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: 2013.TEL.7779

Title: **Development of a Simulation Model for an Inter Terminal Transport System at the Maasvlakte 1+2**

Author: H.J.L. Schroër

Title (in Dutch) Ontwikkeling van een Simulatie Model voor een Inter Terminal Transport Systeem op de Maasvlakte 1+2

Assignment: Computer

Confidential: no

Initiator (university): Dr. Rudy Negenborn

Supervisor: Dr. Rudy Negenborn

Ir. Mark Duinkerken

Date: May 30, 2013

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. Schroër	Assignment type:	Computer
Supervisor (TUD):	Dr. R.R. Negenborn	Creditpoints (EC):	15
	Ir. M.B. Duinkerken	Specialization:	TEL
		Report number:	2013.TEL.7779
		Confidential:	No

Subject: **Comparison of transportation systems for Inter Terminal Transport at Maasvlakte 1+2**

Due to the expansion of the Port of Rotterdam with the Maasvlakte 2 and the continuing rise in global container transport, there will be an increasing demand for Inter Terminal Transport (ITT) of containers at the Maasvlakte 1 + 2. Until 2020 this demand can be met using 3 TEU trucks that drive on the public road, but after 2020 this option will no longer suffice. One of the possibilities to solve the Inter Terminal Transport problem is by using a Closed Transportation Route. Different types of transportation systems to drive on the Closed Transportation Route could be considered; Terminal Tractors with Terminal Chassis, Multi Trailer Systems, AGVs, Lift AGVs, …

The goal of this research assignment is to develop a simulation model with which the potential of these alternative transport systems can be evaluated. Such a model should provide the possibility to evaluate research questions such as:

- How many vehicles and what type of vehicle are necessary to satisfy peak needs? And how many in case of average needs?
- What are the waiting times for the ITT system?
- Which infrastructure and how many cranes are needed for the ITT system?
- How can a capacity balance be found between peak and average needs?
- What is the environmental performance of the different transportation systems?

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. Rudy Negenborn (34B-1-230; r.r.negenborn@tudelft.nl) or Ir. Mark Duinkerken (34B-1-260; m.b.duinkerken@tudelft.nl).

The professor,
Prof.dr.ir. G. Lodewijks

English summary

Due to an increasing demand in global containerized transport over the past decades, 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. 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 Inter Terminal Transport 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. 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, including AGVs, Lift AGVs, and Multi Trailer Systems (MTSs).

A discrete simulation model has been developed that is able to evaluate the potential of the alternative transport systems. The developed ITT simulation model simulates a closed transport route on the Maasvlakte 1 + 2 in the Port of Rotterdam to which 19 different Terminals are connected. The model is able to operate with 3 different vehicles: AGVs, Lift AGVs, and MTSs. The system's main performance indicator is "non-performance". If a container is not delivered within a set time frame, it is accounted as non-performance.

The model has been built using Delphi and TOMAS, and is object oriented. It consists of 8 different objects: Containers, Generators, Roads, Intersections, Terminals, Nodes, Terminal Equipment and Vehicles. The processes for the AGV, Lift AGV, and MTS are all built into the Vehicle object's process. The Vehicles use the Dijkstra algorithm for path planning. The general simulation parameters have to be defined in a configuration file. The properties of the objects have to be defined in the various input files.

The output of the model consists of a number of output files and graphs. There are 3 different output files that hold all results of the simulation run: a general output file, an output file that shows the results per Road and Intersection, and an output file that show the results per Terminal. The results include performance indicators such as: non-performance, waiting times, occupation rates, the total distance traveled by the Vehicles, and the traffic flows on the infrastructure. Data is being gathered during simulation that can be viewed live in various graphs on the TOMAS Collections form. When the simulation is finished the gathered data is written away to a number of .csv files. These files can be imported in spreadsheet software for further analysis.

The model is easily adaptable for testing different scenarios. The vehicle type, the amount of vehicles, and the total amount of containers to be transported per hour, can simply be changed by altering their values in the configuration file. Using a different transport demand and/or road map only requires changing the input data in the designated input files.

The model has been evaluated by the the hand of a number of simulation runs based on a predefined transport demand scenario. All obtained results comply with what was expected. The generated Container flows deviate very little from the intended flows. The results of the simulations show very high values of non-performance, which means that most containers are delivered late. This is easily explained by the absence of planning. The only advanced planning algorithm currently implemented in the model is for path planning. In order to make the model perform better, and to show more realistic results, it is essential that planning algorithms will be implemented in future expansion of the model.

Dutch summary

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. 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. Het Inter Terminal Transport systeem verwerkt alle containers die moeten worden overgedragen tussen de verschillende container terminals op de Maasvlakte. Het is voorspeld dat het ITT tot 2020 kan worden uitgevoerd met 3 TEU trucks die rijden op de openbare weg, maar na 2020 zal deze optie niet langer voldoen. Daarom moet er een nieuw en meer geavanceerde ITT systeem worden ontwikkeld. Een van de opties is een gesloten transport route waarop diverse typen voertuigen zonder interactie met andere soorten verkeer rond kunnen rijden. Verschillende transport systemen die rijden op de gesloten transportroute kunnen worden beschouwd, inclusief AGVs, Lift AGVs, en Multi Trailer Systems (MTSs).

Een discreet simulatiemodel is ontwikkeld dat in staat is om de potentie van de alternatieve transportsystemen te evalueren. Het ontwikkelde ITT simulatiemodel simuleert een gesloten transportroute op de Maasvlakte 1 + 2 in de haven van Rotterdam waarop 19 verschillende terminals zijn aangesloten. Het model is in staat om te opereren met 3 verschillende voertuigen: AGVs, Lift AGVs, en MTSs. De belangrijkste prestatie-indicator van het systeem is "non-performance". Als een container niet binnen een bepaald tijdskader kan worden geleverd dan wordt het geregistreerd als non-performance.

Het model is gebouwd met behulp van Delphi en TOMAS, en is object georiënteerd. Het bestaat uit 8 verschillende objecten: Containers, Generatoren, Wegen, Kruispunten, Terminals, Nodes, Terminal Equipment en Voertuigen. De processen voor de AGV, Lift AGV en MTS zijn allemaal gebouwd in het proces van het Voertuig object. De Voertuigen maken gebruik van het Dijkstra algoritme voor path planning. De algemene simulatie parameters kunnen worden gedefinieerd in een configuratiebestand. De eigenschappen van de objecten kunnen worden gedefinieerd in de verschillende invoerbestanden.

De output van het model bestaat uit een aantal output bestanden en grafieken. Er zijn 3 verschillende output bestanden die alle resultaten van de simulatie bevatten: een algemene output bestand, een output bestand dat de resultaten per Weg en Kruispunt toont en een output bestand dat de resultaten per Terminal toont. De resultaten bestaan uit prestatie-indicatoren zoals: non-performance, wachttijden, bezettingsgraden, de totale afstand die de voertuigen afleggen en de verkeersstromen op de infrastructuur. Tijdens de simulatie worden gegevens verzameld die live kunnen worden bekeken in verschillende grafieken op het TOMAS Collections formulier. Als de simulatie voltooid is wordt de verzamelde data weggeschreven naar een aantal .csv bestanden. Deze bestanden kunnen in spreadsheet software worden gemporteerd voor verdere analyse.

Het model is gemakkelijk aan te passen voor het testen van verschillende scenario's. Het Voertuig type, de hoeveelheid Voertuigen en de totale hoeveelheid Containers die getransporteerd moet worden per uur kunnen eenvoudig worden gewijzigd door de waarden te veranderen in het configuratiebestand. Een andere transport vraag en/of infrastructuur kan worden verkregen door het veranderen van de invoergegevens in de bijbehorende invoer bestanden.

Het model is geëvalueerd aan de hand van een aantal simulaties uitgevoerd op basis van een vooraf gedefinieerde transport scenario. Alle verkregen resultaten komen overeen met de verwachtingen. De gegenereerde Container stromen wijken zeer weinig af van de beoogde stromen. De resultaten van de simulaties laten zeer hoge non-performance waarden zien, wat betekent dat de meeste Containers te laat worden geleverd. Dit kan gemakkelijk worden verklaard door de afwezigheid van planning. Het enige

geavanceerde planning algoritme momenteel gemplementeerd in het model is voor path planning. Om ervoor te zorgen dat het model beter presteert en meer realistische resultaten toont is het essentieel dat het model wordt uitgebreid met geavanceerde planning algoritmes.

Abbreviations

AGV	= Automated Guided Vehicle
ITT	= Inter Terminal Transport
MTS	= Multi Trailer System
O-D	= Origin-Destination
TEU	= Twenty foot Equivalent Unit

Table of Contents

Chapter 1

Introduction

This research will be part of a collaboration project that has been set up to "Design the company that will execute the Internal Transport on MV2 with horizon 2030. Not only the transport vehicles and their infrastructure, but a total concept including innovative technologies, logistics, ICT, business case, organization and governance aspects." [1]. The project is a collaboration between Delft University of Technology, Erasmus University Rotterdam, the University of Hamburg, and the Port of Rotterdam.

1.1 Problem statement

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 [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 Inter Terminal Transport 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. 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, including AGVs, Lift AGVs, and Multi Trailer Systems.

In the development of a new ITT system, different research questions have to be answered before the system can be built. These research questions include the following:

- How many vehicles and what type of vehicle are necessary to satisfy peak needs? And how many in case of average needs?
- What are the waiting times for the ITT system?
- Which infrastructure and how many cranes are needed for the ITT system?
- How can a capacity balance be found between peak and average needs?
- What is the environmental performance of the different transportation systems?

A simulation model will have to be developed in order to be able to evaluate the potential of the alternative transport systems. The model will make it possible to find the answers to these research questions.

1.2 Research objective

The objective of this research is to *develop a discrete simulation model with which the potential of the alternative transport systems can be evaluated*. The model has to make it possible to find answers to the research questions stated in the problem description.

The model should be easily adaptable to test different scenarios with different vehicle types, amounts of vehicles, transport demands, and infrastructure.

1.3 Scope of the research

1.3.1 Maasvlakte 1 + 2 layout

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 between which Inter Terminal Transport will take place over a closed transport route. A map of the container terminals and the roads between them on which the ITT will take place is shown in Figure 1.1. Figure 1.2 shows a table of all possible relations between the different Maasvlakte container terminals.

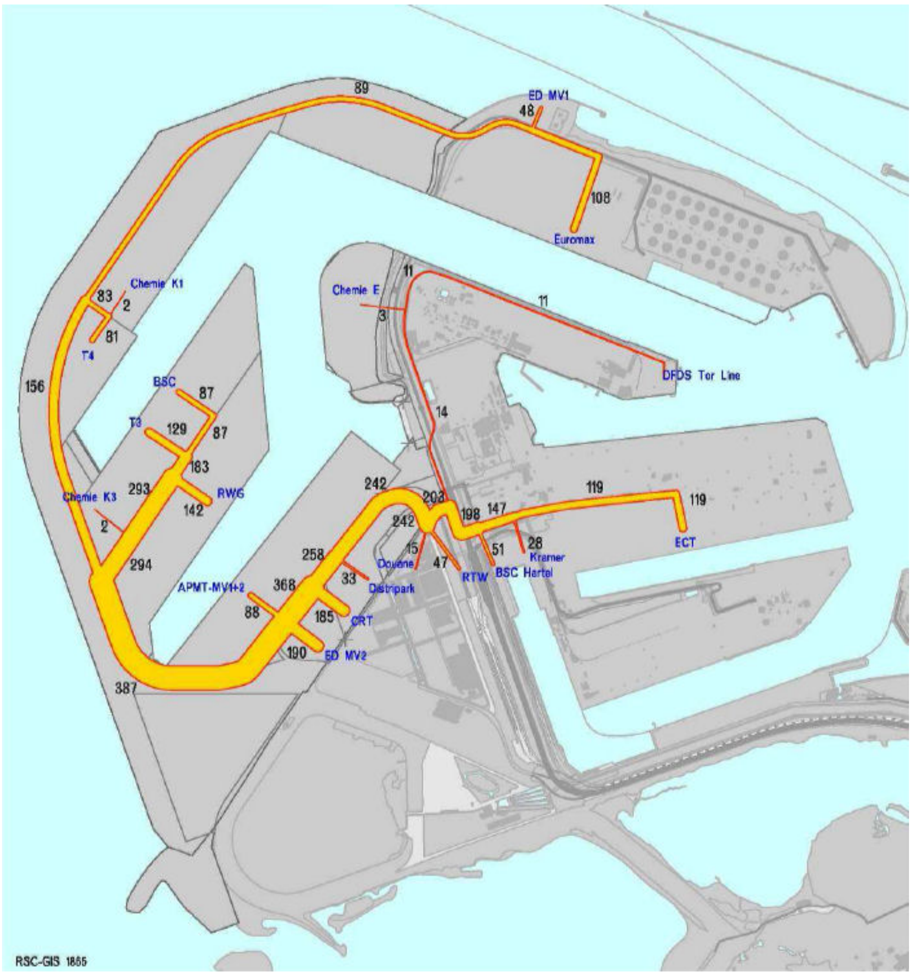


Figure 1.1: Map of the Maasvlakte ITT[2]

In total there are 19 container terminals that will be part of the modeled ITT system. These are the following:

van: \ naar:	ECT	APMT-MV1 + APMT-MV2	RWG	Containerterminal X	Containerterminal Y	ECT-Euromax	Common Rail Terminal *(nw)*	RTW - MV1 *(ECT + nw APMT)*	BSC Hartelhaven *(huidig ECT)*	Common Barge Service Center	Kramer (ED + barge/feeder)	Douane scan	Distripark Maasvlakte	ED 1 (westzijde distripark)	ED 2 (nabij EMX)	DFDS Tor Line	Chemie nieuw E (MV1)	Chemie nieuw K1 (MV2)	Chemie nieuw K3 (MV2)
ECT		+	+	+	+	+	+	+	+	+	+	+	+	+	-	-	-	-	-
APMT-MV1 + APMT-MV2	+		+	+	+	+	+	+	+	+	+	+	+	+	-	-	-	-	-
RWG	+	+		+	+	+	+	+	+	+	+	+	+	+	-	-	-	-	-
Containerterminal X	+	+	+		+	+	+	+	+	+	-	+	+	+	+	-	-	-	-
Containerterminal Y	+	+	+	+		+	+	+	+	+	-	+	+	+	+	-	-	-	-
ECT-Euromax	+	+	+	+	+		+	+	+	+	-	+	+	+	+	-	-	-	-
Common Rail Terminal *(nw)*	+	+	+	+	+	+		-	-	-	+	+	+	+	+	+	+	+	+
RTW - MV1 *(ECT+nw APMT)*	+	+	+	+	+	+	-		-	-	+	+	+	+	+	+	+	+	+
BSC Hartelhaven *(ECT)*	+	+	+	+	+	+	-	-		-	+	+	+	+	+	+	+	+	+
Common Barge SC	+	+	+	+	+	+	-	-	-		+	+	+	+	+	+	+	+	+
Kramer (ED + barge/feeder)	+	+	+	-	-	-	+	+	+	+		+	+	-	-	-	+	-	-
Douane scan	+	+	+	+	+	+	+	+	+	+	+		+	-	-	-	+	+	+
Distripark Maasvlakte	+	+	+	+	+	+	+	+	+	+	+	+		+	+	+	-	-	-
ED 1 (westzijde distripark)	+	+	+	+	+	+	+	+	+	+	-	-	+		-	-	+	-	-
ED 2 (nabij EMX)	-	-	-	+	+	+	+	+	+	+	-	-	+	-		-	-	-	-
DFDS Tor Line	-	-	-	-	-	-	+	+	+	+	-	-	+	-	-		-	-	-
Chemie nieuw E (MV1)	-	-	-	-	-	-	+	+	+	+	+	+	-	+	-	-		-	-
Chemie nieuw K1 (MV2)	-	-	-	-	-	-	+	+	+	+	-	+	-	-	-	-	-		-
Chemie nieuw K3 (MV2)	-	-	-	-	-	-	+	+	+	+	-	+	-	-	-	-	-	-	

Figure 1.2: Relations between the container terminals in the ITT system. (+ = Possible relation, − = No relation)[2]

- ECT Marine terminal
- APMT Marine terminal
- RWG Marine terminal
- T3 Marine terminal
- T4 Marine terminal
- Euromax Marine terminal
- CRT Common Rail Terminal
- RTW Rail Terminal West
- BSC Barge Service Center
- BSCHar Barge Service Center Hartelhaven
- Kramer Empty Depot + Barge/Feeder Service Center
- Customs Customs Scan
- Distri Distripark Maasvlakte
- ED1 Empty Depot MV1
- ED2 Empty Depot MV2
- DFDS DFDS Tor Line, recently bought by Rhenus Logistics
- ChemE Chemical
- ChemK1 Chemical
- ChemK3 Chemical

1.3.2 ITT Vehicles

Three different types of ITT vehicles will be considered: AGVs, Lift AGVs and Multi Trailer Systems (MTSs). The vehicles have been shown in Figure 1.3.

AGV: The conventional 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 one 40 ft. container at a time.

Lift AGV: The (Gottwald) lift AGV is a rather new vehicle and will first be used at the APM Terminal at the Maasvlakte 2 [6]. In many ways it is the same as the conventional 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 Lift AGVs and terminal equipment don't have to wait for each

Figure 1.3: Vehicle types

other to make a move. The downside of the Lift AGV is that it has a lower speed than the conventional AGV. It is also able to carry one 40 ft. container at a time.

MTS: A Multi Trailer System consisting of a manned Terminal Tractor pulling a train of Terminal Chassis. The MTS usually has a capacity of 5 40 ft. containers.

1.4 Research questions

A number of research questions have been defined in order to reach the research objective. These are the following:

- What are the key performance indicators for the ITT system? (Chapter 2)
- Which objects are needed to construct the simulation model? (Chapter 3)
- How can the model be made easily adaptable to test different scenarios? (Chapter 3)
- What output data will the model be able to provide? (Chapter 4)
- How to evaluate the performance of the simulation model? (Chapter 5)
- How can the model be expanded in future research? (Chapter 6)

1.5 Structure of the report

Chapter 2 will discuss the main information defined prior to building the ITT simulation model, including the expected transport demand, what the input and output of the model will consist of, some general assumptions, and the model's performance indicators. Chapter 3 will explain how the ITT simulation model works. All modeled objects will be discussed in detail. Chapter 4 will show what the output of the model looks like. This includes the model interface, the output files, and the graphs. Chapter 5 will discuss the evaluation of the simulation model by the hand of a number of simulation runs based on the transport demand scenario presented in Chapter 2. The report will be concluded in Chapter 6. This Chapter also includes recommendations for future research.

Chapter 2

Assumptions

This chapter will discuss the main information defined prior to building the ITT simulation model. It will discuss the expected transport demand, what the input and output of the model will consist of, some general assumptions, and the model's performance indicators.

2.1 Transport demand

In order to create a realistic transport demand, an origin-destination (O-D) matrix has to be created. Due to confidentiality this matrix has not been given by the Port of Rotterdam. Therefore the matrix has to be estimated from the values and assumptions found in the "Gesloten transport route: Inter Terminal Transport op Maasvlakte 1 en 2" report by Willemien Diekman and Jan Willem Koeman [2]. Since the report is incomplete and contains several contradictions, it is impossible to exactly replicate the transport demands. Therefore some assumptions and estimations have been made in order to create an origin-destination matrix that is somewhere in the range of the matrix used in the report. The creation of the O-D will not be discussed in great detail because the exactness of these data is not very important at this point. It is just to have a somewhat realistic scenario that can be used to evaluate the simulation model, which will be done in Chapter 5.

A scenario is assumed where terminals that are situated next to each other don't use the ITT system, but instead use their "back doors". For 2035 this would give a total of 3.4 million TEU per year [2]. That would mean 685 containers/hour for peak hours, and 228 containers/hour for average hours.

The numbers of containers on a piece of road during a peak hour, which are depicted by the numbers in Figure 1.1, have been used as a basis for the O-D matrix. Adding up all numbers from a terminal to the first intersection gives a total of 1369 containers/hour, which when rounded is equal to $2 * 685$. The total percentage of containers from and to a single terminal is then equal to the number of containers to and from that terminal in a busy hour divided by 1369. An overview of all these values can be seen in Table 2.1. These values, combined with the possible relations shown in Figure 1.2, different assumptions in the report, and various loose assumptions and estimations, have led to the O-D matrix shown in Figure 2.1. All numbers in the matrix represent the percentage of the total amount of containers that has to be transported from that specific origin to that specific destination. This percentage, in combination with the total amount of Containers to be transported per hour, determines the amount of containers that has to be transported from that specific origin to that specific destination. The matrix will be used in Chapter 5 to evaluate the simulation model.

2.2 Model input and output

The input and output of the simulation model has been schematically shown in Figure 2.2. The model input consists of a general configuration file and a number of input files. In these input files the road map, the equipment and vehicle properties, and the transport demand can be defined. This will be explained

From: \ To:	**Containers / hour**	**Percentage of total**
ECT	119	8,69
APMT	88	6,43
RWG	142	10,37
T3	129	9,42
T4	81	5,92
Euromax	108	7,89
CRT	185	13,51
RTW	47	3,43
BSC	87	6,36
BSCHartel	51	3,73
Kramer	28	2,05
Customs	15	1,10
Distripark	33	2,41
ED1	48	3,51
ED2	190	13,88
DFDS	11	0,80
ChemE	3	0,22
ChemK1	2	0,15
ChemK3	2	0,15
Total:	1369	100,00

Table 2.1: Containers per hour

From: \ To:	ECT	APMT	RWG	T3	T4	Euromax	CRT	RTW	BSC	BSCHartel	Kramer	Customs	Distripark	ED1	ED2	DFDS	ChemE	ChemK1	ChemK3	Total
ECT		0,27	0,10	0,00	0,05	3,29	3,39		0,90			0,20	0,69		1,56					10,43
APMT	0,27		0,32	0,29	0,03	0,24	2,51		0,66	0,50	1,09	0,15	0,51		1,15					7,72
RWG	0,00	0,31			0,06	0,22	4,04	1,25	1,07	0,82	1,77	0,23	0,82		1,86					12,44
T3	0,90	0,29			0,05	0,89	3,67	1,14	0,97	0,74		0,21	0,75		1,69					11,30
T4	0,57	0,18	0,07	0,76			2,31	0,71	0,61	0,47		0,13	0,47	0,82						7,10
Euromax	0,90	0,24	1,07	0,98			3,07	0,95	0,81	0,62		0,18	0,62							9,46
CRT	0,62	0,46	0,73	0,67	0,42	0,56								0,41	0,78	0,24		0,14	0,14	5,16
RTW			0,55	0,50	0,31	0,42										0,06	0,22			2,06
BSC	0,26	0,19	0,31	0,28	0,18	0,23								1,03	1,23	0,11				3,82
BSCHartel		0,23	0,38	0,34	0,21	0,29									0,72	0,07				2,24
Kramer		0,41	0,65										0,16							1,22
Customs	0,20	0,15	0,23	0,21	0,13	0,18														1,10
Distripark								0,38		0,42										0,80
ED1					3,30		0,41		1,03											4,74
ED2	3,25	2,41	3,88	3,53			1,63		2,57	1,51										18,78
DFDS							0,56	0,14	0,26	0,15										1,12
ChemE								0,22												0,22
ChemK1							0,14													0,14
ChemK3							0,14													0,14
Total	6,96	5,14	8,30	7,55	4,74	6,32	21,88	4,80	8,90	5,22	2,86	1,10	4,02	2,26	8,98	0,48	0,22	0,14	0,14	**100%**

Figure 2.1: Origin-Destination matrix

in Chapter 3. The model output consists of a number of output files and graphs. They will be discussed in Chapter 4. All simulations are performed within one model.

2.3 General assumptions

Some general assumptions have been made in order to simplify the model. These include the following:

- There is no distinction between 20 ft. and 40ft. containers. Both are accounted as one unit.
- All containers have the same level of urgency.
- Containers are generated one by one and do not come in batches.
- Vehicles don't have to stop for gas or for changing their battery.
- All equipment and vehicles are active 24 hours a day, 7 days a week.
- Vehicles do not interact with each other.
- MTSs are modeled as one vehicle with a certain capacity, and not as a separate terminal tractor and chassis.

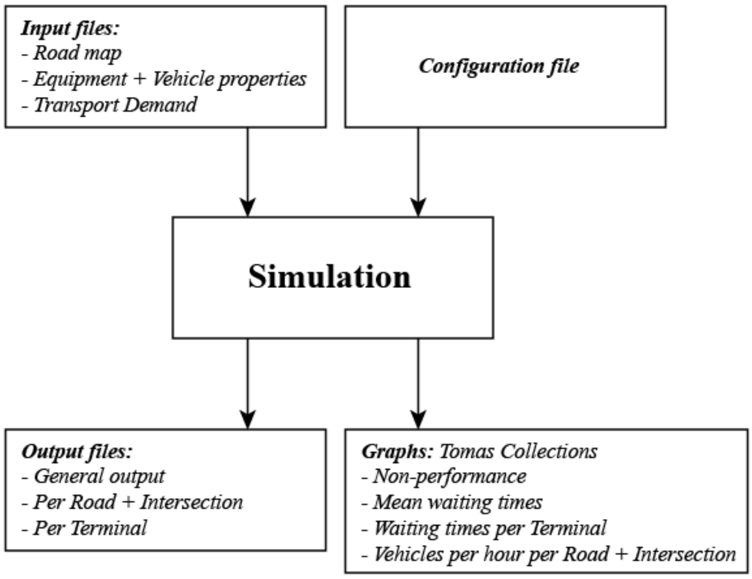


Figure 2.2: Model input and output

- The model does not include congestion. Instead, each intersection takes a certain time for a vehicle to cross. The busier the intersection, the higher the time to cross.

- Terminals do not have separate handling centers. Each terminal has one handling center with multiple terminal equipment.

- There is only one type of terminal, with an adjustable terminal equipment type and number. There is no distinction for terminals like barge or train terminals.

- Except for path planning, no advanced planning will be implemented at this point.

2.4 Performance indicators

In order to be able to evaluate research questions such as stated in Section **??**, the model needs to have some performance indicators. The most import task of the system is to deliver the containers to their destination in time. If a container is not delivered to its destination in time it will be accounted as non-performance. This is conform to the method used by Tierney *et al.* [7], but contrary to the method used by Ottjes *et al.* [4] [5] and Duinkerken *et al.* [3] where the system is also penalized for delivering a container early.

Whether the delivery of a container is accounted for as non-performance depends on the time it takes the system to deliver a container, and on the so called "Cap Factor". The Cap Factor determines how much later the container may be delivered before it counts as non-performance. For instance: if the Cap Factor is 1,25, then the container only counts as non-performance when it is delivered late with more than 25% of its expected handling time. The way this method is applied in the model is explained in Section 3.3.2.

This method is not completely realistic, as in reality containers have different classes of urgency and thereby different cap values, but since there is only one type of container at this point it provides a viable solution. It is expected that the amount of non-performance will be quite high, as no planning is involved at this point. Values may not be completely realistic, but they will be able to show wether the system is performing better or worse with different configurations.

Other performance indicators are:

- Occupation rates of the vehicles.
- Occupation rates of the terminal equipment, both in general and per terminal.
- Waiting times, both in general and per terminal.
- Total distance traveled by the vehicles.
- Number of empty rides.

2.5 Summary

This chapter has given an overview of the basic settings and assumptions made prior to building the simulation model. The model will be evaluated (in Chapter 5) using a transport demand scenario based on an ITT report by Diekman *et al.* [2]. The model will use several input files and a configuration file to define the simulation parameters. The results of the simulations will be written to a series of output files and graphs. A number of general assumptions have been made to simplify the model. The main performance indicator is non-performance, which registers if containers are delivered on time or not.

The next chapter will explain how the ITT simulation model works. All modeled objects will be discussed separately.

Chapter 3

Simulation model

This chapter will explain how the ITT simulation model works. The model has been widely verified throughout every step of the programming process. Every process has been closely monitored in order to make sure everything works exactly as intended.

3.1 Model design

A simulation model for the ITT system has been created using Delphi and the simulation tools provided by TOMAS [8]. TOMAS has been used because is a user friendly shell (written in Delphi) with all the basic needs for a process oriented simulation. It contains functionalities such as visual control, queues, and statistics that make it easier to develop and analyze the logistic simulation model.

The simulation model is object-oriented. It consists of the following objects: Containers, Generators, Roads, Intersections, Terminals, Nodes, Terminal Equipment and Vehicles. The Generators, Terminals, Terminal Equipment, and Vehicles all have a process and are therefore active. The Containers, Roads, Intersection, and Nodes don't have a process and are therefore passive. The relations between the different objects are shown in Figure 3.1.

Figure 3.2 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. Nodes are not physical objects. They are created for every Terminal and Intersection so they can be used for route planning (see Section 3.9).

The most important communications in the model are shown in Figure 3.3. When a Terminal has a Vehicle shortage, it searches for the Terminal with the most idle Vehicles and then orders one of the Vehicles there to perform an empty ride. The Vehicles 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 or Intersection when and for how long they are driving there.

All modeled objects will be discussed in detail in the following sections. They 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.
• Creating the objects	Description of how the objects are created using input files.

3.2 General input

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 3.4.

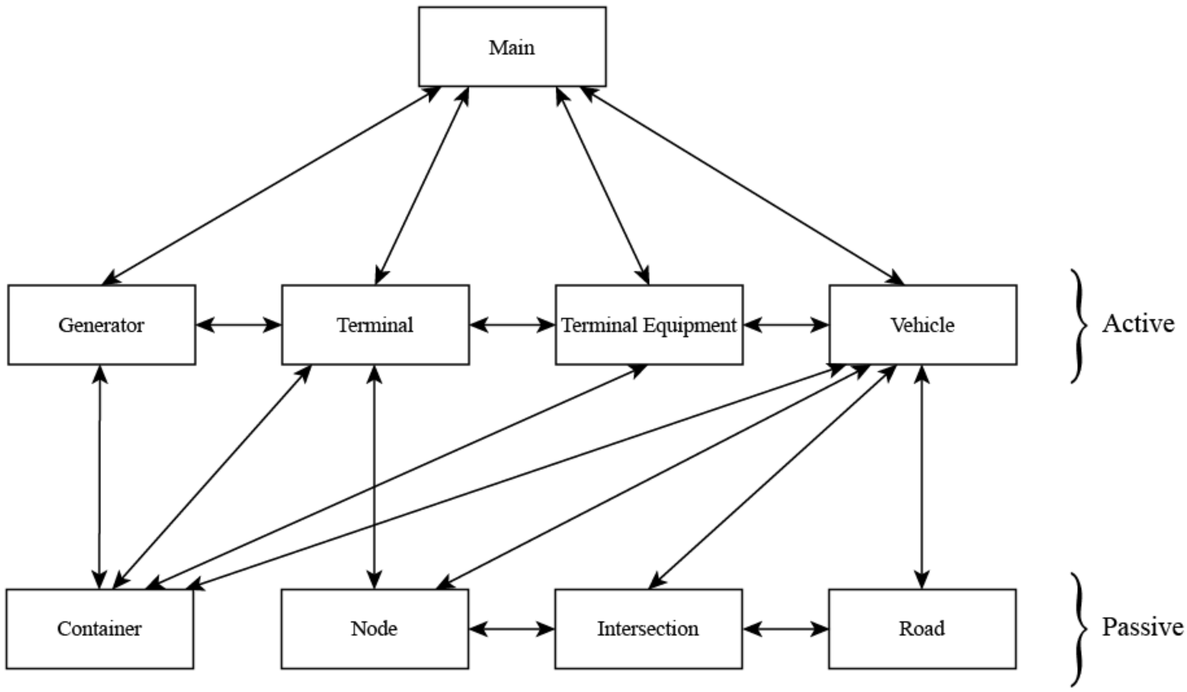


Figure 3.1: Relations between the objects in the ITT simulation model

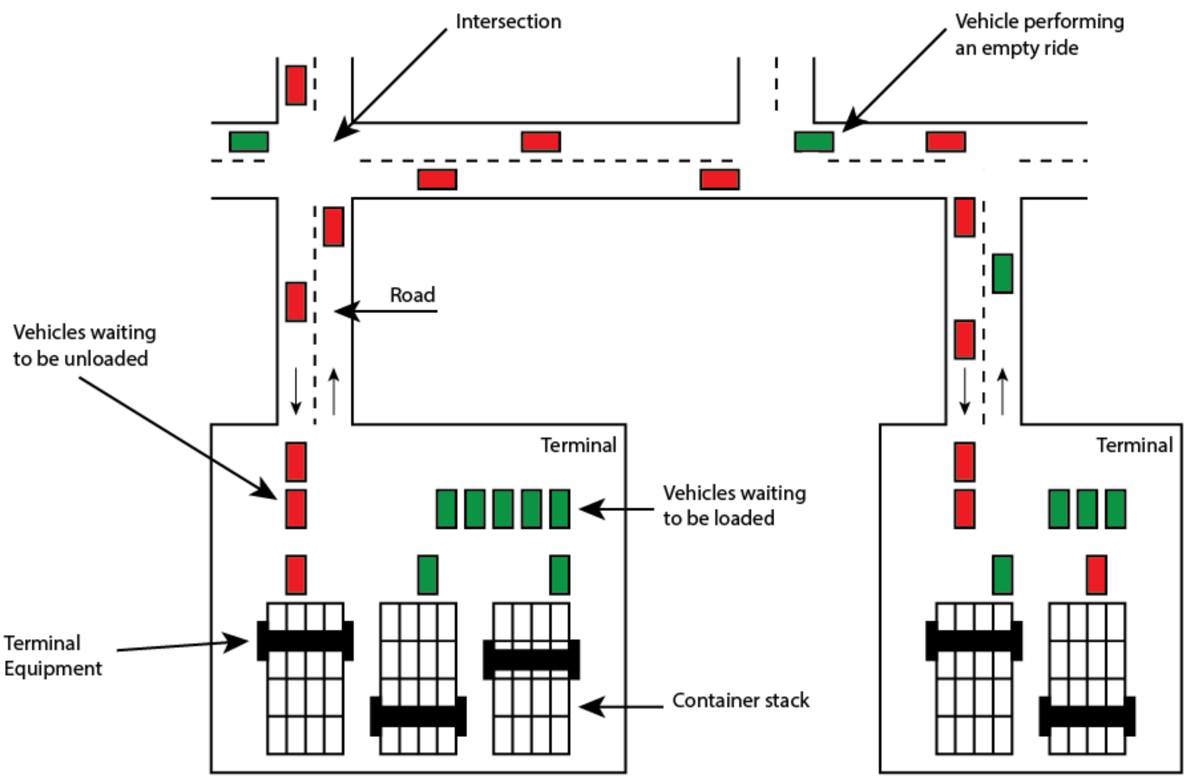


Figure 3.2: Schematic representation of the physical objects in the model

The parameters to be specified are:

- VehicleType — Determines which Vehicle type is used. Must be AGV, LiftAGV, or MTS.
- No. of Vehicles — Determines the total number of Vehicles in the system.
- Runtime — Determines the total simulation time. Must be specified in hours.
- Containers per hour — Determines the average amount of Containers created by the Generators per hour.
- Cap factor — Factor for calculating non-performance (see Section 2.4).

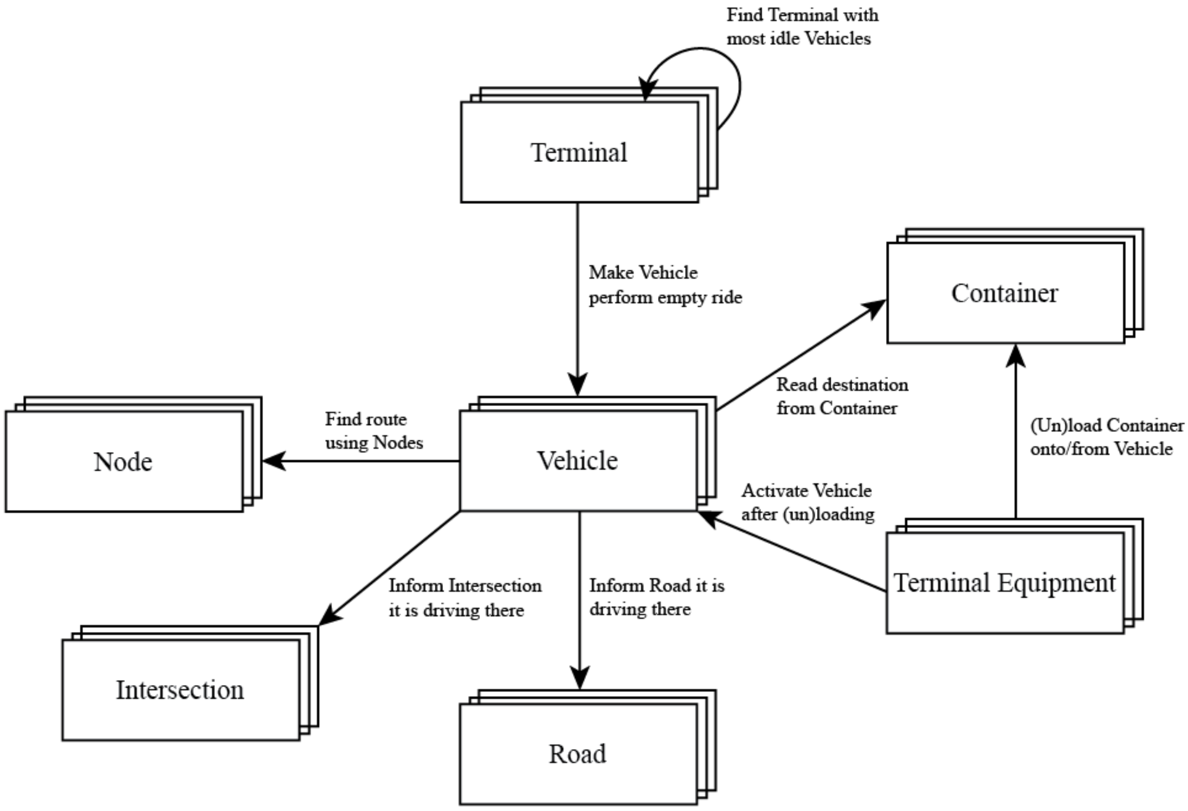


Figure 3.3: Communications between the objects

```
{Configuration file for the ITT Simulation model}
LiftAGV {VehicleType; AGV, LiftAGV or MTS}
200 {no. of vehicles}
10 {Runtime in hours}
200 {Containers per hour}
1,25 {Capfactor for non-performance}
```

Figure 3.4: Configuration file: *configfile.txt*

If VehicleType is not specified as AGV, LiftAGV, or MTS, a popup message saying *"Select proper Vehicle Type. Check configfile.txt."* will be displayed and the simulation will not be started.

3.3 Containers

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 Generators (see Section 3.4). 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.

3.3.1 Attributes

• Origin	Terminal to be transported from
• Destination	Terminal to be transported to
• DepartureTime	Time at which it is placed at the origin terminal
• ArrivalTime	Time at which it is unloaded at the destination terminal
• TravelDistance	Distance from origin to destination
• ExpectedHandlingTime	Expected time from stack at origin to stack at destination
• MyVehicle	Vehicle it is transported with
• RegisterNonPerformance	Method: to register non-performance
• CalculateExpectedHandlingTime	Method: to calculate its ExpectedHandlingTime

3.3.2 Process description

The container doesn't have a process, but it has two methods which can be called for by other objects; *RegisterNonPerformance* and *CalculateExpectedHandlingTime*.

3.3.2.1 CalculateExpectedHandlingTime

The method *CalculateExpectedHandlingTime* is used to calculate the Container's expected handling time in case of no delays. This value is necessary to determine the system's container handling performance. The method uses Formulas 3.1, 3.2 and 3.3 to calculate the expected handling time. All times are in minutes.

$$RoadTime = \frac{TravelDistance}{Speed * 60} \tag{3.1}$$

$$IntersectionTime = \sum_{i=1}^{N} TimeToCross_i \tag{3.2}$$

$$\begin{aligned} ExpectedHandlingTime = OriginHandlingTime + RoadTime + IntersectionTime \\ + DestinationHandlingTime \end{aligned} \tag{3.3}$$

Where:

$RoadTime$	=	Time needed to drive the Roads
$IntersectionTime$	=	Time needed to cross the Intersections
N	=	Total number of Intersections to be crossed
$OriginHandlingTime$	=	Expected handling time of the Equipment at the Origin Terminal
$DestinationHandlingTime$	=	Expected handling time of the Equipment at the Destination Terminal

In case of the Lift AGV scenario (Section 3.9.2.5) the lift times of the AGV have to be included too, so for this scenario Formula 3.3 becomes:

$$\begin{aligned} ExpectedHandlingTime = OriginHandlingTime + LiftLoadTime + RoadTime \\ + IntersectionTime + LiftUnloadTime + DestinationHandlingTime \end{aligned} \tag{3.4}$$

3.3.2.2 RegisterNonPerformance

The method *RegisterNonPerformance* is used to register if the Container has been delivered within the set time frame. More information on non-performance can be found in Section 2.4.

The method is called for by the Terminal Equipment just before the Container is destroyed (see Section 3.6). The process goes as follows:

- Calculate true handling time

- if $Truehandlingtime > ExpectedHandlingTime * CapFactor$
    - Register this Container as non-performance

3.3.3 Interaction with other objects

Generator

Containers are generated and placed at their Origin Terminals by the Generators.

Terminal

Containers wait in the MyContainerQ of their Origin Terminal until they are handled by the Terminal Equipment located at the Terminal.

Terminal Equipment

Containers are loaded from the MyContainerQ onto a vehicle by the Terminal Equipment. The Terminal Equipment is responsible for destroying the Container objects after unloading at the Destination Terminal.

Vehicle

The Vehicles read the information of their Containers in order to determine where they need to go.

3.4 Generators

The Generators are responsible for creating the Containers and placing them in the MyContainerQ of their Origin Terminal. There is one Generator for every possible Origin-Destination combination. The Generators use an exponential distribution, in combination with the mean amount of Containers they have to create per hour, to determine the time in between creating new Containers.

3.4.1 Attributes

• InterContTimeDistribution	Exponential distribution on the mean time between creating Containers
• MyOrigin	Terminal where created Containers need to be be transported from
• MyDestination	Terminal where created Containers need to be be transported to
• PROCESS	Method: describes activities as a function of time

3.4.2 Process description

- Repeat until RunTime is over
    - Wait for InterContTimeDistribution.Sample
    - Create a new Container
    - Assign the Container's Origin and Destination
    - Place the Container in its Origin Terminal's MyContainerQ
- Interrupt the simulation

3.4.3 Interaction with other objects

Container

The Generators create the containers, give them their properties, and place them at their Origin Terminal.

Terminal

The Generators place the containers in the Terminal's MyContainerQ.

3.4.4 Creating the Generators

The Generators are created by reading their properties from an input file called *inputgenerators.txt*. Part of the input file is shown in Figure 3.5. For every generator, 4 properties have to be specified: the Generator's name, the names of the Origin and Destination Terminals of the Containers it creates, and the percentage of the total amount of Containers to be transported that have to be transported for this Origin-Destination combination. These percentages originate from the Origin-Destination matrix defined in Section 2.1. Using the same model with a different transport demand would simply mean updating the Generators input file with the properties of a different Origin-Destination matrix.

```
{Input file for the Generators}

123 {total number of generators}

{GeneratorID - Origin - Destination - Percentage of Containers per hour}

G_ECT1 ECT APMT 0,27
G_ECT2 ECT RWG 0,1
G_ECT3 ECT T4 0,05
G_ECT4 ECT Euromax 3,29
G_ECT5 ECT CRT 3,39
G_ECT6 ECT BSC 0,9
G_ECT7 ECT Customs 0,2
G_ECT8 ECT Distri 0,69
G_ECT9 ECT ED2 1,56
G_APMT1 APMT ECT 0,27
G_APMT2 APMT RWG 0,32
G_APMT3 APMT T3 0,29
G_APMT4 APMT T4 0,03
G_APMT5 APMT Euromax 0,24
G_APMT6 APMT CRT 0,24
```

Figure 3.5: Part of input file: *inputgenerators.txt*

If one of the entered Origin or Destination names does not match with the names of the Terminals, so when a spelling mistake is made, the model will show a popup message displaying exactly for which Generator the mistake was made, and the simulation will not be started.

3.5 Terminals

All Origins and Destinations are modeled as Terminals. Each Terminal consists of a number of queues and a certain amount of Terminal Equipment that is tasked with loading and unloading the Containers. The schematics of the Terminals for the AGV and the MTS scenarios can be seen in Figure 3.6 and for the Lift AGV scenario in Figure 3.7. The schematics are different for the Lift AGV scenario because in this case the Terminal Equipment and Vehicle processes are decoupled (see Sections 3.6 and 3.9.2.5).

3.5.1 Attributes

- MyEquipmentQ — Queue with all Equipment located at the Terminal
- MyEquipmentIdleQ — Queue with idle Equipment
- MyContainerQ — Queue with Containers to be loaded
- MyIdleVehicleQ — Queue with Idle Vehicles, ready to be loaded
- MyLoadedVehiclesQ — Queue with loaded Vehicles, ready to be unloaded
- MyUnloadPlatformQ — Queue with containers to be unloaded by Equipment (only for Lift AGV)
- MyLoadPlatformQ — Queue with containers to be loaded onto a Lift AGV (only for Lift AGV)
- NoIdleVehicles — Method: returns True if an empty ride needs to be requested
- PROCESS — Method: describes activities as a function of time

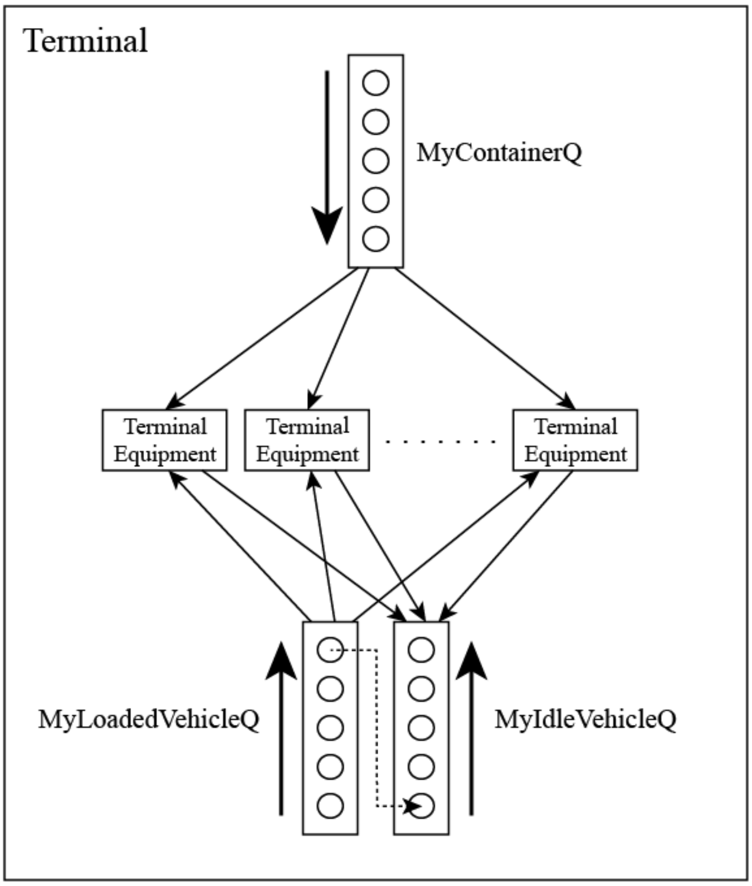


Figure 3.6: Terminal schematics for the AGV and MTS scenarios

3.5.2 Process description

Each Terminal has a process for requesting vehicles from other Terminals. These requests are called "empty rides". In order to determine when it should place such a request, the method *NoIdleVehicles* is used.

3.5.2.1 NoIdleVehicles

The method *NoIdleVehicles* returns *True* to the Terminal process if certain conditions are met, otherwise it returns *False*. The decision tree for this method can be seen in Figure 3.8. The method only returns *True* when there are no idle Vehicles available in MyIdleVehicleQ, there are Containers to be transported in MyContainerQ, and there is a Terminal in the system that has at least 2 idle Vehicles in his MyIdleVehicleQ. In case of the MTS scenario, there has to be at least half the MTS capacity in Containers to be transported available in order to return *True*. There can be only one empty ride request per container.

3.5.2.2 Terminal process

The Terminal's general process is only used to request empty rides. It starts requesting an empty ride as soon as *NoIdleVehicles* returns *True*.

- Repeat
  - Wait while *NoIdleVehicles* = *False*
  - Select Terminal with longest MyIdleVehicleQ
  - Select First Vehicle from the selected Terminal's MyIdleVehicleQ

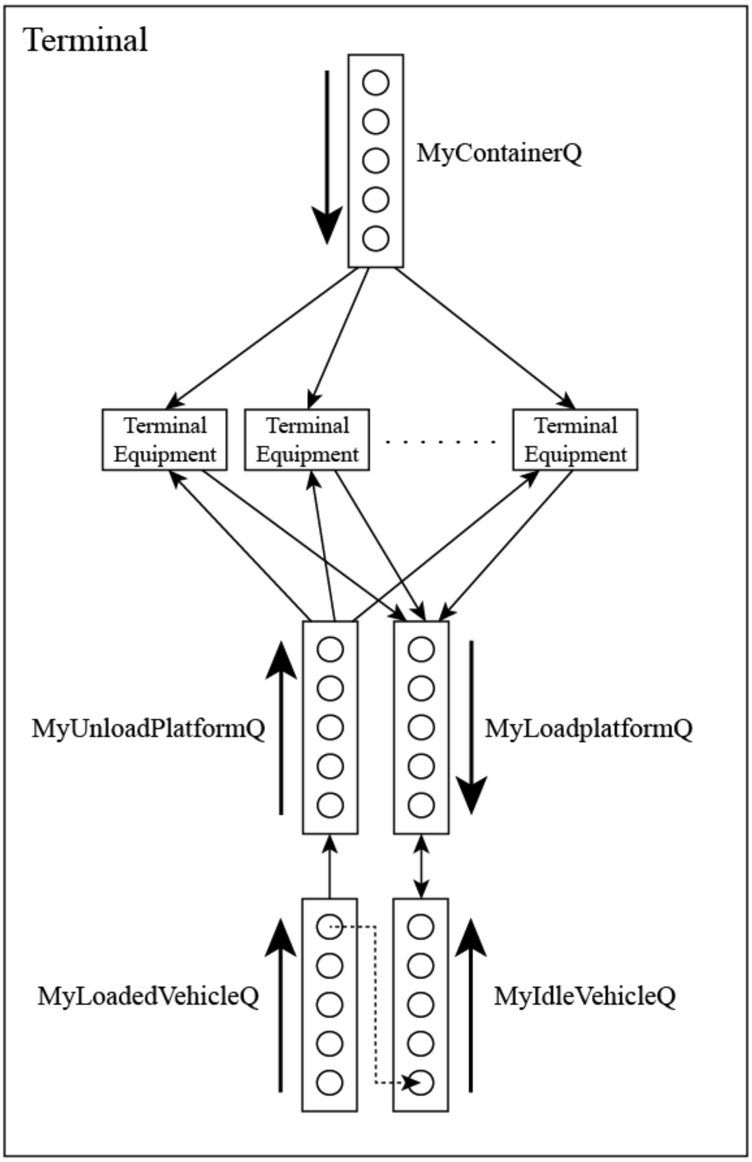


Figure 3.7: Terminal schematics for the Lift AGV scenario

- Remove Vehicle from the selected Terminal's MyIdleVehicleQ
- Set Vehicle's Destination to Self
- Make Vehicle perform an empty ride

3.5.3 Interaction with other objects

Container

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

Generator

Containers are placed in the MyContainerQ by the Generators.

Terminal Equipment

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

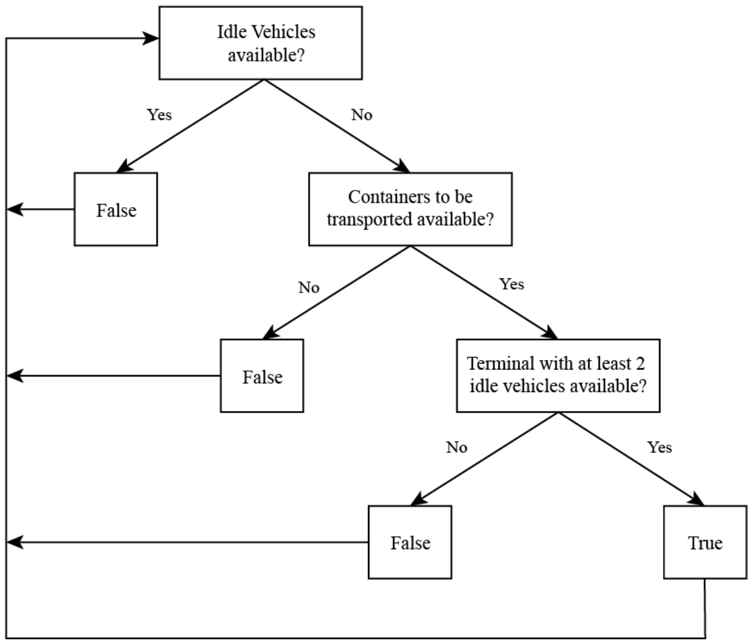


Figure 3.8: Decision tree for the NoIdleVehicles method

Vehicle

Arriving loaded Vehicles are placed in the Terminal's MyLoadedVehicleQ. Idle Vehicles located at the Terminal are placed in the MyIdleVehicleQ.

Node

A Node is made for each Terminal. They are used for Vehicle path planning (see Section 3.8).

3.5.4 Creating the Terminals

The Terminals are created by reading their properties from an input file called *inputTerminals.txt*. The input file is shown in Figure 3.9. For every Terminal 3 properties have to be specified: The Terminal's name, the type of Equipment, and the amount of Equipment located at this Terminal. Three types of equipment can be chosen: Automatic Stacking Crane (ASC), Reachstacker (RS), and Straddle Carrier (SC). If a wrong Equipment type is specified, the model will show a popup message displaying *"Select proper Terminal Equipment Type. Check inputTerminals.txt"* and the simulation will not be started.

3.6 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.

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

19 {total number of Terminals}

{Equipment type  = ASC,RS or SC}

{TerminalID - Equipment Type - TE at this terminal}
ECT ASC 4
APMT ASC 4
RWG ASC 4
T3 ASC 4
T4 ASC 2
Euromax ASC 3
CRT RS 4
RTW RS 5
BSC ASC 3
BSCHar SC 3
Kramer SC 2
Customs RS 1
Distri RS 3
ED1 SC 2
ED2 SC 4
DFDS RS 1
ChemE RS 1
ChemK1 RS 1
ChemK3 RS 1
```

Figure 3.9: Input file: *inputTerminals.txt*

3.6.1 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
• SelectTaskAGV	Method: for deciding which action to take in AGV 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

3.6.2 Process description

The Terminal Equipment uses a different process for each Vehicle type scenario: AGV, Lift AGV, and MTS (see Section 3.9). The process that is used depends on the value of variable *VehicleType* (see Section 3.2). 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.

3.6.2.1 Terminal Equipment process for the AGV scenario

In the AGV scenario, the process uses method *SelectTaskAGV* to determine which action to take. The decision tree of this method is shown in Figure 3.10. 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 3.5). After unloading, the Vehicle is immediately loaded if a Container is available. Otherwise the Vehicle is put in the MyIdleVehicleQ.

- Repeat
  - Wait while *SelectTaskAGV* = *'Wait'*
  - Leave MyEquipmentIdleQ
  - if *SelectTaskAGV* = *'Unload'*
    - Select first AGV in MyLoadedVehicleQ

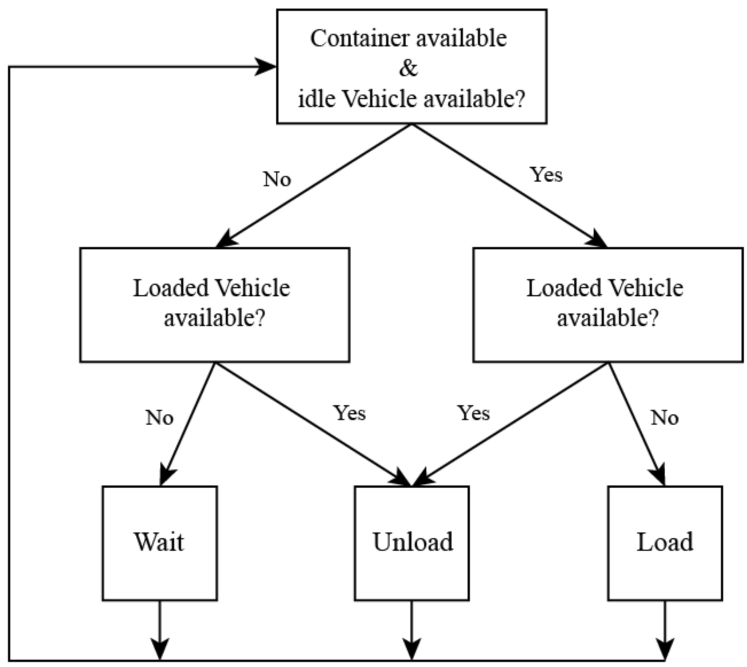


Figure 3.10: Decision tree for the *SelectTaskAGV* method

- Remove AGV from MyLoadedVehicleQ
- Select Container carried by the AGV
- Call for the Container's *CalculateExpectedHandlingTime* method
- Wait for UnloadTime.Sample
- Call for the Container's *RegisterNonPerformance* method
- Destroy the Container object
- if Container to be transported available
    - Select first Container in MyContainerQ
    - Remove Container from MyContainerQ
    - Wait for LoadTime.Sample
    - Allocate Container to selected AGV
    - Activate AGV
- else
    - Put AGV in MyIdleVehicleQ

- if *SelectTaskAGV = 'Load'*
    - Select first AGV in MyIdleVehicleQ
    - Remove AGV from MyIdleVehicleQ
    - Select first Container in MyContainerQ
    - Remove Container from MyContainerQ
    - Wait for LoadTime.Sample
    - Allocate Container to selected AGV
    - Activate AGV

- Enter MyEquipmentIdleQ

3.6.2.2 Terminal Equipment process for the Lift AGV scenario

In the Lift AGV scenario the process uses method *SelectTaskLiftAGV* to determine which action to take. The decision tree of this method is shown in Figure 3.11. 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 Lift AGVs 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 3.5). Each set of platforms has a maximum capacity currently set to 5 for each Terminal, but this could easily be made adaptable per Terminal if it turns out that it is a limiting factor.

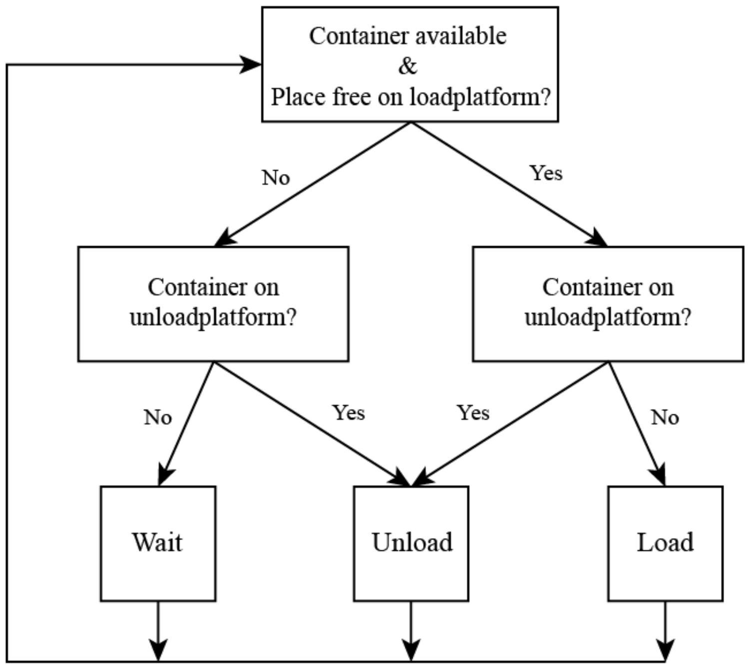


Figure 3.11: Decision tree for the *SelectTaskLiftAGV* method

The general process is described below. All queues belong to the Terminal where the Equipment is situated at (see Section 3.5).

- Repeat
  - Wait while *SelectTaskLiftAGV = 'Wait'*
  - Leave MyEquipmentIdleQ
  - if *SelectTaskLiftAGV = 'Unload'*
    - Select first Container in MyUnloadPlatformQ
    - Remove Container from MyUnloadPlatformQ
    - Wait for UnloadTime.Sample
    - Call for the Container's *RegisterNonPerformance* method
    - Destroy the Container object
  - if *SelectTaskLiftAGV = 'Load'*
    - Select first Container in MyContainerQ
    - Remove Container from MyContainerQ
    - Reserve a spot in MyLoadPlatformQ

- Wait for LoadTime.Sample
- Put Container in MyLoadPlatformQ

- Enter MyEquipmentIdleQ

3.6.2.3 Terminal Equipment process for the MTS scenario

In the MTS scenario, the process uses method *SelectTaskMTS* to determine which action to take. The decision tree of this method is shown in Figure 3.12. The decision tree is almost the same as for the AGV scenario, only in this case there have to be at least $0.5 \cdot MTSCapacity$ in Container jobs available before the Terminal Equipment is allowed to start loading the MTS. An MTS can only be handled by one piece of Equipment at the same time.

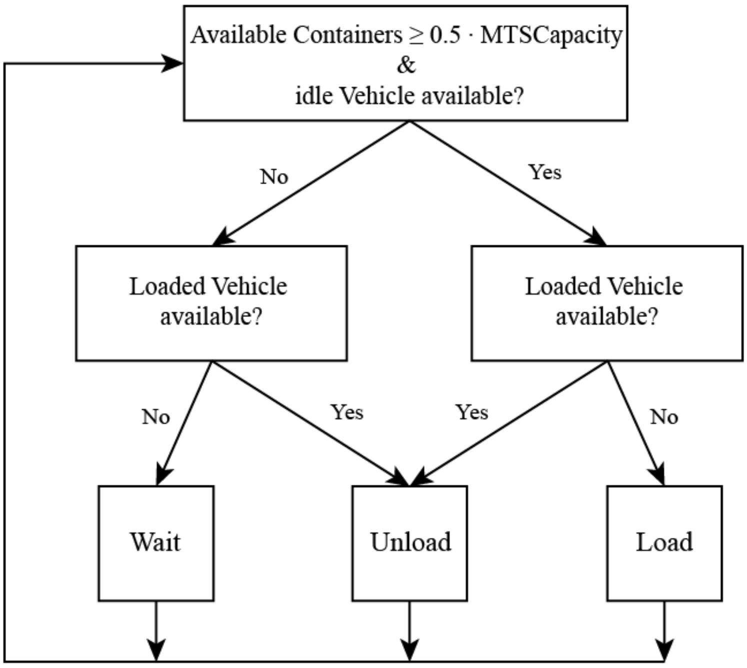


Figure 3.12: Decision tree for the *SelectTaskMTS* method

The general process is described below. All queues belong to the Terminal where the Equipment is situated at (see Section 3.5). The value ContainersToLoad determines the amount of Containers to be loaded onto the MTS. This value has to be at least half the MTS capacity and maximum the full MTS capacity. A full MTS is preferred.

- Repeat
  - Wait while *SelectTaskMTS = 'Wait'*
  - Leave MyEquipmentIdleQ
  - if *SelectTaskMTS = 'Unload'*
    - Select first MTS in MyLoadedVehicleQ
    - Remove MTS from MyLoadedVehicleQ
    - for each Container that needs to be unloaded here
      - Select Container from MTS's MyContainerQ
      - Call for the Container's *CalculateExpectedHandlingTime* method
      - Wait for UnloadTime.Sample
      - Call for the Container's *RegisterNonPerformance* method

    - Destroy the Container object
  - Activate the MTS
- if *SelectTaskMTS* = *'Load'*
  - Select first MTS in MyIdleVehicleQ
  - Remove MTS from MyIdleVehicleQ
  - Determine ContainersToLoad
  - for ContainersToLoad
    - Select first Container in MyContainerQ
    - Remove Container from MyContainerQ
    - Wait for LoadTime.Sample
    - Put Container in MTS's MyContainerQ
  - Activate the MTS
- Enter MyEquipmentIdleQ

3.6.3 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 MyEquipmentQueue and utilizes the other queues belonging to its Terminal to perform its tasks.

Vehicle

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

3.6.4 Creating the Terminals

The type and amount of Terminal Equipment created per Terminal is defined in an input file called *inputTerminals.txt*, as explained in Section 3.5. The properties of the different types of Terminal Equipment can be defined in the input file *inputTerminalEquipment.txt* (see Figure 3.13). At this point only the different Equipment types and their load and unload times can be defined.

```
{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}
```

Figure 3.13: Input file: *inputTerminalEquipment.txt*

3.7 Infrastructure

The infrastructure consists of two different object types: Roads and Intersections. Every single Intersection, and every piece of Road between different Intersections or Terminals, is a separate object with its own ID. The Intersections are defined with IDs I_{01} to I_n, where $n =$ the total amount of Intersections. The Roads are defined with IDs R_{01a} to R_{ma} and R_{01b} to R_{mb}, where $m =$ the total amount of two-way Road sections. R_{01a} Is the opposite Road of R_{01b}. A map of all Roads, Intersections, and Terminals for the Maasvlakte can be seen in Figure 3.14. A larger version of the map, including the Road lengths is given in Appendix A.

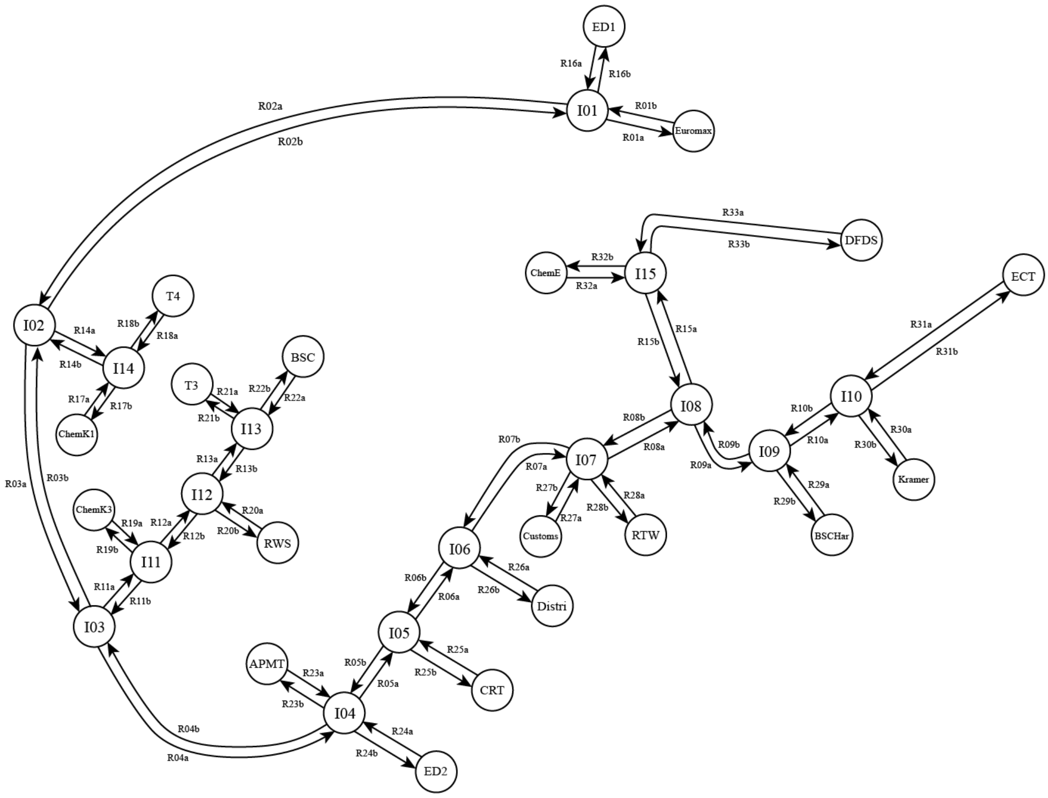


Figure 3.14: Schematic map of the Maasvlakte showing all Terminals, Intersections, and Roads

Every Intersection and Road has its own queue which holds all Vehicles that are present there on a certain time. This makes it possible to see how busy these Roads and Intersections are.

3.7.1 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. A schematic representation of a two-way Road can be seen in Figure 3.15.

3.7.1.1 Attributes

- Length — Length of the Road in meters
- StartNode — Node at which the Road starts (see Section 3.8)
- EndNode — Node at which the Road ends (see Section 3.8)
- MyTrafficQ — Queue containing all Vehicles on the Road at that time

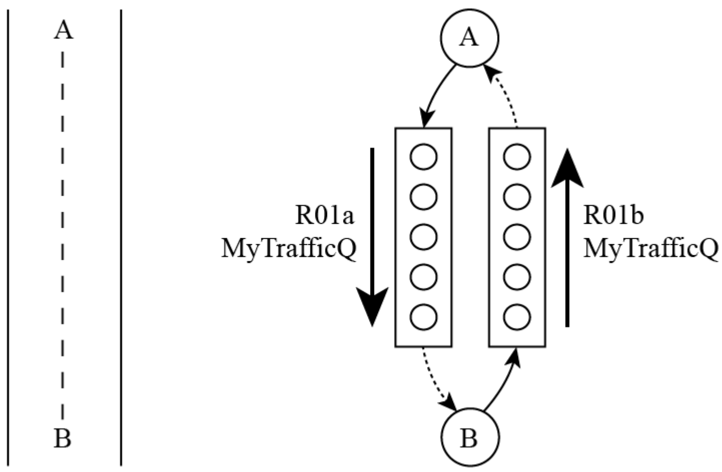


Figure 3.15: Schematic representation of a two-way Road

3.7.1.2 Interaction with other objects

Vehicle

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

Node

Roads form the arcs between the different Nodes.

3.7.1.3 Creating the Roads

The Roads are created by reading their properties from an input file called *inputRoads.txt*. Part of the input file is shown in Figure 3.16. The properties that need to be defined are the Road's ID, Length, Start Node, and End Node.

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

66 {total number of roads}

{RoadID - Length - Start node - end node}

R01a 1510 Euromax I01
R01b 1510 I01 Euromax
R02a 5510 I01 I02
R02b 5510 I02 I01
R03a 2880 I02 I03
R03b 2880 I03 I02
R04a 2520 I03 I04
R04b 2520 I04 I03
R05a 470 I04 I05
R05b 470 I05 I04
R06a 470 I05 I06
R06b 470 I06 I05
R07a 1080 I06 I07
R07b 1080 I07 I06
R08a 180 I07 I08
R08b 180 I08 I07
R09a 540 I08 I09
R09b 540 I09 I08
```

Figure 3.16: Input file: *inputRoads.txt*

If a spelling mistake is made in defining a Road's Start or End Node, a popup message will be displayed saying exactly for which Road the mistake was made, and the simulation will not be started.

3.7.2 Intersections

The Intersections don't have a process. The time it takes a Vehicle to cross an Intersection is a set value called TimeToCross. The busiest Intersections have the highest TimeToCross. Every Vehicle that has to pass the Intersection enters the Intersection's MyTrafficQ, independent of which direction it is coming from. It stays in the MyTrafficQ for the set TimeToCross. A schematic representation of an Intersection can be seen in Figure 3.17.

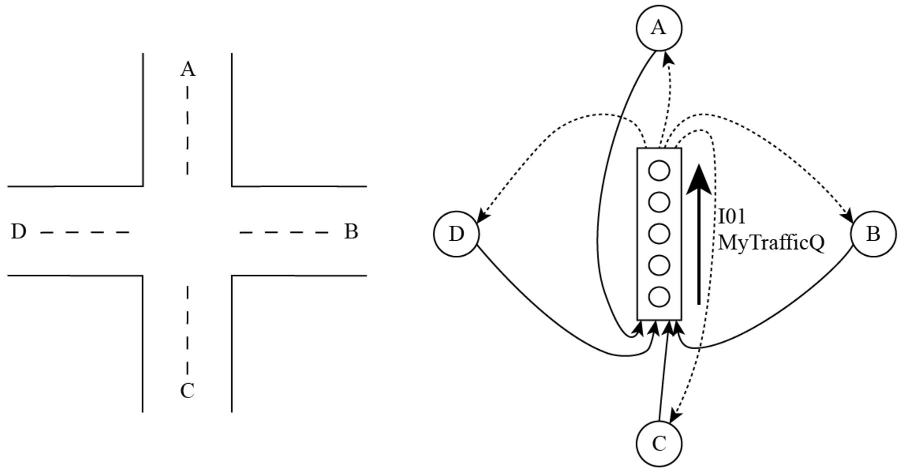


Figure 3.17: Schematic representation of an Intersection

3.7.2.1 Attributes

- TimeToCross Time it takes to cross the Intersection in minutes
- MyTrafficQ Queue containing all Vehicles on the Intersection at that time

3.7.2.2 Interaction with other objects

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 3.8).

3.7.2.3 Creating the Intersections

The Intersections are created by reading their properties from an input file called *inputIntersections.txt*. The input file is shown in Figure 3.18. The properties that need to be defined are the Intersection's ID and its TimeToCross. The TimeToCross values are pure estimations and not based on actual data.

3.8 Nodes

Nodes are automatically created for every Intersection and Terminal. They are used by the Vehicle's methods *FindShortestRoute* and *DriveShortestRoute* (see Section 3.9) for finding and driving the shortest route from origin to destination. By having a StartNode and EndNode attribute, the Roads form the arcs between the different Nodes.

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

15 {total number of intersections}

{IntersectionID - TimeToCross}
I01 0,5
I02 1
I03 2
I04 2,5
I05 2,5
I06 2
I07 2
I08 1,5
I09 1,5
I10 1
I11 2
I12 2
I13 1,5
I14 0,5
I15 0,1
```

Figure 3.18: Input file: *inputIntersections.txt*

3.8.1 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 3.9.2.1

3.8.2 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.

Intersection

A Node is created for every Intersection

Road

Roads form the arcs between the different Nodes.

3.9 Vehicles

Three different Vehicle types are modeled: AGVs, Lift AGVs and MTSs. All 3 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* (see Section 3.2). The Vehicle has three methods that can be called for by each Vehicle type: *FindShortestRoute*, *DriveShortestRoute*, and *DoEmptyRide*.

3.9.1 Attributes

• SpeedLoaded	Vehicle speed when loaded [m/s]
• SpeedEmpty	Vehicle speed when empty [m/s]
• MyRouteQ	Queue containing all Nodes on route
• MyContainer	The Container onboard (for AGV and Lift AGV)
• MyContainerQ	Queue containing all Container onboard (for MTS)
• 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 [min]
• LiftUnloadTime	Time it takes the Lift AGV to unload a Container [min]
• 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

3.9.2 Process description

3.9.2.1 FindShortestRoute

The method *FindShortestRoute* uses the Dijkstra path planning algorithm to find the shortest path from its origin to its destination. 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

3.9.2.2 DriveShortestRoute

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.

- Set Speed depending on the Vehicle being empty or loaded
- Repeat
    - Find Road between the first two Nodes in MyRouteQ
    - Enter the Road's MyTrafficQ
    - Wait for $\frac{Roadlength}{Speed*60}$ [min]
    - 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
    - Enter the Intersection's MyTrafficQ
    - Wait for the Intersection's TimeToCross [min]
    - Leave the Intersection's MyTrafficQ

3.9.2.3 DoEmptyRide

The procedure of doing an empty ride is the same for all three 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. The process goes as follows:

- Call for *FindShortestRoute*
- Call for *DriveShortestRoute*
- Set MyOrigin to current MyDestination
- Enter MyIdleVehicleQ at MyDestination Terminal

3.9.2.4 AGV process

The AGV process can be activated in two ways: by Terminal Equipment after being loaded with a Container, and by a Terminal for performing an empty ride. An AGV is able to carry one container at the same time. It is loaded and unloaded by Terminal Equipment.

- Repeat
    - Wait while not activated by Terminal Equipment or for empty ride
    - if activated by Terminal Equipment
        - Call for *FindShortestRoute*
        - Call for *DriveShortestRoute*
        - Set MyOrigin to current MyDestination
        - Enter MyLoadedVehicleQ at MyDestination Terminal
    - else
        - *DoEmptyRide*

3.9.2.5 Lift AGV process

The Lift AGV 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 for performing an empty ride. The Lift AGV is able to load and unload containers from the platform queues at the Terminals and doesn't have to wait for the Terminal Equipment to load or unload a Container from the Vehicle. The Lift AGV is able to carry one container at the same time.

- Repeat
  - Wait while no Containers on MyOrigin's MyLoadPlatformQ or activated for empty ride
  - if Container available on MyOrigin's MyLoadPlatformQ
    - Leave MyOrigin's MyIdleVehicleQ
    - Select first Container from MyOrigin's MyLoadPlatformQ
    - Remove Container from MyOrigin's MyLoadPlatformQ
    - Wait for LiftLoadTime
    - Call for *FindShortestRoute*
    - Call for *DriveShortestRoute*
    - Call for Container's *CalculateExpectedHandlingTime*
    - Set MyOrigin to current MyDestination
    - Enter MyLoadedVehicleQ at MyDestination Terminal
    - Wait while MyDestination's MyUnloadPlatformQ is full
    - Leave MyLoadedVehicleQ at MyDestination Terminal
    - Reserve a spot in MyDestination's MyUnloadPlatformQ
    - Wait for LiftUnloadTime
    - Put Container in MyDestination's MyUnloadPlatformQ
    - Enter MyDestination's MyIdleVehicleQ
  - else
    - *DoEmptyRide*

3.9.2.6 MTS process

Just like the AGV, the MTS can be activated by both the Terminal Equipment after being loaded and by a Terminal for performing an empty ride. Although the MTS can also be activated by the Terminal Equipment after unloading part of its Containers before its empty. For simplification an MTS is simulated as a Vehicle that is able to carry a certain amount of Containers. This amount of Containers is specified as MTSCapacity. The MTS does not consist of a separate tractor and trailer as in reality, but it is able to drive a lot faster when empty than when full.

- Repeat
  - Wait while not activated by Terminal Equipment or for empty ride
  - if activated by Terminal Equipment after loading
    - Repeat until MyContainerQ is empty
      - Find and select Container with the closest Destination
      - Call for *FindShortestRoute*
      - Call for *DriveShortestRoute*
      - Set MyOrigin to current MyDestination

    - Enter MyLoadedVehicleQ at MyDestination Terminal
    - Wait until all Containers with this Destination are unloaded
  - Enter MyDestination's MyIdleVehicleQ
- else
  - *DoEmptyRide*

3.9.3 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 MyIdleVehicleQ and MyLoadedVehicleQ.

Terminal Equipment

Containers are transferred to and from the Vehicles by the Terminal Equipment, except for in the Lift AGV scenario.

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 MyTrafficQ in order to reach their destination.

Node

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

3.9.4 Creating the Vehicles

The Vehicles are created by reading their properties from an input file, depending on which Vehicle type is selected. The input files are *inputAGVs.txt* (Figure 3.19), *inputLiftAGVs.txt* (Figure 3.20), and *inputMTSs.txt* (Figure 3.21). As discussed in Section 3.2, the type and amount of Vehicles can be specified in *configfile.txt* (Figure 3.4). After creation, Vehicles are evenly distributed over all Terminals.

For the AGV only the driving speed for full and empty rides has to be specified. For the Lift AGV it is also necessary to specify the time it takes to load or unload a Container. For the MTS it has to be specified how much Containers it is able to carry.

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

3 {vehicle speed when loaded in m/s}
6 {vehicle speed when empty in m/s}
5 {no. of containers per MTS}
```

Figure 3.19: Input file: inputAGVs.txt

3.10 Changing the input

As described in the previous Sections, the input of the model consists of several input files. These files can be easily altered in order to use different transport demands, road maps, or equipment and

```
{Input file containing properties of the Lift AGVs}

3 {vehicle speed when loaded in m/s}
4 {vehicle speed when empty in m/s}

0,5 {Container loadtime}
1 {Container unloadtime}
```

Figure 3.20: Input file: inputLiftAGVs.txt

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

3 {vehicle speed when loaded in m/s}
6 {vehicle speed when empty in m/s}
5 {no. of containers per MTS}
```

Figure 3.21: Input file: inputMTSs.txt

vehicle properties. Changing the properties of the equipment simply means changing some values in their associated input file.

Changing the transport demand

In order to change the transport demand a new origin-destination matrix has to be created, and this has to be translated to a number of Generators to be defined in *inputgenerators.txt* (see Section 3.4.4). As long as the road map stays the same, this is all that has to be done.

Changing the road map

Changing the road map would require to make a new map similar to the one in Figure 3.14. The different Roads, Intersections, and Terminals, combined with all their properties, would have to be defined in *inputRoads.txt*, *inputIntersections.txt*, and *inputTerminals.txt*.

3.11 Summary

This chapter has explained how the ITT simulation model works. The model was built using Delphi and TOMAS, and is object oriented. It consists of the following objects: Containers, Generators, Roads, Intersections, Terminals, Nodes, Terminal Equipment and Vehicles. Every object has been discussed in detail. The general simulation parameters have to be defined in the configuration file. The properties of the objects have to be defined in the input files.

The next chapter will show what the output of the model looks like. This includes the model interface, the output files, and the graphs.

Chapter 4

Model output

This chapter will show the output of the ITT simulation model. First the interface of the model will be shown, then the various input files, and the chapter will be concluded with information on the graphs that are created using TOMAS Collections.

4.1 Interface

The interface of the ITT simulation model is shown in Figure 4.1. 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 the general simulation parameters that have been defined in the configuration file (see Section 3.2). Also it will start showing the current values of some of the main performance indicators: non-performance, mean overall Terminal Equipment occupancy, mean overall Vehicle occupancy, mean overall idle Vehicle waiting time, total number of rides, percentage of empty rides, total amount of Containers handled, and the total distance traveled by the Vehicles. 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 4.1. After clicking the *Quit* button, the programm will be closed and the results of the simulation will be saved away to various output files.

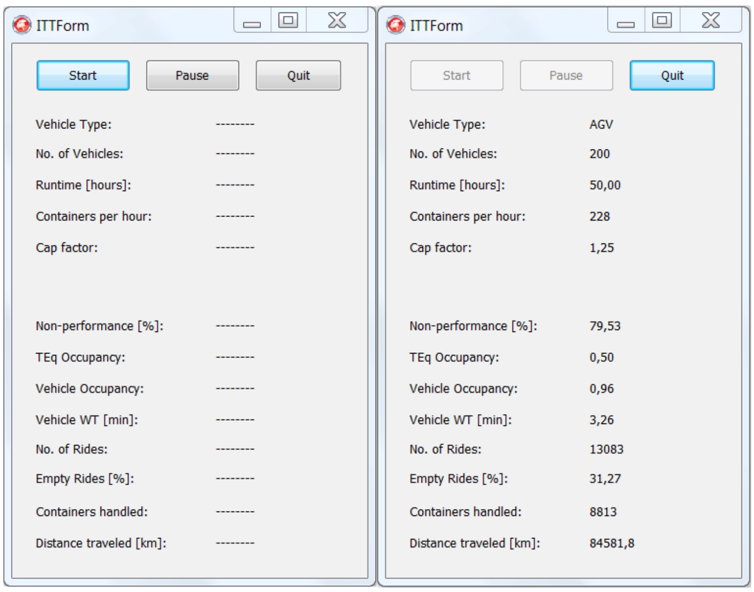


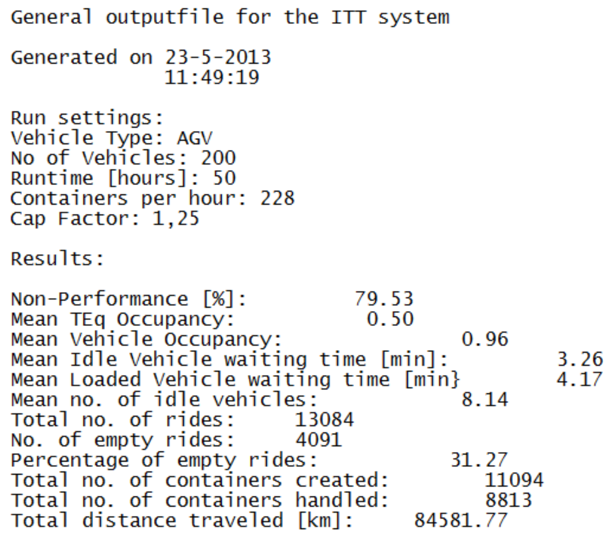
Figure 4.1: Interface of the ITT simulation model; Left: before simulation; Right: after simulation

4.2 Output files

At the end of the simulation, after clicking the *Quit* button, the simulation results are saved to 3 different output files: *outputGeneral.txt*, *outputRoadsIntersections.txt*, and *outputTerminals.txt*.

4.2.1 General output

The General results are saved to the file *outputGeneral.txt*. An example of the file is shown in Figure 4.2. 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.


```
General outputfile for the ITT system

Generated on 23-5-2013
             11:49:19

Run settings:
Vehicle Type: AGV
No of Vehicles: 200
Runtime [hours]: 50
Containers per hour: 228
Cap Factor: 1,25

Results:

Non-Performance [%]:         79.53
Mean TEq Occupancy:           0.50
Mean Vehicle Occupancy:              0.96
Mean Idle Vehicle waiting time [min]:        3.26
Mean Loaded Vehicle waiting time [min}       4.17
Mean no. of idle vehicles:           8.14
Total no. of rides:     13084
No. of empty rides:     4091
Percentage of empty rides:          31.27
Total no. of containers created:       11094
Total no. of containers handled:        8813
Total distance traveled [km]:     84581.77
```

Figure 4.2: Output file: outputGenerators.txt

The following results are shown:

- Non-Performance, in percentage
- Mean overall Terminal Equipment occupancy
- Mean overall Vehicle occupancy
- Mean idle Vehicle waiting time, in minutes
- Mean loaded Vehicle waiting time, in minutes
- Mean number of idle Vehicles
- Total number of rides
- Number of empty rides
- Percentage of empty rides, of total
- Total number of containers created
- Total number of containers handled
- Total distance travel, in km

4.2.2 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 4.3. 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 roads and intersections of the ITT system

Generated on 23-5-2013
             13:17:51

Run settings:
Vehicle Type: AGV
No of Vehicles: 200
Runtime [hours]: 50
Containers per hour: 228

Name | TotalPassed | Mean Veh/h | Max Veh/h | Mean Veh | Max Veh

I01     2023          40.46      86             0.34      5
I02     3332          66.64      130            1.11      8
I03     8825         176.50      234            5.89      20
I04     9618         192.36      227            8.02      22
I05     8914         178.28      213            7.43      23
I06     6436         128.72      165            4.29      18
I07     6185         123.70      172            4.12      15
I08     5187         103.74      165            2.59      16
I09     5045         100.90      166            2.52      14
I10     4348          86.96      157            1.45      13
I11     6685         133.70      201            4.46      21
I12     7068         141.36      223            4.71      20
I13     4263          85.26      147            2.13      14
I14     1583          31.66      91             0.26      6
I15     198            3.96      14             0.00      2
R01a    748           14.96      34             1.55      11
R01b    739           14.78      33             1.49      10
R02a    1015          20.30      48             7.71      27
R02b    1002          20.04      44             7.43      28
R03a    1542          30.84      62             6.03      22
R03b    1517          30.34      62             5.63      19
R04a    3959          79.18      105           13.42      31
R04b    3943          78.86      122           12.53      32
```

Figure 4.3: Part of 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

4.2.3 Output per Terminal

The results per Terminal are saved to the output file *outputTerminals.txt*. An example of the file is shown in Figure 4.4. 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 23-5-2013
             13:17:51

Run settings:
Vehicle Type: AGV
No of Vehicles: 200
Runtime [hours]: 50
Containers per hour: 228

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

ECT      ASC    2     0.83    1.24    3.02   0.56    48.81   1118   12.50   544    6.17
APMT     ASC    2     0.52    2.73    0.80   0.37     2.13   609     6.81   438    4.97
RWG      ASC    3     0.68    0.76    1.00   0.28    22.87   1353   15.13   627    7.11
T3       ASC    3     0.60    0.80    0.65   0.20    23.70   1151   12.87   624    7.08
T4       ASC    2     0.51    1.54    1.29   0.22    47.90   653     7.30   348    3.95
Euromax  ASC    2     0.63    4.95    4.57   0.33   171.91   694     7.76   597    6.77
CRT      RS     5     0.61    1.65    0.90   0.87     0.21   588     6.58   1859  21.09
RTW      RS     1     0.91   12.94   55.07   1.00     1.18   250     2.80   456    5.17
BSC      ASC    2     0.64    3.64    2.00   0.69     0.72   464     5.19   868    9.85
BSCHar   SC     2     0.39    6.46    0.94   0.51     1.64   263     2.94   385    4.37
Kramer   SC     1     0.49    9.95    3.47   0.71     0.39   131     1.47   287    3.26
Customs  RS     1     0.28   18.12    1.64   0.57     4.30   105     1.17   125    1.42
Distri   RS     2     0.29    7.61    0.24   0.67     0.34   152     1.70   319    3.62
ED1      SC     2     0.31    5.94    0.21   0.18   116.78   270     3.02   236    2.68
ED2      SC     5     0.51    3.22    0.23   0.89   476.12   1024   11.45   978   11.10
DFDS     RS     1     0.17   42.55    1.02   3.34    19.21   70      0.78   63     0.71
ChemE    RS     1     0.07   59.04    0.02   2.52     2.69   25      0.28   25     0.28
ChemK1   RS     1     0.03   55.46    0.26   0.84     3.63   11      0.12   15     0.17
ChemK3   RS     1     0.04   90.28    0.11   0.81     0.56   10      0.11   19     0.22
```

Figure 4.4: 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
- Terminal Equipment occupancy
- Mean waiting time for Vehicles in MyIdleVehicleQ
- Mean waiting time for Vehicles in MyLoadedVehicleQ
- Mean number of idle Vehicles
- Mean length of MyContainerQ, so the mean number of Containers waiting to be transported
- Number of Containers loaded
- Percentage of Containers loaded
- Number of Containers unloaded
- Percentage of Containers unloaded

4.3 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, in percentage
- Mean idle Vehicle waiting time, in minutes
- Mean idle Vehicle waiting time per Terminal, in minutes
- 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.

4.4 Summary

This chapter has shown the output of the ITT simulation model. During the simulation, the interface will show the general simulation parameters that have been defined in the configuration file. When the simulation is finished and the Quit button has been clicked, the results of the simulation will be saved to several output files. There is a general output file, an output file that shows the traffic flow per Road and Intersection, and an output file that show the results per Terminal. Data is being gathered during simulation that can be viewed live in various graphs on the TOMAS Collections form. After clicking the Quit button the gathered data is written away to a number of .csv files. These files can be imported in spreadsheet software for further analysis.

The next chapter will discuss the evaluation of the simulation model by the hand of a number of simulation runs based on the transport demand scenario presented in Chapter 2. It will show how the gathered data can be used to analyze the difference between various scenarios.

Chapter 5

Model evaluation

This chapter will discuss the evaluation of the ITT simulation model and will show how the gathered data can be used to analyze the difference between various scenarios. In order to evaluate the developed simulation model, a number of simulation runs have been performed for the scenario described in Chapter 2. The exact values of the simulation results are not very important at this point, they merely serve to verify that a certain change of parameters gives the expected result. For instance: it is expected that using more vehicles would result in a lower non-performance.

Unfortunately the model cannot be completely validated since no real life data is available to which the simulation results could be compared. For proper validation, data would be needed from a real life system that could be implemented in the model. This data includes waiting times, vehicle occupancy, terminal equipment occupancy, and driving times.

5.1 Simulation settings

Unless otherwise noted, simulations have been performed using the following settings:

Vehicle properties (see Section 3.9)

• AGV speed when loaded	4 m/s
• AGV speed when empty	5 m/s
• Lift AGV speed when loaded	3 m/s
• Lift AGV speed when empty	4 m/s
• Lift AGV's container load time	0,5 min
• Lift AGV's container unload time	1 min
• MTS speed when loaded	3 m/s
• MTS speed when empty	6 m/s
• MTS capacity	5 Containers

Terminal Equipment properties (see Section 3.6)

• ASC unload time	3 min
• ASC load time	3 min
• SC unload time	4 min
• SC load time	4 min
• RS unload time	4 min
• RS load time	4 min

Terminal Equipment per Terminal (see Section 3.5)

Table 5.1 shows the amount of Equipment used per Terminal. Calculations are based on the amount of Containers to be handled for an average hour and the capacity (in moves per hour) of the type of Equipment situated at the Terminal. Values are rounded up to determine the number of Terminal Equipment per Terminal.

From: \ To:	**Cont/h peak**	**Cont/h avg**	**Eq. Type**	**Eq. capacity**	**# Eq.**	**# Eq. used**
ECT	119	40	ASC	20	2,0	2
APMT	88	29	ASC	20	1,5	2
RWG	142	47	ASC	20	2,4	3
T3	129	43	ASC	20	2,2	3
T4	81	27	ASC	20	1,4	2
Euromax	108	36	ASC	20	1,8	2
CRT	185	62	RS	15	4,1	5
RTW	47	16	RS	15	1,0	1
BSC	87	29	ASC	20	1,5	2
BSCHartel	51	17	SC	15	1,1	2
Kramer	28	9	SC	15	0,6	1
Customs	15	5	RS	15	0,3	1
Distripark	33	11	RS	15	0,7	1
ED1	48	16	SC	15	1,1	2
ED2	190	63	SC	15	4,2	5
DFDS	11	4	RS	15	0,2	1
ChemE	3	1	RS	15	0,1	1
ChemK1	2	1	RS	15	0,0	1
ChemK3	2	1	RS	15	0,0	1
Total:	1369	456				38

Table 5.1: Equipment per Terminal

Cap factor

The Cap factor is by standard set to 1,5. So the Containers are allowed to be delivered 50% of their expected handling time late before they're accounted as non-performance.

Infrastructure

The road map is as shown in Appendix A, including the distances displayed in the figure. The Intersection's TimeToCross values are as shown in the input file in Figure 3.18.

Transport demand

The transport demand is as described in Chapter 2. The average number of Containers per hour of 228 Containers has been used.

Runtime

Runtime has been set to 10 weeks, or 1680 hours. Figure 5.1 shows the non-performance results for one single run with 300 AGVs. As can be seen from the Figure, the model needs about one to two week to approach an equilibrium position.

5.2 Validation of generated Container flows

The by the Generators generated container flows have been compared to the input flows defined in the O-D matrix in Section 2.1. The O-D matrix in Figure 5.2 shows the deviations between the generated flows and the input flows. As can be seen from the Figure, the deviations are relatively small. The deviation of the total amount of generated Containers is only 0,02%. The average amount of Containers created per hour is 228,07, with a maximum of 281 and a minimum of 184.

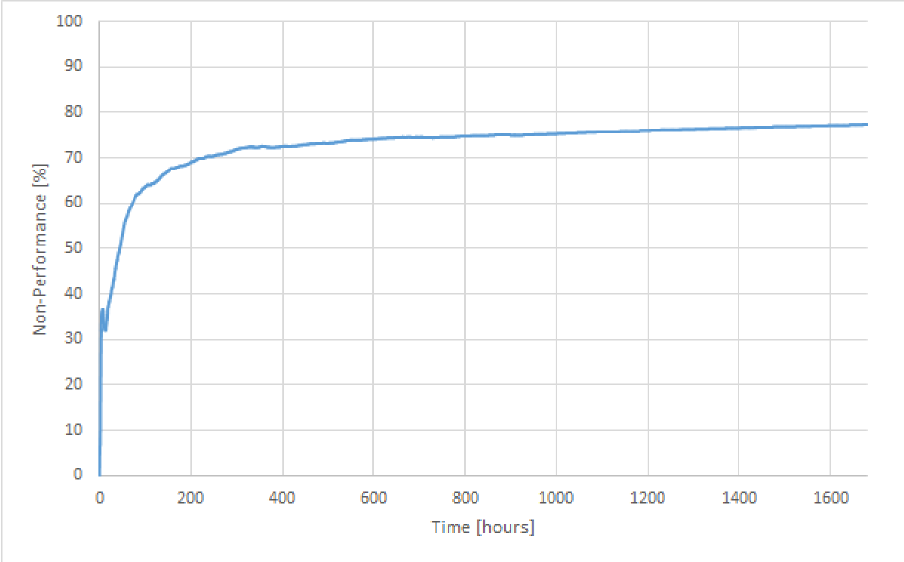


Figure 5.1: Non-Performance for one single run

From: \ To:	ECT	APMT	RWG	T3	T4	Euromax	CRT	RTW	BSC	BSCHartel	Kramer	Customs	Distripark	ED1	ED2	DFDS	ChemE	ChemK1	ChemK3	Total
ECT		4,0	-11,0	0,0	5,7	-0,6	-0,8		1,4			-3,2	1,7		0,3					-0,2
APMT	0,9		3,3	-3,5	-14,2	-5,3	1,4		-3,6	-1,2	1,4	3,4	0,2		0,9					0,3
RWG	0,0	2,8			6,3	0,0	-0,3	0,8	-1,0	-0,8	-0,3	-1,8	1,1		0,8					0,0
T3	-0,3	-0,6			-22,0	-1,1	1,3	0,4	1,8	-2,9		3,4	3,2		-0,9					0,3
T4	-1,2	-1,8	1,7	1,3			-0,7	1,0	-0,1	0,9		-2,7	2,5	-0,2						0,0
Euromax	1,1	-1,8	-1,3	0,4			0,5	2,0	5,6	1,5		6,7	0,8							1,1
CRT	2,4	-1,1	-2,0	-2,6	-2,1	-3,3								-0,2	-0,1	0,0		0,0	0,0	-1,0
RTW			-1,7	-1,8	2,1	0,5										-1,6	-4,5			-1,0
BSC	4,4	-0,7	-2,8	3,4	2,2	-2,0								-0,7	1,6	-2,5				0,5
BSCHartel		7,4	-1,5	-0,4	-6,7	5,0									-0,2	5,8				0,6
Kramer		6,0	2,4										11,1							4,8
Customs	1,9	3,4	-6,1	-1,3	-2,7	1,1														-0,9
Distripark								-6,2		3,3										-1,3
ED1					1,3		-0,5		-1,0											0,6
ED2	-0,4	-0,6	-0,3	-0,4			-2,0		-1,3	-0,6										-0,7
DFDS							1,8	-8,6	-5,1	3,6										-0,9
ChemE								-4,5												-4,5
ChemK1							-7,1													-7,1
ChemK3							7,1													7,1
Total	0,3	0,7	-0,7	-0,4	0,3	-0,8	0,1	-0,1	-0,2	-0,2	0,3	0,9	2,0	-0,4	0,3	0,0	-4,5	0,0	0,0	**0,02**

Figure 5.2: Deviations between the input and output flows of the Generators

5.3 Results

5.3.1 General performance measurements

A series of simulations with varying numbers and types of Vehicles have been performed. For each Vehicle type 9 simulations were done, with numbers varying from 100 to 450 in steps of 50 Vehicles. Figure 5.3 shows the non-performance results of these simulations as a function of the number of Vehicles for the AGV, Lift AGV, and the MTS. The non-performance for all simulations is quite high, which is logical since no planning is involved at this point. A Terminal only starts requesting an empty ride when it notices there are no idle Vehicles available anymore while there are Containers to be transported. This will immediately result in non-performance because the Vehicle still has to drive to the Terminal to pick up the Container. A smart planning system could solve this problem.

Both the AGV and Lift AGV have a downward trend in non-performance when more Vehicles are used. This is mainly due to the fact that the Terminal Equipment has to spend less time waiting for available Vehicles. The Lift AGV performs best, because here the container transport is decoupled from the storage process. The MTS also shows a downward trend, but it stabilizes after about 200 Vehicles. This mainly has to do with the way the non-performance is calculated. The expected handling time calculations assume that a Container is brought straight to its destination, while the MTS takes a route past different Terminals. So all Containers that are not delivered as first of the load will almost always result in non-performance.

Figure 5.4 shows the mean Terminal Equipment occupancy as a function of the number of Vehicles. All graphs show an upward trend that stabilizes after a certain number of Vehicles. The reason that the average Terminal Equipment occupancy doesn't get higher than 0,67 is because there are a couple of

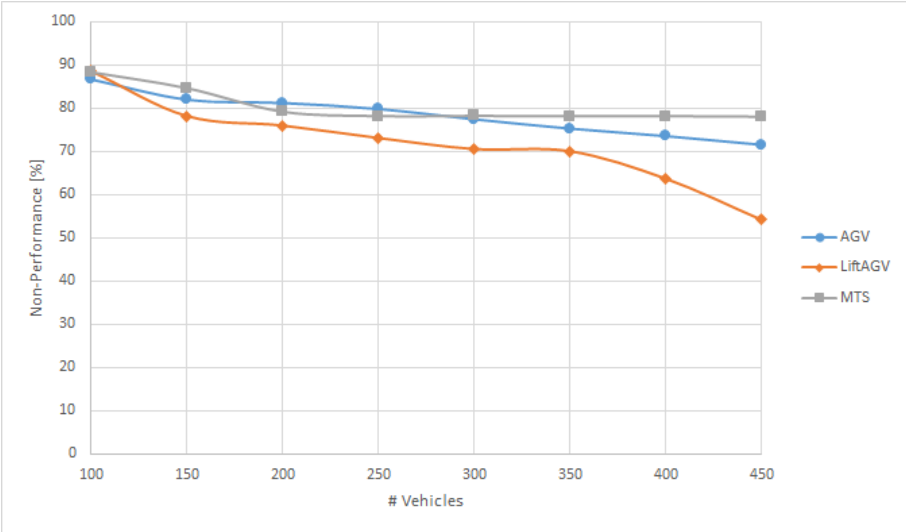


Figure 5.3: Non-performance as a function of the number of Vehicles

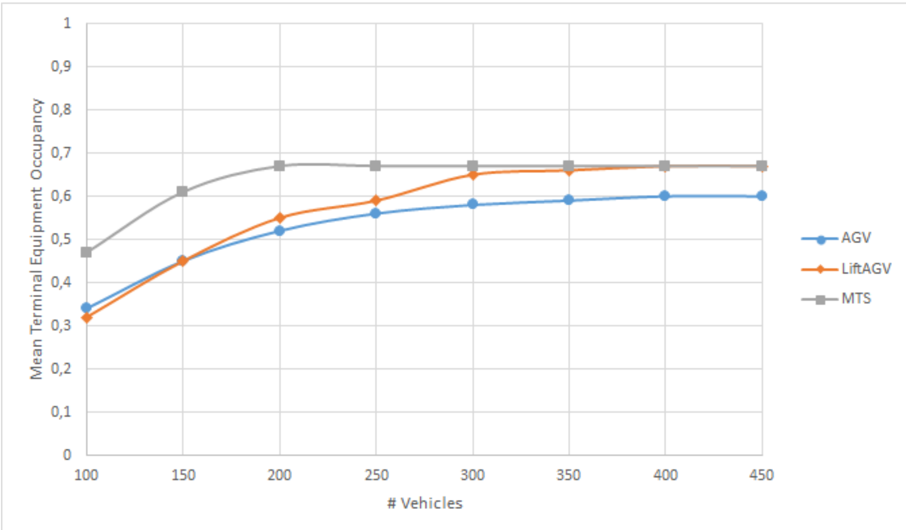


Figure 5.4: Mean Terminal Equipment occupancy as a function of the number of Vehicles

Terminals in the system with very little throughput which already have a huge over capacity with just one piece of Terminal Equipment (see for example Figure 4.4). Using AGVs results in a lower occupancy than when using Lift AGVs because in the Lift AGV scenario the Terminal Equipment doesn't have to wait for the Vehicles to load or unload. The MTS scenarios result in a higher occupancy at lower amounts of Vehicles simply because the MTSs have a higher capacity than the AGVs and Lift AGVs.

Figure 5.5 shows the mean Vehicle occupancy as a function of the number of Vehicles. The MTS occupancy starts to drop down from a much lower amount of Vehicles because of the MTS's higher capacity. The AGV and the Lift AGV have the same capacity, but the results imply that under current conditions there would be fewer Lift AGVs necessary to operate the system than AGVs. These observations also coincide with the mean idle Vehicle waiting times shown in Figure 5.6.

Figure 5.7 shows the percentage of empty rides out of the total amount of rides as a function of the number of Vehicles. The 3 Vehicle types show quite different graphs, but all seem to reach a certain peak after which the percentage of empty rides starts to drop. These peaks coincide with the points in Figure 5.5 where the mean Vehicle occupancy starts to drop. So before these peaks there are less empty rides because most vehicles are busy, and after these peaks there are less empty rides because there are more Vehicles available in the system.

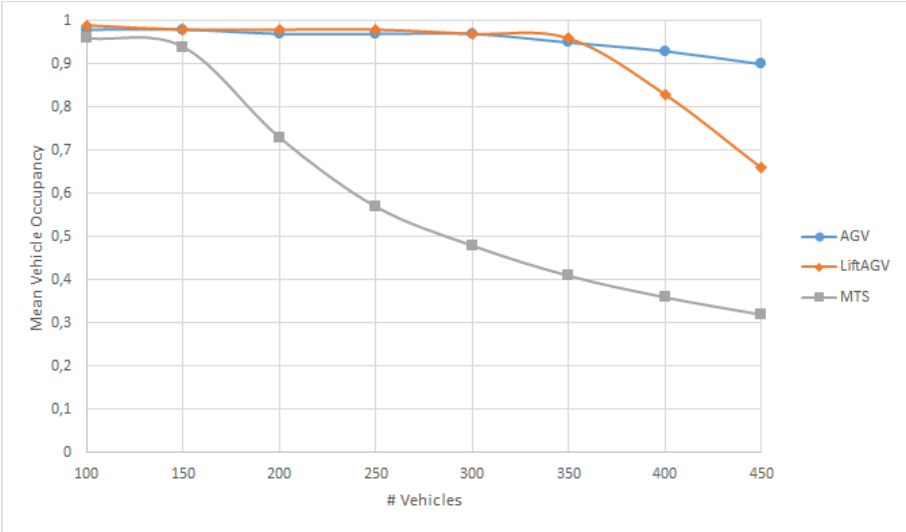


Figure 5.5: Mean Vehicle occupancy as a function of the number of Vehicles

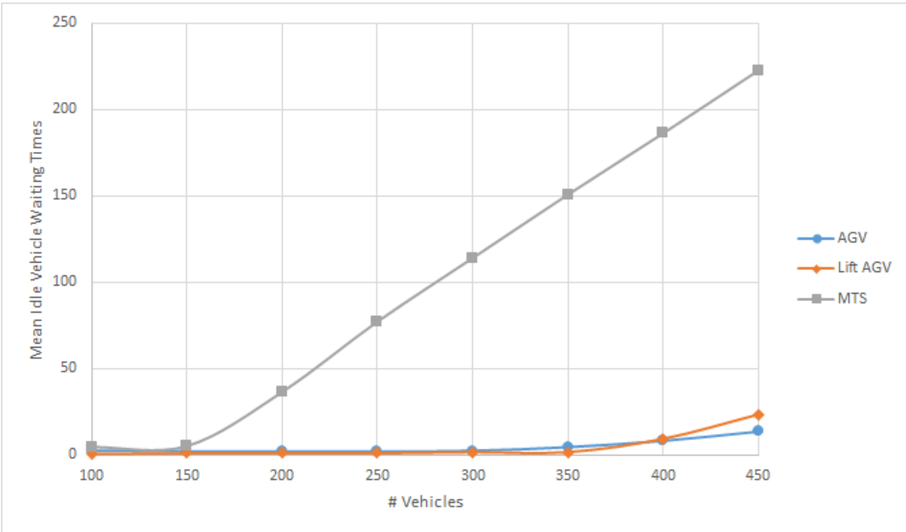


Figure 5.6: Mean idle Vehicle waiting times as a function of the number of Vehicles

5.3.2 Changing the transport demand

Simulations have been performed with a changing number of Containers to be transported per hour, for 0.5, 0.75, 1, 2, and 3 times the average amount of Containers. All simulations are done with 300 AGVs. The non-performance as a function of the number of Containers to be transported per hour is shown in Figure 5.8. As can be seen from the Figure, the non-performance drops drastically when having a smaller transport demand. When having a peak factor of 3 with these settings, the non-performance approaches 100%.

5.3.3 Changing the cap factor

A number of simulations have been performed with varying cap factors to show the impact of it on the non-performance. The results of these simulations, all with 300 AGVs, have been shown in the graph in Figure 5.9. As can be seen from the Figure, a higher cap factor means a lower non-performance. This is according to expectations, as a higher cap factor means the Containers are allowed to be delivered later before being accounted as non-performance.

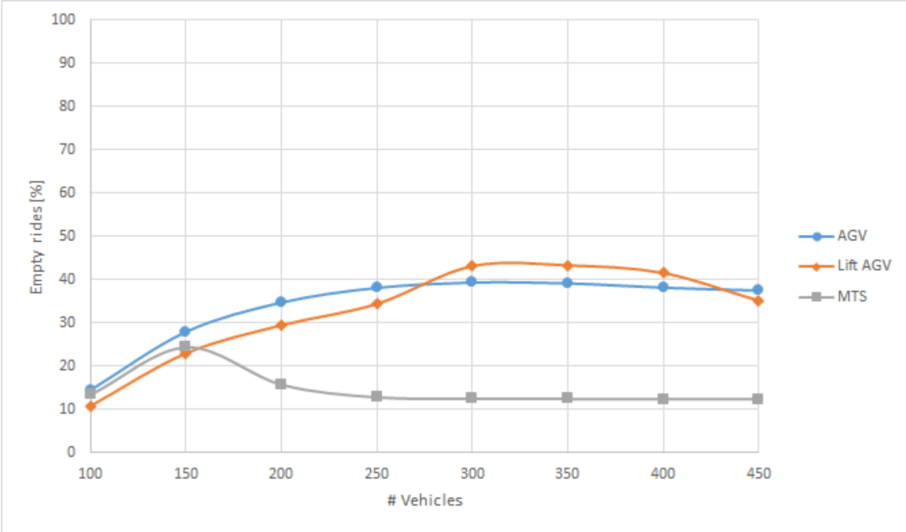


Figure 5.7: Percentage of empty rides as a function of the number of Vehicles

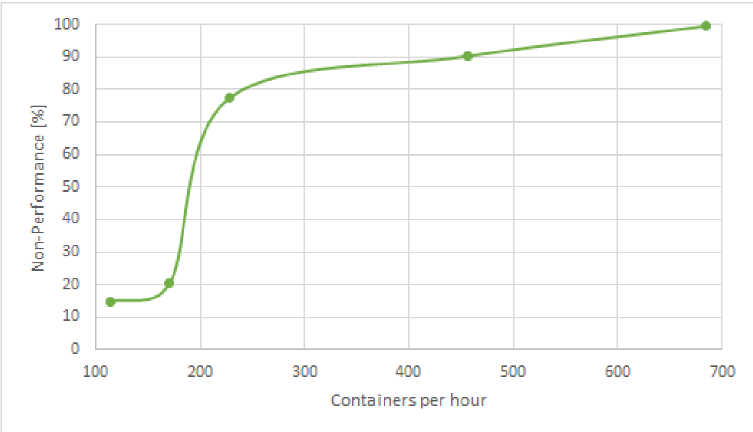


Figure 5.8: Non-Performance as a function of the amount of Containers to be transported per hour

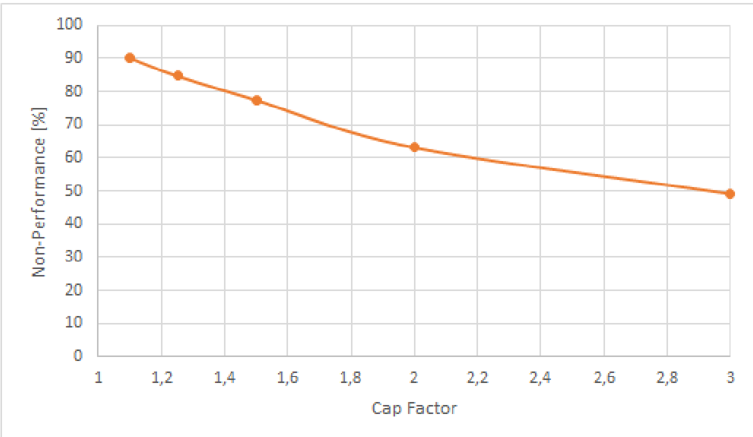


Figure 5.9: Non-Performance as a function of the cap factor

5.3.4 Results per Terminal

Simulations have been performed with 300 AGVs and a varying number of Terminal Equipment for RTW (Rail Terminal West). The results are shown in Table 5.2. As can be seen from the Table, the Terminal is underoccupied when there is just one piece of Terminal Equipment working there. This leads to high Vehicle waiting times, especially for Vehicles waiting to be unloaded. This also shows in the amount of Containers unloaded at this Terminal.

Adding more pieces of Equipment lowers the Equipment occupancy and the Vehicle waiting times. Even with 4 pieces of Equipment the idle Vehicle waiting times still stay quite high, which can be explained by the fact that this Terminal unloads a lot more Containers than it needs to load. This results in a significant abundance of Vehicles at this Terminal. Since there is no planning involved at this point, the Vehicles wait at the Terminal until they are loaded or requested by another Terminal.

# Eq.	Eq. Occupancy	Idle Vehicle WT	Loaded Vehicle WT	Cont. loaded	Cont. unloaded
1	1	11,89	324,65	7838	17232
2	0,51	6,53	1,67	7839	18135
3	0,34	5,9	0,25	7839	18138
4	0,26	5,63	0,03	7839	18118

Table 5.2: Results for RTW with a varying number of Terminal Equipment

5.3.5 Traffic flows

As mentioned in Section 4.3, the number of Vehicles per hour for every Road and Intersection is stored to a .csv file. Excel has been used to create histograms of these data for Intersection I04 (Figure 5.10) and Road R04a (5.11) for a simulation with 300 AGVs. Intersection I04 is the busiest Intersection of all, with an average of 244 and a maximum of 344 Vehicles per hour. Road R04a is one of the busiest Roads, with an average of 97 and a maximum of 151 Vehicles per hour. These data can be used to determine what kind of infrastructure is necessary throughout the transport route.

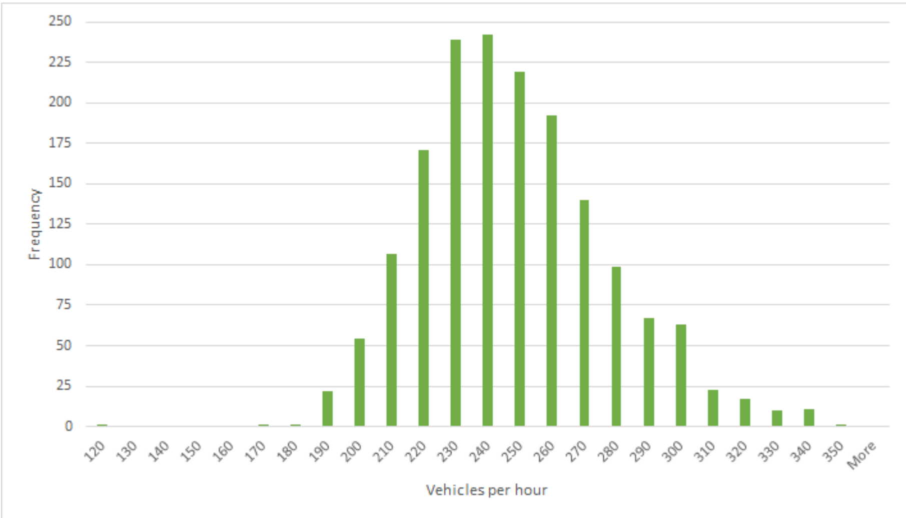


Figure 5.10: Histogram of the number of Vehicles per hour for Intersection I04

5.3.6 Environmental performance

The main output parameters the simulation model gives to determine the system's environmental performance are the total number of Vehicles used, the Vehicle type, and the total distance traveled by the Vehicles. If the emissions per kilometer, the total energy required to make a Vehicle, and the Vehicle's depreciation time are known, it is easy to calculate the environmental performance for each scenario.

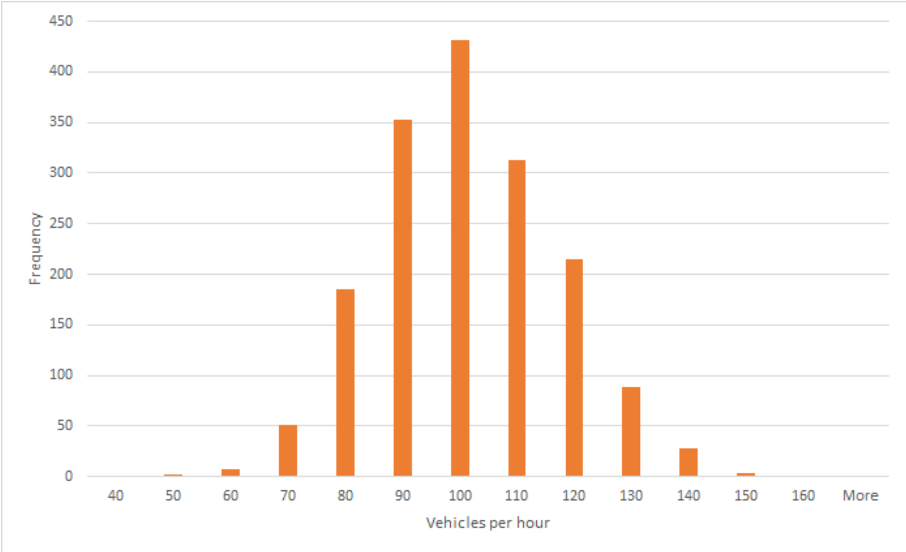


Figure 5.11: Histogram of the number of Vehicles per hour for Road R04a

5.4 Summary

This chapter has discussed the evaluation of the ITT simulation model. This has been done by the the hand of a number of simulation runs based on the transport demand scenario presented in Chapter 2. All obtained results comply with what was expected. The generated Container flows deviate very little from the intended flows. Non-performance is very high, which is logical since no planning is involved at this point.

The next chapter will conclude this report and will suggest possibilities for future research.

Chapter 6

Conclusions and future research

6.1 Conclusions

A discrete simulation model for an Inter Terminal Transport system at the Maasvlakte 1+2 has successfully been developed using Delphi and the object oriented simulation tools provided by TOMAS. The model provides output data that makes it possible to evaluate research questions such as the ones stated in Section 1.1.

The model's key performance indicator is non-performance. This value shows what percentage of the containers is delivered within the set time frame. Other important performance indicators are the occupation rates of the vehicles and the terminal equipment, vehicle waiting times at the terminals, the total distance driven by the vehicles, and the number of empty rides.

The model consists of 8 different objects: Containers, Generators, Roads, Intersections, Terminals, Nodes, Terminal Equipment and Vehicles. The Generators, Terminals, Terminal Equipment, and Vehicles are all active. The Containers, Roads, Intersection, and Nodes are passive. The processes for the AGV, Lift AGV, and MTS, are all built into the Vehicle object's process. The Nodes don't represent physical objects, but they are used by the Vehicles for path planning.

The model has been made easily adaptable for testing different scenarios. The vehicle type, the amount of vehicles, and the total amount of containers to be transported per hour, can simply be changed by altering their values in the configuration file. The transport demand can be altered by translating a new origin-destination matrix to the generators input file. It is also possible to use the model for road maps different than the currently used Maasvlakte scenario. In order to use a different road map, the properties of the roads, intersections, and terminals, can be redefined in their designated input files. The Vehicles use the Dijkstra algorithm for path planning, so no routes have to be defined. The Vehicles can find their own shortest route as long as the road map has been properly defined in the input files.

A large amount of output data is generated by the model. The output data includes: the system's non-performance, waiting times, vehicle and equipment occupancy, the number of empty rides, the total distance traveled by the vehicles, and traffic flows for each separate road and intersection. At the end of every simulation, data is written away to several output files. There is a general output file, an output file that shows the traffic flow per Road and Intersection, and an output file that show the results per Terminal. Data is being gathered during simulation that can be viewed live in various graphs on the TOMAS Collections form. At the end of the simulation the gathered data is written away to a number of .csv files. These files can be imported in spreadsheet software for further analysis.

A number of simulations have been performed in order to evaluate the model. For a simulation period of 10 weeks, the generated container flows deviate on average only 0,02% from the intended container flows. Unfortunately the model cannot be completely validated since no real life data is available to which the simulation results could be compared.

The results of the simulations show very high values of non-performance, which means that most containers are delivered late. This is easily explained by the absence of planning. The only advanced planning algorithm currently implemented in the model is for path planning. In order to make the model perform better, and to show more realistic results, it is essential that planning algorithms will be implemented in future expansion of the model.

6.2 Recommendations for future research

As stated in the conclusions, it is essential to expand the model with planning algorithms in order to make the model perform better and to show more realistic results. Smart planning solutions need to be found for several problems:

- Making sure there is a vehicle at a terminal when a container needs to be transported so terminal equipment doesn't have to wait for a vehicle to arrive.
- Dealing with transport demand peaks. In reality, containers have different levels of priority and urgency. Planning could be used to make sure that low level priority containers are not transported during peak hours, but at times when less capacity is required.
- Minimizing the number of empty rides and the total distance traveled in order to improve environmental performance and to reduce costs.
- Using advanced MTS planning in order to use MTS capacity as efficient as possible.

Besides planning there a several other things that could be implemented in the model:

- Make a distinction between 20 ft. and 40 ft. containers. AGVs will then be able to carry 2 20 ft. containers.
- Let containers have different levels of priority. This could also be done by giving containers a certain time frame within they need to be delivered to their destination terminal.
- Make transport demand more realistic by including train and barge schedules.
- Create different types of terminals.
- Model terminals more realistically by splitting them up in separate handling centers.
- Add barges to the system as an extra vehicle to perform the ITT. It is impossible for barges to perform all ITT at the Maasvlakte since not all Terminals are connected to the waterside, but they could potentially perform part of the ITT.
- Model MTSs in more detail, so consisting of a separate terminal tractor and chassis.
- For the Lift AGV scenario: make the size of the (un)loadplatform adjustable per terminal.
- Implement traffic models for the intersections to get a more realistic view of possible congestions. Different types of intersection could be created to assess their impact on the traffic flows.
- Make vehicle driving times more realistic by including factors such as accelerating, braking, and speed in corners.
- Let vehicles stop for gas or for a battery change when their energy runs low.
- Create animations that show the locations for all vehicles in the system.

Bibliography

[1] Proposal for the Port of Rotterdam. 2012.

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

[3] 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*, 4:469–493, 2006.

[4] 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.

[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*, 4:447–468, 2006.

[6] Port Technology. *http://www.porttechnology.org*.

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

[8] Hans Veeke and Jaap Ottjes. *http://www.tomasweb.com/*.

Appendix A

Maasvlakte map

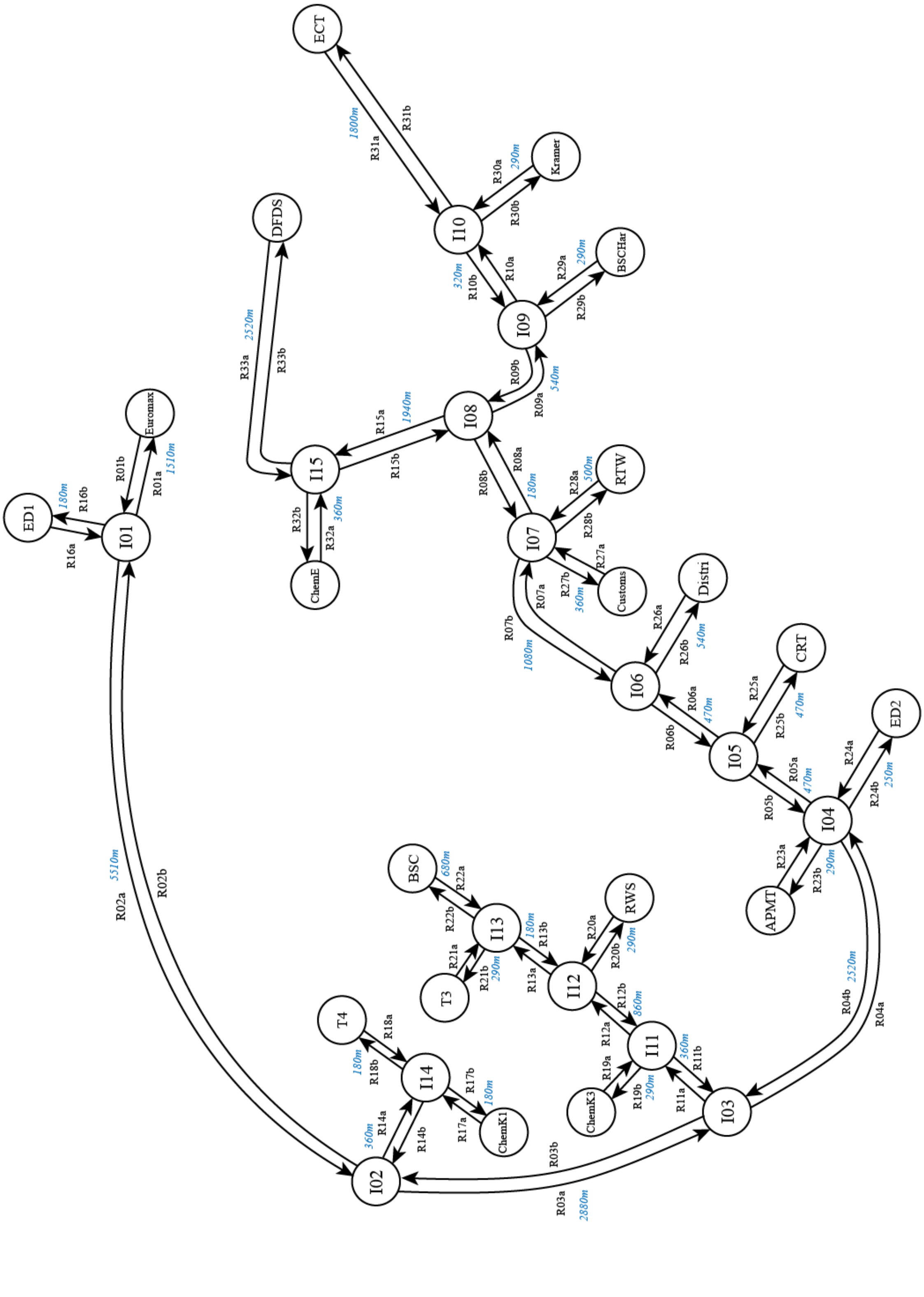


Figure A.1: Schematic map of the Maasvlakte showing all Terminals, Intersections, and Roads + Road Lenghts