

VAKGROEP  
WATERBOUWKUNDE  
Afd. Civiele Techniek  
TH-Delft

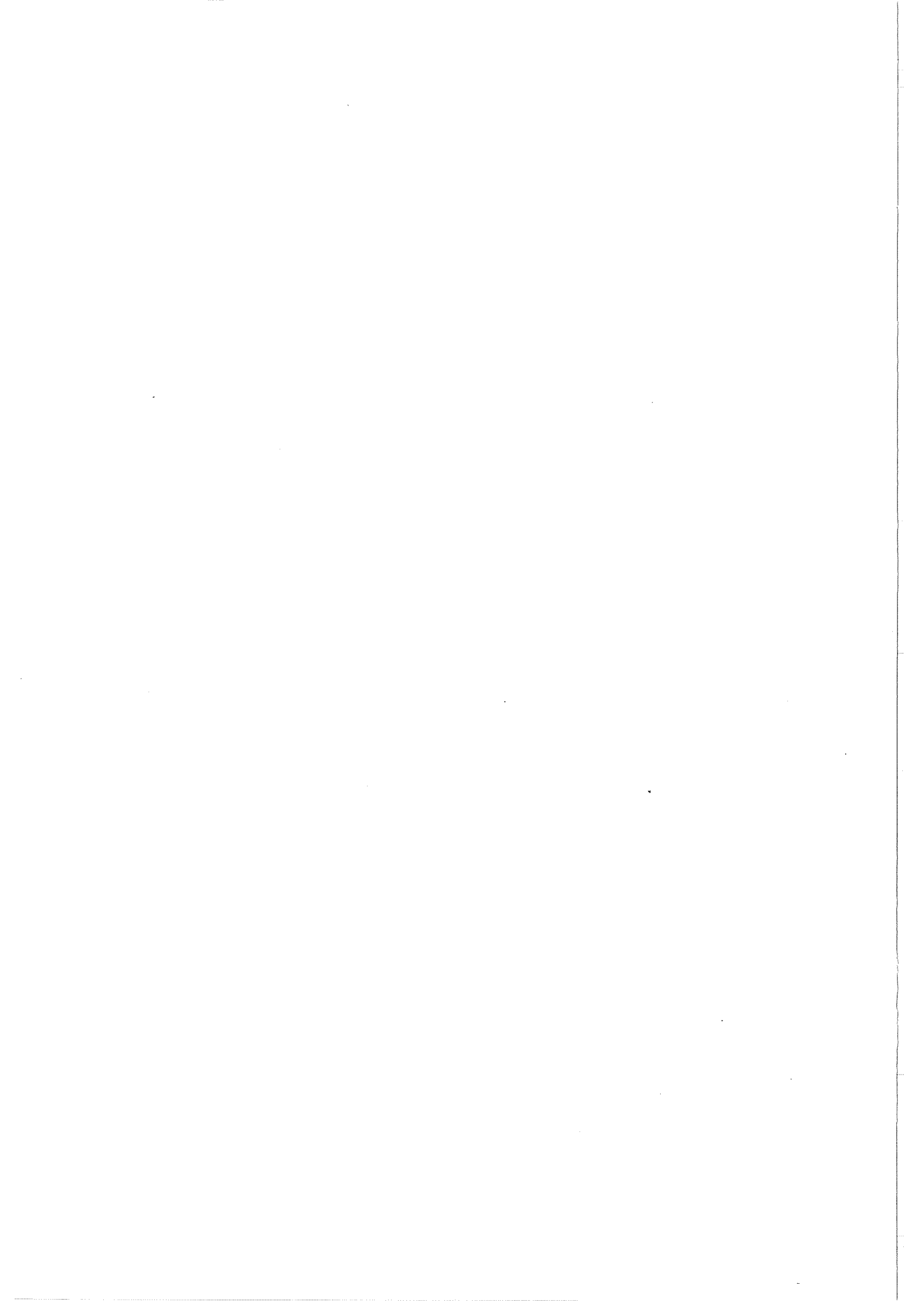
HAVENSIMULATIEMODELLEN

R. Groenveld

nr. 1478020101

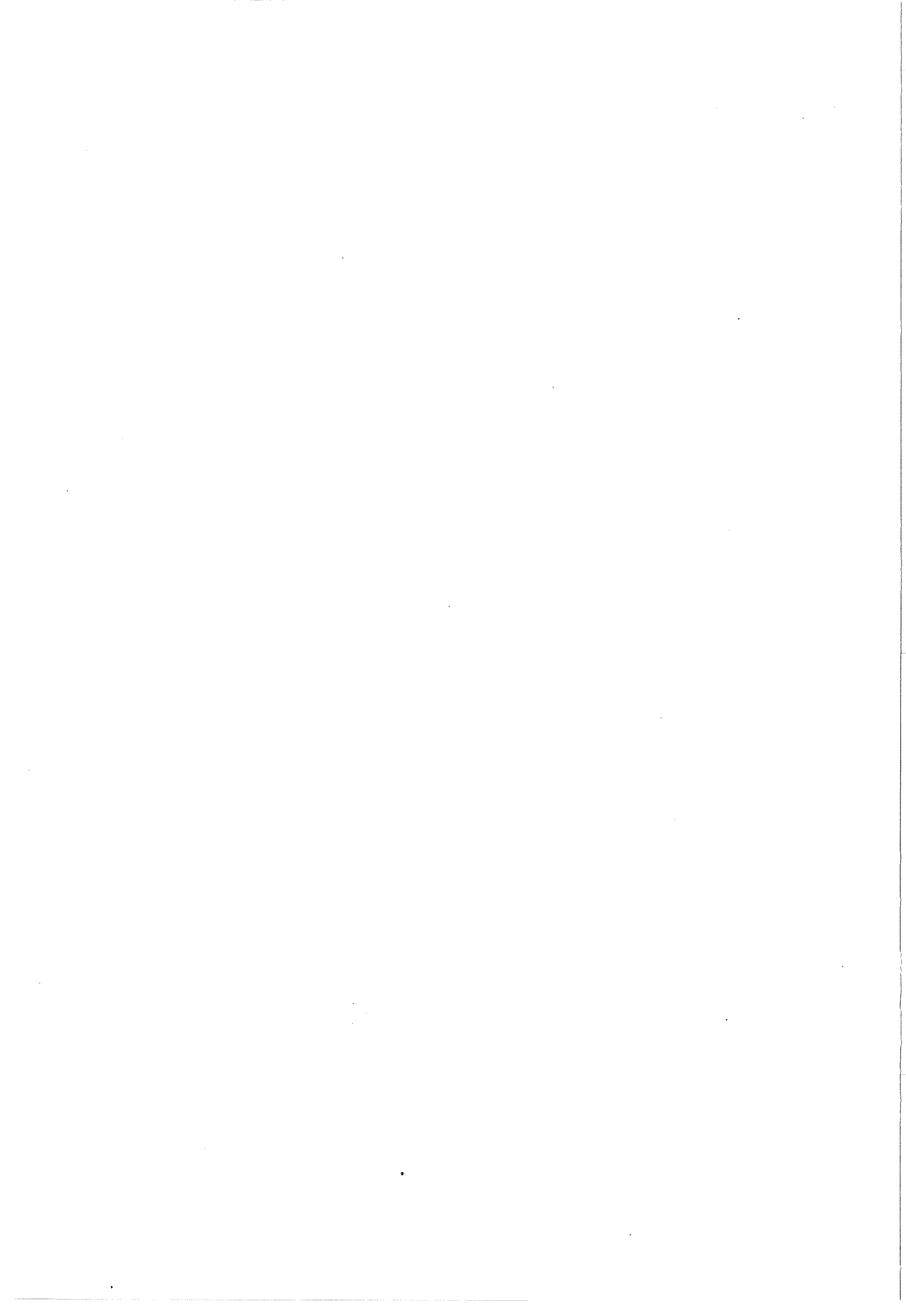
Bibliotheek

Delft, december 1980.



## Inhoud

1. Inleiding
2. Discrete computersimulatie van een stochastisch systeem
  - 2.1 Inleiding
  - 2.2 Begrippen event, component, attribuut
  - 2.3 Beschrijvingsmethoden
  - 2.4 Het trekken van onafhankelijke getallen uit een gegeven kansverdeling
  - 2.5 Voorbeeld van een computersimulatiemodel
3. Verkeerssimulatiemodel haven Zeebrugge
  - 3.1 Geschiedenis haven Zeebrugge
  - 3.2 Doelstelling verkeerssimulatiemodel
  - 3.3 Beschrijving van het systeem en de randvoorwaarden
  - 3.4 Beschrijving model middels de procesmethode
    - 3.4.1 Proces van de havenmeester en de loods
    - 3.4.2 Het proces van de schepen
  - 3.5 Enige invoergegevens
4. Optimalisatie m.b.v. een simulatiemodel



1. Inleiding

Teneinde te kunnen vaststellen op welke wijze de efficiency van een haven op een bepaald moment is te verbeteren of te kunnen voorspellen welke problemen zullen ontstaan in de toekomst wanneer bijvoorbeeld de verkeersomstandigheden zich wijzigen, is het noodzakelijk effecten te kunnen berekenen van iedere verandering in de randvoorwaarden van het havensysteem. De berekening van deze effecten is over het algemeen niet uit te voeren met behulp van een simpele methode. Een haven laat zich over het algemeen niet zover schematiseren waardoor met behulp van wachttijd en stagnatie theorieën van service-systemen de genoemde effecten zich laten kwantificeren. Iedere analyse van het gedrag van een haven gebaseerd op de aanname, dat de overheersende gedragspatronen uitgedrukt in een wiskundige formulering zich niet zullen wijzigen, leidt tot fouten.

De meest betrouwbare en de meest onthullende methode om effecten van veranderingen in de randvoorwaarden te kunnen vaststellen en te kwantificeren is de veranderingen als experiment in het havengebeuren zelf uit te voeren en het resultaat te registreren. Deze methode zal over het algemeen echter niet uitvoerbaar zijn om redenen van tijd en kosten.

Een meer acceptabel alternatief is dan een model van het havensysteem te maken waarmee de experimenten worden uitgevoerd, hetgeen simulatie wordt genoemd. Een simulatie is dan ook eigenlijk niets anders dan het naspelen van een geschematiseerde bestaande of mogelijk toekomstige realiteit.

Hierop aansluitend geeft Prof. Sierenberg de volgende definitie: Simulatie is het ontwerpen van, en het werken met modellen van dynamische (meestal ook stochastische) systemen waarmee het mogelijk is door het doen van experimenten gegevens te verkrijgen die, vaak in statistische zin, relevant zijn voor het oorspronkelijke systeem.

Simulatiemodellen kunnen voor een aantal doeleinden worden gebruikt:

- a. Mogelijk is het vast te stellen wat de gevolgen (beoogde gevolgen) van een ingreep of een reeks van ingrepen in het systeem zijn, zowel op korte als op langere termijn (consequentie-analyse). De ingrepen kunnen van technische aard zoals vergroting van overslagcapaciteit, verbetering van de nautische omstandigheden, vergroten kadelengte dan wel ingrepen betreffen op het strategisch beleidsniveau, zoals het wijzigen van prioriteitsregels.

- b. Door een aantal gerichte experimenten met het simulatiemodel uit te voeren is het mogelijk meer inzicht te verkrijgen in het functioneren van het systeem. Vastgesteld kan worden welke parameters erg kritisch liggen en welke parameters het functioneren nauwelijks beïnvloeden (strukturaanalyse).
- c. De randvoorwaarden van het havensysteem, zoals aankomstpatronen van schepen, zijn over het algemeen tijdsafhankelijk. Mogelijk is het met behulp van een simulatiemodel het gedrag van de systeemvariabelen in de tijd, zoals de bezettingsgraad van een kade of wachttijden van schepen, vast te stellen. Hieruit kunnen dan conclusies volgen betreffende ingrepen (investeringen, beleidsmaatregelen) die op een bepaald tijdstip moeten plaatsvinden, zodat het systeem naar behoren blijft functioneren.
- d. Wanneer het systeem volgens welke een haven gaat functioneren ingeval er wordt gesproken van een nieuw te bouwen haven, bekend is en ook de randvoorwaarden beschikbaar zijn, dan is het mogelijk vast te stellen of het ontwerp voldoet aan de eisen die eraan gesteld zijn.

Boven beschreven simulaties betreffen discrete simulaties en worden uitgevoerd met een digitale computer, waarbij onder een discrete simulatie wordt verstaan een simulatie waarvan de toestandsvariabelen die het dynamisch gedrag van het systeem beschrijven discrete funkties van de tijd zijn. Hiertegenover staat de continue simulatie-techniek die vanuit de regeltechniek tot ontwikkeling is gekomen.

De continue simulatie werd vroeger veelal uitgevoerd met een analoge rekenmachine, echter gezien de nadelen van de kleine nauwkeurigheid, de hoge kosten en het ontbreken van de mogelijkheid om zeer complexe systemen te simuleren, wordt tegenwoordig veelal gebruik gemaakt van de digitale computer. Een bijzonder soort discrete simulatie is de Monte-Carlo simulatie, waarbij de tijd relatie vereenvoudigd is tot een volgorde relatie. Duidelijk is het dat deze vereenvoudiging niet toepasbaar is met betrekking tot de discrete havensimulatiemodellen.

Schematisch kan het bovenstaande als volgt worden afgebeeld:

- |                   |   |
|-------------------|---|
|                   | - analoge machine   |
| computersimulatie | - continue computersimulatie - digitale machine <sup>x)</sup> |
|                   | - hybride machine <sup>x)</sup>                               |
|                   | - discrete computersimulatie - digitale machine               |

x) Uit een combinatie van de analoge en de digitale computer is de hybride rekenmachine ontstaan.

Naarmate de toepassingsgebieden van het simulatiegebeuren uitgestrekter zijn gaan worden, heeft ook het software systeem zijn wijzigingen ondergaan. Het gevolg is geweest een vervaging van de grenzen tussen de genoemde deelgebieden. Zo is vanuit de computertaal Pl 1 het proces georiënteerde simulatiepakket Prosim (Proces simulatie) waarmee zowel continue als discrete simulaties kunnen worden uitgevoerd, ontwikkeld. Gesproken wordt hier van een simulatiepakket omdat Prosim een pakket is dat aan de simulatietaal Pl 1 is toegevoegd. Dit in tegenstelling tot bijvoorbeeld Simula, waar het een simulatietaal betreft met zijn eigen syntactische regels.

Een en ander kan nog nader worden toegelicht met het onderstaande schema:

DIGITALE COMPUTER

<u>General purpose talen</u>	<u>Betekenis naam</u>	<u>Opmerkingen</u>
Algol 60	Algorithmic Language	Voor wiskundige problemen
Fortran	Formula Translation	Idem
Cobol	Common Business Oriented Language	Geschikt voor administratieve problemen
Pl 1	Programming Language	Door IBM gebracht als vervanger van Fortran en Cobol

SIMULATIE TALEN/PAKKETTEN VOOR DISCRETE SIMULATIE

<u>Taal/Pakket</u>	<u>Betekenis naam</u>	<u>Soort + toepassing</u>		
Simula 67 <sup>x</sup>	Simulation Language	D	Pr A E	T
CSL	Controle and Simulation Language	D	A	T
Prosim 78	Proces-Simulatie	D/C	Pr/A	P

SIMULATIE TALEN VOOR CONTINUE SIMULATIE

CSMP <sup>xx</sup> III	Continuous System Modelling Program	C	-	T
Dynamo <sup>xxx</sup>	Dynamic System Modelling	C	-	T

\* Simula 67 gebruikers zijn niet aan een beschrijvingsmethode gebonden

- \*\* Voor SCMP<sup>III</sup> is een goede voorkennis van Fortran nodig en dit wordt voornamelijk gebruikt om continue technische systemen te simuleren.
- \*\*\* Dynamo is een simulatietaal, ontworpen om dynamische feedback modellen van sociale, economische en ecologische systemen te simuleren.

Verklaring afkortingen:

- D = Discrete simulatietaal of pakket  
C = Continue simulatietaal of pakket  
Pr = Proces beschrijvingsmethode  
A = Activiteiten beschrijvingsmethode  
E = Event beschrijvingsmethode

Het simulatiewerk omvat de volgende punten:

1. Specificatie van het probleem en bevredigende antwoorden op de vragen:
  - a. Is het zinvol een simulatie toe te passen (dit hangt o.a. samen met de vraag of het systeem naar behoren te beschrijven is, en of er voldoende gegevens betreffende de randvoorwaarden aanwezig of te verzamelen zijn).
  - b. Kan de simulatie worden gedaan binnen de gestelde tijd en het gestelde budget.
2. Het opstellen van een verbaalmodel van het te simuleren reallife systeem, waarbij de mate van gedetailleerdheid van het te bouwen model (afhankelijk van de probleemstelling) moet worden vastgesteld. De voor- en nadelen van het detail niveau kunnen met behulp van de volgende tabel worden aangegeven.

STERK GESCHEMATISEERD

<u>Voordelen</u>	<u>Nadelen</u>
1. Opstellen van het model is eenvoudig.	1. Grote aannamen die misschien niet onder alle omstandigheden juist zijn.
2. De randvoorwaarden zijn over het algemeen gemakkelijk vast te leggen.	2. Implicaties van de aannamen zijn veelal niet duidelijk en moeilijk te evalueren.
3. Model resultaten zijn meer algemeen toepasbaar.	3. Model resultaten kunnen geen details geven.

Mogelijke redenen om de benadering te verwerpen: Resultaten kunnen onder bepaalde omstandigheden niet geldig zijn of er zijn geen conclusies uit te trekken.

GEDETAILLEERD

Voordelen

1. Basis aannamen zijn eenvoudig en duidelijk.
2. Model resultaten geven details.

Nadelen

1. Het opstellen van het model is een omvangrijk werk.
2. Resultaten zijn specifiek en vele simulaties zijn nodig om al de mogelijkheden na te gaan.
3. Grote hoeveelheden gegevens zijn nodig.

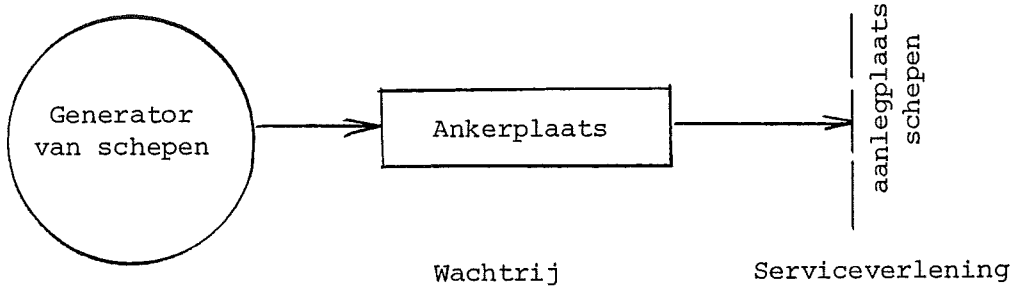
Mogelijke redenen om de benadering te verwerpen zijn: Het gedetailleerd model kan te kostbaar zijn en in een aantal gevallen zijn te weinig gegevens beschikbaar.

3. Het verbaal model omzetten in een computer programma waarbij gebruik gemaakt wordt van een beschrijvingsmethode (zie par. 2.3).
4. De ijking van het model aan de actuele situatie voordat er een verandering plaatsvindt in de randvoorwaarden en voordat er wijziging in de werking van het systeem wordt aangebracht.
5. Het doen van experimenten, zodat gegevens worden verkregen waarmee het gestelde probleem kan worden opgelost.

2. Discrete computersimulatie van een stochastisch systeem

2.1 Inleiding

Zoals in par. 1 gesteld, is het over het algemeen niet mogelijk een haven-systeem, als service systeem, analytisch op te lossen met behulp van wachttijdtheorieën. Een stochastisch systeem waarbij dit nog mogelijk is, kan als volgt worden gekarakteriseerd:



De tijd nodig voor schepen om van de ankerplaats naar de kade te varen wordt hierbij verwaarloosd, hetgeen bij de meeste havenmodellen niet toelaatbaar is.

Wordt nu voor het aankomstproces bijvoorbeeld een Poisson-tussenaankomst-tijdverdeling genomen met een gemiddelde van  $1/\lambda$  en voor de servicetijdverdeling een negatief exponentiële verdeling met gemiddelde  $1/\mu$ , dan is volgens de wachttijdtheorie wanneer het een vertragingssysteem betreft (aan ieder schip dat het systeem binnenkomt wordt ook service verleend) en gerekend wordt met  $s$  aanlegplaatsen, de stagnatiekans:

$$St = \frac{\frac{a^s}{s!} \cdot \frac{s}{s-a}}{1 + a + \frac{a^2}{2!} + \dots + \frac{a^{s-1}}{(s-1)!} + \frac{a^s}{s!} \left(\frac{s}{s-a}\right)}$$

en de gemiddelde wachttijd bedraagt:

$$W = \frac{a}{s-a} \cdot \frac{St}{\lambda}$$

De ankerplaats is hierbij een wachtrij met discipline FIFO (first in first out). Andere wachtrij disciplines zijn LIFO (last in first out), random, SPT (shortest processingtime first) of een wachtrijdiscipline volgens vastgestelde prioriteitsregels.

Indien echter complicaties in dit systeem optreden, bijvoorbeeld wanneer de kade niet is opgedeeld in gedeelten, zodat het aantal ligplaatsen langs de kade varieert met de lengte van de schepen of wanneer de tijd, die nodig is om het schip van de ankerplaats naar de kade te brengen niet meer te verwaarlozen is, dan kan m.b.v. de wachttijdtheorie het probleem niet meer worden opgelost en is de overstap naar (computer) simulatie noodzakelijk.

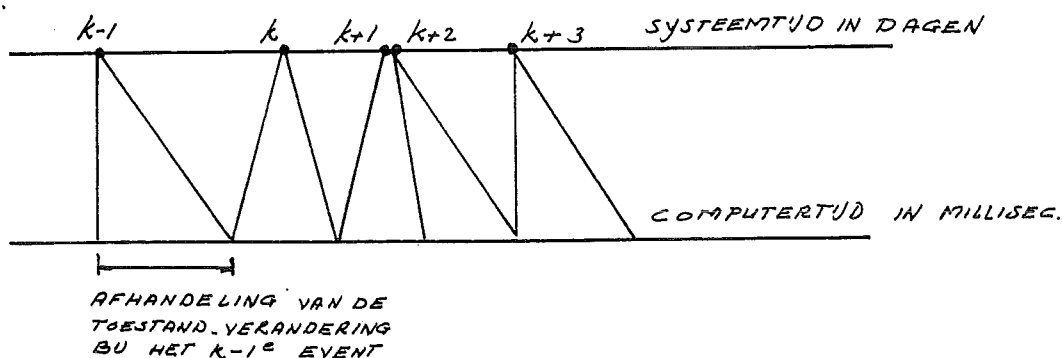
Alvorens een eenvoudig computersimulatiemodel van een kade te behandelen, is het zinvol eerst de begrippen event, component en attribuut toe te lichten, de drie systeembeschrijvingsmethoden te behandelen en daar haven-simulatiemodellen over het algemeen een stochastisch karakter hebben, in te gaan op het trekken van getallen uit een gegeven kansverdeling.

## 2.2 Begrippen event, component, attribuut

Een event is een gebeurtenis waardoor de situatie waarin het systeem zich bevindt, verandert. Bij discrete simulatie verandert de toestand waarin het systeem (model) zich bevindt tussen twee opvolgende gebeurtenissen, niet. In een havensysteem is bijvoorbeeld de aankomst of het vertrek van een schip een event. Het tijdstip waarop dit plaatsvindt wordt eventtime genoemd. Duidelijk is het dat de verschillende gebeurtenissen (events) niet dezelfde veranderingen in het systeem met zich meebrengen; er wordt gesproken van eventklassen. Een complicatie treedt op indien er twee gebeurtenissen tegelijkertijd moeten plaatsvinden, bijvoorbeeld het vertrek van een schip dat langs de kade ligt en de aankomst van een schip op de ankerplaats. Een computer kan namelijk maar één handeling tegelijk verrichten. Het probleem wordt dan als volgt opgelost:

- Zoals gesteld verandert de toestand van het systeem niet tussen twee opvolgende eventtijdstippen (de tussenliggende tijd kan door de computer worden overgeslagen).
- Er is geen tijd nodig om een verandering in de toestand van het systeem aan te brengen.

In schema:



Iedere systeembeschrijvingsmethode formuleert het gedrag en de interactie van de in het systeem voorkomende componenten. In een havensysteem zijn tijdelijke en permanente componenten aanwezig. Een voorbeeld van een tijdelijke component is een schip dat nadat het de haven verlaat, ook uit het systeem verdwijnt terwijl de havenmeester die voortdurend aanwezig is, een voorbeeld van een permanente component is. Componenten dienen nader gespecificeerd te worden. Dit nader specificeren gebeurt met een "attribuutlijst", die op iedere component wordt "geplakt".

Een schip heeft bijvoorbeeld als attributen:

1. aankomsttijd in het systeem
  2. scheepstype
  3. lengte
  4. diepgang
  5. bestemming in de haven
  6. servicetijd
  7. maximaal toelaatbare wachttijd (bijvoorbeeld i.v.m. aan bederf onderhevige goederen)
- enz.

### 2.3 Beschrijvingsmethoden

Nadat het model verbaal is opgesteld moet het worden vertaald naar een computertaal. Dit kan zoals reeds eerder is aangeduid, op een aantal manieren gebeuren. Onderscheiden worden de procesbeschrijving, de activiteitenbeschrijving en de eventbeschrijvingsmethode.

De procesbeschrijving geeft een beschrijving van de processen, die de verschillende componenten in het systeem doormaken, bijvoorbeeld het proces van een schip, het proces van een havenmeester, het proces van een scheepsgenerator, waarbij het duidelijk is, dat de verschillende processen in elkaar grijpen.

De procesbeschrijving wordt over het algemeen als een methode gezien, waarbij het systeem op een natuurlijke manier wordt beschreven; het is tamelijk eenvoudig zich te realiseren wat voor handelingen het schip doormaakt of wat de taak is van een havenmeester. Problemen ontstaan meestal bij de interacties tussen de verschillende processen.

De activiteitenbeschrijving beschrijft alle activiteiten, die er in het systeem voorkomen, bijvoorbeeld:

- plaats een schip in de wachtrij op de ankerplaats,
- leg een schip aan de kade,
- verleen een schip service in de vorm van laad- en losfaciliteiten,
- verwijder het schip uit het havensysteem.

Iedere activiteit vindt plaats wanneer er aan een aantal voorwaarden zijn voldaan. Voorwaarden voor het afmeren van een schip langs de kade kunnen zijn:

1. Er moet een schip in de wachtrij "ankerplaats" liggen.
2. Het schip moet qua lengte passen in de vrije kadelengte.
3. Getij moet gunstig zijn.
4. Vaarweg moet vrij zijn, enz.

Deze methode is niet bijzonder efficiënt vanwege het herhaald nagaan of aan de betreffende voorwaarden voor de activiteit is voldaan.

Tenslotte de eventbeschrijving.

In par. 2.2 is reeds duidelijk gemaakt wat wordt bedoeld met een eventklasse. De eventbeschrijving geeft aan welke toestandsveranderingen moeten plaatsvinden afhankelijk van de eventklasse. Bijvoorbeeld bij de aankomst van een schip dient te worden nagegaan of het schip aan de kade kan worden afgemeerd of dat het in een wachtrij moet plaatsnemen. Op het volgende eventtijdstip verlaat een schip het systeem en moet worden nagegaan of een schip in de wachtrij zijn plaats kan innemen.

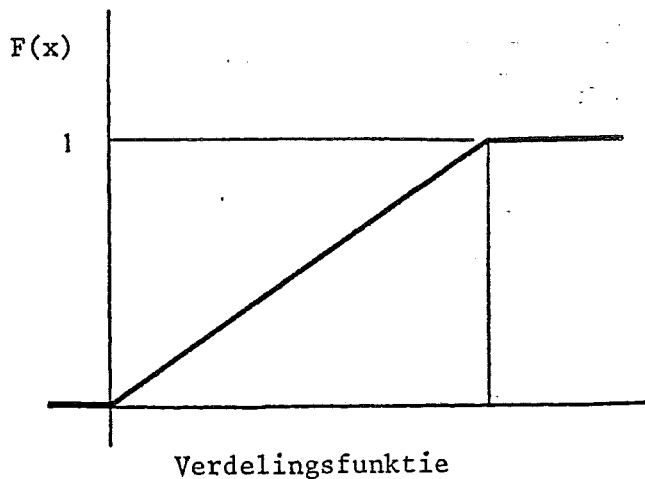
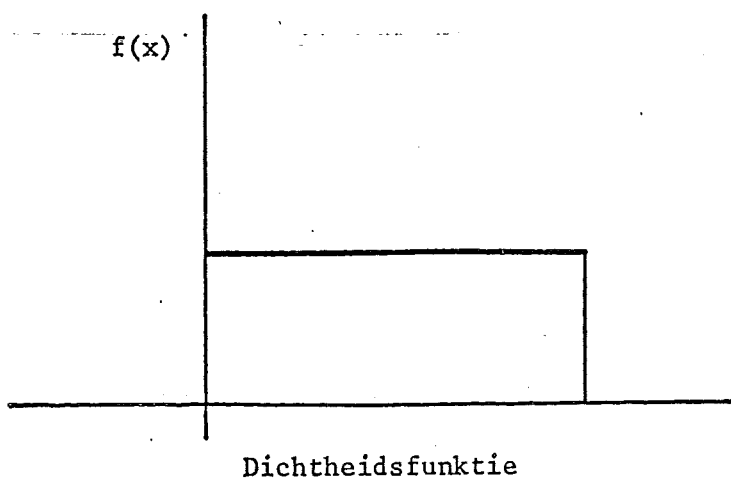
Voor een eventmethodebeschrijving is het noodzakelijk, dat de werking van het systeem zeer grondig wordt gekend.

In par. 1 is in het overzicht van de verschillende simulatietalen/pakketten reeds vermeld dat een aantal simulatietalen/pakketten meerdere beschrijvingsmethoden toelaten. Hierdoor komt het voor, dat hoewel de procesmethode bij de beschrijving van het systeem wordt gehanteerd, er in onderdelen van de processen gebruik is gemaakt van de activiteitenbeschrijving.

#### 2.4 Het trekken van onafhankelijke getallen uit een gegeven kansverdeling

Aankomsten van schepen, servicetijden (los- en laadtijden) zijn over het algemeen onderhevig aan toevalsfluctuaties. Uit een groot aantal waarnemingen

of op grond van theoretische overwegingen kan evenwel een kansverdeling worden opgesteld. Door een trekking te doen uit deze verdelingen kunnen de aankomsttijdstippen, servicetijden enz. worden vastgesteld en kan met behulp van het model de realiteit worden nagespeeld (simulatie). Voor de trekking van onafhankelijke getallen uit een gegeven kansverdeling is het nodig een onafhankelijke trekking te kunnen doen uit een homogene verdeling. Onderstaande figuur geeft de verdelingsdichtheid en de verdelingsfunctie van een uniforme verdeling.



Een proces dat getallen produceert die beantwoorden aan een aselechte (onafhankelijke) trekking uit bovengenoemde homogene verdeling, noemt men een proces dat randomgetallen genereert. Vroeger produceerde men deze getallen met behulp van een hoge hoed waarin bijvoorbeeld tien papiertjes van 0.0, 0.1, 0.2 ..... 0.9 aanwezig waren. Door hieruit papiertjes met terugleggen te trekken, produceerde men randomgetallen. Duidelijk is het, dat een dergelijke methode omslachtig is. Er zijn dan ook tabellen met aselechte getallen opgesteld. Tippet gebruikte hiervoor de laatste vijf cijfers van elk in 20 decimalen getabelleerde logaritme.

Een voorbeeld van trekkingen uit de homogene discrete verdeling op 00, 01, 02 ..... 99 is de methode van Von Neumann:

$i$	$y_i$	$y_i^2$	$i$	$y_i$	$y_i^2$
1	42	1764	1	79	6241
2	76	5776	2	24	0576
3	77	5929	3	57	3249
4	92	8464	4	24	0576
5	46	2116	5	57	3249
6	11	0121	6	24	

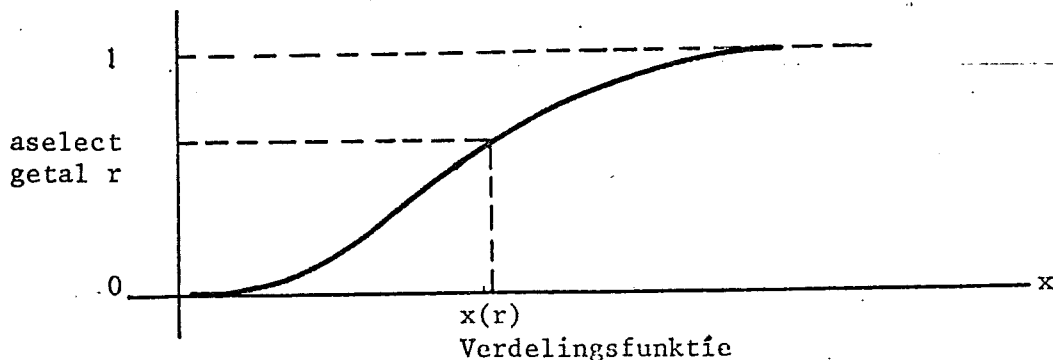
7	12	0144
8	14	0196
9	19	0361
10	36	1296
11	29	0841
12	84	7056
13	05	0025
14	02	0004
15	00	0000

enz.

Duidelijk zijn de bezwaren die aan deze methode kleven. Verfijningen zijn mogelijk om deze moeilijkheden weg te nemen, echter er zijn talrijke andere en betere methoden waar hier niet verder op in zal worden gegaan.

Nu het mogelijk is een randomgetal te trekken, kan ook een trekking worden gedaan uit een bepaalde verdelingsfunctie. Het proces volgens de directe methode geschiedt als volgt:

1. Bepaal met behulp van tabellen of een andere methode het randomgetal r.
2. Bepaal met behulp van de vergelijking  $r = F(x)$  (zie onderstaande figuur) of  $x = F^{-1}(r)$  de waarde van x; x is dan de gevraagde trekking uit de betreffende verdelingsfunctie.



Voorbeeld van een trekking uit een exponentiële verdeling met gemiddelde

(verwachting)  $\mu = \frac{1}{\lambda}$

Stel de kansdichtheidsfunctie is  $f(x) = \lambda e^{-\lambda x}$  voor  $x > 0$ .

De verdelingsfunctie is dan  $F(x) = \int_0^x \lambda e^{-\lambda x} dx = -e^{-\lambda x} + \text{const.}$

Randvoorwaarden zijn:  $x = 0 \rightarrow F(x) = 0 \rightarrow F(x) = 1 - e^{-\lambda x}$

Uitgaande van het aselect getrokken getal r kan dan de trekking uit de betreffende verdeling als volgt worden berekend:

$$r = 1 - e^{-\lambda x} \rightarrow 1 - r = e^{-\lambda x} \quad \ln(1-r) = -\lambda x \rightarrow x = -1/\lambda \ln(1-r)$$

Aangezien r en 1-r homogeen verdeeld zijn op het interval 0-1, mag de onafhankelijke trekking uit de gegeven verdeling ook als volgt worden geschreven:

$$x = -\frac{1}{\lambda} \ln r$$

Voorbeeld van een trekking uit een Bernouilli-verdeling:

Een Bernouilli-verdeling kan als volgt worden omschreven:

$P(\text{succes}) = p$  of  $P(\text{geen succes}) = 1-p$ .

Succes kan bijvoorbeeld zijn een geladen schip en geen succes een ongeladen schip.

De methode is dan als volgt:

1. Bepaal random getal  $r$

$$\text{succes} = r < p$$

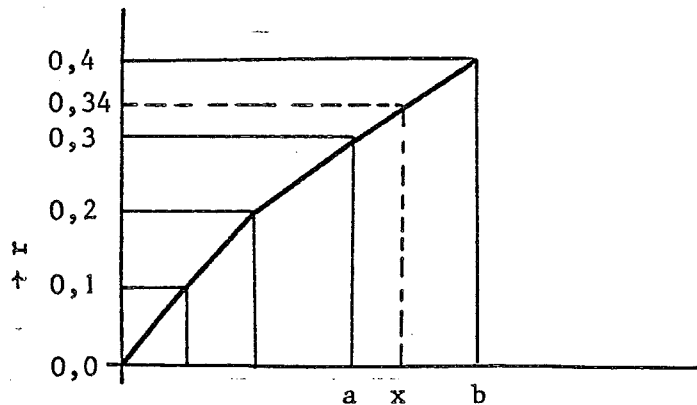
Voorbeeld van een trekking uit een verdeling in tabelvorm:

Bij een groot aantal kansverdelingen is het moeilijk de inverse  $x = F^{-1}(r)$  te bepalen. Wel is het mogelijk om  $x = F^{-1}(r)$  in tabelvorm te geven.

Om het zoeken in de opgestelde tabel efficiënt te doen en geschikt voor computerberekeningen, dient de  $r$ -as equidistant te zijn verdeeld.

$r = 0, h, 2h, 3h \dots \dots \dots nh$ , waarbij  $n \times h = 1$ .

Stel we kiezen voor  $h = 0.1 \rightarrow n = 10$  (zie onderstaande figuur):



Het trekkingsproces gaat als volgt:

1. Bepaal het randomgetal  $r$
2. Bepaal  $i = \text{entier}(n \times r)$
3. Bepaal  $x = \text{TAB}(i) + (\text{TAB}(i + 1) - \text{TAB}(i)) \cdot (r \cdot n - i)$

of zoals in de figuur is aangegeven:

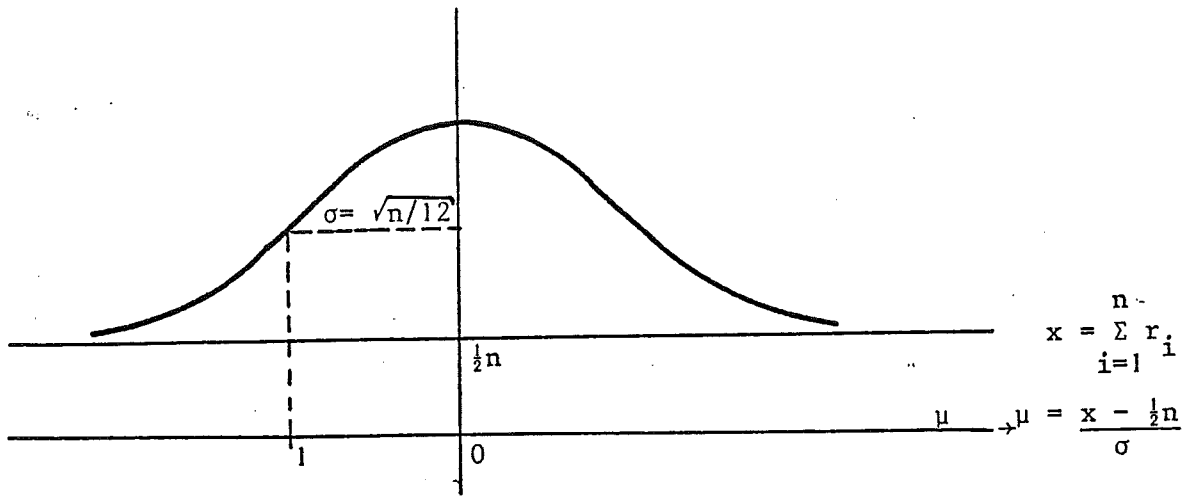
$$r = 0.34$$

$$i = 3$$

$$x = a + (b - a) \times 0.4$$

Voorbeeld trekking uit een normale verdeling met behulp van de indirecte methode.

Gebruik wordt gemaakt van de eigenschap, dat voor grote n de som van n randomgetallen bij benadering normaal verdeeld is met verwachting  $\frac{1}{2}n$  en variantie  $1/12 n$ .



Via transformatie kan nu een trekking uit een normale verdeling met gemiddelde 0 en spreiding 1 worden bepaald:

$$\mu = \frac{\sum_{i=1}^n r_i - \frac{1}{2}n}{\sqrt{1/12n}}$$

## 2.5 Voorbeeld van een computersimulatiemodel

Om de werking van een computersimulatiemodel toe te lichten, zal hier een voorbeeld worden gegeven van een kade waaraan schepen worden behandeld. De simulatie is uitgevoerd met behulp van Prosim (Processimulatie) waarbij de procesbeschrijvingsmethode is gebruikt.

### Het verbaal model

Schepen arriveren volgens een negatief exponentiële tussenaankomstverdeling met een gemiddelde of verwachting van  $1/\lambda = 3$  dagen. De schepen hebben een maximale wachttijd van 0,1 of 2 dagen met een even grote kans van optreden. De lengte van de schepen is uniform verdeeld op het traject 80 tot 120 meter. De servicetijdverdeling is uniform verdeeld op het traject 1 tot 4 dagen. Een schip dat is gearriveerd vraagt aan de havenmeester of vrije kaderuimte voldoende is om aan de kade af te meren. Indien dit het

geval is vaart het schip naar de kade waarbij uiteraard als voorwaarde moet gelden dat de maximale wachttijd niet is overschreden.

Wanneer een schip de kade bezet vindt bij aankomst, voegt het zich achter in de wachtrij. Wordt de maximale wachttijd overschreden voordat er voldoende kadelenkte beschikbaar is, dan wijkt het schip uit naar een andere haven en verdwijnt uit het systeem.

Een schip verlaat de kade en daarmee het systeem wanneer zijn servicetijd verstreken is, dus wanneer de overslagactiviteiten zijn beëindigd.

Bij het verlaten van het systeem gaat de havenmeester na, in volgorde van aankomst, of één of meer schepen die liggen te wachten op de ankerplaats, de plaats van het vertrekkende schip kan of kunnen innemen, gezien de vrije kadelenkte.

#### Beschrijving middels de procesmethode

Een component of klasse van componenten (hier schepen) wordt beschreven middels het proces dat deze component doorloopt. De componenten die in dit model een rol spelen zijn de component main, de generator van schepen, de havenmeester en de component klasse schepen. Bij de procesbeschrijvingsmethode dient steeds te worden vastgesteld wie doet wat of welke activiteit wordt door een bepaalde component uitgevoerd.

In het proces dat wordt uitgevoerd door de component MAIN (ook wel het hoofdproces genoemd) vindt plaats:

- a. definiëring van de componenten, wachtrijen en histogrammen
- b. declaraties
- c. en de runcontrol

In bijlage 1 is het proces van de component main in de prosimtaal weergegeven. De componenten die worden gedefinieerd zijn:

H.MASTER (havenmeester), SHIP (component klasse schepen) en de GENERATOR-SHIP (de scheepsgenerator).

De component MAIN behoeft in de prosimtaal niet te worden gedefinieerd. MAIN bestaat reeds voordat de simulatie begint en wordt systeemcomponent genoemd. Hierna volgt de definiëring van de wachtrijen ANCHORAGE (ankerplaats) en QUAY (kade) en de definiëring van de histogrammen ONE DAY WAITING SHIPS, TWO DAYS WAITING SHIPS, GIVING WAY SHIPS, DHANDLING SHIPS (schepen die direct worden geholpen), DEGREE-ENG (bezettingsgraad van de kade).

Vervolgens wordt een dag als de tijdseenheid en RAND als een randomgetal gedefinieerd. Steeds wanneer RAND wordt aangeroepen wordt er een nieuw randomgetal gegenereerd.

Tenslotte worden in deze DEFINITION AND DECLARATION-sectie nog een aantal parameters gedeclareerd, zoals FREE QUAY LENGTH (de vrije kadelenkte), QUAY LENGTH (de totale kadelenkte), RUNNR (het aantal runs dat reeds is afgehandeld), TOTRUNS (het totaal aantal runs dat moet worden verwerkt) en een aantal hulpgetallen.

Ook de sectie RUNCONTROL wordt uitgevoerd door de component MAIN. In deze sectie wordt gesteld dat de havenmeester zijn werk moet beginnen bij INSPECTION ANCHORAGE en de generator bij STARTGEN (dit zijn twee in het programma opgenomen labels). Wanneer het programma 350 dagen (de lengte van een run) heeft gesimuleerd (de component MAIN schort zijn werkzaamheden 350 dagen op), wordt STATUS (situatie waarin het model zich op dat moment bevindt) afgedrukt en vervolgens de histogrammen zoals vermeld in de bijlage ( 9 ). Daarna worden alle histogrammen schoongeveegd en ook de statistische gegevens van de wachtrijen worden bijgesteld. Het aantal runs wordt met 1 verhoogd en wanneer het totaal aantal runs nog niet bereikt is, schort MAIN zijn werkzaamheden weer 350 dagen op (MAIN is A SUSPENDED COMPONENT), zo niet dan stopt de simulatie: CANCEL GENERATORSHIP (er worden geen nieuwe schepen meer gemaakt) en het proces van MAIN wordt getermineerd.

Het PROCES OF THE GENERATOR begint bij STARTGEN (zie bijlage 2).

De GENERATORSHIP, op dit moment de current component, creëert een nieuw schip met het statement NEWSHIP = NEW SHIP en bepaalt de waarden van de attributen van dat nieuwe schip en wel de lengte, de maximale wachttijd en de servicetijd. De GENERATORSHIP activeert het schip, waarbij het schip zijn werkzaamheden moet beginnen bij het label ARRIVAL, en creëert een nieuw schip wanneer er -3 LOG (RAND) dagen zijn verstreken, hetgeen overeenkomt met een trekking uit een negatief exponentiële tussenaankomsttijdverdeling met een verwachting van drie dagen.

Zoals gesteld begint het PROCES OF A SHIP bij de label ARRIVAL en wordt uitgevoerd door de current component SHIP. Een schip plaatst zichzelf bij aankomst in de wachtrij ANCHORAGE. De HMASTER wordt opnieuw geactiveerd en het schip staakt zijn werkzaamheden. Wanneer daarna het schip opnieuw gereactiveerd wordt, dit gebeurt onder bepaalde voorwaarden door de haven-

meester (zie proces van de havenmeester), verlaat het schip de wachtrij ANCHORAGE. Getest wordt of de  $ACT\ WAITING\ TIME > MAX\ WAITING\ TIME$  (actuele wachttijd > maximale wachttijd). Is dit het geval, dan wordt de SHIPLENGTH genoteerd in het histogram GIVING WAY SHIPS en wanneer dit nog niet gebeurd is, wordt de havenmeester geactiveerd. Indien echter de actuele wachttijd niet is overschreden, dan wordt de SHIPLength genoteerd in het histogram dat overeenkomt met de actuele wachttijd van het schip, waarbij DHANDLING SHIPS het histogram voor schepen is die direct zijn geholpen en dus geen tijd hebben doorgebracht in de wachtrij ANCHORAGE. Het schip wordt geacht aan de kade te liggen en het aantal dagen met de bijbehorende bezettingsgraad van de periode NOW tot het vorige tijdstip waarop de bezetting van de kade is veranderd wordt genoteerd in het histogram DEGREE-ENG.

Het schip stopt nu zijn werkzaamheden gedurende zijn servicetijd (HOLD SERVICETIME). Wanneer de servicetijd is verstreken wordt opnieuw de bezettingsgraad voor de afgelopen periode genoteerd in het histogram DEGREE-ENG en verdwijnt het schip (TERMINATE) uit het systeem. Bij vertrek van een schip wordt de havenmeester weer gereactiveerd om een eventueel wachtend schip toestemming tot aanleggen te geven.

Het PROCES of HARBOURMASTER is het laatste proces (overigens is de volgorde van de verschillende processen niet van belang, wel moet het proces van de component MAIN als eerste worden beschreven). De havenmeester wordt gereactiveerd wanneer een schip aankomt in de wachtrij ANCHORAGE of wanneer de servicetijd van een schip aan de kade verstreken is en begint zijn werkzaamheden bij INSPECTION-ANCHORAGE. Indien er schepen in de wachtrij ANCHORAGE liggen, beschouwt de havenmeester het eerste schip (NEXTSHIP = ANCHORAGE.FIRST), berekent de actuele wachttijd van het schip in deze wachtrij en gaat na of de actuele wachttijd de maximale wachttijd heeft overschreden. Indien de maximale wachttijd is overschreden wordt het schip gereactiveerd. Is dit niet het geval, dan gaat de havenmeester na of de vrije kadelengte voldoende is; zo ja dan wordt het schip geactiveerd en de betreffende kadelengte voor dit schip gereserveerd. Wanneer de vrije kadelengte onvoldoende is blijft het schip in de wachtrij. Hierna beschouwt de havenmeester het volgende schip in de wachtrij (NEXTSHIP = NEXTSHIP.SUCC) en bepaalt zoals beschreven of het schip naar de kade gaat, uitwijkt naar een andere haven danwel in de wachtrij blijft. Dit gaat door totdat de havenmeester alle schepen in de wachtrij heeft bekeken.

### SEQUENTIE MECHANISME

Teneinde het programma correct uit te voeren, moeten de werkzaamheden volgens de in het model geformuleerde tijdschema worden uitgevoerd. Dit gebeurt met behulp van het, in de computertaal Prosim ingebouwde, sequentie mechanisme. Het sequentie mechanisme zorgt ervoor, dat iedere component overeenkomstig de in het model vastgestelde tijden actief wordt. Wanneer twee componenten op hetzelfde tijdstip worden geactiveerd, dan worden ze actief in de volgorde waarin zij in het model zijn geactiveerd. De taal Prosim kent twee uitvoeringsmogelijkheden van het programma, n.l.:

1. Versie ter contrôle van syntactische fouten in het programma waarbij een overzicht wordt gegeven van de eventtijdstippen.
2. De versie waarin het programma wordt uitgevoerd.

Zinvol is het nader in te gaan op de eerste uitvoeringsversie (zie bijlage 4). Op het tijdstip 0.0 wordt de component Main geactiveerd door het systeem. De component Main initialiseert de havenmeester, de component klasse schepen, de component generator, de wachtrijen ANCHORAGE en QUAY en activeert vervolgens de HMASTER en de GENERATORSHIP. Daarna schort de component MAIN zijn werkzaamheden tot het tijdstip 350 op (MAIN SUSPENDED UNTIL 350; d.w.z. MAIN hervat zijn werkzaamheden op het tijdstip 350 vanaf de plaats waar MAIN met de werkzaamheden was gestopt).

De havenmeester is op het tijdstip nul het eerst geactiveerd en wordt daarom nu de current component. Daar er nog geen schepen in het systeem zitten, wordt de havenmeester gepassifeerd.

Het sequentie mechanisme zorgt ervoor, dat vervolgens de generator de current component wordt (nog steeds op het tijdstip nul). De generator creëert schip 1, activeert dit schip en schort zijn werkzaamheden 2.9205 dageenheden op (GENERATORSHI SUSPENDED UNTIL 2.9205). Schip 1 gaat achter in de wachtrij ANCHORAGE liggen, waarna het schip de havenmeester activeert en zelf passief wordt.

De havenmeester constateert dat de maximale wachttijd niet is overschreden, de vrije kadelenkte voldoende is en reactiveert het schip 1 (zie proces van het schip). Schip 1 wordt de current component, vervolgt zijn proces, komt uit de wachtrij, stelt zijn ACT WAITING TIME vast, wordt geboekt in een van de histogrammen, hier het histogram DHANDLING SHIPS, omdat de actuele wachttijd van het schip in de wachtrij nul is en schort zijn werkzaamheden op tot zijn servicetijd beëindigd is op 2.7834. Op dit tijdstip her-

vat schip 1 zijn werkzaamheden, vertrekt en verlaat het systeem (PROCES OF SHIP 1 IS TERMINATED). Het volgende eventtijdstip op de lijst van het sequentie mechanisme is 2.9205. Tot dat tijdstip was de generator een suspended component. De generator maakt weer schip 2, activeert schip 2 en schort zijn werkzaamheden op tot het tijdstip 8.1299 enz. Wanneer het eventtijdstip 350 wordt bereikt neemt de component MAIN de zaak weer over. De component MAIN was gebleven bij het statement HOLD (350). MAIN zorgt er nu achtereenvolgens voor, dat de uitvoer wordt verzorgd, dat de histogrammen worden schoongeveegd en dat de wachtrij opnieuw wordt bijgezet.

In bijlage (8,9) is een indruk gegeven van de uitvoer van het programma met een afdruk van de status van het systeem en een aantal histogrammen na 350 dagen simuleren. De uitvoer die het statement STATUS verzorgt spreekt voor zichzelf. Ook de uitvoer van de histogrammen, waarbij zowel de dichtheidsfunctie als een cumulatieve verdelingsfunctie is afgedrukt, is zelf uitleggend.

```

00040 BEGIN;
00050 /*****/
00060 /**DEFINITION AND DECLARATION SECTION**/
00070 /*****/
00080 DEFINE
00090   HMASTER AS A COMPONENT WITH ATTRIBUTE NEXTSHIP
00100   REFERENCE TO SHIP;
00110 DEFINE
00120   SHIP AS A CLASS OF COMPONENTS EACH WITH ATTRIBUTES
00130   (SHIPLength FLOAT,MAXWAITINGTIME FIXED BIN,
00140   ACTWAITINGTIME FIXED BIN,SERVICETIME FLOAT);
00150 DEFINE
00160   GENERATORSHIP AS A COMPONENT WITH ATTRIBUTE
00170   NEWSHIP REFERENCE TO SHIP;
00180 DEFINE
00190   (ANCHORAGE,QUAY) AS QUEUES;
00200 DEFINE
00210   (ONEDAYWAITINGSHIPS,TWODAYSWAITINGSHIPS,GIVINGWAYSHIPS,
00220   DHANDLINGSHIPS) AS HISTOGRAMS(4,80,10);
00230 DEFINE
00240   DEGREE_ENG AS A HISTOGRAM(5,0,0.2);
00250 DEFINE
00260   A DAY AS A TIME_UNIT,
00270   RAND AS A RANDOMSTREAM WITH SEED(13579);
00280   DECLARE (FREEQUAYLENGTH INIT(180),QUAYLENGTH INIT(180))
00290   FLOAT,
00300   (RUNNR INIT(1),TOTRUNS INIT(3),I) FIXED BIN,
00310   (FIGURE,DEGENG,PREFREQL INIT(180),PREARRIVALTIME INIT(0),
00320   QUAYLENGTHDAYS) FLOAT;
00330
00340
00350 /*****/
00360 /**RUNCONTROL**/
00370 /*****/
00380   ACTIVATE HMASTER FROM INSPECTION_ANCHORAGE;
00390   ACTIVATE GENERATORSHIP FROM STARTGEN;
00400   WHILE RUNNR]=TOTRUNS+1 DO;
00410   HOLD 350;
00420   PRINT STATUS;
00430   PRINT ONEDAYWAITINGSHIPS HEADED BY 'ONE DAY WAITINGSHIPS';
00440   PRINT TWODAYSWAITINGSHIPS HEADED BY 'TWO DAYS WAITINGSHIPS';
00450   PRINT GIVINGWAYSHIPS HEADED BY 'GIVINGWAY SHIPS';
00460   PRINT DHANDLINGSHIPS HEADED BY 'DIRECT HANDLINGSHIPS';
00470   PRINT DEGREE_ENG HEADED BY 'DEGREE ENGAGEMENT';
00480   CLEAR ONEDAYWAITINGSHIPS;
00490   CLEAR TWODAYSWAITINGSHIPS;
00500   CLEAR GIVINGWAYSHIPS;
00510   CLEAR DHANDLINGSHIPS;
00520   CLEAR DEGREE_ENG;
00530   RESET STATISTICS OF ALL QUEUES;
00540   RUNNR=RUNNR+1;END; CANCEL GENERATORSHIP;
00550   TERMINATE;
00560

```

```

00570
00580 /*****
00590 /***PROCES OF THE GENERATOR*****/
00600 /*****
00610 STARTGEN;
00620 NEWSHIP=NEW SHIP;
00630 NEWSHIP,SHIPLength=40*RAND+80;
00640 FIGURE=RAND;
00650 IF FIGURE<=1/3 THEN NEWSHIP.MAXWAITINGTIME=2;
00660 ELSE DO; IF FIGURE<=2/3 THEN NEWSHIP.MAXWAITINGTIME=1;
00670 ELSE NEWSHIP.MAXWAITINGTIME=0;
00680 END;
00690 NEWSHIP.SERVICETIME=1+3*RAND;
00700 ACTIVATE NEWSHIP FROM ARRIVAL;
00710 HOLD -2*LOG(RAND);
00720 REPEAT FROM STARTGEN;
00730
00740
00750 /*****
00760 /***PROCES OF A SHIP*****/
00770 /*****
00780 ARRIVAL;
00790 ENTER ANCHORAGE;
00800 IF HMASTER.STATUS=PASSIVE THEN REACTIVATE HMASTER;
00810 PASSIVATE;
00820 LEAVE THE ANCHORAGE;
00830 IF ACTWAITINGTIME>MAXWAITINGTIME
00840 THEN DO; TALLY SHIPLength IN GIVINGWAYSHIPS;
00850 TERMINATE;
00860 END;
00870 IF ACTWAITINGTIME=1 THEN
00880 TALLY SHIPLength IN ONEDAYWAITINGSHIPS;
00890 IF ACTWAITINGTIME=2 THEN
00900 TALLY SHIPLength IN TWODAYSWAITINGSHIPS;
00910 IF ACTWAITINGTIME=0 THEN
00920 TALLY SHIPLength IN DHANDLINGSHIPS;
00930 QUAYLENGTHDAYS=(NOW-PREARRIVALTIME)*24;
00940 DEGENG=(QUAYLENGTH-PREFREQ)/QUAYLENGTH;
00950 IF QUAYLENGTHDAYS>0 THEN DO;
00960 FOR I=1 TO QUAYLENGTHDAYS DO;
00970 TALLY DEGENG IN DEGREE_LEN;END;END;
00980 PREARRIVALTIME=NOW;PREFREQ=FREEQUAYLENGTH;
00990 HOLD SERVICETIME;QUAYLENGTHDAYS=(NOW-PREARRIVALTIME)*24;
01000 DEGENG=(QUAYLENGTH-PREFREQ)/QUAYLENGTH;
01010 IF QUAYLENGTHDAYS>0 THEN DO;
01020 FOR I=1 TO QUAYLENGTHDAYS DO;
01030 TALLY DEGENG IN DEGREE_LEN;END;END;
01040 FREEQUAYLENGTH=FREEQUAYLENGTH+SHIPLength;
01050 PREFREQ=FREEQUAYLENGTH;PREARRIVALTIME=NOW;
01060 IF HMASTER.STATUS=PASSIVE THEN REACTIVATE HMASTER;
01070 TERMINATE;
01080

```

```

01090
01100 /*****
01110 /***PROCES OF HARBOURMASTER***/
01120 /*****
01130 INSPECTION_ANCHORAGE;
01140 NEXTSHIP=ANCHORAGE.FIRST;
01150 WHILE NEXTSHIP]=NOCOMP DO;
01160 NEXTSHIP.ACTWAITINGTIME=NOW-NEXTSHIP.ARRIVALTIME;
01170 IF NEXTSHIP.MAXWAITINGTIME<NEXTSHIP.ACTWAITINGTIME
01180 THEN REACTIVATE NEXTSHIP;
01190 ELSE DO;
01200 IF FREEQUAYLENGTH>NEXTSHIP.SHIPLength THEN DO;
01210 REACTIVATE NEXTSHIP;
01220 FREEQUAYLENGTH=FREEQUAYLENGTH-NEXTSHIP.SHIPLength;
01230 END;
01240 END;
01250 NEXTSHIP=NEXTSHIP.SUCC;
01260 END;
01270 PASSIVATE;REPEAT FROM INSPECTION_ANCHORAGE;
01280
01290 END;
01300 /*
01310 //
END OF DATA

```

TIME= 0.0000  
MAIN IS CURRENT NOW  
HMASTER INITIALIZED  
CLASS SHIP INITIALIZED  
GENERATORSHI INITIALIZED  
ANCHORAGE INITIALIZED  
QUAY INITIALIZED  
HMASTER ACTIVATED AT 0.0000  
GENERATORSHI ACTIVATED AT 0.0000  
MAIN SUSPENDED UNTIL 350.0000

TIME= 0.0000  
HMASTER IS CURRENT NOW  
HMASTER IS PASSIVATED

TIME= 0.0000  
GENERATORSHI IS CURRENT NOW  
SHIP.....1 INITIALIZED  
SHIP.....1 ACTIVATED AT 0.0000  
GENERATORSHI SUSPENDED UNTIL 1.9470

TIME= 0.0000  
SHIP.....1 IS CURRENT NOW  
SHIP.....1 TO TAIL OF QUEUE(OF) ANCHORAGE  
HMASTER REACTIVATED AT 0.0000  
SHIP.....1 IS PASSIVATED

TIME= 0.0000  
HMASTER IS CURRENT NOW  
SHIP.....1 REACTIVATED AT 0.0000  
HMASTER IS PASSIVATED

TIME= 0.0000  
SHIP.....1 IS CURRENT NOW  
SHIP.....1 OUT OF QUEUE(OF) ANCHORAGE  
SHIP.....1 SUSPENDED UNTIL 2.7834

TIME= 1.9470

GENERATORSHI IS CURRENT NOW  
SHIP.....2 INITIALIZED  
SHIP.....2 ACTIVATED AT 1.9470  
GENERATORSHI SUSPENDED UNTIL 5.4200

TIME= 1.9470  
SHIP.....2 IS CURRENT NOW  
SHIP.....2 TO TAIL OF QUEUE(OFF) ANCHORAGE  
HMASTER REACTIVATED AT 1.9470  
SHIP.....2 IS PASSIVATED

TIME= 1.9470  
HMASTER IS CURRENT NOW  
HMASTER IS PASSIVATED

TIME= 2.7834  
SHIP.....1 IS CURRENT NOW  
HMASTER REACTIVATED AT 2.7834  
PROCESS OF SHIP.....1 IS TERMINATED

TIME= 2.7834  
HMASTER IS CURRENT NOW  
SHIP.....2 REACTIVATED AT 2.7834  
HMASTER IS PASSIVATED

TIME= 2.7834  
SHIP.....2 IS CURRENT NOW  
SHIP.....2 OUT OF QUEUE(OFF) ANCHORAGE  
SHIP.....2 SUSPENDED UNTIL 4.8151

TIME= 4.8151  
SHIP.....2 IS CURRENT NOW  
HMASTER REACTIVATED AT 4.8151  
PROCESS OF SHIP.....2 IS TERMINATED

TIME= 4.8151  
HMASTER IS CURRENT NOW  
HMASTER IS PASSIVATED

TIME= 5.4200  
GENERATORSHI IS CURRENT NOW  
SHIP.....3 INITIALIZED  
SHIP.....3 ACTIVATED AT 5.4200  
GENERATORSHI SUSPENDED UNTIL 11.0215

TIME= 5.4200  
SHIP.....3 IS CURRENT NOW  
SHIP.....3 TO TAIL OF QUEUE(OF) ANCHORAGE  
HMASTER REACTIVATED AT 5.4200  
SHIP.....3 IS PASSIVATED

TIME= 5.4200  
HMASTER IS CURRENT NOW  
SHIP.....3 REACTIVATED AT 5.4200  
HMASTER IS PASSIVATED

TIME= 5.4200  
SHIP.....3 IS CURRENT NOW  
SHIP.....3 OUT OF QUEUE(OF) ANCHORAGE  
SHIP.....3 SUSPENDED UNTIL 7.7823

TIME= 7.7823  
SHIP.....3 IS CURRENT NOW  
HMASTER REACTIVATED AT 7.7823  
PROCESS OF SHIP.....3 IS TERMINATED

TIME= 7.7823  
HMASTER IS CURRENT NOW  
HMASTER IS PASSIVATED

TIME= 11.0215  
GENERATORSHI IS CURRENT NOW  
SHIP.....4 INITIALIZED  
SHIP.....4 ACTIVATED AT 11.0215  
GENERATORSHI SUSPENDED UNTIL 14.2739

TIME= 138.5404  
HMASTER IS CURRENT NOW

SHIP.....59 REACTIVATED AT 138.5404  
SHIP.....60 REACTIVATED AT 138.5404  
SHIP.....61 REACTIVATED AT 138.5404  
SHIP.....62 REACTIVATED AT 138.5404  
HMASTER IS PASSIVATED

TIME= 138.5404  
SHIP.....59 IS CURRENT NOW  
SHIP.....59 OUT OF QUEUE(OFF) ANCHORAGE  
PROCESS OF SHIP.....59 IS TERMINATED

TIME= 138.5404  
SHIP.....60 IS CURRENT NOW  
SHIP.....60 OUT OF QUEUE(OFF) ANCHORAGE  
PROCESS OF SHIP.....60 IS TERMINATED

TIME= 138.5404  
SHIP.....61 IS CURRENT NOW  
SHIP.....61 OUT OF QUEUE(OFF) ANCHORAGE  
PROCESS OF SHIP.....61 IS TERMINATED

TIME= 138.5404  
SHIP.....62 IS CURRENT NOW  
SHIP.....62 OUT OF QUEUE(OFF) ANCHORAGE  
PROCESS OF SHIP.....62 IS TERMINATED

TIME= 138.8993  
GENERATORSHI IS CURRENT NOW  
SHIP.....63 INITIALIZED  
SHIP.....63 ACTIVATED AT 138.8993  
GENERATORSHI SUSPENDED UNTIL 139.2629

TIME= 138.8993  
SHIP.....63 IS CURRENT NOW  
SHIP.....63 TO TAIL OF QUEUE(OFF) ANCHORAGE  
HMASTER REACTIVATED AT 138.8993  
SHIP.....63 IS PASSIVATED

STATUS OF THE SYSTEM

CLOCKTIME= 350.0000  
 MAIN CURRENT COMPONENT  
 HMASTER PASSIVE COMPONENT  
 GENERATORSHI EVENTTIME= 350.2676

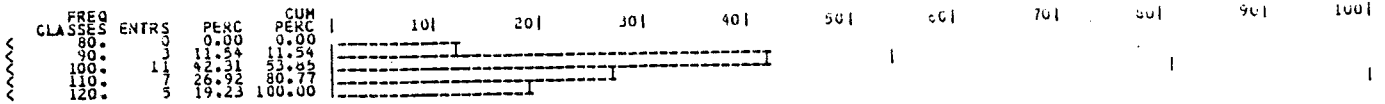
MEMBERS OF CLASS SHIP ARE:  
 SHIP.....163 ARRIVALTIME= 346.0425 EVENTTIME= 351.0308  
 SHIP.....164 ARRIVALTIME= 348.4653 PASSIVE COMPONENT ARRIV  
 ED AT 348.4653 IN ANCHORAGE  
 SHIP.....165 ARRIVALTIME= 349.2559 PASSIVE COMPONENT ARRIV  
 ED AT 349.2559 IN ANCHORAGE

DATA ABOUT QUEUES

NAME	ENTRIES	ZEROENTRIES	LENGTH	MAX.LENGTH	MEAN WAITINGT	MAX.W
AITINGT						
ANCHORAGE	165	57	2	5	1.3382	
5.0680						
QUAY	0	0	0	0	NILL	
NILL						

ONE DAY WAITINGSHIPS

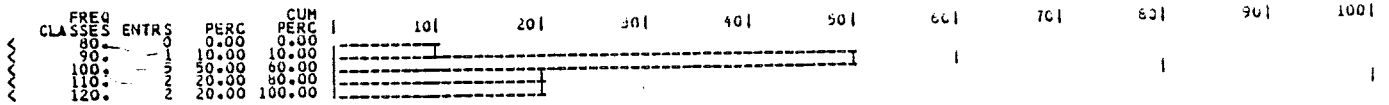
ENTRIES 26  
 ZERO ENTRIES 0  
 MEAN 100.7449  
 STD.DEVIATION 9.20095  
 MINIMUM 83.8629  
 MAXIMUM 115.825



REMAINDER EMPTY

TWO DAYS WAITINGSHIPS

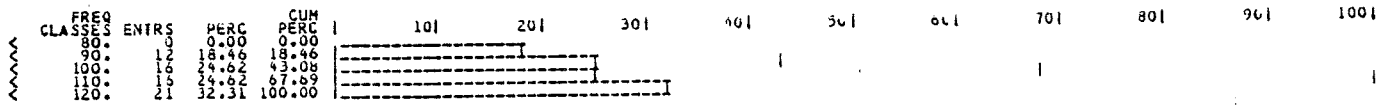
ENTRIES 10  
 ZERO ENTRIES 0  
 MEAN 98.468  
 STD.DEVIATION 8.9315  
 MINIMUM 84.7869  
 MAXIMUM 112.9723



REMAINDER EMPTY

GIVINGWAY SHIPS

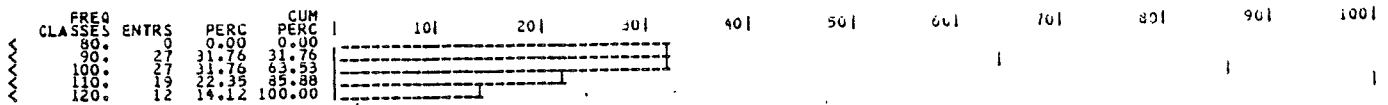
ENTRIES 65  
 ZERO ENTRIES 0  
 MEAN 102.925  
 STD.DEVIATION 11.1425  
 MINIMUM 82.9023  
 MAXIMUM 119.8307



REMAINDER EMPTY

DIRECT HANDLINGSHIPS

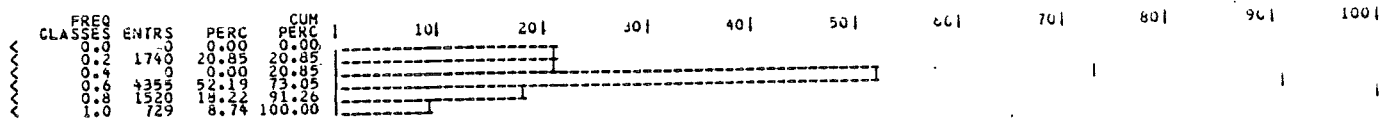
ENTRIES 85  
 ZERO ENTRIES 0  
 MEAN 96.7896  
 STD.DEVIATION 10.51742  
 MINIMUM 80.8639  
 MAXIMUM 119.3189



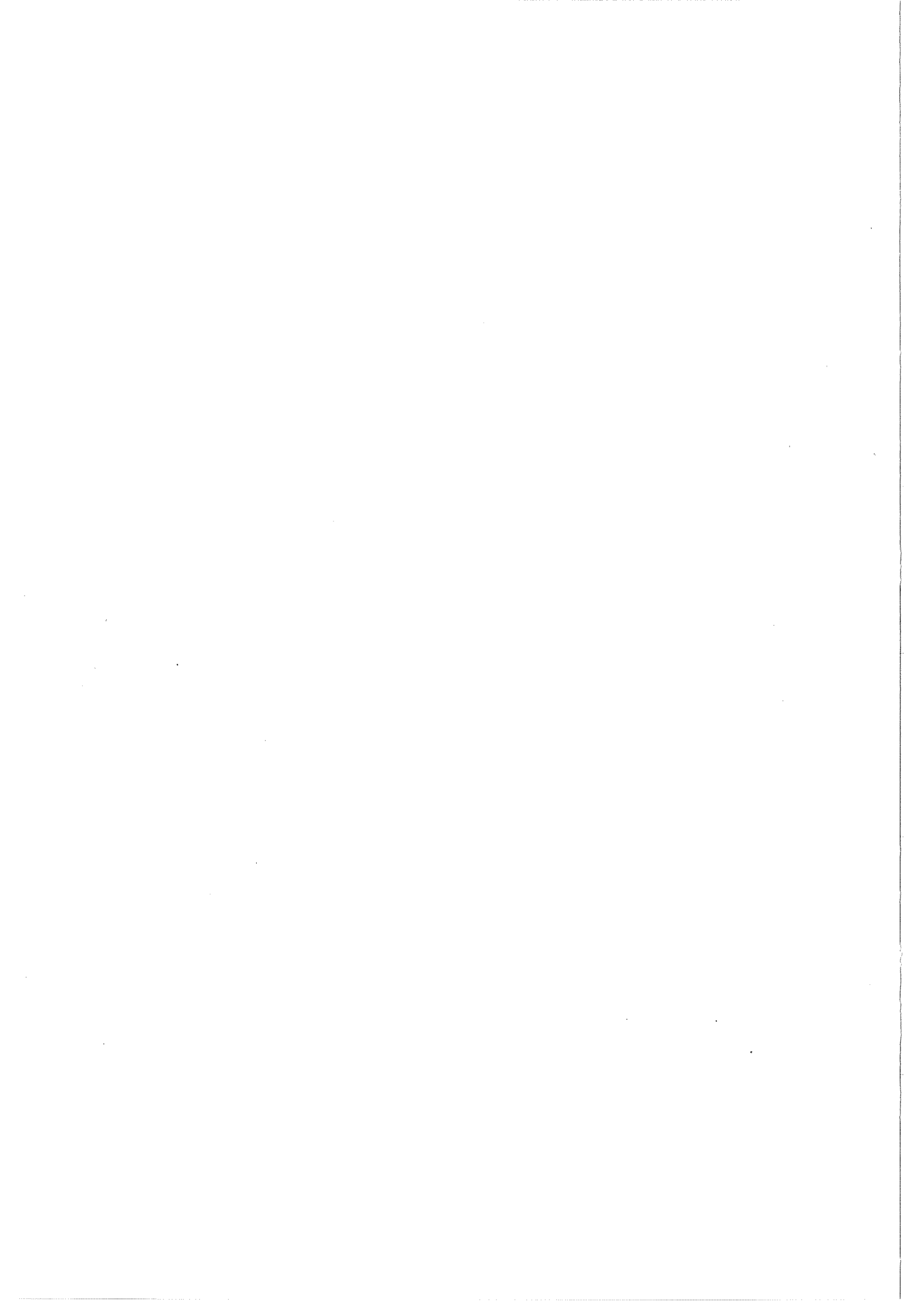
REMAINDER EMPTY

DEGREE ENGAGEMENT

ENTRIES 8344  
 ZERO ENTRIES 1740  
 MEAN 0.479306  
 STD.DEVIATION 0.2750580  
 MINIMUM 0.  
 MAXIMUM 0.998298



REMAINDER EMPTY



### 3. Verkeerssimulatiemodel haven van Zeebrugge

Met het doel een beeld te geven van de mogelijkheden van dit type simulatie zal nu het verkeerssimulatiemodel van Zeebrugge, ontwikkeld bij de vakgroep Waterbouwkunde door de studenten de heren P.H. Hiddinga en C.J. Bunschoten, worden behandeld.

#### 3.1 De geschiedenis van de haven van Zeebrugge kan als volgt worden gekarakteriseerd:

1855 Op initiatief van koning Leopold II is in 1855 besloten een nieuwe zeehaven te bouwen aan de kust vlak voor Zeebrugge. De kunstwerken zijn voltooid in 1907 en bestonden uit: een havendam, een havengeul, vissershaven, zeesluis, ferrydok en het Boudewijnkanaal (verbinding Brugge-Zeebrugge).

1950-1960 In deze periode zijn enkele verbeteringen uitgevoerd, waaronder de voornaamste:

- a. nieuwe terminal voor ferryschepen
- b. uitbreiding van de vissershaven
- c. industriehaven (het Prins Filipdok)
- d. indijking westbank met een oliekade

1960-1970 In de periode 1960-1970 is de haven uitgebreid en geschikt gemaakt voor olietankers van max. 50.000 dwt. Hiertoe is een oostelijke havendam gerealiseerd, hetgeen een rustiger ligging en minder onderhouds- en baggerwerk betekende en is het westhoofd verder uitgebouwd. De volledige voltooiing van het westhoofd is gerealiseerd in 1971, waarbij de westelijke kademuur is uitgerust als een moderne diepzee containerterminal, en de oostelijke kademuur als olieloscade; overal rond het westhoofd is een diepte van 13.00 m beneden laagwater gerealiseerd. De Pas van het Zand en de geul door het Ribzand zijn verdiept tot 9.30 m beneden laagwater met een bodembreedte van 300 m. Ook de voorhaven werd verdiept tot 9.30 m waaronder de zwaikom met een diameter van 500 m. Verder zijn in deze periode gerealiseerd:

- a. kademuur t.b.v. een shortsea containerterminal
- b. een roll on roll off terminal
- c. verbindingsweg tussen de shortsea terminal en de deepsea terminal
- d. een militair dok

1970 In 1970 is door de Belgische regering beslist tot een verdere uitbouw van de haven van Zeebrugge, de beslissing werd genomen op grond van een studie omtrent de mogelijkheden van een nieuwe haven "in volle zee of aan de Belgische kust voor schepen met een grote tonnenmaat". De uitvoering geschiedt verspreid over een aantal jaren en in een aantal fasen. Het project omvat een beschermd open buitenhaven, een nieuwe binnenhaven (ca. 1.000 ha), en de constructie van een nieuwe zeesluis.

1972 De eerste fase omvat de nieuwe zeesluis, de nieuwe achterhaven en een uitbouw van de buitenhaven. De bouwactiviteiten van deze fase zijn gestart in 1972.

1976 In 1976 is een raamovereenkomst afgesloten voor de havendammen met de Tijdelijke Vereniging Zeebouw Zeezand, welke groep bestaat uit een bouwfirma en een studiebureau. Het eerste deelcontract betreft een studiecontract waarin de hydraulische, nautische en sedimentologische aspecten zijn opgenomen.

1977 Het tweede deelcontract omvat strandverbeteringswerken t.p.v. Knokke-Heist.

1978 Het volgende deelcontract betreft de werkhaven, de aanzet van de oostelijke havendam en het westelijk werkterrein.

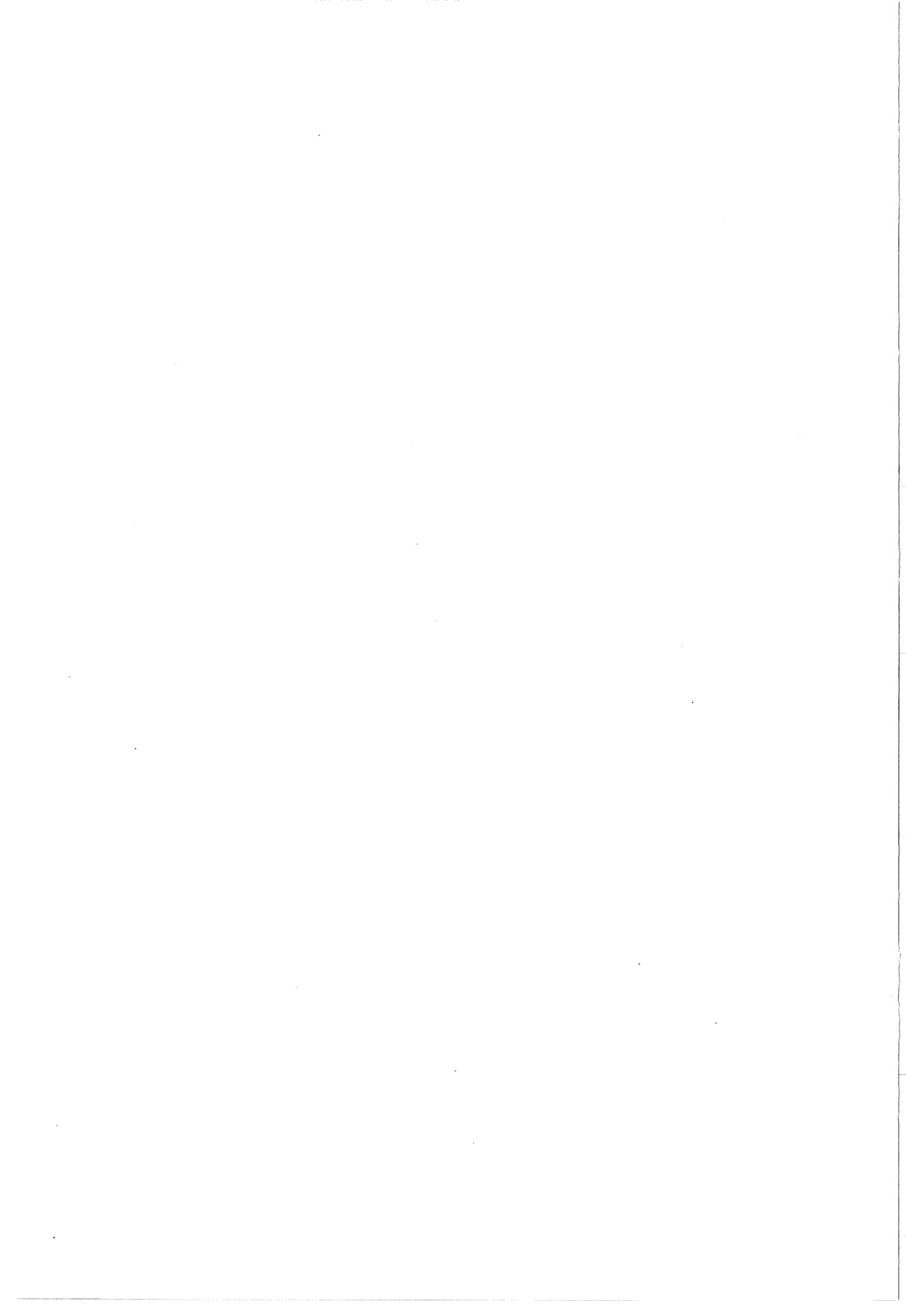
1979 De voortzetting van de oostelijke havendam tot 1.300 m vanaf de kust is in 1979 in een deelcontract vastgelegd.

1985 Het ligt in de bedoeling om in 1985 de L.N.G.-terminal in gebruik te nemen.

### 3.2 Doelstelling verkeerssimulatiemodel

De doelstelling van het verkeerssimulatiemodel van Zeebrugge kan in eerste instantie als volgt worden geformuleerd.  
Het registreren van de wachttijden welke optreden bij een gegeven verkeersprognose gedurende de periode heden tot het jaar 2000. Hierbij wordt aandacht besteed aan de volgende punten:







De randvoorwaarden die via een dataset gewijzigd kunnen worden zijn:

- het weerbeeld over een jaar (de actuele weercijfers van het jaar 1979 zijn in eerste instantie gebruikt)
- getijkrommen
- bestaande/toekomstige lay-out
- vaardiepte Scheur en Pas van het Zand
- simulatieduur, subrun duur
- voor LNGC, VLCC, Bulkcarriers en 2e en 3e generatie containerschepen: een tussenaankomsttijdverdeling en een servicetijdverdeling
- voor bulkcarriers, 2e en 3e generatie containerschepen: grenzen uniform gekozen diepgangverdeling, vaartijden en percentages betreffende de verschillende bestemmingen  
(opmerking: de diepgang van de VLCC is afhankelijk van de maatgevende diepte bij het eerstvolgende hoogwater)

#### 3.4 Beschrijving van het model middels de procesmethode

Overeenkomstig het gegeven voorbeeld van de kade begint ook dit model met het proces van de component main. Het proces kan worden onderverdeeld in:

##### A. Definitie sectie waarin de permanente componenten

1. het weer
2. de loods
3. havenmeester
4. sluismeester
5. de generatoren voor LNGC, tweede generatie containerschepen, derde generatie containerschepen, VLCC en bulkcarriers worden geïnitieerd

Verder wordt de component klasse van schepen met de attributen:

1. diepgang ingaand schip
2. diepgang uitgaand schip
3. maximale dwarsstroom
4. scheepslengte
5. scheepsklasse
6. bestemming
7. afkomst
8. vaartijden voor de verschillende onderdelen van het vaartraject
9. prioriteitscoëfficiënt

gedefinieerd en tenslotte de histogrammen voor wachttijden en bezettingsgraad van de verschillende kaden.

B. Declaraties en initialisaties van diverse toestandvariabelen.

C. Verzorging van de uitvoer en Runcontrol.

Daarna volgt het proces van:

2. de generatoren voor de vijf genoemde schepen
3. het proces van het weer
4. het proces van de loods met als belangrijkste onderdelen:
  - a. bepalen van de curven voor het horizontaal en verticaal getij (zie ook bijlage 10)
  - b. vaststellen welk schip aan de criteria om te mogen varen voldoet betreffende schepen die bij de aankomstboei liggen
5. het proces van de havenmeester met belangrijkste onderdeel het vaststellen of voldaan wordt aan de "uitvaarcriteria" betreffende schepen aan de kade waarvan de servicetijd is verstreken
6. proces van de schepen waarin ondermeer de regeling van de fictieve stoplichten van de vaarwegonderdelen is opgenomen
7. het proces van de sluismeester die de doorgang van schepen door de sluis naar achterhaven regelt (bulkcarriers)

De meest karakteristieke processen van dit model zijn het proces van de havenmeester, de loods en de schepen. Mede gezien de beperkte tijd zullen daarom slechts deze processen hier worden behandeld.

#### 3.4.1 Proces van de havenmeester en de loods (zie bijlage 10, fig. 3.3, 3.4, 3.5)

Teneinde de gecompliceerdheid van een dergelijk verkeerssimulatiemodel aan te geven zal in grote lijnen het proces van de havenmeester en het proces van de loods worden besproken. De havenmeester en loods regelen in nauw overleg het scheepvaartverkeer van en naar de haven. Om tot een zo goed mogelijke benadering van de realiteit te komen zijn de regels die in dit model worden gehanteerd met betrekking tot de havenmeester en de loods (en ook voor de andere processen) opgesteld in samenwerking met het eerdergenoemde studiebureau. In de realiteit worden deze regels in feite toegepast zonder dat ze duidelijk zijn omschreven; wanneer echter de realiteit met behulp van een simulatiemodel wordt nagespeeld moeten de spelregels eenduidig vastliggen.



Schepen die op de ankerplaats liggen te wachten om de haven te mogen aandoen worden in de wachtrij van de loods geplaatst. Evenzo worden schepen die de haven willen verlaten in de wachtrij van de havenmeester gezet. Bij het vaststellen van de wachttijd discipline kan niet worden uitgegaan van het FIFO-principe. In het model komen vijf verschillende scheepstypen voor waarvan sommige aan een strak schema zijn gebonden (LNGC en containerschepen) terwijl voor andere typen, zoals de VLCC en bulkcarriers een wat langere wachttijd minder bezwaarlijk is.

Een prioriteitscoëfficiënt en de tijd die is doorgebracht in de wachtrij bepaalt de prioriteit die aan het betreffende schip wordt toegekend. Voor de in het model voorkomende schepen is uitgegaan (bij dezelfde wachttijd in de wachtrij) van de volgende voorrangsvolgorde:

1. LNG-carriers
2. 2e en 3e generatie containerschepen
3. VLCC en bulkcarriers

Een door de generator gecreëerd schip wordt in de wachtrij van de loods geplaatst en activeert de loods. De loods gaat nu na voor dat eventtjdstip hoe de getijomstandigheden zijn en de weersgesteldheid is. Wanneer de weersgesteldheid niet gunstig is schort de havenmeester zijn werkzaamheden op tot dat dit wel het geval is. De loods beschouwt, indien de weersomstandigheden wel gunstig zijn, het eerste schip in zijn wachtrij, onderzoekt of dit schip aan de vaarcriteria voldoet en zo ja, dan wordt wanneer het geen LNGC betreft, de prioriteit berekend. De loods neemt hierna het volgende schip in zijn wachtrij en herhaalt zijn werkzaamheden vanaf de toetsing van vaarcriteria. Op deze manier wordt de hoogste prioriteit vastgesteld van schepen die in de wachtrij van de loods liggen. De loods activeert nu de havenmeester, die op zijn beurt op overeenkomstige wijze het schip vaststelt met de hoogste prioriteit in de wachtrij van de havenmeester. Een hulpgetal X krijgt de waarde van de hoogste prioriteit (van alle schepen in deze twee wachtrijen) en het schip met deze prioriteit wordt vervolgens geactiveerd.

Aangenomen dat het geactiveerde schip in de wachtrij van de loods ligt, wordt nadat geconstateerd is, dat het aantal schepen in de wachtrij van de havenmeester groter dan nul en een aantal hulpgetallen opnieuw hun

startwaarde krijgen, de havenmeester geactiveerd en de loods, die in dit geval het schip heeft geactiveerd, gepassiveerd. Het schip met de hoogste prioriteit wordt uit de rij van de havenmeester gehaald en (indien het niet een schip betreft dat de sluis moet passeren) wordt het schip geactiveerd. Daar, zoals verondersteld is, reeds een schip bij de loods is vertrokken en er nog schepen in de rij van de loods liggen, wordt nu na 15 minuten (dit om oplopen te voorkomen) de loods weer geactiveerd. Eenzelfde redenering kan ook worden gehouden indien de havenmeester het eerste schip activeert. Het proces is dus zo gebouwd, dat op hetzelfde moment slechts twee schepen geactiveerd kunnen worden, n.l. één door de havenmeester en één door de loods. Echter zoals bij de meeste van dit soort processen zijn het de uitzonderingen die de zaak gecompliceerd maken. Zo ook bij dit proces:

- Uitzonderingen worden gemaakt het betrekking tot LNGC. Voor LNGC heeft geen prioriteit te worden berekend, daar deze schepen onder alle omstandigheden de hoogste prioriteit verkrijgen. Indien door de loods wordt geconstateerd, dat een LNGC in de loodsrij aanwezig is, worden zijn vaarcriteria getoetst:

1. horizontaal getij
2. verticaal getij
3. fictieve stoplichten en als extra
4. kade vrij

Voldoet nu een LNGC aan alle vaarcriteria behalve het kadecriterium, dan wordt de havenmeester geactiveerd. Bevindt zich nu een LNGC in de wachtrij van de havenmeester, dan wordt dit schip (indien het voldoet aan de vaarcriteria) geactiveerd.

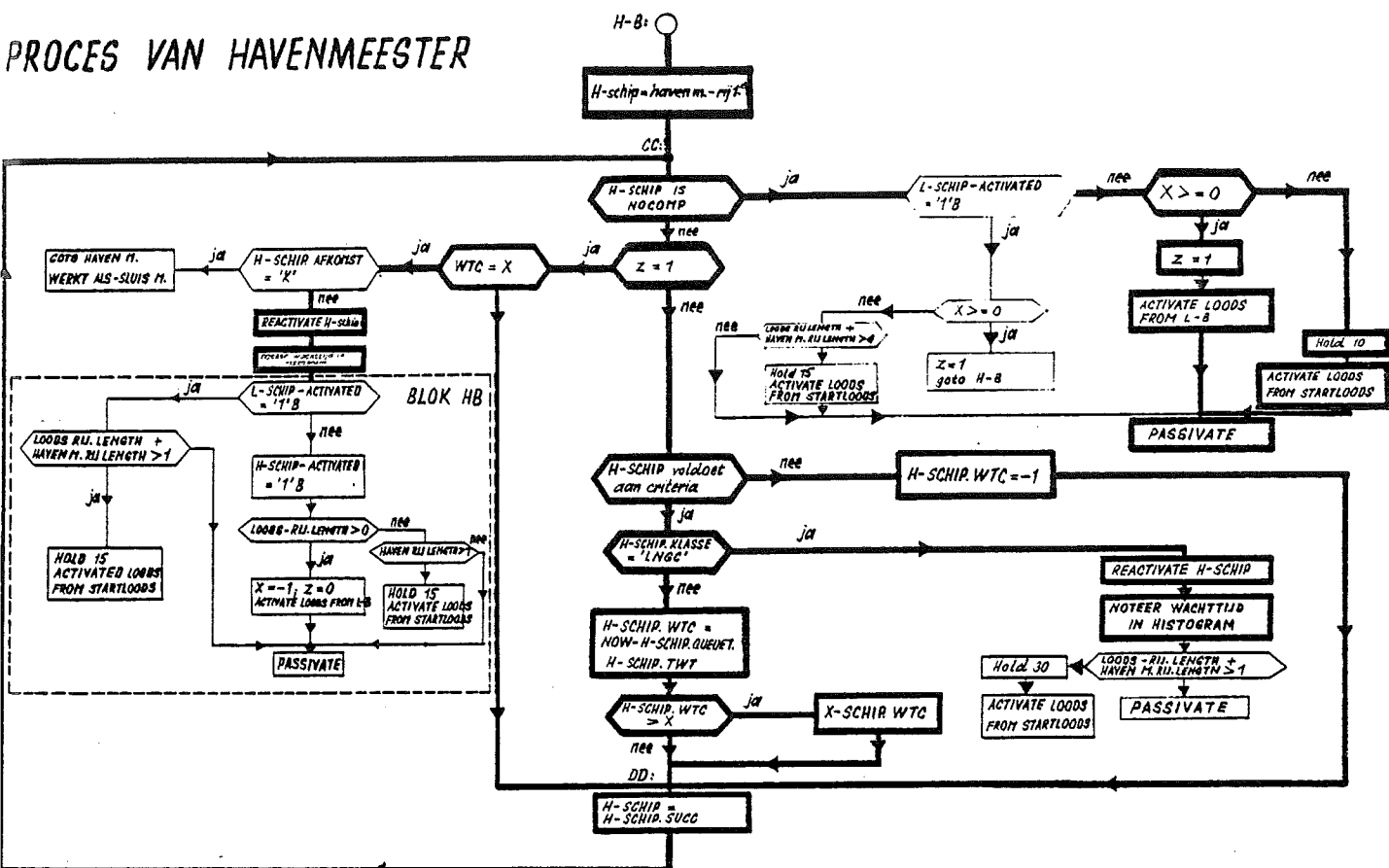
- Ook voor de VLCC moet een extra vaarcriterium worden aangebracht. Nadat in het programma is komen vast te staan dat het VLCC-schip de hoogste prioriteit heeft gekregen, wordt ook hier getest of de kade, die slechts bestaat uit één aanlegplaats, vrij is. Is dit niet het geval, dan krijgt het VLCC-schip, dat aan de kade ligt, wanneer het aan de vaarcriteria voldoet (hierbij wordt niet gelet op de prioriteit) opdracht om te vertrekken. Voldoet het VLCC-schip aan de kade niet aan de vaarcriteria, dan wordt het schip met de op één na hoogste prioriteit in de twee wachtrijen bepaald en onder bepaalde voorwaarden geactiveerd.

- Voor bulkcarriers wordt ook, gezien het bestaan van slechts één aanlegplaats voor deze schepen, de test "kade vrij" uitgevoerd. Het verschil met de VLCC-schepen ligt hierin, dat indien de kade bezet is en de aan kade liggende bulkcarrier voldoet aan de vaarcriteria, dit schip geen opdracht krijgt de haven te verlaten, wanneer zijn prioriteit niet de op één na hoogste is. De reden hiervoor is, dat de aanlegplaats voor bulkcarriers achter de sluis ligt en de daarmee gepaard gaande lange vaartijd.
  
- Het model test met betrekking tot de containerschepen in tegenstelling tot bovenstaande niet of de kade (aantal aanlegplaatsen bedraagt  $\pm 7$ ) geheel bezet is, daar dit volgens de verkregen gegevens niet noodzakelijk is. Wel wordt een controle uitgevoerd door te registreren de bezettingsgraad van de kade, zodat onmiddellijk blijkt op welk tijdstip deze aanname niet meer juist is (bezettingsgraad wordt dan groter dan 100%). Bij containerschepen wordt echter op een andere manier van de algemene regel afgeweken en wel onder de volgende omstandigheden:
  1. wanneer het schip met de hoogste prioriteit in de wachtrij van de loods ligt met bestemming H (zie fig. 3.3);
  2. het schip met de op één na hoogste prioriteit in de wachtrij van de loods een containerschip is met bestemming I of K.

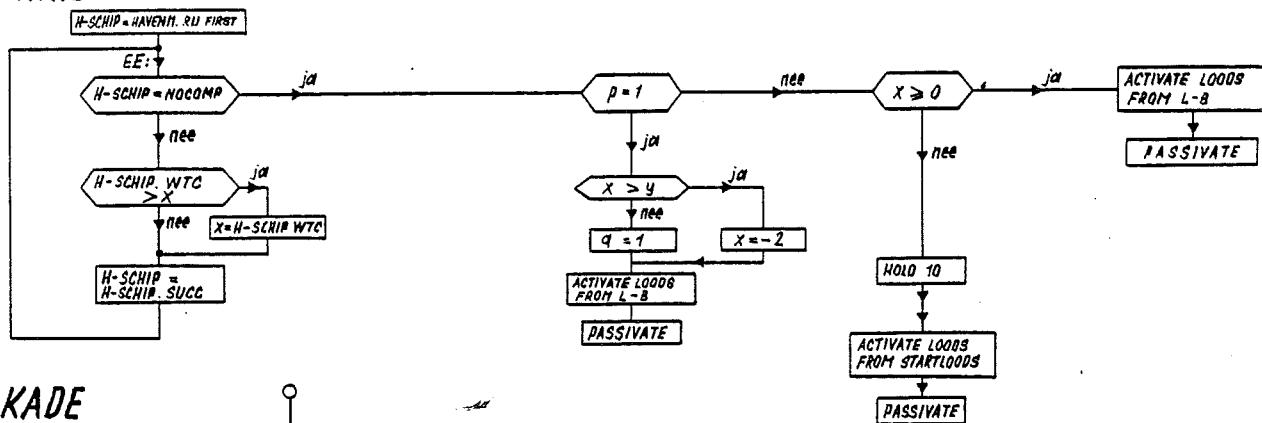
In dat geval krijgt het containerschip met bestemming I of K opdracht eerst te vertrekken. Dit betekent met betrekking tot dit containerschip een tijdwinst van 20 à 30 minuten, de tijd die in de draaicirkel E wordt doorgebracht, vermeerderd met de 15 minuten die in acht worden genomen in verband met het oplopen van schepen. Het schip met bestemming H boekt slechts een verlies van 15 minuten. Deze verandering van voorrang wordt slechts toegepast voor containerschepen, daar de winst van 20 à 30 minuten voor VLCC-schepen en Bulkcarriers teniet zou worden gedaan door hun lagere vaarsnelheid.



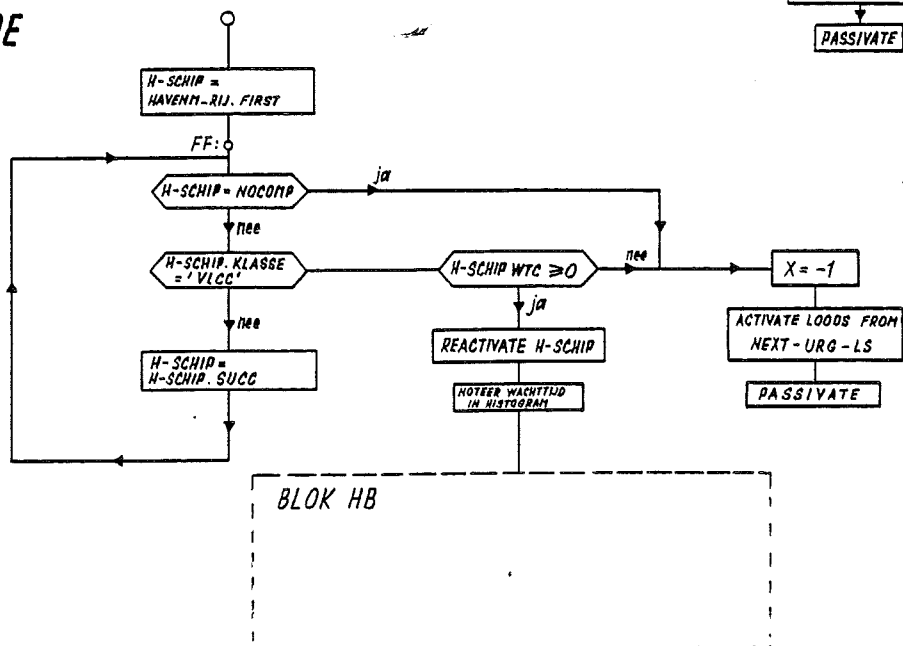
# PROCES VAN HAVENMEESTER



## NEXT-URG-HMS



## VLCC-aan-KADE



### 3.4.2 Het proces van de schepen

De procesbeschrijving van de schepen kan het gemakkelijkst worden toegelicht m.b.v. het bijgevoegde diagram.

Enkele opmerkingen bij het diagram:

#### 1. "HOLD VAARTIJD + SERVICETIJD"

Hier wordt onder verstaan: Vaartijd over het Scheur + vaartijd over de Pas van het Zand + vaartijd in de haven + los- en laadtijd aan de kade (deze tijden worden als attributen meegegeven aan de component schip).

#### 2. "HOLD VAARTIJD"

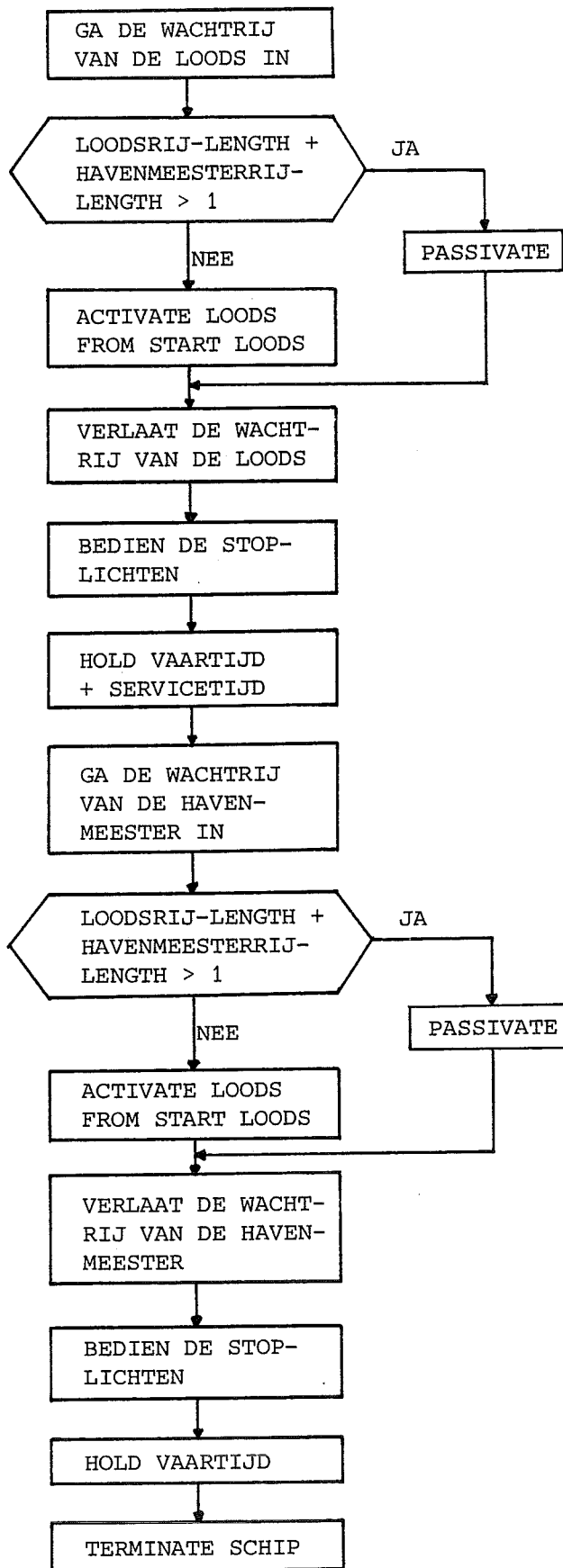
Deze opdracht wordt gegeven nadat het schip toestemming heeft gekregen de haven te verlaten.

De "VAARTIJD" bestaat uit: "Trossen losmaken tijd + verblijftijd in de draaicirkel + vaartijd in de haven + vaartijd in de Pas van het Zand + vaartijd in het Scheur.

Het diagram spreekt verder voor zichzelf.

1x Procesbeschrijving van het schip x1

"START SCHIP"



### 3.5 Enige invoergegevens

Ter oriëntatie zullen enige randvoorwaarden betreffende de situatie in het jaar 1979 worden gegeven.

Dit jaar is gekozen om het model te ijken.

1. Waterdiepte van het Scheur t.o.v. GLLWS bedraagt 11.50 m en van de Pas van het Zand 11.00 m.
2. VLCC:
  - a. tussenaankomsttijden zijn negatief exponentieel verdeeld met een gemiddelde 216 uur
  - b. servicetijd is k-erlang verdeeld met een gemiddelde van 33.00 uur (k=4)
  - c. toelaatbare dwarsstroom = 1.15 m/s
  - d. de diepgang is afhankelijk van het getij
  - e. prioriteitscoëfficiënt = 24
3. Tweede generatie containerschepen:
  - a. tussenaankomsttijd is negatief exponentieel verdeeld met gemiddelde 40.00 uur
  - b. bestemming kade I 100%
  - c. servicetijd is k-erlang verdeeld met k = 8 en gemiddelde 14,7 uur
  - d. toelaatbare dwarsstroom = 1,73 m/s
  - e. prioriteitscoëfficiënt = 2
  - f.
4. Derde generatie containerschepen:
  - a. tussenaankomsttijden negatief exponentieel verdeeld met gemiddelde 125.00 uur
  - b. bestemming kade I 100%
  - c. servicetijd is k-erlang verdeeld met gemiddelde 21.30 (k = 8)
  - d. diepgang is uniform verdeeld op het traject 8.20 - 10.80 m
  - e. prioriteitscoëfficiënt = 2
  - f. toelaatbare dwarsstroom = 1.73 m/s
5. Met betrekking tot het weer zijn de actuele weergegevens van het jaar 1979 ingevoerd.

6. De horizontale en verticale getijkrommen van doodtij en springtij zijn over 12.25 geschematiseerd en middels 15 punten met een tussentijd van 50 minuten ingevoerd.

#### 4. Optimalisatie met behulp van een simulatiemodel

Over het algemeen wordt een optimaal functionerende haven gedefinieerd als een haven waarvan het kostenniveau, wanneer alle relevante kosten in rekening worden gebracht, het minimum hebben bereikt. Opschoon dit een gevoelige doelstelling is, zijn er andere mogelijk.

Een andere definitie van een optimale haven zou kunnen zijn een haven waar met een minimum aan kosten de vertragingen van schepen door het bezet zijn van de kade niet voorkomen. Dit laatste optimum komt overeen met het eerste wanneer scheepsvertragingen altijd kostbaarder zijn dan een uitbreiding in faciliteiten. Over het algemeen wordt het optimum "minimum aan kosten" gehanteerd.

Het optimum "minimum aan kosten" is afhankelijk vanuit welke optiek het systeem wordt beschouwd. Wordt het systeem benaderd vanuit de optiek van de rederij, dan zal worden aangedrongen op investeringen die de wachttijden van schepen minimaliseren. Havenbedrijven zullen aandringen op investeringen die goederenopslag en overslag zo efficiënt mogelijk laten verlopen. Een derde partij, de havenbeheerder, is tenslotte gebaat bij een hoge opbrengst van havengelden, pachten en huren van loodsen en terreinen enz.

Een optimalisatie legt het verband tussen de technische en administratieve veranderingen (systeemdeel veranderingen) en de directe en indirecte kosten en baten. Het optimum wordt bereikt door een aantal systeemdelen te wijzigen, zodat de haven op een minimumniveau aan kosten functioneert.

Het probleem dat zich hierbij voordoet is dat er in principe "oneindig" veel combinaties van veranderingen in deelsystemen zijn samen te stellen. Het aantal mogelijkheden wordt beperkt door:

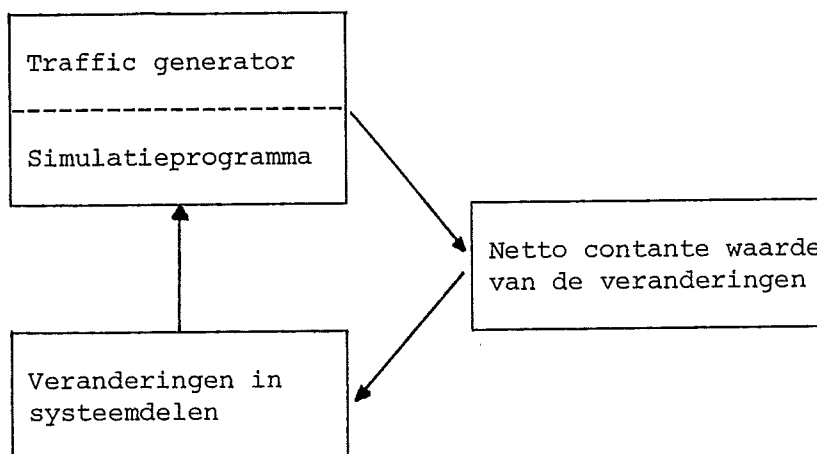
1. Technisch onmogelijke combinaties te elimineren (een haven kan niet uitbreiden t.g.v. de bestaande infrastructuur).
2. Het elimineren van onlogische combinaties, het genereren van grotere schepen zonder dat hiervoor b.v. de nautische aanpassingen worden uitgevoerd.
3. Budgetbeperkingen elimineren een aantal mogelijkheden m.b.t. de investeringsprogramma's.

Drie soorten kosten optimalisaties kunnen worden onderscheiden, te weten: het statisch optimum, het quasi-dynamisch optimum en het dynamisch optimum.

#### Het statisch optimum

Door marginale veranderingen in havenfaciliteiten en havenorganisatie worden kleine bottlenecks opgeheven. De randvoorwaarden van het systeem, zoals aankomstpatronen van schepen, zijn bij het bepalen van het statisch optimum niet aan een trendmatige verandering onderhevig.

De optimalisatie m.b.t. een simulatiemodel kan nu schematisch als volgt worden voorgesteld:



Het simulatiemodel registreert de consequenties van de veranderingen aangebracht in de systeemdelen. Deze veranderingen worden uitgedrukt in tijd en middels tijdfuncties omgezet in geld. Hieruit volgt dan de netto contante waarde van de investering waarmee de effectiviteit van de investering kan worden getoetst.

Voorbeelden van veranderingen in systeemdelen kunnen zijn:

- a. een wijziging in de procedure m.b.t. het in- en uitklaren van de lading waardoor onnodige congestie wordt vermeden;
- b. een vergroting van de overslagcapaciteit op de kade waarmee wachttijden van schepen wordt verkleind;
- c. een vergroting van de capaciteit van de loodsdienst zodat de wachttijden op de ankerplaats worden verminderd.

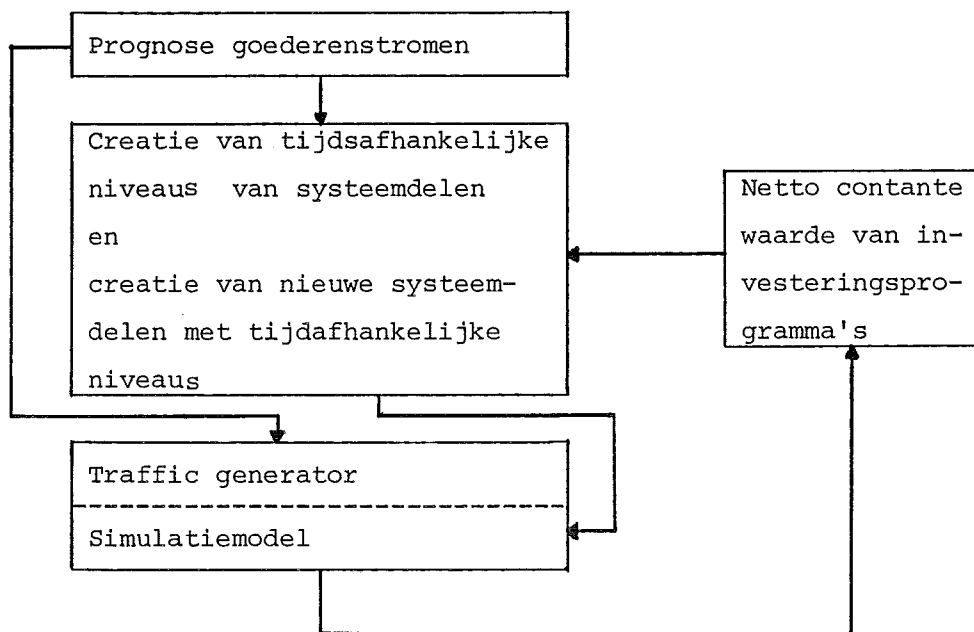
Via een optimalisatieprocedure dient nu de optimale combinatie van veranderingen in de systeemdelen te worden geselecteerd.

#### Het quasi-dynamisch optimum

Het quasi-dynamisch optimum wordt gedefinieerd als het aanpassen van de haven aan trendmatige ontwikkelingen, zodat de haven weer optimaal functioneert. Het verschil met het dynamisch optimum ligt hierin dat de randvoorwaarden (scheepsaanbod en goederenstromen) voor een toekomstig jaar zijn aangepast overeenkomstig de trendmatige ontwikkeling en dat nu relatief grote veranderingen in de systeemdelen zullen worden aangebracht. Echter scheepstypen en overslagmethoden zullen nog ongewijzigd blijven. Een voorbeeld om aan de trendmatige ontwikkeling tegemoet te komen kan een uitbreiding in een stukgoedhaven zijn met een stukgoedkade met een haventerrein, of een verdieping en daarmee een verruiming van het getijvenster van de toegangseul.

#### Het dynamisch optimum

Bij het vinden van het dynamisch optimum wordt in feite getracht een formulering van het optimum te vinden wat een functie is van de tijd. Het dynamisch optimum anticipeert op trendmatige ontwikkelingen en structurele wijzigingen. De randvoorwaarden, zoals aankomstverdeling, zijn tijdsafhankelijk geworden. De niveaus waarop de systeemdelen werken zijn eveneens functies van de tijd geworden, terwijl op gezette tijden nieuwe systeemdelen worden aangebracht. Een en ander kan als volgt in schema worden gezet.



Het deel van het schema waarmee gemanipuleerd wordt is het blok "creatie van tijdsafhankelijke niveaus van systeemdelen en creatie van nieuwe systeemdelen met tijdsafhankelijke niveaus" of met andere woorden het programma van investeringen. Ieder investeringsprogramma levert een netto contante waarde.

Via een optimalisatieprocedure dient nu het optimale investeringsprogramma te worden geselecteerd. Het probleem dat ook hier optreedt is dat in principe oneindig veel investeringsprogramma's zijn op te zetten; door eliminatie van mogelijkheden en het aftasten van de gevoeligheden wordt het aantal investeringsprogramma's beperkt.

