COMPUTER SIMULATIONMODEL
CONTAINER TERMINAL
" TANJUNG PERAK "
SURABAYA, INDONESIA

PART 3 : MODEL DESCRIPTION

BERTHS

model boundaries

container terminal

import

export

 TU Delft
Delft University of Technology
PART 3: COMPUTER SIMULATION MODEL DESCRIPTION

Graduate students: J.A. Helbing
A.C. de Kluizenaar

Head of Port Engineering Group, T.U. Delft:
Prof. ir. H. Velsink

Supervisors: ir. R. Groenveld
ir. T. Horstmeier

TECHNICAL UNIVERSITY DELFT
MARCH 1989
CONTENTS

Part 3 : Computer simulation model description

Chapter 17 : Introduction

Chapter 18 : Model SUREND 1 description

18.1 Module MAIN
18.2 Module SHIPGENERATOR
18.3 Macro SHIPATTRIBUTES
18.4 Macro STACKALLOCATION
18.5 Module SHIPPROCESS
18.6 Module MEASUREMENTS
18.7 Module BERTHMASTER
18.8 Module CRANEMASTER
18.9 Module TTMASTER
18.10 Module QUAYCRANEPROCESS
18.11 Macro QUAYLOAD
18.12 Module TTPROCESS
18.13 Macro TTCONTROL
18.14 Macro DIRECTIONS
18.15 Macro DRIVETIMES
18.16 Module YARDCRANEPROCESS
18.17 Macro TTPRIORITY
18.18 Macro NOPRIORITY
18.19 Macro YARDLOAD
18.20 Module IMPORTGENPROCESS
18.21 Module EXPORTGENPROCESS
18.22 Module TRUCKPROCESS
18.23 Macro INPUTBLOCK
18.24 Macro OUTPUTBLOCK

Chapter 19 : Model SUREND 2 description

19.1 Introduction
19.2 The module MAIN
19.3 The macro STACKALLOCATION
19.4 The module TTPROCESS
19.5 The macro TTCONTROL
19.6 The macro SHIFTCONTROL
19.7 The macro DIRECTIONS
19.8 The macro DRIVETIMES
19.9 The module YARDCRANEPROCESS

Chapter 20 : Model SUREND 3 description

20.1 Introduction
20.2 The module MAIN
20.3 The macro STACKALLOCATION
20.4 The module TTPROCESS
20.5 The macro SHIFTCONTROL
20.6 The macro DIRECTIONS
CHAPTER 17 : INTRODUCTION

This part contains detailed descriptions of the models and is intended for those who wish to analyse the models. Because the three models vary only in certain modules and macro's, we have chosen to describe only the SUREND 1 model completely. Of the other models only the modules and macro's which differ from their counterpart in model SUREND 1 have been described.

The setup of the description is as follows. First we will describe how a part of a module works (a verbal model), after which the part of the program itself is shown. As explanation of the terms used in that part, a definition list is included. The most user-defined words are only explained the first time they occur.

Where necessary, an illustration of how the module works (a flow diagram) is given.
CHAPTER 18 : MODEL DESCRIPTION SUREND1

18.1 Description of module MAIN

[3 - 13] :

In this part of the program the user is asked on his screen whether he wants to specify the run-input or not. This facility has been put in to facilitate the starting of a standard run. If the user wishes to specify or change the input he must answer y(es), in which case the program will skip to the special input section of the main module. If the user answers n(o), the program will call the standard inputblock and proceed with the standard input. If the user answers incorrectly (neither y or n) he will get a message explaining that he must answer correctly and the program will repeat itself from the start.

3 INPUT FROM SCREEN
4 BEGIN :
5 WRITE "DO YOU WANT TO SPECIFY THE INPUT ? (y\n" WITH IMAGE
6 SPECIFICATION ≠ CHREAD
7 IF "n" = SPECIFICATION
8 CALL INPUTBLOCK
9 GOTO INPUTEND
10 END
11 GOTO INPUTSTART IF "y" = SPECIFICATION
12 WRITE "YOU MUST ANSWER y OR n !!!" WITH IMAGE
13 REPEAT FROM BEGIN

SPECIFICATION : Character attribute of MAIN
Used to determine whether input is to be specified or not

INPUTBLOCK : Macro
This macro determines the input for a non-specified run

[15 - 58] :

This part of the program is the special input-section in which the user can determine the values of the run variables. The user is asked on his screen to fill in the values for the different variables. The user has the ability to change the non-physical variables, such as the transporter speeds, etc. The physical dimensions of the terminal can not be varied, neither can certain boundary condition values. At the end of the inputblock the user is asked if all the input is correct. If the user wishes to make corrections he must answer y(es) and the program will repeat itself from the beginning of the input-setion.
INPUTSTART:
WRITE "PLEASE FILL IN THE FOLLOWING INFORMATION" WITH IMAGE
DATE (DD-MM-YYYY) = " WITH IMAGE
LENGTH OF SIMULATION RUN (IN DAYS) = " WITH IMAGE
RUNTIME < 24 x READ
IF (RUNTIME = 0) v (RUNTIME > 8760) WRITE "LENGTH OF SIMULATION RUN IS 0 OR LONGER THAN 1 YEAR" WITH IMAGE
WRITE "PLEASE CORRECT THE RUNLENGTH" WITH IMAGE
REPEAT FROM COR2
COR3:
WRITE "DO TRACTOR-TRAILERS HAVE PRIORITY IN THE YARD? (y\n)n" WITH IMAGE
PRIORITY = CHREAD
GOTO MODVAR IF ("y" = PRIORITY)v("n" = PRIORITY)
WRITE "YOU MUST ANSWER y OR n !!!" WITH IMAGE
REPEAT FROM COR3
MODVAR:
WRITE "AND NOW THE DETERMINATION OF THE MODEL VARIABLES" WITH IMAGE
WRITE "THE NUMBER OF QUAYCRANES USED =" WITH IMAGE
NRCRANES = READ
WRITE "THE NUMBER OF TRACTOR-TRAILERS PER QUAYCRANE =" WITH IMAGE
TTPERCRANE = READ
NRTT5 = TTPERCRANE x NRCRANES
WRITE "THE NUMBER OF YARDCRANES USED =" WITH IMAGE
NRRTGC5 = READ
WRITE "THE MAXIMUM SPEED FOR TRANSPORTERS IN THE YARD = (KM\HR)" WITH IMAGE
YARDSPEED = (1000-3600) x READ
WRITE "THE MAXIMUM SPEED FOR TRANSPORTERS ON THE BRIDGE = (KM\HR)" WITH IMAGE
BRIDGESPEED = (1000-3600) x READ
WRITE "THE MAXIMUM SPEED FOR TRANSPORTERS ON THE QUAY = (KM\HR)" WITH IMAGE
QUAYSPEED = (1000-3600) x READ
WRITE "THE MAXIMUM RELOCATIONSPEED FOR THE YARDCRANE = (KM\HR)" WITH IMAGE
RTGCSPEED = (1000-3600) x READ
COR4:
WRITE "DO YOU WANT TO MAKE CORRECTIONS? (y\n)n" WITH IMAGE
CORRECTION = CHREAD
REPEAT FROM COR4 IF "y" = CORRECTION
REPEAT FROM COR1 IF "n" = CORRECTION
INPUTEND:
DATE : Character attribute of MAIN
RUNTIME : Real attribute of MAIN
The length of simulation in hours
PRIORITY : Character attribute of MAIN
Determines whether tractor-trailers receive priority or not in the yard
NRCRANES : Integer attribute of MAIN
The number of quaycranes used
TTPERCRANE : Integer attribute of MAIN
The number of tractor-trailers per quaycrane
NRTTS : Integer attribute of MAIN
The number of tractor-trailers used

NRRTGCS : Integer attribute of MAIN
The number of yardcranes used (Rubber Tyred Gantry Cranes)

YARDSPEED : Real attribute of MAIN
Assumed driving speed of tractor-trailers in the yard

BRIDGESPEED : Real attribute of MAIN
Assumed driving speed of tractor-trailers on the bridge

QUAYSPEED : Real attribute of MAIN
Assumed driving speed of tractor-trailers on the quay

RTGCSPEED : Real attribute of MAIN
Assumed driving speed of yardcranes along the stacks

CORRECTION : Character attribute of MAIN
Determines whether input must be redetermined or not

[60 - 63] :

In these lines the heading for the outputfile listing all the served ships is made.

'60 WRITE "SHIPNR LENGTH ARRIVAL TURN IMPORT EXPORT IMPORT EXPORT"
   TO SHIPOUTPUT WITH IMAGE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
This part of the program reads in the corresponding values of the shiplengths for the different \texttt{SELECT[i]} values. The values for \texttt{SELECT[i]} are read in from the input datafile \texttt{SPLDISTR}, to which different distributions can be linked. This information will be needed by the macro \texttt{SHIPATTRIBUTES}.

\begin{verbatim}
76 FOR A = 1 TO 20
77 SELECT(A) = READ FROM SPLDISTR
78 END
\end{verbatim}

\texttt{A} : Integer attribute of \texttt{MAIN}
SELECT[A] : Integer attribute of MAIN
For different values of A a ship-length is linked to SELECT[A]
SPLDISTR : Inputstream

[80 - 89] :

This section of the program creates the quaycranes and initializes some of their attributes. Sets belonging to the quaycranes are also created. After creation the cranes are joined to the queue PASSIVECRANES.

```
08 CREATION OF THE QUAYCRANES
01 FOR B = 1 TO NRCRANES
02 THIS QUAYCRANE < NEW QUAYCRANE
03 SHIPONHAND < NONE
04 UNDERCRANE < NEW SET
05 TTSET < NEW SET
06 UNLOADING < TRUE
07 TIMEIDLE < 0
08 JOIN THIS QUAYCRANE TO PASSIVECRANES
09 END
```

B : Integer attribute of MAIN
Parameter
NRCRANES : Integer attribute of MAIN
The number of quaycranes used
QUAYCRANE : Class component
This is a portainer
SHIPONHAND : Reference to ship of class component QUAYCRANE
This is the ship the quaycrane is working on
UNDERCRANE : Reference to set of class component QUAYCRANE
Set in which tractor-trailers waiting underneath a quaycrane are registered
TTSET : Reference to set of class component QUAYCRANE
Set in which all the tractor-trailer assigned to the quaycrane are registered
UNLOADING : Logical attribute of class component QUAYCRANE
Indicates process cycle of quaycrane
TIMEIDLE : Real attribute of class component QUAYCRANE
Time spent idle by quaycrane during a job
PASSIVECRANES : Queue
Contains all the quaycranes which are not assigned to a ship

[91 - 92] :
This section initializes the values for the logical OCCUPIED[i], which indicates which berth is occupied and which not. Both berths will start unoccupied, i.e. OCCUPIED = FALSE.

91 OCCUPIED[1] = FALSE

OCCUPIED[i] : Logical attribute of MAIN
Indicates whether berth[i] is free or not

[94 - 108] :

This section creates the transporters which remain permanently on the terminal. Again all the accompanying attributes are initialized. After creation the transporters are joined to the queue TTAVAILABLE.

94 CREATION OF TRANSPORTERS OF TYPE TT @
95 FOR C = 1 TO NRTTS
96 THIS TRANSPORTER < NEW TRANSPORTER
97 TYPE = "TT"
98 BOX < 0
99 TEU < 0
100 ROW < 0
101 BAY < 0
102 WORKSHOP < NONE
103 SHIFTPROCEDURE < FALSE
104 ENDPRECEDURE < FALSE
105 DESTINATION < "IMPORTSTACK"
106 TASK < "BRINGLOAD"
107 JOIN THIS TRANSPORTER TO TTAVAILABLE
108 END

C : Integer attribute of MAIN
Parameter

NRTTS : Integer attribute of MAIN
The number of tractor-trailers used

TRANSPORTER : Class component
This can be a truck or a tractor-trailer, depending on the type

TYPE : Character attribute of transporter
Indicates whether transporter is truck or tractor-trailer

BOX : Integer attribute of transporter
The number of containers on a transporter

TEU : Integer attribute of transporter
The number of TEU on a transporter

ROW : Integer attribute of transporter
Stack number which transporter must drive to

BAY : Integer attribute of transporter
Place along stack transporter must deliver or pick up container
WORKSHIP: Reference to ship of class component TRANSPORTER
This is the ship the transporter is driving for

SHIFTPROCEDURE: Logical attribute of class component TRANSPORTER
Indicates whether tractor-trailer must shift from import cycle to export cycle

ENDPROCEDURE: Logical attribute of class component TRANSPORTER
Indicates whether tractor-trailer must stop driving for ship or not

DESTINATION: Character attribute of class component TRANSPORTER
Determines which part of the yard the transporter must drive to

TASK: Character attribute of class component TRANSPORTER
Determines what the transporter is coming for

TTAVAILABLE: Queue
Contains the idle tractor-trailers

This part of the program creates the yardcranes. As in the previous similar sections, attributes are initialized and queues created. The new yardcranes are now not joined to a queue but activated in their process (YARDCRANEPROCESS). For each yardcrane a different randomstream CATROLLTIME is defined so that different performance levels could, if desired, be implemented.

CREATION OF YARDCRANES
FOR D = 1 TO NRRTGCS
YCRANE[D] = NEW YARDCRANE
CRANENR OF YCRANE[D] = D
YARDSTACK OF YCRANE[D] = 500
PREVLOCATION OF YCRANE[D] = 0
LOCATION OF YCRANE[D] = 0
IDLETIME OF YCRANE[D] = 0
PAST OF YCRANE[D] = 0
STRATEGY OF YCRANE[D] = TTPRIORITY IF ("y" = PRIORITY)
STRATEGY OF YCRANE[D] = NOPRIORITY IF ("n" = PRIORITY)
ACTIVATE YCRANE[D] FROM ROLLTOLOCATION IN YARDCRANEPROCESS
SEED OF CATROLLTIME[D] = READ FROM SEEDLIST
RESHAPE CATROLLTIME[D] AS SAMPLED FROM DISTRIBUTION UNKNOWN WITH PARAMETERS LB(0.5 MINUTE) MEAN(1.5 MINUTES) DEVIATION(0.5 MINUTE)
END

D: Integer attribute of MAIN Parameter
NRRTGCS: Integer attribute of MAIN
The number of yardcranes used (Rubber
YCRANE[D]: Reference to yardcrane of component MAIN
YARDCRANE: Class component
CRANENR: Integer attribute of YARDCRANE
   Enables identification of yardcranes
YARDSTACK: Integer attribute of YARDCRANE
   Indicates stack volume
PREVLOCATION: Integer attribute of YARDCRANE
   This is the original position of the yardcrane along the stack
LOCATION: Integer attribute of YARDCRANE
   This is the new position of the yardcrane along the stack
IDLETIME: Real attribute of YARDCRANE
   Time spent idle by the yardcrane
PAST: Real attribute of YARDCRANE
   Aid for measuring idletime
STRATEGY: Macro attribute of YARDCRANE
   Gives reference to which macro to use to determine the transporter to serve
TTPRIORITY: Macro
   Gives priority to the tractor-trailers by the yardcranes
NOPRIORITY: Macro
   First-come-first-serve ruling used to determine which transporter to serve
CATROLLTIME[D]: Randomstream
   Time needed to place or retrieve a container by yardcrane d

In this section attributes of MAIN are determined. The attributes are values for boundary variables which are linked to the physical dimensions of the terminal or of minor importance to the operation of the terminal.

\[
\begin{align*}
126 & : S < 1 \\
127 & : \text{TOTTEU} < 0 \\
128 & : \text{SAILTIME} < 1.5 \\
129 & : \text{DEBERTHTIME} < 0.5 \\
130 & : \text{SHIftime} < 0.5 \\
131 & : \text{QUAYSPEED} < 500 \\
132 & : \text{BRIDGETIME} < (1710 + \text{BRIDGESPEED}) \text{ SECONDS} \\
133 & : \text{YARDTIME} < (420 + \text{YARDSPEED}) \text{ SECONDS} \\
134 & : \text{TIME1} < (125 + \text{QUAYSPEED}) \text{ SECONDS} \\
135 & : \text{TIME2} < (375 + \text{QUAYSPEED}) \text{ SECONDS} \\
136 & : \text{UB12} < 0 \\
137 & : \text{UB34} < 0 \\
138 & : \text{UB5} < 0
\end{align*}
\]

S: Integer attribute of MAIN
   Parameter
TOTTEU: Real attribute of MAIN
   Total amount of TEU handled by harbor
SAILTIME: Real attribute of MAIN
   Time needed by ships to sail from the anchorage to the berths
DEBERTHTIME : Real attribute of MAIN
Time needed by a ship to leave berth and sail away

SHIFTTIME : Real attribute of MAIN
Time needed by quaycranes to shift to another ship before resuming operation

QUAYSPACE : Integer attribute of main
The amount of quay length available

BRIDGETIME : Real attribute of MAIN
The time needed by atractor-trailer to drive across the bridge.

YARDTIME : Real attribute of MAIN
The time needed by a transporter to drive the length of the main traffic lane

TIME1 : Real attribute of MAIN
Time needed by a tractor-trailer to reach berth[1] once on the quay

TIME2 : Real attribute of MAIN
Time needed by a tractor-trailer to reach berth[2] once on the quay

UB12 : Integer attribute of MAIN
Upper bound bay value of stack 1 and 2

UB34 : Integer attribute of MAIN
Upper bound bay value of stack 3 and 4

UB5 : Integer attribute of MAIN
Upper bound bay value of stack 5

[140 - 168] :

In these lines the randomstreams are determined. The seeds for the streams are read in from the inputfile SEEDLIST so that these may easily be changed by changing the environment.

140 RANDOMSTREAM PARAMETER DETERMINATION
141 SEED OF F401 < READ FROM SEEDLIST
142 RESHAPE F401 AS SAMPLED FROM DISTRIBUTION EXPONENTIAL WITH PARAMETERS MEAN(64)
143 SEED OF MT401 < READ FROM SEEDLIST
144 RESHAPE MT401 AS SAMPLED FROM DISTRIBUTION EXPONENTIAL WITH PARAMETERS MEAN(56)
145 SEED OF F201 < READ FROM SEEDLIST
146 RESHAPE F201 AS SAMPLED FROM DISTRIBUTION GAMMA SHAPE WITH PARAMETERS LB(0) MEAN(106) DEVIATION(50)
147 SEED OF MT201 < READ FROM SEEDLIST
148 RESHAPE MT201 AS SAMPLED FROM DISTRIBUTION EXPONENTIAL WITH PARAMETERS MEAN(56)
149 SEED OF F40E < READ FROM SEEDLIST
150 RESHAPE F40E AS SAMPLED FROM DISTRIBUTION EXPONENTIAL WITH PARAMETERS MEAN(95)
151 SEED OF MT40E < READ FROM SEEDLIST
152 RESHAPE MT40E AS SAMPLED FROM DISTRIBUTION EXPONENTIAL WITH PARAMETERS MEAN(21)
153 SEED OF F20E < READ FROM SEEDLIST
154 RESHAPE F20E AS SAMPLED FROM DISTRIBUTION GAMMA SHAPE WITH PARAMETERS LB(2) MEAN(144) DEVIATION(84)
155 SEED OF MT20E < READ FROM SEEDLIST
156 RESHAPE MT20E AS SAMPLED FROM DISTRIBUTION EXPONENTIAL WITH PARAMETERS MEAN(14)
157 SEED OF INTERVAL < READ FROM SEEDLIST
158 RESHAPE INTERVAL AS SAMPLED FROM DISTRIBUTION GAMMA SHAPE WITH PARAMETERS LB(0) MEAN(0.73) DEVIATION(0.62)
159 SEED OF LIFTONTIME < READ FROM SEEDLIST
160 RESHAPE LIFTONTIME AS SAMPLED FROM DISTRIBUTION UNKNOWN WITH PARAMETERS LB(0.4 MINUTE) MEAN(0.6 MINUTE) DEVIATION(0.2 MINUTE)
161 SEED OF LIFTOFFTIME < READ FROM SEEDLIST
RESHAPE LIFTOFFTIME AS SAMPLED FROM DISTRIBUTION UNKNOWN WITH PARAMETERS LB(0.4 MINUTE) MEAN(0.6 MINUTE) DEVIATION(0.2 MINUTE)
RESHAPE CRANETIME AS SAMPLED FROM DISTRIBUTION UNKNOWN WITH PARAMETERS LB(1.3 MINUTES) MEAN(1.8 MINUTES) DEVIATION(0.5 MINUTE)
SEED OF UNIF < READ FROM SEEDLIST
RESHAPE UNIF AS SAMPLED FROM DISTRIBUTION UNIFORM
SEED OF UNIFS < READ FROM SEEDLIST
RESHAPE UNIFS AS SAMPLED FROM DISTRIBUTION UNIFORM

F40I : Randomstream
  Full 40 ft. Import
MT40I : Randomstream
  Empty 40 ft. Import
F20I : Randomstream
  Full 20 ft. Import
MT20I : Randomstream
  Empty 20 ft. Import
F40E : Randomstream
  Full 40 ft. Export
MT40E : Randomstream
  Empty 40 ft. Export
F20E : Randomstream
  Full 20 ft. Export
MT20E : Randomstream
  Empty 20 ft. Export
INTERVAL : Randomstream
  Inter arrival time of ships
LIFTONTIME : Randomstream
  Time needed to place container on transporter
LIFTOFFTIME : Randomstream
  Time needed to pick up container from transporter
CRANETIME : Randomstream
  Time needed to handle container by quaycrane
UNIF : Randomstream
  Random number from uniform distribution
UNIFS : Randomstream
  Random number from uniform distribution exclusively for ships

[170 - 171] :

The simulation is started by activating the SHIPGENERATOR which will start to generate ships. The model will run for RUNTIME hours, during which the MAIN module is passive.

ACTIVATE SHIPCREATION FROM GENERATE IN SHIPGENERATOR
WAIT RUNTIME
SHIPCREATION : Component
Generator of ships

[173 - 176] :

The end of the simulation has been reached and the results are analysed and sent to an outputfile called OUTPUTBLOCK. MAIN cancels all further activities, prints the statistics of all the queue's in the model and terminates itself.

173 CALL OUTPUTBLOCK
174 CANCEL ALL
175 PRINT STATISTICS
176 TERMINATE

OUTPUTBLOCK : Macro
Contains simulation run output
PROCESS OF MAIN

SPECIFY INPUT

YES

NO

CALL INPUTBLOCK

ANSWER INPUT " QUESTIONS

READ SPLDISTR

GENERATE "NRCRANES" QUARYCRANES
JOIN TO PASSIVECRANES

GENERATE "HRTTS" TRACTOR-TRAILERS
JOIN TO TOOLAVAILBLE

GENERATE "HRTGCS" YARDCRANES
ACTIVATE YARDCRANES

RESHAPE ALL RANDOMSTREAMS

ACTIVATE SHIPGENERATOR

WAIT RUNTIME

CALL OUTPUT

TERMINATE
18.2 Description of module SHIPGENERATOR

This module is activated by MAIN from the label GENERATE. In these lines of the module a new ship is created and with it all the sets and queue's that belong to a ship. The attributes of the ship are determined by the macro SHIPATTRIBUTES. The macro STACKALLOCATION allocates for each ship a part in the stacking areas. The size of the appointed area depends on the amount of containers to be exchanged. When the ship has been created and all of its attributes are determined she is joined to the queue SEA and activated in its SHIPPROCESS with a delay of 5 days.

```plaintext
3 GENERATE:
4 THIS SHIP < NEW SHIP
5 SHIPNR < S
6 S < S + 1
7 WAITCRANE < NEW SET
8 CRANESET < NEW SET
9 IMPTTLIST < NEW SET
10 FREQLIST < NEW SET
11 MTREQLIST < NEW SET
12 IMPTIQ < NEW SET
13 EXPTIQ < NEW SET
14 CALL SHIPATTRIBUTES
15 CALL STACKALLOCATION
16 JOIN THIS SHIP TO SEA
```

**SHIPNR** : Integer attribute of class component SHIP
Identification number for store files

**S** : Integer attribute of MAIN

**WAITCRANE** : Set attribute of class component SHIP
Set to which serving quaycrane is joined when idle on job

**CRANESET** : Set attribute of class component SHIP
Set to which all quaycranes serving the ship are joined

**IMPTTLIST** : Set attribute of class component SHIP
Set to which tractor-trailers are joined which are in the importcycle for this ship

**FREQLIST** : Set attribute of the class component SHIP
Set to which tractor-trailers are joined when bringing full containers to the ship

**MTREQLIST** : Set attribute of the class component SHIP
Set to which tractor-trailers are
joined when bringing empty containers to the ship

IMPTTQ: Set attribute of the class component SHIP
Set to which tractor-trailers are joined when waiting for an unloading quaycrane

EXPTTQ: Set attribute of the class component SHIP
Set to which tractor-trailers are joined when waiting for a loading quaycrane

SHIPATTRIBUTES: Macro
Determines the characteristics of a ship

STACKALLOCATION: Macro
Determines the stacking areas for the cargo of a ship

[17 - 26]:

For every ship that is generated a new EXPORTGENERATOR is made to create the supply of containers from the hinterland that the ship will take away. In these lines the integer values which the EXPORTGENERATOR will need are determined (the number of 20 Ft., 40 Ft., full and empty containers). After this the EXPORTGENERATOR is activated from the label EXPORT.

```
17 NextExportGen = New ExportGen
18 ExpJob Of NextExportGen = This Ship
19 NrTrucks Of NextExportGen = Exp40 + Exp20
20 Gen20 Of NextExportGen = Exp20
21 Gen20F Of NextExportGen = Req20F
22 Gen20MT Of NextExportGen = Req20MT
23 Gen40 Of NextExportGen = Exp40
24 Gen40F Of NextExportGen = Req40F
25 Gen40MT Of NextExportGen = Req40MT
26 Activate NextExportGen From Export In ExportGenProcess
```

NextExportGen: Reference to class component EXPORTGEN

EXPORTGEN: Class component
Component which creates the trucks to bring the exportcontainers for the ship

ExpJob: Reference to SHIP of NEXTEXPORTGEN

Definition of ship NEXTEXPORTGEN is working for

NrTrucks: Integer attribute of NEXTEXPORTGEN
Number of trucks to be generated

Gen20: Integer attribute of NEXTEXPORTGEN
Number of 20 ft container to be
brought : Integer attribute of NEXTEXPORTGEN
Number of full 20 ft containers to be brought

GEN20MT
Number of empty 20 ft containers to be brought

GEN40
Number of 40 ft containers to be brought

GEN40F
Number of full 40 ft containers to be brought

GEN40MT
Number of empty 40 ft containers to be brought

[27 - 31] :

The SHIPPROCESS is now activated with a delay of 5 days, the SHIPGENERATOR waits a random-time long (the interarrivaltime of a ship) and then repeats from the label GENERATE.

27 ACTIVATE THIS SHIP WITH DELAY 5 DAYS FROM ARRIVAL IN SHIPPROCESS
28 TURNTIME = (NOW + 5 DAYS)
29 INT = (INTERVAL x 24)
30 WAIT INT
31 REPEAT FROM GENERATE

TURNTIME : Real attribute of class component SHIP
Turnaroundtime of a ship

INT : Real attribute of component
Interarrival time till next ship in hours
SHIPGENERATOR

ACTIVATED BY MAIN

GENERATE NEW SHIP

MAKE NEW SETS

CALL SHIPATTRIBUTES

CALL STACKALLOCATION

JOIN SHIP TO SEA

GENERATE EXPORTGENERATOR

ACTIVATE EXPORTGENERATOR

ACTIVATE SHIP WITH DELAY 5 DAYS

WAIT INTERVAL
18.3 Description of macro SHIPATTRIBUTES

In this macro the shiplength, load and neccessary attributes of a generated ship are determined.

[3 - 4] :

The shiplength is determined with a random number between 0 and 20. To each number a shiplength is appointed through SELECT[1]. This function was read in MAIN in the lines [76 - 78].

3

SHIPLENGTH <- SELECT(CEIL(20xUNIFS))

[5 - 16] :

In this block the full import containers of a generated ship are determined. First, with a random number UNIFS, is determined if the ship indeed has full 40 Ft. import containers on board. If so, the amount is determined with a randomstream (in this case the F40I).

The same is done with the full 20 Ft. import containers in the lines [9 - 12].

Then the needed stackingarea in the importstack for this ship is determined; the IMPBAYF. This is the amount of rows of slots where the full import containers can be stacked. Each row (or here called bay) has 7 slots and when the stackingheigth is assumed to be 2, can stack 14 TEU. When the stackingarea is too big (caused by the exponential randomstreams) the block will repeat the descriped actions.

5 REPI:
6 R11 <- UNIFS
7 IMP40F <- 0 IF R11 < 0.24
8 IMP40F <- 2 + (FLOOR(F40I)) IF R11 > 0.24
9 R11 <- UNIFS
10 IMP20F <- 0 IF R11 < 0.04
11 IMP20F <- FLOOR(F20I) IF R11 > 0.04
12 IMPBAYF <- CEIL((IMP20F+IMP40Fx2)+(2x7x2)) IF IMPBAYF > 32
13 WRITE "SHlP";SHIPNR;"IMPBAYF TOO LARGE" TO RESULTS WITH IMAGE
14 REPEAT FROM REPI
15 END

R11: Real attribute of MAIN
Help variable used to determine ship cargo
UNIFS: Randomstream
Random number from uniform distribution
SHIPLENGTH: Length of the ship in meters
IMP40F: Number of full 40 ft import containers
IMP20F : Number of full 20 ft import containers
IMPBAYF : Number of stack bays needed for full import containers

[18 - 29] :

With the same method as described above the amount of empty import containers and the stacking area in the emptystack (IMPBAYMT) are determined.

18 REP2:
19 R11 <= UNIFS
20 IMP40MT <= 0 IF R11 <= 0.40
21 IMP40MT <= FLOOR(MT401) IF R11 > 0.40
22 R11 <= UNIFS
23 IMP20MT <= 0 IF R11 <= 0.63
24 IMP20MT <= 1 + (FLOOR(MT201)) IF R11 > 0.63
25 IMPBAYMT <= CEIL((IMP20MT+IMP40MTx2)+(3x7))
26 IF IMPBAYMT > 32
27 WRITE "SHIP";SHIPNR;"IMPBAYMT TOO LARGE" TO RESULTS WITH IMAGE
28 REPEAT FROM REP2
29 END

IMP40MT : Number of empty 40 ft import containers
IMP20MT : Number of empty 20 ft import containers
IMPBAYMT : Number of stack bays needed for empty import containers

[31 - 33] :

Here the total amount of import 20 Ft., 40 Ft. and TEU are determined. These integers are used in the module QUAYCRANE and macr QUAYLOAD.

31 IMP20 <= IMP20F + IMP20MT
32 IMP40 <= IMP40F + IMP40MT
33 IMPTEU <= IMP20 + (2xIMP40)

IMP20 : Total number of 20 ft import containers
IMP40 : Total number of 40 ft import containers
IMPTEU : Total amount of imported TEU

[35 - 59] :

Again with the same method, now the full export containers, line [35 - 46], and the empty export containers, line [48 - 59], are determined.
22

REP3:
36 R11 < UNIFS
37 REQ20F = 0 IF R11 ≤ 0.03
38 REQ20F = FLOOR(P20E) IF R11 > 0.03
39 R11 < UNIFS
40 REQ40F = 0 IF R11 ≤ 0.12
41 REQ40F = FLOOR(P40E) IF R11 > 0.12
42 EXPBAYF = CEIL((REQ20F+REQ40Fx2)/(3x7x2))
43 IF EXPBAYF > 32
44 WRITE "SHIP";SHIPNR;"EXPBAYF TOO LARGE" TO RESULTS WITH IMAGE
45 REPEAT FROM REP3
46 END
47
48 REP4:
49 R11 < UNIFS
50 REQ40MT = 0 IF R11 ≤ 0.83
51 REQ40MT = 1 + (FLOOR(MT40E)) IF R11 > 0.83
52 R11 < UNIFS
53 REQ20MT = 0 IF R11 ≤ 0.59
54 REQ20MT = 1 + (FLOOR(MT20E)) IF R11 > 0.59
55 EXPBAYMT = CEIL((REQ20MT+REQ40MTx2)/(3x7))
56 IF EXPBAYMT > 32
57 WRITE "SHIP";SHIPNR;"EXPBAYMT TOO LARGE" TO RESULTS WITH IMAGE
58 REPEAT FROM REP4
59 END

REP3:
35 REP3:
36 R11 < UNIFS
37 REQ20F = 0 IF R11 ≤ 0.03
38 REQ20F = FLOOR(P20E) IF R11 > 0.03
39 R11 < UNIFS
40 REQ40F = 0 IF R11 ≤ 0.12
41 REQ40F = FLOOR(P40E) IF R11 > 0.12
42 EXPBAYF = CEIL((REQ20F+REQ40Fx2)/(3x7x2))
43 IF EXPBAYF > 32
44 WRITE "SHIP";SHIPNR;"EXPBAYF TOO LARGE" TO RESULTS WITH IMAGE
45 REPEAT FROM REP3
46 END
47
48 REP4:
49 R11 < UNIFS
50 REQ40MT = 0 IF R11 ≤ 0.83
51 REQ40MT = 1 + (FLOOR(MT40E)) IF R11 > 0.83
52 R11 < UNIFS
53 REQ20MT = 0 IF R11 ≤ 0.59
54 REQ20MT = 1 + (FLOOR(MT20E)) IF R11 > 0.59
55 EXPBAYMT = CEIL((REQ20MT+REQ40MTx2)/(3x7))
56 IF EXPBAYMT > 32
57 WRITE "SHIP";SHIPNR;"EXPBAYMT TOO LARGE" TO RESULTS WITH IMAGE
58 REPEAT FROM REP4
59 END

REQ20F : Number of full 20 ft export containers
REQ40F : Number of full 40 ft export containers
EXPBAYF : Number of stack bays needed for full export containers
REQ40MT : Number of empty 40 ft export containers
REQ20MT : Number of empty 20 ft export containers
EXPBAYMT : Number of stack bays needed for empty export containers

[62 - 77] :

In these lines the last attributes of the ship are determined and the macro returns to the module SHIPGENERATOR.

62 REQ20 = REQ20F + REQ20MT
63 REQ40 = REQ40F + REQ40MT
64 EXP20 = REQ20
65 EXP40 = REQ40
66 EXPTEU = EXP20 + (2xEXP40)
67 IMPTEU = IMP20F + (2xIMP40F)
68 EXPTEUMT = EXP20 + (2xEXP40F)
69 IMPTEUMT = IMP20MT + (2xIMP40MT)
70 EXPTEUEMT = EXP20MT + (2xEXP40MT)
71 TOTMOVES = IMP20 + IMP40 + EXP20 + EXP40
72 EXCHANGE = (IMPTEU + EXPTEU)
73 REGCRANES = 1 IF (TOTMOVES ≤ 120)
74 REGCRANES = 2 IF (TOTMOVES > 120)
75 TURNTIME = 0
76 SERVETIME = 0
77 RETURN
containers

EXP20 : Same as REQ20
EXP40 : Same as REQ40
EXPTEU : Total amount of exported TEU
IMPTEUF : Amount of full imported TEU
EXPTEUF : Amount of full exported TEU
IMPTEUMT : Amount of empty imported TEU
EXPTEUMT : Amount of empty exported TEU
TOTMOVES : Total amount of containers exchanged

EXCHANGE : Same as TOTMOVES
REQCRANES : Number of quaycranes needed
TURNTIME : Real attribute
            Turnaroundtime
SERVETIME : Real attribute
            Time alongside QUAY
18.4 Description of macro STACKALLOCATION

This macro is called upon from the SHIPGENERATOR. When a new ship is generated special areas are appointed to the ship in all the stacking areas (the import-, export- and empty- stack).

Depending on the amount of full import (line 3-11), full export (line 12-20), empty import (line 21-29) and empty export containers (line 30-38) the ship gets a number of bays in each stacking area. To define this area the boundaries are defined with a lowerbound (for example LBIMPF is the first bay of the area to be used for stacking full import containers in the import stack) and an upperbound (UBEXPMT is the last bay of the area to be used for stacking empty export containers in the empty stack).

When an upperbound exceeds the maximum bay number 64 (all stacks exists of 64 slots in the length = 64 bays), the area for the ship is once more defined, now with the new lowerbound bay number 0.

After appointing all the necessary areas to the ship the macro returns to the SHIPGENERATOR.

3 IMPORTFULL:
4 LBIMPF < UB34
5 UBIMPF < LBIMPF + IMPBAYF
6 UB34 < UBIMPF
7 IF UB34 > 64
8 UB34 < 0
9 REPEAT FROM IMPORTFULL
10 END
11
12 EXPORTFULL:
13 LBEXPF < UB12
14 UBEXPF < LBEXPF + EXPBAYF
15 UB12 < UBEXPF
16 IF UB12 > 64
17 UB12 < 0
18 REPEAT FROM EXPORTFULL
19 END
20
21 IMPORTEMPTY:
22 LBIMPMT < UBS
23 UBS < LBIMPMT + IMPBAYMT
24 IF UBS > 64
25 UBS < 0
26 REPEAT FROM IMPORTEMPTY
27 END
28
29 EXPORTEMPTY:
30 LBEXPMT < UBS
31 UBEXPMT < LBEXPMT + EXPBAYMT
32 UBS < UBEXPMT
33 IF UBS > 64
34 UBS < 0
35 REPEAT FROM EXPORTEMPTY
36 END
37
38 RETURN
LBIMPF : Integer attribute of class component SHIP
Lower bound of full import container stack bays

UB34 : Integer attribute of MAIN
Upper bound of import stack bays
(stack 3 and 4)

UBIMPF : Integer attribute of class component SHIP
Upper bound of full import container stack bays

IMPBAYF : Integer attribute of class component SHIP
Amount of bays needed for full import containers

LBEXPF : Integer attribute of class component SHIP
Lower bound of full export container stack bays

UB12 : Integer attribute of MAIN
Upper bound of export stack bays
(stack 1 and 2)

UBEXPF : Integer attribute of class component SHIP
Upper bound of full export container stack bays

EXPBAYF : Integer attribute of class component SHIP
Amount of bays needed for full export containers

LBIMPMT : Integer attribute of class component SHIP
Lower bound of empty import container stack bays

UB5 : Integer attribute of MAIN
Upper bound of empty stack bays
(stack 5)

IMPBAYMT : Integer attribute of class component SHIP
Amount of bays needed for empty import containers

LBEXPMT : Integer attribute of class component SHIP
Lower bound of empty export container stack bays

UBEXPMT : Integer attribute of class component SHIP
Upper bound of empty export container stack bays

EXPBAYMT : Integer attribute of class component SHIP
Amount of bays needed for empty export containers
FIG. 18.4.1
18.5 Description of module SHIPPROCESS

[3 - 17] :

After the ship is activated by the SHIPGENERATOR it leaves the SEA and enters the ANCHORAGE. Here the ship activates the BERTHMASTER and is to wait if no quayspace is available. The BERTHMASTER reactivates the ship when there is enough room and the ship sails to the quay. When the ship arrives at the quay she activates the module MEASUREMENT to start the measurements of the yardcranes efficiency's (if not already active) and the CRANEMASTER to request the cranes.

ANCHORAGE : Queue
Contains all the ships waiting to berth

BERTHALLOCATION: Component
Allocates berths to ships

QUAY : Queue
Contains the ships berthed alongside the quay

CRANEEFFICIENCY: Component
Measures the performance of the yardcranes

BERTH[i] : Queue
Contains a berthed ship

BERTHNR : Integer attribute of class component SHIP
Number of the berthplace of the ship

CRANEREQLST : Queue
Contains all the ships that still need quaycranes

CRANEALLOCATION: Component
Allocates quaycranes to ships

SERVETIME : Real attribute
Time alongside QUAY
Each ship generates its own IMPORTGENERATOR to generate the trucks needed to remove the imported containers of the ship from the yard. In these lines the total number of trucks is defined and the sort of container they should collect. The IMPORTGEN is activated with a delay of 1 day and the ship waits until it is served completely. (i.e. the crane(s) finished servicing the ship and leave the craneset of the ship)

18 NEXTIMPORTGEN $ NEW IMPORTGEN
19 IMPJOB OF NEXTIMPORTGEN $ THIS SHIP
20 NRTRANS OF NEXTIMPORTGEN $ IMP20 + IMP40
21 HIN20 OF NEXTIMPORTGEN $ IMP20
22 HIN20F OF NEXTIMPORTGEN $ IMP20F
23 HIN20MT OF NEXTIMPORTGEN $ IMP20MT
24 HIN40 OF NEXTIMPORTGEN $ IMP40
25 HIN40F OF NEXTIMPORTGEN $ IMP40F
26 HIN40MT OF NEXTIMPORTGEN $ IMP40MT
27 ACTIVATE NEXTIMPORTGEN WITH DELAY 1 DAY FROM IMPORT IN IMPORTGENPROCESS
28 WAIT
29
30 SERVICETIME:
31 WAIT WHILE CRANESET IS NOT EMPTY
32

NEXTIMPORTGEN : Reference to class component
IMPORTGEN : Class component
Component which creates the trucks to pick up the containers brought by a ship
IMPJOB : Reference to SHIP of NEXTIMPORTGEN
Definition of ship NEXTIMPORTGEN is working for
NRTRANS : Integer attribute of NEXTIMPORTGEN
Number of trucks to be generated
HIN20 : Integer attribute of NEXTIMPORTGEN
Number of 20 ft containers to be picked up
HIN20F : Integer attribute of NEXTIMPORTGEN
Number of full 20 ft containers to be picked up
HIN20MT : Integer attribute of NEXTIMPORTGEN
Number of empty 20 ft containers to be picked up
HIN40 : Integer attribute of NEXTIMPORTGEN
Number of 40 ft containers to be picked up
The ship is ready to leave now. A few variables are stored and the ship leaves the CRANEREQLST if the ship still had a request for a crane. The module MEASUREMENT is activated from the label 'stop' to end the measurements if this is the last ship at the quay. The ship has to wait a deberthtime, leaves the QUAY, frees the berth (OCCUPIED[i] = FALSE), updates the quayspace and activates the BERTHMASTER. Finally all the important ship-integers are written to the outfile SHIPOUTPUT and the process of this ship is terminated.

```
33 QUITBERTH:
34 SERVETIME = NOW - SERVETIME
35 TIMEPERBOX = (SERVETIME + TOTMOVES)
36 STORE TIMEPERBOX AS "B"
37 STORE TOTMOVES AS "M"
38 LEAVE CRANEREQLST IF REDCRANES > 0
39 ACTIVATE EACH TRANSPORTER IN EXPTTQ FROM ENWORK IN TTPROCESS
40 ACTIVATE CRANEEFFICIENCY FROM STOP IN MEASUREMENTS IF CRANEEFFICIENCY IS NOT ACTIVE
41 WAIT DEBERTHTIME
42 LEAVE QUAY
43 LEAVE BERTH(BERTHRN)
44 OCCUPIED(BERTHRN) = FALSE
45 QUAYSACE = QUAYSACE + (SHIPLENGTH + 15)
46 ACTIVATE BERTHALLOCATION FROM BERTHING IN BERTHMASTER IF BERTHALLOCATION IS NOT ACTIVE
47 TURNTIME = ((NOW - ARRIVALTIME) - 120)
48 STORE TURNTIME AS "T"
49 ENTER SEA IF (IMPTTQ IS NOT EMPTY) v (EXPTTQ IS NOT EMPTY)
50 WRITE SHIPNR;SHIPLENGTH; (ARRIVALTIME+120);TURNTIME;IMPTEUF;EXPTEUF;IMPTEUMT;EXPTEUMT TO SHIPOUTPUT WITH IMAGE xxxxxxxx
51 TOTTEU = TOTTEU + EXCHANGE
52 TERMINATE
```

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIN40F</td>
<td>Integer attribute of NEXTIMPORTGEN Number of full 40 ft containers to be picked up</td>
</tr>
<tr>
<td>HIN40MT</td>
<td>Integer attribute of NEXTIMPORTGEN Number of empty 40 ft containers to be picked up</td>
</tr>
<tr>
<td>TIMEPERBOX</td>
<td>Real attribute of class component SHIP Average time spent on handling one container</td>
</tr>
<tr>
<td>TOTMOVES</td>
<td>Total amount of containers exchanged</td>
</tr>
<tr>
<td>EXPTTQ</td>
<td>Set attribute of the class component SHIP Set to which tractor-trailers are joined when waiting for a loading quaycrane</td>
</tr>
</tbody>
</table>
DEBERTHTIME : Real attribute of MAIN
Time needed by a ship to leave
berth and sail away

TURNTIME : Real attribute
Turnaroundtime
SHIPPROCESS

ACTIVATED BY SHIPGENERATOR

LEAVE SEA
ENTER ANCHORAGE
ACTIVATE BERTHMASTER

WAIT

SAIL TO BERTH
ENTER QUAY
ACTIVATE MEASUREMENTS
ACTIVATE CRANEMASTER

GENERATE NEW IMPORTGENERATOR
ACTIVATE IMPORTGEN WITH DELAY 1 DAY

WAIT

WAIT WHILE CRANESET OF THIS SHIP IS NOT EMPTY

ACTIVATE MEASUREMENTS
WAIT BERTHI NTIME
LEAVE QUAY

ACTIVATE BERTHMASTER

TERMINATE
18.6 Description of module MEASUREMENTS

[3 - 11]:

This module is activated from the label START by an arriving SHIP. When the length of QUAY is one, the arriving ship is the first at the quay and this module starts to keep track off the waitingtimes of the yardcranes while the terminal services the ship. Then the module waits for reactivation, which will occur when the last ship departs from the quay.

```
3 START:
4 IF LENGTH OF QUAY = 1
5 FOR P = 1 TO NRRTGCS
6   WAITTIME OF YCRANE[P] < 0
7   PAST OF YCRANE[P] < NOW
8 END
9 BERTHTIME < NOW
10 END
11 WAIT
```

P : Integer attribute of MAIN
Parameter

WAITTIME : Real attribute of class component YARDCRANE
Time spent waiting during certain period

PAST : Real attribute of YARDCRANE
Aid for measuring idletime

BERTHTIME : Real attribute of class component YARDCRANE
Length of waittime measuring period

WAITPERC : Real attribute of class component YARDCRANE
Percentage of berthtime period spent waiting

[12 - 27]:

Again the module is activated by a SHIP. This time the module is departing and the module checks the length of QUAY. When the length is one, the departing ship is the last ship to leave the QUAY, and the WAITPERC of all the yardcranes (percentage of time spent waiting while the ships were serviced) is computed and stored in a storefile. Then the module becomes passive again. When the departing ship is not the last ship in QUAY, the module will not store the WAITPERC, but wait until the last ship.
12 STOP ;
13 IF LENGTH OF QUAY = 1
14 BERTHTIME < NOW - BERTHTIME
15 FOR P = 1 TO NRRTGCS
16 IF YCRANE(P) BELONGS TO NOWORK
17 WAITTIME OF YCRANE(P) = WAITTIME OF YCRANE(P) + (NOW - PAST OF YCRANE(P))
18 END
19 WAITPERC OF YCRANE(P) = CEIL(100 * (WAITTIME OF YCRANE(P) + BERTHTIME))
20 STORE WAITPERC OF YCRANE(P) AS "1" IF P = 1
21 STORE WAITPERC OF YCRANE(P) AS "2" IF P = 2
22 STORE WAITPERC OF YCRANE(P) AS "3" IF P = 3
23 STORE WAITPERC OF YCRANE(P) AS "4" IF P = 4
24 STORE WAITPERC OF YCRANE(P) AS "5" IF P = 5
25 END
26 END
27 PASSIVATE
18.7 Description of module BERTHMASTER

This module checks to see if ships in the queue ANCHORAGE can berth or not. The module is activated by ships arriving at the anchorage of the port. When activated the BERTHMASTER first makes sure that a ship is waiting in the anchorage to berth. For the first ship in the anchorage he checks if the available quaylength is sufficient. If the available quayspace is sufficient, but both berthplaces are already occupied (OCCUPIED[1]=OCCUPIED[2]=TRUE), the BERTHMASTER leaves the ship in the anchorage and becomes passive. If this is not the case the BERTHMASTER will check the first section of the quay to see if it is occupied or not. If this section is already occupied he will go on to the next to find a free section of quay.

When the BERTHMASTER has found the free section he immediately allocates this to the ship he is looking for. When the berth at the quay has been allocated he must update the QUAYSPACE. The ship he has allocated the berth to is activated and the BERTHMASTER repeats his process.

If there are no other waiting ships, or the available quaylength is insufficient, he will become passive and wait until activated again.

3 BERTHING:
4 IF ANCHORAGE IS NOT EMPTY
5 NEXTSHIP = FIRST OF ANCHORAGE
6 IF QUAYSPACE > (SHIPLength OF NEXTSHIP + 15)
8 IF OCCUPIED[1] = FALSE
9 OCCUPIED[1] = TRUE
10 BERTHNR OF NEXTSHIP = 1
11 GOTO UPDATESPACE
12 END
14 OCCUPIED[2] = TRUE
15 BERTHNR OF NEXTSHIP = 2
16 GOTO UPDATESPACE
17 END
18 UPDATESPACE :
19 QUAYSPACE = QUAYSPACE - (SHIPLength OF NEXTSHIP + 15)
20 REACTIVATE NEXTSHIP
21 WAIT 5 MINUTES
22 REPEAT FROM BERTHING
23 END
24 QUAYFULL:
25 END
26 PASSIVATE

ANCHORAGE : Queue
Contains all the ships waiting to berth

NEXTSHIP : Reference to class component
SHIP
The ship which is to be served
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUAYSPACE</td>
<td>next: Integer attribute of main. The amount of quay length available.</td>
</tr>
<tr>
<td>SHIPLENGTH</td>
<td>: Length of the ship in meters.</td>
</tr>
<tr>
<td>OCCUPIED[i]</td>
<td>: Logical attribute of MAIN. Indicates whether berth[i] is free or not.</td>
</tr>
<tr>
<td>BERTHNR</td>
<td>: Integer attribute of class component SHIP. Number of the berthplace of the ship</td>
</tr>
</tbody>
</table>
ACTIVATED BY SHIP

IF ANCHORAGE IS NOT EMPTY

YES

IS FREE QUAYSACE SUFFICIENT

YES

ARE BERTH(1) AND BERTH(2) BOTH OCCUPIED?

YES

UPDATE QUAYSACE

NO

IS AREA 1 FREE

YES

OCCUPY AREA 1

NO

IS AREA 2 FREE

YES

OCCUPY AREA 2

NO

UPDATE QUAYSACE

REACTIVATE SHIP WAIT 5 MINUTES

PASSIVATE
18.8 Description of module CRANEMASTER

This module is activated from the label REQLSTCHECK by a ship that has berthed and needs cranes. The first thing the CRANEMASTER does is look if there are any requests for cranes that he needs to handle. Then he next looks if there are any cranes available at all. This is done by looking in the queue PASSIVECRANES. If these two conditions are met, the CRANEMASTER proceeds his actions. The total exchange of containers (MOVES) of the first ship in the queue CRANEREQLST is looked at. If MOVES is less than 120 (= 120 containers) and the requesting ship already has a crane, the ship is removed from the CRANEREQLST and the process of the CRANEMASTER is repeated from the beginning.

CRANEREQLST : Queue
Contains all the ships that still need quaycranes

PASSIVECRANES : Queue
Contains all the quaycranes which are not assigned to a ship

CRANESET : Set attribute of class component SHIP
Set to which all quaycranes serving the ship are joined

REQCRANES : Number of quaycranes needed

MOVES : Integer attribute of CRANEALLOCATION
Amount of containers still to be handled on this ship

Now that the CRANEMASTER has determined that a crane can be allocated to the requesting ship, he proceeds to allocate. The amount of cranes needed by the ship is reduced by one after allocation. If the ship now has all the cranes he needs (REQCRANES = 0) the ship is removed from the CRANEREQLST. Because the allocated crane will need transporters to serve it, the crane is
joined to the queue TTREQLST and the TTMASTER is activated to allocate transporters to the crane. The crane is allocated to a ship and its transporters are on the way. The crane is activated from the label UNLOADSHIP. The ship is signalled that servicing can begin. After this the CRANEMASTER repeats his process from the beginning. If there are no more crane-requesting ships or cranes to allocate, the CRANEMASTER becomes passive.

```
13 IDLECRANE < FIRST OF PASSIVECRANES
14 REMOVE IDLECRANE FROM PASSIVECRANES
15 JOIN IDLECRANE TO CRANESET
16 SHIPONHAND OF IDLECRANE < THIS SHIP
17 REQCRANES + REQCRANES - 1
18 REMOVE THIS SHIP FROM CRANEREQLST IF REQCRANES = 0
19 JOIN IDLECRANE TO TTREQLST
20 ACTIVATE TRACTORALLOCATION FROM TTINSTRUCTIONS IN TTMASTER IF TRACTORALLOCATION IS NOT ACTIVE
21 ACTIVATE IDLECRANE FROM UNLOADSHIP IN QUAYCRANEPROCESS
22 ACTIVATE THIS SHIP FROM SERVICETIME IN SHIPPROCESS IF THIS SHIP IS NOT ACTIVE
23 REPEAT FROM REQLSTCHECK
24 END
25 END
26 PASSIVATE

IDLECRANE : Reference to class of components
QUAYCRANE
Crane which is next to be allocated

TTREQLST : Queue
Contains cranes which need tractor-trailers

TRACTORALLOCATION: Component
This component allocates tractor-trailers to quaycranes
CRANEMASTER

ACTIVATED BY SHIP

IS CRANEREQUEST NOT EMPTY

YES

ARE CRANES AVAILABLE

YES

IF TOTMOVES > 120 AND SHIP ALREADY HAS A CRANE

YES

REMOVE SHIP FROM CRANEREQUEST

NO

JOIN CRANE TO CRANESET OF SHIP

REPLACE SHIP FROM CRANEREQUEST IF NO MORE CRANES NEEDED

ACTIVATE TTMASHER
ACTIVATE QUAYCRANE
REACTIVATE SHIP

PASSIVATE
18.9 Description of module TTMASTER

This module is activated by the CRANEMASTER after it has allocated a crane to a ship. The TTMASTER takes care of transporter allocation and activation for the allocated crane. The module is activated from the label TTINSTRUCTIONS, after which the TTMASTER checks to see if there are any requests from quaycranes for transporters in the queue TTREQLST. The first crane in this queue is helped first. This crane will get TTPERCRANE transporters, depending on the chosen amount of transporters per crane in the input. Depending on where the ship is berthed alongside the quay the driving-times of the transporters on the quay are determined. After the crane has received all its transporters, the crane is removed from the TTREQLST and the TTMASTER repeats his process from the beginning. If there are no more cranes needing transporters, the TTMASTER becomes passive until activated again by the CRANEMASTER.

3 TTINSTRUCTIONS:
4 IF TTREQLST IS NOT EMPTY
5 ALLOCRATE < FIRST OF TTREQLST
6 FOR N < 1 TO TTPERCRANE
7 WAIT WHILE TTAVAILABLE IS EMPTY
8 SERVICETT < FIRST OF TTAVAILABLE
9 WORKSHIP OF SERVICETT < SHIPONHAND OF ALLOCRATE
10 IF WORKSHIP OF SERVICETT BELONGS TO BERTH(1)
11 QUAYTIME OF SERVICETT < TIME1
12 END
13 IF WORKSHIP OF SERVICETT BELONGS TO BERTH(2)
14 QUAYTIME OF SERVICETT < TIME2
15 END
16 JOIN SERVICETT TO TTSET OF ALLOCRATE
17 REMOVE SERVICETT FROM TTAVAILABLE
18 ACTIVATE SERVICETT FROM STARTUP IN TTPROCESS
19 WAIT 0.5 MINUTE
20 END
21 N < 0
22 REMOVE ALLOCRATE FROM TTREQLST
23 REPEAT FROM TTINSTRUCTIONS
24 END
25 PASSIVATE

ALLOCRATE : Reference to class component QUAYCRANE
Quaycrane which is next to receive tractor-trailers

N : Integer attribute of TRACTORALLOCATION Parameter

TTAVAILABLE : Queue
Contains the idle tractor-trailers

SERVICETT : Reference to class component TRANSPORTER
Tractor-trailer allocated to
Quaycrane
: Reference to ship of class component TRANSPORTER
This is the ship the transporter is driving for

QUAYTIME
: Real attribute of class component TRANSPORTER
Driving time of tractor-trailers on the quay area from bridge to berth

TIME1
: Real attribute of MAIN
Time needed by a tractor-trailer to reach berth[1] once on the quay

TIME2
: Real attribute of MAIN
Time needed by a tractor-trailer to reach berth[2] once on the quay

TTSET
: Reference to set of class component QUAYCRANE
Set in which all the tractor-trailer assigned to the quaycrane are registered
ACTIVATED BY CRANEMASTER

THERE IS A REQUEST FROM A CRANE FOR IT?

DEFINE THE CRANE

FOR N < TYPERCANE

REMOVE CRANE FROM REQUESTLIST

WAIT WHILE TYPENABLE IS EMPTY

ALLOCATE IT TO CRANE
DEFINITE DRIVETIME
JOIN IT TO SET OF CRANE

REMOVE IT FROM TYPABLE
ACTIVATE IT
WAIT 0.5 MIN

TIMASTER

PASSIVATE
18.10 Description of module QUAYCRANEPROCESS

[3 - 9] :

When a quaycrane has been allocated to a ship by the CRANEMASTER he is also activated from the label UNLOADSHIP, which is part of the QUAYCRANEPROCESS. To be able to register the time that the quaycrane spends idle during its operation it is necessary to set the time at which operation began.

As long as there are containers to unload from the ship the quaycrane will remain in the UNLOADSHIP-cycle. At the beginning of each cycle the quaycrane checks how many TEU still have to be unloaded. If the amount of TEU to unload is smaller or equal to two and there is already one crane waiting to unload this, then the (possible) second quaycrane can switch to the LOADSHIP-cycle.

3 UNLOADSHIP:
4 RESETTIME <= NOW
5 SHIPUNLOAD:
6 IF IMPTEU OF SHIPONHAND > 0
7 IF (LENGTH OF WAITCRANE OF SHIPONHAND = 1) ^ (IMPTEU OF SHIPONHAND <= 2)
8 GOTO LOADSHIP
9 END

RESETTIME : Real attribute of QUAYCRANE
Time variable to record the time spent idle

SHIPONHAND : Reference to ship of class component QUAYCRANE
This is the ship the quaycrane is working on

WAITCRANE : Set attribute of class component SHIP
Set to which serving quaycrane is joined when idle on job

[10 - 17] :

If the quaycrane goes ahead to unload he must first make sure that there is a transporter to come and pick up the load. The empty transporters are queued in the IMPTTQ of the ship on the quay, waiting to be called to a quaycrane. If this queue is empty the quaycrane must wait. The time spent waiting is registered to be able to measure performance level. The first transporter in the queue is removed from the IMPTTQ and activated to drive to the calling quaycrane. When the quaycrane has called a transporter, the load of this transporter, or rather what is going to be unloaded from the ship, is determined by calling
the macro QUAYLOAD.

THEN
IMPTT : Real attribute of QUAYCRANE
Reference to TRANSPORTER.
Tractor-trailer onto which container is unloaded

IMPTTQ : Set attribute of the class component SHIP
Set to which tractor-trailers are joined when waiting for an unloading quaycrane

WORKCRANE : Reference to class component QUAYCRANE
Defines which quaycrane the tractor-trailer must drive to

QUAYLOAD : Macro
Determines the load placed on the tractor-trailer

[18 - 30] :

Now that a transporter has been allocated, the quaycrane can proceed to work. First he must remove the transporter from the queue IMPTTQ LIST of the ship the quaycrane is working on. In this queue are all the transporters who are already appointed to this ship and available for the unloading of the ship. This queue is necessary for the control and the giving of directions to the transporters by the macro TTCONTROL (see paragraph 18.13). After this the quaycrane picks up a container from the ship and is ready to place it on the allocated transporter. As long as the transporter is not yet under the quaycrane he must wait. The time spent waiting is registered. When the transporter is underneath the quay-crane he can put on the container. If the transporter is to receive another container, the quaycrane will work again. When the transporter has received his load he is removed from underneath the quaycrane and activated to leave. The UNLOADSHIP-cycle now repeats itself.
18 REMOVE IMPTT FROM IMPTT_LIST OF SHIPONHAND
19 ACTIVATE IMPTT FROM DRIVETO.CRANE IN TTPROCESS
20 LEAVE WAITCRANE OF SHIPONHAND
21 WORK CRANETIME
22 WORK LIFTONTIME
23 IF BOX OF IMPTT = 2
24 WORK CRANETIME
25 WORK LIFTONTIME
26 END
27 REMOVE IMPTT FROM UNDERCRANE
28 ACTIVATE IMPTT FROM DRIVETOYARD IN TTPROCESS
29 REPEAT FROM SHIPUNLOAD
30 END

CRANETIME : Randomstream
Time needed to handle container by quaycrane
LIFTONTIME : Randomstream
Time needed to place container on transporter
BOX : Integer attribute of transporter
The number of containers on a transporter
UNDERCRANE : Reference to set of class component QUAYCRANE
Set in which tractor-trailers waiting underneath a quaycrane are registered

[32 - 37] :

When all the import-containers have been removed from the ship, the quaycrane will switch to the LOADSHIP-cycle, in which it will remain until all the export-containers have been put aboard. At the beginning of each cycle the quaycrane checks how many TEU still have to be lifted on board. If the amount of TEU is smaller or equal to 2 and there is already another quaycrane waiting to lift this aboard, the quaycrane will switch to JOBDONE.

32 LOADSHIP:
33 UNLOADING < FALSE
34 IF EXPTEU OF SHIPONHAND > 0
35 IF (LENGTH OF WAITCRANE OF SHIPONHAND = 1) ^ (EXPTEU OF SHIPONHAND <= 2)
36 GOTO JOBDONE
37 END

UNLOADING : Logical attribute of QUAYCRANE
Defines which cycle the quaycrane is in

[38 - 46] :

If there is more than 2 TEU to lift aboard, the quaycrane will stay in the LOADSHIP-cycle. He must now wait until a transporter is underneath
the quaycrane before he can go to work, because he must first pick up the load from the transporters. If there is no transporter underneath, the quaycrane must wait while the queue of transporters (EXPTTQ) servicing the ship is empty. The time spent waiting is registered. If this queue is not empty, the quaycrane will tell the first transporter to which quaycrane he has to drive (workcrane). At this same moment the load of the transporter is asked and already added to the load on board the ship by the macro QUAYLOAD.

38 IF UNDERCRANE IS EMPTY
39 ENTER WAITCRANE OF SHIPONHAND
40 THEN < NOW
41 WAIT WHILE EXPTTQ OF SHIPONHAND IS EMPTY
42 TIMEIDLE = TIMEIDLE + (NOW - THEN)
43 EXPTT = FIRST OF EXPTTQ OF SHIPONHAND
44 REMOVE EXPTT FROM EXPTTQ OF SHIPONHAND
45 WORKCRANE OF EXPTT = THIS QUAYCRANE
46 CALL QUAYLOAD

LIFTOFFTIME : Randomstream
Time needed to pick up container from transporter

TEU : Integer attribute of transporter
The number of TEU on a transporter

LOAD : Character attribute of class component TRANSPORTER
Defines the load the transporter carries

[47 - 62] :

Then the transporter is activated from the label drivetocrane in the TTPROCESS. The quaycrane takes the first container off the transporter. If the TT carries two containers, the crane puts the first on board and lifts the second off the transporter. Before activation of the transporter, some of its attributes are updated and after that the quaycrane repeats the process from label loadship.

47 ACTIVATE EXPTT FROM DRIVETOCRANE IN TTPROCESS
48 LEAVE WAITCRANE OF SHIPONHAND
49 END
50 WORK LIFTOFFTIME
51 IF BOX OF EXPTT = 2
52 WORK CRANETIME
53 WORK LIFTOFFTIME
54 END
55 BOX OF EXPTT = 0
56 TEU OF EXPTT = 0
57 LOAD OF EXPTT = "O"
58 REMOVE EXPTT FROM UNDERCRANE
59 ACTIVATE EXPTT FROM DRIVETOYARD IN TTPROCESS
60 WORK CRANETIME
61 REPEAT FROM LOADSHIP
62 END
This section of the program describes the actions the quaycrane takes when it has finished servicing the ship. First the performance level is calculated by computing the percentage of the time that the crane spent waiting on transporters. After this the quaycrane will detach itself from the ship, remove all the transporters from the TTSET of the quaycrane and go into a shiftprocedure which lasts just as long as the deberthtime of a ship. This time has been chosen in case the quaycrane is allocated to another ship, in which case it would take some time before the quaycrane is installed and operational again. In case the newly allocated ship will berth in the same place as the previous ship the quaycrane will not need to transfer, but will at least have to wait until the new ship is berthed. When the quaycrane has finished the shiftprocedure, it enters the queue PASSIVECRANES and is available for re-alocation. The CRANEMASTER is activated to see if there are any other ships that need cranes. After this the quaycrane becomes passive and awaits reactivation.

```plaintext
64 JOBDONE:
65 IF (NOW - RESETTIME) > 0
66 PERCIDLE < CEIL(100 * (TIMEIDLE + (NOW - RESETTIME))))
67 STORE PERCIDLE AS "Q"
68 END
69 TIMEIDLE < 0
70 DEPARTSHIP < SHIPONHAND
71 LEAVE CRANESSET OF DEPARTSHIP
72 SHIPONHAND < NONE
73 REMOVE EACH TRANSPORTER IN TTSET FROM TTSET
74 ENTER SHIFTCRANES
75 WAIT SHIFTIME
76 LEAVE SHIFTCRANES
77 UNLOADING < TRUE
78 ENTER PASSIVECRANES
79 ACTIVATE CRANEALLOCATION FROM REQLSTCHECK IN CRANEMASTER IF CRANEALLOCATION IS NOT ACTIVE
80 PASSIVATE
```

**PERCIDLE**
- : Real attribute of QUAYCRANE
- The percentage of time spent idle

**TIMEIDLE**
- : Real attribute of class component QUAYCRANE
- Time spent idle by quaycrane during a job

**DEPARTSHIP**
- : Reference to class component SHIP

**SHIFTCRANES**
- : Set attribute of MAIN
- Set in which shifting quaycranes are registered
18.11 Description of macro QUAYLOAD

This macro consists of two parts: one to determine the load taken from the ship to be placed on a transporter [3 - 66], the other to update the load aboard a ship when a load is taken from a transporter [68 - 88].

[3 - 5]:

If the quaycrane calls the macro and he is unloading the ship, then the macro will start here. A random number is generated and compared to the fraction of 20' import-containers in the total amount of import-containers aboard.

```
3 IF UNLOADING = TRUE
4 R1 < UNIF
5 IF R1 ≤ (IMP20 OF SHIPONHAND + IMP40 OF SHIPONHAND)
```

[6 - 44]:

If the random number is smaller than or equal to this fraction the macro will proceed in this block. This block determines whether the transporter will receive one or two 20 Ft. containers, and whether they are empty containers or full. Again a random number is generated and compared to the fraction of full 20 Ft. containers in the total amount of 20 Ft. containers. If the random number is smaller than or equal to the fraction, the first container will be a 20 Ft. full container. If the random number is larger than the fraction, the first container will be a 20 Ft. empty container. After the determination of the type of the first container to be put on the transporter, the amount of that type on board is updated by reducing with one. The load facts of the transporter are also updated. Since the transporters have a two TEU capacity, and there is already one 20 Ft. container allocated, it is logical that a second 20 Ft. container should be allocated. If there are still 20 Ft. containers on board the macro will proceed with the next block [21 - 39]. If not, then the macro will skip to block [40 - 44].

In block [21 - 39] a random number is again generated to determine whether the second 20 Ft. container will be full or empty. The procedure is identical to the one described in the previous blocks. Again the import-load on board the ship is updated according to which type of container
has been allocated to the transporter. The load facts of the transporter are also updated. In block \([40 - 44]\) the attributes of the transporter, necessary to bring away his load successfully, are set. With this done the macro's task for this transporter has finished and the return command is given.

\[
\begin{align*}
6 & \quad R2 \leftarrow \text{UNIF} \\
7 & \quad \text{IF } R2 < ((\text{IMP}20 \text{F OF SHIPONHAND}) + (\text{IMP}20 \text{ OF SHIPONHAND})) \\
8 & \quad \text{LOAD OF IMPTT } \leftarrow \text{"F"} \\
9 & \quad \text{IMP}20 \text{F OF SHIPONHAND } \leftarrow \text{IMP}20 \text{F OF SHIPONHAND} - 1 \\
10 & \quad \text{GOTO POINT}1 \\
11 & \quad \text{END} \\
12 & \quad \text{IF } R2 > ((\text{IMP}20 \text{F OF SHIPONHAND}) + (\text{IMP}20 \text{ OF SHIPONHAND})) \\
13 & \quad \text{LOAD OF IMPTT } \leftarrow \text{"MT"} \\
14 & \quad \text{IMP}20 \text{MT OF SHIPONHAND } \leftarrow \text{IMP}20 \text{MT OF SHIPONHAND} - 1 \\
15 & \quad \text{END} \\
16 & \quad \text{POINT}1: \\
17 & \quad \text{BOX OF IMPTT } \leftarrow 1 \\
18 & \quad \text{TEU OF IMPTT } \leftarrow 1 \\
19 & \quad \text{IMP}20 \text{ OF SHIPONHAND } \leftarrow \text{IMP}20 \text{ OF SHIPONHAND} - 1 \\
20 & \quad \text{IMPTEU OF SHIPONHAND } \leftarrow \text{IMPTEU OF SHIPONHAND} - 1 \\
21 & \quad \text{IF } \text{IMP}20 \text{ OF SHIPONHAND } > 0 \\
22 & \quad R2 \leftarrow \text{UNIF} \\
23 & \quad \text{IF } R2 < ((\text{IMP}20 \text{F OF SHIPONHAND}) + (\text{IMP}20 \text{ OF SHIPONHAND})) \\
24 & \quad \text{IMP}20 \text{F OF SHIPONHAND } \leftarrow \text{IMP}20 \text{F OF SHIPONHAND} - 1 \\
25 & \quad \text{LOAD OF IMPTT } \leftarrow \text{"F"} \text{ IF (LOAD OF IMPTT } = \text{ "F")} \\
26 & \quad \text{LOAD OF IMPTT } \leftarrow \text{"FMT"} \text{ IF (LOAD OF IMPTT } = \text{ "MT")} \\
27 & \quad \text{GOTO POINT2} \\
28 & \quad \text{END} \\
29 & \quad \text{IF } R2 > ((\text{IMP}20 \text{F OF SHIPONHAND}) + (\text{IMP}20 \text{ OF SHIPONHAND})) \\
30 & \quad \text{IMP}20 \text{MT OF SHIPONHAND } \leftarrow \text{IMP}20 \text{MT OF SHIPONHAND} - 1 \\
31 & \quad \text{LOAD OF IMPTT } \leftarrow \text{"MT"} \text{ IF (LOAD OF IMPTT } = \text{ "MT")} \\
32 & \quad \text{LOAD OF IMPTT } \leftarrow \text{"FMT"} \text{ IF (LOAD OF IMPTT } = \text{ "F")} \\
33 & \quad \text{GOTO POINT2} \\
34 & \quad \text{END} \\
35 & \quad \text{POINT2:} \\
36 & \quad \text{BOX OF IMPTT } \leftarrow 2 \\
37 & \quad \text{TEU OF IMPTT } \leftarrow 2 \\
38 & \quad \text{IMP}20 \text{ OF SHIPONHAND } \leftarrow \text{IMP}20 \text{ OF SHIPONHAND} - 1 \\
39 & \quad \text{IMPTEU OF SHIPONHAND } \leftarrow \text{IMPTEU OF SHIPONHAND} - 1 \\
40 & \quad \text{END} \\
41 & \quad \text{TASK OF IMPTT } \leftarrow \text{"BRINGLOAD"} \\
42 & \quad \text{DESTINATION OF IMPTT } \leftarrow \text{"IMPORTSTACK"} \text{ IF (LOAD OF IMPTT } = \text{ "F")} \text{ V} \\
43 & \quad \text{LOAD OF IMPTT } \leftarrow \text{"FMT"} \text{ IF (LOAD OF IMPTT } = \text{ "MT")} \\
44 & \quad \text{DESTINATION OF IMPTT } \leftarrow \text{"IMPORTSTACK"} \text{ IF (LOAD OF IMPTT } = \text{ "F")} \text{ V} \\
45 & \quad \text{LOAD OF IMPTT } \leftarrow \text{"FMT"} \text{ IF (LOAD OF IMPTT } = \text{ "MT")} \\
46 & \quad \text{RETURN} \\
47 & \quad \text{END} \\
\end{align*}
\]

\([45 - 66]\):

If the random number generated in block \([3 - 5]\) is larger than the fraction of 20 Ft. containers in the total amount of import-containers aboard, then the first (and only) container to be loaded on the transporter will be a 40 Ft. container. To be able to determine whether this container is full or empty, the same technique is used as in the previous blocks. A random number is generated, etc. Again the import-load still on board the ship is updated according to which type of container is allocated. The load facts and attributes of the transporter are also set.
If the quaycrane calls the macro and he is loading the ship (UNLOADING=FALSE), then the macro will start here. This block updates the export-load already aboard the ship when a transporter brings an export-load to the quaycrane. If the transporter has two containers, it can only mean that he is bringing two 20 Ft. containers. If the transporter only has one container, the macro must check the transported TEU to see whether the container is 20 or 40 Ft.. After changing the task of the transporter, the macro is finished.

68 IF UNLOADING = FALSE
69 IF BOX OF EXPTT = 2
70 EXP20 OF SHIPONHAND = EXP20 OF SHIPONHAND - 2
71 EXPTEU OF SHIPONHAND = EXPTEU OF SHIPONHAND - 2
72 GOTO TEUCOUNT
73 END
74 IF BOX OF EXPTT = 1
75 IF TEU OF EXPTT = 1
76 EXP20 OF SHIPONHAND = EXP20 OF SHIPONHAND - 1
77 EXPTEU OF SHIPONHAND = EXPTEU OF SHIPONHAND - 1
78 GOTO TEUCOUNT
79 END
80 IF TEU OF EXPTT = 2
81 EXP40 OF SHIPONHAND = EXP40 OF SHIPONHAND - 1
82 EXPTEU OF SHIPONHAND = EXPTEU OF SHIPONHAND - 2
83 END
84 END
85 TEUCOUNT:
86 TASK OF EXPTT = "GETLOAD"
87 RETURN
88 END
MACRO QUAYLOAD

CALL BY QUAYCRANE

IS THE SHIP UNLOADING?

YES

IS THE FIRST BOX A 20 ft. CONTAINER?

NO

IS THE FIRST BOX A 40 ft. CONTAINER?

YES

IS THE CONTAINER FULL?

NO

IS THIS CONTAINER EMPTY?

YES

LOAD ("F")

ADJUST COUNTERS

LOAD ("HT")

DEFINE DESTINATION

NO

NO

IS THE SECOND BOX A FULL CONTAINER?

NO

IS THE SECOND BOX AN EMPTY CONTAINER?

YES

DEFINE LOAD

ADJUST COUNTERS

DEFINE TASK AND DESTINATION OF TT

RETURN

RETURN
18.12 Description of module TTPROCESS

This module describes the actions of a transporter of the type TT. These transporters remain on the terminal at all times and are part of the terminal's container-handling equipment. In this module, labels have been used to be able to skip through the module so that certain parts may be used for different parts of the TT's cycle.

[3 - 9] :

When a TT is passive and queued in TTAVAILABLE, it can be activated by the TTMASTER. The TT must first check with the macro TTCONTROL as to what he is supposed to do (start the unload- or loadcycle or if he is needed at all). After this he may start to leave the yard. The TT will skip to the label QUITYARD (line 50).

If the TT is not needed his ENDPROCEDURE become TRUE in the macro TTCONTROL and the process will direct the TT to the label parking (line 97).

```
3 STARTUP:
4 CALL TTCONTROL
5 IF ENDPROCEDURE = TRUE
6  N OF TRACTORALLOCATION = TTPERCRANE
7  GOTO PARKING
8 END
9 GOTO QUITYARD
```

TTCONTROL : Macro
Determination of shiftprocedures

ENDPROCEDURE : Logical attribute of class component TRANSPORTER
Indicates whether tractor-trailer must stop driving for ship or not

[11 - 17] :

This is the beginning of a standard TT-cycle. The beginning of the cycle has been chosen at the moment the TT joins the IMPTTQ or EXPTTQ of his WORKSHIP. He must wait to be activated again by one of the quaycranes servicing the ship. A quaycrane will activate the waiting TT from the label DRIVETOCRANE. After activation the TT will drive underneath the quaycrane and wait while being loaded or unloaded.

```
11 BEGINCYCLE:
12 WAIT
13
14 DRIVETOCRANE:
15 ENTER UNDERCRANE OF WORKCRANE
16 WAIT
17```
When the quaycane is finished with the TT, he will activate the TT from the label DRIVETOYARD. As the label indicates, the TT can leave the quay and drive to the yard. The next lines describe where the TT is driving and for how long.

18 DRIVETOYARD:
19 ENTER QUAYTRAFFIC
20 WORK QUAYTIME
21 LEAVE QUAYTRAFFIC
22 ENTER QUAYTOYARD
23 WORK BRIDGETIME
24 LEAVE QUAYTOYARD

Upon arriving at the yard, the TT will ask what his further instructions are by calling the macro TTCONTROL again. If he is to end his cycle, he will skip to the label ENDCYCLE. If, for the time being, he may proceed, he will call the macro DIRECTIONS to determine the position where he is to drive to. When this in the yard is known, he can call the macro DRIVETIMES to determine the time necessary to get there.

26 YARDINSTRUCTIONS:
27 CALL TTCONTROL
28 GOTO ENDCYCLE IF (ENDPROCEDURE = TRUE) ^ (BOX = 0)
29 CALL DIRECTIONS
30 CALL DRIVETIMES

DIRECTIONS : Macro
Determination of stack destination
DRIVETIMES : Macro
Determination of driving times

Now the TT knows everything he needs to know, he can proceed to drive to his destination in the yard. The next lines describe where he drives and for how long. When he has reached the position, he must wait on the yardcrane to help him.

32 YARDACTION :
33 ENTER INTRAFFIC
34 WORK DRIVEIN
35 LEAVE INTRAFFIC
36 ENTER LANE[ROW]
37 WORK INLANEDRIVE
38 ENTER TTQ[ROW]
39 IMPOSITION :
40 WAIT

INTRAFFIC : Set attribute of MAIN
Physical part of yard on which
tractor-trailers drive

DRIVEIN
: Real attribute of TRANSPORTER
Driving time from main traffic lane to stack of destination

LANE[i]
: Queue
Physical part of yard on which tractor-trailers drive

INLANEDRIVE
: Real attribute of TRANSPORTER
Driving time into lane

ttractor-trailer queue for yardcrane

TTQ[i]
: Queue

[42 - 48]:

When the yardcrane has finished with the TT, he will activate the TT from the label QUITLANE and the TT will start to leave the yard. The next lines describe where the TT drives and for how long. If the TT has received the instructions from TTCONTROL to end his procedure, he will skip to the label PARKING in the module (line 97).

42 QUITLANE:
43 WORK OUTLANEDRIVE
44 LEAVE LANE(RGW)
45 GOTO PARKING IF ENDPROCEDURE = TRUE
46 ENTER OUTTRAFFIC
47 WORK DRIVEOUT
48 LEAVE OUTTRAFFIC

OUTLANEDRIVE
: Real attribute of TRANSPORTER
Driving time out of lane

OUTTRAFFIC
: Set attribute of MAIN
Physical part of yard on which tractor-trailers drive

DRIVEOUT
: Real attribute of TRANSPORTER
Driving time from lane back to main traffic lane

[50 - 67]:

The TT starts to leave the yard (when leaving the queue MAINTRAFFIC), but when he reaches the entrance from the quayside, he determines whether he has to go back into the yard to pick up or unload other containers. The instructions for this have been received from TTCONTROL when the TT entered the yard, or are determined by the values of attributes. If the TT has to make a loop in the yard, he will go back to the label YARDACTION. There are two reasons why a TT might have to make a loop in the yard. One reason may be because he brought a mixed load of containers, which could not go into the same stack (line 54-58). Because of the traffic circulation scheme,
the TT must make a loop. The other reason may be because the TT must shift from an import-cycle procedure to an export-cycle procedure (line 59-67).

58 QUIT YARD;
59 ENTER MAINTRAFFIC
60 WORK YARDTIME
61 LEAVE MAINTRAFFIC
62 IF (BOX > 0) "DESTINATION = "IMPEMTYSTACK")
63 CALL DIRECTIONS
64 CALL DRIVETIMES
65 REPEAT FROM YARDACTION
66 END
67 IF SHIFTPROCEDURE = TRUE
68 TASK = "GETLOAD"
69 SHIFTPROCEDURE = FALSE
70 CALL TTCONTROL
71 GOTO ENDCYCLE IF ENDPYCEDURE = TRUE
72 CALL DIRECTIONS
73 CALL DRIVETIMES
74 REPEAT FROM YARDACTION
75 END

MAINTRAFFIC : Set attribute of MAIN
Set in which transporters on the main yard traffic lane are registered

YARDTIME : Real attribute of MAIN
The time needed by a transporter to drive the length of the main traffic lane

DESTINATION : Character attribute of class component TRANSPORTER
Determines which part of the yard the transporter must drive to

SHIFTPROCEDURE : Logical attribute of class component TRANSPORTER
Indicates whether tractor-trailer must shift from import cycle to export cycle

TASK : Character attribute of class component TRANSPORTER
Determines what the transporter is coming for

[69 - 78] :

If the TT does not have to make a loop, he can proceed to drive towards the quay and enter the appropriate queue of his WORKSHIP. The process will start all over again from label begincycle (line 11).
DRIVETOQUAY:
ENTER YARDOYQUAY
WORK BRIDGETIME
LEAVE YARDOYQUAY
ENTER QUAYTRAFFIC
WORK QUAYTIME
LEAVE QUAYTRAFFIC
ENTER IMPTTQ OF WORKSHIP IF (DESTINATION = "IMPORTSTACK")
(DESTINATION = "IMPEMPTYSTACK")
ENTER EXPTTQ OF WORKSHIP IF (DESTINATION = "EXPORTSTACK")
(DESTINATION = "EXPEMPTYSTACK")
REPEAT FROM BEGINCYLE

YARDOYQUAY: Set attribute of MAIN
Set in which tractor-trailers driving from the yard to the quay are registered

BRIDGETIME: Real attribute of MAIN
The time needed by tractor-trailer to drive across the bridge

QUAYTRAFFIC: Set attribute of MAIN
Set in which tractor-trailers driving on the quay are registered

QUAYTIME: Real attribute of class component TRANSPORTER
Driving time of tractor-trailers on the quay area from bridge to berth

[80 - 86]:

This section describes the procedure the TT must take when ending a cycle from the quay. For example, when a crane has finished servicing a ship and a TT is still waiting, the TT will be activated to drive back to the yard. This situation should not arise because TTCONTROL excludes the possibility of such a situation. It is more of a safety measure.

ENDWORK:
ENTER QUAYTRAFFIC
WORK QUAYTIME
LEAVE QUAYTRAFFIC
ENTER QUAYTOYARD
WORK BRIDGETIME
LEAVE QUAYTOYARD

[88 - 95]:

This section describes the ending of the TT's cycle from the entrance to the yard. The TT drives to the first lane, enters it drives to the end.
This section describes the TT becoming passive and the resetting of its attribute values. The TT is now available for reactivation by the TTMASTER.

97 PARKING:
98 ENTER TTAVAILABLE
99 SHIFTPROCEDURE = FALSE
100 DESTINATION = "IMPORTSTACK"
101 TASK = "BRINGLOAD"
102 WORKSHIP = NONE
103 ENDPARAMETER = FALSE
104 ROW = 0
105 BAY = 0
106 PASSIVATE
BEGIN CYCLE:

CALL TITCONTROL

REACTIVATED BY YARD CRANE

DRIVE TO CRANE
ENTER UNDERCRANE
WAIT

REACTIVATED BY YARD CRANE

DRIVE TO THE YARD

CALL TITCONTROL

TT HAS NO LOAD AND IS NOT NEEDED ANYMORE.

YES

CALL DIRECTIONS
CALL DRIVETIME

NO

DRIVE TO CRANE
ENTER UNDERCRANE
WAIT

TT MUST DELIVER A BOX IN THE IMPORT/EMPTY STACK.

YES

CALL DIRECTIONS
CALL DRIVETIME

NO

ENTER INTRAFFIC

THE TT MUST SWITCH FROM UNLOADING TO LOADING?

YES

CALL TITCONTROL

NO

ENTER INTRAFFIC

DRIVE TO APOINTEO STACK
ENTER LANE
WAIT

REACTIVATED BY YARD CRANE

LEAVE LANE OF STACK

THE TT IS NOT NEEDED ANYMORE FOR THE PROCESS

YES

CALL DIRECTIONS
CALL DRIVETIME

NO

DRIVE TO QUAY
ENTER DEPOT
OR EXIT

PASSIVATE

CALL DIRECTIONS
CALL DRIVETIME

DRIVE TO YARD IF STILL WAITING AT QUAY.

CALL DIRECTIONS
CALL DRIVETIME

ENTER FIRST LANE

ENTER INTRAFFIC

ADJUST ATTRIBUTES

TT PROCESS

DRIVE TO YARD IF STILL WAITING AT QUAY.

CALL TITCONTROL

IS THE TT NEEDED FOR THE LOADING PROCESS?

YES

CALL DIRECTIONS
CALL DRIVETIME

NO

CALL TITCONTROL

DRIVE TO QUAY
ENTER DEPOT
OR EXIT

PASSIVATE
18.13 Description of macro TTCONTROL

[3 - 13]:

This part of the macro is used by transporters of the type TT which are unloading a ship. Each TT entering the yard calls this macro. This macro decides if, after the TT has brought his current load to the IMPORT- or IMPEMPTYSTACK, the TT is to continue unloading the ship. If the TT is still needed for unloading the ship, he is joined to the ship's set IMPTTLIST (a set containing all the TT's assigned to unloading the ship) and the macro returns to the TTPROCESS.

If there are already enough TT's assigned to unloading the ship (compared to the amount of containers still to be unloaded), the current TT, after having delivered his current load in the stack, can switch to the loading process. The attribute SHIFTPROCEDURE of the TT will become TRUE and the macro is directed to the label TEST (line 15) in the next block. This is done to decide if the TT is indeed neccessary for the loading process.

3 IF TASK ="BRINGLOAD"
4 E = CEIL(IMPTEU OF WORKSHIP+2)
5 F = LENGTH OF IMPTTLIST OF WORKSHIP
6 IF (F>E) \( > (IMPTEU OF WORKSHIP+2)\)
7 SHIFTPROCEDURE \( TRUE \)
8 GOTO TEST
9 END
10 JOIN THIS TRANSPORTER TO IMPTTLIST OF WORKSHIP
11 RETURN
12 END
13

[14 - 24]:

The TT is now in the process of loading the ship. Before entering the yard the TT calls the macro, which checks if the TT is still needed to bring containers to the ship. This is done by comparing the amount of TT's assigned to loading the ship, but which have not yet picked up a load, to the amount of TT's necessary to move the remaining export in the yard to the ship. TT's which have been assigned to loading the ship, but are on their way to picking up the load, are registered in the set FREQLIST if the load is a full export-container, and in the set MTREQLIST if the load is an empty export-container. If the TT is not needed, its attribute ENDPROCEDURE becomes TRUE. The macro skips to the label MACROEND (line 43) and returns to the TTPROCESS.

TT's for which the attribute SHIFTPROCEDURE is TRUE also enter this block to see if they will need to switch to loading the ship after having
delivered their import-containers at the IMPORT- or IMPEMPTYSTACK. The macro will skip to the label MACROEND so the TT's can proceed in delivering their containers. If they do not need to switch to loading the ship, their attribute ENDPROCEDURE will have gotten the value TRUE.

14 IF TASK = "GETLOAD"
15 TEST:
16 E ≤ (CEIL(REQ20F OF WORKSHIP+2) + REQ40F OF WORKSHIP)
17 F ≤ LENGTH OF FREQLIST OF WORKSHIP
18 G ≤ (CEIL(REQ20MT OF WORKSHIP+2) + REQ40MT OF WORKSHIP)
19 H ≤ LENGTH OF MTREQLIST OF WORKSHIP
20 IF (F)E ∧ (H≤G)
21 ENDPROCEDURE = TRUE
22 GOTO MACROEND
23 END
24 GOTO MACROEND IF (SHIFTPROCEDURE = TRUE)

[25 - 45] :

TT's only enter this block if they are needed for the loading process. In this block the macro defines to which stack the TT is sent. When both full and empty containers have to be brought to the ship, a random number decides whether the TT goes to the EXPORTSTACK or the EXPEMPTYSTACK. The TT is joined respectively to the set FREQLIST or MTREQLIST. When either only full or only empty containers are left in the stacking-area of the ship, the TT is directed to the correct stack and joined to the correct set. Hereafter the macro comes to the label MACROEND (line 43), and returns to the TTPROCESS.

25 IF (F>E) ∧ (H≥G)
26 R3 ≤ UNIF
27 24 ≤ (REQ20F OF WORKSHIP + RED40F OF WORKSHIP) + (REQ20MT OF WORKSHIP + RED40MT OF WORKSHIP)
28 DESTINATION = "EXPORTSTACK" IF R3 ≤ 24
29 DESTINATION = "EXPEMPTYSTACK" IF R3 > 24
30 JOIN THIS TRANSPORTER TO FREQLIST OF WORKSHIP IF DESTINATION="EXPORTSTACK"
31 JOIN THIS TRANSPORTER TO MTREQLIST OF WORKSHIP IF DESTINATION="EXPEMPTYSTACK"
32 GOTO MACROEND
33 END
34 IF (F≤E)
35 DESTINATION = "EXPORTSTACK"
36 JOIN THIS TRANSPORTER TO FREQLIST OF WORKSHIP
37 GOTO MACROEND
38 END
39 IF (H<G)
40 DESTINATION = "EXPEMPTYSTACK"
41 JOIN THIS TRANSPORTER TO MTREQLIST OF WORKSHIP
42 END
43 END
44 MACROEND:
45 RETURN
CALL BY TIT PROCESS

IS THE TT IN THE UNLOADING PROCESS?

YES

IS THE TT IN THE LOADING PROCESS?

YES

THIS TT IS NOT NEEDED ANYMORE FOR THE UNLOADING PROCESS

NO

NO

YES

STOP WORKING FOR THIS SHIP

GOTO MACROEND IF SHIFTPROCEDURE = TRUE

MORE TT'S ARE NEEDED TO BRING CONTAINERS TO SHIP

YES

DESTINATION <- EXPORTSTACK OR EXPEN PTSTACK

NO

IT'S ARE NEEDED TO BRING FULL CONTAINERS TO SHIP

YES

DESTINATION <- EXPORTSTACK

NO

IT'S ARE NEEDED TO BRING EMPTY CONTAINERS TO SHIP

YES

DESTINATION <- EXPEN PTSTACK

NO

RETURN

RETURN
18.14 Description of macro DIRECTIONS

[3 - 24]:

The container-yard consists of 5 stacking-areas; 2 for import-containers, 2 for export-containers and 1 for empty containers (import and export). This macro decides to which area the TT will drive: to row 1 or 2 if the destination of the TT is EXPORTSTACK, to row 3 or 4 if the destination of the TT is IMPORTSTACK and to row 5 if the destination of the TT is IMPEMPTYSTACK or EXPEMPTYSTACK.

The TT will also get a reference-point in the stackrow where he must deliver his load (BAY). As each TT knows for which ship he is working and each ship is appointed a special area in all the stacks, the macro gives the TT a random value for BAY in that appointed area. Then the macro returns to the TTPROCESS.

3 R4 < UNIF
4 R10 < UNIF
5 IF DESTINATION = "EXPORTSTACK"
6 ROW < 1 IF R4 ≤ 0.5
7 ROW < 2 IF R4 > 0.5
8 BAY = (CEIL(R10*EXPBAYF OF WORKSHIP) + LBEXPF OF WORKSHIP)
9 END
10 IF DESTINATION = "IMPORTSTACK"
11 ROW < 3 IF R4 ≤ 0.5
12 ROW < 4 IF R4 > 0.5
13 BAY = (CEIL(R10*IMPBAYF OF WORKSHIP) + LBIMPF OF WORKSHIP)
14 END
15 IF DESTINATION = "IMPEMPTYSTACK"
16 ROW < 5
17 BAY = (CEIL(R10*IMPBAYMT OF WORKSHIP) + LBIMPMT OF WORKSHIP)
18 END
19 IF DESTINATION = "EXPEMPTYSTACK"
20 ROW < 5
21 BAY = (CEIL(R10*EXPBAYMT OF WORKSHIP) + LBEXPMT OF WORKSHIP)
22 END
23 RETURN
24
18.15 Description of macro DRIVETIMES

[3 - 9]:

After the ROW and BAY of the transporter have been determined in the macro DIRECTIONS, the transporter must know the drivingtimes in the yard. This is achieved in this macro.
The time to drive from the MAINTRAFFIC-lane to the entrance of the appointed stacking-area-lane, is called DRIVEIN and takes place on the INTRAFFIC-lane.
The time to drive to the appointed bay in the appointed lane, is called INLANEDRIVE and takes place on the LANE OF YCRANE[ROW].
The time to leave LANE OF YCRANE[ROW] after being served by the yardcrane, is called OUTLANEDRIVE.
The time to drive back to the entrance of MAINTRAFFIC from where the transporter can leave the yard, is called DRIVEOUT and takes place on OUTTRAFFIC (see figure 18.15.1 for the driving pattern).
After the calculation of the drivetimes in the yard, the macro returns to the TTPROCESS or TRUCKPROCESS.

1 DETERMINATION OF THE LENGTH OF DRIVETIMES
2 3 DRIVEIN ≤ (CEIL((ROW*30)+YARDSPEED)) SECONDS
4 INLANEDRIVE ≤ (CEIL((BAY*6.4)+YARDSPEED)) SECONDS
5 OUTLANEDRIVE ≤ (CEIL((420-((BAY*6.4)))+YARDSPEED)) SECONDS
6 DRIVEOUT ≤ DRIVEIN
7 YARDTIME ≤ (420+YARDSPEED) SECONDS
8 RETURN
9
yardlayout 1

to quay

export
import
empties

parking area

to hinterland

FIG. 18.15.1
18.16 Description of module YARDCRANEPRECESS

[3 - 14] :

The yardcrane will wait as long as there is no transporter to serve. When a transporter arrives (in the TTQ or TRUCKQ) the idletime of the yardcrane is registered and the yardcrane will call the macro STRATEGY to define the transporter to be served, the 'serveunit'. The crane will then call the macro YARDLOAD to adjust all the counters involved in this process, roll to the location (ROLLTIME) of the transporter and start working.

If the TRUCK or TT is bringing containers, then the yardcrane will follow the procedure as outlined in section [16 - 32].
If the TRUCK or TT is coming to get a container, then the yardcrane will follow the procedure outlined in section [33 - 48].

3 ROLLLOCATION:
4 PAST < NOW
5 ENTER NOWORK
6 WAIT WHILE (TTQ[CRANENR] IS EMPTY)^(TRUCKQ[CRANENR] IS EMPTY)
7 LEAVE NOWORK
8 IDLETIME = IDLETIME + (NOW - PAST)
9 WAITIME = WAITIME + (NOW - PAST)
10 CALL STRATEGY
11 CALL YARDLOAD
12 LOCATION = BAY OF SERVEUNIT x 6.4
13 ROLLTIME = (CEIL(ABS((LOCATION - PREVLOCATION) / RTGCSPEED))))SECONDS
14 WORK ROLLTIME
15

PAST : Real attribute of YARDCRANE
Help variable to determine the time spent idle

NOWORK : Set attribute of MAIN
Set in which idle yardcranes are registered

CRANENR : Integer attribute of YARDCRANE
Enables identification of yardcranes

TRUCKQ[i] : Queue
Truck queue for yardcrane

IDLETIME : Real attribute of YARDCRANE
Time spent idle by the yardcrane

LOCATION : Integer attribute of YARDCRANE
This is the new position of the yardcrane along the stack

YARDLOAD : Macro
Determines the load placed on or picked up from transporters

ROLLTIME : Real attribute of YARDCRANE
Time needed to relocate along stack
This block describes how the yardcrane lifts off containers from the transporters. The TASK of the transporters must be BRINGLOAD. Each transporter will have at least one container. The crane starts by taking the first container from the transporter (WORK LIFTOFFTIME). If the transporter has two containers, which have both been destined to the same stack by the macro YARDLOAD, then the yardcrane will stack the first container (WORK CATROLLTIME) and then lift off the second container. The transporter is now relieved of its load and activated to drive away. The yardcrane process is directed to the label NEWLOC in section [51 - 53].

16 IF TASK OF SERVEUNIT = "BRINGLOAD"
17 WORK LIFTOFFTIME
18 IF (BOX OF SERVEUNIT = 2) AND (LOAD OF SERVEUNIT = "O")
19 WORK CATROLLTIME(CRANENR)
20 WORK LIFTOFFTIME
21 END
22 BOX OF SERVEUNIT < 0 IF LOAD OF SERVEUNIT = "O"
23 IF TYPE OF SERVEUNIT = "TRUCK"
24 ACTIVATE SERVEUNIT FROM BYETRUCK IN TRUCKPROCESS
25 END
26 IF TYPE OF SERVEUNIT = "TT"
27 ACTIVATE SERVEUNIT FROM QUITLANE IN TTPROCESS
28 END
29 WORK CATROLLTIME(CRANENR)
30 GOTO NEWLOC
31 END
32

SERVEUNIT : Reference to class component
TRANSPORTER
The transporter that is served by the yardcrane
CATROLLTIME[D] : Randomstream
Time needed to place or retrieve a container by yardcrane

The yardcrane will follow the procedure in this block if the TASK of the transporter under the yardcrane is now GETLOAD. The yardcrane will lift off the first container for the transporter from the stack (WORK CATROLLTIME) and place it on the transporter (WORK LIFTOFFTIME). If macro YARDLOAD has determined that the transporter will transport two containers, then the yardcrane will work another CATROLLTIME and LIFTOFFTIME. After this the transporter will be activated to drive away. Activation takes place in the process corresponding with the type of the transporter. The transporter is also removed from the queue MYCRANEQ of the yardcrane.
33 IF TASK OF SERVEUNIT = "GETLOAD"
34 AGAINWORK:
35 WORK CATROLLTIME[CRANENR]
36 WORK LIFTONTIME
37 IF BOX OF SERVEUNIT = 2
38 WORK CATROLLTIME[CRANENR]
39 WORK LIFTONTIME
40 END
41 IF TYPE OF SERVEUNIT = "TRUCK"
42 ACTIVATE SERVEUNIT FROM BYETRUCK IN TRUCKPROCESS
43 END
44 IF TYPE OF SERVEUNIT = "TT"
45 ACTIVATE SERVEUNIT FROM QUITLANE IN TPROCESS
46 END
47 END
48

[49 - 51]:

The yardcrane updates its position in the stack and is directed to the start of the process; the label ROLLTOLOCATION.

49 NEWLOC:
50 PREVLOCATION < LOCATION
51 REPEAT FROM ROLLTOLOCATION
52
18.17 Description of macro TTPRIORITY

This macro is called upon by the yardcrane when the yardcrane needs to know which transporter to serve. This macro will give transporters of the type TT priority over transporters of the type TRUCK.

[3 - 7] :

The macro will first look into the yardcrane's queue of waiting transporters of the type TT. If this queue is not empty, the first TT will become the transporter to serve.

3 IF TTQ[CRANENR] IS NOT EMPTY
4 SERVEUNIT = FIRST OF TTQ[CRANENR]
5 REMOVE SERVEUNIT FROM TTQ[CRANENR]
6 RETURN
7 END

[8 - 12] :

Only if there are no TT's waiting will the macro reach this block. The transporter to serve will become the first TRUCK in the yardcrane's queue.

8 IF TRUCKQ[CRANENR] IS NOT EMPTY
9 SERVEUNIT = FIRST OF TRUCKQ[CRANENR]
10 REMOVE SERVEUNIT FROM TRUCKQ[CRANENR]
11 RETURN
12 END
18.18 Description of macro NOPRORITY

This macro is called upon by the yardcrane to determine which transporter to serve first. This macro will give priority to the transporter that has arrived first.

[3 - 7] :

The macro looks into the yardcrane's queue of waiting transporters of the type TT. If this queue is empty he will know that a truck is waiting by the yardcrane because the yardcrane will only activate him when a transporter needs to be served. The transporter to serve will be the first truck in the queue of waiting trucks. When the transporter to serve has been determined the macro will return to the yardcrane.

3 IF TTQ(CRANENR) IS EMPTY
4 SERVEUNIT ← FIRST OF TRUCKQ(CRANENR)
5 REMOVE SERVEUNIT FROM TRUCKQ(CRANENR)
6 RETURN
7 END

[8 - 12] :

This block works exactly as the previous one, except this time the macro looks into the queue of waiting transporters of the type TRUCK.

8 IF TRUCKQ(CRANENR) IS EMPTY
9 SERVEUNIT ← FIRST OF TRUCKQ(CRANENR)
10 REMOVE SERVEUNIT FROM TRUCKQ(CRANENR)
11 RETURN
12 END

[13 - 24] :

If neither the queue of waiting trucks nor the queue of waiting tt's is empty, the determination of the transporter to serve becomes more difficult. The macro will look at the time the first transporter of each queue arrived, the queuetime, and compare them. The transporter that arrived first will be the transporter that will be served by the yardcrane.
73

13 Z2 ← QUEUETIME OF FIRST OF TTQ[CRANENR] IN TTQ[CRANENR]
14 Z3 ← QUEUETIME OF FIRST OF TRUCK[CRANENR] IN TRUCK[CRANENR]
15 IF Z2 ≤ Z3
16  SERVEUNIT ← FIRST OF TTQ[CRANENR]
17  REMOVE SERVEUNIT FROM TTQ[CRANENR]
18  RETURN
19 END
20 IF Z2 > Z3
21  SERVEUNIT ← FIRST OF TRUCK[CRANENR]
22  REMOVE SERVEUNIT FROM TRUCK[CRANENR]
23  RETURN
24 END

Z1,Z2 : Real attribute of MAIN
18.19 Description of macro YARDLOAD

This macro determines the loading and unloading of transporters by the yardcranes, keeping detailed track of the YARDSTACK volume, the flow of export-containers to the ships and the load of the transporters themselves. Because of its general application, the macro can be used by any kind of yardcrane.

When the yardcrane calls this macro, and the TASK of the transporter being served is BRINGLOAD, then the macro will stay in this block. If the transporter has two containers (only transporters of the type TT can have two containers), then the macro determines if the containers are to be unloaded in the same stack or not. If the containers have to be stacked in different stacks, the transporter receives a new destination. In this model the transporters always first go to the IMPORTSTACK (full boxes) and secondly to the EMPTYSTACK.

The volume of the stack and the transporter load facts are also updated here.

3 IF TASK OF SERVEUNIT = "BRINGLOAD"
4 IF (LOAD OF SERVEUNIT = "F") OR (LOAD OF SERVEUNIT = "MT")
5 YARDSTACK = YARDSTACK + (TEU OF SERVEUNIT)
6 TEU OF SERVEUNIT = 0
7 LOAD OF SERVEUNIT = "O"
8 RETURN
9 END
10 IF LOAD OF SERVEUNIT = "FMT"
11 YARDSTACK = YARDSTACK + 1
12 TEU OF SERVEUNIT = 1
13 LOAD OF SERVEUNIT = "MT"
14 BOX OF SERVEUNIT = 1
15 DESTINATION OF SERVEUNIT = "IMPORTSTACK"
16 RETURN
17 END
18 END
19

EXPSHIP : Reference to class component
SHIP
Ship that transporter (which is being served by the yardcrane) is driving for

[20 - 77] :

If the TASK of the transporter is to GETLOAD, and the type TT, then the macro will stay in this block. If the type is TRUCK however, the macro will skip to block [79 - 86]. First the macro looks where the TT is waiting to be served; the EXPORTSTACK (only full containers)
or the EXPEMPTYSTACK (only empty containers).

Using the same technique explained in the description of the macro QUAYLOAD, a first container is generated. If this first container is a 20 Ft. container, the macro will stay in block [31 - 59].

The macro will give a TT a full container if he is waiting by an EXPORTSTACK and an empty container if he is waiting by an EXPEMPTYSTACK. If there are more 20 Ft. containers in the same stack that have to be transported to the same ship, the TT will get a second container. After adjusting the counters, the macro returns to the YARDCRANEPROCESS.

If the first container is a 40 Ft. container, the macro will skip to block [60 - 77]. The TT will only get one container now, because it can only transport two TEU. The macro looks where the TT is waiting, adjusts the appropriate counters of the ship and the yardstack and returns to the yardcraneprocess.

20 IF TASK OF SERVEUNIT = "GETLOAD"
21 IF TYPE OF SERVEUNIT = "TT"
22 IF DESTINATION OF SERVEUNIT = "EXPORTSTACK"
23 EXPSHIP = WORKSHIP OF SERVEUNIT
24 Z1=((REQ2OF EXPSPH)+((REQ2OF EXPSHIP+REQ4OF EXPSHIP))
25 END
26 IF DESTINATION OF SERVEUNIT = "EXPEMPTYSTACK"
27 EXPSHIP = WORKSHIP OF SERVEUNIT
28 Z1=((REQ2MT OF EXPSHIP)+((REQ2MT OF EXPSHIP+REQ4MT OF EXPSHIP))
29 END
30 R5 = UNIF
31 IF R5 < Z1
32 TEU OF SERVEUNIT = 1
33 BOX OF SERVEUNIT < 1
34 IF DESTINATION OF SERVEUNIT = "EXPORTSTACK"
35 REQ2OF EXPSPH = REQ2OF EXPSPH - 1
36 LOAD OF SERVEUNIT = "F"
37 IF REQ2OF EXPSPH > 0
38 TEU OF SERVEUNIT = 2
39 BOX OF SERVEUNIT = 2
40 REQ2OF EXPSPH = REQ2OF EXPSPH - 1
41 END
42 REMOVE SERVEUNIT FROM FREQLIST OF EXPSPH
43 GOTO POINT4
44 END
45 IF DESTINATION OF SERVEUNIT = "EXPEMPTYSTACK"
46 REQ2MT OF EXPSPH = REQ2MT OF EXPSPH - 1
47 LOAD OF SERVEUNIT = "MT"
48 IF REQ2MT OF EXPSPH > 0
49 TEU OF SERVEUNIT = 2
50 BOX OF SERVEUNIT = 2
51 REQ2MT OF EXPSPH = REQ2MT OF EXPSPH - 1
52 END
53 REMOVE SERVEUNIT FROM MTREQLIST OF EXPSPH
54 END
55 POINT4:
56 REQ2OF EXPSPH = REQ2OF EXPSPH - (BOX OF SERVEUNIT)
57 YARDSTACK = YARDSTACK - (TEU OF SERVEUNIT)
58 RETURN
59 END
60 IF R5 > Z1
61 TEU OF SERVEUNIT = 2
62 BOX OF SERVEUNIT = 1
63 IF DESTINATION OF SERVEUNIT = "EXPORTSTACK"
The macro gives transporters of the type TRUCK only one container; a full container if the TRUCK is waiting by the IMPORTSTACK, and an empty container if the TRUCK is waiting in the IMPEMPTYSTACK. The TEU of the container has been determined in the IMPORTGENPROCESS of the truck. The YARDSTACK volume is updated and the macro returns to the YARDCRANEPROCESS.
18.20 Description of module IMPORTGENPROCESS

[3 - 21] :

For every ship that visits the terminal an import-truck-generator is created. This generator is activated with a one day delay, at the moment the ship berths. The generation of transporters of the type TRUCK is spread out over 5 days. The first day 20%, second day 30%, third day 30%, fourth day 10% and the final day 10% of the total amount of trucks is generated.

Depending on which day it is, a number of new trucks are generated with a, for that specific day, constant interarrival-time (This has been determined in block [3 - 21]). Each truck will get the task GETLOAD in order to remove the ship's imported containers from the stacks. A truck can only carry one container, and the TEU
of the container he is to pick up is defined here. The technique used to define this is the same as used in the macro YARDLOAD. Also whether the container is full or empty, is determined here.

The truck is activated in its TRUCKPROCESS, and the generator waits IMPORTPAUSE long before repeating the generation of a truck. When all the trucks for one day (NRTRANS) have been generated, the generator will come into the next block.

```
22 PICKUP :
23 IMPORTPAUSE ≤ (24 + TRANSNR) HOURS
24 FOR K = 1 TO TRANSNR
25 IMPTRANSPORTER = NEW TRANSPORTER
26 TYPE OF IMPTRANSPORTER = "TRUCK"
27 TASK OF IMPTRANSPORTER = "GETLOAD"
28 WORKSHOP OF IMPTRANSPORTER = IMPJOB
29 LOAD OF IMPTRANSPORTER = "0"
30 BOX OF IMPTRANSPORTER = 0
31 TEU OF IMPTRANSPORTER = 0
32 RB < UNIF
33 IF RB ≤ (HIN20 + HIN20 + HIN40)
34 TEU OF IMPTRANSPORTER = 1
35 R9 < UNIF
36 IF R9 ≤ (HIN20 + HIN20)
37 HIN20 ≤ HIN20 - 1
38 HIN20F ≤ HIN20F - 1
39 DESTINATION OF IMPTRANSPORTER = "IMPORTSTACK"
40 GOTO POINT6
41 END
42 IF R9 > (HIN20 + HIN20)
43 HIN20 ≤ HIN20 - 1
44 HIN20MT ≤ HIN20MT - 1
45 DESTINATION OF IMPTRANSPORTER = "IMPEMPTYSTACK"
46 GOTO POINT6
47 END
48 END
49 IF RB > (HIN20 + HIN20 + HIN40)
50 TEU OF IMPTRANSPORTER = 2
51 R9 < UNIF
52 IF R9 ≤ (HIN40F + HIN40)
53 HIN40 ≤ HIN40 - 1
54 HIN40F ≤ HIN40F - 1
55 DESTINATION OF IMPTRANSPORTER = "IMPORTSTACK"
56 GOTO POINT6
57 END
58 IF R9 > (HIN40F + HIN40)
59 HIN40 ≤ HIN40 - 1
60 HIN40MT ≤ HIN40MT - 1
61 DESTINATION OF IMPTRANSPORTER = "IMPEMPTYSTACK"
62 GOTO POINT6
63 END
64 END
65 POINT6:
66 ACTIVATE IMPTRANSPORTER FROM TRUCKARRIVAL IN TRUCKPROCESS
67 WAIT IMPORTPAUSE
68 END
```
Parameter

[69 - 76] :

The counters of the IMPORTGENERATOR are reset and the generator is directed to the following day through a label in block [3 - 21], or terminated.

69 \( K \leq 0 \)
70 \( L \leq L + 1 \)
71 REPEAT FROM 2DAY IF \( L = 1 \)
72 REPEAT FROM 3DAY IF \( L = 2 \)
73 REPEAT FROM 4DAY IF \( L = 3 \)
74 REPEAT FROM 5DAY IF \( L = 4 \)
75 TERMINATE IF \( L = 5 \)
76
18.21 Description of module EXPORTGENPROCESS

[3 - 21] :

For each ship that is generated an export-truck-generator is created to supply the export-containers for the ship.
Like the IMPORTGENPROCESS the EXPORTGENPROCESS is also spread out over 5 days. Each day has his own label and amount of trucks to be generated on that day (TRUCKNR). The first day 5%, the second day 5%, third day 10%, fourth day 20% and the last day 60% of the total amount of trucks (NRTRUCKS) is generated.

```
3 EXPORT:
4 I <= 0
5 J <= 0
6 DAYEXP = FLOOR (NRTRUCKS / 20)
7 DAY1:
8 TRUCKNR = (1 * DAYEXP)
9 GOTO DISPATCH
10 DAY2:
11 TRUCKNR = (1 * DAYEXP)
12 GOTO DISPATCH
13 DAY3:
14 TRUCKNR = (2 * DAYEXP)
15 GOTO DISPATCH
16 DAY4:
17 TRUCKNR = (4 * DAYEXP)
18 GOTO DISPATCH
19 DAY5:
20 TRUCKNR = (NRTRUCKS - (8 * DAYEXP))
21 GOTO DISPATCH
22
```

I : Integer attribute of class component EXPORTGEN
Parameter
J : Integer attribute of class component EXPORTGEN
Parameter
NRTRUCKS : Integer attribute of class component EXPORTGEN
Total number of trucks to generate
DAYEXP : Integer attribute of class component EXPORTGEN
Fraction of total number of trucks
TRUCKNR : Integer attribute of class component EXPORTGEN
Number of trucks arriving on a certain day

[23 - 71] :

Each day a number of trucks with the task BRINGLOAD is generated. The truck have only one-container. The generator determines whether this is a 20 or a 40 Ft., full or empty container. The destination of the truck is also determined. With a full container the truck goes to the
exportstack and with an empty container to the expemptystack.
The truck with load is activated and the generator waits EXPORTPAUSE long before generating the next truck. When all the trucks for the day have been generated, the generator goes to the next block.

24 EXPORTPAUSE < (24 + TRUCKNR) HOURS
25 FOR I = 1 TO TRUCKNR
26 EXPTRANSPORTER = NEW TRANSPORTER
27 WORKSHIP OF EXPTRANSPORTER = EXPJOB
28 TYPE OF EXPTRANSPORTER = "TRUCK"
29 TASK OF EXPTRANSPORTER = "BRINGLOAD"
30 BOX OF EXPTRANSPORTER = 1
31 R6 = UNIF
32 IF R6 < (GEN20+(GEN20 + GEN40))
33 TEU OF EXPTRANSPORTER = 1
34 R7 = UNIF
35 IF R7 < (GEN20F+GEN20)
36 GEN20 = GEN20 - 1
37 GEN20F = GEN20F - 1
38 DESTINATION OF EXPTRANSPORTER = "EXPORTSTACK"
39 LOAD OF EXPTRANSPORTER = "F"
40 GOTO POINTS
41 END
42 IF R7 > (GEN20F+GEN20)
43 GEN20 = GEN20 - 1
44 GEN20MT = GEN20MT - 1
45 DESTINATION OF EXPTRANSPORTER = "EXPEMPTYSTACK"
46 LOAD OF EXPTRANSPORTER = "MT"
47 GOTO POINTS
48 END
49 END
50 IF R6 > (GEN20+(GEN20 + GEN40))
51 TEU OF EXPTRANSPORTER = 2
52 R7 = UNIF
53 IF R7 < (GEN40F+GEN40)
54 GEN40 = GEN40 - 1
55 GEN40F = GEN40F - 1
56 DESTINATION OF EXPTRANSPORTER = "EXPORTSTACK"
57 LOAD OF EXPTRANSPORTER = "F"
58 GOTO POINTS
59 END
60 IF R7 > (GEN40F+GEN40)
61 GEN40 = GEN40 - 1
62 GEN40MT = GEN40MT - 1
63 DESTINATION OF EXPTRANSPORTER = "EXPEMPTYSTACK"
64 LOAD OF EXPTRANSPORTER = "MT"
65 GOTO POINTS
66 END
67 END
68 POINTS:
69 ACTIVATE EXPTRANSPORTER FROM TRUCKARRIVAL IN TRUCKPROCESS
70 WAIT EXPORTPAUSE
71 END

EXPORTPAUSE : Real attribute of EXPORTGEN
Time between truck arrivals

EXPTRANSPORTER : Reference to class component TRANSPORTER
Newly generated export truck

R6 : Real attribute of MAIN
Parameter

R7 : Real attribute of MAIN
Parameter

[72 - 78] :
The counters of the EXPORTGENERATOR are reset and
the generator is directed to the next day's label in block [3 - 22] or, when the current day is the last, terminated.

72 I = 0
73 J = J + 1
74 REPEAT FROM DAY2 IF J = 1
75 REPEAT FROM DAY3 IF J = 2
76 REPEAT FROM DAY4 IF J = 3
77 REPEAT FROM DAY5 IF J = 4
78 TERMINATE IF J = 5
ACTIVATOBY EXPORTGEN

PROCESS

DEFINE TRUCKNR. FOR THIS DAY

FOR I < TRUCKNR

GENERATE TRUCK TO BRING BOXES TO THE YARD

DEFINE ATTRIBUTES

LOAD IS A 20 FT. BOX

BOX IS FULL

ADJUST COUNTERS DESTINATION (-EXPORTSTACK)

ADCJUST COUNTERS DESTINATION (-EXPORTSTACK)

LOAD IS A 40 FT. BOX

BOX IS FULL

ADJUST COUNTERS DESTINATION (-EXPORTSTACK)

ADJUST COUNTERS DESTINATION (-EXPORTSTACK)

ACTIVATE TRUCK

WAIT EXPORTPAUSE

GOTO NEXTDAY

IF DAYNR. < 5

TERMINATE

NO

YES

NO

YES

NO

YES

NO

YES
Transporters of the type TRUCK are generated by the IMPORTGENERATOR and the EXPORTGENERATOR. A truck arrives at, and enters the YARD. He must ask directions on where to go, so he calls the macro DIRECTIONS. Now that he knows where he must drive to, he can call macro DRIVETIMES to determine the time it will take to drive there. He starts to drive and enters the following queue's in successive order: MAINTRAFFIC, INTRAFFIC, LANE[ROW] (the lane he is directed to by the macro DIRECTIONS) and TRUCKQ[ROW] (the servicequeue of the yardcrane in that lane). Once he has arrived here, he will wait until activated by the yardcrane. The yardcrane will activate him when he has been served.

YARD : Queue
Physical area of the container yard

PASSGATE : Integer attribute of MAIN
Number of trucks entering yard

The truck has been served by the yardcrane and can drive away through the queue's LANE[ROW] and OUTTRAFFIC. The truck leaves the YARD and is terminated.
ACTIVATED BY IMPORT/GENERATOR OR EXPORT/GENERATOR

ENTER YARD

CALL DIRECTIONS

CALL DRIVETIMES

DRIVE TO THE APPOINTED STACKROW AND ENTER WAITINGROW

WAIT

LEAVE STACKROW DRIVE TO THE EXIT AND LEAVE YARD

TERMINATE

TRUCKPROCESS

REACTIVATED BY YARDCRANE
18.23 Description of macro INPUTBLOCK

[3 - 14] :

In this macroblock the input which is normally determined by the input-section is read in from the inputfile SITUATION. This macro is called upon by the MAIN module, but only if the user does not wish to specify the input for the simulation run.

3 DATE < "NOT KNOWN"
4 PRIORITY < "y"
5 RUNTIME < READ FROM SITUATION
6 NRCRANES < READ FROM SITUATION
7 NRRTS < READ FROM SITUATION
8 TTPERCRANE < READ FROM SITUATION
9 NRTGCS < READ FROM SITUATION
10 YARDSPEED < \((1000 + 3600) \times \) READ FROM SITUATION
11 BRIDGESPEED < \((1000 + 3600) \times \) READ FROM SITUATION
12 QUAYSPEED < \((1000 + 3600) \times \) READ FROM SITUATION
13 RTGCSPEED < \((1000 + 3600) \times \) READ FROM SITUATION
14 RETURN
18.24 Description of macro OUTPUTBLOCK

[1 - 20]:

In these lines of the macro OUTPUTBLOCK the used input-values are written down. These can be user defined (via the screen) or standard values (via the macro INPUTBLOCK).

1 WRITE "MACRO FOR PRINTING RESULTS" TO RESULTS WITH IMAGE
2 WRITE "SIMULATION RUN DONE WITH THE FOLLOWING VALUES" TO RESULTS WITH IMAGE
3 WRITE "*****************" TO RESULTS WITH IMAGE
4 IF "v" = PRIORITY
5 WRITE "TRACTOR-TRAILERS HAVE PRIORITY IN THE YARD" TO RESULTS WITH IMAGE
6 END
7 IF "n" = PRIORITY
8 WRITE "TRACTOR-TRAILERS HAVE NO PRIORITY IN THE YARD" TO RESULTS WITH IMAGE
9 END
10 WRITE "THE NUMBER OF QUAYCRANES USED =";NRQuaycranes TO RESULTS WITH IMAGE
11 WRITE "THE NUMBER OF TRACTOR-TRAILERS PER QUAYCRANE =";TTpercrane TO RESULTS WITH IMAGE
12 WRITE "THE NUMBER OF YARDCRANES USED =";NRyardcranes TO RESULTS WITH IMAGE
13 WRITE "THE MAX. SPEED IN THE YARD =";Yardspeed x 3.6;"KM\HR" TO RESULTS WITH IMAGE
14 WRITE "THE MAX. SPEED ON THE BRIDGE =";BridgeSpeed x 3.6;"KM\HR" TO RESULTS WITH IMAGE
15 WRITE "THE MAX. SPEED ON THE QUAY =";Quayspeed x 3.6;"KM\HR" TO RESULTS WITH IMAGE
16 WRITE "THE MAX. RELOCATION SPEED OF THE YARDCRANES =";RTGcspeed x 3.6;"KM\HR" TO RESULTS WITH IMAGE
17 WRITE "*****************" TO RESULTS WITH IMAGE
18 WRITE "RESULTS OF SIMULATION RUN AFTER";runtime;"HRS" TO RESULTS WITH IMAGE
19 WRITE "*****************" TO RESULTS WITH IMAGE
20 WRITE "***************" TO RESULTS WITH IMAGE
21

[22 - 39]:

For each yardcrane the percentage of time spent idle is calculated and written down.
'22 FOR M = 1 TO NRRC
23   WRITE "YARDCRANE":M TO RESULTS WITH IMAGE "x"x
24   IF YCRANEM BELONGS TO NOWORK
25     IDLETIME OF YCRANE(M) = IDLETIME OF YCRANE(M)+ (NOW-PAST OF YCRANE(M))
26   END
27   IDLEPERC OF YCRANE(M) = CEIL(100 * (IDLETIME OF YCRANE(M)+NOW))
28   WRITE "PERCENTAGE OF TIME SPENT IDLE = ";IDLEPERC OF YCRANE(M);"%" TO RESULTS WITH IMAGE "x"x
29   END
30 WRITE "THE TOTAL AMOUNT OF TEU HANDLED = ";TOTTEU TO RESULTS WITH IMAGE "x"x

31 WRITE "STORE FILE ABBREVIATIONS ARE AS FOLLOWS" TO RESULTS WITH IMAGE "x"x
32 WRITE "V = PERCENTAGE OF QUAYCRANE TIME SPENT IDLE WHILE ALLOCATED TO A SHIP" TO RESULTS WITH IMAGE "x"x
33 WRITE "T = TURNAROUND TIME OF A SHIP" TO RESULTS WITH IMAGE "x"x
34 WRITE "END OF DATA "x"x" TO RESULTS WITH IMAGE "x"x
35 WRITE "DATE : ";DATE TO RESULTS WITH IMAGE "x"x
36 RETURN
19.1 Introduction

The following modules and macro's of SUREND 2 differ importantly from their counterparts in SUREND 1:

- MAIN
- STACKALLOCATION
- TTPROCESS
- TTCONTROL
- SHIFTCONTROL
- DIRECTIONS
- DRIVETIMES
- YARDCRANEPROCESS

In the following paragraphs we will discuss each of these macro's and modules and explain in which way they are different.

In general, model SUREND 2 differs from model SUREND 1 in the way the container yard is designed, which has consequences for the circulation pattern of the tractor-trailers on the terminal. It is, therefore, no surprise that the modules and macro's which control the tractor-trailers are different.
19.2 The module MAIN

The MAIN module of model SUREND 2 differs only slightly from its counterpart in model SUREND 1. The lines 131 to 137 set the upper bounds of the stacking areas as they have been defined in model SUREND 2 (see Part 1).

As in model SUREND 1, the upper bounds for the different stacking areas are initialized at 0. The value of the variable LASTAREA is initialized at 2 (stackingrow 3 and 4), so that the next ship arriving will receive stacking area 1 (stackingrow 1 and 2) in the macro STACKALLOCATION. The rest of the module is almost identical to that of model SUREND 1.

131 UBI[1] < 0
132 UBI[2] < 0
133 UBE[1] < 0
134 UBE[2] < 0
135 UBIMT < 0
136 UBEMT < 0
137 LASTAREA < 2
138

**UBI[i]**
: Attribute of MAIN
Upperbound bay value in the importstack of area[i]

**UBE[i]**
: Attribute of MAIN
Upperbound bay value in the exportstack of area[i]

**UBIMT**
: Attribute of MAIN
Upperbound bay value in the importemptystack

**UBEMT**
: Attribute of MAIN
Upperbound bay value in the exportemptystack

**LASTAREA**
: Attribute of MAIN
Stacking area of the previous SHIP
19.3 The macro STACKALLOCATON

The concept of this macro is the same for both the models, but there is a slight difference. In model SUREND 2, the container yard has been split into two stacking areas for the import and export containers (see Part 1, par 6.3).

Stack space is allocated to the ships alternatingly in STACKAREA 1 (only stacking rows 1, 2 and the empty stack row) and STACKAREA 2 (only stacking rows 3, 4 and the empty stack row). The procedure used to allocate stacking area in model SUREND 2 is identical to that used in model SUREND 1 except that the procedure is now done for either STACKAREA 1 or 2 and that the upperbound bay limit is now half (32) of that used in model SUREND 1 (64).

```
3 STACKAREA = 1 IF LASTAREA = 2
4 STACKAREA = 2 IF LASTAREA = 1
5 LASTAREA = STACKAREA
6
7 IMPORTFULL:
8 LBIMPF = UBI[STACKAREA]
9 UBIMPF = LBIMPF + IMPBAYF
10 UBI[STACKAREA] = UBIMPF
11 IF UBI[STACKAREA] > 32
12 UBI[STACKAREA] = 0
13 REPEAT FROM IMPORTFULL
14 END
15
16 EXPORTFULL:
17 LBEXPF = UBE[STACKAREA]
18 UBEXPF = LBEXPF + EXPBAYF
19 UBE[STACKAREA] = UBEXPF
20 IF UBE[STACKAREA] > 32
21 UBE[STACKAREA] = 0
22 REPEAT FROM EXPORTFULL
23 END
24
25 IMPORTEMPTY:
26 LBIMPMT = UBIMT
27 UBIMPMT = LBIMPMT + IMPBAYMT
28 UBIMT = UBIMPMT
29 IF UBIMT > 32
30 UBIMT = 0
31 REPEAT FROM IMPORTEMPTY
32 END
33
34 EXPORTEMPTY:
35 LBEXPMT = UBEIMT
36 UBEEXPMIT = LBEXPMT + EXPBAYMT
37 UBEMT = UBEEXPMIT
38 IF UBEMT > 32
39 UBEMT = 0
40 REPEAT FROM EXPORTEMPTY
41 END
42 RETURN
```

STACKAREA : Attribute of class component SHIP
Stacking area of ship in yard where containers are to be stacked
19.4 The module TTPROCESS

This module is for the greatest part the same for model SUREND 1 and model SUREND 2. However, lines 42 to 71 in model SUREND 2 replace lines 42 to 67 in model SUREND 1. The module now incorporates the possibility that a tractor-trailer must relocate along the stack.

Because the general circulation pattern of the tractor-trailers is different, the modules which coordinate this are different too.

[45 - 52]:

This section describes the actions a tractor-trailer undertakes when leaving the lane he is in.

45 WORK OUTLANEDRIVE
46 WORK OUTLANEDRIVE
47 LEAVE LANE[ROW]
48 GOTO PARKING IF ENDPROCEDURE = TRUE
49 ENTER OUTTRAFFIC
50 WORK DRIVEOUT
51 LEAVE OUTTRAFFIC
52 GOTO QUITYARD
53

[54 - 60]:

This section describes the actions undertaken to relocate along the stack. It is also possible that the tractor-trailer is sent into a SHIFTPROCEDURE (shift from importcycle to exportcycle) or ENDPROCEDURE (stop working for a ship).

54 RELOCATE :
55 CALL SHIFTCONTROL
56 REPEAT FROM DRIVING IF ENDPROCEDURE = TRUE
57 REPEAT FROM DRIVING IF SHIFTPROCEDURE = TRUE
58 WORK SHIFTDRIVE
59 ENTER TTQ[ROW]
60 REPEAT FROM INPOSITION
61

SHIFTDRIVE : Real attribute of class component SHIP
Time needed to drive along stack to new location

[62 - 71]:

This section describes how the tractor-trailer leaves the yard and reaches the entrance of the yard approached from the bridge. At this point the
tractor-trailer has the possibility of reentering the yard, should this be necessary.

52 QUIT YARD;
53 ENTER MAINTRAFFIC
54 WORK YARDTIME
55 LEAVE MAINTRAFFIC
56 IF ((BOX > 0)^(DESTINATION = "IMPEMPTYSTACK")|SHIFTPROCEDURE = TRUE)
57 CALL DIRECTIONS
58 CALL DRIVETIMES
59 SHIFTPROCEDURE = FALSE IF BOX = 0
60 REPEAT FROM YARDACTION
61 END
19.5 The macro TTCONTROL

This macro, which controls the tractor-trailers, is almost identical for both models. The only difference is that in model SUREND 2 two statements are added. Line 8 has been inserted in the macro to get a correct SHIFTPROCEDURE for the tractor-trailers.

Line 25 has been changed by adding the statement "^(box = 0)" to it.

3 IF TASK = "BRINGLOAD"
4 E = CEIL(IMPTEU OF WORKSHIP+2)
5 F = LENGTH OF IMPITLIST OF WORKSHIP
6 IF (F>E) ^ (IMPTEU OF WORKSHIP>0)
7 SHIFTPROCEDURE = TRUE
8 TASK = "GETLOAD" IF BOX = 0
9 GOTO TEST
10 END
11 JOIN THIS TRANSPORTER TO IMPITLIST OF WORKSHIP
12 RETURN
13 END
14
15 IF TASK = "GETLOAD"
16 TEST:
17 E = CEIL(REQ20F OF WORKSHIP+2) + REQ40F OF WORKSHIP
18 F = LENGTH OF FREQLIST OF WORKSHIP
19 G = CEIL(REQ20MT OF WORKSHIP+2) + REQ40MT OF WORKSHIP
20 H = LENGTH OF HTREQLIST OF WORKSHIP
21 IF (F>E) ^ (H>G)
22 ENDPROCEDURE = TRUE
23 GOTO MACROEND
24 END
25 GOTO MACROEND IF (SHIFTPROCEDURE = TRUE)^(BOX = 0)
26 IF (F>E) ^ (H>G)
27 R3 = UNIF
28 Z4 = (REQ20F OF WORKSHIP + REQ40F OF WORKSHIP) + (REQ20F OF WORKSHIP + REQ40F OF WORKSHIP + REQ20MT OF WORKSHIP + REQ40MT OF WORKSHIP)
29 DESTINATION = "EXPORTSTACK" IF R3 < Z4
30 DESTINATION = "EXPEMPTYSTACK" IF R3 > Z4
31 JOIN THIS TRANSPORTER TO FREQLIST OF WORKSHIP IF DESTINATION="EXPORTSTACK"
32 JOIN THIS TRANSPORTER TO HTREQLIST OF WORKSHIP IF DESTINATION="EXPEMPTYSTACK"
33 GOTO MACROEND
34 END
35 IF (F>E)
36 DESTINATION = "EXPORTSTACK"
37 JOIN THIS TRANSPORTER TO FREQLIST OF WORKSHIP
38 GOTO MACROEND
39 END
40 IF (H>G)
41 DESTINATION = "EXPEMPTYSTACK"
42 JOIN THIS TRANSPORTER TO HTREQLIST OF WORKSHIP
43 END
44 END
45 MACROEND:
46 RETURN
19.6 The macro SHIFTCONTROL

This macro is completely new in this model. Because of the more complex situations with the tractor-trailers when shifting in procedures, it was necessary to create a macro to control this.

[3 - 6] :

These lines determine the variables necessary for this macro.

\[
\begin{align*}
3 & \text{E} \equiv \text{CEIL} \left( \text{REQ20F OF WORKSHIP} \times 2 \right) + \text{REQ40F OF WORKSHIP} \\
4 & \text{F} \equiv \text{LENGTH OF FREQLIST OF WORKSHIP} \\
5 & \text{G} \equiv \text{CEIL} \left( \text{REQ20MT OF WORKSHIP} \times 2 \right) + \text{REQ40MT OF WORKSHIP} \\
6 & \text{H} \equiv \text{LENGTH OF HTREQLIST OF WORKSHIP}
\end{align*}
\]

[8 - 13] :

In this section the possibility is checked that the tractor-trailers are no longer needed to drive at all, and that they may end their procedures.

\[
\begin{align*}
8 & \text{IF} \ (\text{F} \geq \text{E}) \wedge (\text{H} \geq \text{G}) \\
9 & \text{ENDPROCEDURE} \equiv \text{TRUE} \\
10 & \text{SHIFTPROCEDURE} \equiv \text{FALSE} \\
11 & \text{OUTLANEDRIVE} \equiv \text{CEIL} \left( \left( 440 - (\text{BAY} \times 6.4) \right) \times \text{YARDSPEED} \right) \text{SECONDS} \\
12 & \text{return} \\
13 & \text{END}
\end{align*}
\]

[14 - 24] :

This section covers the possibility that a tractor-trailer is at an importstack and that he can pick up a container from the exportstack, further along the stack (see Part 1 , fig. 6.3.1 ). The variables necessary to do this are determined in this block.

\[
\begin{align*}
14 & \text{IF} \ (\text{DESTINATION} = \text{"IMPORTSTACK"}) \wedge (\text{F} < \text{E}) \\
15 & \text{TASK} \equiv \text{GETLOAD} \\
16 & \text{DESTINATION} \equiv \text{"EXPORTSTACK"} \\
17 & \text{SHIFTPROCEDURE} \equiv \text{FALSE} \\
18 & \text{R10} \equiv \text{UNIF} \\
19 & \text{BAY} \equiv \text{CEIL} \left( \text{R10} \times \text{EXPBAYF OF WORKSHIP} \right) + \text{LBEFPF OF WORKSHIP} \\
20 & \text{SHIFTDRIVE} \equiv \text{OUTLANEDRIVE} + \text{CEIL} \left( \left( 20 - (\text{BAY} \times 6.4) \right) \times \text{YARDSPEED} \right) \text{SECONDS} \\
21 & \text{OUTLANEDRIVE} \equiv \text{CEIL} \left( \left( 210 - (\text{BAY} \times 6.4) \right) \times \text{YARDSPEED} \right) \text{SECONDS} \\
22 & \text{JOIN THIS TRANSPORTER TO FREQLIST OF WORKSHIP} \\
23 & \text{return} \\
24 & \text{END}
\end{align*}
\]

[25 - 30] :

This block covers the case that a tractor-trailer is at an importstack, there are no containers to be
picked up from the exportstack, but there are still containers to be picked up from the empty export container stack.

```
25 IF (DESTINATION == "IMPORTSTACK") ^ (H < G)
26 TASK <= "GETLOAD"
27 DESTINATION <= "EXEMPTYSTACK"
28 JOIN THIS TRANSPORTER TO MTREQLIST OF WORKSHIP
29 RETURN
30 END
```

[31 - 41] :

This block covers the case the tractor-trailer is at an empty import container stack and must pick up a container from the empty export container stack.

```
31 IF (DESTINATION == "IMEMPTYSTACK") ^ (H < G)
32 TASK <= "GETLOAD"
33 DESTINATION <= "EXEMPTYSTACK"
34 SHIFTPROCEDURE <= FALSE
35 R10 <= UNIF
36 BAY <= CEIL(R10*EXPAYMT OF WORKSHIP) + LBEXPMT OF WORKSHIP
37 SHIFTDRIVE <= OUTLANEDRIVE + (CEIL((20 + (BAY*6.4)) + YARDSPEED))SECONDS
38 OUTLANEDRIVE <= (CEIL((210-(BAY*6.4)) + YARDSPEED))SECONDS
39 JOIN THIS TRANSPORTER TO MTREQLIST OF WORKSHIP
40 RETURN
41 END
```

[42 - 47] :

The last possibility is covered in this block. If a tractor-trailer is at an empty import container stack and there are no empty export containers to pick up, and there are still full export containers to be picked up, he will be sent to this stack by this block.

```
42 IF (DESTINATION == "IMEMPTYSTACK") ^ (F < E)
43 TASK <= "GETLOAD"
44 DESTINATION <= "EXPORTSTACK"
45 JOIN THIS TRANSPORTER TO FREQLIST OF WORKSHIP
46 RETURN
47 END
48 END
```
MACRO SHIFTCONTROL

CALLED BY TIPROCESS

YES

DESTINATION = IMPORTSTACK

NO

ENDPROCEDURE = TRUE

SHIFTPROCEDURE = FALSE

RETURN

YES

DESTINATION = IMPORTSTACK

NO

STILL FULL BOXES TO BRING

YES

TASK = GETLOAD
DESTINATION = EXPORTSTACK
JOIN IT TO FREQLIST
RETURN

NO

DESTINATION = IMPORTSTACK

STILL EMPTY BOXES TO BRING

YES

TASK = GETLOAD
DESTINATION = EXPORTSTACK
JOIN IT TO FREQLIST
RETURN

DESTINATION = IMPORTSTACK

EMPTY BOXES TO BRING

YES

TASK = GETLOAD
DESTINATION = EXPORTSTACK
JOIN IT TO FREQLIST
RETURN

TASK = GETLOAD
DESTINATION = EXPORTSTACK
JOIN IT TO FREQLIST
RETURN
This macro has been updated to direct the tractor-trailers to the correct stacking areas of the adapted layout of the container yard (Figure 19.7.1). The concept used to achieve this is the same as in model SUREND 1. The main difference is that the macro has to take account of the STACKAREA of the ship the tractor-trailer is working for (WORKSHIP).

```plaintext
3 R4 <= UNIF
4 R10 <= UNIF
5 IF (DESTINATION = "IMPORTSTACK") v (DESTINATION = "EXPORTSTACK")
6 IF STACKAREA OF WORKSHIP = 1
7 ROW <= 1 IF R4 <= 0.5
8 ROW <= 2 IF R4 > 0.5
9 END
10 IF STACKAREA OF WORKSHIP = 2
11 ROW <= 3 IF R4 <= 0.5
12 ROW <= 4 IF R4 > 0.5
13 END
14 IF DESTINATION = "IMPORTSTACK"
15 BAY <= (CEIL(R10xIMPAYT OF WORKSHIP) + LBIMPMT OF WORKSHIP)
16 END
17 IF DESTINATION = "EXPORTSTACK"
18 BAY <= (CEIL(R10xEXPAYT OF WORKSHIP) + LBEXPT OF WORKSHIP)
19 END
20 IF DESTINATION = "IMPORTVSTACK"
21 ROW <= 5
22 BAY <= (CEIL(R10xIMPAYT OF WORKSHIP) + LBIMPMT OF WORKSHIP)
23 END
24 IF DESTINATION = "EXPORTVSTACK"
25 ROW <= 5
26 BAY <= (CEIL(R10xEXPAYT OF WORKSHIP) + LBEXPT OF WORKSHIP)
27 END
28 RETURN
29
30
31
32
```
yardlayout 2

to quay

to hinterland

import import import import empty

expert expert expert expert empty

parking area

FIG. 19.7.1
19.8 The macro DRIVETIMES

This macro has also been updated in accordance with the new layout.

[4 - 8] :

This section defines the driving times corresponding to a destination lying in the beginning of the stacks (the import containers).

3 DRIVEIN ≤ (CEIL((ROWx30)+YARDSPEED)) SECONDS
4 IF (DESTINATION = "IMPORTSTACK") ∨ (DESTINATION = "IMPORTEMPTYSTACK")
5 INLANEDRIVE ≤ (CEIL((BAyx0.4)+YARDSPEED)) SECONDS
6 OUTLANEDRIVE ≤ (CEIL((210-(BAyx0.4)+YARDSPEED)) SECONDS
7 OUTLANEDRIVE ≤ (CEIL((440-(BAyx0.4)+YARDSPEED)) SECONDS IF TYPE = "TRUCK"
8 END

[9 - 12] :

This block defines the driving times corresponding to a destination lying at the end of the stacks (the export containers).

The last lines determine the time to drive from the lane where the tractor-trailer delivered its load to the main road (DRIVEOUT) and back to the entrance of the yard (YARDTIME).

9 IF (DESTINATION = "EXPORTSTACK") ∨ (DESTINATION = "EXPORTEMPTYSTACK")
10 INLANEDRIVE ≤ (CEIL((230+(BAyx0.4)+YARDSPEED)) SECONDS
11 OUTLANEDRIVE ≤ (CEIL((210-(BAyx0.4)+YARDSPEED)) SECONDS
12 END
13 DRIVEOUT ≤ DRIVEIN
14 YARDTIME ≤ (INLANEDRIVE + OUTLANEDRIVE)
15 RETURN
The only difference in this module is the insertion of a new definition of the variable LOCATION (lines 13 to 15). In case the destination of a tractor-trailer lies in the second part of the stack, the distance the Yardcrane will have to roll is larger. The rest of the module is identical.

3 ROLLTOLOCATION:
4 PAST < NOW
5 ENTER NOWORK
6 WAIT WHILE (TTQCRANENN) IS EMPTY)^(TRUCKQCRANENN) IS EMPTY)
7 LEAVE NOWORK
8 IDLETIME <= IDLETIME + (NOW - PAST)
9 WAITTIME <= WAITTIME + (NOW - PAST)
10 CALL STRATEGY
11 CALL YARDLOAD
12 LOCATION <= BAY OF SERVEUNIT X 6.4
13 IF (DESTINATION OF SERVEUNIT = "EXPORTSTACK") v
14 LOCATION <= LOCATION + 230
15 END
16 ROLLTIME <= (CEIL(ABS((LOCATION - PREVLOCATION) + RTGSPEED))))SECONDS
17 WORK ROLLTIME
18
19 IF TASK OF SERVEUNIT = "BRINGLOAD"
20 WORK LIFTOFFTIME
21 IF (BOX OF SERVEUNIT = 2)^(LOAD OF SERVEUNIT = "O")
22 WORK CATROLLTIME[CRANENN]
23 WORK LIFTOFFTIME
24 END
25 BOX OF SERVEUNIT <= 0 IF LOAD OF SERVEUNIT = "O"
26 IF TYPE OF SERVEUNIT = "TRUCK"
27 Activate SERVEUNIT FROM BYETRUCK IN TRUCKPROCESS
28 END
29 IF TYPE OF SERVEUNIT = "TT"
30 Activate SERVEUNIT FROM QUITLANE IN TTPROCESS
31 END
32 WORK CATROLLTIME[CRANENN]
33 GOTO NEWLOC
34 END
35 IF TASK OF SERVEUNIT = "GETLOAD"
36 AGAINWORK:
37 WORK CATROLLTIME[CRANENN]
38 WORK LIFTONTIME
40 IF BOX OF SERVEUNIT = 2
41 WORK CATROLLTIME[CRANENN]
42 WORK LIFTONTIME
43 END
44 IF TYPE OF SERVEUNIT = "TRUCK"
45 Activate SERVEUNIT FROM BYETRUCK IN TRUCKPROCESS
46 END
47 IF TYPE OF SERVEUNIT = "TT"
48 Activate SERVEUNIT FROM QUITLANE IN TTPROCESS
49 END
50 END
51 NEWLOC:
52 PREVLOCATION <= LOCATION
53 REPEAT FROM ROLLTOLOCATION
20.1 Introduction

The model SUREND 3 can be seen as a combination of model SUREND 1 and model SUREND 2. Many of the modules and macro's are the same as in the other models. Of the modules and macro's listed in par.19.1 the following are different or similar.

- MAIN = different
- STACKALLOCATION = different
- TTPROCESS = different
- TTCONTROL = same as in model SUREND 2
- SHIFTCONTROL = different
- DIRECTIONS = different
- DRIVETIMES = same as in model SUREND 1
- YARDCRANEPROCESS = same as in model SUREND 2

In the following paragraphs we will discuss only the modules and macro's which differ from the other models.
20.2 The module MAIN

Again the MAIN module is slightly different to incorporate the new container yard layout and its consequences. This time the differences are to be found in lines 136 to 141. In these lines the upper bounds for the stackareas are initialized.

```
136 FOR O < 1 TO 5
137 UBIM[O] < 0
138 UBE[O] < 2B
139 END
140 UBIMT < 0
141 LASTAREA < 0
```

20.3 The macro STACKALLOCATION

The concept of this module is the same as in model SUREND 3. However, this time the empty containers are all handled in lines 25 to 29, corresponding to the chosen yard layout (see fig. 6.4.1). It is also assumed that the space for the empty containers is always adequate.

```
3 STACKAREA < LASTAREA + 1
4 STACKAREA < 1 IF STACKAREA > 5
5 LASTAREA < STACKAREA
6
7 IMPORTFULL:
8 LBIMPF < UBIBSTACKAREA)
9 UBIMPF < LBIMPF + IMPBAYF
10 UBIBSTACKAREA < UBIMPF
11 IF UBIBSTACKAREA > 27
12 UBIBSTACKAREA < 0
13 REPEAT FROM IMPORTFULL
14 END
15
16 EXPORTFULL:
17 LBEXPF < UBE[STACKAREA]
18 UBEXPF < LBEXPF + EXPBAYF
19 UBE[STACKAREA] < UBEXPF
20 IF UBE[STACKAREA] > 54
21 UBE[STACKAREA] < 2B
22 REPEAT FROM EXPORTFULL
23 END
24
25 EMPTYs:
26 LBIMPM < 54
27 UBIMPM < 64
28 LBEXPm < LBIMPM
29 UBEXPm < UBIMPM
30
31 RETURN
```
In this module only lines 42 to 83 are different.

This block covers three possibilities for the status of a tractor-trailer. If a tractor-trailer is sent away and must still bring an empty container to the empty import stack, he will get the correct instructions. If the SHIFTPROCEDURE of a tractor-trailer is TRUE, he will be directed to the right place in its process. All other tractor-trailers will reach the next block.

This section describes the actions a tractor-trailer undertakes to leave the lane and the stacking area.

This block describes the actions undertaken by a tractor-trailer to relocate.

This section is a process description of the tractor-
trailer leaving the yard. At the entrance to the bridge, the tractor-trailer may have to drive back into the yard. If this is the case (determined in line 78), the tractor-trailer will receive the instructions needed.

74 QUITYARD;
75 ENTER MAINTRAFFIC
76 WORK YARDATE
77 LEAVE MAINTRAFFIC
78 IF ((BOX > 0) ∧ (DESTINATION = "IMPEMPTYSTACK") ∨ (SHIFTPROCEDURE = TRUE))
79 CALL DIRECTIONS
80 CALL DRIVETIMES
81 SHIFTPROCEDURE = FALSE IF BOX = Ø
82 REPEAT FROM YARDACTION
83 END
20.5 The macro SHIFTCONTROL

This macro is almost identical to that of model SUREND 2. However, because of the different yard layout and, therefore, the driving times of the tractor-trailers, the macro is slightly different. The sequence of the situations run through are the same, however, the instructions within each block may be different.

1 108
2 20.5 The macro SHIFTCONTROL
3 4
4 5
5 6
6 7
7 8
8 9
9 10
10 11
11 12
12 13
13 14
14 15
15 16
16 17
17 18
18 19
19 20
20 21
21 22
22 23
23 24
24 25
25 26
26 27
27 28
28 29
29 30
30 31
31 32
32 33
33 34
34 35
35 36
36 37
37 38
38 39
39 40
40 41
41 42
42 43
43 44
44 45
45 46

J E = CEIL(REQ20F OF WORKSHIP+2) + REQ40F OF WORKSHIP
J F = CEIL(REQ20MT OF WORKSHIP+2) + REQ40MT OF WORKSHIP
J G = GPMT
J H = LENGTH OF MTREQLIST OF WORKSHIP
J 8 TASK = "GETLOAD"
J 9 IF (F>E) ^ (H>0)
J 10 SHIFTPROCEDURE = TRUE
J 11 ENDPROCEDURE = FALSE
J 12 OUTLANEDRIVE = CEIL((420 - (BAYx6.4))xYARDSPEED) SECONDS
J 13 RETURN
J 14 END
J 15 IF (DESTINATION ="IMPORTSTACK") ^ (F < E)
J 16 DESTINATION = "EXPORTSTACK"
J 17 SHIFTPROCEDURE = FALSE
J 18 JOIN THIS TRANSPORTER TO FREQLIST OF WORKSHIP
J 19 RIB = UNIF
J 20 BAY = CEIL(RIBxEXPF OF WORKSHIP) + LBEXPF OF WORKSHIP
J 21 SHIFTPROCEDURE = OUTLANEDRIVE = CEIL(((64-BAY)x6.4)+YARDSPEED) SECONDS
J 22 OUTLANEDRIVE = CEIL((420-(BAYx6.4))xYARDSPEED) SECONDS
J 23 RETURN
J 24 END
J 25 IF (DESTINATION ="IMPORTSTACK") ^ (H < G)
J 26 DESTINATION = "EXPEMPTYSTACK"
J 27 SHIFTPROCEDURE = FALSE
J 28 JOIN THIS TRANSPORTER TO MTREQLIST OF WORKSHIP
J 29 BAY = CEIL(RIB + 54)
J 30 SHIFTDRIVE = OUTLANEDRIVE-CEIL((64-BAY)x6.4)+YARDSPEED) SECONDS
J 31 OUTLANEDRIVE = CEIL((420-(BAYx6.4))xYARDSPEED) SECONDS
J 32 RETURN
J 33 END
J 34 IF (DESTINATION ="IMPEMPTYSTACK") ^ (H < G)
J 35 DESTINATION = "EXPEMPTYSTACK"
J 36 SHIFTPROCEDURE = FALSE
J 37 JOIN THIS TRANSPORTER TO MTREQLIST OF WORKSHIP
J 38 SHIFTDRIVE = 0 SECONDS
J 39 RETURN
J 40 END
J 41 IF (DESTINATION ="IMPEMPTYSTACK") ^ (F < E)
J 42 DESTINATION = "EXPORTSTACK"
J 43 JOIN THIS TRANSPORTER TO FREQLIST OF WORKSHIP
J 44 RETURN
J 45 END
J 46
20.6 The macro DIRECTIONS

This macro has been adapted according to the layout illustrated in fig. 6.4.1 (Part 1). The stack allocation is done quite differently as in the other models. A ship receives two stacks to contain the containers which it will import and export. A next ship receives the next two stacks and so on. Because there are five stacks in total, there are five combinations of stacks possible which have been defined as STACKAREA (stackingrows 1 & 2 or 3 & 4 or 5 & 1 or 2 & 3 or 4 & 5). Once the stack area of the ship is known, the place along the stack is determined according to the destination of the tractor-trailers.

3 R4 $\leq$ UNIF
4 R10 $\leq$ UNIF
5
6 IF STACKAREA OF WORKSHIP = 1
7 ROW $\leq$ 1 IF R4 $\leq$ 0.5
8 ROW $\leq$ 2 IF R4 $>$ 0.5
9 END
10 IF STACKAREA OF WORKSHIP = 2
11 ROW $\leq$ 3 IF R4 $\leq$ 0.5
12 ROW $\leq$ 4 IF R4 $>$ 0.5
13 END
14 IF STACKAREA OF WORKSHIP = 3
15 ROW $\leq$ 5 IF R4 $\leq$ 0.5
16 ROW $\leq$ 1 IF R4 $>$ 0.5
17 END
18 IF STACKAREA OF WORKSHIP = 4
19 ROW $\leq$ 2 IF R4 $\leq$ 0.5
20 ROW $\leq$ 3 IF R4 $>$ 0.5
21 END
22 IF STACKAREA OF WORKSHIP = 5
23 ROW $\leq$ 4 IF R4 $\leq$ 0.5
24 ROW $\leq$ 5 IF R4 $>$ 0.5
25 END
26 IF DESTINATION = "IMPORTSTACK"
27 BAY = CEIL(R10 x IMPBAYF OF WORKSHIP) + LBIMPF OF WORKSHIP
28 END
29 IF DESTINATION = "EXPORTSTACK"
30 BAY = CEIL(R10 x EXPBAYF OF WORKSHIP) + LBEXPF OF WORKSHIP
31 END
32 IF (DESTINATION = "IMPORTSTACK") v (DESTINATION = "EXPEMPTSTACK")
33 BAY = CEIL(R10 x 10) + 54
34 END
35 RETURN
FIG. 20.6.1

yard layout 3

to quay

to hinterland

parking area