DEVELOPING AN AUTOMATED SOLUTION FOR ETA DEFINITION CONCERNING LONG DISTANCE SHIPPING

VELDHUIS, H.D.

S1231685

TECHNISCHE BEDRIJFSKUNDE

Behavioural, Management and Social Sciences

First Supervisor

DR. M.E. IACOB

Second Supervisor

A. DOBRKOVIC MSC

Company Supervisor

ING. S. PIEST

UNIVERSITY OF TWENTE.

19-08-2015

Management Summary

On behalf of Cape Groep a research on automating estimated time of arrival definition for deep sea vessels has been executed. This research was built around a case study from HST Sea- & Airfreight, which is a partner of Cape Groep and is concerned with tracking their freight which mostly travels by sea. The results of this research can be found in this report. The research question around this research was build is:

"*How can a real time track and trace and prediction system for deep sea vessels be developed?"*

Based upon this research, it is concluded that there are two methods to effectively create an automated solution for this question. At this point, HST tracks ships transporting their freight via a number of websites which list information on this subject. Data is collected from these websites and processed within their current IT system. The data collection process could be automated by using web scraping, a technique which simulates human surfing behavior and allows one to automatically receive selected data from websites based upon certain input values.

Furthermore arrival times are often calculated by using historical data. By modeling a ship's route and split this route into parts between various ports, one can calculate the time spend on a route by defining the time needed to cover the distance between these ports and the time spent at these ports. These times can be calculated by using the mean of previous times at which these routes were covered.

By combining these two methods, HST could calculate the estimated time of arrival of these ships in two ways, which results in a way to estimate the reliability of an estimated time of arrival. An application which collects data from various websites based upon one input value has been developed and it is expected that this system could be linked to the current IT infrastructure at HST, so that these input values can be generated automatically and human interference can be minimized in the process of defining estimated arrival times. Not only does this save time, this also allows for automated collection of historical data so that this historical data could be used to calculate estimated times of arrival in the future, alongside the use of web scraping. Moreover, this data might hold interesting information on punctuality of shipping companies and on the influence of for instance certain weather on arrival times.

Preface

I hereby present you the results of the research I have conducted for my bachelor thesis, of which I hope it concludes my bachelor Industrial Engineering & Management at the University of Twente. On behalf of Cape Groep I have examined the possibilities for a track and trace system for deep sea vessels, applied on a case study around their partner HST Sea- & Airfreight.

During my bachelor Industrial Engineering & Management I have often wondered what aspects of this bachelor I would really like to focus on in any future job or education, in which the more IT targeted aspects were surrounded by big question mark. I simply did not know whether this would be a possible job direction I would enjoy. My bachelor thesis seemed like an ideal possibility to try to answer that question. Cape Groep provided me with an interesting research question on this subject, which also had some common ground with other aspects of the Industrial Engineering & Management bachelor, such as logistics. I greatly appreciate the fact that Cape Groep has given me this chance, despite my lack of extensive knowledge in this area.

Besides this I would like to thank my supervisor from the Unversity of Twente, Maria Iacob for bringing me in touch with Cape Groep and for critically evaluating the choices I made, whether these were on structuring of my thesis or on the master I was planning to follow after passing my bachelor. I would also like to thank my second supervisor Andrej Dobrkovic for providing me with some useful literature, for reading through concept versions of my thesis carefully and for providing me with positive yet critical feedback.

Furthermore I would like to thank Sebastian Piest, who fulfilled the role of supervisor within Cape Groep really well and helped me especially when getting started with my research. Last but not last I would like to thank Samet Kaya for his help with configuring eMagiz and María Solórzano for her help with my Mendix project, as well as all other colleagues at Cape Groep for answering small questions and the cheerful time.

Enjoy reading this report!

Rick Veldhuis
19-08-2015, Enschede

Contents

List of abbreviations

AIS	Automatic Identification System
API	Application programming interface
CSS	Cascading Style Sheets
DSRM	Design Science Research Methodology
ETA	Estimated Time of Arrival
ETD	Estimated Time of Departure
GUI	Graphical user interface
HST	HST Sea- & Airfreight Division
HTML	HyperText Markup Language
IT	Information Technology
REST	Representational state transfer
URL	Uniform Resource Locator
XPath	XML Path Language
XSLT	Extensible Stylesheet Language Transformation

1. Introduction

This section will introduce and describe the research conducted as well as the structure of this report. After a short introduction on Cape Groep, the company on behalf of which this research was conducted, the motivation and goal of the research will be described, followed by the research question, sub questions and hypotheses. After this the scope of the research will be described, the methodology used will be pointed out and finally the structure of this report will be outlined.

1.1 General

Cape Groep B.V. is a company located In Enschede, the Netherlands, specialized in integrating information technology (hereafter: IT) solutions. One of their specialties is supply chain solutions, making sure that these solutions give full control and that communication among various systems used is optimized by matching the in- and outputs of these systems. HST, one of Cape Groep's partners, tracks their orders travelling by sea to estimate whether they will arrive according to their planning in order to properly schedule further travelling of the freight.

1.2 Research motivation

Lee and Meng (2015) start their book "Handbook of Ocean Container Transport Logistics" with the following sentences: "As international trade continues to grow rapidly and supply chains become more globalized, many operations have been outsourced and moved offshore. About 90% of the international trade volume was facilitated by ocean transportation". Add to this that "fast and accurate calculation of the estimated time of arrival (ETA) (…) is of high importance in several areas of the ocean shipping industry" (Fagerholt et Al., 2000) and you will find that a lot of attention for the ETA is required within the shipping industry.

As a ship leaves for its destination, a first value for the ETA is estimated. However, under the influence of various (external) factors such as weather influences, this ETA is adjusted during the trip. In practice, this means that only 52% of the vessels arrive on time (Vernimmen et Al., 2007). Other surveys talk about average schedule deviations of one till one and a half day (Lee et Al., 2015). This uncertainty for arrival times creates instability within the supply chain, causing for instance subsequent transport by trucks to be delayed and leaving drivers waiting. Besides this, the uncertain arrival times cause for extra safety stocks to be kept at companies, resulting in extra inventory costs or lower service levels.

This research is conducted on behalf of Cape Groep, looking for a solution for ETA definition at HST. HST currently defines the value of the ETA by checking the location of a ship at various moments during the trip and comparing these locations to shipping schedules, which are collected from the shipping company. All of this is done manually by using data provided by a number of commercial websites, which will be further elaborated on in Chapter 2. However, within an ever growing industry this takes up an ever growing amount of time, demanding automated processes to take over in the future. Therefor this research will look into ways to automate this process. Besides this, both the University of Twente and Cape Groep take part in a project called "SynchromodalIT", which tries to develop synchromodal logistics by incorporating ICT. This research fits well within the context of this project as well.

1.3 Research Goal

The main goal of this research is to design an automated solution for tracking and tracing HST's sea freight in order to define the ETA as precise as possible. To reach this goal effectively and efficiently, three separate parts of the research have been defined: firstly a theoretical model has to be developed, which will form the base for an application, secondly a prototype of this application has to be designed and thirdly this prototype has to be tested and evaluated.

The application should provide reliable ETA values, or provide ETA values combined with some sort of source for estimating the ETA's reliability. To ensure a useful end product for HST, the application has to be able to operate together with current IT systems at HST. Successful implementation of this application will result in more efficient ETA definition, as this process demands less man hours and possibly provides better results with the use of this application.

1.4 Research Questions

The research goal above has to be defined more concretely in order to translate into valid inputs for the research, which is done by translating the research goals into research questions. First the main goal of the research is translated into a question, which is formulated as follows:

"*How can a real time track and trace and prediction system for deep sea vessels be developed?"*

To be able to answer this question, four sub questions have been defined. These questions are structured following the three steps as brought up in the previous section, complemented with a section on data collection.

1.4.1 Theoretical model

The first question to be asked is how an ETA can be calculated. This algorithm will later on be the underlying calculations the application will execute:

1. How can one efficiently calculate the estimated time of arrival using an algorithm?

1.4.2 Data collection

After a theoretical model has been build, a clear overview of the data required by this algorithm is needed. In order to create a (close to) real time application, (close to) real time data should be incorporated. It is important to collect data from various sources and translate them into input data for the application which is going to be built. Combined, this results in the following question:

2. What data sources are available and how can data be collected from them?

1.4.3 Developing

As also defined in Section 1.3, the next step after this is incorporating this algorithm together with a technique on collecting the data into a prototype application. As a question, this looks as follows:

3. How can this algorithm be implemented in a technical solution and what applications and/or software is needed for this?

1.4.4 Testing

Last, but most certainly not least, this prototype has to be tested effectively, to check whether it provides the functionalities required. Simply translated into a question, this looks as follows:

4. How can this technical solution be tested effectively?

1.5 Hypothesis

Initially, a program called RapidMiner was expected to provide the functionalities desired, so the hypothesis is that the final result is a prototype created with RapidMiner. This expectation is based upon expertise within Cape Groep. It is also expected that the use of this program can significantly improve the current process of ETA and ETD definition in terms of invested manhours.

1.6 Scope

In order to conduct this research efficiently and within the time available, a clear scope for the research is required. One way of scoping has been discussed earlier in this chapter, as a case study for HST will be focused on. This means the results of this research may be useful for others as well, however that this is not the main intention of this research. Besides this, airfreight and transport by train or truck will be considered out of scope. Furthermore we assume that the data used is valid data, without checking whether this really is the case.

By incorporating a model into an application without taking our assumptions into consideration, we can create a starting point. With a working prototype finished, the effects of these assumptions can be researched later while using the application, which is why the scope is defined this way.

1.7 Methodology

Within this report, a Design Science Research Methodology (DSRM) was required, as the research question (which is defined in Section 1.4) is a design question. The methodology used is described in "A design science research methodology for information systems research" by Peffers et Al. (2007). This methodology is often used in design researches and fits this report better than business methodologies, as the aim within this report is the development of an information system and thus covers an information science problem rather than a social science problem. This DSRM lists 6 important steps, which are:

1. Identify Problem & Motivate;
2. Define objectives of a solution;
3. Design & Development;
4. Demonstration;
5. Evaluation;
6. Communication.

A graphical overview of the different steps within this methodology is shown in Figure 1.1.How these steps can be found within this report is described in Section 1.8.

1.8 Structure

This report consists out of six chapters, which all correspond with different parts of Peffers' research methodology. To enhance the ease of reading, the chapters of this report will be listed below, including a short description on what information they will provide and how that corresponds with Peffers' methodology.

Chapter 1: Introduction

This chapter covers most of the first step of the Peffers' methodology and describes the problem identification and motivation.

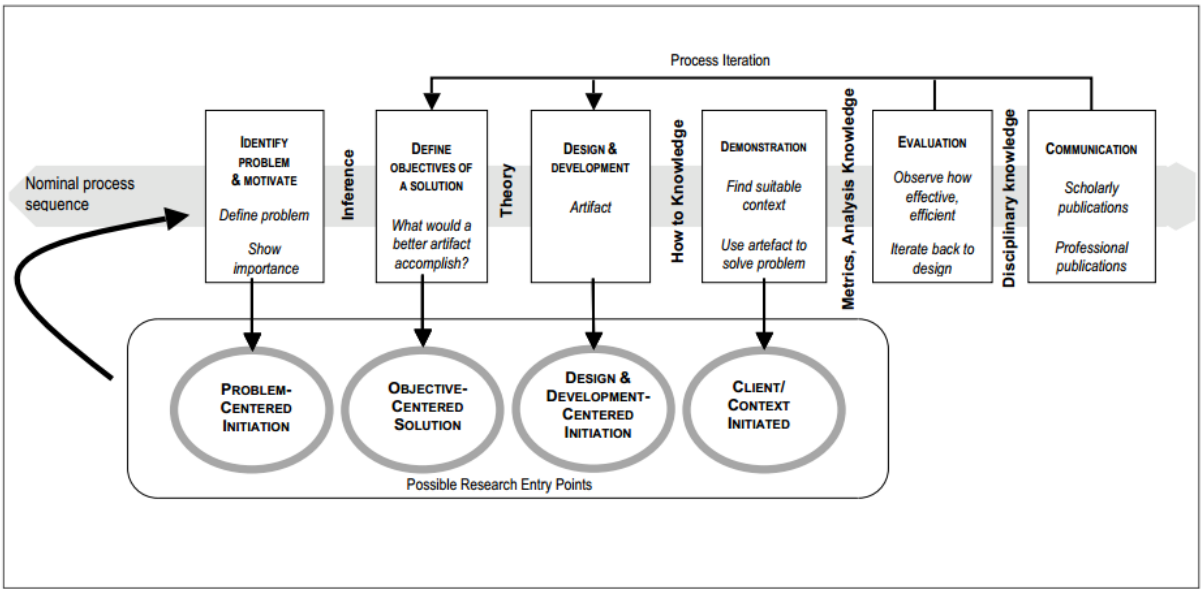

Figure 1.1 Graphical overview of the DSRM by Peffers et Al. (2007)

Chapter 2: Literature review

This chapter lists several findings based upon research publications as well as findings based upon interviews and/or observations and explanation on programs and data used. This corresponds mainly with the second step of Peffers' methodology, although some parts of step one and step three are also included.

Chapter 3: Design and development

This chapter describes the process of thinking out and building a prototype of the application. This corresponds with the third step of Peffers' methodology.

Chapter 4: Demonstration

This chapter shows how the application works in practice. This corresponds with the fourth step of Peffers' methodology

Chapter 5: Validation

In this chapter it will be discussed whether the designed application meets the objectives as described earlier in the report. This corresponds with the fifth step of Peffers' methodology.

Chapter 6: Conclusions and recommendations

This chapter will summarize the conclusions which can be drawn from this research and will denote possible recommendations for future work. Besides this, recommendations for future research will be done. This corresponds with the sixth and final step of Peffers' design methodology.

2. Literature review

This chapter lists several findings based upon research publications as well as findings based upon interviews and/or observations and explanations on programs and data used. First, the current situation of ETA definition at HST is described, after which several theoretical methods to define the ETA will be outlined and programs and data used will be explained.

2.1 Current situation

Within this section, the current situation of estimating the arrival times within the case study is described. This description is based upon an observation and interview of which the questions used can be found in Appendix II. As a result from this the objectives of a solution will be defined.

2.1.1 Observations

HST currently processes orders which will be shipped from point A to point B, including stops along the way. A shipping schedule is collected from the shipping company executing this order. All the orders which are in process are listed within a Microsoft Dynamics environment. Employees check whether these orders are still travelling according to their initial planning daily and save this information within the system. This is done for both the ETA as for the ETD (Estimated Time of Departure), for which different methods are used. This ETD is also important, as a ship needs to leave a port at a certain point in time to arrive in time at the next port and as possible delays will lead to more delays in the future almost inevitably (Heywood et al., 2009).

To determine the ETA, the websites of the terminals in Rotterdam are used in combination with the names of the vessels which are listed within the Microsoft Dynamics environment. An employee copies the name of the vessel and then searches for this name at ECT.nl (see Figure 2.1), as it lists most arriving vessels. If the ship is found, he checks the ETA listed at this website and compares it to the ETA listed in their system. Afterwards, the results of his check are processed within HST's IT infrastructure. Some ship names are not listed at ECT.nl. Whenever this is the case the website of another terminal is used, which is APMTRotterdam.nl. Whereas the search method at this website is a bit different, the employees' process is the same. If ECT.nl does list the data needed, APMTRotterdam.nl is not consulted, so there is no chance of contradicting data.

Terminal	Object	Voyage DisLoa	Operator	Arrival Date	Status	Departure Date	Modality	Cargo-Window
EMX	HANJIN GOLD	J194X / J194X	HJS	07 May 2015 23:00	Expected	09 May 2015 17:00	Deepsea	yes
EMX	HANJIN GOLD	J314X / J314X	HJS	23 Jul 2015 23:00	Expected	25 Jul 2015 14:00	Deepsea	no

De ETA en ETD van feeder reizen op terminal ECT zijn de aangevraagde en onbevestigde tijden.
The ETA and ETD for feeder calls at terminal ECT are requested and unconfirmed times.

Figure 2.1: an overview of the data provided by ect.nl

The process of determining the ETD is a bit more complex as it involves the use of more websites. To check whether a container will still be picked up according to its planning, employees will first check the location of ship, where the container will be loaded onto, using a close to real time track and trace method via MarineTraffic.com. The ship's location can be used to define whether a certain port is visited or not. If the ship's next stop is the port where the container will be picked up, MarineTraffic lists all the data needed, otherwise the ship's planning is needed, which can be found on either Hanjin Shipping's website or Hapag-Lloyd's website. Once again, the processed information is used to determine whether the ship is still travelling according to the initial planning and this information is processed. Unlike with determining the ETA, this process can sometimes require a bit of expertise to estimate whether the ship is really travelling according its planning as this should be based upon locations.

Whenever the initial planning is not up to date anymore, this is inputted into Microsoft Dynamics. The new given ETA or ETD is then used, for which HST relies on the expertise at the shipping company. If no new ETA or ETD is given, HST calls the shipping company.

To summarize the rather abstract description above, we define the following activities for ETA processing:

A. Collect ship name from Microsoft Dynamics;
B. Fill in name at ect.nl and collect data;
C. Check if sufficient data is available;
D. Find ship via apmtrotterdam.nl and collect data (only executed when the result of C is negative);
E. Compare data to data in Microsoft Dynamics;
F. Process whether ETA within Microsoft Dynamics is still valid or not valid anymore.

Following the Activity on Node notation, a graphical representation looks as follows:

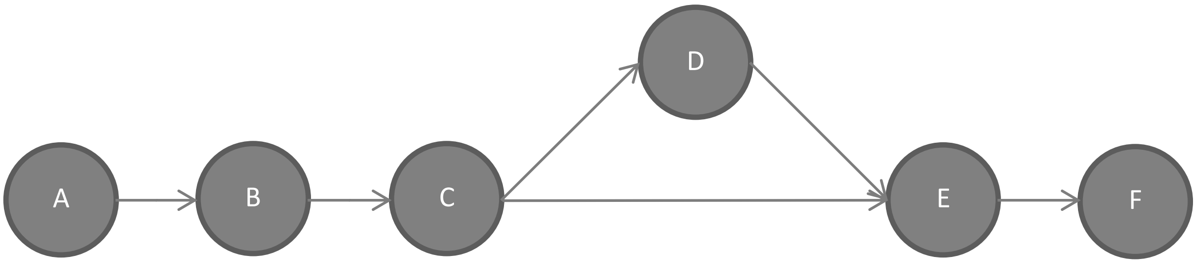

Figure 2.2: Activity on Node representation of defining the ETA process and the ETD process.

Figure 2.2 can also be used for ETD processing, however, with the following activities:

A. Collect ship name from Microsoft Dynamics;
B. Fill in name at MarineTraffic.com and collect data;
C. Check if sufficient data is available;
D. Collect data via ship planning available at hanjin.com or hapag-lloyd.com (only executed when the result of C is negative);
E. Compare data to data in Microsoft Dynamics
F. Process whether ETD within Microsoft Dynamics is still valid or not valid anymore.

Whereas the basics of this current situation are pretty straightforward, the lack of efficiency within the process lays within the use of multiple websites. Looking up information on various websites or checking whether information is available on one website and potentially having to continue that search on another website creates a cumbersome manner of working.

2.1.2 Defining the objectives of a solution

The main objective for the solution of the problem should be the creation of a more straightforward method to check whether the given values for the ETA are still valid. Finding a way to automate this process would not only be beneficial for the time spend on this matter by employees, it will also most likely lead to executing this process more often, which will then lead to more precise data. This way, we can improve data quality while spending less man hours on the execution of the process.

There are various ways to check whether the planned ETA is still up to date, among which the most important ones are the current method of comparing locations with schedules and determination of the ETA based upon data which will be described in detail in Section 2.3. Comparing these values in a more automated way should raise efficiency and give valid data, as multiple methods can be used.

A final objective for a solution is to strive to exclude human expertise and interference as much as possible, as human expertise is hard to automate and human interference does not match with an automated solution.

2.2 Automating the current situation

This section will introduce methods which can be used to automate the current method as described in Section 2.1.

2.2.1 Web scraping

Within the current situation, data from various sources is used to define whether a listed ETA is still realistic or not. In order to be able to automate this, a method to collect this data automatically is needed. This can be realized by a modern technique, which is called web scraping. Following Technopedia, web scraping is defined as follows: "Web scraping is a term for various methods used to collect information from across the internet. Generally, this is done with software that simulates human web surfing to collect specified bits of information from different websites." By applying this technique on the websites currently used, we could automatically gather the information which is now collected manually, which creates a basis for automating the entire process.

Various tools are available for web scraping, ranging from scripting languages to graphical user interface (GUI) guided solutions. A number of these solutions will be further elaborated on in Section 2.4. Web scraping typically builds upon certain selection methods to extract information from websites, among which XPath (short for XML Path Language) and CSS selection (short for Cascading Style Sheets selection) have been found to be the most common types of selection. CSS selection and XPath are quite similar, both using HTML (short for HyperText Markup Language) tags in a generated web page as selection methods. However, CSS selection fits better with the complexity of modern websites and allows for powerful and robust queries (Kanaoka et Al., 2014). This leads to a preference for CSS selection, even though the final method of choice is also dependent on applications available, other selection criteria for these applications besides the selection methods they support.

2.3 ETA Definition

Within this section, various techniques to process data for ETA calculation are discussed and translated to possible use within the case study.

2.3.1 Historical Data

Everyone has a certain trip one will execute on a regular basis. Whether this is your every day trip to work by car or the bike ride towards your favorite pub, you know how much time this trip will approximately take as you have executed this trip so many times. And even though this trip might vary in time as you experience certain delays, you have a clear estimation of the time this will take based upon experience.

What is unconsciously happening in such a case is that you make an estimation based upon historical data. Your estimation will also most likely get better after executing this trip more often. This principle is also applicable as a technical solution within the shipping industry: by collecting data regarding travelling between point A and point B, one could start predicting the time spend on covering the distance between points A and B.

Most journeys as traveled by vessels are journeys with various stops during this journey. If one assumes a trip from point A to point C via point B, the time spend on the journey could be split in the journey between A and B, the journey between B and C and the time spend at point B. Following Heywood et Al. (2009), it is possible to model this in the following way: if we assume that a vessel has current location P, his journey will consist out of the time spend until reaching point B, the time spend on covering the distances between point B and point C, point C and point D, etcetera and the time spend at the stops B, C, D, etcetera.

The distances between these points can be based upon historical data. By following ship locations throughout the distance it covers and saving this data at a number of set points (see Figure 2.3) together with the time spend, a proper estimate can be made in the future. By repeating this process, you can build a database which will be referred to as “HistoricalLeg”. As time passes by, this database will grow larger, meaning one can base estimates on more historical data, providing a more accurate estimation.

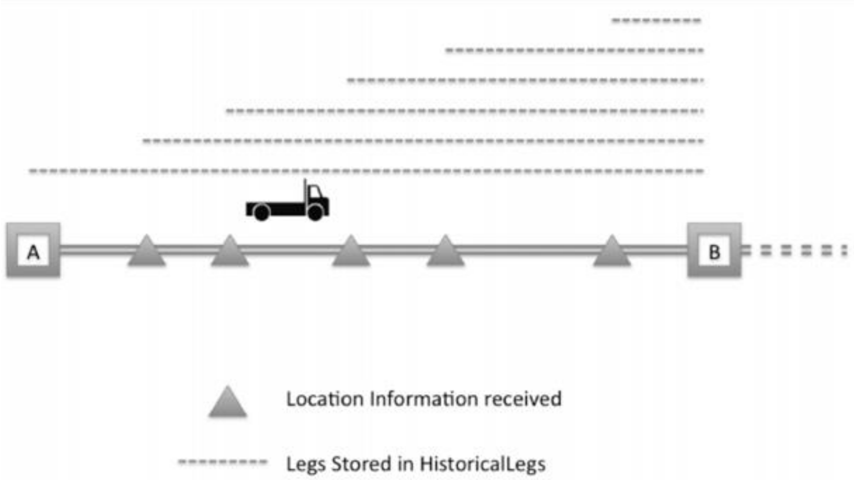

Figure 2.3. HistoricalLeg rows generated as the cargo proceeds from *A* to *B*. Sent location information is indicated by triangles. Each dotted line over the route indicates a startpoint, endpoint, and elapsed time added to HistoricalLeg (Heywood et al., 2009)

Converting this process into a formula, this looks as followed:

$$ETA = t_{PB} + t_{BZ} + t_{transfer}$$

With:

$$t_{PB} = \frac{\sum_{i=1}^{n} t_i}{n}$$
$$t_{BZ} = t_{BC} + t_{CD} + \cdots + t_{YZ}$$
$$t_{transfer} = t_B + t_C + \cdots + t_Z$$

While this solution looks fairly simple, a number of issues are present while using this method. The first problem which comes to mind is the fact that a received location will hardly ever be the exact same location as the locations as stored within the database. This also means that estimations would be based upon less historical data, which is in contrast with earlier statement that estimations would grow to be more precise over time, as they could be based upon more historical data. This could be solved by implementing a sensitivity radius *r* around the current location *P*. The radius proposed by Heywood et al. (2009) is defined as follows, also illustrated by Figure 2.4:

$$r = 0.05 * \overline{AB}$$

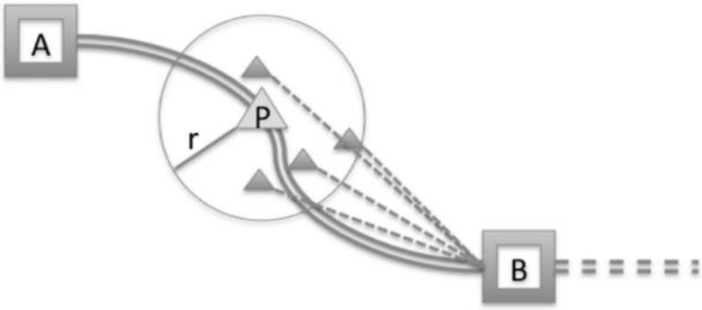

Figure 2.4. Sensitivity radius capturing HistoricalLeg rows with startpoints near *P*. Four legs are deemed within *r* of *P* and also end at *B*. The ETA will be calculated using the average of the elapsed time of these previous shipments. (Heywood et al., 2009).

Another problem arises while considering high traffic density, which leads to slow process in covering distance and thus translate to collecting almost the same location data a number of times. This problem could also be solved by implementing a sensitivity radius, combined with adding a certain weight to this location. If a location within radius *2r* of an earlier location is saved in HistoricalLeg, this location is not saved and the earlier location is given a weight of 2.

The final concern which will be taken into consideration is the value of the ETA when a ship has arrived at a transfer point. When a ship has arrived at for instance point B, the formula stated on the previous page would no longer take the time spend at this point into account. Therefor we introduce a clock t_B which starts a countdown as soon as the ship comes within a range *r* of point *B*. This countdown is taken into account when calculating the ETA up until the point when the ship is a range *2r* away from point *B*.

A final thing to take into consideration is that "if an unusual delay occurs at one transfer point, delays are likely at future transfer points due to a disruption in the planned schedule". (Kara et Al. 2001). Heywood et al. (2009) do mention this problem, however, they do not take its results into account while making calculations. By investigating the data later on, adjustments to the algorithm could be made, taking this into consideration.

The solution as proposed by Heywood et Al. (2009) is based upon a case study at TMO Global Logistics, which sketches a starting point which is very similar to the starting point at HST as described in Section 2.1. Therefore it is concluded that a solution based upon historical data is suitable for the calculation of the ETA considering long distance shipping at HST.

2.3.2 Other ETA calculation methods

Other methods to calculate the ETA which have been investigated are modeling the presence of obstacles on the route as described by Fagerholt et Al. (2000) and taking into consideration the influence of weather as described by Szelangiewicz et Al. (2014). Fagerholt et Al. (2000) describe a method to calculate an ETA based upon the shortest path while taking into account certain obstacles in for instance port areas. Szelangiewicz et Al. (2014) describe a method to calculate vessel speed based upon weather influences, which could be used to calculation ETAs under the assumption of sets routes. However, Fagerholt et Al. (2000) state that their “proposed solution method for calculating the ETA is best suited for short sea voyages with sailing times up to a few days. For deep sea voyages, ocean currents and bad weather that may force ships to make speed reductions or alter course represent too much uncertainty” and Szelangiewicz et Al. (2014) note that their “computer code does not automatically reduce the speed of the ship when the permissible limits of the wave induced ship rolling are exceeded (…), and it is only the navigator who can reduce deliberately the speed of the vessel”. Besides this, the calculations made by Szelangiewicz et Al. (2014) are only valid under a number of restrictions as maintenance of constant RPM or engine power, constant specific fuel consumption or constant ship speed in changing weather conditions. Especially on long distances these conditions might not be met, leading to inaccurate calculations. Furthermore a large variety of ship related data is necessary, which might not always be present. Therefor the results of these methods do prove the relevance of weather influences and obstacles in port regions, however, they are not suitable as underlying model within our case. Therefore it is concluded that it is wise to take into account that both the weather and presence of obstacles influence the ETA, but that the methods as proposed are not suitable within the developed application.

A final useful method is to automate the current method. This would ensure a more efficient method of ETA definition in a possible transition period in which the HistoricalLeg database does not have sufficient data to base ETAs on. By generating the ETA in more than one way once the HistoricalLeg database is filled, one could compare the outcomes as well, which can provide useful information on the reliability of the ETAs. Furthermore, data on the ship type, but also on the weather during the journey could be saved to research their effects in the future and possibly adapt the algorithm following these effects.

2.3.3 AIS

AIS stands for Automatic Identification System. A clear overview of what AIS is, is given by Wijaya et Al. (2013):

> *“AIS is a means for vessels that make it possible to transmit data via radio wave to other vessels as well as to inland stations. The transmitted information can be divided into three categories: (1) Static Information, which includes vessel’s name, International Maritime Organization (IMO) number, Maritime Mobile Service Identity (MMSI) number, and dimension. (2) Dynamic Information, which contains vessel’s position, Speed Over Ground (SOG), Course Over Ground (COG), current status, and rate of turn. (3) Voyage-specific Information, which consists of destination, Estimated Time of Arrival (ETA), and draught.”*

AIS data has been used for various research subjects regarding maritime traffic, such as defining motion patterns and motion detection. Results show that motion behavior can be predicted well and ship routes are more or less set (Ristic et Al. 2008), which pleads for acceptance of set routes. In the following part of this research however, a definition of AIS is sufficient for the understanding of the applied calculations, as it will mainly serve as a source for location data.

2.4 Data Sources and Processing

Within this section, some background information on various sources of data is provided as well as information on the way of processing this data. Together with Section 2.2.1 this section answers the second sub question.

2.4.1 CloudScrape

CloudScrape is a web scraping tool, providing the functionalities as described in Section 2.2.1. CloudScrape has been used as it is an easy to use, browser based application with an easy to understand user interface (see Figure 2.5). CloudScrape follows the actions you go through on a website, saving them and providing the ability to automate them and save certain data from the website along the way. The selection of data works via CSS selection and unlike other applications found, CloudScrape provides a REST API (short for REpresentational State Transfer Application Programming Interface) to ease communication with other applications.

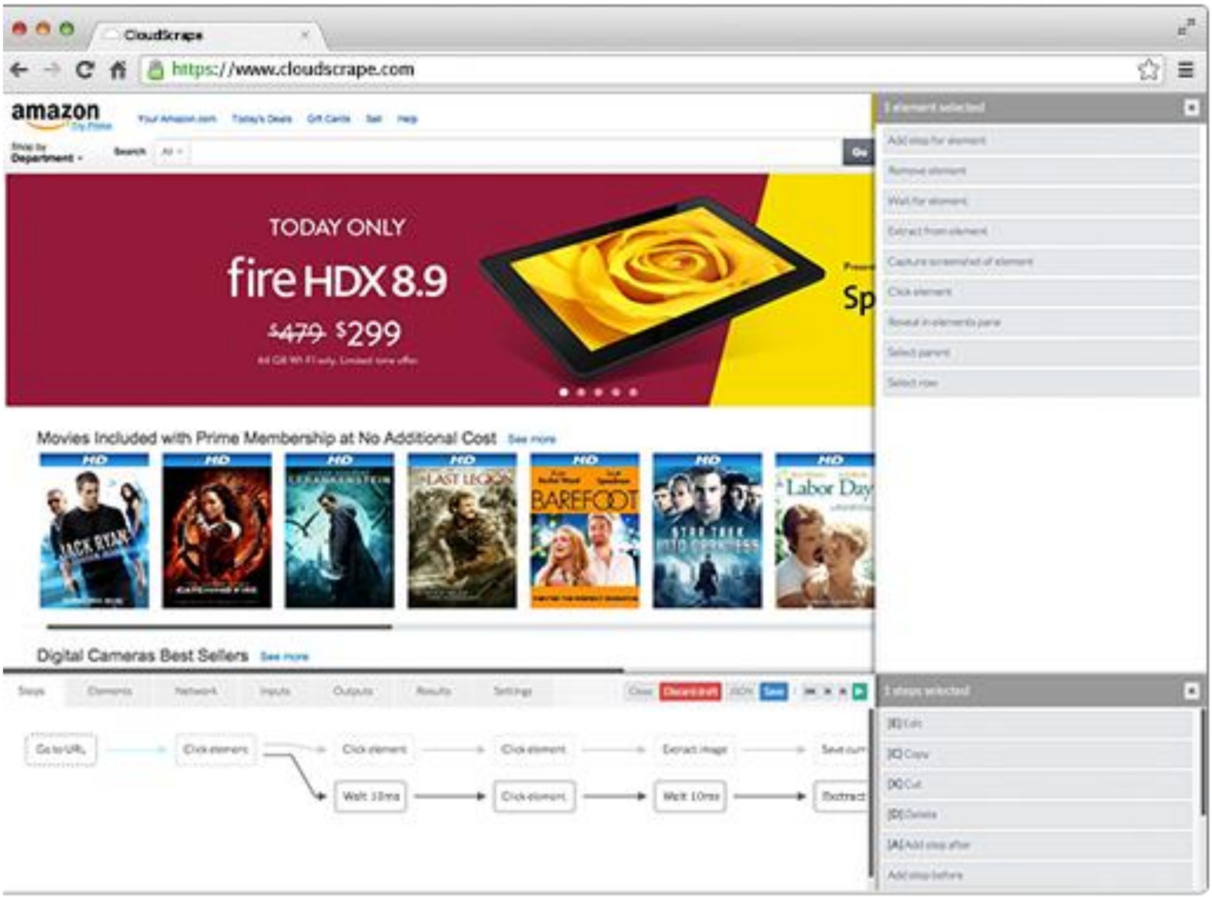

Figure 2.5 CloudScrape interface

Several other applications have been taken into consideration as well, among which RapidMiner, WebHarvest and Mozenda are the most important ones. However, RapidMiner could not scrape all websites which needed scraping as it for instance does not support logging in on websites to gather information from them, WebHarvest was hard to combine with other applications as it lacks an API and Mozenda was extremely expensive compared to CloudScrape. This also means our hypothesis as described in Section 1.5 is false. As a part of the application, CloudScrape is used to automatically collect the data described in the following sections. More detailed information on how this works exactly will be provided in Chapter 3.

2.4.2 Schedule- and ETA data

For schedule and ETA data provided by shipping companies, we are dependent on the various shipping companies HST works with. Therefore there is a limited amount of websites used to collect this data. For ETAs, ECT.nl and APMTRotterdam.nl are used, for the collection of shipping schedules Hapag-Lloyd.com and Hanjin-Shipping.com are used.

2.4.3 AIS Data

AIS Data can be used as a source to define the (close to) real time location of a vessel and to collect the ETA as predicted by the vessel's staff. Mainly the location data is important in the design steps. MarineTraffic.com translates AIS data into clear and usable data, which is why it will be used as our source for AIS data.

2.4.4 Mendix

Mendix is an application Platform as a Service (aPaaS), which can be applied to develop apps quickly, without having to use code. The use of Mendix is common within Cape Groep, so it is logical to use Mendix to develop the application and bring everything together.

2.4.5 eMagiz

eMagiz is a web based PaaS solution, providing loads of functionalities concerning transformation of various data types. eMagiz integrates with Mendix perfectly as standard modules for this integration are available and in the scope of this research its use lays within better API handling comparing it to Mendix as standalone software.

3. Design and Development

Within this section, the designed solution for the main question of this report as defined in Chapter 1 will be described. This solution will be implemented into an application designed for process automation. This chapter will answer the third sub question "How can this algorithm be implemented in a technical solution and what applications and/or software is needed for this?"

3.1 Architecture

Within this section, the application will be explained using its architecture. This architecture mainly serves to set out the ideas behind the application, which will help to understand the design steps better later on.

The architecture of the model has been drawn using the ArchiMate language and Archi application. Not only was the choice for the ArchiMate language made because of earlier experience with this language both within Cape Groep as for the author, also because it "has become for EA design what UML is for software design with its own international open standard". (Iacob et Al. 2014)

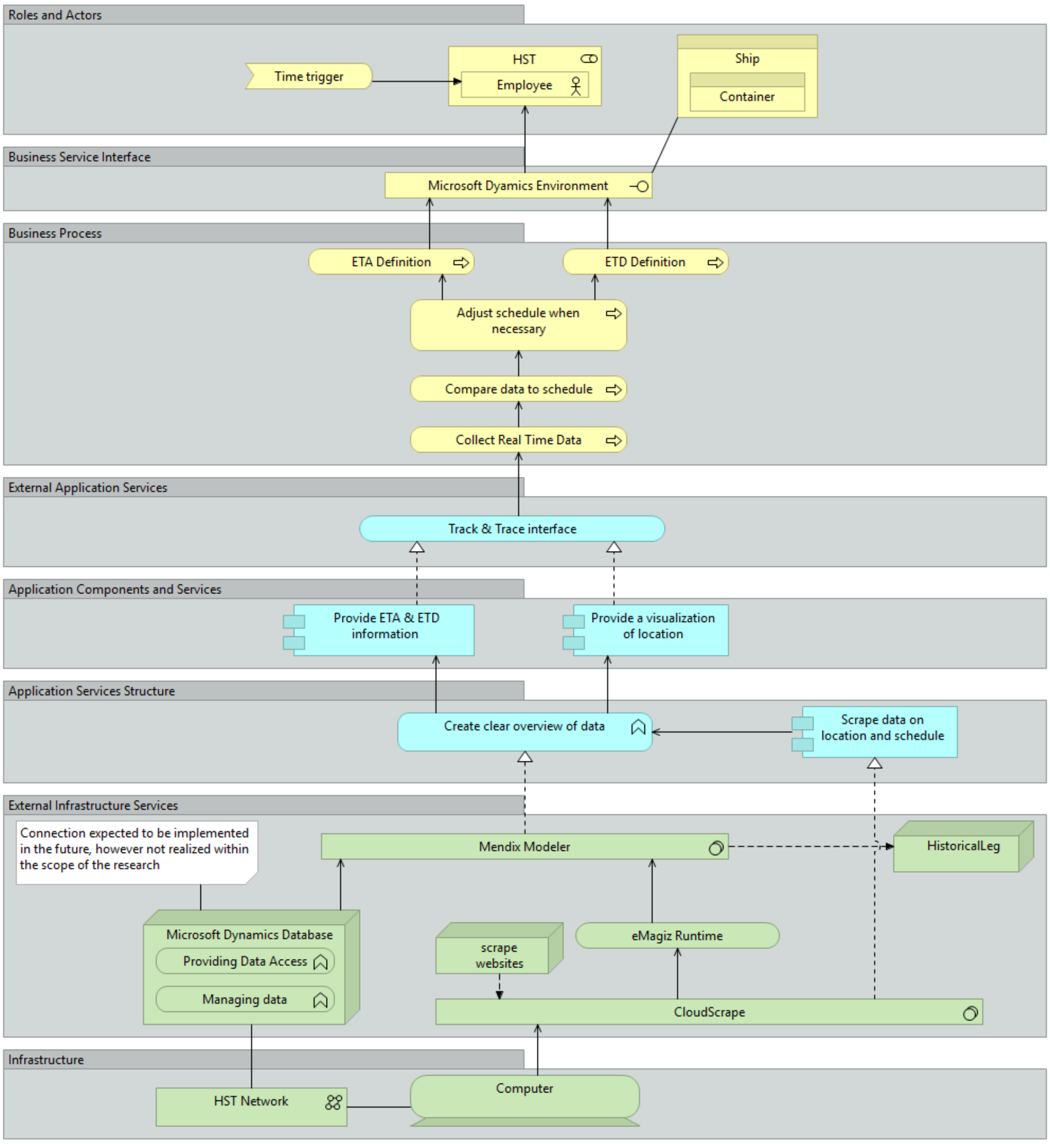

Figure 3.1 Architecture of the designed solution

The architecture shown in Figure 3.1 is a partial translation of the methods as explained in Chapter 2, including several types of software which are able to provide the services we need. It consists out of three main parts, the business layer, the application layer and the infrastructure layer, as also used in ArchiMate's Layered Viewpoint (The Open Group, 2013). The business layer explains processes as executed by actors, the application layer lists processes which are executed by technological solutions and the infrastructure layer lists the hard- and software used to achieve this. The main idea behind the application is to develop a track & trace interface, which serves as a portal where all information available on several websites is showed in a clear format. In the future, this system could be linked to HST's database to fully automate this process. Implementation of these functionalities was expected to require changes at HST's IT infrastructure and therefore take up too much time for the scope of this research. Besides this, using an interface will ease testing and visibility of the developed service.

3.1.1 Business layer

The business layer consists out of the top three group blocks within the ArchiMate structure, as also shown Figure 3.2, which can also be distinguished by the yellow entities. In this particular case, the business layer is relatively clear and straightforward, as our application does not have that many roles and actors involved. Within our case study, the designed application will be used by HST only and more concrete, by one or more of their employees. To connect with the current situation at HST, it is important that the application can communicate with the current IT environment at HST, which is a Microsoft Dynamics environment as described in Section 2.1.1. The HST employee will use this for two processes, the ETA definition and the ETD definition, which is done by collecting real time data, comparing it to the schedule and possibly adjusting the schedule. Besides this, the Microsoft Dynamics Environment lists information on containers which are transported using ships. Finally, the process is executed following a certain time trigger.

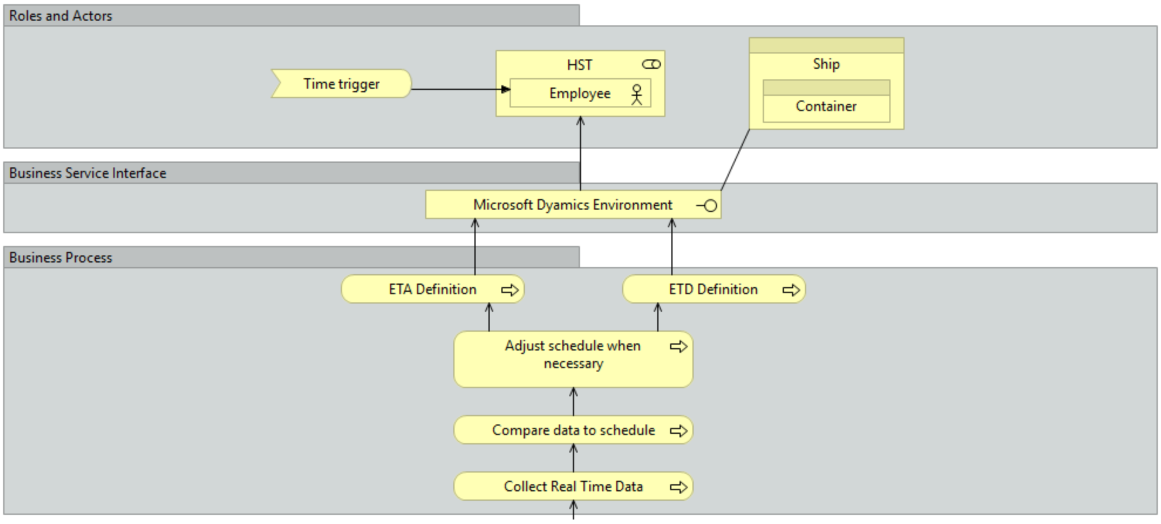

Figure 3.2 : Business layer

3.1.2 Application layer

The next three layers list the application's functionalities, as also shown in Figure 3.3. The main service provided by the application is the track & trace interface, listing the information from various websites on the ETA & ETD and also providing a visualization of the ship's location. To achieve this, the application has to create a clearly structured overview of the data and has to scrape the data from the various websites.

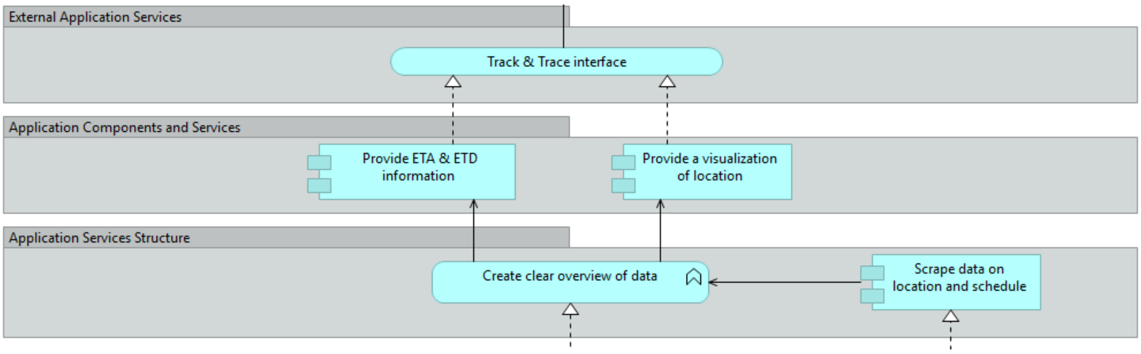

Figure 3.3: Application layer

3.1.3 Infrastructure layer

The infrastructure layer is the more complex part of the ArchiMate structure. To achieve the described functionalities, several applications are used. The main application is CloudScrape, which scrapes all the data from the various scrape websites. CloudScrape is accessed via an application build in the Mendix Modeler, which communicates with the CloudScrape API using the eMagiz runtime. Besides this, the Microsoft Dynamics database is used to compare the scraped data to the data within the database. Last but not least, the scraped data is saved into the HistoricalLeg database, to provide possibilities for future ETA calculation and for future analysis.

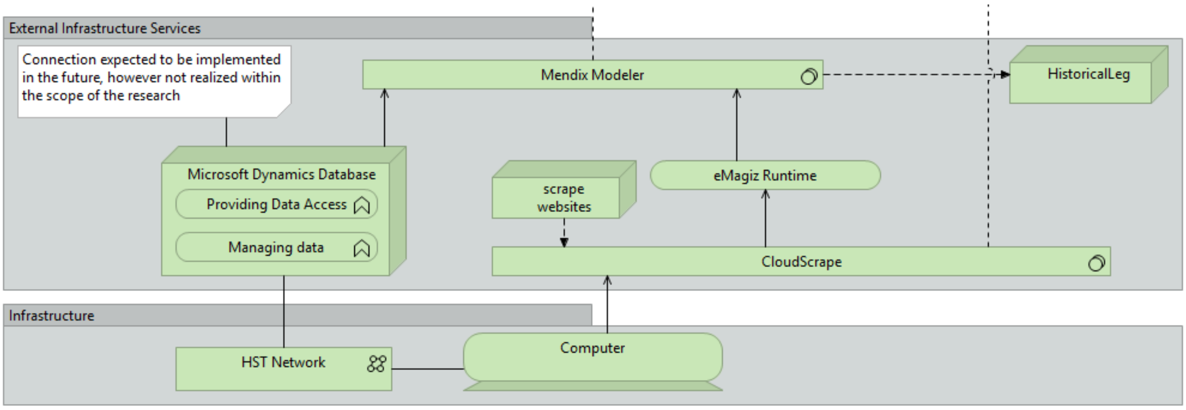

Figure 3.4: Infrastructure layer

3.2 CloudScrape

As also explained in Section 2.4.1, CloudScrape is an application which lets its user scrape information from a website, based upon certain input data. This is done using so called "robots", an automation path for each website one wants to scrape information from. Eventually, CloudScrape is addressed via its REST API functionality.

3.2.1 Robots

A so called robot within CloudScrape lets you open up a certain website, then travel a certain path on that website and scrape certain data off that website along the way. Within this section, the Hanjin-

Shipping.com robot will be described in detail to explain CloudScrape's functionality in detail. The other robots which scrape data from ECT.nl, APMTRotterdam.nl, MarineTraffic.com and Hapag-Lloyd.com work in a similar way.

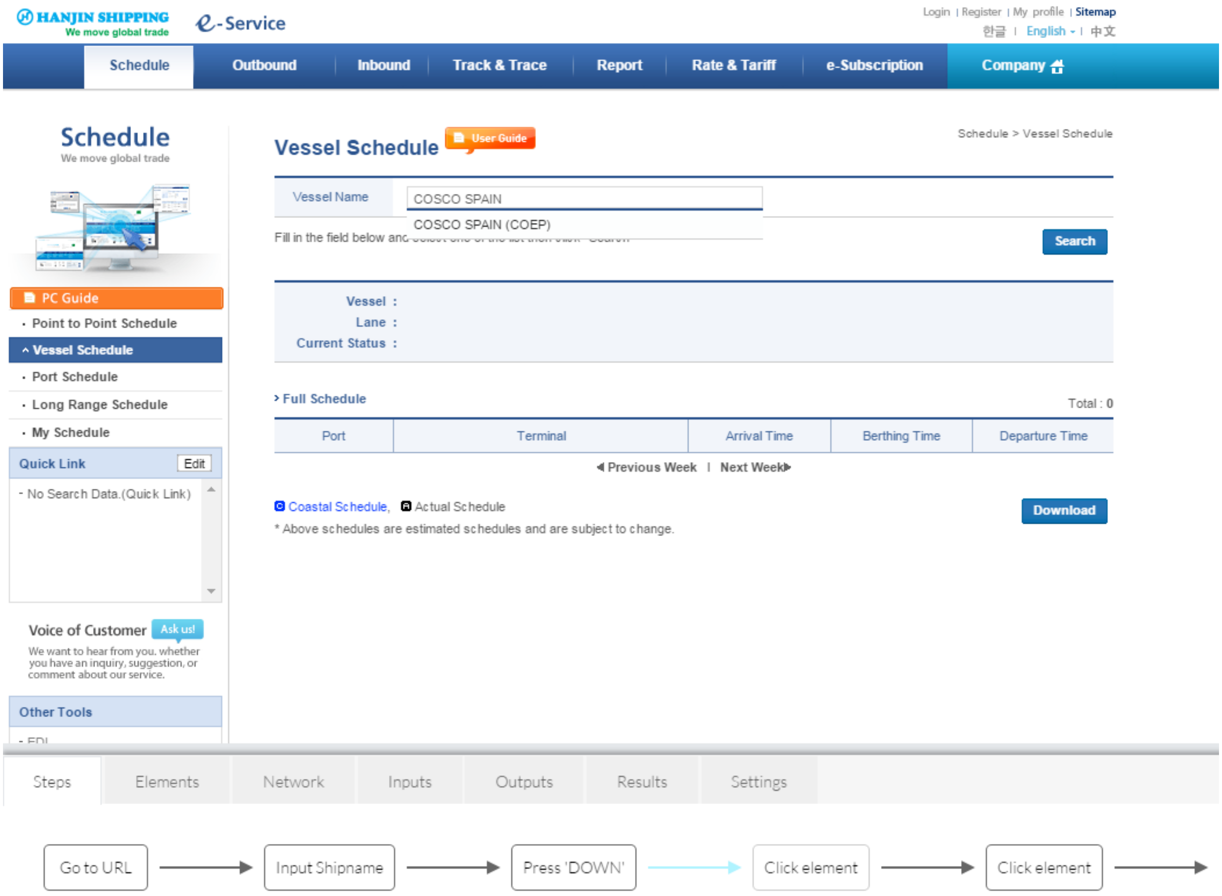

Figure 3.5: The first part of Hanjin Shipping's CloudScrape robot

Figure 3.5 lists the first part of Hanjin Shipping's robot, which is the part making sure the data we need actually appears on the screen. The steps are listed in the lower part of the figure, starting off with going to Hanjin Shipping's uniform resource locator or URL. After this, the ship's name is inputted (in this particular case Cosco Spain) and the down key is pressed, so that the ship shows up with the exact name as listed in Hanjin-Shipping's database. Afterwards, this element is clicked and the search element is clicked, so that the data actually appears on the screen, as shown in Figure 3.6.

CloudScrape works by simply letting you click a certain element and then let you decide whether you need to input something in the field you have clicked, need to click the element, etcetera. If necessary, you can also manually edit the CSS selection path generated by CloudScrape. It is also possible to let the robot run with various values for the input field. This means we can run this robot, scraping data for various ships.

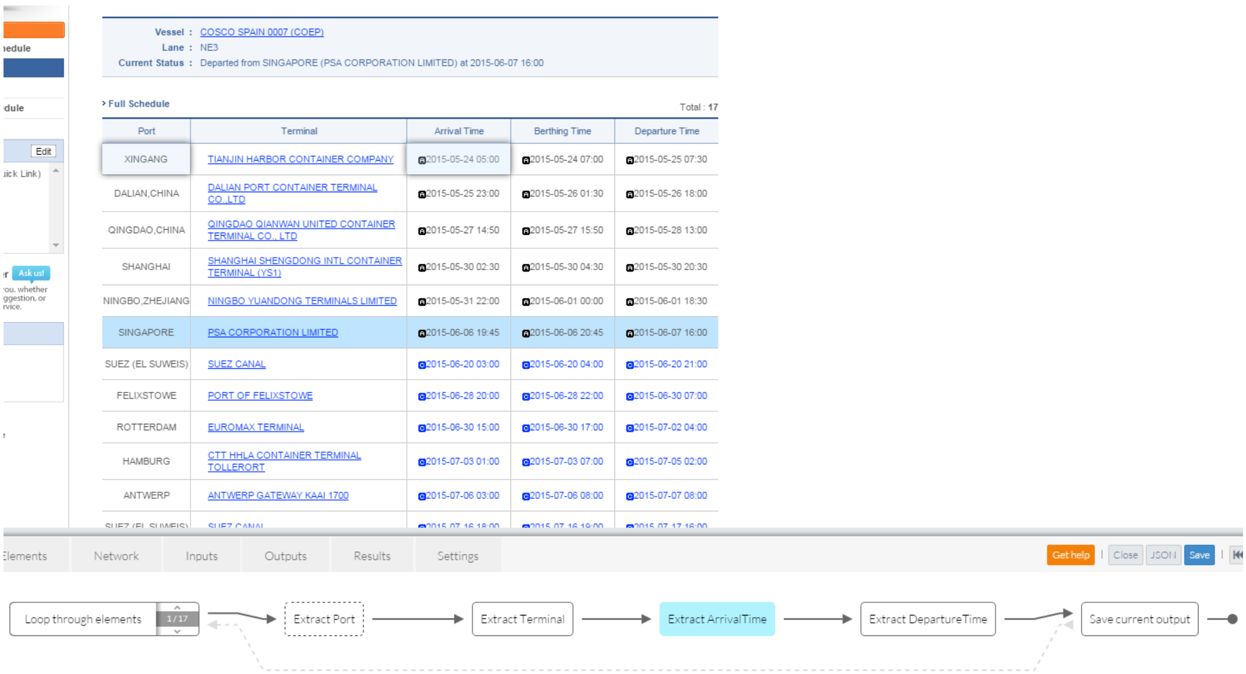

Figure 3.6: The second part of Hanjin Shipping's CloudScrape robot

After the second "click element" step in Figure 3.5, the data is loaded onto the screen as in Figure 3.6 after which the actual data scraping starts. CloudScrape creates a loop, going through all the first elements of the table created on the web page. Within this loop, several values of the rows are scraped and saved into a results table. For Hanjin-Shipping.com, this means that for each row, the port name, terminal name, arrival time and departure time are saved within a results table. By going through the loop, a ship's schedule is extracted from the website, featuring this data. Extra functionalities include removing possible extracted HTML tags and formatting user output to match a certain format.

3.2.2 CloudScrape API

CloudScrape has several possibilities to reach data via their API or to execute certain robots, using a so called run. In the application, the API used is the so called executeWithInputSync API. This API is described as follows: "Starts new execution of run using the input from the body instead of the run itself and outputs the output when it's done. This will also delete the execution from CloudScrape immediately after execution. Please note that you need to ensure your HTTP client does not timeout for long executions. Headers will be outputted when the output is ready." (CloudScrape, 2015). Translated to the case study, this means a certain ship name can be inserted, after which the API will return the ETA values and other relevant data. An example of the in- and output for Hanjin Shipping's robot is added in Appendix III. An overview of which data is collected from which website is listed in Appendix IV.

3.3 eMagiz

eMagiz is used to communicate with the CloudScrape API. As also explained in Section 2.4.6, eMagiz is a lot better with REST integrations than for instance Mendix, which does not provide the functionalities needed on its own. By incorporating eMagiz, the needed functionalities become available, which is why eMagiz is used. The integration works as follows: eMagiz receives a message from Mendix with inputs for CloudScrape, information on which robots to include and a Search ID, so that inputs can be linked to outputs within Mendix. A separation between synchronous and asynchronous calls is made when using eMagiz: synchronous calls always have an answer linked to an input, whereas asynchronous calls separate this. Within this application, an asynchronous call is used as one input message can have multiple output messages, which is not possible with a synchronous call. Certain parts of the process are added automatically by eMagiz, others have to be added or configured within eMagiz or sometimes even with the use of some coding. The following set up has been made using version 6.5.0 of the eMagiz portal, version 3.20.1 of the eMagiz runtime and version 1.20 of the eMagiz Mendix connector.

3.3.1 eMagiz Portal

Within this section, the application as build within the eMagiz portal will be explained. Starting from the eMagiz message bus which gives an overview of the application, more detailed explanation will follow as deeper levels of the design are exposed.

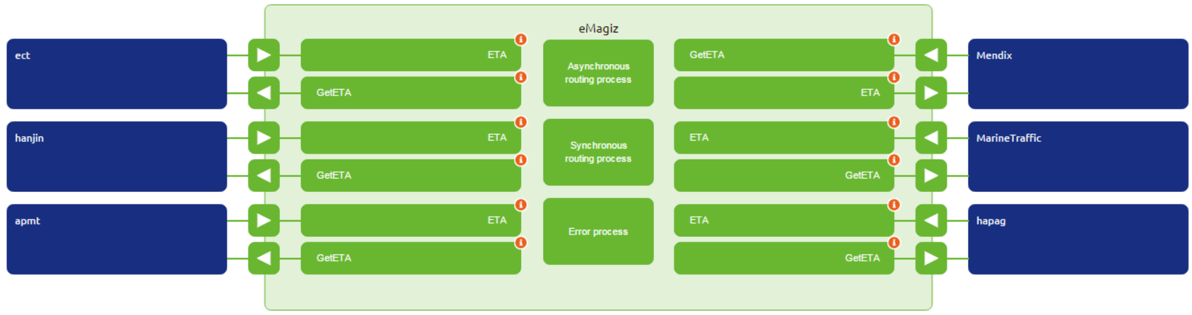

Figure 3.7 eMagiz Message bus

The so called message bus (Figure 3.7) in eMagiz sets out the functionalities provided by the program. The blue blocks represent the various machines eMagiz communicates with, which in this case are Mendix and the websites used to scrape data from. The green blocks represent the actual functionality provided by eMagiz. In short, an asynchronous request message named "GetETA" is provided by Mendix and travels through eMagiz to one or more of the machines, where an answer is provided. As the call is an asynchronous one, the answer message has a different name, in this case ETA. This message is converted into a format functional for Mendix and is then send back to Mendix.

To provide a more detailed overview of what is really happening, we will now look into the processes for a call to Hapag-Lloyd. An incoming message first travels to the GetETA onramp (Figure 3.8). This message travels through the GetETA onramp, which transforms the message in a useful format for eMagiz. The GetETA looks as follows:

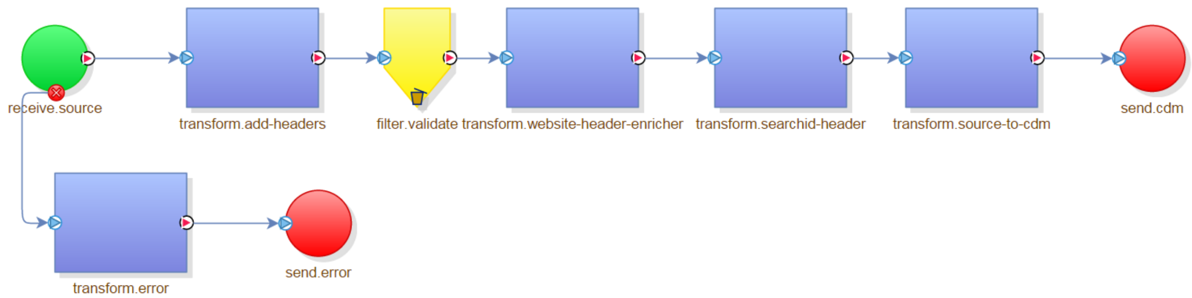

Figure 3.8 GetETA onramp

After a message is received, information on the source of the message is added in the first add-headers block. Thereafter, a validation filter checks whether the message is a valid message.

Subsequently, the Search ID and selected websites are transformed into headers for further processing and the entire message is transformed from Mendix' XML format to CDM format, which is done commonly within eMagiz for possible future integrations. Sending the message to the asynchronous routing process concludes the functionalities of the GetETA onramp.

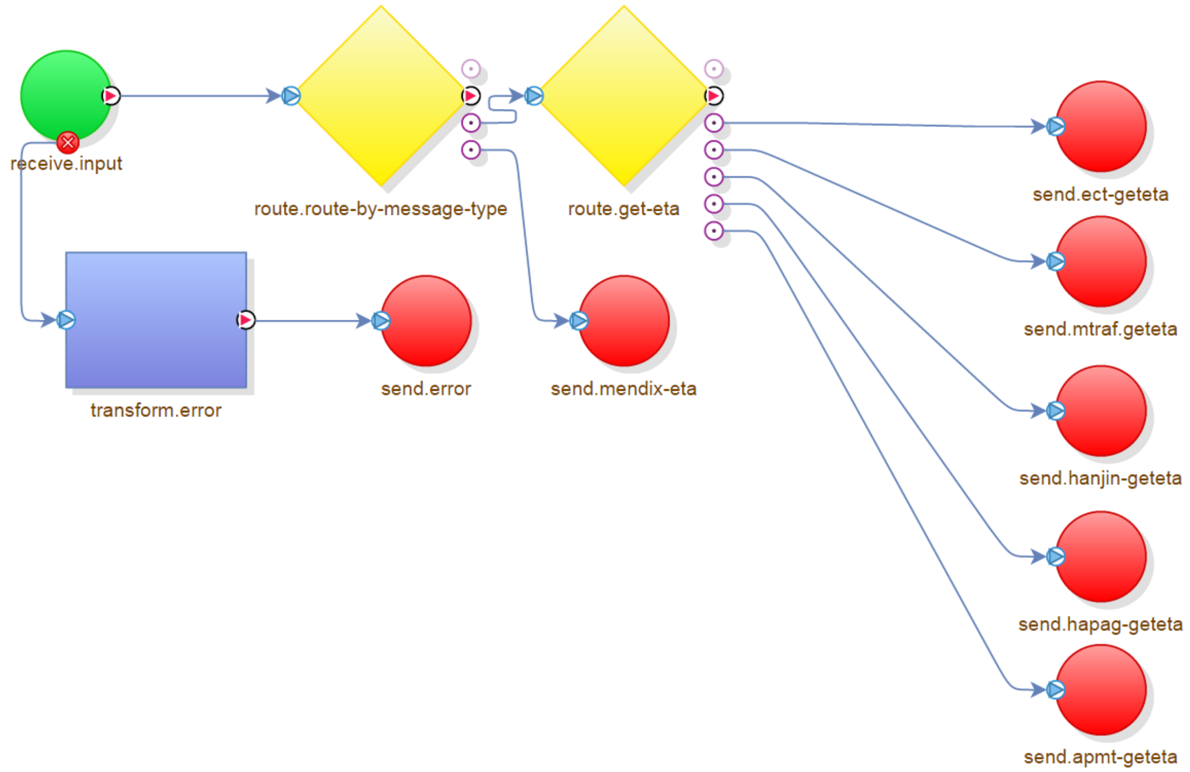

Figure 3.9 Asynchronous Routing Process

Hereafter, this message arrives in the asynchronous routing process (Figure 3.9). The main functionality of the asynchronous routing process is sending messages to the correct places. Both the request message and the answer message travel through the asynchronous routing process, so the first thing done after a message comes in, is checking whether the message is a request- or answer message using a header router. Based upon header values, the messages are sent to the correct places. An answer message travels back to Mendix, whereas a request message is send towards another routing block, which sends the message to the correct machine based upon which websites have been selected, which is why these values have been transferred into header routers before. The message itself is not changed. As a request message for Hapag-Lloyd is described, the message travels to the hapag-geteta block, which sends the message to the GetETA offramp for the Hapag machine.

The main functionality of the offramp is to transform the incoming message to JSON format, as this is the format requested by the CloudScrape API. The offramp looks as follows:

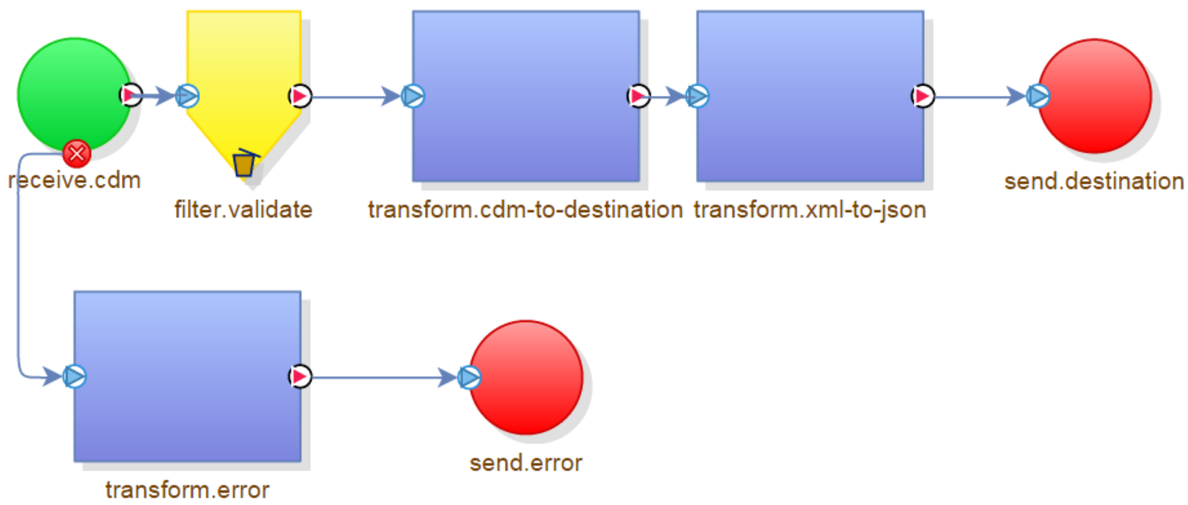

Figure 3.10 GetETA offramp

First, an incoming message is validated as a request message. After this, it is transformed from CDM format to XML format, in such a way that it can easily be converted into JSON afterwards. This is done in an XML transformer, which matches certain input values to certain output values. This is shown in Figure 3.11:

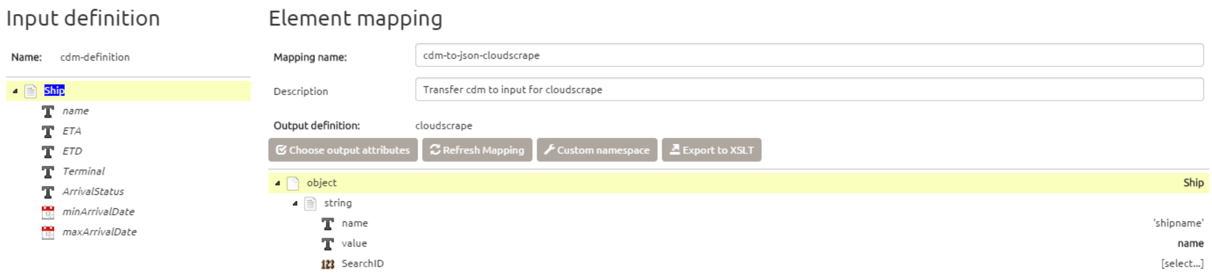

Figure 3.11 XML transformer in the GetETA offramp

After transformation to JSON format, the output exactly matches the format as requested by the CloudScrape API, which is also shown in Appendix III. Hereafter the message is send to the exit connector, which is where the communication with the actual machine happens. The exit connector is shown in Figure 3.12.

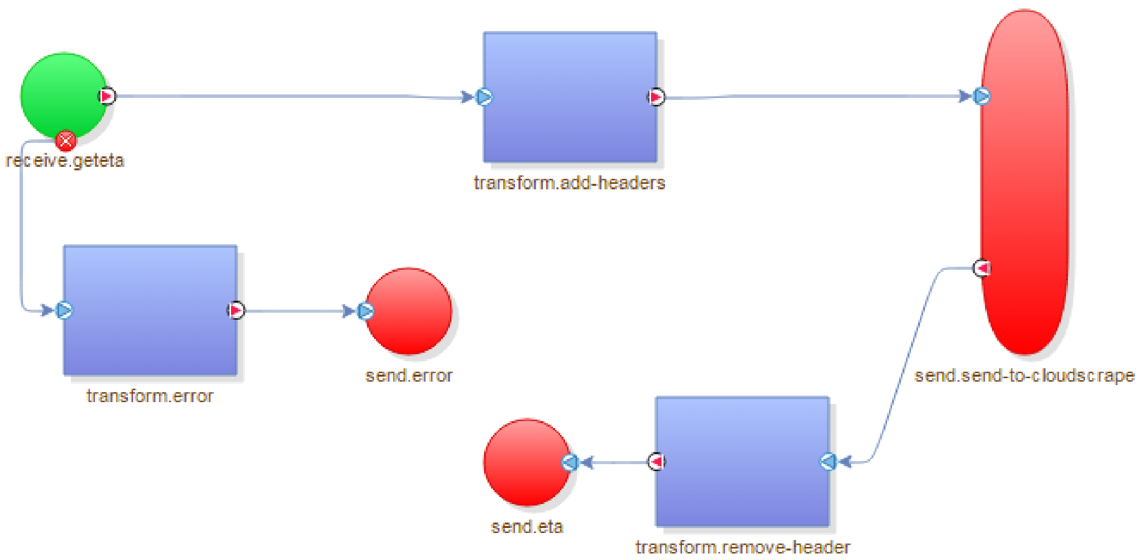

Figure 3.12 Exit connector

After receiving the message, a number of headers required for communication with the CloudScrape API are added to the message. Hereafter, the message is send to CloudScrape and an answer is received. This answer includes a header with meta information like the time at which the answer is received, which comes in a format impossible to process in Mendix, so this header is removed before the answer message is send towards the ETA onramp for the machine Hapag, which is shown in Figure 3.13.

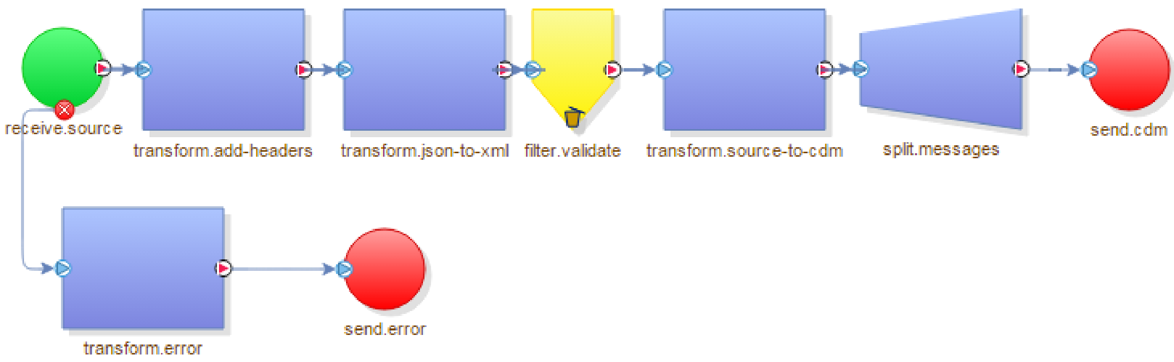

Figure 3.13 ETA onramp

After receiving a message within this ETA onramp, a number of headers is added to the message so that its source is traceable. After this, the JSON message is transformed to XML and validated to check whether this really was a JSON message. Hereafter, the message is transformed to a format which is suitable for processing within Mendix. As the output format of CloudScrape's API is slightly unusual and includes nested arrays, this transformation required the use of a custom Extensible Stylesheet Language Transformation or XSLT. This XSLT transforms the message using certain tags, which ease mapping the values later on in Mendix. The code of this XSLT is added in Appendix V. As Hapag-Lloyd returns a shipping schedule, it includes ETA and ETD values per stop. To ease further processing, the response is split into several messages, so that each message has one ETA value and on ETD value, among other returned values. These messages are then send towards the ETA offramp (see Figure 3.14) via the Asynchronous Routing Process described before.

Within this ETA offramp, the message is validated once more and transformed back to the XML format suitable for processing in Mendix, after which it is sent back to Mendix.

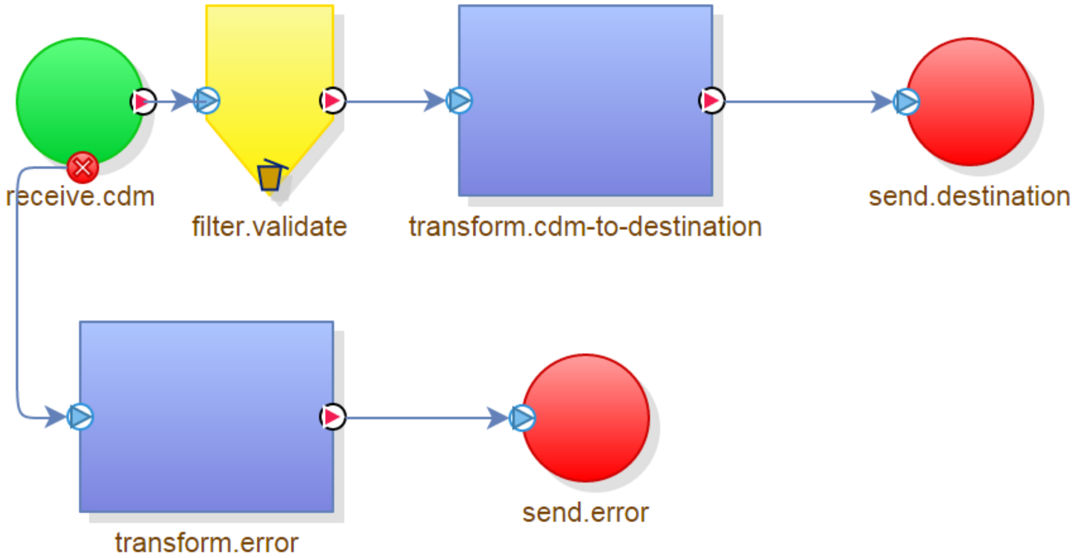

Figure 3.14 ETA offramp

3.3.2 eMagiz Runtime

To be able to test the application locally, a number of configurations had to be made. eMagiz automatically configures a JMS server and a so called container, which can be run locally on a laptop or PC. Besides this, a number of connectors had to be configured to access the various data sources. Configuration files could then be downloaded via the eMagiz Portal and their locations could be added to the connectors, so that the correct parts of the application are run locally in Windows Command Processor. As most of this is configured automatically by eMagiz, no further detailed information is provided on this subject.

3.3.3 eMagiz Mendix Connector

The connection between Mendix and eMagiz is achieved by using the eMagiz Mendix Connector, which is imported as module within Mendix. By importing this into Mendix, eMagiz creates a consumed web service within Mendix, to allow communication between both programs. Configurations can then be made by downloading connection files from the eMagiz portal and uploading them into the Mendix application which runs locally. This concludes the configurations within eMagiz.

3.4 Mendix

The application's User Interface as well as some of the incorporated intelligence of the application is configured within Mendix. A search query is created within Mendix, which is then sent into eMagiz. The results are transferred back into Mendix and a clear overview of the collected data is then provided using Mendix. This application is built using Mendix Business Modeler 5.15.1.

3.4.1 Search Query

Within Mendix, a search query is generated which consists at least out of a ship name, a unique search ID and a selection of which websites have to be included while searching. If APMTrotterdam.nl is included as website to search on, a minimum and maximum expected arrival date also have to be included. These values can be configured within a user interface, after which they are passed on to eMagiz by calling the consumed web service as explained in Section 3.3.3 using a microflow.

3.4.2 Results

The results are imported within Mendix by publishing a web service with an entity in which the results can be mapped. eMagiz then reaches Mendix via the URL of this published web service. Login credentials are inserted within eMagiz, to ensure that not everybody is able to reach this published web service. The actual mapping is done by using the microflow as shown in Figure 3.15.

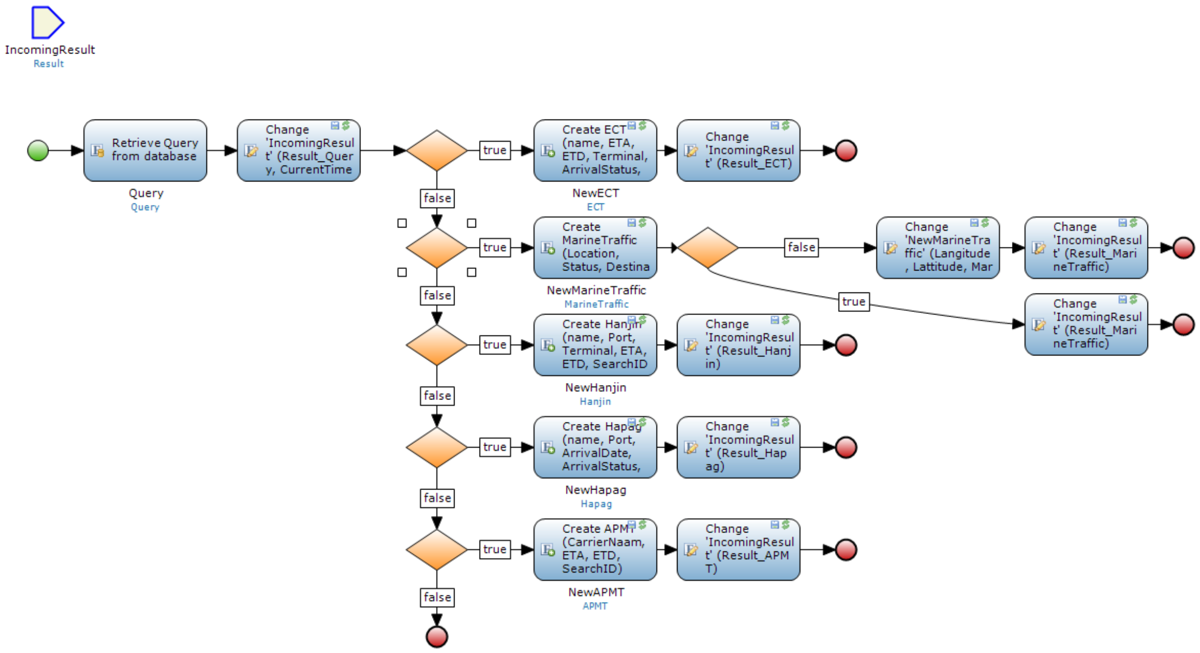

Figure 3.15 Microflow used to process incoming results

The results are mapped using an entity in Mendix' domain model which has attributes with the same name as configured in the XSLT. Once the results are in, the query used to find these results is retrieved from the database, using the Search ID of the incoming data. Subsequently the link between this query and result is configured, to ensure that only results linked to the current search query can be shown on the user interface. After this, the results are mapped into another entity based upon their source within the orange splits. This creates the possibility to later on show the results grouped by source and creates a useful database for future applications. Within these steps, error handling is also added by applying an if-statement for cases where empty results arrive within Mendix.

3.4.3 Overview page

The overview page offers possibilities to alter the search query and shows a clear overview of incoming results, as shown in detail in Chapter 4. Some final intelligence is added when showing results, to ensure that only results for the current search action are shown. The results shown are retrieved from a database, based upon the mapping as explained in the previous section. However, an extra XPath constraint is added, to limit results to the current search action. This XPath constraint looks as follows:

```
[HST.Result_ECT/HST.Result/HST.Result_Query = '[%CurrentObject%]']
```

This XPath constraint compares the link between the result and the query, which is configured within the microflow shown in Figure 3.15 to the current object. It only shows results for which this comparison returns true.

4. Demonstration

Within this section, a demonstration of how the application works will be provided. After logging in, a user interface is reached, which is shown in Figure 4.1. The top half of the screen shows the possibilities for searching, which consist out of adding a ship name and selecting several websites to be included when searching.

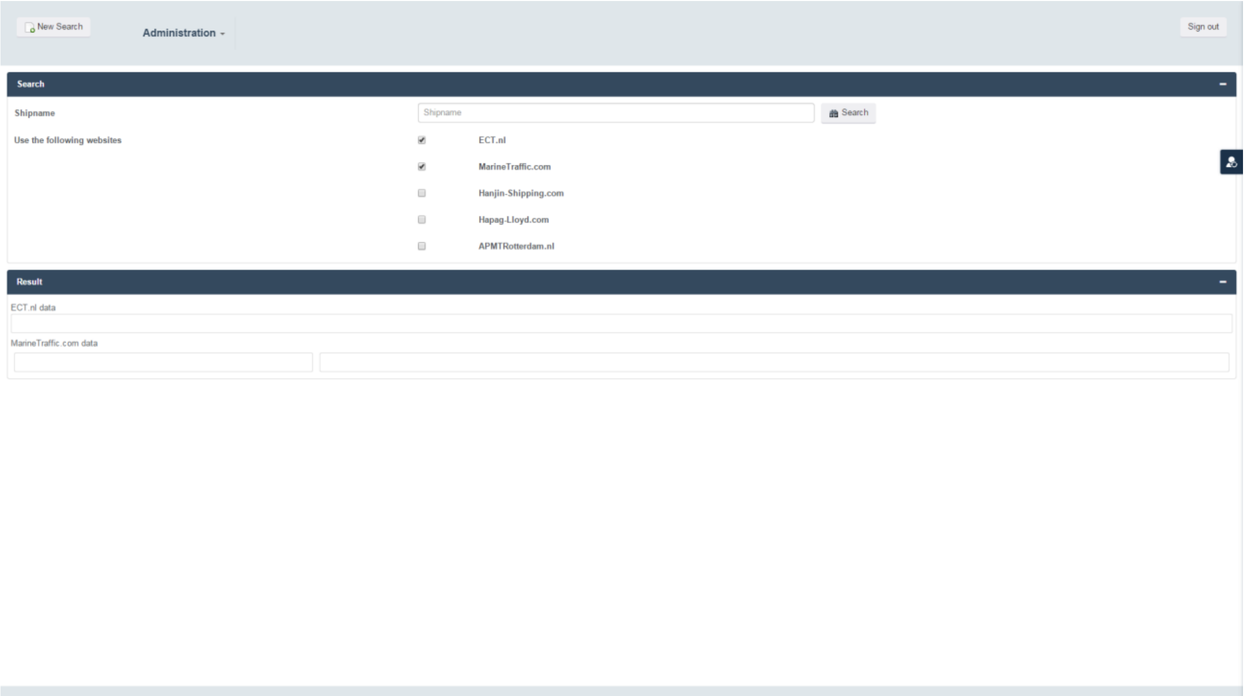

Figure 4.1 Application home screen

As shown in Figure 4.2, the to be added expected minimum and maximum arrival dates appear when APMTrotterdam.nl is checked as one of the websites to be included in the search.

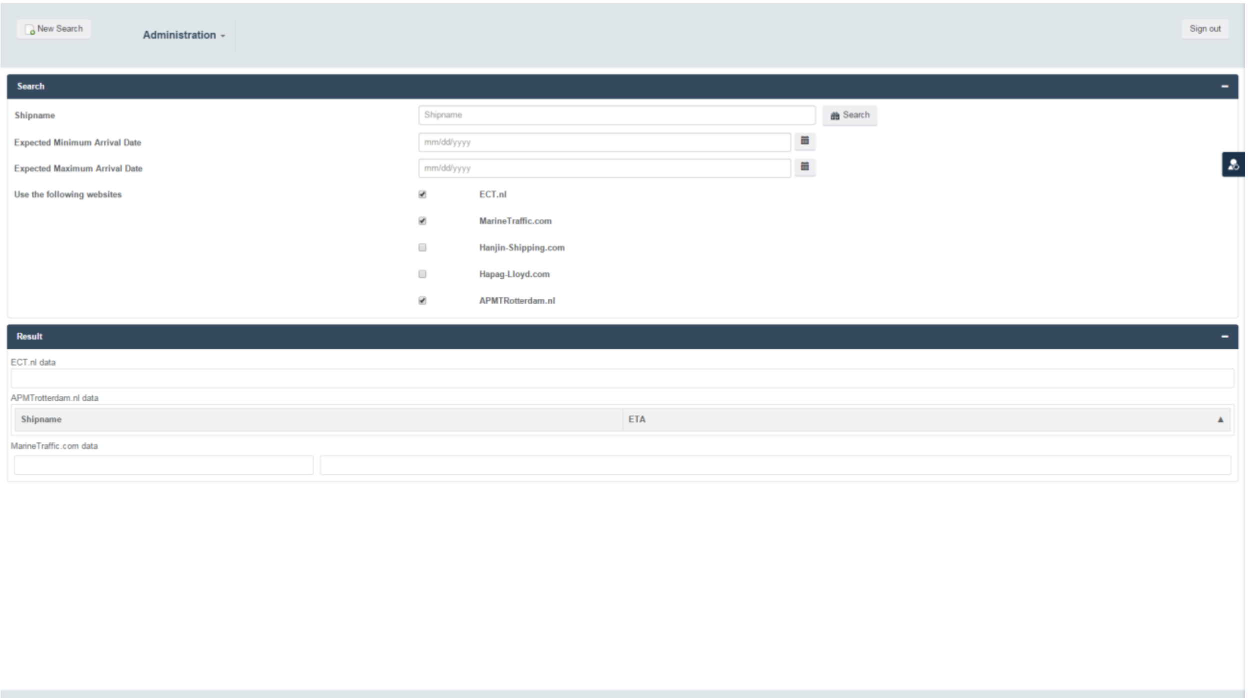

Figure 4.2 Application home screen when searching at APMTrotterdam.nl is checked

Figure 4.3 shows an overview of the results after searching for "Cosco Spain" on ECT.nl, MarineTraffic.com and Hapag-Lloyd.com. The results are shown in a clear way, only showing the results linked to the search query which is shown above. The location data retrieved from MarineTraffic.com is projected using the Google Maps module for Mendix, as such a projection is likely to tell a user lot more than longitude and a latitude values. As no results for the ship "Cosco Spain" can be found on Hapag-Lloyd.com, the table which should show results retrieved from Hapag-Lloyd shows "No results found". The possibilities for searching are not editable anymore after a search has been conducted. To start searching for another ship, the "New Search" button on the top left should be used. This ensures that each search query is unique.

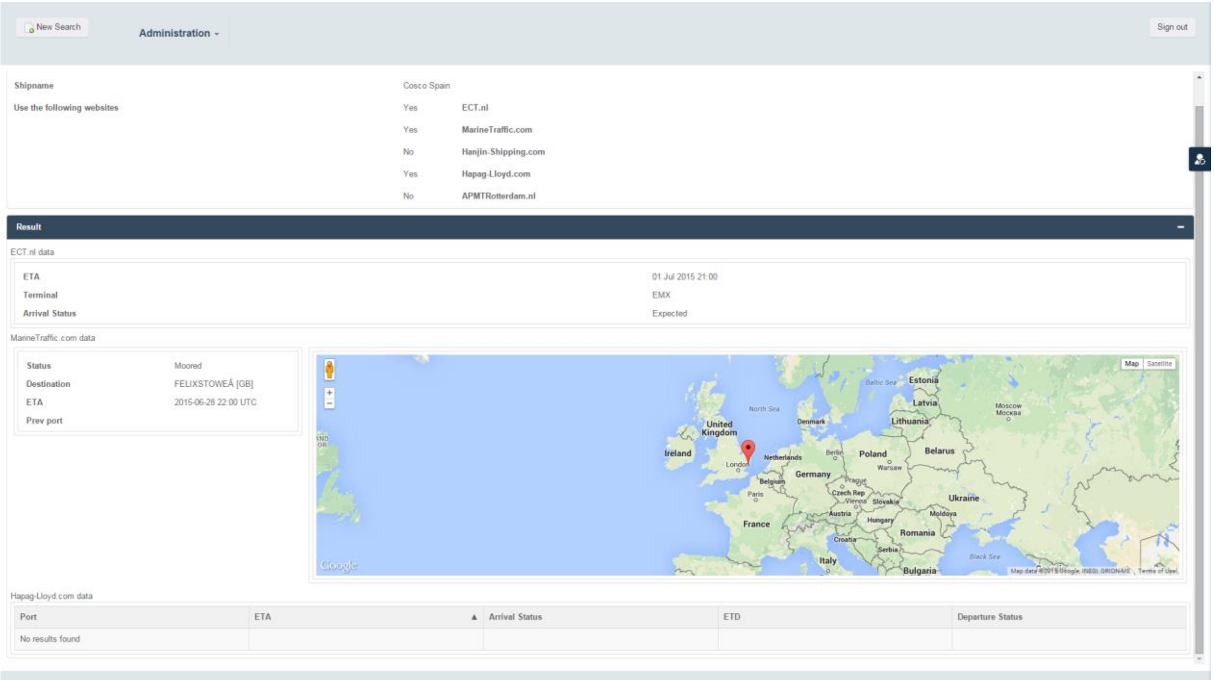

Figure 4.3 results overview

Last but not least, there is also an "Administration" tab available within the application, where accounts can be configured. This is shown in Figure 4.4. Roles for an Administrator, a UserAdministrator, a User and a Webservice have been configured. The difference between the roles lies in the fact that an Administrator can see and configure eMagiz settings within the application, a UserAdministrator can add and configure new users and a User can just use the application. The Webservice role is configured for communication with eMagiz, as discussed in Section 3.4.2.

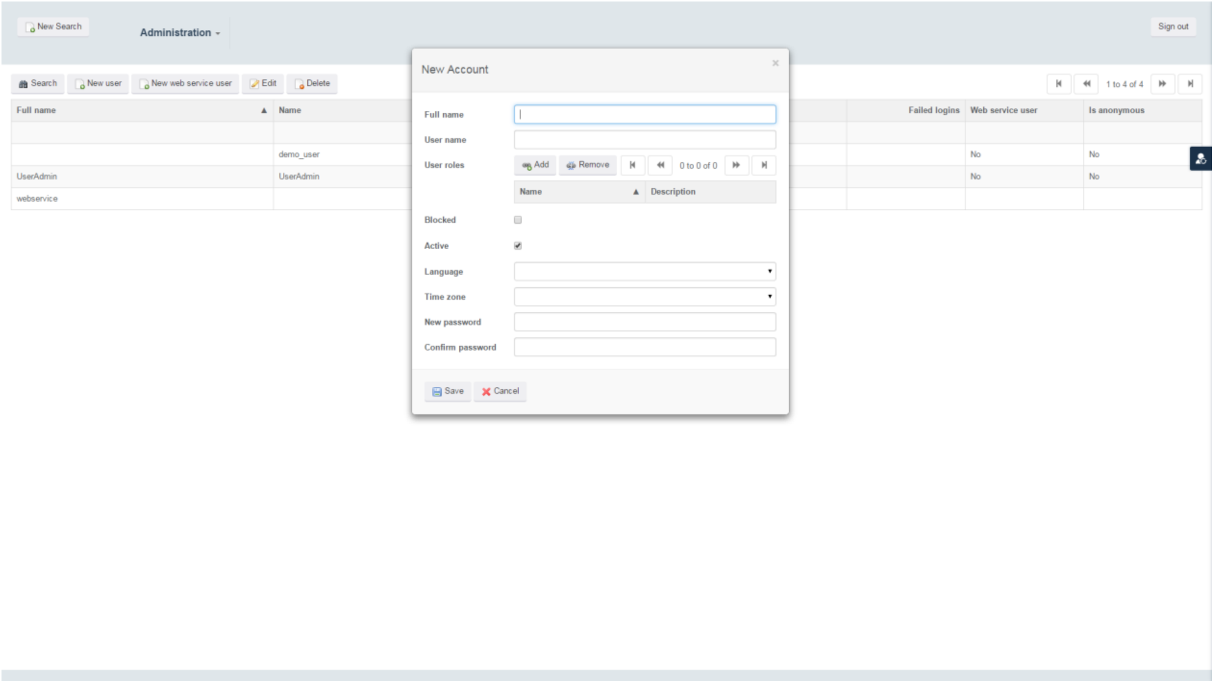

Figure 4.4 Account overview

5. Validation

Within this section, the designed solution will be tested against the initial research question and initial situation, to validate that it is indeed a solution for the problem. According to Peffers et Al. (2007) "such evaluation could include any appropriate empirical evidence or logical proof". In this case, a theoretical overview will be given and feedback will be listed. This will also answer the fourth sub question, "How can this technical solution be tested effectively?"

5.1 Theoretical

In Section 2.1, the current situation was described using an activity on node diagram. This activity on node diagram looked as follows:

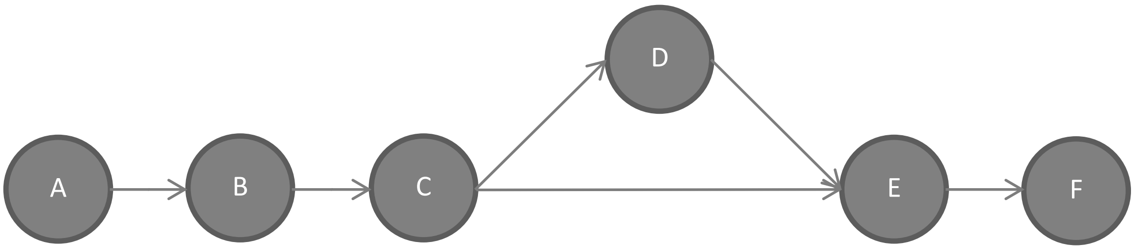

Figure 5.1 Activity on Node diagram – old situation

With the following activities for ETA definition:

A. Collect ship name from Microsoft Dynamics;
B. Fill in name at ect.nl and collect data;
C. Check if sufficient data is available;
D. Find ship via apmtrotterdam.nl and collect data (only executed when the result of C is negative);
E. Compare data to data in Microsoft Dynamics;
F. Process whether ETA within Microsoft Dynamics is still valid or not valid anymore.

And the following activities for ETD definition:

A. Collect ship name from Microsoft Dynamics;
B. Fill in name at MarineTraffic.com and collect data;
C. Check if sufficient data is available;
D. Collect data via ship planning available at hanjin.com or hapag-lloyd.com (only executed when the result of C is negative);
E. Compare data to data in Microsoft Dynamics
F. Process whether ETD within Microsoft Dynamics is still valid or not valid anymore.

Besides this, the objectives of a solution were defined. The main objective was to create a more straightforward method to define ETA values. Furthermore, another goal was to automate as much of the solution as possible, so that human interference was kept down to a minimum.

If we introduce the newly created situation with the use of the application, a different activity on node diagram can be drawn, which would then look like this:

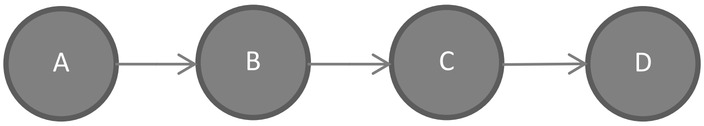

Figure 5.2 Activity on Node diagram – new situation

With the following activities:

A. Collect ship name from Microsoft Dynamics;
B. Fill in ship name and arrival date;
C. Compare data to data in Microsoft Dynamics
D. Process whether ETA/ETD in Microsoft Dynamics is still valid or not valid anymore.

Several activities from the initial diagram have been combined, creating a shorter diagram with fewer activities to be included. This proofs that the process has become a more straightforward process. Unfortunately, there was no possibility to test the prototype on any other source than locally on a laptop, which has negative impact on the performance. Therefore time measurements remain to be meaningless and are therefore not included.

If a link between the application and the Microsoft Dynamics environment can be achieved in the future, the activities as shown could be executed without human interference. This is expected to save a huge amount of time. Based upon the results of this research, both Cape Groep and HST considered proven that executing this process without human interference is possible.

5.2 Prototype

The prototype has been tested in several ways. At first, a status sheet listing 32 different ships has been used to test the application. Ship names were inserted in the application and compared to results shown on the selected websites. These results matched all 32 times.

During the timespan of this research, three visits to HST have been paid. During the first one, the current situation was observed and employees gave in depth information on the way they work. During the second meeting CloudScrape's functionalities were exposed as a possible way to automate data collection. During this meeting, it has been decided that linking the application to the current IT system at HST would take too much time, so that this would not be executed and that the focus would lay on executing the data collection part. During the third meeting the prototype as demonstrated in Chapter 4 has been shown. During the second and third meeting, HST's director and one executive staff member were present. Enthusiasm was slowly developing while progress was being made and several possible applications of the methods presented have been discussed. After showing the prototype, the statement that the process of tracking ETA and ETD data could be automated was considered proven.

The prototype was also demonstrated during a SynchromodalIT meeting where a combination of researchers and people from business life were present. Based upon the prototype, a discussion was held in which these people saw possibilities for full automation of the process by linking the systems together as well as possibilities to apply the theory on other areas than shipping.

6. Conclusions and Recommendations

According to Pfeffer's DSRM, the final step is to communicate the findings of an executed research. This final step is outlined in this chapter, together with recommendations, limitations and possibilities for future research.

6.1 Conclusion

The main goal of this research was to design an automated solution for tracking and tracing HST's sea freight in order to define the ETA as precise as possible. Transferred to a research question, this looks as follows:

"How can a real time track and trace and prediction system for deep sea vessels be developed?"

Within this research the focus lay on long distance shipping for deep sea vessels. Based upon findings in scientific literature, it is concluded that the best way to predict arrival times for these long distance trips is based upon historical data. A ship's trip usually goes from port A to port C via port B, if we define a point P in between point A and point B, the remaining time could be divided into the time to get from point P to port B, the time spend at port B and the time to get from port B to port C. If we base the time spend on these on historical data saved in a database, the times could be calculated using this formula:

$$ETA = t_{PB} + t_{BZ} + t_{transfer}$$

With:

$$t_{PB} = \frac{\sum_{i=1}^{n} t_i}{n}$$
$$t_{BZ} = t_{BC} + t_{CD} + \cdots + t_{YZ}$$
$$t_{transfer} = t_B + t_C + \cdots + t_Z$$

Location data is available as AIS data, of which broadcasting is obligatory for deep see vessels. This data is free to use, so near to real time location data for deep sea vessels is available. A database with historical data could be built by saving ships' locations and times, which might take some time.

Another possibility is to automate the current situation at HST and use data available at various websites to automatically track a ship's arrival time. By using a technique called web scraping, human surf behavior could be simulated and the required data could be collected based upon input values. The collection of this data is applied in an application, which collects the data from selected websites and gives a clear overview of this data. This application is made using CloudScrape, Mendix and eMagiz. The initial hypothesis that RapidMiner would provide the functionalities needed appeared to be false. By linking this functionality to HST's current IT system, the process of ETA and ETD definition could be automated and by saving the data gathered in this automated way a database with historical data could be build. This way, predictions can be made based upon two sources in the future, leading to more accurate predictions.

6.2 Recommendations

On the short term it is recommended to test the developed application on a more advanced system than locally on a laptop. This way, a comparison between the time spend on ETA and ETD checking in the current situation could be compared to the time spend when using the application. Although using the application provides extra functionalities as saving data, eventually the real profit lies within time saving. Therefore it is recommended to start using the application when this saves time.

On the long term, it is highly recommended to investigate the possibilities of combining the possibilities of web scraping with a direct link to their current IT system to achieve a fully automated way of predicting arrival times. If search values could be extracted directly from that IT system and results pushed back to the system, it would save lots of time. By adding notification functionality for changes in ETAs exceeding certain limits, employees would still be able to take action when desired.

The automated system could save data processed so that HST could also calculate ETAs based upon historical data in the future. By combining these data sources, there is also a possibility to check the reliability of the ETAs. Furthermore, this data might hold valuable information on for instance punctuality of various shipping companies or other research fields, so analyzing this data would be wise.

6.3 Limitations

A number of limitations have already been listed when describing the scope within Section 1.6. By focusing on the methods at HST, an application for HST has been developed, but this application might be useless for other shipping companies. Furthermore, we have assumed that data was valid and even though a method to test this in the future has been described, this functionality has not been applied yet.

Furthermore, the various recommended systems have been tested in an application, but especially CloudScrape has not been tested in handling bigger amounts of data and more frequent requests at this point. This means that although it is expected that CloudScrape can handle more frequent requests, this cannot be concluded and errors might arise.

6.4 Future research possibilities

Future research possibilities can be divided into two areas, which are expanding and deepening. At first, the results of this research might not only be applicable in transportation via water, but also at for example transportation by truck, train or airplane. Certain websites listing data for airfreight are known to exist, however other data might be harder to reach and thus harder to include. If various transport methods can be supported, one can also investigate the moment at which freight switches from for instance ship to truck and possibly include information on this subject as well.

Besides this, the results of this research could be deepened by investigating the results once a connection with the IT system of HST has been established. With a good set of data, interesting results may be found when investigating calculation methods or when analyzing data. This could lead to better algorithms when predicting arrival times in which for instance real time weather- or traffic forecasts can be included to achieve even better predictions in the future.

These possibilities have been added to the application's architecture, which is shown in figure 6.1. The red blocks represent future possibilities.

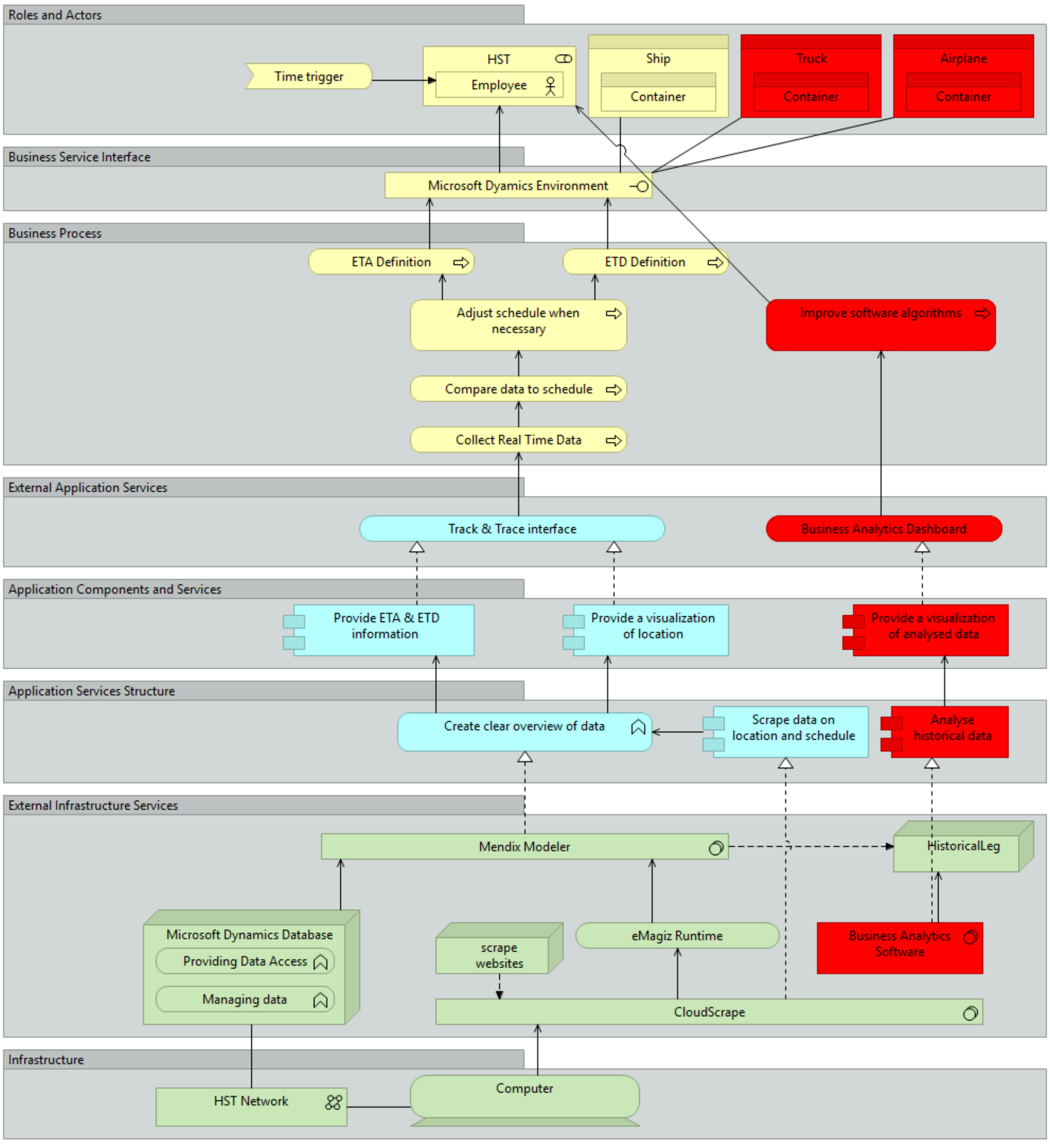

Figure 6.1 Architecture including future research possibilities (in red)

Appendix I: References

Literature

1. Fagerholt, K., Heimdal, S. I., & Loktu, A. (2000). Shortest path in the presence of obstacles: An application to ocean shipping. *Journal of the operational research society*, 683-688.
2. Heywood, C., Connor, C., Browning, D., Smith, M. C., & Wang, B. (2009, April). GPS tracking of intermodal transportation: System integration with delivery order system. In *Systems and Information Engineering Design Symposium, 2009. SIEDS'09.* (pp. 191-196). IEEE.
3. Iacob, M. E., Meertens, L. O., Jonkers, H., Quartel, D. A. C., Nieuwenhuis, L. J. M., & Van Sinderen, M. J. (2014). From enterprise architecture to business models and back. *Software & Systems Modeling*, *13*(3), 1059-1083.
4. Kanaoka, K., Fujii, Y., & Toyama, M. (2014, July). Ducky: a data extraction system for various structured web documents. In *Proceedings of the 18th International Database Engineering & Applications Symposium* (pp. 342-347). ACM.
5. Kara, B. Y., & Tansel, B. Ç. (2001). The latest arrival hub location problem. *Management Science*, *47*(10), 1408-1420.
6. Lee, C. Y., & Meng, Q (2015). *Handbook of Ocean Container Transport Logistics.* Switzerland: Springer.
7. Lo, H. K., McCord, M. R., & Wall, C. K. (1991). Value of ocean current information for strategic routing. *European journal of operational research*,*55*(2), 124-135.
8. Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, *24*(3), 45-77
9. Ristic, B., La Scala, B., Morelande, M., & Gordon, N. (2008). Statistical analysis of motion patterns in AIS data: Anomaly detection and motion prediction. *Information Fusion, 2008 11th International Conference on* (pp. 1-7). IEEE.
10. Szelangiewicz, T., Wiśniewski, B., & Żelazny, K. (2014). The influence of wind, wave and loading condition on total resistance and speed of the vessel. *Polish Maritime Research*, *21*(3), 61-67.
11. Vernimmen, B., Dullaert, W., & Engelen, S. (2007). Schedule unreliability in liner shipping: origins and consequences for the hinterland supply chain.*Maritime Economics & Logistics*, *9*(3), 193-213.
12. Wijaya, W. M., & Nakamura, Y. (2013, December). Predicting Ship Behavior Navigating through Heavily Trafficked Fairways by Analyzing AIS Data on Apache HBase. In *Computing and Networking (CANDAR), 2013 First International Symposium on* (pp. 220-226). IEEE.

Terms	Date	Search Engine	Results	Limited by/found via	Limited to	Used
“Shipping” AND “Estimated Time of Arrival”	18-04	Scopus.com	6	Read the abstracts, removed articles on inland shipping, as this is not the scope of my research. One article was not available as download. One article appeared not useable after reading.	2	- GPS Tracking of intermodal transportation system integration (2009) - Shortest Path in the Presence of Obstacles: An Application to Ocean Shipping (2000) - Smart Container Management (2011)
Article Title	20-04	Scopus.com	1	Referenced in “Shortest Path in the Presence of Obstacles: An Application to Ocean Shipping”.	1	- Value of ocean current information for strategic routing
Article Title	20-04	Scholar.google.nl	1	Recommended by Maria Iacob	1	- A Design Science Research Methodology for Information Systems Research
“Shipping” AND “Weather” AND “Speed”	21-04	Scopus.com	69	Limited to subject Computer Sciences Read Abstracts, removed articles not fully available.	0	
“Shipping” AND “Weather” AND “Speed”	21-04	Scopus.com	69	Excluded “Computer Sciences”, “Environment”, “Energy”, “Agriculture” AND “social sciences”. Limited to 2011 onwards Limited by reading titles and abstracts Excluded articles not fully available	20	- The influence of wind, wave and loading condition on total resistance and speed of the vessel. (2014)
“Estimated time of arrival” AND “weather”	21-04	Scopus.com	9	Read abstracts, removed articles not available	1	- The optimization of ship weather-routing algorithm based on the composite influence of multi-dynamic elements (II): Optimized routings (2015)
Article Title	21-04	Scopus.com	2	Removed articles already	1	- The optimization of ship weather-routing

				found		algorithm based on the composite influence of multi-dynamic elements (2013)
Suggested by Supervisor	21-04	-	9	Read all the articles, useful as background information at least	9	- Towards an approach for long term AID-based prediction of vessel arrival times - Vessel Track Recovery with incomplete AIS data Using Tensor CANDECOM/PARAFAC Decomposition (2014) - Predicting Ship Behavior Navigating Through Heavily Trafficked Fairways by Analyzing AID Data on Apache HBase (2013) - Unsupervised Learning of Maritime Traffic Patterns for Anomaly Detection (2012) - Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction (2013) - Application of Dempster-Shafer Theory for the Quantification and Propagation of the Uncertainty Caused by the use of AIS (2013) - Statistical Analysis of Motion Patterns in AIS Data Anomaly Detection and Motion Prediction (2008) - A framework of Moving Behavior Modeling in the Maritime Surveillance (2011) - Anomaly Detection in Sea Traffic – A comparison of the Gaussian Mixture Model and the Kernel Density Estimator (2009)
Suggested by supervisor	07-05	-	1	-	1	- Schedule Unreliability in Liner Shipping: Origins and Consequences for the Hinterland Supply Chain (2007)
Suggested by colleague	11-05	-	2	-	2	- From enterprise architecture to business models and back (2012) - ArchiMate® in de praktijk v4.0 (Dutch) (2012)

“Web Scraping”	19-05	Scopus.com	33	Read titles, removed ones not available	1	-	Effective Web data extraction with standard XML technologies (2002)
“Web Scraping” AND “CSS”	22-05	Scopus.com	1	-	1	-	Ducky: A Data Extraction System for Various Structured Web Documents (2014)

Websites

1. The Open Group, *An Introduction to ArchiMate® 2* (2012). Retrieved May 18, 2015, from https://www.opengroup.org/archimate/2.0/ArchiMate2_intro.pdf
2. The Open Group, *ArchiMate® 2.1 Specification* (2013). Retrieved May 18, 2015, from http://pubs.opengroup.org/architecture/archimate2-doc/
3. Cory Janssen, *Web Scraping.* Retreived June 9 2015 from http://www.techopedia.com/definition/5212/web-scraping
4. *Cloudscrape Overview (2015).* Retrieved June 4, 2015, from http://cloudscrape.com/

Appendix II: HST Questionnaire

Questions on the executed estimations:

- How do employees currently define the ETA value?
  - Which inputs are used?;
  - Where does this data come from?;
  - How is this data currently saved?;
  - Is the ETA defined on for request by a customer or following a certain time schedule?;
- Do times needed to cover a certain distance differ a lot? If so, what influences this? (shiptype, weather, etc);
- How do delays in the beginning of the schedule influence the rest of the schedule?

Questions on an application concerning automating this process:

- What criteria are important within an automated process? Mainly automating the current process, or improving estimations as well?;
- What part of the process is important to automate? For instance, is it annoying if an employee still has to enter a ship name?

Appendix III: API example

Input:

```
POST /api/runs/7c26a1a4-69ee-46b6-bd60-5d410c69625b/execute/inputs/wait HTTP/1.1
Host: app.cloudscrape.com
X-CloudScrape-Access: 3b0d2b1971eb252a27af8d248196cff4
X-CloudScrape-Account: e2e383a6-ae6a-40de-9b5b-2bcc462d0e75
Accept-Charset: utf-8
Content-Type: application/json
Accept: application/json

{ "Shipname": "Cosco Spain" }
```

Output:


```
{
  "headers": [
    "Shipname",
    "list",
    "Port",
    "Terminal",
    "ArrivalTime",
    "DepartureTime",
    "status"
  ],
  "rows": [
    [
      "Cosco Spain",
      null,
      "SHANGHAI",
      "SHANGHAI SHENGDONG INTL CONTAINER TERMINAL (YS1)",
      "2015-08-06 10:00",
      "2015-08-07 02:00",
      null
    ],
    [
      "Cosco Spain",
      null,
      "SUEZ (EL SUWEIS)",
      "SUEZ CANAL",
      "2015-07-16 18:00",
      "2015-07-17 16:00",
      null
    ],
    [
      "Cosco Spain",
```

```
    null,
    "ROTTERDAM",
    "EUROMAX TERMINAL",
    "2015-06-30 15:00",
    "2015-07-02 04:00",
    null
  ],
  [
    "Cosco Spain",
    null,
    "HAMBURG",
    "CTT HHLA CONTAINER TERMINAL TOLLERORT",
    "2015-07-03 01:00",
    "2015-07-05 02:00",
    null
  ],
  [
    "Cosco Spain",
    null,
    "QINGDAO,CHINA",
    "QINGDAO QIANWAN UNITED CONTAINER TERMINAL CO., LTD",
    "2015-05-27 14:50",
    "2015-05-28 13:00",
    null
  ],
  [
    "Cosco Spain",
    null,
    "DALIAN,CHINA",
    "DALIAN PORT CONTAINER TERMINAL CO.,LTD",
    "2015-05-25 23:00",
    "2015-05-26 18:00",
    null
  ],
  [
    "Cosco Spain",
    null,
    "SHANGHAI",
    "SHANGHAI SHENGDONG INTL CONTAINER TERMINAL (YS1)",
    "2015-05-30 02:30",
    "2015-05-30 20:30",
    null
  ],
  [
    "Cosco Spain",
```

null,
"XINGANG",
"TIANJIN HARBOR CONTAINER COMPANY",
"2015-05-24 05:00",
"2015-05-25 07:30",
null
],
[
"Cosco Spain",
null,
"NINGBO,ZHEJIANG",
"NINGBO YUANDONG TERMINALS LIMITED",
"2015-08-16 08:00",
"2015-08-17 06:00",
null
],
[
"Cosco Spain",
null,
"DALIAN,CHINA",
"DALIAN PORT CONTAINER TERMINAL CO.,LTD",
"2015-08-11 04:00",
"2015-08-12 04:00",
null
],
[
"Cosco Spain",
null,
"XINGANG",
"TIANJIN HARBOR CONTAINER COMPANY",
"2015-08-09 04:00",
"2015-08-10 11:00",
null
],
[
"Cosco Spain",
null,
"ANTWERP",
"ANTWERP GATEWAY KAAI 1700",
"2015-07-06 03:00",
"2015-07-07 08:00",
null
],
[
"Cosco Spain",

```
            null,
            "FELIXSTOWE",
            "PORT OF FELIXSTOWE",
            "2015-06-28 20:00",
            "2015-06-30 07:00",
            null
        ],
        [
            "Cosco Spain",
            null,
            "SUEZ (EL SUWEIS)",
            "SUEZ CANAL",
            "2015-06-20 03:00",
            "2015-06-20 21:00",
            null
        ],
        [
            "Cosco Spain",
            null,
            "SINGAPORE",
            "PSA CORPORATION LIMITED",
            "2015-06-06 19:45",
            "2015-06-07 16:00",
            null
        ],
        [
            "Cosco Spain",
            null,
            "NINGBO,ZHEJIANG",
            "NINGBO YUANDONG TERMINALS LIMITED",
            "2015-05-31 22:00",
            "2015-06-01 18:30",
            null
        ],
        [
            "Cosco Spain",
            null,
            "QINGDAO,CHINA",
            "QINGDAO QIANWAN UNITED CONTAINER TERMINAL CO., LTD",
            "2015-08-12 22:00",
            "2015-08-13 19:00",
            null
        ]
    ]
}
```

Appendix IV: Collected data

ECT.nl

Data name	ETA	ETD	Terminal	ArrivalStatus
Description	Estimated time of arrival in Rotterdam	Estimated time of departure from Rotterdam	Specific terminal where the ship will arrive	Lists whether the ship has already arrived or not.

APMTRotterdam.nl

Data name	CarrierNaam	ETA	ETD
Description	Name of the carrier (as APMT uses a minimum and maximum arrival date as input)	Estimated time of arrival in Rotterdam	Estimated time of departure from Rotterdam

MarineTraffic.com

Data name	Location	Status	Destination	ETA	PrevPort
Description	Longitude and latitude of current location	Lists whether a ship is underway or moored	Destination of the ship's next stop	Estimated time of arrival for this stop	Previous stop

Hanjin-shipping.com

Data name	Port	Terminal	ETA	ETD
Description	Port name for the stop of a ship	Terminal where the ship will stop	Estimated time of arrival for this port	Estimated time of departure for this port

Hapag-Lloyd.com

Data name	Port	ArrivalDate	ArrivalTime	Arrival Status	DepartureDate	DepartureTime	Departure Status
Description	Port name for the stop of a ship	Estimated or actual arrival date for this port	Estimated or actual arrival time for this port	Lists whether the ship has already arrived or not.	Estimated or actual departure date for this port	Estimated or actual departure time for this port	Lists whether the ship has already departed or not.

Appendix V: Hapag-Lloyd XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
   version="2.0"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   xmlns:json="http://www.emagiz.com/ns/json/1.0/"
   exclude-result-prefixes="xs xsl json">

   <xsl:template match="/json:object">

         <ETAlist>
               <xsl:apply-templates select="json:array[@name='rows']/json:array"/>
         </ETAlist>
   </xsl:template>

         <xsl:template match="json:array">
               <ETA>
                     <name><xsl:value-of select="json:*[1]/@value"/></name>
                     <ArrivalStatus><xsl:value-of select="json:*[5]/@value"/></ArrivalStatus>
                     <source>hapag</source>
                     <Port><xsl:value-of select="json:*[2]/@value"/></Port>
                     <ArrivalDate><xsl:value-of select="json:*[3]/@value"/></ArrivalDate>
                     <ArrivalTime><xsl:value-of select="json:*[4]/@value"/></ArrivalTime>
                     <DepartureDate><xsl:value-of select="json:*[6]/@value"/></DepartureDate>
                     <DepartureTime><xsl:value-of
select="json:*[7]/@value"/></DepartureTime>
                     <DepartureStatus><xsl:value-of
select="json:*[8]/@value"/></DepartureStatus>
               </ETA>
         </xsl:template>
</xsl:stylesheet>
```