CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering
Department of Cybernetics

Analysis and sequential mining of logistic data

Bachelor Thesis

Study Programme: Open Informatics

Branch of study: Computer and Information Science

Thesis advisor: Ing. Jíří Kléma, PhD.

Filip Mihalovič

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 24.5.2013

Podpis autora práce

Abstract

We are given a large database of logistic data containing information about shipments and their states in time. We introduce the problem of sequential data mining over such database collected in shipment handling process. We present three approaches on handling such data, which are, respectively, simple sequential data mining tool, advanced sequential data mining tool and transition probability model. We evaluate the differences between the approaches and between the process documentation provided and the output of the approaches. The simple tool can provide most of the common subsequences and sequences, the advanced tool can provide complex overview and using the probability model is a fast, but not very exact method to describe the input dataset.

Abstrakt

Máme k dispozici velkou databázi logistických dat obsahující informace o zásilkách a jejich stavech v čase. Naším cílem je porozumět souvislostem mezi událostmi, které při doručování zásilek nastávají. Využíváme k tomu nástrojů sekvenčního dolování dat. Prezentujeme tři různé přístupy k řešení dané problematiky: 1) aplikaci jednoduchého nástroje pro sekvenční těžení dat, 2) využití pokročilejšího nástroje pro sekvenční těžení dat a 3) vytvoření pravděpodobnostního přechodového modelu s omezenou pamětí. Zkoumáme rozdíly mezi přístupy a využíváme jejich odlišných vlastností k porozumění doméně. Jednoduchý nástroj sekvenčního dolování dat odhalil řadu zajímavých častých podsekvencí, pokročilý nástroj umožnil rozšíření škály vyhledávaných podsekvencí a použití pravděpodobnostního modelu vede k souhrnnému a přibližnému náhledu na posloupnosti událostí.

Contents

1. Introduction

1.1. Preface

DHL is the global market leader in the logistics industry. Every day, DHL is collecting shipments' data from warehouses all over the world containing shipments' detailed information. It has a reporting application ready to provide data to the managers and customers. The problem is that no unite system for surveillance over this huge amount of data is available. Therefore it is complicated to get some useful up-to-date information from everyday shipment data and to dynamically adapt to updated requests from the customers.

My project proposes to involve statistics and basic process mining techniques to improve business decisions. Given environment and data are perfect for such solution, because of its complexity and volume of provided data. Proposed solution can start with some basic information and provide a huge potential to grow with time, even to become a daily used solution for business flow overview. Statistics can start with general information and roll down to basic levels of each shipment. Well-developed data infrastructure guarantees access to needed information with regards to some speed issues while extracting required data.

The aim of my project is to use data caught in application integration layer of DHL and use it in a way it would help to analyze business strategy and provide statistical information for the management and customers. The final data will be provided in an easy-to-read form consisting of tables, graphs and diagrams. Also the solution has to be robust and easily scalable to adapt to everyday business needs.

In the following sections, I will try to describe the proposed solution and aim closer to the important parts of it. I will mainly concentrate on the practical parts of the implementation. Also some enhancements will be proposed to show where the solution can grow and where the weak points of it are.

The first section will describe the needed theoretical knowledge needed for this task, and also provide some insight into the problem of sequential data mining and its possible adaptation to the DHL's corporate business model. Then the particular gained data will be described to understand the form of it, and basic statistical will be carried out to understand the content of the dataset. Later on, various sequential data mining techniques will be applied to identify frequent and unique sequences in the business process. In the end, sequential data mining output will be compared within different tools and also with Markov chain probability model.

1.2. Assignment

The aim of this bachelor thesis is to get familiar with the topic of sequential data mining and its available tools. To achieve this goal, understanding the corporate business process within logistics' industry and data saved during package delivering is required. Analyze the data to understand its structure and format. Extract data from the integration layer for a specific time period in such format so that it could be used as an input to sequential data mining tools. Apply sequential data mining techniques and try to find the most common processes. Compare the common process with the official business process description and discuss similarities and differences. Then try to extract sequences that are not frequent and that do not occur often, so called outliers. Find out why these outliers exist and how a process can become an outlier. Create a state diagram, which would represent most of the found sequences in the sequential data mining stage. In the end, create a transition probability matrix that would represent the probability of transitions between all states. Conclude and discuss the results of the project, how it could help in real life situation and enhance the business flow of the company and the whole logistics industry.

2. Theory

2.1. Used terms

DHL

DHL is the global market leader of the international express and logistics industry, specializing in providing innovative and customized solutions from a single source. DHL offers expertise in express, air and ocean freight, overland transport, contract logistics solutions as well as international mail services, combined with worldwide coverage and an in-depth understanding of local markets. DHL's international network links more than 220 countries and territories worldwide. Some 300,000 employees are dedicated to providing fast and reliable services that exceed customers' expectation.

When critical spare parts delivery to customers within an agreed timescale is needed, DHL's Service Parts Logistics (SPL) solutions will provide this service. Operating globally, DHL designs and maintains systems that not only get customers the parts they need quickly, but also help to anticipate and prepare for that demand. Giving a complete perspective of supply chain and an inventory of what is in stock and in motion. It's all managed by an end-to-end model, integrating transportation, warehousing, and repair-cycle management.

Business Intelligence

Business Intelligence is an emerging discipline within corporate sector; it is a category of various technologies and applications that provide decision-makers needed and very important information from raw running data. Few out of many business intelligence tools are for example Reporting and Online Analytical Processing. Business Intelligence can be simply characterized as a system for decision support.

Reporting

Reporting is a fundamental part of the larger movement towards improved business intelligence and knowledge management. With the dramatic expansion of information technology, and the desire for increased competitiveness in corporations, there has been an increase in the use of computing power to produce unified reports which join different views of the enterprise in one place. This reporting process involves querying data sources with different logical models to produce a human readable report. For example query takes enormous production data from warehouses and shows how efficiently space and time are used across an entire corporation.

OLAP schema

An OLAP schema (Online Analytical Processing) is a logical model that defines a multidimensional data structure. It defines one or more OLAP cubes in a single database that each are defined by one or more dimensions and measures. A cube can be considered a generalization of a two-dimensional spreadsheet. For example, a company might wish to summarize data by product, by time-period, by location to compare actual and real delivery time. Product, time, location and scenario are the data's dimensions. OLAP data

is typically stored in a star schema or snowflake schema in a relational data warehouse or in a special-purpose data management system.

The elements of a dimension can be organized as a hierarchy, a set of parent-child relationships, typically where a parent member summarizes its children. Parent elements can further be aggregated as the children of another parent. Conceiving data as a cube with hierarchical dimensions leads to conceptually straightforward operations to facilitate analysis. Aligning the data content with a familiar visualization enhances analyst learning and productivity.

Insofar as two-dimensional output devices cannot readily characterize four dimensions, it is more practical to project "slices" of the data cube (project in the classic vector analytic sense of dimensional reduction, not in the SQL sense, although the two are conceptually similar), which may suppress a primary key, but still have some semantic significance.

Integration layer

Integration layer is an application or set of applications that provide communication between divided corporate systems within organization or between customer and provider systems. Integration application can adapt to different structure of systems making it easy to interconnect systems that were not designed to communicate together. What is more, Integration layer provides also security enforcement of access privileges, message processing and management of stored messages. There is a variety of Integration system types. Most used are central hub, data hub and bus. Some of the implementations can save information transferred between systems, runtime information, logs and history of transformed messages and therefore it is a perfect place where to look for reporting and data mining sources.

2.2. Sequential Data Mining

Sequential Data Mining

Sequential data mining is a discipline of Data mining, which aim is to extract frequent subsequences, patterns and sequence rules from a given sequences. Sequential mining technique should find the complete set of patterns while supporting the minimum support threshold within given constrains like length constraint, type constraint, gap constraint, etc. In Sequential data mining, it is important to keep good identification of sequence states, distinguishing between sequences and order of states in each sequence. Sequential data mining not only provides information about patterns that do occur together, but also distinguishes the order and time difference between each event in given sequences. To put it another way, Sequential data mining is trying to discover relationships between occurrences of states to find specific order of the occurrences.

The sequential pattern mining problem was first introduced by Agrawal and Srikant in [1]: "Given a set of sequences, where each sequence consists of a list of elements and each element consists of a set of items, and given a user-specified minimum support threshold, sequential pattern mining is to find all frequent subsequences, i.e., the subsequences whose occurrence frequency in the set of sequences is no less than minimum support".

Sequence

Sequence is a series of events of an observed item containing ordered list of these events and timestamp when they occurred. After observing more items with similar properties and expected variation of an event order and time of occurrence, Sequential Data Mining can be deployed to find sequential patterns, frequent subsequences, etc.

Frequent subsequence

Subsequence is a subset of a sequence that occurs in given dataset of sequences meeting minimum support property set for subsequence search.

A sequence $<a_1,a_2\ldots a_n>$ is a subsequence of another sequence $<b_1,b_2\ldots b_m>$ if there exists integers $i_1 < i_2 < \ldots < i_n$ such that $a_1 \subseteq b_{i1}$, $a_2 \subseteq b_{i2}$, … , $a_n \subseteq b_{in}$ [2].

The support for an itemset *i* is defined as the fraction of customers who bought the items in *i* in a single transaction. Thus the itemset I and the l-sequence $< i >$ have the same support. An itemset with minimum support is called a large itemset. Note that each itemset in a large sequence must have minimum support. Hence, any large sequences must be a list of large itemsets [1].

Sequential pattern

Sequential pattern is a sequence of events that frequently occur in a specific order, all events in the same pattern are supposed to have the same transaction time value or within a time gap. Usually all the transactions of a customer are together viewed as a sequence, usually called customer-sequence, where each transaction is represented as an event in

that sequence, all the transactions are list in a certain order with regard to the transaction-time.

Minimum support of a sequence

By using Minimum support of a requested sequence, unneeded or uninteresting sequential patterns can be pruned out. This option ensures that only wanted patterns are found.

To put it into the context, a sequence database S is a set of tuples <*sid*, *s*>, where *sid* is a sequence identification and *s* a sequence. A tuple <*sid*, *s*> is said to contain a sequence α, if α is a subsequence of *s*. The support of a sequence α in a sequence database *S* is the number of tuples in the database containing α, i.e.,

$$support_s(\alpha) = |\{<sid, s>|(<sid, s> \in S) \wedge (\alpha \subseteq s)\}|$$

It can be denoted as *support(α)* if the sequence database is clear from the context. Given a positive integer *minimum support* as the support threshold, a sequence α is called a sequential pattern in sequence database S if $support_s(\alpha) \geq minimum\ support$. A sequential pattern with length *l* is called an *l*-pattern [3].

Episode

An episode is a collection of events that occur relatively close to each other in a given partial order. When such episode is found, rules can be identified within sequences to predict its behavior.

Formally, an episode α is a triple $(V, \leq, g)$ where V is a set of nodes, $\leq$ is a partial order on *V*, and $g : V \rightarrow E$ is a mapping associating each node with an event type. The interpretation of an episode is that the events in $g(V)$ have to occur in the order described by $\leq$. The *size* of α, denoted $|\ \alpha\ |$, is $|V|$.

Episode α is *parallel* if the partial order $\leq$ is trivial (i.e., $x \not\leq y$ for all $x, y \in V$ such that $x \neq y$). Episode α is *serial* if the relation $\leq$ is a total order (i.e., $x \leq y$ or $y \leq x$ for all $x, y \in V$). Episode α is *injective* if the mapping *g* is an injection, i.e., no event type occurs twice in the episode [4].

Sequence window

Sequence window is a constraint defined by a user that defines a time period in which an episode must occur. After applying this constraint, a sequence can be seen as a set of overlapping windows with a predefined size.

Formally, a window on an event sequence $s = (s, T_s, T_e)$ is an event sequence $w = (w, t_s, t_e)$, where $t_s < T_e$ and $t_e > T_s$, and *w* consists of those pairs (A, t) from *s* where $t_S \leq t < t_e$. The time span $t_e - t_s$ is called the *width* of the window *w*, and it is denoted *width*(*w*). Given an event sequence *s* and an integer *win*, we denote by *W*(*s*, *win*) the set of all windows *w* on s such that *width*(*w*) = *win*.

By the definition the first and last window on a sequence extend outside the sequence, so that the first window contains only the first time point of the sequence, and the last

window contains only the last time point. With this definition an event close to either end of a sequence is observed in equally many windows to an event in the middle of the sequence. Given an event sequence $s = (s, T_s, T_e)$ and a window width *win*, the number of windows in $W(s, win)$ is $Te - Ts + win - 1$ [5].

Outlier

Outliers are unexpected patterns with extreme behavior that do not belong to a majority of the processes in the process flow. These outliers can contain very interesting and often hidden properties of the processes. Detection of outliers is closely connected with as good as possible detection of normal behavior, since they are defined as deviation from normal. As it is probably clear, sequence does not have to be a total outlier. Outlier can be called also a series of events which do not correspond with the normal flow, even if the end of the sequence joins the major events set. Outlier detection is a critical task in many safety critical environments as the outlier indicates abnormal running conditions from which significant performance degradation may well result. Outlier detection accomplishes this by analyzing and comparing the time series of usage statistics.

In general, three main design patterns have emerged to detect and extract outliers based on distribution, distance and density. In the distribution-based approach the underlying statistical distribution of the data source is estimated, say M, and a data point d is considered to be an outlier if $P(d|M) < t$ for a user-specified threshold. A known limitation of this approach is that computing the distribution of complex heterogeneous and high-dimensional data sets is non-trivial if not intractable The distance-based paradigm was originally proposed by Knorr and Ng [6] in which each data point is represented as point in a n-dimensional space. Points whose distance to their k-th nearest neighbor is large are considered candidate outliers. Several variations and efficient algorithms on this have been proposed. A limitation of distance based outlier techniques is that they are not flexible to discover local outliers, especially in data sets which have non-uniform density as one moves across the data landscape. This limitation was lifted by Breunig et. Al [7], who introduced the concept of Local Outlier Factor (LOF) that takes the local density into account when checking for outliers.

3. Related work

On 10th May 2010, on DHL SSOW conference in Amsterdam a presentation on e-billing was presented [9]. As main improvement of this solution cost saving and easier access to invoice data was shown. The next focus of this presentation was future development of reporting and analysis from gained data. This option was not possible while paper invoices were used. One of the conclusion thoughts of this presentation were to continue in innovation by implementing e-marketing and data mining techniques.

Earlier that year DHL Data mining project aiming on Customer segmentation with clustering was finished [10]. One of the inhibitors for this project is as stated here: "DHL lacks the information about their customer segmentation and profile to help them to make decisions in marketing, pricing and other business decisions. While they have strong domain knowledge and a rich past transaction data, they lack the expertise to mine out interesting patterns." This research segmented data according to which customer they belong. Then these customers' data were compared and patterns in customer transactions tried to be found. Mainly processed data in graphical form were the output of this research. Interesting results like highest revenue per transaction, revenue per weight and many more are discussed in this document. They were also focusing on the discovery of liable customers and those who have some peaks in cooperation. But despite finding some clusters ready for business decisions, one of their conclusions were that "Due to the limited domain knowledge we have about the logistic industry and the business, we find it hard to come up with detailed recommendations". This meant that due to poor understanding of business data they had to hand found observations to business department.

Research done on Temporal Pattern Mining in Logistics [11] had its targets in finding and identification of patterns in logistic data. Simple data having two properties, state description and time of occurrence, was expected as the input. Prolog was used to set requested parameters of logistic process and then find frequent patterns and identify rules based on Allen's theory of action and time [12]. They proposed a learning algorithm that will set up temporal prediction rules. Using these rules, critical situations in logistic processes are supposed to be identified before they actually happen. Using training sets they were able to predict when rescheduling for a shipment will be needed in an early state of the process. Although this research tried to be practical and proposed solutions said to be ready for real-life logistics situations, it was running only on prepared test scenarios. It is not clear how the proposed learning algorithm will cope with real production data containing unseen situations and patterns.

The research in paper Business process mining: Industrial application [13] collects business data from logs and transaction services (like integration services between applications) and then uses these data to create process models. This research was carried out based on data from Workflow management system of Dutch National Public Works Department. Their propose solution to create variety of models, starting from mining of

simple processes with a few branches to mining complicated organization structures. One of the most interesting finding from my point of view is the main process flow. It revealed a highly informative process model providing information on state usage and time spent in each state. The model was optimized by removing states with low usage and preventing repetition of the same branches by one process.

When it comes to sequential data mining, it would be a sin not to mention the document Process mining: a research agenda by Eindhoven University of Technology [14]. This document gives a perfect overview on process mining start and proposes results a project can focus on. Great illustration of process models is given, and what's more it provides models of unwanted processes in process mining. Such models to look for are for example duplicate tasks, non-free-choice constructs and loops. The document is a good introduction into process mining and gives understandable examples. It also issue setbacks of process mining techniques which occur almost in every application of process mining (loops, duplicated tasks, hidden tasks, lack of completeness limit, etc.).

Another related are to this work are outliers. In Mining for Outliers in Sequential Databases [15] definition of this problem is created and algorithms for finding outliers is proposed. Their method relies on building PST (probabilistic suffix tree) on a database. Then to find an outlier in such tree, it is enough to search close the root.

In paper A Survey of Outlier Detection Methodologies [16], author applies various outlier detection techniques to data of different kinds. In the end no method is suggested and the decision is left on developer. As the conclusion says "In outlier detection, the developer should select an algorithm that is suitable for their data set in terms of the correct distribution model, the correct attribute types, the scalability, the speed, any incremental capabilities to allow new exemplars to be stored and the modeling accuracy."

Practical application of Outlier detection can be found in document OUTLAW: Using Geo-Spatial Associations for Outlier Detection and Visual Analysis of Cargo Routes [17]. The document proposes solution based on Outlier Detection to predict and find illegal cargo movement into USA. Its aims are to gather all available data from different agencies and try to identify anomaly before it happens. Based upon anomaly identification (outlier was found) alerts can be raised. The solution uses combination of alerts to find most of the possible anomalies. Proposed control system not only controls individual cargos, but makes a geographical relation of which cargos can be harmful together.

4. Logistic data

4.1. Business process description

The business process of shipment in the system is very complex. Therefore I will try to write it in bullets for better understanding.

Description:

- Order is created and sent to shipment management application (SMA)
- SMA confirms order creation and acknowledges Event Management
- Order is
    - Successfully accepted, shipment is created
    - Declined (no stock, unknown customer, …), end of process
- Shipment is in SMA in state created
- SMA creates message ETA (expected time of arrival) and acknowledges other applications
- Shipment now has all necessary information, it is confirmed; order is created in application Returns
- Shipment is now ready for pickup, physically ready and available in warehouse
- Courier picks up shipment, state is changed to Pick up info entered
- Multiple change of ETA is possible due to various reasons (car breakdown, …)
- Shipment is delivered, state changed to delivered
- State of order in Returns is changed to First leg received
- If shipment is BAD due to some reasons, state in Returns is changed to second leg shipped, this means the package is travelling back to warehouse
- Package arrives to warehouse, state is changed to second leg received
- Package is sent to repair vendor, state is changed to third leg shipped
- Repair vendor accepts or declines shipment, in second case, new package is prepared
- If repaired part in package arrives back to warehouse, state changes to third leg received, new shipment is created automatically repeating the process; if this shipment is successfully delivered, state in Returns changes to closed

In any state until the shipment is picked up, it can be cancelled.

4.2. Data extraction and description

As mentioned before data are taken from A2A.

> A2A (Application to Application) –connecting all other applications into unite working organism, so called integration layer (see section 2.1. Used terms). It provides data synchronization and event forwarding between applications using xml messages and queues. A2A not only resends these messages, but also saves them, so they can be used for other purposes.

Raw data will be taken from the database of the integration application (A2A). New record is inserted into A2A database every time the state of the shipment is changed. The format of the record is standard xml. It contains a variety of information, but the most important is shipment ID and timestamp, which shows the time when record was created. The table where these records are stored contains also column 'queue_id', which defines the state of the shipment.

Extracted production data contains process data for around 5300 shipments. Total number of lines in this file is 77461, what means around 15 states per shipment.

5. Data understanding

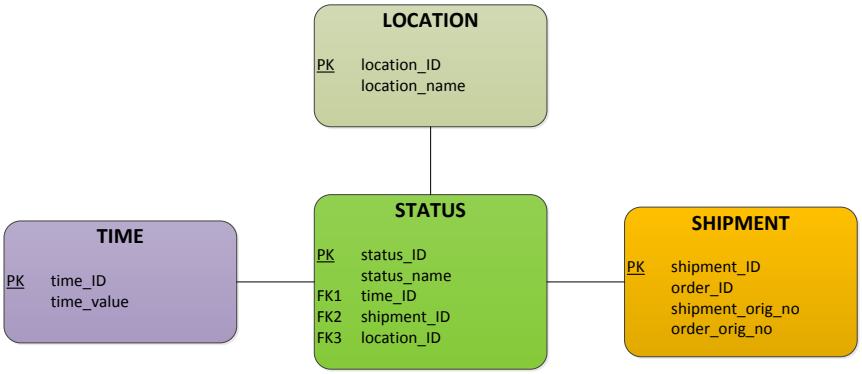
5.1. Data representation

The elements of a dimension can be organized as a hierarchy, a set of parent-child relationships, typically where a parent member summarizes its children. Parent elements can further be aggregated as the children of another parent. Conceiving data as a cube with hierarchical dimensions leads to conceptually straightforward operations to facilitate analysis. Aligning the data content with a familiar visualization enhances analyst learning and productivity [19].

Insofar as two-dimensional output devices cannot readily characterize four dimensions, it is more practical to project "slices" of the data cube (we say project in the classic vector analytic sense of dimensional reduction, not in the SQL sense, although the two are conceptually similar), which may suppress a primary key, but still have some semantic significance.

Implementing OLAP scheme with DHL's integration layer data

A decision was made to implement OLAP scheme to application data of integration application A2A and get information from other databases to create easily accessible and fast database for my further works. Also it creates structure that can be broadened within growing data needs. If working with more details of shipment is needed, it is possible to create another dimension with more detailed data. Starting simple, with data model consisting of four tables interconnected using standard OLAP rules. The logical model is as on following picture.



Status – defines a point in the business process where shipment actually was
Time – timestamp, when shipment A was in status S1
Location – ship from warehouse information, if available
Shipment – table containing all possible shipment identifications

Structure above was used to extract statistic information from the database. One of the OLAP scheme positives is that we can avoid using JOIN command and replace it with more efficient where clause. You can see it on the following demonstration.

SQL query from standard relational database:

```
select * from shipment s
left join delivery d on s.shipment_id = d.shipment_id
left join shipment_status k on s.shipment_status = k.status_id
where k.status_name = 'Delivered;
```

SQL query from OLAP scheme:

```
select * from status, time, shipment
        where
                status.time_id = time.time_id AND
                status.shipment_id = shipment.shipment_id AND
                status.status_name = 'Delivered';
```

5.2. Statistical analysis

In this section, statistic data extracted from database that was found interesting will be provided, and presented in a easy to read form. Following data are from production environment, but please note that exact customer and warehouse names have been hidden by random number or string representation. All gathered data are within period 1st March 2012 and 20th January 2013.

Percentage of shipments per countries

This data presented in a pie graph shows top 10 countries where DHL SPL section is operating and provides percentages of delivered shipments within the time period. This reporting data can provide necessary information for management about current markets, after which some marketing or pricing campaign can start.

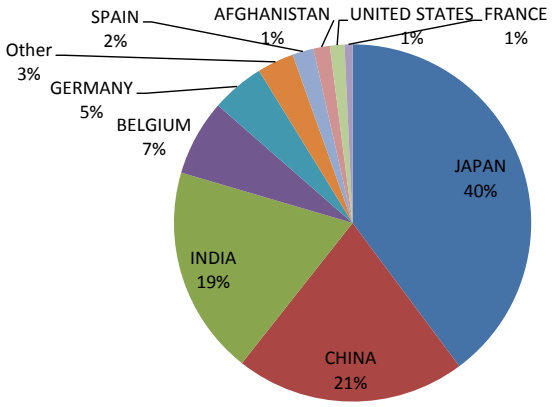


Figure 1: Shipment deliveries in top ten countries

From the above figure, it is visible that shipments are not uniformly distributed all over the world, but depend mainly on 3 markets that are covering 80 percent of shipment deliveries. This also provides necessary information for process mining that concentration should be also aimed on regional location, because some differences might occur.

Actual states of shipments

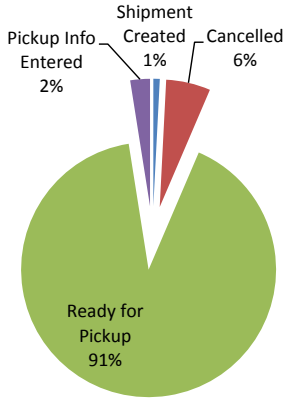


Figure 2: States of shipments in the dataset

Following statistic information gives an eye on what is the actual state of shipments. This information was extracted at 12 pm GMT.

From this, it is clearly visible in which state shipment stays for the longest time. This data should be taken from database in different times of a week, even better month, and then compare them with the results of the process mining concentrating on the shipments' continuance in business defined states. As for the importance of this data, it is necessary to lower shipments in state Ready for Pickup, so that warehouses will be emptied faster and place in warehouses is saved.

Shipment time spent in warehouse

This reporting-like information continues where the previous section stopped. It shows the average time between a shipments' is ready for pick up and the time shipment is actually picked up. Names of the warehouses are hidden behind random numbers. The time on y-axis is in minutes. Red color bar shows the average time, which is 487 minutes. Median is 230 minutes. The best pickup time is 25 minutes.

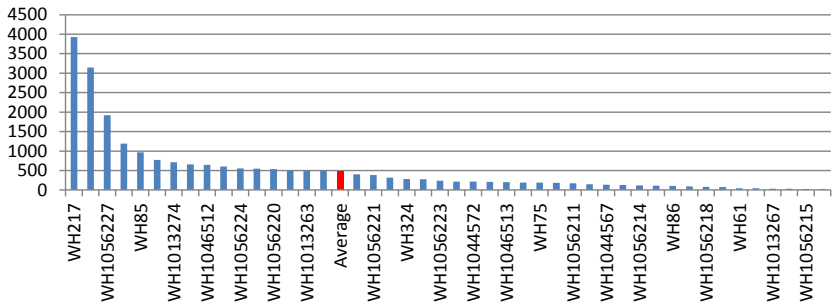


Figure 3: Average shipment time spent in warehouse

As it is visible, average time is higher the median. In process mining, it would be great to identify shipments with high pickup time in early phase and then lower average closer to median. This graph shows, that in some warehouses, shipments are prepared way earlier, than they could be. It leads to inefficient materialization of warehouse and human capacities.

Cancelled shipments

Number of cancelled shipment per warehouse confirms trend, that small amount of warehouses have a high influence on the average. In this case, only five warehouses create more than 50 percent of cancelled shipments. Process mining section should identify such shipments in an early phase.

Couriers

There are altogether 503 registered couriers in the system. The delivery time of couriers vary from 1 minute, which is probably value caused by human mistake, to 20500 minutes, which can be caused by late information entry to the system. In future, it would be efficient to link couriers to warehouses and analyze this on two layers. Firstly analyze how warehouses handle with shipment process and, secondly, in which amount do couriers affect this handling by their speed.

Shipment growth since application start

Since the application for shipment management, which handles shipment deliveries, was launched in March 2012, number of shipments still has a growing trend. Following graph shows the number of shipments achieving state within selected month.

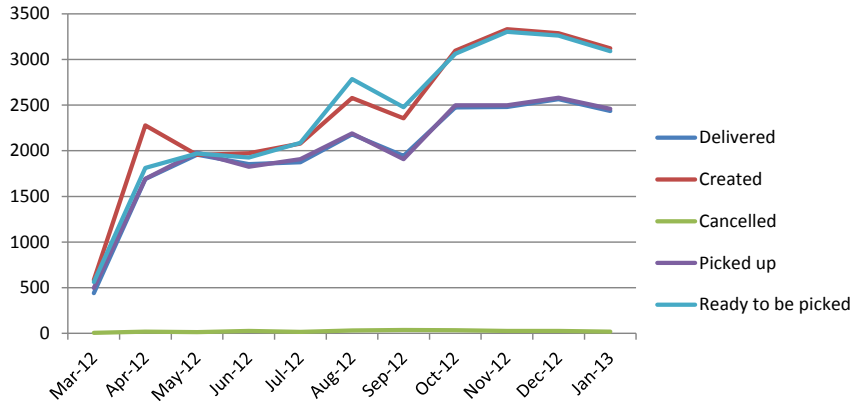


Figure 4: Evolution of shipment states within one year

What really does stand for a comment is the gap between created and delivered shipments. It is clearly visible, that monthly more shipments are created than the process can handle. This fact postpones delivery of shipments into another month, and the trend is that the gap will only become bigger if nothing is done. It is visible that within previous time period, maximum deliveries that DHL SPL was able to make was around 2500, whereas number of monthly created shipments rises to 3300. Therefore optimization of shipment deliveries has to be made, because we believe this gap can be closed by smarter package handling and process improvements that can be found by applying process mining techniques.

6. Experimental protocol

6.1. Used tools

To obtain the best results possible, more tools were considered to be applied. After counting the pros and cons of various tools, two of them were chose and respectively dmt4sp tool [18] and Sequential and Pattern Mining Framework (SPMF) [8]. Also other tools came into consideration, for instance SPAM [20] or Ferda Data Miner [21], but they were not chosen due to the fact, that dmt4sp and SPMF have better documentation and were considered better for academic purposes. However choosing tool for sequential data mining is pure personal thing and it depends on the given task or user habits.

Dmt4sp is Linux based command line application with one present algorithm and a number of options, described in the practical application part, which can be applied to dataset to extract the data requested. Output is then saved to selected text file with all related details.

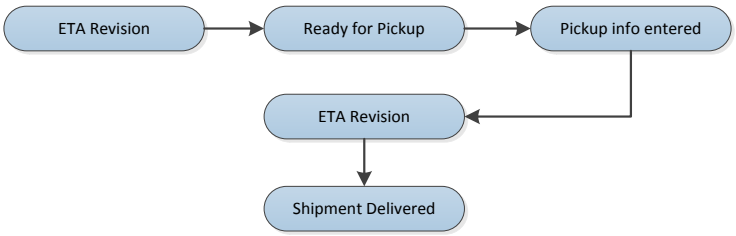
Sequential and Pattern Mining Framework can be used as a library to existing JAVA application, command line application or standalone program with graphical user interface. Containing several algorithms for each association rule mining, sequential rule mining and sequential pattern mining, it allows experimenting with a dataset looking for the best possible results. To see actually used algorithms see section *6.4.Advanced techniques*. As well as the dmt4sp, it stores the output to a text file in a fixed format making it easy to use this file for post processing or for analytical purposes.

6.2. Input and expected output

One of the most difficult parts was to adapt the data that were not designed for sequential data mining to a form, in which they could be easily processed by existing sequential data mining tools. As the best source for this task, data from the integration layer logs was considered. This data contains all the necessary information to be preprocessed into sequences. There is a strict naming convention of states, identification of sequence is done by database number given to a shipment and every single record has a timestamp when it was created. The identification is used throughout the system, so the details for a specific shipment can be selected from other databases. The final preprocessed dataset was in such form that it was easily used with most of the sequential data mining applications only by applying minor changes. The dataset used for sequential data mining contained exactly 6311 sequences, where each sequence represents a process of a shipment. Unless stated in the experiment differently, the default sequence windows is set to 1, which means that two states connected with arrow do not have any other states between them.

6.3. Data mining starts

Using mode pattern serial episode rule in dmt4sp tool it is possible to demonstrate its wide applications by setting the preferred length and confidence of the episode rules. Changing parameters enables to change focus from frequent to rare rules, discovering standard flow of the business process and also rare rules occurring in unusual situations.



Rule containing 2 ETA revisions

First task of the experiment was to find rules with low confidence which differ from the standard process model. Rule ending with shipment delivery above has been found only in 20 sequences. This means that 2 ETA (expected time of arrival) revisions for one shipment is a rare event that happens only in small amount of cases. Also ETA revision after the shipment has been already picked up means that something unusual happened in these cases.

Then I tried to find maximal subsequence in exported data. Various setting of the parameters where tried, starting always from the lower values and proceeding to the optimal one. With lower values, minimal length set to 3 and minimal occurrence set to 100, the output of the application contained a lot of sequences, from which it was difficult to extract those with relevant states for this task. On the other hand, when higher constrains were set (minimal occurrence = 2200, minimal length = 8), the output data contained a few sequences, and for even higher parameter settings the output was empty. After trying different options, the final configuration of parameters in dmt4sp was set these values: minimal occurrence is at least 1000 and minimal length of sequence at least 5 states. From this is it clearly visible that the most frequent shipment subsequence is as following:

This subsequence was expected to be found, and even the amount is corresponding to forecasts.

Another expected idea was that the shipment creation should always be before shipment confirmation. This is due to the design of the business model. But as it was found out, these two states are not always in the same window. According to the sequences, shipment can also be created using shipment confirmation message. If this message contains valid data, and corresponding shipment is not created, it will be created in state ready for pickup. This option is not described in the business model, and should be added.

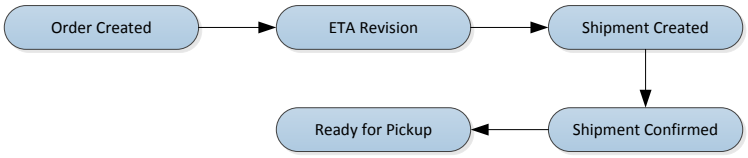
Regarding ETA revision message, one is default, which is sent after the shipment is confirmed. Therefore one ETA revision should not be taken as a special state, because it does happen in every sequence that gets to the state ready for pickup. This means that whenever shipment confirmation is done, ETA revision is expected to be the next action.

Focus was also given on windows between states pickup info entered and delivery, how big the window should be to guarantee confidence equal to 75 %. This experiment did not succeed, probably due to variability of production data.

6.4. Advanced techniques

6.4.1. Sequence discovery

The first main step of sequence discovery was to find a subsequence set that is frequent and does occur in normal shipment process situation. The subsequence to be found was expected to have very high support due to the robustness and quality of the business process. The candidates for most frequent subsequences are part of the process that are mostly automatized and do not depend on a human factor. Sequential Pattern Mining Framework offers variety of algorithms for discovering frequent subsequences. For this special task PrefixSpan was chosen, because of its simplicity, speed, easy defining of requested output and the readability of the output. First attempt started with support higher than 66%, which returned subsequences occurring in more than 2/3 of the processes, but as expected, the number of these subsequences was enormous. So the sequence mining support was set to higher level equal to 75%. This run of the PrefixSpan algorithm also provided almost one hundred subsequences, but these were mainly shorter variations of the longer found subsequence, which was decided to be the longest and the most supported at the same time.



Frequent subsequence with support 75%

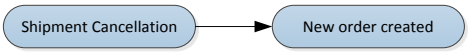
As it is clearly noticeable, this subsequence with support slightly over 78% starts with Order Created message from a customer. Before this event, an order operation can be expected, for example Order request from the customer and order confirmation. In this found subsequence, the only state that can be affected by the customer is the "Order Created", which is sent from customer's system. Once this state is processed in a warehouse, everything else is done by systems and employees of the warehouse. Process shown above is a typical shipment preparation in the warehouse. Firstly expected time of shipment arrival (ETA) is counted using the addresses in the shipment creation message. The shipment is officially created, provided with expected time of delivery, confirmed that it contains all requested parts and prepared for courier pickup. After this, it is just on couriers side to deliver it on time. This subsequence might look as something that has to happen for almost any shipment, but it is not so, situations like changed part requirements from customer, unavailability of the requested part in warehouse or shipment cancellation can happen.

After founding this frequent subsequence, it became rather interesting which are the subsequences that do not occur so often. The main aspect was mainly what happens when a shipment is cancelled, what are the next possibilities of the process. For this specific purpose, SPMF has built in item set discovery algorithm called Apriori Inverse. This algorithm needs to be provided with 2 values, which are the minimum support and the maximum support of an itemset. To find really the lowest values, the minimum limit was set to 0.01% and the maximum support was set to 1%. If the minimum itemset support was set to zero, the algorithm did not work correctly and provided invalid output. But in the case of bounds set to [0.00001; 0.001] the output consisted of four itemsets (only if item sets with length longer than one are counted). All of these item sets contained state "Shipment Cancelled".

Shipment cancellation on customer's request

First found itemset starting with the state Shipment Cancelled was found with support equal to 0.16%. Stated cancellation procedure happens, when customer declines the need of the to-be-shipped parts. In this case, the customer is acknowledged that cancellation request was successful and the shipments process ends without being actually sent.

Another kind of cancellation is caused by some unexpected event, error in system or by human mistake. For example part can be damaged while being handled, can get lost or just incorrect part is chosen from the warehouse.



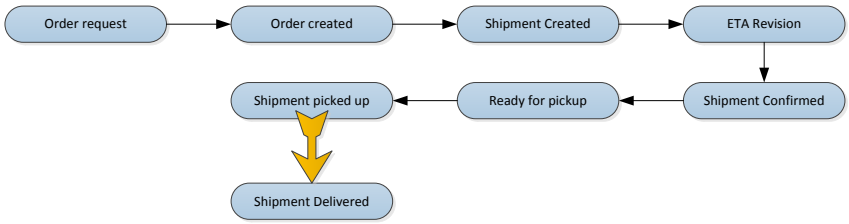
Shipment cancellation due to unexpected event

In this case new order is created from the original order and the process starts again from the early beginning. Support for this item set is as low as 0.11%. This support might look small at first glance, but when compared to the total number of delivered shipments and many years' experience in logistic it is understandable.

6.4.2. Episode rule discovery

Episode rules provide a closer look at how the states of shipment actually follow each other and how the shipments states fit in together till the shipment is delivered.

The main catalyzer for this part is to confirm that the business process that is known on manager's level is also the one present in a warehouse. To find this out, episode rule with high confidence ending in final state, state Delivered, has to be found. SPMF comes with built in algorithms for rule discovery, from which FP Growth was chosen. After choosing parameters minimum support set to 50% and minimum confidence set to 90%, the algorithm provided following rule as the best describing.



Frequent episode rule ending in state Delivered

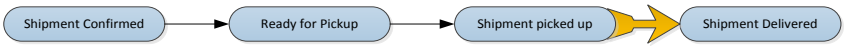
The rule above shows the whole shipment process in a high level, easy to understand view. It almost copies a part of the business process description, it can be said that this is the core and the most important part of the whole process. This assumption can be also proved by the confidence with which this episode rule was found, the confidence is equal to 92%.

One of the essential things that were in the scope of this episode rule discovery was to find episode rule that has extremely high confidence. This is supposed to prove that two such states exist, that after one comes the second one with almost 100% probability. To find this out, the confidence and support has to be set to great levels as much as 99%. The results were surprising, because following rule came out as the result.

Episode rule with support as much as 99.97%

After shipment is confirmed there is enormous 99.97% confidence that it will be picked up by a courier. To put it another way, this part of the process is done exclusively on warehouse location, often within one building. That means that process within part handling in warehouse is on a very high quality level, but now a lot more those 0.03% are very interesting to be examined.

To have a closer look what is the connection between the processes in warehouse and the delivery of shipment, windows size was included into episode rule discovery. Setting the window size to 2 and lower the confidence to 90% provided rule for successful delivery after there is no extra state done in warehouse.



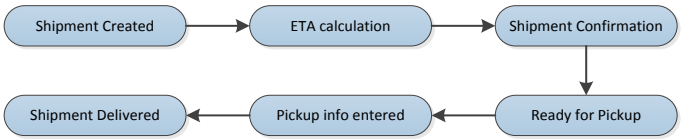
Frequent episode rule within window size 2

If handling of the shipment was done in an ideal way and the courier picked up the shipment from the dispatch area in time with no additional events, there is a 91% confidence that shipment delivery will be present in following two states. This comes from the window size set to 2.

6.4.3. Sequences by countries

Statistical analysis has revealed that 80% of the shipment deliveries happen in only three countries (see section 5.2.Statistical analysis). From sequential data mining point of view, it would be essential to discover how sequences differ between these countries. Also the differences between countries with big and small market share within DHL SPL section will be point of interest.

To identify similarities and differences across countries, same properties of sequential data mining have to be chosen for each country. To find out the most frequent subsequences, PrefixSpan algorithm was chosen with minimum support set to 66%. Then the algorithm ran with datasets of regarding countries. The longest found frequent subsequence was chosen for comparing with other countries.



The most frequent subsequence in countries with big market share

After the run of the algorithm for top 3 countries with the biggest number of delivered shipment, the most frequent subsequence was as shown above. There were no differences between these 3 countries, what was quite expected. These three countries use the same set of application and the business process does not differ at all.

The most frequent subsequences in countries with lower market share

"Small" countries that have less than 10 % of the global transferred shipments within the service parts logistics branch do not use order process before the shipment creation. It was found that order is sent as a part of shipment creation request. Also ETA (expected time of arrival) is calculated before the shipment creation. The subsequence above shows the most frequent subsequence in these "small" countries.

The difference found between countries with high and low market share show that the business process used does depend on the country market size. For smaller markets, more simple solutions can be used, because the business flow does not have to be so robust, whereas on bigger markets, every state of the shipment has to be well documented.

6.4.4. Applying outlier techniques

The next step after finding frequent episodes and episode rules was to concentrate on processes differentiating from them. To separate wanted sequences for further examination, simple outlier techniques were applied to the original dataset. An application in JAVA was developed to remove sequences containing frequent subsequences. This application stores data in Array List and searches for predefined subsequences. If sequence contains frequent subsequence found in the previous steps of Sequential data mining, whole sequence is not considered in the following outlier discovery.

Many different subsequences were tried to be removed, but finally the best results were created when firstly sequences with subsequence

Confirmed->Ready for Pickup-> Shipment Picked up

were removed, and after this the same technique was applied once again, but now all the sequences ending in state *Delivered* were deleted. The second action did not provide assumed upturn, so finally only the first thinning was applied. After acquiring the final dataset for research, SPMF tool was applied. Pruning out the dataset meant that it contained only sequences that did not belong to the majority of the original one. So the logic of finding requested results was the same as when the dataset was not pruned out. To put it another way, Sequential data mining was applied to special, non-frequent sequences. The final number of such sequences was around 498 out of original 6311.

Outlier dataset revealed processes that happened mainly because of:

- Wrong usage of applications
- Parts of business process designed for special purposes (Shipment Cancellation, Returning of a Shipment)
- Overloading of applications, which leads to prolonged message read time
- Possible bugs in applications

The first interesting sequence that was found probably happened due to the third reason, which is overloading of the applications. From the start the sequence looks ordinary, actually whole sequence would be probably pruned out if it was picked up. But it was not; instead delivery message came into the system. The problem is that the shipment cannot be delivered before it is picked up. In this rare case, when delivery was forced by the end user, the shipment delivery message was found in the integration layer exactly 8 times for the corresponding shipment. So the sequence looks something like this:

High level process: *Order –> Shipment –> Ready for pickup –> Delivery –> Delivery -> ..... -> Delivery*

Another interesting, but more frequent sequences have their states shuffled in illogical order or some of the states are missing. For instance a shipment can be delivered without calculated expected time of delivery or shipment is missing an order that would clarify what it has to contain. Such issues are quite frequent in the acquired dataset and these mistakes cannot be overlooked. The reason of their occurrence can be various, but from the research it is clearly visible that they happened mainly due to software discrepancies. Simply the integration layer timeout was not enough to handle real time data and the timestamps of corresponding states were postponed until the time another state, which should be later in the designed business process, was moved before its ancestor. This happens when a lot of messages are sent to one queue in the integration layer and the server is not able to deal with them instantly or the message is not read correctly so it has to be resent. This created some fuss in the potential outlier discovery, but by knowing the business process in detail such sequences can be instantly marked as unreliable and not taken into account.

Many processes are left in an unfinished state due to incorrect usage by end users. For instance Delivery is not entered into the system, so there are shipments that are physically delivered, but in the system they are shown as picked up or even ready for pickup. At first, it seemed that these are regular process that were not finished at the time of data extraction, but by looking into the original database and regarding time stamps, they were marked as non-finished processes. Described processes overload the database, because purging is set only for shipments in final states, which are delivered or cancelled. This founding opened a question whether purging is set correctly and/or whether users of applications are trained enough to work with them.

Returns process occurs when a customer does not have interest in the shipment after it was delivered because it is defective or it was used only once for analytical purposes. Returns process does also belong to the outliers of the dataset. 15 percent of the shipments within outlier dataset were actually returned with bad or good state. This number might look big, but one more time, it has to be considered that also good parts can be returned if they are not used. Such situation can occur when a server is broken, and it is not known which part should be replaced. Then more parts are ordered (processor, RAM …) and after the root cause was found, unneeded parts are returned back into the warehouse. Returns process does work perfectly, no issues with delivering returned parts into warehouses were found.

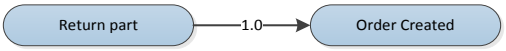
Expected time of arrival (ETA) revision is a default action when the shipment is being created. If it revised after pickup, something unexpected happened and it should be investigated. Therefore revising ETA after pickup requires reason code of this action, which must be entered. In serious cases, these revisions can be done even more than once. Naturally, for all the events new ETA should be provided. ETA revision happened in 44.58 percent of the outlying processes. The number is so high mainly because of the outlier dataset extraction method that is described in section above.

There is a 7 percent probability that after shipment creation it will be cancelled, within the outlier dataset. Such information was found by applying sequence window technique with focus on these two states. By selecting states with such a high percentage of cancelling, more attention could be given to such sequences.

The discovery in the following example shows the growth of confidence with respect to the window size between states Order acceptance and Shipment cancellation.

Window size	*Support*	*Confidence*
1	30	12.61%
2	36	12.13%
3	42	17.65%
4	48	20.17%

The growth indicates that shipment cancellation has no predefined position in the process and can happen suddenly in various cases. It was also found, that shipment cancellation is not affected by previous states and therefore cannot be predicted in an early phase of shipment process.



Returns process was found only with specific state occurring after it, state Order Created. This means that every part that is requested to be returned back to a warehouse is returned there with 100 % probability. DHL simply needs back the good unused parts or parts that were broken and have to undergo claim.

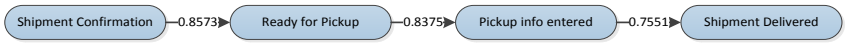
6.5. Markov chain

After evaluating output of the sequential data mining output some skeptic questions came in mind:

- Is this process random and can be characterized as memory less?
- Does the next step depend only on the current one or does it depend also on states that precede it?
- Can this process be identified using Markov chain?
- Does Markov chain represent the process in a better way than output of Sequential Data mining?

Firstly the task has to be converted to a Markov chain model. Since this is a sequence task, the aim of interest is transition probability between states. Therefore a transition matrix, present in Appendix B, was created, saying what the probability of getting from state A to state B is. For filling in the table an application in JAVA was developed. To explain how it works, let's say we have states A and B. The probability of transition from state A to state B was computed as result of occurrence B right after A divided by the total count of state A occurrence. This application enabled quick and easy filling of the table within extracted data.

Then one of the questions could be answered, yes, the task can be adapted to Markov chain, but is it really worthy representation? To answer this question, independent probability of some chosen frequent subsequence was counted. For instance, let's look at the transition probability between states "Confirmed", "Ready for Pickup", "Picked up" and "Delivered"



Afterwards a first order probability with transition from the state "Confirmed" to "Delivered" was computed. The sequence memory at level one needs to be confirmed or disproved.

According to these equations

$$P((SC \to RP) \wedge (RP \to PE) \wedge (PE \to SD)) =$$
$$P((SC \to RP) \cap (RP \to PE) \cap (PE \to SD)) =$$
$$P(SC \to RP) * P(RP \to PE) * P(PE \to SD) =$$
$$0.8573 * 0.8375 * 0.7551$$

the first order probability of the occurrence of these three transitions is 54.21%.

Result value using Markov model was then compared with the actual probability computed from dataset. The probability of states above occurring in respectable order was divided by the total occurrence of the state "Shipment Confirmation" followed by any

three states. Found likelihood from dataset was 67.55%, what is not a big difference, but still it can be said that the following state not only depends on the previous one, but on more states preceding its occurrence.

Just for curiosity, the same method was to be applied to outlier dataset. The same subsequence was chosen to be identified. Found difference between computed first order probability and actual probability from dataset was not as huge as in regular dataset. Computed from dataset, the probability was 75.32% whereas the Markov model first order probability value was 76.16%. From the small difference, it can be concluded, that the outlier dataset is closer to memory level 1 than the original one.

From the findings above, Sequential Data mining can be considered as more complicated, but still, more enhanced method of describing the shipment sequences. Markov chain has the positive of complete process overview in a relatively short time, but does not take the dependency of states into account. More on this issue can be read in the conclusion.

7. Conclusion

7.1. Comparison of attempts

To work with extracted data, three main attempts were tried. Sequential data mining was done using two different tools, and in the end, Markov chain was applied. All of these attempts have specific features which make each one more or less suitable for given task and given data.

DMT4SP tool was a great thing to start with. It comes with elementary sequential data mining features. While not so complicated but robust, it provides parameters for adjusting the results found. Despite the fact that it has only one algorithm for each data extraction technique, it was the best when it comes to speed and has one great advantage. Unlike other tool used, it can natively work with timestamps that came with data. Although all its minor negatives do not affect the work much, there is one that does. The output is a text file with immense formatting and plenty of unneeded characters. This makes it really hard to be postprocessor because of text formatting and big size of the output file. However it still produced adequate results and helped to understand sequences and all its properties.

The second attempt after better understanding the data and the sequential data mining principles was done using Sequential and Pattern Mining Framework (SPMF). Before even using it, I was amazed by the variety of styles it can be used, for instance command line interface, as a framework for JAVA application to a standalone application with graphical user interface (GUI). It comes with detailed documentation and a big set of examples that really make the work easier. When first starting the SPMF GUI, it is noticeable that many algorithms for each sequential data mining task are present. Each part even has a group of implemented algorithms. I stayed with using traditional algorithms for each discipline, for example PrefixSpan for sequential pattern mining, Apriori for Itemset mining and RuleGrowth for sequential rules mining. SPMF is fast, but what helped me more is that it comes with option to make SPMF complaint dataset from almost any extracted set from a database. Also the output was easily readable and perfect for further post processing. The biggest, and the only negative found from my perspective was that SPMF does not support timestamps within states. This removed data properties that were available and could have improved the results, but on the other hand, results would not be so specific and easily accessible if this tool was not used.

Markov chain was the last method used that came into considering after finishing sequential data mining. There was a possibility that sequential data mining is unnecessarily complicated method and that the dataset and the whole process can be better described by probabilities of state transitions. The key point was to find, whether the states have memory equal to one or not. In the end, sequential data mining was considered as better approach, but the advantages of likelihood transition matrix were appreciable. Markov chain provides a quick overview of the process behavior, which is not as exact as mined sequences, but can be made much faster with less understanding of the dataset. It may also identify discrepancies in the flow and provide an easily understandable output.

7.2. Analysis of results

To sum up, the benefit of this project was enormous for me. Not only have I tried something new, but I have gone deeper into the topic of sequential data mining than I was expecting in the early stages of the project. There are numerous advantages of such project for my further studying and career. I have understood the business process that I work with almost every day on a much deeper level, got to know its hidden parts and create an overview of its weaknesses as stated in the experimental protocol. The next, even bigger, advantage is that I succeeded applying sequential data mining to a task, which has never been designed to it. Applying ideas and found tools to such a problem of the logistics industry looked as a complex task at start, but I managed to use gained knowledge and software to make it possible and provide results and relevant conclusions. Many outputs of the applied techniques are visible in the experimental protocol section of this document. Out of these found sequences, I created a single transition diagram that can be found in the *Appendix B section*. The transition diagram can be considered as a representation of the business process found by sequential data mining. It contains all the necessary states and directed transitions showing the flow of the process. The diagram was later compared with the business process description (*see section 4.1. Business process description*). The first visible difference between the official description and the one found by sequential data mining is that the created transition diagram is more detailed and describes the process on a more detailed level, whereas the official description gives only high level overview. If the question is, whether each process from the diagram is present in the official description, the answer would be yes. For example the order handling process is not described so thoroughly, but the created diagram broadened the process description and provided more states of the order, so it became rather detailed.

Although the task terminated successfully giving numerous achievements, there occurred several problems. The initial setback was gathering the data from the application database and storing it in defined form in a custom made database. The problem was, that each state of shipment used different identification to which process it does belong. No documentation was found for connecting states with different identification within the same process together, therefore some testing was needed to link states to a processes. This led to the final storing design, where all the states were linked to some process based on identification uniting within all states. Another setback that brought necessary feedback in the end was to identify whether unexpected sequences that do not meet process definition standards are mistakes in data extraction, malfunctions in the application or processes that are allowed but not described. Data extraction mistakes were completely removed as the project was growing. The last problem, or can be called also a dilemma, is how to interpret the findings so they could be easily readable and understandable for wide management and development teams in the organization. I decided that the best idea is to use flow charts and my assumptions were correct, nobody had problem understanding the findings and they were described as perfectly understandable.

If I was to name several achievements that positively helped the company, the transition diagram creation would be definitely one of them. All the positives were already said before in previous sections. What can be stated as another big achievement is that malfunctions of applications were found and due to the good communication with the development team, they were also removed. The most important and expressive is the following one. A process was found, where after the shipment has been picked up a shipment confirmation occurred. Shipment confirmation is forbidden in such part of the process, but the behavior was that the shipment's state changed back to confirmed and all previously changes were discarded. This mistake was found within outlier detection and was successfully fixed.

7.3. Future work

The proposed solution on applying sequential data mining on logistic data definitely solves all the tasks provided in the assignment and personal goals. However, there are some ways that could make it a more advanced and more frequently used solution. Some of the findings, especially the statistics of the production data, can be automatized to run on regular basis and used in automatically generated reports. Reporting them on regular basis could help the decision-makers to curve the business strategies to requested paths. Sequential data mining can also be partly automatized, but I find its strong points mainly when deciding where the application development should aim. On the other hand, such decisions are not made on daily basis, so such analysis of production data can be made before bigger business decisions or application updates.

The part of the solution, which definitely needs to use some smarter handling, is outlier detection. Since this was not a main task in my project, I used only simple self-designed outlier algorithm that is suitable, but not optimal. Several more advanced algorithms do exist, from which it could be possible to apply distance based model, distance K-based model or statistical method. These algorithms will provide more appropriate output with regards to the quantity and quality.

Since the task is specifically made for DHL and its applications, it would be certainly an advantage for the whole industry of logistics to implement intelligent algorithms for business decision making. This would lead to lower spending and more effective operation of the whole industry. Definitely, sequential data mining is emerging discipline which application to business decisions will become more and more frequent.

References

[1] Agrawal, R.; Srikant, R.: Mining Sequential Patterns, Proc. 1995 Int'l Conf. Data Eng. (ICDE '95), pp. 3-14, Mar. 1995.

[2] Srikant, R.; Agrawal, R.: Mining Sequential Patterns: Generalizations and Performance Improvements. In Proc. 5th Int. Conf. Extending Database Technology, EDBT, Vol. 1057 (FebruaryMay--FebruarySeptember~ 1996), pp. 3-17, 1996.

[3] Pei, J.; Han, J.; Mortazavi-Asl, B.; Wang, J.; Pinto, H.; Chen, Q.; Dayal, U.; Hsu,M.C.: Mining Sequential Patterns by Pattern-Growth:The PrefixSpan Approach; IEEE Transactions on knowledge and data engineering, vol. 16, no. 10, October 2004.

[4] Mannila, H.; Toivonen, H.; Verkamo, A.: Discovery of Frequent Episodes in Event Sequences. Data mining and Knowledge Discovery 1, Kluwer Academic Publichers, pp. 262, 1997.

[5] Mannila, H.; Toivonen, H.; Verkamo, A.: Discovery of Frequent Episodes in Event Sequences. Data mining and Knowledge Discovery 1, Kluwer Academic Publichers, pp. 261, 1997.

[6] Knorr, E.M.; Ng, R.T.: Algorithms for Mining Distance-Based Outliers in Large Datasets. Department of Computer Science, University of British Columbia. 1998.

[7] Breunig, M. M.; Kriegel, H. P.; Ng, R. T.; Sander, J.: LOF: Identifying Density-Based Local Outliers, Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (2000), pp. 93–104, 2000.

[8] SPMF: A Sequential Pattern Mining Framework, http://www.philippe-fournier-viger.com/spmf.

[9] Shared Services & Outsourcing Network: e-Billing in DHL. Amsterdam. May 2011.

[10] Tan, T.C.Y; MISRA, A.H.; Ji, J.J.Y.: DHL Data Mining Project: Customer Segmentation with Clustering. March 2010.

[11] Lattner, A.D.; Bogon, T.; Schumann, R.; Timm, I.J.: Temporal Pattern Mining in Logistics. Johann Wolfgang Goethe-Universität Frankfurt, Institure of Computer Science, Information Systems and Simulation, May 2008.

[12] Allen, J.F.: Maintaning Knowledge about Temporal Intervals. The University of Rochester. October 2005.

[13] Van der Aalst, W.M.P.; Reijers, H.A.; Weijters, A.J.M.M.; van Dongen, B.F.; Alves de Medeiros, A.K.; Song, M.; Verbeek, H.M.W.: Business Process Mining: An industrial application. In Information Systems**,** Elseiver Ltd. pp. 713-732, July 2007.

[14] Van der Aalst, W.M.P.; Weijters, A.J.M.M.: Process mining: a research agenda. In Computer in Industry, Elsevier B.V. pp. 231-244, April 2004.

[15] Sun, P.; Chawla, S.; Arunasalam, B.: Mining for Outliers in Sequential Databases. The School of Information Technologies, University of Sydney, May 2009.

[16] Hodge, V.J.; Austin, J.: A Survey of Outlier Detection Methodologies. Artificial Intelligence Review, Kluwe Academic Publishers, January 2004.

[17] Janeja, V.; Vijayalakshmi, A.; Adam, N.R.: OUTLAW: Using Geo-Spatial Associations for Outlier Detection and Visual Analysis of Cargo Routes. MSIS Department and CIMIC Rutgers University, 2002.

[18] dmt4sp, http://liris.cnrs.fr/~crigotti/dmt4sp.html.

[19] The OLAP Council: OLAP AND OLAP Server Definitions. The OLAP Council, January 1995.

[20] Himalaya Data Mining Tools: SPAM, http://himalaya-tools.sourceforge.net/Spam.

[21] Martin Ralbovsky: Ferda Data Miner, http://ferda-data-miner-x64.smartcode.com/info.html.

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

BACHELOR PROJECT ASSIGNMENT

Student: Filip Mihalovič

Study programme: Open Informatics

Specialisation: Computer and Information Science

Title of Bachelor Project: Analysis and Sequential Mining of Logistic Data

Guidelines:

1. Get familiar with the topic of sequential data mining and its available tools.
2. Examine the available corporate logistic data from integration layer.
3. Carry out basic statistical analysis of data ad 2.
4. Apply the sequential tools ad 1 to the logistic data ad 2.
5. Evaluate the found patterns and episodal rules, categorize them, discuss namely their practical applicability for definiton of standard shipments and identification of the nonstandard ones.

Bibliography/Sources:

[1] Rajaraman, A.; Leskovec, J.; Ullman, J. D.: Mining of Massive Datasets. Cambridge University Press, 2012.
[2] Mannila, H., Toivonen, H., Verkamo, A: Discovery of Frequent Episodes in Event Sequences. Data Mining and Knowledge Discovery 1, Kluwer Academic Publishers, pp. 259–289, 1997.
[3] Srikant, R., Agrawal, R.: Mining Sequential Patterns: Generalizations and Performance Improvements. In Proc. 5th Int. Conf. Extending Database Technology, EDBT, Vol. 1057 (FebruaryMay--FebruarySeptember~ 1996), pp. 3-17, 1996.
[4] SPMF: A Sequential Pattern Mining Framework, http://www.philippe-fournier-viger.com/spmf.

Bachelor Project Supervisor: Ing. Jiří Kléma, Ph.D.

Valid until: the end of the winter semester of academic year 2013/2014

prof. Ing. Vladimír Mařík, DrSc.
Head of Department

VYSOKÉ UČENÍ TECHNICKÉ V PRAZE Fakulta elektrotechnická

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, January 10, 2013

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Filip Mihalovič

Studijní program: Otevřená informatika (bakalářský)

Obor: Informatika a počítačové vědy

Název tématu: Analýza a sekvenční dolování logistických dat

Pokyny pro vypracování:

1. Seznamte se s problémem a základními nástroji dolování sekvenčních dat.
2. Seznamte se s povahou reálných firemních logistických dat z integrační vrstvy.
3. Proveďte základní statistickou analýzu dat ad 2.
4. Aplikujte vhodné nástroje sekvenčního dolování na logistická data ad 2.
5. Nalezené vzory a pravidla přehledně kategorizujte, posuďte jejich praktickou využitelnost pro definici standardních a vyhledávání nestandardních zásilek.

Seznam odborné literatury:

[1] Rajaraman, A.; Leskovec, J.; Ullman, J. D.: Mining of Massive Datasets. Cambridge University Press, 2012.
[2] Mannila, H., Toivonen, H., Verkamo, A: Discovery of Frequent Episodes in Event Sequences. Data Mining and Knowledge Discovery 1, Kluwer Academic Publishers, pp. 259–289, 1997.
[3] Srikant, R., Agrawal, R.: Mining Sequential Patterns: Generalizations and Performance Improvements. In Proc. 5th Int. Conf. Extending Database Technology, EDBT, Vol. 1057 (FebruaryMay--FebruarySeptember~ 1996), pp. 3-17, 1996.
[4] SPMF: A Sequential Pattern Mining Framework, http://www.philippe-fournier-iger.com/spmf.

Vedoucí bakalářské práce: Ing. Jiří Kléma, Ph.D.

Platnost zadání: do konce zimního semestru 2013/2014

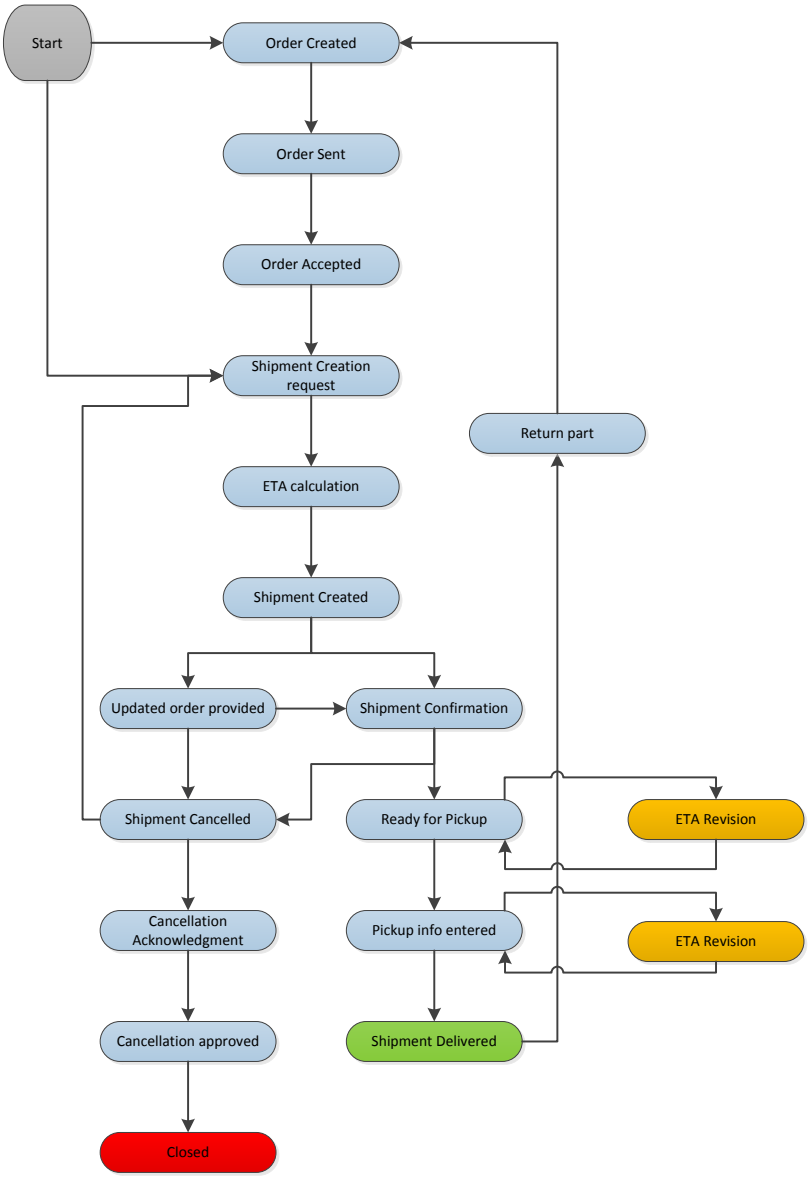
L.S.

prof. Ing. Vladimír Mařík, DrSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 10. 1. 2013

Appendix B – Transition diagram



Appendix C – Markov chain transition matrix

from\to	Order Created	Order Sent	Create Order Failure	Shipment Creation request	ETA Revision	Shipment Created	Order Accepted	Shipment Confirmed	Ready for Pickup	Pickup info Entered	Delivered	Return part	Stock not available (Order related)	Cancelled	Updated Order provided	Stock not available	Terminal state
Order Created	0	0.267674	0.024825	0.262	0.03909	0.03215	0.227585	0.011025	0	0	0	0	7.71E-04	0	0	0	0.134916
Order Sent	0.294937	6.02E-04	0.021061	0.0265	0	0	0.656924	0	0	0	0	0	0	0	0	0	0
Create Order Failure	0.432335	2.46E-04	6.57E-04	0.2899	0.01075	0.00673	0.249159	0.009356	0	0	8.21E-05	8.21E-05	4.10E-04	0	3.28E-04	0	1.89E-15
Shipment Creation request	0.04754	0	0.026096	0	0.21492	0.22477	0.42053	0.066067	0	0	0	0	0	0	0	0	7.88E-05
ETA Revision	0.009805	6.95E-05	0.00459	0	0.04764	0.38769	0.019541	0.362796	0.012726	0.072531	0.081015	1.39E-04	0	8.34E-04	5.56E-04	0	6.95E-05
Shipment Created	0.019613	0	0.008165	0	0.47675	0	0.029553	0.436723	0.020501	0.006834	1.77E-04	0	0	0.001154	5.32E-04	0	0
Order Accepted	0.042637	0	0.038961	0.4277	0.21335	0.17912	8.17E-05	0.097852	0	0	0	0	0	0	3.27E-04	0	0
Shipment Confirmed	0.088389	0.001446	0	0	1.61E-04	0	0	4.02E-04	0.857372	5.62E-04	0	0	0	0	8.04E-05	0	0.051587
Ready for Pickup	0	0.001644	0	0	0.12089	0.01257	9.67E-05	9.67E-05	2.90E-04	0.837524	0.008607	0	0	0	2.90E-04	0	0.017988
Pickup info Entered	0.016912	1.97E-04	0	0	0.14031	0	0	9.83E-05	1.97E-04	0	0.755064	0	0	0	9.83E-05	0	0.087119
Delivered	0	0.034722	0	0	0.00694	0	0	0.006944	0	0.17361	0.5625	0	0	0	0.006944	0	0.208333
Return part	0.806452	0	0.184588	0.009	0	0	0	0	0	0	0	0	0	0	0	0	0
Stock not available (Order)	0	0	0.027027	0.2162	0	0	0.297297	0	0	0	0	0	0.459459	0	0	0	0
Cancelled	0	0	0	0.25	0	0	0	0	0	0	0	0	0	0	0	0	0.75
Updated Order provided	0	0	0	0	0	0	0	0	0	0	0	0	0	0.393939	0	0	0.606061
Stock not available	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1