Port Call Synchronization

Bachelor's thesis in Computer Science and Engineering

SIMON CEDER
NIKLAS GUSTAFSSON
HENRIK NILSSON
ARI VON NORDENSKJÖLD
PONTUS STJERNSTRÖM
JOHANNA TORBJÖRNSSON

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2018

Abstract

The shipping industry is at the time of writing undergoing a process of modernisation and digitalisation. Port Collaborative Decision Making (PortCDM) is part of Sea Traffic Management (STM), a research project aiming to show how modern maritime information sharing can be implemented and how it can benefit actors in the shipping industry.

Port Call Synchronization has been developed as a proof-of-concept service for automating the generation of recommended times of arrival for vessels arriving at ports, based on the greater access to information that PortCDM enables. This information makes it possible to implement digital representations of port actors, and simulate the process by which they arrive at consensuses regarding what time it is suitable for a vessel to enter a port.

A rudimentary system for optimizing berth schedules has also been developed, based on the automated system.

Keywords: PortCDM, Maritime IT, Sea Traffic Management, Collaborative Decision Making, Information Sharing, Machine Learning, Scheduling

Sammandrag

Sjöfartsindustrin genomgår i skrivande stund en moderniserings- och digitaliseringsprocess. Port Collaborative Deceision Making (PortCDM) är en del av Sea Traffic Management (STM), ett forskningsprojekt som avser att visa hur modern maritim informationsdelning kan implementeras och hur det kan gagna aktörer inom sjöfartsindustrin.

Port Call Synchronisation har utvecklats som en *proof-of-concept*-tjänst för att automatisera skapandet av rekommenderade ankomsttider för fartyg på väg in till hamnar, baserat på den högre informationsåtkomsten som PortCDM möjliggör. Denna information gör det möjligt att implementera digitala representationer av hamnaktörer, och simulera den process som används för att dem emellan komma fram till konsensus angående vilken hamnankomsttid som är passande för ett fartyg.

Ett grundläggande system för att optimera kajscheman har också utvecklats, baserat på det automatiserade systemet.

Nyckelord: PortCDM, Maritim IT, Sea Traffic Management, Collaborative Decision Making, informationsdelning, maskininlärning, schemaläggning

Contents

List of Figures

List of Tables

Abbreviations and Special Terminology

AIS: Automatic Identification System

API: Application Programming Interface

ANN: Artificial neural networks

ATA: Actual Time of Arrival

ATFCM: Air Traffic Flow and Capacity Management

ECDIS: Electronic Chart Display and Information System, a digital system for vessel navigation that can communicate with PortCDM.

ETA: Estimated Time of Arrival

ETD: Estimated Time of Departure

FCFS: First-come, first-served, a scheduling algorithm

HTTP: Hyper Text Transfer Protocol

IMO: International Maritime Organisation, an organisation issuing unique vessel identification numbers

Java EE: Java Enterprise Edition

JSON: JavaScript Object Notation

kNN algorithm: k-Nearest Neighbours algorithm

ML: Machine Learning

PACT: Port Call Actor Coordination Tool, a tool for handling PortCDM messages

PCA: Principal component analysis

PortableCDM: A mobile app handling PortCDM messages

Port call: A vessel's visit to the port

PortCDM: Port Collaborative Decision Making, an implementation of a message standard adapted to port related events

RESTful API: Representational State Transfer API

RTA: Recommended Time of Arrival

SGD: Stochastic gradient descent

STM: Sea Traffic Management, a research project aiming to show how modern maritime information sharing can be implemented and how it can benefit actors in the shipping industry

TTA: Target Time of Arrival

WAR-file: Web Application Runnable file

1

Introduction

The shipping industry is responsible for carrying about 80% of global trade by volume [1]. Importing and exporting goods would simply not be possible as we know it today without cargo shipping lines. The international shipping standards and laws have their roots several hundred years back and are primitive compared to those of other similar industries [2]. It is arguably one of our oldest industries, since sea transport was an important factor in the rise of the first civilisations [3].

Some of the greatest challenges for the industry today arise as a result of the complex nature of international shipping, where many countries and organisations need to agree on the same rules, procedures and which IT systems to use [4]. As a result, industry progress and development is generally slow [5]. Many laws and standards are based on ancient conventions [2], which inhibits technological development within the industry.

1.1 Seafaring Today

During a port call, i.e. a vessel's visit to a port, there are many actors involved in handling the vessel, such as pilots, tugboats and terminals. The actors need to agree upon when events should occur, and synchronize the procedure, see Figure 1.1. The communication between such actors, including the vessel itself, is currently handled using phone calls, emails, or even fax messages. There are ports where some of the operators use IT systems for communication [6] to a certain extent, but they are not regulated by any international standards, and communication is generally cumbersome. The lack of efficient communication regularly causes long waiting times for vessels [7].

1.2 Fuel Consumption and Environmental Impact

The long waiting times can lead to vessels setting higher speeds at sea in order to compensate for delays [8]. In most ports, it is also the case that berth slots are allocated according to an antiquated principle known as "first-come, first-served", i.e. if only a single slot is available, it is given to the vessel that arrives first [8]. This

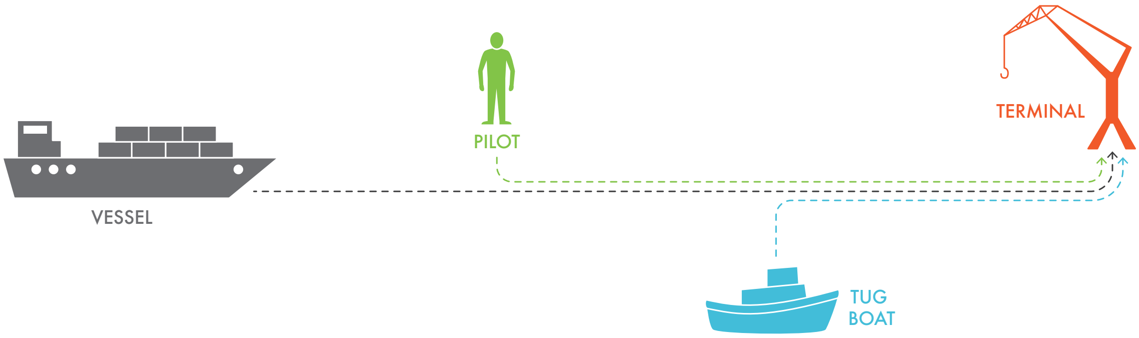


Figure 1.1: Some of the actors in a port.

regularly leads to vessels racing each other to the port. Fuel consumption is drastically increased when vessels travel at higher speeds than what they are designed for [9]. Lying anchored outside the port also results in a higher environmental impact without increasing the transport supply [10], partly due to the fact that the engines are usually kept running while waiting for a slot. The reason for why they are kept running is that restarting them after they have cooled down can lead to increased fuel consumption, and it is often preferred to keep certain systems running at all times.

1.3 Port Collaborative Decision Making

Port Collaborative Decision Making (PortCDM) is part of Sea Traffic Management (STM), which is a project financed by the European Union with the goal of increasing the efficiency of sea traffic around the world [8]. PortCDM focuses on the port-related aspects of STM, and will act as a standard for ports to follow in order to enable smooth communication. This is done by letting port actors share standardised messages in real time over the internet. These messages contain time-stamp data of the actors' planned and completed actions and can be accessed by all actors in the port. An implementation of PortCDM has been developed and serves as a proof-of-concept of the standard [11]. It is this implementation that is referred to in this report when PortCDM is mentioned.

1.4 Port Call Synchronization

An important aspect of a port call occurs when actors agree on a recommended time of arrival (RTA) to the port. Establishing an RTA allows the vessel to adapt its speed, and actors to plan and prepare properly. The captain of the vessel sends a suggested time, a target time of arrival (TTA), and the port responds with an RTA, as shown in Figure 1.2. It is possible to respond to TTAs via the graphical interfaces for PortCDM, which are discussed in Section 2.4. However, the agreement process is still manual and actors need to agree on what time to respond with.

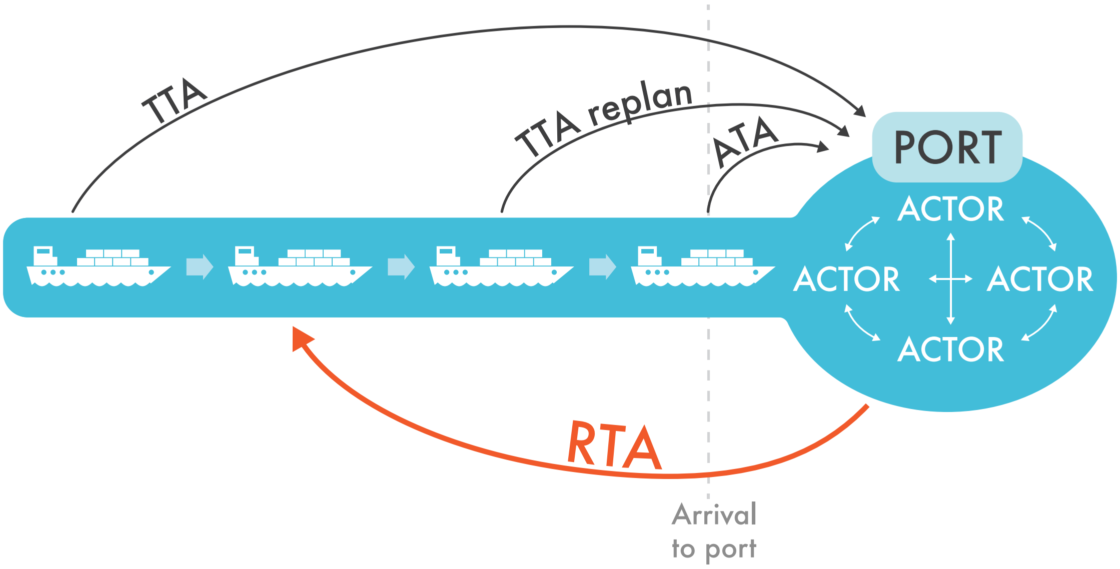


Figure 1.2: The role of port call synchronization. The vessel updates its TTA and an RTA is sent as a response. Finally, an Actual Time of Arrival (ATA) is sent, when the vessel has arrived at the port.

1.5 Purpose and Objective

The goal of this project has been to produce a proof-of-concept service that reduces the manual aspects of the agreement process between actors by automatically generating suitable RTAs. These RTAs would then serve as aids for port actors, saving time and thereby potentially reducing delays. It is hoped that if vessels receive an RTA well ahead of arriving at the port, their speed can be optimized in order to minimise the time spent anchored outside the port.

An additional goal has been to optimize the scheduling of port calls, by exposing potential improvements to the order of port calls, and consequently RTAs. The main objective of such rescheduling would be to reduce the amount of time that vessels spend anchored outside a port.

Since it is believed that these concepts are valuable for ports and have great potential to increase efficiency and reduce the environmental impact of the shipping industry, it is hoped that the project can expose them to real port actors and businesses involved with maritime IT solutions. As part of the evaluation of the results of the project, another goal of the project has been to gather the input of persons experienced within the sea traffic industry in order to assess the relevance of the results for the industry.

As the systems developed in order to meet these goals were conceived as API services, the terms *service(s)*, *system(s)*, and *product(s)* will be used interchangeably throughout the rest of the report.

1.6 Scope and Delimitations

The scope of this project is to present a proof-of-concept implementation of how port call synchronization could be done in the future. It also serves the purpose of showing that synchronizing and optimizing ports can potentially be done automatically in the future, if digitalisation is pushed forward.

The scope of the project includes the following:

- Use of real-time, real-world port data from the PortCDM platform
- Use of real-world AIS data
- Generation of an RTA, as a proposal to the user
- A runnable application that optimizes each berth in the port
- An API that provides the RTA automation and the berth optimization
- A graphical demo interface to present the possibilities of the API

The following is outside the scope of the project:

- Graphical User Interface (beyond the basic demo interface)
- RTAs that can be used in the real world
- Providing RTAs in the port call message format
- Use of port specific data such as; mooring schedules, specific berth information, geographical port limitations, port rules and limitations and port opening hours.
- Use of weather data
- Vessel limitations
- Economical consequences of optimization
- API documentation

The factors taken into consideration for synchronization and optimization are based on relevance and accessibility. For a more thorough discussion, see Section 7.1.1.

Conducting a pre-study in order to evaluate industry demand for services like PCS or gather design input before initiating development was considered to be outside the scope of the project.

1.6.1 Limitations of PortCDM

The PortCDM project is only a proof-of-concept, which means that data in PortCDM do not fully reflect all aspects of the real port. Although enough data is available to provide an interesting result, important information is still missing as only a fraction of port actors utilise PortCDM. Since there is currently no data quality assurance, the data in PortCDM varies in quality and might sometimes be incorrect.

1.6.2 Unpredictable Events

In the shipping industry, many events are hard to predict. One reason is the fact that captains have the authority to take their vessel wherever they see fit. Captains may defy recommendations from PortCDM or refrain from reporting their actions to PortCDM, or other IT services, without breaking any laws. They could also refrain from reporting their actions correctly to PortCDM. Predicting these types of events is outside the scope of this project.

Furthermore, some vessels can not comply with RTAs due to fines which are charged for not arriving within a certain time. Other vessels have pre-booked slots at berths and simply can not change their RTA, no matter how beneficial it would be. Since no information regarding these restrictions is available, they are considered to be outside the scope of the project as well.

1.7 Related Work

PortCDM and STM are relatively new research projects. While a fair amount of academic work has been published as a part of those research projects, the field of collaborative decision making and information sharing at sea as such is in a nascent stage and just emerging at the time of writing [12]. One of the most important research topics that PortCDM and STM are concerned with is the automation of communicational procedures [13]. Automating the generation of RTAs is not a problem that has been explored previously, and as such, the project is an original contribution to the research within the field.

A lot of research on the topic of scheduling in general has been conducted, including many aspects related to the shipping industry [14][15]. The Berth Allocation Problem [16], for example, is a well-known research problem. Most of this research is, however, purely theoretical, or assumes that the resources are controlled by a single entity. By leveraging the benefits of collaborative information sharing, efficient scheduling can be applied to more complex systems and with a better understanding of contingent factors. [17] discusses this from a theoretical perspective.

Air Traffic Flow and Capacity Management (ATFCM) is a more mature research field, and also served as an inspiration for launching the STM and PortCDM research projects [18]. Within ATFCM, problems comparable to the ones explored in this project have been researched more thoroughly [19]. For example, Computer-assisted Slot Allocation is a relatively old and well-researched field [20]. [21] discusses problems similar to those explored in this project in the context of ATFCM. The nature of air traffic flow and the systems that have been developed for controlling it are, however, not similar enough to the shipping industry and PortCDM for any direct use to be made of the results developed within that field for the purposes of this project.

2

Port Procedure and PortCDM

This chapter offers information regarding common procedures for a vessel's port visit. It also explains PortCDM in greater detail. This is useful for readers unfamiliar with the shipping industry or the PortCDM project, and can be helpful for understanding the following chapters.

2.1 The Port and Port Actors

Ports exist all around the globe with capabilities ranging from a few hundred vessels annually to 130000 as is the case in the Port of Singapore – the world's busiest port [22]. Most ports consist of several docks, containing berths for passengers, bulk cargo, or fluids.

A lot of different companies and organisations are involved in port operations, and they all constitute one or several *port actors*. When a port call occurs (i.e. a vessel arrives at the port, gets served and departs), the port actors need to collaborate to achieve maximal efficiency. The actors that can be found in most ports are described below, based on [23], while Section 2.2 outlines the typical port call process, which is also illustrated in Figure 2.1.

- The **Shipping Agent** is a single person who acts as the contact between the port and the vessel and is responsible for all communication between them. The agent is expected to be up to date with the latest information about the port call. It is also the agent that books other required actors such as pilotage and towage.
- **Maritime Pilot Organisations** employ pilots that assist incoming vessels in manoeuvring them to the berth. Pilotage is normally booked a few hours ahead by the agent.
- **Towage Operators** own tugboats and escort tugs which assist incoming vessels and help manoeuvre them to the berth by pushing or pulling the vessel. Tugboats are normally booked a few hours ahead by the agent.
- **Terminals** are responsible for the loading and unloading of cargo. For this purpose they own one or more berths which are usually booked before the vessel has departed from its departure port.

- **Mooring Operators** moor the vessel to the berth so that cargo operations can commence.
- The **Port Authority** Grants vessels permission to enter the coastal territory.
- **Sludge Operators** take care of sludge – such as oil residues and crew waste – when the vessel is berthed.

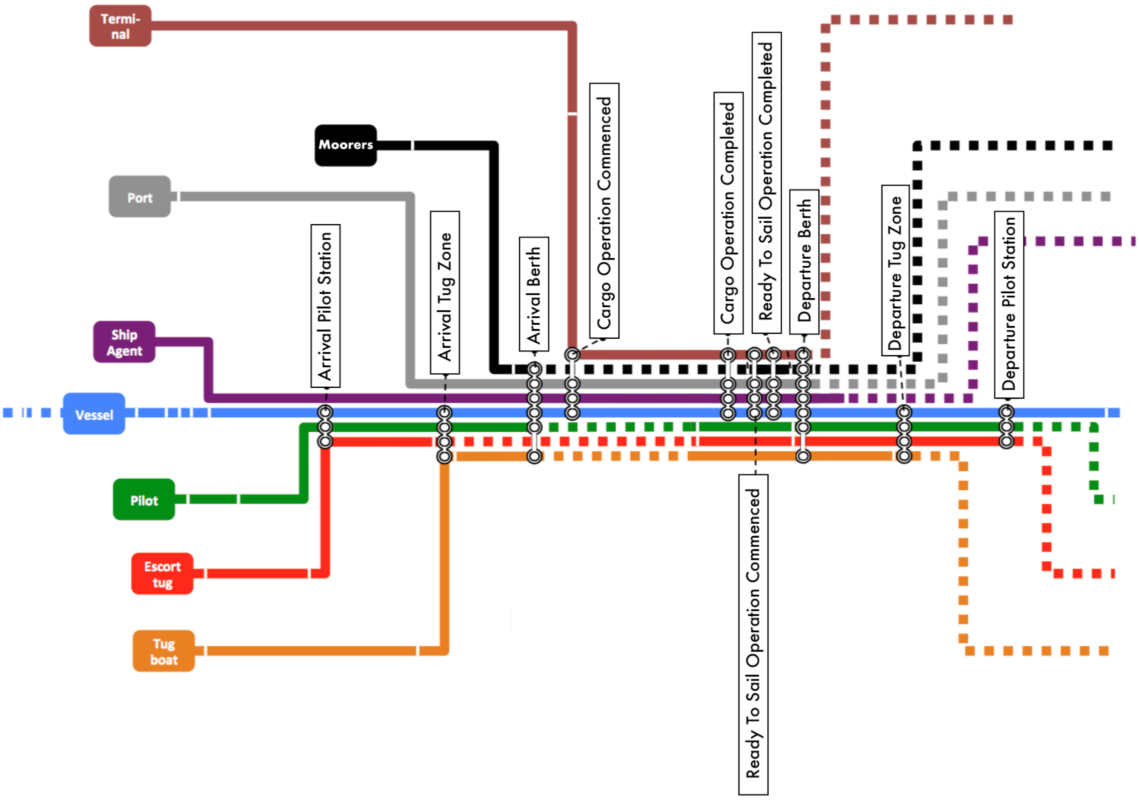


Figure 2.1: The actors involved in a port call. The white nodes indicate points at which the actors must coincide. Dashed lines indicate when the actors are not actively involved. Modified figure taken from [11], with permission.

2.2 The Port Call Process

The following section describes a common port call process, based on information from [23] and [24].

The port call process dictates that all involved actors should come to an agreement that satisfies all of them. Since complete cooperation is required, it does not matter if all actors but one can reach an agreement. In practice the agreement process is currently administrated by a shipping agent that tries to establish a consensus in terms of a target time of arrival between the vessel and all the actors that constitute the port.

When the vessel begins its voyage, a targeted time of arrival is communicated to the agent. At this point, the agent can start planning the actual port call. This includes booking the pilot and tugboats, and communicating with other port actors. The agent usually contacts the targeted berth's terminal a few hours before arrival for information about equipment, average berth duration, and other practical details, in order to calculate the duration of the port visit. This information is important for booking tugboats and pilots for departure as well as for subsequent port visits. The agent keeps all involved actors informed and communicates new information until the vessel arrives.

If all goes well, the vessel is approved by the port authority and arrives at the pilot station. If a pilot is needed, the pilot boards the vessel in order to steer it to its assigned berth. The escort tugs, if required, escort the vessel from the pilot station to the tug zone. From the tug zone, the required number of tugboats commence the towing while maintaining communication with the pilot to allow for a smooth transfer to the berth. When the berth is reached, the vessel is moored, and the process of unloading and/or loading cargo can commence. This process is illustrated in Figure 2.1.

In addition to loading and unloading cargo, other services can be performed when a vessel is berthed. These include the draining of sludge, crew replacements, and lesser reparations of the vessel. When finished, the moorings are released from the vessel, allowing potential towing and piloting to commence in order to transport the vessel out of the port and on to its next voyage.

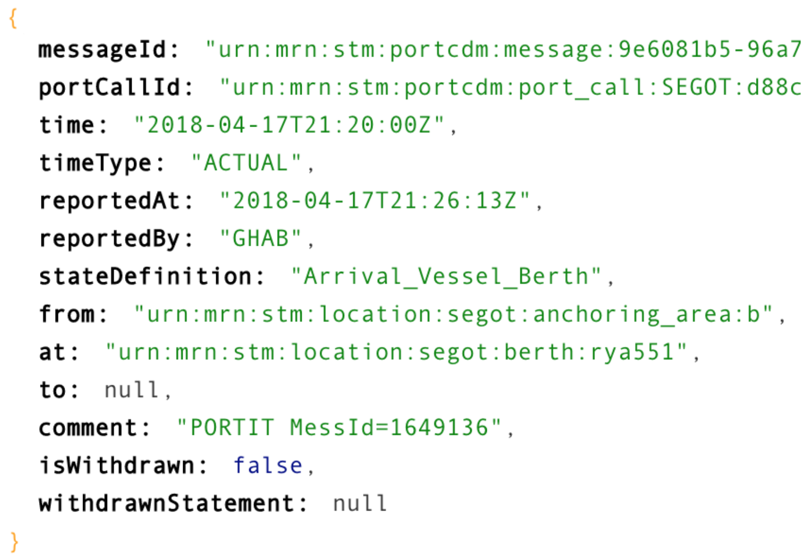
2.3 PortCDM

Recall from Section 1.3 that PortCDM is a standard and a system for sharing operational information – mainly time-stamps for various events – related to port calls.

The implementation of PortCDM has at the time of writing been tested over a period of three years [25]. It has collected time-stamp data and additional information about port calls from 13 different ports [25]. In the Port of Gothenburg alone, more than eleven thousand port calls have been registered since 2016. This data is accessible through the PortCDM API.

In the PortCDM database, each port call has a unique ID, a vessel ID, and a collection of events and port call messages attached to it. A port call message is the smallest component in PortCDM and is submitted to the system through the PortCDM API in JavaScript Object Notation (JSON) format, see Figure 2.2. It concerns an action and a time, and consists of the following main components:

- Time - the time for the action to take place
- Time type - e.g estimated, actual etc.
- State definition - a name of the action e.g "Arrival vessel berth" or "Departure Vessel Berth"
- From, at or to - possible locations of the action e.g. a specific berth



```
{
  messageId: "urn:mrn:stm:portcdm:message:9e6081b5-96a7
  portCallId: "urn:mrn:stm:portcdm:port_call:SEGOT:d88c
  time: "2018-04-17T21:20:00Z",
  timeType: "ACTUAL",
  reportedAt: "2018-04-17T21:26:13Z",
  reportedBy: "GHAB",
  stateDefinition: "Arrival_Vessel_Berth",
  from: "urn:mrn:stm:location:segot:anchoring_area:b",
  at: "urn:mrn:stm:location:segot:berth:rya551",
  to: null,
  comment: "PORTIT MessId=1649136",
  isWithdrawn: false,
  withdrawnStatement: null
}
```

Figure 2.2: A message in JSON format

Actions are parts of events, for example the “Arrival Vessel Berth” and “Departure Vessel Berth” actions are both part of a “Vessel at berth” event. In this case, the “arrival” action decides the start time of the event, while the “departure” action decides the end time. In the PortCDM database, all messages in a port call concerning the same event are grouped together and the most recent data from these messages is compiled to show the current state. An event is composed of the following main parts:

- Start time and start time type
- End time and end time type
- Definition ID - the name of the event, e.g “Vessel at berth”
- From, at and to - possible locations of the event
- A collection of messages, related to this event

The structure of port calls, events and messages is shown in Figure 2.3.

In addition, the database also contains a vessel registry and a location registry. The vessel registry holds information about all vessels registered with an IMO number, which is a unique vessel identification number that is used as a reference when creating port calls. The location registry holds information about all locations in the port, such as the corresponding coordinates and names.

2.4 PortableCDM and PACT

PortCDM itself is only a service that other IT systems can connect to. To demonstrate the full potential of PortCDM, two main applications for human interaction

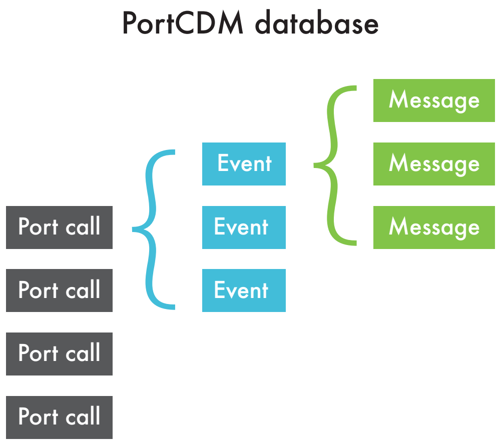


Figure 2.3: The structure of the PortCDM database

were developed. PACT is a web based desktop application that provides access to all of the features of PortCDM, such as sending all message types, viewing the current status of the port, or browsing the vessel registry.

PortableCDM is a mobile application available for the iOS and Android operating systems. It serves the same purpose as PACT, though it is a version that has been scaled down slightly. It is possible to respond to TTAs with RTAs directly from both PACT and PortableCDM.

3

Method

This chapter offers an overview of how the project progressed. The initial approach is reviewed and a refined approach is motivated and explained. The evaluation process that was employed as well as the methodology used for managing the project are also discussed.

The implementation of the system is described in Chapter 5, while Chapter 4 offers the theoretical framework needed to understand the implementation.

3.1 Initial System Architecture

The initial intention was to solve the two main objectives, automating the synchronization process and providing an optimal RTA, using a single Machine Learning (ML) model that would consider relevant variables in the port and derive an optimal RTA from them. Thus, the initial focus was to identify those variables while simultaneously investigating suitable machine learning models. It was believed that the data in PortCDM was quite noisy. Since artificial neural networks are well suited for problems with noisy training data [26] they were studied thoroughly. Other ML algorithms were studied as well in order to enable a comparison between algorithms.

The availability of three actors; pilots, tugboats and terminals (i.e berths), were identified as the most decisive variables for RTAs. It was concluded that if machine learning was to be used to perform the synchronization, the RTA provided would depend on the probabilities of these actors being available at certain times. It was decided that such an RTA would simply be too unreliable to be of use, and thus the project took a new approach.

3.2 Refined System Architecture

Instead of using ML for synchronization it was decided to create a system that models the port. The goal was that the system would keep track of the availability of actors using data from PortCDM. Since the availability of all actors would be known, the model would be able to provide an RTA at which all actors are available. Since the number of tugboats required in a port call is not available in PortCDM it was decided that ML algorithms for classification should be used to approximate the data.

Instead of incorporating the optimization as a part of the synchronization it was decided to make the optimization and synchronization two separate services. The purpose of this was to allow users to utilise the synchronization without the RTA being affected by an optimization metric. Furthermore, separating the services allowed for parallel development of the optimization and synchronization services.

Since both services required data from PortCDM, the first step was to develop a service that could provide that data, called the data module. Once the data module was completed the synchronization service was implemented. In order for the system to be modular we decided that the synchronization service should be composed of several classes implementing a common interface, called `Actor`. Since the availability of berths was deemed the most relevant factor in planning an RTA, it was the first `Actor` to be implemented. Two other `Actors`, for pilotage and towage, were developed after this, as they were considered to be central to the port call process and since the data needed to model them was available.

While developing the synchronization service, the delays of vessels were also identified as a relevant variable to consider. Delays can be reported via PortCDM, but there are no regulations specifying when that should be done. Thus, it would be beneficial to be able to predict delays that have not been reported yet. An ML algorithm for regression was therefore implemented, with the purpose of predicting delays from berth.

The optimization service was developed in parallel with the synchronization service. Several algorithms for optimization were investigated, out of which one was implemented. In particular, we investigated the CPU scheduling algorithms Shortest Process First and Priority Scheduling as well as the general scheduling algorithm Earliest End First. The optimization service implements the algorithm Earliest Deadline First. This was implemented first, since it was believed to have the best potential. The others were not implemented due to a lack of time. Due to the team's lack of prior knowledge on the topics of scheduling and optimization, only algorithms for optimizing one variable were considered.

During the development process, many different supportive functionalities had to be implemented in order to produce a functioning system. As stated earlier in this section, a connection to PortCDM was necessary in order to retrieve and work with the data. For this purpose there are several classes in the data module to help handle HTTP requests, parse and cache data. While handling requests, the system needed a way to keep track of the specific data for a request. As such, wrappers for the data were developed as well as helper classes to build schedules and utility classes to simplify common tasks. Similarly, for the ML parts of the system, supporting functionality was needed to gather, analyse, prepare, and import the training data. Two approaches were used when gathering data. In the first approach, data from PortCDM was used. To make this possible several classes were developed to fetch data, process the data, and build a result with features and keys for the ML parts to train on. For the second approach, the data was gathered from a database of positional data and a number of database views were created to analyse and retrieve the data needed. The data was then exported from the database and imported into the system were it was processed by many of the same classes used in the first approach.

3.3 Project Evaluation – Industry Survey

In order to evaluate the quality and relevance of the project, a survey was conducted near the end of the project. Ten respondents with relevant shipping industry experience were shown a demonstration of the synchronization and optimization services with made-up scenarios, and then asked to fill out a survey with questions regarding the services. An extensive summary of the responses can be found in Section 6.4. The survey was designed according to the methodology outlined in [27] and [28], and Likert scales [29] were used for measuring attitudes regarding certain aspects of the services. The survey and an English translation of it are available in Appendix B.

The respondents came from a number of diverse but relevant parts of the shipping industry. Some of the them worked at the port of Gothenburg, and some of them worked as lecturers at the department of Shipping and Marine Technology, with previous experience of working in the shipping industry. Figure 3.1 summarises the number of years of industry experience that they had. All respondents except one were previously aware of the PortCDM project.

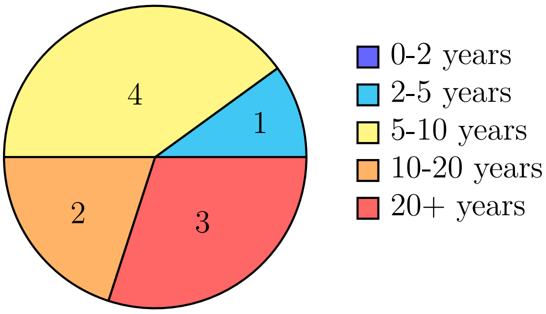


Figure 3.1: Industry experience of respondents

As we were physically present when the respondents filled out the surveys, we occasionally received further elaboration verbally. We have tried to include such additional information in our documentation of the responses.

3.4 Project Management

This project followed the Scrum methodology for software development, and for writing the documentation. It is a modern and well-known process where each week represents a “sprint”, as seen in Figure 3.2. Each sprint contains a number of tasks that are assigned estimated durations.

A “daily Scrum” was conducted via the Slack communication tool, where we answered three questions: “What have I done? What am I doing? What will I do?”, every day. At the end of every sprint, a sprint review and retrospective were held. In the sprint review, the members of the group discussed whether the tasks were under- or overestimated, as well as what had been done and what had not been done. The retrospective also consisted of three questions, namely “What did go well? What did not go well? What do we need to improve?”. The answers to these questions and an overview of the completed tasks were then added to the project diary.

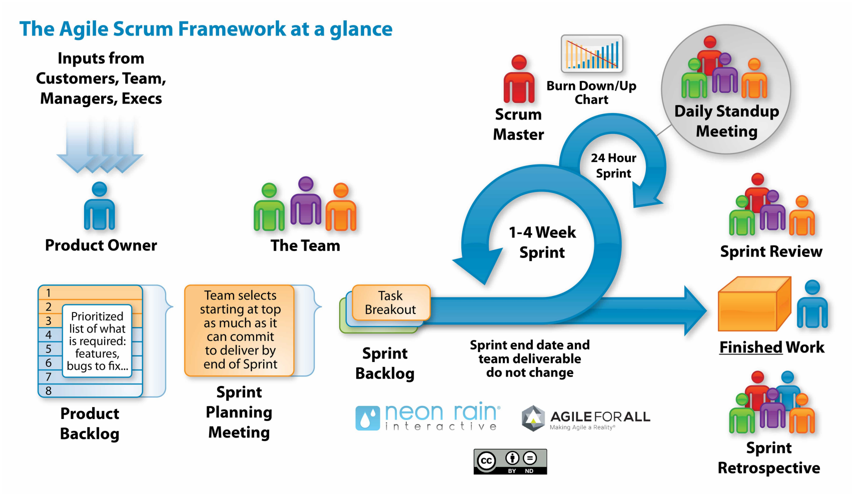


Figure 3.2: An overview of the Scrum methodology. [30]

Administrative work was handled by assigning each team member a role with specific corresponding responsibilities. These roles were comprised of a Scrum master, a project leader, a protocol manager, a diary and time log manager, a planning manager, and a report manager.

3.5 Tools

For the Scrum backlog, the industry standard tool Jira was used, which produces a burndown chart automatically and has other useful features. We used Git as our version control system with a repository on Stash. We used the IDE IntelliJ for Java, Visual Studio Code for JavaScript and Spyder for Python. Other tools that were used include; Toggl for time logging, Slack for communication within the team, Google Drive for sharing notes and documents, and ShareLaTeX for report writing.

4

Theoretical Framework

This chapter offers an explanation of the different methods and theories needed to understand the implementation of the system. Machine Learning concepts are explained, along with several algorithms and related methods. Algorithms for optimizing schedules are also explained, and the chapter ends with an explanation of what a RESTful API and the Automatic Identification System are.

4.1 Machine Learning

Machine Learning (ML) methods for classification and regression were used during the course of the project to approximate unavailable data. This section presents the regression and classification problems, after which the underlying theories for several machine learning algorithms that solve the problems are presented. A description of how the algorithms were implemented can be found in Section 5.5.

4.1.1 Regression

Regression is the process of approximating a real-valued function [26]. Linear regression is perhaps the simplest form of regression, since it is simply the linear combination of the input variables that approximates the given function as closely as possible [26]. Other algorithms accomplish this in other ways, but they are all a function of the input variables and their goal is also to approximate a given function as closely as possible [26].

4.1.2 Classification Problem

Within machine learning, the classification problem can be defined as follows: given a set of categories and an observation, identify which of the categories the observation belongs to. This is accomplished by utilising a training set consisting of observations where the categories are known.

An observation is defined as a set of quantifiable features. The value of each feature makes observations unique and distinguishes them from each other. Using this data, a machine learning algorithm can learn what quantities in each feature distinguishes a member of a certain category.

Classification Example

A simple example of the classification is the matter of classifying a fruit as either a lemon or an apple. The set of categories could be defined as $[Lemon, Apple]$. An example of the set of features that defines each observation could be the following: $[Width, Height, Yellow]$ where $Width$ and $Height$ are real numbers while $Yellow$ is a binary feature, i.e. it has the value 0 or 1. Given some training data a machine learning algorithm can learn what features distinguish a lemon from an apple. For example, the algorithm could conclude that if the conditions $Width > 5, Height < 6, Yellow = Yes$ are met by a new observation, it is most likely a lemon.

4.1.3 Key and Feature Selection

The variable that is to be predicted by a machine learning method is called the *key* and the variables used for making the prediction are called the *features*. For example, if the number of apples purchased by a customer is to be predicted, then the number of apples would be the key and the features could be the price of the apples, the customer's income, the customer's age and so forth. Selecting features can be done in several ways. One way is to consider the correlation between the features and the key, which can be measured by, for example, the Pearson correlation coefficient [31]. The Pearson correlation coefficient measures linear correlation between the features and the key [31].

4.1.4 Dimensionality Reduction Using PCA

Dimensionality reduction methods are used when a machine learning algorithm might be impeded by correlation between features or an excessive number of features [32]. Principal component analysis (PCA) is a method for dimensionality reduction [32]. PCA is performed by identifying a number of vectors, called principal components, that best express the variance of the data [33]. Once the principal components have been identified, all data is transformed such that it is expressed in terms of the principal components instead of their original basis [33]. The number of principal components can be chosen by the percentage of variance that is to be preserved [34].

4.1.5 Data Normalisation

In algorithms based on geometrical distance, features with a large range of values are implicitly assigned a higher weight than features with a small range. In order to make sure that all features are weighted equally, normalisation is performed before the algorithm is applied. One way to achieve this is to normalise all features into the range $[0, 1]$ [35]. In addition, normalisation can be performed to increase the training speed of an artificial neural network [36].

The following formula is used for normalising a single variable in a data set, with x being the original value, x_{norm} being the normalised value, and s being the full data set.

$$x_{norm} = \frac{x - mean(s)}{max(s) - min(s)} \tag{4.1}$$

The output given by an algorithm applied on normalised input and training data is also normalised. The output is *denormalised* by solving for the original value x from the same formula.

4.1.6 Categorical Data

Training data sometimes contains categorical variables, like colours. For the variable to be used as input to an ML algorithm based on linear techniques it needs to be one-hot encoded [37]. That is, the variable should be represented by several binary inputs, where each input corresponds to one category.

One-hot Encoding Example

Assume that a categorical variable represents three colours, blue green and red. These could be represented using two binary inputs, x_0 and x_1.

Colour	x_0	x_1
Blue	0	0
Red	0	1
Green	1	0

Unlike continuous variables, PCA is not valuable for categorical variables and should therefore be avoided [38].

4.1.7 Evaluating Algorithms with k-Folds Cross Validation

k-Folds Cross Validation measures the overall accuracy of a classification algorithm by splitting the training data D into k subsets and then training the algorithm k times, each time training it on $D \setminus D_t$ and testing it on D_t, $t \in [1..k]$ [39]. The cross-validation accuracy is the number of correct classifications divided by the number of instances [39].

4.1.8 Artificial Neural Networks

Artificial Neural Networks (ANNs) are algorithms (or hardware) that mimic the structure of neurons in the cerebral cortex with the purpose of approximating functions [40] and can be used both for regression and classification problems. ANNs consist of artificial neurons that are organised in layers interconnected with each other [40]. A network usually consists of an input layer, an output layer, and one or more hidden layers which process the input data [40]. This is visualised in Figure 4.1, where each neuron is a node and each connection is an arrow.

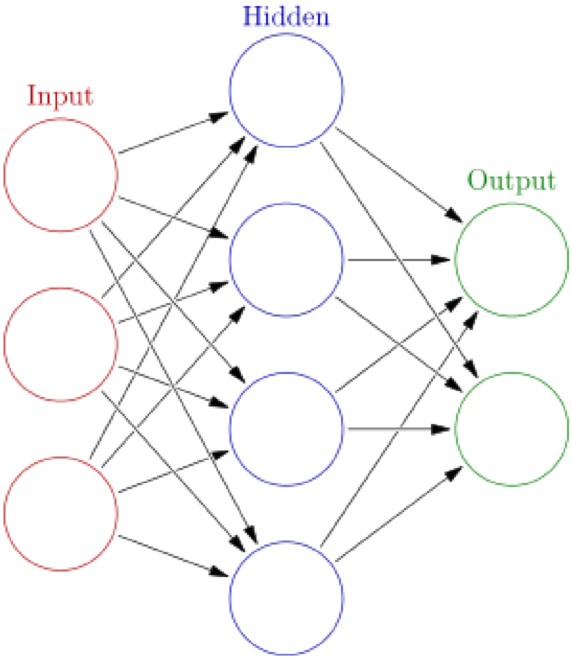


Figure 4.1: Visualisation of an ANN. From [41]. CC BY-SA 3.0.

4.1.8.1 Neurons in Artificial Neural Networks

Before the network's function as a whole can be understood, the function of a single neuron must be understood. The following explanation is based on [42]. A neuron has a number of binary inputs, $x_1, x_2, \ldots\ x_n$ and produces a binary output y. Each input is associated with a weight $w_1, w_2, \ldots\ w_n$, denoting the significance of that input. The output is produced by applying the neuron's activation function f on the weighted sum of the inputs, that is $y = f(\sum_{i=1}^{n} w_i \cdot x_i)$. There are several types of activation functions. A basic example is a type of neuron called perceptron, which activates once the input is greater than a threshold, usually called the bias and denoted by b. The function is defined as follows:

$$y = \begin{cases} 1, & \text{if } \sum_{i=1}^{n} w_i \cdot x_i > \text{b} \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

Neural Network Example

Assume that a perceptron that tells whether to bring an umbrella outside or not is wanted, and that it has the following inputs and weights:

- $x_1 = 1$ if it is raining, 0 otherwise, $w_1 = 6$
- $x_2 = 1$ if it is cloudy, 0 otherwise, $w_2 = 2$
- $x_3 = 1$ if the forecast is "no rain", 0 else, $w_3 = -4$

Lastly, assume that the threshold for the activation function is 0. By calculating the weighted sum of the inputs the perceptron can now "decide" if an umbrella should be brought or not. For example, if it is raining, not cloudy and the forecast said "no rain" the weighted sum is $6 * 1 + 0 * 2 - 4 * 1 = 2$, and an umbrella should thus be brought. Note that by changing the weights and the threshold, the behaviour of the perceptron can be changed.

4.1.8.2 The Layered Structure of an Artificial Neural Network

Conceptually, each neuron can be thought of as making one minor decision. A single neuron cannot make very complex decisions – the power of an ANN comes from combining multiple neurons [42]. The layered structure of ANNs allows the neurons to make their decision based on the decisions made in preceding layers, i.e. an ANN can make complex decision by combining several minor decisions [42].

The ANN processes data by passing the inputs from the input layer through the hidden layers [42]. The outputs of the neurons in the first hidden layer are computed and passed on to the next hidden layer and so forth until the output layer is reached, thereby combining several minor decisions into one [42]. It is worthwhile to note that every neuron in the input layer as well as in the hidden layers has a single output, but that this single output is an input to every other node in the succeeding layer [42], as seen in Figure 4.1.

4.1.8.3 Training an Artificial Neural Network

From the passages above it should be clear that, given the correct weights and biases, an ANN can be used to approximate a function of its inputs. It should also be clear that by changing weights and biases, the behaviour of the network changes. Thus, given a set of inputs X and their known outputs Y for the function f, X can be passed to the ANN and its outputs $\hat{Y}$ can be compared with Y, after which the weights and biases can be adjusted in order for the ANN to closer approximate f [42].

A cost function, C, is defined in order to measure how well the ANN manages to approximate f [42]. A common choice of C is to consider the cross-entropy between the training data and the output from the ANN [43]. Another option is to use the mean square error:

$$C(w, b) = \frac{1}{n} \sum_{i=0}^{n} ||f(\vec{x_i}) - \hat{f}(\vec{x_i})||^2 \tag{4.3}$$

Where w and b are the weights and biases, n the number of inputs, $\vec{x_i}$ an input consisting of a number of features (thus a vector), $f(x_i)$ the known output for x_i and $\hat{f}(x_i)$ the output produced by the ANN. It follows that the ANN closely approximates f when $C(w, b) \approx 0$ [42]. An ANN learns to approximate f by minimising C using the gradient descent and backpropagation algorithms [42].

4.1.8.4 Designing an Artificial Neural Network

When designing an ANN the number of layers in the network, the number of neurons in each layer and which activation functions to use [42] must be chosen. These choices are important, as they directly impact the ANN's ability to approximate f [42]. Furthermore, the type of problem also influences the design [42].

The choice of activation function in the output layer is highly dependent on the type of problem [43]. Common choices are the softmax function for classification problems, while a linear (a linear combination of the inputs) function is common for regression problems [43]. Activation functions for neurons in the hidden layer is an active field of research and there are few definitive guidelines. However, the rectified linear function is a common default choice [43].

There are few definitive guidelines regarding the number of hidden layers and the number of neurons in each such layer [43]. Some general conclusions can, however, be drawn. An ANN with a single hidden layer can represent any function, but that layer may need to be exponential in size, and the ANN may still fail to learn the function [43]. Using a network with several layers can reduce the number of nodes required in each layer [43]. Furthermore, deeper networks are suitable for functions which are thought to be divisible into several simpler functions [43]. Ultimately, though, the optimal number of layers and optimal number of neurons in each layer must be found via experimentation [43].

4.1.9 The k-Nearest Neighbours Algorithm

The k-Nearest Neighbours algorithm is a machine learning method for classification based on the geometrical distance between observations [26]. Given a new observation, the algorithm calculates the k observations in the training data with the shortest geometrical distance to the new observation, after which it is assigned the category held by the majority of the k nearest observations [26].

Geometrical distance can be calculated in multiple ways. For continuous variables, the Euclidean distance is commonly used [26]. Assuming that observations consist of n features, an observation can be defined as a vector with n real values. That is, observation i is defined as $x_i = (x_{i1}, x_{i2}, ...x_{in})$, where x_{ip}, $p \in [1..n]$ represents the value of feature p in observation i. The Euclidean distance between x_i and x_j is defined as $d_{ij} = \sqrt{\sum_{p=1}^{n}(x_{ip} - x_{jp})^2}$ [26].

When implementing the algorithm it has to be determined which k to use. It must be a positive integer – that is, $k \in N_+$ – since some neighbours need to be considered. However, different choices of k yield different results [26]. The optimal k can be found via a grid search, which tries different values of hyper-parameters and selects the one yielding the best cross-validation accuracy [44].

If the number of features are limited to two, the algorithm can be visualised in the two-dimensional plane, as in Figure 4.2. In the figure, there are two features – estimated salary and age – and two categories, zero and one. The red and green points represent the known observations that constitute the training data. The red and the green areas represent the category a new observation belonging to that area would be assigned. It is worthwhile to note that in this example the features have been normalised, which explains why all the values of the features are between negative and positive three.

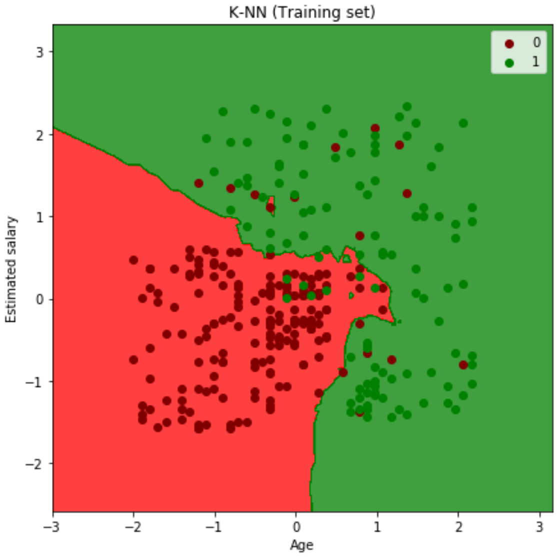


Figure 4.2: Visualisation of the k-Nearest Neighbours algorithm.

4.2 Algorithms for Optimal Scheduling

Literature on the subject of port call scheduling is scarce. Additionally, the literature which does exist on the subject is mainly concerned with scheduling specific types off vessels, rather than all of the vessels which are anticipated to arrive in a port. One reason for this may be that most ports employ the system of “first-come first-served” (FCFS) [45].

While domain-specific literature is scarce, general scheduling algorithms are partially applicable. Three main aspects have to be considered when optimizing port call schedules.

1. A variable to optimize must be selected. In the context of port call scheduling it could be the lateness of a vessel.

2. All vessels need to be served eventually. This requirement is called freedom from starvation [46].

3. No vessel must be forced to wait for an excessively long time. This is called the fairness requirement.

4.2.1 First-come, First-served

In ports that employ FCFS, the vessel that is first to arrive at the port is served first. In unpropitious scenarios, vessels will need to wait for long periods of time, and in the worst cases, vessels race against each other in order to get a slot [47]. Furthermore, the "losing" vessels may have to wait for unnecessarily long periods of time for their slots.

FCFS is quite naïve, and while it will perform well in a lot of scenarios, it also tends to lead to sub-optimal outcomes in many other scenarios [48]. FCFS does, however, fulfil the freedom from starvation and fairness requirements [48].

4.2.2 Earliest Deadline First

Earliest deadline first is an algorithm for scheduling the use of a general resource with the purpose of minimising lateness. Assuming that there is a set of n requests $[r_1, r_2, ..., r_n] = R$ for using the resource with deadlines d_i and processing time t_i the algorithm determines a start time s_i and finish time f_i for all requests. That is, $\forall r_i \in R$ the algorithm will determine s_i and f_i where $f_i = s_i + t_i$.

The algorithm will satisfy all requests and is thus free of starvation. They are, however, allowed to be late. The lateness of a request i is $l_i = f_i - d_i$, $l_i = 0$ if $f_i < d_i$. The goal is to minimise the maximum lateness. This approach means that it is better if everyone is a little bit late than someone being very late. Since all vessels are served relative to their deadline, the algorithm is fair.

Assuming that the resource is available from time s, the algorithm will take the request with the earliest deadline, r_1, and set $s_1 = s$, $f_1 = s_1 + t_1$, then $s_2 = f_1$ and so on. This greedy algorithm has been proven to be optimal in minimising the maximum lateness [49].

Figure 4.3 illustrates a scenario where the earliest deadline first algorithm is better than FCFS.

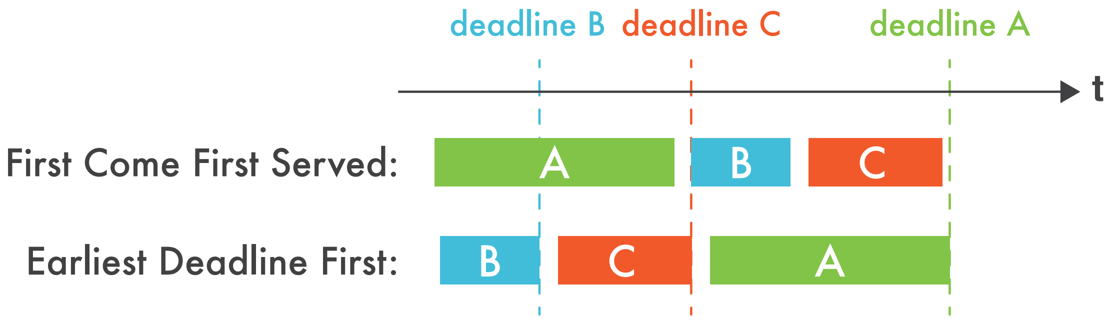


Figure 4.3: *First come, first served* versus *earliest deadline first.*

4.3 RESTful API

A RESTful (Representational State Transfer) API is a convenient way to provide an application with data through the Hyper Text Transfer Protocol (HTTP) [50]. To access the API, the user (in almost all cases another software application) performs an HTTP method request to a URL specified by the API, and the server produces a result [50]. This is one of the main advantages of using a REST API, since almost all IT systems today are capable of performing simple HTTP requests. The response from the API is in the JSON format. It is a widely used format, well suited for object-oriented applications [51].

4.4 Automatic Identification System

Automatic identification system (AIS) is a system that is placed on vessels at sea and constantly broadcasts its own position, together with a plethora of other data [52]. The main purpose of AIS is to help vessels at sea avoid collisions by sharing their own position with all other vessels close enough to receive the transmission [52].

The AIS data for 2017 from around the port of Gothenburg are available in a database that was made available to us. To reduce the amount of data, the positional points have been converted into segments, with a start point and end point containing position, time stamp, and other meta-data. This format decreases the amount of data, since several points on a fairly straight line are now represented by a single segment instead.

These segments can be used to make calculations and comparisons. It is, for example, possible to retrieve the number of tugboats needed for a given ship by comparing tugboat segments to vessel segments to see if they overlap in space and time.

5

Implementation

This chapter describes the implementation of the synchronization and optimization services. The implementation should be seen as a prototype, or a proof-of-concept for the synchronization and optimization services. It uses data from PortCDM to offer synchronization and optimization, and acts as an optional extension to the current PortCDM implementation. The implementation is made available via a RESTful API interface. A small web interface was implemented to present the result and test the application.

5.1 System Overview

In order to provide an RTA and potential optimizations, an application consisting of three principal systems, as seen in Figure 5.1, was developed. The brunt of the work is performed by the model, which fetches data from PortCDM and processes it as described below, before exposing the results through the API.

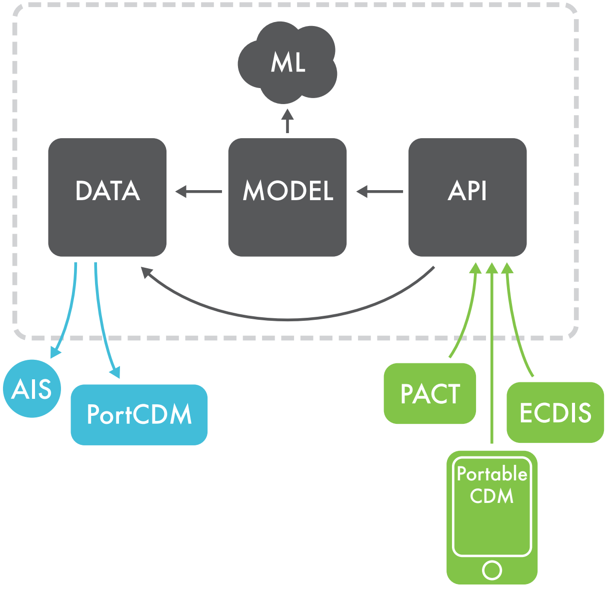


Figure 5.1: A high-level overview of the system. The grey arrows represent internal calls. The green arrows represent requests over the internet to the service. The blue arrows represent requests from the system to external services.

This API can then be accessed by external IT applications. Three such systems are shown in Figure 5.1. Recall from Section 2.4 that PACT and PortableCDM are two PortCDM interface applications. Electronic Chart Display and Information System (ECDIS) is a widely used digital system for vessel navigation [53], that can communicate with PortCDM.

The application was written using Java Enterprise Edition (Java EE), but the main model also depends on an external machine learning module, as discussed in Section 5.5, which was written in Python.

5.2 Model Implementation

The model is the core of the system and provides two services, Port Call Synchronization and Port Call Optimization, as shown in Figure 5.2. It has a connection to the API module, through which it receives and responds to requests. It is also connected to the data module which provides the model with useful data – both recent and historical – from PortCDM. Finally, there is a connection to a machine learning module, which is written in Python. It is used for making predictions when data is non-existent, sparse, or too unreliable.

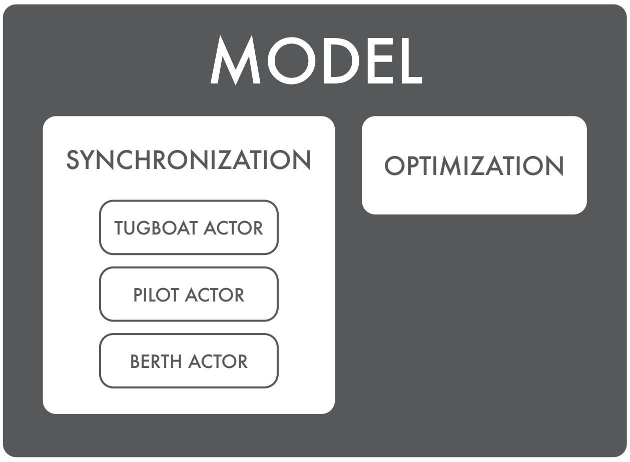


Figure 5.2: An overview of the model.

5.3 Port Call Synchronization

The `Port Call Synchronizer` is the part of the model that coordinates different `Actors` with the purpose of finding the earliest time that a vessel can enter the port – i.e, an RTA. The `Actors` are analogous to real actors and processes in the port, and both the data and the machine learning modules are used to model them. The `Port Call Synchronizer` receives requests for RTAs via the API.

The synchronization is performed according to the following algorithm:

1. The Synchronizer queries each Actor for the earliest time, following a desired time, at which it is able to accommodate a vessel. In the first iteration the desired time is the TTA given through the API.
2. Each Actor responds with a time.
3. If the resulting times are different, the latest of them is chosen as a new desired time.
4. Step 1 through 3 are repeated until the desired time and the responses from all of the Actors are the same. At this point the algorithm finishes, and the time agreed upon is used as RTA.

The algorithm is also illustrated in Figure 5.3. The algorithm finds the earliest possible time at which all Actors agree, as proven in Appendix A.

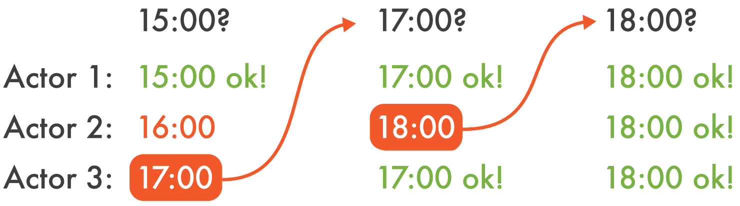


Figure 5.3: The synchronizer process.

As a simplification, the Port Call Synchronizer queries all the Actors for the same time. In reality however, the towage and pilotage would take place prior to the berth visit. Additionally, the Port Call Synchronizer always assumes that the port call associated with the RTA request will require pilotage.

5.4 Actor Implementation

Four Actors were implemented during the course of the project - Berth, Pilot, Tugboat and Berth Delay. The model was designed so that Actors are modular, and additional Actors implemented in the future could thus be easily integrated into the system. The specific Actors which were implemented during the course of this project were chosen based on real-world significance as well as the feasibility of implementing them. The assumption that the availability of other Actors will follow was made. For example, we assume that moorers are available when the berth is available.

The Port Call Synchronizer provides a start and end time when making a request to one of the Actors. The Actor then constructs one or several schedules using time stamp data of planned events relevant to the Actor fetched from the PortCDM database. By default, only port calls from the last two days and forward are considered, but the time limit can also be set via the configuration interface, see

Section 5.10. Earlier port calls are considered since their operations may interfere with future port calls. A duration is derived from the start and end time after which the `Actor` searches the schedules in order to find the first gap following the start time where the duration would fit, possibly postponing it.

5.4.1 Berth Actor

The purpose of the `Berth Actor` is to determine the next time at which a specified berth is available. In addition to a start and end time, the `Berth Actor` requires the name of the berth for which to construct a schedule. The resulting time of this `Actor` is thus the time when the specified berth is available.

In order to build a schedule for a specific berth, the `Berth Actor` iterates over the events of every relevant port call (recall the structure of port calls from Section 2.3). If an event that corresponds to the berth is found, it is added to the schedule. If the end time is missing, the average berth time is used to derive an estimated end time. After the schedule has been built, the new port call is scheduled for the first available slot after the given start time that is sufficient for the vessel. The duration of the time slot needs to be at least the duration of the visit plus a marginal time that represents the turnaround time required in the real world.

5.4.2 Pilot Actor

The purpose of the `Pilot Actor` is to determine the next time at which any pilot is available. There could be several pilots working in the port simultaneously, but each port call requires at most one pilot at the same time. Therefore, the `Pilot Actor` creates a number of schedules, corresponding to the number of pilots in the port, containing the times at which they are busy. The earliest possible start time is derived from each schedule, and the earliest of these is then returned as an answer to the `Port Call Synchronizer`. The user can set the number of pilots via the web interface, see Section 5.10.

Schedules are constructed by iterating over the events of all port calls and discarding those that are not events concerning pilotage. The remaining events are used to create the schedules.

5.4.3 Tugboat Actor

The purpose of the `Tugboat Actor` is to determine when enough tugboats are available for a specific port call. The number of tugboats needed for a port call is not specified in PortCDM. Instead, ML is used to predict the number of tugboats needed. Section 5.5.2 describes the implementation of the ML algorithm. After the number of tugboats have been predicted, the `Actor` proceeds to calculate a time at which enough tugboats are available. Similar to the `Pilot Actor`, this `Actor` creates a number of schedules, corresponding to the number of tugboats in the port. The `Tugboat Actor` then searches these schedules for the earliest possible time at which enough tugboats are available and returns it as a response to the `Port Call Synchronizer`.

The `Tugboat Actor` creates schedules in the same manner as the `Pilot Actor`, with the only difference being that the events that are of concerns are those that regard towage operations rather than pilotage events.

5.4.4 Berth Delay Actor

The purpose of the `Berth Delay Actor` is to check a specified berth for potential delays. That is, it will check if the berth is occupied by a vessel and – if so – if that vessel will be delayed. Like the other actors, the `Berth Delay Actor` will return the first time, after the desired time, at which it is “available”. In the case of the `Berth Delay Actor`, the first time at which the berth is available when taking delays into account is returned, and not only the estimated time of departure as in the case of the `Berth Actor`.

The implementation of the `Berth Delay Actor` utilises machine learning in order to predict the delay. The implementation of the ML algorithm is described in Section 5.5.1. The implementation of the `Berth Delay Actor` works as follows.

1. Given a port call, check if the relevant berth is occupied at the requested time.
2. If it is not, return the requested time as the first available time.
3. If it is, predict the delay using ML, add it to the estimated time of departure of the berthed vessel, and return it.

5.4.5 Assumptions by Actors

Information regarding which pilots and which tugboats a port call uses is not available to us. Hence, there is no way to ensure that the schedules produced by the `Tugboat Actor` and `Pilot Actor` are consistent with reality. To cope with this, they both assume the following:

- As few pilots and tugboats as possible are used. For example, if there are four pilots available in the port but all requests can be satisfied using only one pilot, only one pilot will be used.
- Any tugboat and any pilot can be used for any vessel.
- All berths, tugboats, and pilots require the same marginal time between operations. The marginal time can be set by the user.

5.5 Machine Learning Implementation

ML is used in the `Tugboat Actor` to predict the number of tugboats needed and in `Berth Delay Actor` to predict the delay at the berth. Both of these ML implementations utilise training data from the `Training Data Provider`. The implementation of `Training Data Provider`, including data pre-processing, is described in Section 5.6.

5.5.1 Predicting Delay at a Berth

The delay of a vessel could theoretically be any real value, and regression was therefore used to approximate it. An ANN was implemented to make the approximation, using the features in Table 5.1. The key used when training the ANN was the delay of a vessel measured in minutes. The delay is defined as the difference between the estimated and actual time of departure from berth.

Features were chosen based on their correlation with the key and based on the accuracy of the resulting ANN. Correlation was measured in two ways – by plotting the feature versus the key, and by calculating the Pearson correlation coefficients. The day and month of arrival were considered as features but were discarded since, as their plots in Figure 5.4 and 5.5 show, the correlation between them and the key was weak. Furthermore, the accuracy decreased when they were included. Although several of the features used had a low correlation coefficient, the plots – as well as the results being less accurate when they were excluded – indicated that they should be used.

Feature	**Pearson correlation coefficient**	**Used/Discarded**
Vessel length	0.0056	Used
Vessel beam	-0.0175	Used
Vessel type	Not a continuous variable	Used
Delay to berth	-0.2153	Used
Delay of departure from traffic area	0.01432	Used
Duration spent at the berth	0.1643	Used
Hour of arrival	0.0064	Used
Berth the vessel is at	Not a continuous variable	Used
Number of ETAs sent	0.1333	Used
Day of arrival	-0.0158	Discarded
Month of arrival	-0.0293	Discarded

Table 5.1: Features for the ANN predicting delays at berth.

The ANN was implemented in Python using the Keras library [54] and the class `Sequential`, which is a linear stack of layers. The class `Dense` was used for creating layers. Table 5.2 shows a specification of how the ANN is set up. By setting the *init* parameter to “glorout_uniform” when instantiating objects of `Dense`, all weights

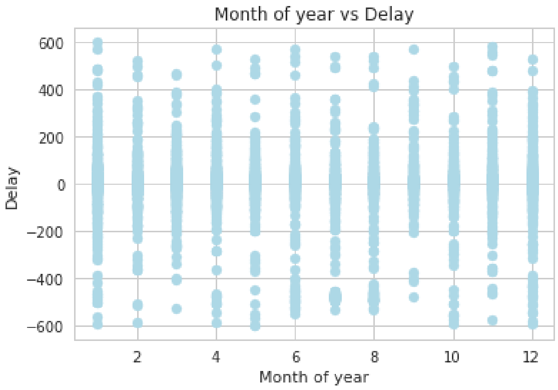


Figure 5.4: Delay of vessels plotted against the month at which they arrived.

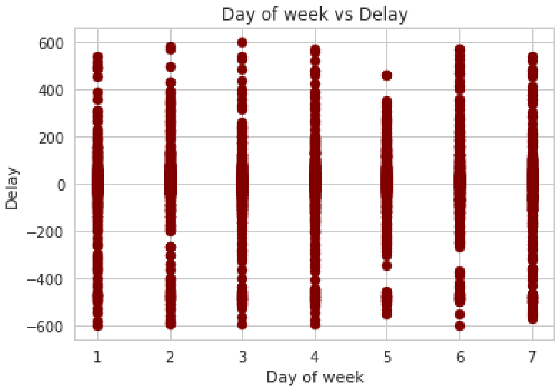


Figure 5.5: Delay of vessels plotted against the day at which they arrived.

were initialised to small none-zero values prior to training. The network was trained using Stochastic Gradient Descent (SGD) with the mean absolute percentage error as loss function, and with the mean square error, mean absolute error, and mean absolute percentage error as performance metrics.

Layer	**Nodes**	**Activation function**
Input	27	-
Hidden layer 1	8	Rectified linear function
Hidden layer 2	8	Rectified linear function
Output layer	1	-

Table 5.2: Setup for the ANN predicting delays at berth.

The number of layers was found by trial and error. The number of nodes in each hidden layer was initially set to the average of the number of nodes in the input layer and output layer. It was then modified to the numbers in Table 5.2 as they yielded the best results.

Since the `Berth Delay Actor` was implemented in Java, the ANN was imported to Java using the DL4J library. The library allows for an hd5 file, specifying the configuration of an ANN, to be imported and represented by an instance of the `Multi Layer Network` class. The configuration of the trained ANN was saved to such a file using `Keras` and imported as a `Multi Layer Network`, which the `Berth Delay Actor` can use to make predictions by simply calling a method.

5.5.2 Predicting the Number of Tugboats in a Port Call

Since the number of tugboats required in a port call is an integer in the range 0 to 5, predicting the number of tugboats is a classification problem. That is, predicting the number of tugboats required is actually the matter of identifying which out of 6 distinct classes a port call belongs to. The training data contains only one record of

5 tugboats being used. Since this is simply too few records to train an ML algorithm on, the record was discarded as an outlier.

In order to achieve as high accuracy as possible we implemented two ML algorithms for classification, an ANN and the k-Nearest Neighbours (kNN) algorithm, and used the one that performed the best. Naturally, in both cases the key is the number of tugs utilised in a port call. The features used by both algorithms are the following:

- Type of vessel
- Length of vessel
- Width of vessel

5.5.2.1 Artificial Neural Network Implementation

The ANN for predicting the number of tugboats required for a port call was implemented in almost exactly the same way as the ANN for predicting the delays at berth, described in Section 5.5.1 above. Table 5.3 shows a specification of how the ANN is set up.

Layer	**Nodes**	**Activation function**
Input	24	-
Hidden layer 1	12	Rectified linear function
Hidden layer 2	12	Rectified linear function
Hidden layer 3	12	Rectified linear function
Output layer	5	Sigmoid function

Table 5.3: Setup for the ANN predicting the number of tugboats needed.

5.5.2.2 k-Nearest Neighbours Implementation

The k-Nearest Neighbours (kNN) algorithm was implemented in Python using the `KNeighborsClassifier` class from the `scikit-learn` library [55]. Three parameters are needed when instantiating the class. These were set according to Table 5.4.

The optimal value of k was found via a grid search. The search was performed using the `GridSearchCV` class from `scikit-learn`. The class requires the number of sets in k-fold cross validation as well as the parameters to search to be specified. The number of subsets was set to 10 and the parameters to search was set to $k \in [1..20]$. The results showed that $k = 10$ was optimal.

5.5.2.3 Model Selection

The ANN and kNN models were evaluated using k-folds cross validation, as described in 4.1.7. The accuracy of the kNN implementation was ∼82.8%, and the

Parameter	Value	Description
n_neighbours	10	The number of neighbours to consider, k.
metric	'Minkowski'	Distance metric to use, 'Minkowski' with p=2 yields Euclidean distance.
p	2	Specifies the power of the Minkowski metric, p=2 yields Euclidean distance.

Table 5.4: Parameters passed to the `KNeighborsClassifier`.

accuracy of the ANN implementation was also ∼82.8%. Since the accuracies of both models were similar it was decided to use the model which was easiest to import to Java. The ANN was thus chosen as the final model, as it could be imported using the procedure described in Section 5.5.1.

5.6 Machine Learning Training Data

The training data used in the project consists of data from PortCDM and historical AIS data. A `Training Data Provider`, which acts as a standalone application, was implemented to provide such data to the machine learning module. The `Training Data Provider` fetches data, encodes categorical features, performs normalisation and PCA, and writes the data to a file.

5.6.1 Training Data

Both ML algorithms used by the `Tugboat Actor` and `Berth Delay Actor` naturally required some training data. For `Berth Delay Actor`, the data provided by PortCDM was adequate for the ML algorithm used.

The first approach that was chosen in order to acquire training data for the ML algorithm used by `Tugboat Actor` was to use data straight from PortCDM. This data contained variables which were thought to have a strong correlation to the number of tugboats. These variables included the vessel type, berth, and length among other features. However, we ran into issues with the most important piece of information – the actual number of tugboats that was required for a given port call. In order to acquire this data, the overlapping tugboat events within a port call were counted. Although several tugboat events were found within a lot of port calls, the majority of these events were invalid and none of them overlapped with each other. This rendered the training data useless, as it could only indicate whether or not tugboats were used, but not how many.

Therefore, another source of data had to be used. AIS data was the only other option and turned out to be much more reliable. It was not used immediately due to the fact that it was not easily available and time consuming to acquire. The AIS data used contained all port calls made in the port of Gothenburg during 2017. The tugboat key was obtained by counting the number of segments reported by tugboats overlapping the geographical segments of each vessel. The segments had to overlap in time, and be at least within 100 meters from each other to be counted. Even though more accurate results were obtained using the AIS data, fewer features were available within the AIS data set. The vessel type, length and width were the only features available.

5.6.2 Normalising and Denormalising Training Data

To improve the performance and results of the ANN the data was normalised. The normalisation was performed using the `StandardNormaliser` class from the `ND4J` Java library. Once a `StandardNormaliser` has been fit to the data set the same object must be used for new records to get any valuable results. The output from the ANN was also normalised and hence needed to be *denormalised* in order to be of any use. In practice, this means that the normaliser that is used must be saved along with the trained ANN. If the wrong normaliser is used with the wrong ANN, the results will be useless.

5.6.3 Applying PCA on Training Data

Since both data sets included several features, a principal component analysis (PCA) was performed on both of them. This is done to reduce dimensionality in the data as described in 4.1.4. The transformation is performed right before starting the training and after normalisation, as seen in Figure 5.6. Since PCA is not meaningful for categorical variables, they were excluded from the analysis.

Applying PCA on the data set for `Berth Delay Actor` yielded a less accurate network, even though 99% of the variance was kept. Since the accuracy declined, PCA was not applied to the training data.

In the data set for `Tugboat Actor` we decided that 99% of the variance should be kept, which resulted in two principal components, which was also the number of continuous features.

PCA was performed using the `ND4J`Java library. The objects that were used for transforming the data sets were saved and are used continuously by the `Tugboat`

Actor to fit new records, derived from incoming requests, to the correct dimensions before passing them to the corresponding ML algorithms.

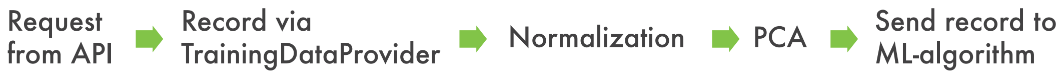


Figure 5.6: When an RTA is requested, the Actors that use ML use the Data module to fetch and normalise the data, and also to perform the PCA. The transformed data is then fed to the trained ANN and the result is denormalised before being used for calculations.

5.6.3.1 Categorical Variables in the Training Data

The training data consists of two categorical variables – berth and vessel type. These are one-hot encoded by the training data provider according to Section 4.1.6, as illustrated in Figure 5.7.

	CARGO	CONTAINER	...	RORO
CARGO ⟶	1	0	...	0
CONTAINER ⟶	0	1	...	0
⋮				
RORO ⟶	0	0	...	1

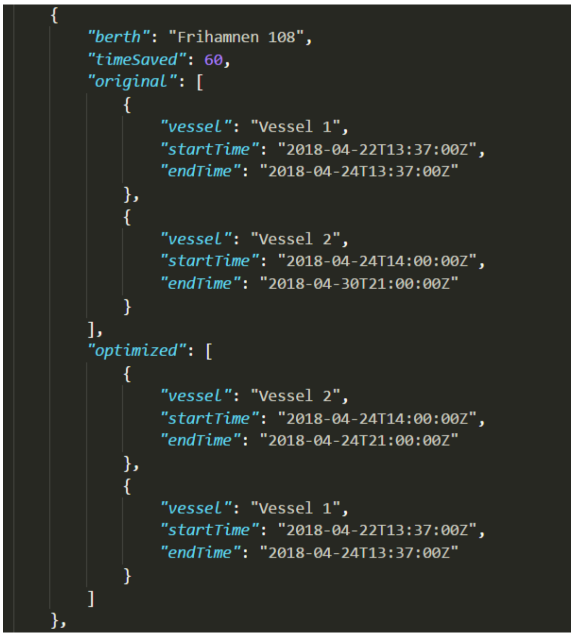
Figure 5.7: The vessel type can only be one of 14 different vessel types. This can be represented as a single one in an array of zeros.

5.7 Port Call Optimizer

The purpose of the Port Call Optimizer is to expose the optimal scheduling of a set of port calls. We have implemented this service for the scheduling of berths. Given a set of port calls, the Port Call Optimizer optimizes for the reduction of maximum lateness.

When optimizing for maximum lateness, the Port Call Optimizer minimises the maximum lateness using the Earliest Deadline First algorithm described in Section 4.2.2. At the time of writing, port calls do not specify a deadline. The implementation of the algorithm uses the estimated time of departure from berth as a deadline, where the delay is defined as the difference between the actual and estimated time of departure. The estimated time of departure was used since no deadline is specified in a port call message. There are however plans on implementing a field specifying the deadline in PortCDM, and once that is done the algorithm can be easily modified. The estimated time of departure is used temporarily since it is believed to reflect the desired time of departure. When the `Optimizer` is queried to optimize for maximum lateness it does the following:

1. Two maps, `Original` and `Optimal`, which map a berth to a list of their incoming port calls, are initialised.
2. Two `OptPortCall` objects, containing only the information relevant to the optimization, are created for each port call given.
3. All `OptPortCalls` are inserted into the list corresponding to the berth it is headed for. One copy is inserted into `Original` and one into `Optimal`.
4. All lists in `Optimal` are sorted by deadline. Simultaneously, all start and end times in the `OptPortCalls` are updated.
5. An `OptimizationResult` is created for each berth, containing the original order, the optimal order, and the time saved, as seen in Figure 5.8. A list of all `OptimizationResults` is returned to the user.



```
{
    "berth": "Frihamnen 108",
    "timeSaved": 60,
    "original": [
        {
            "vessel": "Vessel 1",
            "startTime": "2018-04-22T13:37:00Z",
            "endTime": "2018-04-24T13:37:00Z"
        },
        {
            "vessel": "Vessel 2",
            "startTime": "2018-04-24T14:00:00Z",
            "endTime": "2018-04-30T21:00:00Z"
        }
    ],
    "optimized": [
        {
            "vessel": "Vessel 2",
            "startTime": "2018-04-24T14:00:00Z",
            "endTime": "2018-04-24T21:00:00Z"
        },
        {
            "vessel": "Vessel 1",
            "startTime": "2018-04-22T13:37:00Z",
            "endTime": "2018-04-24T13:37:00Z"
        }
    ]
},
```

Figure 5.8: An `OptimizationResult` with two arrays of optimized and original `OptPortCall` objects in the JSON format.

5.8 The Data Module

When a request is sent to the API, a port call ID and message ID are the only parameters required. Additional information is, however, needed in order for the model to perform the synchronization described above. The data module provides this information.

The data module retrieves the information from PortCDM. PortCDM can provide both historical and real-time data. Historical data is used for analysis such as finding the average anchorage time or to provide training data for the machine learning algorithms. The real-time data is necessary when constructing `Actor` schedules.

5.8.1 The Data Analyser

The main purpose of the data module is to simply provide data, but it can also perform some basic analysis on it. For instance, the `Berth Actor` requires the average time at berth for its calculations. The `Data Analyser` enables several ways of analysing the data.

5.8.2 PortCDM HTTP Handler

In order for data to be fetched from the PortCDM database, a `PortCDM HTTP Handler` was created. The OkHttp library is used to create and send HTTP requests to the PortCDM API. Conceptually, the `PortCDM HTTP Handler` wraps the PortCDM API, thus allowing data from the PortCDM database to be fetched with ease.

When real-time data is requested, it simply downloads the required data from PortCDM. Historical data, however, is retrieved from a cache.

5.8.3 Data Cache

Since the application is accessed through API requests, it is inconvenient if it has to download all historical data for every request. Still, the historical data needs to be updated once in a while to take recent information into consideration. This is implemented in a `Cache Handler`, that for each request either downloads all required information or provides the cached information.

5.8.4 PortCDM Parser

The `PortCDM parser` is a simple tool for parsing the JSON data provided by the `PortCDM HTTP Handler` or the `Data Cache` into Java objects. This is accomplished by using the Gson library, which utilises predefined templates in order to parse the JSON objects.

5.9 The API Module

The API (Application Programming Interface) is built similarly to the other PortCDM modules. It packages the complete application into a WAR-file (Web Application Runnable) which can be run on any Java EE application server. In this case, it will run on a Wildfly server, together with the other PortCDM modules, making it an extension to PortCDM. Since it extends PortCDM, it is also convenient to have PortCDM on the same server as the application, since all requests to the PortCDM API can be done locally, without using the internet. The application is based on a RESTful API created with the RestEasy Java library. The API serves mainly two purposes; providing access to the synchronization and optimization algorithms as well as to the system configuration. To demonstrate what the REST API can be used for and to show why it is a valuable extension to PortCDM, a web interface that is connected to the API was created.

The services developed in this project were made available through a RESTful API. The motivation for and a general overview of RESTful APIs is given in Section 4.3.

To specify the API, the markup language YAML was used. This was then interpreted by the service Swagger, which generated most of the implementation. Swagger leaves a gap in the implementation for making calls to the internal application structure, i.e. the `Port Call Synchronizer` and other internal services. The response is the result from the `Port Call Synchronizer` along with a complete trace of the agreement process, to visualise the iterations required to achieve consensus in the web interface.

The REST API is supposed to be used as an extension by other PortCDM applications, such as the mobile application PortableCDM or the web based PACT system, see Section 2.4. It could also be integrated with any other IT system that is capable of making HTTP requests.

5.10 Port Call Synchronization Web Interface

The web interface was designed to allow for visual demonstrations, configuration and full-stack testing by developers. It also serves as an example front-end application that uses the synchronization service. The web interface is not supposed to be the main application for the synchronization service, since that is outside the scope of this project, see Section 1.6. It was developed with the JavaScript front-end library ReactJS and is served on the same URL as the synchronizer API.

6

Results

This chapter offers an overview of the resulting product. The web interface is presented, after which the specific results of the `Synchronizer` and its ML algorithms are reviewed. The `Optimizer` is presented briefly, and the chapter ends with a compilation of the survey results that were gathered in order to evaluate the product.

The actual implementation of the synchronization and optimization services should be considered to be the chief result of the project.

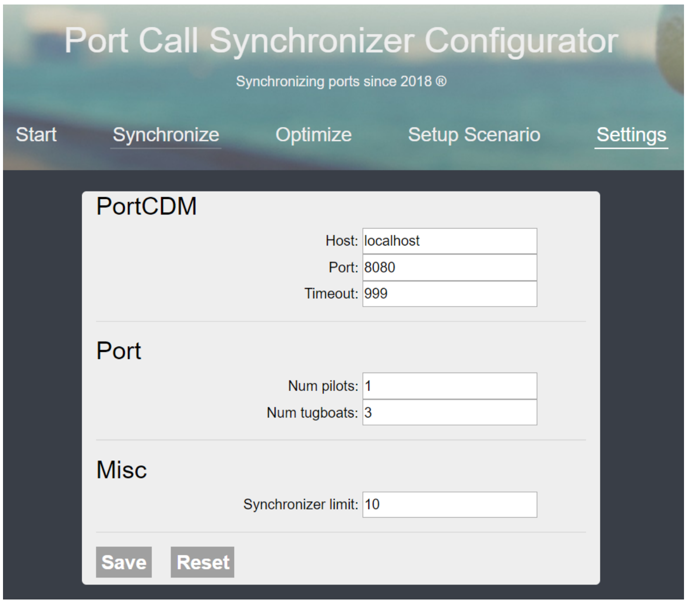
6.1 Web Interface

The web interface contains four pages; *Synchronize*, *Optimize*, *Setup Scenario* and *Settings*. The *Settings* page is illustrated in Figure 6.1. From this page the user can configure the following settings of the application:

- PortCDM server-related settings, including host, port and and timeout for the server.
- Port-related settings, including the number of pilots and tugboats operating in the port.
- A limit for how many iterations the `Synchronizer` will run when trying to find an RTA.

In the *Setup Scenario* page, the user can setup example port calls on a PortCDM server in order to test the application. There are nine different vessels to choose from when setting up port calls. Each port call contains planned towage, pilotage, and berth visit. The user can choose to create the port calls one by one by clicking the buttons *Setup vessel 1*, *Setup vessel 2* and so on, or all nine at once by clicking the *Build scenario* button. The *Setup Scenario* page is presented in Figure 6.2.

From the *Synchronize* page, the user an request an RTA by choosing a port call in a drop down menu and clicking the *Request RTA* button. There is a small log view and a progress bar, giving the user information on the progress of the request. When an RTA has been generated, the process of finding consensus is visualised on the same page. It shows each iteration of the `Synchronizer` and each `Actor's` first available time. It points out all `Actors` that cannot serve the vessel on the requested time and marks the latest `Actor` with a thick red border. It is possible to


Port Call Synchronizer Configurator
Synchronizing ports since 2018 ®
Start
Synchronize
Optimize
Setup Scenario
Settings
Save
Reset

PortCDM

Host: localhost
Port: 8080
Timeout: 999

Port

Num pilots: 1
Num tugboats: 3

Misc

Synchronizer limit: 10

Figure 6.1: The *Settings* page.

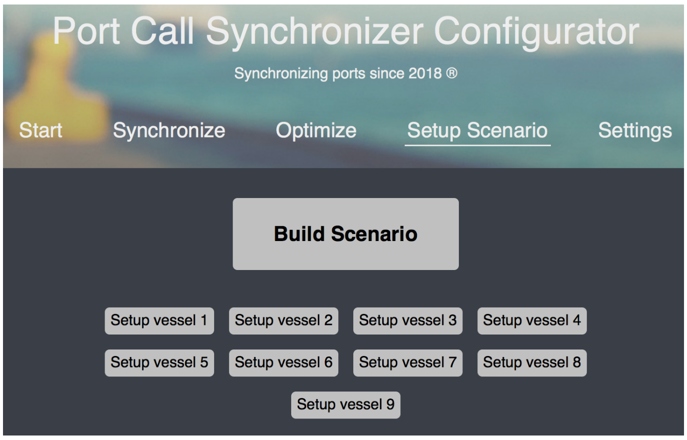
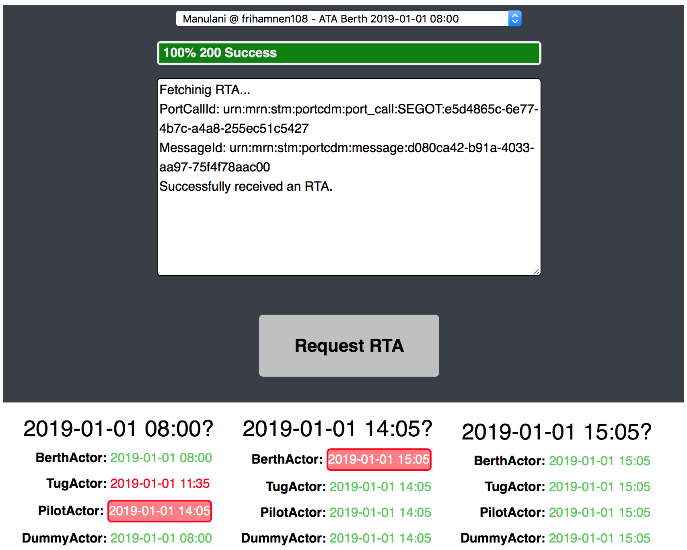


Figure 6.2: The *Setup scenario* page.

click on each `Actor` to see its reliability and its description. The *Synchronize* page is visualised in Figure 6.3.

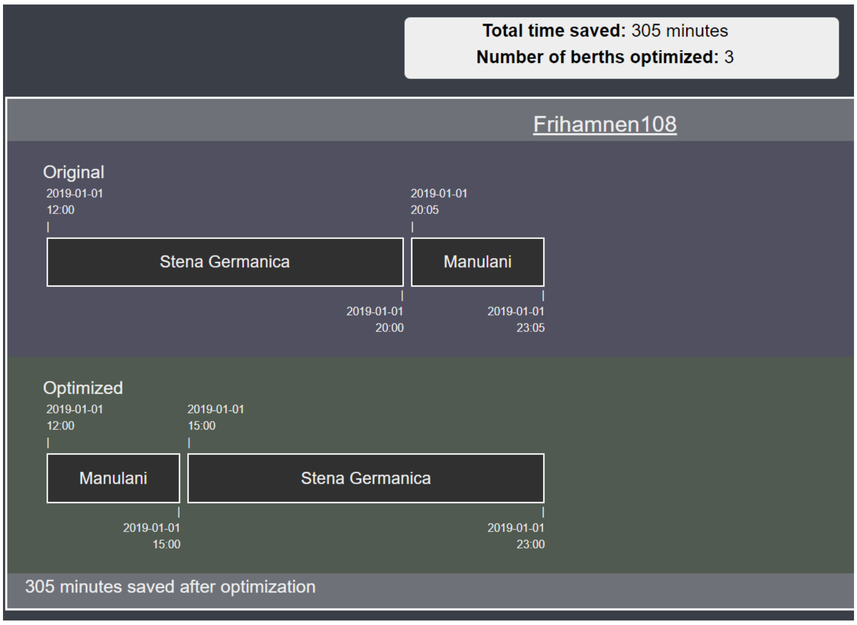
In the *Optimize* page, the user can request an optimization of the port calls in the port. Just like in the *Synchronize* page, there is a log view and a progress bar, giving the user information regarding the progress of the request. When the request is done, a visualisation of the optimization is presented, see Figure 6.4. For each berth, a visualisation shows the original schedule, and an optimized one along with the number of minutes saved. The name of the vessel along with the original and optimized time of arrival and departure for each port call is also displayed.



```
Fetchinig RTA...
PortCallId: urn:mrn:stm:portcdm:port_call:SEGOT:e5d4865c-6e77-
4b7c-a4a8-255ec51c5427
MessageId: urn:mrn:stm:portcdm:message:d080ca42-b91a-4033-
aa97-75f4f78aac00
Successfully received an RTA.
```

2019-01-01 08:00?	2019-01-01 14:05?	2019-01-01 15:05?
BerthActor: 2019-01-01 08:00	**BerthActor:** 2019-01-01 15:05	**BerthActor:** 2019-01-01 15:05
TugActor: 2019-01-01 11:35	**TugActor:** 2019-01-01 14:05	**TugActor:** 2019-01-01 15:05
PilotActor: 2019-01-01 14:05	**PilotActor:** 2019-01-01 14:05	**PilotActor:** 2019-01-01 15:05
DummyActor: 2019-01-01 08:00	**DummyActor:** 2019-01-01 14:05	**DummyActor:** 2019-01-01 15:05

Figure 6.3: The *Synchronize* page.



Total time saved: 305 minutes
Number of berths optimized: 3

305 minutes saved after optimization

Figure 6.4: The *Optimization* page.

6.2 Port Call Synchronizer results

The Port Call Synchronizer works as described in Section 5.3. The result of the ML utilised by the Port Call Synchronizer through the Berth Delay Actor and the Tugboat Actor is described below.

6.2.1 Berth Delay Actor

The ANN used by the Berth Delay Actor is very inaccurate. The mean absolute percentage error of the final network is ∼ 90%. Since the predictions are useless in practice, the Berth Delay Actor was excluded from the final version of the system. Furthermore, plotting the predicted values versus an ideal prediction reveals that the network's approximation is always roughly the same, as shown in Figure 6.5. Why this is the case is discussed in Section 7.3.2.

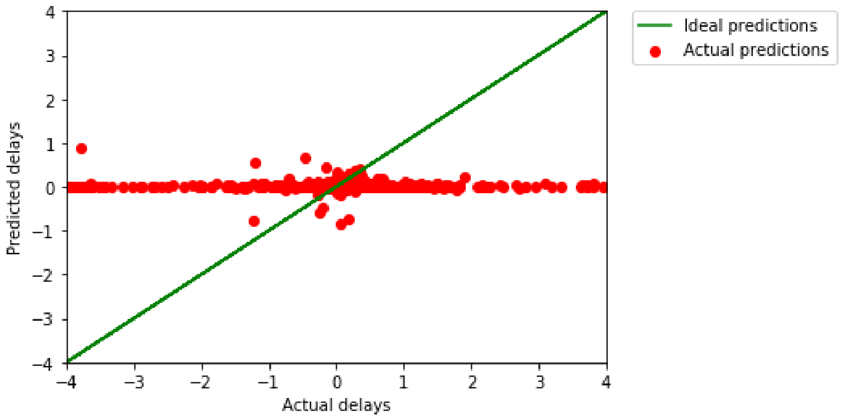


Figure 6.5: The predictions of Berth Delay Actor plotted with an ideal prediction.

6.2.2 Tugboat Actor

The cross validation accuracy of the ANN used by Tugboat Actor is ∼ 82.8%. The confusion matrix, which shows the predicted and actual classes of records in a test set, offers further insight in the predictions of the ANN. The confusion matrix in Table 6.1 was produced when training the ANN on a randomly selected set of records, constituting 80% of the training data, and testing it using the remaining records. The rows in Table 6.1 correspond to the actual number of tugboats used and the columns correspond to the predictions of the ANN. That is, the number in row i, column j, corresponds to the number of records that used i tugboats and were predicted to use j tugboats. Table 6.2 offers further interpretation of the confusion matrix. Note that the accuracy is different from the cross validation accuracy, as the ANN was trained on one, randomly selected, set of records. If the same procedure would be repeated another outcome would of course be possible, which is why the confusion matrix should be interpreted as a description of the predictions rather than an exact measurement of accuracy. Section 7.3 offers a discussion of these results.

Number of tugboats	0	1	2	3	4
0	3540	0	6	0	1
1	541	1	4	2	1
2	114	0	14	4	7
3	18	0	3	2	14
4	4	0	0	3	14

Table 6.1: A confusion matrix produced when training the ANN

Total number of records:	4293
Number of correct predictions:	3571
Number of errors:	722
Accuracy:	∼ 83.2%

Table 6.2: Table 6.1 in terms of total errors and correct predictions.

6.3 Port Call Optimizer Results

The `Port Call Optimizer` works as intended and will reduce the overall delays as much as possible whenever the current schedule can be improved. The improvements that can be obtained can save a substantial amount of time. An example of this can be seen in Figure 6.4.

6.4 Product Evaluation Results

In this Section the results of the product evaluation are presented. The respondents' quantifiable answers have been summarised using pie charts, and their answers to open-ended questions have been transcribed. Some respondents chose to not answer some of the questions, including the quantifiable questions, which is why the number of answers do not always add up to 10.

6.4.1 Respondents' Answers to the General Questions

When asked if it occurs that they encounter areas related to their work where they think digitalisation would be beneficial, 9 of the 10 respondents answered yes.

When asked to explain how, most of the respondents answers included information sharing. One respondent thought it would reduce the need to chase information and perform manual calculations, while another mentioned that not needing to call or email to get information would save time.

When asked if they had previous knowledge of PortCDM, 9 of the 10 respondents answered that they did.

6.4.2 Respondents' Opinions on the Synchronization Service

Figure 6.6 shows how relevant the respondents thought that the synchronization service was to their work.

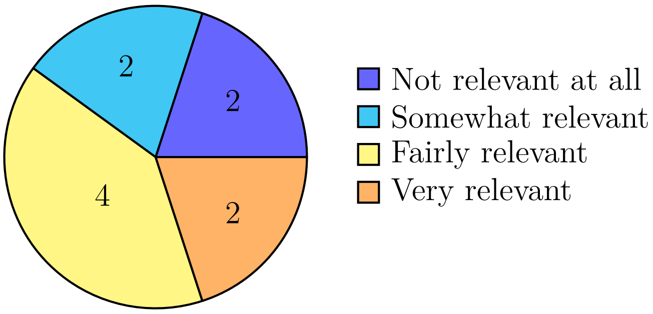


Figure 6.6: Relevancy of synchronization service to respondents' work.

As seen in Figure 6.7, most respondents for whose work the synchronization service would be relevant were interested in using it for their own work.

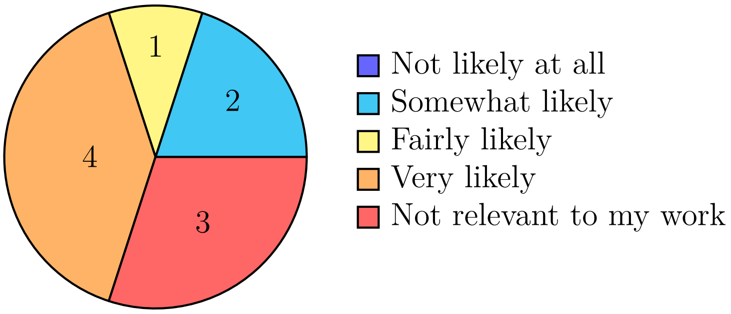


Figure 6.7: Respondents' interest in using the synchronization service in their work, if it was available free of charge.

When asked if they could think of any improvements or additions that would make the synchronization service more attractive, one respondent answered that cargo should be taken into consideration – that loading and unloading will be affected if the product is not ready or if the weather prohibits the cargo from being moved. Another respondent thought that including other berth options could be useful. Most of the respondents thought that more information from other sources like weather, daylight, geography and cargo type needed to be taken into consideration.

Figure 6.8 shows how respondents answered when asked how likely they were to be interested in using the synchronization service with the suggested improvements in their work.

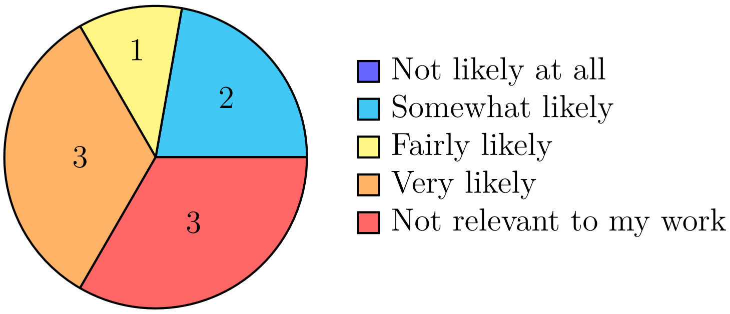


Figure 6.8: Respondents' interest in using the synchronization service in their work, if it were available with the suggested improvements or additions.

As seen in Figure 6.9, most respondents thought that the synchronization service would be somewhat likely or very likely to reduce delays in the port, with a majority answering that it was fairly likely.

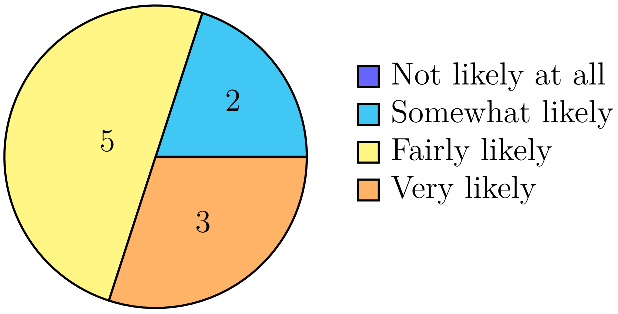


Figure 6.9: Respondents' opinions regarding potential for the synchronization service to reduce delays in the port.

As seen in Figure 6.10, most respondents thought that the synchronization service would be fairly likely or very likely to reduce the administrative burden of port actors, with a majority answering that it was fairly likely.

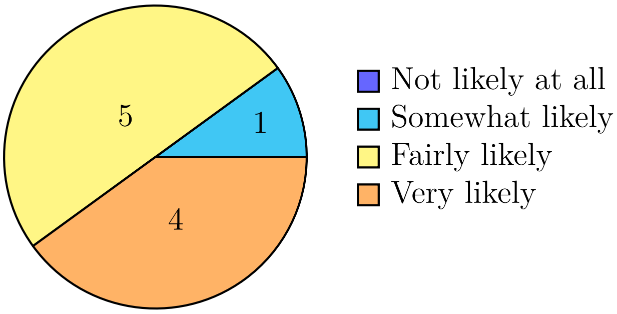


Figure 6.10: Respondents' opinions regarding potential for the synchronization service to reduce administrative burden of port actors.

As seen in Figure 6.11, all respondents thought that the synchronization service would be fairly likely or very likely be economically beneficial to port actors. The majority, six out of the ten respondents, thought that it would be very likely.

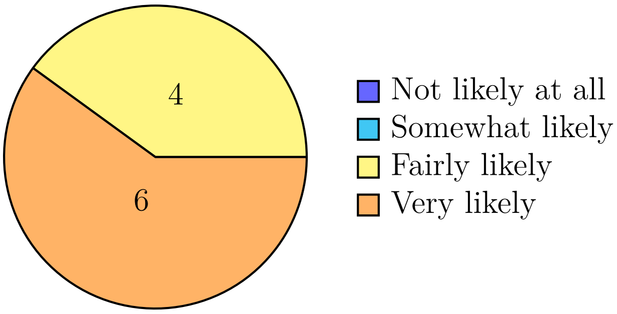


Figure 6.11: Respondents' opinions regarding potential economic value of the synchronization service.

When asked if they could think of any other benefits associated with the synchronization service, several respondents thought that it would be beneficial for the environment. One respondent also thought that it would reduce stress for those responsible for loading vessels with cargo, while another respondent thought that it would result in better vessel utilisation, safer transports, and more reliable data. One respondent thought that it could be used for small autonomous vessels in the future, and another respondent thought that it would be more likely that actors would be on time if they had agreed to a time. Another respondent noted that it was important for a Cargo Manager to know when they are allowed to approach and leave the berth, so that they can save money by booking in new cargo earlier and keep the vessel and cargo moving.

When asked if they could think of any problems or risks associated with the synchronization service, several respondents mentioned that people might become over-reliant on the system and be less up to date with the "real" situation, and ignore important human factors, which might result in queues, increased waiting times, and perhaps even create dangerous situations if the port were to become too crowded.

One respondent also noted that it would be difficult to include all time stamps in the system, and that there might be unexpected delays that would not be included in the system. Another respondent thought that it could result in vessels slowing down in order to meet a later RTA, but that it later turns out that the earlier window was actually available. Two respondents wondered whose responsibility it would be if a berth would turn out to be unavailable despite the service indicating that it was available, and one of them also noted that weather changes and special port regulations might result in the RTA not working as intended. One respondent noted that people might be afraid of automation.

When asked whether they had any other thoughts regarding the synchronization service, one respondent said that another actor that could be included would be the cargo inspector. Another respondent thought that the project was very exciting, but that in the end machine learning would not be necessary if data and logic could be used instead.

6.4.3 Respondents' Opinions on the Optimization Service

Figure 6.12 shows how relevant the respondents thought that the optimization service was to their work.

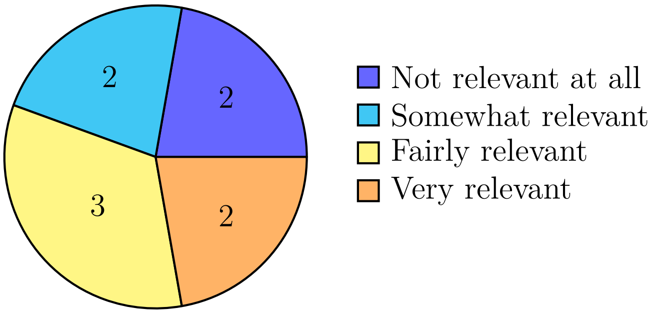


Figure 6.12: Relevancy of optimization service to respondents' work.

As seen in Figure 6.13, most respondents for whose work the optimization service would be relevant were interested in using it for their own work.

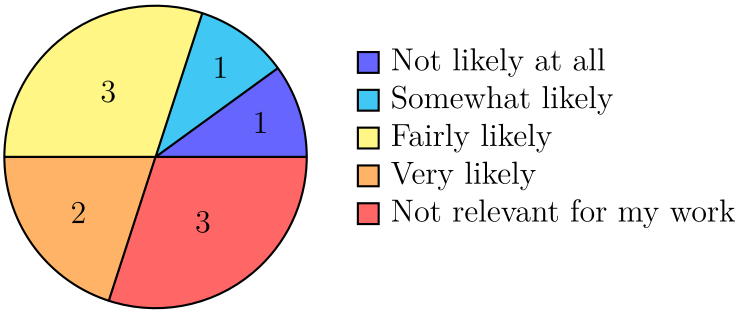


Figure 6.13: Respondents' interest in using the optimization service in their work, if it was available free of charge.

When asked if they could think of any improvements or additions that would make the optimization service more attractive, there were a lot of responses with a wide array of suggestions. Some respondents wanted different terminals and vessels to be taken into considerations. Two respondents would have liked for the optimization service to optimize for economic gains. One respondent wanted it to be more clear how much time would be saved. Another respondent wanted more concern to be shown to the customers, i.e. the owners of the vessels, noting that if specific vessels were to be frequently down-prioritised, they might choose another port. Another respondent wanted more options so that different actors could weight optimization fields on their own. Another respondent wanted that the complexity of port calls and the number of actors involved should be taken into consideration.

As seen in Figure 6.14, the majority of respondents were fairly likely or very likely to be interested in using the optimization service in their work, if the suggested improvements or additions were to be implemented.

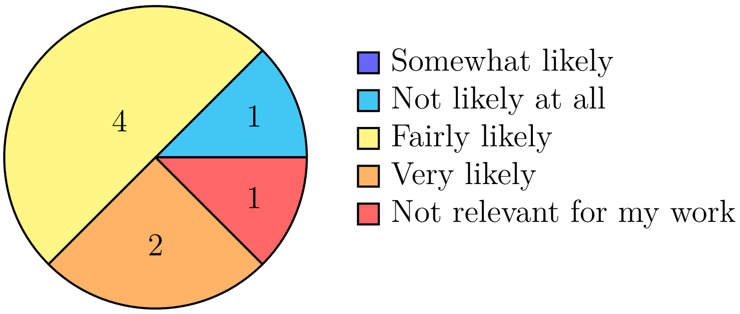


Figure 6.14: Respondents' interest in using the optimization service in their work, if it were available with the suggested improvements or additions.

As seen in Figure 6.15, one respondent thought that the optimization service was somewhat likely to reduce delays in the port, while the rest were evenly split between thinking that it was fairly likely or very likely.

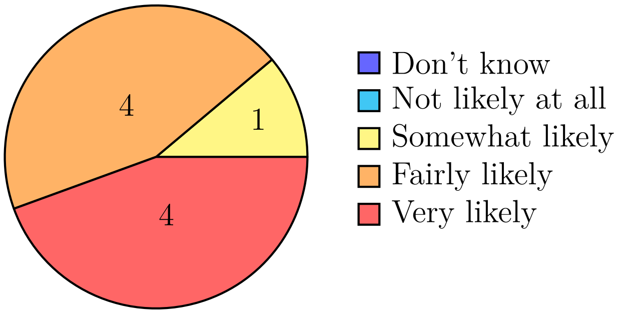


Figure 6.15: Respondents' opinions regarding potential for the optimization service to reduce delays in the port.

As seen in Figure 6.16, most respondents thought that the optimization service would be somewhat likely or very likely to reduce the administrative burden of port actors, with a majority answering that it was very likely.

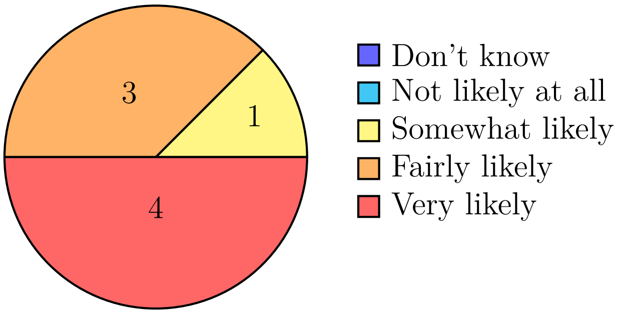


Figure 6.16: Respondents’ opinions regarding potential for the optimization service to reduce administrative burden of port actors.

As seen in Figure 6.17, views regarding the potential economic value of the optimization service for port actors were not decisive, although more than half the respondents thought it was fairly likely or very likely that the service would be economically valuable for port actors.

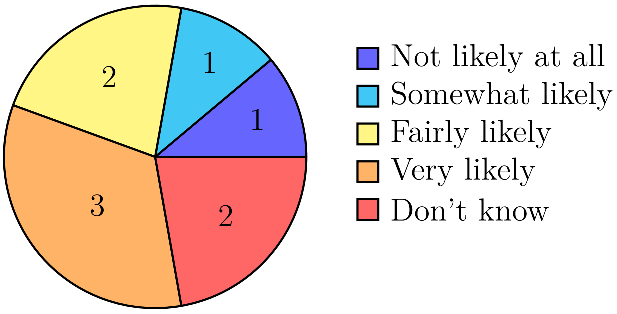


Figure 6.17: Respondents’ opinions regarding potential economic value of the optimization service.

When asked if they could think of any other benefits associated with the optimization service, one respondent thought that it would be a good planning aid and one thought it would make administrative work easier. Another respondent saw clear advantages for the environment due to reduced fuel consumption. They also thought it could be economically advantageous, as personnel could be replaced by software.

When asked if they could think of any problems or risks associated with the optimization service, one respondent mentioned the risk of replacing human jobs with software. They also highlighted the need for a change to the way that contracts between shipping companies and cargo owners are drafted. One respondent noted that vessels might feel that they are being prioritised in an unfair manner and choose other ports. One respondent thought that the optimization was focusing on the wrong metrics, and that more focus should be put on costs and penalties. Another respondent thought that optimization would prove to be too complex, as there are too many factors to consider.

When asked whether they had any other thoughts regarding the optimization service, one respondent said that they thought it was a good and positive idea, but that it requires more aspects. One respondent noted that it should be taken into considerations that vessels that require longer berthing times may incur higher penalty fees due to the greater volume of the cargo. Another respondent thought that combinations of various optimization principles for optimization should be used.

7

Discussion

This chapter discusses the result in relation to the theory. It discusses how well the results of the implementation correspond to reality and the motivation behind the implemented `Actors`. For the optimization, the objective was to produce a simple prototype showing its potential, rather than producing a final product. The machine learning turned out to be a difficult and time consuming part of the project. The final product does not quite reflect the effort put into the machine learning. It did, however, produce some interesting results and discoveries. The ethical and societal considerations are also discussed along with the approach of the project. Finally, how the product could be further developed in the future is discussed.

7.1 Correspondence of the Synchronizer with Reality

As previously mentioned, we chose to focus on the three actors which we deemed to be most essential, and as a result the `Synchronizer` is a proof-of-concept which needs to model more port actors in order to reach a point were it can be trusted for industrial deployment. However, it is important to stress that the main goal of this project never was to present a finished solution, but rather to show that some aspects of port operations could by automated. Apart from the lack of actors, the actors themselves are more complex than what we have been able to model using such a limited amount of time, resources, and access to data. The challenges that would need to be overcome in order to create a more realistic synchronizer are far from trivial.

Firstly, the communication processes between various actors often allow for flexible and uncertain outcomes which are difficult to predict. Additionally, it is often important to confirm mutual awareness of certain factors that impact the port call process, and therefore human-to-human communication is required.

Moreover, the implementation also deviates from the current port call synchronization process. It is not set to automate in a first-come first-served manner, but instead is carried out as book-first come-first, i.e. the vessels that book a given time slot first are given that slot. This can be quite problematic, as bookings would most likely be made a long time in advance. Although this can be a good thing since there would be no last minute surprises, it can also lead to cases were bookings are made despite the fact that the details are yet to be mediated in order to make sure that the time slots do not get taken. As a result, many bookings might be irrelevant after final routes have been decided, which could lead to the entire service being unreliable.

Another simplification that had to be made was to give each `Actor` the exact same time slot. For instance, the `Tug Actor` and `Berth Actor` have the same time allocated for a given port call even though they hardly overlap in practise. Fixing this would have made the model more realistic, but was not prioritised.

7.1.1 Choice of Actors

It was decided that a `Tug`, `Pilot` and `Berth Actor` had to be included in order to accurately simulate the port. Even though there are other important factors, such as weather, to take into consideration in a real life scenario, we did not have time to implement them all and had to choose the ones that were most crucial. The fact that data about port resources were unavailable made the `Actors` more difficult to implement than what was first anticipated.

7.1.2 Berth Actor

The `Berth Actor` is the simplest `Actor` of the three and was straight-forward to implement. Just like the other `Actors`, it contains a marginal time which is supposed to be a rough estimation of the time required between two events. No good data on this was found and the estimation can thus be far off in certain cases, as for instance different sized vessels have drastically varying turnaround times. The same problem applies to the other `Actors`. The marginal can, however, be set by the user via the API if the default value is found to be unsuitable.

We have not been able to take the fact that berths within terminals are often interchangeable into account. This is a salient shortcoming of our model, but unfortunately not one that could realistically be solved since the required data is not possible to obtain, and terminal procedures are very complex. Whether berths within terminals are interchangeable or not often depends on contracts, the order of vessel arrivals, the locations and sizes of other vessels in the terminal, and what personnel is available at that specific time.

7.1.3 Pilot and Tugboat Actor

Certain aspects of port operations, such as the number of tugboats available, depend on human decisions made in order to meet the demands imposed on the port by the arriving vessels. It is therefore often unwise to draw conclusions about factors such as port accessibility from current information. It is likely that the relevant factors have already been taken into account by the concerned parties, and that potential problems will be appropriately remedied at a later time.

Consequently, the way in which we have modelled the `Pilot Actor` and `Tugboat Actor` is quite naïve. In order for a model of their behaviour to be feasible, extensive changes would have to be made to administrative procedure and how information is communicated and shared. In particular, if the schedules of pilots and tugboats were available, the actors could be modelled to reflect reality more precisely. There have been discussions within the STM research community about how port resources could be modelled, and how information about their status could be shared, in real-time. Currently, however, such information is generally only available to the specific actors which control those resources, and that does not seem likely to change in the near future.

Both the `Pilot Actor` and `Tugboat Actor` are more complex than the `Berth Actor`, as they need to take a working crew into account, and in addition, the number of tugboats available to the `Tugboat Actor` varies. Since data regarding crew schedules and the number of tugboats are not available, it is difficult to realistically model the actors, and we instead had to rely on ML and input from the user.

7.2 Optimization

The optimization algorithm is first and foremost a proof of concept and is hence quite naïvely implemented. The optimization is only made with respect to the berth schedules and thus it may create complications in a real scenario, as the availability of other actors can not be guaranteed. However, rearranging the order should work in most cases where the scheduling was done using the `Synchronizer`, which assures that all the other actors are available. It is important to stress, though, that there are edge cases were another order may require, for instance, more tugboats than are available at the time.

The current implementation of PortCDM is not ideal for optimization. Firstly, the optimization algorithm assumes that a deadline exists, which specifies how the port call can be rescheduled. However, PortCDM has not implemented deadlines at the moment and the Estimated Time of Departure (ETD) is used instead. Using the ETD is not optimal as it does not present any leeway for a later deadline that most likely would be feasible in a real scenario, which inhibits the optimization algorithm. There are, however, plans on implementing time intervals for specifying such leeways in PortCDM, and in the future these could be used to derive a proper deadline.

Secondly, PortCDM acts passively. That is, data is only provided to users upon request. Furthermore it only provides data to the particular user who performed the request, even though others might be affected. This is one of the reasons why we implemented the optimization service as a tool to be used by those who plan port calls, rather than integrating it with the synchronization service.

For the `Optimizer` to be integrated with the `Synchronizer` the `Optimizer` would need to be able to change RTAs already sent, since if the conditions of the port change, previously generated RTAs might no longer be optimal. This is, however, not possible due to the current implementation of PortCDM, where RTAs can not be updated in that manner. If PortCDM were to be extended so that RTAs could be updated, integrating the `Synchronizer` and `Optimizer` would be more feasible.

Another thing that may make this implementation less desirable for real-life deployment is the goal of the algorithm. The algorithm focuses on reducing the overall lateness of the vessels. Optimizing for cost may be more desirable. In the current state a vessel with valuable cargo may have to stand aside for a vessel that in comparison would reap negligible financial benefits for being in time. However, data regarding costs were unavailable to us, which is why this was not done.

7.3 The Use of Machine Learning

The ANNs of `Berth Delay Actor` and `Tugboat Actor` are quite different and are discussed in detail below. With that said, the results of both ANNs could probably be improved by exploring other features than those used. In both cases the use of weather data, vessel specific information (e.g. cargo or the number of propellers), berth specific information (e.g. workload or current stoppages in operations), might allow for higher accuracy.

Additionally, it is worthwhile to note that no project members are experienced in the field of ML. All of the information required for implementation and evaluation was gathered during the course of the project. If others with more experience within ML were to conduct the project they might have chosen different methods or algorithms more suitable for the task.

Furthermore designing an ML model that makes perfect predictions is a difficult, if not impossible, task. Using ML to predict delays at berth or the number of tugboats to use will probably always leave some room for error. Thus, instead of making predictions based on historical data, real-time data should be used if errors are to be avoided.

7.3.1 Tugboat Predictions

As seen in Section 6.2.2 the accuracy of the ANN used by `Tugboat Actor` is $\sim 82\%$. It was decided to use the ANN in the final version of the system, but higher accuracy is probably needed for real life applications. One must remember that $\sim 18\%$ of the predictions are inaccurate, and each faulty prediction would, in a real life application, generate costs. In fact, it might even be dangerous since most vessels are unable to manoeuvre correctly in the port without the help of tugboats. When it was initially decided to use ML for this and other similar problems, it was thought that it would be possible to obtain much higher predictive accuracy, and thus a product that would be significantly more reliable and thus more applicable to real-world usage.

From Table 6.1 it is clear that the most common prediction of the ANN is zero no matter the actual value. This can be explained partially by the fact that the majority of the records use zero tugboats. In fact, 17589 out of 21462 records, ∼ 82% use zero tugboats. Furthermore Figure 7.1 reveals that distinguishing between the records using length and width is not easily done. For example, judging from Figure 7.1 a vessel with a length of 200 metres and width of 30 metres might require four, three, two, one or zero tugboats as there are yellow, black, green, blue and red points in the surrounding area of (200,30).

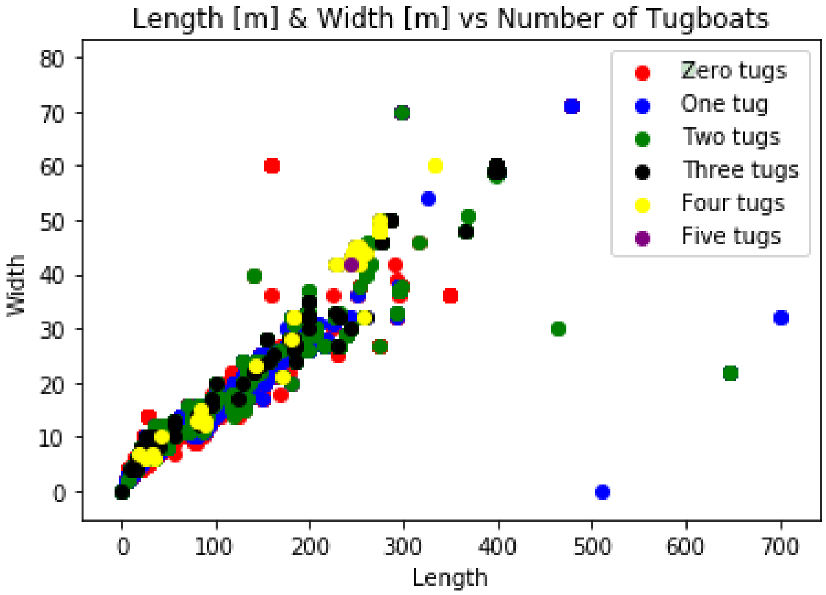


Figure 7.1: The length and width of the records, along with the number of tugboats required for each record.

As Figures 7.2 and 7.3 show, the type of the vessel probably allows an ML algorithm to draw some conclusions. From the figure it is clear that as the number of tugboats increase, the colours become darker. That is, only some types of ships, like tankers and cargo ships, require more tugboats. However, it is also clear that a lot of times those ships also require zero tugboats, which makes predictions harder.

Additional features could possibly have improved the performance of the ANN. For example, there could exist a strong correlation between the number of tugboats needed and which berth the vessel utilises. However, since the training data originates from the AIS system, each record would have to be mapped to a port call for PortCDM data to be used along with it. Since only a few weeks of the project remained when the training data from AIS was ready, the records were not mapped to port calls due to a lack of time.

The ambiguity in the data also tells us that other ML algorithms would probably not have been able to perform better than the one implemented. As we saw in Section 5.5.2, the kNN algorithm's accuracy was similar to the ANNs. Furthermore, other features might be closely correlated to the key, but if the related data contains the same degree of ambiguity as the features in figures 7.1, 7.2 and 7.3, performance is unlikely to increase. For an ML algorithm to perform better than the one implemented, other features with less ambiguity are needed.

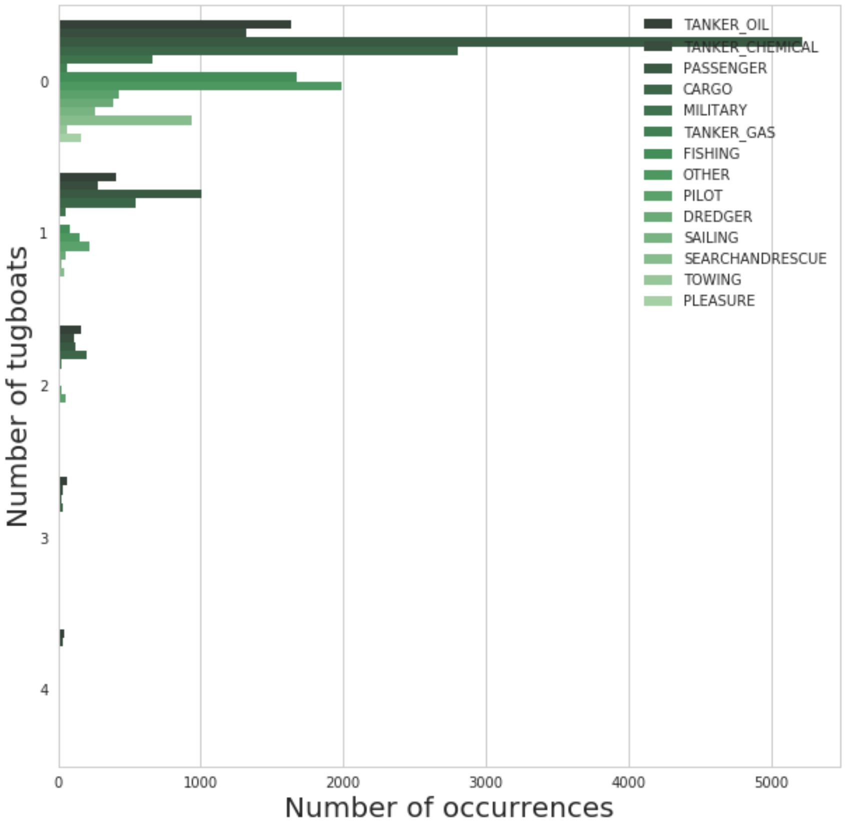


Figure 7.2: The number of tugboats used by each type and the number of such records.

7.3.2 Berth Delay Actor

As seen in Section 6.2, the accuracy of `Berth Delay Actor`'s predictions are quite poor. This is mainly due to ambiguity in the training data, which does not seem to allow for higher accuracy.

Even though the Pearson Correlation Coefficient only measures linear correlation, Table 5.1 reveals that the correlations between the features used and the key are quite weak. Furthermore, by plotting several of the features against the key it is clear that several of them simply do not allow for a higher accuracy.

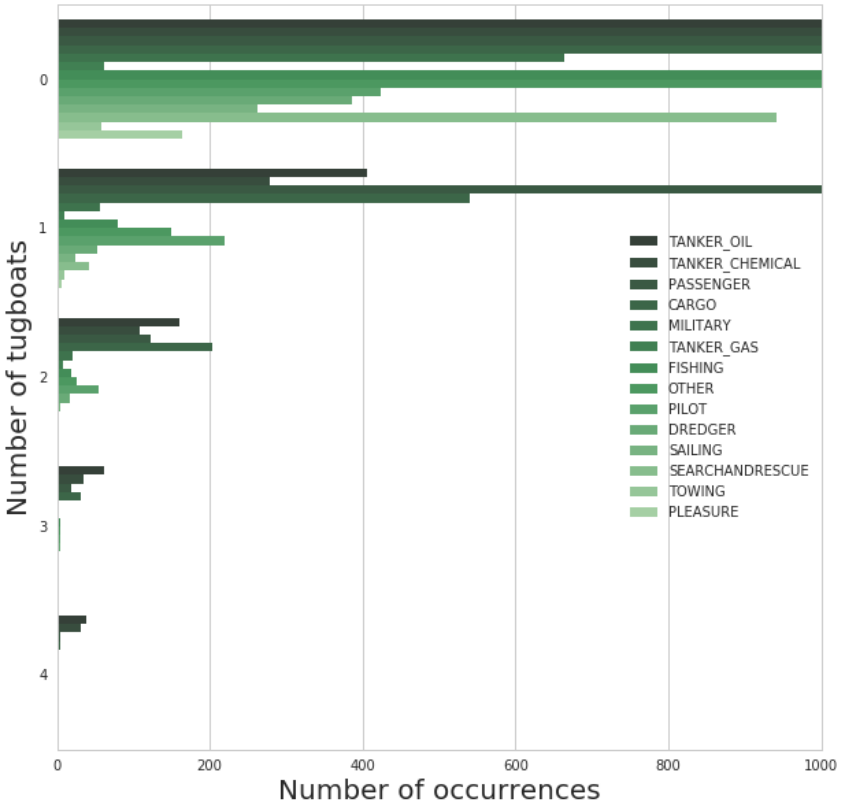


Figure 7.3: Figure 7.2 limited to 1000 occurrences.

The size of the vessel was expected to have some correlation to the delay. From figures 7.4 and 7.5 it is clear that ships which are wider and longer are less delayed. However, for ships with beam in the interval 10 to 30 metres and a length between 50 and 250 metres, the potential delay roughly ranges from -600 to 600 minutes. It is probable that this ambiguity prevents an ANN from drawing conclusions of records within that range. Other features, such as the duration of the stay and the number of ETAs sent showed a clearer correlation with the delay from berth, although the majority of the records, which have lower values, show ambiguity as seen in figures 7.6 and 7.7.

It is possible that higher accuracy could have been achieved by the use of other ML algorithms such as polynomial regression or logistic regression. Since no comparisons to other algorithms were made, this can not be said with certainty. However, it seems plausible that different algorithms would have yielded similar results, as the ambiguity in the data is believed to be the underlying problem. Other algorithms were not implemented due to a lack of time.

Another approach could have been to model the prediction of delays as a classification problem by letting intervals of delays correspond to classes. For example, one class could represent all port calls with delays within 0-99 minutes, another could represent all port calls with a delay of 100 - 199 minutes, and so on. Then the ANN would only have to differentiate between a small, finite, number of classes instead of approximating a continuous value. Unfortunately, we did not have time to try the classification approach.

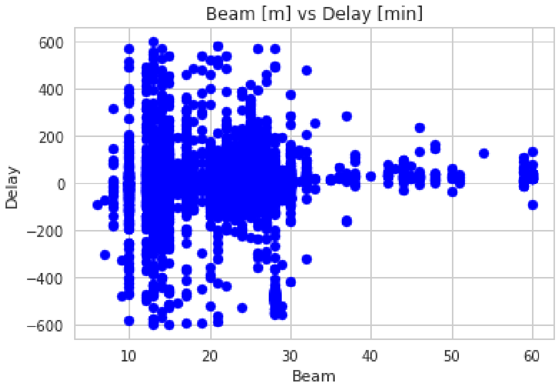


Figure 7.4: Delay of vessels plotted against their beam.

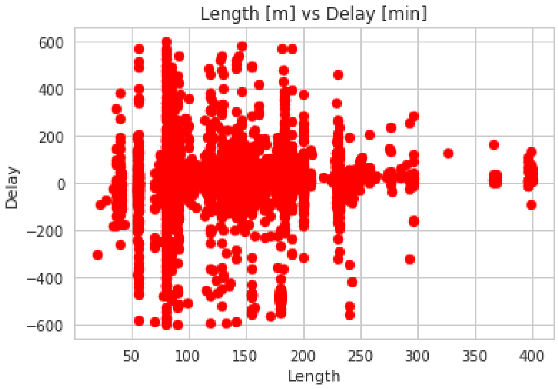


Figure 7.5: Delay of vessels plotted against their length.

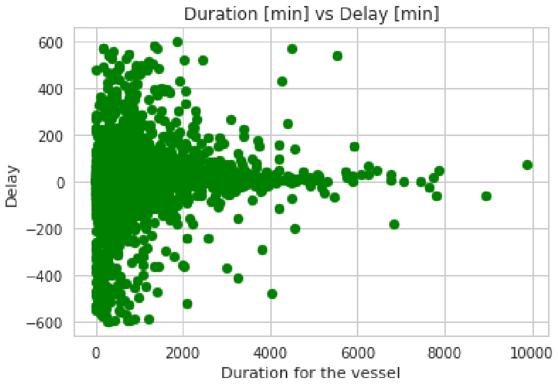


Figure 7.6: Delay of vessels plotted against the duration which the vessel has spent in berth.

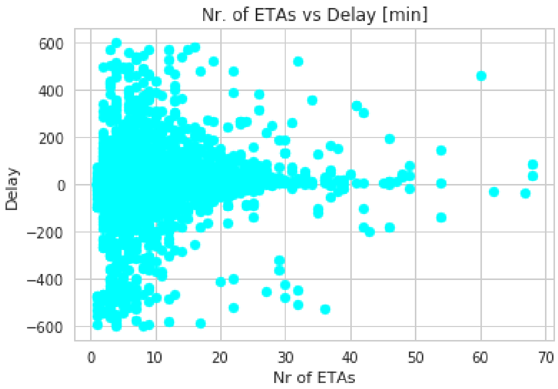


Figure 7.7: Delay of vessels plotted against the number of ETAs.

7.4 Reflections on the Project

This section will discuss the approach of the project as well as the time spent on working on supporting functions of the project. The rationale for not conducting an initial user study to find the functional requirements of our system will also be discussed.

7.4.1 Reflections on the Approach

While the product produced during the project satisfies the proof-of-concept approach taken, it is a very simplified version of the reality. For this reason some decisions made during the project have to be questioned. Should so much effort have been put into ML, even though the results barely have any impact on the product? Similarly, should so much effort have been put into obtaining the AIS data, since the same proof-of-concept could have been achieved with the existing PortCDM data? Should more effort instead have been put into making the model resemble reality more? These are some of the questions that surface as the project nears its conclusion.

Even though it is natural for these questions to surface and it is easy to be critical of the decisions that were made in hindsight, it was difficult to predict the outcomes of the choices made. Realistically, there was no way of knowing that processing the AIS data would be such a time-consuming task, or that most of the problems could be solved without Machine Learning.

Not knowing the outcome of a choice is a common situation. The choice to dive into Machine Learning came from the fact that it was a big part of the initial project idea and what drew the members of the group to the project. Machine Learning was also found to be one of the more theoretical and academic aspects of the project, which contributed to the choice of putting a lot of effort toward ML.

The choice to analyse AIS data was based on the fact that better data was needed to make it possible to predict more than zero or one tugboats with ML. Due to PortCDM's lack of port resource data, it was only possible to deduce if a vessel had tugboats or not. We were also curious about the potential benefits of AIS data, which contributed toward the choice of pursuing the analysis of it.

These decisions led to the model being somewhat neglected. There was not really any further development of it after the basic working model had been developed. As such, the model reflects the reality of a port call poorly in some cases.

Finally, the proof-of-concept approach gave us the opportunity to make critical decision like these, since only a working concept was needed. As such it was possible to make the decision to work on the synchronization and optimization parts separately. This decision was made so that smaller working prototypes could be developed. This also made it possible to work on these parts in parallel, instead of having one depend on the other in a full-scale, industry-ready application.

7.4.1.1 Omission of Initial Industry Survey

At the start of this project the team felt that the goal of the project was reasonably clear. The few things that weren't clear were also expected to be cleared up at an early meeting with the product owners. Because of this, an initial industry survey was never considered. As the first meeting with the product owners was delayed, the team felt that work on the project had to begin. As such, the team made decisions based on their current knowledge and started working. This is what lead to the initial system architecture described in Section 3.1. Later, when the first meeting with the product owners eventually took place, the team got pointed in a new direction. The product owners' new directives is what led to the refined system architecture described in Section 3.2. During this process an initial user study still was not considered. The team did not feel there was any need for it, as they had already received specific directives.

In hindsight and with the industry survey at hand, it is clear that a different path could have been taken at the start of the project. The industry survey revealed interesting suggestions and possible additions to the system that would have been useful at the start of the project.

7.4.2 Reflections on Supporting Functionality

As described at the end of Section 3.2, much of the project has been spent working on supporting functionality. This work has, however, not been discussed in our explanations of the implementation and results. The reason for this is that the team views the supporting functionality as a prerequisite for the project – both the knowledge to implement them and the time spent doing it. As such, even though time has been spent working on the supporting functionality and to some extent reading and researching them, this work has been omitted from the report. Another reason for this is that the supporting functionality is as the name suggests, simply supporting. It is not at the core of the project, even though it is necessary for the project to work. The team felt that the report needed to be concise and to the point of an already complex system. Therefore, sections dealing with exactly how connecting to PortCDM works or what SQL queries were used with the positional database were deemed superfluous and out of scope.

This decision might of course lead to some readers feeling that this information is missing from the report, that the project is not fully represented and can not be easily reproduced. The team believes that these readers will be few and that most readers will appreciate the more concise structure of the report.

7.5 Reflections on the Results of the Evaluation

The evaluation has given us insights into the value of the product, as all respondents were positive to the product as a whole.

As seen in Section 6.4.2, not all of the respondents thought that the synchronization service would be relevant to their work. The reason for those answers was that those respondents did not work at the port at the time of the interview. As mentioned in Section 3.3, some of the respondents worked as lecturers at Chalmers. The respondents did, however, think that the synchronization service would be economically beneficial, reduce administrative burden of port actors and reduce delays in the port. It is clear that there is a demand for a product like the synchronization service in the industry, since all of the respondents thought that it would be beneficial in many ways and some of them have already been asking for something similar to our product.

In Section 6.4.3, it is clear that the respondents were somewhat conflicted about the optimization service. Only half of the respondents found it relevant to their work. The respondents did think that it would reduce the administrative burden on actors in the port as well as reduce delays in the port. They were, however, split regarding if the optimization service would be economically beneficial or not. This is probably due to the current state of sea traffic, with contracts resulting in economical penalties for delays. This means that even if it would be beneficial to use the optimization service for the port and sea traffic as a whole, individual vessels and organisations might be penalised if they were to be scheduled later due to the optimization.

Some of the respondents mentioned positive effects on the environment as an outcome of the product. Environmental impact is discussed further in Section 7.6.1. An issue that was brought up was that humans could lose their jobs since the application can perform tasks previously performed by people. This problem is discussed further in Section 7.6.2. Other risks mentioned were whose responsibility it would be if the application were to fail to give a correct RTA, and that the services do not consider physical limitations of the port which could lead to congestion. These risks are discussed further in Section 7.6.3.

7.6 Ethical and Societal Considerations

In this section, the most important ethical and societal considerations related to the services that have been developed are discussed. More specifically, the discussions concern the potential consequences that could result from a wider industry adoption of services of this kind. The environmental impact, which has the potential to be very positive, as well as more problematic aspects such as potential job loss or increased risks are also discussed.

7.6.1 Environmental Impact

An important motivation for the project was to highlight ways in which the environmental impact of the shipping industry could be reduced. This could be realised if the services or services similar to those developed during this project were to become part of the standard practice in the shipping industry.

Currently, a lot of fuel is wasted when vessels make port calls due to inefficient procedures [56]. As mentioned in Section 1.2, vessels often race to the port due to competition for berth slots. That would not occur if the berth slot allocation process was formalised and determined by a strict protocol. This is one of the main purposes of the STM project and an automated system for generating RTAs would be very useful when further developing the platform.

Currently, RTAs are often not assessed in an optimal fashion, since there are so many parties and factors to consider. If RTAs could be communicated earlier to vessels, this would enable a more optimized port call process, thus potentially reducing fuel consumption greatly. For example, if unexpected delays at the port result in the need to postpone the arrival of a vessel, and the later RTA is communicated to the captain at an early stage, the speed of the vessel can be reduced, thus saving fuel.

If more efficient scheduling were adopted in ports, the potential for reducing fuel consumption would be even greater. The first-come first-serve system is, as we have seen in Section 4.2, not very efficient at all. Even our naïve implementation of schedule optimization highlights the gains that stand to be made by taking a more formal approach to the order of ship arrivals.

The beneficiaries of reduced fuel consumption are not limited to the shipping industry, though their financial incentives fortunately coincide with more general ethical considerations. The shipping industry is responsible for about 3% of global CO_2 greenhouse gas emissions – and they may increase by 50 to 250 percent until 2050 if actions are not taken [57], as well as the proliferation of other chemicals which are harmful to the environment [58]. Even marginal improvements would be greatly beneficial for the entire planet.

7.6.2 Replacing Humans

One may also wish to consider the impact of automation on the livelihoods of those whose work could be rendered redundant by more efficient processes. It is not entirely unreasonable to be concerned over this, but the specific work-related tasks which concern us in in this project are not the source of any significant amount of employment. Neither the manual labour nor the human guidance currently necessary for port calls is likely to become redundant in the near future, and even less likely as a consequence of the sort of automation discussed in this report. The tasks of certain employees may, however, become less stressful, and their administrative burden reduced – allowing them to focus their attention on more significant aspects of their work, thus also reducing the risk of accidents occurring.

Still, if one takes a very long view, it may be reasonable to expect the workload of certain roles, such as the agents', to be replaced by automated systems to such a degree that fewer individuals would need to be employed. But one then needs to counterbalance this with the economic growth that would occur in the shipping sector as a whole as a result of automation. Economic growth as the result of technological development is generally thought to lead to the creation of more jobs than it removes [59] – a process known as Creative Destruction [60] that has been widely studied in the field of economics.

It is important to stress that the purpose of the automation described in this report is not to actually replace the work which is currently carried out by human personnel, but to provide a basis for making more informed and timely decisions.

7.6.3 Operational Risks

Something that would need to be considered before services of the kind that we have developed can be deployed in the real world is the question of who is responsible if their usage results in undesirable consequences.

There is always a risk that oversights or mistakes could cause IT systems of this kind to malfunction. The sort of undesirable results that this would lead to is hard to predict, but there are a few that can be identified.

The most conspicuous risk is that the times or rescheduling suggested would be worse than what was planned before consulting the service. This could lead to substantial financial harm for actors whose work is impacted by the use of services of this kind, as well as environmental harm. There is a long way to go until systems of this kind can be fully automated rather than being an aid for making more informed decisions, and it is not what our services were designed for. Still, it is important that those using services like ours exercise their own sound judgement and not be too reliant on what the systems tell them, as the ultimate responsibility would have to fall on whoever sends out the RTAs.

A much less likely but more severe risk is that using the services might lead to congestion or levels of activity in the port that would be so high as to increase the risk of severe accidents occurring. This would obviously be extremely undesirable, as it could lead to the loss of life or severe environmental harm in the case of environmentally hazardous cargo. It would not be as a direct result of the use of systems like the ones described in this report, though, as it would require that key actors like the port authority consciously allow such risks to be taken.

It is important, however, that anyone using services like the ones described in this report be aware of the risks and caveats associated with them, and that they are not infallible.

7.7 In the Future

This Section offers an overview of possible improvements that can be implemented in the future. Additional `Actors` and more collaboration in the port can improve the correspondence of the system with reality.

7.7.1 Additional Actors

Even though the actors that were deemed the most relevant are included in the model, several other actors are required for the model to fully reflect all aspects of a port. These actors include moorers, sludge operators and the port authority. Some of these would be feasible to implement if the data in PortCDM were extended to include timestamps regarding their operations.

7.7.2 A More Collaborative Port

Many of the challenges associated with port operations stem from the many and diverse interests of the involved actors. Commercial incentives often align in a way that does not foster more collaboration than strictly necessary. Part of the purpose behind the development of PortCDM was to promote closer collaboration and more generous information sharing between port actors.

The world that the PortCDM project envisions is, unfortunately, not the world which we live in today. It is difficult to obtain the information and collaboration necessary to build functional and advanced maritime IT systems like the one developed during the course of this project. There are plans within the PortCDM research community to develop a system for sharing information about and tracking port resources, but it is not certain if the collaboration necessary to implement such a system will be possible to achieve at this stage.

It can be hoped that port actors can come to the realisation that collaborating and thereby working towards the “greater good”, while perhaps not always the most profitable strategy in the short term, will be in the best interests of everyone – including themselves – in the long run. PortCDM is leading the way in showing the considerable benefits that collaboration can lead to. We hope that our project can also contribute to this, and to highlighting the improvements that could be made if information were shared more gregariously.

8

Conclusions

The project has succeeded in showing that it is possible to automate the synchronization of port actors in a relatively easy and efficient manner. As long as accurate real-time data regarding the actors that are to be synchronized is accessible, this process has the potential to work quite well.

As it turned out, the necessary data described above was in many cases not available, and ML methods were used to partially compensate for this. It would be preferable, however, to not have to rely on predictive models, as they make high reliability very difficult to achieve.

As the results show, the approximations made by the `Berth Delay Actor` are not very accurate due to inadequate and ambiguous data. Thus, if delays are to be taken into consideration in future PortCDM services we recommend that, instead of predicting delays, real-time data be used. However, the `Berth Delay Actor` was created since the real-time data regarding delays was deemed unreliable. Therefore, if real-time data is to be used, time estimates need to be updated more frequently than they are now in order to increase reliability.

Furthermore, the `Tugboat Actor` also relies on ML to predict the number of tugboats needed in a port call. Although the accuracy of the `Tugboat Actor` is greater than that of the `Berth Delay Actor`, it is far from perfect. Additionally, it relies on data from AIS, which might be inaccurate. Rather than relying on predictions, we recommend that the number of tugboats needed is shared in real-time via PortCDM. If that were the case there would be no need for predictions, and thus less room for errors.

We have also been able to show that even with our simple schedule optimization algorithm, the potential for reducing waiting times in the port is great. The optimization would, however, greatly benefit from being more purpose-specific, and possibly more complex in order to meet wider optimization goals. As the PortCDM system does not have any effective authority, it is not particularly well-suited for implementing actual schedule optimizations for the real world. Another issues is again that the data coverage is incomplete, and PortCDM does not contain data regarding vessels that have not communicated an ETA.

The system that we developed during the course of this project is modular and thus extensible. It could be extended in order to take additional actors into account in the future. At the time of writing we were, however, unable to conceive of any additional actors that would be feasible to implement given the time-frame of the project and the character of the data in PortCDM. In order to extend the system with additional actors, the relevant data would have to be made available.

It is important to stress once more that the results are to be understood as a proof-of-concept, and not as a functional service that could be deployed as-is in the real world. It is our hope that our work should make it clear that a system of this kind is feasible to implement, given that the requisite data is available.

There have been discussions among PortCDM researchers regarding the possibility to develop a system for monitoring port resources. Such a system would require port actors to share more information, and would thus make it possible to greatly improve the reliability and functionality of our service.

The survey that was conducted showed that there is a lot of interest within the shipping industry for services of the kind that were developed during this project, and it would probably be easy to convince port actors to adopt such services. Many of the respondents also had good suggestions for improvements and additions to the services. It could therefore be valuable to further develop these in collaboration with the actors.

A central idea behind PortCDM and this project has been to highlight the benefits that can be reaped from increased collaboration. Many of the issues that have been outlined could be solved if port actors shared information more liberally and collaborated more closely. If PortCDM – with our project hopefully taking a small part – can lead the way in convincing the industry to adopt its vision of a more collaborative approach, there is reason to be optimistic about the future.

Bibliography

[1] UNCTAD. *Review of Maritime Transport 2017.* United Nations, 2018. ISBN: 9789211129229.

[2] Proshanto K Mukherjee. “Maritime law and admiralty jurisdiction: Historical evolution and emerging trends”. In: *The Admiral VI. Ghana Shippers’ Council.—2012.—42 p* (2012).

[3] Lincoln Paine. *The Sea and Civilisation – A Maritime History of the World.* Atlantic Books, 2014. ISBN: 978-1782393559.

[4] Trevor D Heaver. “The Evolving Roles of Shipping Lines in International Logistics”. In: *International journal of maritime economics* 4.3 (Sept. 2002), pp. 210–230. ISSN: 1476-0592. DOI: `10.1057/palgrave.ijme.9100042`. URL: `https://doi.org/10.1057/palgrave.ijme.9100042`.

[5] Jane Lister, René Taudal Poulsen, and Stefano Ponte. “Orchestrating transnational environmental governance in maritime shipping”. In: *Global Environmental Change* 34 (2015), pp. 185–195. ISSN: 0959-3780. DOI: `https://doi.org/10.1016/j.gloenvcha.2015.06.011`. URL: `http://www.sciencedirect.com/science/article/pii/S0959378015300054`.

[6] SAAB. *PORT MANAGEMENT INFORMATION SYSTEMS - PORTCONTROL™.* URL: `https://saab.com/security/maritime-traffic-management/traffic-management/portcontrol/` (visited on 05/06/2018).

[7] Mikael Lind et al. *Balancing just-in-time operations – Coordinating value creation.* URL: `http://fathom.world/wp-content/uploads/2018/03/STM-concept-note-6.pdf` (visited on 05/06/2018).

[8] Mikael Lind et al. “Sea Traffic Management – Beneficial for all Maritime Stakeholders”. In: *Transportation Research Procedia* 14 (2016). Transport Research Arena TRA2016, pp. 183–192. ISSN: 2352-1465. DOI: `https://doi.org/10.1016/j.trpro.2016.05.054`. URL: `http://www.sciencedirect.com/science/article/pii/S2352146516300540`.

[9] Anna Jivén et al. *Sjöfartens energianvändning – Hinder och möjligheter för omställning till fossilfrihet.* 2017. URL: `https://www.energimyndigheten.se/globalassets/klimat--miljo/transporter/sjofartens-energianvandning_slutrapport_feb.pdf` (visited on 01/30/2018).

[10] Naya Olmer et al. "Greenhouse gas emissions from global shipping, 2013–2015". In: (2017).

[11] Mikael Lind et al. "Port collaborative decision making – closing the loop in sea traffic management". In: *14th International Conference on Computer Applications and Information Technology in the Maritime Industries, Ulrichshusen, Germany.* 2015.

[12] Carl Enbom and Carl Frölund. *En analys av industriframväxt: Insikter från Sea Traffic Management.* 2015.

[13] Mikael Lind et al. "Digital Infrastructures for enabling Sea Traffic Management". In: *The 10th International Symposium ISIS*. 2014.

[14] Marielle Christiansen, Kjetil Fagerholt, and David Ronen. "Ship Routing and Scheduling: Status and Perspectives". In: *Transportation Science* 38.1 (2004), pp. 1–18. DOI: 10.1287/trsc.1030.0036. eprint: https://pubsonline.informs.org/doi/pdf/10.1287/trsc.1030.0036. URL: https://pubsonline.informs.org/doi/abs/10.1287/trsc.1030.0036.

[15] Dirk Steenken, Stefan Voß, and Robert Stahlbock. "Container terminal operation and operations research - a classification and literature review". In: *OR Spectrum* 26.1 (Jan. 2004), pp. 3–49. ISSN: 1436-6304. DOI: 10.1007/s00291-003-0157-z. URL: https://doi.org/10.1007/s00291-003-0157-z.

[16] Akio Imai, Etsuko Nishimura, and Stratos Papadimitriou. "The dynamic berth allocation problem for a container port". In: *Transportation Research Part B: Methodological* 35.4 (2001), pp. 401–417. ISSN: 0191-2615. DOI: https://doi.org/10.1016/S0191-2615(99)00057-0. URL: http://www.sciencedirect.com/science/article/pii/S0191261599000570.

[17] Adi Budipriyanto et al. "Berth allocation problem under uncertainty: A conceptual model using collaborative approach". In: *Procedia Manufacturing* 4 (2015), pp. 429–437.

[18] Sea Traffic Management. *About Sea Traffic Management.* URL: http://stmvalidation.eu/about-stm/ (visited on 05/10/2018).

[19] Andrew Cook. *European air traffic management: principles, practice, and research.* Ashgate Publishing, Ltd., 2007.

[20] Peter BM Vranas. "Optimal slot allocation for European air traffic flow management". In: *Air Traffic Control Quarterly* 4.4 (1996), pp. 249–280.

[21] Nikola Ivanov et al. "Air Traffic Flow Management slot allocation to minimize propagated delay and improve airport slot adherence". In: *Transportation Research Part A: Policy and Practice* 95 (2017), pp. 183–197. ISSN: 0965-8564. DOI: https://doi.org/10.1016/j.tra.2016.11.010.

URL: http://www.sciencedirect.com/science/article/pii/S0965856416300052.

[22] Zengqi Xiao and Jasmine Siu Lee Lam. "A systems framework for the sustainable development of a Port City: A case study of Singapore's policies". In: *Research in Transportation Business & Management* 22 (2017). Revisiting Port Governance and Port Reform: A Multi-country Examination, pp. 255–262. ISSN: 2210-5395. DOI: https://doi.org/10.1016/j.rtbm.2016.10.003. URL: http://www.sciencedirect.com/science/article/pii/S2210539516300694.

[23] Sternberg Henrik. "Transportation visibility and information sharing: a case study of actors' requirements". In: *World Review of Intermodal Transportation Research* 2.1 (2008), pp. 54–71.

[24] Göteborgs Hamn. *Allmänna hamnföreskrifter för Göteborgs Hamn.* 2017. URL: https://www.goteborgshamn.se/FileDownload/?contentReferenceID=12722 (visited on 05/10/2018).

[25] Sea Traffic Management. *STM Validation Project.* URL: http://stmvalidation.eu/projects/stm-validation/ (visited on 05/10/2018).

[26] Tom M. Mitchell. *Machine Learning.* 1997, pp. 231–236. ISBN: 0070428077.

[27] Yvonne Rogers, Helen Sharp, and Jenny Preece. *Interaction design: beyond human-computer interaction.* John Wiley & Sons, 2011.

[28] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. *Research methods in human-computer interaction.* Morgan Kaufmann, 2017.

[29] Tomoko Nemoto and David Beglar. *Likert-scale questionnaires.* 2014.

[30] Neon Rain Interactive. *Scrum Process.* URL: https://cdn-8a82.kxcdn.com/wp-content/uploads/2017/02/scrum_process_afa_5000.jpg.

[31] *SPSS Tutorials: Pearson Correlation.* 2018. URL: https://libguides.library.kent.edu/SPSS/PearsonCorr (visited on 05/03/2018).

[32] Jeffrey Stuckman, James Walden, and Riccardo Scandariato. "The Effect of Dimensionality Reduction on Software Vulnerability Prediction Models". In: *IEEE Transactions on Reliability* 66.1 (2017), p. 22.

[33] Lindsay I Smith. *A tutorial on Principal Component Analysis.* Feb. 2002. URL: http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf (visited on 04/26/2018).

[34] Fei-Fei Li. *Principal Component Analysis (PCA).* 2015. URL: http://www.cs.princeton.edu/courses/archive/fall08/cos429/CourseMaterials/lecture2/PCA_handout.pdf (visited on 04/16/2018).

[35] Selim Aksoy and Robert M. Haralick. *Feature Normalization and Likelihood-based Similarity Measures for Image Retrieval.* 2000. URL: http://www.cs.bilkent.edu.tr/~saksoy/papers/prletters01_likelihood.pdf.

[36] Sebastian Ruder. "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747* (2016). URL: https://arxiv.org/pdf/1609.04747.pdf.

[37] Joseph E. Beck and Beverly Park Woolf. "High-level Student Modeling with Machine Learning". In: (2018), p. 5. URL: https://pdfs.semanticscholar.org/f1b8/f25aedbe3e06f9aae23ffb9f01f5168e7f38.pdf.

[38] Stanislav Kolenikov, Gustavo Angeles, et al. "The use of discrete data in PCA: theory, simulations, and applications to socioeconomic indices". In: *Chapel Hill: Carolina Population Center, University of North Carolina* (2004), pp. 1–59.

[39] Ron Kohavi. "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection". In: Morgan Kaufmann, 1995, pp. 1137–1143.

[40] Josef Burger. *A Basic Introduction To Neural Networks*. Nov. 2010. URL: http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html (visited on 03/29/2018).

[41] Glosser.ca. *Colored neural network*. License: https://creativecommons.org/licenses/by-sa/3.0/. 2013. URL: https://upload.wikimedia.org/wikipedia/commons/4/46/Colored_neural_network.svg (visited on 03/29/2018).

[42] Michael A. Nielsen. *Neural Networks and Deep Learning*. 2015. URL: http://neuralnetworksanddeeplearning.com/ (visited on 04/13/2018).

[43] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. Chap. 6. URL: http://www.deeplearningbook.org (visited on 04/16/2018).

[44] Chih-Chung Chang Chih-Wei Hsu and Chih-Jen Lin. "A Practical Guide to Support Vector Classification". In: (2016), pp. 5–8.

[45] Siri Pettersen Strandenes. "Port pricing structures and ship efficiency". In: *Review of Network Economics* 3.2 (2004).

[46] Edi Shmueli and Dror G Feitelson. "Backfilling with lookahead to optimize the performance of parallel job scheduling". In: *Workshop on Job Scheduling Strategies for Parallel Processing*. Springer. 2003, pp. 229–230.

[47] Erik Røsæg. "A System for Queuing in Ports". In: *Scandinavian Institute of Maritime Law Yearbook* (2010). DOI: http://dx.doi.org/10.2139/ssrn.1697404.

[48] Neetu Goel and RB Garg. "A Comparative Study of CPU Scheduling Algorithms". In: *arXiv preprint arXiv:1307.4165* (2013).

[49] Erland Holmström. *Lecture 6*. 2018. URL: http://www.cse.chalmers.se/edu/course/TIN093_Algorithms_LP4/mtrl/lectures/F6.pdf (visited on 04/24/2018).

[50] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures (Ph.D.), Chapter 5*. 2000.

URL: http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm (visited on 05/06/2018).

[51] Ecma International.
Standard ECMA-404 2nd Edition - The JSON Data Interchange Syntax. 2017.
URL: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf (visited on 05/06/2018).

[52] IALA / AISM. *IALA GUIDELINES ON THE UNIVERSAL AUTOMATIC IDENTIFICATION SYSTEM (AIS) Volume 1, Part I – Operational Issues Edition 1.1.* 2002.
URL: http://www.ucakikazlambasi.com.tr/upload/pdf/d4ff31ce-f16b-4910-979e-2e05482e3519.pdf (visited on 05/06/2018).

[53] Wan Xiaoxia and Gan Chaohua.
"Electronic chart display and information system".
In: *Geo-spatial Information Science* 5.1 (2002), pp. 7–11.
DOI: 10.1007/BF02863487. eprint: https://doi.org/10.1007/BF02863487.
URL: https://doi.org/10.1007/BF02863487.

[54] Francois Chollet et al. *Keras.* https://keras.io. 2015.

[55] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python".
In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[56] Hannes Johnson and Linda Styhre. *Increased energy efficiency in short sea shipping through decreased time in port.* 2015. URL: http://www.sciencedirect.com/science/article/pii/S0965856414002857.

[57] Tristan Smith et al.
Third IMO GHG Study 2014: Executive Summary and Final Report.
July 2014.

[58] Liesbeth Schrooten et al.
"Emissions of maritime transport: A European reference system".
In: *Science of The Total Environment* 408.2 (2009), pp. 318–323.
ISSN: 0048-9697.
DOI: https://doi.org/10.1016/j.scitotenv.2009.07.037. URL: http://www.sciencedirect.com/science/article/pii/S0048969709007219.

[59] Dale T. Mortensen and Christopher A. Pissarides.
"Technological Progress, Job Creation, and Job Destruction".
In: *Review of Economic Dynamics* 1.4 (1998), pp. 733–753. ISSN: 1094-2025.
DOI: https://doi.org/10.1006/redy.1998.0030. URL: http://www.sciencedirect.com/science/article/pii/S1094202598900308.

[60] Philippe Aghion and Peter Howitt.
A model of growth through creative destruction. Tech. rep.
National Bureau of Economic Research, 1990.

A

Proof of Optimality of Synchronization Algorithm

A.1 Synchronization Algorithm

A.1.1 The Purpose and Goal of the Algorithm

The purpose of the synchronization algorithm is to find the first time at which all actors are able to accommodate the vessel the query concerns. We will prove that the synchronization algorithm we've created finds the first overlapping gap between all actors that is big enough to accommodate the vessel.

The problem can be formulated in a formal manner. There are n actors, $a_1, a_2, ...a_n$. The set of all actors is denoted A. At all times, each actor is either available or unavailable. Conceptually one can think of actors as time-lines containing intervals indicating when the actor is available. The purpose of the algorithm is to find a start time t_{start} such that all actors are available for a long enough time to service the vessel if they were to begin at t_{start}. Note that the time needed by each actor depends on the nature of the actor as well as the vessels properties.

In order for the algorithm to work it has to be able to query each actor if they're available from a certain start time. The algorithm thus makes use of a function called `available`. The function takes two arguments; the actor to query and the start time. It returns the earliest time after the given start time at which the actor is available for long enough. Hence, if the function was passed the start time t and the actor is available at t the function will also return t. If the actor isn't available at that time the function will return t' which is the first time after t that the actor is available for d. It follows that $t' > t$.

Thus, the goal of the algorithm is to find the *earliest* t_{start} such that:
$\forall a_i \in A : \texttt{available}(a_i, t_{start}) = t_{start}$

A.1.2 Description of how the algorithm works

The synchronization is performed according to the following algorithm:

1. The algorithm runs `available(`a_i, t`)` $\forall a_i \in A$. In the first iteration the desired time, t, is the TTA given through the API.
2. Each `Actor` responds with a time.
3. If the resulting times are different, the latest of these is chosen as a new desired time.
4. Step 1 through 3 are repeated until the desired time and the responses from each `Actor` are the same, then t is t_{start}.

A.1.3 Proof

We will prove that the algorithm is optimal by showing that our solution stays ahead of an optimal solution. Assume that there are n actors and call the desired start time t. Assume that there is an optimal solution t_{opt} which is the earliest possible start time after (or at) t. Starting from the first iteration all actors will be queried for their earliest start time.

There are two possibilities, either all of the actors return the same time or they don't. If they do the algorithm is done and t_{start} has been found. We know that $t_{start} \leq t_{opt}$ since t_{opt} also started from t and the earliest possible start time was t.

If all times aren't the same the algorithm has to proceed in choosing a new t to query all actors with. Let's assume that the new value of t isn't the latest of the times returned by the actors, t_{latest}. If the algorithm would proceed with this value of t the actor that returned t_{latest} will return t_{latest} again, since that was the earliest time at which that actor was available. It follows that the earliest possible start time will be restricted by t_{latest}, i.e $t_{latest} \leq t_{opt}$. Hence, there is no point in choosing any other value for t than t_{latest}. As our algorithm chooses $t = t_{latest}$ we will still have $t \leq t_{opt}$.

In the next iteration of our algorithm the same two possibilities present themselves again. Either all actors return the same time and t_{start} is found. We know that $t_{start} \leq t_{opt}$ since $t_{start} = t$. If they don't agree the new t_{latest} will place a new bound on the optimal solution, and since the algorithm will choose that time as it's new t our solution will still stay ahead of t_{opt}. This process will be repeated until an optimal solution is found.

Note that an optimal solution is assumed to exist. That is, it's assumed that at some point all actors have to return the same time. Theoretically this doesn't have to be the case, in the worst case the set of available intervals for the actors are mutually exclusive. In that case the algorithm wouldn't terminate. However this isn't the case in the context the algorithm is applied. All actors in the port will be available at some point, although it might be later than the initially requested time.

B

Survey

B.1 Survey Questions (in Swedish)

B.1.1 Generella frågor

1. Hur länge har du jobbat inom sjöfarten?

☐ 0-2 år
☐ 2-5 år
☐ 5-10 år
☐ 10-20 år
☐ 20+ år

2. Vad är ditt arbetsområde?

3. Händer det att du stöter på områden i ditt arbete där du tror att digitalisering skulle underlätta?

☐ Ja
☐ Nej

4. Om ja, förklara gärna på vilket sätt

5. Vårt projekt är ett tillägg till PortCDM. Känner du till PortCDM sedan tidigare?

☐ Ja
☐ Nej

B.1.2 Synkronisering

6. Hur relevant är synkroniseringstjänsten för ditt arbete?

☐ Inte alls relevant
☐ Lite relevant
☐ Ganska relevant
☐ Mycket relevant

7. Ifall synkroniseringstjänsten fanns tillgänglig utan kostnad, hur sannolikt är det då att du personligen skulle vilja använda den i ditt arbete?

□ Inte alls sannolikt
□ Lite sannolikt
□ Ganska sannolikt
□ Mycket sannolikt
□ Inte relevant för mitt arbete

8. Finns det några förbättringar eller tillägg som du tror skulle göra synkroniseringstjänsten mer attraktiv?

9. Ifall synkroniseringstjänsten fanns tillgänglig med dessa förbättringar, hur sannolikt är det då att du personligen skulle vilja använda den i ditt arbete?

□ Inte alls sannolikt
□ Lite sannolikt
□ Ganska sannolikt
□ Mycket sannolikt
□ Inte relevant för mitt arbete

10. Tror du att den här synkroniseringstjänsten skulle resultera i mindre förseningar för fartyg i hamnen?

□ Inte alls sannolikt
□ Lite sannolikt
□ Ganska sannolikt
□ Mycket sannolikt

11. Tror du att den här synkroniseringstjänsten skulle resultera i lägre administrativ börda för aktörer i hamnen?

□ Inte alls sannolikt
□ Lite sannolikt
□ Ganska sannolikt
□ Mycket sannolikt

12. Tror du att den här synkroniseringstjänsten skulle vara ekonomiskt gynnsam för aktörer i hamnen?

□ Inte alls sannolikt
□ Lite sannolikt
□ Ganska sannolikt
□ Mycket sannolikt

13. Kan du komma på några andra fördelar som synkroniseringstjänsten skulle kunna medföra?

14. Ser du några problem eller risker som användning av synkroniseringstjänsten skulle kunna medföra?

15. Har du några andra tankar kring synkroniseringstjänsten?

B.1.3 Optimering

6. Hur relevant är optimeringstjänsten för ditt arbete?

□ Inte alls relevant
□ Lite relevant
□ Ganska relevant
□ Mycket relevant

7. Ifall optimeringstjänsten fanns tillgänglig utan kostnad, hur sannolikt är det då att du personligen skulle vilja använda den i ditt arbete?

□ Inte alls sannolikt
□ Lite sannolikt
□ Ganska sannolikt
□ Mycket sannolikt
□ Inte relevant för mitt arbete

8. Finns det några förbättringar eller tillägg som du tror skulle göra optimeringstjänsten mer attraktiv?

9. Ifall optimeringstjänsten fanns tillgänglig med dessa förbättringar, hur sannolikt är det då att du personligen skulle vilja använda den i ditt arbete?

□ Inte alls sannolikt
□ Lite sannolikt
□ Ganska sannolikt
□ Mycket sannolikt
□ Inte relevant för mitt arbete

10. Tror du att den här optimeringstjänsten skulle resultera i mindre förseningar för fartyg i hamnen?

□ Inte alls sannolikt
□ Lite sannolikt
□ Ganska sannolikt
□ Mycket sannolikt

11. Tror du att den här optimeringstjänsten skulle resultera i lägre administrativ börda för aktörer i hamnen?

□ Inte alls sannolikt
□ Lite sannolikt
□ Ganska sannolikt
□ Mycket sannolikt

12. Tror du att den här optimeringstjänsten skulle vara ekonomiskt gynnsam för aktörer i hamnen?

□ Inte alls sannolikt
□ Lite sannolikt

☐ Ganska sannolikt
☐ Mycket sannolikt

13. Kan du komma på några andra fördelar som optimeringstjänsten skulle kunna medföra?

14. Ser du några problem eller risker som användning av optimeringstjänsten skulle kunna medföra?

15. Har du några andra tankar kring optimeringstjänsten?

B.2 Survey Translation

B.2.1 General Questions

1. For how long have you worked in with shipping?

☐ 0-2 years
☐ 2-5 years
☐ 5-10 years
☐ 10-20 years
☐ 20+ years

2. What is your professional field?

3. Does it occur that you encounter areas related to your work where you think digitalisation would be beneficial?

☐ Yes

☐ No

4. If yes, please explain how

5. Our project is an extension of PortCDM. Are you familiar with PortCDM?

☐ Yes
☐ No

B.2.2 Synchronization

6. How relevant is the synchronization service for your work?

☐ Not relevant at all
☐ Somewhat relevant
☐ Fairly relevant
☐ Very relevant

7. If the synchronization service were available free of charge, how likely is it that you would personally want to use it in your work?

☐ Not likely at all
☐ Somewhat likely
☐ Fairly likely
☐ Very likely
☐ Not relevant for my work

8. Are there any improvements or additions that you think would make the synchronization service more attractive?

9. If the synchronization service were available free of charge, with these improvements, how likely is it that you would personally want to use it in your work?

☐ Not likely at all
☐ Somewhat likely
☐ Fairly likely
☐ Very likely
☐ Not relevant for my work

10. Do you think that the synchronization service would result in less delays for vessels in the port?

☐ Not likely at all
☐ Somewhat likely
☐ Fairly likely
☐ Very likely

11. Do you think that the synchronization service would result in less administrative burden for port actors?

☐ Not likely at all
☐ Somewhat likely
☐ Fairly likely
☐ Very likely

12. Do you think that the synchronization service would be economically beneficial for port actors?

☐ Not likely at all
☐ Somewhat likely
☐ Fairly likely
☐ Very likely

13. Can you think of any other benefits that the synchronization service could result in?

14. Can you think of any problems or risks that the synchronization service could result in?

15. Do you have any other thoughts regarding the synchronization service?

B.2.3 Optimisation

6. How relevant is the optimisation service for your work?

☐ Not relevant at all
☐ Somewhat relevant
☐ Fairly relevant
☐ Very relevant

7. If the optimisation service were available free of charge, how likely is it that you would personally want to use it in your work?

☐ Not likely at all
☐ Somewhat likely
☐ Fairly likely
☐ Very likely
☐ Not relevant for my work

8. Are there any improvements or additions that you think would make the optimisation service more attractive?

9. If the optimisation service were available free of charge, with these improvements, how likely is it that you would personally want to use it in your work?

☐ Not likely at all
☐ Somewhat likely
☐ Fairly likely
☐ Very likely
☐ Not relevant for my work

10. Do you think that the optimisation service would result in less delays for vessels in the port?

☐ Not likely at all
☐ Somewhat likely
☐ Fairly likely
☐ Very likely

11. Do you think that the optimisation service would result in less administrative burden for port actors?

☐ Not likely at all
☐ Somewhat likely
☐ Fairly likely
☐ Very likely

12. Do you think that the optimisation service would be economically beneficial for port actors?

□ Not likely at all
□ Somewhat likely
□ Fairly likely
□ Very likely

13. Can you think of any other benefits that the optimisation service could result in?

14. Can you think of any problems or risks that the optimisation service could result in?

15. Do you have any other thoughts regarding the optimisation service?