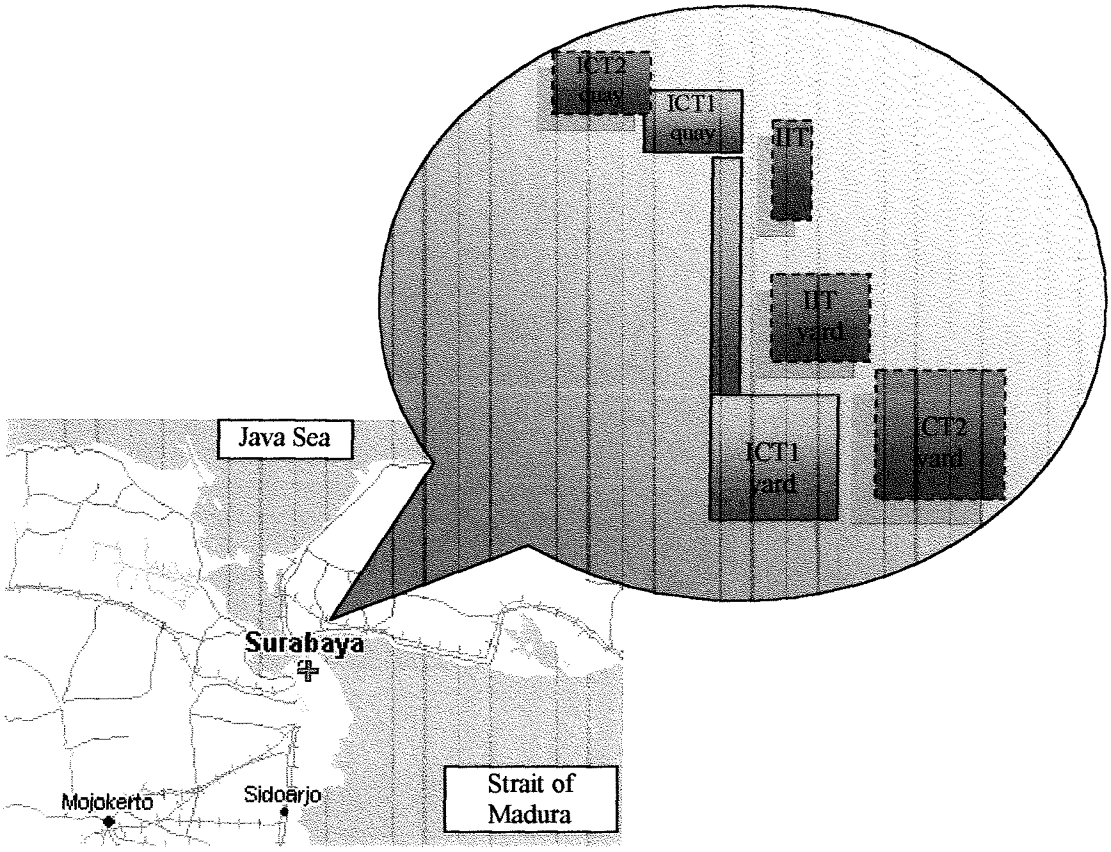
Computer simulation model

International Container Terminal

“Tanjung Perak”

Surabaya, Indonesia



Final graduation report

S. Wanders
September 1998

Computer simulation model

International Container Terminal

"Tanjung Perak"

Surabaya, Indonesia

Final graduation report
S. Wanders
September, 1998

Under supervision of:

Prof. ir. H. Ligteringen
Professor of Ports and Inland Waterways (part of the Hydraulic and Offshore Engineering section of the Division Hydraulic and Geotechnical Engineering, Faculty of Civil Enigineering, Technical University of Delft)

ir. R. Groenveld
Staff scientist of Ports and Inland Waterways

ir. J. C. Rijsenbrij
Interim manager Department Transport Technology, Faculty Mechanical Engineering and Marine Technology

Foreword

The Division of Hydraulic and Geotechnical Engineering of the Faculty Civil Engineering, Delft University of Technology, is conducting research in corporation with the Technical University of Surabaya (ITS) and the Port Corporation III in Indonesia. In the framework of this research corporation various port efficiency studies have been carried out.

The present graduation report concerns the expansion of the International Container Terminal of the Surabaya Tanjung Perak port. For this study computer simulation models of the terminal have been developed and tested using terminal records and other information on the terminal operations made available by the terminal management in Indonesia.

Special thanks are in place for ir. J. Mutsaers who, based on his on-site experience, was able to provide valuable background information and advise on the terminal operations. Finally, I thank ir. R. Groenveld of the Division Hydraulic and Geotechnical Engineering for his advise and guidance during this research.

Summary

The Tanjung Perak harbour of the city of Surabaya on the island Java, Indonesia has experienced a considerable growth of container traffic. In order to adequately deal with the expected continuing increase of container traffic in the future, the International container terminal is presently being expanded.

In future there will be three independently operating terminals: two terminals for international container traffic (ICT) and one terminal for inter island traffic (IIT). The Inter Island Terminal will handle coasters operating between the larger and smaller (more inland located) ports. The quays of these three terminals will be connected to the container yards by one bridge.
Evidently the expansion of the container terminal will create a new situation on the whole terminal, in particular on the bridge connecting the quays to the container yards. The objective of this thesis research is to investigate the capacity of the future container terminal and to analyse possible future bottlenecks (with emphasis on the bridge situation). For this purpose computer simulation models have been created of the three terminals and the bridge.

The first phase of the research concerned the creation of a simulation model of the terminal as it is operating at present. This model was verified and validated using the 1997 data records of the terminal. Since the operations of this terminal will not change drastically once the expansion of the terminal is completed, this simulation model of the present terminal has also been used as the basis for the simulation of the future terminals.
In order to be able to cope with the anticipated increase of container traffic, upgrading of various terminal operations will have to be considered. This can be achieved by improving efficiency of various terminal operations and/or by upgrading equipment. In this phase a computer model was created to generate vessels with a certain inter-arrival time, load and length to call at the terminal. The data lists generated by this model have been used as input by the future terminal models.

In the second phase three simulation models have been developed: one for the new international container terminal, one for the inter island terminal and a model of the bridge. The second international container terminal has been considered a replica of the present terminal since both terminals are scheduled to use the same equipment and to have the same capacity. The main difference will be the location of the quay and the container yard. As the new ICT quay will be located west of the present quay and the new ICTcontainer yard next to the present container yard (on the eastern side), the distance between the new ICT yard and the quay will be larger compared to the present operating terminal. The inter island terminal quay is presently being built along the bridge that is also connecting the ICT quays to the container yards.

Using above simulation models the future terminal operations have been analysed and in the process also an indication of the capacity of the terminal is obtained. The models enabled a better insight of the possible container throughput of the terminal in relation to the efficiency level of terminal operations and upgrading of equipment. The traffic load on the bridge has been investigated by analysing the varying traffic flows generated by the terminal models.

Concluding it appears that there are two factors that may limit the anticipated container throughput increases per terminal: the capacity of the portainers and the stacking area of the container yards. In case the container throughput per terminal increases as expected and the efficiency levels can not be improved (thus limiting the production of quay cranes), the possibility of a serious congestion of the terminal may be the result. This would result in increased anchorage waiting times for the vessels and in a further increase of the quay occupancy (more than the present 70%). Also the capacity of the stacking areas may prove to be insufficient to cope with this increased container throughput. These effects also imply that the terminal will not have an adequate safety margin to handle unforeseen delays, for instance equipment breakdown.

If on the other hand the efficiency of terminal operations can be improved (thus enabling a higher portainer production) and if the dwell time of the containers in the container yard can be reduced, then the terminal will be able to handle a maximum of about 1,400,000 boxes (which is about 2 million TEU) per year, assuming that the present quay occupancy of approximately 70% is maintained.

It is noted that in case the efficiency level of terminal operations can not be increased beyond the present efficiency level (and maintaining the maximum quay occupancy of 70%), a box throughput of about 1,000,000 boxes (about 1,450,000 TEU) is expected to be possible on a year basis by the terminal. This capacity corresponds with more than twice the number of containers that are handled by the present operative terminal.

It appears that the bridge will not constitute a limiting factor for terminal operations once all three terminals are operational, even if these terminals are operating at maximum capacity.

Contents

1. Preface

1.1 General picture of the port

The city Surabaya lies in the north-eastern part of Java (see figure 1.1), which is the most populated island of Indonesia. Surabaya is the second major city in Indonesia, after the capital Jakarta, and is the provincial capital of East Java.

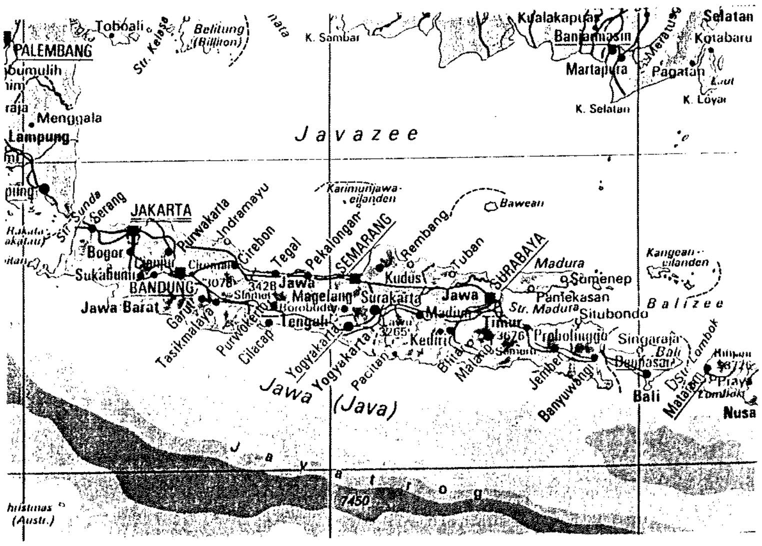


figure 1.1: The island Java

The Tanjung Perak harbour, the port of Surabaya, is the second major port of Indonesia and the main port of East Java. The harbour is located in the Northern part of the city, on the south bank of the straits of Madura (see figure 1.2).

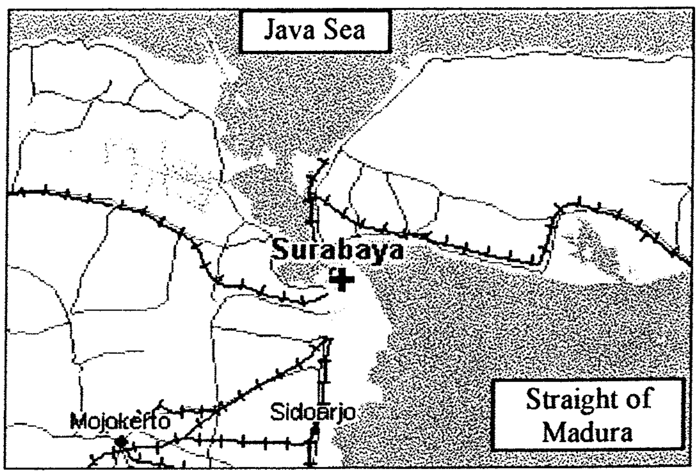


figure 1.2: The location of the City of Surabaya

The main entrance of the harbour is from the North by the Western Fairway: a partly dredged channel with a length of about 45 kilometres. Approach of the harbour from the East by the Eastern Fairway is only possible for small vessels due to the fact that the Gulf of Madura is very shallow.

In 1982 the government of Indonesia introduced the "Four Gateway Ports System", an investment programme to stimulate the development of four ports among which the Tanjung Perak harbour. Most of East Java and a part of Central Java constitute the hinterland of the T. P. harbour. The harbour acts as an interface and link in the transportation system for a large variety of goods. Besides containers, general cargo, bulk goods and liquid bulk goods are handled by the harbour. In 1986 about 19% of the international trading goods in Indonesia were shipped by container. By 1993 this percentage increased up to 50%, corresponding with about 30% of the total volume of goods handled by the harbour.

This tendency of increased shipping of goods by means of containers has since 1993 been maintained: between 1993 and 1997 the container throughput increased from 410,000 TEU (about 280,000 box moves) to about 560,000 TEU (380,000 box moves), which is an increase of about 36% (see figure 1.3). Also the traffic between the islands ("inter island traffic") follows this tendency towards increased traffic of goods by means of containers.

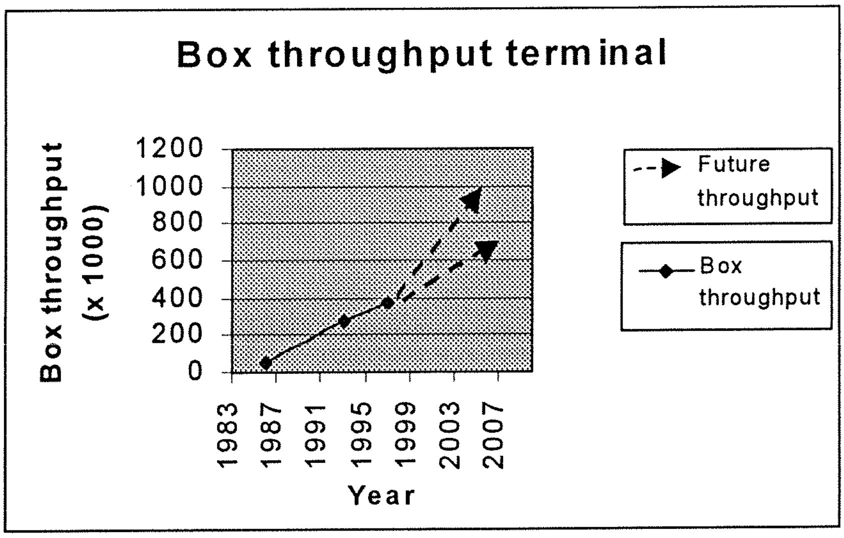


figure 1.3: Box throughput terminal

Before the economic crisis hit Asia, it was expected that the container flow would continue to increase to such an extent that the present container terminal facilities might not be able to handle the anticipated container flow in the near future.
Following this growth scenario (see figure 1.3), congestion of different sections of the terminal has been foreseen, resulting in longer waiting times before the vessels are serviced as well as in longer turn-around times of vessels (possibly increasing up to 24 hours). The anticipated increased turn-around times would constitute an unfavourable development as certain shipping companies might start looking for alternative container terminals for their vessels.

In order to deal with the anticipated increase of container throughput, the container terminal is currently being expanded with (see also figure 1.4, the not coloured sections):

- another 500 meters quay for international container traffic located west of the present quay;
- 450 meters quay for inter-island traffic along the bridge;
- expansion of the container yard;
- corresponding upgrading of the terminal equipment.

In future (the year 2005) there will thus be three independently operating terminals: two for international container traffic (ICT1 and ICT2) and one terminal for inter-island container traffic (IIT). The inter-island terminal will handle coasters operating between the larger and smaller harbours.

1.2 Present International Container Terminal

The present International Container Terminal consists of an offshore berth island connected to the main land by means of a 1800 meters long bridge (see figure 1.4, coloured sections, see also appendix 1). The berth island has a quay of 500 meters length, consisting of three berthing places for servicing international container vessels. The vessels are serviced by four rail-mounted quay cranes (portainers). The transport of containers to and from the berth island is done by means of tractor-trailers.

The present container yard consists of ten stacks in which rubber-tyred gantry cranes are used to stack the containers. The stacks in the container yard are each 66 TEU 20 ft long and 7 TEU (7x8 ft) wide. The containers can be stacked up to four containers high. The off size and left over containers are stacked at the southern end of the stacks and for refrigerated containers (reefers) there are slots reserved in two stacks. The offices and Container Freight Station (CFS) are located at the western and southern side of the terminal grounds (appendix 1).

Trucks are used for the delivery and picking up of containers. These trucks, arriving by road, have to wait in the parking lots outside the terminal grounds till granted permission to enter the yard for the delivery or collection of their load. These trucks are not allowed to drive onto the bridge and deliver or pick-up their load directly at the vessels berthed.

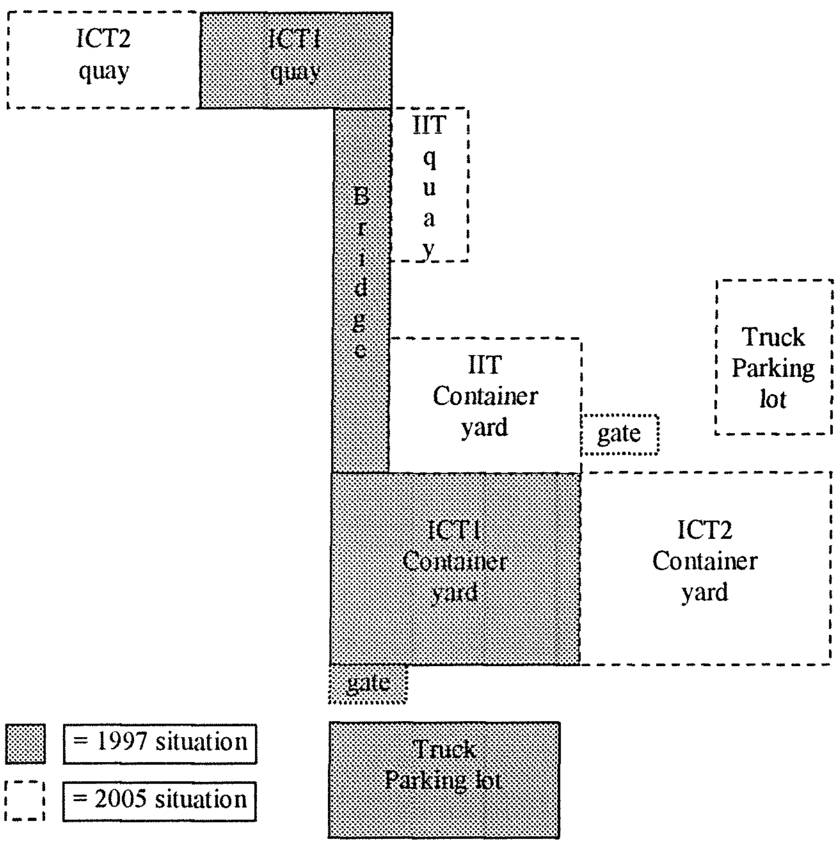


figure 1.4: Tanjung Perak International Container Terminal

2. Problem description

2.1 Problem analysis

Taking the present ICT operations as a starting point (about 380,000 box moves per year), the operation of a similarly equipped second International Container Terminal would in principle enable the terminal to handle roughly twice the number of containers by 2005 (in case the present capacity and efficiency of the various terminal operations are maintained).

When taking into account the future operations of also the Inter Island Terminal (using the same bridge connecting the quays to the container yard), it is evident that such increased traffic will result in a new situation on the terminal grounds and on the bridge.

It is not known whether and for which terminal operations problems of congestion may arise as result of such increased container throughput in the future. Particularly the bridge is mentioned as a potential bottleneck, with possible negative effects on other terminal operations.

It is however not clear to which extent the capacity of the terminal can be increased and which alterations of terminal facilities and /or improvements of terminal operations will be required to achieve this. Neither is it known what the impact of such measures (e.g. a higher portainer production rate) will be on other terminal operations, both in the yard and on the bridge.

2.2 Research objective

The main objective of this thesis research is to investigate the capacity of the future container terminal after completion of the scheduled expansion and to

- assess the impact on various terminal operations (with particular attention for the increased traffic load on the bridge);
- identify possible constraints and analyse the measures that can be taken in order to further increase the terminal capacity in future .

This complex container terminal system will be investigated using the computer simulation language PROSIM. With his tool, first a computer simulation model of the present (1997) operating International Container Terminal will be created. After verification and validation of this model a computer simulation model of the future terminal will be developed.
For the investigations of the future terminal, in total four computer simulation models will be developed: two International Container Terminal models, one Inter Island Terminal model and one model of the bridge. Using these models an adequate insight in the utilisation of the equipment, the service times of the ships and the capacity of the whole terminal will be obtained. Based on these simulation models possible bottlenecks can be identified and possible solutions analysed.

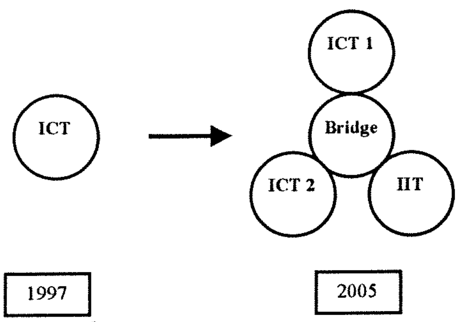


figure 2.1: Outline models to be created

3. Problem approach

As described in the research objective, this research will focus on the future situation of the terminal after its ongoing expansion: what effects will this expansion (and the anticipated increased container flow) have on the operations and the efficiency of the terminal? Four computer simulation models have been developed to provide more insight in the future terminal operations.

It is planned that eventually there will be three independently operating terminals at this location. Although the operations within the terminal will not be changed drastically, an upgrading of the operations is called for in order to cope with the anticipated increased container traffic. This can be achieved by increasing the efficiency of operations and/or by upgrading the equipment.

The first phase of research concerned the development of a simulation model for the terminal as it is presently operating. This model representing the International Container Terminal at present (and also in the future) was verified and validated using parameters obtained from the 1997 data records of the terminal. As such data are not available for the future situation, the parameters for the future situation have been generated in an extra model. This model, generating the time of ship arrival, its load and length, will in fact act as the planning maker for the future terminal.

In the second phase, the two new terminals have been created. The second International Container Terminal will be the same as the present terminal: both will use similar equipment and have the same capacity. The main difference will be the location of the berths and the container yard. Since the Inter Island Terminal, scheduled to handle inter-island container traffic, will have its berths located along the bridge, the quays of all three terminals will be connected to one and the same bridge. This bridge component has therefore been looked upon as a potential bottleneck for the future terminal (in any case, the traffic on the bridge will increase considerably).

Using above models, it will be possible to investigate the efficiency of the new terminal operations and thus the capacity of the terminal. By increasing the efficiency and upgrading the equipment, the service times of the vessels can be reduced, thus increasing the throughput of the terminal.
The questions to be answered are, whether the anticipated increase in activity of the whole terminal will have a negative effect on the operations of the terminals and whether specific bottlenecks are to be expected?

By running the models of the separate terminals at maximum capacity (while maintaining a maximum 70% quay occupancy), a maximum traffic load on the terminal grounds is created: in this situation it can be estimated which increase in efficiency of various processes and which upgrading of equipment will be required in order to assure optimal terminal operations. If problems occur, solutions may then be examined.

With above maximum throughput of the terminal, also a maximum traffic load on the bridge will be simulated. At this maximum traffic load, the possible bottlenecks of the bridge will be examined. As this bridge could - in case of traffic jams - effect the operations of all the terminals, the situation at the bridge will be analysed separately.
To conclude predictions are made regarding some of the improvements that may be made to keep the terminal operating as efficiently as possible and the corresponding capacities that can be reached.

4. Problem analysis tools

4.1 The computer simulation language PROSIM

PROSIM is a computer simulation language suitable for discrete, continuous and combined simulations. This language has been developed at the Technical University of Delft.

PROSIM is designed to simulate real life systems. A model is used because experimenting in the real system is either not possible or too complex and expensive. To create a model of the real life system, this system must be analysed, schematised and broken down into components, which all have their own specific task. The function and the activities of each component have to be described in its own module. A computer simulation model will thus consist of different modules, which are connected by the fact that the components interact with each other. These interactions are described in the modules.

4.2 General description of the language

The real life system is considered to be modelled as a set of interrelated components. The components constitute the basic elements of the PROSIM language.

A component can be either permanent or temporary: a permanent component will be present in the model permanently whereas a temporary component will only be present for a certain period of time. Every component has certain characteristics that are of importance for the proper functioning of this component and for the interacting with other components. These characteristics are defined in the model and are called "*attributes*".
When components have the same type of attributes, they form a "*class of components*".

Note: The attributes that are of the same type don't necessarily have the same value. For instance in this model, a container terminal will be simulated with vessels arriving; these vessels form the class of components called SHIP and one of the attributes will be the load.

A PROSIM model consists of two sections:
- a *definition* section in which all the components and their attributes, queues and other variables are defined. This definition section is called the DEFINE module;
- a *dynamic* section in which the dynamic behaviour of the components is described. This section consists of "modules" and "macros".

A PROSIM model always starts with a module called MAINMOD. This module takes care of the run control and the creation and activation of the first components. The other modules are activated either by the MAINMOD module or by another module.

A macro is activated by a module. Such a macro contains a description of certain activities to be executed. These activities are repeated regularly by the module and are therefore put in a macro in order to keep the module as simple as possible.

5. Present time International Container Terminal (1997)

5.1 Introduction

Before a simulation model of the terminal, as it was in 1997, can be built, it is necessary to perform an extensive data research. This can be achieved in various ways. By interviewing people who have been on site at the container terminal, by collecting records of past terminal operations and by examining survey results.
In this chapter first a brief view of the main terminal activities is given. Then the data records of the terminal during 1997 and the results of a survey are discussed. Subsequently, the operations occurring on the terminal are reviewed. Finally the workings of the computer simulation model of the container terminal are explained.

5.2 Overview ICT

In this section a brief overview of the terminal is given. Figure 5.1 below gives a simplified view of the terminal layout.
The quay has a length of 500 meters and consists of maximal three berth places. The quay is connected with the container yard by means of an 1800 meters long bridge. The container yard, as can be seen in figure 5.1, contains 10 container stacks: 5 stacks for export containers, 4 stacks for import containers and one stack for empty containers. North of the stacks there is a parking lot for trucks waiting in the yard to enter one of the stacks. The RTGC parking lot is also located north of the stacks. The tractor-trailer parking lot is located in the north-eastern corner of the container yard. The entrance gate, through which the trucks enter and leave the yard, is located in the south-western corner of the terminal grounds.

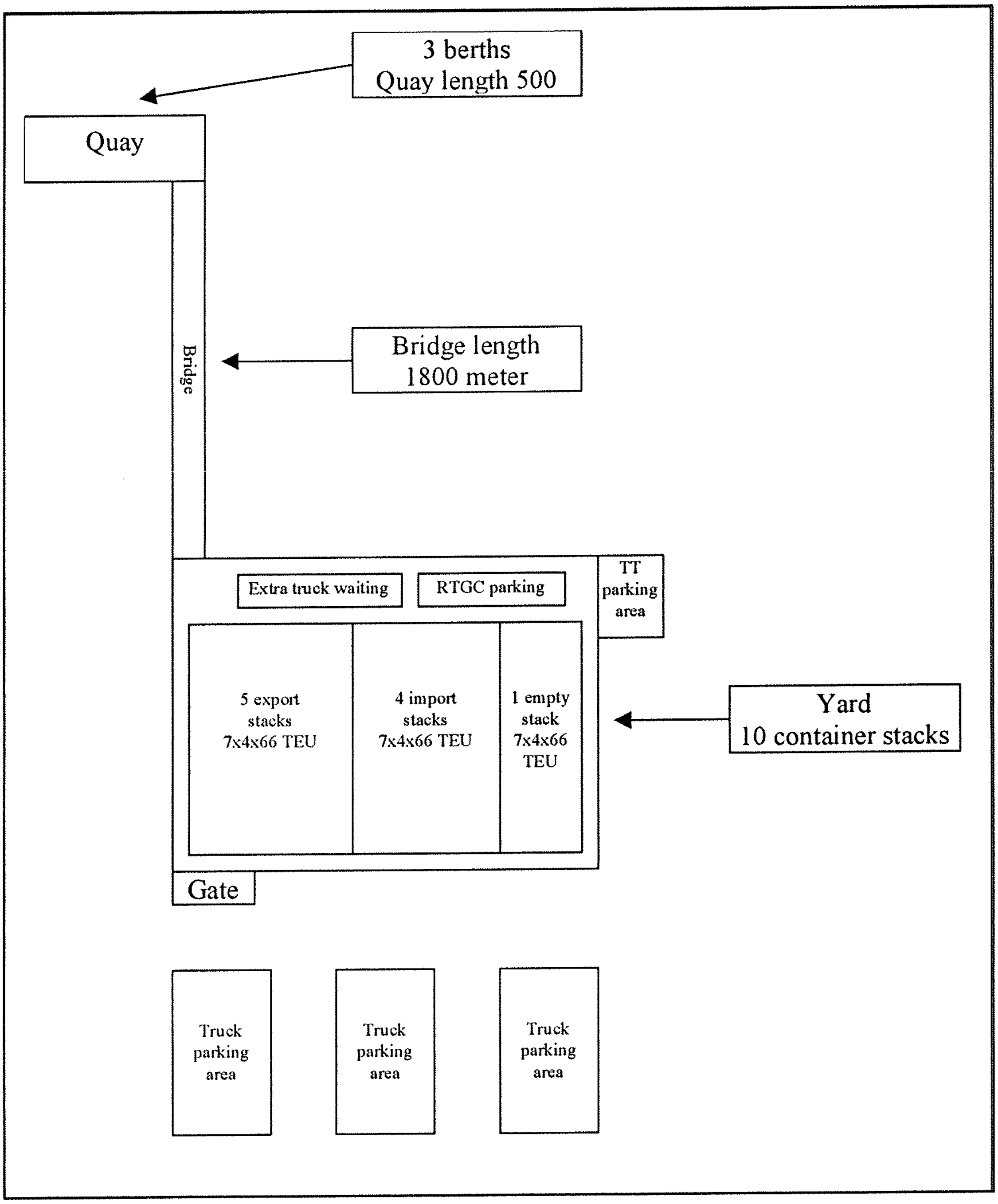


figure 5.1: Overview terminal layout 1997

5.3 Data statistics

In order to get a proper insight of the operations and performance of the terminal, it is important to collect and analyse as much information as possible concerning this terminal. The terminal records of 1997 provide information regarding the actual arrival and departure times of the vessels, vessel length, number of box moves per vessel and portainer production. From these records information is obtained about the service and inter-arrival times of the ships, providing a good insight in the terminal performance. These terminal records are analysed first.

Subsequently, the results of a survey carried out on the RTGC-operations in the yard, are discussed.

5.3.1 Vessels

5.3.1.1 Arrival times

The arrival times of the ships at the terminal (in 1997, 992 ships called at the terminal) provide an insight in the "*inter-arrival times*" of the ships. The inter-arrival time is the time between the arrival of two successive ships at the terminal.

Analysis of the data has been done with the program BestFit. With BestFit a distribution function is fitted to the data and simultaneously analysed according to two goodness of fit tests: the Chi-square Test and the Kolmogorov-Smirnov Test. BestFit also gives a ranking of which distribution fits the data best. The analysis results obtained with BestFit can be reviewed in table 5.1 and figures 5.3 and 5.4. The inter-arrival times are assumed exponentially distributed. The average inter-arrival time in 1997 was 8 hours and 52 minutes.
The statistical data and the exponential shaped distribution are illustrated in figure 5.2 and in the BestFit results section below.

Excel data statistics:

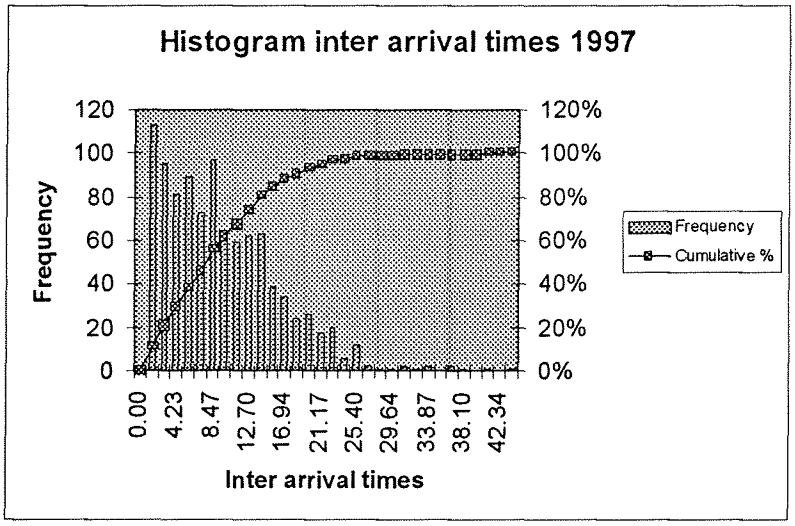


figure 5.2: Histogram & cumulative distribution vessel inter-arrival time

BestFit results:

Minimum=	0.00
Maximum=	43.75
Mode=	0.33
Mean=	8.83
Std Deviation=	6.74
Variance=	45.46

Best Fit Results				
Function	Chi-Square	Rank	K-S Test	Rank
Gamma(1.30,6.82)	0.11	1	0.07	1
Expon(8.83)	0.15	2	0.10	2
Normal(5.67,7.77)	5.15	3	0.25	3

table 5.1: BestFit analysis and ranking results

BestFit: Exponential distributions

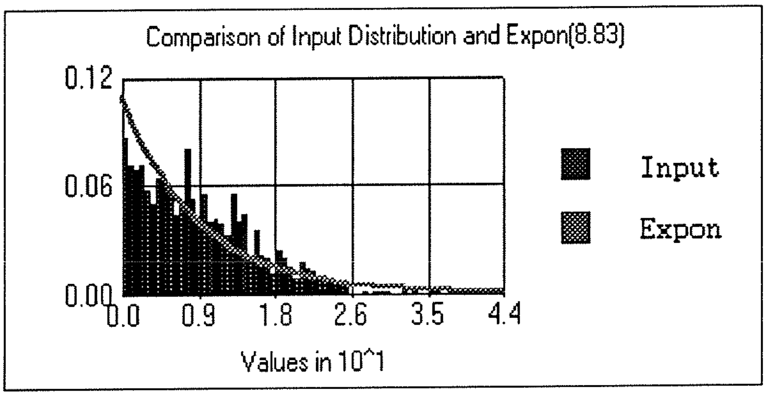


figure 5.3: P.d.f. exponential distribution

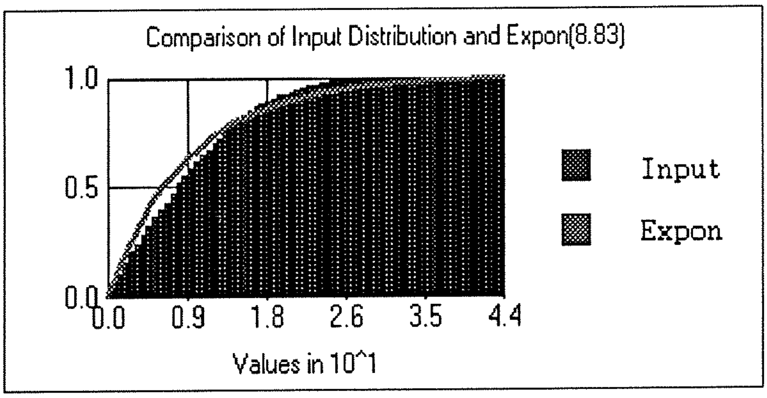


figure 5.4: C.d.f. exponential distribution

5.3.1.2 Service times

The "*service time*" of a ship is defined, as the time needed to completely service a vessel by the terminal. As the definition of the actual arrival and departure times of the ships in the terminal records is not exactly clear (see appendix 5.1), the time the ship spends at the quay has been defined as the "*turn-around time*" of the ship.
According to the 1997 records, the average turn-around time of a vessel at the terminal would be 19 hours and 10 minutes. Excluding some extremely high turn-around times (exceeding 40 hours), an average turn-around time of 18 hours and 53 minutes appears to be realistic.

Note: For unknown reasons, 1 % of all the vessels serviced at the terminal in 1997 had such an extreme berth time: possibly this was caused by some sort of mechanical failure of the vessels.

The Excel data statistics section (see figure 5.5 below) shows a cumulative distribution of the data. The irregular shape of the histogram may be due to the fact that the majority of ships arrive and depart just before or after the changing of the day shifts (occurring at 0:00, 8:00 and 16:00).

BestFit results:

Minimum=	2.75
Maximum=	68.00
Mode=	14.92
Mean=	19.16
Std Deviation=	7.35
Variance=	54.05

table 5.2: BestFit analysis

Excel data statistics:

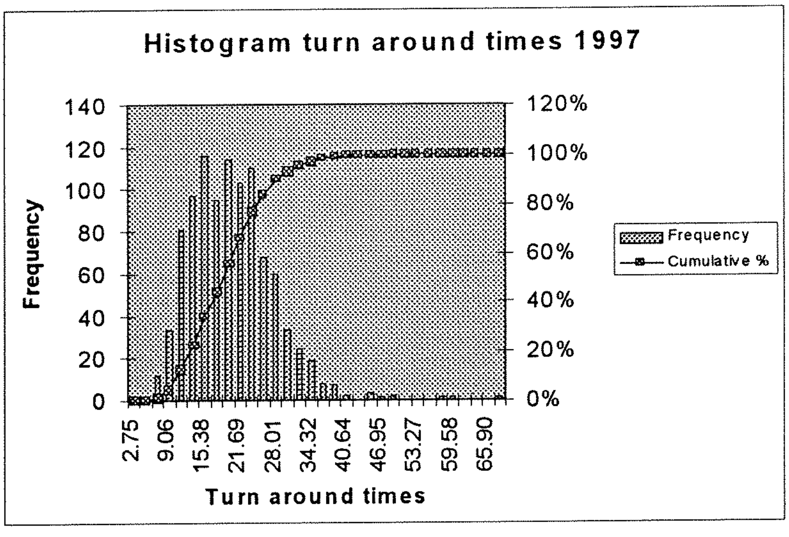


figure 5.5: Histogram & cumulative distribution vessel turn-around time

5.3.1.3 Load distribution

The 1997 terminal records indicate that there is a difference between the expected number of box moves to be made and the total number of actually realised box moves: more box moves have been made than in first instance expected.

The average number of boxes unloaded and loaded per ship was 393 in 1997. The distribution of the number of moves made during 1997, is shown in figure 5.6.

The distribution of (full or empty) containers imported and exported in 1997, is summarised in table 5.3 below and illustrated in the pie charts (figures 5.7 & 5.8 next page): obviously there is more export than import traffic. Consequently, the average number of containers moved per ship is higher when loading than during discharging, whereas less empty containers are loaded than discharged. The much higher percentage of full containers justifies the fact that there is only one empty container stack against five export and four import stacks.

Note: The "less than full" containers are included in the "full" container percentage. There is hardly any traffic in container sizes other than the 20 and 40 feet long containers.

Boxes	Import Discharged	Export Loaded
total boxes	184,059	205,651
% average	47	53
av. per ship	186	207
% 20' full	43	54
% 40' full	28	41
% 45' full	0	0
% 20' empty	13	2
% 40' empty	16	3
% 45' empty	0	0

TEU's	Import Discharged	Export Loaded
total TEU's	265,846	295,740
% average	47	53
av. per ship	268	298
% 20' full	30	37
% 40' full	38	56
% 45' full	0	1
% 20' empty	9	2
% 40' empty	22	4
% 45' empty	1	0

table 5.3: % discharged and loaded distribution boxes and TEU's 1997

Excel data statistics:

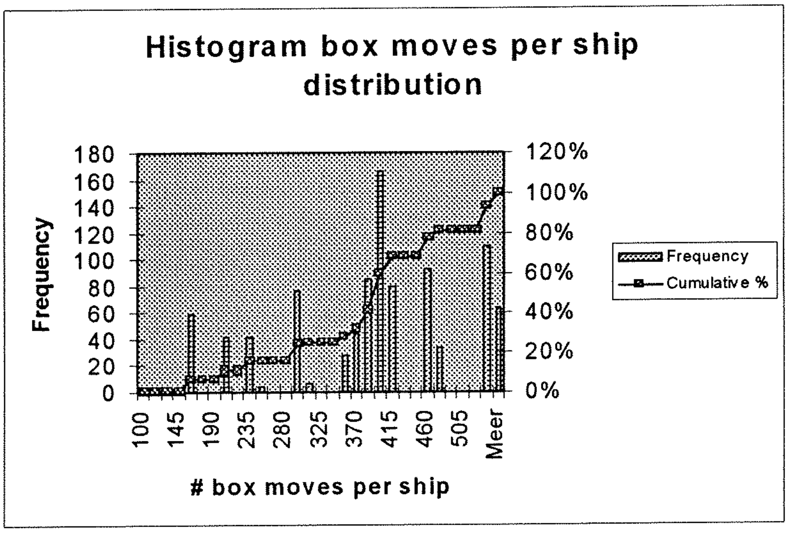


figure 5.6: Histogram & cumulative distribution box moves per vessel

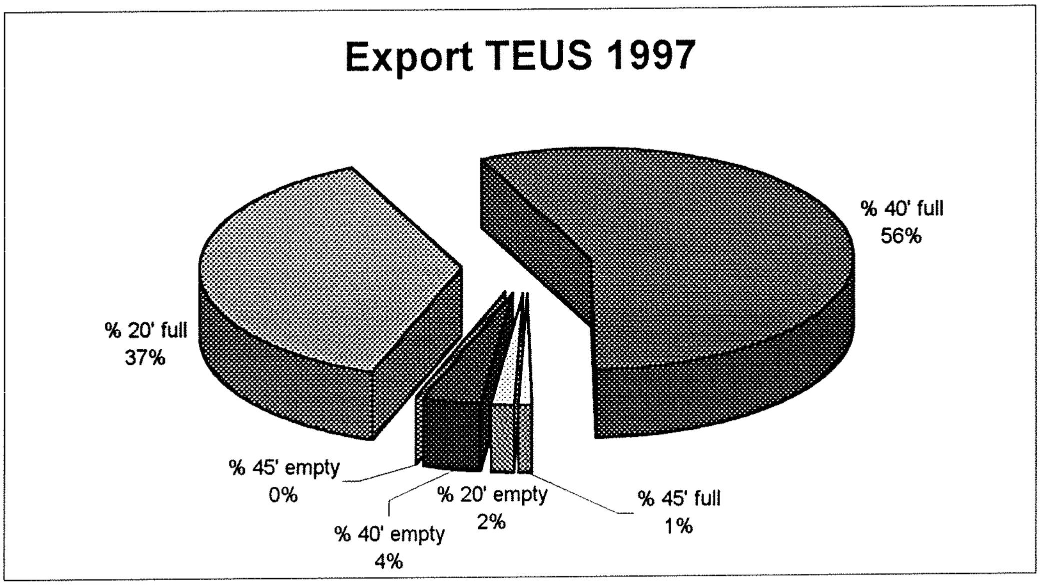


figure 5.7: Export TEU's 1997

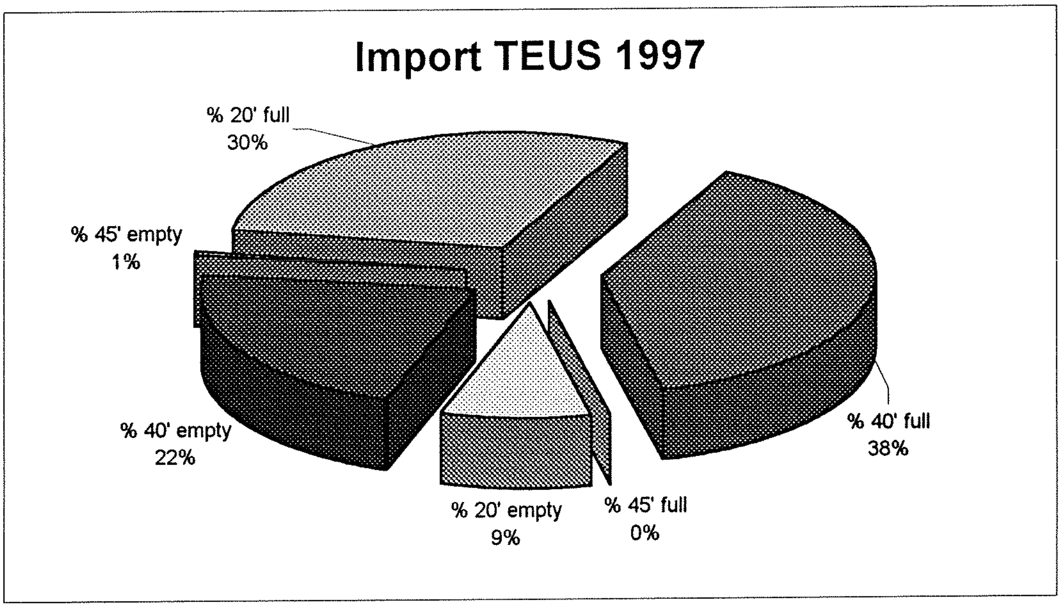


figure 5.8: Import TEU's 1997

5.3.1.4 Length distribution

The container vessels arriving at the Tanjung Perak terminal are first and second generation vessels. The average vessel length in 1997 was 153 meter (see table 5.4). The ship length distribution in 1997 is shown below in figure 5.9. There is no specific distribution that fits the ship length data.

BestFit results:

Minimum=	114.00
Maximum=	208.00
Mode=	121.00
Mean=	152.58
Std Deviation=	24.50
Variance=	600.17

table 5.4: BestFit analysis

Excel data statistics:

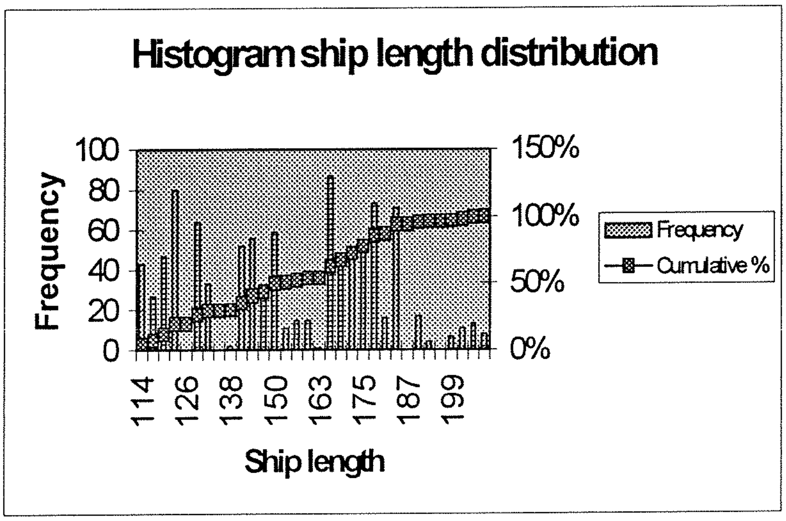


figure 5.9: Histogram & cumulative distribution ship length

5.3.2 Portainer production

According to the 1997 terminal records, the portainer production for unloading and loading containers was approximately 21 box moves per crane per hour.
Based on this information, the working speed of the portainers has been predicted.

For reasons of simplicity, the operations of a portainer have been simplified into three possible actions in the model, as follows:

- *"lift-on"*: action of the portainer spreader (holding a container), lifting a container onto the tractor-trailer and returning to its former position;
- *"lift-off"*: action of the spreader lifting a container off the tt and returning to its upper most position;

- *"lift-cycle"*: action of the portainer following the lift-on or lift-off actions. This lift-cycle action comprises the movement of the spreader towards a position above the vessel (to either deliver or pick-up a container) and the movement back to its former position. It is assumed that the lift-cycle action takes twice as long as either one of the other two actions.

The lift-on and lift-off actions are characterised by a uniform distribution with a mean of 55 seconds. The lift-cycle action has a gamma shaped distribution with a mean of 109 seconds.

Note: By characterising the portainer actions with a uniform distribution it is assumed that the non effective portainer moves (e.g. shifting of the portainer once in a while) are also included.

5.3.3 Yard survey results

A survey was carried out to determine the frequency and duration of the different actions of the "rubber-tired gantry crane" (RTGC). From the survey it appears that there may be either one, two or three RTGC's at work simultaneously in the same stack (mostly however only one or two RTGC's are working in the same stack). The number of RTGC's at work in a stack, depends on the number of free RTGC's, the number of ships at the quay and the number of carriers waiting to be serviced.

The RTGC has as main function to lift-off containers from a transporter and to place these into the stack (or vice-versa). A RTGC also has to drive to other locations in the stack once in a while (approximately once every five container lift actions). Occasionally the stack the stack also will have to be re-shuffled by a RTGC, this occurs about once after four container lift actions.

Note: The time needed for a RTGC to re-position itself after lifting of a container, has not been recorded during the survey. This action has been assumed identical to the lift-on and lift-off actions.

From the analysis of the data using the BestFit program, the following can be concluded (see table 5.5):

- mean lift-on, lift-off and repositioning actions of 51 seconds;
- mean re-shuffle action of 39 seconds;
- mean drive time of 45.5 seconds.

From above it is estimated that a RTGC can make approximately 30 box moves per hour. These RTGC actions are best characterised by a gamma shaped distribution.

BestFit results:

Lifton-off times:

Minimum=	17.30
Maximum=	107.00
Mode=	39.73
Mean=	51.27
Std Deviation=	16.82
Variance=	282.84
Skewness=	0.41

Reshuffle times:

Minimum=	19.60
Maximum=	73.00
Mode=	27.61
Mean=	38.90
Std Deviation=	13.31
Variance=	177.29
Skewness=	0.77

Drive times:

Minimum=	5.55
Maximum=	235.10
Mode=	39.98
Mean=	45.52
Std Deviation=	42.58
Variance=	1812.66
Skewness=	2.62

table 5.5: BestFit analysis RTGC operations times

5.4 Terminal operations

5.4.1 Work periods

The terminal is operational 360 days per year, 24 hours per day. A day is divided into three 8-hour shifts with a 1-hour break per shift for eating, drinking and resting.

Note: The survey does not specify whether this 1-hour break includes the time required for shutting down and starting up of the equipment, etc. Nor is it clear how much time it takes to change shifts and whether in practice this results in time losses.

The regular shift times are as follows:

- from 08:00 to 16:00 (break from 12:00 to 13:00),
- from 16:00 to 00:00 (break from 20:00 to 21:00),
- from 00:00 to 08:00 (break from 04:00 to 05:00).

During the Ramadan (holy month, when Muslims do not take food nor drinks during daytime) the breaks are extended with half an hour. During this period, the 04:00 o'clock break begins half an hour earlier instead of lasting half an hour longer (sun set in Eastern Java between 04:00 and 05:00). The break times during the Ramadan period are as follows:

- from 12:00 to 13:30,
- from 20:00 to 21:30,
- from 03:30 to 05:00.

5.4.2 Vessels

The vessels calling at the terminal are mostly of the first and second-generation container vessels. Larger vessels are not able to reach the terminal due to channel depth restrictions and container terminal facility limitations.
Vessels calling at the terminal first have to go through an approach channel with the help of a pilot and/or a tugboat. After having passed the channel, vessels wait in the anchorage until permission is granted to moor at the quay.

Note: Unfortunately for 1997 no information is available regarding the situation at the anchorage. According to the terminal management, the arrival times recorded in 1997 concern the arrival times of the vessels at the quay only.

Important factors for a vessel calling at a terminal, are the waiting time before being serviced and the time required for the terminal to service the ship.
These arrival and service times are discussed in the following sections.

5.4.2.1 Arrival times

In order to be able to meet the large demand of vessels wanting to be serviced, each vessel calling at the terminal is assigned a time window in which the vessel is expected to berth at the quay.

Note: The terminal records do not clearly define the actual arrival and departure time of a vessel. It is not specified whether the time required for mooring operations during arrival is included in the actual arrival time or that the arrival time relates to the time when servicing of the ship starts. Nor is it clear whether the preparation time preceding the actual departure of a vessel, is included in the actual departure time.

In any case, the recorded arrival times do give adequate information of the *"inter-arrival times"* of the vessels calling at the terminal.
This inter-arrival time of ships at the terminal, provides an indication of how busy the terminal is and how efficiently it operates: a longer inter-arrival time indicates that the terminal is either less busy or operating with a poor efficiency (and vice versa).

From the terminal records of 1997 it can be concluded that the average inter-arrival time of the ships was 8 hours and 52 (chapter 5.3.1.1).

5.4.2.2 Service times

The vessel service times give an indication of the performance of the terminal. The aim of the container terminal in Tanjung Perak is to service every vessel within 24 hours.
The time it takes to service a ship is the "*service time*" of that ship. The total time a ship spends at the quay is referred to as the "*turn-around time*".

Note: As the terminal records do not clearly define the actual arrival and departure times of the ships, it is assumed that the difference between these arrival and departure times is the turn-around time.

The terminal records of 1997 showed an average turn-around time of 18 hours and 53 minutes (chapter 5.3.1.2).

5.4.3 Portainers

There are four rail-mounted gantry cranes (portainers) at the berth (see photo below). The cranes lift containers on to and from the vessel. Tractor-trailers deliver and collect the containers. These cranes are able to change position by rail in 15 to 20 minutes provided the way is not obstructed.
The 1997 terminal records showed an average portainer production of about 21 boxes per crane per hour.

Note: In the near future it is envisaged to increase the portainer production to 30 boxes per crane per hour by increasing the efficiency of the terminal (no information is available about maintenance schedules or breakdown times).

The assignment of the portainers to a vessel goes according to the following rules:

- one vessel at the quay: two portainers are assigned to this vessel;
- two vessels at the quay: both vessels receive two portainers;
- in case a third ship arrives at the quay, the ship that is already the longest at the quay will continue to be serviced by two portainers whereas the other two ships will each remain with one portainer.

This portainer assignment schedule was reported to be applied in a consistent manner.

Note: However, analysis of the terminal records (see appendix 5.2) indicate that the turn-around times determined following above rules, do not always match the actual turn-around times (even if uncertainties are taken into consideration): apparently, sometimes not the longest residing ship but another ship at the quay is being assigned with two portainers.

Presently the terminal is experimenting with the simultaneously unloading and loading of a vessel: after unloading a few ship bays, one portainer continues unloading while the other portainer can start loading the vessel. It is assumed that this method may be applied with approximately 30 % of the ships calling at the terminal and that this will result in shorter turn-around times.

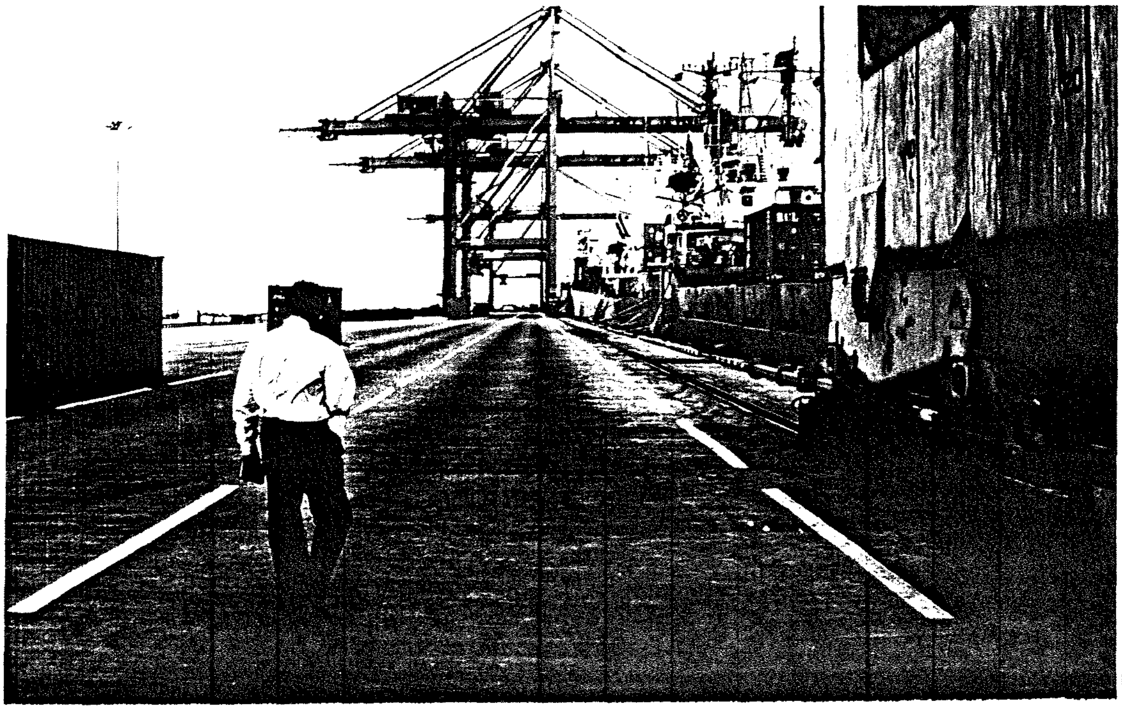


figure 5.10: Photo portainer

5.4.4 Stacks

In the container yard there are 10 stacks available for the storage of containers, as follows:

- five "*export stacks*" for containers with an export destination;
- four "*import stacks*" for containers with an import destination;
- one "*empty stack*" for empty containers.

These stacks, numbered from A to J (see figure 5.11 below), have each a maximum capacity of 1848 TEU (stacked 66 TEU long, 7 TEU wide and 4 TEU high). At the end of each stack there is space available for off-standard containers and containers which have been left behind. Stacks B and C have 56 slots for refrigerated containers (reefers). The occupancy of the export stacks is about 65 % and of the import and empty stacks about 80 %.

Stacking procedures:
The method of stacking area allocation differs between export and import destination:

- export stacks: area is assigned to a vessel and the containers are stacked according to their size and destination;
- import and empty stacks: random stacking whereby the containers are placed wherever it is most convenient. The precise location of each container in the stacks is recorded on paper and afterwards registered by computer.

Loading of a vessel is carried out according to a stowage plan. This plan, prepared by the ship's officers, is delivered to the terminal in order to determine the sequence in which the containers are to be delivered to the vessel. Preferably such stowage plans should be known to the terminal well before loading of the vessel is started. If the stowage plan is known in advance, the terminal can make the necessary preparations so as to avoid delays in loading (if required, involving re-arrangement of containers in the stacks). Unfortunately however, often stowage plans are only delivered to the terminal personnel at the moment of arrival of the ship at the terminal.

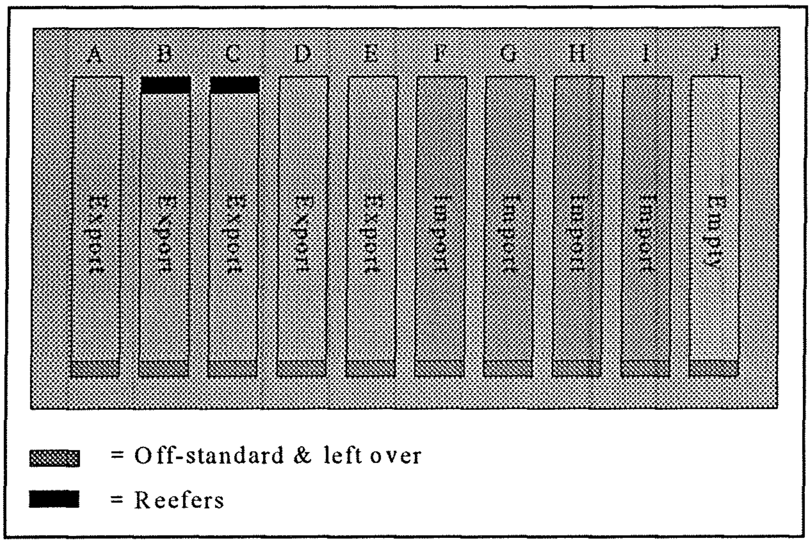


figure 5.11: Overview stacks in container yard

5.4.5 RTGC's

There are 11 rubber tired gantry cranes (RTGC) available in the yard (see photo below). These RTGC's, which can be deployed in any one of the ten stacks, take care of the stacking and reclaiming of the containers into respectively out of the stacks. Both tractor-trailers and trucks are loaded and unloaded by the RTGC's. The speed of an RTGC while driving to another location, varies between 5 and 10 km per hour. The survey reports (see 5.3.3) showed that the average RTGC production is 30 boxes per hour.

Most of the time each stack receives one RTGC. In practice however, during unloading or loading of a vessel some of the RTGC's may be re-arranged in the yard. There is not a clear set of rules concerning the procedure of assigning RTGC's to certain stacks. A description of what is basically happening in the yard is given hereafter.

Method of RTGC assignment:

- in case a vessel is being serviced by two portainers, the import stack assigned to that vessel will receive two RTGC's during unloading of the vessel. During loading of such a vessel, the export stack assigned to that vessel will also get two RTGC's. In this way the tractor-trailers are serviced as quickly as possible. In the event that two vessels are being unloaded or loaded at the same time and from the same stack, this could mean that even four RTGC's are operating in one stack. In practice however this situation (two or three vessels being unloaded or loaded from the same stack at the same time) rarely applies.
- a third RTGC may be assigned to a stack in cases that trucks are waiting to deliver containers which will be needed to load a vessel that is due to arrive within a few days. The assignment of a third RTGC in a stack, will only be done if such a RTGC does not interfere with the other activities;
- if not operational, a RTGC is parked in a parking lot at the head of the stacks (see figure 5.12).

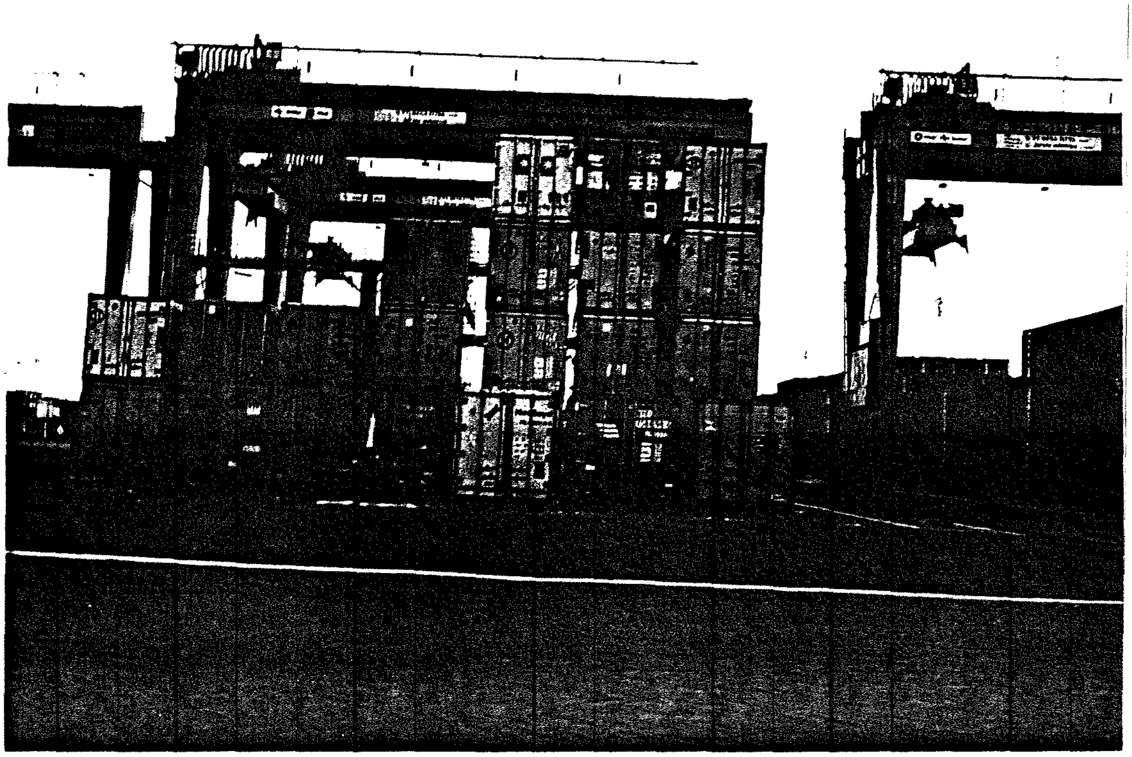


figure 5.12: Photo RTGC

5.4.6 Tractor trailers

Tractor-trailers (TT's), also called "head trucks", are used for the transport of containers from the stacks to the quay and vice versa. The maximum speed of a tt is set at 30 km per hour (mainly to avoid accidents). The tt's can transport a maximum load of 2 TEU: either one 40 ft container or two 20 ft containers. In case a tt is not operational, it is parked in the tractor-trailer parking lot at the NE corner of the terminal yard (see figure 5.1).
According to terminal management, seven tractor-trailers are working for each portainer in operation, or in total 28 tt's (4 portainers x 7 tt's). Apparently there are more tractor-trailers available (about 40 tractor-trailers according to appendix 5.1). Based on information obtained from people with first-hand experience, in practice the number of operational tt's does not reach 28 in case all four portainer are in use. It is estimated that the actual number of tt's operational per portainer, is only 5 or 6 instead of 7 (or 20 to 24 tt's in total). The reasons for this lower number of operational tt's are not known. A brief description of the tractor-trailer movements is given below.

Tractor-trailer traffic:
The destination of the tractor-trailer is dependent on the load it is carrying:

- a tractor-trailer arriving at the yard to pick-up a container, will drive to the RTGC with the shortest transporter queue and join this queue. Once loaded the tt driver receives a slip of paper with instructions for delivery to a ship and portainer. The tt proceeds to the quay to deliver the container to the portainer and ship designated, handing-in the slip of paper to the portainer operator;
- a tractor trailer going to pick-up a container from a vessel, drives to the unloading portainer with the shortest queue and joins this queue. After collecting his load, the tt driver receives a slip of paper with instructions for delivery in the yard. The tt then proceeds to the correct stack, handing-in the slip of paper, with container information, to the RTGC operator.

It is noted that the tt drivers tend to drive to the portainer that is nearest on the quay. This practice may result in a slightly lower production of the portainers working at the end of the quay.
In the event that loading and unloading activities are taking place at the same time, it is possible for a tt driver to cut down "empty return traffic": after delivery of his load to a stack, the tt may be able to

pick-up another portainer before returning to the quay (or vice versa). In this way not only the productivity of the tractor-trailers is increased, but also the salary of the drivers (who are paid according to the number of effective moves made).

5.4.7 Trucks

Different types of trucks are used to transport containers to and from the terminal. Although precise information is lacking, it has been reported that the majority of trucks have a capacity to carry two TEU at a maximum speed of 30 km per hour.

Truck traffic procedures:

- Upon arrival at the harbour, a truck is first directed to the truck parking area's outside the terminal (see figure 5.1). After having full-filled the various administrative and custom procedures at the terminal office, the driver receives instructions to proceed to a certain stack.
- Before being granted permission to enter the yard, trucks have to wait at the gate. Particularly in case that three vessels are being serviced at the quay, trucks may have to wait several hours resulting in dozens of trucks lined-up at the gate (RTGC's operating in a stack assigned to a ship that is berthed, give priority to tractor-trailers while the amount of trucks permitted in this stack is minimised). In order to minimise this waiting time, trucks are allowed to enter once there is a gap in tt traffic in a stack. It is not clear how these gaps are determined by the terminal management and how many trucks are admitted during such gaps: apparently this practice depends on the experience and skill of the people involved.
- Once a truck is admitted it proceeds to the designated stack. If it's too busy in the stack, a truck may have to wait in a parking lot (capacity about 25 trucks) in the northern part of the yard (see figure 5.1).
- After having delivered or picked-up its load, the truck leaves the yard through the gate without further delay.

The timing of export and import container traffic to and from the terminal, is further explained in the next two sections.

5.4.7.1 Export container arrival process

According to the terminal authorities, the export containers are delivered by the trucks in a time span of 7 days starting 8 days before arrival of the vessel (with an average container dwell time of four to five days). In this way, one day before arrival of the vessel all the containers for this vessel are present on the terminal. The distribution of container arrivals in 1997 is shown in figure 5.13 below.

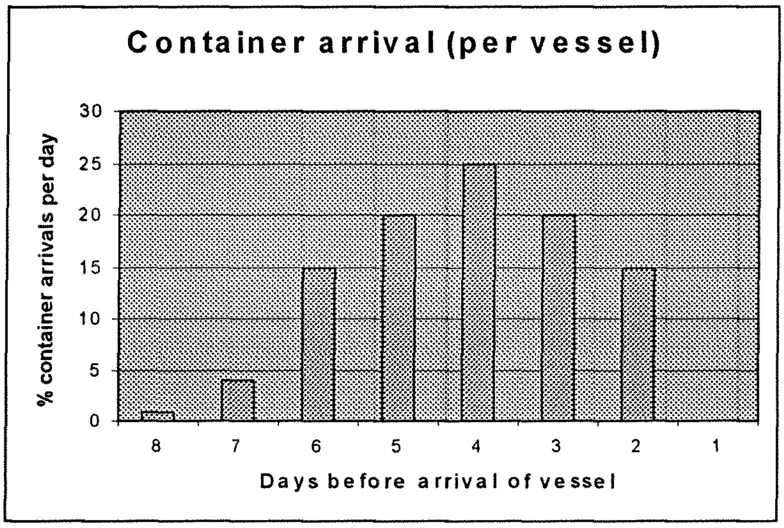


figure 5.13: Distribution of export container arrival per vessel

5.4.7.2 Import container departure process

Import containers first have to remain one day at the terminal because of custom procedures. During the following 10 days the containers are being picked-up. The average dwell time of the containers is about 8 days. The distribution of container departure in 1997 is illustrated in figure 5.14 below.

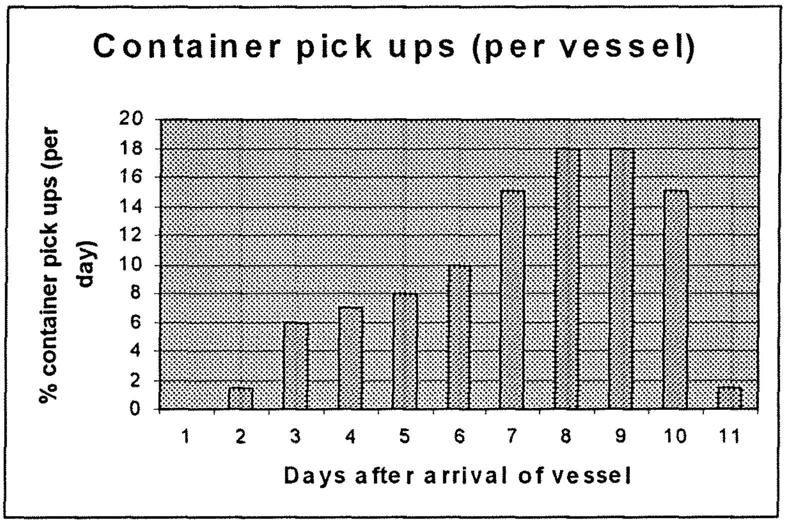


figure 5.14: Distribution of import container pick up per vessel

5.5 Computer simulation model ICT1

5.5.1 Component description and modelling assumptions

In this section the various boundaries, assumptions and components of the computer model are explained.

5.5.1.1 Model boundaries and general assumptions

This model has two "*fictive boundaries*": one located on the seaside and one on the landside (see figure 5.15):

- seaside: this boundary is set at the harbor anchorage site, where the vessels arrive from the sea eight days after having been generated (eight days is the time needed for the trucks to deliver all the necessary export containers). In reality the vessels have to navigate through the approach channel which may cause a delay. In the model, it has been assumed that pilot and tugboat assistance is always available and that there is no restriction concerning the number of vessels that can anchor in the so-called anchorage while waiting for admittance to the terminal quay;
- landside: the gate of the terminal yard is the landside boundary. Trucks are generated to arrive at this gate and wait there in parking lots until further notice (this corresponds with the actual practice).

Terminal records of 1997 show the ship length and the approximate number of box moves to be made as well as the time of arrival and departure of each ship.
In order to verify and validate the model with the actual data as best as possible, runs have been made using an input file containing the actual 1997 ship arrival times, loads and ship lengths. Since such information is not available for the future situation, these input parameters have been created by a separate model for the future terminals (this future situation is further explained in the next chapter).

Note: The total number of box moves to be made per ship, is an expected value transmitted to the terminal by the shipping companies. In reality and according to the 1997 terminal records, which present an overview of the actual number of box moves made per month, there have been deviations due to the fact that vessels required more or less box moves than expected.

It has been assumed that there is no container traffic between the stacks in the yard and no container movement between vessels. In the model therefore, all containers arriving at the terminal from the vessels will leave the terminal by truck (or vice versa). In addition, container traffic between the stacks and the container freight station has been neglected. These assumptions have been made because of lack of information on the actual occurrence and time requirement of these activities.

Because of lack of information on the handling procedure of refrigerated (reefer) and off-size containers and in view of their small numbers, also these special containers have been neglected in the model.

Since there is no exact information available with respect to the frequency and time that vessels have to wait at the anchorage, the model can not be validated on this aspect.

Note: A comparison between the expected arrival times of the vessels and the actual arrival times, showed that in a significant number of cases the actual arrival time of a vessel was later than the expected time. In some months up to 90% of the vessels calling at the terminal had a later arrival time than expected (e.g. due to the fact that vessels may be delayed during their voyage or that vessels are not permitted to moor at the quay after arriving at the anchorage). The assumption that vessels have to wait at the anchorage, therefore appears reasonable.

The model has three *"traffic lights"*: one at the entrance to the bridge from the yard, one when entering the bridge from the quay side and one traffic light at the gate where trucks enter the container yard. In this way it is ensured that tt's keep at a certain distance when driving over the bridge. The same applies for trucks entering the yard.

Note: In practice there are no traffic lights at these points on the terminal grounds.

Factors that effect the ship service times considerably, but have not in first instance been included in the model because of lack of information, are:

- *"down-time of equipment"*: extent of equipment breakdown not recorded;
- unforeseen delays as the result of bad weather conditions and strikes;
- *"human factors"*: loss of effective time in connection with possible non-adherence to prescribed schedules for breaks and changing of shifts.

In first instance, the model will run without taking into account above unknown influences and thus result in shorter ship service times. By comparing these service times with the actual, recorded service times, it will be possible to estimate the effect of the combination of above listed factors.
Following this comparison, a variable is introduced in the model. This variable, which represents all the factors influencing (delaying) the servicing of the vessels, is actually a measure of the efficiency of the operations at the terminal. This factor is referred to as the *"influence factor"* and is further explained in 5.4.3.

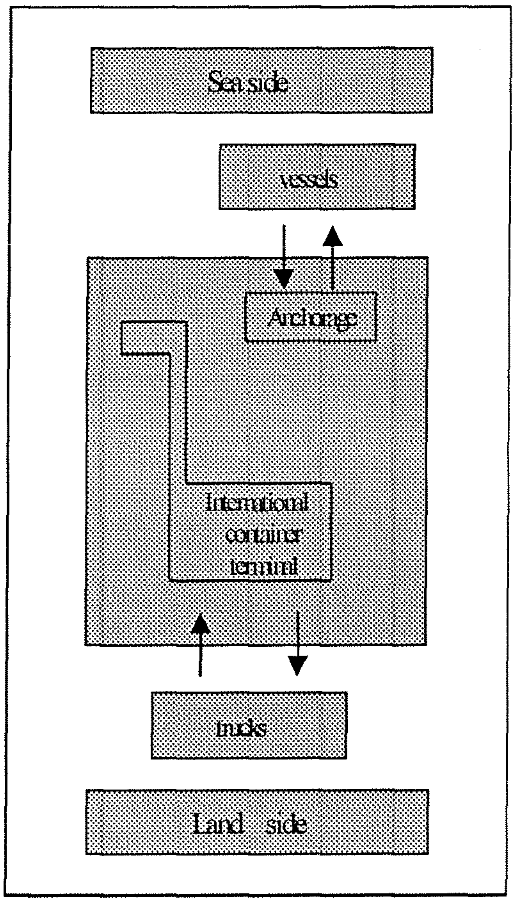


figure 5.15: Model boundaries ICT

5.5.1.2 Model components

The model comprises a number of components interacting with each other in such a way that they reflect the actual system as close as possible. The components used in the model are listed below and explained in the following paragraphs.

Components	Role
"shipgenerator"	to create the container vessels with their load and attributes;
"berthmaster"	to manage the berth distribution;
"portainermaster"	to instruct the portainers;
"tt-master"	to instruct the tractor-trailers;
"trafficmasterquay"	to manage the traffic on the quay;
"trafficmasteryard"	to manage the traffic in the yard;
"truckgatemaster"	to manage the traffic arriving at the terminal gate;
"traffic light"	to manage the traffic over the bridge and at the terminal gate.

Each PROSIM model requires the so-called "MAIN" component, taking care of the simulation run control.

The following components are "class components":
- Ship;
- Portainer;
- RTGC;
- Transporter, tractor-trailers and trucks: transporter components;
- "importgenerator": generating trucks to pick-up import containers;
- "exportgenerator": generating trucks to deliver containers for export.

In addition, the model uses "MACRO's" which are activated by a module to perform specific actions. The names of these MACRO's and their functions will be treated in the next paragraphs.

5.5.1.3 The component shipgenerator

The component "*shipgenerator*" generates the vessels. The shipgenerator does what in practice the shipping companies are doing: generating a vessel with its specific information to call at the terminal for servicing.

After a vessel has been generated, it receives attributes and information on what is to be unloaded or loaded: this is done in the "*MACRO shipattributes*". These macros are separate modules describing certain actions which are repeated by the components. In this way the component description is simplified.

For each vessel, the shipgenerator also creates an "*exportgenerator*": this is done in the "*Macro expgen_gen*". The macro's are further explained under a and b below.

Eight days after the exportgenerator has been started for a ship, the vessel arrives at the harbor anchorage and will wait there for further orders. In these eight days all the containers required for by the ship will arrive at the terminal by truck as was earlier explained in 5.4.7.
The shipgenerator waits a certain inter-arrival time before creating the next ship. This inter-arrival time is an attribute of the vessel that has just been created.

a) Macro shipattributes

In the "*macro shipattributes*" values are given to certain ship attributes. The values of the ship length, the inter-arrival time, the total number of box moves to be made and the container specifics, are read from an input file. Each ship has a certain number of containers for import and a certain number for export. These containers can be either 20 ft (1 TEU) or 40ft long (2 TEU) and can be either full or empty. These specifics read from the input file, have been explained in 5.3.1.3.
In this macro also a stack in the container yard is assigned to the ship for the import containers and a stack for the export containers.

b) Macro expgen_gen

In the "*Macro expgen_gen*" the exportgenerator is created and activated. At this stage the number of trucks required for handling the exact number and types of containers is determined. Subsequently, the exportgenerator is activated, creating the trucks that deliver the export containers of correct size and load to the terminal.

5.5.1.4 The class component ship

The ships constitute the most important component of this system: after been generated by the ship generator the ships transport the containers to and from the terminal.
After eight days the ship arrives at the harbor anchorage where it waits until permission is granted to proceed to the quay. Once permission is granted by the berthmaster, the vessel sails to its designated berth place at the quay.

Note: It is assumed that the vessel next in line to berth is already set to berth when she is granted permission by the berthmaster. The sailing time from the anchorage to the quay is therefore assumed to be negligible. Furthermore it is assumed that upon arrival at the berth place, one hour is required for mooring operations before the ship is ready to be serviced.

After mooring is completed, the ship activates the "*portainermaster*" (joining a request list) and by doing so is requesting for portainers. At this stage, also the "*importgenerator*" is created: this is done in the "*Macro impgen_gen*". This generator, responsible for the creation of the trucks required to pick-up the import containers, is created and activated with a delay of one day (see 5.4.7).

Subsequently, the vessel waits in a passive state until it is reactivated by another component.
Once the vessel has been unloaded and loaded, preparations will start to get the ship ready to leave the quay and to resume its voyage. These preparations for departure are assumed to take one hour. Finally the vessel leaves its berthing place at the quay and sails through the channel to sea thereby leaving the simulation. While leaving the quay the vessel activates the berthmaster to grant permission to other vessels to enter the terminal quay.

5.5.1.5 The component berthmaster

The "*berthmaster*" component is responsible for assigning a berth to a vessel. The following options apply:

Sequence of berthmaster operations:

- once a vessel has arrived at the anchorage, the berthmaster is activated to check whether there is a berthing place available for the vessel. If there are less than 3 vessels moored at the quay and the available quay length is sufficient to receive the vessel, then a berthing place is assigned to that vessel. At that stage the available quay length is updated and the ship re-activated;
- subsequently, the berthmaster checks if there is another ship waiting in the anchorage. In case there is another ship waiting, the berthmaster checks the quay situation again. If there is no other ship, the berthmaster becomes passive until re-activated;
- if the available quay length is not sufficient or if there are no free berths, then the ship remains in the anchorage and the berthmaster becomes passive;
- when a ship leaves the quay, it activates the berthmaster. The berthmaster again becomes passive if he cannot help the ships in the anchorage or if there are no ships;

In the model it is assumed that the berthmaster works via the "first come first serve" (FCFS) principle. In practice however, the berthmaster may be in the position to assign an available berthplace to a later ship (e.g. when such a ship fits the available quay length). In practice a decision to change the sequence of ships for berthing, appears to be far more complex and depending on various factors, including the expertise of the people in charge.

Since it is not clear how these decisions are reached, above mentioned FCFS assumption has been maintained throughout the model.

5.5.1.6 The component portainermaster

The portainermaster is responsible for assigning portainers to the ships at the quay. The moment a vessel finished mooring at the quay, it activates the portainermaster to check the situation at the quay. Three situations apply: one, two or three vessels moored at the quay. Depending on the number of vessels at the quay the portainermaster assigns the portainers to the vessels (see explanatory boxes hereafter).

Option: one vessel berthed

In case only one vessel is berthed at the quay, the portainermaster assigns two portainers to this vessel with directions to the precise location at the quay. These directions depend on the berthing place of the vessel at the quay:

- in case a vessel is moored at the first berth, portainer nr. 1 is directed to "position 1" and portainer nr. 2 to "position 2" at the quay.

For a vessel at berth place two or three, portainers three and four are assigned to this vessel:

- these portainers are directed to portainer positions three and four, in case the vessel is moored at berth place two;
- and to positions five and six in case the vessel is moored at berth place three.

Note: To simplify the model, it is assumed that there are six positions at the quay where portainers can be located or directed to (see figure 5.16 below illustrating the portainer positions at the quay). Depending on instructions from the portainermaster, a portainer can either remain at its position or be directed towards a new position.

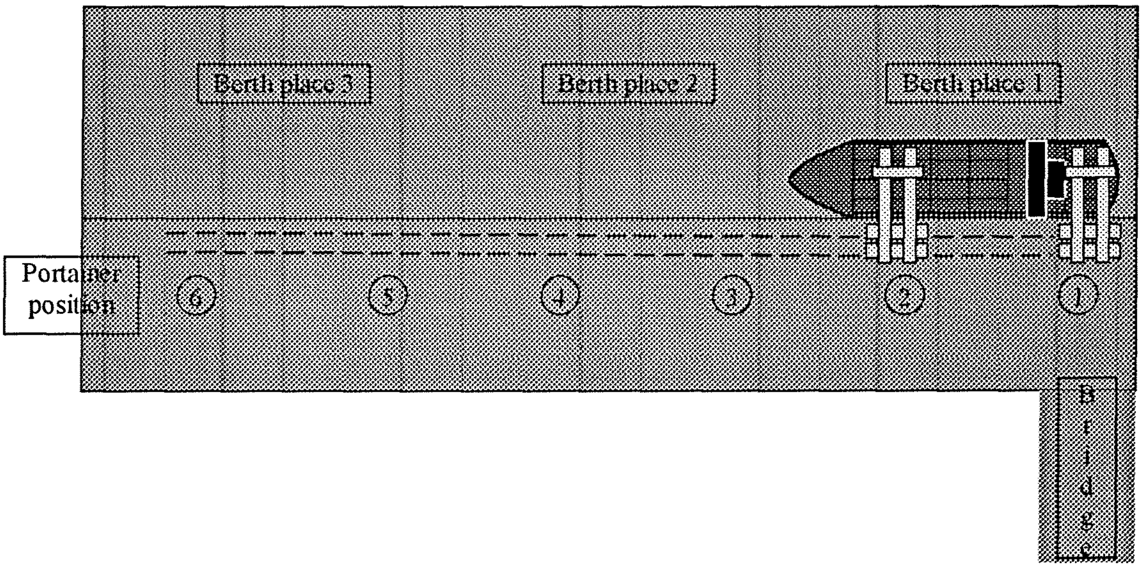


figure 5.16: Option one vessel berthed

Option: two vessels berthed When there are two vessels at the quay, they will both be assigned two portainers. • ships moored at berth places one and two: portainers one and two assigned to ship at berth place one and portainers three and four to the ship at berth place two; • ships moored at berth places two and three: portainers one and two assigned to ship at berth place two and portainers three and four to the ship at berth place three; • ships moored at berth places one and three: portainers one and two assigned to ship at berth place one and portainers three and four to ship at berth place three.

Above situation is clarified in figure 5.17 below.

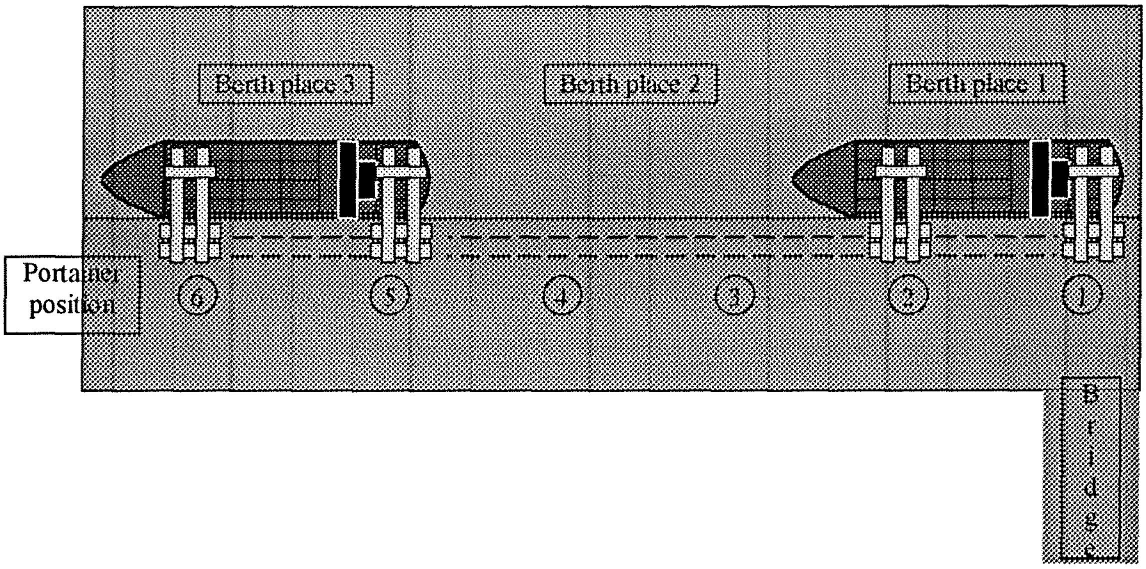


figure 5.17: Option two vessels berthed

Option: three vessels berthed
If there are three vessels at the quay, then the ship that arrived first at the quay is assigned two portainers and the other two ships one portainer each.

- ship with longest berth time moored at berth place one: portainers one and two are assigned;
- ship with longest berth time moored at berth place two: portainers two and three are assigned;
- ship with longest berth time moored at berth place three: portainers three and four are assigned to this ship (see figure 5.18 below).

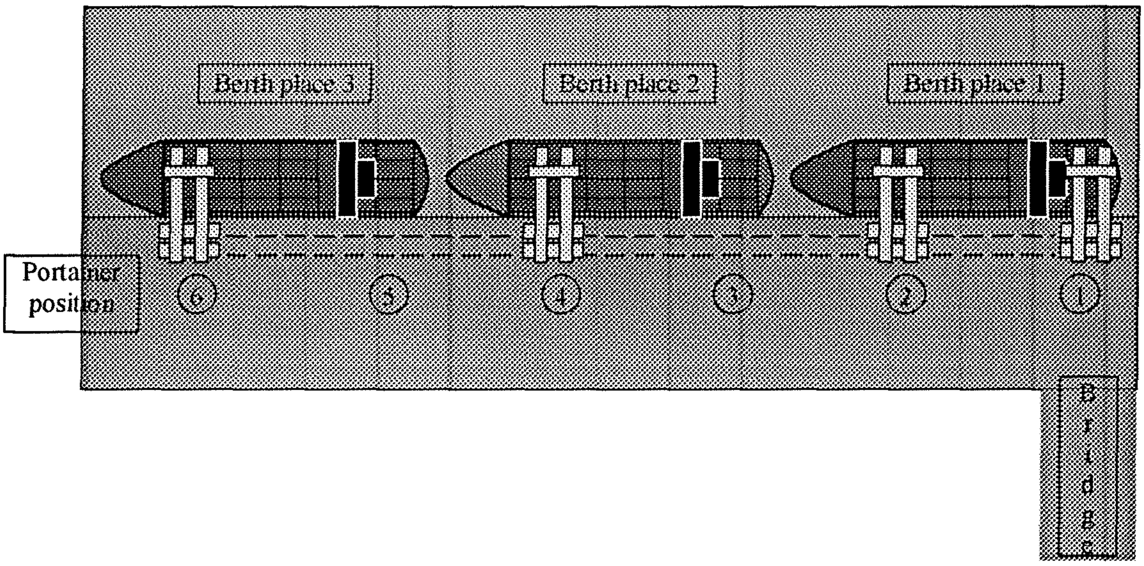


figure 5.18: Option three vessels berthed

It may happen that a portainer when being assigned to a vessel, is at another location at the quay: in such a case the portainer is directed towards its assigned location.

Note: This repositioning of a portainer is only possible when there are no other portainers in its path (all portainers use the same rails). In case its path is blocked by another portainer, it has to wait till this portainer has cleared the way. Once the path is clear, the portainer moves to its destination. This repositioning of a portainer is assumed to take 15 minutes.

If a portainer, while in the process of loading or unloading of a ship, is being assigned to another ship, then this portainer will receive instructions to halt this activity and shift to its new location as soon as possible.
Once a ship has received the portainers assigned to it, this ship is taken from the list of portainer requests. Subsequently, the portainermaster is activated to check whether there are other ships on the request list.

The assignment of a portainer to a certain ship and to a certain position at the quay, is handled by the "*Macro portainerstart*". In this macro the "*tractor-trailer master*" is activated to assign tractor-trailers to this portainer if needed (i.e. when a portainer does not yet have tractor-trailers assigned to it). In this macro, also the portainer is activated.

Macro portainerstart
In the "*Macro portainerstart*" the portainer is directed to a certain position (if not already at the right location).
In the case two portainers are assigned to a ship, also a second RTGC is directed to the stack assigned to that particular vessel and activated. This additional RTGC is directed to either the import or export stack (depending on whether a ship is being unloaded or loaded).
If a portainer does not have tt's assigned to it, this portainer is added on to the request list for tt's. The "*tractor-trailer master*" is then activated to assign tt's to the portainer in question. Subsequently the portainer is activated.

5.5.1.7 The class component portainer

The portainers are designated to unload or load vessels that are berthed. The "*portainermaster*" assigns a portainer to a particular vessel and directs it to a certain location at the quay, taking into account the situation at that moment. If the portainer finds itself already at the correct location, it first checks whether the vessel in question needs to be unloaded:

- if yes, unloading is started;
- if not, the portainer checks whether its vessel needs to be loaded. If this is confirmed, the portainer starts to load the ship.

It may happen that a portainer, which is not at the correct location at the quay, has to wait until it is allowed to move to its correct location. Other portainers that are still operating may stand in the way. These portainers are by then also instructed by the portainermaster to clear the way as soon as possible (such portainers may be in the process of unloading or loading which activity has then first to be finished).

When a portainer is unloading a vessel, it starts to retrieve a container from the vessel. With this container clamped into its spreader, the portainer waits until a tractor-trailer has arrived under the portainer and then lowers the container onto the tt (the portainer will always try to load 2 TEU onto a tt).
Once a tt is loaded, the portainer retrieves a container for the next tt. In between these activities, the portainer driver checks whether or not his portainer has been assigned to another vessel: if not, unloading is continued. In case the portainer is instructed to move to another location, it will do so if possible (repositioning of a portainer is only possible when there is no container clamped into its spreader).

Once unloading of a vessel is finished, the portainer starts loading this vessel. In this case the empty spreader waits for a loaded tt to arrive under the portainer. These loaded tt's are waiting in the queue near the portainer. After loading a container in the vessel, the portainer returns to its former position (ready to pick-up the next container). The attributes of each loaded container are given to the portainer driver. The portainer drives then checks whether there is a change of plan involving a repositioning of the portainer along the quay.

Before a portainer starts loading a vessel, it first checks whether that vessel is assigned a second portainer. In case two portainers are assigned to load a ship, one additional RTGC is activated in the stack from which the export containers are to be picked-up.
The portainer becomes passive once a vessel is loaded and when there is nothing left to do at that location. At that stage, the tt's assigned to this portainer receive instructions to return to the tt parking lot as soon as possible. Likewise, the additional RTGC assigned to such ship becomes passive as well. Finally the portainermaster is activated to check whether there is any other vessel on the portainer request list.

If a portainer receives instructions to change its location while still loading/unloading a tt, it first finishes servicing this tt.

- in the case this portainer was busy **unloading** a ship, the portainer driver directs the tt's - which are at that time still waiting in the queue - to the quay master for new instructions. Only after all tt's - waiting for the portainer - have left, then the portainer is ready to move to a new location (provided that the path is free);
- in the case that the portainer receives instruction to move, while still **loading** a ship, it first continues to unload all the tt's waiting in the queue with their loads. Other tt's that are at that time arriving with a load for the same vessel, are instructed to either go to another portainer or wait on the quay until another portainer is ready to unload these tt's at the correct ship. When there are no more tt's left in the queue, the portainer is ready to move to a new location.

Once the portainer is ready to move to a new location, the "*portainermaster*" is activated in order to give instructions to the portainer.

The tt's assigned to a particular portainer, are only sent to the parking lot when that portainer has finished loading a vessel. At that moment these tt's become passive. In case a portainer is simply changing its position along the quay, the tt's remain operative.

Note: In practice however, the tractor-trailers only go to the parking lot when the portainer they are assigned to, has no more vessels to service.

In the "*Macro quayload*", it is decided which type of container is to be unloaded (full or empty and 20ft or 40ft) and what the destination in the container yard will be. This information is passed-on to the tt driver upon reception of his load.
This macro keeps track of the number of containers still remaining to be unloaded. Similarly, this macro records the number of containers loaded on to a ship.

5.5.1.8 The component ttmaster

The "*ttmaster*" is responsible for assigning tractor-trailers to portainers requesting tt's.
The ttmaster is activated by the "*portainer master*" at the moment that an activated portainer has no tt's assigned to it. A portainer is being assigned a standard number of tractor-trailers.

Note: In practice however, a ttmaster appears to be more flexible in assigning tt's, taking into account the number of operational tt's at his disposal at a given time. Such decision will also depend on the time pressure to service a vessel, the expected arrival of more vessels and whether the terminal is operating on schedule.

Once the ttmaster has assigned the tt's to a portainer, he checks whether other portainers have put in a request for tt's.

- in case there is a request, he assigns tt's to such a portainer;
- if there is no request, the ttmaster becomes passive;
- if these are no tt's left in the parking lot, the ttmaster waits till tt's arrive in the parking lot.

5.5.1.9 The class component tractor-trailer (tt)

Tractor-trailers are used to transport containers between the portainers at the quay and the RTGC's in the container yard. The tractor-trailers can carry a maximum of 2 TEU: either two 20ft containers or one 40ft container. Whenever possible the tractor-trailers transport the maximum load of 2 TEU.

The "*ttmaster*" activates the tt's that where requested for by a portainer. In accordance with the actual practice, the activated tt's operate totally independent of this portainer and transport containers to and from any vessel berthed at a given moment.

Figure 5.19 below illustrates the possible driving cycles of the tt's.

Once a tt is activated, it leaves the parking lot and drives around the container yard until it arrives at the north-western corner of the yard where the tt receives instructions from the "*yard trafficmaster*". The yard trafficmaster is activated each time a tt passes this point of the yard: all tt's coming from the quay or from a stack have to pass this yard trafficmaster for instructions.

In case a tt is without a load, it is instructed to either pick-up a container in one of the stacks, or to drive to the quay to pick-up a container there.

Example of tractor-trailer cycle: A tt instructed to pick-up a container in one of the stacks, drives to the assigned stack and joins the queue for the corresponding RTGC. Once loaded, the tt proceeds to the bridge and drives over the bridge to the quay. Arriving at the quay, the tt passes the "*quay trafficmaster*" who directs the tt to the portainer servicing the correct vessel. The tt waits in the queue for this portainer before being unloaded. After unloading, the tt drives to the trafficmaster of the quay for new instructions. The tt is instructed either to pick-up a container at another portainer or to return to the container yard. In case the tt is instructed to pick-up another container at the quay, it drives to and joins the queue at this particular portainer. Once loaded and passing the quay trafficmaster, the tt (with its new load) drives to the bridge entrance and proceeds to the container yard. Arriving at the container yard, the tt passes the yard trafficmaster for instructions on the stack where the load is to be delivered.

Each time a tt enters the bridge, it passes traffic lights that are activated by the tractor-trailer. Depending on the activity on the bridge, a tt may either have to wait or proceed without delay.

It may happen that portainers are in the process of changing positions along the quay, at the moment that a tt arrives with a load for a vessel: in such a case the tt waits till a portainer is again operational at the correct vessel.

A tractor-trailer may carry a mixed load of one full and one empty 20 ft container. It is assumed that such mixed loading is only done by a portainer (mixed loading from stacks is unlikely because tt drivers would loose too much time in driving between two different stacks). In case of a mixed load, the tt delivers the full container first.

Once a portainer, after it has finished servicing a vessel, becomes passive, the tractor-trailers are instructed to return to the parking lot.

- if at that moment a tt is empty, it returns to the parking lot;
- if the tt is loaded (or: waiting in a queue to be loaded), it first delivers the container before proceeding to the parking lot.

Once arrived in the parking lot, a tt becomes passive and waits until re-activated by the ttmaster.

The "*yard and quay trafficmasters*" have been created in order to be able to direct a tt - after the delivery of its load to a portainer or RTGC - back to a portainer or RTGC to pick-up another load before returning to the yard or quay.

Note: Also in practice there is staff present to keep track of this tt traffic and to give proper instructions and directions to the drivers. As it is not known how many persons are in practice regulating this tt traffic (nor what exactly their job description is), the model uses one trafficmaster in the yard and one at the quay.

It is further assumed that the tractor-trailers drive at a constant speed, both in the yard, at the quay and on the bridge (for safety reasons and to avoid situations of overtaking). In addition, the model assumes that the tractor-trailers always follow the same routes on the terminal site: to ensure efficiency and allow a clear overview of traffic.

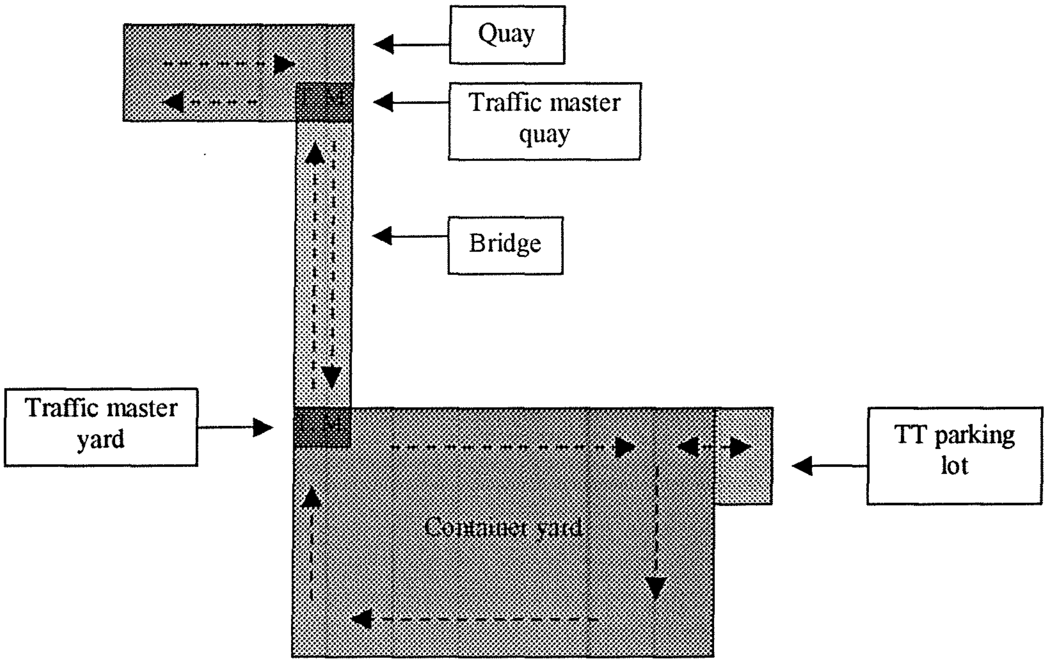


figure 5.19: Driving cycles of the tt's

5.5.1.10 The component trafficmaster of the yard

The yard trafficmaster is responsible for directing the tt's to the correct location.
The "*yard traficmaster*" is activated when a tractor-trailer is passing.

Example of yard trafficmaster instructions: If a tt is passing with a load to be delivered to one of the import or empty stacks, the trafficmaster directs the tt to the correct stack. In case an empty tt is passing, the trafficmaster first checks whether there are vessels being loaded at that moment. If there are only vessels being unloaded, then the tt is directed back to the quay to pick-up a load. In the case vessels are indeed being loaded, the tt is instructed by the trafficmaster to drive to the yard to pick-up a load for a vessel (whose portainer was sent a tt the longest time ago). In the *"Macro yardcontainer"* the tt is instructed from which stack its new load is to be collected (this can either be the export or empty stack).

Once a tractor-trailer has received its instructions, the trafficmaster checks whether there are other tt's waiting for instructions:

- in case that there are indeed other tt's, the trafficmaster remains active;
- if there are no more tt's, then the trafficmaster becomes passive until re-activated.

5.5.1.11 The component trafficmaster of the quay

The quay trafficmaster is responsible to direct the tt's to the portainers. The "*quay trafficmaster*" is activated by the tt's when they drive onto the quay or just before the tt's drive onto the bridge in the direction of the yard.

Example of quay trafficmaster instructions:
- empty tt's: the quay trafficmaster directs an empty tt to a portainer that is unloading a vessel. In case there are more than one portainers unloading, the tt is directed to the portainer that was assigned a tt the longest time ago. The trafficmaster directs a tt to the yard, in case there is no portainer unloading a vessel, or if the queue of other tt's waiting for an active portainer is too long.
- loaded tt's: the quay trafficmaster directs a tt with a load for one of the vessels to the portainer that was assigned a tt the longest time ago. It may happen that there are no portainers active at the vessel for which the tt is carrying a load: in such a case the tt driver is instructed to wait until further notice.

Once a tractor-trailer has received its instructions, the quay trafficmaster checks whether there are other tt's waiting for instructions:
- in case that there are indeed other tt's, the trafficmaster remains active;
- if there are no more tt's, then the trafficmaster becomes passive until re-activated.

5.5.1.12 The component traffic light

The three traffic lights are situated at the bridge entrance from the yard and quayside and at the gate entrance to the container yard from outside of the terminal grounds. By means of these traffic lights, a standard distance is maintained between two tt's or trucks.

Note: These "*traffic-lights*" have been included in the model to ensure that - such as in practice - vehicles are kept at a certain distance, both in the yard and at the bridge.

It may happen that there are more tt's or trucks arriving at the bridge or gate at the same time: in this case the vehicles have to wait 4 seconds until the traffic light turns green. This setting, based on the speed of the vehicles and their approximate acceleration rate, results in a minimum distance between the vehicles corresponding with roughly the length of these vehicles.

5.5.1.13 The class component RTGC (yard crane)

Containers, delivered or picked-up by tractor-trailers and trucks, are lifted onto or from the stacks by the RTGC's. A particular "*RTGC*" is activated the moment a tt or truck arrives in the queue for that RTGC.

Once activated, the yard crane unloads or loads the tt or truck in question.

Note: In case there are both tractor-trailers and trucks in the queue, it depends on the predetermined service strategy which carrier will be serviced first. In accordance with the common practice, it is assumed in the model that tt's have priority over trucks: this means that trucks in the queue will only be serviced once there are no more tt's left.

Besides lifting containers onto and from the stacks, a RTGC regularly changes its position and reshuffles the stack. The survey showed that a RTGC reshuffles the stack on average after every 5 container moves to and from the stack and that a RTGC is re-positioned on average after every 4 container moves. In the RTGC process the reshuffling and re-positioning actions have been simulated to take place after every container lift-on or lift-off move by the RTGC.

Note: Other factors effecting the working speed of the RTGC's (such as the stack volume) have not been included separately in the RTGC process. In case of a full stack, a RTGC will in practice need more time to lift a specific container on-to or from the stack, because of the larger height of lifting required (the same applies to reshuffling a full stack).
Above influence is to some extent taken into account in the model: the time required to perform these RTGC actions, is namely changed in accordance with a certain distribution (see 5.3.2).

Once a tt or truck is serviced, the RTGC checks whether there are other carriers in the queue waiting to be serviced. In case an additional RTGC is operating in the stack, the RTGC checks after each container move whether such additional RTGC is still active in the stack.

The "*Macro yardload*" keeps track of the number of containers in the stacks. At the moment that a tt or truck is loaded it will also receive the destination (assigned quay and ship).

5.5.1.14 The class component exportgenerator

The "*exportgenerator*" is responsible to create and activate trucks to deliver the containers that are to be loaded on the vessel calling-in at the terminal (these export containers are dispatched by clients from the hinterland).

The exportgenerator generates a number of trucks, including type of loading, in accordance with the number and type of export containers that the vessel is instructed to load at the terminal. The exportgenerator is generated the moment a vessel is generated and given its attributes.
The arrival of the trucks is assumed to be spread-out over 7 days: one day before arrival of the vessel, all containers are assumed to be present in the container yard (see specifics discussed in chapter 5.4.7).

Note: According to above specifics, each day a certain percentage of trucks arrive. As the terminal facilities are open 24 hours a day, a truck may arrive any time of the day or night. It is assumed that most trucks arrive during daytime between 8 o'clock in the morning and 4 o'clock in the afternoon.

The truck arrival distribution is shown in the graph below. Each truck arriving at a particular day receives an arrival time obtained from this distribution with a random number.

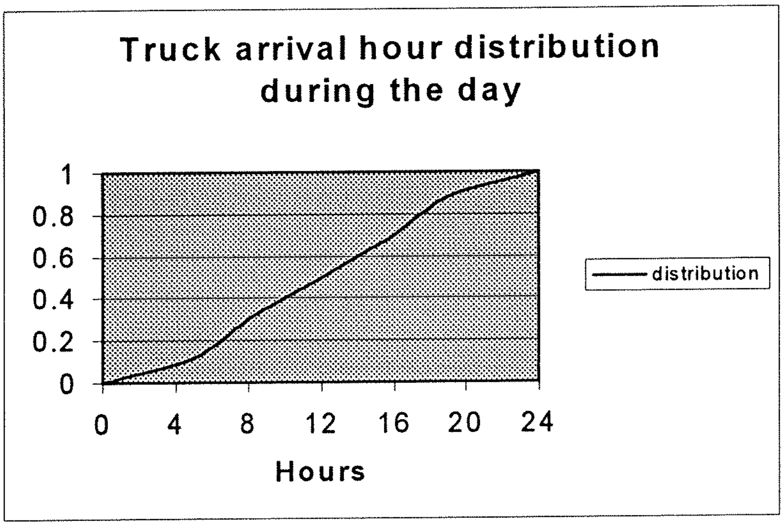


figure 5.20: Truck arrival distribution during the day

5.5.1.15 The class component importgenerator

The "*importgenerator*" is generated at the moment a vessel is moored and ready to be serviced. With a delay of one day (see chapter 5.5.1.4.), this importgenerator generates and activates the trucks required to pick-up containers unloaded from the vessel and deliver these to the clients in the hinterland. The number of trucks generated is in accordance with the number of containers delivered by the vessel to the terminal.

Note: The trucks arrive at the terminal according to the distribution discussed in chapter 5.3.7. During the day, the arrival pattern of the trucks at the terminal is assumed to be the same as for the trucks arriving with export containers (as explained in chapter 5.4.1.14).

In the "*Macro teupertruck*" the trucks receive instructions regarding the load to be picked-up in the container yard.

5.5.1.16 The class component truck

Transport of containers between the terminal and the hinterland, is carried out by trucks that are generated and activated by the "*import and export generators*". These generators also give instructions to the trucks regarding the type of containers to be picked-up or delivered to the yard.

An arriving truck joins the terminal gate queue (to be considered as a parking lot) until the gatemaster of the terminal grants the truck permission to enter the container yard. At the moment a truck arrives at the gate the "*gatemaster*" is activated. Once a truck is allowed to enter the yard, it drives to the stack where the load is to be delivered.

Note: In order to prevent chaotic traffic situations in the yard, trucks follow the same route as the tt's (see figure 5.21 below). In addition, trucks are not allowed to drive over the bridge to the quay to deliver or pick-up their load themselves.

Once a truck has been loaded or unloaded, it drives back to the terminal gate where the "*gatemaster*" is again activated at the moment the truck is leaving the terminal grounds.

It is assumed that the trucks always carry either a full or an empty load (and never a mixed load). Such load consists of either 1 TEU or 2 TEU (it is assumed that a 2 TEU load can only be one 40 ft container).

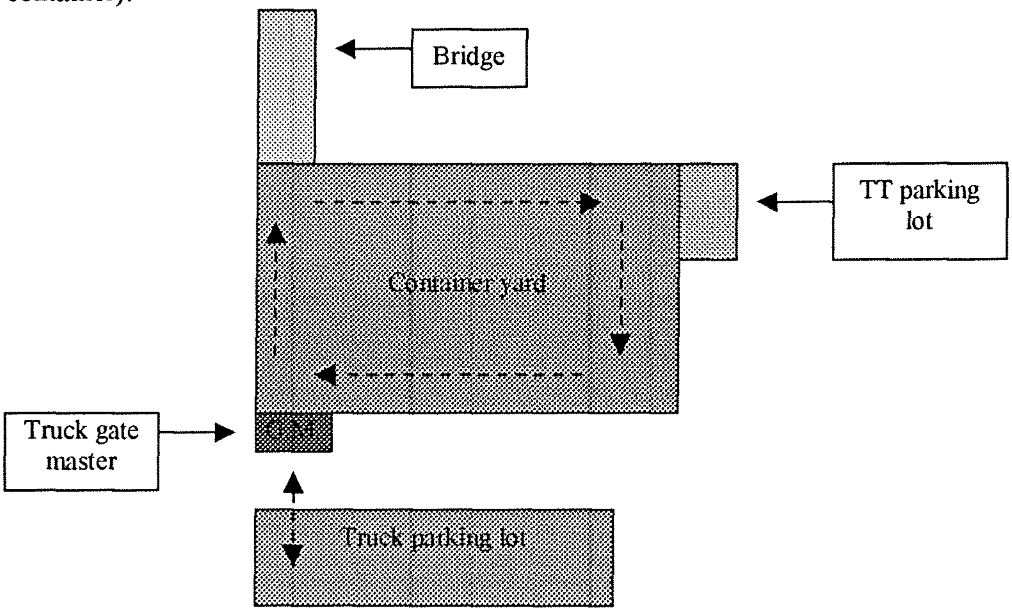


figure 5.21: Driving cycle of the trucks in the yard

5.5.1.17 The component truckgatemaster

The "*gatemaster*", activated by a truck arriving at the gate, grants permission to trucks to enter the container yard.
The gatemaster checks whether it is possible for the truck to enter the yard and the designated stack:

- the truck is allowed to enter in case the stack is open for trucks and when there are not too many trucks present already;
- in case there are already too many trucks in the stack, the truck remains passive at the gate until further notice from the gatemaster. Subsequently, the gatemaster checks the situation for the next truck in the parking lot.
- in case the stack is closed for trucks (see 5.4.7), the truck remains passive, and the gatemaster checks the situation for the next truck in the parking lot.

The gatemaster is also activated by trucks leaving the yard. In case the leaving truck did come from the same stack, such departure will at that time also result in the permission for the waiting truck to proceed to that particular stack.

5.5.2 Verification and validation of model

5.5.2.1 Verification

Model verification implies that the model must be able to handle a number of tests, thus making sure that it works correctly.

The personal PROSIM run control system offers excellent verification facilities. With the "*trace function*" a specific component can be selected to be followed during the run, whereby also the attributes can be viewed. In this way the programmer, who is familiar with the sequence of the actions, can immediately notice any incorrect action.

Animation constitutes another possibility for a rapid verification of the model. The advantage of making an animation of a major action in the model, is that the programmer is able to observe on the screen what is exactly happening during the run.

The present model has been verified with the help of both the trace function and the animations.

5.5.2.2 Validation

The model generates simulated computer data. Validation of the model consists of various investigations carried out in order to ascertain whether the model is valid. This validation requires checking whether the data generated by the model are similar to the actual recorded data.

The present model has been validated by comparing the data obtained from the model with the data derived from the terminal records of 1997. The validation results are discussed in the paragraph 5.5.3 Sensitivity analysis & computer simulation results.

5.5.3 Sensitivity analysis & computer simulation results

A number of simulation runs have been made with different parameters to test the sensitivity of the model.

Because in 1997 the number of vessels serviced by the terminal and thus the number of container moves made, varied from month to month, it was decided to run the model with the input data from a selection of months. For the validation of the operations of the model, the input data for the months of January, February, October and November have been used.

Note: In addition, a couple of runs have been made with the input data for the whole year. However, as these runs proved to be quite time consuming, it was decided to validate the model on the basis of the output obtained from the runs with the four above mentioned months only.

An "*influence factor*" was incorporated in the model.

Note: This influence factor represents all the unknown or unpredictable factors which in practice effect the service time of a vessel, such as breaks and shift irregularities, equipment breakdown, bad weather etc.

With this influence factor set at 1 it is assumed that there is no disturbing influence of these factors. By reducing the value of this factor, the influence of these unknown factors on the turn-around time of the vessels is increased.

The present model was run with different values of this influence factor in order to determine which value results in the best representation of the actual terminal operations.

Note: In the model this influence factor changes the production rate of the portainers and thus the berthing time of the vessels. In practice however the net portainer production rate itself is constant whereas only the overall production rate is effected. On the other hand when a vessel - due to circumstances - has to remain longer at the quay, this results automatically in a smaller average number of container moves per hour. In the model, this same effect is achieved by the influence factor.

In addition, the number of tt's assigned to a portainer has been varied in the model (in view of the fact that the actual numbers were not exactly recorded). Also the effects have been investigated for the situations in which:

- trucks are continuously admitted to the container yard;
- tt's are not given priority in the yard.

5.5.3.1 Portainer production & influence factor measurements

According to the terminal records, the portainer production during 1997 varied between 19 and 21 box moves per hour per crane (in the simulation runs a net portainer production rate of 21 box moves per hour is applied).

When the model was run with the influence factor set at 1 (i.e. assuming no disturbing or delaying influences), the portainer production reached an average of 20.5 box moves per hour with 6 tt's assigned to the portainer. The lower value of the portainer production rate as compared to the net portainer production rate is caused by the fact that the portainer apparently occasionally has to wait for a tractor-trailer.
Table 5.6 below illustrates that this production rate decreased when this influence factor is reduced: more delays occurring during the period that a vessel is moored, thus resulting in an average lower number of box moves per hour.

January & February:

	Portainer production (box moves/hour/crane)			
	Influence factor			
# tt per pt	1	0.7	0.65	0.6
4	18.6	14.2	13.3	
5	20.0	14.6	13.6	
6	20.5	14.7	13.7	12.6
7	20.6	14.7	13.6	
10	20.7	14.6	13.6	

table 5.6: Portainer production rate versus efficiency level

5.5.3.2 Vessel turn-around time

According to the terminal records of 1997 (and excluding extreme values), the average turn-around time of a vessel was 17.4 hours in January/February, 18.4 hours during the months of October/November and 18.8 hours for the whole year.

Assuming no delaying factors (influence factor = 1), the model results in an average turn-around time of about 12 hours in January/February and 12.5 hours in October/November.

By introducing the influence factor, thereby taking into account the effect of various delaying factors, the model results in realistic, higher turn-around times of the vessels as anticipated (see tables and graphs below):

- for the months of January/February, an influence factor between 0.7 and 0.65 resulted in an average turn-around time closest to the actual recorded average (see table 5.8);
- for the months of October/November, the best fit was obtained with the influence factor set at 0.7 (see table 5.9).

For the runs using the input data from the whole year of 1997, an influence factor of approximately 0.67 resulted in the average turn-around time of 18.8 hours as calculated from the 1997 terminal records (see table 5.7).

1997:

Turn-around time		
influence factor		
0.7	0.68	0.65
17.5	18.1	19.2

table 5.7: Turn-around time (hours) vessels in 1997 with varying influence factor (and 6 tt per portainer)

January and February:

	Turn-around time			
	Influence factor			
# tt per pt	1	0.7	0.65	0.6
4	12.9	17.0	18.4	
5	12.1	16.5	17.8	
6	11.8	16.4	17.7	19.6
7	11.7	16.4	17.8	
10	11.7	16.6	17.9	

table 5.8: Turn-around time (hours) vessels in January & February versus influence factor

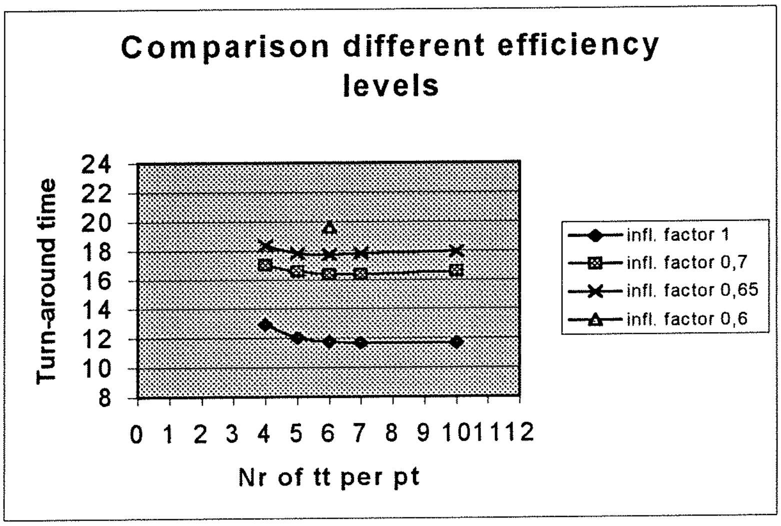


figure 5.22: Effect influence factor and number of tt's on turn-around time (hours) vessels in Jan. & Feb.

October and November:

	Turn-around time			
	Influence factor			
# tt per pt	1	0.75	0.7	0.65
5	12.9		18.8	
6	12.6	16.9	18.6	20.6
7	12.5		18.6	

tTable 5.9: Turn-around time (hours) vessels versus influence factors

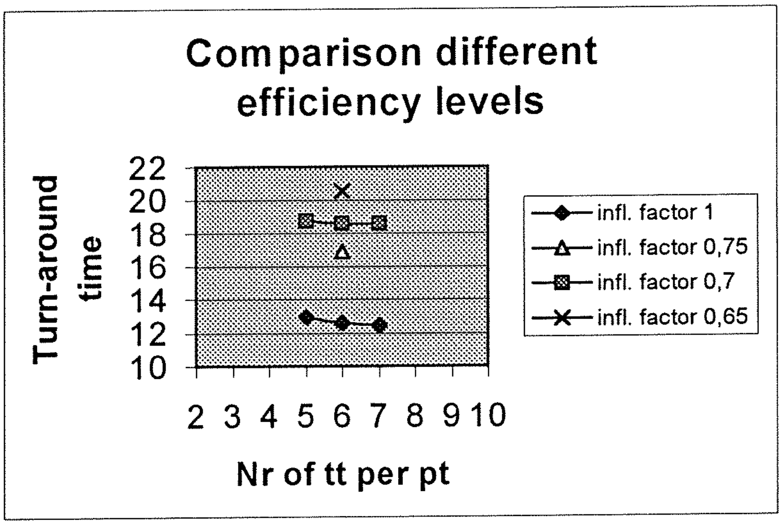


figure 5.23: Effect influence factor and number of tt's on turn-around time (hours) vessel in Oct. & Nov.

5.5.3.3 Number of tt's per portainer

For 1997 it has been assumed that a portainer was assigned 6 tt's on average (see appendix 5.1):

- the portainer production rate is effected if assigned with less than 5 tt's: apparently a portainer sometimes has to wait for tt's (see tables in 5.5.3.1);
- the portainer production rate does not increase significantly, if assigned with more than 5 tt's: in this case the tt's have to wait, which does not effect the portainer operations as such.

The model confirmed above effect on the turn-around time of the vessels:

- the turn-around time increases with less than 5 tt's;
- but is hardly reduced further if more than 5 tt's are assigned.

5.5.3.4 Quay & portainer occupancy

The "*quay occupancy rate*" corresponds with the % of the time that in a given period, vessels are berthed at the quay. According to terminal management, this quay occupancy rate can be as high as 70 %.

With the influence factor set at 1 (i.e. assuming no delaying factors), the corresponding "theoretical" occupancy rate of the quay will naturally be much lower than above 70 %: namely 40% during January/February and 50 % for the period October/November (see tables 5.10 and 5.11 below).

With the influence factor set at 0.65 the quay occupancy during January/February increased to about 60%. During the period October/November this occupancy rate reached 73 % with the influence factor set at 0.7.
With above influence factors, roughly the same turn-around times of vessels are obtained as the actual, recorded times.

Note: The difference in quay occupancy between above two periods, is attributed to the fact that during the months of October/November considerably more vessels are being serviced with a corresponding larger number of container moves. This results in higher quay and portainer occupancy rates as compared to the January/February period.

The runs for the whole of 1997, resulted in an average quay occupancy rate of about 70 % with an influence factor between 0.7 and 0.65 (see table 5.9 below).

The "*portainer occupancy rate*" reflects the % of time that in the simulation run the portainers are assigned to vessels (not passive). In case servicing of a vessel is delayed because of disturbing factors, then servicing of such a vessel will take longer, which in turn will result in the situation that portainers are assigned to a vessel during a longer period. If the number of vessels berthed is not changing, the portainer occupancy rate will therefore increase if the value of the influence factor is reduced (i.e. assuming more influence of delaying factors).

With the influence factor set at 1 (i.e. no delaying factors), the vessels will theoretically be served quicker, resulting in portainers becoming passive more often: consequently the model resulted in a low "theoretical" portainer occupancy rate between 50 and 60 %.
With the influence factor set at 0.65, this occupancy rate reached almost 70 % during the period January/February. For the period October/November this occupancy rate reached 80 % with the influence factor of 0.7.
The runs for the whole of 1997, resulted in a comparable average portainer occupancy rate of about 79 % when the influence factor is reduced to 0.65 (see table 5.9 below).

Above confirms that the portainers are quite occupied, particularly so during the busy October/November months.

1997:

influence factor					
0.7		0.68		0.65	
av. pt. occ.	av. quay occ.	av. pt. occ.	av. quay occ.	av. pt. occ.	av. quay occ.
0.74	0.66	0.76	0.68	0.79	0.72

table 5.9: Effect influence factor on quay and portainer occupancies in 1997 (and 6 tt per portainer)

January & February:

	influence factor							
	1		0.7		0.65		0.6	
# tt per pt	av. pt occ.	av. quay occ.	av. pt occ.	av. quay occ.	av. pt occ.	av. quay occ.	av. pt occ.	av. quay occ.
4	0.51	0.45	0.67	0.59	0.71	0.63		
5	0.48	0.42	0.65	0.57	0.7	0.62		
6	0.47	0.41	0.64	0.57	0.69	0.61	0.75	0.68
7	0.46	0.41	0.64	0.57	0.69	0.62		
10	0.46	0.405	0.65	0.57	0.7	0.62		

table 5.10: Effect influence factor and number of tt's on quay and portainer occupancies in Jan. & Feb.

October & November:

	influence factor							
	1		0.75		0.7		0.65	
# tt per pt	av. pt occ.	av. quay occ.	av. pt occ.	av. quay occ.	av. pt occ.	av. quay occ.	av. pt occ.	av. quay occ.
5	0.6	0.5			0.81	0.74		
6	0.58	0.5	0.76	0.67	0.81	0.73	0.86	0.81
7	0.58	0.5			0.81	0.73		

table 5.11: Effect influence factor and number of tt's on quay and portainer occupancies in Oct. & Nov.

5.5.3.5 Yard operation measurements

The effect of admitting trucks to the yard without restrictions and of not giving tt's priority has been simulated in the model using the October/November input data. In these runs, the RTGC's in the stack served the first carrier in its queue, while maintaining a maximum of 15 transporters per stack.

This situation did not have significant influence on the turn-around time of the vessels in the case 6 tt's are assigned to a portainer (whichever influence factor is introduced, see table 5.12). This indicates that increased yard traffic does not constitute a limiting factor.

Note: As mentioned earlier and herewith confirmed by the model, the capacity of the RTGC's in the yard exceeds the capacity of the portainers at the quay.

The queue statistics (see appendix 5.3) confirm that the length of the truck queue at the gate of the container yard is zero in case trucks are admitted without restrictions. On the other hand, the length of the queues in the yard increases slightly whereas the length of the tt queue's at the portainers does hardly change.
Above indicates that the yard cranes have enough capacity to cope with such increased yard traffic.

Note: The smaller the number of tt's assigned to a portainer, the more the yard effects the turn-around time of the vessels: apparently portainers have to wait occasionally for tt's to return to the quay.

influence factor	1	0,7
ttpriority	12.6	18.61
nopriority	12.69	18.65

table 5.12: Turn-around times (hours) Oct. & Nov.

5.5.4 Conclusions

After verification and validation and based on the sensitivity analysis, it can be concluded that the model performs as expected, with an output comparable to the in 1997 recorded actual data.

In case the influence factor is introduced (thus taking into account the various factors delaying the terminal operations), the simulated turn-around times of the vessels approach the actual recorded turn-around times:

- for the months of January/February the best results are obtained with an influence factor between 0.65 and 0.7;
- for the months of October/November an influence factor of approximately 0.7 gives the correct results;
- for the whole of 1997, an influence factor of about 0.67 proves to result in the best fit.

In case there is no or little interference with the container operations, the "theoretical" portainer production in the model reaches about 20 box moves per hour per crane. Because of the unavoidable presence of delaying factors, this "theoretical" production rate is seldom reached in practice.

The quay occupancy in the model approaches the estimated actual occupancy rate reached in 1997 (according to the terminal management).

Assuming the portainers to operate with an influence factor of about 0.67, the following can be concluded:

- approximately 1.67 portainers operating per vessel on average;
- net operation time equals 68 % of the total berthing time of a vessel (assuming 2 hours mooring and deberthing time loss included in the turn-around time). In case this mooring and deberthing time loss is not included in the actual total berthing time (such as obtained from the terminal records), then the net operation time amounts to about 61 % of the total berthing time;
- quay occupancy of approximately 72 %.

The queue statistics (see appendix 5.3) show that in the model the average waiting time in the anchorage, is between 1 and 2.5 hours.

Note: Such waiting times have been expected since the model works following a "first come first serve" system, which does not allow re-scheduling of vessels (certain vessels being served earlier or later than others in the data list). Above values can not be verified because there is no information on the actual waiting times in the anchorage.

The results of the simulation runs confirm that the yard has a larger capacity to serve the carriers than the quay has to serve these tt's. This is understandable since an RTGC can make about 30 moves and a portainer approximately 21 moves per hour. This difference in servicing capacity further increases in case two portainers are servicing a vessel with two RTGC's operating in a corresponding stack.

The import and empty container stacks have, as earlier mentioned, a higher occupancy rate than the export stacks. This is explained by the fact that the import containers tend to have a longer dwell time than the export containers, while the numbers of export and import containers handled by the terminal are nearly the same.
From the animation it is observed that the empty container stack has a higher occupancy rate than the import stacks. This is due to the fact that 32 % of the imported containers are empty whereas there is only one stack against four stacks for import containers.

Note: As in practice these empty containers appear to be high in demand in Indonesia, it is assumed that the majority will be fetched as soon as possible. Judging from the 80 % occupancy rate recorded by the terminal management for this empty container stack, above assumption appears to be valid. Based on these assumptions, the simulation model resulted in occupancy rates of the import and empty stacks approaching the actual recorded rates.

6. Planning strategy

6.1 Introduction

A terminal can be seen as a link in a larger transport chain and as an interface between transport modes. To stay in business it is of utmost importance for a terminal to operate as efficiently as possible. A planning strategy is required to assure that the terminal operates efficiently, both for the present and for future operations.

In the previous chapter the model of the terminal has been verified and validated using data obtained from the terminal records of 1997. In this chapter the planning method applied at the Tanjung Perak terminal is discussed. The planning strategy followed by the terminal needs to be included in the model in order to be able to simulate the future situations as best as possible. Subsequently, the implementation of the planning strategy is explained when analysing the results of a number of simulation runs.

6.2 Present planning strategy followed by the terminal

In order to be able to meet the large demand for berthing time, a time-window is assigned to each vessel calling at the terminal. The planning of the time-windows is done according to the following schedule:

- before the 21st of each month, the shipping companies have to submit a list of vessels that they intend to call-in at the terminal during the next month. This list contains the vessels that are expected to be serviced, the approximate number of box moves to be made and the dates and times that the vessels are estimated to arrive at the terminal;
- on the 23rd of each month, a meeting is held between the terminal management and the shipping companies. During this meeting the time schedule for the next month is announced. This schedule is made by hand and provides information on the time-windows assigned to the vessels during the next month.

In case vessels arrive before their assigned time-window, the terminal authorities do not regard this as waiting time. In practice, vessels that arrive before or after their assigned time-window, can if necessary be re-scheduled. According to terminal management, vessels arriving within their assigned time-window are given priority over vessels that arrive outside their assigned time-windows.

6.3 Implementation of planning strategy in the simulation model

In this chapter, first all the model components and modelling assumptions are briefly described, followed by a discussion of the results from a number of computer simulation runs.
To check the validity of these results, the terminal model is subsequently run with the data obtained from the planning model. Finally the performance of the planning model is evaluated.

6.3.1 Component description and modelling assumptions

6.3.1.1 Model boundaries and general assumptions

The inter-arrival times of the vessels calling at the terminal are generated by PROSIM from an exponential distribution. The arrival of a vessel at the terminal is planned as follows:

- first the number of vessels that will call at the terminal, are simulated by PROSIM using an exponential distribution;
- subsequently, a time schedule is created in which these vessels are fitted in.

This final list of vessels with their inter-arrival times is used as an input file for the terminal model. In the planning model the "first come first serve" system is followed.

Note: This differs from the actual situation whereby the planning and fitting of vessels in the time schedule is done by hand, which allows certain vessels to be planned before or after other vessels if preferred from a planning point of view.

The purpose of this planning model is to create a list of vessels with a specific average inter-arrival time to serve as input for the terminal model.

The terminal model, in which the list of vessels with their inter-arrival times is used, also follows the "first come first serve" principle.

Note: In practise however it is possible that vessels arrive earlier or later than scheduled. Consequently, vessels may be serviced earlier or later than planned. This situation does not occur in the model. Similarly, in case the next vessel scheduled to be served is too long to fit at the quay, this quay place will in the model remain empty until another vessel has left. In practice however a smaller vessel may be chosen to moor at the quay instead of the longer vessel.

6.3.1.2 Model components

The planning model comprises the following components:

- a "*planningmaster*": to manage the final planning;
- a "*class component ship*";

Besides the "MAIN" component, taking care of the simulation run control, the model includes the "*Macro firstplanning*" responsible for generating the first list of vessels with their attributes.

6.3.1.3 The macro firstplanning

At the start of the model run, the "MAINMOD" module first activates the *"MACRO firstplanning"*. This macro creates a list of vessels, each with the following specific attributes: inter-arrival time, ship length and number of box moves to be made. All this data is stored in a file that is subsequently being used in the model.

- the inter-arrival time of a vessel is created from an exponential distribution having a specific mean. In 1997, this mean value was about 8.8 hours. The aim of the planning model is to create a list of vessels arriving at the terminal with an inter-arrival time of approximately 8.8 hours;

- the number of box moves to be made varies per vessel. The average number of box moves per vessel was about 393 in 1997 (see 5.3.1.3). Because this data fits no particular distribution, a number is picked from a list (see table 6.1 below). By generating a number between 1 and 20, the number of box moves to be made by that vessel is found. The values for the box moves in the list, are in accordance with the peaks in the histogram (see figure 6.1 below).

	# box moves per vessel		# box moves per vessel
1	160	11	400
2	205	12	415
3	295	13	415
4	295	14	460
5	370	15	460
6	385	16	460
7	385	17	475
8	400	18	535
9	400	19	535
10	400	20	550

table 6.1: Box moves list per vessel

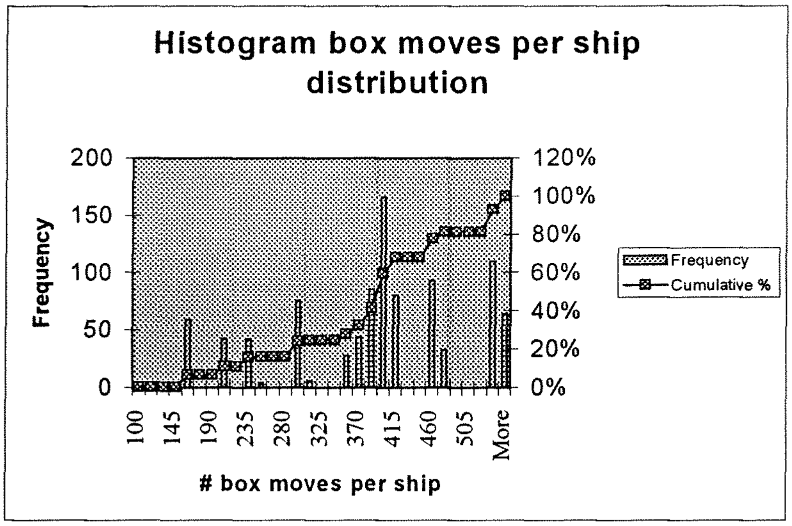


figure 6.1: Histogram & cumulative distribution box moves per vessel

- the average ship length in 1997 was approximately 153 meter (see 5.3.1.4). Since the ship length distribution during 1997 also does not really fit a specific distribution (see figure 6.2 below), a ship length is chosen from a list. This list is shown in the table 6.2 below and is created in accordance with the peaks in the histogram.

	Ship length		Ship length
1	114	11	166
2	120	12	166
3	123	13	169
4	123	14	172
5	129	15	175
6	132	16	178
7	141	17	178
8	144	18	184
9	147	19	184
10	150	20	205

table 6.2: vessel length list

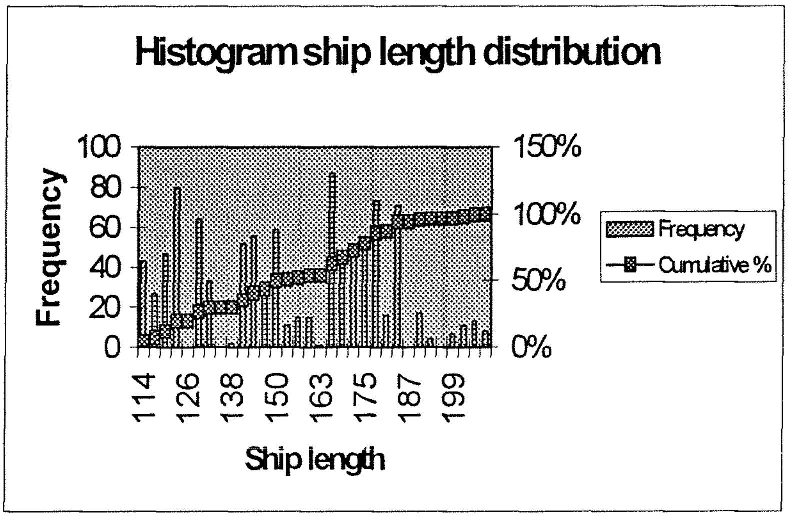


figure 6.2: Histogram & cumulative distribution vessel length

6.3.1.4 The component Planningmaster

This "*planningmaster*" module is activated in the "MAINMOD" after the first data list has been created by the "*Macro firstplanning*". The planningmaster is responsible for making the time schedule for the vessels in the same order in which they have been created previously.

The planningmaster assigns a berth place to a vessel provided that there is a free berth place and that the quay space is sufficient for the vessel (if not, the vessel waits until a vessel leaves and until there is enough quay space to moor the vessel). After a vessel is assigned a berth place, the vessel is activated in the module "*shipserviceplan*". After each vessel has been serviced, the data is updated and the quay situation examined for the next ship on the data list.

The vessels with their inter-arrival time, length and number of box moves to make, are stored in a second file, which file is used as input file for the terminal model.

The module shipserviceplan

On the basis of the number of box moves to be made, this module calculates the expected time requirement to service the vessels. After the vessel has been serviced, the vessel is terminated and the data regarding the number of ships at the quay and the length of free quay space updated.

6.3.2 Computer simulation results

First of all the results of the planning model are discussed.
In order to arrive at a data list of vessels with an average inter-arrival time of about 8.8 hours, the model has to generate vessels from an exponential distribution with a mean of about 7.9 hours.

The computer model has run for about 2683 hours or about 112 days. The number of vessels served in this period was 300 and the number of box moves made 117,365. In 1997, roughly the same number of vessels were served and the same number of box moves were made during this time-span.

Note: In case vessels are generated from an exponential distribution with a mean of 8.8 hours, the resulting average inter-arrival time of the vessels would exceed the projected 8.8 hours. This is due to the fact that in the model the "first come first serve" principle is followed, constituting a rather rigid planning method as compared to the flexible planning method used in practice.

The planning model creates a second data list, which is used as input for the terminal model. Hereafter, these simulated values for the inter-arrival times, ship lengths and number of box moves to be made per vessel, are compared with the actual 1997 data.

- The average inter-arrival time of the data in the data list reaches 8.9 hours (see table 6.3 below). This is achieved when a mean value of 7.9 hours is applied in the exponential distribution for the first generation of vessels. This distribution matches the distribution obtained from the terminal which was used as input in the terminal model (see appendix 6.1).

BestFit results:

Minimum=	0.00
Maximum=	37.44
Mode=	7.54
Mean=	8.89
Std Deviation=	6.80
Variance=	46.23

table 6.3: Inter-arrival time results

- On average the number of box moves per vessel reached 393 (see table 6.4 below) which is equal to the 1997 average. Also the distribution of these values is similar to the distribution of the 1997 values (see appendix 6.2).

BestFit results:

Minimum=	160.00
Maximum=	550.00
Mode=	400.00
Mean=	392.53
Std Deviation=	106.62
Variance=	11366.84

table 6.4: Box moves per vessel

- The ship length appears to have an average of about 153 meters (see table 6.5 below). Also the nature of the ship length distribution is similar to the distribution discussed in 5.3.1.4 (see appendix 6.3).

BestFit results:

Minimum=	114.00
Maximum=	205.00
Mode=	178.00
Mean=	152.74
Std Deviation=	25.70
Variance=	660.55

table 6.5: Vessel length

After the creation of this data list, this list was used as input file to run the terminal model. In the previous chapter it was concluded that the model resulted in a reasonable simulation of the actual 1997 terminal operations with an influence factor of approximately 0.68.
Two simulation runs have been made with above input file, with the value for the influence factor set at 0.7 and 0.68. The run results are shown in the table 6.6 below.

The total period covered by the terminal model runs corresponds with the time needed according to the planning model. Also the average turn-around time reaches nearly the same value as the 1997 value (terminal records). The average inter-arrival time of the vessels remains nearly the same as the value obtained from the planning model.
The average waiting time of the vessels in the anchorage is however slightly higher than the average waiting time resulting from the model in the previous chapter. The queue statistics (see appendix 6.4) show that the average waiting times in the other queue's on the terminal are similar to the average waiting times obtained during the runs in the previous chapter.

Run results:

Influence factor	0.7	0.68
Time	2872	2873
turn around time	18	18.6
Inter arrival time	8.9	8.9
portainer occ.	0.65	0.71
quay occ.	0.67	0.69

table 6.6: Simulation run results

6.3.3 Conclusions

Judging from the results obtained by the terminal model, using the data list created by the planning model, it can be concluded that the planning model performs correctly:

- the average turn-around time, number of box moves made and the time in which all the vessels are serviced, match the results obtained by the terminal model in case it is using the actual arrival times of the vessels together with the other parameters.
- the quay and portainer occupancies match the earlier found values.
- the queue statistics show that the situation in the queue's is also similar to the situation earlier simulated;
- the value of the influence factor in the terminal model also matches the value that was predicted to match the 1997 situation: this value should be approximately 0.68.

Considering the less flexible "FCFS" planning method used, in this terminal model a lower mean value has been chosen for the generation of inter-arrival times out of the exponential distribution.

7. Future terminal operations

7.1 Introduction

This chapter describes the development of a simulation model of the Tanjung Perak container terminal in Surabaya. Since the present international container terminal will be expanded with another similar international container terminal and an inter-island container terminal, two extra terminal models will be made. As these three terminals will be operating independently from each other, it is justified to create three independent simulation models. The future layout of the terminal is illustrated in figure 7.1 below.

In this chapter the two expected international containers terminals and the inter-island container terminal have been reviewed seperately. Since these three terminals will all have the same bridge connecting their quays with their terminal yard, a model of the connecting bridge has been made to simulate the increased traffic load on the bridge. This model is discussed at the end of this chapter.

Since there is no information regarding the expected container throughput in the future (especially considering the present economic crisis in Asia), the theoretical maximum container throughput (while maintaining a 70% quay occupancy) of the terminals will be used as a basis. By simulating the maximum traffic flow possible in the three terminals, the maximum load for the whole terminal is created.

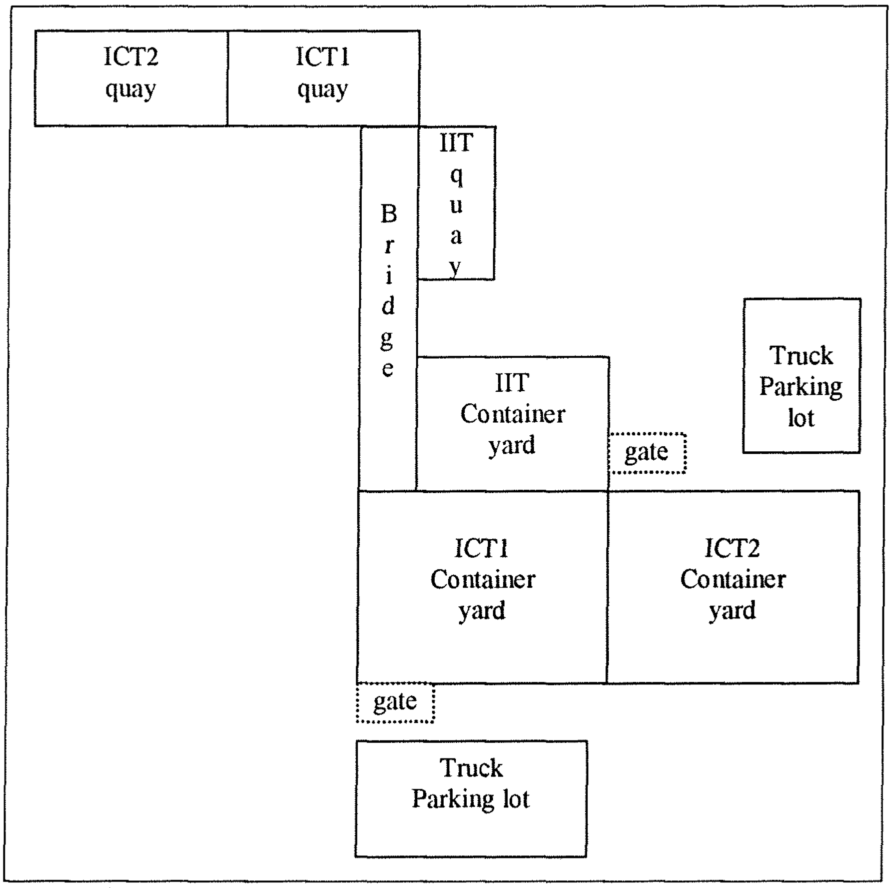


Figure 7.1: Overview future terminal

7.2 First International Container Terminal (ICT1)

7.2.1 Terminal operations 2005

The objective is to assess the theoretical maximum container throughput the international container terminal can deal with in an acceptable manner. In order to remain a viable option for the shipping companies, the terminal should in this case still be in the position to service the vessels within a certain time without increasing the anchorage waiting time of the vessels.

According to terminal management, it is not envisaged to change the present terminal equipment. This means that after the expansion, the same quay cranes, RTGC's and tractor-trailers will be used at this terminal. The container yard of the ICT1 terminal will not be extended and will thus keep its present capacity.

The international container terminal in 1997 already operated with a 70 % occupancy rate of the quay's and with the import and empty stacks already being filled for approximately 80 %. Without other measures, the anticipated increase in container throughput may therefore lead to longer turn-around times and longer waiting times of the vessels in the anchorage: such development is not acceptable for the terminal.

The following measures may be taken into consideration:

- According to the analysis in the previous chapters, about 10 % of the berthing time is lost due to mooring and deberthing activities of the vessels. Approximately 68 % of the berthing time left is effectively used for the actual loading and unloading activities. It is not known why the effective operational time is limited to about 68 % of the berthing time: numerous factors may in practice play a - unknown and unpredictable - role, such as equipment breakdown, personnel poor adherence to break and shift times and inefficient terminal processes.

- It is assumed that the efficiency of terminal operations can be improved to such an extent that the effective operational time reaches 80 % of the berthing time (while maintaining the 10 % allocation of time for mooring and deberthing activities);

- According to terminal management, the portainer production can be increased up to about 30 box moves per crane per hour by improving the overall efficiency of terminal operations;

- It is anticipated that the capacity of the container stacks in the yard will not be sufficient to handle the foreseen increased container throughput. The obvious solution to create a higher container storage capacity, would be to increase the number and/or the capacity of the stacks. However, this option is not being considered in the terminal plans for 2005: no extension of the terminal yard foreseen. For this "zero-growth scenario of the yard", a possible solution may be to reduce the average dwell time of the containers in the yard. In 1997 the maximum container dwell time for export containers was 8 days (average dwell time was four to five days), including the day preceding the arrival of a vessel at the terminal (during which day no more containers are delivered). For import containers the maximum dwell time was 11 days (average dwell time was about 8 days), including one day for customs procedures and administration. The dwell time of the containers in the yard can be reduced either by limiting the time span during which the containers can be delivered and picked-up, or by changing the arrival pattern of the majority of the trucks;

- Another possibility to improve the efficiency of the terminal operations, may be to increase the number of tt's operating per portainer. At present there are about 6 tt's operating per portainer.

7.2.2 Computer simulation model ICT1

In this section the theoretical maximum container throughput is determined, taking into account a number of factors. The results of sensitivity analysis are presented in relation to this maximum container capacity.

7.2.2.1 Modelling assumptions

In order to be able to predict the future operations of the terminal, the following assumptions have been made:

- Portainer production

It is assumed that the portainer production can be increased to 30 box moves per hour per portainer by improving the efficiency of the terminal operations. This implies that on average 40 seconds is required for the ptlift-on and -off actions and on average 80 seconds for the ptlift-cycle;

- Terminal efficiency

The model results for 1997 revealed that the net, effective servicing time of the vessels was about 68 % of the actual berthing time. This can be seen as a measure of the terminal efficiency. By increasing the efficiency of terminal operations, it is assumed that this percentage can be increased to 80 % (an influence factor value of 0.80 will be used);

- Quay occupancy

In 1997 the quay occupancy was 70 % on average. It is assumed that this percentage will be maintained in the future.

Note: A higher occupancy rate may in fact have unfavourable consequences for a terminal because little "room" is left for delays: for instance an unexpected, prolonged stay of a vessel at the quay, may delay berthing of a next vessel and thus result in higher waiting times in the anchorage.

- Anchorage waiting time

For 1997 the model was based on an average anchorage waiting time of about 2 hours. This average waiting time in the anchorage is not supposed to increase dramatically;

- Container distribution

It is assumed that the distribution of import and export containers per vessel and the distribution of 20 feet and 40 feet containers and the load of these containers remain the same as in 1997.

- Stack volume

In 1997 the import and empty stacks were on average nearly 80 % filled and the export stacks about 60 % filled. In order to leave room for peak periods the average occupancy rate of the stacks should not exceed 80 %. To achieve this with the anticipated increased container traffic, it is assumed that the majority of the import containers will be picked-up from the container yard within eight days (average container dwell time of about 5 or 6 days in stead of about 8 days). The majority of the export containers will be delivered in a time span of 6.5 days (average container dwell time of 4 to 5 days) before the arrival of the vessel (as in 1997). The distributions are shown in figures 7.2 and 7.3 below.

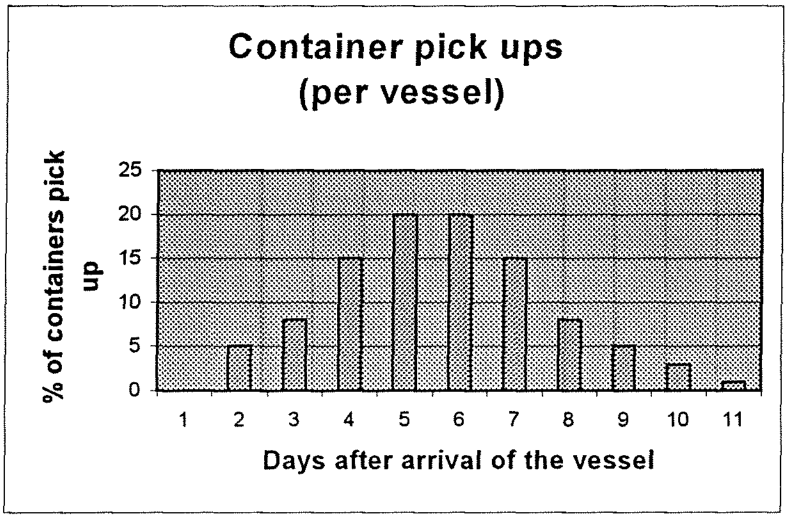


figure 7.2: New import container departure

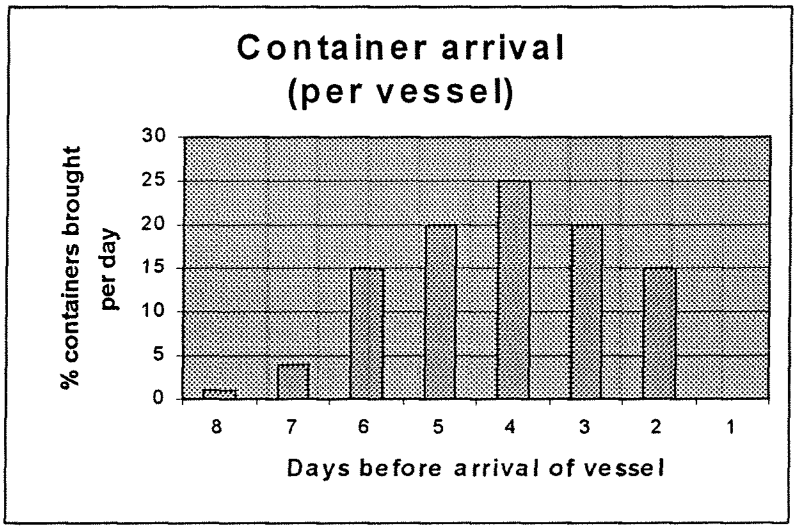


figure 7.3: Export container arrival

7.2.2.2 Sensitivity analysis and computer simulation results

The first step consisted of the estimation of maximum container throughput, based on above listed assumptions and further assuming on average 1.5 portainers servicing a vessel.
This resulted in a maximum throughput corresponding with about 590,000 box moves to be made with 1500 vessels having a turn-around time of about 12 hours.

Running the planning model with these predicted values and using the data lists created as input in the terminal model, gave the following results:

box moves/year	Vessels	quay occ.	t.a.t.	i.a.t.	av. waiting time anchorage (hours)
517,147	1316	61%	12.0	6.5	0.1
579,345	1474	72%	12.7	5.8	2.0
650,034	1654	85%	13.3	5.1	16.1

table 7.1: Results ICT1 run with 6 tt per portainer

Above listed results have been obtained by running the terminal model with 6 tt's per operating portainer. The results of investigations whereby larger numbers of tt's were assigned to a portainer, are summarised in table 7.2 below and can be reviewed in appendix 7.1.

tt's per pt	box moves/year	Vessels	quay occ.	t.a.t.	i.a.t.	av. waiting time anchorage (hours)
6	579,341	1474	72%	12.7	5.8	2.0
8	579,276	1474	71%	12.5	5.8	1.7
10	579,111	1474	71%	12.6	5.8	1.75

table 7.2: Results ICT1 run with different numbers of tt's per portainer

With six tt's assigned to a portainer approximately 580,000 box moves can be made in a year while the quay occupancy remains about 70%. Also the average anchorage waiting time of the vessels remains within the assumed limits.

The effect of increasing the number of tt's per portainer is apparently very limited: the average anchorage waiting times and the turn-around times are only slightly reduced when more tt's are assigned to the portainers. This implies that there are enough tt's driving between the portainers and the container yard and that the portainers seldom have to wait for tt's during the servicing of a vessel.

The situation in the container yard, however has drastically changed in comparison with the 1997 situation. The import and empty stacks are frequently filled 100 % in case the model is run with the new increased container throughput (580,000 box moves) and with the present truck arrival distributions for 1997 (this was to be expected since these stacks were already in 1997 about 80 % filled).
By changing the truck arrival distribution in accordance with the assumptions mentioned in 7.2.2.1, this problem is solved: with the new truck arrival distribution, the import and empty stacks are now approximately 80 % filled.

Since the number of empty import containers is quite high (about 32 % of the number of import containers), the majority of the empty import containers is expected to be picked-up within 4 instead of 8 days. Considering the high demand for empty containers (see chapter 5), it seems likely that these empties will, as was the case in the past, indeed be picked-up earlier than the full containers. The number of empty containers delivered for export is insignificant in this case.

7.2.2.3 Conclusions

The maximum number of box moves that can be handled by the terminal while meeting the earlier assumed requirements amounts up to 580,000. This number of box moves can be reached while maintaining a quay occupancy rate of 70 % (with an average anchorage waiting time of the vessels of about 2 hours). If the number of box moves is increased further, the turn-around times of the vessels, the quay occupancy and the average anchorage waiting time will also increase significantly.

Increasing the number of tt's assigned to a portainer, apparently does not have a significant effect on the container throughput: the turn-around times and the anchorage waiting times are only slightly reduced. The fact that the turn-around time of the vessels is hardly effected in case more tt's are assigned, indicates that the portainers are seldom waiting for tt's while servicing a vessel.

Note: From the queue statistics (see appendix 7.1) it can be concluded that the assignment of increasing numbers of tt's to a portainer, will result in higher tt waiting times in the queue's for the portainers as well as in higher tt waiting times in the queue's in the yard.

Since the container yard itself is not planned to be extended, the dwell time of the containers in the container yard need to be adjusted. The desired reduction of the dwell time is realised by changing the arrival and departure pattern of the containers as follows:

- majority of containers meant for import expected to be picked-up within 8 days with an average dwell time of about four to five days;
- majority of export containers expected to be delivered in a time-span of one day to 6.5 days before vessel arrival (similar to the 1997 situation);
- majority of empty import containers are expected to be picked-up earlier than the majority of the full import containers (already common practice at present).

The simulation runs confirmed that the stacks are able to cope with the increased container throughput in case the import containers are picked-up as proposed above,

Even at this increased container throughput, the capacity of the RTGC's in the container yard remains sufficient. This can also be concluded from the yard queue statistics: even with a more than 48 % increased container throughput, the waiting times in the queue's remain acceptable (see appendix 7.1).

7.3 Second International container Terminal (ICT2)

7.3.1 Terminal operations 2005

The new second international container terminal will operate in the same way as the present terminal. According to the terminal authorities, the new terminal will be a replica of the existing one: its quay will also have a length of 500 meters, starting at the end of the present terminal quay (see figure 7.1). The container yard is located east of the present terminal.

- The capacity and number of stacks are the same as in the ICT1 yard and 11 RTGC's will be operating in the yard;
- There will be four portainers operating at the quay, with - according to the terminal authorities - seven tt's operating per active portainer (in view of the longer distance between the quay and the yard);
- The entrance gate to the container yard of ICT1, will remain at the same location and will also serve as entrance for the trucks with the ICT2 as destination. The present truck parking lot will be used for trucks for both terminals: it is expected that the parking lot capacity will not be sufficient to accommodate all these trucks.

The terminals will be operating independently as they will be owned by different companies. Never the less it seems likely that the quay length and perhaps also the equipment, will be shared when convenient.

7.3.2 General assumptions

Since it is not known how large the container throughput will actually be in the future, it is assumed that this second terminal will reach approximately the same theoretical maximum container throughput as estimated for ICT1. Since the second terminal is in fact a replica of ICT1, above assumption appears justified.
Also the other assumptions made for ICT1, have been applied to this new terminal

7.3.3 Computer simulation model ICT2

7.3.3.1 Model boundaries and modelling assumptions

The computer model for ICT2 is exactly the same as the ICT1 model described in chapter 5. The model boundaries, model components and assumptions have been assumed to be the same and will therefore not be repeated in this chapter.

The only differences with the present terminal, concern the larger distances the tt's have to drive on the quay and in the terminal yard:
- on the ICT2 quay, the tt's have to drive an additional 500 meters to reach the portainers and the bridge;
- when leaving the bridge and driving to the container yard, the tt's will now first have to drive 300 meters past the container yard of ICT1 (the same applies when leaving the container yard);
- this additional travel distance applies also for the trucks directed to the ICT2 container yard.

7.3.3.2 Sensitivity analysis & computer simulation results

For the present terminal ICT1 it was estimated that the maximum number of box moves that can be made amounted to approximately 580,000 (see chapter 7.2). Since the second terminal is a replica of the present one (apart from the distances the tt's and trucks have to travel), the ICT2 terminal model has been checked with this same container throughput.

In this way, this new terminal can be compared with the ICT1 and the influence of the change in terminal layout investigated.

The terminal model has been run with varying values for the number of tt's assigned to a portainer in order to be able to assess the effects of the longer travel times on the operations at the quay. In particular the possible effect on the portainer production is investigated, as this would have a direct influence on the turn-around times of the vessels. The results of the simulation runs are summarised in table 7.3 below.

Tt's per pt	box moves/year	vessels	quay occ.	t.a.t.	i.a.t.	av. waiting time anchorage (hours)
6	578,578	1472	75%	13.3	5.8	3.2
7	579,111	1474	73%	12.9	5.8	2.1
8	578,973	1473	72%	12.7	5.8	1.9
10	579,111	1474	72%	12.7	5.8	1.8

table 7.3: Results ICT2 run

In comparison with the present terminal, the turn-around times, quay occupancy and anchorage waiting times of the ICT2 terminal have increased (i.e. in the case that in both terminals 6 tt's are assigned). This is to be expected since the tt's and trucks have to drive longer distances.
The simulation runs confirmed that by increasing the number of tt's, the turn-around times, quay occupancy and anchorage waiting times could be reduced.

The container yard operates in an acceptable manner when trucks arrive following the same distribution as applied for the present terminal. The queue statistics confirmed that the waiting times in the queues are similar to the waiting times in the container yard of the ICT1 (see appendix 7.2).

7.3.3.3 Conclusions

The new international container terminal can handle about the same number of containers as the present container terminal. The only differences concern the slightly larger turn-around times and anchorage waiting times that vessels have to face when call at the new terminal (as compared to the present terminal). This due to the fact that the tt's and trucks have to drive longer distances in the new terminal.

By assigning more tt's to the portainers these differences can be reduced.

Note: When more tt's are assigned to the portainers the waiting times of the tt's at the portainers will slightly increase for both terminals. The waiting times for tt's in the ICT2 queue's are slightly lower than the waiting times in the queue's of the present terminal, because the tt's of the new terminal have to spend more time on driving (see appendix 7.2).

The results indicate that the ICT2 will operate efficiently when 7 to 8 tt's are assigned to the portainers. The assignment of more tt's will hardly effect the turn-around times of the vessels (but will merely result in increased waiting times for the tt's, both in the queues and in the yard).

The tt waiting times in the ICT2 yard are comparable to those in the container yard of the present terminal, confirming that the container yard still has enough capacity to cope with the increased container flow (provided that the truck arrival distribution is changed as outlined). In case this distribution is not changed, then the capacity of the import stacks and empty stack will be insufficient.

7.4 Inter Island Terminal (IIT)

7.4.1 Introduction

At present an Inter-Island Terminal (IIT) is being constructed, scheduled to be operational in the year 2000. Its quay will be located alongside the bridge starting 450 meters before the ICT quay is reached (see figure 7.1). The container yard will be located north of the ICT yards as illustrated in figure 7.1.

The present IIT terminal, operating at a different location in the harbour, is servicing all kinds of coasters (coasters are vessels that are smaller than the vessels arriving at the ICT). These vessels travel between the larger harbours and smaller harbours that cannot accommodate the larger vessels.
At the present IIT terminal 30 % of the vessels have a load smaller than 100 TEU. The new terminal will only service vessels with a load larger than 100 TEU (vessels with a smaller load will continue to be serviced at the old terminal).

Note: The relocation of this inter island terminal for larger vessels to the vicinity of the international container terminals, may have a number of advantages. The expected interaction between these terminals on the same terminal grounds, may create time saving opportunities (as compared to the present situation in which trucks have to travel between the IIT and the ICT). This aspect is beyond the scope of this research and will therefore not be investigated further.

In the next paragraphs first the terminal is described, followed by an explanation of the computer simulation program (with its assumptions and components) and the results of the sensitivity analysis and the conclusions.

7.4.2 2000 situation

The new IIT is expected to be in service by the year 2000. Based on a report on this future terminal (December 1997) the new IIT lay-out can be summarised as follows:

Quay
The 450 meters long quay will be located as depicted in figure 7.1. The entrance to this quay for the tt's will be situated at the southern end of the quay, 450 meters from the bridge connecting the IIT with the ICT quay (see figure 7.4 below).

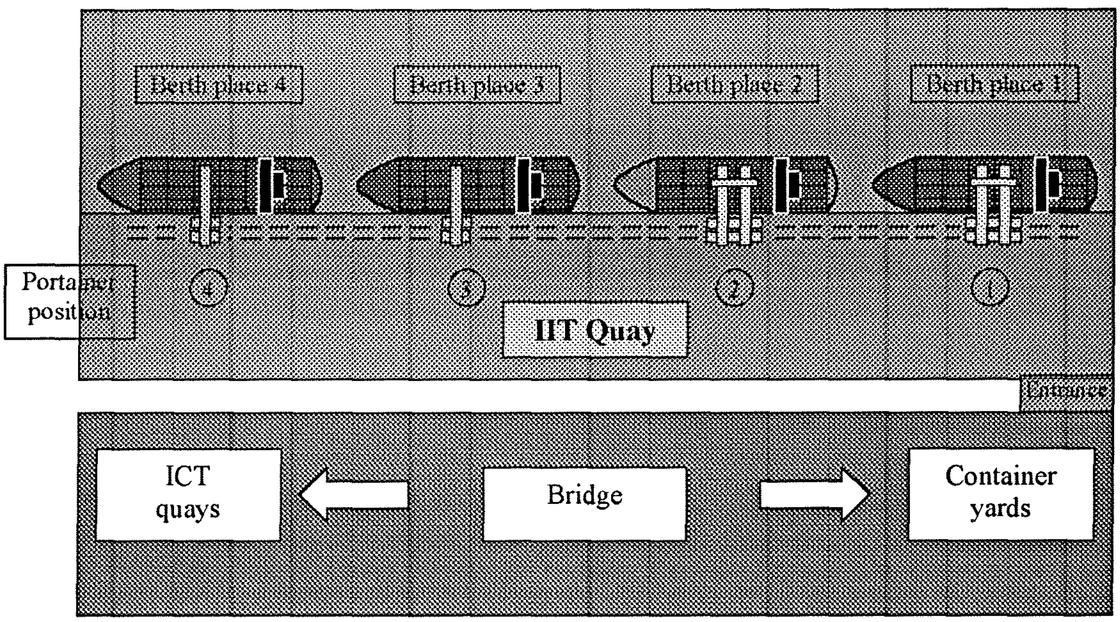


figure 7.4: Quay & portainers

Portainers
Initially the IIT berths will be equipped with two portainers having a production rate of about 16 box moves per hour. The other berths will be serviced by shiptainers in the case of self unloading and loading vessels. Otherwise the vessels will be serviced by non-dedicated quay cranes. The quay cranes presently being used in the active IIT, have a production rate of approximately 9 box moves per hour.

Note: Initially the aim was to have four portainers available with a production rate of about 16 box moves per hour (partly because of the economic crisis, this plan has in the mean time been changed).

Container yard
The container yard will be located on a newly reclaimed area of 50,000 m^2 north of the ICT container yard (see figure 7.1). The yard will accommodate six stacks, each 50x20 ft long and 7x8 ft wide (see figure 7.5 below). The containers will be stacked 4 tiers high. This present area does not allow further expansion of the stacks: the only possibility to increase the container yard would be to reclaim more ground to the north (seaside direction).
The IIT container yard will be operating independently from the international container terminals.

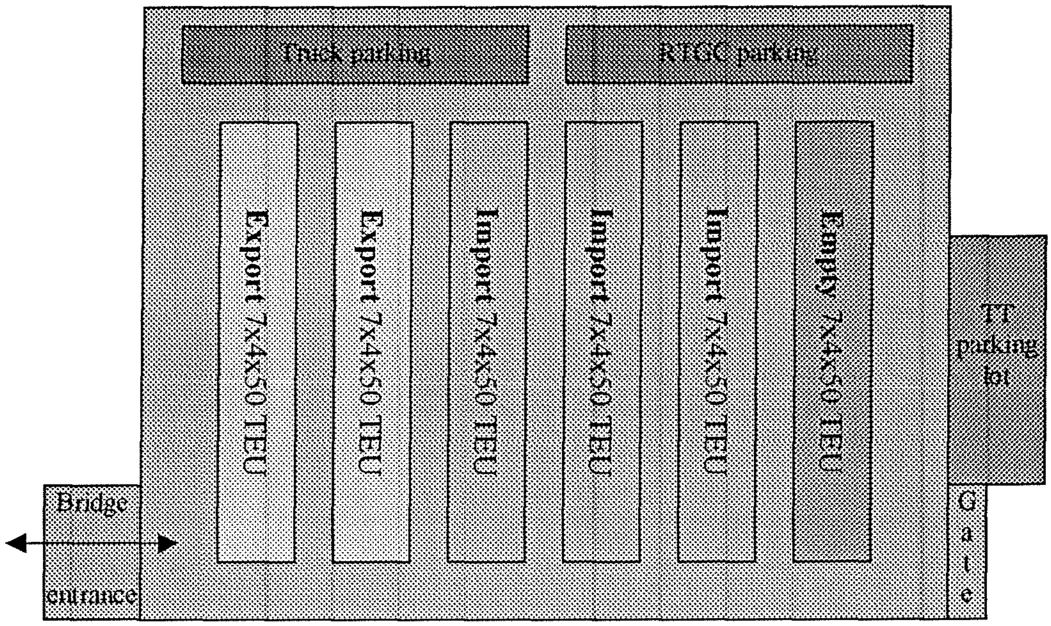


figure 7.5: Container yard & stacks

RTGC's
According to the terminal management, six RTGC's will be operating in the container yard.

Tractor-trailers
It is expected that approximately 5 tractor-trailers will be operating per active portainer or quay crane (to be further investigated). The actually required number will depend on the portainer production and the distance between the container yard and the quay.

Trucks
The entrance gate for the trucks to the container yard is located on the south-eastern end of the container yard (see figure 7.1). This gate entrance is connected to a road passing the container yard at the East Side of the terminal. The IIT truck parking lot will be situated outside the gate and will thus be separated from the parking lot for the trucks that arrive for the ICT's.

Vessels
According to above mentioned report, only vessels with a load exceeding 100 TEU will be serviced by this new terminal. It has been estimated that by the year 2000 approximately 1407 vessels will be serviced at this terminal. Assuming 190 box moves per vessel, this would correspond with about 270,000 box moves to be made in 2000. This estimation was based on 4 portainers operating at the quay each with a production rate as mentioned above.

Note: In case only 2 portainers plus 2 quay cranes (with a lower production rate) will be available, the total number of possible box moves per year might also turn out to be lower.

Container distribution
Unfortunately the container distribution is not clear: the report indicates that 90 % of the containers moved are 20 ft containers and 10 % 40 ft containers. There is no information regarding the full or empty container distribution nor about the distribution of import and export containers.
According to the report, the export containers are scheduled to arrive at the container yard between 7 and 2 days before the vessel arrives at the terminal. The import containers are expected to be picked-up inside a time span of 10 days after vessel arrival.
The containers arriving at the IIT by truck and leaving the terminal by vessel are called export containers (even though their destination is further inland). The import containers are containers that arrive at the terminal by vessel and leave the terminal by truck (presumably taken to an international container terminal for further transport).

7.4.3 Computer simulation model IIT

In this section the model components, model boundaries and modelling assumptions are described, followed by a discussion of the results obtained from a number of simulation runs checking the model on its sensitivity.

Note: It has to be noted that little is known regarding the expected performance of this terminal. The model has therefore been based on the models already made for the ICT's, be it with a number of adjustments.

A theoretical maximum container throughput has been simulated in order to create the maximum load on the total terminal (including the ICT's). Since the terminal was originally planned to be equipped with four portainers of the same capacity, the maximum load feasible with these four portainers operating at the quay, will also be examined.

7.4.3.1 Component description and modelling assumptions

7.4.3.1.1 Model boundaries and general assumptions

The IIT model has the same model boundaries as the ICT models:

- seaside: the vessels are assumed to arrive at the anchorage in the harbour 8 days after having been generated at sea. During this period the vessels arrive at the terminal passing through the approach channel (which in practice may cause a delay). In the model it is assumed that pilot and tugboat assistance is available when required and there is no restriction as far as the number of vessels waiting in the anchorage is concerned.
- landside: at the gate of the terminal, constituting the land side boundary, the trucks are generated to arrive and wait in the parking lot until further notice. The trucks leave the terminal through the same gate (and at that time are terminated).

Since the construction of this terminal is still ongoing, information is lacking with respect to the organisation of the future terminal operations. Consequently, a number of assumptions had to be made (see hereafter).

Note: For the sake of simplicity and considering the fact that the basic terminal operations will remain the same, the IIT computer model could to a large extent be based on the already available models for the international container terminals. Although incorporating a number of adjustments, basically the same model has been used: various modules have been duplicated with minor adjustments.

Portainers

Because it is not clear what the future organisation of this terminal will be, it has been assumed that the IIT portainers will always stay at one and the same berth place (even if there is no vessel to be served at that berth). Two portainers, with a production rate of about 16 box moves per hour each, will be positioned at the first two berths: vessels are expected to be moored by preference at these first two berth places (provided they are free). The other two berth places will have a quay crane with a production rate of about 9 box moves per hour. It is assumed that one quay crane will service a vessel, even if such a vessel has its own shiptainers (in practice a combination of these cranes might be possible).
For the simulation of the terminal equipped with four similar portainers (each with a production rate of about 16 box moves per hour), is has been assumed that also these portainers will stay at their berth (the portainers will not be re-positioned to another berth). Although this constitutes a major simplification, this assumption has been maintained throughout the model, since it is not known what strategy will be followed in the future.
For both scenarios above, the theoretical maximum number of box moves has been simulated while assuming a maximum quay occupation of 70 %.

RTGC's

It has been assumed that each RTGC in the container yard is assigned to a particular stack and that an RTGC will not shift between stacks in case it is passive. In practice however, a passive RTGC might be shifted to another stack in case there is heavy traffic in that stack.
As it is not known what type of yard cranes are scheduled to be used, it has been assumed that the IIT yard will be equipped with the same RTGC's as the ICT container yard.

Container distribution

As information is lacking regarding the distribution of the containers expected to be passing through the IIT terminal, it has been assumed that 50 % of the containers are for export and 50 % for import (this approaches the distribution of container passing through the ICT's). The above referred to report indicates that 90 % of the containers might be 20 ft containers and 10 % 40 ft containers. As the percentage of full and empty containers is not known, it has been assumed that approximately the same distribution will apply as for the ICT's:

- export containers (to be loaded onto vessels and transported further inland): about 70 % full and 30 % empty containers;
- import containers (to be loaded onto trucks): about 95 % full and 5 % empty containers.

The export containers are assumed to arrive in a time-span of 7 days before arrival of the vessel, whereby all containers are expected to be in the stacks one day before arrival. The import containers are assumed to be picked-up from the container yard 10 days after arrival of the vessel (including one day for custom and handling procedures).

Taking into consideration the predicted 270,000 box moves per year (ref. December report) and the yard capacity, a likely truck arrival distribution has been assumed similar to the truck arrival distribution for the ICT1 in 1997 (for the exact distribution, see the figures in chapter 5.3.7).
Because of lack of information, also the following assumptions have been made:

- no container traffic between the stacks in the IIT yard and the stacks in the ICT yards.
- negligible container movements between the stacks and the container freight station;
- reefer containers and off-size containers neglected.

All containers arriving at the terminal by vessel will leave the terminal by truck and all containers delivered by trucks will leave the terminal by vessel.

Note: One of the objectives of locating these terminals next to each other is to allow faster container movements between the terminals without relying on trucks driving between these terminals. As there is no information about such inter-terminal container movement, this option has not been taking into consideration.

Influence factor

The waiting times of vessels in the anchorage, equipment breakdowns, bad weather, inefficient operations, strikes and other factors that may delay terminal operations, can not be specified: therefore, following the example of the ICT models an influence factor has been introduced in the model to simulate these effects.
It is assumed that also at this terminal 10 % of the berthing time will be required for mooring and deberthing activities and that the vessels will be completely serviced in 80 % of the remaining berthing time (corresponding with an influence factor of 0.80 in the model runs).

Note: This value represents a rather high efficiency level of terminal operations, enabling to simulate a maximum traffic load in the terminal and on the bridge: a high portainer production means that more tt's can be serviced and thus that the tt's will be driving more frequently between the container yard and the quay.

Traffic procedures

The model includes a traffic master at the quay and a traffic master in the yard (similar to the ICT's). In addition three traffic lights are used, placed at the same locations as in the ICT's, to ensure a minimum distance between the tt's driving on the bridge.
Also in practice there will have to be some sort of traffic master to direct IIT related traffic to and from the bridge considering the possible interference with the traffic over the bridge coming from and going to the ICT quay: this traffic master may in fact be a traffic light.

Note: In case a tt coming from the yard wants to drive onto the IIT quay, it will have to cross the traffic lane used by the tt's coming from the ICT quay since road traffic in Indonesia drive on the left hand side. In the model this tt traffic coming from the ICT quay has in first instance been neglected: all the different traffic flows on the bridge will be investigated further in the separate bridge model.

Planning model

The planning model used to create a data list of vessels calling at the terminal, will basically be the same as used for the ICT's. Because in this case little is known about the actual distribution of vessel lengths and number of boxes per vessel, these distributions have been assumed to be normally distributed. Because the terminal will be operating with different portainers, also the distributions of the inter-arrival times and turn-around times of the vessels are not known. Consequently, the planning model will create a list of vessels with inter-arrival times out of an exponential distribution.

Note: Since basically the same model has been used as for the ICT's, for further background information reference is made to chapter 5.

For planning the IIT operations, the following two alterations have been made in the planning model for the ICT's (the related results are discussed in 7.4.3.3):
The number of box moves to be made per vessel, is generated by PROSIM out of a normal distribution with a mean of 190 and a deviation of 32;
The vessel length is also generated out of a normal distribution with a mean of 75 and a deviation of 15.

7.4.3.1.2 Model components

In the IIT model the same components and class components have been used as in the ICT model (reference is made to chapter 5.5.1.2).
The component processes that have been changed are explained hereafter.

Component process	Role
"berthmaster"	to manage berth distribution
"portainermaster"	to instruct portainers
"portainer"	to manage the portainer actions
"tt"	to manage the tractor-trailers
"quay trafficmaster"	to direct the tt's at the quay
"RTGC"	to manage the RTGC actions

In the "MAIN" module some values of parameters have been changed, but most of the module remained the same.

The majority of the MACRO's have also been used with the exception of the "Macro portainerstart" responsible for the portainer receiving the correct instructions.

7.4.3.1.3 The component berthmaster

The IIT quay has four berths to assign vessels to. The actions of the "*berthmaster*" are the same as in the ICT terminal model and it is activated by the same components.
The berthmaster is activated when a vessel arrives at the anchorage and when a vessel leaves the quay. The berthmaster checks whether there are vessels in the anchorage and then investigates the situation at the quay: in case a place is free and the available quay space is sufficient for the first vessel at the anchorage, this berth place is assigned to that vessel. Also for the IIT model the "first come first served" (FCFS) principle is followed. In case the available space is not sufficient, or if there is not a free place, the berthmaster becomes passive until re-activated.

7.4.3.1.4 The component portainermaster

The portainermaster is responsible for assigning the portainers to the vessels moored at the quay. As portainers have been assumed not to change berths (each berth having its own portainer), the "*ptmaster*" only has to check whether there is a vessel requesting for a portainer.

- In case there is another vessel, the ptmaster activates the portainer and adds the portainer to a list in which the portainer requests the "*ttmaster*" for tractor-trailers.
- In case there is no vessel requesting for a portainer, the "*ptmaster*" becomes passive until he is activated by a vessel.

7.4.3.1.5 The class component portainer

The component portainer is responsible for servicing the vessel and is activated by the "*ptmaster*". Since the IIT unloading and loading processes are similar to those described in chapter 5.5.1.7 for the ICT's, they have not been repeated hereafter. The only difference in the portainer process is that the portainers do not change position along the quay: consequently the portainer can directly start unloading a vessel and once finished proceed with loading the vessel. After servicing a vessel the portainer becomes passive until further notice.
The "*Macro quayload*" keeps track of the loaded and unloaded containers. This macro also determines which containers are to be unloaded and to which place these containers are to be delivered in the container yard (this information is handed-over to the tt driver).

7.4.3.1.6 The class component tractor-trailer (tt)

The tractor-trailers are activated by the "*ttmaster*". It is assumed that the same tt's will be used in the IIT as in the ICT. The difference with the tt process in the ICT model is that the location of the container yard and the quay are slightly different. On the whole this process is similar. For questions the reader is referred to 5.5.1.9.

7.4.3.1.7 The component trafficmaster of the quay

The trafficmaster of the quay has the same responsibility as the trafficmaster of the ICT quay's, namely directing tt's arriving at the quay where to their destination:

- in the case of the IIT, a loaded tt has only one portainer to go to;
- an empty tt is directed by the trafficmaster to the portainer that has last received a tt for unloading. If its queue is too long the tt will be sent to the container yard to pick-up a load.

The trafficmaster is activated by the tt's arriving at the quay and the tt's on that are on their way to the bridge.

7.4.3.1.8 The class component RTGC (yard crane)

The RTGC is activated by tt's or trucks waiting in a queue to be served by the RTGC. The RTGC is responsible for lifting containers onto and from the stacks and for loading respectively unloading the transporters. It is assumed that the same RTGC's will be used as in the ICT yard. For further information on the RTGC actions the reader is therefore referred to 5.5.1.13. The only difference relates to the assumption that the RTGC will remain in its stack and that there will always be just one RTGC in a stack (and not two as was the case in the ICT RTGC process).

7.4.3.2 Sensitivity analysis & computer simulation results

The inter island terminal has been modelled for two situations. These two situations which have been run and checked, can be described as follows:

- situation one (a): two portainers with a production rate of 16.4 box moves per hour each supplemented by two quay cranes with a lower production rate (9 box moves per hour);
- situation two (b): four identical portainers, each with a production rate of 16.4 box moves per hour.

Both terminal situations have been run with a theoretical container throughput. These models require different data lists of vessels that are created by a planning model.

a) Terminal situation with two different types of portainers active at the quay

When assuming the cranes at the quay operate with an average of about 14 box moves per hour a data list of vessels can be created with an average inter arrival time of 7 hours. With this vessel distribution a total of about 230,000 box moves can be made by the terminal.

The results obtained from a number of runs of the IIT model with above container throughput, have been summarised in the table below. Other runs indicate that a higher container throughput will result in an increased quay occupancy as well as in longer turn around-times and anchorage waiting times (if the container throughput is decreased, the opposite occurs).

The terminal is able to handle a total number of about 230,000 box moves per year while the quay occupancy remains about 70%. This constitutes the theoretical maximum number of box moves that can be handled by the terminal (corresponding with about 1200 vessels calling at the terminal).

tt's per pt	box moves/year	vessels	quay occ.	t.a.t. (hours)	i.a.t. (hours)	av. waiting time anchorage (hours)
5	231,070	1216	70%	20.0	7.0	1.7
8	231,045	1216	70%	19.9	7.0	1.6

table 7.4: Run results IIT with two different types of portainers

From above table it appears that the increase of tt's per portainer does not effect the servicing of the vessels. Also the queue statistics (see appendix 7.3) show that the waiting times of the tt's at a portainer will only increase when more tt's are activated. This indicates that the quay cranes production rate is the main factor determining the turn-around times of the vessels.

b) Terminal situation with four identical portainers

In this case the IIT operates with four identical portainers having a higher production rate than the combination of different portainers and cranes above. Consequently, the maximum container throughput in this case proves to be higher, with corresponding lower turn-around times and inter-arrival times of the vessels (see the results in the table below).

When assuming a maximum quay occupancy of 70%, the maximum number of box moves that this terminal can handle per year, can be estimated at about 285,000 (corresponding with about 1500 vessels calling at the terminal). This estimation was used to predict the inter-arrival time of the vessels required as input for the planning model creating the data lists of vessels planning to call at the terminal.

The results of the runs of the terminal model using above data list have been summarised in table 7.5 below.

tt's per pt	box moves/year	vessels	quay occ.	t.a.t. (hours)	i.a.t. (hours)	av. waiting time anchorage (hours)
5	282,844	1489	70%	16.2	5.7	0.45
8	282,890	1489	69%	16.2	5.7	0.4

table 7.5: Run results IIT with four identical portainers

Other runs indicate that the effects of increasing the number of vessels calling at the terminal are the same as for the previous terminal situation. In case the container throughput is increased, the quay occupancy and the anchorage waiting times also increase.

Also in this situation it appears that an increase of the number of tt's hardly effects the terminal operations but merely results in increased waiting lines for the tt's at the quay and in the yard (see appendix 7.4). The same was concluded for the previous situation.

7.4.3.3 Conclusions

The IIT model has been simulated for two different situations.
The first simulation of the terminal is assumed with two different types of quay cranes active at the quay (in accordance with the present planning of the IIT when operational in 2000).
For the second simulation four portainers with the same capacity have been assumed (each having the same capacity). This corresponds with the original terminal planning, as described in the terminal report. Since this situation is likely to be the next step in upgrading the IIT in the future (resulting in a larger container throughput per year), its simulation has also been included.

Based on a visual verification of the model runs (by means of animation and the TRACE function) it can be concluded that the IIT models are operating according to the assumptions made.
Reviewing the queue statistics and the simulation run outputs, the following can be concluded:

- the terminal situation with two different types of portainers can handle a total number of boxes of about 230,000 per year, whereas the terminal with four identical portainers can handle about 280,000. These are the theoretical maximum number of box moves the terminal can handle in an efficient manner considering the before made assumptions. The queue statistics (see appendixes 7.3 and 7.4) show that the waiting times in the queue's are acceptable;
- the capacity of the terminal yard is also sufficient for the in the model assumed dwell times of the containers;
- also in this terminal the portainers constitute the main factor determining the container throughput per year;
- increasing the number of tt's in the model does not have a significant effect on the total number of containers that can be handled;
- the truck queue's outside the terminal grounds are shorter than the truck queue's for the ICT's (because the number of trucks arriving per vessel are much smaller).

It should be noted that this IIT model constitutes a simplified representation of the IIT operations that may be realised in the future. This simplification was made due to present lack of information regarding the future operations of this terminal.

7.5 Bridge situation

7.5.1 Introduction present bridge use

With one international container terminal in operation, the present bridge connecting the ICT quay with the main land, is used as a two-way road. According to terminal management and on-site investigations, this bridge is wide enough to accommodate a third lane.
With the expansion of the terminal with another ICT and an inter-island terminal, the use of the bridge needs to be reconsidered. For instance, as the IIT will be located along the bridge, it can not be avoided that tractor-trailers with different quay destinations will have to cross each others traffic lanes in order to deliver or pick-up their loads.

In this chapter a possible traffic strategy is discussed based on a computer simulation model for the bridge (with its assumptions and components). Following a description of the model, the results of the sensitivity analysis are given and conclusions formulated.

7.5.2 Future bridge use

Following the scheduled expansion of the terminal, the situation on the bridge will change considerably: the number of active tt's at the terminal at a given time will increase from about 24 tt's at present to about 72 tt's in the future (i.e. assuming that the number of operational portainers and tt's in the new set-up will be in accordance to the planning described in chapter 7).

This increased traffic flow on the bridge, compounded by the fact that different traffic flows will have to cross each other's traffic lanes, makes it necessary to review the bridge situation (see figure 7.6 below).

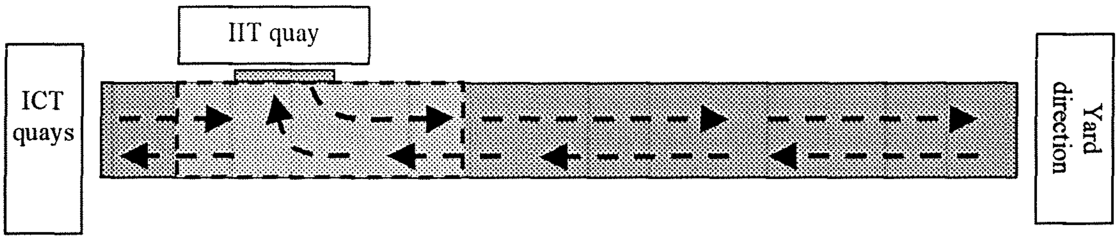


figure 7.6: Overview bridge traffic flows

As stated above, the present bridge is wide enough to accommodate three traffic lanes. Since the planning of the future terminal does not include a strategy concerning the different traffic flows on the bridge, a likely strategy has been developed hereafter.

For the bridge situation, investigated by means of a computer model, the following assumptions have been followed:

- the tt's drive on the left hand side of the road (as everywhere in Indonesia);
- it is assumed that the tt's driving to and from the ICT quay's, have priority over the tt's driving to and from the IIT quay (the ICT's are thus given priority over the IIT);
- at the point where the tt's driving to the IIT quay leave the bridge traffic lane to enter the quay, they will have to cross the lane with traffic in direction of the international terminal yards. Before crossing this lane, these IIT bound tt's will be driving in the central lane without disrupting traffic heading for the ICT quay's. If necessary these tt's will have to wait before they can cross the traffic lane without influencing the other traffic;

- it is assumed that the tt's will keep a certain minimum distance between each other on the bridge and that the maximum driving speed (30 km/hour) is respected (in this case this minimum distance has the same value as the driving speed, or 30 meters).

Figure 7.7 below presents an overview of the bridge situation, illustrating the use of the different traffic lanes.

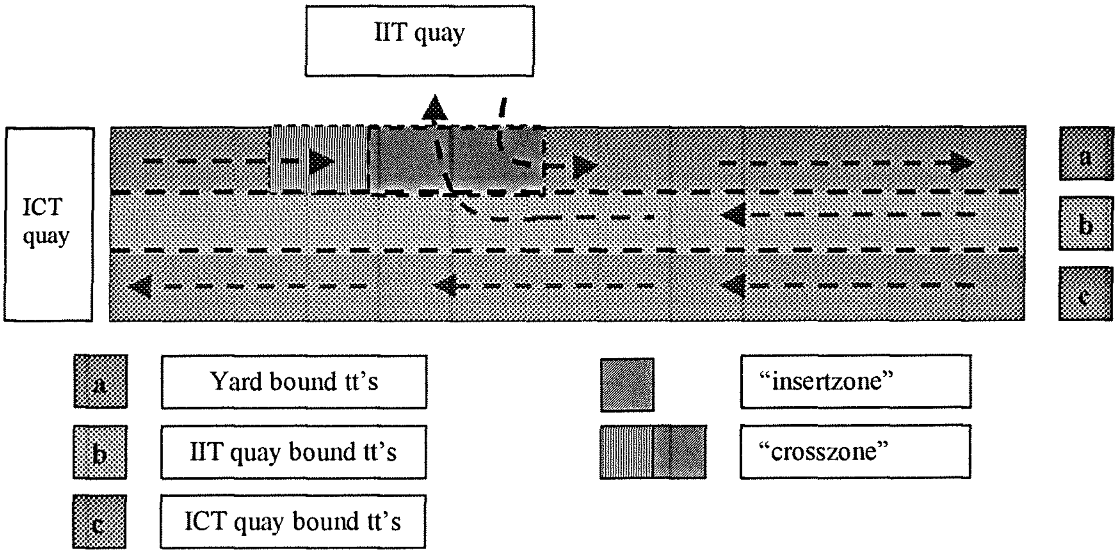


figure 7.7: Overview usage of traffic lanes on the bridge

Note: Above situation represents just one of the many possible use's of the bridge and traffic strategies that can be followed: in practice, crossing of the traffic lanes and driving onto the bridge will be a flexible process largely depending on the experience and insight of the tt driver's.

7.5.3 Computer simulation model of the bridge component

7.5.3.1 Description and modelling assumptions

In this section the model boundaries, various assumptions and model components are further explained.

7.5.3.1.1 Model boundaries and general assumptions

The model has two boundaries: one at the bridge entrance from the yard-side and one at the bridge entrance from the quayside:

- the tt's with the quay as destination are created at the yard-side entrance and terminated once they leave the bridge and arrive at the quay;
- the quay-side boundary actually consists of two bridge entrances: one for the ICT quays and one for the IIT quay. Also in the model a distinction is made between these two quays as far as the driving process of the tt's is concerned. The tt's with yard destination are created at the quayside and subsequently terminated once they arrive at the yard.

The model generates tt's arriving at the bridge entrance with a certain inter-arrival time. It has been assumed that the inter-arrival time of the tt's follows an exponential distribution.

Note: This assumption appears justified considering the distribution of the inter-arrival times of the tt's driving onto the bridge in the ICT models and the IIT model.

The program BestFit has been used to check the distribution of the inter-arrival times. In this program only data sets of 3000 values can be checked and therefore a number of data sets have been checked. The results of checking these data sets with the help of BestFit, indicate that an exponential distribution gives a reasonable reflection of the distribution generated by the terminal models (see appendix 7.4).

Note: It appears that also the Gamma shaped distribution gives a good representation of the distribution. When a Gamma shaped distribution is used, PROSIM calculates a value for "k": in case this "k" value is less than 1 (which appears to be the case), PROSIM advises the user to apply an exponential distribution. Consequently, the traffic flows in the bridge model have been generated by PROSIM from an exponential distribution.

The traffic with the IIT quay as destination and the traffic coming from the IIT quay, are in the model managed by means of traffic lights.

Note: In practice it is possible that a "human traffic officer" will carry out this task or that the tt drivers themselves will make the decision whether or not they can safely leave or enter the bridge.

There are also traffic lights for the tt's driving onto the bridge from the terminal yard and from the ICT quays. This is done to ensure that the tt's maintain a minimum distance between each other when driving on the bridge (the value of the minimum distance corresponding with the value of the maximum speed on the bridge). In practice this is already the case when the tt's drive onto the bridge.

It is further assumed that the tt's, after having been given the green light at the traffic light, accelerate until they reach a speed of 30 km per hour and that they maintain this speed until leaving the bridge. Furthermore it is assumed that a tt once it has reached its maximum speed on the bridge, will still have the predetermined minimum distance of 30 meters (see chapter 7.5.2) between itself and the following tt. All tt's are assumed to drive at the same speed (thus not allowing overtaking on the bridge).
The tt's that have the IIT quay as destination, drive at the maximum speed until they arrive at the quay entrance. At this point the tt waits in a queue at the traffic light until given the green light to accelerate and cross the bridge onto the quay. A tt coming from the IIT quay, accelerates onto the bridge after having been given the green light from the traffic light at the bridge entrance.

Note: It is possible that a tt gets mechanical failure while driving on the bridge: since the bridge is wide enough for three traffic lanes, this will not constitute a major obstacle. On the other hand, an accident on the bridge involving a number of tt's, could create an obstacle on the bridge and thus result in portainers waiting at the quay for tt's. In the bridge model this situation has been neglected.

7.5.3.1.2 Model components

The bridgemodel is a totally different model as compared to the described models of the ICT's and the IIT in the previous chapters. The components used in the bridge model are listed below and further explained in the following paragraphs.

Component	Role
"*ttgenerator*"	'class' component creating the tractor-trailers with their inter-arrival times;
"*tractor-trailer*" (tt)	'class' component representing the tractor-trailers used to transport containers between the container yards and the quays (they make up the only traffic on the bridge);
"*crosslight*"	traffic light to manage the traffic crossing the traffic lane to reach the IIT quay;
"*insertlight*"	traffic light to manage the traffic driving onto the bridge coming from the IIT quay;
"*yardlight*"	traffic light to manage the traffic driving onto the bridge coming from the yard;
"*ictquaylight*"	traffic light to manage the traffic coming from the ICT quays with destination yard.

The module MAINMOD is also present and takes care of the simulation run control. The actions of the tt's on the bridge are divided in two processes, each in a different module: one process describing the traffic coming from the yard with destination quay and the other process describing the traffic in the opposite direction. MACRO's have not been used in this model.

In order to run, the model requires input data as follows:
- first an input file is created containing the mean values of the exponential distribution out of which the inter-arrival times of the tt's are generated;
- before the model starts its run, it requires information from the user concerning the maximum driving speed on the bridge and the traffic light follow-up time.

7.5.3.1.3 The class component ttgenerator

The class component "*ttgenerator*" is created and activated in the module MAINMOD. The following six ttgenerators are created and activated, whereby each ttgenerator creates tt's with different origin:

- "y-ict1": this generator generates tt's coming from the yard-side which have the ICT1 quay as destination;
- "q-ict1": this generator generates tt's coming from the ICT1 quay-side which have the yard as destination;
- "y-ict2": this generator generates tt's from the yard-side which have the ICT2 quay as destination;
- "q-ict2":this generator generates tt's from the ICT2 quay heading for the yard;
- "y-iit": this generator generates tt's coming from the yard which have the IIT quay as destination;
- "q-iit": this generator generates tt's from the IIT quay which have the yard as destination.

The ttgenerators generate tractor-trailers, according to inter-arrival times obtained from an exponential distribution having a certain mean value provided by the user.
The generated tt's are activated by the ttgenerator in either the "*ttyardprocess*" module (the tt's coming from the yard) or the "*ttquayprocess*" module (the tt's coming from the quay), depending on which generator created the tt.

7.5.3.1.4 The class component tractor-trailer (tt)

A tt is created and activated by a ttgenerator. The actions of the tractor-trailers are described in two modules:
- "*ttyardprocess*": describing the actions of the tt's generated to come from the yard;
- "*ttquayprocess*": describing the actions of the tt's generated to come from the quay.

These two different modules are explained below:

ttyardprocess module
First a tt joins a queue ("yardq") and waits until granted permission by the traffic light to proceed its course. This traffic light (module yardlightprocess) is activated the moment a tt enters the queue. The traffic light actions are explained in paragraph 7.5.3.1.6 yardlight & ictquaylight.

Once re-activated by the yardlight, the tt accelerates until it reaches its maximum driving speed (30 km/hour) and then proceeds at a constant speed until it reaches the IIT quay entrance is (after 1350 meters). At his point the tt's with destination ICT quay and IIT quay, have to follow different actions:

- tt's driving to one of the ICT quays, continue at the same speed for another 450 meters until they arrive at their quay where they are terminated;
- tt's with the IIT quay as destination, enter the central traffic lane and join a queue ("*crossq*") to wait until further notice. The traffic light that manages this traffic flow is activated the moment a tt enters the queue (as explained in paragraph 7.5.3.1.5: module crosslightprocess). Once re-activated by the traffic light, the tt crosses the traffic lane to drive onto the IIT quay where the tt is terminated.

ttquayprocess module
In this module a ttgenerator activates the tt to start its actions. Depending on which quay the tt came from, the following tt traffic procedures will apply:

- tt coming from one of the ICT quays:

This tt first enters the queue of the ICT quay traffic light and waits until granted permission to drive onto the bridge. The moment a tt enters the queue, this traffic light is activated in the module ictquaylightprocess (see paragraph 7.5.3.1.6). After being re-activated by this traffic light, the tt accelerates until it reaches the maximum speed limit on the bridge (30 km/hour). Once this speed is reached, the tt proceeds at this speed and enters the "*crosszone*" and subsequently the "*insertzone*" before passing the IIT quay (after 450 meter). Having passed these zones, the tt proceeds at its maximum speed for another 1350 meters until it reaches the yard where the tt is terminated.

- tt coming from the IIT quay:

Also this tt first enters a traffic light queue and waits until it receives permission to drive onto the bridge. The moment a tt enters the queue, it activates the module insertlightprocess that manages the actions of this traffic light (see paragraph 7.5.3.1.5). After being re-activated by this module, the tt drives onto the bridge and increases its speed until reaching the maximum speed limit on the bridge. Subsequently the tt proceeds at this speed till reaching the end of the bridge (after 1350 meters) where the tt is terminated.

7.5.3.1.5 The component crosslight & insertlight

The actions of the traffic light managing the crossing of the tt's from the bridge onto the IIT quay are described in the module "*crosslightprocess*". The traffic coming from the IIT quay onto the bridge is regulated in the module "*insertlightprocess*". As these modules are identical except for one action, both modules are hereafter discussed mentioning their difference.

These modules are activated the moment a tt enters the queue for the particular traffic light. The traffic light managing the crossing re-activates the first tt in the queue the moment the "crosszone" section on the bridge is free of tractor-trailers (see figure 7.7). The traffic light managing the tt traffic onto the bridge re-activates the first tt in the queue the moment the "insertzone" section of the bridge is empty (see figure 7.7). After having re-activated the tt, the traffic light waits for a certain "*follow up time*" as specified by the user, before it checks whether the queue is empty or not. In case the queue is empty, the traffic light becomes passive until re-activated by a tt. If on the other hand the queue is not empty, the traffic light re-activates the next tt the moment the specified section of the bridge is empty (either the "crosszone" or the "insertzone" depending on the traffic light).

7.5.3.1.6 The component yardlight & ictquaylight

The actions of the traffic lights located at the ICT quay and the yard are also identical. Both modules are activated the moment a tt enters the queue for the traffic light. The traffic light then re-activates the tt and subsequently waits a certain "follow up time" (the same time as explained earlier in this chapter). In case the queue is empty, the traffic light becomes passive till re-activated by the next tt entering the queue. In case the queue is not empty, the traffic light repeats its actions.

7.5.3.2 Sensitivity analysis & computer simulation results

To test the sensitivity of the model a number of simulation runs have been made with different values for the following three parameters:

a) mean values of the exponential distributions used to generate the inter-arrival times of the tractor-trailers;
b) maximum driving speed on the bridge;
c) traffic light follow-up time.

a) Sensitivity for the mean values of the exponential distribution
Four sensitivity runs have been made with the values 25, 50, 100 and 200 seconds as mean value for the exponential distributions out of which the inter-arrival times of the tractor-trailers are generated. These runs have been made with a maximum speed limit of 30 km/hour and a 4 seconds follow-up time at the traffic lights. The results are summarised in table 7.6 below.

exp.distr.	crossq		insertq		yardq		Ictquayq	
Mean (sec.)	waiting time	max. length	waiting time	max. length	Waiting time	Max. length	Waiting time	max. length
25	8.3	13	5.3	8	1.8	9	0.9	7
50	2.3	4	1.6	4	0.6	4	0.4	4
100	1	4	0.7	3	0.3	4	0.2	3
200	0.4	2	0.3	2	0.1	3	0.1	2

table 7.6: Bridge run results with varying mean value (waiting time in seconds)

The queue statistics and the graphs of the queues are presented in appendix 7.6. A smaller mean value, corresponding with a shorter inter-arrival time of the tt's, results in a longer queue length and increased average waiting times in the queues (see figure 7.8 below). This is to be expected as more traffic is generated while maintaining the follow-up time of the traffic lights. The queues at the crossing and insertion points also increase when the inter-arrival times of the tt's is reduced: because of increased traffic the tt's have to wait longer before being allowed to cross the traffic lane or to drive onto the bridge.
The graphs in appendix 7.6 show that for larger mean values the density of traffic on the bridge is automatically reduced.

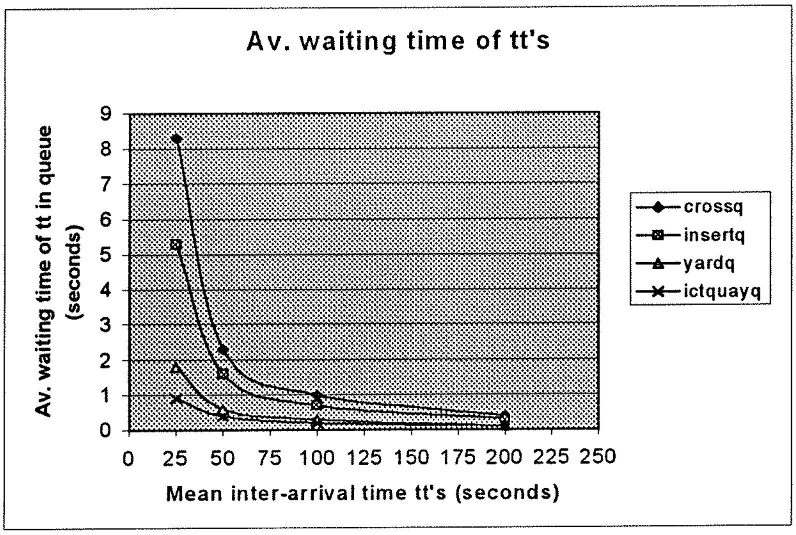


figure 7.8: Waiting times in the queues with varying mean inter-arrival time of the tt's

b) Sensitivity for the maximum driving speed on the bridge
Two additional sensitivity runs have been made with the following values for the maximum speed limit on the bridge: 20 and 40 km per hour. The results are summarised in table 7.7 below.

	waiting time		yardtoiit		littoyard	
max. speed	insertq	crossq	max. length	Time driving	Max. length	Time driving
20	0.5	0.97	21	245.2	22	243.7
30	0.7	0.98	16	165.5	17	163.2
40	0.9	0.97	14	126	16	123

table 7.7: Bridge run results with varying speed limit (time in seconds)

The results confirm that a higher maximum speed limit (for a constant mean value of 100 seconds for the inter-arrival times and a follow-up time of 4 seconds) leads to a lower traffic density on the bridge: evidently less time is required to drive over the bridge (see also graphs and queue statistics in appendix 7.7). The queue at the crossing traffic light is hardly effected by the changing speed limit. The queue statistics of the traffic light at the insertion point however, show a more significant effect. At a higher driving speed the free distance ("insertzone") required to permit the tt to drive onto the bridge (from the IIT quay) increases more rapidly than the "crosszone" distance required for the tt to cross the traffic lane. This is also due to the assumption that the acceleration rate of the tt's remains the same.

c) Sensitivity for the traffic light follow-up time
The model has been run with four different values for the traffic light follow-up time: 2, 4, 8 and 12 seconds. Some results are shown in table 7.8 below. The queue statistics and graphs are presented in the appendix 7.8.

follow-up time (sec.)	waiting time (sec.)			
	crossq	insertq	yardq	ictquayq
2	0.9	0.6	0.06	0.04
4	1	0.7	0.3	0.2
8	1.3	0.9	1.2	0.8
12	0.9	1.4	3.3	2

table 7.8: Bridge run results with varying follow-up time

The results indicate that a longer follow-up time results in higher average waiting times in the queues for the corresponding traffic lights. In the case of the crosslight queue, the waiting time will increase until the follow-up time reaches a value at which the tt's have the same or a longer distance between each other as the length of the "crosszone" section (see figure 7.9 below). In case the maximum speed limit is set at 30 km per hour and the follow-up time at 9 seconds, the distance between the tt's will be about the same as the length of the "crosszone": resulting in a short time for the tt's to wait in the queue before crossing the traffic lane. The queue at the insertion point remains dependent of the follow-up time of the traffic light.

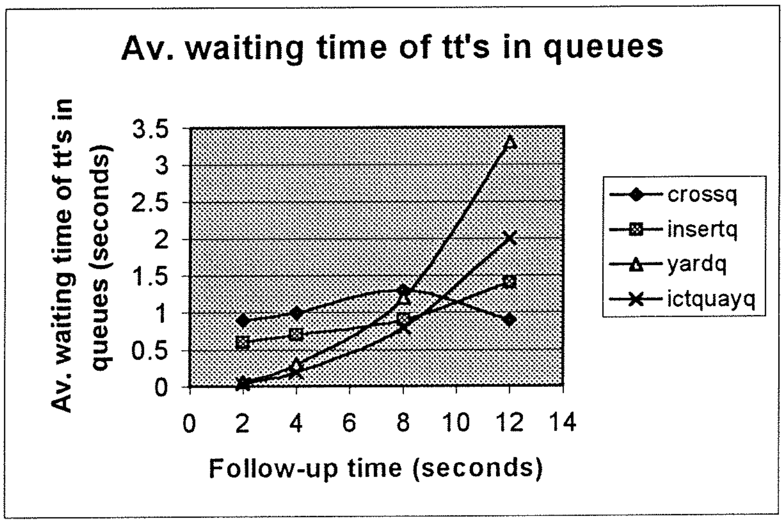


figure 7.9: Waiting times in the queues with varying follow-up times of the traffic lights

7.5.3.3 Conclusions

Based on the sensitivity runs made it can be concluded that the bridge model gives reasonable and expected results.

Higher inter-arrival times of the tt's generated by the model, result in shorter average waiting times in the queues and in a reduced average maximum length of these queues (the opposite occurs in case the inter-arrival times of the tt's are reduced). This is to be expected because the time between two tt's driving onto the bridge is determined by the pre-set follow-up time, which time was not changed during these simulation runs

In case the follow-up time is increased, the average waiting time of the tt's in the queues will increase. The waiting time of the tt's in the queue waiting to cross the traffic lane and to enter the IIT quay, depends largely on the follow-up time.

- In case this follow-up time results in a distance between two tt's smaller than the length of the "crosszone" (this zone needs to be free of tt's to assure a safe crossing of the bridge), the waiting times of the tt's in this queue will increase when the mean inter-arrival time of the tt's reduces ;
- in case the value of the follow-up time results in larger minimum distances between the tt's than the "crosszone", the waiting times of the tt's in this queue will reduce.

Evidently at an increased maximum speed limit on the bridge, the tt's require less time on the bridge. If the other variables are not changed, the number of tt's on the bridge will also be less. The results also show that the waiting times in the insertion queue increase with higher speed limits. This effect is understandable since the tt's driving onto the bridge need a certain acceleration time before reaching the maximum speed allowed on the bridge: the larger the maximum speed of the tt's on the bridge, the larger the distance will have to be between the accelerating tt and the tt coming from the ICT quays.

Note It should be kept in mind that although the accuracy of the computer model is not in the order of seconds, the results obtained from this computer model do give an indication of the bridge capacity. The results indicate that the bridge is not likely to become a bottleneck in the future, unless the terminal capacity is increased far beyond the in this report investigated situations.

8. Overall computer simulation results

8.1 Introduction

In the previous chapters the terminals have been treated separately as these are intended to be operating independently from each other in the future. Each terminal has been discussed with its own terminal operations. Hereafter, an indication of the overall production level of the terminals is presented.
The bridge connecting the quays to the container yards, may constitute a potential limiting factor for the production level of the whole terminal. As tractor-trailers of all three terminals will make use of the bridge simultaneously, this link has been analysed in more detail in order to verify its capacity to cope with the total traffic load of all three terminals.

In chapters 8.2 and 8.3 hereafter, first an overall picture of the terminal capacities (ICT1 and IIT) is given with some tentative analysis of the consequences on the terminal operations if the assumptions made can not be realised.

In chapter 8.4 the traffic load on the bridge is further analysed, whereby the bridge model is run with the anticipated traffic flows following the expansion of the terminal. These analysis concern the expected traffic flows on the bridge in case the maximum container throughput is reached in the terminals as simulated in chapter 7 (thus simulating the highest possible traffic load on the bridge). Above chapter is followed by an analysis of the situation in which a calamity occurs at one of the quays (or at all of the quays): the possible effects of such calamities are examined as far as the traffic flows on the bridge and the operations in the container yards are concerned.

Finally an approximation of the maximum capacity of the bridge is discussed, including the effects on the terminal operations and their maximum container throughput.

8.2 ICT capacity and its possible limitations

In chapter 5 the present ICT container terminal operations have been simulated and discussed: the terminal handled about 380,000 box moves in 1997 at an average portainer production rate of about 20.5 box moves per hour.

According to terminal management, the capacity of the portainers can be increased to 30 box moves per hour per crane if the efficiency of terminal operations could be improved. In chapter 7 therefore, the 2005 terminal operations have been simulated based on above assumptions: for this hypothetical situation, the simulation runs resulted in a capacity of 580,000 box moves in 2005, while maintaining a quay occupancy rate of about 70 % as obtained in 1997 (as explained in chapter 7).

In addition, in chapter 7 is was assumed that in the future a different arrival distribution could be implemented for the trucks delivering and picking-up the containers, in order to alleviate the pressure on the stacks necessary to accommodate the anticipated increased container traffic. Such a "stricter" arrival distribution may however in reality be difficult to realise.

The possibility must however be taken into consideration that in the future above assumptions may prove to be unreachable, in case - because of persisting lower efficiency levels - the portainer production can not be increased to the extent required to handle the anticipated increase in container traffic.

A number of additional simulation runs have been made to demonstrate the possible consequences of above situations:

- first the terminal has been simulated with a larger and smaller container throughput than in 1997, while maintaining the present portainer production rate (with an influence factor of 0.68);
- subsequently, similar runs have been made for the possible terminal operations in 2005 assuming an increased portainer production (with an influence factor of 0.8).

The possible consequences are summarised and illustrated in the table 8.1 and graphs below (see also appendix 8.1).

box moves/year	vessels	Quay occ.	t.a.t.	i.a.t.	av. waiting time anchorage	portainer production level
337,415	859	56%	17.0	10.0	0.1	20.5
382,411	973	70%	18.7	8.8	1.9	20.5
427,531	1088	83%	19.9	7.9	17.9	20.5
517,147	1316	61%	12.0	6.5	0.1	30
579,345	1474	72%	12.7	5.8	2.0	30
650,034	1654	85%	13.3	5.1	16.1	30

table 8.1: Overall view ICT1 with different container throughputs & production levels

From these results it is evident that:

- in case the portainer production remains at the 1997 level: an increase of the number of vessels calling at the terminal will result in a higher quay occupancy and longer waiting times for the vessels in the anchorage. Also the turn-around times of the vessels will increase due to this increased traffic;
- in case the portainer production increases to 30 box moves per hour as assumed for the 2005 situation: the same effects can be observed, but this time at much higher throughput levels.

As stated earlier, the portainer production rate appears to be the main limiting factor at the terminal, since the operations in the container yard do not cause delays for the quay operations (i.e. under the assumption that the truck arrival distribution can be changed).

Figure 8.1 below shows that - in case the terminal aims to keep the quay occupancy at about 70 % or less - the portainer production will have to increase in order to enable the terminal to handle the increased number of containers anticipated.

While it is the objective to remain close to the 70 % quay occupancy line in the figure 8.1 below, figure 8.2 clearly illustrates that when the efficiency level cannot be improved, the terminal will have to face a rapidly increasing quay occupancy in servicing the expected increased container throughput (at the same time vessels will be confronted with increased anchorage waiting times).

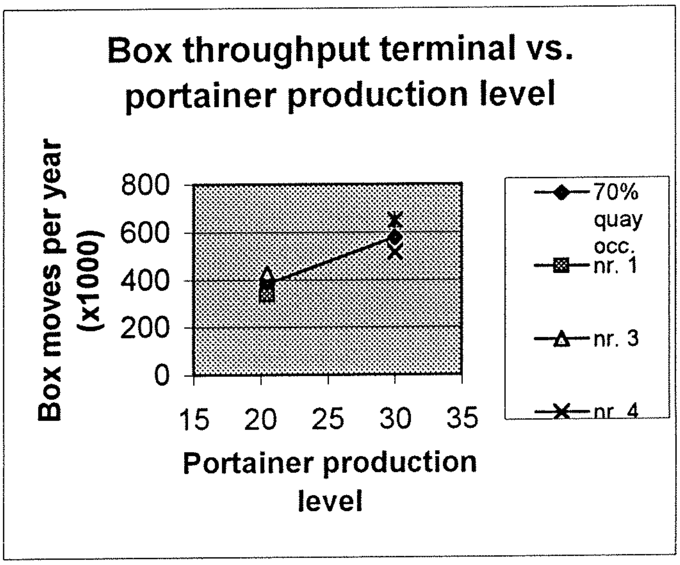


figure 8.1: Box throughput versus portainer production level

Explanation of the legenda figure above:

- The 70% quay occupancy line represents the approximate portainer production rate required to ensure a 70 % occupation of the quay;
- Nr. 1 representing a lower container throughput with the 1997 production rate (thus a lower quay occupancy);
- Nr. 3 representing a higher container throughput with the 1997 production rate (an higher quay occupancy);
- Nr. 4 representing a lower container throughput with the for 2005 assumed higher production rate;
- Nr. 6 representing a higher container throughput with the for 2005 assumed higher production rate.

Note: above used numbers are also used in the figure below.

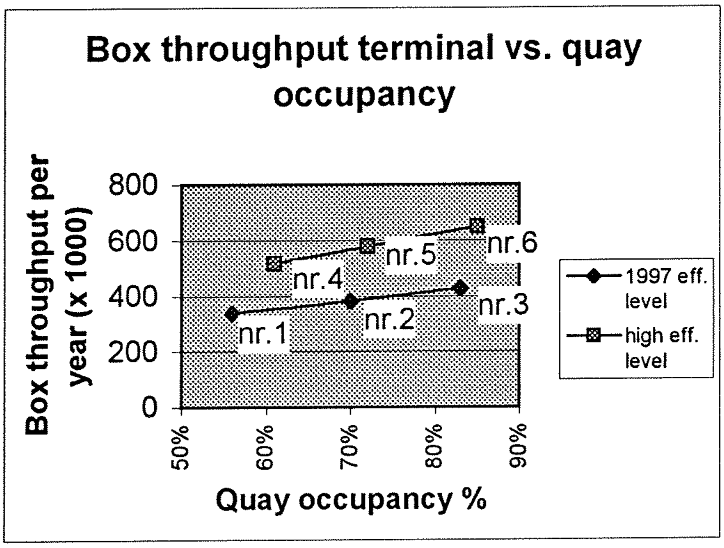


figure 8.2: Box throughput per year versus quay occupancy

In case the truck arrival distribution can indeed be changed as assumed in chapter 7.2.2.1, there will hardly be a problem in the container yard (the queue statistics in appendix 8.1 show that the queues in the yard do not change either). However in the case this truck arrival distribution can not be adjusted then a shortage of stack volume must be anticipated at the expected higher container throughput level (in particular so in the empty and import stacks).

Note: As a temporary measure, this problem could be solved by placing such containers in the export stacks. However this can only be a temporary solution since also these export stacks are increasingly filling-up. In addition this would complicate the required proper record keeping of containers in the different stacks.

8.3 IIT capacity and its possible limitations

In chapter 7.5 the Inter Island Terminal was simulated for two portainer situations: one with two different portainers and one with four identical portainers in operation at the quay. This distinction has also been maintained hereafter.
For both terminal situations a theoretical maximum container throughput was simulated in chapter 7.5, assuming a high 80 % level of efficiency of the terminal operations.

Obviously, these maximum container throughputs will not be reached in case of a lower efficiency level and if the quay occupancy rate is not allowed to exceed 70 %.

Note: Since the operations of this IIT have been based on the ICT's, it is assumed that about the same relations apply for the IIT as for the ICT's.

The indicative numbers of box moves that can be handled by the IIT at different efficiency levels, are analysed hereafter.

IIT1 - use of two different types of cranes
When assuming a similar efficiency level (68 % and an influence factor of 0.68) as applied for the ICT operations in 1997, a total number of about 196,000 box moves can be made by the IIT while maintaining a 70 % quay occupancy. With the higher efficiency level (80 % and an influence factor of 0.8) as assumed in chapter 75, a container throughput of about 230,000 can be reached (see table 8.2 and figure 8.3 below).

IIT1	Box moves per year	efficiency level	quay occ.
	196,000	68%	70%
	230,000	80%	70%

table 8.2: IIT1 indication box throughput per year

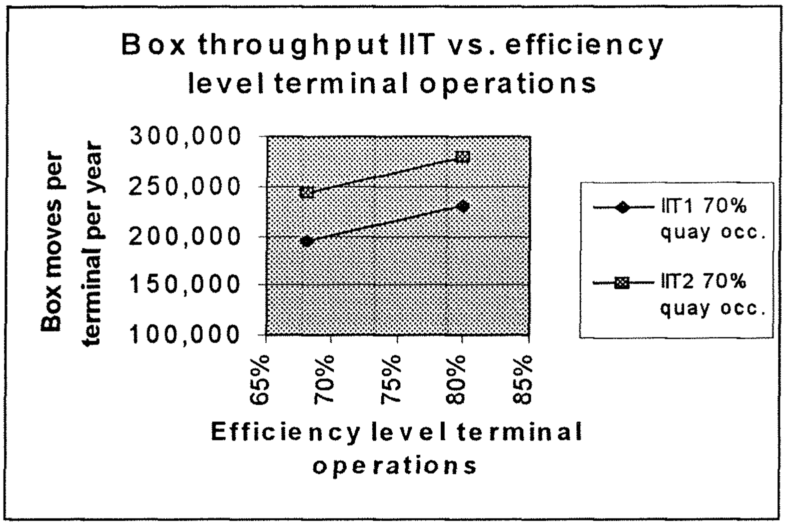


figure 8.3: IIT1 & IIT2 box throughput versus efficiency level terminal operations

For the analysis of the possible consequences it is assumed that for the IIT similar observations apply as earlier explained in 8.2 for the ICT:

- in case the portainer production cannot be increased as planned (due to various efficiency limitations of the terminal operations), the anticipated higher container throughput will result in a higher quay occupancy as well as in longer anchorage waiting times (the opposite will occur when the portainer production rate increases while the container throughput doesn't);
- for the container yard the same applies as for the ICT container yards: in case the truck arrival distribution can not be changed, the stacks in the yard will occasionally be filled 100 %. This means either that containers have to be stacked in another stack or that the trucks have to wait until the stacks have space. Both solutions may have negative consequences on terminal operations: stacking in other stacks implies extra work and trucks waiting outside the terminal may ultimately result in vessels having to wait for their load. The export stacks are on the whole less full than the empty and import stacks due to the fact that the trucks picking-up these containers from the yard arrive inside a shorter time-span.

IIT2 -use of four identical portainers

The theoretical maximum container throughput calculated in chapter 7.5 amounted to about 280,000 box moves per year with four identical portainers servicing the vessels. This capacity was achieved assuming the high 80 % efficiency level of terminal operations (with an influence factor of 0.8). In case however that this efficiency level cannot be improved beyond the 1997 level of 68 %, only about 243,000 box moves can be made per year (see table 8.3 below and figure 8.3 above).

IIT2	box moves per year	efficiency level	quay occ.
	243,000	68%	70%
	280,000	80%	70%

table 8.3: IIT2 indication box throughput per year

Otherwise the same observations apply as above under IIT1.

8.4 Traffic load on the bridge

In this paragraph the traffic load on the bridge has been investigated for the situation that all three terminals are operating, each at their highest traffic load as defined in chapter 7. In this chapter it was found that both ICT's can handle about 580,000 boxes per year each and the IIT (with two different quay cranes) about 230,000 box moves or (in case four identical portainers are operative) about 280,000 box moves per year. The following four runs have been made:

- Run 1: both ICT's are operating with 6 tt's per portainer and the IIT (with two different quay cranes) with 5 tt per portainer;
- Run 2: both ICT's are operating with 8 tt's per portainer and the IIT (with two different quay cranes) also with 8 tt's per portainer;
- Run 3: the same situation as run 1 except that the IIT operates with four identical portainers;
- Run 4: the same situation as run 2 except that the IIT operates with four identical portainers.

Observations:

- Run 1

The average tt inter-arrival time on the bridge amounts to 90 seconds for the ICT1, 93 seconds for ICT2 and 230 seconds for the IIT (see appendix 8.3).

Note: The higher inter-arrival time for the IIT is a logical consequence of the much lower portainer production at the quay (the queue statistics and graphs are presented in appendix 8.7).

The average waiting time for the tt's in the queue on the bridge is 1 second and the maximum queue length 3 tt's. For the insertion queue the average waiting time amounts to less than 1 second whereas the maximum length of tt's remains three. The waiting times of the tt's at the yard and at the ICT quays are negligible and the maximum length of the queue is 4 tt's.

- Run 2

In this situation, the average tt inter-arrival time on the bridge is 70 seconds for the ICT1 bound tt's, 75 seconds for ICT2 and 65 seconds for the IIT (see appendix 8.4).

Note: The lower inter-arrival time for the IIT bound tt's is due to the fact that the model sends tt's back to the yard when a certain queue length at the portainers is reached. This is a simplification of the terminal model to make sure that the queue lengths at the quay do not become too large. In reality this can be solved depending on the expertise of the traffic managers in charge.

In this situation the queue lengths and the average waiting time of the tt's in the queues are only slightly higher than in run 1 (see appendix 8.8).

- Run 3

The only difference with run 1 concerns the inter-arrival time of the IIT bound tt's which is 220 seconds instead of 230 seconds (see appendix 8.5).
Consequently, the results do not differ much from the run 1 results (see appendix 8.9).

- Run 4

This run has been simulated with a slightly lower value for the inter-arrival time of the IIT bound tt's, namely 64 seconds (as compared to 65 seconds in run 2). The BestFit analysis of this data can be viewed in appendix 8.6.
The results hardly differ from the run 2 results (see appendix 8.10).

Concluding it appears that the traffic load on the bridge created for the situation that all three terminals operate at the theoretical maximum container throughput, does not negatively effect the terminal operations at all. Also an increase of the numbers of tt's operating per portainer, hardly effects the traffic operations on the bridge. This is also clearly illustrated in figure 7.8 in chapter 7.5.

Note: It is noted that the bridge situation corresponding with the ICT's operating at the 1997 production levels, has not been simulated since the traffic density on the bridge will even be less than in the four situations simulated above.

For comparison:
For the situation described in chapter 8.2 with quay occupancy levels exceeding 70 %, the tt inter-arrival times are as follows (see appendix 8.2):

- in case 427,531 boxes are handled at the 1997 portainer production rate, the tt inter-arrival times are approximately 106 seconds;
- in case 650,034 box moves are made, the inter-arrival times are about 81 seconds.

8.5 Traffic load on the bridge in case of calamity at the quay

In case a calamity occurs at the quay with result that one of the berths temporarily can not be used for servicing a vessel, this may cause filling up of the anchorage. This build up of vessels will lead to an higher than normal quay occupancy once the "calamity" is solved.
This exceptionally high peak load at the quays, will cause an extra traffic load on the bridge and at the quay. This "calamity" situation has been simulated by generating five vessels to arrive almost simultaneously in the anchorage: in this way the quay occupancy is increased temporarily and thus the traffic density.
This "calamity " has been simulated for each terminal (and with varying numbers of tt's per portainer). The resulting average inter-arrival times of the tt's on the bridge are illustrated in table 8.4 below (see also appendix 8.11).

Terminal	tt per portainer	av. i.a.t.
ICT1	6 8	81 67
ICT2	6 8	85 70
IIT1 (two different quay cranes)	5 8	168 45
IIT2 (four identical quay cranes)	5 8	157 47

table 8.4: Inter-arrival time tt's on bridge due to a "calamity" at the quay (seconds)

The situation with 8 tt's per portainer active for all three terminals has been simulated and the queue statistics and graphs can be read in appendix 8.12.

Note: Since the situation with the ICT's operating with 6 tt's per portainer and the IIT operating with 5 tt's per portainer, is similar to the situation simulated in 8.4, no additional runs have been made for this situation.

From these queue statistics it appears that the waiting times in the queues are slightly higher than the waiting times obtained for the maximum container throughput situation. This is to be expected since, due to the temporary extra high quay occupancy, all the portainers and thus also all the tt's will be operative (when there is no such "calamity" less tt's will be in use).
The extent of the waiting times and the maximum lengths of the queues, are still too small to have a negative effect on the terminal operations. The waiting times in the queues of the IIT bound tt's, also remain small (ICT bound tt's have much larger inter-arrival times than IIT tt's).

8.6 Maximum capacity of the bridge

In this chapter an indication is given of the maximum capacity of the bridge.

Note: The "maximum capacity" is reached if and when the density of traffic on the bridge is such that the traffic flow is seriously disturbed and consequently effecting the other terminal operations.

If tt's are delayed by the traffic on the bridge for such a long time that the portainers have to wait for the tt's to arrive, then the terminal operations will be seriously effected. For this to happen in the 1997 situation, the tt's with ICT quay destination would have to be delayed for about 18 minutes (see queue statistics in appendix 6.4). The IIT bound tt's would have to be delayed even more than the ICT bound tt's.

Reviewing the queue statistics of the bridge model runs made in 8.4, it appears that occasionally a tt will have to wait longer than average before being able to resume its course. These maximum waiting times (maximum of about 45 seconds for a tt in "crossq" in run 2) however, do not come close to the values required in order to effect the terminal operations on the quay.
The same can be concluded from the results of the bridge model in case of a "calamity" (a calamity being defined as a situation causing a temporary higher traffic load on the bridge): although the corresponding traffic load is higher, the bridge appears to be able to handle such a peak-load adequately without causing disturbances in other terminal operations. This is also illustrated in figure 7.8 in chapter 7.5.

It is noted that figure 7.8 only gives an indication of the waiting times in a queue as a function of the mean inter-arrival time of the tt's. This graph is made assuming that the tt's with different destinations have the same mean inter-arrival times. In chapter 8.5 the ICT bound tt's have a much larger inter-arrival time than the IIT bound tt's. Therefore the IIT bound tt's will naturally be effected less than if the ICT bound tt's had smaller inter-arrival times.

Concluding:
Even with the theoretical maximum container throughput of 1,160,000 containers per year (as discussed in chapter 7), the bridge will not effect the operations of the new terminal. As above simulated maximum throughput corresponds with an, under the present economic conditions, unlikely increase of the international trade with 300 % compared to the present 1997 situation, it can be concluded that the bridge is not likely to constitute a limiting factor effecting the terminal operations in the near future.

9. Conclusions

In this thesis research project the two International Container Terminals (ICT1 & ICT2), the Inter Island Terminal (IIT) and the bridge of the Tanjung Perak harbour have been studied. Simulation models have been developed to analyse the operations of these terminals with a particular emphasis on the resulting bridge traffic.

Analysis of the 1997 operations of the at present operational ICT, revealed that approximately 68% of the berthing time of a vessel is effectively used for unloading and loading the vessels. Various unspecified delaying factors appear to effect the efficiency of the terminal operations, such as:
- "human factors": possible loss of effective time in connection with the breaks (three breaks per day) and the changing of working shifts (three shifts per day);
- equipment breakdown: possible cause for loss of effective time;
- unforeseen delays as result of bad weather conditions or for instance strikes which are beyond the control of terminal management.

As far as the yard operations are concerned, there appear to be no major problems (in spite of the fact that the import and empty stacks reach an occupancy of approximately 80%).

In anticipation of an increased container throughput per ICT in the future, it was assumed that certain terminal operations might have to be reconsidered, particularly the portainer operations and the container yard operations:

1. Portainer operations

In the present 1997 situation the portainer production rate is limited to about 21 box moves per hour. According to terminal management, this production rate can be increased to 30 box moves per hour per crane by improving the efficiency of terminal operations.

The simulation models showed that if the expected increase in container throughput cannot be matched by above assumed increased portainer production rate, this may have a serious impact on the servicing of the vessels, such as:
- Increased quay occupancy and corresponding increase of average waiting time of the vessels;
- Increased turn-around times of the vessels;
- A higher quay occupancy rate (in this case exceeding 70%) means that the terminal will have limited room for unavoidable, unforeseen delays, thus causing more delays.

Obviously, above developments would not be appreciated by the shipping companies and are thus to be avoided.

Based on the assumption that the terminal can improve its efficiency level (thereby making possible a portainer production rate of 30 box moves per hour), one ICT may be able to handle about 580,000 box moves per year (while maintaining a 70% quay occupancy).

Note: As compared to the present ICT situation, this implies that after completion the scheduled ICT's may in theory be able to cope with a 300% increase of the number of containers.

The major factor determining the servicing the vessels, appears to be the portainer production rate itself: the simulated increase in tt's hardly does effect the turn-around time of the vessels (merely resulting in increased waiting times for the tt's at the assigned portainers).

The main difference between the two ICT's, consists of the longer distance the tt's have to drive between the quay and the container yard. The simulation models showed that a slight increase in the number of operational tt's is to be recommended (about 8 instead of 6 tt's per portainer).

Also for the scheduled IIT, the portainers also constitute the main factor determining the container throughput of the terminal:

- With two portainers and two quay cranes (having a lower production rate) operative, a maximum box throughput of about 230,000 can be reached while maintaining a 70% occupancy of the quay;
- With four identical portainers (as originally planned), the box throughput may increase to about 280,000 per year (also at a 70% quay occupancy).

2. Yard operations

The yard facilities and the stacking areas are sufficient to deal with the present (1997) container throughput. For the anticipated increased container throughput, a number of stacks in the container yard will however be filled 100%. If no further measures are taken, this situation may result in:

- Increased loss of time for the trucks awaiting delivery or collection of their load (resulting in further filling-up of the parking lots outside the terminal);
- Temporarily stacking of especially import and empty containers in other stacks (complicating the proper monitoring of containers in these stacks).

Ultimately, above situation can result in increased service times of the vessels at the quay.

The results of the simulation models showed that above stacking area constraints occurring at higher container throughput levels, can be overcome by adjusting the truck arrival pattern, i.e. by reducing the dwell time of the containers.

At present the truck parking lot outside the terminal grounds already has a high occupancy. With the scheduled operation of a second terminal, this parking lot area may prove to be insufficient to accommodate about twice the number of trucks as is the case at the moment (since the capacity of this parking lot is outside the scope of this research, this component has not been researched further).

The yard cranes (RTGC's) appear to have an adequate capacity to handle the traffic load in the yard, also if the container throughput is increased as planned. If required it can be decided to assign two or three RTGC's to a stack in case of heavy traffic in that particular stack. This adequate RTGC capacity has also been confirmed by the simulation runs for the situation in which trucks are permitted to enter the yard at all times and for the option that tt's have no priority over the trucks. In both cases the tt waiting times do not increase up to the extent that portainer operations are effected. In practice, the number of trucks present in the container yard is limited and so that the tt's serviced without too much delay.

The container yard of the IIT has enough stacking area to accommodate the flow of containers following the simulated maximum IIT container throughput. It has been assumed that the capacity of the parking lot for the IIT bound trucks is sufficient (at a different location than the ICT parking lot). Otherwise for the IIT yard operations, the same conclusions apply as for the ICT yard operations.

3. Bridge traffic flow

Since it was not known whether the bridge, connecting all three scheduled terminals, might constitute a possible bottleneck for the future, it was decided to pay particular attention to this bridge.

At present the bridge functions as a two-way road. Following the scheduled operations of the new IIT located along this bridge, it is clear that tractor-trailers with different destinations (ICT or IIT) will have to cross each others traffic lanes in order to deliver or collect their load.
Since the bridge is wide enough to accommodate three traffic lanes, the bridge model simulated a situation in which the IIT bound tt's make use of a central traffic lane when arriving at the quay entrance. In this way, traffic coming from and going to the ICT quays is not disturbed (assumed that ICT bound tt's have priority over IIT bound tt's).

A traffic light (such as simulated in the model) or a traffic officer can be used to manage the traffic crossing the traffic lane and entering the traffic lane coming from the IIT quay. In practice it may also be possible to let the tt drivers decide whether it is safe to cross or enter the traffic lane.

The simulation models showed that even in case all three terminals are operating with an high efficiency level (with the corresponding increased portainer production rate), the traffic load generated on the bridge does not lead to delays for the tt's that will ultimately effect terminal operations (obviously the same implies in case the future traffic load is based on the present 1997 container throughput).

Also simulation runs for situations involving a "calamity" at the quay (resulting in temporary high terminal traffic loads), the bridge did not significantly influence the traffic flow.

Final observations:

- It has to be kept in mind that the scheduled new international container terminals in any case will be able to cope with twice the number of containers as handle at present by the single ICT (in case the efficiency level of the terminal operations and the portainer production rate remain as was the case in 1997);
- The simulation models showed that the total container terminal scheduled to be in operation in the future, will in theory be able to handle a much higher container throughput without the need of increasing the number of portainers or the container yard area.

10. Recommendations

1. In the computer simulation models a number of assumptions had to be made both for reasons of simplicity and because of lack of information on certain terminal operations. A more precise simulation of the terminal operations will be possible if more detailed information is collected and analysed concerning the following operations:
   - The specifics of the portainer process: survey to assess the actual time required for lifting containers onto and off the vessels and to analyse the shifting strategy of the portainers applied in practice;
   - Factors effecting terminal efficiency: through a survey, a better understanding of the various "influence factors" is to be pursued in order to be able to investigate and simulate the effects of the various effects causing a loss a of effective time to service a vessel. If a complete picture can be made of the importance and impact if the various factors effecting the efficiency of the terminal operations, recommendations may be formulated to enable the terminal management to take preventive or corrective measures;
   - The anchorage waiting times: more detailed information to be collected;
   - The "feedering" of containers: because of lack of information, in the present study the assumption was made that all the containers arriving by vessel will leave the terminal by truck. In practice, a number of containers will leave the terminal by another vessel, thus requiring less trucks delivering containers. The dwell times of these containers and the frequency of occurrence can also be investigated by means of a survey;
   - The strategy of truck admittance and RTGC assignment to the stacks: at times when there is a gap in the servicing of tt's , trucks are allowed to enter a stack. During busy periods a third RTGC may be placed in a stack. A survey to analyse the strategy followed in practice regarding above processes, may result in more precise information required to adequately simulate the stack occupancy;
   - The occupancy level of the truck parking lots: the present capacity and occupancy level of the truck parking lot are not known. Through a survey relevant information may be collected required to verify whether this parking lot may become a bottleneck for the anticipated increase in truck traffic.

2. As the Inter Island terminal (IIT) has yet to be completed, its operations have been simulated in a more simplified manner as compared to the International Container Terminals. In this study, the portainers have been assumed always to remain at the same berth place. In practice however, these portainers are likely to shift (thus more efficient use of the portainers). If more information would become available regarding the strategy followed by the terminal management, a more realistic simulation of the IIT operations will be possible.

3. In this research it has been assumed that the three terminals will be operating independently from each other. In practice however, a certain degree of interaction between the ICT's and the nearby IIT will be aspired, aiming to improve the efficiency of certain operations. Exactly which interactions will be possible after the completion of the terminals is not known. The following interactions which possibly may result in an increase of efficiency in terminal operations are recommended to be investigated further:
   - It seems likely that a certain number of containers, delivered to the IIT by vessel will ultimately leave the terminal by vessel via the ICT. In this case, containers can be directly transported from one terminal to another, thus requiring less trucks arriving at the ICT and collecting containers at the IIT (the same applies in the opposite direction). It is not known whether such containers will first be stacked in the IIT yard or directly in the ICT yard (or vice versa). Ultimately, such interaction may result in less and more efficient yard traffic;
   - In future it may also be possible that a vessel initially scheduled to arrive at one ICT will be assigned a berthing place of the other ICT if required. This may result in reduced anchorage waiting times of the vessels and thus the total loss of time;

- In future it may also be possible to temporarily use available stacking area in one container yard if in the other container yard a shortage is expected.

4. Economic picture: an economic analysis of the present and future terminal operations was not part of this study. A study on certain shortcomings of the terminal operations and the corresponding economic losses, may support proposed changes in terminal operations.

References

Sierenberg & de Gans b.v. (1987-1993) "Personal prosim Reference Manual", Third edition 3.00, Waddinxveen, The Netherlands

Sierenberg & de Gans b.v. (1987, 1988) "Prosim Textbook", bijgewerkt voor release 3.03, Waddinxveen, The Netherlands

Velsink, H. prof. ir. (October 1994) "Ports and Terminals", Planning and Functional Design, lecture notes f12/f13, Delft University of Technology, Delft, The Netherlands

Groenveld, R. ir. (October1994) "Service Systems in Ports and Inland Waterways", lecture notes f12/f13, Delft University of Technology, Delft, The Netherlands

Raport Inter Island Container Terminal, Surabaya (December 1997) "Rencana Pengoperasian Dermaga Peti Kemas Antar Pulau" (in Indonesian)

Internet site:
http://www.petra.ac.id/english/science/manufactures/pelabuhan/index.htm
"General Picture- P.T. Pelabuhan Indonesia III"

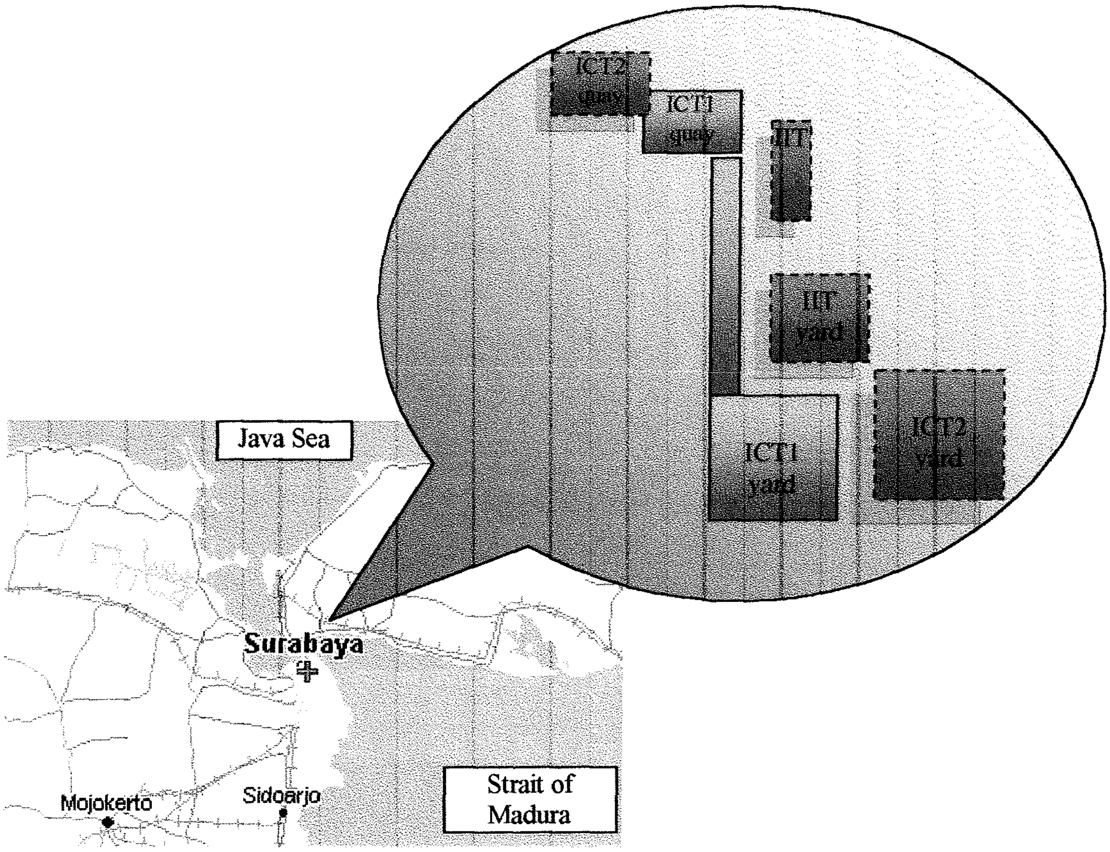
Computer simulation model

International Container Terminal

"Tanjung Perak"

Surabaya, Indonesia

User manual



Final graduation report

S. Wanders
September 1998

International Container Terminal

"Tanjung Perak"

Surabaya, Indonesia

Final graduation report

User manual

S. Wanders

September, 1998

Under supervision of:

Prof. ir. H. Ligteringen
Professor of Ports and Inland Waterways (part of the Hydraulic and Offshore Engineering section of the Division Hydraulic and Geotechnical Engineering, Faculty of Civil Enigineering, Technical University of Delft)

ir. R. Groenveld
Staff scientist of Ports and Inland Waterways

ir. J. C. Rijsenbrij
Interim manager Department Transport Technology, Faculty Mechanical Engineering and Marine Technology

Contents

1. Introduction

This user manual contains detailed descriptions of the models and is intended for those whom wish to analyse the models.

The second chapter gives a detailed description of the ICT 1 model that has been verified and validated using the 1997 data records of the terminal. This model has been used as basis for the future international container terminals since no major changes are foreseen in the operations of the present terminal.
The third chapter describes the planning model that creates data lists of vessels expected to call at a terminal. Since this model is practically the same when used for different terminal models, only the version that has been used for the creation of data lists for the ICT 1 terminal will be discussed.
Since the two future ICT models differ only slightly from the basis model, the fourth chapter only describes those modules and macros that have been modified.
The fifth chapter describes the model of the Inter Island Terminal (IIT). For this terminal model, the in chapter one described ICT model has also been used as basis. Of this IIT model only the modified modules and macros have been discussed.
The final chapter contains a detailed description of the bridge model.

All the computer models are described in the following way. Each module and macro is described in a separate paragraph. The computer printout of the module (or macro) is shown on the left page. The corresponding description is given on the right page, including a definition list explaining new terms. For the module DEFINE (present in every PROSIM model) only the printout is shown, because this module only gives a list of all the components and their attributes, queues and other variables. These terms are explained in the paragraph where they occur for the first time. In the appendix a list containing most of the used terms can be reviewed.

```
 1 CLASS                    : ship portainer rtgc transporter importgen exportgen
 2 COMPONENT                : shipgenerator berthmaster ptmaster ttmaster
                              trmasterquay trmasteryard truckgatemaster tlyard
                              tlquay tlgate
 3 QUEUE                    : anchorage berth[3] quay cranereqlst passivept
                              ttreqlst ttavailable yard ttq[10] truckq[10]
                              lane[10] ptset rtgcset xrtgcset unloadingship
                              loadingship quayqueue yardqueue tlyqueue tlqqueue
                              tlgqueue tgmq[10]
 4 TABLE(10)                : trucktab
 5 RANDOMSTREAM             : unif ptliftcycle ptliftontime ptliftofftime
                              rtgcrepostime rtgcliftontime rtgcliftofftime
                              rtgcdrivetime rtgcreshuftime
 6 INPUTSTREAM              : situation seedlist realinv
 7 OUTPUTSTREAM             : shipoutput yttonbridge qttonbridge iatlist
 8 TIMEUNIT                 : hours
 9 FIGURE                   : stackvol[10] berthocc[3] ptatquay[6] anchfig
                              channelinfig ttparking rtgcfig[10] movept[6]
                              ptimport[6] ptexport[6] tgmqfig[10] rowopenfig[10]
                              ttuptfig[4] ptboxfig[4] ptnrboxfig[4]
                              pttottimefig[4]
10
11 ATTRIBUTES OF MAIN:
12 INTEGER                  : a b c d e x w nrpt nrtt nrrtgc shipsatquay nr dd
                              rtgcstackvol[10] nrtab[10] runnr
13 REAL                     : totteu shsailtime ptshifttime quayspace bridgespeed
                              bridgetime mainlanetime mainlanespeed berth1time
                              berth2time berth3time quayspeed runtime ttperpt
                              rtgcspeed shinterarrivaltime landboxin landboxout
                              seaboxin seaboxout tringate troutgate shdeparttime
                              totptwtime totpt2time totberthtime shoccberth[3]
                              shiptotstime shiptotanchtime netoperation iattime
                              iattimetot shiparrivaltime
14 DOUBLE                   : totbox
15 LOGICAL                  : occupied[3] portaineractive[4] reservation[3]
                              placeocc[6]
16 CHARACTER(1)             : priority specification correction
17 CHARACTER(10)            : date simdate
18 REFERENCE TO SET         : sea channelin channelout shiftingpt quaytoyard
                              intraffic importpt exportpt outtraffic maintraffic
                              yardtoquay noworkrtgc xrtgcworkset quayintraffic
                              quayouttraffic gatequeue
19 REFERENCE TO portainer   : mmpt
20
21 ATTRIBUTES OF ship       :
22 INTEGER                  : shnr shexp20 shexp20f shexp20mt shexp40 shexp40f
                              shexp40mt shimp20 shimp20f shimp20mt shimp40
                              shimp40f shimp40mt shlength shimportteu
                              shexportteu shimportteuf shimportteumt
                              shexportteuf shexportteumt shexchange shreqpt
                              shberthnr shexportbay shimportbay nrrtgcw stackimp2
                              stackimp20f stackimp20mt stackimp40 stackimp40f
                              stackimp40mt stackexp20 stackexp20f stackexp20mt
                              stackexp40 stackexp40f stackexp40mt nrofship
23 REAL                     : shturnaroundtime shservicetime timearrival
                              loadingtime unloadingtime shinanchtime
                              shiparrival
24 DOUBLE                   : shtotmoves
25 REFERENCE TO SET         : shpt shfulltt shemptytt waitinglist xrtgcwimp
                              xrtgcwexp
26 REFERENCE TO importgen   : shimportgen
27
28 ATTRIBUTES OF shipgenerator:
29 REFERENCE TO exportgen   : shexportgen
30
31 ATTRIBUTES OF importgen  :
32 INTEGER                  : k l p q ignrtrans igimp20 igimp40 igimp20f
                              igimp20mt igimp40f igimp40mt igtransnr igimprow
                              igtrucks igtrucknr
33 REAL                     : r8 igdayimp presentimptime
34 REFERENCE TO SET         : trimplst
35 REFERENCE TO ship        : igjob
36 REFERENCE TO transporter: igimptransp
37
38 ATTRIBUTES OF exportgen  :
39 INTEGER                  : egnrtruck egexp20 egexp40 egexp20f egexp20mt
                              egexp40f egexp40mt i j m o egtrucknr egtrucks
40 REAL                     : r6 r7 egdayexp presentexptime
41 REFERENCE TO SET         : trexplst
42 REFERENCE TO ship        : egjob
43 REFERENCE TO transporter: egtransp
44
```

2. Description model ICT_1

2.1 Module DEFINE

```
ATTRIBUTES OF tlyard       :
LOGICAL                    : trlighty
REFERENCE TO transporter: tlytt

ATTRIBUTES OF tlquay       :
LOGICAL                    : trlightq
REFERENCE TO transporter: tlqtt

ATTRIBUTES OF tlgate       :
LOGICAL                    : trlightg
REFERENCE TO transporter: tlgtt

ATTRIBUTES OF portainer :
INTEGER                    : ptnr ptgoal ptloc ptbox ptteu
REAL                       : timept pttime1 pttime2 pttottime pttot2time
                             ptlasttime tmqlasttime
DOUBLE                     : ptnrbox
LOGICAL                    : ptunloading ptshiftprocedure ptend ptmove
CHARACTER(3)               : ptload
CHARACTER(13)              : ptdestination
REFERENCE TO SET           : ttofpt ttunderpt ttimportpt ttexportpt loadtt
                             unloadtt
REFERENCE TO ship          : shiponhand departship
REFERENCE TO transporter: ptimptt ptexptt
REFERENCE TO rtgc          : xrtgcpt

ATTRIBUTES OF rtgc         :
INTEGER                    : row rtgcnr
REAL                       : rtgcrolltime
LOGICAL                    : xrtgcstop
CHARACTER(5)               : rtgctype
REFERENCE TO transporter: rtgcserveunit
REFERENCE TO ship          : rtgcship
MACRO                      : rtgcstrategy

ATTRIBUTES OF transporter:
INTEGER                    : trbox trteu trrow trbay
REAL                       : trquaytime truckat trinrtgcqtime ttupttime
                             ttinrtgcqtime trarrival trqueuetime
LOGICAL                    : trendprocedure ttwait
CHARACTER(3)               : trload
CHARACTER(5)               : trtype
CHARACTER(13)              : trdestination
REFERENCE TO ship          : trworkship
REFERENCE TO portainer     : trworkpt ptfirst

ATTRIBUTES OF berthmaster:
REFERENCE TO ship          : nextship

ATTRIBUTES OF ptmaster     :
INTEGER                    : spnr
REAL                       : timeatberth[3]
DOUBLE                     : totmoves
REFERENCE TO portainer     : pmpt
REFERENCE TO rtgc          : xrtgc

ATTRIBUTES OF ttmaster     :
INTEGER                    : n
REFERENCE TO portainer     : tmpt
REFERENCE TO transporter : tmttforpt

ATTRIBUTES OF trmasterquay    :
REFERENCE TO transporter : tmqtt
REFERENCE TO portainer     : tmqpt
REFERENCE TO ship          : tmqship

ATTRIBUTES OF trmasteryard    :
REAL                       : r10
REFERENCE TO transporter : tmytt
REFERENCE TO portainer     : tmypt
REFERENCE TO ship          : tmyship
REFERENCE TO rtgc          : xrtgcrow10

ATTRIBUTES OF truckgatemaster    :
INTEGER                    : tgmrow
LOGICAL                    : tgmrowopen[10]
REFERENCE TO transporter : tgmtruck
REFERENCE TO SET           : tgmtruckq[10] leavingtrucks

```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ initial values and run parameters input @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

begin:
WRITE "Do you want to specify the input (y/n)?" WITH IMAGE
    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
specification<CHREAD
IF "n" = specification
   CALL inputblock
   GOTO inputend
END
GOTO inputstart IF "y"=specification
WRITE "you must answer (y)es or (n)o !!" WITH IMAGE
    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
REPEAT FROM begin

inputstart:
WRITE "please fill in the following information" WITH IMAGE
    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
cor1:
WRITE "Date [dd-mm-yyyy]: " WITH IMAGE
    xxxxxxxxxxxxxxxxxxx
date<CHREAD
cor2:
@WRITE "Length of simulation run [0 -365 days]: " WITH IMAGE
     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
@runtime<24*READ
@IF (runtime=0) | (runtime>8760)
@   WRITE "simulation time is 0 or longer than 1 year" WITH IMAGE
         xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
@   WRITE "please correct the runlength" WITH IMAGE
         xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
@   REPEAT FROM cor2
@END
cor3:
WRITE "Do tt's have priority in the yard (y/n)? " WITH IMAGE
    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
priority<CHREAD
GOTO modvar IF ("y"=priority) | ("n"=priority)
WRITE "you must answer (y)es or (n)o !!" WITH IMAGE
    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
REPEAT FROM cor3
modvar:
WRITE "-----------------------------------------------" WITH IMAGE
    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
WRITE "and now the determination of the model variables" WITH IMAGE
    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
WRITE "The number of portainers used: " WITH IMAGE
    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
nrpt<READ
WRITE "The net operation time per vessel: " WITH IMAGE
    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
netoperation<READ
WRITE "Write the number of tt's per portainer: " WITH IMAGE
    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
ttperpt<READ
nrtt<ttperpt*nrpt
WRITE "The number of yardcranes used: " WITH IMAGE
    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
nrrtgc<READ
WRITE "Maximum speed for transporters in the yard [km/h]: " WITH IMAGE
    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
mainlanespeed<(1000/3600)*READ
WRITE "Maximum speed for tt's on the bridge [km/h]: " WITH IMAGE
    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
bridgespeed<(1000/3600)*READ
WRITE "Maximum speed for tt's on the quay [km/h]: " WITH IMAGE
    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
quayspeed<(1000/3600)*READ
WRITE "The maximum relocation speed for rtgc's [km/h]: " WITH IMAGE
    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
rtgcspeed<(1000/3600)*READ
cor4:
WRITE "Do you want to change anything (y/n)? " WITH IMAGE
    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
correction<CHREAD
REPEAT FROM cor1 IF "y"=correction
REPEAT FROM cor4 IF "n"≠correction

inputend:

```

2.2 Module MAIN

This module is present in every computer simulation model created in the language PROSIM. The MAIN module takes care of the run control. A simulation run will always start in this module.

[5-14]:
In this section of the module the user is asked whether he/she wants to specify the input manually. If so, the program will ask the user various questions (see lines 16-61 below). If not, the program will activate the macro INPUTBLOCK in which all the input variables the program requires, are read from a predefined input file. If there is no such file the program will not be able to complete its simulation run.

specification	Character attribute of MAIN: defines whether the input data will be specified by the user or is to be attained from an input file
inputblock	Macro: defines the values of various variables that the computer model requires

[16-61]:
This section is followed in the case the user wishes to specify the input data manually and not via an input file. The user will receive on the monitor a variety of questions that are to be answered within certain boundaries.

date	Character attribute of MAIN: defines the present date
runtime	Real attribute of MAIN: defines the length of the simulation run in hours
priority	Character attribute of MAIN: defines which servicing strategy the RTGC's are to follow in the yard
nrpt	Integer attribute of MAIN: total number of portainers present at the terminal
netoperation	Real attribute of MAIN: value of the to be used *influence factor* in the runs (see also the report chapter 5.5.3 for specifics of this variable)
ttperpt	Integer attribute of MAIN: number of tractor-trailers (tt's) to be assigned per portainer
nrtt	Integer attribute of MAIN: total number of tt at the terminal (product of nrpt and ttperpt)
nrrtgc	Integer attribute of MAIN: total number of RTGC's present at the terminal
mainlanespeed	Real attribute of MAIN: driving speed of the tt's and trucks in the container yard
bridgespeed	Real attribute of MAIN: driving speed of the tt's on the bridge
quayspeed	Real attribute of MAIN: driving speed of the tt's on the quay
rtgcspeed	Real attribute of MAIN: driving speed of the RTGC's
correction	Character attribute of MAIN: defines whether the user wishes to make any changes in the input

```
@@@@@@@@@@@@@@@@
@ set creation @
@@@@@@@@@@@@@@@@

sea<NEW SET CALLED "sea"
channelin<NEW SET CALLED "channel in"
channelout<NEW SET CALLED "channel out"
shiftingpt<NEW SET CALLED "shifting pt's"
noworkrtgc<NEW SET CALLED "not working rtgc's"
maintraffic<NEW SET CALLED "main traffic"
intraffic<NEW SET CALLED "in traffic"
outtraffic<NEW SET CALLED "out traffic"
quaytoyard<NEW SET WITH STATISTICS CALLED "quay to yard"
yardtoquay<NEW SET WITH STATISTICS CALLED "yard to quay"
importpt<NEW SET CALLED "importpt"
exportpt<NEW SET CALLED "exportpt"
xrtgcworkset<NEW SET CALLED "xrtgc10wset"
quayintraffic<NEW SET CALLED "quayintraffic"
quayouttraffic<NEW SET CALLED "quayouttraffic"
gatequeue<NEW SET WITH STATISTICS CALLED "gatequeue"

@@@@@@@@@@@@@@@@@@@@@@@@@@
@ CREATION OF PORTAINERS @
@@@@@@@@@@@@@@@@@@@@@@@@@@

FOR b<1 TO nrpt
   THIS portainer<NEW portainer
   ptnr<b
   ptloc<b
   MOVE ptatquay[ptloc] TO 1    @anim@
   ptgoal<b
   timept<0
   shiponhand<NONE
   ttunderpt<NEW SET WITH STATISTICS CALLED "underpt"
   ttofpt<NEW SET CALLED "ttofpt"
   ttimportpt<NEW SET CALLED "ttimportpt"
   ttexportpt<NEW SET CALLED "ttexportpt"
   loadtt<NEW SET CALLED "loadingtt"
   unloadtt<NEW SET CALLED "unloadingtt"
   ptunloading<TRUE
   ptshiftprocedure<FALSE
   ptend<FALSE
   ptmove<FALSE
   JOIN THIS portainer TO passivept
   JOIN THIS portainer TO ptset
END
portaineractive[1]<FALSE
portaineractive[2]<FALSE
portaineractive[3]<FALSE
portaineractive[4]<FALSE
nr<0

@@@@@@@@@@@@@@@@@@@@
@ berth occupation @
@@@@@@@@@@@@@@@@@@@@

reservation[1]<FALSE
reservation[2]<FALSE
reservation[3]<FALSE
occupied[1]<FALSE
occupied[2]<FALSE
occupied[3]<FALSE
shipsatquay<0
timeatberth[1]<0
timeatberth[2]<0
timeatberth[3]<0
placeocc[1]<TRUE
placeocc[2]<TRUE
placeocc[3]<TRUE
placeocc[4]<TRUE
placeocc[5]<FALSE
placeocc[6]<FALSE

@@@@@@@@@@@@@@@@@@@@@@@@@@
@ INITIALIZE GATE MASTER @
@@@@@@@@@@@@@@@@@@@@@@@@@@

leavingtrucks<NEW SET CALLED "leavtruc"
FOR e<1 TO 10
   tgmrowopen[e]<TRUE
   tgmtruckq[e]<NEW SET
END

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ CREATION OF TRACTOR TRAILERS @
```

[63-82]:
In this section various sets are created. These sets are sets with which all the movements and locations of the components during the simulation run can be followed.

sea	Set attribute of class component ship: set which every ship is joined to before arriving at the anchorage and after she has left the terminal
channelin	Set attribute of MAIN: set to which a ship is joined when leaving the anchorage for the quay
channelout	Set attribute of MAIN: contains ships bound for sea
shiftingpt	Set attribute of MAIN: contains the portainers changing their position
noworkrtgc	Set attribute of MAIN: set to which the passive RTGC's are joined
maintraffic	Set attribute of MAIN: set to which tt's and trucks are joined when driving north along the first stack
intraffic	Set attribute of MAIN: set to which the tt's and trucks driving north of the stacks are joined to
outtraffic	Set attribute of MAIN: set to which tractor-trailers and trucks coming from the stack and driving south of the stacks are joined to
quaytoyard	Set attribute of MAIN: set to which all the yard bound tt's driving on the bridge are joined to
yardtoquay	Set attribute of MAIN: set to which the quay bound tt's driving on the bridge are joined to
importpt	Set attribute of MAIN: contains all portainers unloading
exportpt	Set attribute of MAIN: contains all portainers loading
xrtgcworkset	Set attribute of MAIN: set to which available extra RTGC's are joined
quayintraffic	Set attribute of MAIN: set to which all the portainer bound tt's driving on the quay are joined to
quayouttraffic	Set attribute of MAIN: set to which the tt's are joined to when driving from the portainer to the bridge
gatequeue	Set attribute of MAIN: set to which trucks are joined to upon arrival at the gate

[84-113]:
This section of the module concerns the creation of the portainers and the initialisation of some of their attributes.

ptnr	Integer attribute of portainer: number of portainer
ptloc	Integer attribute of portainer: present portainer position
ptgoal	Integer attribute of portainer: represents the destination of the portainer (portainer position)
timept	Real attribute of portainer: time at which the last tt was sent to a portainer
shiponhand	Reference to class component ship: ship portainer is assigned to
ttunderpt	Set attribute of portainer: contains the tractor-trailers waiting to be served by the portainer
ttofpt	Set attribute of portainer: set to which the tt's are joined that are activated on behalf of this portainer
ttimportpt	Set attribute of portainer: contains all the tt's on the quay waiting to be serviced by the portainer
ttexportpt	Set attribute of portainer: contains all the tt's on the quay waiting to be serviced by the portainer
loadtt	Set attribute of portainer: contains the tt presently being loaded by the portainer
unloadtt	Set attribute of portainer: contains the tt presently being unloaded by the portainer
ptunloading	Logical attribute of portainer: defines whether or not a portainer is in the process of unloading a ship
ptshiftprocedure	Logical attribute of portainer: defines whether the portainer is instructed to stop its actions and shift to another position at the quay
ptend	Logical attribute of portainer: defines if portainer is to become passive

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

FOR c<1 TO nrtt
   THIS transporter<NEW transporter
   trtype<"tt"
   trbox<0
   trteu<0
   trrow<0
   trworkpt<NONE
   ptfirst<NONE
   trworkship<NONE
   trendprocedure<FALSE
   ttwait<FALSE
   trdestination<"importquay"
   JOIN THIS transporter TO ttavailable
   MOVE ttparking TO LENGTH OF ttavailable    @anim@
END

@@@@@@@@@@@@@@@@@@@@@@@
@ CREATION OF RTGC'S @
@@@@@@@@@@@@@@@@@@@@@@@

FOR dd<1 TO 2
   FOR d<1 TO nrrtgc
      THIS rtgc<NEW rtgc
      rtgcnr<d
      rtgctype<"nrtgc" IF dd=1
      rtgctype<"xrtgc" IF dd=2
      rtgcstrategy<ttpriority IF ("y" = priority)
      rtgcstrategy<nopriority IF ("n" = priority)
      JOIN THIS rtgc TO rtgcset IF dd=1
      JOIN THIS rtgc TO xrtgcset IF dd=2
      ACTIVATE THIS rtgc FROM rolltolocation IN rtgcprocess IF dd=1
   END
END

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ driving speeds and sailing times @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

totteu<0
shsailtime<0 HOURS
shdeparttime<20 MINUTES
ptshifttime<15 MINUTES
quayspace<510
bridgetime<(1800/bridgespeed) SECONDS
mainlanetime<(440/mainlanespeed) SECONDS
berth1time<(125/quayspeed) SECONDS
berth2time<(300/quayspeed) SECONDS
berth3time<(425/quayspeed) SECONDS
x<5
w<0
tringate<0
troutgate<0
landboxin<0
landboxout<0
seaboxin<0
seaboxout<0

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ seeds for random number generation @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

SEED OF unif<READ FROM seedlist
SEED OF ptliftontime<READ FROM seedlist
SEED OF ptliftofftime<READ FROM seedlist
SEED OF ptliftcycle<READ FROM seedlist
SEED OF rtgcdrivetime<READ FROM seedlist
SEED OF rtgcliftontime<READ FROM seedlist
SEED OF rtgcliftofftime<READ FROM seedlist
SEED OF rtgcrepostime<READ FROM seedlist
SEED OF rtgcreshuftime<READ FROM seedlist

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ reshaping of distributions @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

RESHAPE unif AS SAMPLED FROM DISTRIBUTION UNIFORM
RESHAPE ptliftontime AS SAMPLED FROM DISTRIBUTION UNIFORM WITH PARAMETERS      L
   B((54/netoperation) SECONDS) UB((56/netoperation) SECONDS)
RESHAPE ptliftofftime AS SAMPLED FROM DISTRIBUTION UNIFORM WITH PARAMETERS     L
   B((54/netoperation) SECONDS) UB((56/netoperation) SECONDS)
RESHAPE ptliftcycle AS SAMPLED FROM DISTRIBUTION GAMMASHAPE WITH PARAMETERS    L
   B((99/netoperation) SECONDS) MEAN((109/netoperation) SECONDS) DEVIATION(5 SECOND
   S)
```

ptmove	Logical attribute of portainer: defines if portainer has to change its position
passivept	Queue: contains all the passive portainers
ptset	Queue: contains all the portainers
portaineractive[nr]	Logical attribute of MAIN: defines if portainer is active or not

[115-133]:
This section concerns the initialisation of various attributes.

reservation	Logical attribute of MAIN: defines if a berthing place is to be occupied
occupied[ ]	Logical attribute of MAIN: defines which berth place is occupied
shipsatquay	Integer attribute of MAIN: number of ships at the quay
timeatberth[ ]	Real attribute of ptmaster: time the ship has been moored
placeocc[ ]	Logical attribute of MAIN: defines if portainer position is occupied

[135-143]:
This section concerns various gate master attributes.

leavingtrucks	Set attribute of truckgatemaster: set to which trucks are joined to when leaving the yard
tgmrowopen[ ]	Logical attribute of Truckgatemaster: defines if stack is open for trucks to deliver or pick-up their load
tgmtruckq[ ]	Set attribute of truckgatemaster: number of trucks present in that particular row in the yard

[145-163]:
This section concerns the creation of the tractor-trailers and the initialisation of some of their attributes.

trtype	Character attribute of transporter: defines type of transporter (tt or truck)
trbox	Integer attribute of tranporter: number of boxes on the transporter
trteu	Integer attribute of transporter: TEU of the total transporter load
trrow	Integer attribute of transporter: number of row
trworkpt	Reference to portainer: portainer the tt is to drive to
ptfirst	Reference to portainer: portainer the tt is originally assigned to
trworkship	Reference to ship: the ship where the tt is sent to (tmqship)
trendprocedure	Logical attribute of transporter: defines if the tt is to return to the tt parking lot
ttwait	Logical attribute of transporter: defines if a tt has to wait on the quay before continuing its drive to a portainer
trdestination	Character attribute of tranporter: defines the destination of the transporter (depends on its load)
ttavailable	Queue: contains the tt's available to be assigned to a portainer

[165-183]:
In these lines the RTGC's are created and given attributes.

rtgcnr	Integer attribute of RTGC: number of RTGC
rtgctype	Character attribute of RTGC: defines if the RTGC is the second RTGC or the first RTGC operating in the stack
rtgcstrategy	Macro: defines the servicing strategy of the RTGC's
rtgcset	Queue: contains the available RTGC's
xrtgcset	Queue: contains the available second RTGC that can be assigned to a stack

```
RESHAPE rtgcdrivetime AS SAMPLED FROM DISTRIBUTION GAMMASHAPE WITH          P
ARAMETERS LB(1.2 SECONDS) MEAN(11 SECONDS) DEVIATION(9 SECONDS)
RESHAPE rtgcreshuftime AS SAMPLED FROM DISTRIBUTION GAMMASHAPE WITH         P
ARAMETERS LB(3.9 SECONDS) MEAN(7.8 SECONDS) DEVIATION(2.7 SECONDS)
RESHAPE rtgcrepostime AS SAMPLED FROM DISTRIBUTION GAMMASHAPE WITH          P
ARAMETERS LB(17 SECONDS) MEAN(51 SECONDS) DEVIATION(17 SECONDS)
RESHAPE rtgcliftontime AS SAMPLED FROM DISTRIBUTION GAMMASHAPE WITH         P
ARAMETERS LB(17 SECONDS) MEAN(51 SECONDS) DEVIATION(17 SECONDS)
RESHAPE rtgcliftofftime AS SAMPLED FROM DISTRIBUTION GAMMASHAPE WITH        P
ARAMETERS LB(17 SECONDS) MEAN(51 SECONDS) DEVIATION(17 SECONDS)

@@@@@@@@@@@@@@@@
@ truck table @
@@@@@@@@@@@@@@@@

nrtab[1]<4
nrtab[2]<6
nrtab[3]<7
nrtab[4]<8
nrtab[5]<12
nrtab[6]<16
nrtab[7]<17
nrtab[8]<18
nrtab[9]<20
nrtab[10]<24

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ start simulation by activating the ship generator @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

ACTIVATE shipgenerator FROM generate IN shipgeneratorprocess

@@@@@@@@@@@@@@@@@@@
@ end simulation @
@@@@@@@@@@@@@@@@@@@

WAIT
endrun:
WAIT 24 HOURS
CANCEL ALL
CALL output

TERMINATE
```

[185-205]:
This section concerns the initialisation of various attributes.

totteu	Real attribute of MAIN: total number of TEU exchanged
shsailtime	Real attribute of MAIN: time it takes for the ship to reach the quay once it has left the anchorage
shdeparttime	Real attribute of MAIN: time it takes for a ship to leave the quay and start its voyage to sea
ptshifttime	Real attribute of MAIN: time needed for a portainer to change its position
quayspace	Real attribute of MAIN: length of quay not occupied
bridgetime	Real attribute of MAIN: time needed to drive over the bridge
mainlanetime	Real attribute of MAIN: time needed to drive the length of a stack
berth1time	Real attribute of MAIN: time a tt needs to drive on the quay to reach the first vessel berthed
berth2time	Real attribute of MAIN: time a tt needs to drive on the quay to reach the second vessel berthed
berth3time	Real attribute of MAIN: time a tt needs to drive on the quay to reach the third vessel berthed
tringate	Real attribute of MAIN: number of trucks that have entered the yard
troutgate	Real attribute of MAIN: number of trucks that have left the yard
landboxin	Real attribute of MAIN: number of boxes that have entered the yard coming from the landside
landboxout	Real attribute of MAIN: number of boxes that have left the yard by truck
seaboxin	Real attribute of MAIN: total number of boxes arrived at the terminal from sea
seaboxout	Real attribute of MAIN: total number of boxes leaving the terminal by sea

[207-233]:
In this section of the module the *randomstreams* are determined. The seeds for the randomstreams are read from an input file called *seedlist*.

seedlist	Inputstream
unif	Randomstream: random number from uniform distribution
ptliftontime	Randomstream: time to perform the *lift-on* action as described in the report chapter 5.3.2
ptliftofftime	Randomstream: time to perform the *lift-off* action as described in the report chapter 5.3.2
ptliftcycle	Randomstream: time to perform the *lift-cycle* as described in the report chapter 5.3.2
rtgcdrivetime	Randomstream: time needed to drive to the correct position (report chapter 5.3.3)
rtgcliftontime	Randomstream: time needed to lift a container onto the transporter (report chapter 5.3.3)
rtgcliftofftime	Randomstream: time needed to lift a container off of a transporter (report chapter 5.3.3)
rtgcrepostime	Randomstream: time needed for repositioning RTGC (report chapter 5.3.3)
rtgcreshuftime	Randomstream: time needed to reshuffle the stack (report chapter 5.3.3)

[235-247]:
This section contains integer values that are needed for the creation of a graph out of which data is retrieved.

nrtab[ ]	Integer attribute of MAIN: point in the graph

[249-265]:
In these lines the module SHIPGENERATORPROCESS is activated. The MAIN module subsequently becomes passive until reactivated by the SHIPPROCESS module when the last vessel leaves the quay. Before terminating the simulation run the MAIN module activates the macro output in which certain output data is written.

output	Macro: contains simulation run output

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ ship generator process @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@
generate:
FOR a<1 TO 992
THIS ship<NEW ship
nrofship OF THIS ship<a
CALL shipattributes
shpt<NEW SET CALLED "shpt"
shfulltt<NEW SET CALLED "shfulltt"
shemptytt<NEW SET CALLED "shemptytt"
xrtgcwimp<NEW SET CALLED "xrtgcimport"
xrtgcwexp<NEW SET CALLED "xrtgcexport"
nrrtgcw OF THIS ship<1
loadingtime OF THIS ship<0
unloadingtime OF THIS ship<0
CALL expgen_gen
JOIN THIS ship TO sea
ACTIVATE THIS ship WITH DELAY 8 DAYS FROM arrival IN shipprocess
WAIT shinterarrivaltime
END
TERMINATE
```

2.3 Module SHIPGENERATOR

[5-26]:
This module is generated by the module MAIN from the label *generate*. A number of ships (in this case 992 vessels) are created in this module. After the creation of a new ship the following actions are performed:

- The new ship receives a ship number;
- The ship attributes are determined by the macro SHIPATTRIBUTES;
- Various sets are created belonging to this specific ship;
- The macro EXPGEN_GEN is activated. In this macro the exact number of export containers (the number of 20 ft, 40 ft, full and empty containers) needed for this ship is determined;
- This ship is activated in its module SHIPPROCESS with a delay of eight days;
- The SHIPGENERATOR waits a certain *shinterarrivaltime* time. This waiting time is an attribute of MAIN determined in the macro SHIPATTRIBUTES above.

If the total number of ships to be created has not yet been reached, the shipgenerator will repeat its actions for the next ship. Otherwise it will be terminated.

a	Integer attribute of MAIN: parameter
nrofship	Integer attribute of class component ship: number of ship
shipattributes	Macro: determines characteristics of a ship
shemptytt	Set attribute of class component ship: set to which tractor-trailers that are on their way to pick-up an empty container in the yard are joined
shpt	Set attribute of class component ship: set to which portainers servicing the ship are joined
shfulltt	Set attribute of class component ship: set to which tractor-trailers that are on their way to pick-up a full container in the yard, are joined
xrtgcwimp	Set attribute of class component ship: if a second RTGC is active in the import stack of this ship, it is joined to this stack
xrtgcwexp	Set attribute of class component ship: if a second RTGC is active in the export stack of this ship, it is joined to this stack
nrrtgcw	Integer attribute of class component ship: total number of RTGC active for this ship
loadingtime	Real attribute of class component ship: time needed for loading the ship
unloadingtime	Real attribute of class component ship: time needed for unloading the ship
EXPGEN_GEN	Macro: determines the export container specifics
shinterarrivaltime	Real attribute of MAIN: the time between the creation of this ship and the next ship

MODEL ICT 1
MAC SHIPATTRIBUTES

Date: 98/09/09
Time: 11:24:41

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ assigning ship attributes for ship generator @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

shipnumber:
shnr<READ FROM realinv

shiplength:
shlength<READ FROM realinv
shinterarrivaltime<READ FROM realinv

shipexportcontainerbay:
w<w+1
shexportbay<w
w<0 IF w=5

shipimportcontainerbay:
x<x+1
shimportbay<x
x<5 IF x=9

containers:
shtotmoves<READ FROM realinv

exportcontainers:
shexp20f< READ FROM realinv
shexp20mt< READ FROM realinv
shexp40f< READ FROM realinv
shexp40mt< READ FROM realinv

importcontainers:
shimp20f< READ FROM realinv
shimp40f< READ FROM realinv
shimp20mt< READ FROM realinv
shimp40mt< READ FROM realinv

totaal:
shimp20<shimp20f+shimp20mt
shimp40<shimp40f+shimp40mt
shexp20<shexp20f+shexp20mt
shexp40<shexp40f+shexp40mt
shimportteu<shimp20+(2*shimp40)
shexportteu<shexp20+(2*shexp40)
shimportteuf<shimp20f+(2*shimp40f)
shexportteuf<shexp20f+(2*shexp40f)
shimportteumt<shimp20mt+(2*shimp40mt)
shexportteumt<shexp20mt+(2*shexp40mt)
shexchange<(shimportteu+shexportteu)
shreqpt<1 IF (shtotmoves≤30)
shreqpt<2 IF (shtotmoves>30)
shturnaroundtime<0
shservicetime<0

stack:
stackimp20<0
stackimp20f<0
stackimp20mt<0
stackimp40<0
stackimp40f<0
stackimp40mt<0
stackexp20<0
stackexp20f<0
stackexp20mt<0
stackexp40<0
stackexp40f<0
stackexp40mt<0

RETURN
```

2.4 Macro SHIPATTRIBUTES

[1-35]:
This macro is activated by the module SHIPGENERATOR and is responsible for assigning values to a number of ship attributes. The following integer attributes are read from the input file ***realinv***:

- *Shnr*: the number of the ship (obtained from the 1997 terminal records)
- *Shlength*: the length of the vessel
- *Shinterarrivaltime*: the inter-arrival time of this ship and the next ship to be created
- *Shtotmoves*: the total number of box moves to be made when this ship is moored
- The number of export containers and their type (size and load)
- The number of import containers and their type (size and load)

An import and export stack is assigned to this ship.

Integer attributes of class component ship:

shnr	Number of this ship which is the same number as in the terminal records of 1997
shlength	Vessel length
shexportbay	Number of the stack for export containers that is assigned to this ship
shimportbay	Number of the stack for import containers that is assigned to this ship
shexp20f	Total number of 20 ft, full export containers
shexp20mt	Total number of 20 ft, empty export containers
shexp40mt	Total number of 40 ft, empty export containers
shimp20f	Total number of 20 ft, full import containers
shimp40f	Total number of 40 ft, full import containers
shexp40f	Total number of 40 ft, full export containers
shimp20mt	Total number of 20 ft, empty import containers
shimp40mt	Total number of 40 ft, empty import containers
shtotmoves	Double attribute of class component ship: total number of box moves to be made

[37-52]:
Once the number of import and export containers (including their size and load) to be moved are known, the last integer attributes are determined.

Integer attribute of class component ship:

shimp20	Total number of 20 ft import containers
shimp40	Total number of 40 ft import containers
shexp20	Total number of 20 ft export containers
shexp40	Total number of 40 ft export containers
shimportteu	Total number of import containers (TEU)
shexportteu	Total number of export containers (TEU)
shimportteuf	Total number of full import containers (TEU)
shexportteuf	Total number of full export containers (TEU)
shimportteumt	Total number of empty import containers (TEU)
shexportteumt	Total number of empty export containers (TEU)
shexchange	Total number of containers (TEU) to be exchanged
shreqpt	Number of portainers to be requested by this ship
shturnaroundtime	Real attribute of class component ship: turn-around time of this ship
shservicetime	Real attribute of class component ship: inter-arrival time of this ship

[54-68]:
The number of containers that are in the stack are kept track of and the values of these integer attributes are set at zero at this point. Hereafter the macro returns to the module SHIPGENERATOR.

Integer attribute of class component ship:

stackimp20	Total number of 20 ft import containers in the assigned import stack
stackimp20f	Total number of 20 ft full import containers in the assigned import stack
stackimp20mt	Total number of 20 ft empty import containers in the assigned import stack
stackimp40	Total number of 40 ft import containers in the assigned import stack
stackimp40f	Total number of 40 ft full import containers in the assigned import stack
stackimp40mt	Total number of 40 ft empty import containers in the assigned import stack
stackexp20	Total number of 20 ft export containers in the assigned export stack
stackexp20f	Total number of 20 ft full export containers in the assigned export stack
stackexp20mt	Total number of 20 ft empty export containers in the assigned export stack
stackexp40	Total number of 40 ft export containers in the assigned export stack
stackexp40f	Total number of 40 ft full export containers in the assigned export stack
stackexp40mt	Total number of 40 ft empty export containers in the assigned export stack

```
@@@@@@@@@@@@@@@@@@
@ ship process @
@@@@@@@@@@@@@@@@@@

arrival:
LEAVE sea
ENTER anchorage
shinanchtime<NOW
MOVE anchfig TO LENGTH OF anchorage @anim@
ACTIVATE berthmaster FROM berthing IN berthmasterprocess
IF berthmaster IS NOT ACTIVE
WAIT
shiptotanchtime<shiptotanchtime + (NOW-shinanchtime)
LEAVE anchorage
MOVE anchfig TO LENGTH OF anchorage @anim@
timearrival<NOW
ENTER channelin
MOVE channelinfig TO LENGTH OF channelin @anim@
WORK shsailtime
LEAVE channelin
MOVE channelinfig TO LENGTH OF channelin @anim@
ENTER quay
ENTER berth[shberthnr]
iattime<NOW-shiparrivaltime IF shnr‡1
iattimetot<iattimetot + iattime IF shnr‡1
WRITE iattime TO iatlist WITH IMAGE xxx.xx IF shnr‡1
STORE iattime AS "iat" IF shnr‡1
shiparrivaltime<NOW
shiparrival<NOW
MOVE berthocc[shberthnr] TO LENGTH OF berth[shberthnr] @anim@
shservicetime<NOW
WAIT 1 HOUR
occupied[shberthnr]<TRUE
shipsatquay<shipsatquay+1
ENTER cranereqlst
ACTIVATE ptmaster FROM reqlstcheck IN ptmasterprocess IF ptmaster IS NOT
ACTIVE
CALL impgen_gen
WAIT

ENTER unloadingship
tgmrowopen[shimportbay]<FALSE
MOVE rowopenfig[shimportbay] TO 1          @anim@
unloadingtime<NOW
WAIT WHILE shimportteu‡0
LEAVE unloadingship
tgmrowopen[shimportbay]<TRUE
MOVE rowopenfig[shimportbay] TO SINK       @anim@

ENTER loadingship
tgmrowopen[shexportbay]<FALSE
MOVE rowopenfig[shexportbay] TO 1          @anim@
loadingtime<NOW
WAIT WHILE shexportteu‡0
LEAVE loadingship
tgmrowopen[shexportbay]<TRUE
MOVE rowopenfig[shexportbay] TO SINK       @anim@

quitberth:
shipsatquay<shipsatquay-1
occupied[shberthnr]<FALSE
WAIT 1 HOUR
shservicetime<(NOW-shservicetime)
STORE shservicetime AS "sst"
shiptotstime<shiptotstime+shservicetime
LEAVE cranereqlst IF BELONGS TO cranereqlst
ENTER channelout
REACTIVATE MAIN FROM endrun IN MAINMOD IF nrofship=992
shoccberth[shberthnr]<shoccberth[shberthnr]+(NOW-shiparrival)
WORK shdeparttime
LEAVE quay
LEAVE berth[shberthnr]
LEAVE channelout
MOVE berthocc[shberthnr] TO LENGTH OF berth[shberthnr] @anim@
reservation[shberthnr]<FALSE
quayspace<(quayspace+shlength+10) IF shberthnr‡3
quayspace<(quayspace+shlength) IF shberthnr=3
ACTIVATE berthmaster FROM berthing IN berthmasterprocess
IF berthmaster IS NOT ACTIVE
shturnaroundtime<(NOW - timearrival)
ENTER sea
totteu<totteu+shexchange

LEAVE sea
TERMINATE
```

2.5 Module SHIPPROCESS

This module is activated by the module SHIPGENERATOR and concerns the ships actions.

[5-11]:
After the ship leaves the set *sea* and enters the queue *anchorage* the module BERTHMASTER is activated. The ship waits in the *anchorage* until she is reactivated by the BERTHMASTER.

anchorage	Queue: contains all the ships waiting in the anchorage to be serviced at the quay
shinanchtime	Real attribute of ship: time the ship enters the queue anchorage

[12-37]:
When the ship is reactivated it will leave the *anchorage* and enter the set *channelin*. After sailing a *shsailtime* the ship leaves *channelin* and will enter the queues *quay* and *berth*. At this point the inter-arrival time between this ship arrival and the previous ship arrival is calculated. After a one hour waiting period, during which it is assumed that mooring activities are completed, the ship will join the queue *cranereqlst* and the module PTMASTER is activated. Before actual servicing of the ship starts, also the macro IMPGEN_GEN is activated.

shiptotanchtime	Real attribute of MAIN: total time the ships spend in the anchorage
timearrival	Real attribute of ship: time of ship arrival at the quay
quay	Queue: contains all the ships moored
berth[shberthnr]	Queue (3): contains a ship to be serviced
iattime	Real attribute of MAIN: inter-arrival time between this ship arriving at the quay and the last ship
iattimetot	Real attribute of MAIN: total of inter-arrival times of the ships
shiparrival	Real attribute of ship: time the ship arrives at the quay
cranereqlst	Queue: contains all the ships requesting for a portainer
IMPGEN_GEN	Macro: determines the import container specifics

[39-46]:
After the ship is reactivated by the PTMASTER she will enter the queue *unloadingship* until the ship has been completely unloaded.

unloadingship	Queue: contains the ships that are being unloaded
unloadingtime	Real attribute of ship: time unloading of ship starts

[48-55]:
The ship now joins the queue *loadingship* where she stays until she has been completely loaded.

loadingship	Queue: contains the ships that are being loaded
loadingtime	Real attribute of ship: time loading of ship commences

[57-81]:
After the ship has left the queue *loadingship* she will wait one hour (time assumed to be required for deberthing activities) before leaving the queues *quay* and *berth*.
It will take a certain time (*shdeparttime*) for the ship to leave the berth. Before the ship enters the set *sea*, the module BERTHMASTER is activated again. After entering the set *sea* the ship leaves the set and is terminated.

shiptotstime	Real attribute of MAIN: total of all ship service times

MODEL ICT 1
MOD BERTHMASTERPROCESS

Date: 98/09/09
Time: 11:24:43

```
@@@@@@@@@@@@@@@@@@@@@@@@@
@ berth master process @
@@@@@@@@@@@@@@@@@@@@@@@@@

berthing:
IF anchorage IS NOT EMPTY
   nextship<FIRST ship IN anchorage
   GOTO quayfull IF shipsatquay=3
   IF (reservation[1]=FALSE)
      IF quayspace ≥ (shlength OF nextship+10)
         quayspace<quayspace-(shlength OF nextship+10)
         reservation[1]<TRUE
         shberthnr OF nextship<1
         GOTO updatespace
      END
      GOTO quayfull
   END
   IF (reservation[2]=FALSE)
      IF quayspace ≥ (shlength OF nextship+10)
         quayspace<quayspace-(shlength OF nextship+10)
         reservation[2]<TRUE
         shberthnr OF nextship<2
         GOTO updatespace
      END
      GOTO quayfull
   END
   IF (reservation[3]=FALSE)
      IF quayspace ≥ shlength OF nextship
         quayspace<quayspace-shlength OF nextship
         reservation[3]<TRUE
         shberthnr OF nextship<3
         GOTO updatespace
      END
   END
   GOTO quayfull
   updatespace:
   REACTIVATE nextship
   WAIT 5 MINUTES
   REPEAT FROM berthing
END

quayfull:
PASSIVATE
```

2.6 Module BERTHMASTERPROCESS

This module is activated by the module SHIPPROCESS and starts at the label *berthing*. This module checks to see whether there are ships in the anchorage that can sail to the quay for servicing.

[5-43]:
First of all the module checks if there are ships in the anchorage; if there aren't the BERTHMASTER becomes passive. If there are, the BERTHMASTER checks the number of ships already berthed. If there are 3 ships berthed or the available free quay space is insufficient for berthing the ship in the *anchorage*, the BERTHMASTER also becomes passive. Otherwise the ship is assigned a berthing place and this berth place is reserved and the available free quay space updated. The BERTHMASTER will first check the first berth and then the second and finally the third berth.

The BERTHMASTER then reactivates this ship and waits 5 minutes before repeating its actions from the label *berthing*.

nextship	Reference to class component ship: the ship that is to be served next
shberthnr	Integer attribute of ship: number of berth assigned to ship

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ assigning portainers to ships @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

reqlstcheck:

IF cranereqlst IS NOT EMPTY
   THIS ship<FIRST OF cranereqlst
   IF occupied[shberthnr OF this ship]=FALSE
      REMOVE this ship from cranereqlst
      REPEAT FROM reqlstcheck
   END
END
IF shipsatquay=1
IF cranereqlst IS NOT EMPTY
   THIS ship<FIRST OF cranereqlst
   totmoves<(shexchange of this ship)
   IF (totmoves<30) & (shpt OF THIS ship IS NOT EMPTY)
      REMOVE THIS ship FROM cranereqlst
      shreqpt<0
      REPEAT FROM reqlstcheck
   END
   nr<1
   FOR spnr<1 TO 3
      IF shberthnr OF THIS ship=spnr
         pmpt<FIRST portainer IN ptset WITH ptnr=nr
         IF portaineractive[nr]=FALSE
            ptgoal OF pmpt<1 IF spnr=1
            ptgoal OF pmpt<3 IF spnr=2
            ptgoal OF pmpt<5 IF spnr=3
            CALL portainerstart
            shreqpt<shreqpt-1
            REMOVE THIS ship FROM cranereqlst IF shreqpt=0
            REPEAT FROM reqlstcheck
         END
         IF (totmoves<30) & (shpt OF THIS ship IS NOT EMPTY)
            REMOVE THIS ship FROM cranereqlst
            shreqpt<0
            REPEAT FROM reqlstcheck
         END
         nr<nr+1
         pmpt<FIRST portainer IN ptset WITH ptnr=nr
         IF portaineractive[nr]=FALSE
            ptgoal OF pmpt<2 IF spnr=1
            ptgoal OF pmpt<4 IF spnr=2
            ptgoal OF pmpt<6 IF spnr=3
            CALL portainerstart
            shreqpt<shreqpt-1
            REMOVE THIS ship FROM cranereqlst IF shreqpt=0
            REPEAT FROM reqlstcheck
         END
      END
      nr<3 IF (spnr=1)|(spnr=2)
   END
END
END

IF shipsatquay=2
IF cranereqlst IS NOT EMPTY
   THIS ship<FIRST OF cranereqlst
   totmoves<(shexchange of this ship)
   IF (totmoves<30) & (shpt OF THIS ship IS NOT EMPTY)
      REMOVE THIS ship FROM cranereqlst
      shreqpt<0
      REPEAT FROM reqlstcheck
   END
   goon:
   IF (occupied[1]=TRUE) & (occupied[2]=TRUE)
      nr<1
      FOR spnr<1 TO 2
         pmpt<FIRST portainer IN ptset WITH ptnr=nr
         IF shberthnr OF THIS ship=spnr
            IF (portaineractive[nr]=FALSE)
               WAIT WHILE LENGTH OF shpt OF THIS ship=2
               ptgoal OF pmpt<1 IF spnr=1
               ptgoal OF pmpt<3 IF spnr=2
               CALL portainerstart
               shreqpt<shreqpt-1
               REMOVE THIS ship FROM cranereqlst IF shreqpt=0
               REPEAT FROM reqlstcheck
            END
            if nr=1
```

2.7 Module PTMASTERPROCESS

In this module the PTMASTER assigns portainers to the ships. The PTMASTER is activated by the module SHIPPROCESS the moment a ship has reached its berth place. There are three situations possible at the quay:

1. One ship berthed ([14-56]);
2. Two ships berthed ([58-223]);
3. Three ships berthed ([225-502]).

[5-13]:
The module starts its actions at the label *reqlstcheck* and will first check the *cranereqlst*. If this list is empty the PTMASTER will become passive. If this list is not empty the first ship on the list is selected.

1. [14-56]:
This section of the PTMASTERPROCESS concerns the assignment of portainers to their correct location at the quay in the case one ship is berthed. Depending on the berth place of the ship two portainers are given a destination if they are not already servicing a ship. The destination they are given represents the location on the quay where they are expected to go to or be (there are 6 possible portainer locations at the quay). If the portainer is not active the macro PORTAINERSTART is activated. This macro is responsible for the defining of the various attributes and for activating the portainer.

nr	Integer attribute of MAIN: parameter representing the portainer number
spnr	Integer attribute of ptmaster: parameter representing the berth place
pmpt	Reference to portainer: portainer to be instructed
PORTAINERSTART	Macro: determines various attributes and activates the portainer

2. [58-223]:
In the case two ships are moored this section of the module will be followed. There are three possible quay situations:

a) Ships berthed at places one and two (lines [68-117]):
b) Ships berthed at places one and three (lines [118-171]);
c) Ships berthed at places two and three (lines [172-223]).

a) [68-117]:
This is the case when *occupied[1]* and *[2]* are both TRUE. First the portainer situation of the selected ship is checked. The first portainer to be servicing this ship is checked whether it is active. If not, this portainer receives a destination and the macro PORTAINERSTART is activated. If this portainer is already active it is checked whether it is stationed at the correct location at the quay. When this portainer is stationed at the wrong place it will get instructions to stop its actions as soon as possible and to wait for further instructions. If this portainer is stationed at the correct place the PTMASTER will check the situation of the second portainer to be assigned to the ship.

> Brief example:
> Assume selected ship is berthed at berth place one. The PTMASTER will first check portainer 1 on its whereabouts and after that portainer 2. These are to be stationed at portainer "positions one and two" (see also report chapter 5.5.1.6).

For the other two situations (b and c) the same principle is followed as described for a above.

```
                  goto goon1 if (ptloc of pmpt=1)|(ptgoal of pmpt=1)
               end
               if nr=3
                  goto goon1 if (ptloc of pmpt=3)|(ptgoal of pmpt=3)
               end
               ptshiftprocedure of pmpt<true
               goon1:
               IF (totmoves<30) & (shpt OF THIS ship IS NOT EMPTY)
                  REMOVE THIS ship FROM cranereqlst
                  shreqpt<0
                  REPEAT FROM reqlstcheck
               END
               nr<nr+1
               pmpt<FIRST portainer IN ptset WITH ptnr=nr
               IF portaineractive[nr]=FALSE
                  WAIT WHILE LENGTH OF shpt OF THIS ship=2
                  ptgoal OF pmpt<2 IF spnr=1
                  ptgoal OF pmpt<4 IF spnr=2
                  CALL portainerstart
                  shreqpt<shreqpt-1
                  REMOVE THIS ship FROM cranereqlst IF shreqpt=0
                  REPEAT FROM reqlstcheck
               END
               if nr=2
                  goto goon2 if (ptloc of pmpt=2)|(ptgoal of pmpt=2)
               end
               if nr=4
                  goto goon2 if (ptloc of pmpt=4)|(ptgoal of pmpt=4)
               end
               ptshiftprocedure of pmpt<true
            END
            goon2:
            nr<3
         END
      END
      IF (occupied[1]=TRUE) & (occupied[3]=TRUE)
         nr<1
         spnr<1
         start:
         IF shberthnr OF THIS ship=spnr
            pmpt<FIRST portainer IN ptset WITH ptnr=nr
            IF portaineractive[nr]=FALSE
               WAIT WHILE LENGTH OF shpt OF THIS ship=2
               ptgoal OF pmpt<1 IF spnr=1
               ptgoal OF pmpt<5 IF spnr=3
               CALL portainerstart
               shreqpt<shreqpt-1
               REMOVE THIS ship FROM cranereqlst IF shreqpt=0
               REPEAT FROM reqlstcheck
            END
            if nr=1
               goto goon3 if (ptloc of pmpt=1)|(ptgoal of pmpt=1)
            end
            if nr=3
               goto goon3 if (ptloc of pmpt=5)|(ptgoal of pmpt=5)
            end
            ptshiftprocedure of pmpt<true
            goon3:
            IF (totmoves<30) & (shpt OF THIS ship IS NOT EMPTY)
               REMOVE THIS ship FROM cranereqlst
               shreqpt<0
               REPEAT FROM reqlstcheck
            END
            nr<nr+1
            pmpt<FIRST portainer IN ptset WITH ptnr=nr
            IF portaineractive[nr]=FALSE
               WAIT WHILE LENGTH OF shpt OF THIS ship=2
               ptgoal OF pmpt<2 IF spnr=1
               ptgoal OF pmpt<6 IF spnr=3
               CALL portainerstart
               shreqpt<shreqpt-1
               REMOVE THIS ship FROM cranereqlst IF shreqpt=0
               REPEAT FROM reqlstcheck
            END
            if nr=2
               goto goon4 if (ptloc of pmpt=2)|(ptgoal of pmpt=2)
            end
            if nr=4
               goto goon4 if (ptloc of pmpt=6)|(ptgoal of pmpt=6)
            end
            ptshiftprocedure of pmpt<true
         END
         goon4:
         IF spnr=1
            nr<3
```

```
         spnr<3
         GOTO start
      END
   END
   IF (occupied[2]=TRUE) & (occupied[3]=TRUE)
      nr<1
      FOR spnr<2 TO 3
         pmpt<FIRST portainer IN ptset WITH ptnr=nr
         IF shberthnr OF THIS ship=spnr
            IF portaineractive[nr]=FALSE
               WAIT WHILE LENGTH OF shpt OF THIS ship=2
               ptgoal OF pmpt<3 IF spnr=2
               ptgoal OF pmpt<5 IF spnr=3
               CALL portainerstart
               shreqpt<shreqpt-1
               REMOVE THIS ship FROM cranereqlst IF shreqpt=0
               REPEAT FROM reqlstcheck
            END
            if nr=1
               goto goon5 if (ptloc of pmpt=3)|(ptgoal of pmpt=3)
            end
            if nr=3
               goto goon5 if (ptloc of pmpt=5)|(ptgoal of pmpt=5)
            end
            ptshiftprocedure of pmpt<true
            goon5:
            IF (totmoves<30) & (shpt OF THIS ship IS NOT EMPTY)
               REMOVE THIS ship FROM cranereqlst
               shreqpt<0
               REPEAT FROM reqlstcheck
            END
            nr<nr+1
            pmpt<FIRST portainer IN ptset WITH ptnr=nr
            IF portaineractive[nr]=FALSE
               WAIT WHILE LENGTH OF shpt OF THIS ship=2
               ptgoal OF pmpt<4 IF spnr=2
               ptgoal OF pmpt<6 IF spnr=3
               CALL portainerstart
               shreqpt<shreqpt-1
               REMOVE THIS ship FROM cranereqlst IF shreqpt=0
               REPEAT FROM reqlstcheck
            END
            if nr=2
               goto goon6 if (ptloc of pmpt=4)|(ptgoal of pmpt=4)
            end
            if nr=4
               goto goon6 if (ptloc of pmpt=6)|(ptgoal of pmpt=6)
            end
            ptshiftprocedure of pmpt<true
         END
         goon6:
         nr<3
      END
   END
END
END

Threeships:
IF shipsatquay=3

   THIS ship<FIRST ship IN quay WITH shberthnr=1
   timeatberth[1]<(NOW-shservicetime OF THIS ship)
   THIS ship<FIRST ship IN quay WITH shberthnr=2
   timeatberth[2]<(NOW-shservicetime OF THIS ship)
   THIS ship<FIRST ship IN quay WITH shberthnr=3
   timeatberth[3]<(NOW-shservicetime OF THIS ship)

IF ((timeatberth[1])>(timeatberth[2])) & ((timeatberth[1])>(timeatberth[3]))
   spnr<1
   THIS ship<FIRST ship IN quay WITH shberthnr=spnr
   totmoves<(shexchange of this ship)
   FOR nr<1 TO 2
      IF (totmoves<30) & (shpt OF THIS ship IS NOT EMPTY)
         REMOVE THIS ship FROM cranereqlst IF shreqpt OF this ship>0
         shreqpt<0
         GOTO go1
      END
      pmpt<FIRST portainer IN ptset WITH ptnr=nr
      IF portaineractive[nr]=FALSE
         WAIT WHILE LENGTH OF shpt OF THIS ship=2
         ptgoal OF pmpt<1 IF nr=1
         ptgoal OF pmpt<2 IF nr=2
         CALL portainerstart
         shreqpt<shreqpt-1
         REMOVE THIS ship FROM cranereqlst IF shreqpt=0
```

```
         GOTO go1
      END
      IF nr=1
         IF (ptloc of pmpt=1)&(ptmove of pmpt=TRUE)
            remove pmpt from shpt of shiponhand of pmpt
            join pmpt to passivept
            portaineractive[1]<FALSE
            ptmove of pmpt<FALSE
            cancel pmpt
         END
         goto go1 if (ptloc of pmpt=1)|(ptgoal of pmpt=1)
      END
      if nr=2
         IF (ptloc of pmpt=2)&(ptmove of pmpt=TRUE)
            remove pmpt from shpt of shiponhand of pmpt
            join pmpt to passivept
            portaineractive[2]<FALSE
            ptmove of pmpt<FALSE
            cancel pmpt
         END
         goto go1 if (ptloc of pmpt=2)|(ptgoal of pmpt=2)
      end
      ptshiftprocedure of pmpt<true
      go1:
      IF (totmoves<30) & (shpt OF THIS ship IS NOT EMPTY)
         REMOVE THIS ship FROM cranereqlst IF shreqpt OF this ship>0
         shreqpt<0
      END
   END
   nr<3
   FOR spnr<2 TO 3
      THIS ship<FIRST ship IN quay WITH shberthnr=spnr
      pmpt<FIRST portainer IN ptset WITH ptnr=nr
      IF portaineractive[nr]=FALSE
         WAIT WHILE LENGTH OF shpt OF THIS ship=2
         ptgoal of pmpt<3 if spnr=2
         ptgoal OF pmpt<6 IF spnr=3
         CALL portainerstart
         shreqpt<shreqpt-1
         REMOVE THIS ship FROM cranereqlst IF shreqpt=0
         GOTO go2
      END
      if nr=3
         IF (ptloc of pmpt=3)&(ptmove of pmpt=TRUE)
            remove pmpt from shpt of shiponhand of pmpt
            join pmpt to passivept
            portaineractive[3]<FALSE
            ptmove of pmpt<FALSE
            cancel pmpt
         END
         goto go2 if (ptloc of pmpt=3)|(ptgoal of pmpt=3)
      end
      if nr=4
         IF (ptloc of pmpt=6)&(ptmove of pmpt=TRUE)
            remove pmpt from shpt of shiponhand of pmpt
            join pmpt to passivept
            portaineractive[4]<FALSE
            ptmove of pmpt<FALSE
            cancel pmpt
         END
         goto go2 if (ptloc of pmpt=6)|(ptgoal of pmpt=6)
      end
      ptshiftprocedure of pmpt<true
      go2:
      nr<nr+1
   END
END
IF ((timeatberth[2])>(timeatberth[3])) & ((timeatberth[2])>(timeatberth[1]))
   spnr<1
   nr<1
   THIS ship<FIRST ship IN quay WITH shberthnr=spnr
   pmpt<FIRST portainer IN ptset WITH ptnr=nr
   IF portaineractive[nr]=FALSE
      WAIT WHILE LENGTH OF shpt OF THIS ship=2
      ptgoal OF pmpt<1
      CALL portainerstart
      shreqpt<shreqpt-1
      REMOVE THIS ship FROM cranereqlst IF shreqpt=0
      goto go3
   end
   IF (ptloc of pmpt=1)&(ptmove of pmpt=TRUE)
      remove pmpt from shpt of shiponhand of pmpt
      join pmpt to passivept
      portaineractive[1]<FALSE
      ptmove of pmpt<FALSE
```

3. [225-502]:
When three ships are moored this section of the module will be followed by the PTMASTER. As explained in the report the ship moored the longest time will be assigned two portainers while the other two ships are assigned one each. The following three situations can apply:

a) The ship at berth place one is assigned two portainers ([235-319]);
b) The ship at berth place two is assigned two portainers ([320-411]);
c) The ship at berth place three is assigned two portainers ([412-496]).

a) [235-319]:
Each portainer is checked (in ascending portainer number sequence) whether it is active and what its position at the quay is. If a portainer is stationed at the wrong position it will receive an order to stop its actions and wait for further instructions. If a portainer is inactive it will receive a destination and will be activated by the macro PORTAINERSTART that is activated. The specific portainer positions are explained in the report chapter 5.5.1.6.

Brief example: Assuming situation a: first portainer 1 is checked whether it is active and at its position. Subsequently portainers two, three and four are checked. Portainers one and two are assigned to the ship moored at berth place one. Portainer three is assigned to the ship moored at berth place two and portainer four to the ship moored at berth place three.

Situations b and c function in a similar way as described for situation a.

Once the PTMASTER has followed these lines, it will check whether there are any portainers passive. If there are, it will repeat its actions from line 225. If all the portainers are active, the PTMASTER becomes passive until reactivated.

```
      cancel pmpt
   END
   goto go3 if (ptloc of pmpt=1)|(ptgoal of pmpt=1)
   ptshiftprocedure of pmpt<true
   go3:
   spnr<2
   THIS ship<FIRST ship IN quay WITH shberthnr=spnr
   totmoves<(shexchange of this ship)
   FOR nr<2 TO 3
      IF (totmoves<30) & (shpt OF THIS ship IS NOT EMPTY)
         REMOVE THIS ship FROM cranereqlst IF shreqpt OF this ship>0
         shreqpt<0
         GOTO go4
      END
      pmpt<FIRST portainer IN ptset WITH ptnr=nr
      IF portaineractive[nr]=FALSE
         WAIT WHILE (LENGTH OF shpt=2)
         ptgoal OF pmpt<3 IF nr=2
         ptgoal OF pmpt<4 IF nr=3
         CALL portainerstart
         shreqpt<shreqpt-1
         REMOVE THIS ship FROM cranereqlst IF shreqpt=0
         GOTO go4
      end
      if nr=2
         IF (ptloc of pmpt=3)&(ptmove of pmpt=TRUE)
            remove pmpt from shpt of shiponhand of pmpt
            join pmpt to passivept
            portaineractive[2]<FALSE
            ptmove of pmpt<FALSE
            cancel pmpt
         END
         goto go4 if (ptloc of pmpt=3)|(ptgoal of pmpt=3)
      end
      if nr=3
         IF (ptloc of pmpt=4)&(ptmove of pmpt=TRUE)
            remove pmpt from shpt of shiponhand of pmpt
            join pmpt to passivept
            portaineractive[3]<FALSE
            ptmove of pmpt<FALSE
            cancel pmpt
         END
         goto go4 if (ptloc of pmpt=4)|(ptgoal of pmpt=4)
      end
      ptshiftprocedure of pmpt<true
      go4:
      IF (totmoves<30) & (shpt OF THIS ship IS NOT EMPTY)
         REMOVE THIS ship FROM cranereqlst IF shreqpt OF this ship>0
         shreqpt<0
      END
   END
   spnr<3
   THIS ship<FIRST ship IN quay WITH shberthnr=spnr
   nr<4
   pmpt<FIRST portainer IN ptset WITH ptnr=nr
   IF portaineractive[nr]=FALSE
      WAIT WHILE LENGTH OF shpt OF THIS ship=2
      ptgoal OF pmpt<6
      CALL portainerstart
      shreqpt<shreqpt-1
      REMOVE THIS ship FROM cranereqlst IF shreqpt=0
      GOTO go5
   END
   IF (ptloc of pmpt=6)&(ptmove of pmpt=TRUE)
      remove pmpt from shpt of shiponhand of pmpt
      join pmpt to passivept
      portaineractive[4]<FALSE
      ptmove of pmpt<FALSE
      cancel pmpt
   END
   goto go5 if (ptloc of pmpt=6)|(ptgoal of pmpt=6)
   ptshiftprocedure of pmpt<true
   go5:
END
IF ((timeatberth[3])>(timeatberth[2])) & ((timeatberth[3])>(timeatberth[1]))
   nr<1
   FOR spnr<1 TO 2
      THIS ship<FIRST ship IN quay WITH shberthnr=spnr
      pmpt<FIRST portainer IN ptset WITH ptnr=nr
      IF portaineractive[nr]=FALSE
         WAIT WHILE LENGTH OF shpt OF THIS ship=2
         ptgoal OF pmpt<1 IF spnr=1
         ptgoal of pmpt<4 if spnr=2
         CALL portainerstart
         shreqpt<shreqpt-1
```

```
         REMOVE THIS ship FROM cranereqlst IF shreqpt=0
         GOTO go6
      END
      if nr=1
         IF (ptloc of pmpt=1)&(ptmove of pmpt=TRUE)
            remove pmpt from shpt of shiponhand of pmpt
            join pmpt to passivept
            portaineractive[1]<FALSE
            ptmove of pmpt<FALSE
            cancel pmpt
         END
         goto go6 if (ptloc of pmpt=1)|(ptgoal of pmpt=1)
      end
      if nr=2
         IF (ptloc of pmpt=4)&(ptmove of pmpt=TRUE)
            remove pmpt from shpt of shiponhand of pmpt
            join pmpt to passivept
            portaineractive[2]<FALSE
            ptmove of pmpt<FALSE
            cancel pmpt
         END
         goto go6 if (ptloc of pmpt=4)|(ptgoal of pmpt=4)
      end
      ptshiftprocedure of pmpt<true
      go6:
      nr<nr+1
   END
   spnr<3
   THIS ship<FIRST ship IN quay WITH shberthnr=spnr
   totmoves<(shexchange of this ship)
   FOR nr<3 TO 4
      IF (totmoves<30) & (shpt OF THIS ship IS NOT EMPTY)
         REMOVE THIS ship FROM cranereqlst IF shreqpt OF this ship>0
         shreqpt<0
         GOTO go7
      END
      pmpt<FIRST portainer IN ptset WITH ptnr=nr
      IF portaineractive[nr]=FALSE
         WAIT WHILE LENGTH OF shpt OF THIS ship=2
         ptgoal OF pmpt<5 IF nr=3
         ptgoal OF pmpt<6 IF nr=4
         CALL portainerstart
         shreqpt<shreqpt-1
         REMOVE THIS ship FROM cranereqlst IF shreqpt=0
         GOTO go7
      END
      if nr=3
         IF (ptloc of pmpt=5)&(ptmove of pmpt=TRUE)
            remove pmpt from shpt of shiponhand of pmpt
            join pmpt to passivept
            portaineractive[3]<FALSE
            ptmove of pmpt<FALSE
            cancel pmpt
         END
         goto go7 if (ptloc of pmpt=5)|(ptgoal of pmpt=5)
      end
      if nr=4
         IF (ptloc of pmpt=6)&(ptmove of pmpt=TRUE)
            remove pmpt from shpt of shiponhand of pmpt
            join pmpt to passivept
            portaineractive[4]<FALSE
            ptmove of pmpt<FALSE
            cancel pmpt
         END
         goto go7 if (ptloc of pmpt=6)|(ptgoal of pmpt=6)
      end
      ptshiftprocedure of pmpt<true
      go7:
      IF (totmoves<30) & (shpt OF THIS ship IS NOT EMPTY)
         REMOVE THIS ship FROM cranereqlst IF shreqpt OF this ship>0
         shreqpt<0
      END
   END
END
FOR nr<1 TO 4
   REPEAT FROM threeships IF portaineractive[nr]=FALSE
END
END

PASSIVATE
```

MODEL ICT 1
MAC PORTAINERSTART

Date: 98/09/09
Time: 11:27:46

```
@assigning portainers to ship@

start:
THIS ship<FIRST ship IN quay WITH shberthnr=spnr
pmpt<FIRST portainer IN passivept WITH ptnr=nr
portaineractive[nr]<TRUE
MOVE ptatquay[ptloc OF pmpt] TO 1 IF ptloc OF pmpt>0    @anim@
REMOVE pmpt FROM passivept
shiponhand OF pmpt<THIS ship
JOIN pmpt TO shpt OF THIS ship
ptmove OF pmpt<TRUE IF (ptgoal OF pmpt≠ptloc OF pmpt)
IF LENGTH OF shpt OF THIS ship=2
   IF shimportteu OF THIS ship>0
      xrtgc<FIRST rtgc IN xrtgcset WITH rtgcnr=shimportbay OF THIS ship
      JOIN xrtgc TO xrtgcwimp OF THIS ship IF xrtgcwimp IS EMPTY
      rtgcship OF xrtgc<THIS ship
      MOVE rtgcfig[rtgcnr OF xrtgc] TO 1               @anim@
      nrrtgcw OF THIS ship<2
      xrtgcstop OF xrtgc<FALSE
      ACTIVATE xrtgc FROM rolltolocation IN rtgcprocess IF xrtgc IS NOT
      ACTIVE
   END
   IF ((shimportteu OF THIS ship=0)&(shexportteu OF THIS ship>0))
      xrtgc<FIRST rtgc IN xrtgcset WITH rtgcnr=shexportbay OF THIS ship
      JOIN xrtgc TO xrtgcwexp OF THIS ship IF xrtgcwexp IS EMPTY
      rtgcship OF xrtgc<THIS ship
      MOVE rtgcfig[rtgcnr OF xrtgc] TO 1               @anim@
      nrrtgcw OF THIS ship<2
      xrtgcstop OF xrtgc<FALSE
      ACTIVATE xrtgc FROM rolltolocation IN rtgcprocess IF xrtgc IS NOT
      ACTIVE
   END
END
IF waitinglist OF THIS ship IS NOT EMPTY
   REACTIVATE EACH transporter IN waitinglist OF THIS ship
END
JOIN pmpt TO ttreqlst IF ptshiftprocedure OF pmpt=FALSE
ptshiftprocedure OF pmpt<FALSE
ACTIVATE ttmaster FROM ttinstructions IN ttmasterprocess IF ttmaster
IS NOT ACTIVE
ACTIVATE pmpt FROM unloadship IN portainerprocess IF ptmove OF pmpt=FALSE
ACTIVATE pmpt FROM rolltoplace IN portainerprocess IF ptmove OF pmpt=TRUE
REACTIVATE THIS ship IF THIS ship IS NOT ACTIVE
RETURN
```

2.8 Macro PORTAINERSTART

This macro is activated by the module PTMASTER and is responsible for defining various attributes and for activating various components.

[3-41]:
The macro starts at the label *start* and assigns the portainer to a certain ship. The portainer is also joined to a *ttreqlst* if necessary. This is a list containing all the portainers that are requesting for tractor-trailers. Consequently the TTMASTER is activated in the module TTMASTERPROCESS.
Subsequently, the portainer is activated in the module PORTAINERPROCESS. Depending on the position of the portainer (whether it needs to change position or not), the portainer is activated at a certain label in the PORTAINERPROCESS module, either *rolltoplace* or *unloadship*.
Finally the macro reactivates the ship that was previously selected by the PTMASTER and returns to the module.

xrtgc	Reference to rtgc: second RTGC to be active in stack
rtgcship	Reference to ship: ship the second RTGC will be servicing
xrtgcstop	Logical attribute of rtgc: defines if second RTGC should stop its actions in the stack
waitinglist	Set attribute of ship: set to which tractor-trailers, that are waiting at the quay for a portainer to reach its position are joined

```
@@@@@@@@@@@@@@@@@@@@@@@
@ portainer process @
@@@@@@@@@@@@@@@@@@@@@@@

rolltoplace:
IF ptnr=1
   WAIT UNTIL ((placeocc[2]=FALSE)&(placeocc[3]=FALSE)) IF ptloc=1
END
IF ptnr=2
   WAIT UNTIL placeocc[3]=FALSE IF ptloc=4
   WAIT UNTIL ((placeocc[4]=FALSE)&(placeocc[3]=FALSE)) IF
   ((ptloc=2)&(ptgoal=4))
   WAIT UNTIL placeocc[3]=FALSE IF ((ptloc=2)&(ptgoal=3))
   WAIT UNTIL placeocc[4]=FALSE IF ((ptloc=3)&(ptgoal=4))
END
IF ptnr=3
   WAIT UNTIL placeocc[4]=FALSE IF ptloc=3
   WAIT UNTIL ((placeocc[4]=FALSE)&(placeocc[3]=FALSE)) IF
   ((ptloc=5)&(ptgoal=3))
   WAIT UNTIL (placeocc[4]=FALSE) IF ((ptloc=5)&(ptgoal=4))
   WAIT UNTIL (placeocc[3]=FALSE) IF ((ptloc=4)&(ptgoal=3))
END
IF ptnr=4
   WAIT UNTIL ((placeocc[5]=FALSE)&(placeocc[4]=FALSE)) IF ptloc=6
END
ptmove<FALSE
IF ptloc=0
   ENTER shiftingpt
   WAIT ptshifttime
   LEAVE shiftingpt
   GOTO start
END
ENTER shiftingpt
placeocc[ptloc]<FALSE
MOVE ptatquay[ptloc] TO SINK              @anim@
ptloc<0
WAIT ptshifttime
LEAVE shiftingpt
start:
ptunloading<false if shimportteu of shiponhand =0
GOTO shift IF ptshiftprocedure=TRUE
ptloc<ptgoal
placeocc[ptloc]<TRUE
MOVE ptatquay[ptloc] TO 1                 @anim@
ptgoal<0

unloadship:
pttime2<NOW
GOTO loadship IF shimportteu OF shiponhand=0
shipunload:
ENTER importpt
MOVE ptimport[ptloc] TO 1                 @anim@
pttime1<NOW
WORK ptliftcycle
CALL quayload IF shimportteu OF shiponhand>0
pttottime<pttottime+(NOW-pttime1)
MOVE pttottimefig[ptnr] TO pttottime      @anim@
unload:
IF (shimportteu OF shiponhand>0)|(ptload≠"o")
   IF ptshiftprocedure=TRUE
      IF ptload="o"
         LEAVE importpt
         MOVE ptimport[ptloc] TO SINK     @anim@
         GOTO shiftprocedure IF ptload="o"
      END
      WAIT WHILE LENGTH OF ttunderpt=0
      GOTO go
   END
   GOTO go IF ptload≠"o"
   pttime1<NOW
   WORK ptliftcycle
   CALL quayload IF shimportteu OF shiponhand>0
   pttottime<pttottime+(NOW-pttime1)
   MOVE pttottimefig[ptnr] TO pttottime  @anim@
   REPEAT FROM unload IF ptload="o"
   go:
   IF ttunderpt IS EMPTY
      WAIT 10 SECONDS
      REPEAT FROM unload
   END
   ptimptt<FIRST OF ttunderpt
   REMOVE ptimptt FROM ttunderpt
```

2.9 Module PORTAINERPROCESS

This module is activated by the PTMASTER and concerns the portainer actions. The portainer is activated to start its actions from either the label *rolltoplace* [line 5] or *unloadship* [line 45].

[5-43]:
If the portainer is instructed to move to a different position at the quay before servicing a ship, it will follow the actions described in these lines. This section checks whether the situation at the quay allows the portainer to drive to the position it is instructed to. This is only possible if no other portainers are blocking its way. If this is the case the portainer will have to wait until these portainers have moved.
It is possible, while the portainer is waiting for other portainers (that are blocking its path) to move or while the portainer is moving itself, it may occur that the situation at the quay changes, resulting in new instructions. In this case the portainer, after having reached its position, will follow the actions defined in lines 227-283 (starting from label *shiftprocedure*) and wait for new instructions from the PTMASTER.

ptnr	Integer attribute of portainer: number of the portainer

[45-125]:
This section concerns the unloading of a ship by the portainer.
If there are import containers on the ship the portainer will lift a box (either 1 TEU or 2 TEU) out of the ship and wait for a tractor-trailer to drive under the portainer spreader. The type of box is to be lifted out of the ship is determined in the macro QUAYLOAD that is activated when necessary.
Once all the import containers have been unloaded, the portainer will start loading the ship (label *loadship* line 127). Before the loading process starts, first the extra RTGC operating in the import stack is given orders to stop its actions and to become passive.
If the portainer receives orders from the PTMASTER to drive to another position at the quay (logical attribute *ptshiftprocedure* becomes TRUE) it will first finish the ongoing actions and then follow the actions described in lines 227 to 283 starting from label *shiftprocedure*.

pttime2	Real attribute of portainer: time portainer starts unloading or loading process
pttime1	Real attribute of portainer: net starting time portainer actions
QUAYLOAD	Macro: determines the type of container to be unloaded (size and load)
pttottime	Real attribute of portainer: total time of net portainer actions
ptload	Character attribute of portainer: defines the nature of the container load (full, empty or both)
ptteu	Integer attribute of portainer: TEU of the container (1 or 2 TEU)
ptbox	Integer attribute of portainer: number of boxes (1 or 2)
ptimptt	Reference to transporter: the tractor-trailer being served by the portainer
trload	Character attribute of transporter: nature of the load on the transporter
ptnrbox	Double attribute of portainer: total number of boxes a portainer has moved

```
   MOVE ttuptfig[ptnr] TO LENGTH OF ttunderpt   @anim@
   IF trbox of ptimptt >0
      remove ptimptt from ttexportpt
      ACTIVATE ptimptt FROM leaveportainer IN ttprocess
      REPEAT FROM go
   END
   seaboxin<seaboxin+1
   ptnrbox<ptnrbox+1
   pttime1<NOW
   JOIN ptimptt TO loadtt
   REMOVE ptimptt FROM ttimportpt
   WORK ptliftontime
   IF ptbox=2
      WORK ptliftcycle
      WORK ptliftontime
      seaboxin<seaboxin+1
      ptnrbox<ptnrbox+1
      MOVE ptnrboxfig[ptnr] TO ptnrbox          @anim@
   END
   trload OF ptimptt<ptload
   trbox OF ptimptt<ptbox
   trteu OF ptimptt<ptteu
   trdestination OF ptimptt<ptdestination
   REMOVE ptimptt FROM loadtt
   ACTIVATE ptimptt FROM leaveportainer IN ttprocess
   ptload<"o"
   ptteu<0
   ptbox<0
   MOVE ptboxfig[ptnr] TO SINK                  @anim@
   ptimptt<NONE
   pttottime<pttottime+(NOW-pttime1)
   MOVE pttottimefig[ptnr] TO pttottime         @anim@
   REPEAT FROM unload
END

IF (LENGTH OF xrtgcwimp OF shiponhand=1)
   xrtgcpt<FIRST rtgc IN xrtgcwimp OF shiponhand WITH rtgcnr=shimportbay OF
   shiponhand
   xrtgcstop OF xrtgcpt<TRUE
   nrrtgcw OF shiponhand<1
   REMOVE xrtgcpt FROM xrtgcwimp OF shiponhand
   MOVE rtgcfig[rtgcnr OF xrtgcpt] TO SINK IF xrtgcpt IS NOT ACTIVE   @anim@
   CANCEL xrtgcpt IF xrtgcpt IS NOT ACTIVE
END
MOVE ptimport[ptloc] TO SINK                    @anim@
LEAVE importpt

loadship:
ENTER exportpt
MOVE ptexport[ptloc] TO 1                       @anim@
IF (LENGTH OF shpt OF shiponhand=2)&(LENGTH OF xrtgcwexp OF shiponhand=0)
   xrtgcpt<FIRST rtgc IN xrtgcset WITH rtgcnr=shexportbay OF shiponhand
   JOIN xrtgcpt TO xrtgcwexp OF shiponhand
   MOVE rtgcfig[rtgcnr OF xrtgcpt] TO 1         @anim@
   nrrtgcw OF shiponhand<2
   xrtgcstop OF xrtgcpt<FALSE
   ACTIVATE xrtgcpt FROM rolltolocation IN rtgcprocess IF xrtgcpt IS
  NOT ACTIVE
END
load:
ptunloading<FALSE
IF shexportteu OF shiponhand>0
   IF ptshiftprocedure=TRUE
      IF (LENGTH OF ttexportpt=0)&(LENGTH OF ttimportpt=0)
         LEAVE exportpt
         MOVE ptexport[ptloc] TO SINK           @anim@
         GOTO shiftprocedure
      END
   END
   IF ttunderpt IS EMPTY
      WAIT 10 SECONDS
      REPEAT FROM load
   END
   ptexptt<FIRST OF ttunderpt
   IF trbox OF ptexptt=0
      REMOVE ptexptt FROM ttimportpt
      REMOVE ptexptt FROM ttunderpt
      MOVE ttuptfig[ptnr] TO LENGTH OF ttunderpt   @anim@
      ACTIVATE ptexptt FROM leaveportainer IN ttprocess
      REPEAT FROM load
   END
   REMOVE ptexptt FROM ttunderpt
   MOVE ttuptfig[ptnr] TO LENGTH OF ttunderpt   @anim@
   REMOVE ptexptt FROM ttexportpt
   seaboxout<seaboxout+1
```

[127-206]:
This section concerns the loading process of a ship by a portainer.
First a second RTGC is instructed to operate in the export stack of the ship (in case there isn't already a second RTGC active).
The loading process now really starts with the portainer waiting until a tractor-trailer with its load is standing underneath the spreader. The load is lifted onto the ship and the spreader returns to its original position and waits for the next tractor-trailer to arrive.
When the ship has been completely loaded the portainer will follow the actions described in the lines 208-225 starting from the label *jobdone*, after giving the second RTGC (if there is one) in the export stack instructions to become passive.
Depending on the situation at the quay the portainer can also receive instructions to change its position. When this happens, the portainer will finish whatever actions it is doing at that moment and follow the actions described in lines 227-283.

ptexptt	Reference to transporter: tt being served by the portainer

```
    ptnrbox<ptnrbox+1
    MOVE ptnrboxfig[ptnr] TO ptnrbox                @anim@
    pttime1<NOW
    JOIN ptexptt TO unloadtt
    ptload<trload OF ptexptt
    ptbox<trbox OF ptexptt
    MOVE ptboxfig[ptnr] TO 1                        @anim@
    ptteu<trteu OF ptexptt
    WORK ptliftofftime
    CALL quayload
    IF trbox OF ptexptt=2
       WORK ptliftcycle
       WORK ptliftofftime
       seaboxout<seaboxout+1
       ptnrbox<ptnrbox+1
       MOVE ptnrboxfig[ptnr] TO ptnrbox             @anim@
    END
    ptbox<0
    MOVE ptboxfig[ptnr] TO SINK                     @anim@
    ptteu<0
    ptload<"o"
    trbox OF ptexptt<0
    trteu OF ptexptt<0
    trload OF ptexptt<"o"
    REMOVE ptexptt FROM unloadtt
    trworkship OF ptexptt<NONE
    ACTIVATE ptexptt FROM leaveportainer IN ttprocess
    WORK ptliftcycle
    pttottime<pttottime+(NOW-pttime1)
    MOVE pttottimefig[ptnr] TO pttottime            @anim@
    REPEAT FROM load
END

MOVE ptexport[ptloc] TO SINK                        @anim@
LEAVE exportpt
IF (LENGTH OF xrtgcwexp OF shiponhand=1)
    xrtgcpt<FIRST rtgc IN xrtgcwexp OF shiponhand WITH rtgcnr=shexportbay OF
    shiponhand
    xrtgcstop OF xrtgcpt<TRUE
    nrrtgcw OF shiponhand<1
    REMOVE xrtgcpt FROM xrtgcwexp OF shiponhand
    MOVE rtgcfig[rtgcnr OF xrtgcpt] TO SINK IF xrtgcpt IS NOT ACTIVE  @anim@
    CANCEL xrtgcpt IF xrtgcpt IS NOT ACTIVE
END

jobdone:
pttot2time<pttot2time+(NOW-pttime2)
ptend<TRUE
FOR EACH transporter IN ttofpt
    trendprocedure<TRUE
END
LEAVE ttreqlst IF BELONGS TO ttreqlst
departship<shiponhand
shiponhand<NONE
ptunloading<TRUE
ENTER passivept
LEAVE shpt OF departship
portaineractive[ptnr]<FALSE
MOVE ptatquay[ptloc] TO SINK                        @anim@
ptshiftprocedure<FALSE
ptend<FALSE
ACTIVATE ptmaster FROM reqlstcheck IN ptmasterprocess
IF ptmaster IS NOT ACTIVE
PASSIVATE

shiftprocedure:
pttot2time<pttot2time+(NOW-pttime2)
shift:
IF (LENGTH OF shpt OF shiponhand=2)&(nrrtgcw OF shiponhand=2)
    xrtgcpt<FIRST rtgc IN xrtgcwimp OF shiponhand IF xrtgcwimp OF
    shiponhand IS NOT EMPTY
    xrtgcpt<FIRST rtgc IN xrtgcwexp OF shiponhand IF xrtgcwexp OF
    shiponhand IS NOT EMPTY
    xrtgcstop OF xrtgcpt<TRUE
    nrrtgcw OF shiponhand<1
    REMOVE xrtgcpt FROM xrtgcwimp OF shiponhand IF xrtgcwimp OF shiponhand IS
    NOT EMPTY
    REMOVE xrtgcpt FROM xrtgcwexp OF shiponhand IF xrtgcwexp OF shiponhand IS
    NOT EMPTY
    MOVE rtgcfig[rtgcnr OF xrtgcpt] TO SINK IF xrtgcpt IS NOT ACTIVE  @anim@
    CANCEL xrtgcpt IF xrtgcpt IS NOT ACTIVE
END
IF ptshiftprocedure=TRUE
   IF ptunloading=TRUE
      IF LENGTH OF ttimportpt=0
```

[208-225]:
This section of the module concerns the actions performed when a portainer has finished servicing a ship and has not yet received new instructions. The portainer, before becoming passive, first activates the PTMASTER.

departship	Reference to ship: ship in the process of departing
pttot2time	Real attribute of portainer: total time a portainer is assigned to ship

[227-283]:
This section concerns the actions the portainer has to perform once it gets instructions from the PTMASTER to quit its actions in order to change position at the quay.

First of all, if there is a second RTGC active in the stack, this RTGC will receive orders to also stop its actions.
When the portainer is in the process of unloading a ship, it will (before becoming passive) send away all the tt's expecting to be serviced by this particular portainer. If the portainer is in the process of loading a ship, it will first service all the tt's waiting in the queue and the tt's on their way to the portainer (that are already on the quay) before becoming passive.
The PTMASTER is then activated to direct the portainer to its new position at the quay.

```
            THIS ship<shiponhand OF THIS portainer
            JOIN THIS ship TO cranereqlst IF shreqpt=0
            shreqpt<shreqpt+1
            LEAVE shpt OF shiponhand
            GOTO final
         END
         WAIT WHILE ttunderpt IS EMPTY
         ptimptt<FIRST OF ttunderpt
         REMOVE ptimptt FROM ttunderpt
         MOVE ttuptfig[ptnr] TO LENGTH OF ttunderpt  @anim@
         REMOVE ptimptt FROM ttimportpt
         trdestination OF ptimptt<"yard"
         ACTIVATE ptimptt FROM leaveportainer IN ttprocess
         REPEAT FROM shift
      END
      IF ptunloading=FALSE
         IF LENGTH OF ttexportpt=0
            THIS ship<shiponhand OF THIS portainer
            JOIN THIS ship TO cranereqlst IF shreqpt=0
            shreqpt<shreqpt+1
            LEAVE shpt OF shiponhand
            GOTO final
         END
         WAIT WHILE ttunderpt IS EMPTY
         ptexptt<FIRST OF ttunderpt
         REMOVE ptexptt FROM ttunderpt
         MOVE ttuptfig[ptnr] TO LENGTH OF ttunderpt  @anim@
         REMOVE ptexptt FROM ttexportpt
         trdestination OF ptexptt<"exportquay"
         ACTIVATE ptexptt FROM leaveportainer IN ttprocess
         REPEAT FROM shift
      END
      final:
      shiponhand<NONE
      ptunloading<TRUE
      ENTER passivept
      portaineractive[ptnr]<FALSE
      MOVE ptatquay[ptloc] TO SINK IF ptloc>0      @anim@
      ACTIVATE ptmaster FROM reqlstcheck IN ptmasterprocess
   IF ptmaster IS NOT ACTIVE
      PASSIVATE
   END
```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ determining how the tt's are (un)loaded by the portainers @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

LOCAL:
REAL: r1 r2

IF ptunloading=TRUE
   r1<unif
   IF r1≤ (shimp20 OF shiponhand/(shimp20 OF shiponhand+shimp40 OF shiponhand))
      r2<unif
      IF r2≤ shimp20f OF shiponhand/shimp20 OF shiponhand
         ptload<"f"
         shimp20f OF shiponhand<shimp20f OF shiponhand -1
         GOTO point1
      END
      IF r2> shimp20f OF shiponhand/shimp20 OF shiponhand
         ptload<"mt"
         shimp20mt OF shiponhand<shimp20mt OF shiponhand -1
      END
      point1:
      ptbox<1
      MOVE ptboxfig[ptnr] TO 1
      ptteu<1
      shimp20 OF shiponhand<shimp20 OF shiponhand -1
      shimportteu OF shiponhand<shimportteu OF shiponhand -1
      IF shimp20 OF shiponhand>0
         r2<unif
         IF r2≤ shimp20f OF shiponhand/shimp20 OF shiponhand
            shimp20f OF shiponhand<shimp20f OF shiponhand -1
            ptload<"f" IF (ptload="f")
            ptload<"fmt" IF (ptload="mt")
            GOTO point2
         END
         IF r2> shimp20f OF shiponhand/shimp20 OF shiponhand
            shimp20mt OF shiponhand<shimp20mt OF shiponhand -1
            ptload<"mt" IF (ptload="mt")
            ptload<"fmt" IF (ptload="f")
         END
         point2:
         ptbox <2
         ptteu <2
         shimp20 OF shiponhand<shimp20 OF shiponhand -1
         shimportteu OF shiponhand<shimportteu OF shiponhand -1
      END
      ptdestination<"importstack" IF (ptload="f")|(ptload="fmt")
      ptdestination<"impemptystack" IF (ptload="mt")
      RETURN
   END
   IF r1>(shimp20 OF shiponhand/(shimp20 OF shiponhand+shimp40 OF shiponhand))
      r2<unif
      IF r2≤shimp40f OF shiponhand/shimp40 OF shiponhand
         ptload<"f"
         shimp40f OF shiponhand<shimp40f OF shiponhand -1
         ptdestination<"importstack"
         GOTO point3
      END
      IF r2>shimp40f OF shiponhand/shimp40 OF shiponhand
         ptload<"mt"
         shimp40mt OF shiponhand<shimp40mt OF shiponhand -1
         ptdestination<"impemptystack"
      END
      point3:
      ptbox<1
      MOVE ptboxfig[ptnr] TO 1
      ptteu<2
      shimp40 OF shiponhand<shimp40 OF shiponhand -1
      shimportteu OF shiponhand<shimportteu OF shiponhand -2
      RETURN
   END
END

IF ptunloading=FALSE
   IF ptbox=2
      shexp20 OF shiponhand<shexp20 OF shiponhand -2
      shexportteu OF shiponhand<shexportteu OF shiponhand -2
      IF ptload="f"
         shexp20f OF shiponhand<shexp20f OF shiponhand -2
      END
      IF ptload="mt"
         shexp20mt OF shiponhand<shexp20mt OF shiponhand -2
      END
```

2.10 Macro QUAYLOAD

This Macro is activated by the module PORTAINERPROCESS.

[8-71]:
This section determines the nature of the containers (size and load) to be unloaded by the portainers and keeps track of the number of containers still in the ship to be unloaded. The container lifted out of the hold is given information on its specifics (load, TEU and destination). This information is later passed on to the tractor-trailer.
A portainer will either lift two 20 ft containers onto a tractor-trailer (if there are enough in the ships hold) or one 40 ft container. The two 20 ft containers can be either full or empty or one of each. The 40 ft container will be full or empty.
The macro is activated only once by the module also in the case the portainer lifts two containers from the ship hold and performs the lifting actions twice. Therefore the import container data can be updated in one time.
After the number of containers on the ship has been updated, the macro returns to the module.

[73-111]:
If the ship is in the process of being loaded, this section of the macro is followed. The information on the container attributes loaded by the portainer is updated by the macro.
The tractor-trailer will deliver either one or two boxes. If two boxes are delivered they will both be 20 ft boxes and if one box is delivered it can be either a 20 ft or 40 ft box. In the case of two boxes, the portainer will perform its lifting actions twice, but the macro is only activated once thus the macro will update the tractor-trailer load in one operation.

```
      GOTO teucount
    END
    IF ptbox=1
      IF ptteu=1
        shexp20 OF shiponhand<shexp20 OF shiponhand -1
        shexportteu OF shiponhand<shexportteu OF shiponhand -1
        IF ptload="f"
          shexp20f OF shiponhand<shexp20f OF shiponhand -1
        END
        IF ptload="mt"
          shexp20mt OF shiponhand<shexp20mt OF shiponhand -1
        END
        GOTO teucount
      END
      IF ptteu=2
        shexp40 OF shiponhand<shexp40 OF shiponhand -1
        shexportteu OF shiponhand<shexportteu OF shiponhand -2
        IF ptload="f"
          shexp40f OF shiponhand<shexp40f OF shiponhand -1
        END
        IF ptload="mt"
          shexp40mt OF shiponhand<shexp40mt OF shiponhand -1
        END
      END
    END
    teucount:
    ptdestination<"yard"
    RETURN
END
```

MODEL ICT 1
MOD TTMASTERPROCESS

Date: 98/09/09
Time: 11:41:08

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ assign tt's to portainers/ships @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

ttinstructions:
IF ttreqlst IS NOT EMPTY
   tmpt<FIRST OF ttreqlst
   FOR n<1 TO ttperpt
      IF (ptend OF tmpt=FALSE)
         WAIT WHILE ttavailable IS EMPTY
         GOTO check IF tmpt BELONGS NOT TO ttreqlst
         tmttforpt<FIRST OF ttavailable
         JOIN tmttforpt TO ttofpt OF tmpt
         ptfirst OF tmttforpt<tmpt
         trrow OF tmttforpt<10
         REMOVE tmttforpt FROM ttavailable
         MOVE ttparking TO LENGTH OF ttavailable    @anim@
         ACTIVATE tmttforpt FROM startup IN ttprocess
         WAIT 4 SECONDS
      END
   END
   n<0
   REMOVE tmpt FROM ttreqlst
   check:
   REPEAT FROM ttinstructions
END
PASSIVATE
```

2.11 Module TTMASTERPROCESS

[5-27]:
This module is activated by the PTMASTER if the TTMASTER is not already active. The TTMASTER is responsible for assigning tractor-trailers to portainers that are requesting tt's. The TTMASTER will assign tt's to the first portainer on the *ttreqlist.*
The tt's are activated from the label *startup* in the module TTPROCESS in intervals of 4 seconds. When the standard number of tt's to be activated per portainer is reached, the portainer is removed from the ttreqlst. The TTMASTER will then check the ttreqlst and perform the same actions for the next portainer on the list if there is one. If the list is empty the TTMASTER becomes passive until being reactivated by the PTMASTER.
If there aren't any available tt's in the parking lot the TTMASTER waits until a tractor-trailer arrives in the parking lot.

ttreqlst	Queue: contains the portainers requesting for tractor-trailers
tmpt	Reference to portainer: portainer the tt is assigned to
ttperpt	Real attribute of MAIN: standard number of tt's assigned per portainer
tmttforpt	Reference to transporter: tractor-trailer being assigned to a portainer

```
@@@@@@@@@@@@@@@
@ tt process @
@@@@@@@@@@@@@@@

startup:

leaveparking:
ENTER lane[10]
WORK mainlanetime
LEAVE lane[10]
ENTER outtraffic
WORK (CEIL((trrow*30)/mainlanespeed)) SECONDS
LEAVE outtraffic
ENTER maintraffic
WORK mainlanetime
LEAVE maintraffic
ENTER yardqueue
ACTIVATE trmasteryard FROM begin IN trmyardprocess IF trmasteryard IS NOT ACTIVE
WAIT
GOTO parking IF trdestination="parking"
GOTO yardaction IF (trdestination="exportstack")|(trdestination="expemptystack")

drivetoquay:
ENTER tlyqueue
ACTIVATE tlyard FROM check IN tlyprocess IF tlyard IS NOT ACTIVE
WAIT
LEAVE tlyqueue
WRITE NOW TO yttonbridge WITH IMAGE xxxx.xxxxx
ENTER yardtoquay
WORK bridgetime
LEAVE yardtoquay
restart:
ENTER quayqueue
ACTIVATE trmasterquay FROM begin IN trmquayprocess IF trmasterquay IS NOT ACTIVE
WAIT
IF ttwait=TRUE
   WAIT WHILE (LENGTH OF shpt OF trworkship=0)
   WAIT WHILE (ptshiftprocedure OF trworkpt=TRUE)
   ttwait<FALSE
   REPEAT FROM restart
END
GOTO drivetoyard IF (trdestination="yard")|(trdestination="parking")

quaytoportainer:
ENTER quayintraffic
WORK trquaytime
LEAVE quayintraffic
ENTER ttunderpt OF trworkpt
ttupttime<NOW
MOVE ttuptfig[ptnr OF trworkpt] TO LENGTH OF ttunderpt OF trworkpt
WAIT
STORE (NOW-ttupttime) AS "ttuptti"

leaveportainer:
ENTER quayouttraffic
WORK trquaytime
LEAVE quayouttraffic
GOTO restart IF trdestination="exportquay"
ENTER quayqueue
ACTIVATE trmasterquay FROM begin IN trmquayprocess IF trmasterquay IS NOT ACTIVE
WAIT
GOTO quaytoportainer IF trdestination="importquay"

drivetoyard:
ENTER tlqqueue
ACTIVATE tlquay FROM check IN tlqprocess IF tlquay IS NOT ACTIVE
WAIT
LEAVE tlqqueue
WRITE NOW TO qttonbridge WITH IMAGE xxxx.xxxxx
ENTER quaytoyard
WORK bridgetime
LEAVE quaytoyard
ENTER yardqueue
ACTIVATE trmasteryard FROM begin IN trmyardprocess IF trmasteryard IS NOT ACTIVE
WAIT
GOTO parking IF trdestination="parking"
GOTO drivetoquay IF trdestination="importquay"

yardaction:
trrow<10 IF trdestination="impemptystack"
ENTER intraffic
WORK (CEIL((trrow*30)/mainlanespeed)) SECONDS
```

2.12 Module TTPROCESS

This module concerns the tractor-trailer actions and is activated after the TTMASTER has instructed a tt to start working.

[5-21]:
The tt is activated in this module at the label *startup* and will first leave the parking lot. Once it has left the parking lot, the tt will drive to the traffic master of the yard to receive further instructions. The traffic master of the yard (module TRMYARDPROCESS) is activated the moment the tt enters the *yardqueue*. The traffic master then reactivates the tt and the tt will either drive to the parking lot, the container yard or onto the bridge to the quay.

lane[ ]	Queue: contains the tractor-trailers and trucks present in that particular lane
yardqueue	Queue: contains all the tt's waiting for instructions from the traffic master of the yard

[23-42]:
This section starts at the label *drivetoquay* and describes the actions a tt performs when driving from the yard to the quay via the bridge. Before driving on the bridge, the tt enters the traffic light queue (*yardqueue*) which activates the TLYPROCESS module. After the traffic light reactivates the tt, it will proceed its course. Upon arrival at the quay entrance, the tt enters a *quayqueue* and the module TRMQUAYPROCESS (traffic master of the quay) is activated. This traffic master will either send the tt to a portainer or back to the yard.

tlyqueue	Queue: contains the quay bound tt's waiting for the traffic light
quayqueue	Queue: contains the tt's waiting for instructions from the traffic master of the quay

[44-52]:
This section starts at the label *quaytoportainer* and describes the tt actions from the quay entrance to the portainer. Upon arriving at the portainer the tt will wait until it is served by the portainer.

trquaytime	Real attribute of transporter: time needed to drive from the quay entrance to the portainer

[54-62]:
This section starts at the label *leaveportainer* and describes the tt actions after it has left the portainer. The tt will drive to the bridge entrance and join the *quayqueue*. Here the traffic master of the quay is again activated and the tt will either drive to the yard or to another portainer.

[64-77]:
This section concerns the drive from the quay to the yard and starts at the label *drivetoyard*. Before driving to the yard the tt enters the queue of the traffic light at the quay (*tlqqueue*). At this point the traffic light module is activated (module TLQPROCESS) and when the tt is reactivated it will drive to the yard. Upon arrival at the yard the tt enters the *yardqueue* and by doing so activates the traffic master of the yard. This traffic master will give the tt instructions if need be. The tt will now drive to the yard, parking lot or return to the quay.

tlqqueue	Queue: contains all the tt's waiting for the traffic light on the quay

```
LEAVE intraffic
ENTER lane[trrow]
WORK (CEIL(210/mainlanespeed)) SECONDS
ENTER ttq[trrow]
trqueuetime<NOW
WAIT

leaveyard:
WORK (CEIL(210/mainlanespeed)) SECONDS
LEAVE lane[trrow]
ENTER outtraffic
WORK (CEIL((trrow*30)/mainlanespeed)) SECONDS
LEAVE outtraffic
ENTER maintraffic
WORK mainlanetime
LEAVE maintraffic
GOTO drivetoquay IF (trbox>0)&(trdestination="exportquay")
GOTO yardaction IF trdestination="impemptystack"
ENTER yardqueue
ACTIVATE trmasteryard FROM begin IN trmyardprocess IF trmasteryard IS NOT ACTIVE
WAIT

GOTO drivetoquay IF trdestination="importquay"
GOTO yardaction IF trdestination="exportstack"
GOTO yardaction IF trdestination="expemptystack"
GOTO parking IF trdestination="parking"

parking:
ENTER intraffic
WORK (CEIL((300)/mainlanespeed)) SECONDS
LEAVE intraffic
trworkpt<ptfirst
LEAVE ttofpt OF trworkpt
ENTER ttavailable
MOVE ttparking TO LENGTH OF ttavailable      @anim@
trworkpt<NONE
trworkship<NONE
ptfirst<NONE
trendprocedure<FALSE
trdestination<"importquay"
trrow<0

PASSIVATE
```

[79-88]:
This section concerns the tt actions in the container yard and starts at the label *yardaction*. The tt will drive to a specific row to either deliver or pick-up a load. Having arrived at the correct stack, the tt enters a queue (*ttq[ ]*) and will wait until a RTGC has unloaded or loaded it.

ttq[ ]	Queue: contains all the tt's waiting to be served by a RTGC
trqueuetime	Real attribute of transporter: time of arrival in waiting queue

[90-108]:
This section starts at the label *leaveyard* and describes the tt actions after it has been serviced and reactivated by a RTGC. Once the tt has left the row and driven to the bridge entrance, it will either drive straight on to the quay (when it has a load for one of the ships moored), drive back to the yard (when it had two different loaded containers) or enter the *yardqueue* to receive instructions from the traffic master of the yard. After entering the *yardqueue* and once reactivated by the traffic master, it will drive to the quay, parking lot or the container yard.

[110-125]:
This section describes the tt actions when it is on its way to the parking lot from the traffic master of the yard location. The section starts at the label *parking*. Once arrived at the parking lot, the tt joins the *ttavailable* queue and becomes passive.

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ determining where the tt's have to drive on the quay @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

begin:
tmqtt<FIRST transporter IN quayqueue
REMOVE tmqtt FROM quayqueue

IF (trendprocedure OF tmqtt=TRUE) & (trbox OF tmqtt=0)
   trdestination OF tmqtt<"parking"
   GOTO final
END
IF trbox OF tmqtt=0
   IF LENGTH OF importpt=0
      trdestination OF tmqtt<"yard"
      GOTO final
   END
   IF LENGTH OF importpt=1
      tmqpt<FIRST portainer IN importpt
   END
   IF LENGTH OF importpt=2
      tmqpt<FIRST portainer IN importpt WITH SMALLEST tmqlasttime
      tmqlasttime OF tmqpt<NOW
   END
   IF LENGTH OF importpt>2
      tmqpt<FIRST portainer IN importpt WITH SMALLEST tmqlasttime
      tmqlasttime OF tmqpt<NOW
   END
   IF LENGTH OF ttimportpt OF tmqpt>4
      trdestination OF tmqtt<"yard"
      GOTO final
   END
   trworkpt OF tmqtt<tmqpt
   tmqship<shiponhand OF tmqpt
   trworkship OF tmqtt<tmqship
   JOIN tmqtt TO ttimportpt OF tmqpt
   trdestination OF tmqtt<"importquay"
END

load:
IF (trbox OF tmqtt>0)&(trdestination OF tmqtt="exportquay")
   tmqship<trworkship OF tmqtt
   IF LENGTH OF shpt OF tmqship>2
      trworkpt OF tmqtt<FIRST portainer in shpt OF tmqship
      ttwait OF tmqtt<TRUE
      GOTO final
   END
   IF LENGTH OF shpt OF tmqship=2
      tmqpt<FIRST portainer IN shpt OF tmqship WITH SMALLEST timept
      trworkpt OF tmqtt<tmqpt
      IF (ptshiftprocedure OF tmqpt=TRUE)|(tmqpt BELONGS TO importpt)
         tmqpt<FIRST portainer IN shpt OF tmqship WITH GREATEST timept
         trworkpt OF tmqtt<tmqpt
         GOTO towait IF ptshiftprocedure OF tmqpt=TRUE
      END
      timept OF tmqpt<NOW
      JOIN tmqtt TO ttexportpt OF tmqpt IF ptshiftprocedure OF tmqpt=FALSE
   END
   IF LENGTH OF shpt OF tmqship=1
      tmqpt<FIRST portainer IN shpt OF tmqship
      trworkpt OF tmqtt<tmqpt
      timept OF tmqpt<NOW
      JOIN tmqtt TO ttexportpt OF tmqpt IF ptshiftprocedure OF tmqpt=FALSE
   END
   towait:
   IF (LENGTH OF shpt OF tmqship=0)|(ptshiftprocedure OF tmqpt=TRUE)
      ttwait OF tmqtt<TRUE
      remove tmqtt from ttexportpt IF tmqtt belongs TO ttexportpt
   END
END
IF (trbox OF tmqtt>0)&(trdestination OF tmqtt≠"exportquay")
   GOTO final
END

IF trworkship OF tmqtt BELONGS TO berth[1]
   trquaytime OF tmqtt<berth1time
END
IF trworkship OF tmqtt BELONGS TO berth[2]
   trquaytime OF tmqtt<berth2time
END
IF trworkship OF tmqtt BELONGS TO berth[3]
   trquaytime OF tmqtt<berth3time
```

2.13 Module TRMQUAYPROCESS

[5-7]:
This module concerns the actions of the traffic master of the quay. If a tractor-trailer enters the *quayqueue* the traffic master is activated from the label *begin* (if not already active) and the first tractor-trailer in the queue is selected.

[9-38]:
This section describes the instructions given to a tt driver when he has no load. Depending on the situation at the quay the tt is sent to the yard or to a portainer that is in the process of unloading a ship. The tt is then reactivated and the traffic master waits four seconds before instructing the next tt in the *quayqueue*. If there are no other tt's waiting in the queue, the traffic master becomes passive till the arrival of a tt.

tmqtt	Reference to transporter: tractor-trailer presently being instructed by the traffic master of the quay
tmqpt	Reference to portainer: portainer the tt is instructed to drive to
tmqship	Reference to ship: the ship the portainer (tmqpt) is servicing
tmqlasttime	Real attribute of portainer: time at which portainer is sent a tt by the traffic master of the quay

[40-89]:
When a tt does have a load, the actions then taken by the traffic master are described in these lines of the module. The tt has either a load for the yard or for a ship. When the tt is yard bound, the traffic master sends the tt to the yard. When the tt is quay bound, the traffic master will direct the tt to a portainer servicing the ship the load is intended for.
The tt is then reactivated and the traffic master will become passive when the *quayqueue* is empty, otherwise the traffic master will instruct the next tt in the queue.

```
END

final:
REACTIVATE tmqtt
WAIT 4 SECONDS
REPEAT FROM begin IF quayqueue IS NOT EMPTY
PASSIVATE
```

MODEL ICT 1 Date: 98/09/09
MOD TRMYARDPROCESS Time: 12:13:21

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ determining where the tt's have to drive in the yard @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

begin:
IF LENGTH OF lane[10]≥10
   IF xrtgcworkset IS EMPTY
      xrtgcrow10<FIRST rtgc IN xrtgcset WITH rtgcnr=10
      ACTIVATE xrtgcrow10 FROM rolltolocation IN rtgcprocess
      JOIN xrtgcrow10 TO xrtgcworkset
      MOVE rtgcfig[10] TO 1              @anim@
   END
END

tmytt<FIRST transporter IN yardqueue
REMOVE tmytt FROM yardqueue

IF (trendprocedure OF tmytt=TRUE) & (trbox OF tmytt=0)
   trdestination OF tmytt<"parking"
   GOTO final
END

IF (trbox OF tmytt=1)|(trbox OF tmytt=2)
   IF (trdestination OF tmytt="importstack")|(trdestination OF tmytt="impemptys
tack")
      r10<unif
      IF trdestination OF tmytt="importstack"
         tmyship<trworkship OF tmytt
         trrow OF tmytt<shimportbay OF tmyship
      END
      IF trdestination OF tmytt="impemptystack"
         tmyship<trworkship OF tmytt
         trrow OF tmytt<10
      END
   END
   GOTO final
END

IF trbox OF tmytt=0
   r10<unif
   IF (LENGTH OF exportpt=0)&(LENGTH OF loadingship>0)
      trdestination OF tmytt<"importquay"
      GOTO final
   END
   IF LENGTH OF exportpt>0
      tmypt<FIRST portainer IN exportpt WITH SMALLEST ptlasttime
      ptlasttime OF tmypt<NOW
      IF (LENGTH OF ttexportpt OF tmqpt>4)
         trdestination OF tmytt<"importquay"
         GOTO final
      END
      trworkship OF tmytt<shiponhand OF tmypt
      tmyship<trworkship OF tmytt
      go:
      CALL yardcontainer
      IF trdestination OF tmytt="exportstack"
         trrow OF tmytt<shexportbay OF tmyship
      END
      IF trdestination OF tmytt="expemptystack"
         trrow OF tmytt<10
      END
   END
   IF (LENGTH OF exportpt=0)&(LENGTH OF loadingship=0)
      trdestination OF tmytt<"importquay"
   END
END

final:
REACTIVATE tmytt
WAIT 4 SECONDS
REPEAT FROM begin IF yardqueue IS NOT EMPTY
PASSIVATE
```

2.14 Module TRMYARDPROCESS

This module concerns the actions of the traffic master of the yard. The traffic master is activated the moment a tt enters the *yardqueue* (if the traffic master is not active already).

[5-21]:
The module starts at the label begin and first of all activates a second RTGC in the empties stack if necessary (in case there is a lot of traffic in that lane). Then the first tt in the queue is selected to receive instructions.

xrtgcrow10	Reference to rtgc: second RTGC to be activated in row 10 (empties stack)
tmytt	Reference to transporter: tt being instructed by traffic master of the yard

[23-36]:
These lines are followed if the tt has a load. The tt is instructed to drive to a specific row in the container yard to deliver its load. After the instructions are given, the tt is reactivated and the traffic master waits four seconds before either becoming passive or instructing the next tt in the queue (if there is one).

tmyship	Reference to ship: ship the tt will service

[38-71]:
When the tt has no load, the actions described in this section are followed by the traffic master. The tt is either sent to the quay or to the container yard to pick-up a load. When sent to the container yard the macro YARDCONTAINER is activated in which the row to drive to is specified (empties or export stack).
When this has been done, the tt is reactivated and the traffic master will become passive when the *yardqueue* is empty. If not, the traffic master instructs the next tt where to go.

tmypt	Reference to portainer: portainer the tt will drive to
ptlasttime	Real attribute of portainer: time at which the last tt was instructed to drive to that particular portainer
yardcontainer	Macro: defines the load to be picked-up in the yard (size and load)

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ determining where the tt's have to drive in the yard @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

LOCAL:
REAL: e f g h r3 z4

IF trbox OF tmytt=0
   test:
   IF (trdestination OF tmytt="yard")|(trdestination OF tmytt="importquay")
      e<CEIL((stackexp20f OF tmyship/2)+(stackexp40f OF tmyship))
      f<LENGTH OF shfulltt OF tmyship
      g<CEIL((stackexp20mt OF tmyship/2)+(stackexp40mt OF tmyship))
      h<LENGTH OF shemptytt OF tmyship
      IF (f≥e) & (h≥g)
         trdestination OF tmytt<"importquay"
         GOTO macroend
      END
      IF (f<e) & (h<g)
         r3<unif
         z4<((stackexp20f OF tmyship+stackexp40f OF tmyship)/(stackexp20f OF
        tmyship+stackexp40f OF tmyship+stackexp20mt OF tmyship+stackexp40mt OF
tmyship))
         trdestination OF tmytt<"exportstack" IF (r3≤z4)
         trdestination OF tmytt<"expemptystack" IF (r3>z4)
         JOIN tmytt TO shfulltt OF tmyship IF trdestination OF tmytt="exportsta
ck"
         JOIN tmytt TO shemptytt OF tmyship IF trdestination OF tmytt="expempty
stack"
         GOTO macroend
      END
      IF (f<e)
         trdestination OF tmytt<"exportstack"
         JOIN tmytt TO shfulltt OF tmyship
         GOTO macroend
      END
      IF (h<g)
         trdestination OF tmytt<"expemptystack"
         JOIN tmytt TO shemptytt OF tmyship
      END
   END
END

macroend:
RETURN
```

2.15 Macro YARDCONTAINER

[8-41]:
This macro is activated by the traffic master of the yard (module TRMYARDPROCESS) and defines the row the tractor-trailer is to drive to.
The tt is instructed to either go to the empties stack or to the export stack. The specifics of the load are determined later in the RTGCPROCESS module.

shfulltt	Set attribute of ship: set to which all tt's on their way to pick-up a full export load in the yard for a specific ship, are joined to
shemptytt	Set attribute of ship: set to which all tt's on their way to pick-up an empty export load in the yard for a specific ship, are joined to

MODEL ICT 1
MOD TLQPROCESS

Date: 98/09/09
Time: 12:13:57

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ trafficlightprocess quay to bridge @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

start:
WAIT UNTIL tlqqueue IS NOT EMPTY

check:
tlqtt<FIRST transporter IN tlqqueue
REACTIVATE tlqtt
trlightq<FALSE
WAIT 4 SECONDS
trlightq<TRUE
REPEAT FROM start IF tlqqueue IS EMPTY
REPEAT FROM check IF tlqqueue IS NOT EMPTY
```

MODEL ICT 1
MOD TLYPROCESS

Date: 98/09/09
Time: 12:13:57

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ trafficlightprocess yard to bridge @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

start:
WAIT UNTIL tlyqueue IS NOT EMPTY

check:
tlytt<FIRST transporter IN tlyqueue
REACTIVATE tlytt
trlighty<FALSE
WAIT 4 SECONDS
trlighty<TRUE
REPEAT FROM start IF tlyqueue IS EMPTY
REPEAT FROM check IF tlyqueue IS NOT EMPTY
```

2.16 Module TLQPROCESS

[5-15]:
This module concerns the traffic light actions at the quay. This traffic light is activated (when not already active) the moment a yard bound tt enters the traffic light queue (*tlqqueue*).
This traffic light reactivates the first tt in the queue and will reactivate the next tt in the queue after four seconds. When there aren't any other tt's in the queue, the traffic light will wait till a tt enters the queue.

tlqtt	Reference to transporter: first tt to be reactivated by the traffic light
trlightq	Logical attribute of tlquay: defines if the tt can drive on

2.17 Module TLYPROCESS

[5-15]:
This module concerns the actions of the traffic light at the yard and is activated the moment a tt enters the *yardqueue* (if not already active).
This module operates identical to the previous described module TLQPROCESS (chapter 2.15) and thus only the attributes that differ in name will be explained.

tlytt	Reference to transporter: first tt to be reactivated by the traffic light
trlighty	Logical attribute of tlyard: defines if the tt can drive on

MODEL ICT 1
MOD RTGCPROCESS

Date: 98/09/14
Time: 10:20:11

```
@@@@@@@@@@@@@@@@
@ RTGC process @
@@@@@@@@@@@@@@@@

rolltolocation:
start:
IF rtgctype="xrtgc"
   IF xrtgcstop=TRUE
      xrtgcstop<FALSE
      rtgcship<NONE
      MOVE rtgcfig[rtgcnr] TO SINK           @anim@
      PASSIVATE
   END
END
ENTER noworkrtgc
WAIT UNTIL (ttq[rtgcnr] IS NOT EMPTY)|(truckq[rtgcnr] IS NOT EMPTY)
LEAVE noworkrtgc
CALL rtgcstrategy
WORK rtgcdrivetime
WORK rtgcreshuftime

IF trbox OF rtgcserveunit>0
   WORK rtgcrepostime
   WORK rtgcliftofftime
   IF (trbox OF rtgcserveunit=2) & (trload OF rtgcserveunit≠"fmt")
      WORK rtgcrepostime
      WORK rtgcliftofftime
   END
   CALL yardload
   IF trtype OF rtgcserveunit="truck"
      ACTIVATE rtgcserveunit FROM byetruck IN truckprocess
   END
   IF trtype OF rtgcserveunit="tt"
      ACTIVATE rtgcserveunit FROM leaveyard IN ttprocess
   END
   GOTO newlocation
END

IF trbox OF rtgcserveunit=0
   CALL yardload
   WORK rtgcliftontime
   IF trbox OF rtgcserveunit=2
      WORK rtgcrepostime
      WORK rtgcliftontime
   END
   IF trtype OF rtgcserveunit="truck"
      ACTIVATE rtgcserveunit FROM byetruck IN truckprocess
   END
   IF trtype OF rtgcserveunit="tt"
      ACTIVATE rtgcserveunit FROM leaveyard IN ttprocess
   END
   WORK rtgcrepostime
END

newlocation:
IF rtgctype="xrtgc"
   IF rtgcnr=10
      IF LENGTH OF lane[10]≤5
         LEAVE xrtgcworkset
         MOVE rtgcfig[10] TO SINK            @anim@
         PASSIVATE
      END
   END
END
REPEAT FROM rolltolocation
```

2.18 Module RTGCPROCESS

This module describes the RTGC actions and is activated by tractor-trailers or trucks entering a queue to be serviced by the RTGC. Originally the RTGC's are activated by the MAIN module.

[5-20]:
The RTGC's will wait until a tractor-trailer or truck enters the queue to be serviced by the RTGC. Depending on the servicing strategy (macro RTGCSTRATEGY is activated), the RTGC will select either the first vehicle or the first tt present in the queue. The RTGC will then perform some standard actions, such as driving to the correct location and reshuffle the stack. For further information on these actions the reader is referred to the report chapter 5.3.3.

[22-37]:
This section concerns the unloading of a tractor-trailer or truck. The RTGC will first reposition itself and then lift the container(s) off of the transporter. The macro YARDLOAD is activated during the servicing of a transporter. This macro is responsible for updating the stack specifics and giving the transporter a temporary new destination. When the transporter is unloaded it is reactivated and leaves the lane it is presently in.
The RTGC will now service the next transporter in the queue if there is one, otherwise it will wait till one arrives.

rtgcserveunit	Reference to transporter: transporter being serviced by the RTGC
YARDLOAD	Macro: defines the updating of the containers in the stack and determines which load is to be lifted out off the stack

[39-65]:
This section concerns the loading of a transporter by the RTGC. First the macro YARDLOAD is activated in which the load to be lifted onto the transporter is defined. After performing the necessary actions the RTGC reactivates the transporter.
If the RTGC is the second RTGC active in the empties stack it will check if it is still required to remain operating. If not, it will become passive. Otherwise it will service the next transporter waiting in the queue.
If the RTGC isn't the second RTGC it will service the next transporter in the queue (if there is one).

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ determining how the tt's are (un)loaded by the rtgc's @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

LOCAL:
REAL: z1 r5

rtgcship<trworkship OF rtgcserveunit
IF (trtype OF rtgcserveunit="tt")
   IF trbox OF rtgcserveunit>0
      IF trload OF rtgcserveunit="fmt"
         stackimp20f OF rtgcship<stackimp20f OF rtgcship+1
         stackimp20 OF rtgcship<stackimp20 OF rtgcship+1
         rtgcstackvol[rtgcnr]<rtgcstackvol[rtgcnr]+1
         MOVE stackvol[rtgcnr] TO rtgcstackvol[rtgcnr]      @anim@
         trteu OF rtgcserveunit<1
         trload OF rtgcserveunit<"mt"
         trbox OF rtgcserveunit<1
         trdestination OF rtgcserveunit<"impemptystack"
         RETURN
      END
      IF trload OF rtgcserveunit="f"
         IF trbox OF rtgcserveunit=1
            stackimp20f OF rtgcship<stackimp20f OF rtgcship+1 IF trteu OF rtgcserveunit=1
            stackimp20 OF rtgcship<stackimp20 OF rtgcship+1 IF trteu OF rtgcserveunit=1
            stackimp40f OF rtgcship<stackimp40f OF rtgcship+1 IF trteu OF rtgcserveunit=2
            stackimp40 OF rtgcship<stackimp40 OF rtgcship+1 IF trteu OF rtgcserveunit=2
            rtgcstackvol[rtgcnr]<rtgcstackvol[rtgcnr]+trteu OF rtgcserveunit
            MOVE stackvol[rtgcnr] TO rtgcstackvol[rtgcnr]  @anim@
         END
         IF trbox OF rtgcserveunit=2
            stackimp20f OF rtgcship<stackimp20f OF rtgcship+2 IF trteu OF rtgcserveunit=2
            stackimp20 OF rtgcship<stackimp20 OF rtgcship+2
            rtgcstackvol[rtgcnr]<rtgcstackvol[rtgcnr]+2
            MOVE stackvol[rtgcnr] TO rtgcstackvol[rtgcnr]  @anim@
         END
      END
      IF trload OF rtgcserveunit="mt"
         IF trbox OF rtgcserveunit=1
            stackimp20mt OF rtgcship<stackimp20mt OF rtgcship+1 IF trteu OF rtgcserveunit=1
            stackimp20 OF rtgcship<stackimp20 OF rtgcship+1 IF trteu OF rtgcserveunit=1
            stackimp40mt OF rtgcship<stackimp40mt OF rtgcship+1 IF trteu OF rtgcserveunit=2
            stackimp40 OF rtgcship<stackimp40 OF rtgcship+1 IF trteu OF rtgcserveunit=2
            rtgcstackvol[rtgcnr]<rtgcstackvol[rtgcnr]+trteu OF rtgcserveunit
            MOVE stackvol[rtgcnr] TO rtgcstackvol[rtgcnr]  @anim@
         END
         IF trbox OF rtgcserveunit=2
            stackimp20mt OF rtgcship<stackimp20mt OF rtgcship+2 IF trteu OF rtgcserveunit=2
            stackimp20 OF rtgcship<stackimp20 OF rtgcship+2
            rtgcstackvol[rtgcnr]<rtgcstackvol[rtgcnr]+2
            MOVE stackvol[rtgcnr] TO rtgcstackvol[rtgcnr]  @anim@
         END
      END
      trdestination OF rtgcserveunit<"importquay"
      trworkship OF rtgcserveunit<NONE
      trteu OF rtgcserveunit<0
      trload OF rtgcserveunit<"o"
      trbox OF rtgcserveunit<0
      RETURN
   END
   IF trbox OF rtgcserveunit=0
      IF trdestination OF rtgcserveunit="exportstack"
         z1<((stackexp20f OF rtgcship)/(stackexp20f OF rtgcship+stackexp40f OF rtgcship))
      END
      IF trdestination OF rtgcserveunit="expemptystack"
         z1<((stackexp20mt OF rtgcship)/(stackexp20mt OF rtgcship+stackexp40mt OF rtgcship))
      END
      r5<unif
      IF r5≤z1
         trteu OF rtgcserveunit<1
```

2.19 Macro YARDLOAD

This macro is activated by the RTGC while servicing a tractor-trailer or truck.

[8-132]:
When the transporter being serviced is a tractor-trailer these lines are followed. This tt can either carry a load or be empty.
If the tt carries a load the situation in the stack is updated and the tt receives a temporary destination. If the tt carries a mixed load (a full and an empty container) first the full container is added to the stack and the tt will remain with one empty container. This tt then receives the empties stack as destination.
If the tt carries no load, the tt is loaded with full or empty export containers. Whenever possible the tt receives two 20 ft containers or one 40 ft container.
After the stack volumes have been updated, the macro returns to the module RTGCPROCESS.

rtgcstackvol Integer attribute of MAIN: number of containers in the stack (in TEU)

[133-178]:
These lines are followed when a truck is being serviced by the RTGC. A truck will be either loaded (with a full or empty load) or empty.
If the truck is loaded, the stack volume is updated and when the truck is empty, it will receive a one or two TEU load. After this is accomplished, the macro returns to the RTGCPROCESS module.

```
            trbox OF rtgcserveunit<1
            IF trdestination OF rtgcserveunit="exportstack"
               stackexp20f OF rtgcship<stackexp20f OF rtgcship-1
               stackexp20 OF rtgcship<stackexp20 OF rtgcship-1
               rtgcstackvol[rtgcnr]<rtgcstackvol[rtgcnr]-1
               MOVE stackvol[rtgcnr] TO rtgcstackvol[rtgcnr]  @anim@
               trload OF rtgcserveunit<"f"
               IF stackexp20f OF rtgcship>0
                  trteu OF rtgcserveunit<2
                  trbox OF rtgcserveunit<2
                  stackexp20f OF rtgcship<stackexp20f OF rtgcship-1
                  stackexp20 OF rtgcship<stackexp20 OF rtgcship-1
                  rtgcstackvol[rtgcnr]<rtgcstackvol[rtgcnr]-1
                  MOVE stackvol[rtgcnr] TO rtgcstackvol[rtgcnr] @anim@
               END
               REMOVE rtgcserveunit FROM shfulltt OF rtgcship
               GOTO point4
            END
            IF trdestination OF rtgcserveunit="expemptystack"
               stackexp20mt OF rtgcship<stackexp20mt OF rtgcship-1
               stackexp20 OF rtgcship<stackexp20 OF rtgcship-1
               rtgcstackvol[rtgcnr]<rtgcstackvol[rtgcnr]-1
               MOVE stackvol[rtgcnr] TO rtgcstackvol[rtgcnr]  @anim@
               trload OF rtgcserveunit<"mt"
               IF stackexp20mt OF rtgcship>0
                  trteu OF rtgcserveunit<2
                  trbox OF rtgcserveunit<2
                  stackexp20mt OF rtgcship<stackexp20mt OF rtgcship-1
                  stackexp20 OF rtgcship<stackexp20 OF rtgcship-1
                  rtgcstackvol[rtgcnr]<rtgcstackvol[rtgcnr]-1
                  MOVE stackvol[rtgcnr] TO rtgcstackvol[rtgcnr] @anim@
               END
               REMOVE rtgcserveunit FROM shemptytt OF rtgcship
            END
            point4:
            trdestination OF rtgcserveunit<"exportquay"
            RETURN
         END
         IF r5>z1
            trteu OF rtgcserveunit<2
            trbox OF rtgcserveunit<1
            IF trdestination OF rtgcserveunit="exportstack"
               stackexp40f OF rtgcship<stackexp40f OF rtgcship-1
               stackexp40 OF rtgcship<stackexp40 OF rtgcship-1
               rtgcstackvol[rtgcnr]<rtgcstackvol[rtgcnr]-2
               MOVE stackvol[rtgcnr] TO rtgcstackvol[rtgcnr]  @anim@
               trload OF rtgcserveunit<"f"
               REMOVE rtgcserveunit FROM shfulltt OF rtgcship
            END
            IF trdestination OF rtgcserveunit="expemptystack"
               stackexp40mt OF rtgcship<stackexp40mt OF rtgcship-1
               stackexp40 OF rtgcship<stackexp40 OF rtgcship-1
               rtgcstackvol[rtgcnr]<rtgcstackvol[rtgcnr]-2
               MOVE stackvol[rtgcnr] TO rtgcstackvol[rtgcnr]  @anim@
               trload OF rtgcserveunit<"mt"
               REMOVE rtgcserveunit FROM shemptytt OF rtgcship
            END
            trdestination OF rtgcserveunit<"exportquay"
            RETURN
         END
      END
END
IF trtype OF rtgcserveunit="truck"
   IF trload OF rtgcserveunit="f"
      stackexp20f OF rtgcship<stackexp20f OF rtgcship+1 IF trteu OF rtgcserveunit=1
      stackexp20 OF rtgcship<stackexp20 OF rtgcship+1 IF trteu OF rtgcserveunit=1
      stackexp40f OF rtgcship<stackexp40f OF rtgcship+1 IF trteu OF rtgcserveunit=2
      stackexp40 OF rtgcship<stackexp40 OF rtgcship+1 IF trteu OF rtgcserveunit=2
      rtgcstackvol[rtgcnr]<rtgcstackvol[rtgcnr]+trteu OF rtgcserveunit
      MOVE stackvol[rtgcnr] TO rtgcstackvol[rtgcnr]        @anim@
      trteu OF rtgcserveunit<0
      trload OF rtgcserveunit<"exp"
      trbox OF rtgcserveunit<0
      RETURN
   END
   IF trload OF rtgcserveunit="mt"
      stackexp20mt OF rtgcship<stackexp20mt OF rtgcship+1 IF trteu OF rtgcserveunit=1
      stackexp20 OF rtgcship<stackexp20 OF rtgcship+1 IF trteu OF rtgcserveunit=1
      stackexp40mt OF rtgcship<stackexp40mt OF rtgcship+1 IF trteu OF rtgcserve
```

```
unit=2
      stackexp40 OF rtgcship<stackexp40 OF rtgcship+1 IF trteu OF rtgcserveunit
=2
      rtgcstackvol[rtgcnr]<rtgcstackvol[rtgcnr]+trteu OF rtgcserveunit
      MOVE stackvol[rtgcnr] TO rtgcstackvol[rtgcnr]        @anim@
      trteu OF rtgcserveunit<0
      trload OF rtgcserveunit<"exp"
      trbox OF rtgcserveunit<0
      RETURN
   END
   IF trload OF rtgcserveunit="o"
      trbox OF rtgcserveunit<1
      IF trteu OF rtgcserveunit=1
         stackimp20f OF rtgcship<stackimp20f OF rtgcship-1 IF trdestination
         OF rtgcserveunit="importstack"
         stackimp20mt OF rtgcship<stackimp20mt OF rtgcship-1 IF trdestination
         OF rtgcserveunit="impemptystack"
         stackimp20 OF rtgcship<stackimp20 OF rtgcship-1
         rtgcstackvol[rtgcnr]<rtgcstackvol[rtgcnr]-1
         MOVE stackvol[rtgcnr] TO rtgcstackvol[rtgcnr]     @anim@
      END
      IF trteu OF rtgcserveunit=2
         stackimp40f OF rtgcship<stackimp40f OF rtgcship-1 IF trdestination
         OF rtgcserveunit="importstack"
         stackimp40mt OF rtgcship<stackimp40mt OF rtgcship-1 IF trdestination
         OF rtgcserveunit="impemptystack"
         stackimp40 OF rtgcship<stackimp40 OF rtgcship-1
         rtgcstackvol[rtgcnr]<rtgcstackvol[rtgcnr]-2
         MOVE stackvol[rtgcnr] TO rtgcstackvol[rtgcnr]     @anim@
      END
      trload OF rtgcserveunit<"f" IF trdestination OF rtgcserveunit="importstac
k"
      trload OF rtgcserveunit<"mt" IF trdestination OF rtgcserveunit="impemptys
tack"
   END
   RETURN
END
```

MODEL ICT 1
MAC EXPGEN_GEN

Date: 98/09/09
Time: 12:22:47

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ initialisation of ship export truck generator @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

shexportgen<NEW exportgen
egjob OF shexportgen<THIS ship
egnrtruck OF shexportgen<(shexp40+shexp20)
egexp20 OF shexportgen<shexp20
egexp20f OF shexportgen<shexp20f
egexp20mt OF shexportgen<shexp20mt
egexp40 OF shexportgen<shexp40
egexp40f OF shexportgen<shexp40f
egexp40mt OF shexportgen<shexp40mt
ACTIVATE shexportgen FROM export IN exportgenprocess
RETURN
```

MODEL ICT 1
MAC IMPGEN_GEN

Date: 98/09/09
Time: 12:22:48

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ initialisation of ship import truck generator @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

shimportgen<NEW importgen
igjob OF shimportgen<THIS ship
ignrtrans OF shimportgen<(shimp20+shimp40)
igimp20 OF shimportgen<shimp20
igimp20f OF shimportgen<shimp20f
igimp20mt OF shimportgen<shimp20mt
igimp40 OF shimportgen<shimp40
igimp40f OF shimportgen<shimp40f
igimp40mt OF shimportgen<shimp40mt
igimprow OF shimportgen<shimportbay
ACTIVATE shimportgen WITH DELAY 1 DAY FROM import IN importgenprocess
RETURN
```

2.20 Macro EXPGEN_GEN

[5-15]:
This macro is activated by the module SHIPGENERATORPROCESS. In this macro an export generator is created responsible for generating the total number of trucks required to deliver all the export containers to the container terminal for that particular ship. Also the number of different types of containers required are defined here. These trucks are generated in the module EXPORTGENPROCESS that is activated before the macro returns to the SHIPGENERATOR module.

egjob	Reference to ship: ship this export generator generates trucks for
egnrtruck	Integer attribute of exportgen: total number of trucks needed to deliver the required total number of containers
egexp20	Integer attribute of exportgen: number of 20 ft containers
egexp20f	Integer attribute of exportgen: number of full 20 ft containers
egexp20mt	Integer attribute of exportgen: number of empty 20 ft containers
egexp40	Integer attribute of exportgen: number of 40 ft containers
egexp40f	Integer attribute of exportgen: number of full 40 ft containers
egexp40mt	Integer attribute of exportgen: number of empty 40 ft containers

2.21 Macro IMPGEN_GEN

[5-16]:
This macro is activated by the SHIPPROCESS module and creates an import generator that will generate the trucks responsible for picking-up all the containers brought to the terminal by that particular ship. The specific number of trucks needed and the load that has to be fetched are determined in this macro. After activating the import generator in the module IMPORTGENPROCESS with a delay of one day, the macro returns to the SHIPPROCESS module.

igjob	Reference to ship: ship this import generator generates trucks for
ignrtrans	Integer attribute of importgen: total number of trucks needed to pick-up the required total number of containers
igimp20	Integer attribute of importgen: number of 20 ft containers
igimp20f	Integer attribute of importgen: number of full 20 ft containers
igimp20mt	Integer attribute of importgen: number of empty 20 ft containers
igimp40	Integer attribute of importgen: number of 40 ft containers
igimp40f	Integer attribute of importgen: number of full 40 ft containers
igimp40mt	Integer attribute of importgen: number of empty 40 ft containers
igimprow	Integer attribute of importgen: import stack number of the ship whose containers are to be picked-up

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ generating the trucks that bring the containers to the yard @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

export:
egdayexp<(egnrtruck/100)
trexplst<NEW SET
FOR j<1 TO 7
   egtrucknr<FLOOR(5*egdayexp)     IF j=1
   egtrucknr<FLOOR(10*egdayexp)    IF j=2
   egtrucknr<FLOOR(20*egdayexp)    IF j=3
   egtrucknr<FLOOR(25*egdayexp)    IF j=4
   egtrucknr<FLOOR(20*egdayexp)    IF j=5
   egtrucknr<FLOOR(10*egdayexp)    IF j=6
   egtrucknr<(egnrtruck-egtrucks)  IF j=7
   presentexptime<NOW HOURS
   GOTO nextday IF egtrucknr=0

   FOR m<1 TO 10
      TABULATE nrtab[m] IN trucktab AT 0.1*m
   END
   FOR i<1 TO egtrucknr
      r6<unif
      egtransp<NEW transporter
      trworkship OF egtransp<egjob
      trarrival OF egtransp<(NOW + VALUE OF trucktab AT(r6))
      trtype OF egtransp<"truck"
      JOIN egtransp TO trexplst @RANKED BY trarrival OF egtransp
      trbox OF egtransp<1
      egtrucks<egtrucks+1
   END
   FOR o<1 TO egtrucknr
      egtransp<FIRST transporter IN trexplst WITH SMALLEST trarrival
      WAIT ((trarrival OF egtransp)-NOW) HOURS
      REMOVE egtransp FROM trexplst
      r6<unif
      IF r6≤(egexp20/(egexp20+egexp40))
         trteu OF egtransp<1
         r7<unif
         IF (r7≤0.5)&(egexp20f=0)
            r7<0.7
         END
         IF (r7>0.5)&(egexp20mt=0)
            r7<0.3
         END
         IF r7≤0.5
            egexp20<egexp20-1
            egexp20f<egexp20f-1
            trdestination OF egtransp<"exportstack"
            trrow OF egtransp<shexportbay OF egjob
            trload OF egtransp<"f"
            GOTO activatetruck
         END
         IF r7>0.5
            egexp20<egexp20-1
            egexp20mt<egexp20mt-1
            trdestination OF egtransp<"expemptystack"
            trrow OF egtransp<10
            trload OF egtransp<"mt"
            GOTO activatetruck
         END
      END
      IF r6>(egexp20/(egexp20+egexp40))
         trteu OF egtransp<2
         r7<unif
         IF (r7≤0.5)&(egexp40f=0)
            r7<0.7
         END
         IF (r7>0.5)&(egexp40mt=0)
            r7<0.3
         END
         IF r7≤0.5
            egexp40<egexp40-1
            egexp40f<egexp40f-1
            trdestination OF egtransp<"exportstack"
            trrow OF egtransp<shexportbay OF egjob
            trload OF egtransp<"f"
            GOTO activatetruck
         END
         IF r7>0.5
            egexp40<egexp40-1
            egexp40mt<egexp40mt-1
```

2.22 Module EXPORTGENPROCESS

[5-95]:
This module is created and subsequently activated by the macro EXPGEN_GEN. Its actions start at the label *export*.
The total number of trucks is generated in a time span of 7 days according to the distribution described in the report (chapter 5.4.7.1). The arrival distribution per day is also described in the report (chapter 5.5.1.14).
Each generated truck is given a certain load and subsequently activated in the module TRUCKPROCESS.
The export generator is terminated after the last truck is generated and activated in the TRUCKPROCESS module.

egdayexp	Real attribute of exportgen: total number of trucks to be generated divided by 100
trexplst	Set attribute of exportgen: set to which trucks that will arrive on the same day (at different times) are joined to
egtrucks	Integer attribute of exportgen: number of trucks already generated
presentexptime	Real attribute of exportgen: present time
trucktab	Table of MAIN: table with arrival distribution of the trucks per day
egtransp	Reference to transporter: truck that is given attributes by the export generator
trarrival	Real attribute of transporter: truck arrival time at the terminal

```
            trdestination OF egtransp<"expemptystack"
            trrow OF egtransp<10
            trload OF egtransp<"mt"
            GOTO activatetruck
         END
      END
      activatetruck:
      ACTIVATE egtransp FROM truckarrival IN truckprocess
   END
   nextday:
   WAIT ((presentexptime+24)-NOW) HOURS
END
TERMINATE
```

MODEL ICT 1
MOD IMPORTGENPROCESS

Date: 98/09/09
Time: 12:22:49

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ generating the trucks that take the containers from the yard @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

import:
igdayimp<(ignrtrans/100)
trimplst<NEW SET
FOR k<1 TO 10
    igtrucknr<FLOOR(1*igdayimp)    IF k=1
    igtrucknr<FLOOR(3*igdayimp)    IF k=2
    igtrucknr<FLOOR(5*igdayimp)    IF k=3
    igtrucknr<FLOOR(8*igdayimp)    IF k=4
    igtrucknr<FLOOR(10*igdayimp)   IF k=5
    igtrucknr<FLOOR(10*igdayimp)   IF k=6
    igtrucknr<FLOOR(13*igdayimp)   IF k=7
    igtrucknr<FLOOR(13*igdayimp)   IF k=8
    igtrucknr<FLOOR(25*igdayimp)   IF k=9
    igtrucknr<(ignrtrans-igtrucks) IF k=10
    presentimptime<NOW HOURS
    GOTO nextday IF igtrucknr=0

    FOR p<1 TO 10
       TABULATE nrtab[p] IN trucktab AT 0.1*p
    END
    FOR l<1 TO igtrucknr
       igimptransp<NEW transporter
       igtrucks<igtrucks+1
       trworkship OF igimptransp<igjob
       trtype OF igimptransp<"truck"
       trload OF igimptransp<"o"
       trbox OF igimptransp<0
       trteu OF igimptransp<0
       r8<unif
       trarrival OF igimptransp<(NOW + VALUE OF trucktab AT(r8))
       JOIN igimptransp TO trimplst @RANKED BY trarrival OF igimptransp
    END
    FOR q<1 TO igtrucknr
       igimptransp<FIRST transporter IN trimplst WITH SMALLEST trarrival
       CALL teupertruck
       WAIT ((trarrival OF igimptransp)-NOW) HOURS
       REMOVE igimptransp FROM trimplst
       ACTIVATE igimptransp FROM truckarrival IN truckprocess
    END
    nextday:
    WAIT ((presentimptime+24)-NOW) HOURS
END

TERMINATE
```

2.23 Module IMPORTGENPROCESS

[5-48]:
This module is activated by the macro IMPGEN_GEN and is responsible for generating trucks that pick-up the import containers retrieved from the ship.
The trucks are generated in a time span of 10 days according to the distribution mentioned in the report chapter 5.4.7.2. The arrival distribution of the trucks per day is the same as for the trucks generated by the export generator.
Before a truck is activated in the module TRUCKPROCESS, the macro TEUPERTRUCK is activated which determines the exact nature of the load to be picked-up by the truck.
When the last truck is activated in the TRUCKPROCESS module, the import generator is terminated.

igdayimp	Real attribute of importgen: total number of trucks to be generated divided by 100
trimplst	Set attribute of importgen: set to which trucks that will arrive at the terminal on the same day (at different times) are joined
igtrucknr	Integer attribute of importgen: number of trucks to be generated on a day
igtrucks	Integer attribute of importgen: number of trucks already generated
presentimptime	Real attribute of importgen: present time
igimptransp	Reference to transporter: truck being given attributes to
TEUPERTRUCK	Macro: defines the load to be picked-up by the truck

MODEL ICT 1
MAC TEUPERTRUCK

Date: 98/09/09
Time: 12:26:13

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ determining the truck import load @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

LOCAL:
REAL: r8 r9

r8<unif
IF r8≤(igimp20/(igimp20+igimp40))
   trteu OF igimptransp<1
   r9<unif
   IF (r9≤0.3)&(igimp20f=0)
      r9<0.7
   END
   IF (r9>0.3)&(igimp20mt=0)
      r9<0.2
   END
   IF r9≤0.3
      igimp20<igimp20-1
      igimp20f<igimp20f-1
      trdestination OF igimptransp<"importstack"
      trrow OF igimptransp<igimprow
      RETURN
   END
   IF r9>0.3
      igimp20<igimp20-1
      igimp20mt<igimp20mt-1
      trdestination OF igimptransp<"impemptystack"
      trrow OF igimptransp<10
      RETURN
   END
END
IF r8>(igimp20/(igimp20+igimp40))
   trteu OF igimptransp<2
   r9<unif
   IF (r9≤0.3)&(igimp40f=0)
      r9<0.7
   END
   IF (r9>0.3)&(igimp40mt=0)
      r9<0.2
   END
   IF r9≤0.3
      igimp40<igimp40-1
      igimp40f<igimp40f-1
      trdestination OF igimptransp<"importstack"
      trrow OF igimptransp<igimprow
      RETURN
   END
   IF r9>0.3
      igimp40<igimp40-1
      igimp40mt<igimp40mt-1
      trdestination OF igimptransp<"impemptystack"
      trrow OF igimptransp<10
      RETURN
   END
END
RETURN
```

2.24 Macro TEUPERTRUCK

[8-57]:
This macro is activated by the import generator to determine the exact load that the truck has to pick-up from the terminal yard.
The macro determines the nature of the load that is to be picked-up. The load can be either one or two TEU's. The truck also receives instructions as to which stack it has to drive to: either the import stack or the empties stack.
Once the truck has received its load information, the macro will return to the module IMPORTGENPROCESS.

MODEL ICT 1
MOD TRUCKPROCESS

Date: 98/09/14
Time: 13:55:01

```
@@@@@@@@@@@@@@@@@@@
@ truck process @
@@@@@@@@@@@@@@@@@@@

truckarrival:
ENTER gatequeue
ENTER tgmq[trrow]
MOVE tgmqfig[trrow] TO LENGTH OF tgmq[trrow]
ACTIVATE truckgatemaster FROM begin IN truckgateprocess IF truckgatemaster IS NOT ACTIVE
WAIT
LEAVE gatequeue
LEAVE tgmq[trrow]
MOVE tgmqfig[trrow] TO LENGTH OF tgmq[trrow]
ENTER tlgqueue
ACTIVATE tlgate FROM check IN tlgprocess IF tlgate IS NOT ACTIVE
WAIT
LEAVE tlgqueue
ENTER yard
tringate<tringate+1
landboxin<landboxin+1 IF (trload="f")|(trload="mt")
ENTER maintraffic
WORK mainlanetime
LEAVE maintraffic
ENTER intraffic
WORK (CEIL((trrow*30)/mainlanespeed)) SECONDS
LEAVE intraffic
ENTER lane[trrow]
WORK (CEIL(210/mainlanespeed)) SECONDS
ENTER truckq[trrow]
trqueuetime<NOW
WAIT

byetruck:
WORK (CEIL(210/mainlanespeed)) SECONDS
LEAVE lane[trrow]
ENTER outtraffic
WORK (CEIL((trrow*30)/mainlanespeed)) SECONDS
LEAVE outtraffic
ENTER leavingtrucks
ACTIVATE truckgatemaster FROM atruckisleaving IN truckgateprocess
LEAVE yard
troutgate<troutgate+1
landboxout<landboxout+1 IF (trload="f")|(trload="mt")
TERMINATE
```

2.25 Module TRUCKPROCESS

[5-30]:
A truck is generated by either an export or import generator. The truck follows the actions described in this module.
Upon arrival at the terminal gate, the truck enters a queue at the gate (the *gatequeue*). By doing so the gate master is activated (module TRUCKGATEPROCESS). This truck gate master is responsible for giving the truck permission to enter the terminal yard.
When the truck is granted permission to enter the terminal yard, the truck first enters a queue for a traffic light (*tlgqueue*). When the traffic light (module TLGPROCESS) reactivates the truck, the truck enters the yard and drives to the stack designated to be serviced by a RTGC.

tgmq[ ]	Queue: contains trucks waiting for permission to enter the terminal yard
tlgqueue	Queue: contains trucks waiting for the traffic light
yard	Queue: contains trucks present in the yard

[32-43]:
This section concerns the truck actions after it has been serviced by a RTGC. The truck will drive to the gate and enters the set *leavingtrucks* which activates the gate master (if not already active) and then leaves the yard. After it has left the yard the truck is terminated.

MODEL ICT 1
MOD TRUCKGATEPROCESS

Date: 98/09/09
Time: 12:26:42

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ determining if the truck can enter the yard @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

begin:
tgmtruck<FIRST transporter IN gatequeue

actruck:
IF tgmrowopen[trrow OF tgmtruck]=TRUE
   IF LENGTH OF tgmtruckq[trrow OF tgmtruck]≤5
      JOIN tgmtruck TO tgmtruckq[trrow OF tgmtruck]
      REACTIVATE tgmtruck
   END
   GOTO check
END
IF tgmrowopen[trrow OF tgmtruck]=FALSE
   GOTO check
END

atruckisleaving:
tgmtruck<FIRST transporter IN leavingtrucks
tgmrow<trrow OF tgmtruck
REMOVE tgmtruck FROM tgmtruckq[trrow OF tgmtruck]
REMOVE tgmtruck FROM leavingtrucks
tgmtruck<FIRST transporter IN gatequeue WITH trrow=tgmrow
PASSIVATE IF tgmtruck IS NONE
REPEAT FROM actruck

check:
tgmtruck<SUCC OF tgmtruck IN gatequeue
REPEAT FROM actruck IF tgmtruck IS NOT NONE

PASSIVATE
```

2.26 Module TRUCKGATEPROCESS

[5-18]:
This module is activated the moment a truck enters the *gatequeue* and concerns the gate master actions. The first truck in this set is selected and the gate master checks whether the truck can be granted permission to enter the yard. If this is the case, the truck is reactivated. If not, the truck remains in the set. The next truck in the set (if any) is then selected and the actions are repeated.

tgmtruck	Reference to transporter: truck for which the gate master is checking the yard situation

[20-33]:
This section describes the actions taken by the gate master when activated by a truck that is leaving the yard. The gate master will check whether there is a truck that it can give permission to enter the yard (in case a truck has just left the yard). If not, the gate master becomes passive. If there is, the gate master reactivates this truck and then repeats the actions described above.

tgmrow	Integer attribute of truckgatemaster: row number of the truck just leaving the yard

MODEL ICT 1
MOD TLGPROCESS

Date: 98/09/09
Time: 12:26:42

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ trafficlightprocess gate to yard @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

start:
WAIT UNTIL tlgqueue IS NOT EMPTY

check:
tlgtt<FIRST transporter IN tlgqueue
REACTIVATE tlgtt
trlightg<FALSE
WAIT 4 SECONDS
trlightg<TRUE
REPEAT FROM start IF tlgqueue IS EMPTY
REPEAT FROM check IF tlgqueue IS NOT EMPTY
```

2.27 Module TLGPROCESS

[5-15]:
This module concerns the actions of the traffic light at the gate and is activated the moment a truck enters the *tlgqueue* (if not already active). The traffic light reactivates the truck and will repeat its actions after four seconds if there are other trucks waiting in the queue to enter the yard. Otherwise the traffic light becomes passive.

tlgtt	Reference to transporter: truck that will be reactivated by the traffic light
trlightg	Logical attribute of tlgate: defines if the truck is allowed to enter the yard

MODEL ICT 1
MAC TTPRIORITY

Date: 98/09/09
Time: 12:15:59

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ tt's have priority in the yard @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

IF ttq[rtgcnr] IS NOT EMPTY
   rtgcserveunit<FIRST OF ttq[rtgcnr]
   REMOVE rtgcserveunit FROM ttq[rtgcnr]
   RETURN
END
IF truckq[rtgcnr] IS NOT EMPTY
   rtgcserveunit<FIRST OF truckq[rtgcnr]
   REMOVE rtgcserveunit FROM truckq[rtgcnr]
   RETURN
END
```

MODEL ICT 1
MAC NOPRIORITY

Date: 98/09/14
Time: 13:55:00

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ tt's have no priority in the yard @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

LOCAL:
REAL: z2 z3

IF ttq[rtgcnr] IS EMPTY
   rtgcserveunit<FIRST OF truckq[rtgcnr]
   REMOVE rtgcserveunit FROM truckq[rtgcnr]
   RETURN
END
IF truckq[rtgcnr] IS EMPTY
   rtgcserveunit<FIRST OF ttq[rtgcnr]
   REMOVE rtgcserveunit FROM ttq[rtgcnr]
   RETURN
END

z2<trqueuetime OF FIRST OF ttq[rtgcnr]
z3<trqueuetime OF FIRST OF truckq[rtgcnr]
IF z2≤z3
   rtgcserveunit<FIRST OF ttq[rtgcnr]
   REMOVE rtgcserveunit FROM ttq[rtgcnr]
END
IF z2>z3
   rtgcserveunit<FIRST OF truckq[rtgcnr]
   REMOVE rtgcserveunit FROM truckq[rtgcnr]
END
RETURN
```

2.28 Macro TTPRIORITY

[5-14]:
In case the user has specified that a tt priority strategy is to be followed in the terminal yard, this macro will be activated by the RTGC when selecting a transporter from the waiting queue.
In this case a tt will be serviced first, even if a truck did arrive earlier in the queue. When only trucks are present in the queue, a truck will be selected.
After a transporter has been selected the macro returns to the RTGCPROCESS module.

2.29 Macro NOPRIORITY

[8-29]:
This macro is activated if the user has specified that the tractor-trailers will have no priority in the container yard. When a RTGC is ready to service a transporter, this macro is activated.
This macro selects the transporter that has entered the waiting queue first. After having done this, the macro returns to the RTGCPROCESS module.

MODEL ICT 1
MAC INPUTBLOCK

Date: 98/09/09
Time: 11:15:12

```
@@@@@@@@@@@@@@@@@@@@@@@
@ preset model input @
@@@@@@@@@@@@@@@@@@@@@@@

date<"not known"
netoperation<READ FROM situation
priority<"y"
runtime<READ FROM situation
nrpt<READ FROM situation
nrtt<READ FROM situation
ttperpt<READ FROM situation
nrrtgc<READ FROM situation
mainlanespeed<(1000/3600)*READ FROM situation
bridgespeed<(1000/3600)*READ FROM situation
quayspeed<(1000/3600)*READ FROM situation
rtgcspeed<(1000/3600)*READ FROM situation
RETURN
```

MODEL ICT 1
MAC OUTPUT

Date: 98/09/09
Time: 12:37:01

```
LOCAL:
INTEGER: z zz

WRITE "run nummer:";runnr;"on";simdate TO shipoutput WITH IMAGE xxxxxxxxxx^xx^
xx^xxxxxxxxxx
WRITE "number of tt per pt:";ttperpt;",total number of tt's:";nrtt;"influence f
actor:";netoperation TO shipoutput WITH IMAGE xxxxxxxxxxxxxxxxxxx^xx^xxxxxxxxx
xxxxxxxxxxxxx^xx^xxxxxxxxxxxxxxxxxx^x.xx

WRITE "Average inter arrival time:";iattimetot/nrofship TO shipoutput WITH IMAG
E     xxxxxxxxxxxxxxxxxxxxxxxxxxx^xx.xx
WRITE "Average ship turn-around time:";shiptotstime/992 TO shipoutput WITH IMAG
E xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx^xx.xx
WRITE "Leaving time last ship:";NOW TO shipoutput WITH IMAGE xxxxxxxxxxxxxxxxxx
xxxxx^xxxx.xx
WRITE "Average waitingtime in anchorage:";shiptotanchtime/992 TO shipoutput WIT
H IMAGE  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx^xx.xx
FOR z<1 TO 4
    mmpt<FIRST portainer IN passivept WITH ptnr=z
    WRITE "portainer";z;"production:";(ptnrbox OF mmpt/pttottime OF mmpt);
    "box/hour| real production:";(ptnrbox OF mmpt/pttot2time OF mmpt);
    "box/hour"TO shipoutput WITH IMAGE xxxxxxxxxxx^x^xxxxxxxxxxx^xx.x^xxxxxxxxx
xxxxxxxxxxxxxxxx^xx.x^xxxxxxxx
    WRITE "portainer";z;"occupancy:";pttot2time OF mmpt/(NOW-192) TO
  shipoutput WITH IMAGE xxxxxxxxx^x^xxxxxxxxxx^xx.xx
    totbox<totbox + ptnrbox OF mmpt
    totptwtime<totptwtime + pttottime OF mmpt
    totpt2time<totpt2time + pttot2time OF mmpt
END
WRITE "Av. portainer production:";(totbox/totptwtime);"box/hour| real productio
n:";(totbox/totpt2time);"box/hour"TO shipoutput WITH IMAGE xxxxxxxxxxxxxxxxxxxx
xxxxx^xx.x^xxxxxxxxxxxxxxxxxxxxxxxxxxx^xx.x^xxxxxxxx
WRITE "Av. portainer occupancy:";totpt2time/(4*(NOW-192)) TO shipoutput WITH IM
AGE xxxxxxxxxxxxxxxxxxxxxxxxx^xx.xx
FOR zz<1 TO 3
    WRITE"berth";zz;"occupancy";shoccberth[zz]/(NOW-192) TO shipoutput WITH
    IMAGE xxxxx^x^xxxxxxxxx^xx.xxx
    totberthtime<totberthtime + shoccberth[zz]
END
WRITE "Average quay occupancy:";totberthtime/(3*(NOW-192)) TO shipoutput WITH I
MAGE xxxxxxxxxxxxxxxxxxxxxxx^xx.xxx

RETURN
```

2.30 Macro INPUTBLOCK

This macro is activated by the MAIN module in case the user decides not to specify the required input data manually. This data has been put in another data file and is referred to by the name *situation*. This has been done before the computer model is activated.

[5-17]:
The module requires values for the following attributes:

- date;
- netoperation;
- priority;
- runtime;
- nrpt;
- nrtt;
- ttperpt;
- nrrtgc;
- mainlanespeed;
- bridgespeed;
- quayspeed;
- rtgcspeed.

After these attributes are given their values, the macro returns to the MAIN module.

situation Inputstream: file containing input data

2.31 Macro OUTPUT

This macro is activated by the module MAIN after the simulation run has ended. This macro is responsible for generating and writing output results to an output data file referred to by the name *shipoutput*. This output data can then be reviewed.
The following data is written to this output file:

- the runnr and date;
- number of tt's per portainer and the total number of tt's present
- value of the influence factor;
- average inter-arrival time of the ships arriving at the quay;
- average turn-around time of the ships;
- leaving time of last ship;
- average waiting time of the ships in the anchorage;
- total portainer production and net portainer production;
- portainer occupancy;
- total average portainer production and total net portainer production;
- total average portainer occupancy;
- berth occupancy;
- total average quay occupancy.

After completing its task, the macro returns to the MAIN module.

shipoutput Outputstream: file containing output data

MODEL PLAN_ICT
MOD DEFINE

Date: 98/09/09
Time: 12:47:22

```
CLASS                 : ship
COMPONENT             : planningmaster
QUEUE                 : planlist1 planlist2 waitlist
RANDOMSTREAM          : shiatdistr unif unifs
INPUTSTREAM           : seedlist selectlength selectbox
OUTPUTSTREAM          : firstplan secondplan shipoutput
TIMEUNIT              : hours

ATTRIBUTES OF MAIN    :
INTEGER               : a b shipsatquay runnr selectshl[20] selectshb[20]
REAL                  : quayspace shinterarrivaltime totiat totiat2 totbox
CHARACTER(10)         : date simdate

ATTRIBUTES OF planningmaster:
REAL                  : shiparrivaltime iat lastshiptime
REFERENCE TO ship     : shsecondlst lastship

ATTRIBUTES OF ship    :
INTEGER               : shnr shtotmoves shlength
REAL                  : interarrivaltime shiat shworktime
REFERENCE TO SET      : waitinglist
```

3. Description model Plan_ICT

3.1 Module DEFINE

MODEL PLAN ICT
MOD MAINMOD

Date: 98/09/09
Time: 12:47:22

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ initial values and run parameters input @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

begin:
quayspace<510

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ seeds for random number generation @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

SEED OF unif<READ FROM seedlist
SEED OF unifs<READ FROM seedlist
SEED OF shiatdistr<READ FROM seedlist

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ reshaping of distributions @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

RESHAPE unif AS SAMPLED FROM DISTRIBUTION UNIFORM
RESHAPE unifs AS SAMPLED FROM DISTRIBUTION UNIFORM
RESHAPE shiatdistr AS SAMPLED FROM DISTRIBUTION EXPONENTIAL WITH PARAMETERS MEAN(7.9)

FOR a<1 TO 20
 selectshl[a]<READ FROM selectlength
END
FOR b<1 TO 20
 selectshb[b]<READ FROM selectbox
END

@@@@@@@@@@@@@@@@@@@@@@
@ start shipplanning @
@@@@@@@@@@@@@@@@@@@@@@

CALL firstplanning
start:
ACTIVATE planningmaster FROM begin IN planningprocess
WAIT

@@@@@@@@@@@@@@@@@@
@ end simulation @
@@@@@@@@@@@@@@@@@@

CALL output
CANCEL ALL
TERMINATE
```

3.2 Module MAIN

[1-22]:
In this section of the module the *randomstreams* are determined. The seeds for the randomstreams are read from an input file called *seedlist*.

unif	Randomstream: random number from a uniform distribution
unifs	Randomstream: random number from a uniform distribution
shiatdistr	Randomstream: inter-arrival time of a vessel, retrieved from an exponential distribution
seedlist	Inputstream

[24-29]:
In this section different values for the *selectsh[a]* and *selectb[b]* are read from input files called *selectlength* respectively *selectbox*. The values of the selectsh[a] represent ship lengths and the values of selectb[b] represent the number of box moves to be made per ship. These values are linked to the macro FIRSTPLANNING in which the ships are generated.

selectsh[a]	Integer attribute of MAIN: for different values of *a* a ship length is linked to *selectsh[a]*
selectb[b]	Integer attribute of MAIN: for different values of *b* a number of box moves to be made per vessel is linked to *selectb[b]*
selectlength	Inputstream
selectbox	Inputstream

[35-46]:
In this section the actual planning occurs. First the macro FIRSTPLANNING is activated which is responsible for creating a data list of vessels that will call at the terminal. Once this macro has completed its actions the module PLANNINGPROCESS is activated and the MAIN module becomes passive. When reactivated the macro OUTPUT is activated and finally the simulation run is terminated.

FIRSTPLANNING	Macro: creates a data list containing ships that will call at the terminal (including inter- arrival time, vessel length and number of box moves to be made)
planningprocess	Module: fits the ships (in the data list created by the macro Firstplanning) in a time schedule and subsequently creates a final data list
output	Macro: contains simulation run output

MODEL PLAN ICT
MAC FIRSTPLANNING

Date: 98/09/09
Time: 12:47:23

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ making the first planning charts @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

LOCAL:
INTEGER: a

FOR a<1 TO 300
   THIS ship<NEW ship
   shnr OF THIS ship<a

   interarrivaltime:
   interarrivaltime OF THIS ship<shiatdistr
   interarrivaltime OF THIS ship<45 IF interarrivaltime OF THIS ship>45 HOURS

   shiplength:
   shlength OF THIS ship<selectshl[CEIL(20*unif)]

   containers:
   shtotmoves OF THIS ship<selectshb[CEIL(20*unifs)]

   JOIN THIS ship TO planlist1
   WRITE interarrivaltime;shlength;shtotmoves TO firstplan WITH IMAGE xxx.xx^xx
xxx^xxxxx
   totbox<totbox + shtotmoves OF THIS ship
   totiat<totiat + interarrivaltime OF THIS ship
END
RETURN
```

3.3 Macro FIRSTPLANNING

[5-27]:
This macro is activated by the MAIN module and is responsible for the following actions:

- Creating a number of ships;
- Defining an inter-arrival time per created ship;
- Defining the accompanying vessel length;
- Defining the accompanying number of box moves to be made by that ship.

Each created ship is subsequently joined to the queue *planlist1*. When all the ships are created and joined to the data list the macro returns to the MAIN module.

shnr	Integer attribute of ship: number of the ship
interarrivaltime	Real attribute of ship: inter-arrival time of ship
shlength	Integer attribute of ship: ship length
shtotmoves	Integer attribute of ship: number of box moves to be made
planlist1	Queue: contains all the ships expected to call at the terminal
firstplan	Outputstream
totbox	Real attribute of MAIN: total number of box moves to be made
totiat	Real attribute of MAIN: total of inter-arrival times

MODEL PLAN ICT Date: 98/09/09
MOD PLANNINGPROCESS Time: 12:47:22

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ rescheduling of the ships by the planningmaster @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

begin:
IF LENGTH OF planlist1≠0
   shsecondlst<FIRST ship IN planlist1
   shiparrivaltime<NOW
   IF shipsatquay=3
      WAIT UNTIL shipsatquay≠3
   END
   IF quayspace < (shlength OF shsecondlst)
      WAIT UNTIL quayspace ≥ shlength OF shsecondlst
   END
   REMOVE shsecondlst FROM planlist1
   JOIN shsecondlst TO planlist2
   lastship<PRED OF shsecondlst IN planlist2 IF shnr OF shsecondlst≠1
   shiat OF lastship<ABS(NOW-lastshiptime) IF shnr OF shsecondlst≠1
   quayspace<quayspace-(shlength OF shsecondlst)
   shipsatquay<shipsatquay+1
   lastshiptime<NOW
   ACTIVATE shsecondlst FROM start IN shipserviceplan
   iat<interarrivaltime OF shsecondlst-(NOW-shiparrivaltime)
   WAIT iat IF iat>0
   WRITE shiat OF lastship;shlength OF lastship;shtotmoves OF lastship TO secon
  dplan WITH IMAGE xxx.xx÷xxxxx÷xxxxx  IF shnr OF shsecondlst≠1
   totiat2<totiat2 + shiat OF lastship IF shnr OF shsecondlst≠1
   REPEAT FROM begin
END
IF LENGTH OF planlist1=0
   TERMINATE
END
```

3.4 Module PLANNINGPROCESS

[5-14]:
This module is activated by the MAIN module and is responsible for fitting the ships (generated in the Macro FIRSTPLANNING) as best as possible in a time schedule (by the planningmaster).
This is done by simulating the expected situation at the quay. The next ship to berth can only do so, when less than three ships are berthed and there is enough free quay space for the ship. When the ship can not berth she will wait till it is possible.

shsecondlst	Reference to ship: ship currently being planned
shiparrivaltime	Real attribute of planningmaster: time planningmaster started fitting current ship
shipsatquay	Integer attribute of MAIN: number of ships at the quay
quayspace	Real attribute of MAIN: length of quay that is unoccupied

[15-31]:
Once the ship is allowed to berth, she is joined to the queue *planlist2* and various attributes are updated. Among the attributes updated is the inter-arrival time of the ship. Subsequently this ship is activated in the module SHIPSERVICEPLAN.
After waiting a specific inter-arrival time (depending on the inter-arrival time of the ship) the situation at the quay is checked for the next ship in the *planlist1*. If there are no other ships in the *planlist1* the module is terminated.

planlist2	Queue: contains the ships after they have been fitted in a time schedule
lastship	Reference to ship: predecessor of ship currently being fitted in the time schedule
shiat	Real attribute of ship: inter-arrival time of *lastship*
lastshiptime	Real attribute of planningmaster: arrival time of ship at quay
iat	Real attribute of planningmaster: original inter-arrival time of ship (determined in macro FIRSTPLAN)
secondplan	Outputstream

MODEL PLAN_ICT
MOD SHIPSERVICEPLAN

Date: 98/09/09
Time: 12:47:22

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ expected time needed for ship to load and unload @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

start:
shworktime<(shtotmoves)/((1.05)*20.5) HOURS
WORK shworktime
shipsatquay<shipsatquay-1
quayspace<quayspace+shlength
IF shnr=300
   REACTIVATE MAIN IF MAIN IS NOT ACTIVE
END
TERMINATE
```

MODEL PLAN_ICT
MAC OUTPUT

Date: 98/09/09
Time: 12:47:24

```

WRITE "average iat planning1:";totiat/300 TO shipoutput
  WITH IMAGE xxxxxxxxxxxxxxxxxxxxxx^xx.xx
WRITE "average iat planning2:";totiat2/299 TO shipoutput
  WITH IMAGE xxxxxxxxxxxxxxxxxxxxxx^xx.xx
WRITE "total boxmoves:";totbox;"average box/ship:";totbox/300 TO shipoutput
  WITH  IMAGE xxxxxxxxxxxxxxx^xxxxxxx^xxxxxxxxxxxxxxxxx^xxx
RETURN
```

3.5 Module SHIPSERVICEPLAN

[5-13]:
This module is activated by the module PLANNINGPROCESS and in this module the ship is served. The time needed to serve a ship depends on the number of box moves that are to be made. Once the ship is served she leaves the quay and some attributes are updated before the module is terminated. When this was the last ship in the simulation run the MAIN module is reactivated.

shworktime	Real attribute of ship: approximate time needed to service ship

3.6 Macro OUTPUT

[2-5]:
This macro is activated by the MAIN module just before the simulation run is terminated. The following simulation run output is stored in an output file (called *shipoutput*):

- Average inter-arrival time of the ships in the first data list (*planlist1*);
- Average inter-arrival time of the ships in the final data list (*planlist2*);
- Total number of box moves made;
- Average number of box moves per ship.

Once this has been done the macro returns to the MAIN module.

shipoutput	Outputstream

MODEL ICT1 A
MAC SHIPATTRIBUTES

Date: 98/09/09
Time: 12:30:56

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ assigning ship attributes for ship @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

shipnumber:
shnr<nrofship

shipexportcontainerbay:
w<w+1
shexportbay<w
w<0 IF w=5

shipimportcontainerbay:
x<x+1
shimportbay<x
x<5 IF x=9

shiat<READ FROM planlist
shlength<READ FROM planlist
shtotmoves<READ FROM planlist

exportcontainers:
shexp20f< FLOOR(0.285*shtotmoves)
shexp20mt< FLOOR(0.011*shtotmoves)
shexp40f< FLOOR(0.216*shtotmoves)
shexp40mt< FLOOR(0.016*shtotmoves)

importcontainers:
shimp20f< FLOOR(0.203*shtotmoves)
shimp40f< FLOOR(0.133*shtotmoves)
shimp20mt< FLOOR(0.062*shtotmoves)
shimp40mt< FLOOR(0.075*shtotmoves)

totaal:
shtotmoves<shexp20f+shexp40f+shexp20mt+shexp40mt+shimp20f+shimp40f+shimp20mt+shimp40mt
shimp20<shimp20f+shimp20mt
shimp40<shimp40f+shimp40mt
shexp20<shexp20f+shexp20mt
shexp40<shexp40f+shexp40mt
shimportteu<shimp20+(2*shimp40)
shexportteu<shexp20+(2*shexp40)
shimportteuf<shimp20f+(2*shimp40f)
shexportteuf<shexp20f+(2*shexp40f)
shimportteumt<shimp20mt+(2*shimp40mt)
shexportteumt<shexp20mt+(2*shexp40mt)
shexchange<(shimportteu+shexportteu)
shreqpt<1 IF (shtotmoves≤30)
shreqpt<2 IF (shtotmoves>30)
shturnaroundtime<0
shservicetime<0

stack:
stackimp20<0
stackimp20f<0
stackimp20mt<0
stackimp40<0
stackimp40f<0
stackimp40mt<0
stackexp20<0
stackexp20f<0
stackexp20mt<0
stackexp40<0
stackexp40f<0
stackexp40mt<0

RETURN
```

4. Description future ICT model

4.1 Introduction

Since the future International Container Terminals (ICT1 & ICT2) only differ slightly from the ICT model described in chapter one, this chapter will only described the modules and macros that have been modified.

For the computer model of ICT1 (future ICT model) only the macro SHIPATTRIBUTES will be described. Of the ICT2 model (new ICT) the TTPROCESS module will be described (the macro shipattributes is identical to the macro discussed for ICT1).

4.2 Description model ICT1_A

Macro SHIPATTRIBUTES:

[5-20]:
In this section the ship is assigned a particular stack for both the import and export containers.
The following three attributes of a ship are given a value that is read from an input file (called *planlist*) that was created in the planning model (discussed in chapter 3):

- Ship inter-arrival time (shiat);
- Ship length (shlength);
- Total number of box moves to be made (shtotmoves).

planlist Inputstream

[22-66]:
In this section the same attributes are given a value as in the macro described in chapter 2.4. For explanation the reader is therefore referred to this chapter.
Once the macro has performed all its actions it will return to the SHIPGENERATOR module.

```
@@@@@@@@@@@@@@
@ tt process @
@@@@@@@@@@@@@@

startup:

leaveparking:
ENTER lane[10]
WORK mainlanetime
LEAVE lane[10]
ENTER outtraffic
WORK (CEIL((600)/mainlanespeed)) SECONDS
LEAVE outtraffic
ENTER maintraffic
WORK mainlanetime
LEAVE maintraffic
ENTER yardqueue
ACTIVATE trmasteryard FROM begin IN trmyardprocess IF trmasteryard IS NOT ACTIVE
WAIT
GOTO parking IF trdestination="parking"
GOTO yardaction IF (trdestination="exportstack")|(trdestination="expemptystack")

drivetoquay:
ENTER tlyqueue
ACTIVATE tlyard FROM check IN tlyprocess IF tlyard IS NOT ACTIVE
WAIT
LEAVE tlyqueue
WRITE NOW TO yttonbridge WITH IMAGE xxxx.xxxxx
ENTER yardtoquay
WORK bridgetime
LEAVE yardtoquay
restart:
ENTER quayqueue
ACTIVATE trmasterquay FROM begin IN trmquayprocess IF trmasterquay IS NOT ACTIVE
WAIT
IF ttwait=TRUE
   WAIT WHILE (LENGTH OF shpt OF trworkship=0)
   WAIT WHILE (ptshiftprocedure OF trworkpt=TRUE)
   ttwait<FALSE
   REPEAT FROM restart
END
GOTO drivetoyard IF (trdestination="yard")|(trdestination="parking")

quaytoportainer:
ENTER quayintraffic
WORK trquaytime
LEAVE quayintraffic
ENTER ttunderpt OF trworkpt
ttupttime<NOW
MOVE ttuptfig[ptnr OF trworkpt] TO LENGTH OF ttunderpt OF trworkpt
WAIT

leaveportainer:
ENTER quayouttraffic
WORK trquaytime
LEAVE quayouttraffic
GOTO restart IF trdestination="exportquay"
ENTER quayqueue
ACTIVATE trmasterquay FROM begin IN trmquayprocess IF trmasterquay IS NOT ACTIVE
WAIT
GOTO quaytoportainer IF trdestination="importquay"

drivetoyard:
ENTER tlqqueue
ACTIVATE tlquay FROM check IN tlqprocess IF tlquay IS NOT ACTIVE
WAIT
LEAVE tlqqueue
WRITE NOW TO qttonbridge WITH IMAGE xxxx.xxxxx
ENTER quaytoyard
WORK bridgetime
LEAVE quaytoyard
ENTER yardqueue
ACTIVATE trmasteryard FROM begin IN trmyardprocess IF trmasteryard IS NOT ACTIVE
WAIT
GOTO parking IF trdestination="parking"
GOTO drivetoquay IF trdestination="importquay"
```

4.2 Description Model ICT2_A

Module TTPROCESS:

This module concerns the actions of a tractor-trailer (tt). The tt actions are the same as the tt actions described in chapter 2.12. This chapter will only discuss the differences. For a description of the whole module the reader is referred to the before mentioned chapter.
The differences occur in the sections discussed below.

[5-21]:
This section concerns the tt actions when a tt leaves the parking lot and drives to the traffic master of the yard (which is assumed to be located in the same location as for the other ICT). The distance a tt has to drive south of the stacks is 300 meters longer than for a tt driving for the other ICT.

[44-61]:
This section describes the tt actions on the quay. A tt driving for ICT2 will have to drive 500 meters more on the quay to reach the portainers (this drive time is called *trquaytime*). A tt will also have to drive 500 meters more when driving from the portainers to the bridge entrance when compared to the tt's driving for ICT1.

[78-106]:
This section concerns the tt actions in the container yard. To reach the container yard a tt will first have to drive 300 meters past the container yard of the first ICT.
This is also the case when a tt leaves the stack and drives in direction of the traffic master of the yard or the bridge entrance.

[108-123]:
This section concerns the tt actions when the tt is on its way to return to the tt parking lot. A tt will have to drive 300 meters extra (past the container yard of ICT1) to reach the parking lot when compared to the tt's driving for the other ICT.

```
yardaction:
trrow<10 IF trdestination="impemptystack"
ENTER intraffic
WORK (CEIL((300+trrow*30)/mainlanespeed)) SECONDS
LEAVE intraffic
ENTER lane[trrow]
WORK (CEIL(210/mainlanespeed)) SECONDS
ENTER ttq[trrow]
WAIT

leaveyard:
WORK (CEIL(210/mainlanespeed)) SECONDS
LEAVE lane[trrow]
ENTER outtraffic
WORK (CEIL((300+trrow*30)/mainlanespeed)) SECONDS
LEAVE outtraffic
ENTER maintraffic
WORK mainlanetime
LEAVE maintraffic
GOTO drivetoquay IF (trbox>0)&(trdestination="exportquay")
GOTO yardaction IF trdestination="impemptystack"
ENTER yardqueue
ACTIVATE trmasteryard FROM begin IN trmyardprocess IF trmasteryard IS NOT ACTIVE
WAIT

GOTO drivetoquay IF trdestination="importquay"
GOTO yardaction IF trdestination="exportstack"
GOTO yardaction IF trdestination="expemptystack"
GOTO parking IF trdestination="parking"

parking:
ENTER intraffic
WORK (CEIL((600)/mainlanespeed)) SECONDS
LEAVE intraffic
trworkpt<ptfirst
LEAVE ttofpt OF trworkpt
ENTER ttavailable
MOVE ttparking TO LENGTH OF ttavailable   @anim@
trworkpt<NONE
trworkship<NONE
ptfirst<NONE
trendprocedure<FALSE
trdestination<"importquay"
trrow<0

PASSIVATE
```

ttexportpt	Set attribute of portainer: contains all the tt's on the quay waiting to be serviced by the portainer
ttimportpt	Set attribute of portainer: contains all the tt's on the quay waiting to be serviced by the portainer
ttnr	Double attribute of tt: number of the tt
ttofpt	Set attribute of portainer: set to which the tt's are joined that are activated on behalf of this portainer
ttperpt	Integer attribute of MAIN: number of tractor-trailers (tt's) to be assigned per portainer
ttq[ ]	Queue: contains all the tt's waiting to be served by a RTGC
ttreqlst	Queue: contains the portainers requesting for tractor-trailers
ttspeed	Real attribute of MAIN: maximum driving speed of the tractor-trailers on the bridge in km/hour
tttype	Character attribute of tt: the ttgenerator that generated the tractor-trailer
ttunderpt	Set attribute of portainer: contains the tractor-trailers waiting to be served by the portainer
ttwait	Logical attribute of transporter: defines if a tt has to wait on the quay before continuing its drive to a portainer
unif	Randomstream: random number from uniform distribution
unifs	Randomstream: random number from a uniform distribution
unloadingship	Queue: contains the ships that are being unloaded
unloadingtime	Real attribute of class component ship: time needed for unloading the ship
unloadtt	Set attribute of portainer: contains the tt presently being unloaded by the portainer
waitinglist	Set attribute of ship: set to which tractor-trailers, that are waiting at the quay for a portainer to reach its position are joined
xrtgc	Reference to rtgc: second RTGC to be active in stack
xrtgcrow10	Reference to rtgc: second RTGC to be activated in row 10 (empties stack)
xrtgcset	Queue: contains the available second RTGC that can be assigned to a stack
xrtgcstop	Logical attribute of rtgc: defines if second RTGC should stop its actions in the stack
xrtgcwexp	Set attribute of class component SHIP: if a second RTGC is active in the export stack of this ship, it is joined to this stack
xrtgcwimp	Set attribute of class component SHIP: if a second RTGC is active in the import stack of this ship, it is joined to this stack
xrtgcworkset	Set attribute of MAIN: set to which available extra RTGC's are joined
yard	Queue: contains trucks present in the yard
YARDCONTAINER	Macro: defines the load to be picked-up in the yard (size and load)
YARDLOAD	Macro: defines the updating of the containers in the stack and determines which load is to be lifted out off the stack
yardq	Queue: contains the tt's waiting for admission onto the bridge from the yard-side
yardqueue	Queue: contains all the tt's waiting for instructions from the traffic master of the yard
yardtoiit	Queue: contains all the quay bound tt's driving on the bridge between the yard and the IIT quay
yardtoquay	Set attribute of MAIN: set to which the quay bound tt's driving on the bridge are joined to
yardtt	Reference to tt: selected tt from the queue *yardq*
ytt_ict1	Randomstream: inter-arrival time of an ICT1 quay bound tt generated at the yard-side
ytt_ict2	Randomstream: inter-arrival time of an ICT2 quay bound tt generated at the yard-side
ytt_iit	Randomstream: inter-arrival time of an IIT quay bound tt generated at the yard-side

shipattributes	Macro: determines characteristics of a ship
shiponhand	Reference to class component ship: ship portainer is assigned to
shipoutput	Outputstream: file containing output data
shipsatquay	Integer attribute of MAIN: number of ships at the quay
shiptotanchtime	Real attribute of MAIN: total time the ships spend in the anchorage
shiptotstime	Real attribute of MAIN: total of all ship service times
shlength	Integer attribute of ship: ship length
shnr	Integer attribute of ship (Plan model): number of the ship
shnr	Integer attribute of ship: number of this ship which is the same number as in the terminal records of 1997
shpt	Set attribute of class component SHIP: set to which portainers servicing the ship are joined
shreqpt	Integer attribute of class component ship: number of portainers to be requested by this ship
shsailtime	Real attribute of MAIN: time it takes for the ship to reach the quay once it has left the anchorage
shsecondlst	Reference to ship: ship currently being planned
shservicetime	Real attribute of class component ship: inter-arrival time of this ship
shtotmoves	Double attribute of class component ship: total number of box moves to be made
shtotmoves	Integer attribute of ship (Plan model): number of box moves to be made
shturnaroundtime	Real attribute of class component ship: turn-around time of this ship
shworktime	Real attribute of ship: approximate time needed to service ship
situation	Inputstream: file containing input data
specification	Character attribute of MAIN: defines whether the input data will be specified by the user or is to be attained from an input file
spnr	Integer attribute of ptmaster: parameter representing the berth place
stackexp20	Integer attribute of class component ship: total number of 20 ft export containers in the assigned export stack
stackexp20f	Integer attribute of class component ship: total number of 20 ft full export containers in the assigned export stack
stackexp20mt	Integer attribute of class component ship: total number of 20 ft empty export containers in the assigned export stack
stackexp40	Integer attribute of class component ship: total number of 40 ft export containers in the assigned export stack
stackexp40f	Integer attribute of class component ship: total number of 40 ft full export containers in the assigned export stack
stackexp40mt	Integer attribute of class component ship: total number of 40 ft empty export containers in the assigned export stack
stackimp20	Integer attribute of class component ship: total number of 20 ft import containers in the assigned import stack
stackimp20f	Integer attribute of class component ship: total number of 20 ft full import containers in the assigned import stack
stackimp20mt	Integer attribute of class component ship: total number of 20 ft empty import containers in the assigned import stack
stackimp40	Integer attribute of class component ship: total number of 40 ft import containers in the assigned import stack
stackimp40f	Integer attribute of class component ship: total number of 40 ft full import containers in the assigned import stack
stackimp40mt	Integer attribute of class component ship: total number of 40 ft empty import containers in the assigned import stack
TEUPERTRUCK	Macro: defines the load to be picked-up by the truck
tgmq[ ]	Queue: contains trucks waiting for permission to enter the terminal yard
tgmrow	Integer attribute of truckgatemaster: row number of the truck just leaving the yard

pttot2time	Real attribute of portainer: total time a portainer is assigned to ship
pttottime	Real attribute of portainer: total time of net portainer actions
ptunloading	Logical attribute of portainer: defines whether or not a portainer is in the process of unloading a ship
qtt_ict1	Randomstream: inter-arrival time of an ICT1 yard bound tt generated at the quay-side
qtt_ict2	Randomstream: inter-arrival time of an ICT2 yard bound tt generated at the quay-side
qtt_iit	Randomstream: inter-arrival time of an IIT yard bound tt generated at the quay-side
quay	Queue: contains all the ships moored
quayintraffic	Set attribute of MAIN: set to which all the portainer bound tt's driving on the quay are joined to
QUAYLOAD	Macro: determines the type of container to be unloaded (size and load)
quayouttraffic	Set attribute of MAIN: set to which the tt's are joined to when driving from the portainer to the bridge
quayqueue	Queue: contains the tt's waiting for instructions from the traffic master of the quay
quayspace	Real attribute of MAIN: length of quay that is unoccupied
quayspeed	Real attribute of MAIN: driving speed of the tt's on the quay
quaytoyard	Set attribute of MAIN: set to which all the yard bound tt's driving on the bridge are joined to
quaytt	Reference to tt: selected tt from the queue *ictquayq*
reservation	Logical attribute of MAIN: defines if a berthing place is to be occupied
rtgcdrivetime	Randomstream: time needed to drive to the correct position (report chapter 5.3.3)
rtgcliftofftime	Randomstream: time needed to lift a container off of a transporter (report chapter 5.3.3)
rtgcliftontime	Randomstream: time needed to lift a container onto the transporter (report chapter 5.3.3)
rtgcnr	Integer attribute of RTGC: number of RTGC
rtgcrepostime	Randomstream: time needed for repositioning RTGC (report chapter 5.3.3)
rtgcreshuftime	Randomstream: time needed to reshuffle the stack (report chapter 5.3.3)
rtgcserveunit	Reference to transporter: transporter being serviced by the RTGC
rtgcset	Queue: contains the available RTGC's
rtgcship	Reference to ship: ship the second RTGC will be servicing
rtgcspeed	Real attribute of MAIN: driving speed of the RTGC's
rtgcstackvol	Integer attribute of MAIN: number of containers in the stack (in TEU)
rtgcstrategy	Macro: defines the servicing strategy of the RTGC's
rtgctype	Character attribute of RTGC: defines if the RTGC is the second RTGC or the first RTGC operating in the stack
runtime	Real attribute of MAIN: defines the length of the simulation run in hours
sea	Set attribute of class component ship: set which every ship is joined to before arriving at the anchorage and after she has left the terminal
seaboxin	Real attribute of MAIN: total number of boxes arrived at the terminal from sea
seaboxout	Real attribute of MAIN: total number of boxes leaving the terminal by sea
secondplan	Outputstream
seedlist	Inputstream
selectb[b]	Integer attribute of MAIN: for different values of *b* a number of box moves to be made per vessel is linked to *selectb[b]*
selectbox	Inputstream
selectlength	Inputstream
selectsh[a]	Integer attribute of MAIN: for different values of *a* a ship length is linked to *selectsh[a]*
shberthnr	Integer attribute of ship: number of berth assigned to ship

iattime	Real attribute of MAIN: inter-arrival time between this ship arriving at the quay and the last ship
iattimetot	Real attribute of MAIN: total of inter-arrival times of the ships
ictquayq	Queue: contains all the tt's waiting on the quay for permission to drive onto the bridge
icttoiit	Queue: contains all the tt's driving between the ICT quay and the IIT quay entrance in direction of the yard
igdayimp	Real attribute of importgen: total number of trucks to be generated divided by 100
igimp20	Integer attribute of importgen: number of 20 ft containers
igimp20f	Integer attribute of importgen: number of full 20 ft containers
igimp20mt	Integer attribute of importgen: number of empty 20 ft containers
igimp40	Integer attribute of importgen: number of 40 ft containers
igimp40f	Integer attribute of importgen: number of full 40 ft containers
igimp40mt	Integer attribute of importgen: number of empty 40 ft containers
igimprow	Integer attribute of importgen: import stack number of the ship whose containers are to be picked-up
igimptransp	Reference to transporter: truck being given attributes to
igjob	Reference to ship: ship this import generator generates trucks for
ignrtrans	Integer attribute of importgen: total number of trucks needed to pick-up the required total number of containers
igtrucknr	Integer attribute of importgen: number of trucks to be generated on a day
igtrucks	Integer attribute of importgen: number of trucks already generated
iittoict	Queue: contains all the ICT quay bound tt's driving between the IIT quay entrance and the ICT quay
iittoyard	Queue: contains all the yard bound tt's driving between the IIT quay entrance and the yard
IMPGEN_GEN	Macro: determines the import container specifics
importpt	Set attribute of MAIN: contains all portainers unloading
inputblock	Macro: defines the values of various variables that the computer model requires
insertq	Queue: contains all the tt's waiting to drive onto the bridge from the IIT quay
inserttt	Reference to tt: selected tt from the queue *insertq*
insertzone	Queue: contains all the tt's driving in the insertzone section
insertzonelength	Real attribute of MAIN: length of the insertzone
interarrivaltime	Real attribute of ship: inter-arrival time of ship
intraffic	Set attribute of MAIN: set to which the tt's and trucks driving north of the stacks are joined to
landboxin	Real attribute of MAIN: number of boxes that have entered the yard coming from the landside
landboxout	Real attribute of MAIN: number of boxes that have left the yard by truck
lane[ ]	Queue: contains the tractor-trailers and trucks present in that particular lane
lastship	Reference to ship: predecessor of ship currently being fitted in the time schedule
lastshiptime	Real attribute of planningmaster: arrival time of ship at quay
leavingtrucks	Set attribute of truckgatemaster: set to which trucks are joined to when leaving the yard
loadingship	Queue: contains the ships that are being loaded
loadingtime	Real attribute of class component SHIP: time needed for loading the ship
loadtt	Set attribute of portainer: contains the tt presently being loaded by the portainer
mainlanespeed	Real attribute of MAIN: driving speed of the tt's and trucks in the container yard
mainlanetime	Real attribute of MAIN: time needed to drive the length of a stack
maintraffic	Set attribute of MAIN: set to which tt's and trucks are joined when driving north along the first stack
netoperation	Real attribute of MAIN: value of the to be used *influence factor* in the runs (see also the report chapter 5.5.3 for specifics of this variable)
nextship	Reference to class component ship: the ship that is to be served next

MODEL SUBRIDG1
MOD YARDLIGHTPROCESS

Date: 98/09/09
Time: 10:36:32

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ DETERMINING IF THE TT'S CAN DRIVE ONTO THE BRIDGE FROM THE YARDSIDE @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

start:
WAIT UNTIL yardq IS NOT EMPTY
yardtt<FIRST tt IN yardq
REACTIVATE yardtt
WAIT followuptime SECONDS
REPEAT FROM start
```

MODEL SUBRIDG1
MOD ICTQUAYLIGHTPROCESS

Date: 98/09/09
Time: 10:36:34

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ DETERMINING IF THE TT'S CAN DRIVE ONTO THE BRIDGE FROM THE ICT QUAY @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

start:
WAIT UNTIL ictquayq IS NOT EMPTY
quaytt<FIRST tt IN ictquayq
REACTIVATE quaytt
WAIT followuptime SECONDS
REPEAT FROM start
```