Log Data Analysis for Performance Measurement in Warehousing Systems

Master Thesis
of

Katharina Landl, BSc

supervised by

O.Univ.-Prof. Dipl.-Ing. Dr.techn. Paul O'Leary

Chair of Automation
University of Leoben
Austria

Affidavit

I declare in lieu of oath that I wrote this thesis and performed the associated research myself, using only literature cited in this volume.

Eidesstattliche Erklärung

Ich erkläre hiermit eidesstattlich, dass ich diese Arbeit selbstsändig verfasst, andere als die angegebenen Quellen nicht benutzt und mich auch sonst keiner unerlaubten Hilfsmittel bedient habe.

Leoben, am 10.10.2017 Katharina Landl, BSc

Acknowledgements

I would like to thank David for giving me the opportunity to write my master thesis at ARIDA Technology, for his constant support and the freedom to implement my ideas and the trust in them. Many thanks also go to the team at ARIDA Technology that supported me whenever they could.

I would like to express my gratitude to my supervisor Prof. Paul O'Leary for his great guidance, motivation, knowledge and patience during the completion of this thesis and the associated work. I appreciate your determination to push and encourage me to deliver the best work possible. Many thanks also to the team of the Chair of Automation, especially to Petra for her organizational support.

I could not thank my family and friends enough for supporting and encouraging me throughout my studies. Special thanks go to my parents Ulrike and Thomas, my siblings Magdalena and Sebastian, my grandmothers and my godmothers Gabi and Eva.

Thanks to you, Hans, for your love and support – you are the best.

Many thanks to all my friends in Leoben who celebrated, studied, discussed, played, danced and so much more with me throughout the years. You have made the time here truly memorable and fun.

Abstract

This thesis investigates methods for the analysis of log files from warehousing systems with the aim of evaluating performance. For the practical problem at hand, the data originate from different warehousing systems where human-machine interaction plays a major role. More precisely the log files are created by warehouse control systems in response to human and machine activity. The aim of this thesis is to determine the temporal activity of humans in the warehouses by analysing the logs for the purpose of productivity measurement.

To accomplish this, three different approaches have been developed: One approach incorporates additional a-priori knowledge in the form of expert knowledge; the second assumes that there is a characteristic temporal distance between two logs for the same activity; the third approach uses entropy as a measure of information content, in this special case information content as a function of time – assuming that entropy is increasing during working times and stagnating during pauses. Histograms and thresholding are key concepts which have been used in each approach to identify the working time.

The three approaches have been evaluated using ten days of log data from two different warehouses with different degrees of automation. For the warehousing system with less automation the average daily deviation between the allocated time and the time calculated by the approaches varies between 9.4 minutes and almost 3.5 hours depending on the approach. For the warehousing system with a higher degree of automation this figure lies between nine and almost 13 hours. For significant improvement the process of log design would have to be improved.

The approaches have been assessed using an objective assessment scheme. From the evaluation results, possible further improvement of the approaches has been proposed, leading to recommendations for a version that could be implemented in practice.

Index Terms

log files, working time, thresholding, data analytics, warehouse, entropy

Kurzfassung

Diese Masterarbeit untersucht Methoden zur Analyse von Log-Dateien aus Lagersystemen mit dem Ziel der Leistungsermittlung. In dem vorliegenden, praktischen Problem stammen die Daten aus unterschiedlichen Lagersystemen in welchen Mensch-Maschine-Interaktionen eine Rolle spielen; genauer gesagt, werden die Log-Dateien von "Warehouse Control Systems" in Reaktion auf menschliche und maschinelle Aktivität erstellt. Das Ziel dieser Arbeit ist die Ermittlung der temporären menschlichen Aktivität im Lager zum Zweck der Produktivitätsbestimmung durch die Analyse der Logs.

Um das zu erreichen, wurden drei unterschiedliche Ansätze entwickelt: Ein Ansatz verwendet zusätzliche a-priori-Information; der zweite nimmt an, dass es eine charakteristische zeitliche Distanz zwischen zwei Logs für die gleiche Aktivität gibt; der dritte Ansatz verwendet Entropie als Maß für den Informationsgehalt – in diesem Fall Informationsgehalt als Funktion der Zeit – in der Annahme, dass die Entropie während der Arbeitszeit ansteigt und während Pausen stagniert. Histogramme und Schwellwertberechnung sind Schlüsselkonzepte, die in jedem Ansatz zur Anwendung kommen, um die Arbeitszeit zu bestimmen.

Die drei Ansätze wurden unter Verwendung von Log-Daten von zehn Tagen aus zwei unterschiedlichen Lagersystemen mit unterschiedlichem Automatisierungsgrad beurteilt. Für das Lager mit geringerem Automatisierungsgrad schwankt die durchschnittliche tägliche Abweichung zwischen der Soll-Arbeitszeit und den Zeiten, die von den Ansätzen berechnet wurden, zwischen 9,4 Minuten und fast 3,5 Stunden je nach Ansatz. In dem Lagersystem mit höherem Automatisierungsgrad liegt dieser Wert zwischen neun und fast 13 Stunden. Für eine signifikante Verbesserung müsste der Prozess der Log-Gestaltung verbessert werden.

Die Zugänge wurden unter Verwendung eines objektiven Bewertungsschemas bewertet. Abgeleitet von den Bewertungsergebnissen wurden mögliche weitere Verbesserungen der Methoden aufgezeigt, was zur Empfehlung einer Version für die praktische Implementierung führte.

Schlagwörter

Log-Dateien, Arbeitszeit, Schwellwertberechnung, Datenanalyse, Lagerhaltung, Entropie

Contents

Chapter 1

Introduction

The aim of this master thesis is to investigate methods which enable the extraction of temporal activity of humans in warehouses from log files from warehousing systems.

Behind most business processes there are software systems which document and control them. This also applies to warehousing systems. The primary software activities are documented in log files; consequently large volumes of log data are produced. These are often only stored and not used for various reasons, although potential insights on how the processes work could be gained by analysing them. The motivation for this thesis is to investigate the possibility of exploiting the possible gain which can be made by analysing the data.

The working time, which shall be calculated from the log files, is used for productivity calculations. Productivity is one of the key performance indicators (KPI) in any business process. It is important to calculate it correctly, to make it comparable and evaluate possible potential for improvement. To achieve this, all components of productivity have to be determined exactly.

Logistics performance contributes significantly to the overall success of a business. Warehousing is a means to reach logistics goals and so supports corporate objectives. The performance of a warehouse can be measured by its productivity[1] which includes the working time of human workers in the warehouse. So by determining the working time well a contribution to the achievement of the corporate goals is made.

In the course of this thesis approaches are presented and evaluated which analyse log files from the warehouse control systems (WCS) of different KNAPP warehouses. These files have not yet been analysed for a similar purpose, so there is no experience in handling and

[1] Although warehouses do not produce in the classical sense, in warehousing terminology, *efficiency* in a warehouse is still called *productivity*.

analysing these available files.

The work for this thesis has been conducted for *ARIDA Technology* – a software development company. ARIDA Technology's largest client is KNAPP – a company that offers complete intralogistics solutions. All data used for development and evaluation of the approaches come from KNAPP warehouses.

1.1 Problem Statement

Currently warehouse productivity is calculated with allocated times rather than actual working times. This distorts the productivity ratio and makes it incomparable to other plants and to the ratio from the same plant at other points in time. So rather than using a working time, which has been set as "target value", the actual working time should be determined. In this thesis approaches are developed which use log files from the warehouse control system (WCS) for data analysis and try to find the working times from the events recorded as logs in these files. At the moment log files are not analysed automatically. Occasionally text searches are conducted to help gather information about particular incidents in the system.

The outcome of this thesis should be one or more prototype(s) of a programme that uses log files from a WCS as input and delivers working times as output. The working times should correspond more precisely to the time actually worked by humans in the warehousing system than the allocated times used for productivity calculation at the moment.

Log messages texts differ from system to system as they can be set rather freely by the software developers of the respective warehousing systems – this solution should be applicable to all different systems implemented by KNAPP, regardless of their components or developers.

1.2 Thesis Overview

Several fields have influence on this thesis. They are presented in the following chapters. As the log files originate from warehousing systems, some aspects of warehousing are elaborated in Chapter 2. The subsequent chapter deals with data mining and process mining. As the data analysed are log files Chapter 4 is dedicated to this topic. In this chapter also the specific log files used for this work are defined. In Chapter 6 three different approaches to solve the problem are presented. The subsequent chapter describes the evaluation of these approaches and compares the applicability to solve the problem. Finally a conclusion about the results is given, followed by an outlook into future use of this work.

Chapter 2

Warehousing

The VDI 2411 guideline [31] defines warehousing as all planned sitting of goods, interrupting the material flow. A warehouse is an area or space used to store goods which are recorded on a value basis or quantitatively.

In [34] Zsifkovits defines the main purpose of warehousing as balancing the inflow and outflow of goods by bridging temporal and quantitative differences of these flows.

Also these other purposes of a warehouse are listed:

1. Bundling: By keeping stock in a warehouse economies of scale can be exploited in transport, production and purchasing.
2. Safety: Keeping safety stocks protects a company from shortages.
3. Assorting: In warehouses goods are assorted according to customer orders.
4. Speculation: Price fluctuations can be exploited for the benefit of the company by building up stocks at the right time.
5. Refinement: While storing goods in a warehouse refining processes can take place; e.g. maturing, fermentation or refrigeration.

To accomplish all these purposes many different processes take place at warehouses. These processes are summarized into so-called functions of warehouses. In [27] the following warehouse functions are identified and described:

1. Goods acceptance and receipt: This contains all steps from the determination of the delivery date, the acceptance of the goods when they arrive at the premises, the in-

spection of the incoming goods, representation of the delivery in the software system, to the formation of storage units that fit into the warehouse.

2. Storage: This process consists of the selection of the storage place and the transport of the storage unit to this place.

3. Retrieval/picking: Goods are retrieved from their storage space. Different strategies can be applied to optimize the material flow. Besides the physical flow of the goods, also the correction of the inventory level is an important task.

4. Consolidation: The consolidation point is a central point in the material flow in the warehouse where all handled goods are identified and order statuses are updated.

5. Order-picking: During the order picking process the customer order is composed by picking the demanded amount of the requested good. A main challenge in this process is how to bring together the picker and the goods to be picked.

6. Packaging: The goods picked according to the customer order are checked and packed for future transportation. Here also the dimensions of the package shall be optimized. This is often supported by computational optimization tools.

7. Shipping: This function focusses on the final preparation of shipping units for shipment. Additionally – if not yet decided – the optimal method and mode of shipment have to be determined considering various factors.

In the following it is described how the goods go through all these processes:

After goods acceptance and goods receipt the goods are either passed on to a reserve warehouse or an order-picking warehouse. An order-picking warehouse can also receive replenishments from an reserve warehouse. From the order-picking warehouse goods are picked and consolidated before they are moved on to the packaging area – or are moved directly to the packaging area if no consolidation is required. Then the packages are conveyed to the shipping area from where they leave the warehouse. If complete storage units are requested for shipping they may also be taken directly form the reserve warehouse.

2.1 Productivity Measurement

As a large variety of data occur in warehousing systems, it is difficult to assess and optimize a warehouse based on these. However, it is still a necessity to use this data and assess, optimize

and compare warehouses on this basis. Although warehouses do not produce in the classical sense, in warehousing terminology, *efficiency* in a warehouse is still called *productivity*. Productivity was selected because, according to [27] productivity belongs to the elementary basis data of a warehouse, and is also a logistics key performance indicator (KPI). KPIs are used to detect anomalies and to monitor whether or not objectives are met. If KPIs are based on approximated or averaged values, – which is often the case in practice – KPIs are not a good representation of the actual situation. There is also suggested that KPIs are not a good representation of the actual situation as they are often based on approximated or averaged values. It is the main goal of this thesis to improve the accuracy of the productivity measurement of warehouses by calculating the components of this KPI more exactly. In [17] productivity p is defined as follows:

$$p = \frac{s_o}{s_i} \tag{2.1}$$

s_o stands for the output of the system and s_i represents the input to the system. In this context output refers to the performance or revenue accomplished, like the number of items produced or the value of the goods produced. Input could be working hours, materials used or capital employed. Depending on the input different types of productivity are differentiated. The productivity used in the context of this thesis is labour productivity and is defined in the following way: The output of the warehousing system are the *handled orders* and the input is the *working time*.

The working time is determined by the approaches presented in Chapter 6.

2.2 Warehousing Software Systems

This section has been adapted from [27]. Warehousing activities are supported by various software systems. Each of the systems has different tasks and interfaces to other systems. The Enterprise Resource Planning system (ERP system) is a company-wide system with comprehensive functionalities. Concerning warehousing ERP systems monitor stock and stock flows. The Warehouse Management System (WMS) conducts the warehousing processes and optimization. The Warehouse Control System (WCS) controls source-sink-relations and is responsible for the handling of individual orders and processes. The WCS communicates with the Programmable Logic Controllers (PLC) of the automated hardware parts of the warehouse - it acts as the interface between the WMS and the automated hardware of the warehouse controlled by PLCs. The hierarchical context of the systems is

illustrated in Figure 2.1.

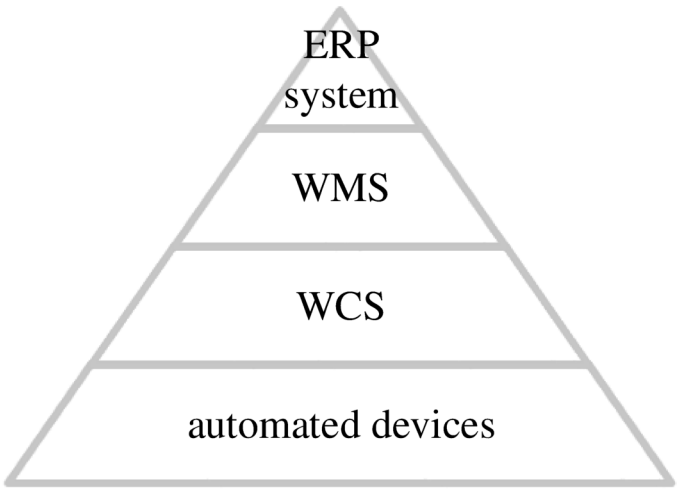

Figure 2.1: Hierarchy of software in a warehouse, according to David Pietzka. The hierarchical order of software in a warehouse from top to bottom is: Enterprise resource planning system, warehouse management system, warehouse control system, automated devices.

Chapter 3

Data Mining

Data mining is a multidisciplinary field, therefore a universal definition of the term does not exist. Han et al. describe data mining as the whole process of turning large amounts of data into knowledge [16]. Knowledge Discovery in Databases (KDD) is defined by Fayyad et al. [9] as *the overall process of discovering useful knowledge from data*. This process consists of the following nine steps: Learning the application domain, creating a target dataset, data cleaning and preprocessing, data reduction and projection, choosing the function of data mining, choosing the data mining algorithm(s), data mining, interpretation, using discovered knowledge. In this process data mining is one step of many to extract knowledge from data. In contrast, according to Embrechts et al. data mining is *the automated extraction of novel and interesting information from large data sets* [5]. There is no consent about the output of data mining. Also none of the above mentioned works defines what is considered information and knowledge and what the differences might be. Embrechts et al. suggested a hierarchy according to the data mining wisdom pyramid depicted in Figure 3.1, but fail to define the terms used. In [8] Fayyad et al. even use the terms information and knowledge as synonyms. The data-information-knowledge-(understanding-)wisdom hierarchy has been discussed extensively in other works, for example in [2], [11] and [23], however without the context of data mining.

Figure 3.1: Data mining wisdom pyramid according to Embrechts [5]

What output should be mined from a data set, depends on what is useful for the specific application. Several objectives, also called tasks, of data mining were identified. The most common *functionalities* of data mining are listed below:

1. Characterization and discrimination [16]: When characterizing data, the main features of a class of data are described by examining this data set. In contrast, in data discrimination the characteristics of a class of data are defined by comparison to another data set.

2. Mining frequent patterns [16]: The identification of patterns or relationships that occur often in a given data set is called frequent pattern mining.

3. Classification [16]: The process of extracting models describing predefined data classes. The models are used to classify new data for which the class label is unkown. According to [6], typically the data mining algorithm is trained on a dataset in order to learn which features distinguish one class from another in order to be able to assign a class to an unknown dataset.

4. Clustering [25]: Cluster analysis categorizes data into groups according to their similarity . According to [6], the aim of clustering is to find homogeneous natural groups that are as distinct as possible from other groups. In contrast to classification, the clusters are not known in advance.

5. Outlier analysis [16]: Outliers may be data elements represented by the model assumed for the data, but lie outside a specific confidence interval; so these values have a low probability to occur and are thus considered outliers. Outliers may also be effects of extraneous events and thus not represented by the assumed model.

The data that is used for data mining can be categorized in many ways; one of them being the following: In [25] Shanhanawaz et al. classify data according to their dependency on time:

1. Static data: This type of data has no temporal reference.
2. Sequences: This is data without explicit reference to time, but with temporal relationship between data items.
3. Time stamped: Data with explicit reference to time.
4. Time series: Time series data is defined as the special form of time stamped data in which the time distance between two items is equal.[1]
5. Fully temporal: This category includes fully time dependent data.

Additionally data streams were identified as a class of data:

6. Data streams [15]: They are characterized by their potentially infinite volumes, temporal order and fast change. Stream data flow in and out of a computer system continuously and due to the volume generally cannot be stored completely or scanned more than once. When data streams occur there are usually many parallel streams that need to be processed which adds to the challenges of handling such huge amounts of data.

The data used in this thesis is time stamped data. However, to be able to apply some useful methods, it was turned into time series data in some of the approaches in Chapter 6.

According to Laxman and Sastry [18] a main feature that distinguishes temporal data mining is the size and nature of the data sets and the origin of the data. Temporal data mining is typically dealing with large datasets which are symbolic or nominal-valued. They also point out that data is often gathered not for the purpose of data mining. Therefore often the data miner has no control over how the data is collected which can lead to challenges for the realisation of the data mining goals as some data might not be gathered in an optimal way for the specific application.

[1] According to [3] time series are commonly, but not necessarily, spaced equally.

3.1 Cross-Industry Standard Process for Data Mining

The Cross-Industry Standard Process for Data Mining (CRISP-DM) is a process model for data mining projects that gives an overview of the stages of a data mining project; it is documented in [4]. The model is flexible enough for being used as a general guide trough any data mining project. The model consists of the following six phases that can be repeated any number of times.

1. During the **Business Understanding** phase the business background and requirements of the data mining project are documented. Taking this as a starting point data mining goals are determined and a rough project plan is designed.

2. The **Data Understanding** phase start with initial data collection. By documenting data features and quality and performing an initial exploratory analysis the analyst can get to know the data better.

3. **Data Preparation** covers all tasks to produce the final dataset used for data mining. This includes selection, cleaning, construction, merging and formatting of data from the initial raw dataset.

4. During the **Modelling** phase modelling techniques are selected and models are built and assessed.

5. The model gained and the process how it was produced is assessed from a business point of view during the **Evaluation** phase. Finally it is decided how the results of the procedure are going to be used.

6. The **Deployment** phase consists of documenting the project experiences and outcomes and planning the integration into the organisation.

The relations between the phases are illustrated by Figure 3.2. It shows that there are many relations among the phases of the CRISP-DM and thus that the model as such is flexible as the phases can be passed in a variable order depending on the nature of the project and the outcomes of the previous phases.

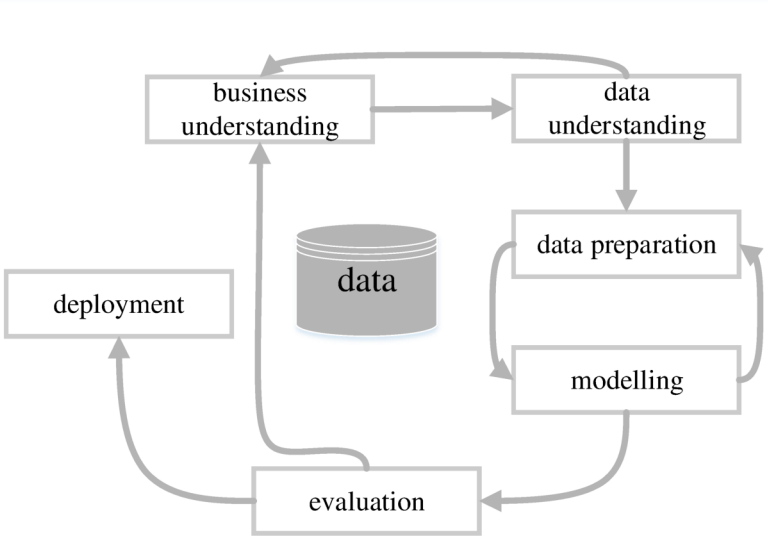

Figure 3.2: CRISP-DM process model [4]

This model has been used as a guideline through the work for this thesis. The CRISP-DM has been used as guideline for the work carried out for this thesis. For this data mining project this model has been chosen, because – unlike other models used like SEMMA (Sample, Explore, Modify, Model, Assess) described in [1] and Knowledge Discovery in Databases (KDD) described in [9] – it acknowledges the phase of business understanding as a vital part of data mining projects, as stated in [1]. However, this thesis only covers a part of the cycle, as it does not deal with the deployment of the developed approaches. Business understanding was described briefly in Chapter 1; the data used will be described in Chapter 4; the approaches used will be described in Chapter 6, followed by an evaluation of the approaches.

3.2 Process Mining

This section is based on the Process Mining Manifesto [29] which has been published by the IEEE Task Force on Process Mining and [30], a book which Aalst has dedicated to this topic. At the end of this section the topic's relevance for this work is discussed. The content of the rest of the section are taken from the above mentioned sources, if not stated otherwise.

Process mining extracts knowledge from event log data to discover, monitor and improve processes. The discipline unites the fields of computational intelligence and date mining with process modelling and analysis. It sees itself as *"missing link between data mining and Business Process Management (BPM)"*.

The data used for process mining are event logs. These event logs are described to have a special structure. Each event has to be linked to an activity and a case - a particular series of events. Typically events could also refer to a timestamp, a resource and cost.

In [30] two different types of processes are disinguished: lasagna processes and spaghetti processes. *Lasagna processes* are relatively easy to analyse, because they are structured and there are only few deviations from the set process model. In contrast *spaghetti processes* are rather difficult to analyse due to a high variety of activities which can not be represented by a process model of reasonable size easily. Thus not all process mining techniques can be applied to spaghetti processes.

There are three different types of process mining which are illustrated in Figure 3.3.

1. Discovery: Here a process model is produced from an event log. A method presented in detail by Aalst for this type of process mining is the α-algorithm. For this algorithm to work it is assumed that the input logs are complete. Still it has some shortcomings; e.g. different models can result from the same input logs.[2] In [30] also an overview of more advanced algorithms which handle noise and incompleteness well is given; these are: heuristic mining, fuzzy mining and genetic process mining.

2. Conformance: During conformance checking, deviations between a recorded event log and a given process model are identified – the so-called *fitness* of the model is determined. The fitness is defined as *the proportion of behaviour in the event log possible according to the model*.

3. Enhancement: This type of process mining aims at altering an existing process model using an event log. Examples for such process enhancement activities are *repair* - which corrects deviations between the specified process model and the real process - and *extension* - which aims at extending the functionality of the existing process model. A repairing activity could be removing unused or infrequently used paths form the model, after discovering this in conformance checking. Extension could refer to mining additional *perspectives* of e.g. organizational structures, the usage of resources or the case perspective (for which decision tree learning could be used).

[2]This lies in the nature of the problem as process discovery can be classified as an inverse problem and as stated in [14] inverse problems do not have unique solutions.

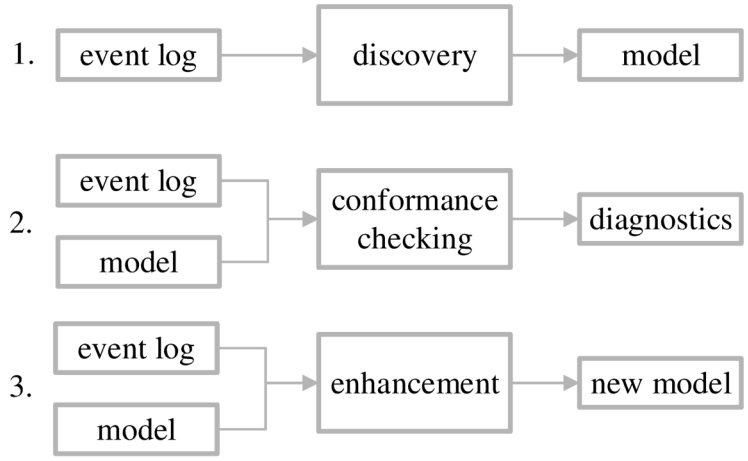

Figure 3.3: Types of process mining: 1. discovery, 2. conformance, 3. enhancement

In [30] most process models are represented as petri nets as this is a simple representation but still allows for concurrency, iterations and a choice of paths. Also other representation techniques are mentioned: workflow nets (which are a subclass of petri nets), YAWL (Yet Another Workflow Language), BPMN (Business Process Modeling Notation), EPC (Event-driven Process Chains) and causal nets.

Aalst describes the data mining process models explained and mentioned in Section 3.1 as *providing little support* and *not tailored toward process mining projects.* So, a process model specifically for process mining is presented. The *L* life-cycle* model consists of five stages:

0. Plan and justify: As a start the process mining project has to be planned and the expectations on the project be defined.

1. Extract: After the start of the project, event log data, process models, process mining objectives and questions which should be answered by the project have to be extracted. Sources for these are the IT-systems, domain experts and managers.

2. Create control-flow model and connect event log: Next the control-flow of the process to be analysed has to be determined, using process discovery techniques. Also existing process models can be used. It is important that the specified model and the event log correlate with each other. This means that the activities in the log are reflected in the model.

3. Create integrated process model: In this stage the control-flow model is extended by additional perspectives, e.g. an organizational perspective or the case-perspective with

a focus on the properties of the individual cases. This model can be used for simulation and can help to gain insight into the actual process.

4. Operational support: Implementing an operational support system is the most challenging form of process mining. The system offers automated feedback and analysis to the participants in the process without the intervention of a process mining analyst.

The event logs as defined by Aalst are described to have a special structure. Each event has to be linked to an activity and a case - a particular series of events. *This is – in the opinion of the author of this thesis – a strong restriction.*[3] Thus not all process mining techniques presented can be used on many log data. The data used for the work of this thesis is described in Chapter 4. It does not have the properties described above. Thus the methods presented in [30] cannot be used for the work of this thesis as the log files that have to be used for the analysis do not have the required structure.

[3]This text has been italicised to emphasise that is the expression of a personal opinion.

Chapter 4

Available Log Files

Log files [28] are text files created automatically by computer programmes. Typically the file consists of multiple *records* or *logs*. The content of log files can vary widely. Usually each of the records contains a timestamp. Additionally variable names and values, delimiters, keywords and comments may be included. Log files are used for software debugging especially during the testing phase and for system monitoring after the software has been deployed.

Log Files from Warehouse Control Systems (WCS)

The log files used during this thesis originate from a WCS of a KNAPP system. There were distinct sets of files available during the development of the approaches in Chapter 6 and the evaluation of these approaches. For each of these two phases files of at least two different plants covering various days were made available.

One log file consists of many lines; the electronic logs are ASCII files with carriage returns as end of line markers. Each line is one log. The structure of these logs is described below in words and using the Backus-Naur Form. Each file contains all logs made in one hour and is produced automatically by the WCS. The 24 files that are produced during one calendar day, are packed into folders (together with some overhead that is irrelevant for this work), compressed and stored at a server. How this exactly works and how the log files can be retrieved from this location is beyond the scope of this thesis.

Each folder name consists of the date of the logs in the following format: logYYMMDD.gz.tar The file names include the date and time during which the logs have been written in the following format: YYMMDD.hh00.gz

The total size of the daily folders varies - depending on the activity of the system - between

approximately 600 KB and 500 times that size. The sizes of the individual log files vary between 24 KB and more than 20 MB. The more active the system was, the more logs it contains and the larger is the size of the log files. This is an example log:

LOG 20161005 00:00:00.098 WMS_ALERTWATCH/WCSIMP/
IMP_ALERTRECEIVER [WMS_ALERTWATCH]
Code[MSG_DB_WAIT_ALERT] - ALERT_NAME = WMS2WCS -
IMP_EVENT.ID = 12744284

In the following the structure of the logs is defined using the Backus-Naur Form:

<log> ::= <prefix> “ ” <timestamp> “ ” <log-message>

<prefix> ::= “LOG” | “ERR” | “DBG”

<timestamp> ::= <date> “ ” <time>

<date> ::= <year> <month> <day>
<year> ::= <digit> <digit> <digit> <digit>
<digit> ::= “0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9”
<month> ::= “01” | ... | “12”
<day> ::= “01” | ... | “31”

<time> ::= <hours> “:” <minutes> “:” <seconds> “.” <milliseconds>
<hours> ::= “00” | ... | “23”
<minutes> ::= “00” | ... | “59”
<seconds> ::= “00” | ... | “59”
<milliseconds> ::= <digit> <digit> <digit>

<log-message> ::= <event-handler> “/” <package> “/” <procedure> “ [” <event> “] ”
<residual>

<event-handler> ::= “WMS_ALERTWATCH” | ...
<package> ::= “WCSIMP” | ...
<procedure> ::= “IMP_ALERTRECEIVER” | ...
<event> ::= “WMS_ALERTWATCH” | ...
<residual> ::= “Code[MSG_DB_WAIT_ALERT] - ALERT_NAME = WMS2WCS -
IMP_EVENT.ID = 12744284” | ...

For the parts of the log text not all possible content is known, thus only an example could be mentioned.

In the following the structure of the logs is described in text. Each log consists of three main parts:

1. three characters which categorize the log
2. a timestamp structured as follows. YYYYMMDD hh:mm:ss.sss
3. the log-text – a description of the event including event data (the structure depends on the type of event)

The log-text does also have a distinct structure: It starts with the name of the event handler, then the package name and the name of the procedure. These three parts are separated by "/". After one space the event name follows in squared brackets. Subsequently additional descriptions about the event follow. For all types of systems considered in this thesis this structure is the same. However, the content of the of the log text can vary from system to system. This is because the systems are created individually for the customers and there is no guideline on what should be logged or how log texts should be formulated. Moreover there is no description available on how to interpret the log texts of each system.

This presents one of the biggest challenges in this thesis, as it is impossible to analyse the contents of the individual logs because the largely depend on what the the programmers of the respective systems consider important.

In Chapter 6 approaches are presented which meet this challenge in different ways by the means of the methods explained in the following chapter.

Chapter 5

Applied Methods

Here the methods which have been used in the different approaches presented in Chapter 6 are introduced. Histograms and thresholding were applied in each of the approaches. The Shannon entropy and Scale Space theory were used in the approach in Section 6.3.

5.1 Histograms

A histogram [7][15] is a data structure that shows the frequency of element values in a data set. The complete range of values in the data is divided into a set of contiguous subranges, called *bins*. The value range of one bin is called *width*. Typically, but not necessarily, the bins have equal widths. The depth of each bin is the number of values in the value range of the bin. The frequency of values in the value range is called *count* and is represented by the depth of the bin. An issue with using histograms is the possibly arbitrary choice of the number and the width of the bins . As listed in [24] several rules for the calculation of the number of bins were suggested. Many of these methods require some knowledge about the underlying distribution of the data which was not available for the data used to solve the problem of this thesis. Depending on the output during the development of the approaches, the auto-binning of Matlab and the square root rule which is the standard method used in MS Excel were applied:

$$k = \sqrt{n} \tag{5.1}$$

Here the number of bins k is the square root of the number of data points n.

Histograms have been applied to the analysis of log-files, see for example [10]. However, in that case the aim was to detect anomalies by representing the log messages in histograms and thus making it easier for the human observer to find unusual events.

During the work for this thesis only one-dimensional histograms were used.

5.2 Thresholding

Thresholding [13] is a method that is commonly used to segment data. In particular in digital image processing a lot of research has been dedicated to this theme. There a threshold is calculated for image segmentation - to find objects in grayscale images and segment them from the image background. For this thesis binary thresholding is of particular interest. This is done by calculating a histogram of gray levels of the image and then determine a threshold which best divides two occuring modes of the binary distribution. There are numerous different thresholding algorithms, many of which can also be applied to problems beyond digital image processing. The results of the application of thresholding with respect to the problem at hand are presented in Chapter 6.

5.2.1 Otsu's Method

This thresholding method was published by Otsu in [22] and bases on the gray-level histogram of an image without any other background knowledge. The threshold determined by maximising the separation of the two modes:

$$\sigma_b^2(k) = \frac{(\mu_T \omega(k) - \mu(k))^2}{\omega(k)(1 - \omega(k))} \tag{5.2}$$

with $\omega(k)$ and $\mu(k)$ being the zeroth and first order cumulative moments of the histogram to the kth bin and μ_T being the total mean level of the image.

Otsu's Method is implemented by Matlab in its *graythres()* method.

5.2.2 Triangular Method

This threshold calculation is also based on the grayscale histogram of an image. In [12] this method is defined as follows. The line between the bin with maximum height of the histogram and the bin representing the values of maximum intensity is determined. Then the maximum normal distance between a bin and this line is calculated as indicated in Figure 5.1. The threshold is the centre of the range of this bin.

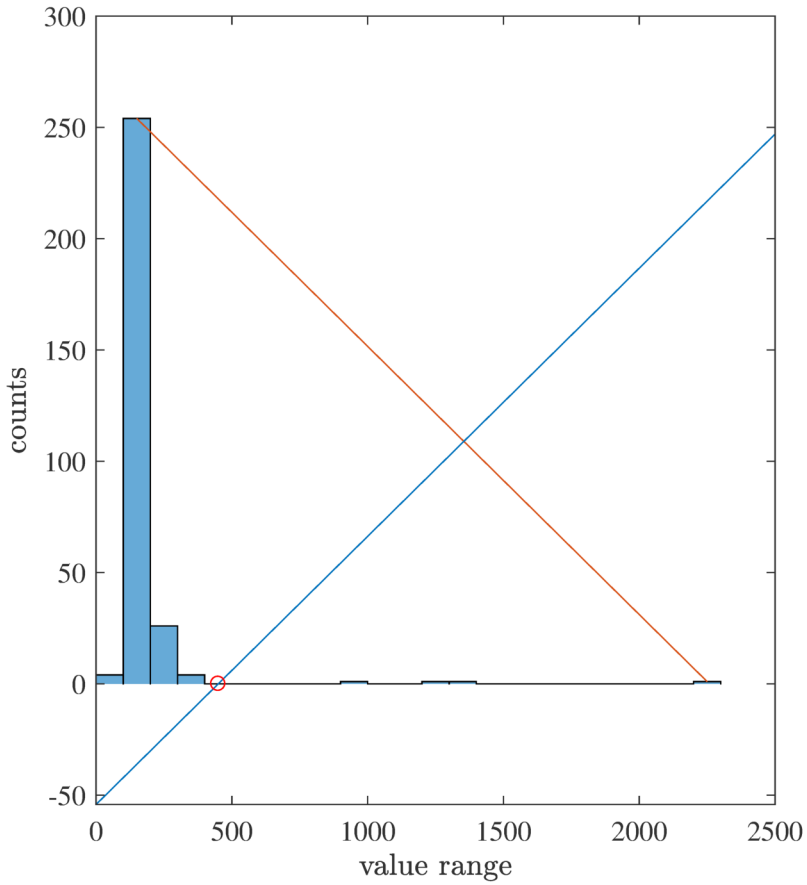

Figure 5.1: Triangular thresholding method: Data is distributed over the whole value range. The orange line connects the bin with maximum height of the histogram and the bin representing the values of maximum intensity. The blue line is the normal line with the maximum distance between a bin and the orange connecting line. The threshold for this distribution is marked with the red circle.

In the work for this thesis this method was applied in the original version as described above and in a slightly modified version. Instead of connecting the bin with maximum height of the histogram to the bin representing the values of maximum intensity is determined, the line is constructed between the peaks of the two modes, i.e. between two local maxima of the distribution. The triangular method is used in Chapter 6 because it also works, if there is only one (local) maximum in the histogram and if there are many, it is easy to calculate and no a-priori knowledge about the distribution is needed. The selection of the two different version described above depends on the actual application.

5.3 Shannon Entropy

The Shannon entropy is a concept in information theory and was first published in [26]. It is a measure for the information content of a message which may be subject to perturbation. Shannon entropy originates from communication theory. It tackles the problem of correctly identifying a message at a destination in the presence of noise. It uses the expectation that the entropy of the sent and received message should be the same.

The Shannon entropy H is defined as follows:

$$H = -\sum_{i=1}^{n} p_i \log p_i \tag{5.3}$$

whereby p_i is the probability of sign i occurring in a message and n is the number of signs in the message, i.e. the length of the message.

H is a maximum when all p_i are equal – this is the situation with most uncertainty. On the other hand H is zero when all p_i are equally zero, except one. This is the situation with complete certainty.

Shannon entropy can be used as an approximate measure for the information content in data as for example shown in [20]. Entropy is also commonly used for attribute selection when building decision trees for classification problems, as stated for example in [33].

Entropy will be used in Section 6.3 to identify working times, assuming that – as entropy measures information content – entropy in the log data is increasing during working times and stagnating during pauses.

5.4 Scale Space Theory

Scale Space theory [19][32] is a theory from digital signal processing which has also been applied to image processing. The idea is that real-world objects in images and real-world events in signals are only meaningful in a certain range of size and extent, also known as scale. So the only possibility to detect them is to represent them at different scales. Consequently, a hierarchical process is implemented whereby fine-scale information should be suppressed from step to step. Finding the "right" scale or an "interesting" scale for representation is difficult. Typically a scale space representation is calculated by a convolution of the original signal with Gaussian kernels followed by decimation.

As the problem dealt with in this thesis is neither feature detection in an image nor signal

processing, the ideas, but not the exact procedure, are used in Chapter 6.

Chapter 6

Developed Approaches

The inputs to these algorithms are text files and the outputs are the working times during the corresponding time periods, whereby each algorithm is an embodiment of a specific approach to computing the required values. The prototypes have been implemented in Matlab version R2016b. Data is analysed on a daily basis. Each approach starts with reading in the text files, the extraction of the timestamps and the extraction of the log messages. Then the approaches differ in how they process the information and finally how the working time is extracted. The approaches are evaluated in Chapter 7.

For an overview of the structure of the log files and the individual logs please refer to Chapter 4. In the following, *timestamps* and *log messages* always refer to <timestamp> and <log-message> as defined in Chapter 4. Any other expressions in angle brackets also refer to the definitions in Chapter 4.

The following two steps are the start of each of the three approaches and will only be explained at this point to avoid repetition in the presentation of the approaches.

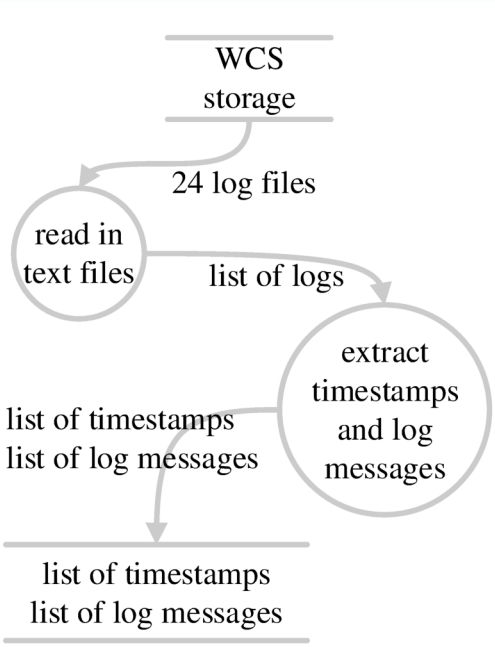

Figure 6.1: Starting procedure of each approach

6.0.1 Read in text files

The electronic logs are read in and evaluated for 24 hours intervals, i.e. on a day-by-day basis.

6.0.2 Extract timestamps and log messages

Each log has a structure as described in Chapter 4. To filter out the overhead of the files, for further processing only lines starting with an "L" (the first character of LOG) are considered. To be able to process the data further to gather more information, the timestamp and the log messages are separated and saved into two lists – one with all timestamps and one with all log messages in the same order.

6.1 Expert Knowledge

This approach starts with the assumption that expert knowledge is available. The expert defines, from experience, which keywords need to be evaluated in which manner to obtain the desired KPI. Knowledge from a domain expert has been employed to find the working times

by analysing the log messages. The expert is associating meaning with specific keywords within the log messages: this is knowledge in the form of *metadata*[1].

The expert has shared, from his experience, the following keywords with their ability to indicate human action in the warehousing system:

1. **DIALOG** – stands for any direct human interaction with the computer system through a user interface; e.g. the feedback after a manual pick.
2. **MOD_WS** – modify working step. This message is triggered when a task , e.g. a pick, is completed and a response is given to the host; in this case the WMS.
3. **MSG_PICKOK** – message pick ok. This log message is recorded after a successful pick in an automated or semi-automated picking system.
4. **DDC_RECEIVE** – display data concentrator receive. This keyword is found when a response is issued from a *pick to light system* to the WCS; e.g. after completing the last pick for an order.

Due to the different components in the warehousing systems, not all keywords are contained in log files from any of the systems. The procedure of finding the working time using the expert knowledge approach is outlined in Figure 6.2.

[1]In [21] metadata is defined as structured information that describes, explains, locates or otherwise makes it easier to retrieve, use, or manage an information resource.

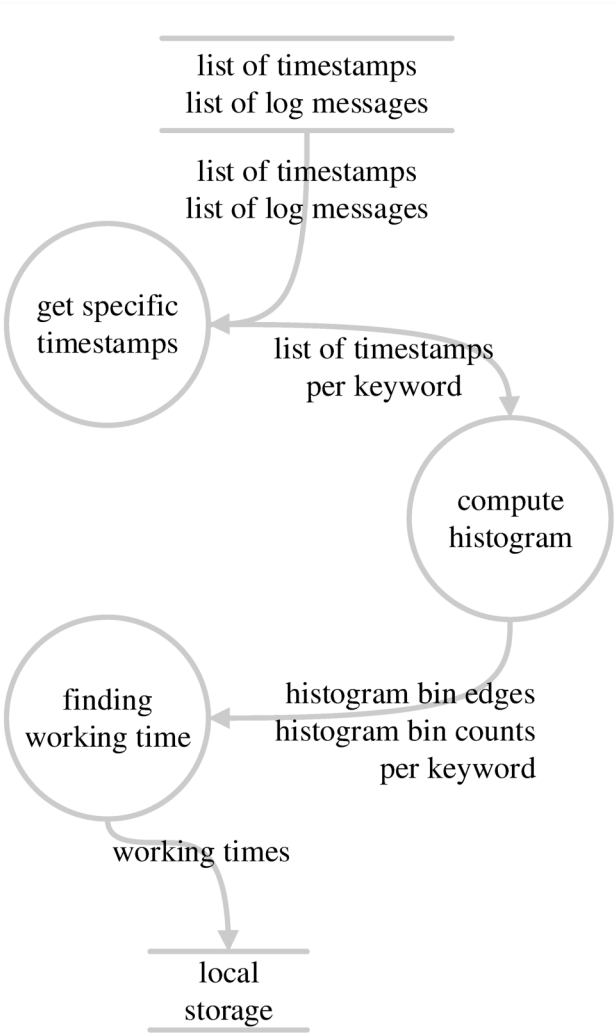

Figure 6.2: Procedure of the expert knowledge approach for finding working times

6.1.1 Get specific timestamps

The timestamps from all messages which contain the four keywords defined earlier in Section 6.1 are extracted. For each of the keywords all log messages containing the keyword are identified and the timestamps associated with the messages are determined. These timestamps are saved into a list for each of the keywords.

6.1.2 Compute histogram

For each of the keywords a minutely histogram is computed from the list of timestamps from the previous step. The number of occurrences of each keyword per minute is determined

and used as the count value for the corresponding *minute bin*. The output of this step are four lists – one for each keyword – which contain the bin counts of the histograms.

6.1.3 Find working time

The assumption is: if in a given minute one of the keywords occurs, then a person is working and thus this minute counts as a "working minute". The minutely bin counts of the four histograms are summed. Then all values > 0 are detected and, using the histogram bin edges, the respective working minutes are determined. Contiguous minutes are summarized to working time blocks as presented in Figure 6.3.

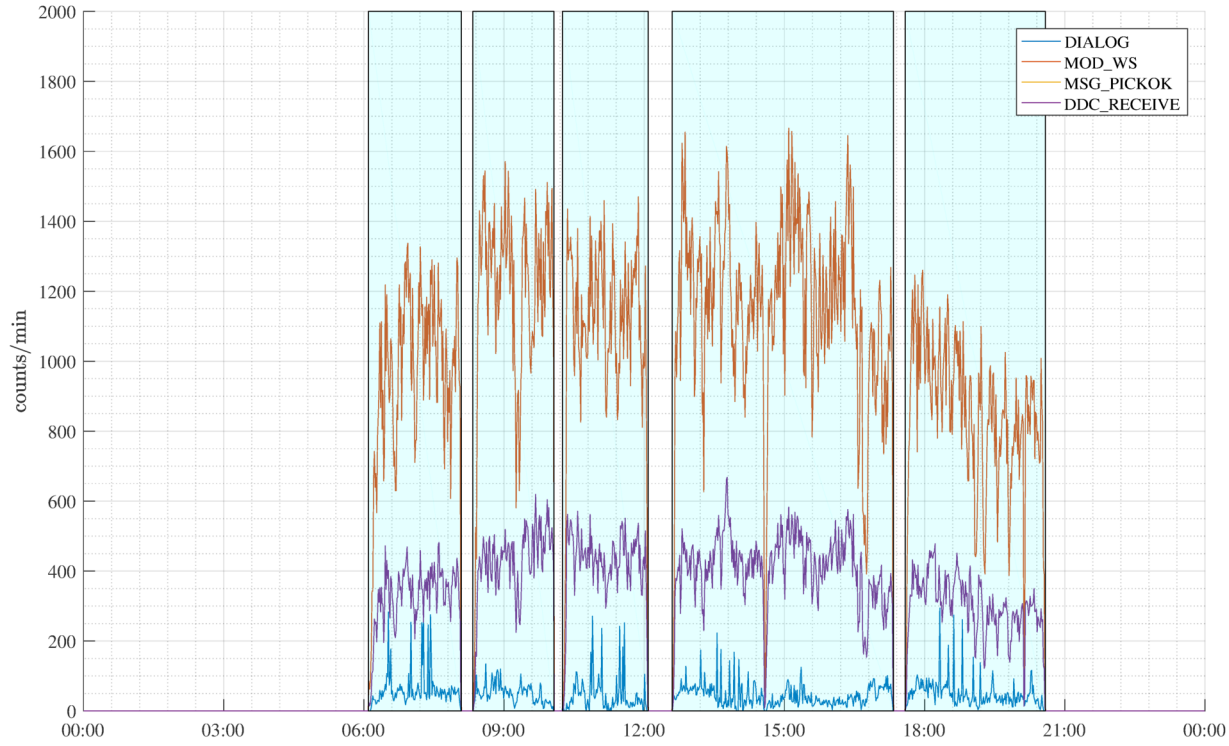

Figure 6.3: Example output for the expert knowledge approach for finding working times – the lines show the number of times each of the keywords occur in logs per minute. Working times are marked in light blue.

This approach will be evaluated and compared to the following approaches in Chapter 7.

6.2 Unique Texts

In contrast to the previous approach, this approach does not use any a-priori knowledge about the log message content. It is assumed that there is a characteristic temporal distance

between two logs for the same activity. Hence an activity can be declared as inactive, if the temporal distance between two logs are longer than the characteristic temporal distance. The procedure for finding the working time using the unique texts approach is depicted in Figure 6.4.

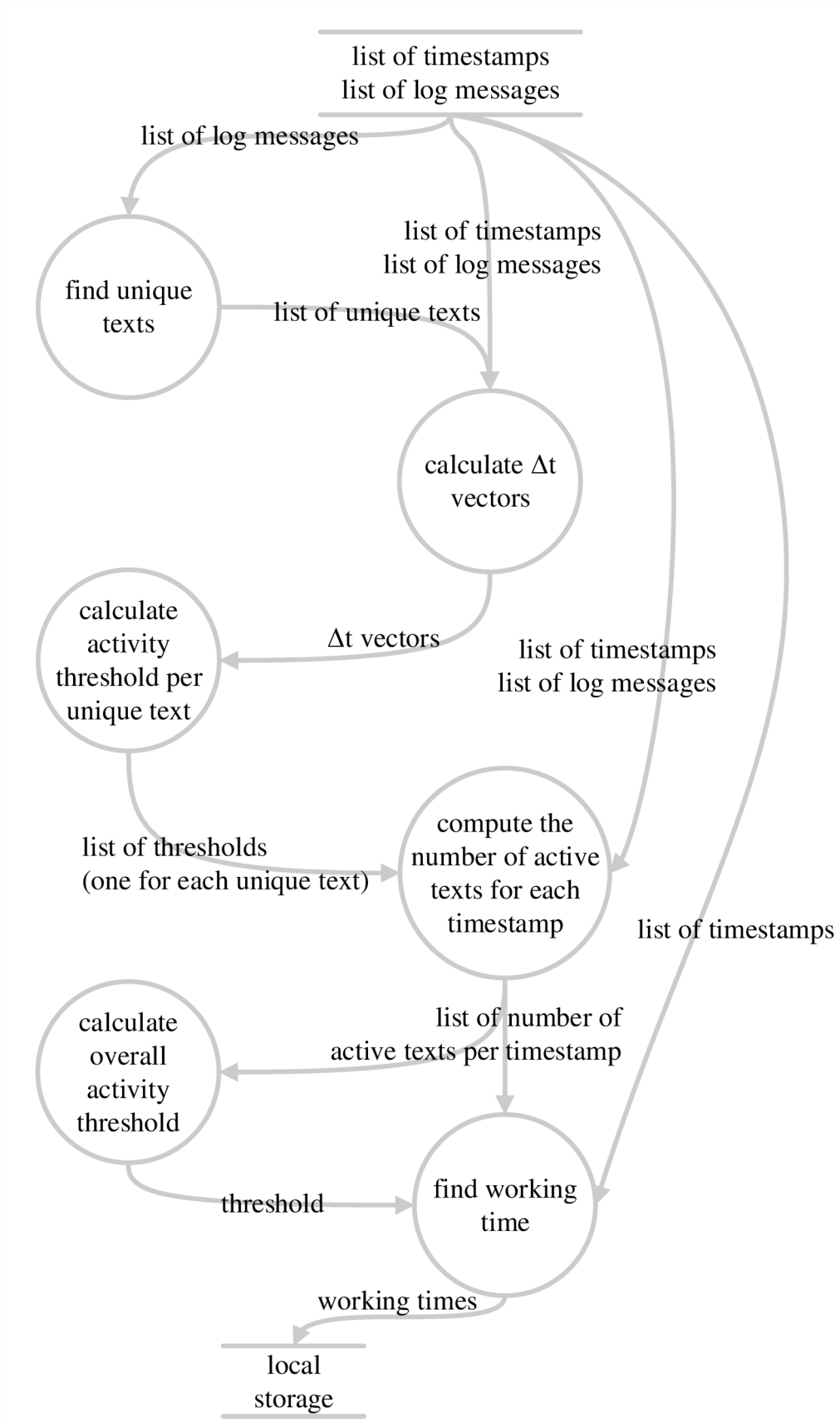

Figure 6.4: Procedure of the unique texts approach for finding working times

6.2.1 Find unique texts

For this method only a part of the log message is used. The first part of the log message containing the <event-handler>, the <package>, the <procedure> and the <event> is extracted – the log message is *trimmed*. It is assumed that these parts of the log message define the type of event that led to this log entry. Thus the rest of the log message is not relevant for future analysis and will not be subject to any further evaluation.

Then a list with all different trimmed log messages is created. These are the *unique texts*.

6.2.2 Calculate Δt vectors

For each of the unique texts, the associated timestamps are extracted and saved in a list. From these lists the Δt vectors are calculated. These vectors contain the temporal distances between the timestamps. For each unique text, there is one Δt vector available for further computations.

6.2.3 Calculate activity thresholds per text

The following procedure has to be carried out for each of the Δt vectors to find a threshold which divides phases of activity and phases of inactivity for each unique text: the threshold is calculated using the triangular thresholding method as described in Section 5.2.2. The input is the Δt vector. In this case it is very important to check, if the Δt vector is mono-modal. This means that the logs occur very regularly – in very similar time distances. If logs with a specific log message occur with regular temporal distances, they cannot help distinguish working time from pauses. Thus they shall be ignored in further analysis. To simplify further computations these thresholds are set to infinity, so no activity will be detected in log messages with these specific texts.

6.2.4 Compute number of active texts for each timestamp

For each of the timestamps associated with a log the number of *active texts* is calculated. This is done by going through all timestamps and keeping a count on how many texts are active – meaning that the time distance to the next log message containing this text is lower than the threshold previously calculated. If the activity of a certain text starts at a timestamp, the count is raised by one and if the activity of a certain text ends at a

timestamp, one is deducted from the activity count. The result of this step is a vector with an activity count for each timestamp.

6.2.5 Calculate overall activity threshold

Now a threshold has to be found which divides phases of activity and phases of inactivity, not for the unique texts, but for the whole system. For this the active texts vector calculated in the previous step is used. This time, a threshold has to be found, even if there is only one local, then global, maximum. So, again the triangular thresholding method from Section 5.2.2 is applied. This time however, if the vector is mono-modal, then the bin with the maximum value is used as the second peak in the thresholding algorithm (see Section 5.2.2), in the case of bi-modal data, the two mode locations are used.

6.2.6 Find working time

After finding an activity threshold, all times with an activity level higher than the threshold have to be found. All groups of contiguous timestamps where the activity level is higher than the calculated threshold are merged to a connected working time fragment.

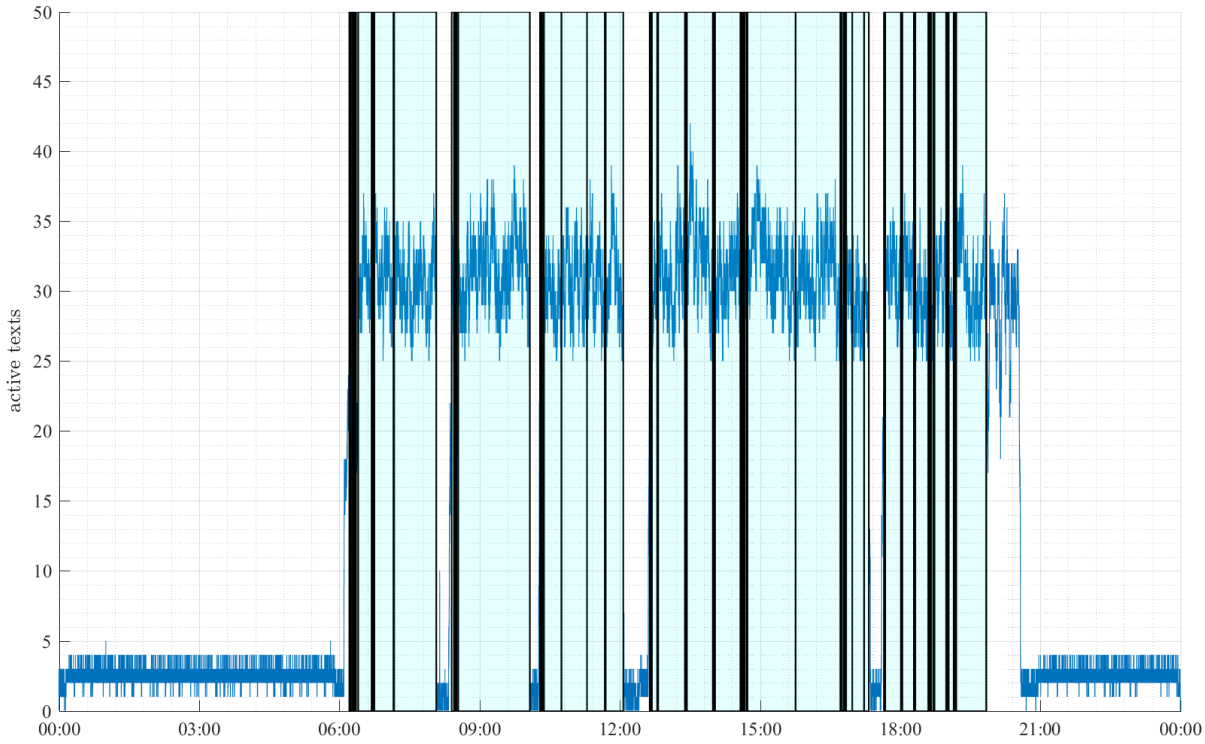

Figure 6.5: Example output for the unique texts approach for finding working times without clustering – the blue line represents the number of active texts. Working times are marked in light blue. Without clustering there are almost 40000 working time segment separated by tiny pauses.

After this first clustering of the neighbouring working time fragments, there are still many working time blocks left – partly separated by only very short pauses: this is depicted in Figure 6.5. To cluster these into larger working time segments, scale space calculation as described in Section 5.4 is applied to the pauses. Pauses are eliminated in *minute steps*. Then the decline of the number of pauses per step is calculated. The desired scale is the scale with the greatest decline between this scale and the next lower scale. The result is the desired working time as shown in Figure 6.6.

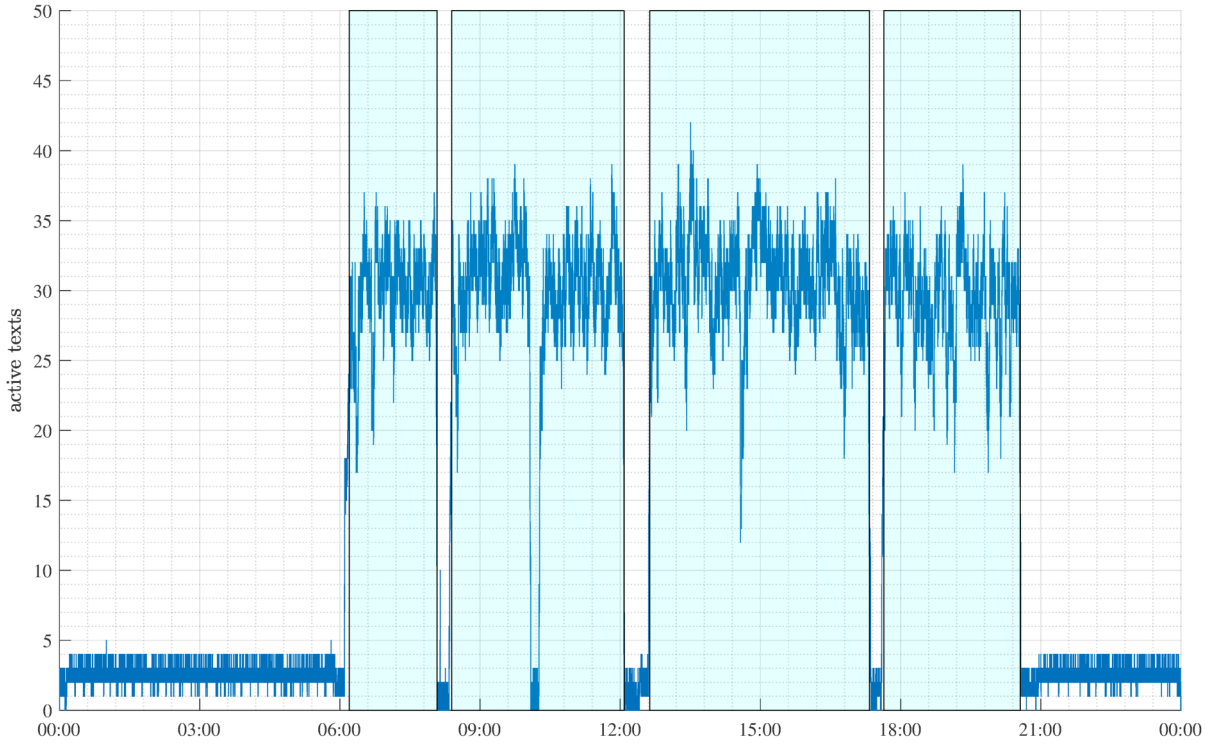

Figure 6.6: Example output for the unique texts approach for finding working times – the blue line represents the number of active texts. Working times are marked in light blue.

6.3 Entropy

In this approach Shannon entropy is used as a measure for information content. The concept of Shannon entropy as defined in Section 5.3 has been applied to calculate the working time, assuming that entropy rises during working times and remains comparably constant during pauses. The procedure for computing the working time applying the concept of entropy is outlined in Figure 6.7.

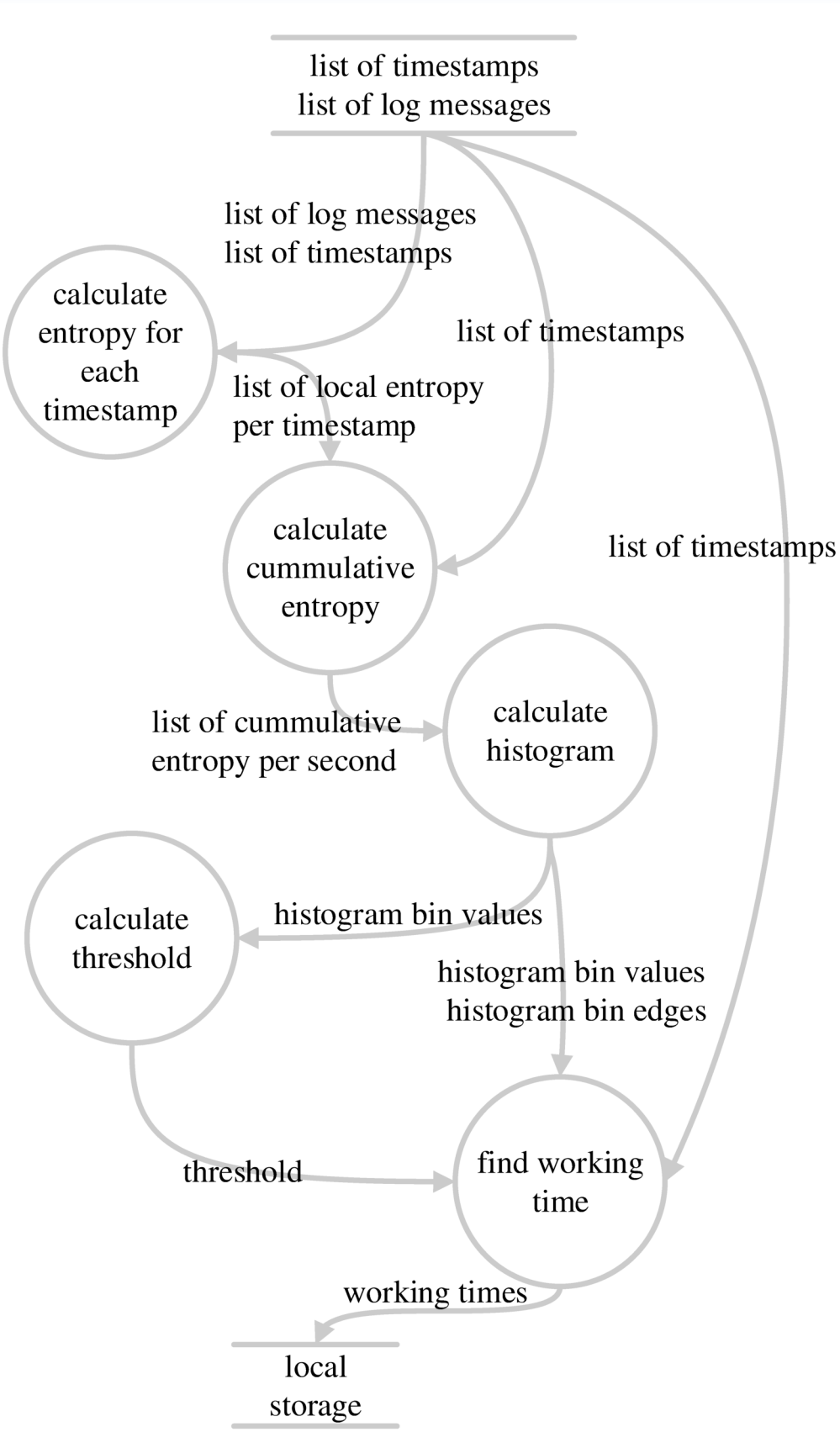

Figure 6.7: Procedure of the entropy approach for finding working times

6.3.1 Calculate entropy for each timestamp

The behaviour of the entropy throughout the day has to be calculated. This is a rather elaborate procedure.

Firstly, as described in Section 6.2.1, only parts of the log messages are used and from these trimmed messages all unique ones have to be found.

From this list of unique texts the probability of a message occurring p_i is calculated for each unique text.

$$p_i = \frac{m_i}{m_t} \tag{6.1}$$

whereby m_i is the number of times this unique text occurs in a log message on this day and m_t is the total number of logs of this day which can be represented as

$$m_t = \sum_{i=1}^{n} m_i \tag{6.2}$$

with n being the number of unique texts found.

In this case p_i is calculated day by day – when implementing this algorithm in a real-world system it might be better to calculate it over various days or with all data available.

Then the local entropy H_l is calculated for each timestamp using the probability of occurrence of the log message.

$$H_l = p_i \log p_i \tag{6.3}$$

These local entropies are summed up secondly, so that for each second of the day all local entropies that are associated with that second are included. This yields the *secondly local entropy*.

A cumulative sum over the secondly local entropy is calculated – the *secondly entropy*. This yields the entropy profiles as depicted in Figure 6.9.

6.3.2 Calculate histogram

From the secondly entropy a histogram over the entropy is calculated – the frequency distribution of the secondly entropy. The number of bins is determined using the square root rule as described in Section 5.1. The entropy histogram for the example in Figure 6.9 is displayed in Figure 6.8.

6.3.3 Calculate Threshold

The histogram calculated previously is used to find the working time during the day. The histogram will have peaks at entropy levels at times with low activity. A threshold has to be calculated to separate activity from inactivity. This is done using the triangular thresholding method as defined in Section 5.2.2. It is applied as in the previous approach described in Section 6.2.5. The calculated threshold is represented by the horizontal orange line in Figure 6.8. This threshold enables the detection of the entropy levels at which pauses are made. The histogram bins with counts above the calculated threshold are declared pauses.

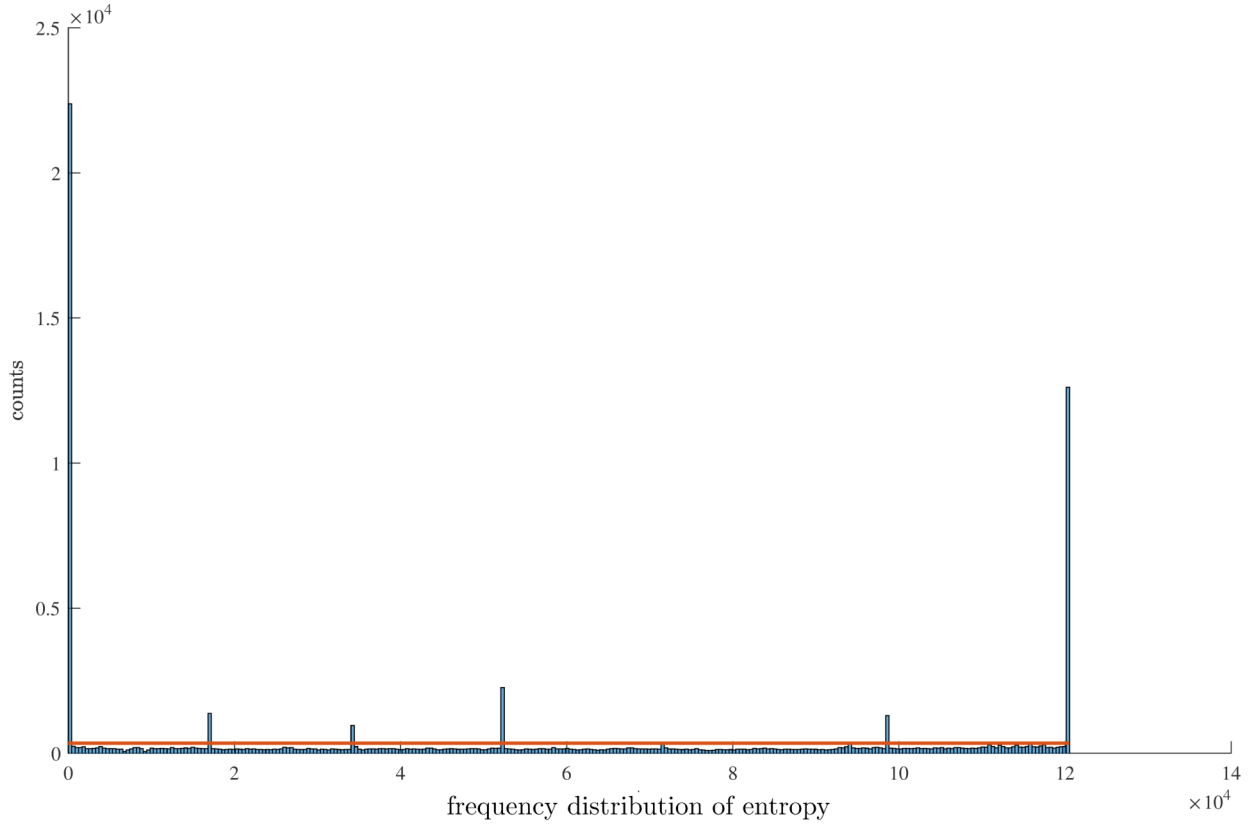

Figure 6.8: Frequency distribution of entropy – the orange line is the threshold separating activity from inactivity. The peaks are the pauses.

When implementing this algorithm in practice: taking longer periods of time for the computation or even using the value from the previous day would make the threshold value more significant and stable over time.

6.3.4 Find working time

From the previous step the entropy levels at which no work is done are available. Now the time periods associated with these entropy levels are determined using the secondly entropy and the corresponding timestamps.

Using this approach no consolidation step is used, however, if results are not satisfactory at this point a consolidation of working time blocks could be implemented.

So the working times can be found as illustrated in Figure 6.9.

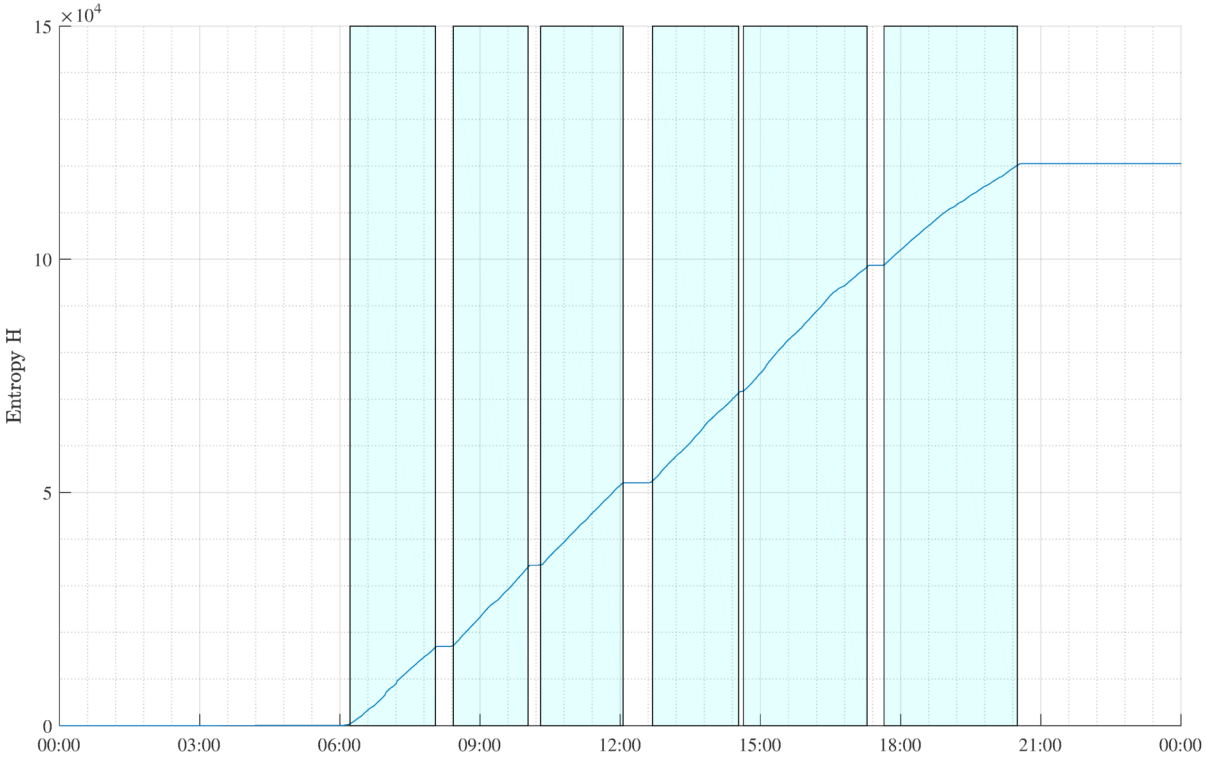

Figure 6.9: Example output for the entropy approach for finding working times – the blue line marks the behaviour of the entropy throughout the day. The working times are marked in light blue.

The three approaches – expert knowledge, unique texts and entropy – are evaluated in the following chapter using actual log data.

Chapter 7

Approach Evaluation

This chapter presents the evaluation of the proposed approaches. Firstly it will be described which criteria have been used for evaluation; then what the result of this evaluations are; and what can be learned from them regarding their applicability to solve the problem.

The following criteria have been considered for the evaluation of the three approaches described in Chapter 6:

1. Total daily deviation: target working time minus actual working time according to approach. This will be the main criterion and can also easily be presented graphically.

2. Hourly deviation: Target working time per hour minus actual working time according to approach per hour. This is the same as first criterion, just broken down on the 24 hours of the day. To use this criterion for the evaluation as one ratio – rather than 24 – the positive hourly values are added up. This is due to the fact that, considering that the working time is used for productivity calculation – detecting too much working time is better than too little.

3. Plus: All times classified by the respective approach as working time, although they should not be working times.

4. Minus: All times that should be working times, but have not been classified as working times by the respective approach.

5. Deviation of the number of working time blocks: target number of working time blocks – actual number of working time blocks. This will be the least influential criterion, as it has virtually no effect on the accuracy of productivity, but still gives information about the accuracy of the algorithm.

Generally detecting too much working time is less of a problem than detecting too little. This is due to the fact that the working time will be used to calculate productivity and how the productivity is calculated.

7.1 Results

The above mentioned evaluation criteria have been calculated for log files from two different systems each with five sample days. The log files originate from the WCS of KNAPP warehousing systems. The two systems consist of different components and have different degrees of automation. Additionally to the log files the respective allocated working times are available for comparison.

After the calculation of the criteria averages are calculated for each system and criterion over the five days available. Then for each of the criteria the different approaches have been ranked from one to three, with one being the best rating and three the worst. Weights have been allocated to the criteria to enable the calculation of one assessment ratio for each approach and system. Then from the rankings a weighted average has been computed for each of the developed approaches from Chapter 6 – expert knowledge, unique texts and entropy – and for each of the considered systems.

The following table shows the weights of the criteria.

Table 7.1: Weights assigned to criteria for evaluation

Criterion	Weight
Total daily deviation	0.25
Hourly deviation	0.30
Plus	0.10
Minus	0.25
Deviation of the number of working blocks	0.10

The weights have been assigned bearing in mind that the working time will be used for productivity calculations as defined in Section 2.1. As previously stated, the deviation of the number of working time blocks is the criterion with least weight, as the number of working blocks has almost no influence on the accuracy of the productivity calculation and only gives information about the accuracy of the algorithm. The time that is declared working time although it should not be has also been assigned low weight, because it is less of a problem, if there is too much working time due to how the productivity will be calculated.

The highest weight is assigned to the hourly deviation value. The productivity will also

be calculated on an hourly basis. Thus the values of the hourly deviations have the most influence on the accuracy of the productivity calculations. To merge the 24 values of the hourly deviation per day into one value, the positive hourly values are added up. The negatives are ignored, because – as already mentioned before – detecting too little working time is less relevant.

The two other criteria – the total deviation of working time per day and the sum of all working time parts that have not been classified as working time although they should have been – have been weighted equally at 0.25. Both represent the aim of calculating the working time accurately.

The five criteria have been calculated for two different plants and for five days for each of the plants by comparing the allocated working times of the day to the working times calculated by the three approaches. The two plants differ in the degree of automation. Plant 2 is automated to a higher extent than plant 1. After the calculation of the criteria averages are calculated for each system and criterion over the five days available. Table 7.2 shows the results of these calculations for plant 1 – the less automated plant. Table 7.3 shows the results for plant 2 – the plant with the higher degree of automation. The values can be interpreted as proportion of the whole day. Generally the closer the value is to zero, the more similar is the calculated working time to the allocated working time.

Table 7.2: Evaluation criteria values of all three approaches for plant 1

Criterion	**Average over five days plant 1**			
	Expert Knowledge	Unique Texts	Entropy	Dimension
Total daily deviation	0.157	3.457	1.908	hours/day
Hourly deviation	0.507	3.457	2.073	hours/day
Plus	0.410	0.022	0.200	hours/day
Minus	0.567	3.478	2.109	hours/day
Deviation of the number of working blocks	-7	-19.8	-0.4	blocks/day

Table 7.3: Evaluation criteria values of all three approaches for plant 2

Criterion	**Average over five days plant 2**			
	Expert Knowledge	Unique Texts	Entropy	Dimension
Total daily deviation	9.060	12.888	10.185	hours/day
Hourly deviation	9.190	12.952	10.546	hours/day
Plus	0.180	0.070	0.372	hours/day
Minus	9.240	12.958	10.557	hours/day
Deviation of the number of working blocks	-110	-17.6	-25.4	blocks/day

After theses calculations for each of the criteria and plants the three approaches are ranked from one to three. A one is assigned for the lowest criterion value, as this means that the deviation between the allocated working time and the calculated working time are the least. After allocating the ranks, averages are calculated for each of the plants and approaches using the weights stated in Table 7.1. The results of these calculations are represented in Table 7.4.

Table 7.4: Results of averaging the assigned ranks per plant and approach

Plants	**Average of the ranks**		
	Expert Knowledge	Unique Texts	Entropy
Plant 1	1.3	2.8	1.9
Plant 2	1.3	2.6	2.1

For both different systems the final ranking of the three approaches is the same:

1. Expert Knowledge
2. Entropy
3. Unique Texts

The following section discusses the reasons for and implications of this result.

7.2 Discussion

In this section the results and evaluation mechanism presented in the previous section will be discussed.

To compute the evaluation criteria, the working times calculated by the three approaches have been compared to the allocated working times provided by the operating companies of the warehousing systems. Only the allocated working times and not the actual working times were available, because, generally, it is a very difficult task to find the actual working times in the systems. At each component of the warehousing system a person would have to be positioned to observe the component and write down the time intervals during which human workers are active in the system – this is virtually impossible. So shift times and break times were used for the comparisons, although these are the times that are supposed to be improved by the approaches presented in this thesis. For this assessment it is assumed that the allocated working times are equal to the actual working times. However, events

such as hardware breakdowns or longer shifts due to higher workload are not reflected in the allocated working time, although they affect the actual working time of humans in the warehousing system. So in general one would expect results of the working time calculations to differ slightly, but not too much from the allocated working times – which is difficult to quantify. So one should keep in mind that this is no ultimate ranking of approaches and also the approaches might work much better than it is suggested by the values in Table 7.2 and Table 7.3.

When looking at Table 7.2 and Table 7.3 one difference is striking at first sight: the values in the second table which evaluates the approaches at plant 2 are much higher than in Table 7.2. The reason for this is most likely the difference of the degree of automation of the two plants. The general observation is that the higher the degree of automation, the less human intervention is needed to produce logs. As the values are higher for all three approaches this also shows that the expert knowledge approach described in Section 6.1 works less exactly with the logs from the more automated plant. This may indicate that the keywords listed by the expert may be inappropriate for detecting working time correctly in such highly automated plants or at least the list may be incomplete, so much less working time is detected than it should.

Another problem is that there is no standard for how long a break for a human is. During the work for this thesis different mechanisms were used to find these breaks (through thresholding and clustering).

The following figure shows an example of the working time detected in the more automated plant using the unique texts approach from Section 6.2. There is work done at the plant throughout the day with only two interrupting pauses from 10:00 to 10:15 and from 13:55 to 14:10. The problems although the it seems that most of the work is done during daytime – it might be even possible that there is more staff present during daytime. This causes the activity level of the day being much higher than the activity level during nighttime.

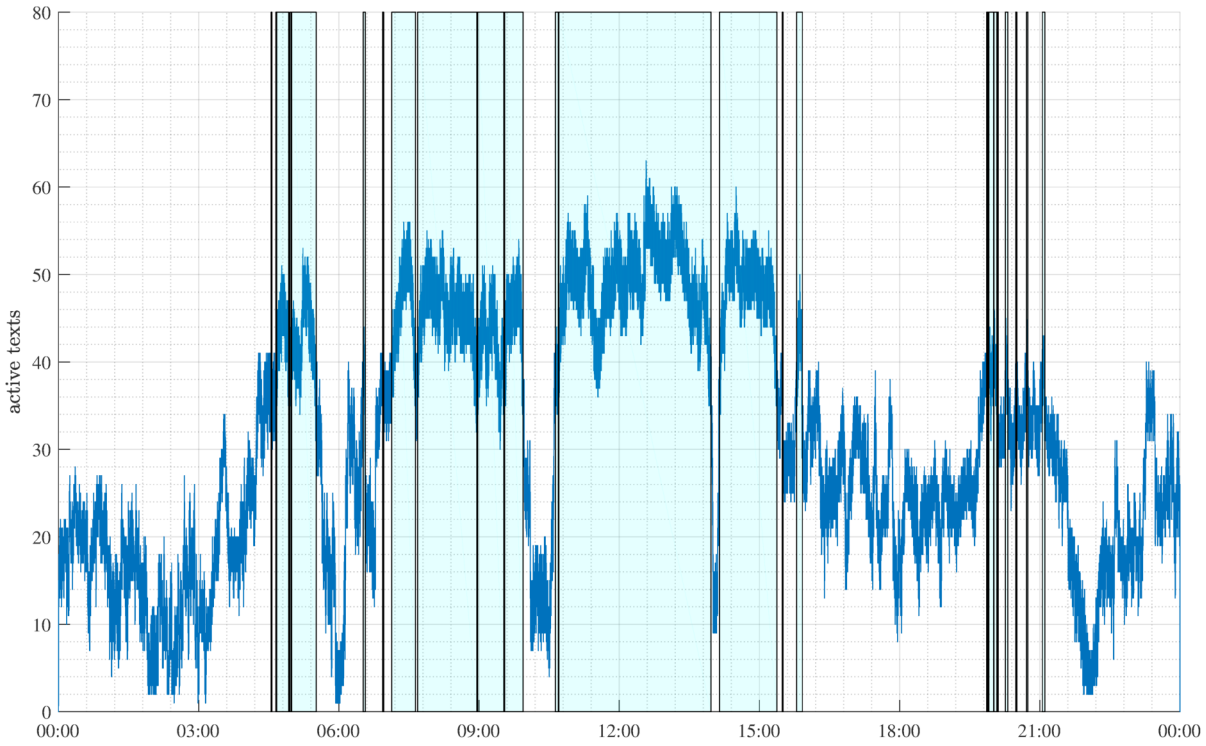

Figure 7.1: Example output for the unique texts approach for finding working times – the blue line marks the behaviour of the entropy throughout the day. The working times are marked in light blue.

Another point to the assessment of the algorithms in general is that already little variations might have an effect on the output. From the components and ideas that have been used creating the approaches presented in this thesis, numerous other algorithms would have been possible by combining them differently. For simplicity reasons, however, in this thesis only three possible approaches have been investigated and evalutated in further detail.

The factors which have most impact on the outcome of each approach – apart from the obvious main idea or method used – are:

1. There are many different **thresholding methods** that could be used for thresholding. Also in some approaches multiple thresholding could be used.

2. As histograms are used in each of the approaches the choice of the **number of bins** and the resulting **bin widths** have an impact.

3. For **clustering** the idea of scale space computation was used. There the method which chooses the scale could be varied. However, one could also use different methods for it, e.g. calculate a threshold for the length of pauses or working time blocks.

As can be seen in Tables 7.2 and 7.3 the algorithms tend detect to too little working time than too much. This is a problem because it raises productivity artificially. This could, at least for the unique texts and the entropy approach, changed by altering the thresholding methods used.

Another point that adds to the possible inaccuracy of the calculations is that the thresholds are calculated on a daily basis, not taking into account any other data. If the algorithms are used in practice the threshold computation will be extended and thresholds will be computed as floating weighted averages, so that the threshold calculation of the particular day is not only dependent on the data of this day, but also on all previous days, integrating the threshold of the previous day into the computation. This is due to the fact that the nature of how the system works cannot change dramatically from one day to the other. This enables the algorithms to detect also days on which no work is done – which would be impossible otherwise.

Although the entropy algorithm does not have the best ratings it will be used as basis for future development and implementation in practice, as it has much potential for improvement and it can be used without any expert knowledge.

Chapter 8

Conclusion and Outlook

In this thesis three approaches have been presented which aim at determining the working time of humans in a warehouse from the log files originating from the warehouse control system. Objective measures have been defined and applied to enable comparison of the different approaches. The evaluation results vary widely between the different approaches and the different warehousing systems.

It has been shown that, although the used log data was not designed for automated analysis, as presented in this thesis, information could be gained from it. Particularly, if no knowledge about the content of the log messages was incorporated, the working times could be found partly with sufficient accuracy. To make more definitive statements about the approaches, significantly more log data would be needed for evaluation.

Especially for warehousing systems with a comparably low level of automation the working times could be found sufficiently reliably. For warehousing systems with a higher degree of automation the approaches still need some enhancement since the logs are not strictly indicative of human activity. In this case not even the use of expert knowledge led to acceptable outcomes.

The particular problem would have been much easier to solve, if the log-files were defined in a more definitve manner; e.g. stating more clearly which log messages indicate working time and which do not. If log messages were structured more stringently, even process mining-approaches and -tools could be used.

Outlook

It is necessary to define which information and knowledge should be extracted from the log data and to determine which data is required to enable this. The logs should be structured in some form of formal markup, e.g. a left logical language, so that the data and its contextual dependencies can be uniquely determined. The use of formal definitions would also enable the automatic generation of lexicographic- and parsing tools.

Particular attention should be paid to the logging in warehousing systems with a higher degree of automation, as the results of the approaches suggested in this thesis are much worse for the system that is automated to a higher degree.

When implementing one of the approaches in practice, the threshold calculation should also take data of the past into account; the concept of *infinite impulse response filters (IIRF)* might be of use to reflect the quasi-persistent nature of the systems.

If requested for a particular future application, near real time evaluation would be possible implementing the concept of a *dead man's handle*.

An altered and improved version of the entropy approach will be implemented in pratice in *PL/SQL* based on the findings of this thesis.

List of Figures

List of Tables

Bibliography

[1] A. I. R. L. Azevedo and M. F. Santos, "Kdd, semma and crisp-dm: a parallel overview," *IADS-DM*, 2008.

[2] J. Bernstein, "The data-information-knowledge-wisdom hierarchy and its antithesis," *NASKO*, vol. 2, no. 1, pp. 68–75, 2011.

[3] S. Brandt, *Data Analysis*. Springer, 1999.

[4] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, "Crisp-dm 1.0 step-by-step data mining guide," 2000.

[5] M. J. Embrechts, B. Szymanski, and K. Sternickel, *Computationally Intelligent Hybrid Systems: The Fusion of Soft and Hard Computing*, 2005, ch. Introduction to Scientific Data Mining: Direct Kernel Methods & Applications.

[6] P. Esling and C. Agon, "Time-series data mining," *ACM Computing Surveys*, vol. 45, no. 1, 2012.

[7] L. Fahrmeir, R. Künstler, I. Pigeot, and G. Tutz, *Statistik*. Springer, 2007.

[8] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Magazine*, vol. 17, 1996.

[9] ——, "The kdd process for extracting useful knowledge from volumes of data," *Communications of the ACM*, vol. 39, no. 11, pp. 27–34, 1996.

[10] A. Frei and M. Rennhard, "Histogram matrix: Log file visualization for anomaly detection," in *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*. IEEE, 2008.

[11] M. Frické, "The knowledge pyramid: a critique of the dikw hierarchy," *Journal of information science*, vol. 35, no. 2, pp. 131–142, 2009.

[12] S. A. L. G. W. Zack, W. E. Rogers, "Automatic measurement of sister chromatid exchange frequency," *The Journal of Histochemistry and Cytochemistry*, 1977.

[13] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Pearson Prentice Hall, 2008.

[14] C. W. Groetsch, *Inverse Problems*. The Mathematical Association of America, 1999.

[15] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. Morgan, 2006.

[16] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2012.

[17] R. M. Laura Roberts and E. Winter, *Gabler Wirtschaftslexikon*. Gabler, 2010.

[18] S. Laxman and P. S. Sastry, "A survey of temporal data mining," *Sadhana*, vol. 31, pp. 173–198, 2006.

[19] T. Lindeberg, "Scale-space theory: A framework for handling image structures at multiple scales," in *CERN School of Computing, Egmond aan Zee, The Netherlands*, 1996.

[20] P. O'Leary, M. Harker, R. Ritt, M. Habacher, K. Landl, and M. Brandner, "Mining sensor data in larger physical systems," *IFAC-PapersOnLine*, vol. 49, no. 20, pp. 37 – 42, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896316316561

[21] N. I. S. Organization, "Understanding metadata," NISO Press, 4733 Bethesda Avenue, Suite 300 Bethesda, MD 20814 USA, 2004. [Online]. Available: http://www.niso.org/publications/press/UnderstandingMetadata.pdf

[22] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems Man, and Cybernetics*, 1979.

[23] J. Riley and K. Delic, "Enterprise knowledge clouds: Applications and solutions," in *Handbook of Cloud Computing*, B. Furht and A. Escalante, Eds. Springer, 2010, pp. 437–452.

[24] T. A. Runkler, *Data Mining: Methoden und Algorithmen intelligenter Datenanalyse*. Vieweg+Teubner, 2010.

[25] M. Shanhanawaz, A. D. M. Ranjan, A. Ranjan, and M. Danish, "Temporal data mining: An overview," *International Journal of Engineering and Advanced Technology*, vol. 1, no. 1, 2011.

[26] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, 1948.

[27] M. ten Hompel and T. Schmidt, *Warehouse Management: Automatisierung und Organisation von Lager-und Kommissioniersystemen*. Springer, 2008.

[28] J. Valdman, "Log file analysis," *Tech. Rep. DCSE/TR-2001-04*, 2001.

[29] W. van der Aalst and A. Adriansyah et al., *Process Mining Manifesto*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 169–194.

[30] W. M. P. van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, 1st ed. Springer Publishing Company, Incorporated, 2011.

[31] VDI, *VDI 2411 - Begriffe und Erläuterungen im Förderwesen*, 1970, no. 2411.

[32] A. Witkin, "Scale-space filtering," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84.*, vol. 9. IEEE, 1984.

[33] I. H. Witten, E. Frank, and M. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., 2011.

[34] H. Zsifkovits, *Logistik*. UTB, 2012.

Appendix

During the work for this thesis the author used own code as well as code that was available at the Chair of Automation.

Used Tools

Matlab R2016b