Inter-terminal transport on Maasvlakte 1 and 2 in 2030

Towards a multidisciplinary and innovative approach on future inter-terminal transport options

Deliverable 3.2

Evaluation of Inter Terminal Transport Configurations at the Maasvlakte 1 and 2 using Discrete Event Simulation

For more information, contact:

Dr. Rudy Negenborn
r.r.negenborn@tudelft.nl
Transport Engineering & Logistics
Delft University of Technology

February 28, 2014

Delft University of Technology

FACULTY MECHANICAL, MARITIME AND MATERIALS ENGINEERING
Department Marine and Transport Technology

Mekelweg 2
2628 CD Delft
the Netherlands
Phone +31 (0)15-2782889
Fax +31 (0)15-2781397
www.mtt.tudelft.nl

Specialization:	Transport Engineering and Logistics
Report number:	2014.TEL.7826
Title:	**Evaluation of Inter Terminal Transport Configurations at the Maasvlakte 1 and 2 using Discrete Event Simulation**
Author:	H.J.L. Schroër

Title (in Dutch)	Evaluatie van Inter Terminal Transport Configuraties op Maasvlakte 1 en 2 met behulp van Discrete Simulatie

Assignment:	MSc Project
Confidential:	no
Initiator (university):	dr. R.R. Negenborn
Initiator (company):	HbR, Maurits van Schuylenburg
Professor:	prof. dr. ir. G. Lodewijks
Supervisor:	dr. ir. F. Corman
Date:	February 10, 2014

Delft University of Technology

FACULTY OF MECHANICAL, MARITIME AND MATERIALS ENGINEERING
Department of Marine and Transport Technology

Mekelweg 2
2628 CD Delft
the Netherlands
Phone +31 (0)15-2782889
Fax +31 (0)15-2781397
www.mtt.tudelft.nl

Student:	H.J.L. Schroër	Assignment type:	Master Thesis
Supervisor:	Dr.ir F. Corman	Report number:	2014.TEL.7826
Specialization:	TEL	Confidential:	No
Creditpoints (EC):	35		

Subject: **Development of a discrete simulation model for an inter terminal transport system at the Maasvlakte 1+2**

Due to an increasing demand in global containerized transport the Port of Rotterdam was forced to expand its Maasvlakte 1 with the new Maasvlakte 2. With this rise in container transport and new container terminals being built at the Maasvlakte 2, there will also be a rise in Inter Terminal Transport (ITT). The ITT system handles all containers that have to be transferred between the different container terminals within the Maasvlakte area. It is predicted that until 2020 the ITT can be performed using 3 TEU trucks that drive on the public road, but after 2020 this option will no longer suffice. Therefore a new and more sophisticated ITT system has to be developed.

This new ITT system is being analyzed within a collaboration project between the Port of Rotterdam and several universities. Within this project, expected transport demand scenarios for 2035 have been defined. An integer programming model was used to find rough estimations of the optimal transport configurations for given transport demand scenarios. The question that remains is how well these defined transport configurations actually perform.

The goal of this research is to create a reliable discrete simulation model with realistic outcomes, with which the potential of all defined ITT configurations can be evaluated.

To achieve this goal your assignment is to:

- Determine which scenarios the simulation model should be able to evaluate.
- Determine the requirements of the simulation model.
- Determine which methods should be used to meet the requirements.
- Based on your findings, develop a reliable discrete simulation model with realistic outcomes, with which the potential of all defined ITT configurations can be evaluated.
- Use the developed simulation model to evaluate the given ITT configurations and demand scenarios.

Based on the assignment, it is expected that you conclude with a recommendation for future research opportunities and potential for more ideas and/or applications. The report must be written in English and must comply with the guidelines of the section. Details can be found on the website.

For more information, contact Dr.ir F. Corman (B-1-320; F.Corman@tudelft.nl)

The professor,

Prof. dr. ir. G. Lodewijks

The supervisor,

Francesco Corman

F. Corman

This research is carried out within the framework of the TUDelft, Erasmus University and the Port of Rotterdam Authority joint project "'Inter-terminal transport on Maasvlakte 1 and 2 in 2030 - Towards a multidisciplinary and innovative approach on future inter-terminal transport options."

Summary

Over the past decades there has been an increasing demand in global containerized transport. Because of this demand the Port of Rotterdam was forced to expand its Maasvlakte 1 with the new Maasvlakte 2. It is expected that in 2040 the combined Maasvlakte 1 + 2 will handle at least 30 million TEU, which is almost four times as much as the entire Port of Rotterdam is handling now [10]. With this rise in container transport and new container terminals being built at the Maasvlakte 2, there will also be a rise in Inter Terminal Transport (ITT). Inter terminal transport is the transport of containers between terminals in a port.

The ITT system for the Maasvlakte is being analyzed within the project "Inter-terminal transport on Maasvlakte 1 and 2 in 2030 - Towards a multidisciplinary and innovative approach on future inter-terminal transport options.". It is a joint project between Delft University of Technology, Erasmus University Rotterdam and the Port of Rotterdam Authority. The goal of the project is to develop innovative, non-conventional concepts for ITT for the port of Rotterdam. Within this project, expected transport demand scenarios for 2030 have been defined by Rick Jansen [27]. An integer programming model was used by Frans Nieuwkoop [41] to find rough estimations of the optimal transport configurations for the given transport demand scenarios. The question that remains is "Which of the defined ITT vehicle configurations is the best configuration seen from an operational perspective?".

In order to find out how well the configurations perform, a discrete event simulation model for an Inter Terminal Transport system at the Maasvlakte 1 and 2 has been developed. The model makes it possible to evaluate all ITT vehicle configurations defined by Frans Nieuwkoop [41].

The input of the simulation model consists of 3 parts: the Maasvlakte infrastructure, the transport demand and the ITT vehicle configurations. The Maasvlakte infrastructure consists of 2 traffic networks, a road network and a water network, which connect a total of 18 container terminals and service centers. Although the simulation model is used in this research for the Maasvlakte area, it can be used for any possible ITT system by simply changing the network maps. The transport demand input consists of 3 different scenarios which have been determined by Rick Jansen [27]. The scenarios are predictions for 2030 and consist of an annual transport demand of respectively 3.340.000, 2.150.000 and 1.420.000 TEU. A total of 4 different vehicle configurations per scenario has to be evaluated. The configurations are: a number of AGVs, a number of ALVs, a number of MTSs and a combination of barges and trucks. The barges are not able to operate on their own because they are not able to reach every terminal in the system.

By far the most import task of the ITT system is to deliver the containers to their destination in time. In order to measure to what extend the system is able to perform this task, the performance indicator "non-performance" is used. If a container is delivered too late it is accounted as non-performance. Non-performance is the key performance indicator of the ITT system and will show the percentage of containers that has not been delivered in time. Other important performance indicators include the occupation rates of the vehicles and the terminal equipment, vehicle waiting times at the terminals, the number of idle vehicles and the total distance traveled by the vehicles.

Because of the discrete nature of the ITT system, the simulation model also needs to be discrete. Therefore a discrete event simulation model was developed using Delphi and the object oriented simulation tools provided by TOMAS. A number of dispatching rules is built into the system which decide on matters like choosing the modality with which to transport a container when barges are used and requesting empty vehicles from other terminals to transport a container.

Unlike previous built ITT simulation models, the new simulation model has a built in traffic modeling

system. Vehicles can experience delays at the intersections in the system. Each intersection decides which vehicle is allowed to cross the intersection first. Two different algorithms can be used to decide which vehicle to choose: a simple First-In-First-Out algorithm and a more advanced priority algorithm which considers container priority, wether vehicles are going in the same direction and wether they are able to cross at the same time without conflicts.

The simulation model is simulated at container level and it is object-oriented. It consists of the following objects: Containers, a Generator, an UrgencyCheck, Roads, Intersections, Terminals, Terminal Controls, Nodes, Terminal Equipment, Vehicles, Quay Cranes and Barges. The Containers, Roads and Nodes do not have a process and are therefore passive. All other objects are active. The vehicles (AGVs, ALVs, MTSs and Trucks) and barges travel through the system over a network of nodes and arcs. The nodes represent the terminals and intersections and the arcs represent the roads. The vehicles and barges both have a separate network. They use the Dijkstra algorithm to plan their path across the networks. Each terminal has its own control system which is able to request empty vehicles from other terminals to transport a container when no vehicles are available at the terminal itself. It is also used for the MTS scenarios to assign the terminal tractor part of the MTS to a trailer.

A number of experiments has been performed to evaluate the ITT configurations defined by Frans Nieuwkoop [41] and to gain more insight into the working of the ITT system. The non-performance values for the 12 ITT configurations have been given in Table 1. The ITT configurations are the results of the integer programming model, which means that these should have a non-performance of roughly 0 % in that model. As can be seen in the table, this is not the case for the simulation model.

Scenario	**Configuration**	**Non-performance [%]**	**Average lateness for late containers [hour]**
1	51 ALVs	18,3	7,67
	65 AGVs	41,5	37,24
	16 MTSs	40,7	78,18
	41 Trucks + 2 Barges	98,6	261,49
2	33 ALVs	11,2	6,10
	42 AGVs	39,4	20,83
	12 MTSs	26,7	13,75
	22 Trucks + 3 Barges	98,5	444,17
3	24 ALVs	2,5	0,60
	32 AGVs	21,7	3,83
	9 MTSs	19,3	3,69
	17 Trucks + 2 Barges	98,7	353,85

Table 1: Non-performance and time too late for the various ITT configurations

By far the most important performance indicators are how many containers are delivered in time and how much too late they have been delivered. Therefore the choice of the best ITT configuration will only be based on the non-performance and the average time that containers have been delivered too late.

Since the ALV configurations have by far the lowest non-performance and lateness values for each of the 3 scenarios, the ALV configurations are the best configurations.

However, this can only be concluded under the currently used dispatching rules and vehicle properties. Results have shown that the vehicle speed has a big influence on the system performance, which can be explained by the large distances in the ITT system. Vehicles spend most of their time driving. In the experiments the speed of the AGV and ALV have both been set to the same value, although the current ALVs are a bit slower than the current AGVs. This difference in speed might actually make the AGVs perform better than the ALVs. Also adding a proper planning system might make the less flexible configurations perform better than they do now.

The barge configurations score by far the worst for all 3 scenarios. The main reason these configurations score so poorly is due to the way they were modeled in Nieuwkoop's integer programming model. The barges were modeled continuous, instead of integer, because of memory issues. The result of this is that each container can be transported separately by a segment of a barge, without having to wait until a barge is full. This makes a barge of 50 TEU more or less work as a set of 1 or 2 TEU trucks, which are all used in an optimal way. In reality it does not work as efficient and flexible as this.

Barges do not seem to be a good option to be used in the ITT system. Handling them takes too much time; mooring alone already takes about an hour per visited terminal. Because of their large capacity, the large number of different terminals and the relatively short allowed delivery time of the containers, it is very hard to optimally use their capacity. The only way they might work is when they sail between terminals that share a lot of containers that allow a long delivery time.

More research is required in order to draw final conclusions from the ITT simulation model's results. Nieuwkoop's integer programming model should be rerun with updated input values. All configurations should be solved integer. The ITT simulation model should then be used to find the number of vehicles required to obtain a certain level of non-performance, with the integer programming model's updated configurations as a starting point, for all 12 instances.

Samenvatting

Wegens een toenemende vraag in wereldwijd container transport in de afgelopen decennia heeft de haven van Rotterdam haar Maasvlakte 1 uit moeten breiden met de nieuwe Maasvlakte 2. De verwachting is dat in 2040 de gecombineerde Maasvlakte 1 + 2 ten minste 30 miljoen TEU zullen behandelen, wat bijna vier keer zoveel is als de hele Rotterdamse haven nu behandeld [10]. Met deze stijging van het containervervoer en de nieuwe container terminals die gebouwd worden op de Maasvlakte 2 zal er ook een stijging van de Inter Terminal Transport (ITT) zijn. Inter terminal transport is het vervoer van containers tussen terminals in een haven .

Het ITT systeem voor de Maasvlakte wordt geanalyseerd binnen het project "Inter-terminal transport on Maasvlakte 1 and 2 in 2030 - Towards a multidisciplinary and innovative approach on future inter-terminal transport options.". Het is een gezamenlijk project van de Technische Universiteit Delft , de Erasmus Universiteit Rotterdam en het Havenbedrijf Rotterdam . Het doel van het project is om innovatieve, niet-conventionele concepten voor ITT voor de haven van Rotterdam te ontwikkelen. Binnen dit project zijn transportvraag scenario's voor 2030 vastgesteld door Rick Jansen [27] . Een integer programming model werd gebruikt door Frans Nieuwkoop [41] om ruwe schattingen van de optimale voertuig configuraties voor de gegeven transportvraag scenario's te vinden. De vraag die overblijft is "Welke van de gedefinieerde ITT voertuig configuraties is de beste configuratie gezien vanuit een operationeel perspectief?".

Om erachter te komen hoe goed de configuraties werken is er een discreet simulatie model voor een Inter Terminal Transport systeem op de Maasvlakte 1 en 2 ontwikkeld. Het model maakt het mogelijk om alle door Frans Nieuwkoop [41] gedefinieerde ITT voertuig configuraties te evalueren .

De input van het simulatiemodel bestaat uit 3 delen : de Maasvlakte infrastructuur, de transportvraag en de ITT voertuig configuraties. De Maasvlakte infrastructuur bestaat uit 2 verkeersnetwerken , een wegennetwerk en een waternetwerk, die een totaal van 18 container terminals en service centers verbinden. Hoewel het simulatiemodel in dit onderzoek wordt gebruikt voor de Maasvlakte kan het ook worden gebruikt voor andere ITT systemen door simpelweg de netwerk kaarten te veranderen. De input van de transportvraag bestaat uit 3 verschillende scenario's die zijn vastgesteld door Rick Jansen [27] . De scenario's zijn voorspellingen voor 2030 en bestaan uit een jaarlijkse vraag van respectievelijk 3.340.000, 2.150.000 en 1.420.000 TEU. Een totaal van 4 verschillende voertuig configuraties per scenario moet worden gevalueerd . De configuraties zijn : een aantal AGVs , een aantal ALVs, een aantal MTSs en een combinatie van binnenvaartschepen en vrachtwagens. De binnenvaartschepen zijn niet in staat om alleen te opereren omdat ze niet elke terminal kunnen bereiken.

Veruit de belangrijkste taak van de ITT systeem is om te zorgen dat containers op tijd op hun bestemming zijn. Om te meten in hoeverre het systeem in staat is om deze taak uit te voeren wordt de prestatie-indicator " non-performance " gebruikt. Als een container te laat wordt afgeleverd wordt hij geregistreerd als non-performance. Non-performance is de belangrijkste prestatie-indicator van de ITT systeem en zal het percentage laten zien van containers die niet op tijd zijn geleverd. Andere belangrijke prestatie-indicatoren zijn de bezettingsgraad van de voertuigen en de terminal apparatuur, voertuig wachttijden bij de terminals, het aantal inactieve voertuigen en de totale afstand afgelegd door de voertuigen.

Door de discrete aard van de ITT systeem , moet het simulatiemodel ook discreet zijn . Daarom werd een discrete event simulatiemodel ontwikkeld met behulp van Delphi en de object-georinteerde simulatie tool TOMAS . Een aantal beslissingsregels zijn ingebouwd in het systeem die beslissen over zaken zoals het kiezen van de modaliteit waarmee een container wordt vervoerd wanner binnenvaartschepen worden gebruikt en het aanvragen van lege voertuigen van andere terminals om een container te vervoeren.

In tegenstelling tot eerdere ITT simulatiemodellen heeft het nieuwe simulatiemodel een ingebouwd ver-

keersmodelleer systeem. Voertuigen kunnen vertragingen oplopen bij de kruisingen in het systeem. Elke kruising bepaalt welk voertuig het eerste mag oversteken. Twee verschillende algoritmen kunnen worden gebruikt om te bepalen welk voertuig te kiezen: een eenvoudige First-In-First-Out algoritme en een meer geavanceerde prioriteit algoritme dat rekening houdt met container prioriteit, of voertuigen in dezelfde richting gaan en of ze op hetzelfde moment zonder conflicten kunnen oversteken.

Het simulatiemodel wordt gesimuleerd op container niveau en het is object-georinteerd. Het bestaat uit de volgende objecten: containers, een generator, een urgencycheck, wegen, kruispunten, terminals, terminal controls, nodes, terminal apparatuur, voertuigen, kade kranen en binnenvaartschepen . De containers , wegen en nodes hebben geen proces en zijn dus passief . Alle andere objecten zijn actief . De voertuigen (AGVs, ALVs, MTSs en vrachtwagens) en binnenvaartschepen verplaatsen zich door het systeem via een netwerk van nodes en arcs. De nodes vormen de terminals en kruispunten en de arcs representeren de wegen. De voertuigen en schepen hebben beiden een apart netwerk. Ze maken gebruik van het Dijkstra algoritme om hun weg over hun netwerk te plannen. Elke terminal heeft een eigen controle systeem dat lege voertuigen kan aanvragen van andere terminals. Het wordt ook gebruikt voor de MTSs om de terminal tractor van de MTS toewijzen aan een trailer .

Een reeks experimenten is uitgevoerd om de ITT configuraties gedefinieerd door Frans Nieuwkoop te evalueren en om meer inzicht te krijgen in de werking van het ITT systeem. De non-performance waarden voor de 12 ITT configuraties zijn gegeven in Tabel 2. De ITT configuraties zijn de resultaten van de integer programmeer model, wat betekent dat deze een non-performance van ongeveer 0% zou moeten hebben in dat model. Zoals te zien is in de tabel is dit niet het geval voor het simulatiemodel.

Scenario	**Configuratie**	**Non-performance [%]**	**Gemiddeld te laat voor te late containers [uur]**
1	51 ALVs	18,3	7,67
	65 AGVs	41,5	37,24
	16 MTSs	40,7	78,18
	41 Trucks + 2 Barges	98,6	261,49
2	33 ALVs	11,2	6,10
	42 AGVs	39,4	20,83
	12 MTSs	26,7	13,75
	22 Trucks + 3 Barges	98,5	444,17
3	24 ALVs	2,5	0,60
	32 AGVs	21,7	3,83
	9 MTSs	19,3	3,69
	17 Trucks + 2 Barges	98,7	353,85

Table 2: Non-performance en tijd te laat voor de 12 ITT configuraties

Veruit de belangrijkste prestatie-indicatoren zijn hoeveel containers op tijd worden geleverd en hoeveel te laat ze zijn afgeleverd. Daarom wordt de keuze van de beste ITT configuratie alleen gemaakt op basis van de non-performance en de gemiddelde tijd dat containers te laat worden afgeleverd.

Aangezien de ALV configuraties veruit de laagste non-performance en te late waarden scoren voor alle 3 scenario's zijn de ALV configuraties de beste configuraties.

Dit kan echter alleen worden geconcludeerd onder de momenteel gebruikte beslissingsregels en voertuigen eigenschappen. De resultaten hebben aangetoond dat de voertuigsnelheid een grote invloed heeft op de prestaties van het systeem, hetgeen kan worden verklaard door de grote afstanden in de ITT systeem. Voertuigen zijn het grootste deel van de tijd aan het rijden. In de experimenten zijn de snelheid van de AGV en ALV beide dezelfde waarde hoewel de huidige ALV iets langzamer is dan de huidige AGV. Dit verschil in snelheid zou ervoor kunnen zorgen dat de AGVs beter presteren dan de ALVs. Ook het toevoegen van een goed planning systeem zou kunnen zorgendat de minder flexibele configuraties beter presteren dan dat ze nu doen.

De binnenvaartschepen configuraties scoren veruit het slechtst voor alle 3 scenario's. De belangrijkste reden waarom deze configuraties zo slecht scoren is te wijten aan de manier waarop ze werden gemodelleerd in Nieuwkoop's integer programmeermodel. De binnenvaartschepen werden continue gemodelleerd in plaats van integer wegens geheugenproblemen. Het resultaat hiervan is dat elke container afzonderlijk kan worden getransporteerd door een segment van een schip zonder te hoeven wachten tot een schip vol is. Dit zorgt dat een schip van 50 TEU min of meer werkt als een set van 1 of 2 TEU vrachtwagens die allemaal optimaal worden gebruikt In werkelijkheid werkt het niet zo efficint en flexibel.

binnenvaartschepen lijken geen een goede optie voor gebruik in de ITT systeem . De afhandeling ervan kost te veel tijd, alleen aanleggen en afmeren duurt al ongeveer een uur per bezochte terminal. Door hun grote capaciteit, het grote aantal verschillende terminals en de relatief korte toegestane levertijd van de containers is het erg moeilijk om optimaal gebruik maken van hun capaciteit. De enige manier waarop ze zou kunnen werken is als ze varen tussen terminals die veel containers uitwisselen die een lange levertijd toestaan.

Meer onderzoek is nodig om definitieve conclusies te trekken uit de resultaten van het ITT simulatiemodel. Nieuwkoop's integer programming model moet opnieuw worden gedraaid met bijgewerkte invoerwaarden. Alle configuraties moeten integer worden opgelost. Het ITT simulatiemodel moet vervolgens worden gebruikt om het aantal voertuigen te bepalen dat nodig is om een bepaald non-performance niveau te verkrijgen voor alle 12 configuraties, met de bijgewerkte configuraties uit het integer programmeermodel als uitgangspunt.

Abbreviations

AGV	= Automated Guided Vehicle
ALV	= Automated Lifting Vehicle
ASC	= Automatic Stacking Crane
FIFO	= First-In-First-Out
GA	= Genetic Algorithm
ITT	= Inter Terminal Transport
MTS	= Multi Trailer System
MV1	= First Maasvlakte
MV2	= Second Maasvlakte
O-D	= Origin-Destination
RS	= Reach Stacker
SC	= Straddle Carrier
TEU	= Twenty foot Equivalent Unit
TNow	= Current time in the simulation model
TOMAS	= Tool for Object-oriented Modeling And Simulation

Contents

Chapter 1

Introduction

Over the past decades there has been an increasing demand in global containerized transport. Because of this demand the Port of Rotterdam was forced to expand its Maasvlakte 1 with the new Maasvlakte 2. The Maasvlakte 2 is being built right next to the already existing Maasvlakte 1 and fully consists of reclaimed land from the sea, including almost 1.000 hectare of net terrain. A large part of this terrain will be used for container terminals. The full Maasvlakte area is shown in Figure 1.1, with the Maasvlakte 2 in orange.

It is expected that in 2040 the combined Maasvlakte 1 + 2 will handle at least 30 million TEU, which is almost four times as much as the entire Port of Rotterdam is handling now [10]. With this rise in container transport and new container terminals being built at the Maasvlakte 2, there will also be a rise in Inter Terminal Transport (ITT).

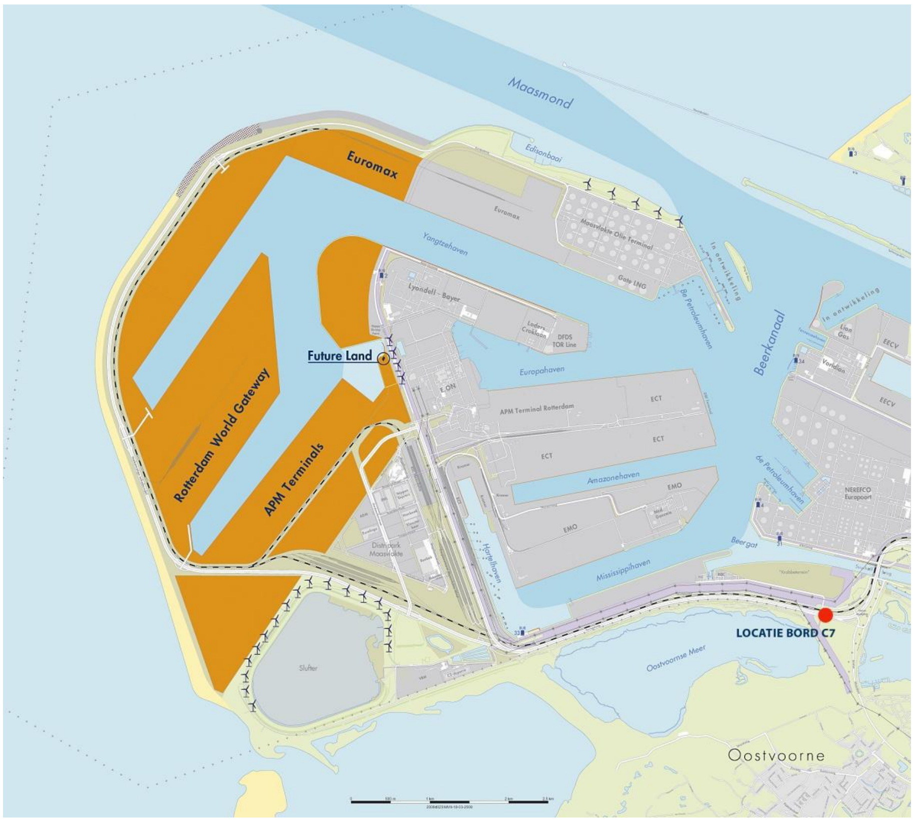

Figure 1.1: Maasvlakte 1 & 2 [1]

1.1 Inter Terminal Transport

Inter terminal transport is the transport of containers between terminals in a port. Not only marine terminals, but also service centers for dedicated handling of hinterland transport modes and support activities such as empty depots, customs, and distribution parks. Not all containers are directly transhipped to another modality like train, barge, or truck. Part of them first have to be moved to a different terminal in the port before they go their way. For instance to be stored in an empty depot, or to be put on a train or barge with containers from different deep sea terminals. An overview of the container flows in a port can be seen in Figure 1.2.

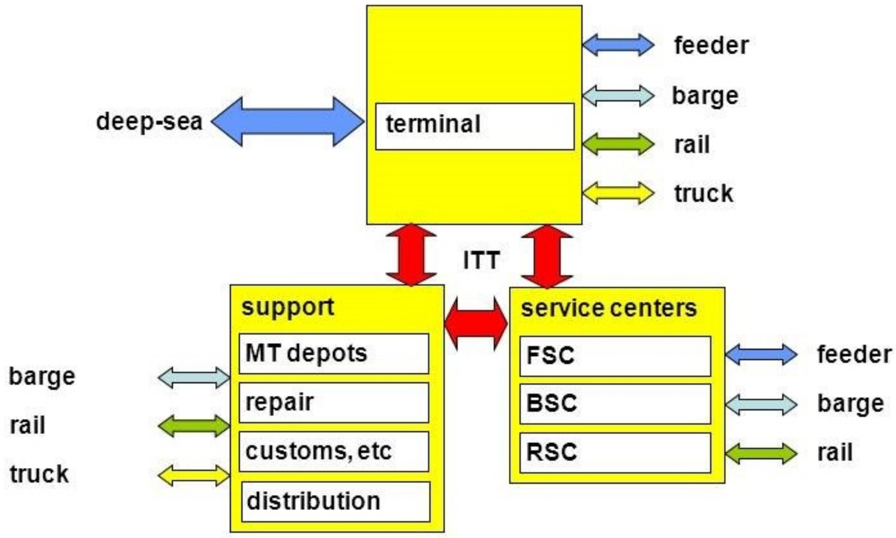

Figure 1.2: Container flows in a port [10]

Most of the research on ITT centers around a large research project commissioned by Incomaas [26], which was started in 1994 to evaluate a possible Inter Terminal Transport system for the Maasvlakte 1 in the Port of Rotterdam. The project was carried out by the section Logistic Technology of Delft University of Technology and the section Econometrics and Operation Research of the Erasmus University Rotterdam, under the guidance of Research School “TRAIL” [42][55].

As part of this project a detailed discrete simulation model was built [12]. The simulation model consists of several components, including 4 different models: a Generator Model, Advanced Planning, the ITT Simulation model and a Traffic density model. The relations between these components is shown in Figure 1.3.

The main component is the ITT Simulation Model. Within this model all interactions between the different modeled objects take place. There are 6 main objects: containers, a generator, handling centers, equipment, vehicles, and control [43]. The ITT system is simulated at container level, so a container object is created for every container that needs to be transported by the ITT. Each Terminal in the system consist of a number of handling centers. Each handling center has a number of equipment working there to transfer containers from and onto the vehicles. The vehicles are tasked with transporting containers from one handling center to another. Three different types of vehicle can be modeled: Automated Guided Vehicles (AGVs), Automated Lifting Vehicles (ALVs) and Multi Trailer Systems (MTSs). The control object is tasked with dividing the vehicles over the different handling centers and giving the equipment permission to load or unload a vehicle.

The Advanced Planning model is only used in the MTS scenario. It then takes over the role of the control object. The output of the Generator Model provides the input for the simulation model. It creates container flows for the ITT system based on information about incoming and outgoing ships, trains and barges. The Traffic density model has been created afterwards and does not communicate with the simulation model. It was only used to analyze the traffic flows of the MTS scenario using a transport demand and a network of nodes and arcs. Congestion has not been taken into account.

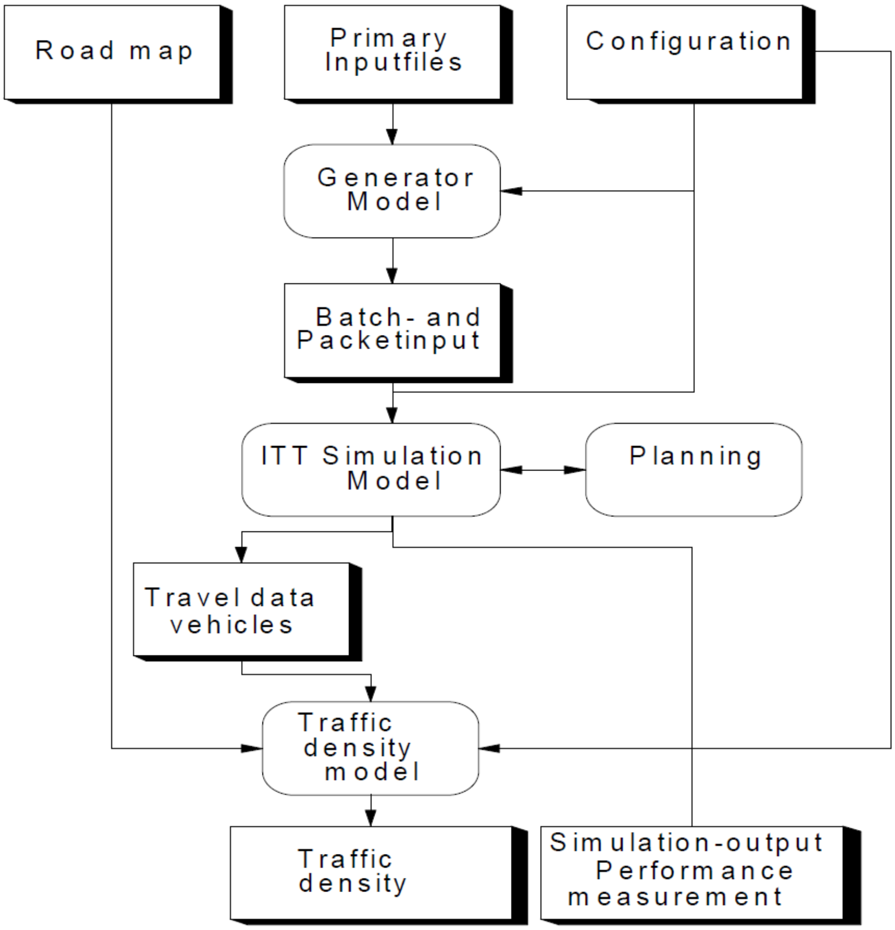

Figure 1.3: Overview and relations between models and files [15]

The main performance indicator in the model is non-performance. Every container has to be delivered within a certain time frame. If the system cannot deliver a container within its set time then it is accounted as non-performance. Other important performance indicators include: vehicle occupation rates, equipment occupation rates and the number of vehicles waiting at the terminals.

As part of the FAMAS.MV2 project [56], Ottjes *et al.* [45] used a new simulation model to evaluate conceptual multi-terminal designs for the second Maasvlakte (MV2), including ITT, in coherence with the existing terminals on the first Maasvlakte (MV1). The research assumed a transport demand scenario for 2025 with a fully developed MV2. However, the assumed MV2 layout does not correspond with the way the area finally turned out to be constructed. The research's focus is mostly on stack content and not so much on the performance of the system. Traffic flows within the ITT system have been analyzed, but congestion was not taken into account.

Tierney *et al.* [52] present a novel integer programming model for analyzing inter terminal transportation in new and expanding seaports. The model can operate with 4 different vehicle types: AGVs, ALVs, MTSs and barges. The barges are able to operate in parallel with the other vehicle types. The model takes congestion into account by giving intersection arcs a maximum capacity. Unlike in the discrete simulation model by Ottjes *et al.* [42], containers are only accounted as non-performance when they are delivered too late and not when they are delivered too early. The model is rather abstract, so its output consists of rough estimations of optimal ITT configurations. Therefore the authors suggest to use the model's output as input for a discrete event simulation of ports in order to provide a complete view of the impact of strategic decisions on port efficiency.

Diekman and Koeman [10] investigated wether the capacity of the existing infrastructure on the Maasvlakte is enough for the expected ITT transport. They predict that until 2020 the ITT can be performed using 3 TEU trucks that drive on the public road, but after 2020 this option will no longer suffice. Therefore a new and more sophisticated ITT system has to be developed. One of the options is to use a closed transportation route on which various types of vehicles could drive without interaction with other kinds of traffic. Different types of transportation systems to drive on the closed transportation route could be considered.

1.2 Inter Terminal Transport project group

The ITT system for the Maasvlakte is being analyzed within the project "Inter-terminal transport on Maasvlakte 1 and 2 in 2030 - Towards a multidisciplinary and innovative approach on future inter-terminal transport options.". It is a joint project between Delft University of Technology, Erasmus University Rotterdam and the Port of Rotterdam Authority. The goal of the project is to develop innovative, non-conventional concepts for ITT for the port of Rotterdam. The project consists of six different, yet interconnected, subprojects. These are the following:

- Task 1: Scenario definition
- Task 2: Truck and AGV configuration
- Task 3: Asset light configuration
- Task 4: Cost/Benefits evaluation
- Task 5: Information exchange evaluation
- Task 6: Operational evaluation

This research considers task 6: operational evaluation. The task is defined as follows: *"For the transport situation in the Port of Rotterdam a model will be developed that can be used to evaluate the different transport configurations from an operational point of view. This software will be able to run demand scenarios from Task 1. The simulation model will be a simplified derived from the one in Duinkerken et al. (2006) and will be developed to determine expected waiting times; type of infrastructure and number of cranes required; capacity balance peak and average needs."*

A simulation model is presented which is able to evaluate various ITT configurations from an operational point of view. The simulation model's input comes from two other tasks in the project: "scenario definition" and "truck and AGV configuration".

The "scenario definition" task has been performed by Rick Jansen [27] and has resulted in 3 transport demand scenarios for the ITT system. These scenarios have been used as input for the "truck and AGV configuration" task performed by Frans Nieuwkoop [41]. Nieuwkoop used an integer programming model based on a model developed by Tierney *et al.* [52] to find rough estimations of the optimal transport configurations for the various transport demand scenarios. The question that remains is how well these transport configurations actually perform seen from an operational perspective. In order to evaluate the performance of the different transport configurations, a more realistic simulation model needs to be developed.

1.3 Research question

The objective of this research is to evaluate the transport configurations defined by Nieuwkoop [41]. Therefore a simulation model with realistic outcomes needs to be developed, with which the potential of all defined ITT configurations can be evaluated. The simulation model should be able to determine performance indicators such as the amount of containers delivered in time, waiting times, vehicle and terminal equipment occupancy and delays due to traffic. The model should be easily adaptable to test different scenarios with different vehicle types, amounts of vehicles, transport demands, and infrastructure.

In order to reach the research objective, the following research question needs to be answered:

Which of the defined ITT vehicle configurations is the best configuration seen from an operational perspective?

The research question will be answered by the hand of a set of sub questions. These are the following:

- Which scenarios and configurations should the model be able to evaluate?

- What are the key performance indicators for the ITT system?
- Which modeling methods should be used in order to meet the requirements for the simulation model?
- Which physical objects need to be modeled?
- Which interaction need to take place in the simulation model?
- How can the model be verified?

1.4 Structure of the report

The report will answer all research questions stated in the previous section. It is structured as follows: Chapter 2 discusses the instances that need to be simulated by the ITT simulation model, including the Maasvlakte area, the transport demand scenarios, the ITT configurations and the main performance indicators. Chapter 3 discusses modeling methods on the subjects of simulation, vehicle scheduling, and traffic modeling. Choices are made on which methods to implement in the ITT simulation model. Chapter 4 gives a detailed explanation of the working of the model. Chapter 5 explains the verification of the model. Chapter 6 shows the results of the simulation runs that have been performed in order to evaluate the given ITT configurations. Chapter 7 concludes the report and gives recommendations for future research.

1.5 Research contribution

This research will result in a simulation model which can be used to evaluate all kinds of different operational aspects of a not yet existing inter terminal transport system. Thereby is will help build an understanding of how such an ITT system would operate, and which effects the adjustment of parameters will have on the performance of the system. The model will not only be able to test the performance of varying vehicle configurations, but also of varying infrastructure, transport demand and terminal capacity. The model will be the first ITT simulation model to incorporate traffic modeling. Delays occurring due to traffic will have an impact on the system's performance.

Chapter 2

Model input and output

The ITT simulation model will need to be able to evaluate different transport scenarios. Each scenario consists of 2 parts: a transport demand and an ITT configuration. Together with the Maasvlakte infrastructure, these two parts form the input of the simulation model. By the hand of these input variables, the model needs to calculate the operational performance of the various settings.

2.1 Maasvlakte

The area to be simulated is the combined Maasvlakte 1 + 2 in the Port of Rotterdam. The area consists of a number of container terminals and service providers between which Inter Terminal Transport will take place over a closed transport route. A map of the terminals and service providers and the roads between them on which the ITT will take place is shown in Figure 2.1.

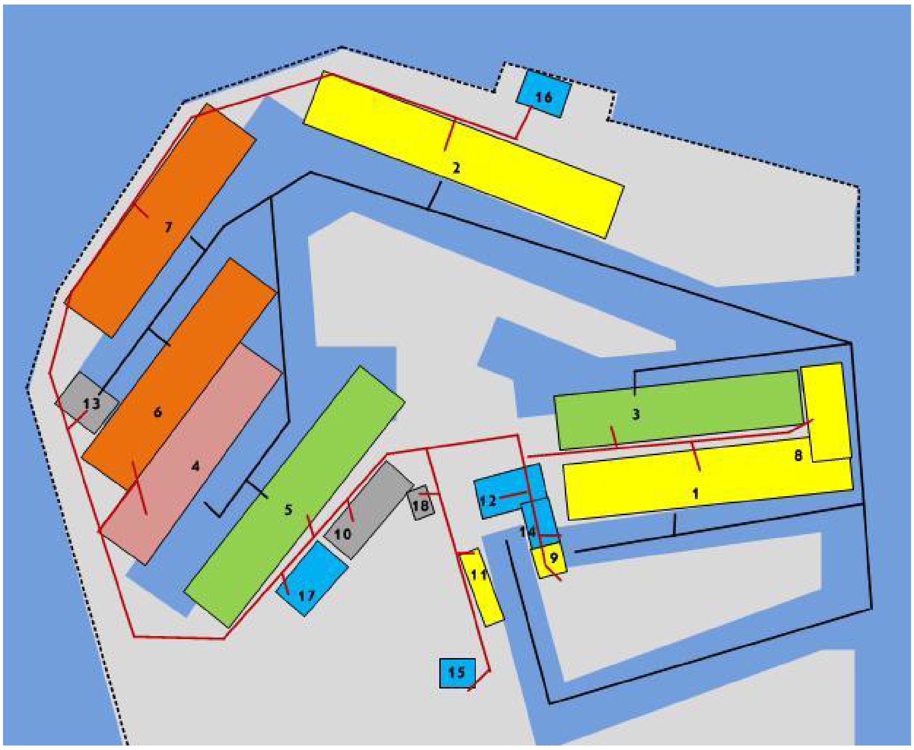

Figure 2.1: Map of the Maasvlakte [27]

In total there are 18 terminals that will be part of the ITT system. All of them are accessible by road, but only part of them are accessible by water. The terminals that are part of the ITT system are shown in Table 2.1

Number	Terminal	Type
1	ECT Delta Terminal	Deep Sea Terminal
2	Euromax Terminal	Deep Sea Terminal
3	APM MV1 Terminal	Deep Sea Terminal
4	RWG	Deep Sea Terminal
5	APM MV2 Terminal	Deep Sea Terminal
6	T3	Deep Sea Terminal
7	T4	Deep Sea Terminal
8	ECT Delta Barge Feeder Terminal	Deep Sea Terminal
9	Delta Container Services	Deep Sea Terminal
10	Common Rail Terminal	Common Rail Terminal
11	Rail Terminal West	Common Rail Terminal
12	Barge Service Center Hartelhaven	Common Barge terminal
13	Common Barge Service Center	Common Barge terminal
14	Kramer Delta Depot	Empty Depot
15	Van Doorn Container Depot	Empty Depot
16	Empty Depot MV1	Empty Depot
17	Empty Depot MV2	Empty Depot
18	Douane	Customs

Table 2.1: Container terminals and service providers in the ITT system [27]

2.2 Transport demand scenarios

The different transport demand scenarios for the ITT system have been defined by Rick Jansen [27] as part of the "scenario definition" task of the "Inter-terminal transport on Maasvlakte 1 and 2 in 2030" project. In total there are 3 different scenarios. The annual transport demand for these scenarios has been given in Table 2.2. More information on the transport demand scenarios can be found in Appendix E.

Scenario	Annual transport demand [TEU]	Mean amount of containers to be transported per hour
1	3.340.000	223
2	2.150.000	144
3	1.420.000	95

Table 2.2: Annual transport demand per scenario [27]

The transport demand input for the simulation model consists of a list of container transport jobs. There is one list of container transport jobs for each scenario. These lists have been created using an Arena [2] based demand generator. The generator is deterministic, so if the input does not change it will generate the same list every time.

For each container to be transported the following data is available:

- Release time Time the container enters the system
- Origin Terminal the container needs to be transported from
- Destination Terminal the container needs to be transported to
- TEU Wether the container is 1 or 2 TEU
- Due time Latest time the container is allowed to be delivered at its destination

An example of a transport demand input file has been given in Appendix B.

2.3 ITT configurations

For each transport demand scenario, optimal ITT vehicle configurations have been defined using an integer programming model based on a model developed by Tierney *et al.* [52]. This research was carried out by Frans Nieuwkoop [41] as part of the "truck and AGV configuration" task of the "Inter-terminal transport on Maasvlakte 1 and 2 in 2030" project.

Nieuwkoop defined 4 optimal vehicle configurations per transport demand scenario, so a total of 12 configurations. Each configuration consists of two values: the type of vehicle and the amount of vehicles. In total there are 5 different vehicle types: AGVs, ALVs, MTSs, trucks and barges. These vehicles will be discussed in Section 2.4. A configuration only with barges is not possible because the barges are not able to reach every terminal, therefore the barges work alongside a number of trucks. The results of Nieuwkoop's model and therefore the configurations that need to be evaluated using the simulation model have been given in Table 2.3. These results mean that with these configurations, more than 99% of the containers should be able to be delivered in time.

Scenario	Configuration
1	51 ALVs 65 AGVs 16 MTSs 41 Trucks + 2 Barges
2	33 ALVs 42 AGVs 12 MTSs 22 Trucks + 3 Barges
3	24 ALVs 32 AGVs 9 MTSs 17 Trucks + 2 Barges

Table 2.3: Required vehicles for the various scenarios resulting from the research by Nieuwkoop[41]

2.4 ITT vehicles

As discussed in the previous section, five different types of ITT vehicles will be considered: AGVs, ALVs, MTSs, barges and trucks. The vehicles have been shown in Figure 2.2.

Figure 2.2: Vehicle types

Automated Guided Vehicle (AGV): The AGV is an autonomous vehicle that is used in many large container terminals to perform the transport from the quay to the stack. It is able to carry 2 TEU.

Automated Lifting Vehicle (ALV): The (Gottwald) ALV is a rather new vehicle and will first be used at the APM Terminal at the Maasvlakte 2 [3]. In many ways it is the same as the AGV, but it has a lifting system which allows it to pick up a container from a platform. Due to this system the container transport is decoupled from the storage process, so the ALV and terminal equipment don't have to wait for each other to make a move. The downside of the ALV is that it a bit heavier, and therefore possibly slower, than the conventional AGV. It is also able to carry 2 TEU.

Multi Trailer System (MTS): A Multi Trailer System consists of a manned terminal tractor pulling a train of terminal chassis. A tractor usually pulls 5 trailers, which gives the MTS a capacity of 10 TEU.

Barge: The barge is the only vehicle type that does not use the closed transport route, but instead travels over the water. The barge in not able to visit all terminals at the Maasvlakte, and will therefore only operate alongside one of the other vehicle types. Barges can usually carry 50 to 100 TEU.

Truck: The truck, or terminal tractor, is a manned vehicle able to carry 2 TEU. From a modeling perspective, its process is the same as for the AGV.

The properties of the vehicles used in the simulations have been given in Section 6.2.

2.5 Performance indicators

The simulation model should be able to evaluate the performance of the ITT system under the set input parameters. Therefore it needs to calculate a number of performance indicators.

By far the most import task of the ITT system is to deliver the containers to their destination in time, in other words; before their due time. In order to measure to what extend the system is able to perform this task, the performance indicator "non-performance" is used. If a container is delivered too late it will be accounted as non-performance. The moment non-performance will be measured is when the container enters the stack of the destination terminal. This method of registering non-performance is conform to the method used by Tierney *et al.* [52] and Nieuwkoop [41]. Non-performance is the key performance indicator of the ITT system and will show the percentage of containers that has not been delivered in time.

Besides the non-performance, there are a number of other important performance indicators. These include the following:

- Average time containers are being delivered too late.
- Occupation rates of the vehicles - How much of the vehicle capacity is used?
- Number of idle vehicles
- Loading rates of the vehicles
- Occupation rates of the terminal equipment - How much of the equipment capacity is used? Does the terminal have an over- or under-capacity?
- Waiting times at the terminals
- Total distance traveled by the vehicles
- Total distance traveled empty by the vehicles - How much energy is wasted by having to let vehicles drive empty?
- Delays due to traffic

2.6 Summary

This chapter has explained the main input and output for the simulation model. The area to be simulated is the combined Maasvlakte 1 + 2 in the Port of Rotterdam. The area consists of 18 terminals that have to be connected both by a road network and a water network. The model needs to be able to read the transport demand input files created by a transport demand generator. Per container job the following information is provided: release time, origin, destination, the number of TEU and the due time. The model needs to be able to simulate 5 different ITT vehicles: AGVs, ALVs, MTSs, barges and trucks. The barges should be able to operate alongside the road vehicles. The model should be able to calculate a number of performance indicators including: non-performance, the average time containers are being delivered late, occupation rates, waiting times and delays due to traffic. Non-performance is the system's main performance indicator and shows the percentage of containers that has been delivered too late.

The next chapter will discuss several modeling techniques that will have to be used to meet the model requirements.

Chapter 3

Modeling methods

This chapter will discuss modeling methods. A literature survey has been performed in the field of simulation, vehicle scheduling and dispatching, and traffic modeling. Choices will be made based on solutions found in literature.

The main difference with the old ITT simulation models discussed in Section 1.1 will be that the new model will be the first to incorporate traffic modeling. This means that delays occurring due to traffic will have an impact on the performance of the system. This will provide a more realistic representation of the system's performance.

3.1 Simulation

The design of a process often needs to be evaluated for correctness and engineering properties before its implementation. Simulation is a cost-effective mechanism to evaluate system and process design [36]. A simulation model can be either discrete or continuous. The type of simulation that is most appropriate depends on the behavior of the system that needs to be modeled. If the system state instantaneously changes at discrete points in time, like in the ITT system, discrete event simulation should be used.

Discrete event simulation has been around since the late 1950s [39], and has since been used extensively for the modeling of complex logistic and production systems. For instance for analyzing bulk terminal operations[9], for improving warehouse operations [20], and for production scheduling [60]. When looking at discrete simulation applications in ports, most research centers around container terminals. From a modeling point of view, the transfer system of a container terminal is in many ways similar to the ITT system, as it also employs vehicles to transport containers from a certain origin to a certain destination. Vehicles are tasked with driving a container from a certain quay crane at the quay to a certain yard crane at the stack, and vice versa. Sha [50] developed a simulation model for such a system, which is used to determine the productivity and cycle times of the terminal's intra-terminal transport system. A similar model is developed by Duinkerken *et al.* [13]. The model is applied to the ECT Delta terminal in the Port of Rotterdam, and can be used to find the bottlenecks in the system. Carpenter *et al.* [7] developed a discrete simulation model for marine terminals that can provide assistance in the terminal planning process by simulating different various terminal layouts. Yang *et al.* [63] use a simulation model to analyze the performance of different transport systems in an automated container terminal. Gambardella *et al.* [21] [37] present a container terminal simulation model which is used as a decision support tool in the management of a real world intermodal terminal. The model is focussed on resource allocation, and uses operation research techniques in order to generate resource allocation plans.

All discrete event simulations contain an operation routine for the management of the event calendar and simulation clock [44]. The operation routine depends on the world view, and may be based on events, activities, or processes [25]. A suitable world view for modeling complex logistic processes is the process interaction world view, which focuses on the flow of entities through a model. This approach views systems as sets of concurrent, interacting processes. A process class describes the behavior of each class of entities during its lifetime. Zeigler *et al.* [65] describe process interaction simulation as a combined event scheduling-activity scanning procedure. The description of the dynamics of a model element can be implemented as a unit, rather than being separated into a number of unconnected events and activities. Therefore the programming structure maintains a closer relation to the model structure and consequently the real system that is being modeled. Applying the process-interaction approach can be broken down into three steps:

1. Decompose the system into relevant element classes, preferably similar to the real world system's elements.
2. Identify the attributes of each element class.
3. Distinguish the active element classes and define their processes.

Whenever simulation is being used to investigate complex control problems, the majority of time in the modeling phase is spent on programming the algorithms. This time can be cut down by using certain specific software packages. A software package especially developed for discrete event simulation of complex control problems in logistic and production environments is TOMAS: Tool for Object-oriented Modeling And Simulation [57] [59]. TOMAS is implemented as a toolbox in the application-development environment of Delphi (using the Object Pascal language), and is described by means of the process-oriented approach. TOMAS provides several tools that make it easier to construct, analyze and verify logistic simulation models. TOMAS has already been used to model various logistic systems, mainly for automated container terminals [14] [58].

3.2 Simulation model requirements

Because of the discrete nature of the ITT system, the simulation model also needs to be discrete. The discrete event simulation model will be developed using Delphi and TOMAS.

In order to meet the requirements, the model needs to be able to:

- Simulate the complete Maasvlakte infrastructure. Including terminals and a separate road and water network.
- Read the given transport demand input files and create the containers on the list with all their given properties. Each container should be a separate object that can be handled by the ITT system
- Simulate 5 types of vehicles: AGVs, ALVs, MTSs, trucks and barges. The barges should be able to operate alongside the other vehicle types.
- Provide realistic values of the performance indicators mentioned in Section 2.5.
- Create a realistic interaction between the storage process and the inter terminal transport by the means of terminal equipment and quay cranes.
- Integrate traffic modeling. Delays due to traffic will have an impact on the system's performance.

The model will need to simulate the following physical objects: containers, roads, intersections, terminals, terminal equipment, vehicles (AGVs, ALVs, MTSs and trucks), quay cranes and barges. Besides the physical objects, control objects should be added to assist in the decision making processes.

A basic version of a discrete ITT simulation model has previously been developed by the author as part of a research assignment [49]. This model forms the basis of the ITT simulation model developed during this research. The basic model proved to be too simplistic in several areas, which resulted in unrealistic outcomes. In order for the model to provide more realistic outcomes and to be able to evaluate all ITT configurations, it requires multiple improvements and expansions.

The properties of the simulation model will be discussed in detail in Chapter 4.

3.3 Vehicle scheduling and dispatching

The vehicle scheduling system decides when, where, and how a vehicle should act to perform given tasks. If all tasks are known prior to the planning period, the scheduling problem can be solved offline. However in practice, exact information about tasks is usually known at a very late instant, which makes offline scheduling very difficult. Therefore, dynamic (online) scheduling or dispatching systems are needed to control vehicles [4].

3.3.1 Dynamic scheduling vs. dispatching rules

A vehicle dispatching system may be considered as a scheduling system with a zero planning horizon, where a dispatching decision is made when a vehicle finishes its job or when a new job comes available. The system uses dispatching rules to control the vehicles. There are two main types of online dispatching systems: decentralized and centralized systems [4]. Decentralized control systems dispatch vehicles based on local information only. There is no communication between vehicles and the central control system. The main advantage of the decentralized control system is its simplicity, but its efficiency is low. The centralized control system is more complicated but can provide a better performance [54]. In centralized dispatching systems, a central controller keeps track of all information related to the vehicles in the system. The controller assigns loads to vehicles based on prespecified dispatching rules. The dispatching rules can be divided into two categories: handling center-initiated (transport jobs at the handling centers can claim vehicles) and vehicle-initiated (vehicles can claim transport jobs) [17].

Because of their simplicity, vehicle dispatching rules are relatively easy to implement. However, for complex systems like the ITT system, vehicle scheduling would be more efficient [38] [48]. A proper scheduling system would make sure that more containers are delivered in time.

In practice, environments are usually stochastic (job arrival times, travel times, load- and unload times, etc.), so the schedules have to be adapted dynamically in time. The vehicle schedules should be updated when new transportation request information arrives. An approach is to schedule vehicles using a rolling horizon in which vehicle routes are updated after a predetermined time period.

3.3.2 Solution approaches for vehicle scheduling

The solution approaches for vehicle scheduling can be divided into two sections: centralized and distributed approaches. Centralized approaches assume a global view of the problem, and use all available information in finding a solution. [35]. Distributed approaches make use of multi agent technology, where decisions are made based on local information.

3.3.2.1 Centralized approaches

Centralized approaches solve operation research problems either exactly or approximately. These can be split up in exact methods and heuristics, which provide approximate solutions.

Exact solutions

A popular example of an exact solution is the branch and bound technique. Le *et al.* [30] focus on the scheduling of automated lifting vehicles. They look for an optimal solution by combining a DC (Difference of Convex functions) algorithm with the branch and bound method. The branch and bound technique has also been used for other port scheduling operations. Peterkofsky *et al.* [46] use it to find a solution for the static quay crane scheduling problem in a maritime container terminal. They present a branch and bound method which searches for the optimal schedule by minimizing delay costs. A similar method for the quay crane scheduling problem is presented by Kim *et al.* [29]. The branch and bound technique is also used by Ng *et al.* [40] to define an optimal schedule for yard cranes to perform a given set of loading/unloading jobs with different ready times. The objective is to minimize the sum of job waiting times.

The biggest downside of exact solution methods like the branch and bound technique is that computational time increases rapidly when problem sizes increase, which makes it difficult to use in practice [29]. Therefore, a heuristic approach might be a better option.

Heuristics

The objective of a heuristic is to quickly produce a solution that is good enough for solving the problem. They don't always provide the optimal solution, but at least an approximation. Heuristics also include dispatching rules (Section 3.3.1).

Various dispatching heuristics for AGV systems are presented in literature [28] [31], but the downside of this type of heuristic is that they only look at the current state of the system; they don't look ahead. Kim *et al.* [29] propose a look-ahead dispatching heuristic for AGVs in a container terminal. The heuristic is compared to several other dispatching rules that don't look ahead, and is shown to be much more efficient. On average the heuristic deviates 10% from the optimal solution, but is uses only 0.01% of the computational time.

An example of a heuristic procedure is a greedy algorithm. It makes choices based on what seems best at that moment and then solves the subproblems that arise later. Xue *et al.* [62] propose a greedy algorithm and a local search algorithm for the scheduling of yard trucks and quay cranes in a container terminal. Bish *et al.* [5] also propose a greedy algorithm for the dispatching of AGVs in a mega container terminal. The algorithm proves to deliver a near optimal solution, with an average deviation of 1.55% from the optimal solution. This heuristic dispatching strategy is relatively easy to implement and has been used in at least one seaport [8].

The most widely used heuristic algorithms in vehicle routing and scheduling are the genetic algorithms (GA) [35]. Genetic algorithms mimic the process of natural evolution. They start with a population of randomly generated individuals and try to find a solution through an iterative process of mutations, crossovers, inversions and selection operators. Many different strategies can be applied. Gudelj *et al.* [22] use a genetic algorithm for the scheduling of AGVs in a marine container terminal. Their goal is to minimize ship processing time and to minimize the number of AGVs involved, while maintaining the system throughput. The same has been done for straddle carriers by Bse *et al.* [6]. Genetic algorithms have also been used for the scheduling of AGVs in automated warehouse systems [34] and collaborative manufacturing systems [47].

3.3.2.2 Distributed approaches

Distributed control is a way of dividing large control problems into multiple smaller control problems by the means of multi-agent systems (MASs). A MAS can be defined as a loosely coupled network of problem solvers that interact to solve problems that are beyond the individual knowledge of each problem solver [16]. These problem solvers are better known as agents. They are autonomous, which means that they can perform desired tasks without external guidance. In a MAS the separate agents do not possess enough information or capabilities for solving the overall problem. By communicating with each other and performing their own separate tasks they can solve the overall problem together [61]. In a MAS there is no system global control, the data are decentralized and computation is asynchronous [51]. Note that the centralized approaches discussed in the previous section can also be applied within the multi agent framework, for instance in a single agent or as a subsystem.

The main advantages of MASs over centralized systems are:

- Computational efficiency: different computations can be done simultaneously. Although this is not a benefit from a modeling point of view when the simulation model would run on one single computer.
- Reliability: the failure of one agent will not fail the whole system. Other agents will automatically be allocated to perform the failed agent's task.
- Extensibility: the number of agents working on a problem can be altered.
- Responsiveness: anomalies can be handled locally and don't have to be propagated through the whole system.

Henesey *et al.* [24] [23] present a MAS based simulator for evaluating different AGV systems for container terminals. The agents use the Contract Net Protocol to coordinate tasks. This protocol implies that one agent will take the role of manager, which initiates a job to be performed by one or more agents. Agents can bid for the job and the best candidates are selected by the management agent. Ye *et al.* [64] use a MAS as the basis for an intelligent truck dispatching system using the Contract Net Protocol and fuzzy reasoning. The system consists of a Container Truck Scheduling Agent and multiple Container

Truck Agents. Each Container Truck has its own Container Truck agent. Li *et al.* [32] present a system for modeling and simulation of yard trailer dispatching at CTs based on MAS. The modeling is aimed at moderating traffic jams at the quay side and storage yard, and minimizing the total working hours of the QCs. The system is continuously in pursuit of minimizing the completion time for every step in the process, and consequently the handling time of the ships. Zheng *et al.* [66] propose a distributed control model for AGVs in a manufacturing plant. The model is mainly aimed at avoiding collisions and deadlocks.

3.3.3 Discussion

As discussed in the previous section; for complex systems like the ITT system, vehicle scheduling would be more efficient than using dispatching rules. A proper scheduling system would make sure that more containers are delivered in time. Scheduling systems are more and more used for real life transport systems so adding this to the simulation model would likely result in more realistic outcomes for a new future system. However, vehicle scheduling brings along a couple of practical problems:

- Scheduling algorithms require a lot of processing time and power. Even the more efficient heuristic scheduling methods would require a tremendous processing power for a model with so many variables as the Maasvlakte ITT system. Especially since the scheduling system would have to run every so many hours of simulated time in order to update the schedule. This might not be a big problem for a real system where actions happen in real time and an hour is really an hour, but it would be a big problem in the simulation model. Simulations are supposed to be run over a time lapse of weeks, but running such a simulation is not supposed to take weeks. It's supposed to take minutes, or possibly hours.
- Developing a proper scheduling algorithm takes a lot of time. This MSc research is supposed to take about half a year. The development of a scheduling algorithm could be a complete research in itself and would be too extensive to be a small part of this research.
- With the current transport demand input data, container jobs are not known to the system before they enter the system. When they enter the system they are immediately allowed to be transported because there is no penalty for being delivered early, only for being delivered late. In other words: with the current input data there is no possible planning horizon.

For these reasons there is no other way than to implement a dispatching system and not a scheduling system. A dispatching system can be implemented in a lot of different ways. Choices need to be made on how to regulate various problems, including the following:

- Deciding if a container should be transported by barge or by road.
- Requesting empty vehicles from another terminal to pick up a container. When to request an empty vehicle and where to get it?
- How to operate the barges.
- Deciding when an MTS trailer is allowed to be transported if it's not yet full.

The manners in which these problems have been solved are explained throughout Chapter 4. The dispatching system is handling center initiated, so the terminals can claim vehicles to transport a certain container. The system is somewhere between a centralized and decentralized dispatching system. If possible, the problem of finding a vehicle to transport a container is solved locally. If there are no vehicle available locally they are searched for globally.

3.4 Traffic modeling

In order to acquire more realistic outcomes, traffic delays have to be taken into account. Intersections will have to be modeled in such a way that vehicles are delayed when an intersection is too crowded.

3.4.1 Solutions approaches for traffic modeling

Van Burgsteden *et al.* [53] make a distinction between two types of interactions in traffic that can cause delays: lateral interactions and longitudinal interactions. Delays through longitudinal interactions occur when one vehicle drives slower than the others, so the others have to adjust their speed to the first one. When assuming that all vehicles drive the same speed, this type of delay won't occur. Most delays are caused by lateral interactions, which occur at intersections. These delays can be divided in three categories:

- Solving of conflicts: these occur at intersections where there is no signaling. Depending on the layout of the intersection and the traffic regulations, vehicles can pass the intersection by priority.
- Waiting for traffic control signals: including waiting for a red light.
- Queuing: when multiple vehicles approach an intersection, and the first vehicle has to wait. All other vehicles have to queue behind it.

In a discrete event simulation the central area of a conflict point can be seen as a server with several queues. The decision which queue to pick is dependant on the priority rules adopted at that point. Process time is dependant on the time the vehicle needs to clear the area. At a conflict point, the priority of every connecting direction over every other connecting direction (see Figure 3.1) needs to be defined. At an intersection with traffic control, every direction is given a time slice in which vehicles can pass freely.

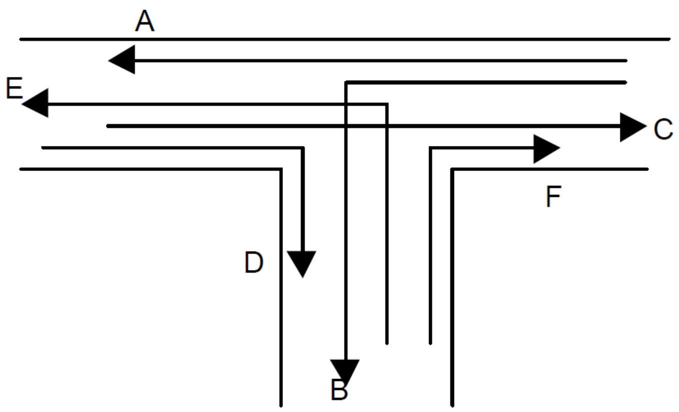

Figure 3.1: Directions at a conflict point [53]

The AGV system for container terminals proposed by Liu *et al.* [33] uses a control logic to guarantee the smooth traffic flow within the yard. When longitudinal conflicts arise due to different AGV traveling speeds (e.g. loaded and empty speeds), the control logic enforces “Low Speed Zones”. The speeds of all AGVs in that zone are then set to the speed of the slowest AGV. When lateral conflicts arise at intersections, the control logic applies a “Modified First Come First Pass” concept. This concept consists of a priority algorithm that determines which vehicle is allowed to go first. Once a vehicle has entered an intersection, the other vehicles have to wait until the area is cleared.

Egbelu *et al.* [18] present a traffic flow model for AGV based systems. Vehicles travel over a network of nodes and arcs. It is assumed that all vehicles travel at the same speed, so from a discrete simulation point of view the movement of a vehicle from a source node to a destination node can be modeled as a series of discrete jumps from one node to another. The jump time is a function of the traveling speed of the vehicle and the distance between the adjacent nodes. Only static routing is considered, so the path between two nodes is predefined and is not altered while traveling. Intersections are modeled as such that every incoming direction has a buffer where vehicles can wait. It is assumed that these buffers have an infinite capacity. The traffic flow model employs a conflict resolution algorithm which determines which vehicle is allowed to cross the intersection first.

Evers *et al.* [19] introduce the concept of semaphore as an abstraction of a traffic light which controls the admissions of approaching vehicles individually. The semaphore concept is borrowed from computer science [11]. It makes it possible for a vehicle to claim a certain node for a period of time. Figure 3.2

shows a four way intersection consisting of a node N which is controlled by semaphore S. When giving S a maximum capacity of 1 this means that it can be claimed by one vehicle at a time. When a vehicle leaves N, the claim is released and another vehicle is allowed to enter. When there are two or more vehicles blocked on S, one of them has to be selected, so an access protocol will have to be introduced. Various rules may be taken into consideration, including: First-In-First-Out, priority to a vehicle in the same direction as the predecessor, and priority to the vehicle with the earliest due time. The semaphore concept was used by Duinkerken *et al.* [13] to develop a control system to coordinate the traffic flows of AGVs. The control system is called TRACES, which stands for Traffic Control Engineering System. The system has successfully been implemented at the ECT Delta terminal at the Maasvlakte in the port of Rotterdam.

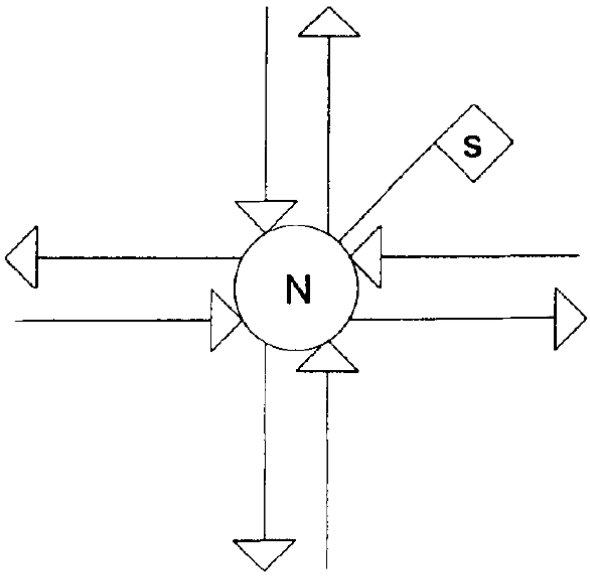

Figure 3.2: Elementary node N controlled by semaphore S [19]

3.4.2 Discussion

In the ITT simulation model it will be assumed that all vehicles travel at the same average speed. Therefore delays will not occur at the straights, but only at the intersections. Regarding the intersection delays, all solutions found in literature share a similar basis. There is one central point, a server, and a number of buffers or queues for every direction entering the intersection. Vehicles can claim the intersection for a certain time, or the intersection can let the vehicle claim it.

Different algorithms can be used to determine which vehicle is allowed to travel the intersection first. Two different algorithms will be implemented in the ITT simulation model:

- A simple First-In-First-Out algorithm. The first vehicle arriving at the intersection will be allowed to travel it first. When the vehicle has cleared the intersection the next vehicles is allowed to go. Only one vehicle can cross the intersection at the same time.

- A more advanced priority algorithm. This algorithm will analyze all containers on the vehicles present at the intersection and select the one with the highest priority. The algorithm will also consider wether vehicles are going in the same direction or wether they are able to cross at the same time without conflicts. This method should decrease delays and increase throughput.

The manner in which the intersections are modeled is explained in Section 4.10.

3.5 Summary

This chapter has discussed various modeling methods. Because of the discrete nature of the ITT system, the simulation model also needs to be discrete. The discrete event simulation model will be developed using Delphi and TOMAS.

Literature research has shown that for complex systems like the ITT system, vehicle scheduling would be more efficient than using dispatching rules. However, scheduling algorithms would require too much processing power and time, they would require to much time to implement, and with the current input data there is no possible planning horizon. Therefore a dispatching system will be implemented in the simulation model. Dispatching rules will be used to regulate problems such as: requesting empty vehicles, deciding if a container needs to be transported by barge or road, and how to operate the barges.

The ITT simulation model will incorporate a traffic modeling system. This means that delays occurring due to traffic will have an impact on the performance of the system. Vehicles will only experience delays at intersections. 2 Different algorithm will be able to be used to determine which vehicle is allowed to cross an intersection first.

The next chapter will explain how the simulation model works, including the modeling methods selected in this chapter.

Chapter 4

Simulation model

This chapter will explain how the ITT simulation model works. First the overall system will be discussed, followed by a detailed explanation of every modeled object. All modeled objects will be discussed by the following points:

- Attributes — A description of the object's attributes.
- Process description — Description of the process for every active component.
- Interaction with other objects — Short description of the interaction with other objects.

4.1 Model design

A simulation model for the ITT system has been created using Delphi and the simulation tools provided by TOMAS [57]. The simulation model is object-oriented. It consists of the following objects: Containers, a Generator, an UrgencyCheck, Roads, Intersections, Terminals, Terminal Controls, Nodes, Terminal Equipment, Vehicles, Quay Cranes and Barges. The Containers, Roads and Nodes do not have a process and are therefore passive. All other objects are active. Short descriptions of the functions of the different object are given in Table 4.1.

4.1.1 Model input and output

The input and output of the simulation model has been schematically shown in Figure 4.1. The model input consists of a general configuration file and a number of input files. In these input files the infrastructure, the equipment and vehicle properties, and the transport demand can be defined. The input files are shown if Appendix B. The model output consists of a number of output files and graphs. The output files are shown if Appendix C. The graphs are created using the Tomas Collections form which is part of TOMAS [57]. The graphs can be used to monitor various output values over time. The graphs are automatically stored to csv files which can be imported in spreadsheet software like Excel for further analysis. All simulations are performed within one simulation model.

4.1.2 Model schematics

Figure 4.2 shows a schematic representation of the physical objects in the model. A Terminal consists of a number of Terminal Equipment and a Container stack. Vehicles drive between the Terminals, where they are loaded or unloaded. The Vehicles drive over a network of Roads and Intersections to reach their destination. The Barges use a separate network of waterways which is connected to all Terminals with waterside operations.

Name	Type	State	Function description
Container	Physical	Passive	Object that is transported by the ITT system.
Generator	Non-physical	Active	Creates the Containers and places them at their origin Terminal.
Urgency Check	Control	Active	Checks Containers for urgency.
Vehicle	Physical	Active	Transports Containers by road; Truck, AGV, ALV or MTS.
Terminal	Physical	Active	Origins and destinations for the Containers.
Terminal Control	Control	Active	Every Terminal has 1 Terminal Control coupled to it. Requests empty Vehicles from other Terminals.
Terminal Equipment	Physical	Active	(Un)loads Containers at the Terminals for transport by road.
Road	Physical	Passive	Vehicles drive the Roads in order to reach their destination. Used for water and road network.
Intersection	Physical	Active	Vehicles cross the Intersections in order to reach their destination. Used for water and road network.
Node	Non-physical	Passive	Object made for every Terminal and Intersection. Used for path planning.
Barge	Physical	Active	Transports Containers by water.
Quay Crane	Physical	Active	(Un)loads Containers at the Terminals for transport by water.

Table 4.1: Modeled objects

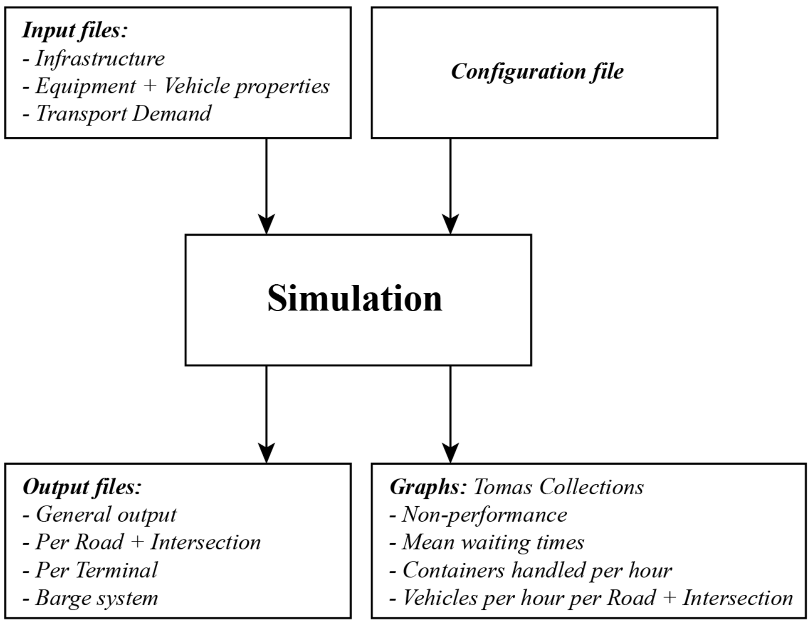

Figure 4.1: Model input and output

4.1.3 General assumptions

A number of assumptions have been made. These include the following:

- All equipment is able to lift 2 TEU at the same time. This can be 1 2-TEU container or 2 1-TEU containers if 2 of those containers are available.

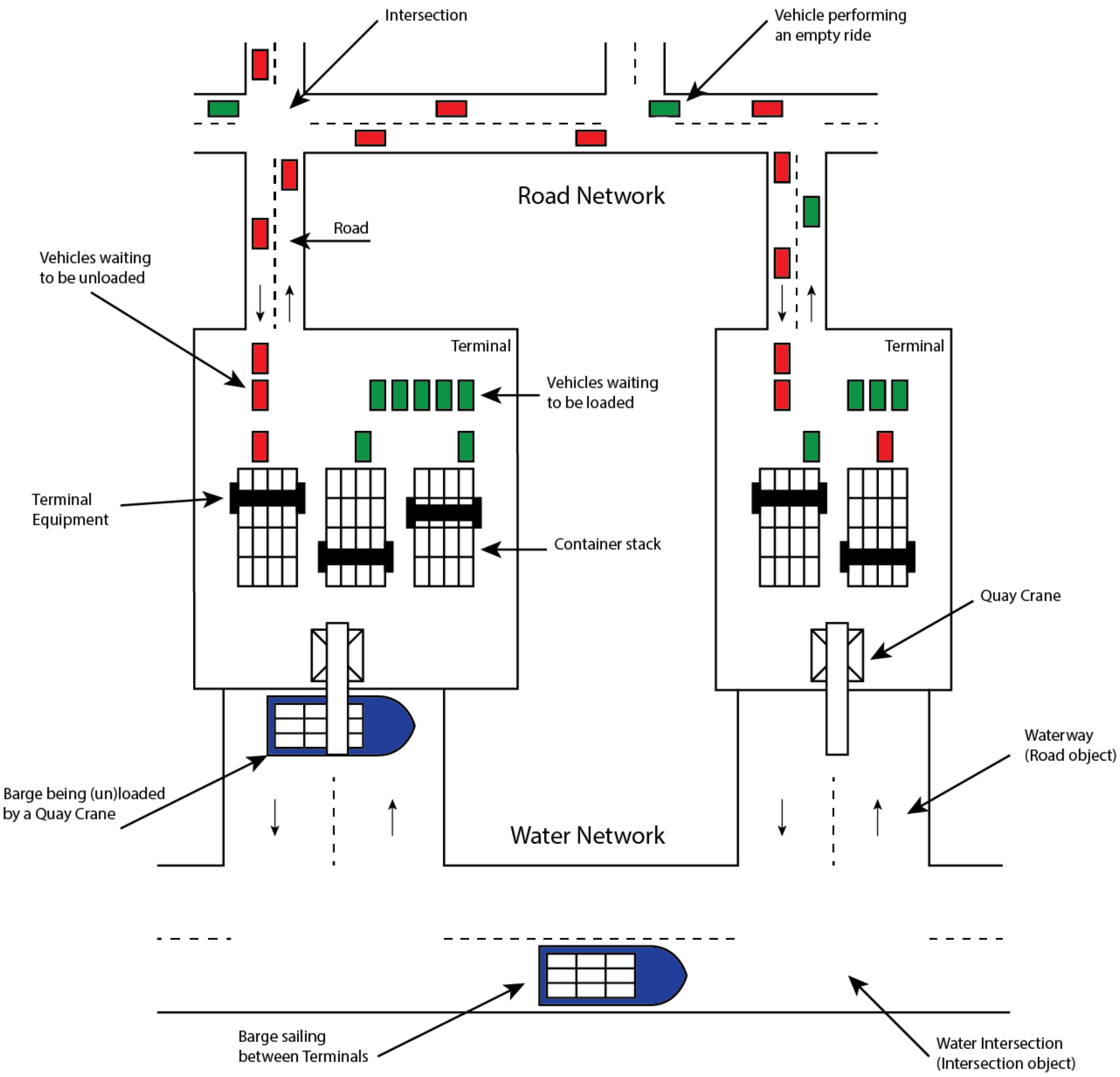

Figure 4.2: Schematic representation of the modeled objects

- All vehicles travel at the same speed. Therefore delays will only occur at intersections and not on the roads.
- Vehicles cannot overtake other vehicles.
- Every road or waterway has one lane.
- Vehicles don't have to stop for gas or a battery change.
- All vehicles and equipment work 24 hours a day, 7 days a week.
- Vehicles transport their entire load from one origin to one destination. They don't stop at different terminals along the way. Only the barge does.

4.1.4 Communications

The most important communications in the model are shown in Figure 4.3. These and other communications between objects will be discussed in more detail in the following sections.

When a Terminal has a Vehicle shortage, its Terminal Control searches for the Terminal with the most idle

Vehicles and then orders one of the Vehicles there to perform an empty ride. The Vehicles (in the AGV and truck scenarios) are loaded and unloaded by the Terminal Equipment. After receiving a Container, the Vehicles read their destination from the Container and then use the Nodes to find the shortest route. While driving the route, they inform each Road when and for how long they are driving there. When a Vehicle enters an Intersection is has to wait until the Intersection's traffic modeling system allows it to cross the Intersection.

The Barges use a separate water network, which is also built up out of Road and Intersection objects. The Barges are (un)loaded and activated by the Quay Cranes. The Barges don't read their next destination from a Container, but from a list of route points that can be set before the simulation begins (see Appendix B).

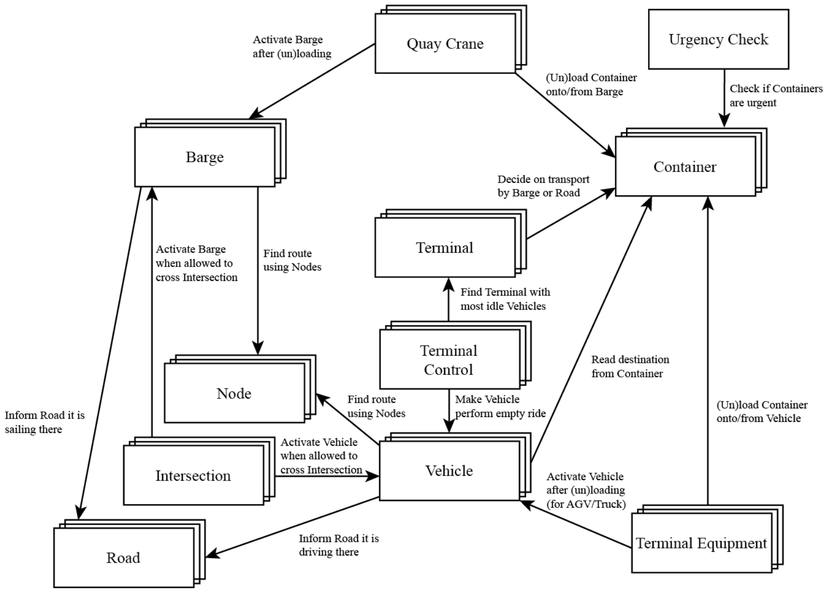

Figure 4.3: Most important communications in the simulation model

4.1.5 Maasvlakte infrastructure

In the model, the ITT infrastructure on the Maasvlakte consists of two separate networks: a Road network and a Barge network. Both networks consist of a number of Nodes (Intersection and Terminals) with arcs (Roads) between them. The Barges are only able to travel the Barge Network and the road Vehicles can only travel the Road network. The maps of these two networks are shown in Figure 4.4 and 4.5. Bigger versions of the networks are shown in Appendix D.

The networks can easily be altered by changing the input files for the Terminals, Roads and Intersections. These input files are shown in Appendix B.

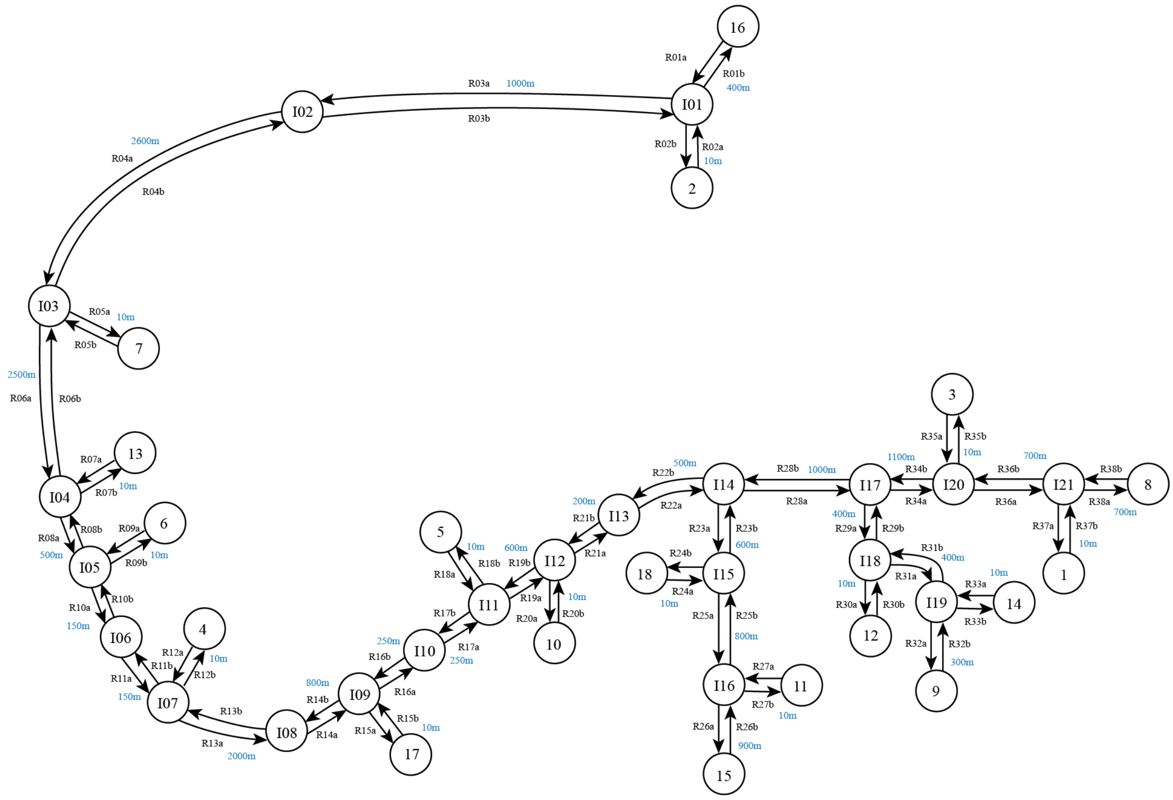

Figure 4.4: ITT Maasvlakte Road network

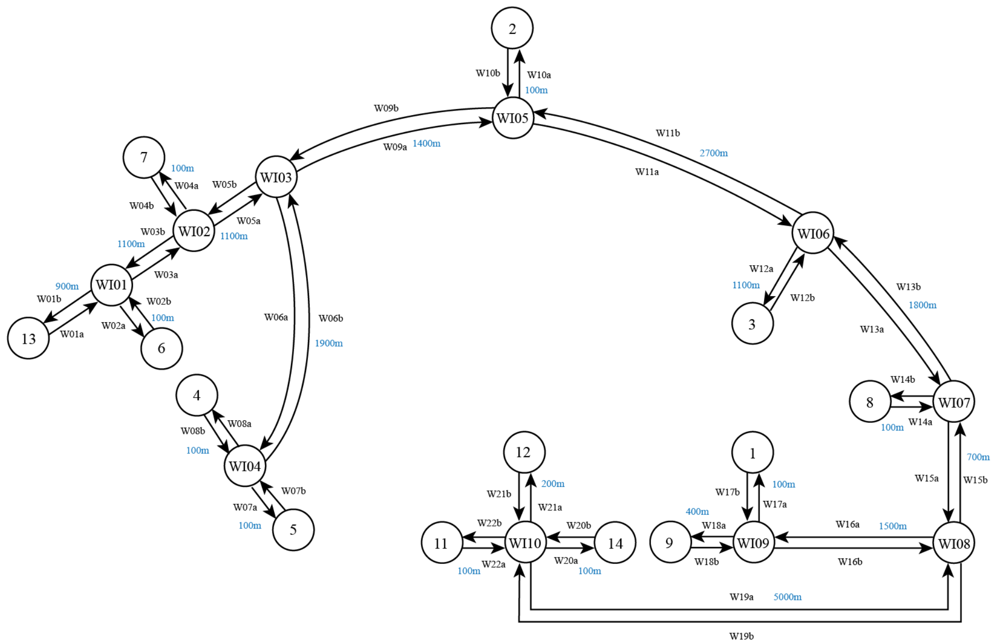

Figure 4.5: ITT Maasvlakte Barge network

4.2 Container

Containers are the objects that have to be transported by the ITT. They are created and placed at certain Terminals at certain times by the Generator (see Section 4.3). The Inter Terminal Transport is simulated at Container level, so for every physical Container a Container object is created in the simulation. Each Container holds information that is essential for its transport. After it has been delivered to its destination and the non-performance has been registered, the Container object is destroyed.

4.2.1 Attributes

• Origin	Terminal to be transported from
• Destination	Terminal to be transported to
• TEU	Type of Container; 1 or 2 TEU
• ReleaseTime	Time at which it is created
• DueTime	Latest allowed arrival time at destination
• PriorityTime	Calculated by Intersection to determine priority
• TravelDistance	Distance from origin to destination
• ExpectedHandlingTime	Expected time from stack at origin to stack at destination via road
• ExpectedBargeHandlingTime	Expected time from stack at origin to stack at destination via Barge
• HandlingTimeLeft	Expected handling time left to transport Container
• FinalStartTime	Container must leave before this time to be delivered on time (see Formula 4.1)
• MyVehicle	Vehicle it is transported by
• Urgent	Whether the Container is urgent or not
• RegisterNonPerformance	Method: to register non-performance

The *HandlingTimeLeft* value is updated while the Container is being transported by a Vehicle. This in order to determine the priority of the Container. The *ExpectedHandlingTime* is only calculated when the Container enters the system (see Section sec:Generator). Before the Container is handled by the system: $HandlingTimeLeft = ExpectedHandlingTime$.

$$FinalStartTime = DueTime - ExpectedHandlingTime \tag{4.1}$$

4.2.2 Register non-performance method

The Container object does not have a process, but it has a method which can be called for by other objects; *RegisterNonPerformance*.

The method *RegisterNonPerformance* is used to register if the Container has been delivered before its *DueTime*. More information on non-performance can be found in Section 2.5.

The method is called for by the Terminal Equipment or Quay Crane just before the Container is destroyed (see Section 4.8 and 4.13). The process goes as follows:

- if $TNow > DueTime$
  - Register this Container as non-performance

4.2.3 Interaction with other objects

Generator

Containers are generated and placed at their origin Terminal by the Generator.

Terminal

Containers wait in the container queues of their origin Terminal until they are handled by the Terminal Equipment or Quay Crane located at the Terminal.

Terminal Equipment

Containers are loaded from the Terminal's container queue onto a Vehicle by the Terminal Equipment. The Terminal Equipment is responsible for destroying the Container objects after unloading at the destination Terminal.

Quay Crane

Containers are loaded from the Terminal's container queue onto a Barge by the Quay Cranes. The Quay Cranes are responsible for destroying the Container objects after unloading at the destination Terminal.

Vehicle

The Vehicles transport the Containers over the road network to their Destination Terminal. They read information from the Container in order to determine where they need to go.

Barge

The Barges transport the Containers over the water network to their destination Terminal.

Intersection

The Intersections read information from the Containers in order to determine which Vehicle has the highest priority.

Terminal Control

The Terminal Control analyzes the Containers in the container queues at the Terminals to decide on requesting empty rides from other Terminals.

4.3 Generator

The Generator is responsible for creating the Containers and placing them in the *MyNewContainerQ* of their origin Terminal. The Generator reads when the Containers should be generated and what their properties should be from the transport demand input file: *inputTransportDemand.txt* (see Appendix B).

4.3.1 Attributes

• CalculateExpectedHandlingTime	Method: to calculate the Container's expected handling time
• FindRoute	Method: for calculating expected handling time of new containers
• PROCESS	Method: describes activities as a function of time

4.3.2 Process description

The Generator has a process that is repeated until the end of the set simulation time. Afterwards it is responsible for completing the simulation. The process goes as follows:

- Repeat until end of simulation time
  - Read ReleaseTime for next Container
  - Wait until ReleaseTime
  - Create a new Container
  - Read Container's properties from input file and assign them to the Container
  - *CalculateExpectedHandlingTime*
  - Add Container to its origin Terminal's MyNewContainerQ
  - Add Container to AllContainerQ sorted by DueTime
- Interrupt the simulation

4.3.2.1 CalculateExpectedHandlingTime method

The method *CalculateExpectedHandlingTime* is used to calculate the generated Container's expected handling time in case of no delays, when transported by road. This value is used by the Terminals to determine which Container should be transported first. The method calls for the *FindRoute* method to find the shortest route to the Container's destination and then uses Formulas 4.2, 4.3 and 4.4 to calculate the expected handling time. The *FindRoute* method is a simplification of the Vehicle's *FindShortestRoute* method, which will be discussed in Section 4.5.

$$RoadTime = \frac{TravelDistance}{VehicleSpeed * 3600} \tag{4.2}$$

$$IntersectionTime = \sum_{i=1}^{N} TimeToCross_i \tag{4.3}$$

$$\begin{aligned} ExpectedHandlingTime = OriginHandlingTime + RoadTime + IntersectionTime \\ + DestinationHandlingTime \end{aligned} \tag{4.4}$$

Where:

$RoadTime$	=	Time needed to drive the Roads [hours]
$VehicleSpeed$	=	Average speed of the used Vehicle in [m/s]
$IntersectionTime$	=	Time needed to cross the Intersections [hours]
N	=	Total number of Intersections to be crossed
$OriginHandlingTime$	=	Expected handling time of the Equipment at the Origin Terminal [hours]
$DestinationHandlingTime$	=	Expected handling time of the Equipment at the Destination Terminal [hours]

In case of the ALV scenario (Section 4.5.2.5) the lift times of the ALV have to be included too, so for this scenario Formula 4.4 becomes:

$$\begin{aligned} ExpectedHandlingTime = OriginHandlingTime + LiftLoadTime + RoadTime \\ + IntersectionTime + LiftUnloadTime + DestinationHandlingTime \end{aligned} \tag{4.5}$$

4.3.3 Interaction with other objects

Container

The Generator creates the Containers, gives them their properties, and places them at their origin Terminal.

Terminal

The Generator places the Containers in the Terminal's MyNewContainerQ.

4.4 Urgency Check

The UrgencyCheck's only function is to check whether a Container needs to be made urgent.

4.4.1 Attributes

- PROCESS Method: describes activities as a function of time

4.4.2 Process description

The UrgencyCheck's process checks all Containers in the system for urgency on a set interval between checks. The formula to decide wether a Container needs to be made urgent is shown in the process description below.

Repeat:

- For each Container in AllContainerQ
  - if $HandlingTimeLeft * UrgencyFactor > DueTime - TNow$
    - Make Container Urgent
- Wait for UrgencyCheckInterval time

When a Container is made Urgent it receives special priority at the Intersections. How soon it is made urgent is dependent on the *UrgencyFactor* which is set in the config file (see Appendix B).

4.4.3 Interaction with other objects

Container

The Control checks the Containers and makes them urgent if necessary.

4.5 Vehicle

Four different types of vehicles are modeled within the Vehicle object: AGVs, Trucks, ALVs and MTSs. All different Vehicle processes are built in one large process for the object Vehicle. The part of the process that is used depends on the value of variable *VehicleType* which can be defined in the config file (see Appendix B). The Vehicle has three methods that can be called for by each Vehicle type: *FindShortestRoute*, *DriveShortestRoute*, and *DoEmptyRide*.

4.5.1 Attributes

• Speed	Average Vehicle speed [m/s]
• Loaded	True if Terminal Equipment finished loading it
• Unloading	True if Vehicle is being unloaded
• Capacity	Vehicle capacity in number of TEU
• TEUonBoard	Number of TEU currently carried by Vehicle
• MyRouteQ	Queue containing all Nodes on route
• MyContainer	First Container in MyContainerQ
• MyContainerQ	Queue containing all Containers onboard
• MyOrigin	Terminal it drives from
• MyDestination	Terminal it needs to drive to
• RouteDistance	Total distance of the current route [m]
• LiftLoadTime	Time it takes the Lift AGV to load a Container [hours]
• LiftUnloadTime	Time it takes the Lift AGV to unload a Container [hours]
• EmptyRide	True if performing an empty ride
• ClearTimeFactor	Used to determine how long it takes a Vehicle to clear an Intersection
• ClearTime	Time it takes to cross the Intersection; calculated by Intersection [hours]
• MyRoad	Road it's driving on
• MyIntersection	Intersection it's on
• WaitAtIntersection	True if waiting at an Intersection
• MyIntersectionQ	Used to determine on which side to enter an Intersection
• MyIntersectionExit	Used to determine on which side to exit an Intersection
• CouplingTime	Time it takes an MTS to (un)couple a trailer
• UnCouple	True when an MTS is allowed to uncouple its trailer
• FindShortestRoute	Method: for finding the shortest route, using the Dijkstra algorithm
• DriveShortestRoute	Method: for driving the shortest route
• DoEmptyRide	Method: for performing an empty ride
• PROCESS	Method: describes activities as a function of time

4.5.2 Process description

4.5.2.1 FindShortestRoute method

The method *FindShortestRoute* uses the Dijkstra path planning algorithm to find the shortest path from its origin to its destination. The result is a set of route points (Nodes) for the shortest route in MyRouteQ. The process goes as follows:

- Copy all Nodes to DijkstraQ
- For all Nodes
  - Distance = Infinity
  - PreviousNode = NIL
- For Origin Node: Distance = 0
- Put Origin Node at front of DijkstraQ
- While DijkstraQ is not empty
  - Select first Node from DijkstraQ
  - if this Node is the destination Node: the shortest route has been found
    - Put all Nodes in the shortest route in MyRouteQ using the PreviousNode attribute
    - Exit the process
  - Remove selected Node from DijkstraQ
  - Find all Roads with selected Node as Start Node and their End Node still in DijkstraQ

- for all found Roads
    - TotalDistance = selected Node's Distance + the length of the Road
    - if TotalDistance < Distance of the Road's EndNode
        - Set Distance of the Road's EndNode to TotalDistance
        - Set PreviousNode of the Road's EndNode to the selected Node
    - Sort the Road's EndNode in DijkstraQ according to Distance, shortest distance first

4.5.2.2 DriveShortestRoute method

The method *DriveShortestRoute* is used to virtually drive the shortest route found by the *FindShortestRoute* method. The process assumes that it always has to drive a Road first, then an Intersection, then a Road, etc. The term *Find Road* in the process description means that it searches for a Road that has the first Node in MyRouteQ as StartNode and the second Node in MyRouteQ as EndNode. The process goes as follows:

- Repeat
    - Find Road between the first two Nodes in MyRouteQ
    - Enter the Road's MyTrafficQ
    - Wait for $\frac{Roadlength}{Speed*3600}$ [hours]
    - Update HandlingTimeLeft for Containers on board
    - Leave the Road's MyTrafficQ
    - Remove first Node from MyRouteQ
    - If length of MyRouteQ ≤ 1: destination has been reached
        - Clear the RouteQ
        - Exit the process
    - Select first Node from MyRouteQ
    - Select Intersection with the same Name as the selected Node
    - Determine where the Vehicle has to enter and exit the Intersection
    - Enter the Intersection's entry side traffic queue
    - Wait until activated by the Intersection
    - Wait for ClearTime, which is calculated by the Intersection
    - Update HandlingTimeLeft for Containers on board

4.5.2.3 DoEmptyRide method

The procedure of doing an empty ride is the same for all Vehicle types. Therefore, a separate *DoEmptyRide* method was created that can be called for by the Vehicle processes when an empty ride is requested by a Terminal Control. The process goes as follows:

- Enter destination Terminal's MyTravelingVehicleQ
- *FindShortestRoute*
- *DriveShortestRoute*
- Set MyOrigin to current MyDestination
- Leave destination Terminal's MyTravelingVehicleQ

- Enter MyIdleVehicleQ at MyDestination Terminal

4.5.2.4 AGV and Truck process

The AGV and Truck both have the same process, because modeling wise they have the same interactions. In the model it doesn't matter if a Vehicle is manned or autonomous. The process can be activated in two ways: by Terminal Equipment after being loaded with a Container, and by a Terminal Control for performing an empty ride. An AGV or Truck is able to carry 2 TEU. It is loaded and unloaded by Terminal Equipment. The process goes as follows:

Repeat:

- Wait while not activated by Terminal Equipment or for empty ride
- if activated for empty ride
  - *DoEmptyRide*
- else
  - Read destination from first Container in MyContainerQ
  - Enter destination Terminal's MyTravelingVehicleQ
  - *FindShortestRoute*
  - *DriveShortestRoute*
  - Set MyOrigin to current MyDestination
  - Leave destination Terminal's MyTravelingVehicleQ
  - Enter MyLoadedVehicleQ at MyDestination Terminal

4.5.2.5 ALV process

The ALV process can be activated in two ways: it can activate itself when there is a new Container to be transported available in the MyLoadPlatformQ of the Terminal it is situated at, and it can be activated by a Terminal Control for performing an empty ride. The ALV is able to load and unload containers from the platform queues at the Terminals and does not have to wait for the Terminal Equipment to load or unload a Container from or to the Vehicle. The ALV is able to carry 2 TEU. The process goes as follows:

Repeat:

- Wait while no Containers on MyOrigin's MyLoadPlatformQ or activated for empty ride
- if activated for empty ride
  - *DoEmptyRide*
- else
  - Leave MyOrigin's MyIdleVehicleQ
  - Select first Container from MyOrigin's MyLoadPlatformQ
  - Remove Container from MyOrigin's MyLoadPlatformQ
  - Put Container in MyContainerQ
  - Read destination from first Container in MyContainerQ
  - Enter destination Terminal's MyTravelingVehicleQ
  - If Container is 1 TEU
    - Search for 1 TEU Container with same destination in MyLoadPlatformQ

    - If such a Container is found then also put this Container in MyContainerQ
  - Wait for LiftLoadTime
  - *FindShortestRoute*
  - *DriveShortestRoute*
  - Set MyOrigin to current MyDestination
  - Leave destination Terminal's MyTravelingVehicleQ
  - Enter MyLoadedVehicleQ at MyDestination Terminal
  - Wait while MyDestination's MyUnloadPlatformQ is full
  - Leave MyLoadedVehicleQ at MyDestination Terminal
  - Enter MyUnloadingVehicleQ at MyDestination Terminal
  - Reserve a spot in MyDestination's MyUnloadPlatformQ
  - Wait for LiftUnloadTime
  - Leave MyUnloadingVehicleQ at MyDestination Terminal
  - Put Container in MyDestination's MyUnloadPlatformQ
  - Enter MyIdleVehicleQ at MyDestination Terminal

4.5.2.6 MTS process

The MTS can be activated by the Terminal Control at its Terminal for coupling or uncoupling a trailer, or by a Terminal Control from another Terminal for performing an empty ride. It is assumed that there is an unlimited amount of trailers available which can be coupled by the tractor part of the MTS. This is explained in more detail in Section 4.7. The MTS is able to carry 10 TEU, but this capacity can be changed by altering the vehicle properties in the Vehicle input file (see Appendix B). The process goes as follows:

Repeat:

- Wait while not activated by Terminal Control or for empty ride
- if activated for empty ride
  - *DoEmptyRide*
- else
  - Wait for CouplingTime: couple a trailer
  - Read destination from first Container in MyContainerQ
  - Enter destination Terminal's MyTravelingVehicleQ
  - *FindShortestRoute*
  - *DriveShortestRoute*
  - Set MyOrigin to current MyDestination
  - Leave destination Terminal's MyTravelingVehicleQ
  - Enter MyLoadedVehicleQ at MyDestination Terminal
  - Wait until activated by Terminal Control
  - Enter MyUnloadingVehicleQ at MyDestination Terminal
  - Wait for CouplingTime: uncouple a trailer
  - Leave MyUnloadingVehicleQ at MyDestination Terminal

- Enter MyIdleVehicleQ at MyDestination Terminal

4.5.3 Interaction with other objects

Container

Containers are transported by the Vehicles

Terminal

Vehicles drive to and from Terminals. Vehicles can reside in the Terminal's vehicle queues.

Terminal Control

Vehicles can be activated by the Terminal Control.

Terminal Equipment

Containers are transferred to and from the Vehicles by the Terminal Equipmentin the AGV and Truck scenarios.

Road

Vehicles need to drive the Roads on their route by entering and leaving their MyTrafficQ in order to reach their destination.

Intersection

Vehicles need to cross the Intersections on their route by entering and leaving their traffic queues in order to reach their destination.

Node

Vehicles use Nodes in order to find and drive the shortest route between their origin and destination.

4.6 Terminal

All Origins and Destinations are modeled as Terminals. Each Terminal consists of a number of queues and a certain amount of Terminal Equipment and Quay Cranes that are tasked with loading and unloading the Containers. The layout for the AGV and Truck scenarios is show in Figure 4.6.

The layout of the Terminal differs a bit per Vehicle type, but the main part is the same for every scenario. New Containers are placed in the MyNewContainerQ by the Generator. The method *SelectModality* is used to determine wether a Container needs to be transported by Road (to MyContainerQ) or by Barge (to MyBargeContainerQ). Containers are added there sorted by a value that is equal to its $DueTime - ExpectedHandlingTime$. The Container with the lowest value will therefore always be at the front of the queue and will thereby be the first Container that is handled by the Terminal Equipment or Quay Cranes. Barges arrive in the MyBargeQ and are handled there by the available Quay Cranes. Only one Quay Crane is able to work on one Barge at the same time.

In the AGV and Truck scenarios loaded vehicles arrive in the MyLoadedVehicleQ. They have to wait for Terminal Equipment to unload their Container(s). Afterwards they move to the MyIdleVehicleQ and wait until they are selected by a piece of Terminal Equipment to be loaded.

In the ALV scenarios, Containers are not directly loaded and unloaded onto and from the Vehicles. Instead, the Terminal Equipment place the Containers in MyLoadPlatformQ. The ALVs waiting in MyIdleVehicleQ can then lift a Container from the MyLoadplatformQ and start driving. When a full ALV arrives at a Terminal it places its Container(s) in MyUnloadPlatformQ and the Terminal Equipment can unload it from there. The platform queues have a limited capacity. The layout for the ALV scenarios is show in Figure 4.7.

The Terminal layout for the MTS scenarios is similar to the ALV scenarios, only in this case the platform queues are replaced with the MyMTSLoadContainerQ and the MyMTSUnloadContainerQ. These queues simulate an unlimited capacity of MTS trailers. New Containers in MyContainerQ are immediately

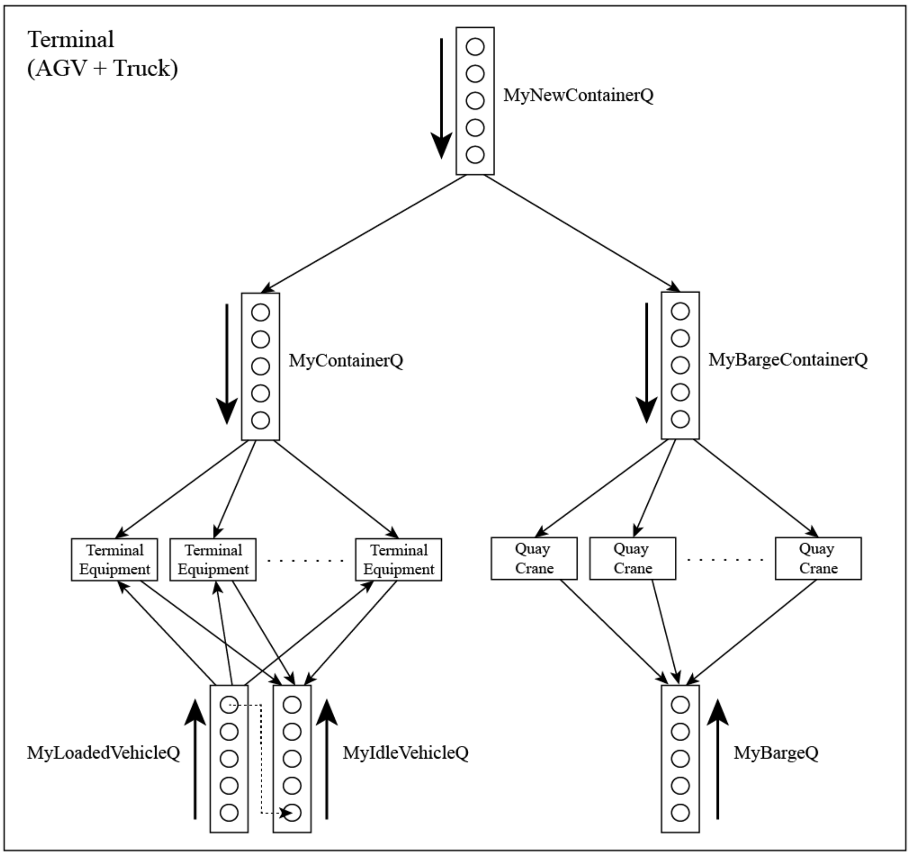

Figure 4.6: Terminal layout for Truck and AGV scenarios

transferred to MyMTSLoadContainerQ by the Terminal Equipment. The Terminal Control (see Section 4.7) monitors this queue and as soon as a full trailer of Containers for one destination is available, or when one of the Containers of a not yet full trailer comes close to its due time, it assigns the tractor part of the MTS to couple to this trailer and drive it to its destination. The trailers are not actually modeled, but there are seen as packages of Containers of maximum the MTS capacity. When a loaded MTS arrives at its destination terminal it drops of its package of Containers in the MyMTSUnloadContainerQ, uncouples, and enters the MyIdleVehicleQ. The layout for the MTS scenarios is show in Figure 4.8.

Modeling an unlimited amount of MTS trailers might result in a slightly better performance for the MTS scenarios. However, with a limited amount of trailers and a proper scheduling system results should be roughly the same.

4.6.1 Attributes

- MyEquipmentQ — Queue with all Terminal Equipment located at the Terminal
- MyEquipmentIdleQ — Queue with idle Equipment located at the Terminal
- MyQuayCraneQ — Queue with Quay Cranes located at the Terminal
- MyQuayCraneIdleQ — Queue with idle Quay Cranes located at the Terminal
- MyNewContainerQ — Queue with Containers placed at the Terminal by the Generator
- MyBargeContainerQ — Queue with Containers to be Transported by Barge
- MyContainerQ — Queue with Containers to be Transported by road
- MyIdleVehicleQ — Queue with idle Vehicles, ready to be loaded
- MyLoadedVehiclesQ — Queue with loaded Vehicles, ready to be unloaded
- MyTravelingVehicleQ — Queue with Vehicles traveling towards the Terminal
- MyUnloadPlatformQ — Queue with containers to be unloaded by Equipment (only for ALV)
- MyLoadPlatformQ — Queue with containers to be loaded by ALV (only for ALV)

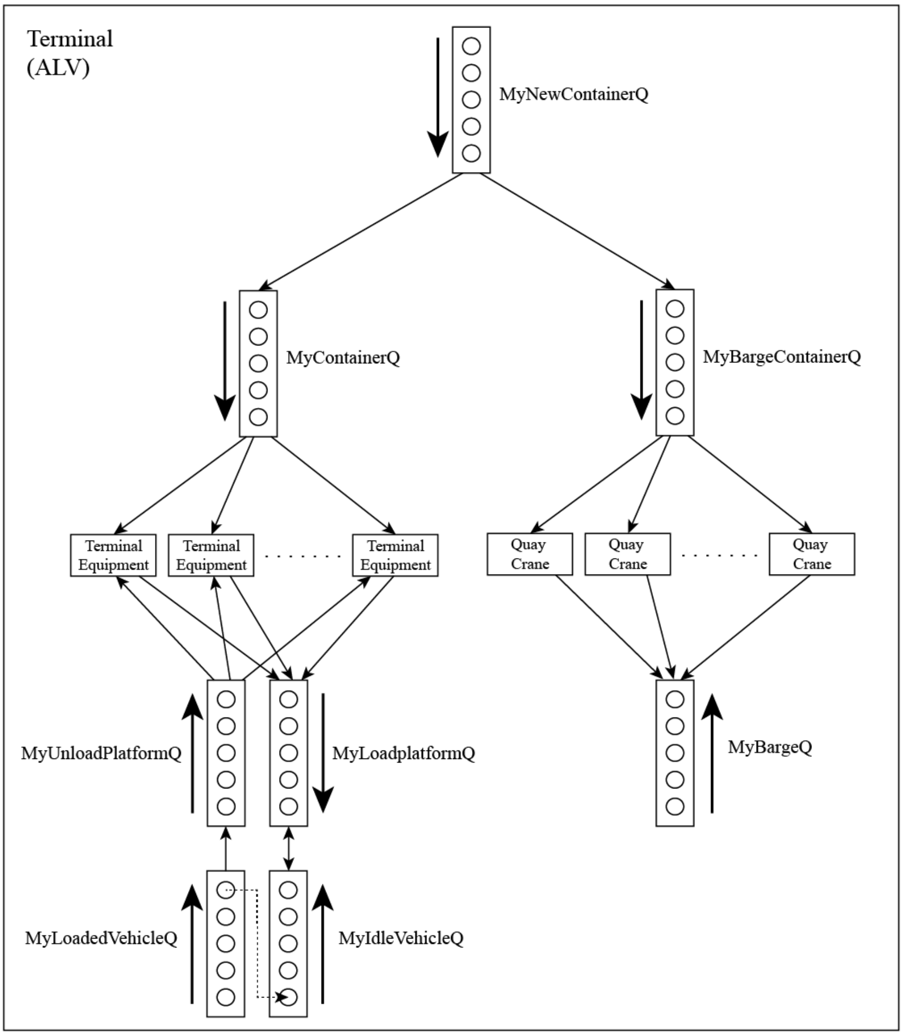

Figure 4.7: Terminal layout for ALV scenarios

• LoadPlatformCapacity	Number of load platforms, set as 2 per Terminal Equipment (only for ALV)
• UnloadPlatformCapacity	Number of unload platforms, set as 2 per Terminal Equipment (only for ALV)
• MyMTSLoadContainerQ	Queue with Containers ready for transport by MTS (only for MTS)
• MyMTSSelectContainerQ	Queue with first Containers on trailers that are allowed to be transported by MTS (only for MTS)
• MyMTSUnloadContainerQ	Queue with Containers dropped off by MTS (only for MTS)
• MyEquipmentType	Type of Equipment working at the Terminal
• MyBargeQ	Queue with Barges located at the Terminal
• BargeTerminal	True if the Terminal is connected to the water network
• MyTerminalControl	Terminal Control coupled to this Terminal
• SelectModality	Method: determines if a Container should be transported by Barge or by Road
• PROCESS	Method: describes activities as a function of time

4.6.2 Process description

The Terminal's process is used for deciding wether to transport a Container by Barge or by Road. It also sorts the Containers in order to make sure that the Container with the highest priority is always transported first. The modality is determined using the method *SelectModality*. The general process goes as follows:

Repeat:

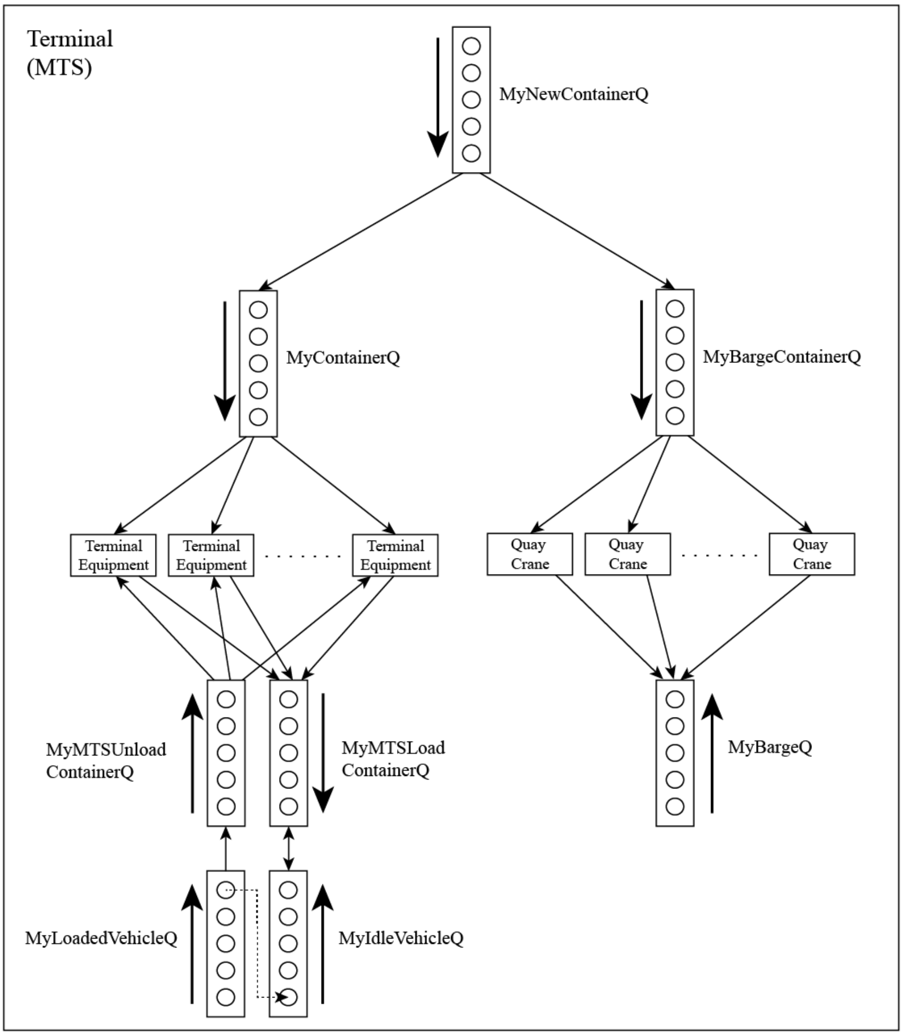

Figure 4.8: Terminal layout for MTS scenarios

- Wait while no Containers in MyNewContainerQ
- Select first Container in MyNewContainerQ
- Remove Container from MyNewContainerQ
- if SelectModality returns 'Road'
    - Add Container to MyContainerQ sorted by FinalStartTime (see Section 4.2)
- if SelectModality returns 'Barge'
    - Add Container to MyBargeContainerQ sorted by BargeHandlingTimeLeft

4.6.2.1 SelectModality method

The method *SelectModality* uses a decision tree to determine wether a new Container should be transported by Barge or by Road. If it is possible to deliver a Container to its destination in time then the Barge is preferred. The method can return two values: *Barge* and *Road*. The decision tree is shown in Figure 4.9.

4.6.3 Interaction with other objects

Container

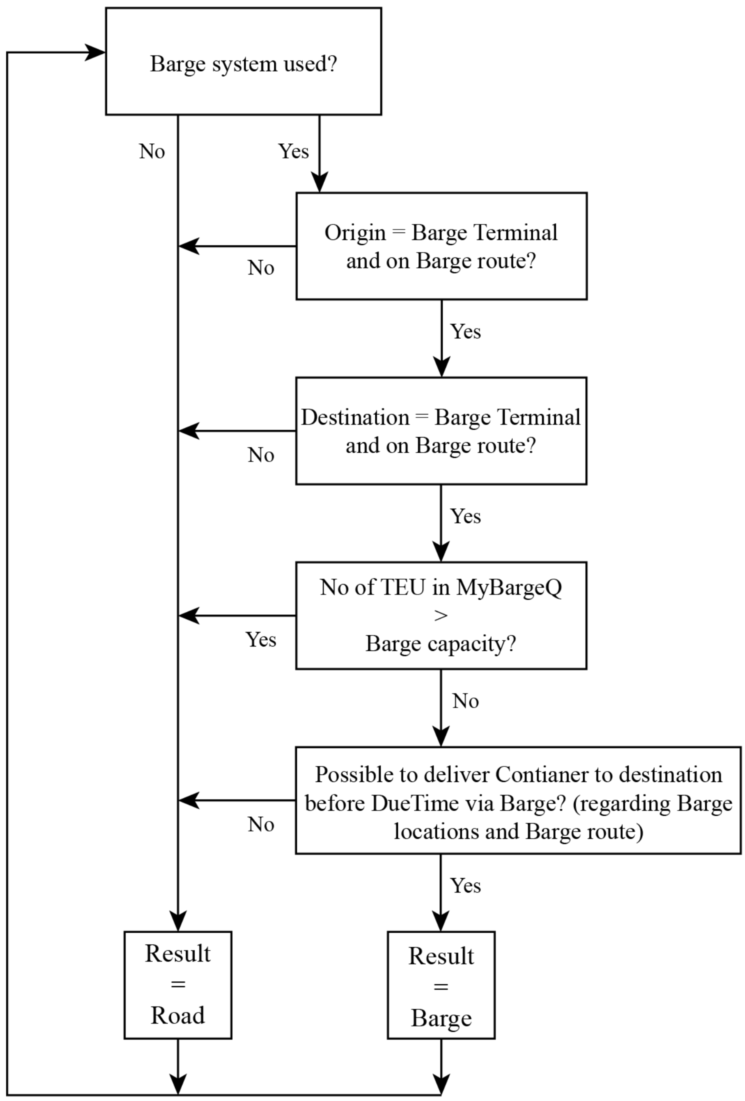

Figure 4.9: Decision tree for the *SelectModality* method

Containers remain in the Terminal's container queues until they are handled by the Terminal Equipment.

Generator

Containers are placed in the MyNewContainerQ by the Generator.

Terminal Equipment

A prespecified number of Terminal Equipment is allocated to operate at a certain Terminal and are placed in the Terminal's MyEquipmentQ.

Quay Crane

A prespecified number of Quay Cranes is allocated to operate at a certain Terminal and are placed in the Terminal's MyQuayCraneQ.

Vehicle

Vehicles reside in the Terminal's various vehicle queues.

Barge

Barges reside in the Terminal's MyBargeQ.

Terminal Control

Each Terminal has a Terminal Control coupled to it.

Node

A Node is made for each Terminal. They are used for Vehicle path planning (see Section 4.11).

4.7 Terminal Control

Every Terminal has a Terminal Control coupled to it. The Terminal Control is tasked with requesting empty rides from other Terminals when necessary. In case of the MTS scenario, the Terminal Control also decides when an MTS is allowed to (un)couple a certain trailer.

4.7.1 Attributes

• MyTerminal	Terminal it's coupled to
• FullTrailer	True if a trailer is full and ready to leave (only for MTS)
• RequestVehicle	Method: used for checking how many empty Vehicles need to be requested
• SelectTask	Method: used for selecting tasks in MTS scenario
• FullTrailerCheck	Method: used for checking if full trailers are available
• PROCESS	Method: describes activities as a function of time

4.7.2 Process description

The Terminal Control process is different for the MTS scenarios than for the other ones. Therefore they will be discussed separately. The Terminal Control has 3 methods that can be called for: *RequestVehicle*, *SelectTask* and *FullTrailerCheck*.

4.7.2.1 Terminal Control process for the AGV, Truck or ALV scenario

In case of the AGV, Truck or ALV scenarios the Terminal Control process is only used for requesting empty rides. The process goes as follows:

Repeat:

- Wait while *RequestVehicle* = 0
- Select Terminal with longest MyIdleVehicleQ
- Calculate if the selected Terminal can spare a Vehicle
- If so
  - Select First Vehicle from the selected Terminal's MyIdleVehicleQ
  - Remove Vehicle from the selected Terminal's MyIdleVehicleQ
  - Set Vehicle's Destination to Self
  - Make Vehicle perform an empty ride
- else
  - Wait for 0,01 hours and try again

4.7.2.2 Terminal Control process for the MTS scenario

In case of the MTS scenarios the Terminal Control process is not only used for requesting empty rides, but also for loading and unloading the MTSs. The process goes as follows:

Repeat:

- Wait while *SelectTask* = 'Wait'
- if *SelectTask* = 'Unload'
    - Select first MTS in MyLoadedVehicleQ
    - Remove MTS from MyLoadedVehicleQ
    - Transfer all Containers in MTS's MyContainerQ to MyMTSUnloadContainerQ
    - Activate MTS
- if *SelectTask* = 'Load'
    - Select first MTS in MyIdleVehicleQ
    - Remove MTS from MyIdleVehicleQ
    - Select first Container in MyMTSSelectContainerQ
    - Select up to 10 TEU of Containers from MyMTSLoadContainerQ for same destination as Selected Container
    - Transfer Containers to MTS's MyContainerQ
    - Activate MTS
- if *SelectTask* = 'RequestVehicle'
    - Select Terminal with longest MyIdleVehicleQ
    - Calculate if the selected Terminal can spare a Vehicle
    - If so
        - Select first Vehicle from the selected Terminal's MyIdleVehicleQ
        - Remove Vehicle from the selected Terminal's MyIdleVehicleQ
        - Set Vehicle's Destination to Self
        - Make Vehicle perform an empty ride
    - else
        - Wait for 0,01 hours and try again

4.7.2.3 RequestVehicle method

The method *RequestVehicle* is used in case of the AGV, Truck and ALV scenarios. It is called for by the general Terminal Control process and returns the number of empty Vehicles that needs to be requested. The decision tree for the method is given in Figure 4.10. An empty ride is allowed to be requested for a Container when $FinalStartTime - TNow <= RequestVehicleTime$. The decision process makes sure that only one Vehicle can be requested for one Container, and only when the Container approaches its last possible time it should be transported to be delivered in time. The value *RequestVehicleTime* determines how close to the Container's due time an empty Vehicle is allowed to be requested to transport it. The value can be defined in the configuration file (see Appendix B).

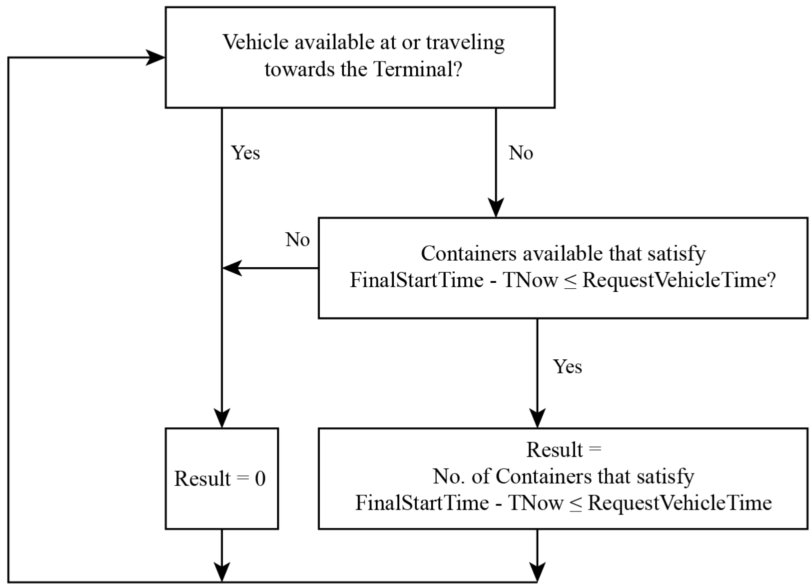

Figure 4.10: Decision tree for the *RequestVehicle* method

4.7.2.4 SelectTask method

The method *SelectTask* is used in case of the MTS scenarios. It can return four different results: *Unload*, *Load*, *RequestVehicle* and *Wait*. The decision tree for the method is given in Figure 4.11 Containers that are put into the MyMTSSelectContainerQ are seen as the first Container of a trailer that is allowed to be transported by an MTS. The value *MTSthreshold* determines how long before a Container's DueTime a not yet full trailer is allowed to be transported. *MTSthreshold* can be defined in the configuration file (see Appendix B).

4.7.2.5 FullTrailerCheck method

The method *FullTrailerCheck* is used in case of the MTS scenarios. It is called for by the Terminal Equipment each time it puts a new Container in the MyMTSLoadContainerQ. It then checks if this new Container fills up a trailer. The method's process goes as follows:

- MyContainer = new Container loaded by Terminal Equipment
- Calculate no. of TEU in MyMTSLoadContainerQ with same destination as MyContainer
- If $no.ofTEU >= MTSCapacity$
  - Put MyContainer in MyMTSSelectContainerQ

4.7.3 Interaction with other objects

Container

The Terminal Control determines which Containers are allowed to be transported by the MTSs.

Terminal Equipment

The method *FullTrailerCheck* is activated by the Terminal Equipment after it has transferred a new Container to MyMTSLoadContainerQ.

Vehicle

The Terminal Control activates Vehicles for empty rides and assigns Containers to the MTSs.

Terminal

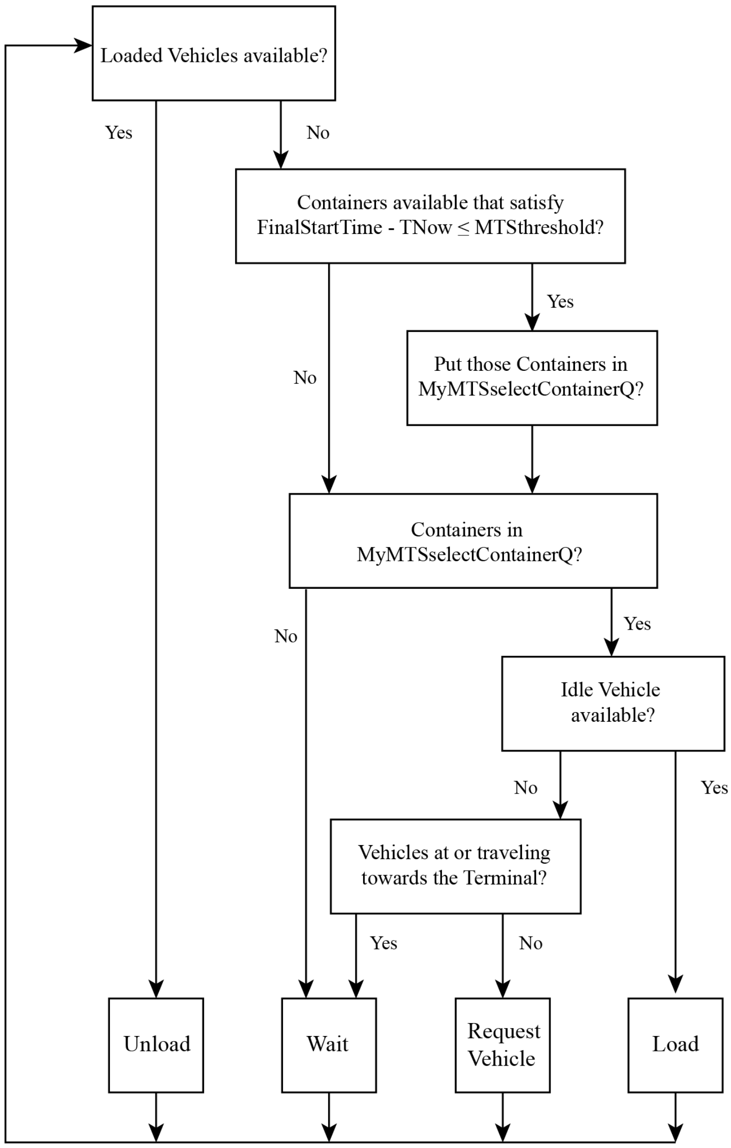

Figure 4.11: Decision tree for the *SelectTaskTC* method

Each Terminal Control is coupled to a Terminal.

4.8 Terminal Equipment

Terminal Equipment is a collective name for Container transfer equipment like Straddle Carriers, Automatic Stacking Cranes and Reachstackers. They are very different, but for this level of simulation they are almost the same. The only difference is their handling times. Each piece of Terminal Equipment is coupled to a Terminal, where it is responsible for loading and unloading the Containers. For all Terminal Equipment it is assumed that they can transfer one 40 ft. Container or two 20 ft. Containers at the same time.

4.8.1 Attributes

• MyTerminal	Terminal it is located at
• UnloadTime	Exponential distribution on the mean time it takes to unload a Container
• LoadTime	Exponential distribution on the mean time it takes to load a Container
• MyVehicle	Vehicle it's (un)loading
• SelectTaskAGV	Method: for deciding which action to take in AGV or Truck scenario
• SelectTaskLiftAGV	Method: for deciding which action to take in Lift AGV scenario
• SelectTaskMTS	Method: for deciding which action to take in MTS scenario
• PROCESS	Method: describes activities as a function of time

4.8.2 Process description

The Terminal Equipment uses a different process for different Vehicle type scenarios. The process is the same for the AGV and Truck, but different for the ALV and MTS. For each Vehicle type there is a separate method for determining which action to take: *SelectTaskAGV*, *SelectTaskLiftAGV* and *SelectTaskMTS*. The results can be: *Unload*, *Load*, or *Wait*. The Terminal Equipment will be discussed per Vehicle scenario.

4.8.2.1 Terminal Equipment process for the AGV or Truck scenarios

In the AGV or Truck scenarios, the process uses method *SelectTaskAGV* to determine which action to take. The decision tree of this method is shown in Figure 4.12. The process waits while there is nothing to load or unload. When both loading and unloading jobs are present, unloading is prioritized.

The general process is described below. All queues belong to the Terminal where the Equipment is situated at (see Section 4.6).

Repeat:

- Wait while *SelectTaskAGV = 'Wait'*
- Leave MyEquipmentIdleQ
- if *SelectTaskAGV = 'Unload'*
    - Select first Vehicle in MyLoadedVehicleQ
    - Remove Vehicle from MyLoadedVehicleQ
    - Put Vehicle in MyUnloadingVehicleQ
    - Wait for UnloadTime.Sample
    - For all Containers in Vehicle's MyContainerQ
        - Call for the Container's *RegisterNonPerformance* method
        - Destroy the Container object
    - Remove Vehicle from MyUnloadingVehicleQ

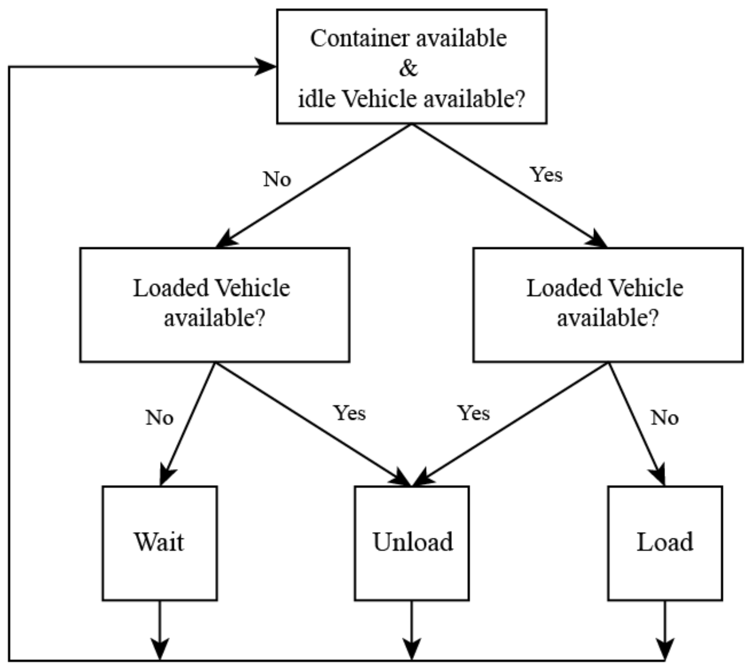

Figure 4.12: Decision tree for the *SelectTaskAGV* method

- Put Vehicle in MyIdleVehicleQ
- if *SelectTaskAGV = 'Load'*
    - Select first Vehicle in MyIdleVehicleQ
    - Remove Vehicle from MyIdleVehicleQ
    - Select first Container in MyContainerQ
    - Remove Container from MyContainerQ
    - Put Container in Vehicle's MyContainerQ
    - If Container is 1 TEU
        - Search for 1 TEU Container with same destination in MyContainerQ
        - If such a Container is found then also put this Container in Vehicle's MyContainerQ
    - Wait for LoadTime.Sample
    - Activate Vehicle
- Enter MyEquipmentIdleQ

4.8.2.2 Terminal Equipment process for the ALV scenario

In the ALV scenarios the process uses method *SelectTaskALV* to determine which action to take. The decision tree of this method is shown in Figure 4.13. In this scenario the Terminal Equipment does not directly transfer the Containers to and from the Vehicles. Instead, Containers are placed on a platform. The ALVs themselves are capable of lifting Containers from and onto these platforms. For modeling simplicity they are split in separate loading and unloading platforms (see Section 4.6). Each set of platforms has a maximum capacity currently set to 2 per piece of Terminal Equipment, but this could easily be made adaptable per Terminal if it turns out that it is a limiting factor.

The general process is described below. All queues belong to the Terminal where the Equipment is situated at (see Section 4.6).

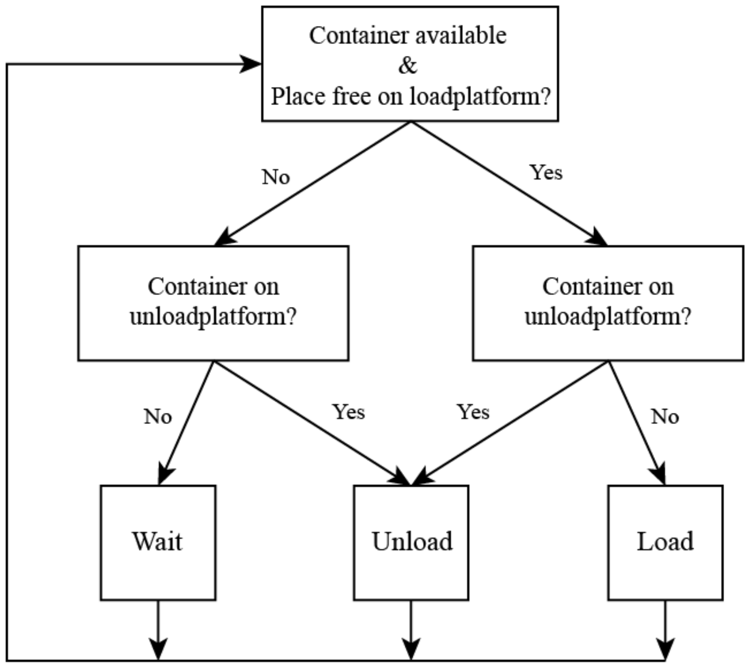

Figure 4.13: Decision tree for the *SelectTaskALV* method

Repeat:

- Wait while *SelectTaskALV* = *'Wait'*
- Leave MyEquipmentIdleQ
- if *SelectTaskALV* = *'Unload'*
    - Select first Container in MyUnloadPlatformQ
    - Remove Container from MyUnloadPlatformQ
    - If Container is 1 TEU
        - Search for 1 TEU Container with same destination in MyUnloadPlatformQ
        - If such a Container is found then also remove Container from MyUnloadPlatformQ
    - Wait for UnloadTime.Sample
    - Call for the Container(s)' *RegisterNonPerformance* method
    - Destroy the Container object(s)
- if *SelectTaskALV* = *'Load'*
    - Select first Container in MyContainerQ
    - Remove Container from MyContainerQ
    - If Container is 1 TEU
        - Search for 1 TEU Container with same destination in MyContainerQ
        - If such a Container is found then also remove Container from MyContainerQ
    - Reserve a spot in MyLoadPlatformQ
    - Wait for LoadTime.Sample
    - Put Container(s) in MyLoadPlatformQ
- Enter MyEquipmentIdleQ

4.8.2.3 Terminal Equipment process for the MTS scenario

In the MTS scenarios, the process uses method *SelectTaskMTS* to determine which action to take. The decision tree of this method is shown in Figure 4.14.

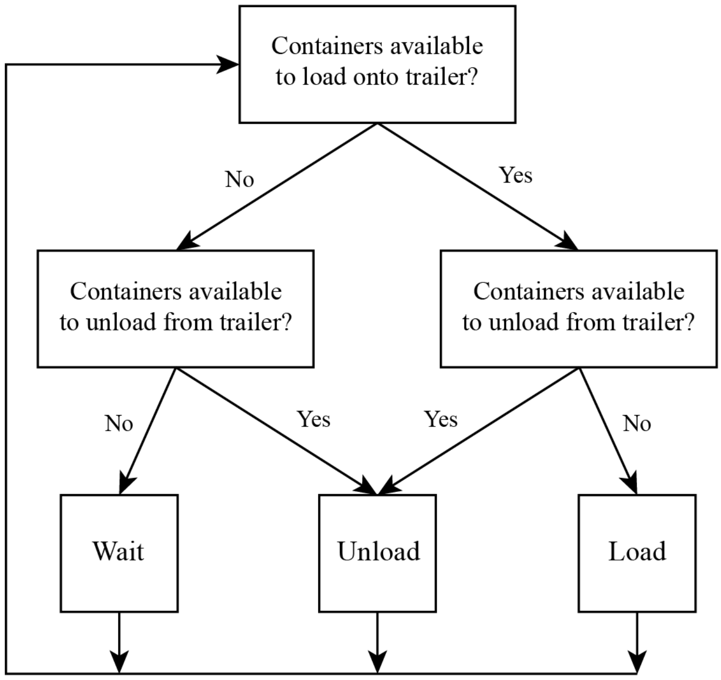

Figure 4.14: Decision tree for the *SelectTaskMTS* method

SelectTaskMTS figure

The general process is described below. All queues belong to the Terminal where the Equipment is situated at (see Section 4.6).

Repeat:

- Wait while *SelectTaskMTS = 'Wait'*
- Leave MyEquipmentIdleQ
- if *SelectTaskMTS = 'Unload'*
    - Select first Container from MyMTSUnloadContainerQ
    - Remove Container from MyMTSUnloadContainerQ
    - If Container is 1 TEU
        - Search for 1 TEU Container with same destination in MyMTSUnloadContainerQ
        - If such a Container is found then also remove Container from MyMTSUnloadContainerQ
    - Wait for UnloadTime.Sample
    - Call for the Container(s)' *RegisterNonPerformance* method
    - Destroy the Container object(s)
- if *SelectTaskMTS = 'Load'*
    - Select first Container in MyContainerQ
    - Remove Container from MyContainerQ
    - If Container is 1 TEU

        - Search for 1 TEU Container with same destination in MyContainerQ
        - If such a Container is found then also remove Container from MyContainerQ
    - Wait for LoadTime.Sample
    - Put Container(s) in MyMTSLoadContainerQ sorted by FinalStartTime
    - Call for Terminal Control's *FullTrailerCheck* method
- Enter MyEquipmentIdleQ

4.8.3 Interaction with other objects

Container

The Terminal Equipment is tasked with transferring the Containers.

Terminal

Every piece of Terminal Equipment is located at a certain Terminal. It resides in its Terminal's MyEquipmentQ and utilizes the other queues belonging to its Terminal to perform its tasks.

Terminal Control

The Terminal Control's method *FullTrailerCheck* is activated by the Terminal Equipment after it has transferred a new Container to MyMTSLoadContainerQ.

Vehicle

The Terminal Equipment transfers Containers to and from the AGVs and Trucks. After loading it activates the Vehicle process.

4.9 Roads

The Roads don't have a process. Every Road has queue called MyTrafficQ. Vehicles stay in this queue for the time it takes to cross the Road, depending on the Road's length and the Vehicle's speed. The Barge network also uses the Road objects, but upon creation these objects are added to a subset in order to create a separate network for the Barges. A schematic representation of a two-way Road can be seen in Figure 4.15.

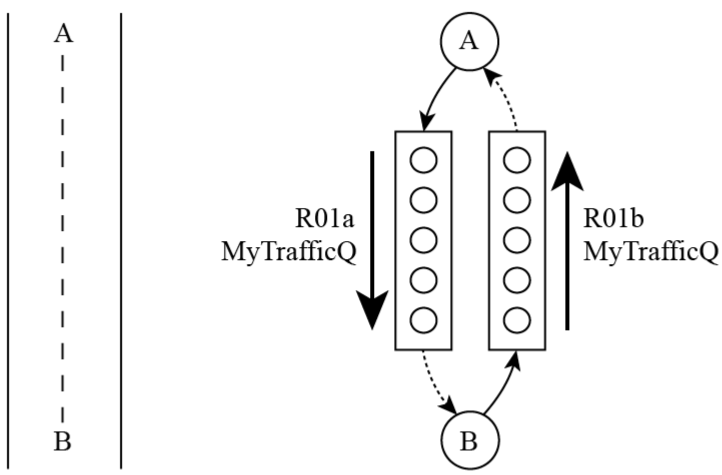

Figure 4.15: Schematic representation of a two-way Road

4.9.1 Attributes

- Length — Length of the Road in meters
- StartNode — Node at which the Road starts (see Section 4.11)
- StartOrientation — If Node is an Intersection: side of the Intersection it's connected to.
- EndNode — Node at which the Road ends (see Section 4.11)
- EndOrientation — If Node is an Intersection: side of the Intersection it's connected to.
- MyTrafficQ — Queue containing all Vehicles on the Road at that time

4.9.2 Interaction with other objects

Vehicle

Vehicles reside in the Road's MyTrafficQ while virtually driving that Road.

Barge

Barges reside in the Road's MyTrafficQ while virtually sailing that Road.

Node

Roads form the arcs between the different Nodes.

4.10 Intersection

The Intersections are modeled in such a way that congestion can occur. Different types of Intersections are modeled and different algorithms are used to determine which Vehicle is allowed to cross the Intersection first.

The different Intersection types are the following:

- Type 1 — Water Intersection; for the barge network.
- Type 2 — Crossing with rail or public road.
- Type 3 — 3 Way crossing.
- Type 4 — 4 way crossing.

A schematic representation of Intersection type 2 is given in Figure 4.16. For all other Intersections the system shown in Figure 4.17 is used. The 3 way crossing is modeled as a 4 way crossing, where 1 entrance/exit is not used. All possible ways to cross the Intersection are implemented in the model. This way it can be checked wether Vehicles are able to cross the Intersection at the same time without conflicts.

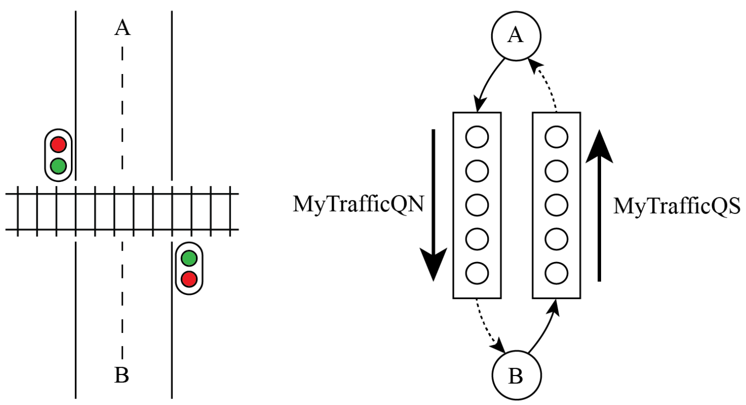

Figure 4.16: Schematic representation of Intersection type 2

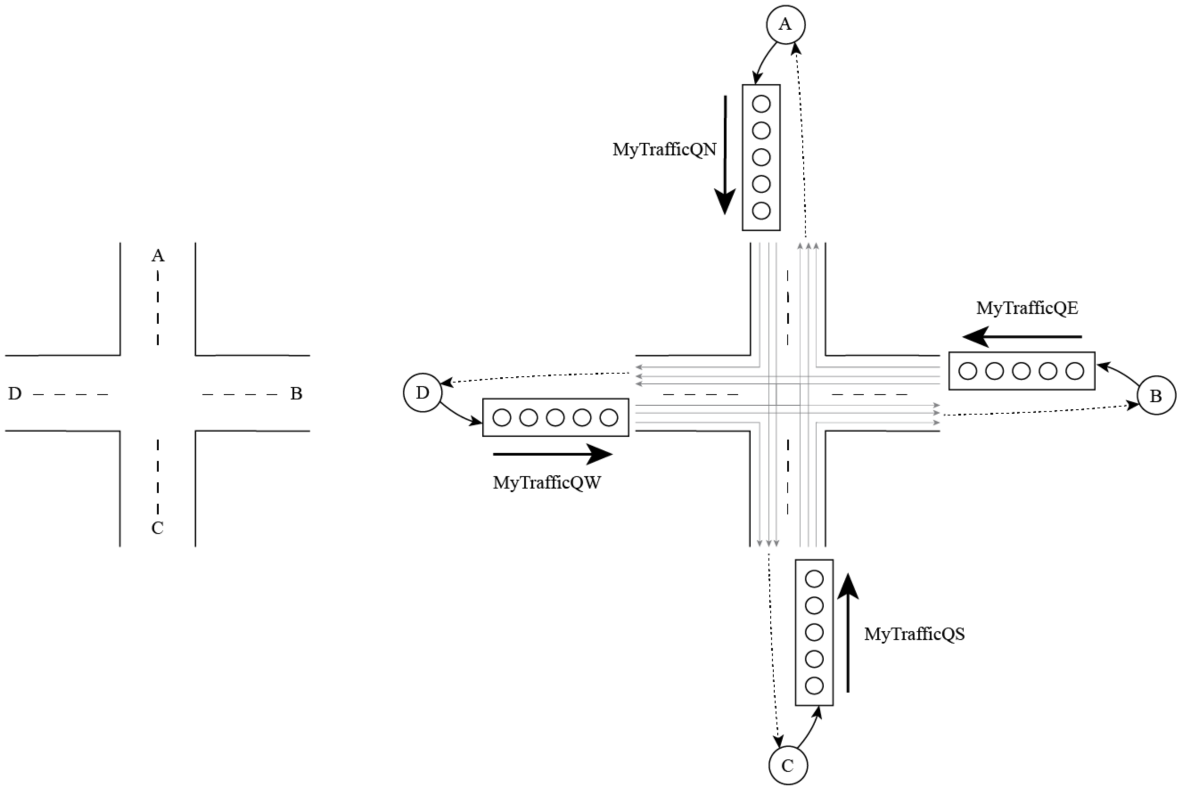

Figure 4.17: Schematic representation of Intersection type 1,3 and 4

4.10.1 Attributes

• TimeToCross	Time it takes to cross the Intersection for a Vehicle with a ClearTimeFactor of 1
• IntersectionType	Type of Intersection; 1, 2, 3 or 4.
• GreenLightTime	Exponential distribution on the mean time Vehicles have a green light (for type 2)
• RedLightTime	Exponential distribution on the mean time Vehicles have a red light (for type 2)
• MyTrafficQN	Traffic queue North.
• MyTrafficQE	Traffic queue East.
• MyTrafficQS	Traffic queue South.
• MyTrafficQW	Traffic queue West.
• MyAllTrafficQ	Queue containing all Vehicles on the Intersection at that time.
• MyUrgentQ	Queue with all urgent Containers at the Intersection.
• MyPriorityQ	Queue with Containers on Vehicles at front of TrafficQs sorted by PriorityTime
• MaxClearTime	Longest ClearTime of all Vehicles selected to cross the Intersection.
• SelectVehicle	Method: selecting Vehicles to cross Intersection.

4.10.2 Process description

For each Vehicle crossing an Intersection the time that it will take the Vehicle to cross the Intersection is calculated. This time is called the ClearTime and it is calculated using Formula 4.6. Vehicle are not able to overtake each other in the traffic queues.

$$ClearTime = (TimeToCross + VehiclePosition * QTime) * ClearTimeFactor \tag{4.6}$$

Where:

$ClearTime$	=	Vehicle attribute: time it takes the Vehicle to clear the Intersection. [hours]
$TimeToCross$	=	Intersection attribute: basic crossing time. [hours]
$VehiclePosition$	=	Vehicle's position in traffic queue. If at front: $VehiclePosition = 0$.
$QTime$	=	Time it takes a Vehicle to drive 1 extra position in a traffic queue.
$ClearTimeFactor$	=	Vehicle attribute. Slower and longer Vehicles have a higher ClearTimeFactor.

The Intersection process will be explained per Intersection type.

4.10.2.1 Type 1: water Intersection

The water Intersections always use the First-In-First-Out algorithm to determine which Barge is allowed to cross the Intersection first. The process goes as follows:

Repeat:

- Select first Barge in MyAllTrafficQ
- Calculate Barge's ClearTime (see Formula 4.6)
- Remove Barge from traffic queues
- Activate the Barge
- Wait for ClearTime

4.10.2.2 Type 2: crossing with rail or public road

The type 2 Intersection simulates a crossing with rail or public road. The Intersection has a traffic light which can be red or green. When the light is green, Vehicles are allowed to cross the Intersection freely. When it is red, the Vehicles have to wait until it is green again. The mean time that a light should be green before turning red, or the other way around, can be set per Intersection in the Intersection input file (see Appendix B). The process goes as follows:

Repeat:

- Calculate $EndOfGreenLight = TNow + GreenLightTime.Sample$
- repeat until $TNow = EndOfGreenLight$
    - Wait while no Vehicles at Intersection
    - For all Vehicles at Intersection
        - Calculate Vehicle's ClearTime (see Formula 4.6)
        - Remove Vehicle from traffic queues
        - Activate Vehicle
    - Wait for longest Vehicle ClearTime
- If TrafficLights turned ON then wait for RedLightTime.Sample

4.10.2.3 Type 3 or 4: 3 or 4 way crossing

The 3 and 4 way crossings both use the same process. These Intersections can use a First-In-First-Out (FIFO) algorithm or a priority algorithm. Which one is used can be defined in the configuration file (see Appendix B). The priority algorithm uses the method *SelectVehicle* to select a number of Vehicles that is allowed to cross the Intersection at the same time. It thereby considers urgent Containers, priority, if Vehicles are going in the same direction and if Vehicles from different directions are able to cross without conflicts.

If the **FIFO algorithm** is used the process goes as follows:

Repeat:

- Select first Vehicle in MyAllTrafficQ
- Calculate Vehicle's ClearTime (see Formula 4.6)
- Remove Vehicle from traffic queues
- Activate the Vehicle
- Wait for ClearTime

If the **priority algorithm** is used the process goes as follows:

Repeat:

- Call for the *SelectVehicle* method
- For all Vehicles in MySelectVehicleQ
  - Remove Vehicle from traffic queues
  - Activate the Vehicle
- Wait for longest Vehicle ClearTime; until the Intersection is cleared

4.10.2.4 SelectVehicle method

The method SelectVehicle is called for by the Intersection process (type 3 and 4, priority algorithm) to select Vehicles present at the Intersection that are allowed to cross at the same time. When a Vehicle is selected, its Intersection clear time is calculated and it is put in the MySelectVehicleQ. The method calculates the priority of Containers at the Intersection using the value PriorityTime (Formula 4.7). The Container with the lowest PriorityTime needs to be transported the soonest and has therefore got the highest priority.

$$PriorityTime = (DueTime - HandlingTimeLeft) - TNow \tag{4.7}$$

The process goes as follows:

- Check if there are urgent Containers at the Intersection
- If there are urgent Containers
  - If multiple; select urgent Container with highest priority
  - For all Vehicles in traffic queue up to Vehicle with selected Container
    - Calculate Vehicle's ClearTime (see Formula 4.6)
    - Put Vehicle in MySelectVehicleQ
- Calculate priority for all Containers on Vehicles at the front of the 4 traffic queues (see Formula 4.7)
- If there are no urgent Containers
  - Select Vehicle that holds Container with highest priority
  - For selected Vehicle + all Vehicles behind it going in the same direction
    - Calculate Vehicle's ClearTime (see Formula 4.6)
    - Put Vehicle in MySelectVehicleQ
- If all Vehicles in MySelectVehicleQ are going in the same direction
  - Select Vehicle holding Container with highest priority that can cross without conflicts
  - For selected Vehicle + all Vehicles behind it going in the same direction

- Calculate Vehicle's ClearTime (see Formula 4.6)
- Put Vehicle in MySelectVehicleQ

4.10.3 Interaction with other objects

Container

The Intersection read information from the Containers in order to determine which Vehicle is allowed to cross the Intersection first.

Vehicle

Vehicles reside in the Intersection's MyTrafficQ while virtually crossing that Intersection.

Node

A Node is made for every Intersection. They are used for Vehicle path planning (see Section 4.11).

4.11 Node

Nodes are automatically created for every Intersection and Terminal. They are used by the Vehicles and Barges for finding and driving/sailing the shortest route from origin to destination. By having a StartNode and EndNode attribute, the Roads form the arcs between the different Nodes.

4.11.1 Attributes

- Distance — Registers the distance from origin to this Node for the Dijkstra algorithm
- PreviousNode — Registers its previous node in the shortest route for the Dijkstra algorithm

The Dijkstra Algorithm is explained in Section 4.5.2.1

4.11.2 Interaction with other objects

Terminal

A Node is created for every Terminal.

Vehicle

The Vehicle's *FindShortestRoute* and *DriveShortestRoute* methods use Nodes for path planning.

Barge

The Vehicle's *FindShortestRoute* and *SailShortestRoute* methods use Nodes for path planning.

Generator

The Generator uses Nodes to calculate the Containers' expected handling time.

Intersection

A Node is created for every Intersection

Road

Roads form the arcs between the different Nodes.

4.12 Barge

The Barge is the only object that travels the water network. The Barge sails a set route past several Terminals on a set schedule. The route can be altered by changing the Barge route input file (see Appendix B). This way it is possible to make the Barge make a route past all Terminals with a water side, but also to just let it sail between 2 Terminals. The schedule is determined by the value *DepartureInterval*, which sets an amount of time between departures. The only reason a Barge is allowed to leave late is when not all Containers for the current destination have been unloaded. The Barge is allowed to leave early if it is full.

4.12.1 Attributes

• Speed	Average Barge speed [m/s]
• Capacity	Vehicle capacity in number of TEU
• TEUonBoard	Number of TEU currently carried by Vehicle
• MyRouteQ	Queue containing all Nodes on route
• MyContainerQ	Queue containing all Containers onboard
• MyOrigin	Terminal it drives from
• MyDestination	Terminal it needs to drive to
• RouteDistance	Total distance of the current route [m]
• MyIntersectionQ	Used to determine on which side to enter an Intersection
• MyIntersectionExit	Used to determine on which side to exit an Intersection
• ClearTime	Time it takes to cross the Intersection; calculated by Intersection
• ClearTimeFactor	Used to determine how long it takes a Vehicle to clear an Intersection
• MyRoad	Road it is driving on
• MyIntersection	Intersection it is on
• WaitAtIntersection	True if waiting at an Intersection
• MyLocation	Terminal its located at or sailing towards
• DepartureTime	Time the Barge should depart from its current MyLocation
• FindShortestRoute	Method: for finding the shortest route, using the Dijkstra algorithm
• SailShortestRoute	Method: for driving the shortest route
• PROCESS	Method: describes activities as a function of time

4.12.2 Process description

The Barge uses the methods *FindShortestRoute* and *SailShortestRoute* for finding its way to the next Terminal. They are the same ones as the Vehicle's *FindShortestRoute* and *DriveShortestRoute* methods (see Section 4.5), but adapted to the Barge object and network. The Barge's general process goes as follows:

Repeat:

- Wait while not activated by Quay Crane
- Calculate $DepartureTime = TNow + DepartureInterval$
- Wait for MooringTime.Sample
- Read next stop on route from Barge route input file
- Set MyLocation to next stop
- *FindShortestRoute*
- *SailShortestRoute*
- Set MyOrigin to current MyDestination
- Wait for MooringTime.Sample
- Enter MyBargeQ at MyDestination Terminal

4.12.3 Interaction with other objects

Container

Containers are transported by the Barges.

Terminal

Barges sail to and from Terminals. Barges can reside in the Terminal's MyBargeQ.

Quay Crane

Containers are transferred to and from the Barges by the Quay Cranes.

Road

Barges need to sail the waterways (Road objects) on their route by entering and leaving their MyTrafficQ in order to reach their destination.

Intersection

Barges need to cross the Intersections on their route by entering and leaving their traffic queues in order to reach their destination.

Node

Barges use Nodes in order to find and sail the shortest route between their origin and destination.

4.13 Quay Crane

The Quay Cranes are tasked with transferring Containers from and to the Barges. They are in many ways similar to the Terminal Equipment; the Terminal Equipment handle the Terminal's landside transfer and the Quay Cranes the waterside transfer. It is assumed that the Quay Cranes can transfer one 2-TEU Container or two 1-TEU Containers at the same time. A Barge can only be handled by one Quay Crane at the same time.

4.13.1 Attributes

• MyTerminal	Terminal it is located at
• UnloadTime	Exponential distribution on the mean time it takes to unload a Container
• LoadTime	Exponential distribution on the mean time it takes to load a Container
• PROCESS	Method: describes activities as a function of time

4.13.2 Process description

The Quay Crane runs through a process of unloading and then loading a Barge. It activates the Barge when it is full or when $TNow = DepartureTime$. The process goes as follows:

Repeat:

- Wait while no Barges in MyBargeQ
- Leave MyQuayCraneIdleQ
- Select first Barge in MyBargeQ
- Remove Barge from MyBargeQ
- For all Containers on Barge that have this Terminal as destination
  - Call for the Container's *RegisterNonPerformance* method
  - Destroy the Container object

- Repeat until Barge is full or $TNow \geq DepartureTime$
  - If only place left on Barge for 1 TEU
    - Search for 1 TEU Container in MyBargeContainerQ
    - Remove Container from MyBargeContainerQ
    - Put Container in Barge's MyContainerQ
    - Wait for LoadTime.Sample
  - else
    - Search for 1 TEU Container in MyBargeContainerQ
    - Remove Container from MyBargeContainerQ
    - Put Container in Barge's MyContainerQ
    - If Container is 1 TEU then wait for $0.5 * LoadTime.Sample$
    - else wait for LoadTime.Sample
- Activate Barge
- Enter MyQuayCraneIdleQ

4.13.3 Interaction with other objects

Container

The Quay Crane is tasked with transferring the Containers.

Terminal

Every Quay Crane is located at a certain Terminal. It resides in its Terminal's MyQuayCraneQ.

Barge

The Quay Crane transfers Containers to and from the Barges. After loading it activates the Barge's process.

4.14 Summary

The discrete event simulation model has been developed using Delphi and TOMAS. The simulation model is object-oriented and consists of the following objects: Containers, a Generator, an UrgencyCheck, Roads, Intersections, Terminals, Terminal Controls, Nodes, Terminal Equipment, Vehicles, Quay Cranes and Barges. Every Terminal has a Terminal Control coupled to it. Vehicles travel over a network of nodes and arcs. A Node object is made for every Intersection and Terminal. The arcs are represented by the Road objects. There are 2 separate road and water networks.

The model input can be defined in a number of input files (see Appendix B). The model output is formed by a number of output files and graphs (see Appendix C).

The next chapter will explain how the model has been verified and wether it is possible to validate it.

Chapter 5

Verification & Validation

Verification and validation is the process of checking wether a system works as it should. Verification deals with the question "Are we building the product right?. It is a static method for verifying if a model is error free and well-engineered. Validation deals with the question "Are we building the right product?". It is concerned with checking that the system satisfies the customer's actual needs. Verification will help to determine whether the software is of high quality, but it will not fully ensure that the system is useful.

This chapter will explain how the model has been verified and wether it is possible to validate it.

5.1 Verification using Delphi and Tomas

The ITT simulation model has been widely verified throughout the programming process. Delphi and Tomas offer several tools which make it possible to closely monitor all elements and processes in the simulation model. The simulation model's interface is represented by a Tomas Form (Figure 5.1). The form shows live information of a number of performance indicators during simulation. Tomas also provides a trace function which shows every action that is taken within the model. For example: the piece of trace information in Figure 5.2 shows that ASC number 2, located at terminal 5, starts loading container number 1603 at TNow $\bar{1}$5.58 hours. Besides a trace function, Tomas also provides a list of all elements and the queues they reside in (Figure 5.3), and a list of all queues and their statistics (Figure 5.4). All information can be viewed while running the simulations.

Delphi provides a debugging function (Figure 5.5) where it is possible to stop the simulation run at a certain line of programming code and then trace the rest of the process line by line. At each line it can be seen which values the attributes of the interacting elements have. This makes it easy to check wether the process does what it is supposed to do. This function has been used all throughout the programming process to verify that each piece of process works properly.

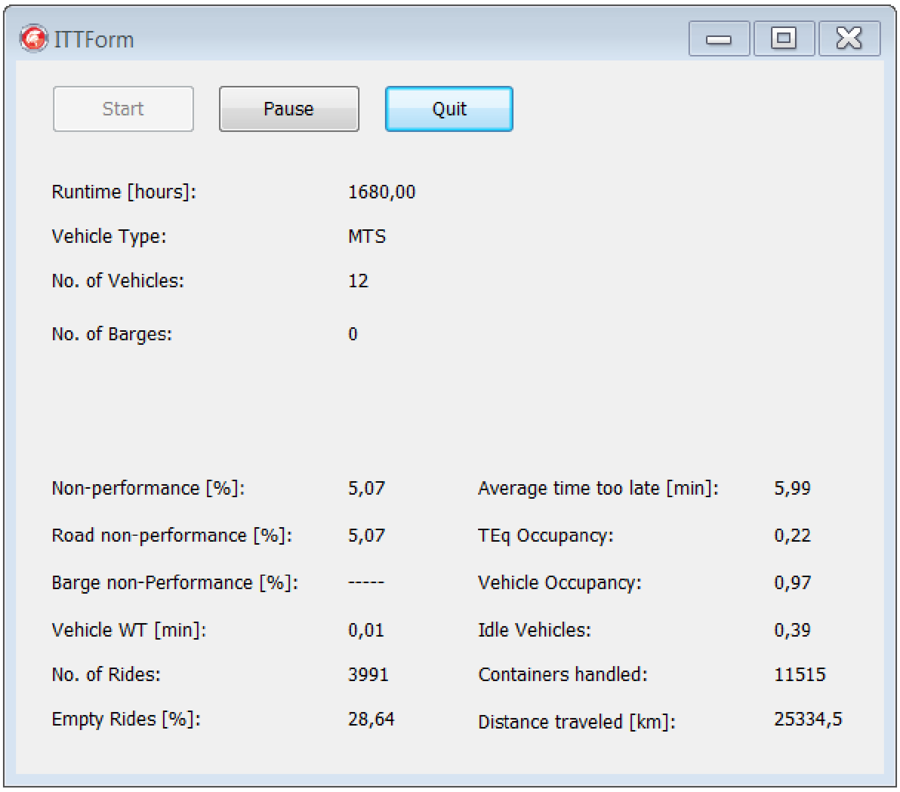

Figure 5.1: Tomas Form: simulation model interface

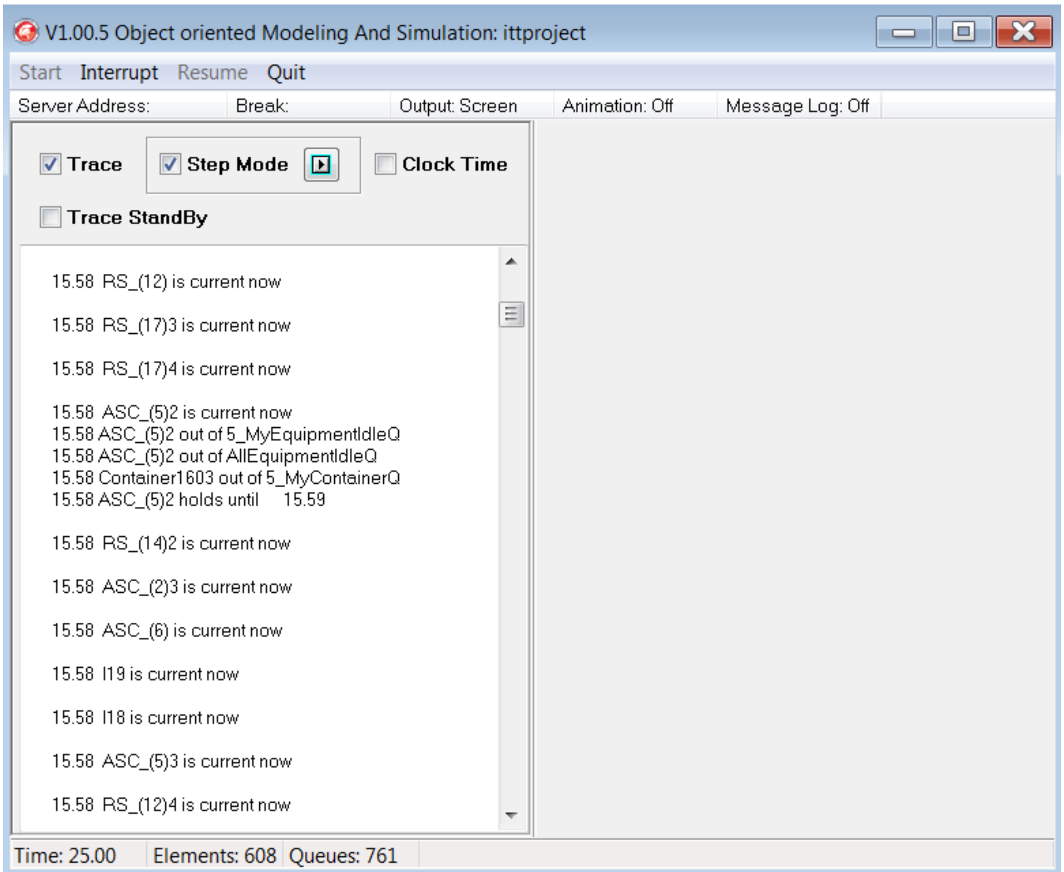

Figure 5.2: Tomas trace

Elements

Name	ArrivalTime	Status	Eventtime	Queue	QueueTime
				AllEquipmentIdleQ	0,00
16	0,00	Scheduled on Time	0,00	AllTerminalQ	0,00
16_Control	0,00	Scheduled on Time	0,00		
RS_(16)	0,00	Scheduled on Time	0,00	16_MyEquipmentQ	0,00
				16_MyEquipmentIdleQ	0,00
				AllEquipmentQ	0,00
				AllEquipmentIdleQ	0,00
RS_(16)2	0,00	Scheduled on Time	0,00	16_MyEquipmentQ	0,00
				16_MyEquipmentIdleQ	0,00
				AllEquipmentQ	0,00
				AllEquipmentIdleQ	0,00
RS_(16)3	0,00	Scheduled on Time	0,00	16_MyEquipmentQ	0,00
				16_MyEquipmentIdleQ	0,00
				AllEquipmentQ	0,00
				AllEquipmentIdleQ	0,00
17	0,00	Scheduled on Time	0,00	AllTerminalQ	0,00
17_Control	0,00	Scheduled on Time	0,00		
RS_(17)	0,00	Scheduled on Time	0,00	17_MyEquipmentQ	0,00
				17_MyEquipmentIdleQ	0,00
				AllEquipmentQ	0,00
				AllEquipmentIdleQ	0,00
RS_(17)2	0,00	Scheduled on Time	0,00	17_MyEquipmentQ	0,00
				17_MyEquipmentIdleQ	0,00
				AllEquipmentQ	0,00
				AllEquipmentIdleQ	0,00
RS_(17)3	0,00	Scheduled on Time	0,00	17_MyEquipmentQ	0,00
				17_MyEquipmentIdleQ	0,00
				AllEquipmentQ	0,00
				AllEquipmentIdleQ	0,00
RS_(17)4	0,00	Scheduled on Time	0,00	17_MyEquipmentQ	0,00
				17_MyEquipmentIdleQ	0,00
				AllEquipmentQ	0,00
				AllEquipmentIdleQ	0,00
18	0,00	Scheduled on Time	0,00	AllTerminalQ	0,00
18_Control	0,00	Scheduled on Time	0,00		
RS_(18)	0,00	Scheduled on Time	0,00	18_MyEquipmentQ	0,00
				18_MyEquipmentIdleQ	0,00
				AllEquipmentQ	0,00
				AllEquipmentIdleQ	0,00
RS_(18)2	0,00	Scheduled on Time	0,00	18_MyEquipmentQ	0,00
				18_MyEquipmentIdleQ	0,00
				AllEquipmentQ	0,00
				AllEquipmentIdleQ	0,00
MTS	0,00	Scheduled on Time	0,00	AllVehicleQ	0,00
				AllIdleVehicleQ	0,00
				1_MyIdleVehicleQ	0,00
MTS2	0,00	Scheduled on Time	0,00	AllVehicleQ	0,00
				AllIdleVehicleQ	0,00
				2_MyIdleVehicleQ	0,00
MTS3	0,00	Scheduled on Time	0,00	AllVehicleQ	0,00
				AllIdleVehicleQ	0,00
				3_MyIdleVehicleQ	0,00
MTS4	0,00	Scheduled on Time	0,00	AllVehicleQ	0,00
				AllIdleVehicleQ	0,00
				4_MyIdleVehicleQ	0,00

Figure 5.3: Tomas elements

Queues

Name	Length	Avg.Lgth	Max.Lgth	Tot.Passed	Direct	Avg. WT	Max. WT
AllVehicleQ	12	0,0	12	0	0	0,00	0,00
AllVehicleQ	0	0,0	0	0	0	0,00	0,00
AllIdleVehicleQ	0	0,6	12	1611	0	0,01	0,61
AllLoadedVehicleQ	0	0,3	4	982	0	0,01	0,02
AllContainerQ	306	269,3	368	3480	0	2,75	14,92
AllTerminalQ	18	0,0	18	0	0	0,00	0,00
AllTerminalSortedQ	0	0,0	0	0	0	0,00	0,00
AllRoadQ	76	0,0	76	0	0	0,00	0,00
AllIntersectionQ	21	0,0	21	0	0	0,00	0,00
AllWaterIntersectionQ	10	0,0	10	0	0	0,00	0,00
AllWaterwaysectionQ	44	0,0	44	0	0	0,00	0,00
AllEquipmentQ	61	0,0	61	0	0	0,00	0,00
AllEquipmentIdleQ	52	51,3	61	6553	617	0,29	12,26
AllBargeTerminalQ	0	0,0	0	0	0	0,00	0,00
AllNodeQ	39	0,0	39	0	0	0,00	0,00
AllWaterNodeQ	23	0,0	23	0	0	0,00	0,00
DijkstraQ	0	0,0	39	257746	257746	0,00	0,00
DijkstraRoadQ	0	0,0	2	151248	151248	0,00	0,00
ShortestRouteQ	0	0,0	19	42811	42811	0,00	0,00
AllQuayCraneIdleQ	0	0,0	0	0	0	0,00	0,00
AllQuayCraneQ	0	0,0	0	0	0	0,00	0,00
I01_MyTrafficQN	0	0,0	0	0	0	0,00	0,00
I01_MyTrafficQE	0	0,0	1	79	75	0,00	0,01
I01_MyTrafficQS	0	0,0	1	128	120	0,00	0,01
I01_MyTrafficQW	0	0,0	1	204	200	0,00	0,00
I01_MyAllTrafficQ	0	0,0	1	411	395	0,00	0,01
I01_MyUrgentQ	0	0,0	3	35	35	0,00	0,00
I01_MySelectVehicleQ	0	0,0	1	411	411	0,00	0,00
I01_MyPriorityQ	0	0,0	8	909	909	0,00	0,00
I02_MyTrafficQN	0	0,0	0	0	0	0,00	0,00
I02_MyTrafficQE	0	0,0	1	204	204	0,00	0,00
I02_MyTrafficQS	0	0,0	0	0	0	0,00	0,00
I02_MyTrafficQW	0	0,0	1	205	205	0,00	0,00
I02_MyAllTrafficQ	0	0,0	1	409	409	0,00	0,00
I02_MyUrgentQ	0	0,0	0	0	0	0,00	0,00
I02_MySelectVehicleQ	0	0,0	0	0	0	0,00	0,00
I02_MyPriorityQ	0	0,0	0	0	0	0,00	0,00
I03_MyTrafficQN	0	0,0	1	204	186	0,00	0,01
I03_MyTrafficQE	0	0,0	9	109	88	0,00	0,01
I03_MyTrafficQS	0	0,0	2	291	279	0,00	0,01
I03_MyTrafficQW	0	0,0	0	0	0	0,00	0,00
I03_MyAllTrafficQ	0	0,0	9	604	553	0,00	0,01
I03_MyUrgentQ	0	0,0	3	43	43	0,00	0,00
I03_MySelectVehicleQ	0	0,0	9	604	604	0,00	0,00
I03_MyPriorityQ	0	0,0	8	1230	1230	0,00	0,00
I04_MyTrafficQN	0	0,0	4	290	261	0,00	0,01
I04_MyTrafficQE	0	0,0	1	104	97	0,00	0,01
I04_MyTrafficQS	0	0,0	2	344	320	0,00	0,01
I04_MyTrafficQW	0	0,0	0	0	0	0,00	0,00
I04_MyAllTrafficQ	0	0,0	4	738	678	0,00	0,01
I04_MyUrgentQ	0	0,0	3	44	44	0,00	0,00
I04_MySelectVehicleQ	0	0,0	4	738	738	0,00	0,00
I04_MyPriorityQ	0	0,0	11	1486	1486	0,00	0,00
I05_MyTrafficQN	0	0,0	2	342	311	0,00	0,01
I05_MyTrafficQE	0	0,0	1	89	79	0,00	0,01

Figure 5.4: Tomas queues

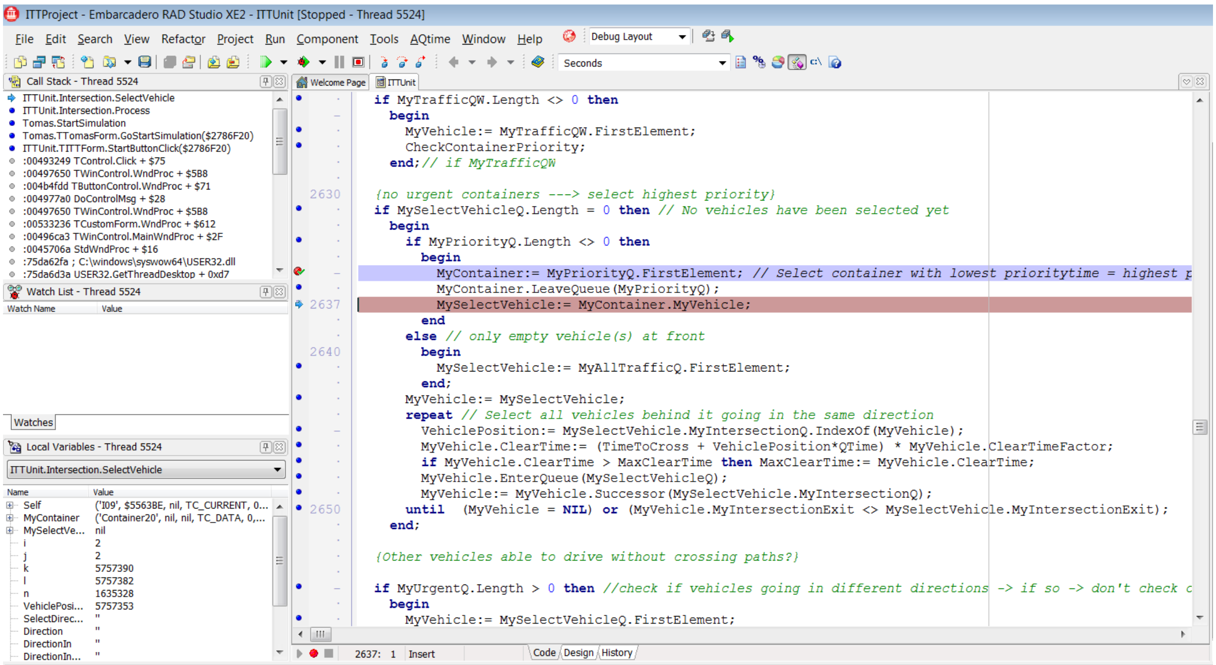

Figure 5.5: Delphi debugging interface

5.2 Verification using a simplified simulation input

The ITT simulation model has been verified using a simplified simulation input for which expected results can be calculated by hand. A simple ITT network has been made containing 5 terminals and 3 intersections. The map of the ITT network is shown in Figure 5.6. The black lines form the road network, the red lines form the barge network.

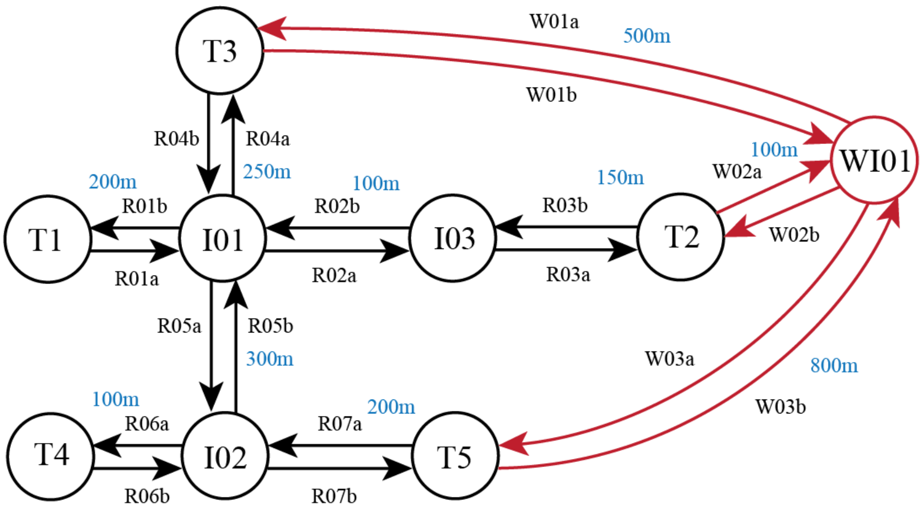

Figure 5.6: Simplified ITT map

5.2.1 Verification run settings

The following settings have been used for the verification run:

- Runtime 1000 hours
- Vehicle type ALV
- No. of vehicles 10
- No. of barges 2
- TimeToCross for I01 and I02 10 s
- TimeToCross for I03 and WI01 0 s
- Barge route T3 - T2 - T5 (and then back to T3)

A special transport demand input has been created. Each hour 5 containers are created. Their properties are displayed in Table 5.1.

Container number	**Release time [h]**	**Origin**	**Destination**	**No. Of TEU**	**Due time [h]**
1	0,1	T3	T2	2	20
2	0,3	T3	T2	2	2
3	0,5	T2	T4	2	10
4	0,7	T4	T3	2	10
5	1	T5	T3	2	20

Table 5.1: Transport demand input

All settings that have not been discussed are the same as the settings for the general experiments (see Section 6.2)

5.2.2 Verification run results

A verification run has been performed using the settings described above. The results for this run are given in the following output files: the general output file (Figure 5.7), the road and intersection output file (Figure 5.8), the terminal output file (Figure 5.9) and the barge output file (Figure 5.10).

```
Non-Performance [%]:          0.000
Average time too late [min]:           0.00
Non-Performance Road [%]:              0.000
Non-Performance Barge [%]:             0.000
Mean TEq Occupancy:            0.07
Mean QC Occupancy:             0.34
Mean Vehicle Occupancy:                0.01
Mean Idle Vehicle waiting time [hour]:         1.96
Mean Loaded Vehicle waiting time [hour]:               0.00
Mean no. of idle vehicles:             9.89
Total no. of rides:      3000
No. of empty rides:      0
Percentage of empty rides:             0.00
Total no. of containers created:         5001
Mean no. of containers created per hour:               5.00
No. of containers created in peak hour: 5
Mean no. of containers handled per hour:               4.99
Total no. of containers handled:         4989
Containers handled via Road:     3000
Containers handled via Barge:    1989
Mean Vehicle loading rate [%]:       100.00
Mean Barge loading rate [%]:           12.33
Total distance traveled by Vehicles[km]:            1800.00
Total distance traveled empty by Vehicles [km]:        0.00
Total distance traveled by Barges [km]:      601.60
Total delay due to traffic [hour]:             0.00
Mean delay due to traffic per ride [s]:        0.00
Mean ride time [min]:          1.18
Mean ride distance [km]:               0.60
```

Figure 5.7: General output file for the verification run

Name	TotalPassed	Mean Veh/h	Max Veh/h	Mean Veh	Max Veh	Total Delay [hour]	Vehicle Delay [s]
I01	3000	3.00	4	0.00	1	0.00	0.00
I02	2000	2.00	3	0.00	1	0.00	0.00
I03	2000	2.00	2	0.00	1	0.00	0.00
WI01	645	0.65	3	0.00	1		
R01a	0	0.00	0	0.00	0		
R01b	0	0.00	0	0.00	0		
R02a	1000	1.00	1	0.00	1		
R02b	1000	1.00	1	0.00	1		
R03a	1000	1.00	1	0.00	1		
R03b	1000	1.00	1	0.00	1		
R04a	1000	1.00	2	0.00	1		
R04b	1000	1.00	1	0.00	1		
R05a	1000	1.00	1	0.00	1		
R05b	1000	1.00	2	0.00	1		
R06a	1000	1.00	1	0.00	1		
R06b	1000	1.00	2	0.00	1		
R07a	0	0.00	0	0.00	0		
R07b	0	0.00	0	0.00	0		
W01a	214	0.21	2	0.00	2		
W01b	215	0.21	2	0.00	2		
W02a	215	0.21	2	0.00	1		
W02b	216	0.22	2	0.00	1		
W03a	216	0.22	2	0.01	2		
W03b	214	0.21	2	0.01	2		

Figure 5.8: Road and Intersection output file for the verification run

Name	EqType	NoOfEq	EqOccupancy	MeanWT IVehicleQ	MeanWT LVehicleQ	MeanL IVehicleQ	MeanL ContQ	Containers loaded No. + %		Containers unloaded No. + %	
T1	ASC	1	0.00	0.00	0.00	0.00	0.00	0	0.00	0	0.00
T2	ASC	1	0.10	2.17	0.00	2.17	0.00	1000	20.02	1000	20.04
T3	ASC	1	0.10	1.53	0.00	1.54	0.00	1000	20.02	1000	20.04
T4	RS	1	0.13	2.18	0.00	2.18	0.00	1000	20.02	1000	20.04
T5	SC	1	0.00	0.00	0.00	0.00	0.00	0	0.00	0	0.00

Figure 5.9: Terminal output file for the verification run

Name	# QC	QCOccupancy	MeanL ContQ	Containers loaded No. + %		Containers unloaded No. + %	
T2	1	0.35	0.00	0	0.00	997	19.98
T3	1	0.32	2.48	997	19.96	992	19.88
T5	1	0.34	2.18	998	19.98	0	0.00

Figure 5.10: Barge output file for the verification run

Modality check

The container properties (Table 5.1) have been chosen in such a way that if the system works properly, it can be predicted by which modality the containers will be transported.

- Container 1 Transport by barge, because barges are prioritized when there is enough time to deliver the container
- Container 2 Transport by road, because there is not enough time to deliver the container by barge
- Container 3 Transport by road, because T4 is not connected to the barge network
- Container 4 Transport by road, because T4 is not connected to the barge network
- Container 5 Transport by barge, because barges are prioritized when there is enough time to deliver the container

The verification run has a runtime of 1000 hours and each container is created once every hour, so every container will be transported 1000 times. This means that 3000 containers should be transported by road and 2000 by barge. This is consistent with the data in the output files. The number of containers loaded and unloaded for the barge network are a bit lower than 1000 because at the end of the simulation a couple of containers will still be on, or waiting for, a barge.

Vehicles per road and intersection - road network

Each created container is a 2 TEU container, so each ALV can only take 1 container. Also, the container's origins and destinations have been chosen in such a way that no empty rides have to be created. This means that the number of vehicles passing a certain point on the road map equals the number of containers. 3 Containers are transported by road, each has a different route.

The following 3 routes will be traveled across the road network:
- Container 2 T3 - R04b - I01 - R02a - I03 - R03a - T2
- Container 3 T2 - R03b - I03 - R02b - I01 - R05a - I02 - R06a - T4
- Container 4 T4 - R06b - I02 - R05b - I01 - R04a - T3

Each route will be traveled by 1000 vehicles, which makes it easy to calculate the number of vehicles per road and intersection. Intersection I01 is part of all 3 routes, so 3000 vehicles should pass it. Intersection I02 is part of 2 routes, so 2000 vehicles should pass it. Etcetera... All values are consists with the ones shown in the road and intersection output file (Figure 5.8).

Mean ride distance

3 Different routes are driven by the vehicles. They have the following distances:
- T3 - T2 $250 + 100 + 150 = 500m$
- T2 - T4 $150 + 100 + 300 + 100 = 650m$
- T4 - T3 $100 + 300 + 250 = 650m$

Each route is driven an equal amount of times, so the mean ride distance should be $\frac{500+650+650}{3} = 600m$. This is correct.

Mean ride time

The ALVs drive with an average speed of 40 km/h and the mean ride distance is 600m. Intersections I01 and I02 have a TimeToCross of 10 s. There is 1 of these intersections in route T3 - T2, both are in route T2-T4, and both are in route T4 - T3.

So the mean ride time should be $\frac{0.6}{40} * 60 + \frac{1+2+2}{3} * \frac{10}{60} = 1.18$ min. This is correct.

Barge loading rate

Two barges transport 2000 2-TEU containers. Barges sail at best every 3 hours and have a capacity of 50 TEU. Therefore, the mean barge loading rate should be a bit higher than $\frac{2000*2}{2*\frac{1000}{3}*50} * 100\% = 12\%$. This is correct.

Terminal equipment occupancy

Terminals T2 and T3 both load and unload a total of 2000 containers. The equipment working at these terminals is the Automatic Stacking Crane (ASC), which has an average handling time of 3 min. This means that the mean terminal equipment occupancy at these terminals should be $\frac{2000*\frac{3}{60}}{1000} = 0.10$. This is correct.

Terminal T4 also handles 2000 containers, but the equipment working here is the Reach Stacker (RS) which has an average handling time of 4 min. This means that the mean terminal equipment occupancy at this terminal should be $\frac{2000*\frac{4}{60}}{1000} = 0.13$. This is correct.

Terminals T1 and T5 handle 0 road containers and therefore have a mean terminal equipment occupancy of 0.

5.3 Validation

Validation of the model is unfortunately impossible because there are no existing inter terminal transport systems to which the results of the simulation model could be compared. The only way at this point is to compare the results to those of different ITT models using the same input, but this is still not a real validation. The results of the simulation model will be compared to those of Frans Nieuwkoop's integer programming model [41] throughout Chapter 6.

5.4 Summary

This chapter has explained how the ITT simulation model has been verified. The ITT simulation model has been widely verified throughout the programming process. Delphi and Tomas offer several tools which make it possible to closely monitor all elements and processes in the simulation model. The ITT simulation model has been verified using a simplified simulation input for which expected results can be calculated by hand. Validation of the model is unfortunately impossible because there are no existing inter terminal transport systems to which the results of the simulation model could be compared.

Now it has been verified that the simulation model works as intended, the next chapter will explain how the model has been used to conduct a series of experiments. The results of these experiments will be given and discussed.

Chapter 6

Experiments & results

The developed ITT simulation model simulation model has been used to conduct a series of experiments. This chapter will show the experimental plan, the simulation settings and the results. The results include the evaluation of the ITT configurations defined by Frans Nieuwkoop [41].

6.1 Experimental plan

The experimental plan has been given in Table 6.1. In total there are 37 different experiments. Because the model is deterministic, and not stochastic, every experiment has to be run only once.

Experiment 1 to 12 are the main experiments of this research. The configurations for these experiments result from the different ITT configurations determined by Frans Nieuwkoop [41] (see Chapter 2). According to Nieuwkoop's integer programming model, these configurations should make sure that more than 99% of the containers are delivered in time. Experiments will show if this is also the case for the more realistic simulation model.

Experiment 13 to 15 are meant to show the influence of changing the Barge route. In experiment 13, barges sail a round route past all waterside terminals. In experiment 14, barges will sail between 2 terminals that are far away from each other by land but close by water. In experiment 15, barges will sail between the 2 terminals that exchange the highest amount of containers.

Experiment 16 to 18, combined with 9, will show how much influence the vehicle speed has on the system's performance. Vehicle speeds chosen for the ITT configurations may have been overestimated. How will the same amount of vehicles perform with lower speeds?

Experiment 19 to 23, combined with 9, will show the relation between the number of used ALVs and the system's performance. This way it can be determined how many vehicles are necessary to obtain a certain level of non-performance. Because of time limitations this has only been done for 1 configuration, and not for all 12. This is advised to be done in future research.

Experiment 24 to 27, combined with 9, 22 and 23, will show the difference between the two used Intersection algorithms regarding occurring delays. The more advanced priority algorithm will most likely result in less delays, but will it also lower non-performance?

Experiment 28 and 29 will show if adding more Barges improves the system performance. Will more containers be transported by barge if more barges are available?

Experiment 30 to 33 will show the influence of turning on the traffic lights at the crossings with public road and rail. The simulation model has the possibility to simulate delays due to interactions with the outside world, for instance the influence of trains crossing the closed transport route. Will this have a significant effect on the system's performance and will this be an interesting area for future research?

Experiment 34 to 37 will show what happens when the number of Terminal Equipment at 1 Terminal is decreased. Terminal 11 has been chosen because it is the busiest Terminal. Therefore decreasing the capacity here will have the biggest influence on the total system.

Number	Scenario	Configuration	Remarks
1	1	51 ALVs	
2	1	65 AGVs	
3	1	16 MTSs	
4	1	41 Trucks + 2 Barges	
5	2	33 ALVs	
6	2	42 AGVs	
7	2	12 MTSs	
8	2	22 Trucks + 3 Barges	
9	3	24 ALVs	
10	3	32 AGVs	
11	3	9 MTSs	
12	3	17 Trucks + 2 Barges	
13	3	30 Trucks + 2 Barges	Barge round route
14	3	30 Trucks + 2 Barges	Barge between 2 & 3 (distance)
15	3	30 Trucks + 2 Barges	Barge between 1 & 11 (volume)
16	3	24 ALVs	Speed = 50 km/h
17	3	24 ALVs	Speed = 30 km/h
18	3	24 ALVs	Speed = 20 km/h
19	3	20 ALVs	
20	3	22 ALVs	
21	3	26 ALVs	
22	3	30 ALVs	
23	3	50 ALVs	
24	3	20 ALVs	FIFO for intersections
25	3	24 ALVs	FIFO for intersections
26	3	30 ALVs	FIFO for intersections
27	3	50 ALVs	FIFO for intersections
28	3	17 Trucks + 5 Barges	
29	3	17 Trucks + 10 Barges	
30	3	24 ALVs	Traffic lights on 2-2
31	3	24 ALVs	Traffic lights on 8-2
32	3	24 ALVs	Traffic lights on 18-2
33	3	24 ALVs	Traffic lights on 38-2
34	3	24 ALVs	Terminal 11: 5 SCs
35	3	24 ALVs	Terminal 11: 4 SCs
36	3	24 ALVs	Terminal 11: 3 SCs
37	3	24 ALVs	Terminal 11: 2 SCs

Table 6.1: Experimental plan

6.2 Simulation settings

Unless otherwise noted under remarks (Table 6.1), the settings described below have been used for all simulation runs. In order to make the results comparable, the general ITT system settings have been set to the same values as the ones used in the integer programming model by Frans Nieuwkoop [41].

6.2.1 Runtime

The runtime for the simulations has been set to 10 weeks (1680 hours) with a warm up period of 2 weeks (336 hours). This warm up period is used to take out the first performance peak in the system. This brings the total simulation time for each run to 12 weeks (2016 hours). In reality it takes on average 1 hour to run a 12 week simulation. The simulation times are shown in Figure 6.1. The graph shows the non-performance over time for a run with 24 ALVs for scenario 3. The graph does not reach a steady state because the transport demand input changes over time.

The transport demand input files used for the experiments (see Appendix B) contain a list of container

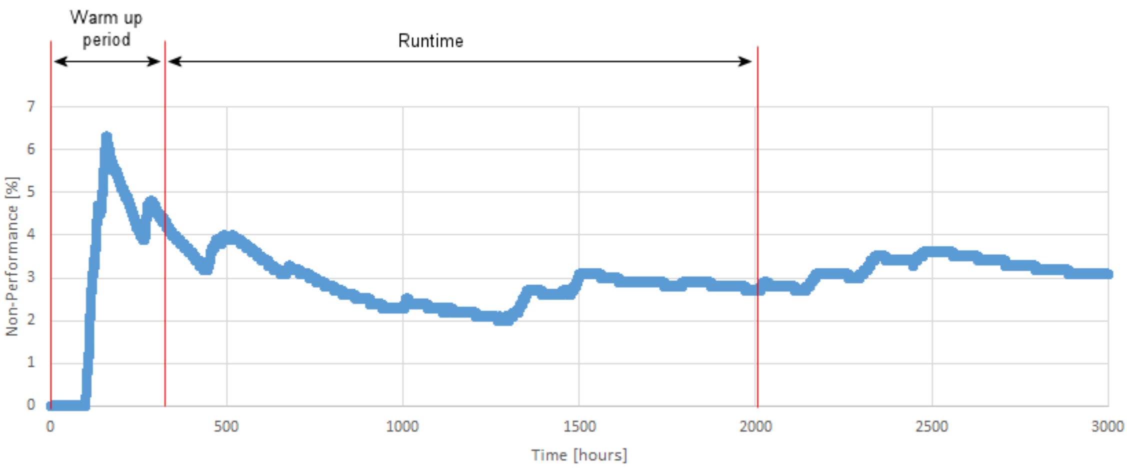

Figure 6.1: Runtime

transport jobs for more than the total runtime of 12 weeks. The lists are created using a demand generator made by Rick Jansen [27]. The demand generator does not use seeds, which means that if the scenario input does not change it will always produce exactly the same list. Therefore it is impossible to do multiple replications per experiment. However, results per scenario can be compared because the same transport demand file is used for every experiment.

6.2.2 Vehicle and equipment properties

The used vehicle and equipment properties have been given in respectively Table 6.2 and Table 6.3.

Vehicle type	**Average speed [km/h]**	**Capacity [TEU]**	**Clear time factor**	**Load time [min]**	**Unload time [min]**	**Coupling time [min]**
Truck	40	2	1			
AGV	40	2	1			
ALV	40	2	1	0,5	0,5	
MTS	30	10	2,67			0,5
Barge	12	50				

Table 6.2: Vehicle properties

Equipment type	**Unload time [min]**	**Load time [min]**
ASC	3	3
SC	4	4
RS	4	4
QC	3	3

Table 6.3: Equipment properties

6.2.3 Terminal Equipment per Terminal

The integer programming model by Frans Nieuwkoop [41] has an infinite capacity at the terminals, but in the ITT simulation model the capacity can be set by giving each terminal a number of terminal equipment and quay cranes. The number of terminal equipment per terminal has been determined by calculating the required capacity for the busiest scenario: scenario 1. This is shown in Table 6.4. The number of containers to be handled per year is calculated directly from the origin-destination matrix for scenario 1 (see Appendix F). The number of equipment required is calculated using Formula 6.1. In order to make sure that the system doesn't have to cope with large delays due to insufficient terminal capacity, each terminal has been given an overcapacity. This is calculated using Formula 6.2. Each terminal that has a waterside operation has been given 1 quay crane.

No.	**Equipment Type**	**Handling time [min]**	**Containers to be handled per year**	**Equipment required**	**Equipment used**	**Quay Cranes used**
1	ASC	3	400979	2,29	**4**	**1**
2	ASC	3	295598	1,69	**4**	**1**
3	ASC	3	201793	1,15	**3**	**1**
4	ASC	3	246946	1,41	**3**	**1**
5	ASC	3	186112	1,06	**3**	**1**
6	ASC	3	222815	1,27	**3**	**1**
7	ASC	3	203928	1,16	**3**	**1**
8	SC	4	30866	0,23	**2**	**1**
9	RS	4	9611	0,07	**2**	**1**
10	SC	4	436563	3,32	**5**	**0**
11	SC	4	499969	3,80	**6**	**1**
12	RS	4	298801	2,27	**4**	**1**
13	SC	4	225841	1,72	**4**	**1**
14	RS	4	89379	0,68	**3**	**1**
15	RS	4	121301	0,92	**3**	**0**
16	RS	4	137236	1,04	**3**	**0**
17	RS	4	261131	1,99	**4**	**0**
18	RS	4	55293	0,42	**2**	**0**

Table 6.4: Equipment per terminal

$$Equipment\ required = \frac{Containers\ per\ year}{\frac{365*24*60}{Handling\ time}} \tag{6.1}$$

$$Equipment\ used = Round(Equipment\ Required) + 2 \tag{6.2}$$

6.2.4 Infrastructure

The maps for the Maasvlakte road and barge network have been given in Appendix D. The lengths of all the roads and waterways can be seen on these maps. The settings for the different intersections have been given in Table 6.5. Intersections of type 2 can also be given a RedLighTime and GreenLightTime, but in the general experiments the traffic lights at these intersections are turned off so they don't have to be specified. The traffic lights have been turned off for the general experiments because they are not incorporated in the integer programming model by Frans Nieuwkoop [41]. Vehicles can now freely cross these intersections without having to wait. These intersections have been added to the map so the influence of having to wait for a crossing with rail or public road can be analyzed (see Section 6.3.7).

The intersection clear times for the simulation model and the integer programming model have been shown in Table 6.6. The calculation of the vehicle clear times is explained in Section 4.10.2.

6.2.5 Barge route

In the barge configuration experiments, the barges sail a round route past all terminals that have waterside operations. The barges visit the terminals in the following order: $9 \rightarrow 1 \rightarrow 11 \rightarrow 12 \rightarrow 14 \rightarrow 8 \rightarrow 3 \rightarrow 2 \rightarrow 7 \rightarrow 13 \rightarrow 6 \rightarrow 4 \rightarrow 5$. After terminal 5 the barge starts at the beginning of the list again and sails to terminal 9. The barges travel across the Maasvlakte barge map shown in Appendix D.

Intersection ID	**Time to cross [s]**	**Type**
I01	7,5	3
I02	0	2
I03	7,5	3
I04	7,5	3
I05	7,5	3
I06	0	2
I07	7,5	3
I08	0	2
I09	7,5	3
I10	0	2
I11	7,5	3
I12	7,5	3
I13	0	2
I14	7,5	3
I15	7,5	3
I16	7,5	3
I17	7,5	3
I18	7,5	3
I19	7,5	3
I20	7,5	3
I21	7,5	3
WI01	0	1
WI02	0	1
WI03	0	1
WI04	0	1
WI05	0	1
WI06	0	1
WI07	0	1
WI08	0	1
WI09	0	1
WI10	0	1

Table 6.5: Intersection properties

	Integer programming model		**Simulation model**		
Vehicle type	**Intersection capacity [vehicles/min]**	**Clear time [s]**	**Time to cross [s]**	**Clear time factor**	**Clear time [s]**
Truck	8	**7,5**	7,5	1	**7,5**
AGV	8	**7,5**	7,5	1	**7,5**
ALV	8	**7,5**	7,5	1	**7,5**
MTS	3	**20**	7,5	2,67	**20**
Barge	8	**0**	0		**0**

Table 6.6: Comparison of intersection properties

6.2.6 General settings

The general settings that have been used are displayed below. All values have been explained in Chapter 4 and Appendix B.

- DepartureInterval 3 hours
- MooringTime 30 minutes
- BargeExtraTime 2 hours
- Intersection strategy Priority
- Traffic Lights OFF
- Urgency Factor 2
- QTime 3 seconds
- RequestVehicleTime 5 hours
- MTSthreshold 2 hours

6.3 Results

6.3.1 Evaluation of ITT configurations

The 12 ITT configurations discussed in Section 2.3 have been run by the ITT simulation model. The ITT configurations are the results of the integer programming model, which means that these should have a non-performance of roughly 0 % in that model. As can be seen in Table 6.7, this is not the case for the simulation model.

Scenario	**Configuration**	**Non-performance [%]**	**Average lateness [min]**	**Average lateness for late containers [min]**	**Mean vehicle occupancy**
1	51 ALVs	18,3	84,1	460,18	0,91
	65 AGVs	41,5	927,5	2234,38	0,97
	16 MTSs	40,7	1908,0	4690,51	0,97
	41 Trucks + 2 Barges	98,6	15468,5	15689,39	0,99
2	33 ALVs	11,2	40,9	366,00	0,95
	42 AGVs	39,4	492,9	1249,62	0,99
	12 MTSs	26,7	220,6	825,07	0,98
	22 Trucks + 3 Barges	98,5	26258,5	26650,49	1
3	24 ALVs	2,5	0,9	36,17	0,92
	32 AGVs	21,7	49,7	229,55	0,93
	9 MTSs	19,3	42,8	221,45	0,98
	17 Trucks + 2 Barges	98,7	20965,6	21231,16	1

Table 6.7: Evaluation of ITT configurations

Table 6.7 shows that, with exception of the barge configurations, the non-performance is the highest for scenario 1 and lowest for scenario 3. This was to be expected because scenario 1 is the busiest and scenario 3 the quietest scenario. The more containers that have to be transported, the bigger the difference will be between the near-optimal working integer programming model by Nieuwkoop [41] and the non-optimal simulation model.

The ALV configurations score the lowest non-performance for all 3 scenarios. The ALVs are the most flexible of all configurations because they don't have to wait for a crane to load a container or for a full trailer that can be coupled. Therefore they operate the best without planning. However, for scenario 1 the ALV configuration still results in 18.3% of all containers not being delivered in time. The MTSs have the second lowest, and the AGVs the third lowest non-performance values. With the exception of scenario 2, these two configurations are relatively close to each other in terms of operational performance

The barge configurations score by far the worst for all 3 scenarios, with a non-performance of over 98% and containers being delivered weeks too late. The main reason these configurations score so poorly is due to the way they were modeled in the integer programming model. The barges were modeled continuous, instead of integer, because of memory issues. The result of this is that each container can be transported separately by a segment of a barge, without having to wait until a barge is full. This makes a barge of 50 TEU more or less work as a set of 1 or 2 TEU trucks, which are all used in an optimal way. In reality it does not work as efficient and flexible as this.

In the simulation model the barges are only allowed to transport a container when it is expected to be able to deliver the container in time. With the current transport demand, containers have on average 8 hours to be delivered. This is relatively short for a transportation system as slow as the barges. Therefore just a small percentage of the containers is able to be transported by barge. All others have to be transported by road, which results in big truck capacity shortage for all 3 scenarios.

The MTS scenarios also score quite poorly. Nieuwkoop also modeled the MTSs continuous, which makes them behave just like explained for the barges above. Besides that, the integer programming model also used bundled terminals. The used network consisted of 5 different origins and destinations while the simulation model has 18. Because there are less terminals, less empty trips would be required and vehicle capacity can be used more optimal. This is especially the case for vehicles with a larger capacity like the MTS. If there are less terminals, an MTS would have to wait less long before it has a full trailer than can be transported to another terminal.

The ALV and AGV were both modeled integer by Nieuwkoop. However, there is a significant difference in performance with the simulation model. This is mainly explained by the difference in the way containers are transferred at the terminals. Nieuwkoop's integer programming model assumes an unlimited capacity at the terminals. The time it takes a vehicle to pick up or drop off a container is the average time it would take the terminal equipment to unload it (for AGV) or for the vehicle itself to lift it onto or from a platform (for ALV). Unlike in the simulation model, vehicles never have to wait until terminal equipment is available. Translated to the simulation model this means the AGV basically works as an ALV with a lift (un)load time equal to the terminal equipment's (un)load time. In reality the main benefit of the ALV over the AGV is that the container transport is decoupled from the storage process, so the ALV and terminal equipment do not have to wait for each other to make a move. This difference between the two is modeled in the simulation model, but not in the integer programming model.

6.3.2 Varying the barge route

For the evaluation of ITT configurations a set barge round route was used, but the model makes it possible to use all kind of different routes. It is also possible to make the barges sail between just 2 terminals. This was done for 2 terminals that are very far away from each other by road but very close by water (terminal 2 & 3) and for the 2 terminals that exchange the highest number of containers (terminal 1 & 11). The results of these experiments with 30 trucks and 2 barges, for scenario 3, are shown in Table 6.9.

Configuration	**Barge route**	**Non-performance [%]**	**Barge loading rate [%]**	**Containers transported by barge**	**Containers transported by road**
2 Barges + 30 Trucks	Round route	27,9	4,88	975	160665
2 Barges + 30 Trucks	Between 2 & 3	27,6	0,49	162	161462
2 Barges + 30 Trucks	Between 1 & 11	22,7	18,69	5933	155703

Table 6.8: Various barge routes

The table shows that the barges have very low loading rates, even when barges just sail between terminal 1 & 11. Transport by barge is always prioritized, but if a container cannot be delivered in time by barge then it is transported by road. If containers would have more time to be delivered, the barge loading rate would become much higher.

Results show that letting barges sail between terminals that are very far away from each other by road but very close by water makes no sense when there are not enough containers available that can actually be transported by the barges. Only letting them sail directly between terminals with a lot of container exchange seems to have potential.

6.3.3 Varying the number of barges

Experiments have been run for scenario 3 in order to see if adding more barges would improve the system performance. The results are shown in Table 6.9. The results show that even when using 10 barges the non-performance would still only drop to 96,8% and the amount of containers transported by barge would still be much less than the amount of containers transported by road. This shows that with the current container transport demand and the long times that the barges need to transport the containers, the barges would not be a good alternative to the road vehicles.

Configuration	Barge route	Non-performance [%]	Barge loading rate [%]	Containers transported by barge	Containers transported by road
2 Barges + 17 Trucks	Round route	98,75	4,67	955	107542
5 Barges + 17 Trucks	Round route	98,05	3,37	1695	107539
10 Barges + 17 Trucks	Round route	96,80	3,12	3091	107120

Table 6.9: Varying the number of barges

6.3.4 Effect of vehicle speed

The vehicle speeds used in the experiments are a lot higher than those of the vehicles currently used at container terminals. Speeds were chosen this high for these simulations because results have to be comparable to those of Nieuwkoop's integer programming model [41]. In 2030 it might be possible that these vehicles can travel at such high speeds, but they cannot right now. In order to show the influence of vehicle speed on the system performance, simulations were run with varying vehicle speed. Simulations are run for scenario 3 with 24 ALVs. The effect of the vehicle speed on the system's non-performance is shown in Figure 6.2.

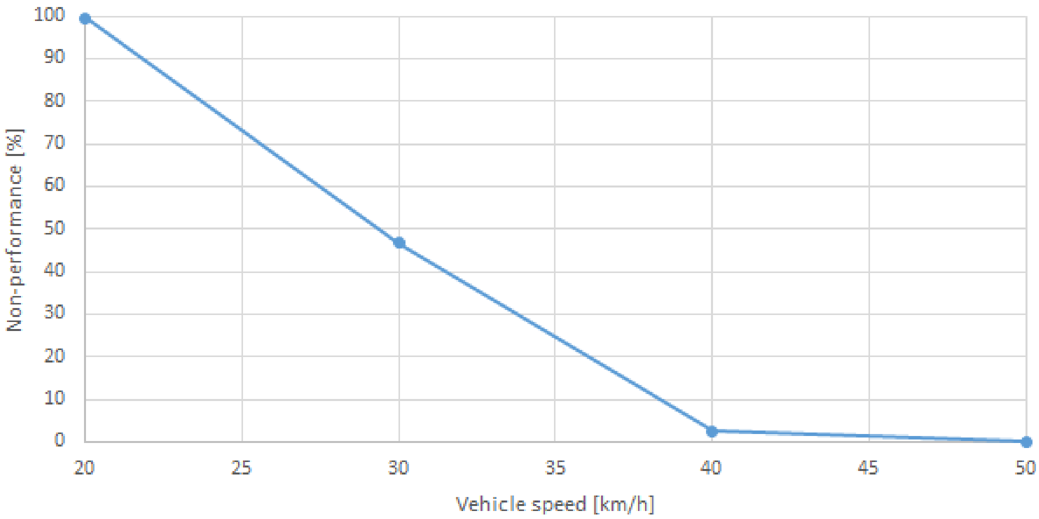

Figure 6.2: Effect of vehicle speed

Results show that the vehicle speed has a very big influence on the system's performance. Distances in the ITT system are big, with an average ride distance of about 6,7 km. Therefore vehicles spend most of their time driving from one terminal to another. If slower vehicles would be used, a lot more vehicles would be required to deliver the same amount of containers. This will likely results in more congestion problems.

6.3.5 Varying the number of vehicles

The ITT configurations that have to be evaluated already include a prespecified number of vehicles, but how well would the system perform if a different number of vehicles would be used? Experiments have been run to find this out for the vehicles with the lowest non-performance values: the ALVs. The graph in Figure 6.3 shows the non-performance for a varying number of ALVs for scenario 3.

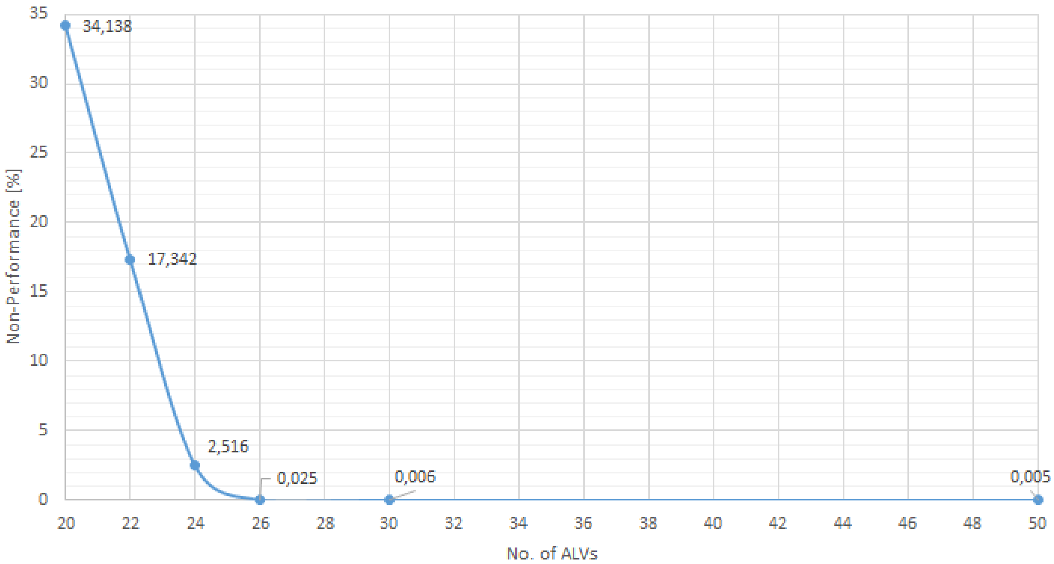

Figure 6.3: Varying the number of ALVs

As can be seen from the figure, there is a clear point where there is just enough capacity to make the system run smoothly. This is around 26 vehicles, 2 vehicles more than the configuration determined by the integer programming model. Using more than 26 vehicles won't result in large differences in performance.

Even with 50 vehicles the non-performance is still not 0%. This is due to the properties of the transport demand. The transport demand is generated in such a way that there is a very small percentage of containers for which it is impossible to deliver them in time, because the time they have to be delivered is smaller than the time needed to transport them to their destination.

The results given in Figure 6.3 show that the ITT simulation model makes it possible to find out how many vehicles are needed to obtain a certain level of non-performance, for instance lower than 1% or 0,5%. This research should be performed for all 12 instances, but because of time limitations this has not been done within this research. Therefore this is advised for future research.

6.3.6 FIFO vs priority algorithm

For the evaluation of the ITT configurations the priority algorithm was used for all simulation runs, but it is also possible to use a more simple First-In-First-Out algorithm. Figure 6.4 shows the differences in vehicle delay per ride between these two algorithms. The mean ride time is about 11 minutes. Table 6.11 shows the differences in non-performance. All experiments have been run for scenario 3.

No. Of ALVs	Priority	FIFO
20	34,138	33,405
24	2,516	2,495
30	0,006	0,002
50	0,005	0,004

Table 6.10: Non-performance [%] for the FIFO and priority algorithm

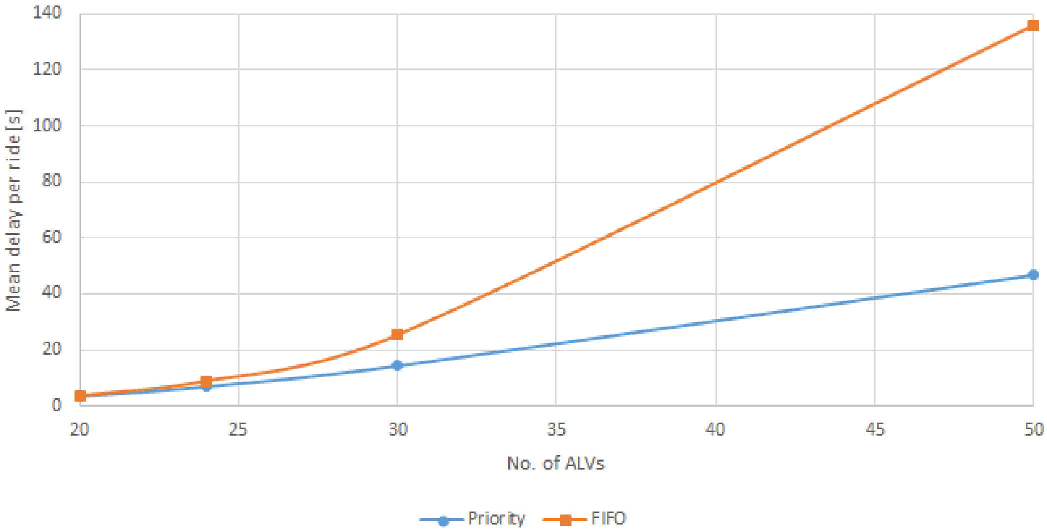

Figure 6.4: Mean vehicle delay per ride

The results show that using the priority algorithm results in much lower vehicle delays, especially when a larger amount of vehicles travels across the network. This is because the priority algorithm can select multiple vehicles that are allowed to cross at the same time, while for the FIFO algorithm the vehicles are only allowed to cross an intersection one at a time. However, results also show that the priority algorithm does not result in a lower non-performance. In fact, it is a little bit higher. This might be explained by the fact that the priority algorithm only considers what happens within the intersection and not around it. Empty vehicles automatically get the lowest priority, so they have to wait at the intersection until all other non-empty vehicles from other directions have crossed or there is a vehicle with an urgent container behind it. Therefore it is possible that an empty vehicle has to wait for a long time while there might be a container with a high priority waiting for the vehicle at a terminal.

6.3.7 Interactions with the outside world

The Maasvlakte ITT map has a number of crossings with rail or public road built in, but because they were not considered in the integer programming model, vehicles are able to pass through without delays in the ITT configuration evaluation simulation runs. These intersections are I02, I06, I08, I10 and I13 (see Appendix D). Several experiments have been run where vehicles cannot just cross these intersection freely, but they sometimes have to wait for a certain period of time because they have to wait for a train or vehicles on the public road to pass. All simulations have been run with 24 ALVs for scenario 3. The results are shown in Figure 6.5 and Figure 6.6. The “Time intersections are closed” means the average percentage of time that vehicles have to wait at these intersections. 0% Means that vehicles can just pass through freely, like in the ITT configuration evaluation simulation runs.

The results show that if the ITT system would have crossings with rail or public road, it would have a significant impact on the system. The delays per ride would increase and the performance would drop. Therefore more vehicles would be necessary to transport the same amount of containers. In these experiments there are just 5 of these crossings, but there might actually be many more in the final version of the Maasvlakte ITT system.

6.3.8 Varying the number of terminal equipment

For the evaluation of ITT configurations, each terminal has a prespecified number of equipment working at the terminal. This number of equipment was set in such a way so that each terminal would have an over capacity, but what would happen to the system performance if the number of equipment was reduced? In order to find this out a number of simulation runs has been performed with a varying number of equipment at the busiest terminal: terminal 11. The equipment working at this terminal is the Straddle

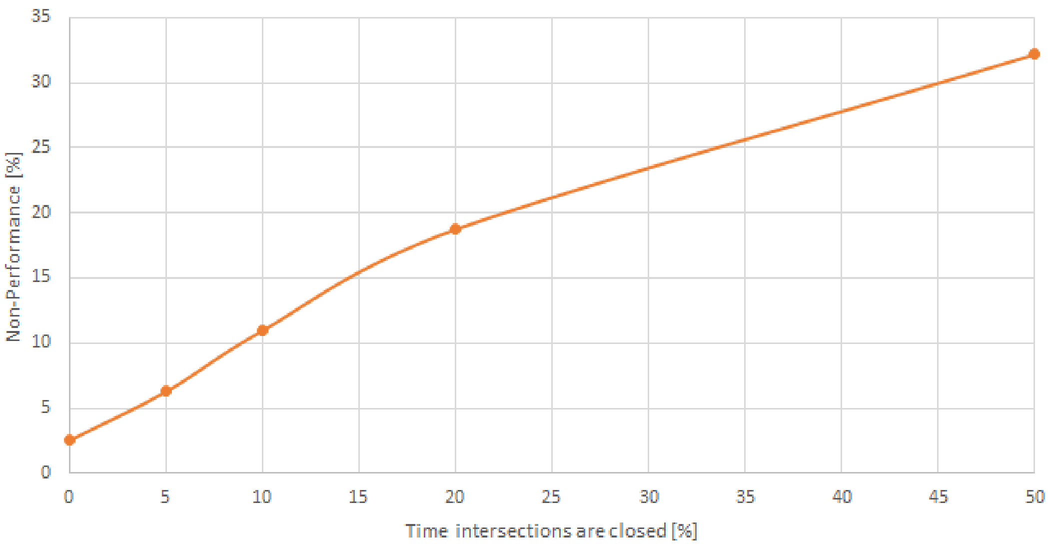

Figure 6.5: Non-performance when intersections are closed due to interaction with rail or public road

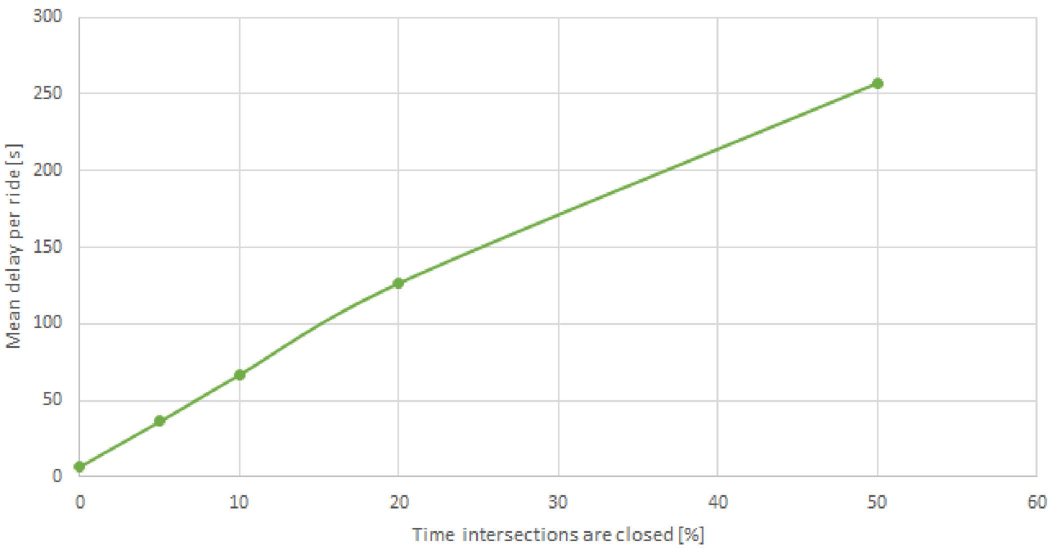

Figure 6.6: Mean delay per ride when intersections are closed due to interaction with rail or public road

Carrier (SC). Simulations have been run with 24 ALVs for scenario 3. The results are shown in Table 6.11. The non-performance is a global variable, the other ones are specific for terminal 11.

No. of SCs	**Non-performance [%]**	**Mean equipment occupancy**	**Mean length idle vehicles queue**	**Mean length loaded vehicles queue**
6	2,52	0,42	0,00	0,00
5	3,67	0,50	0,02	0,00
4	4,69	0,63	0,14	0,00
3	20,50	0,84	1,43	0,00
2	91,50	0,99	8,58	0,00

Table 6.11: Varying the number of terminal equipment

The situation with 6 SCs is the one simulated for the evaluation of ITT configurations. Decreasing the number of SCs to 5 or 4 raises the non-performance a bit, but big problems do not start to arise until there are only 3 SCs available. When there are only 2 SCs available there is a serious under capacity at the terminal and over 25% of the available vehicles are waiting at the terminal before they can pick up a container. This shows that the transfer capacity at the terminals is a major bottleneck and that an under capacity at only one terminal could have a big impact on the overall system performance.

The mean length of the loaded vehicles queue stays 0,00 even when there are only 2 SCs available. This is explained by the fact that unloading is always prioritized in the simulation model. The equipment capacity that is available is first used to unload containers. When there are no containers to unload, it starts loading containers.

6.4 Summary

A total number of 37 experiments has been performed. Because the model is deterministic, and not stochastic, every experiment has to be run only once. Every simulation run has a runtime of 10 weeks with a warm up period of 2 weeks. In reality it takes on average 1 hour to run a 12 week simulation. In order to make the results comparable, the general ITT system settings have been set to the same values as the ones used in the integer programming model by Frans Nieuwkoop [41].

Results show that the ALV configurations score the best for all 3 scenarios. The barges score by far the worst with a non-performance of over 98%. The reason they score so poorly is due to the unrealistic way they were modeled by Nieuwkoop. Varying the number of barges and the barge route has shown that barges can only work when they operate between terminals that exchange a lot of containers that allow a long delivery time. Results show that vehicle speed has a large influence on the system's performance. It has been shown that the ITT simulation model makes it possible to find out how many vehicles are needed to obtain a certain level of non-performance. This is advised to be done for all 12 instances in future research.

This research will be concluded in the next chapter. Various recommendations are made for future research.

Chapter 7

Conclusions & future research

7.1 Conclusions

The main research question for this research is "*Which of the defined ITT vehicle configurations is the best configuration seen from an operational perspective?*". In order to answer this question a discrete event simulation model for an Inter Terminal Transport system at the Maasvlakte 1 and 2 has been developed. The simulation model has been used to conduct a number of experiments to evaluate the ITT configurations defined by Frans Nieuwkoop [41] and to gain more insight into the working of the ITT system. The non-performance and average lateness values for the 12 given ITT configurations have been shown in Table 7.1. The ITT configurations are the results of Nieuwkoop's integer programming model, which means that these should have a non-performance of roughly 0 % in that model. As can be seen in the table, this is not the case for the simulation model.

Scenario	**Configuration**	**Non-performance [%]**	**Average lateness for late containers [hour]**
1	51 ALVs	18,3	7,67
	65 AGVs	41,5	37,24
	16 MTSs	40,7	78,18
	41 Trucks + 2 Barges	98,6	261,49
2	33 ALVs	11,2	6,10
	42 AGVs	39,4	20,83
	12 MTSs	26,7	13,75
	22 Trucks + 3 Barges	98,5	444,17
3	24 ALVs	2,5	0,60
	32 AGVs	21,7	3,83
	9 MTSs	19,3	3,69
	17 Trucks + 2 Barges	98,7	353,85

Table 7.1: Non-performance and time too late for the various ITT configurations

Choosing the best ITT configuration from an operational point of view could be done by calculating a weighted average over the most important performance indicators such as the non-performance, occupation rates, loading rates, distance traveled empty, etc. However, by far the most important performance indicators are how many containers are delivered in time and how much too late they have been delivered, and when differences in these values are so big as they are, it does not make sense to even consider the other much less important performance indicators. Therefore the choice of the best ITT configuration will only be based on the non-performance and the average time that containers have been delivered too late.

Since the ALV configurations have by far the lowest non-performance and lateness values for each of the 3 scenarios, the ALV configurations are the best configurations.

However, this can only be concluded under the currently used dispatching rules and vehicle properties. As shown in Section 6.3.5 the vehicle speed has a big influence on the system performance, which can be explained by the large distances in the ITT system. Vehicles spend most of their time driving. In the experiments the speed of the AGV and ALV have both been set to the same value, although the current ALVs are a bit slower than the current AGVs. This difference in speed might actually make the AGVs perform better than the ALVs. Also adding a proper planning system might make the less flexible configurations perform better than they do now.

The barge configurations score by far the worst for all 3 scenarios. The main reason these configurations score so poorly is due to the way they were modeled in Nieuwkoop's integer programming model. The barges were modeled continuous, instead of integer, because of memory issues. The result of this is that each container can be transported separately by a segment of a barge, without having to wait until a barge is full. This makes a barge of 50 TEU more or less work as a set of 1 or 2 TEU trucks, which are all used in an optimal way. In reality it does not work as efficient and flexible as this.

The integer programming model also used bundled terminals. The used network consisted of 5 different origins and destinations while the simulation model has 18. Because there are less terminals, less empty trips would be required and vehicle capacity can be used more optimal. This is especially the case for vehicles with a larger capacity like the MTS. If there are less terminals, an MTS would have to wait less long before it has a full trailer than can be transported to another terminal.

Barges do not seem to be a good option to be used in the ITT system. Handling them takes too much time; mooring alone already takes about an hour per visited terminal. Because of their large capacity, the large number of different terminals and the relatively short allowed delivery time of the containers, it is very hard to optimally use their capacity. The only way they might work is when they sail between terminals that share a lot of containers that allow a long delivery time.

The simulation model developed during this research is not the first ITT simulation model that has been developed, but it is the first model which incorporates traffic modeling into the ITT system. It is the first ITT simulation model where delays occurring due to traffic have an impact on the system's performance. The model is not only able to simulate delays within the system, but also delays due to crossings with rail or public road. The Port of Rotterdam expects that traffic delays will be a major problem for the ITT system. This research has provided a tool which is able to analyze this problem in detail.

More research is required in order to draw final conclusions from the ITT simulation model's results. Recommendations for future research will be given in the following section.

7.2 Future research

This research has provided a discrete event simulation model for an Inter Terminal Transport system. The model could be used to evaluate different operational aspects of the system. Also, several expansions and improvement could be thought of. The following researches are recommended:

- Because of memory issues, Nieuwkoop's integer programming model was run with 5 bundled terminals and the barge and MTS configurations were solved continuous instead of integer. It should be investigated how to solve all configurations integer with 18 terminals. This might result in a better approximation of the optimal number of MTSs and Barges required.
- New transport demand scenarios for 2013 are being constructed. Also, vehicle speeds used in the current simulation might be overestimated. Therefore, both the integer programming model and the simulation model should be rerun using the updated values. Lower vehicle speeds will likely result in a much higher number of required vehicles, which will likely results in more congestion problems.
- Use the simulation model to find the number of vehicles required to get a non-performance below 1% (or 0.5%, or 0.1%, etc.) for all 12 instances.
- The simulation model could be used to investigate the influence of crossings with rail or public road. These crossings were not included in the evaluation of ITT configurations, but experiments showed that these crossings could have a big influence of the performance of the system (see Section 6.3.7).

More research is needed in finding out where these intersections exactly are, what time delays they would cause, and what the effect of this would be on the system.

- The simulation model could be expanded in order to be able to evaluate the asset light configurations defined as part of the "asset light configuration" task of the "Inter-terminal transport on Maasvlakte 1 and 2 in 2030" project. These configurations contain a variable amount of vehicles that operate in the ITT system. A new 'vehicle generator' object would have to be added to the simulation model which is able to add vehicles to and remove vehicles from the ITT system at certain points in time.
- For the ITT configuration evaluation simulation runs a certain strategy was chosen on how to operate the barges. If barges were considered to be a viable option by the Port of Rotterdam , the simulation model could be used to investigate different barge routing strategies.
- The current simulation model is not optimized. Research could be done in how to optimize the simulation model without having processing power issues, for instance by using cloud based services. Cloud based services could make it possible to use the processing power of multiple servers at the same time.
- The current simulation model interface does not show what happens within the model. An extra model could be attached to the ITT simulation model which visualizes the interactions of the ITT system. It could show information like where the vehicles are in the system and what their statuses are.

Bibliography

[1] http://blomdsw.nl/nieuws/daf/het-maasvlakte-vraagstuk.

[2] http://www.arenasimulation.com/.

[3] http://www.porttechnology.org.

[4] Tuan Le Anh. *Intelligent Control of Vehicle-Based Internal Transport Systems.* PhD thesis, Erasmus University Rotterdam, 2005.

[5] Ebru K. Bish, Frank Y. Chen, Yin Thin Leong, Barry L. Nelson, Jonathan Wing, Cheong Ng, and David Simchi-Levi. Dispatching vehicle in a mega container terminal. *OR Spectrum*, 27:491–506, 2005.

[6] Jürgen Böse, Torsten Reiners, Dirk Steenken, and Stefan Voss. Vehicle dispatching at seaport container terminals using evolutionary algorithms. *Proceedings of the 33rd Hawaii International Conference on System Sciences*, pages 1–10, 2000.

[7] Beth C. Carpenter and Thomas Ward. The use of computer simulation for marine terminal planning. *Proceedings of the 1990 Winter Simulation Conference*, pages 802–804, 1990.

[8] Yong-Leong Cheng, Hock-Chan Sen, and Karthik Natarajan. Dispatching automated guided vehicles in a container terminal. Technical report, Singapore-MIT Alliance Program, 2003.

[9] K.P. Dahal, S.J. Galloway, G.M. Burt, J.R. McDonald, and I. Hopkins. A case study of process facility optimization using discrete event simulation and genetic algorithm. *GECCO05*, pages 2197–2198, 2005.

[10] Willemien Diekman and Jan Willem Koeman. Gesloten transport route: Inter terminal transport op Maasvlakte 1 en 2. 2010.

[11] E.W. Dijkstra. *Co-operating sequential processes.* Academic Press, New York, 1968.

[12] Mark B. Duinkerken, Rommert Dekker, Stef T. G. L. Kurstjens, Jaap A. Ottjes, and Nico P. Dellaert. Comparing transportation systems for inter-terminal transport at the Maasvlakte container terminals. *OR Spectrum*, 28:469–493, 2006.

[13] Mark B. Duinkerken, Joseph J.M. Evers, and Jaap A. Ottjes. Traces: Traffic control engineering system. *Proceedings of the 1999 Summer Computer Simulation Conference*, pages 461–465, 1999.

[14] Mark B. Duinkerken, Jaap A. Ottjes, and Gabriel Lodewijks. The application of distributed simulation in TOMAS: Redesigning a complex transportation model. *PProceedings of the 2002 Winter Simulation Conference*, pages 1207–1213, 2002.

[15] M.B. Duinkerken, J.A. Ottjes, J.J.M. Evers, S.T.G.L. Kurstjens, R.Dekker, and N.P. Dellaert. Simulation studies on inter terminal transport at the Maasvlakte. Technical report, TRAIL Research School, 1996.

[16] E.H. Durfee and V.R. Lesser. Negotiating task decomposition and allocaltion using partial global planning. *Distributed Artificial Intelligence*, 2:229–224, 1989.

[17] Pius J. Egbelu and J.M.A. Tanchoco. Characterization of automated guided vehicle dispatching rules. *International Journal of Production Research*, 22:359–374, 1984.

[18] Pius J. Egbelu and J.M.A. Tanchoco. Potential for bi-directional guide-path for automated guided vehicle based systems. *International Journal of Production Research*, 24:1075–1097, 1986.

[19] Joseph J.M. Evers and Stijn A.J. Koppers. Automated guided vehicle traffic control at a container terminal. *Transportation Research Part A: Policy and Practice*, 30:21–34, 1996.

[20] Jean Philippe Gagliardi, Jacques Renaud, and Angel Ruiz. A simulation model to improve warehouse operations. *Proceedings of the 2007 Winter Simulation Conference*, pages 2012–2018, 2007.

[21] Luca Maria Gambardella, Andrea E. Rizzoli, and Marco Zaffalon. Simulation and planning of an intermodal container terminal. *SIMULATION on Harbour and Maritime Simulation*, 71:107–116, 1998.

[22] Anita Gudelj, Danko Kezic, and Stjepan Vidacic. Planning and optimization of agv jobs by petri net and genetic algorithm. *JIOS*, 36:99–122, 2012.

[23] L. Henesey, P. Davidsson, and J.A. Persson. Comparison and evaluation of two automated guided vehicle systems in the transhipment of containers at a container terminal. *Proceedings of the 5th International Conference on Computer Applications and Information Technology in the Maritime Industries*, 5:54–64, 2006.

[24] L. Henesey, P. Davidsson, and J.A. Persson. Evaluation of automated guided vehicle systems for container terminals using multi agent based simulation. *Multi-Agent-Based Simulation*, 9:85 – 96, 2009.

[25] J.W. Hooper. Strategy-related characteristics of discrete-event languages and models. *Simulation*, 46:153–159, 1986.

[26] Incomaas. Report definition phase. Technical report, 1994.

[27] Rick Jansen. Determining the cost savings for the participants in a joint inter terminal transport system at the Port of Rotterdam. Master's thesis, Erasmus University Rotterdam, 2013.

[28] N. Jawahar, P. Aravindan, S.G. Ponnambalam, and R.K. Suresh. Agv schedule integrated with production in flexible manufacturing systems. *The International Journal of Advanced Manufacturing Technology*, 14:428–440, 1998.

[29] Kap Hwan Kim and Young-Man Park. A crane scheduling method for port container terminals. *European Journal of Operation Research*, 156:752–768, 2004.

[30] Hoai Minh Le, Adnan Yassine, and Riadh Moussi. Dca for solving the scheduling of lifing vehicle in an automated port container terminal. *Computational Management Science*, 9:273–286, 2012.

[31] J. Lee, M. Tangjarukij, and Z. Zhu. Load selection of automated guided vehicles in flexible manufacturing systems. *International Journal of Production Research*, 12:3383–3400, 1996.

[32] B. Li, W. Li, Y. Zhang, Y. Ge, H. Chen, and X. Liang. Modeling and simulation of yard trailer dispatching at container terminals. *Proceedings of the IEEE International Conference on Automation and Logistics*, pages 29–34, 2009.

[33] Chin-I Liu, Hossein Jula, Katarina Vukadinovic, and Petros Ioannou. Automated guided vehicle system for two container yard layouts. *Transport Research Part C*, 12:349–368, 2004.

[34] Sainan Liu. Modeling and optimization of integrated scheduling of automated warehouse system. *Advanced Materials Research*, 230-232:35–39, 2011.

[35] Tamás Máhr. *Vehicle Routing under Uncertainty.* PhD thesis, Technische Universiteit Delft, 2011.

[36] Rajesh Mansharamani. An overview of discrete event simulation methodologies and implementation. *Sadhana*, 22:611–627, 1997.

[37] Monaldo Mastrolilli, Nicoletta Fornara, Luca Maria Gambardella, Andrea E. Rizzoli, and Marco Zaffalon. Simulation for policy evaluation, planning and decision support in an intermodal container terminal. *Society for Computer Science International*, pages 33–38, 1998.

[38] Patrick Johannes Maria Meersman. *Optimization of Container Handling Systems.* PhD thesis, Erasmus University Rotterdam, 2002.

[39] Richard E. Nance. A history of discrete event simulation programming languages. Technical report, Virginia Polytechnic Institute and State University, 1993.

[40] W.C. Ng and K.L. Mak. Yard crane scheduling in port container terminals. *Applied Mathematical Modelling*, 29:263–276, 2005.

[41] F.E. Nieuwkoop. Determining itt configurations on the Maasvlakte by integer programming. Master's thesis, Delft University of Technology, 2013.

[42] J.A. Ottjes, J.J.M. Evers, R.Dekker, N.P. Dellaert, M.B. Duinkerken, S.T.G.L. Kurstjens, A.H. Mey, and H. Borsje. Simulatiestudie inter terminal transport Maasvlakte. Technical report, TRAIL Research School, 1996.

[43] Jaap A. Ottjes, Mark B. Duinkerken, Joseph J.M. Evers, and Rommert Dekker. Robotised inter terminal transport of containers. *Proc. 8th European Simulation Symposium*, pages 621–625, 1996.

[44] Jaap A. Ottjes and Hans P.M. Veeke. Prototyping in process oriented modeling and simulation. *Proceedings of the 16th European Simulation Multiconference*, 2002.

[45] Jaap A. Ottjes, Hans P.M. Veeke, Mark B. Duinkerken, Joan C. Rijsenbrij, and Gabriel Lodewijks. Simulation of a multiterminal system for container handling. *OR Spectrum*, 28:447–468, 2006.

[46] Roy I. Peterkofsky and Carlos F. Daganzo. A branch and bound solution method for the crane scheduling problem. *Transportation Research Part B*, 24B:159–172, 1990.

[47] Naifei Ren, Dan Liu, Yan Zhao, and Xiaobing Ge. Agv scheduling optimizing research of collaborative manufacturing. *Applied Mechanics and Materials*, 300-301:55–61, 2013.

[48] Ihsan Sabuncuoglu and Omer Batuhun Kizilisik. Reactive scheduling in a dynamic and stochastic fms environment. *International Journal of Production Research*, 41:4211–4231, 2003.

[49] Herbert Schroër. Development of a simulation model for an inter terminal transport system at the Maasvlakte 1 + 2. Technical report, Tu Delft, 2013.

[50] Mei Sha. A simulation model for intra-terminal transport of container terminal operations system. *IEEE International Conference on Service Operations and Logistics, and Informatics*, 2:2810–2814, 2008.

[51] K.P. Sycara. Multiagent systems. *AI Magazine*, 19:79–92, 1998.

[52] Kevin Tierney, Stefan Voss, and Robert Stahlbock. A mathematical model of inter-terminal transportation. *European Journal of Operation Research*, 2012.

[53] Marco C. van Burgsteden and Mark Hullegie. Modmodel road traffic on airport premisis. *Proceedings of the 2000 Winter Simulation Conference*, pages 1154–1163, 2000.

[54] Jaap Robert van der Meer. *Operational Control of Internal Transport.* PhD thesis, Erasmus University Rotterdam, 2000.

[55] J.R. van der Meer, R. de Koster, P. Meersman, and R. Dekker. Famas/newcon. 1999.

[56] M. van Schuylenburg, W.a. Stijlen, M.J. Stubenitsky, H.A. Weustenenk, and C. Pons. Maasvlakte integrale container logistiek. Technical report, 2002.

[57] Hans P.M. Veeke and Jaap A. Ottjes. TOMAS: Tool for Object-oriented Modeling And Simulation. *Proceedings of the Business and Industry Simulation Symposium*, pages 76–81, 2000.

[58] Hans P.M. Veeke and Jaap A. Ottjes. Applied distributed discrete process simulation. *Proceedings of the 15th European Simulation Multiconference*, 2001.

[59] Hans P.M. Veeke and Jaap A. Ottjes. Tomasweb: www.tomasweb.com. 2002.

[60] Bharath S. Viadyanathan, David M. Miller, and Young H. Park. Application of discrete event simulation in production scheduling. *Proceedings of the 1998 Winter Simulation Conference*, 965-971, 1998.

[61] M. Woolridge. *An Introduction to MultiAgent System.* John Wiley & Sons, 2002.

[62] Zhaojie Xue, Canrong Zhang, Lixin Miao, and Wei-Hua Lin. An ant colony algorithm for yard truck scheduling and yard location assignment problem with precedence constraints. *Journal of Systems Science and Systems Engineering*, 22:021–037, 2013.

[63] Chang Ho Yang, Yong Seok Choi, and Tae Young Ha. Simulation-based performance evaluation of transport vvehicle at automated container terminals. *OR Spectrum*, 26:149–170, 2004.

[64] M. Ye, S. Wang, and C. Yun. A dispatching method for trucks at container terminal by using fuzzy-cnp concept. *Proceedings of the 2010 International Conference on Logistics Engineering and Intelligent Transportation Systems*, pages 1 – 4, 2010.

[65] Bernard P. Zeigler, Herbert Praehofer, and Tag Gon Kim. *Theory of Modeling and Simulation.* Academic Press, San Diego, 2000.

[66] Kun Zheng, Dunbing Tang, Wenbin Gu, and Min Dai. Distributed control of multi-agv system based on regional control model. *Production Engineering*, 7:433–441, 2013.

Appendix A

Scientific research paper

Evaluation of Inter Terminal Transport Configurations at the Maasvlakte 1 and 2 using Discrete Event Simulation

Herbert Schroër, Francesco Corman, Rudy Negenborn, Gabriel Lodewijks

Delft University of Technology

Abstract

The Port of Rotterdam is investigating a new Inter Terminal Transport (ITT) system for the Maasvlakte 1 + 2 area. This paper presents a discrete event simulation model which can be used to evaluate various possible ITT configurations at the Maasvlakte. The model is the first ITT simulation model to incorporate traffic modeling, which means that delays occurring due to traffic will have an impact on the system's performance. The model is applied to 12 different ITT vehicle configurations, including Automated Guided Vehicles (AGVs), Automated Lifting Vehicles (ALVs), Multi Trailer Systems (MTSs) and a combination of barges and trucks.

I. Introduction

Over the past decades there has been an increasing demand in global containerized transport. Because of this demand the Port of Rotterdam was forced to expand its Maasvlakte 1 with the new Maasvlakte 2. It is expected that in 2040 the combined Maasvlakte 1 + 2 will handle at least 30 million TEU, which is almost four times as much as the entire Port of Rotterdam is handling now [1]. With this rise in container transport and new container terminals being built at the Maasvlakte 2, there will also be a rise in Inter Terminal Transport (ITT). Inter terminal transport is the transport of containers between terminals in a port.

The ITT system for the Maasvlakte is being analyzed within the project "Inter-terminal transport on Maasvlakte 1 and 2 in 2030 - Towards a multidisciplinary and innovative approach on future inter-terminal transport options.". It is a joint project between Delft University of Technology, Erasmus University Rotterdam and the Port of Rotterdam Authority. The goal of the project is to develop innovative, non-conventional concepts for ITT for the port of Rotterdam. Within this project, expected transport demand scenarios for 2030 have been defined by Rick Jansen [3]. An integer programming model was used by Frans Nieuwkoop [4] to find rough estimations of the optimal transport configurations for the given transport demand scenarios. The question that remains is "Which of the defined ITT vehicle configurations is the best configuration seen from an operational perspective?".

In order to find out how well the configurations perform, a discrete event simulation model for an Inter Terminal Transport system at the Maasvlakte 1 and 2 has been developed. The model makes it possible to evaluate all ITT vehicle configurations defined by Frans Nieuwkoop [4].

II. Model input and output

I. Model input

The input of the simulation model consists of 3 parts: the Maasvlakte infrastructure, the transport demand and the ITT vehicle configurations. The Maasvlakte infrastructure consists of 2 traffic networks, a road network and a water network, which connect a total of 18 container terminals and service centers. A map of the terminals and service providers and the roads between them on which the ITT will take place is shown in Figure 1. Although the simulation model is used in this research for the Maasvlakte area, it can be used for any possible ITT system by simply changing the network maps.

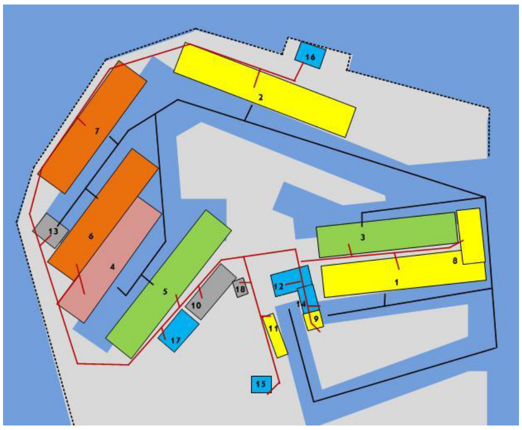

Figure 1: *Map of the Maasvlakte [3]*

The transport demand input consists of 3 different scenarios which have been determined by Rick Jansen [3]. The scenarios are predictions for 2030 and consist of an annual transport demand of respectively 3.340.000, 2.150.000 and 1.420.000 TEU.

A total of 4 different vehicle configurations per scenario has to be evaluated. The configurations are: a number of AGVs, a number of ALVs, a number of MTSs and a combination of barges and trucks. The barges are not able to operate on their own because they are not able to reach every terminal in the system.

II. Performance indicators

By far the most import task of the ITT system is to deliver the containers to their destination in time. In order to measure to what extend the system is able to perform this task, the performance indicator "non-performance" is used. If a container is delivered too late it is accounted as non-performance. This method of registering non-performance is conform to the method used by Tierney *et al.* [6] and Nieuwkoop [4]. Non-performance is the key performance indicator of the ITT system and will show the percentage of containers that has not been delivered in time. Other important performance indicators include the occupation rates of the vehicles and the terminal equipment, vehicle waiting times at the terminals, the number of idle vehicles and the total distance traveled by the vehicles.

III. Simulation model

Because of the discrete nature of the ITT system, the simulation model also needs to be discrete. Therefore a discrete event simulation model was developed using Delphi and the object oriented simulation tools provided by TOMAS. A number of dispatching rules is built into the system which decide on matters like choosing the modality with which to transport a container when barges are used and requesting empty vehicles from other terminals to transport a container.

The simulation model is simulated at container level and it is object-oriented. It consists of the following objects: Containers, a Generator, an UrgencyCheck, Roads, Intersections, Terminals, Terminal Controls, Nodes, Terminal Equipment, Vehicles, Quay Cranes and Barges. The Containers, Roads and Nodes do not have a process and are therefore passive. All other objects are active. The vehicles (AGVs, ALVs, MTSs and Trucks) and barges travel through the system over a network of nodes and arcs. The nodes represent the terminals and intersections and the arcs represent the roads. The vehicles and barges both have a separate network. They use the Dijkstra algorithm to plan their path across the networks. Each terminal has its own control system which is able to request empty vehicles from other terminals to transport a container when no vehicles are available at the terminal itself. It is also used for the MTS scenarios to assign the terminal tractor part of the MTS to a trailer.

Figure 2 shows a schematic representation of the physical objects in the model. A Terminal consists of a number of Terminal Equipment and a Container stack. Vehicles drive between the Terminals, where they are loaded or unloaded. The Vehicles drive over a network of Roads and Intersections to reach their destination. The Barges use a separate network of waterways which is connected to all Terminals with waterside operations.

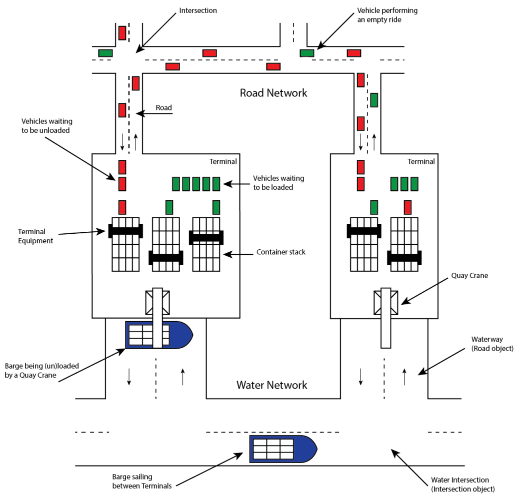

Figure 2: *Schematic representation of the modeled objects*

Unlike previous built ITT simulation models [2] [5], the new simulation model has a built in traffic modeling system. Vehicles can experience delays at the intersections in the system. Each intersection decides which vehicle is allowed to cross the intersection first. Two different algorithms can be used to decide which vehicle to choose: a simple First-In-First-Out algorithm and a more advanced priority algorithm which considers container priority, wether vehicles are going in the same direction and wether they are able to cross at the same time without conflicts. Vehicles can also experience delays at crossings with rail or public road. These crossing have a traffic light that can be set to red or green for certain periods of time, simulating for instance passing trains.

IV. Results & conclusions

A number of experiments has been performed to evaluate the ITT configurations defined by Frans Nieuwkoop [4] and to gain more insight into the working of the ITT system. The non-performance values for the 12 ITT configurations have been given in Table 1. The ITT configurations are the results of the integer programming model, which means that these should have a non-performance of roughly 0 % in that model. As can be seen in the table, this is not the case for the simulation model.

Scenario	Configuration	Non-performance [%]
1	51 ALVs	18,3
	65 AGVs	41,5
	16 MTSs	40,7
	41 Trucks + 2 Barges	98,6
2	33 ALVs	11,2
	42 AGVs	39,4
	12 MTSs	26,7
	22 Trucks + 3 Barges	98,5
3	24 ALVs	2,5
	32 AGVs	21,7
	9 MTSs	19,3
	17 Trucks + 2 Barges	98,7

Table 1: *Non-performance for the various ITT configurations*

By far the most important performance indicators are how many containers are delivered in time and how much too late they have been delivered. Therefore the choice of the best ITT configuration will only be based on the non-performance and the average time that containers have been delivered too late.

Since the ALV configurations have by far the lowest non-performance and lateness values for each of the 3 scenarios, the ALV configurations are the best configurations.

However, this can only be concluded under the currently used dispatching rules and vehicle properties. Results have shown that the vehicle speed has a big influence on the system performance, which can be explained by the large distances in the ITT system. Vehicles spend most of their time driving. In the experiments the speed of the AGV and ALV have both been set to the same value, although the current ALVs are a bit slower than the current AGVs. This difference in speed might actually make the AGVs perform better than the ALVs. Also adding a proper planning system might make the less flexible configurations perform better than they do now.

The barge configurations score by far the worst for all 3 scenarios. The main reason these configurations score so poorly is due to the way they were modeled in Nieuwkoop's integer programming model. The barges were modeled continuous, instead of integer, because of memory issues. The result of this is that each container can be transported separately by a segment of a barge, without having to wait until a barge is full. This makes a barge of 50 TEU more or less work as a set of 1 or 2 TEU trucks, which are all used in an optimal way. In reality it does not work as efficient and flexible as this.

Barges do not seem to be a good option to be used in the ITT system. Handling them takes too much time; mooring alone already takes about an hour per visited terminal. Because of their large capacity, the large number of different terminals and the relatively short allowed delivery time of the containers, it is very hard to optimally use their capacity. The only way they might work is when they sail between terminals that share a lot of containers that allow a long delivery time.

The simulation model developed during this research is not the first ITT simulation model that has been developed, but it is the first model which incorporates traffic modeling into the ITT system. It is the first ITT simulation model where delays occurring due to traffic have an impact on the system's performance. The model is not only able to simulate delays within the system, but also delays due to crossings with rail or public road. The Port of Rotterdam expects that traffic delays will be a major problem for the ITT system. This research has provided a tool which is able to analyze this problem in detail.

V. Future research

This research has provided a discrete event simulation model for an Inter Terminal Transport system. The model could be used to evaluate different operational aspects of the system. Also, several expansions and improvement could be thought of. The following researches are recommended:

- Because of memory issues, Nieuwkoop's integer programming model was run with 5 bundled terminals and the barge and MTS configurations were solved continuous instead of integer. It should be investigated how to solve all configurations integer with 18 terminals. This might result in a better approximation of the optimal number of MTSs and Barges required.
- New transport demand scenarios for 2013 are being constructed. Also, vehicle speeds used in the current simulation might be overestimated. Therefore, both the integer programming model and the simulation model should be rerun using the updated values. Lower vehicle speeds will likely result in a much higher number of required vehicles, which will likely results in more congestion problems.
- Use the simulation model to find the number of vehicles required to get a non-performance below 1% (or 0.5%, or 0.1%, etc.) for all 12 instances.
- The simulation model could be used to investigate the influence of crossings with rail or public road. These crossings were not included in the evaluation of ITT configurations, but experiments showed that these crossings could have a big influence of the performance of the system. More research is needed in finding out where these intersections exactly are, what time delays they would cause, and what the effect of this would be on the system.

- The simulation model could be expanded in order to be able to evaluate the asset light configurations defined as part of the "asset light configuration" task of the "Inter-terminal transport on Maasvlakte 1 and 2 in 2030" project. These configurations contain a variable amount of vehicles that operate in the ITT system. A new 'vehicle generator' object would have to be added to the simulation model which is able to add vehicles to and remove vehicles from the ITT system at certain points in time.
- For the ITT configuration evaluation simulation runs a certain strategy was chosen on how to operate the barges. If barges were considered to be a viable option by the Port of Rotterdam , the simulation model could be used to investigate different barge routing strategies.
- The current simulation model is not optimized. Research could be done in how to optimize the simulation model without having processing power issues, for instance by using cloud based services. Cloud based services could make it possible to use the processing power of multiple servers at the same time.
- The current simulation model interface does not show what happens within the model. An extra model could be attached to the ITT simulation model which visualizes the interactions of the ITT system. It could show information like where the vehicles are in the system and what their statuses are.

References

[1] Willemien Diekman and Jan Willem Koeman. Gesloten transport route: Inter terminal transport op Maasvlakte 1 en 2. 2010.

[2] Mark B. Duinkerken, Rommert Dekker, Stef T. G. L. Kurstjens, Jaap A. Ottjes, and Nico P. Dellaert. Comparing transportation systems for inter-terminal transport at the Maasvlakte container terminals. *OR Spectrum*, 28:469–493, 2006.

[3] Rick Jansen. Determining the cost savings for the participants in a joint inter terminal transport system at the Port of Rotterdam. Master's thesis, Erasmus University Rotterdam, 2013.

[4] F.E. Nieuwkoop. Determining itt configurations on the Maasvlakte by integer programming. Master's thesis, Delft University of Technology, 2013.

[5] Jaap A. Ottjes, Hans P.M. Veeke, Mark B. Duinkerken, Joan C. Rijsenbrij, and Gabriel Lodewijks. Simulation of a multiterminal system for container handling. *OR Spectrum*, 28:447–468, 2006.

[6] Kevin Tierney, Stefan Voss, and Robert Stahlbock. A mathematical model of inter-terminal transportation. *European Journal of Operation Research*, 2012.

Appendix B

Input files

This appendix gives examples of all input files that are required to run the ITT simulation model. These input files can be altered in order to evaluate different scenarios and configurations.

B.1 Configuration file

The general simulation parameters are set by specifying them in a configuration file called *configfile.txt*. An example of the file is shown in Figure B.1.

```
{Configuration file for the ITT Simulation model}
ALV {VehicleType; AGV, ALV, Truck or MTS}
24 {no. of vehicles}
1680 {Runtime in hours}
NO {Use barges?: YES or NO}
2 {No. of Barges}
3 {DepartureInterval for barges in hours}
30 {MooringTime for barges in min}
2 {BargeExtraTime in hours}
Priority {Intersection strategy: FIFO or Priority}
OFF {Traffic lights ON or OFF}
2 {Urgency factor}
3 {QTime in seconds}
5 {RequestVehicleTime in hours}
2 {MTSthreshold in hours}
336 {Statistics reset time}
```

Figure B.1: Configuration file: *configfile.txt*

The parameters to be specified are:

- VehicleType — Determines which Vehicle type is used. Must be AGV, ALV, Truck or MTS.
- No. of Vehicles — Determines the total number of Vehicles in the system.
- Runtime — Determines the total simulation time [hours].
- Use barges? — Determines wether the barge system is used. Must be 'YES' or 'NO'.
- DepartureInterval — Determines after how many hours a barge will leave for the next Terminal.
- MooringTime — Mooring time for Barges [min] (exponential distribution).
- BargeExtraTime — Parameter used by Terminal for deciding on transport by water or road.
- Intersection strategy — Determines which priority algorithm is used at Intersections. Must be 'Priority' of 'FIFO'.
- Traffic Lights — Determines if Vehicles can have a red light at crossings with rail or public road. Must be 'ON' of 'OFF'.
- Urgency Factor — Factor used to determine how soon a Container becomes urgent.
- QTime — Time it takes a Vehicle to travel 1 position in a traffic queue [s].
- RequestVehicleTime — Determines how many hours before due time an empty ride is allowed to be generated for a Container [hours].
- MTSthreshold — Determines how many hours before due time a non-full trailer is allowed to be transported by MTS [hours].
- Statistics reset time — System warm up time. All statistics are reset after this time [hours].

B.2 Transport demand input file

The transport demand is defined in the transport demand input file called *inputTransportDemand.txt*. The file consists of a long list of Containers that need to be transported. An example of part of the file is shown in Figure B.2.

```
{Input file for the Transport Demand - Scenario 3}

{ReleaseTime - Origin - Destination - # TEU - DueTime}

 0.0000  1  4 1 4.2003
 0.0000  2 17 2 4.2043
 0.0000  2 11 1 7.8127
 0.0000  4 18 1 12.1130
 0.0000  5 12 2 9.9995
 0.0000  6 11 1 10.4033
 0.0000  7 11 2 11.7227
 0.0000  8 17 2 9.1114
 0.0000  9 11 1 14.8135
 0.0000 10 18 2 7.0010
 0.0000 11  6 2 7.9636
 0.0000 12 16 2 12.7778
 0.0000 13 18 2 5.7680
 0.0000 14  6 2 6.2693
 0.0000 15  1 1 6.8016
 0.0000 16  7 2 8.3289
 0.0000 17  2 2 14.9058
 0.0000 18  1 2 8.0696
 0.0174 16 11 2 5.6656
 0.0203 17 11 2 2.2666
 0.0570 12  5 1 8.4733
 0.0649  4 16 2 14.1841
 0.0710 12 15 2 9.4621
 0.0753 11  4 2 4.1993
 0.0869  1 15 2 11.9147
 0.0988 16 12 2 14.5006
 0.1045 15  2 2 14.0669
 0.1233 12 17 2 4.4764
 0.1251 12 17 1 7.0181
 0.1618 11  6 2 11.4556
 0.1679  2 15 2 5.0626
 0.1680 15  3 1 5.3250
 0.1848  1 11 2 5.9315
 0.1918  2 11 2 5.7574
 0.1920  6 17 1 11.8334
 0.2089  6 12 2 6.8445
 0.2171 17  1 2 7.0963
 0.2375 16  7 2 12.2174
 0.2489  4 11 2 8.6648
 0.2555 11  3 1 5.7112
```

Figure B.2: Transport demand input file: *inputTransportDemand.txt*

For each Container the following parameters have to be specified:

- ReleaseTime Time the Container enters the system.
- Origin Name of the origin Terminal.
- Destination Name of the Destination Terminal.
- # TEU The type of Container; 1 or 2 TEU.
- DueTime Final time the Container is allowed to be delivered at its destination.

The files have been generated using an Arena [2] based demand generator created by Rick Jansen [27].

B.3 Terminal input file

The Terminals in the system are defined in the terminal input file called *inputTerminals.txt*. An example of the file is shown in Figure B.3.

```
{Input file containing properties of the terminals}

18 {total number of Terminals}

{Equipment type  = ASC,RS or SC}

{TerminalID - Equipment Type - No. of TEq - No. of QC}
1 ASC 4 1
2 ASC 4 1
3 ASC 3 1
4 ASC 3 1
5 ASC 3 1
6 ASC 3 1
7 ASC 3 1
8 SC 2 1
9 RS 2 1
10 SC 5 0
11 SC 6 1
12 RS 4 1
13 SC 4 1
14 RS 3 1
15 RS 3 0
16 RS 3 0
17 RS 4 0
18 RS 2 0
```

Figure B.3: Terminal input file: *inputTerminals.txt*

For each Terminal the following parameters have to be specified:

- Terminal ID Name of the Terminal.
- Equipment Type Type of equipment working at the Terminal. Must be 'ASC', 'RS' or 'SC'.
- No. of TEq Number of equipment working at the Terminal.
- No. of QC Number of Quay Cranes working at the Terminal. If 0: Terminal is not a Barge Terminal.

B.4 Terminal Equipment input file

The Terminal Equipment (and the Quay Crane) properties are defined in the Terminal Equipment input file called *inputTerminalEquipment.txt*. Four different types of equipment are defined: Automatic Stacking Cranes, Straddle Carriers, Reachstackers and Quay Cranes. An example of the file is shown in Figure B.4.

For each type of equipment the following parameters have to be specified:

- Unloadtime Average time it takes to unload a Container (from a Vehicle) [min] (exponential distribution).
- Loadtime Average time it takes to load a Container (onto a Vehicle) [min] (exponential distribution).

```
{Input file containing properties of the terminal equipment}

{Automatic Stacking Crane}

ASC {Equipment Type}
3 {Unloadtime in min}
3 {Loadtime in min}

{Straddle Carrier}

SC {Equipment Type}
4 {Unloadtime in min}
4 {Loadtime in min}

{Reachstacker}

RS {Equipment Type}
4 {Unloadtime in min}
4 {Loadtime in min}

{QuayCrane}

QC {Equipment Type}
3 {Unloadtime in min}
3 {Loadtime in min}
```

Figure B.4: Terminal Equipment input file: *inputTerminalEquipment.txt*

B.5 Vehicle input file

The Vehicle properties are defined in the Vehicle input file called *inputVehicles.txt*. Five different types of Vehicles are defined: Trucks, AGVs, ALVs, MTSs and Barges. An example of the file is shown in Figure B.5.

```
{Input file containing properties of the vehicles}

{Truck}

Truck {Vehicle Type}
40 {Vehicle speed km/h}
1 {ClearTimeFactor}
2 {Capacity in TEU}

{AGV}

AGV {Vehicle Type}
40 {Vehicle speed km/h}
1 {ClearTimeFactor}
2 {Capacity in TEU}

{ALV}

ALV {Vehicle Type}
40 {Vehicle speed km/h}
1 {ClearTimeFactor}
0,5 {Container loadtime}
0,5 {Container unloadtime}
2 {Capacity in TEU}

{MTS}

MTS {Vehicle Type}
30 {Vehicle speed km/h}
2,67 {ClearTimeFactor}
10 {Capacity in TEU}
0,5 {MTS coupling time in min}

{Barge}

Barge {Vehicle Type}
12 {Vehicle speed km/h}
50 {Capacity in TEU}
```

Figure B.5: Vehicle input file: *inputVehicles.txt*

The following parameters have to be specified:

- Vehicle speed — Average speed [km/h].
- ClearTimeFactor — Factor used to determine how long it takes a Vehicle to clear an Intersection.
- Capacity — The number of TEU a Vehicle is ably to carry.
- Container LoadTime — Time it takes an ALV to lift 2 TEU from a platform.
- Container UnloadTime — Time it takes an ALV to lift 2 TEU onto a platform.
- MTS coupling time — Time it takes a MTS to (un)couple a trailer.

B.6 Intersection input file

The Intersections are defined in the Intersection input file called *inputIntersections.txt*. An example of the file is shown in Figure B.6.

```
{Input file containing properties of the different intersections}
31 {total number of intersections}

{IntersectionID - TimeToCross (s) - Type - GreenLightTime (min) - RedLightTime (min)}
I01 7,5 3 0 0
I02 0 2 18 2
I03 7,5 3 0 0
I04 7,5 3 0 0
I05 7,5 3 0 0
I06 0 2 18 2
I07 7,5 3 0 0
I08 0 2 18 2
I09 7,5 3 0 0
I10 0 2 18 2
I11 7,5 3 0 0
I12 7,5 3 0 0
I13 0 2 18 2
I14 7,5 3 0 0
I15 7,5 3 0 0
I16 7,5 3 0 0
I17 7,5 3 0 0
I18 7,5 3 0 0
I19 7,5 3 0 0
I20 7,5 3 0 0
I21 7,5 3 0 0
WI01 0 1 0 0
WI02 0 1 0 0
WI03 0 1 0 0
WI04 0 1 0 0
WI05 0 1 0 0
WI06 0 1 0 0
WI07 0 1 0 0
WI08 0 1 0 0
WI09 0 1 0 0
WI10 0 1 0 0
```

Figure B.6: Intersection input file: *inputIntersections.txt*

For each Intersection the following parameters have to be specified:

- Intersection ID — Name of the Intersection.
- TimeToCross — Time it takes a Vehicle with a ClearFactor of 1 to cross the Intersection [s].
- Type — Type of Intersection (see Section 4.10) Must be 1, 2, 3 or 4.
- GreenLightTime — Average time Vehicles are able to cross until next red light [min]. Only for type 2 (exponential distribution).
- RedLightTime — Average time Vehicles have to wait until next green light [min]. Only for type 2 (exponential distribution).

The Intersection ID for the Intersections in the water network have to start with 'W'. Based on their ID the model adds them to a subset of Water Intersections.

B.7 Road input file

The Roads are defined in the Road input file called *inputRoads.txt*. An example of part of the file is shown in Figure B.7.

```
{Input file containing properties of the different roads}

120 {total number of roads}

{RoadID - Length [m] - Start node - Start orientation - end node - end orientation}

R01a 400 16 0 I01 E
R01b 400 I01 E 16 0
R02a 10 2 0 I01 S
R02b 10 I01 S 2 0
R03a 1300 I01 W I02 E
R03b 1300 I02 E I01 W
R04a 2600 I02 W I03 N
R04b 2600 I03 N I02 W
R05a 10 7 0 I03 E
R05b 10 I03 E 7 0
R06a 2500 I03 S I04 N
R06b 2500 I04 N I03 S
R07a 10 13 0 I04 E
R07b 10 I04 E 13 0
R08a 500 I04 S I05 N
R08b 500 I05 N I04 S
R09a 10 6 0 I05 E
R09b 10 I05 E 6 0
R10a 150 I05 S I06 W
R10b 150 I06 W I05 S
R11a 150 I06 E I07 W
R11a 150 I07 W I06 E
R12a 10 4 0 I07 N
R12b 10 I07 N 4 0
R13a 2000 I07 E I08 W
R13b 2000 I08 W I07 E
R14a 800 I08 E I09 W
R14b 800 I09 W I08 E
R15a 10 I09 S 17 0
R15b 10 17 0 I09 S
R16a 250 I09 E I10 W
R16b 250 I10 W I09 E
R17a 250 I10 E I11 W
R17b 250 I11 W I10 E
R18a 10 5 0 I11 N
R18b 10 I11 N 5 0
```

Figure B.7: Road input file: *inputRoads.txt*

For each Road the following parameters have to be specified:

• Road ID	Name of the Road.
• Length	Length of the Road [m].
• Start Node	Node where the Road begins.
• Start orientation	If Node is an Intersection: side of the Intersection it's connected to. Must be 'N', 'E', 'S' or 'W'.
• End Node	Node where the Road ends.
• End orientation	If Node is an Intersection: side of the Intersection it's connected to. Must be 'N', 'E', 'S' or 'W'.

The Road ID for the Roads in the water network have to start with 'W'. Based on their ID the model adds them to a subset of Waterways.

B.8 Barge route input file

The route that the Barges have to sail can be defined in the input file called *BargeRoute.txt*. The file consists of a list of Terminals that the Barges will sail to. The Barge will always sail to the Terminal that's below its own location in the list. If it's at the location at the bottom of the list it will start from the top again. An example of the file is shown in Figure B.8.

```
{Barge Route}

9
1
11
12
14
8
3
2
7
13
6
4
5
EndOfFile
```

Figure B.8: Barge route input file: *BargeRoute.txt*

Appendix C

Model output

This appendix gives an overview of the output of the ITT simulation model.

C.1 Interface

The interface of the ITT simulation model is shown in Figure C.1. The picture on the left shows the interface before starting the simulation. The simulation will begin after pressing the *Start* button. The interface will then show some of the general simulation parameters that have been defined in the configuration file (see Appendix B). Also it will start showing the current values of some of the main performance indicators. The simulation can be paused by clicking the *Pause* button. After the simulation is finished the interface will look like the picture on the right in Figure C.1. After clicking the *Quit* button, the programm will be closed and the results of the simulation will be saved away to various output files.

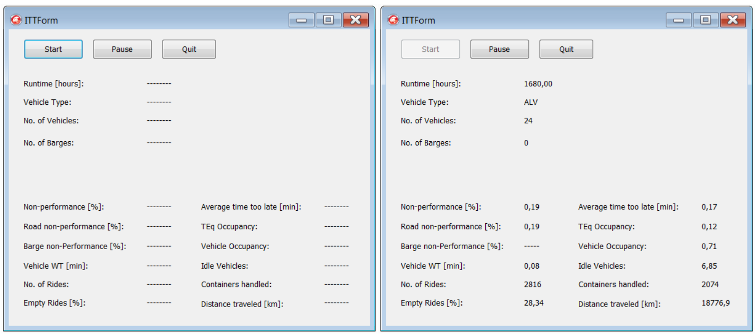

Figure C.1: Interface of the ITT simulation model; Left: before simulation; Right: after simulation

C.2 Output files

At the end of the simulation, after clicking the *Quit* button, the simulation results are saved to the various output files.

C.2.1 General output

The General results are saved to the file *outputGeneral.txt*. An example of the file is shown in Figure C.2. The file first shows the time the simulation was performed and the most important input parameters defined in the configuration file. After that it shows the results of the simulation.

The following results are shown:

```
General outputfile for the ITT system

Generated on 13-11-2013
             1:35:45

Run settings:
Runtime [hours]: 1680
Vehicle Type: Truck
No of Vehicles: 30
Barge system used?: YES
No of Barges: 2
Barge departure interval [hours]: 3
ContBargeLimit [hours]: 0
Intersection strategy: Priority
Traffic lights: OFF
Look ahead time [hours]: 5

Results:

Non-Performance [%]:          25.421
Average time too late [min]:          106.43
Non-Performance Road [%]:             26.382
Non-Performance Barge [%]:             0.034
Mean TEq Occupancy:           0.17
Mean QC Occupancy:            0.06
Mean Vehicle Occupancy:               0.96
Mean Idle Vehicle waiting time [hour]:        0.01
Mean Loaded Vehicle waiting time [hour]:              0.00
Mean no. of idle vehicles:            1.25
Total no. of rides:     162638
No. of empty rides:     21560
Percentage of empty rides:           13.26
Total no. of containers created:        161634
Mean no. of containers created per hour:              96.21
No. of containers created in peak hour: 137
Mean no. of containers handled per hour:              96.22
Total no. of containers handled:        161650
Containers handled via Road:    155755
Containers handled via Barge:   5895
Mean Vehicle loading rate [%]:       81.43
Mean Barge loading rate [%]:         15.47
Total distance traveled by Vehicles[km]:        1122952.68
Total distance traveled empty by Vehicles [km]:  130297.49
Total distance traveled by Barges [km]:    7155.80
Total delay due to traffic [hour]:          105.53
Mean delay due to traffic per ride [s]:       4.88
Mean ride time [min]:          11.29
Mean ride distance [km]:              6.90
```

Figure C.2: General output file: *outputGenerators.txt*

- Non-Performance [%]
- Average time too late [min]
- Non-Performance of Containers handled by road [%]
- Non-Performance of Containers handled by Barge [%]
- Mean overall Terminal Equipment occupancy
- Mean overall Vehicle occupancy
- Mean idle Vehicle waiting time [hours]
- Mean loaded Vehicle waiting time [hours]
- Mean number of idle Vehicles
- Total number of rides
- Number of empty rides
- Percentage of empty rides [%]
- Total number of Containers created (result of transport demand input)
- Mean number of Containers created per hour (result of transport demand input)
- Max. number of Containers created in one hour (result of transport demand input)
- Mean number of Containers handled per hour
- Total number of containers handled
- Number of Containers handled by road
- Number of Containers handled by Barge

- Mean Vehicle loading rate [%]
- Mean Barge loading rate [%]
- Total distance traveled by Vehicles [km]
- Total distance traveled empty by Vehicles [km]
- Total distance traveled by Barges [km]
- Total delay due to traffic [hours]
- Mean delay due to traffic per ride [s]
- Mean ride time [min]
- Mean ride distance [km]

C.2.2 Output per Road and Intersection

The results per Road and Intersection are saved to the output file *outputRoadsIntersections.txt*. An example of part of the file is shown in Figure C.3. The file first shows the time the simulation was performed and the most important input parameters defined in the configuration file. After that it shows the results of the simulation.

```
Outputfile for the roads and intersections of the ITT system

Generated on 13-11-2013
             1:35:45

Run settings:
Runtime [hours]: 1680
Vehicle Type: Truck
No of Vehicles: 30
Barge system used?: YES
No of Barges: 2
Barge departure interval [hours]: 3
ContBargeLimit [hours]: 0
Intersection strategy: Priority
Traffic lights: OFF
Look ahead time [hours]: 5

Name | TotalPassed | Mean Veh/h | Max Veh/h | Mean Veh | Max Veh | Total Delay [hour] | Vehicle Delay [s]

I01     48346       28.78      63         0.00      8            2.82             0.21
I02     48282       28.74      62         0.00      1            0.00             0.00
I03     62197       37.02      65         0.00      10           3.77             0.22
I04     60737       36.15      59         0.00      4            2.79             0.17
I05     76697       45.65      65         0.00      12           5.88             0.28
I06     74021       44.06      61         0.00      1            0.00             0.00
I07     89941       53.54      76         0.00      14           7.99             0.32
I08     87198       51.90      72         0.00      1            0.00             0.00
I09     112018      66.68      91         0.00      14          12.90             0.41
I10     98330       58.53      89         0.00      1            0.00             0.00
I11     112461      66.94      99         0.00      13          11.98             0.38
I12     108416      64.53      98         0.00      6            8.78             0.29
I13     108417      64.53      98         0.00      2            0.00             0.00
I14     132656      78.96      119        0.01      5           18.55             0.50
I15     80033       47.64      98         0.00      11           5.63             0.25
I16     77153       45.92      99         0.00      15           7.70             0.36
I17     78613       46.79      74         0.00      5            6.93             0.32
I18     39835       23.71      58         0.00      9            2.22             0.20
I19     10132        6.03      40         0.00      15           0.73             0.26
I20     40957       24.38      50         0.00      8            4.60             0.40
I21     27064       16.11      53         0.00      11           2.27             0.30
WI01    0            0.00      0          0.00      0
WI02    0            0.00      0          0.00      0
WI03    0            0.00      0          0.00      0
WI04    0            0.00      0          0.00      0
WI05    0            0.00      0          0.00      0
WI06    0            0.00      0          0.00      0
WI07    0            0.00      0          0.00      0
WI08    1287         0.77      3          0.00      1
WI09    1287         0.77      3          0.00      1
WI10    1289         0.77      2          0.00      1
R01a    8934         5.32      22         0.05      9
R01b    8933         5.32      22         0.05      7
R02a    15272        9.09      29         0.00      7
R02b    15271        9.09      31         0.00      1
R03a    24142       14.37      31         0.47      13
```

Figure C.3: Part of Road and Intersection output file: *outputRoadsIntersections.txt*

Per Road and Intersection the following results are shown:

- Name of the Road or Intersection
- Total amount of Vehicles passed
- Mean amount of Vehicles per hour
- Maximum amount of Vehicles in an hour, so in the busiest hour
- Mean amount of Vehicles on this Road or Intersection at one point in time
- Maximum amount of Vehicles on this Road or Intersection at one point in time
- Total delay due to traffic [hours] (only for Intersections)
- Mean delay per Vehicle [s] (only for Intersections)

C.2.3 Output per Terminal

The results per Terminal are saved to the output file *outputTerminals.txt*. An example of the file is shown in Figure C.4. The file first shows the time the simulation was performed and the input parameters defined in the configuration file. After that it shows the results of the simulation.

```
Outputfile for the Terminals of the ITT system

Generated on 13-11-2013
             1:35:45

Run settings:
Runtime [hours]: 1680
Vehicle Type: Truck
No of Vehicles: 30
Barge system used?: YES
No of Barges: 2
Barge departure interval [hours]: 3
ContBargeLimit [hours]: 0
Intersection strategy: Priority
Traffic lights: OFF
Look ahead time [hours]: 5

Name | EqType | NoOfEq | EqOccupancy | MeanWT IVehicleQ | MeanWT LVehicleQ | MeanL IVehicleQ | MeanL ContQ | Containers loaded No. + % | Containers unloaded No. + %

1     ASC   4      0.16      0.02      0.00      0.14      3.31      9679     5.21    12496     7.73
2     ASC   4      0.21      0.00      0.00      0.02    102.89      17537    9.45    13067     8.08
3     ASC   3      0.08      0.08      0.00      0.34      0.00      0        0.00    8061      4.99
4     ASC   3      0.17      0.02      0.00      0.10      3.60      7797     4.20    9465      5.86
5     ASC   3      0.16      0.02      0.00      0.11      4.54      7772     4.19    8954      5.54
6     ASC   3      0.17      0.02      0.00      0.10      3.53      7703     4.15    9418      5.83
7     ASC   3      0.14      0.02      0.00      0.08      3.21      6363     3.43    7769      4.81
8     SC    2      0.05      0.04      0.00      0.04      0.74      926      0.50    1277      0.79
9     RS    2      0.02      0.05      0.00      0.02      0.48      394      0.21    470       0.29
10    SC    5      0.00      0.00      0.00      0.00      0.00      0        0.00    0         0.00
11    SC    6      0.39      0.00      0.00      0.02    289.30      36804   19.83    28482    17.62
12    RS    4      0.29      0.00      0.00      0.04     29.13      16413    8.84    15057     9.31
13    SC    4      0.00      0.00      0.00      0.00      0.00      0        0.00    0         0.00
14    RS    3      0.09      0.02      0.00      0.05      3.49      3748     2.02    3409      2.11
15    RS    3      0.21      0.00      0.00      0.05     10.10      8365     4.51    7749      4.79
16    RS    3      0.22      0.00      0.00      0.02     17.44      9417     5.07    8446      5.22
17    RS    4      0.37      0.00      0.00      0.07     35.16      20526   11.06    19397    12.00
18    RS    2      0.09      0.02      0.00      0.04      2.17      2314     1.25    2238      1.38
```

Figure C.4: Terminal output file: *outputTerminals.txt*

Per Terminal the following results are shown:

- Name of the Terminal
- Terminal Equipment type operating at the Terminal
- Number of Terminal Equipment operating at the Terminal
- Mean Terminal Equipment occupancy
- Mean waiting time for Vehicles in MyIdleVehicleQ
- Mean waiting time for Vehicles in MyLoadedVehicleQ
- Mean number of Vehicles in MyIdleVehicleQ
- Mean length of MyContainerQ, so the mean number of Containers waiting to be transported
- Number of Containers loaded onto road Vehicles
- Percentage of Containers loaded onto road Vehicles, of total
- Number of Containers unloaded from road Vehicles
- Percentage of Containers unloaded from road Vehicles, of total

C.2.4 Output for Barge network

The results for the Barge network are saved to the output file *outputBarge.txt*. Results are given per Terminal that is connected to the Barge network. An example of the file is shown in Figure C.5. The file first shows the time the simulation was performed and the input parameters defined in the configuration file. After that it shows the results of the simulation.

```
Outputfile for the Barge system

Generated on 13-11-2013
             1:35:45

Run settings:
Runtime [hours]: 1680
Vehicle Type: Truck
No of Vehicles: 30
No of Barges: 2
ContBargeLimit [hours]: 0
Barge departure interval [hours]: 3
Intersection strategy: Priority
Traffic lights: OFF
Urgency factor: 2

Name | # QC | QCOccupancy | MeanL ContQ | Containers loaded No. + % | Containers unloaded No. + %

1       1           0.39         1.85          2688        1.45     3207         1.98
2       1           0.00         0.00          0           0.00     0            0.00
3       1           0.00         0.00          0           0.00     0            0.00
4       1           0.00         0.00          0           0.00     0            0.00
5       1           0.00         0.00          0           0.00     0            0.00
6       1           0.00         0.00          0           0.00     0            0.00
7       1           0.00         0.00          0           0.00     0            0.00
8       1           0.00         0.00          0           0.00     0            0.00
9       1           0.00         0.00          0           0.00     0            0.00
11      1           0.40         2.23          3205        1.73     2688         1.66
12      1           0.00         0.00          0           0.00     0            0.00
13      1           0.00         0.00          0           0.00     0            0.00
14      1           0.00         0.00          0           0.00     0            0.00
```

Figure C.5: Barge system output file: *outputBarge.txt*

Per Terminal the following results are shown:

- Name of the Terminal
- Number of Quay Cranes operating at the Terminal
- Mean Quay Crane occupancy
- Mean length of MyBargeContainerQ, so the mean number of Containers waiting to be transported by Barge
- Number of Containers loaded onto Barge
- Percentage of Containers loaded onto Barge, of total
- Number of Containers unloaded from Barge
- Percentage of Containers unloaded from Barge, of total

C.3 Graphs

A number of graphs is automatically created using the TOMAS Collections form. The graphs can be opened, with live view, directly on the form, but results are also automatically saved to .csv files so they can later be imported in spreadsheet software like Excel.

Graphs are created for the following output parameters:

- Non-Performance [%]
- Mean idle Vehicle waiting time [hours]
- Number of Containers handled per hour
- Number of Vehicles per hour, per Intersection and Road

At this point graphs are only created for these parameters, but if desired new graphs can easily be added for any system parameter.

Appendix D

ITT Maasvlakte network maps

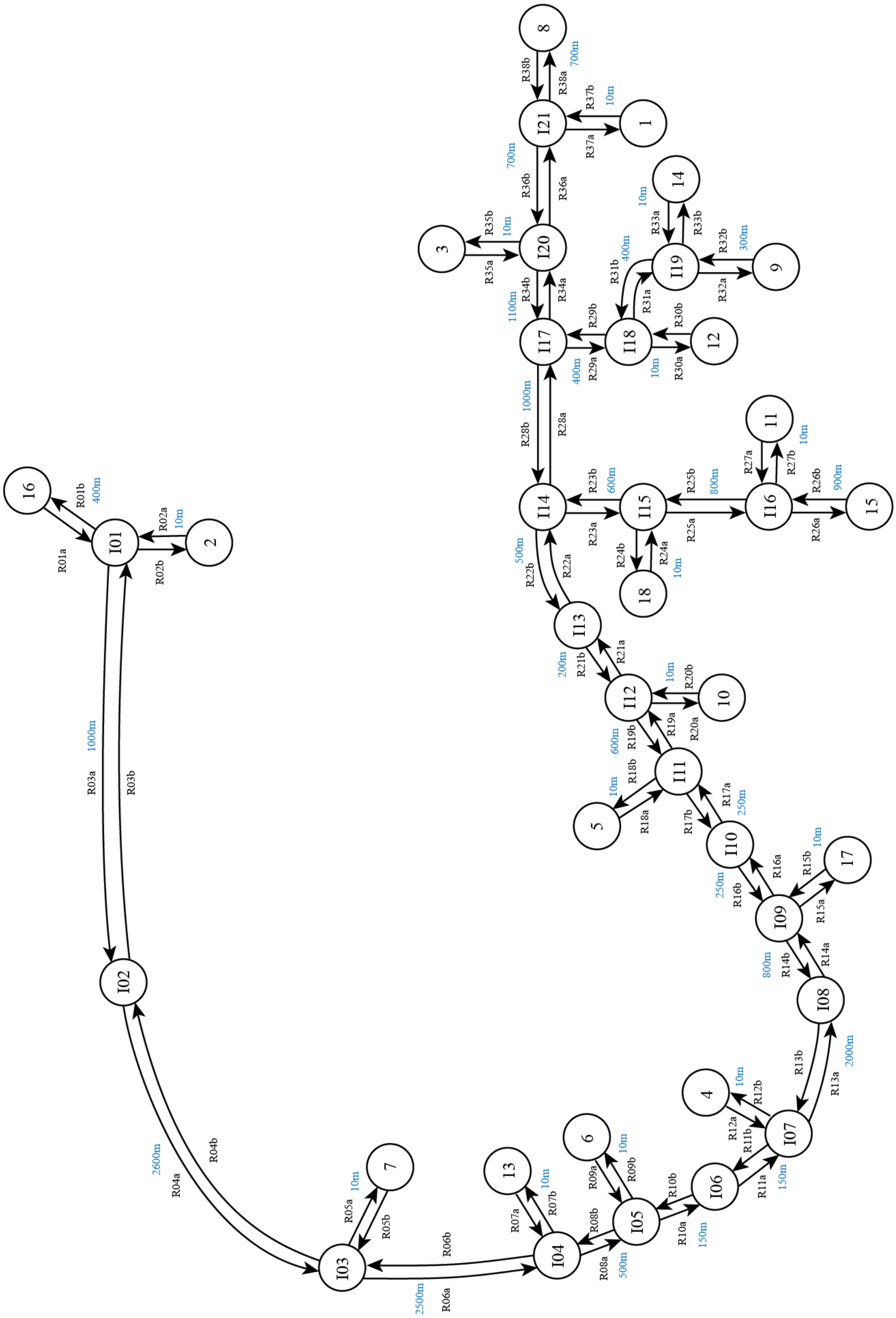

Figure D.1: ITT Maasvlakte Road network

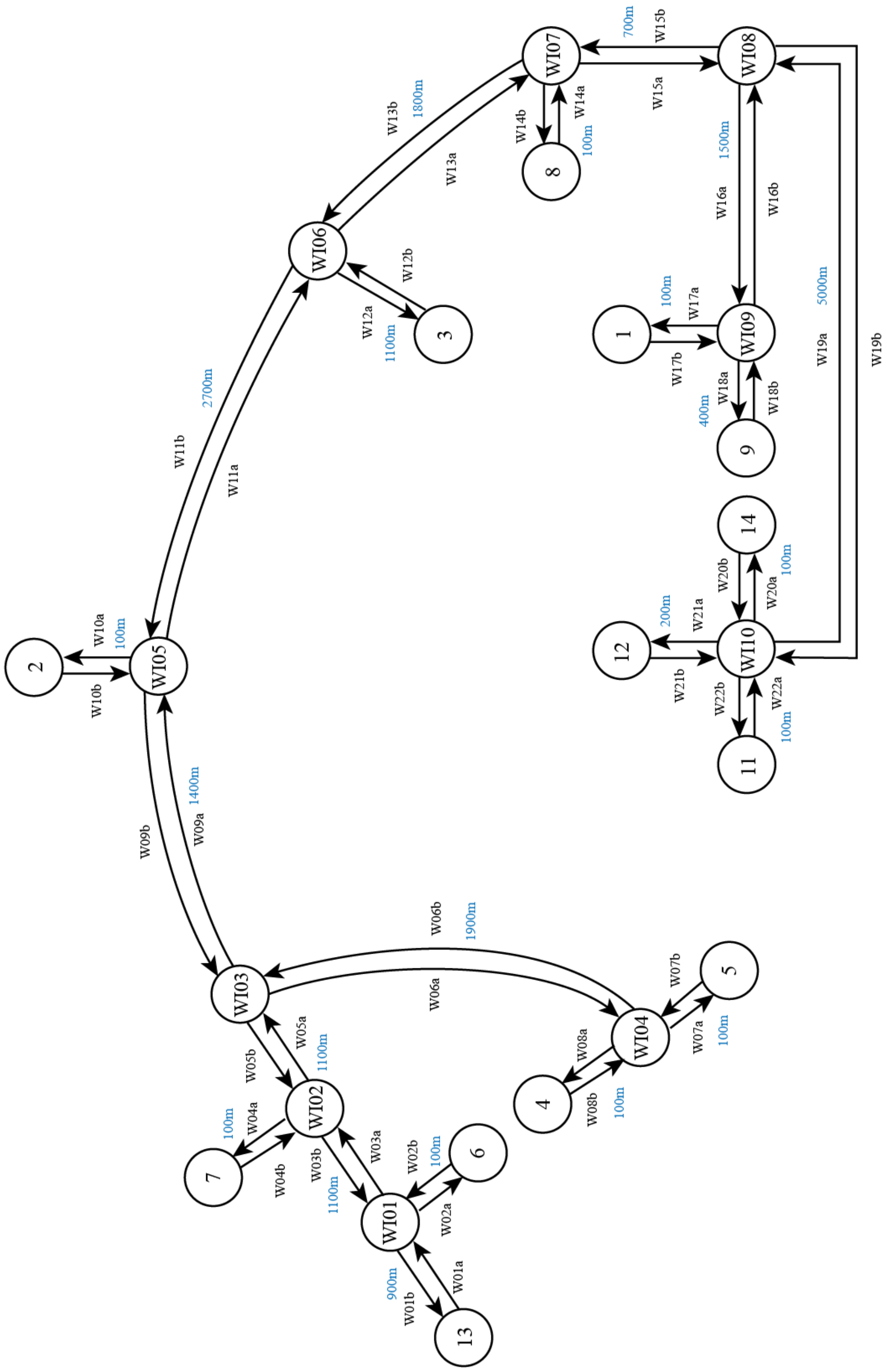

Figure D.2: ITT Maasvlakte Barge network

Appendix E

Container demand scenarios

This appendix holds Chapter 2 from the Master Thesis "Determining the cost savings for the participants in a joint inter terminal transport system at the Port of Rotterdam" by Rick Jansen [27], which explains the 3 transport demand scenarios used as input for the simulation model.

Chapter 2 Container demand scenarios

The demand of containers that will be transported by the ITT system in 2030 is uncertain. In this chapter, three scenarios will be constructed that describe the possible demand of containers. Based on the predictions of the Port Authority, the container flows sizes between the terminals are derived. The containers do not arrive equally over time. Based on data about the current container operations, the size of the peak factor will be determined. The size of the container flow together with the peak factor determine the capacity of the ITT system. Each scenario will be described by an Origin/Destination matrix and a peak factor. The demand scenarios are input for the ITT simulation model.

2.1 Yearly number of containers

The Port of Rotterdam Authority makes predictions about future container transport in the Port Vision 2030 (Port of Rotterdam Authority, 2011b). Based on the predictions, the total container transport is between 19 million TEU in a negative prediction and 31 million TEU in a positive prediction. The development of the Port of Rotterdam as a hub function for other ports in Europa is important for the ITT system. If the Port of Rotterdam becomes a container hub, many transhipment containers are handled in the port area. Larger vessel will stop only at a single terminal in the port area. Containers will be redistributed to other terminals before continuing the next leg of transportation. That will increase the demand for the ITT system.

The scenarios are based on the available handling capacity for ITT in the Port of Rotterdam. All available handling capacity in barge- and rail terminals and empty depots will be used for ITT. In scenario 2, 30% of the barge- and rail terminal and empty depot capacity is used commercially. The commercial use does not generate an ITT movement. When there will be no common barge terminal and common rail terminal, the demand will be according scenario 3. Furthermore the ITT container flows are unbalanced. The imbalance in container movements of the ITT systems is levelled by the normal container movements. The import of full containers exceeds the export of full containers. Empty containers are exported more.

Scenario 1: High demand scenario

The assumptions made for scenario 1 are summed up below. Table 2 shows the amount of containers transported by the ITT system in scenario 1. How the assumptions results in ITT containers flows is explained in Appendix 2: Construction of a scenario.

- The demand of ITT containers between deep sea terminals is taken as 1% of the transshipment containers from the Global Economy scenario of the Port of Rotterdam Authority. The 1% is based on the same assumption as used in the report of (Diekman and Koeman, 2010).
- Deep sea terminals have facilities for X-ray scanning, nuclear detection and physical inspection of containers available within the terminal area. Containers for second-line

scanning are transported to the central customs facility by the ITT system. Only 0.5% of all containers have to visit the central customs facility.

- The transportation of empty containers to and from empty depots can only be done by use of the ITT system.
- The capacity of the common barge and rail terminals is used completely. Every container handling move at common barge or rail terminal generates an ITT movement.
- The capacity of the rail terminals and barge terminals is restricted to 1.756 million and 0.935 million respectively.
- Of the containers that are transported by the ITT system, 35% are empty and 65% are full. Of the empty containers, 35% is import and 65% is export. Of the full containers, 60% is import and 40% is export.
- Empty depots have a capacity of 25,000 containers per hectare.
- 45% of the empty containers are transported by rail and 55% by barge.
- The new ITT system replaces the current ITT system with MTS service at the MV1.

Table 2: ITT container flows in scenario 1 (TEU/year)

To / From	Deep sea terminals	Barge Terminals	Rail Terminals	Customs	Empty depots
Deep sea terminals	94000	425865	628690	155000	266175
Barge terminals	283910	0	0	0	146396
Rail terminals	943035	0	0	0	119779
Customs	155000	0	0	0	0
Empty depots	494325	78829	64496	0	0

Scenario 2: Reduced demand

Scenario 2 (Table 3) takes the same assumptions as scenario 1 but differs at the following points:

- The total transshipment containers is equal to 8 million, which corresponds with the European scenario of the predictions of the Port of Rotterdam Authority (Port of Rotterdam Authority, 2011b).
- The demand for second-line scanning at the central customs facility is equal to 0.25% of all containers handled in the port.
- Commercial parties operate the empty depots at the MV area. Trucks can bring and pick up empty containers next to the ITT system. The number of ITT movements is not equal to the capacity of the empty depot, but equal to 70%. The other capacity will be transported by commercial trucks that do not make use of the ITT system.
- Commercial parties operate also the common barge and rail terminals. Transshipment of containers from short-sea to inland shipping uses handling capacity that cannot be used for ITT. Scenario 2 assumes that only 70% of the handling operations of the barge- and rail terminals generate an ITT movement.

- The import/export ratio is equal to 40%/60% for empty container and 55%/45% for full containers.
- The current ITT system with MTS service stays available.

Table 3: ITT container flows in scenario 2 (TEU/year)

To / From	Deep sea terminals	Barge terminals	Rail terminals	Customs	Empty depots
Deep sea terminals	80000	260876	486801	75000	196560
Barge terminals	213444	0	0	0	108108
Rail terminals	594979	0	0	0	88452
Customs	75000	0	0	0	0
Empty depots	294840	72072	58968	0	0

Scenario 3: No common barge and rail terminal

Scenario 3 (Table 4) takes the same assumptions as scenario 1 and 2 but there will be no common barge and rail terminal at the MV2 area. Intermodal change for low frequent Hinterland connections will be done by the barge terminal and rail terminal at the MV1.

Table 4: ITT container flows in scenario 3 (TEU/year)

To / From	Deep sea terminals	Barge terminals	Rail terminals	Customs	Empty depots
Deep sea terminals	80000	97251	219051	75000	196560
Barge terminals	79569	0	0	0	108108
Rail terminals	267729	0	0	0	88452
Customs	75000	0	0	0	0
Empty depots	294840	72072	58968	0	0

2.2 Container peak factor

The arrival and departure of containers at terminals is not equally distributed over time. The deep sea terminal tries to balance the water side operation. The landside operation of the deep sea terminal has more peaks due to limited operational hours of trucks and trains. In general it can be said that the first trucks arrive around 6 AM and the last trucks leave at 6 PM. Trains are scheduled from Monday afternoon until Saturday afternoon (Van Schuylenburg, 2013). Barges approach the terminal more or less constant over the day. For the design of the ITT system, it is important to know the average and peak flows. Designing the ITT system for average flows leads to under-capacity, while designing for peak flows leads to over-capacity.

Direct data about containers flows from the deep sea terminals is not available in this study. The deep sea terminal balances the arrival of containers at the water side of the terminal. To estimate the peak flows that will be handled by the ITT system, data about the container flows at the landside of the terminals is obtained. The peaks in the number of containers arriving in the ITT system can be caused by the capacity restrictions at the landside at the landside of the deep sea terminal. In the year 2030, the peaks are likely to flatten because trains will run 24 hours a day and the share of trucks will be reduced. Therefore, departure data is gathered about the number of trucks, trains and barges leaving the different terminals at the current Maasvlakte area. The available data about arrivals of trucks, trains and barges will be combined to construct a peak factor (Appendix 1).

Truck data

Direct data about truck arrivals at the gate of the deep sea terminals is not available in this study. But data is available about the number movements on the A15 highway heading towards the Maasvlakte area. The database of Regiolab Delft (Regiolab-Delft) holds data per minute from loop detectors of the A15 highway. The data describes the number of movements, the speed and the direction of the movement. The point of observation is close the Maasvlakte Area to exclude non-container carrying transport as much as possible. The intermodal split between trucks, trains and barges is used to scale the highway movements to truck movements. The average load per truck is assumed to be equal to the TEU factor of 1.7 (Port of Rotterdam Authority, 2012b).

Train data

Train data is obtained from Keyrail about the number of arriving and departing trains from the emplacement Maasvlakte West (Keyrail, 2013). The number of containers per train is gained from the Prorail handbook (ProRail, 2011). The average length of a train is 500, which are 25 railcars that can carry 3 TEU. The maximum number of TEU on a train is 105. An average train is loaded for 90%. Trains can be operated as ideal shuttle or start up shuttle. In ideal shuttle is loaded completely at the origin. A start up shuttle hops between several stops before the train is fully loaded. The assumption is made that due to the shuttle process the trains are loaded for 70% when leaving the Maasvlakte. The average number of TEU per train is equal to 47.

Barge data

Currently barges are handled at the three deep sea terminals (ECT Delta, APM MV1 and EuroMax). The number of arriving and departing barges is determined for each terminal (APM Terminals Rotterdam, 2013), (ECT, 2013). The average number of containers at a barge is based on the information from NextLogic (Nextlogic, 2013). A real life performance meeting identified that the average call size of a barge at a deep sea terminal is equal to 44 TEU.

Combined data

Hourly data about container movements of trucks, trains and barges is aggregated in Figure 6. Hinterland transport by truck is concentrated on weekdays. Trains are mainly scheduled from Monday afternoon to Saturday afternoon. Barges operate equally over the week. The number of containers transported is almost twice as high on weekdays than on weekend days. Figure 7 shows the number of containers at the landside of the deep sea terminals per hour for an average weekday. There is a peak in the morning and a slight peak in the afternoon. During the night the demand is much lower. In weekend days the number of containers are more balanced over the day as shown in Figure 8.

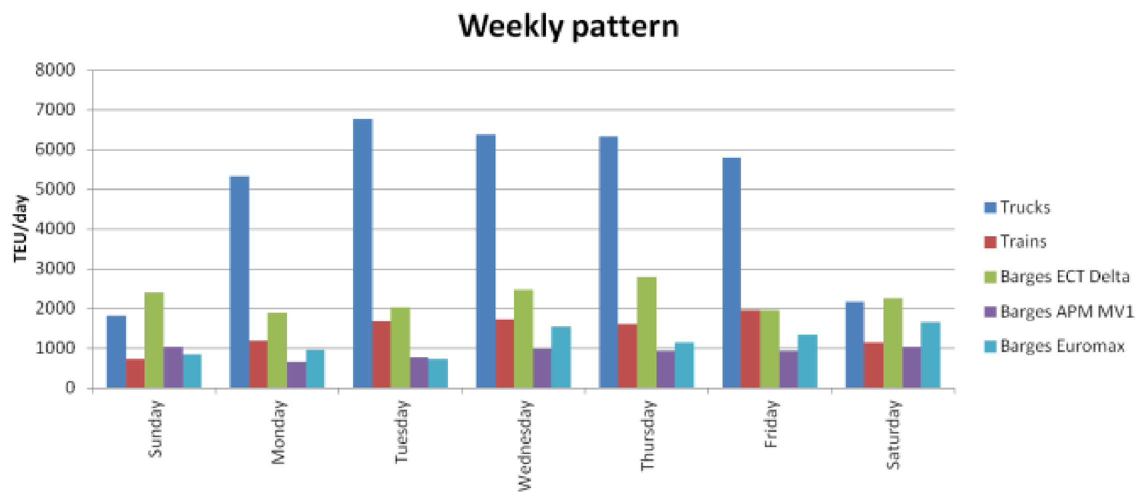

Figure 6: Containers per day per modality

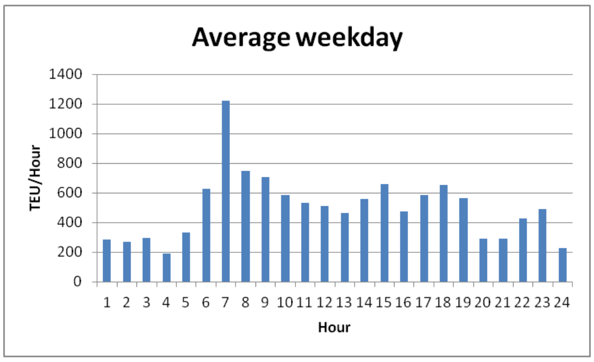

Figure 7: Demand per hour of the day for an average weekday

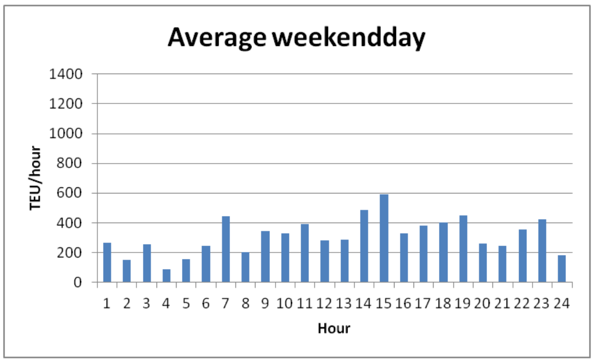

Figure 8: Demand per hour of the day for an average weekend day

The peak factor is determined by the average demand of containers per timeslot of three hours divided by the average demand of containers per day. A distinction will be made for weekdays and weekend days.

$$\text{Peak factor} = \left[\frac{\text{Average demand of containers per 3 hours}}{\text{Average demand of containers per day}}\right]_{\text{weekday,weekendday}}$$

The yearly number of containers and the peak factor will be combined in the container demand scenarios.

2.3 Container demand scenarios

The design of the ITT system for the year 2030 has many uncertainties. The throughput of containers of the ITT system is determined by economic developments. To deal with the uncertainty in demand of containers, three scenarios are created.

The three scenarios cover the extremes in demand for the ITT system. The chance of actual realization of the ITT system is higher when it has flexibility for different demands of containers. Also during the years before the final capacity is reached, the ITT system has to be competitive. Table 5 summarizes the yearly demand in containers for the different scenarios. The main causes of ITT transport is the transport between deep sea terminals and the common rail and barge facilities. The transport between empty depots and deep sea terminals are also significant container flows. The yearly number of containers are assigned to individual terminals based on the container flows as described in paragraph 2.1. In the Origin/Destination matrices of Appendix 2, the container flows are assigned to terminals based on the capacity of the terminal and the backdoor connections (Appendix 6).

Table 5: Total yearly ITT movements per scenario

	Total yearly movements in TEU
Scenario 1	3.34 million
Scenario 2	2.15 million
Scenario 3	1.42 million

Scenario 1 (Table 6) will have also the highest peak factors that coincidence with the current peak factors as found the data about trucks, trains and barges. A main cause of the peak factors are the operating hours of trucks and trains. In the future, the operating hours of trucks and trains will be more evenly spread over the day and therefore dampen the peak factors. No peak factors are assumed in scenario 3.

Table 6: Peak factors container arrival times

		0-3h	3-6h	6-9h	9-12h	12-15h	15-18h	18-21h	21-24h
Scenario 1	Weekday	0.57	0.77	1.79	1.09	1.12	1.14	0.76	0.76
	Weekend day	0.71	0.52	1.05	1.06	1.45	1.18	1.01	1.02
Scenario 2	Weekday	0.80	1.25	1.25	1.00	1.00	1.00	0.90	0.80
	Weekend day	0.90	0.90	1.00	1.10	1.10	1.10	1.00	0.90
Scenario 3	Weekday	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Weekend day	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Appendix F

Origin-destination matrices

This appendix holds the origin-destination matrices for the 2 transport demand scenarios used as input for the simulation model. The matrices have been constructed by Rick Jansen [27].

Participant	ECT Delta Terminal	Euromax Terminal	APM MV1 Terminal	Rotterdam World Gateway	APM MV2 Terminal	T3	T4	ECT Delta barge Feeder terminal	Delta Container Services	Common Rail Terminal	Rail Terminal West	Rotterdam Container Terminal Hartelhaven	Common Barge Service Center	Kramer Delta depot	Van Doorn Container depot	Empty depot MV1	Empty depot MV2	Customs
ECT Delta Terminal	x	3905	2580	3096	3572	3096	2560	397	119	66687	62559	47754	39795	7107	9645	12183	25787	9662
Euromax Terminal	3696	x	1907	2288	2640	2288	0	293	88	52080	48855	37294	31078	5550	7532	0	20138	7546
APM MV1 Terminal	2293	1790	x	1419	1638	1419	1174	182	55	34402	32272	24635	20529	3666	4975	6285	13303	4984
Rotterdam World Gateway	2818	2201	1454	x	2013	0	1443	224	67	41283	38727	29562	24635	4399	5970	7542	15963	5981
APM MV2 Terminal	3326	2598	1716	2059	x	2059	1703	264	79	0	44685	34110	28425	5076	6889	8702	0	6902
T3	2818	2201	1454	0	2013	x	1443	224	67	41283	38727	29562	0	4399	5970	7542	15963	5981
T4	2273	0	1173	1407	1623	1407	x	180	54	34138	32024	24446	20371	3638	4937	6236	13200	4946
ECT Delta barge Feeder terminal	321	250	165	198	229	198	164	x	8	5293	4965	3790	3158	564	765	967	2047	767
Delta Container Services	95	74	49	59	68	59	49	8	x	1588	1489	1137	948	169	230	290	614	230
Common Rail Terminal	100031	78120	51603	61924	0	61924	51206	7939	2382	x	0	0	0	8026	10893	13759	0	0
Rail Terminal West	93838	73283	48408	58090	67027	58090	48036	7447	2234	0	x	0	0	7529	10218	12907	27321	0
Rotterdam Container Terminal	31836	24862	16423	19708	22740	19708	16297	2527	758	0	0	x	0	10370	14074	17778	37630	0
Common Barge Service Center	26530	20719	13686	16423	18950	0	13581	2106	632	0	0	0	x	8642	11728	14815	31358	0
Kramer Delta depot	13198	10307	6808	8170	9427	8170	6756	1047	314	4322	4054	5584	4653	x	0	0	0	0
Van Doorn Container depot	17911	13988	9240	11088	12794	11088	9169	1422	426	5865	5502	7578	6315	0	x	0	0	0
Empty depot MV1	22625	0	11672	14006	16161	14006	11582	1796	539	7409	6950	9573	7977	0	0	x	0	0
Empty depot MV2	47889	37399	24705	29646	0	29646	24515	3801	1140	0	14711	20262	16885	0	0	0	x	0
Customs	9662	7546	4984	5981	6902	5981	4946	767	230	0	0	0	0	0	0	0	0	x

Table F.1: OD-matrix for demand scenario 1

Participant	ECT Delta Terminal	Euromax Terminal	APM MV1 Terminal	Rotterdam World Gateway	APM MV2 Terminal	T3	T4	ECT Delta barge Feeder terminal	Delta Container Services	Common Rail Terminal	Rail Terminal West	Rotterdam Container Terminal Hartelhaven	Common Barge Service Center	Kramer Delta depot	Van Doorn Container depot	Empty depot MV1	Empty depot MV2	Customs
ECT Delta Terminal	x	4239	0	2688	3102	2688	2223	0	0	57907	53453	0	26855	0	7925	10011	21189	4575
Euromax Terminal	3981	x	2053	1971	2275	1971	0	316	95	45223	41744	25633	20972	4560	6189	0	16548	3573
APM MV1 Terminal	0	1912	x	1213	1399	1213	1003	0	0	29873	27575	0	13854	0	4088	5164	10931	2360
Rotterdam World Gateway	2338	1826	1206	x	1336	0	958	186	56	28678	26472	16255	13300	2892	3925	4958	10494	2266
APM MV2 Terminal	2752	2149	1420	1363	x	1363	1127	218	66	0	30544	18756	15346	3337	4529	5720	0	2614
T3	2338	1826	1206	0	1336	x	958	186	56	28678	26472	16255	0	2892	3925	4958	10494	2266
T4	1892	0	976	937	1081	937	x	150	45	23714	21890	13442	10998	2391	3246	4100	8678	1874
ECT Delta barge Feeder terminal	0	264	0	168	193	168	139	x	0	4596	4242	0	2131	0	629	794	1682	363
Delta Container Services	0	78	0	50	57	50	41	0	x	1379	1273	781	639	139	189	238	505	109
Common Rail Terminal	70775	55272	36511	35051	0	35051	28984	5617	1685	x	0	0	0	5973	8107	10240	0	0
Rail Terminal West	65331	51020	33703	32354	37332	32354	26755	5185	1556	0	x	0	0	5514	7483	9452	20008	0
Rotterdam Container Terminal	0	20972	0	13300	15346	13300	10998	0	639	0	0	x	0	0	10480	13238	28020	0
Common Barge Service Center	21972	17159	11335	10881	12556	0	8998	1744	523	0	0	0	x	6318	8574	10831	22925	0
Kramer Delta depot	0	6841	0	4338	5005	4338	3587	0	209	3982	3676	0	4212	x	0	0	0	0
Van Doorn Container depot	11888	9284	6133	5887	6793	5887	4868	943	283	5404	4989	6987	5716	0	x	0	0	0
Empty depot MV1	15016	0	7746	7437	8581	7437	6149	1192	358	6827	6302	8825	7221	0	0	x	0	0
Empty depot MV2	31784	24822	16396	15741	0	15741	13016	2523	757	0	13338	18680	15284	0	0	0	x	0
Customs	4575	3573	2360	2266	2614	2266	1874	363	109	0	0	0	0	0	0	0	0	x

Table F.2: OD-matrix for demand scenario 2

Participant	ECT Delta Terminal	Euromax Terminal	APM MV1 Terminal	Rotterdam World Gateway	APM MV2 Terminal	T3	T4	ECT Delta barge Feeder terminal	Delta Container Services	Common Rail Terminal	Rail Terminal West	Rotterdam Container Terminal Hartelhaven	Common Barge Service Center	Kramer Delta depot	Van Doorn Container depot	Empty depot MV1	Empty depot MV2	Customs
ECT Delta Terminal	x	4239	0	2688	3102	2688	2223	0	0	0	50110	0	0	0	7925	10011	21189	4575
Euromax Terminal	3981	x	2053	1971	2275	1971	0	316	95	0	39133	17374	0	4560	6189	0	16548	3573
APM MV1 Terminal	0	1912	x	1213	1399	1213	1003	0	0	0	25850	0	0	0	4088	5164	10931	2360
Rotterdam World Gateway	2338	1826	1206	x	1336	0	958	186	56	0	24816	11018	0	2892	3925	4958	10494	2266
APM MV2 Terminal	2752	2149	1420	1363	x	1363	1127	218	66	0	28634	12713	0	3337	4529	5720	0	2614
T3	2338	1826	1206	0	1336	x	958	186	56	0	24816	11018	0	2892	3925	4958	10494	2266
T4	1892	0	976	937	1081	937	x	150	45	0	20521	9111	0	2391	3246	4100	8678	1874
ECT Delta barge Feeder terminal	0	264	0	168	193	168	139	x	0	0	3977	0	0	0	629	794	1682	363
Delta Container Services	0	78	0	50	57	50	41	0	x	0	1193	530	0	139	189	238	505	109
Common Rail Terminal	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0
Rail Terminal West	61245	47830	31595	30331	34997	30331	25081	4861	1458	0	x	0	0	11487	15590	19692	41682	0
Rotterdam Container Terminal	0	14215	0	9014	10401	9014	7454	0	433	0	0	x	0	0	19054	24069	50945	0
Common Barge Service Center	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0
Kramer Delta depot	0	6841	0	4338	5005	4338	3587	0	209	0	7658	0	0	x	0	0	0	0
Van Doorn Container depot	11888	9284	6133	5887	6793	5887	4868	943	283	0	10393	12703	0	0	x	0	0	0
Empty depot MV1	15016	0	7746	7437	8581	7437	6149	1192	358	0	13128	16046	0	0	0	x	0	0
Empty depot MV2	31784	24822	16396	15741	0	15741	13016	2523	757	0	27788	33963	0	0	0	0	x	0
Customs	4575	3573	2360	2266	2614	2266	1874	363	109	0	0	0	0	0	0	0	0	x

Table F.3: OD-matrix for demand scenario 3