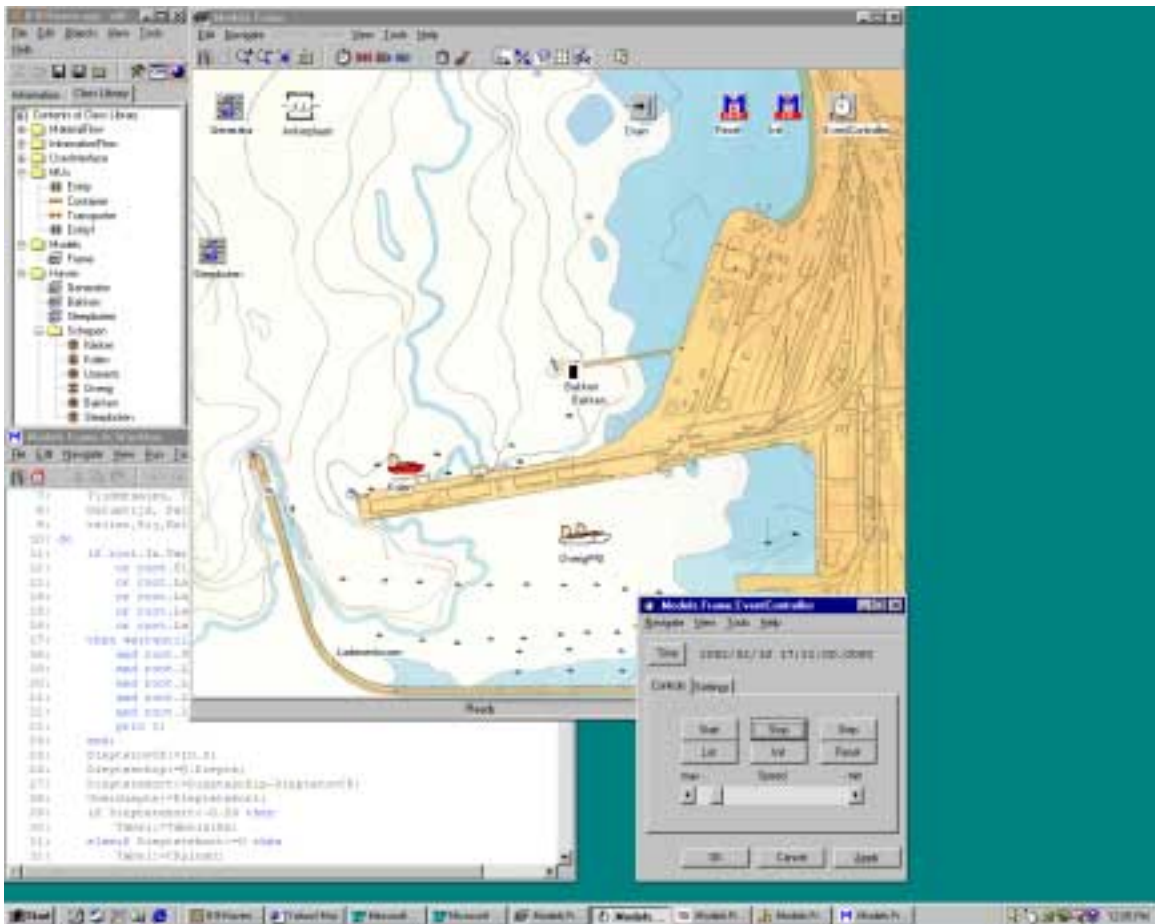


Concept rapport

Onderzoek naar een simulatiepakket voor de havenlogistiek



Voorwoord

Voor u ligt het afstudeerverslag van P.N. van Zijl. Dit verslag is geschreven naar aanleiding van het afstuderen aan de Technische Universiteit Delft bij de sectie "Havens en scheepvaartwegen" van de faculteit Civiele Techniek, in opdracht van Boskalis Westminster.

Boskalis Westminster is een internationaal opererend bedrijf dat zich heeft gespecialiseerd in het baggeren van alle mogelijke bodemmateriële. Eén van de afdelingen van het bedrijf is Hydronamic, het interne ingenieurbureau. Dit bureau is (onder andere) bezig met onderzoek naar havens en scheepvaartwegen, door de grote vraag naar baggerwerkzaamheden in deze sector. Hieruit ontstond de in dit rapport gestelde onderzoeksvraag: *Welk simulatiepakket is geschikt voor het simuleren van de vervoersstromen in hedendaagse havens en heeft dit pakket voldoende animatiemogelijkheden.* Het laatste deel van het rapport is gewijd aan een casestudie met het betreffende simulatieprogramma.

In dit voorwoord wil ik mijn begeleiders en de mensen van de faculteit TBM bedanken. Zonder uw medewerking was dit rapport niet tot stand gekomen. In het bijzonder wil ik de volgende mensen bedanken:

- Prof. ir. H. Ligteringen als hoofd van de afstudeercommissie.
- Ir R. Groenveld simulatie deskundige binnen de sectie havens en scheepvaartwegen.
- Ir. J.J. van de Looij, havenexpert binnen Hydronamic en initiator van het onderzoekproject.
- Ir T.W. Tewoldeberhan promovendus van systeemkunde binnen de faculteit Techniek Bestuur en Management die geholpen heeft met het programmeren van een simulatie met het programma eM-Plant.

Tevens wil ik de afdeling systeemkunde van de faculteit TBM bedanken voor het ter beschikking stellen van de computers, het simulatieprogramma en de benodigde expertise.

P.N. van Zijl
Papendrecht, november 2002



Samenvatting

Simulatie is in de laatste decennia met de opkomst van de computer een steeds belangrijker plaats gaan innemen bij het oplossen van complexe problemen. Met de komst van steeds snellere computers met betere grafische mogelijkheden is naast het oplossen van problemen ook het uitbeelden van de problemen steeds belangrijker geworden. De animatie wordt steeds vaker gebruikt om de opdrachtgever of het grote publiek te overtuigen van de kracht van simulatie. Door de complexe vervoersstromen in de hedendaagse havens heeft simulatie ook in deze sector een belangrijke plaats ingenomen. De naam van het simulatiepakket dat hier vaak voor gebruikt wordt is Prosim, een pakket dat zeer geschikt is voor het simuleren van de havenlogistiek. Ondanks dat het pakket zeer geschikt is voor de simulatie van havens zitten er een aantal nadelen aan het pakket. Een van de nadelen is de beperkte mogelijkheid tot het maken van animatie. Hierdoor is de onderzoeksvraag ontstaan:

Welk simulatiepakket is geschikt voor het simuleren van de vervoersstromen in hedendaagse havens en heeft dit pakket uitgebreide animatiemogelijkheden ?

Om tot de oplossing te komen van de vraagstelling zijn vijf pakketten door middel van een multi criteria analyse (mca) met elkaar vergeleken. Tevens is er een testcase in de vorm van een model met het gekozen pakket gemaakt. De onderzochte pakketten zijn:

1. eM-Plant (Tecnomatix)
- 1 Arena (Rockwell simulation software automation)
- 2 Automod (Brooks automation)
- 3 Prosim (Prosim B.V.)
- 4 Tomas (H.P.M. Veeke en J.A. Ottjes)

De volgorde van geschiktheid op basis van de gestelde criteria en de gewichten van deze criteria is weergegeven op basis van de rangorde van de bovenstaande tabel. De grootste verschillen tussen de verschillende pakketten zijn te vinden in animatie, flexibiliteit en documentatie.

De volgende opmerkingen zijn basis van het eerste deel van het rapport gemaakt:

- Als er in toekomst vaker een pakketkeuze gemaakt gaat worden is het belangrijk te weten dat: de keuze van een pakket is een delicate kwestie is. Het gevaar bestaat dat een programma gekozen wordt dat achteraf niet geschikt blijkt voor de gewenste modellering (dit maakt flexibiliteit een belangrijke pakketeigenschap).
- De keuze van een van de softwarepakketten wil niet zeggen dat de overige pakketten niet geschikt zijn. Het wil slechts zeggen dat met de gestelde criteria dit pakket het meest geschikt is.

Door middel van een casestudie is er volgens gekeken of het gekozen simulatiepakket daadwerkelijk geschikt is voor het maken van havensimulaties. De case studie betrof een haven waarin een aantal problemen spelen. Deze problemen worden vooral veroorzaakt door een gebrek aan capaciteit en een te geringe diepte. Voor de haven zijn een aantal alternatieven opgesteld:

- 1) Behouden van de huidige situatie (0-alternatief).
- 2) De aanleg van een extra aanlegplaats in combinatie met een vergroting van de diepte naar CD -11 meter bij twee van de vier aanlegplaatsen.



3) De aanleg van een extra aanlegplaats in combinatie met een vergroting van de diepte naar CD -13 meter bij twee van de vier aanlegplaatsen.

Deze alternatieven zijn vervolgens uitgewerkt in de vorm van simulatiemodellen. Uit de simulaties zijn de volgende conclusies gekomen.

- Door de aanleg van een extra aanlegplaats en het verdiepen van de haven tot CD - 11 dalen de kosten ten opzichte van het 0 alternatief met ongeveer 4 miljoen dollar per jaar in 2000 en 17 miljoen dollar in 2010.
- De aanleg van een extra aanlegplaats en het verdiepen van de haven tot CD -13 geeft een besparing ten opzichte van het nulalternatief van 7 miljoen dollar in 2000 en 21 miljoen dollar in 2010.
- Verdiepen van de haven geeft een besparing in het benodigd aantal schepen.

Samenvattende kan er gesteld worden dat eM-Plant geschikt is voor het maken van havensimulaties. Tevens beschikt eM-Plant over voldoende animatiekwaliteiten om duidelijke presentaties te geven. Als opmerking moet er bij dit onderzoek vermeld worden dat er naast de onderzochte simulatie pakketten nog meer pakketten op de markt zijn. Een van de aanbevelingen is om deze pakketten in een vervolgstudie te evalueren.



Inhoudsopgave

VOORWOORD.....	I
SAMENVATTING.....	II
INHOUDSOPGAVE	IV
1 INLEIDING.....	1
2 SIMULATIE.....	2
2.1 WAT IS SIMULATIE ?	2
2.2 WAAROM SIMULEREN ?	2
2.3 VOOR- EN NADELEN VAN SIMULATIE.....	2
2.4 SIMULATIEBEGRIPPEN.....	3
2.4.1 "Discrete event" simulatie.....	3
2.4.2 "Continue" simulatie	3
2.4.3 Object georiënteerd programmeren	3
2.4.4 Stroom georiënteerd programmeren	4
3 ONDERZOEKSVRAAG EN OPBOUW.....	5
3.1 VRAAGSTELLING	5
3.2 DOELSTELLING.....	5
3.3 OPBOUW VAN HET ONDERZOEK.....	5
3.3.1 Fase 1 Onderzoek naar en kiezen van een pakket	5
3.3.2 Fase 2 Modelkeuze en uitwerken model.....	5
3.3.3 Fase 3 Evaluatie.....	5
4 EVALUATIE CRITERIA SOFTWARE PAKKETTEN.....	6
4.1 SOFTWARE	6
4.2 MODEL EN INPUT	6
4.3 DRAAIEN VAN DE SOFTWARE	7
4.4 ANIMATIE	7
4.5 TESTEN EN EFFICIËNTIE.....	7
4.6 UITVOER	7
4.7 AANSCHAFKOSTEN	8
5 ALGEMENE BESCHRIJVING SOFTWAREPAKKETTEN	9
5.1 PROSIM.....	9
5.2 AUTOMOD.....	10
5.3 ARENA.....	11
5.4 EM-PLANT.....	12
5.5 TOMAS.....	13
6 PAKKETKEUZE	14
6.1 WAARDE TOEKENNING CRITERIA.....	14
6.1.1 Software.....	14
6.1.2 Model en input	14
6.1.3 Draaien van het programma.....	15
6.1.4 Animatie en "testen en efficiëntie".....	15
6.1.5 Uitvoer.....	15
6.2 MULTI CRITERIA ANALYSE.....	16
6.3 TOELICHTING WEGING	17



6.3.1	Software.....	17
6.3.2	Model en input	17
6.3.3	Draaien van het programma.....	19
6.3.4	Animatie.....	19
6.3.5	Testen en efficiency	19
6.3.6	Uitvoer.....	20
6.3.7	Aanschafkosten.....	20
6.4	DEFINITIEVE PAKKETKEUZE	21
6.4.1	Civieltechnische methode.....	21
6.4.2	De methode van de faculteit TBM	22
6.4.3	Keuze van methode.....	22
7	MODELEREN IN EM-PLANT.....	24
7.1	OPBOUW VAN DE MODELLEN.....	24
7.2	BOUWEN VAN MODELLEN	25
7.2.1	Tabblad 1.....	25
7.2.2	Tabblad 2.....	26
7.2.3	Tabblad 3.....	27
7.3	EENVOUDIG VOORBEELD	27
8	PROJECT BESCHRIJVING	29
8.1	PRODUCTEN	29
8.2	LAY-OUT.....	30
8.3	ALTERNATIEVEN	31
9	CONCEPTUALISATIE	32
9.1	INLEIDING	32
9.2	FYSISCH GRENSEN	32
9.3	OVERIGE GRENZEN	33
10	SPECIFICATIE.....	34
10.1	SCHEEPSGEGEVENS	34
10.1.1	Analyse	34
10.2	TOEKOMSTIGE SCHEEPSAFMETINGEN	35
10.3	TUSSENAANKOMSTTIJDEN	35
10.4	KLINKER	36
10.4.1	Laad en los cyclus.....	36
10.4.2	Huidige import.....	37
10.4.3	Toekomstige doorvoer.....	37
10.5	KOLEN.....	37
10.5.1	Laad en los cyclus.....	37
10.5.2	Huidige export kolen.....	38
10.5.3	Prognose kolen export.....	38
10.6	IJZERERTS	38
10.6.1	Laad en los cyclus.....	38
10.6.2	Huidige export ijzererts.....	39
10.6.3	Prognose export ijzererts.....	39
10.7	VERBLIJFTIJDEN	39
10.8	BAKKENTRANSPORT	40
10.9	RELATIE DIEPGANG EN TONNAGE	40
10.10	GETIJ	42
10.11	PROCESSEN EN STROOMDIAGRAMMEN	43
11	MODEL BESCHRIJVING EM-PLANT	49
11.1	BESCHRIJVING HOOFDFRAME.....	49
11.2	KOPPELING STROOMDIAGRAM AAN SIMULATIEMODEL	50



12	BEHANDELINGSOPZET, VERIFICATIE EN VALIDATIE.....	55
12.1	BEHANDELINGSOPZET	55
12.1.1	<i>Opstarttijd.....</i>	55
12.1.2	<i>Runlengte.....</i>	56
12.1.3	<i>Bepalen aantal replicaties</i>	56
12.2	VERIFICATIE	57
12.2.1	<i>Autocorrelatie</i>	58
12.3	VALIDATIE.....	59
13	EXPERIMENTEN.....	60
13.1.1	<i>Havengerelateerde kosten.....</i>	61
13.1.2	<i>Vaargerelateerde kosten.....</i>	61
13.2	UITVOERWAARDEN	62
13.2.1	<i>Uitvoer 2000.....</i>	62
13.2.2	<i>Uitvoer 2010.....</i>	64
14	CONCLUSIES EN AANBEVELINGEN	66
14.1	CONCLUSIES EERSTE DEEL	66
14.2	CONCLUSIES TWEEDE DEEL.....	67
14.3	BEANTWOORDING ONDERZOEKSVRAAG.....	67
14.4	BEGRENZING.....	67
14.5	AANBEVELINGEN	68
15	AFSTUDEERCOMMISSIE	69
16	LITERATUURLIJST	70
	BIJLAGE 1 SIMULATIE PAKKETTEN	71
	BIJLAGE 2 DATAFIT.....	87
	BIJLAGE 3 VERIFICATIE EN VALIDATIE.....	90



1 Inleiding

In het verleden is er regelmatig onderzoek gedaan naar de aanleg van havens. In de jaren voor de uitvinding van de computer werden deze onderzoeken met de hand doorgerekend. Na de introductie van de computer werd snel duidelijk dat de complexe vervoersstromen in de havengebieden beter en sneller met behulp van de computer doorgerekend konden worden. Vanaf dat moment werden de eerste havensimulaties van de vervoersstromen in de havengebieden gemaakt.

Voor het simuleren van vervoersstromen kunnen verschillende programma's gebruikt worden. Deze programma's zijn onder andere:

- Arena
- Automod
- eM-Plant
- Tomas
- Prosim

Prosim is het simulatiepakket dat op dit moment veel gebruikt wordt. Dit pakket is echter op het gebied van animatie minder sterk dan andere vergelijkbare pakketten. Hierdoor is binnen de organisatie van Boskalis Westminster de vraag ontstaan of er grafisch sterkere pakketten op de markt zijn, die rekentechnisch gelijkwaardig zijn.

Veel van de grote baggerprojecten, waar Boskalis aan werkt hebben direct te maken met havens. Er kan hierbij gedacht worden aan het verdiepen van bestaande havens of het creëren van nieuwe havens (Nieuwe Maasvlakte). Vaak gaat er aan dit soort projecten jaren van onderzoek vooraf. Voor Boskalis is het interessant om die onderzoeken zelf uit voeren om op deze manier de kosten te beperken. Bij deze onderzoeken horen ook de simulaties van de betreffende haven. Dit is één van de redenen tot initiatie van deze studie.

In het eerste deel van het rapport worden de volgende items behandeld:

- De eigenschappen van de verschillende simulatie programma's
- De keuze van één van deze programma's
- De vraag waarom er gekozen moet worden voor simulatie

Het pakket dat in het eerste deel gekozen is zal in het tweede deel van het rapport als basis dienen voor een case studie van een haven.

De te simuleren haven heeft een aantal capaciteitsproblemen. Deze problemen worden grotendeels veroorzaakt door een te geringe diepte en kades die niet geschikt zijn om te laden of lossen. Het tweede probleem heeft men in het verleden gedeeltelijk op weten te lossen door een bakkentransport in te zetten via welke de schepen geladen of gelost worden. Doordat de verschillende vervoersstromen elkaar beïnvloeden ontstaat een complex probleem. De oplossingen voor dit probleem zijn moeilijk met de hand te berekenen. Daarom is deze haven als casestudie voor dit onderzoek gekozen.



2 Simulatie

In dit hoofdstuk wordt beschreven wat simulatie is en waarom het op dit moment bij veel bedrijven wordt gebruikt. Verder zullen de redenen voor het gebruik van simulatie als hulp voor het nemen van beslissingen worden genoemd en zal er inzicht worden gegeven in de verschillende simulatie technieken, die op dit moment worden gebruikt.

2.1 Wat is simulatie ?

De definitie van simulatie is het nabootsen van een proces of systeem uit de werkelijkheid over een bepaald tijdbestek door middel van een model. Onder simulatie valt ook het genereren van kunstmatige geschiedenis (in de vorm van data) en het observeren hiervan. Hieruit kunnen conclusies en aanbevelingen voor het werkelijke systeem worden gehaald, wat een bestaand of nog te realiseren systeem kan zijn. Alvorens een proces of systeem kan worden gesimuleerd moet eerst een model van het systeem worden gemaakt.

2.2 Waarom simuleren ?

Door het nabootsen van een systeem of proces door middel van simulatie kan op een relatief goedkope manier het werkelijke systeem of proces worden getest op tekortkomingen zonder dat het in werkelijkheid gebouwd of aangepast hoeft te worden. Simuleren is echter niet altijd nodig, in geval van een simpel systeem of proces kan rekenen aan het systeem voldoende nauwkeurige informatie opleveren. De wachttijdtheorie is hier een voorbeeld van.

Over het algemeen wordt simulatie toegepast om te helpen bij het nemen van beslissingen bij ingewikkelde processen. In de havenindustrie moet hierbij gedacht worden aan het simuleren van complete havensystemen of delen hiervan. In het verleden zijn onder andere de Maasvlakte en de sluis van IJmuiden gesimuleerd.

2.3 Voor- en nadelen van simulatie

Uit de literatuur komen vooral de voordelen van de simulatiepakketten duidelijk naar voren. Er zijn echter ook een aantal nadelen aan het simuleren verbonden. In Tabel 2.3.1 worden deze voor- en nadelen weergegeven.



Voordelen	Nadelen
Mogelijkheid tot het uitwerken van verschillende alternatieven zonder grote investeringen.	Tijdrovend
Simulaties zijn vaak een goede basis om beslissingen op te nemen.	Dure soft- en hardware nodig
Het systeem kan stap voor stap doorgenomen worden, zodat er meer inzicht in het systeem ontstaat.	Bij het simuleren van een nog te realiseren systeem is het moeilijk in te schatten of het systeem een betrouwbaar beeld van de werkelijkheid geeft.
Makkelijk aan te passen aan nieuwe situaties.	Vanwege het gemak van het invoeren van nieuwe alternatieven worden er vaak erg veel en nutteloze alternatieven gesimuleerd.

Tabel 2.3.1 Voor- en nadelen van simuleren

2.4 Simulatiebegrippen

Er wordt onderscheid gemaakt tussen twee verschillende technieken, de eerste techniek is de discrete event simulatie en de tweede is de continue simulatie. Daarnaast worden de begrippen object georiënteerd en stroom georiënteerd programmeren in deze alinea behandeld.

2.4.1 "Discrete event" simulatie

De "discrete event" methode simuleert het gedrag van grootheden als er op een bepaald punt in de tijd een gebeurtenis plaats vindt. De grootheden zijn de componenten van het systeem die beschreven zijn door variabelen. De momenten in de tijd waarop de variabelen veranderen worden de gebeurtenissen genoemd. Een belangrijke eigenschap van deze methode is, dat er tussen de gebeurtenissen in niets gebeurt.

2.4.2 "Continue" simulatie

In tegenstelling tot de "discrete event" methode wordt bij continue simulatie de tijd wel continu genomen (zoals de naam al zegt). De tijd wordt in kleine stukjes verdeeld en per tijdstapje wordt een nieuwe situatie berekend.

2.4.3 Object georiënteerd programmeren

Bij object georiënteerd programmeren wordt het systeem geschematiseerd als een verzameling objecten. Deze objecten zijn over het algemeen bestaande dingen zoals bijvoorbeeld schepen. Voor ieder object wordt een apart stukje programma geschreven dat los staat van de overige objecten. Het programma bestaat dus uit verschillende autonome objecten, die informatie met elkaar uitwisselen.



2.4.4 Stroom georiënteerd programmeren

Bij stroom georiënteerd programmeren wordt de werkelijkheid geschematiseerd als een aaneenschakeling van processen / acties, de entiteiten stromen hier als het ware doorheen. Een voorbeeld van een proces dat vrij makkelijk door middel van "Stroom georiënteerd programmeren" kan worden gesimuleerd is een fabriekshal, waar de entiteiten van machine naar machine "stromen" door middel van lopende banden. De entiteiten ondergaan hierbij de behandelingen.



3 Onderzoeksvraag en opbouw

3.1 Vraagstelling

Welk simulatiepakket is geschikt voor het simuleren van de vervoersstromen in hedendaagse havens en heeft dit pakket voldoende animatiemogelijkheden.

3.2 Doelstelling

Het doel is: het vinden van een geschikt pakket voor het maken van simulatiemodellen voor een havenontwerp/dimensionering, vaarwegen en andere complexe werken binnen de baggersector. Tevens zal een havensysteem geanalyseerd moeten worden om te valideren of het gekozen pakket aan de gestelde eisen voldoet.

3.3 Opbouw van het onderzoek

Om een gestructureerde en duidelijke opbouw van het onderzoek te verkrijgen is er gekozen om het onderzoek op te delen in drie fasen. De eerste fase betreft het kiezen van het simulatie pakket. De tweede fase gaat over de keuze van het model dat wordt gesimuleerd en in de derde fase zal een evaluatie worden gegeven het onderzoek. Hieronder worden de verschillende fasen verder uitgewerkt.

3.3.1 Fase 1 Onderzoek naar en kiezen van een pakket

In fase 1 zullen de evaluatiecriteria worden genoemd, zodat het eenvoudiger is om de verschillende pakketten te vergelijken om uiteindelijk tot een pakketkeuze te komen. De volgende criteria zullen onder meer meegenomen worden:

- Is de continuïteit van het pakket gegarandeerd ?
- Is het pakket betrouwbaar ?
- Is er voldoende documentatie over het pakket ?

De overige selectie criteria zijn gegeven in hoofdstuk 4 (evaluatie criteria softwarepakketten). Aan het eind van dit deel zal met behulp van een uitgebreide evaluatie van de verschillende software pakketten tot een definitieve pakketkeuze worden gekomen.

3.3.2 Fase 2 Modelkeuze en uitwerken model

In de tweede fase zal een havensysteem worden gemodelleerd om op deze manier een beeld te krijgen van de sterke en zwakke punten van het gekozen softwarepakket. Het begin van deze fase is gericht op het kiezen van het model gevolgd door het uitwerken van het betreffende model.

3.3.3 Fase 3 Evaluatie

In fase één is het softwarepakket gekozen en in fase twee is met het gekozen software pakket een casestudie uitgevoerd. Fase drie is een evaluatie van het onderzoek.



4 Evaluatie criteria software pakketten

Om een goede vergelijking en keuze te kunnen maken tussen de verschillende pakketten moeten er een aantal criteria (J. Nikoukaran 1999) worden opgesteld waaraan het benodigde pakket moet voldoen. Deze criteria kunnen worden verdeeld in drie hoofdgroepen: softwarebedrijf, software en gebruiker. De hoofdgroepen kunnen ook verdeeld worden in een aantal subgroepen. De hoofdgroep softwarebedrijf is opgedeeld in drie subgroepen: Verleden, documentatie en ondersteuning. De hoofdgroep software wordt opgedeeld in vijf sub-criteria; ontwikkeling van het model en input, draaien van de software, testen en efficiëntie, animatie en uitvoer. De laatste hoofdgroep is de gebruiker welke opgedeeld is in: simulatietype, algoritmes en financieel. In dit hoofdstuk zullen de verschillende criteria worden behandeld.

4.1 Software

De criteria van het softwarebedrijf zijn vooral gericht op de geloofwaardigheid van het softwarebedrijf, als mede de geloofwaardigheid van het product. Met de onderstaande criteria moet een beeld worden gekregen over de betrouwbaarheid en leerbaarheid van het product. De criteria zijn onderverdeeld in drie groepen genaamd:

Softwarebedrijf

Algemene indruk en prestaties van het bedrijf

Documentatie

Bestaat er goede en duidelijke documentatie van de software?

Worden er voorbeeldmodellen bij de software geleverd?

Ondersteuning

Wordt er telefonische of andere ondersteuning door het bedrijf geleverd?

Zijn er op de Technische Universiteit Delft mensen aanwezig die ondersteuning kunnen/willen geven?

4.2 Model en input

Grafisch bouwen van modellen

De modellen kunnen op verschillende manieren worden gemaakt. De twee uitersten liggen bij het schrijven van het model in code en het bouwen van het model in een grafische omgeving. De meeste software zal een combinatie van code en een grafische omgeving zijn, waarbij de ene software meer neigt naar code en de andere grafischer is ingesteld.

Invoer

Er zijn verschillende manieren om de invoer te genereren, bijvoorbeeld interactief, lezen uit een file, automatisch uit een systeem inlezen.

Statistische verdelingen

Sommige invoer in het programma zal statistisch verdeeld moeten worden. Het programma moet in ieder geval een aantal verdelingen aan kunnen zoals normaal, exponentieel en uniform.

Mogelijkheid tot invoer code

De mogelijkheid tot de invoer van code is een graadmeter voor de flexibiliteit van het programma, is het mogelijk om zelf code in het programma in te voeren of kan zelfs de source code van het programma aangepast worden? (dus op modelniveau of op programma niveau).



Wachtrij-systemen

Kan het programma verschillende wachtrij-systemen zoals FIFO en LIFO aan?

Ingebouwde variabelen

Kan het verschil tussen integer en real worden aangegeven of is het zelfs mogelijk aan te geven of een waarde een tijd of een snelheid is.

Debugmogelijkheden

Sommige programma's hebben functies die het eenvoudiger maken om de fouten in de zelf gemaakte code op te sporen.

4.3 Draaien van de software**Aanpassen snelheid**

Mogelijkheid tot het aanpassen van de snelheid. Wanneer het programma op een lagere snelheid wordt gedraaid is het vaak makkelijker problemen en programmeerfouten op te sporen.

Executable

Kan het programma van een model een zelfdraaiend model maken (executable)?

4.4 Animatie**Geïntegreerd animatiepakket**

Is het animatiepakket geïntegreerd in de software of wordt het er apart bijgeleverd?

Iconen

Worden er iconen bij het pakket geleverd?

Is het mogelijk om nieuwe iconen aan te maken?

Kunnen er iconen uit andere software ingevoerd worden (zoals bijvoorbeeld CAD of bitmap)?

4.5 Testen en efficiëntie**Multitasking**

Is het mogelijk om tijdens het bewerken van een model op hetzelfde moment een model te "runnen" en kunnen er meerdere vensters "open" staan?

4.6 Uitvoer**Rapporten**

Welke rapporten komen standaard voor in het programma?

Kunnen er zelf rapporten worden opgesteld?

(Standaard rapporten zijn meestal wachtrijlengtes en wachttijden).

Grafieken

De pakketten leveren in een aantal gevallen ook mogelijkheden tot het maken van grafieken zoals histogrammen, staafdiagrammen en andere grafiekvormen.



4.7 Aanschafkosten

Financieel

Kan de software tegen een redelijke prijs voor een bepaalde periode gehuurd worden en wat zijn de kosten van het gehele pakket?

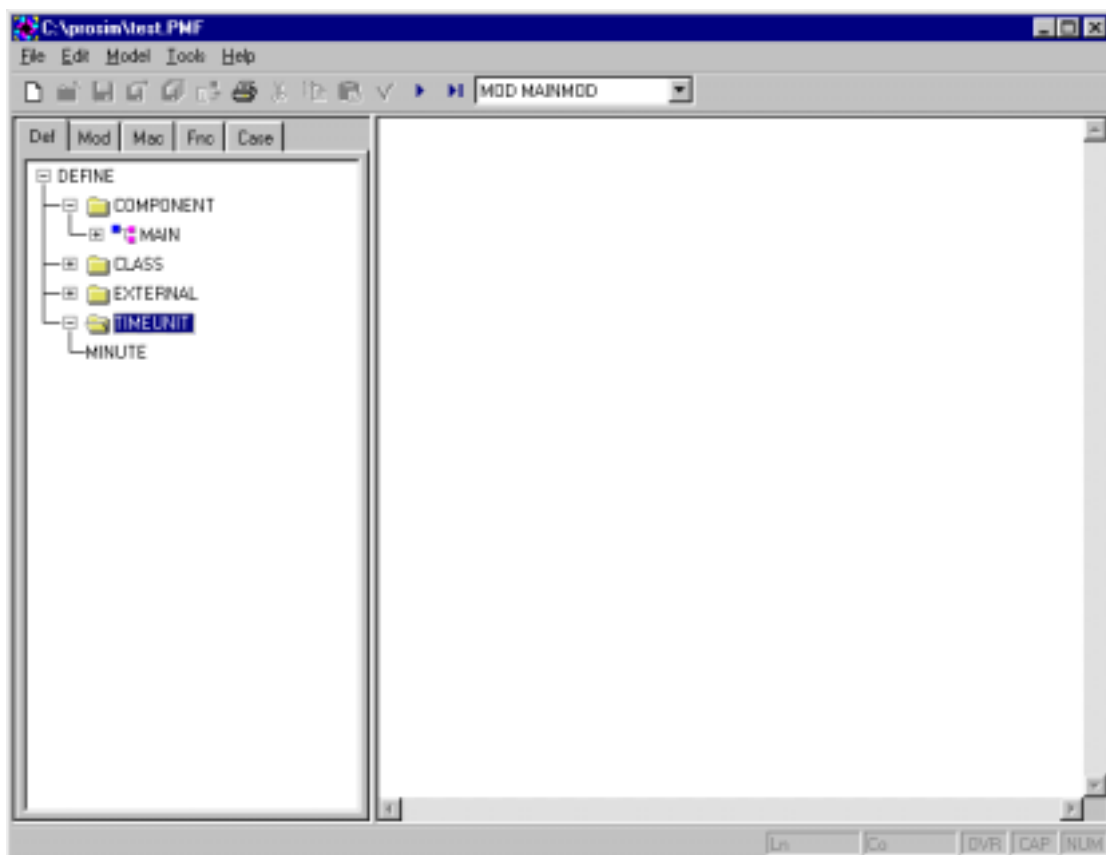


5 Algemene beschrijving softwarepakketten

In dit hoofdstuk worden vijf verschillende software pakketten beschreven om op deze manier een beter beeld te krijgen van de onderlinge verschillen. Het eerste pakket dat zal worden besproken is Prosim, aangezien dit het huidige programma is dat gebruikt wordt voor het simuleren van havens. De overige pakketten zijn Automod, Arena, eM-Plant en Tomas. Bij het schrijven van dit hoofdstuk is gebruik gemaakt van "simulation software review" van E Valentin, die hiervoor de selectie criteria van Nikokouran gebruikte. Voor de uitgebreide evaluatie wordt verwezen naar Bijlage 1.

5.1 PROSIM

Prosim is een modelleertaal die zowel discrete als continue processen kan simuleren, de taal wordt vooral gebruikt voor planning, logistieke systemen en chemische processen. Volgens de handleiding kan de taal in 4 tot 5 dagen worden geleerd, hierna kan op beginners niveau met de taal worden omgegaan. Er zijn in het totaal zijn 20 tot 50 pakketten verkocht (volgens de site "Software tools for process integration"). De hardware eisen zijn betrekkelijk laag, om het programma te "runnen" kan volstaan worden met een computer met een pentium processor. In Figuur 5.1.1 wordt een voorbeeld gegeven van een mogelijke schermopbouw in Prosim.



Figuur 5.1.1 Schermvoorbeeld Prosim

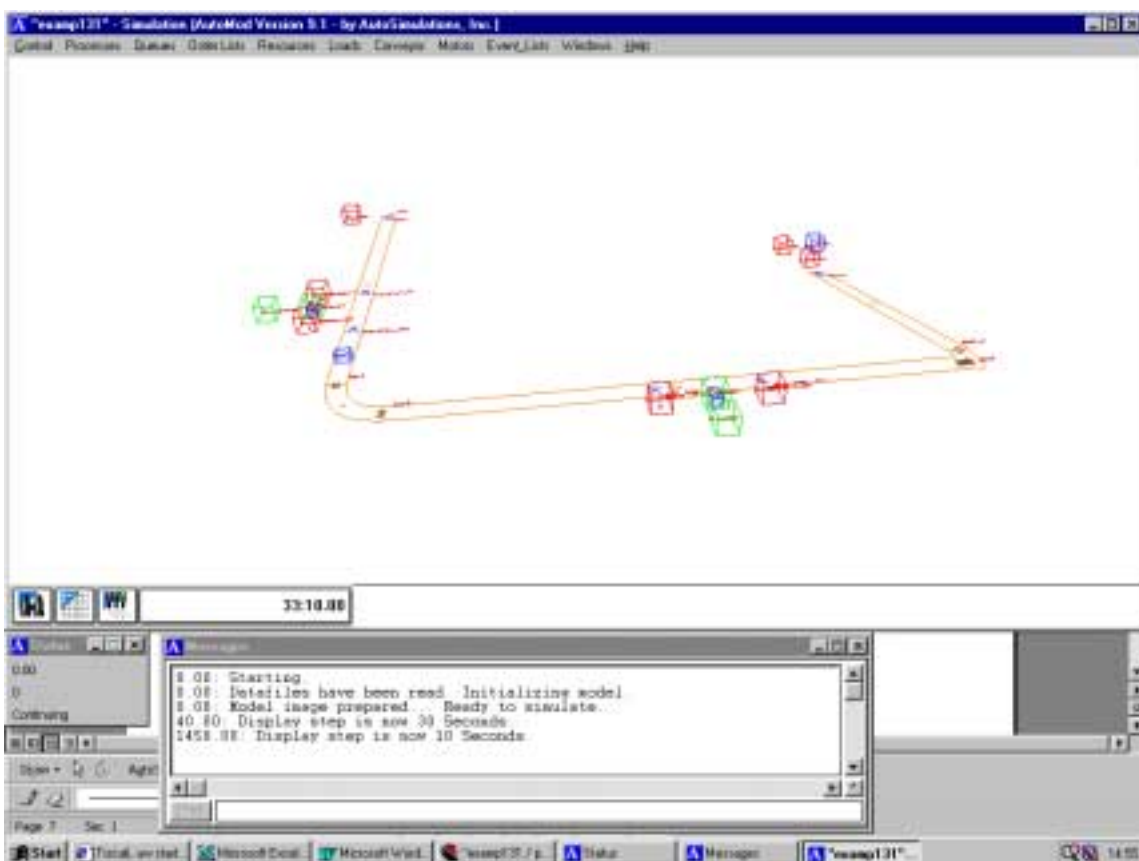


5.2 Automod

Het simulatie programma Automod is gebaseerd op een grafische omgeving waar de objecten en processen in “geplaatst” kunnen worden. Het programma bevat een bibliotheek waaruit de verschillende simulatieonderdelen gekozen kunnen worden, om ze vervolgens in de grafische omgeving op schaal te plaatsen. Het voordeel hiervan is dat tijdens het programmeren direct op schaal gewerkt wordt, dit verschaft veel inzicht in de lay-out van de verschillende onderdelen. Het nadeel is echter dat er gekozen moet worden uit een bibliotheek van onderdelen en het is niet duidelijk of deze onderdelen ook zelf gemaakt kunnen worden. Door deze grafische opbouw en de duidelijke handleiding kan het programma wel snel worden geleerd.

De software kan volgens de handleiding zowel discrete als continue processen simuleren. Verder wordt er een geïntegreerde 3d omgeving bijgeleverd, die het ruimtelijk inzicht in het probleem vergroot. Met behulp van Autostat kunnen de verschillende onderdelen van het programma weergegeven worden in grafiekvorm.

De softwareproducent Brooks Automation is een van de grote simulatie leveranciers, dat de kans op continuïteit aanzienlijk vergroot. In Figuur 5.2.1 wordt een schermopbouw van Automod weergegeven.

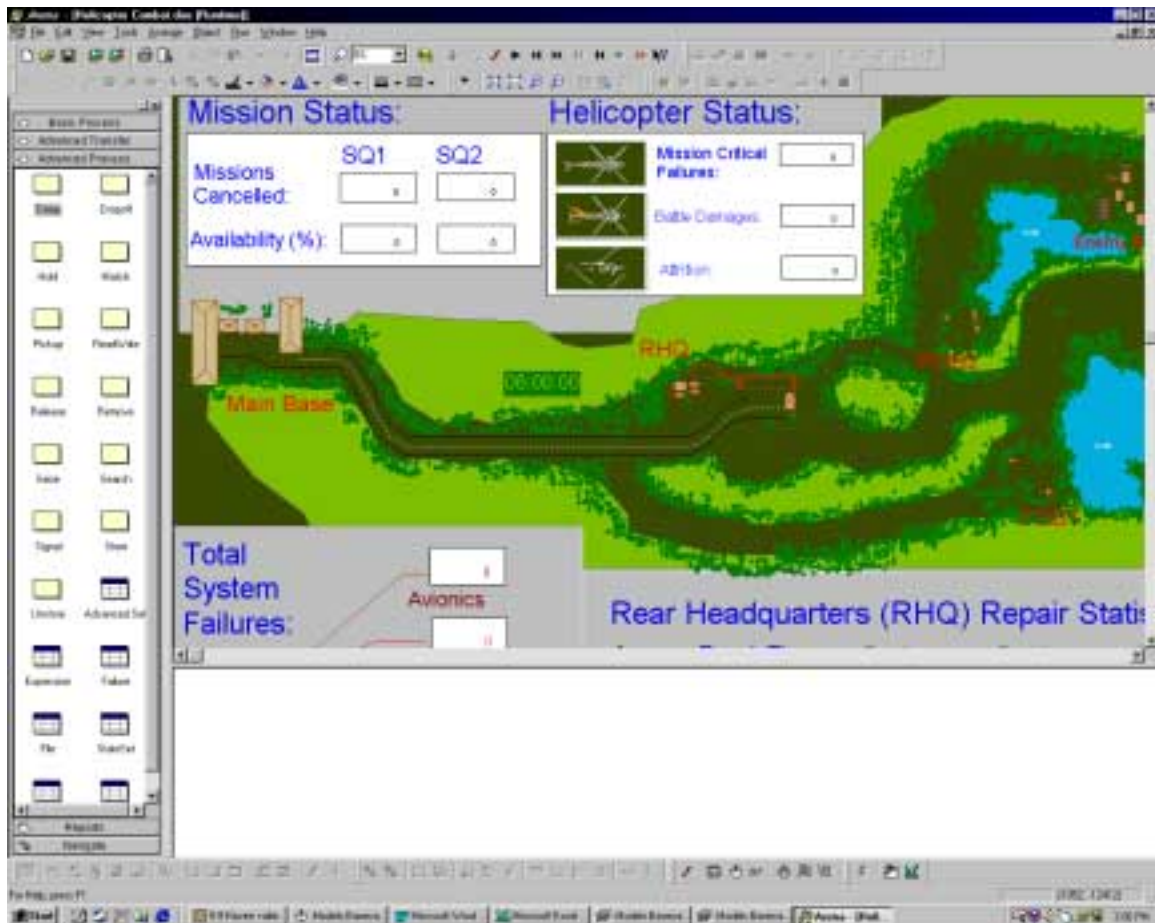


Figuur 5.2.1 Schermvoorbeeld Automod

5.3 Arena

Arena is een pakket dat in veel opzichten lijkt op Automod. Zo beschikt Arena net als Automod over een grafische omgeving waar objecten in geslept kunnen worden (zie Figuur 5.3.1) en kan er gekozen worden uit een bibliotheek van simulatieonderdelen. Het pakket wordt uitgebracht door Rockwell Software en bestaat al meer dan 15 jaar. Doordat Arena grafisch gericht is kan de basis van deze taal snel worden aangeleerd. Het programmeren van de onderliggende code is echter ingewikkeld. Door middel van deze code kunnen echter wel zelf programmablokken worden gecreëerd, zodat een eigen bibliotheek kan worden opgezet.

Arena heeft een uitstekende documentatie, omdat er bij het pakket meer dan 150 kleine simulatie modellen worden geleverd, die voor de meest voorkomende problemen een oplossing geven. Arena beschikt ook over voldoende documentatie in de vorm van leerboeken en handboeken.



Figuur 5.3.1 Schermvoorbeeld Arena

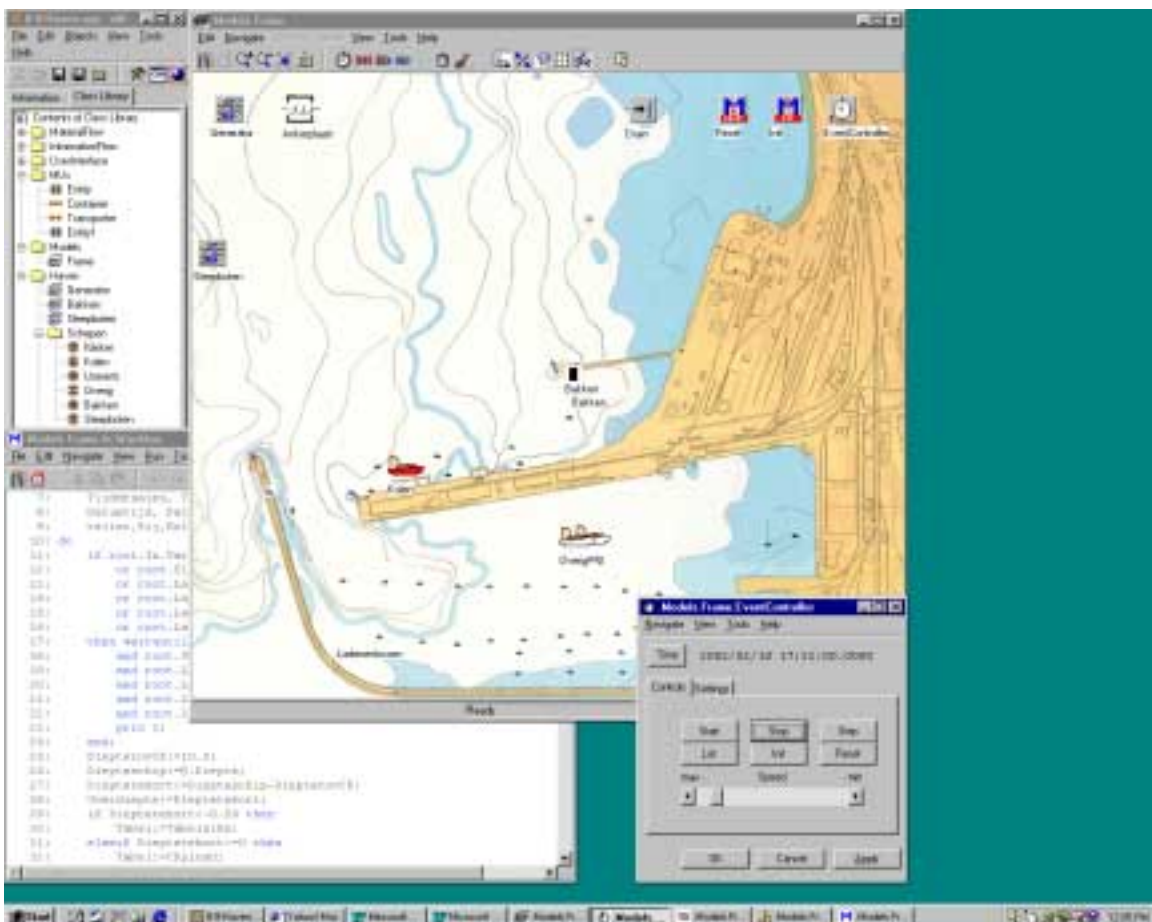


5.4 eM-plant

Een aantal jaren geleden is het programma Simple++ overgenomen door Tecnomatix, met deze overname is ook de naam Simple++ veranderd in eM-Plant, tevens zijn een aantal fouten uit het programma verwijderd.

eM-Plant is een programma dat vooral in de auto-industrie grote populariteit geniet. Vandaar dat het programma ook gericht is op deze tak van industrie. Het pakket kan echter door middel van programmeren van nieuwe programmablokken wel zodanig aangepast worden, dat het geschikt gemaakt kan worden voor andere doeleinden. Het relatief makkelijk programmeren van nieuwe blokken is ook een van de sterke punten van dit pakket. Verder kan met behulp van eM-Plant grafische uitvoer worden gerealiseerd. Deze grafische omgeving is echter wel vooral 2d gericht, hoewel het ook mogelijk zou moeten zijn om 3d uitvoer te genereren (zie Figuur 5.4.1.).

Aan dit pakket zijn twee nadelen verbonden namelijk: de noodzaak tot het gebruik van zeer snelle computers (omdat het pakket zelf redelijk langzaam is) en de zeer hoge aanschafprijs. Ook de documentatie en de meegeleverde voorbeelden van het pakket zijn niet optimaal. Van de geëvalueerde pakketten is deze het meest object georiënteerd, wat inhoudt dat geschreven blokken vrij makkelijk in een programma kunnen worden geïntegreerd, zodat stukken software steeds opnieuw gebruikt kunnen worden.



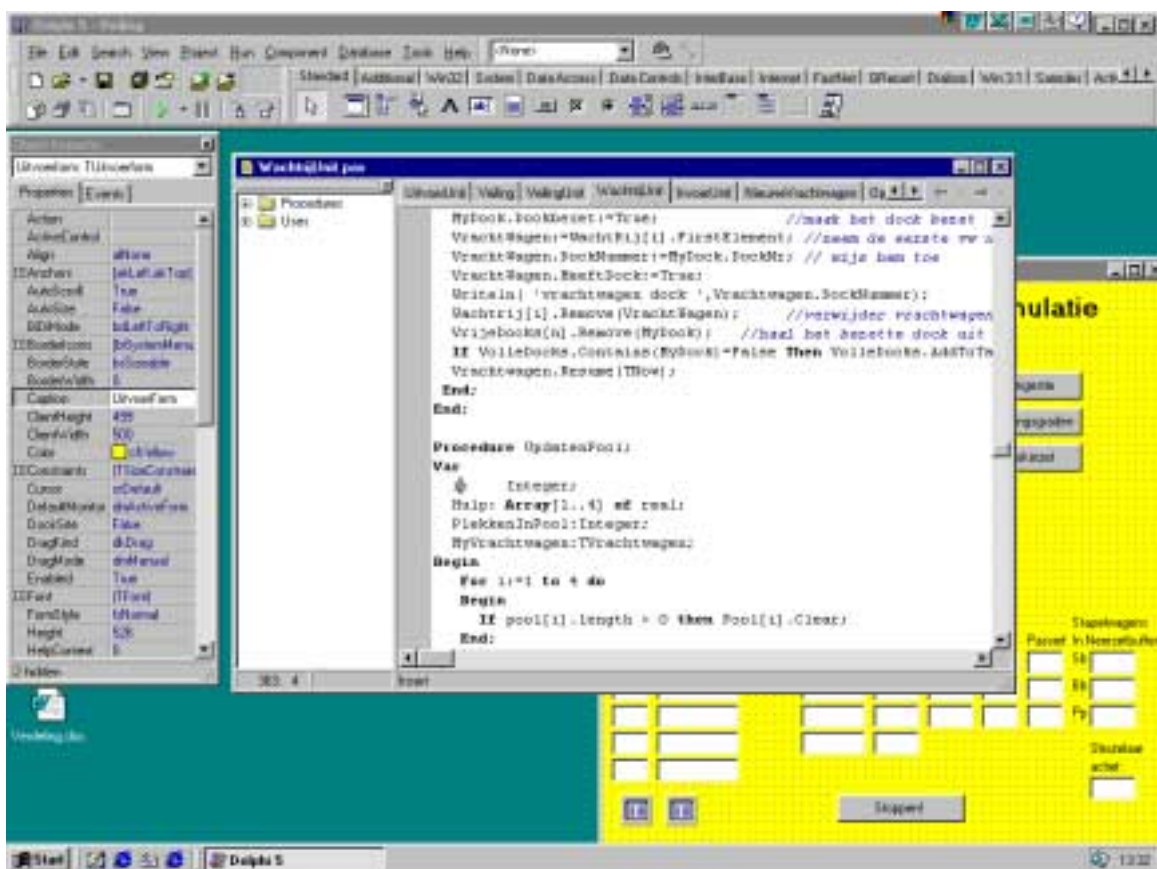
Figuur 5.4.1 Schermvoorbeeld eM-Plant



5.5 Tomas

Tomas is een applicatie die het beste met Prosim te vergelijken is. Het pakket is net als Prosim op code gebaseerd (zie Figuur 5.5.1). In de praktijk komt het er op neer dat het programma in Delphi moet worden geprogrammeerd, waarbij gebruik gemaakt kan worden van de discrete Tomas omgeving in Delphi. Een duidelijk voordeel van het gebruik van code is dat (hoewel dit veel tijd kost) rekentechnisch bijna alles mogelijk is. Grafisch is het gebruik van code echter veel minder makkelijk, het kost veel tijd om een relatief simpele animatie te maken.

Omdat Tomas nog een taal in ontwikkeling is, is de documentatie over de taal nog zeer gering. Bij het pakket worden wel een aantal voorbeeld modellen geleverd, die een goed beeld geven van de mogelijkheden van Tomas.



Figuur 5.5.1 Schermvoorbeeld Tomas



6 Pakketkeuze

In het vorige hoofdstuk zijn de verschillende pakketten beschreven, zodat een beeld werd verkregen van wat een bepaald pakket kan en wat de voor- en nadelen van het pakket zijn. In dit hoofdstuk zullen de voor- en nadelen van de pakketten naast elkaar worden gezet om op deze manier aan het eind van dit hoofdstuk een pakketkeuze te kunnen maken. Om een goede en betrouwbare keuze te kunnen maken is gebruik gemaakt van een multi criteria analyse. In Tabel 6.2.1 is deze analyse weergegeven, de letters A tot en met E staan voor de verschillende pakketten en de cijfers voor de waardering van het criterium voor het betreffende programma.

6.1 Waarde toekenning criteria

In de onderstaande multi criteria analyse (mca) zijn de voor dit onderzoek relevante criteria uit de indeling van J. Nikouranan gebruikt, deze indeling is verder uitgewerkt in hoofdstuk 4 (Evaluatie criteria software pakketten). Aan ieder criterium in de mca is een bepaalde waarde gehangen naar aanleiding van het gewicht van het item. Zo zijn een goede documentatie en de mogelijkheid tot de invoer van code de twee belangrijkste criteria, beide met een gewicht van 50. In de onderstaande paragrafen zullen de overige criteria met gewicht en het cijfer voor de verschillende programma's uitgewerkt worden.

6.1.1 Software

Het eerste punt in de mca is het softwarebedrijf. Dit punt is onderverdeeld in drie verschillende onderdelen: algemene indruk van het bedrijf, documentatie en ondersteuning. Het belangrijkste item is de documentatie van het product. Dit criterium heeft het zwaarste gewicht, omdat zonder een goede documentatie in de vorm van helpfiles, een leerboek of een handleiding het maken van een simulatie een bijzonder lastige opgave wordt. In dit geval zal veel extra tijd geïnvesteerd moeten worden in het leren omgaan met het programma. Op de tweede plaats staat de algemene indruk van het bedrijf. Zonder de garantie op continuïteit is het niet verstandig om één van de pakketten aan te schaffen, mede vanwege het feit dat de pakketten over het algemeen behoorlijk kostbaar zijn. Het laagst gewaardeerde item onder de kop softwarebedrijf is de ondersteuning.

6.1.2 Model en input

Het item "model en input" bestaat uit zeven criteria: het grafisch bouwen van modellen, invoer, statistische verdelingen, mogelijkheid tot invoer van code, wachtrij systemen, ingebouwde functies en debugmogelijkheden.

Van deze zeven criteria krijgt de mogelijkheid tot de invoer van code de hoogste waardering. Deze functie staat voor de mogelijkheid van het creëren van eigen programmablokken. In de havenlogistiek kunnen we hier bijvoorbeeld denken aan het creëren van vaarroutes of terminals. Zonder deze functie is het niet mogelijk om zonder hulp van de leverancier simulaties te maken van niet standaard objecten (b.v. sluisen). De invoer van verschillende statistische verdelingen en de aanwezigheid van wachtrij systemen krijgen beide een gewicht van veertig. Deze functies zijn niet onmisbaar maar zorgen wel voor een betere simulatie.



Het grafisch bouwen van modellen wil zeggen, dat door middel van "copy pasten" van bestaande blokken een simulatie in elkaar gezet kan worden. Op deze manier kan snel een simulatie worden opgezet. Vanwege het mindere belang is het gewicht van dit criterium dertig. Eenzelfde gewicht wordt toegekend aan "ingebouwde functies" en "debugmogelijkheden".

6.1.3 Draaien van het programma

Aanpassen van de snelheid en de mogelijkheid tot het maken van een executable zijn de criteria vallend onder de titel: "Draaien van het programma". Van deze twee functies is "aanpassen van de snelheid" de belangrijkste, deze functie zorgt ervoor dat de fouten eenvoudiger uit de programmastructuur gehaald kunnen worden. Het maken van een executable (zelfdraaiende simulatie) is met 20 punten minder belangrijk, want ook zonder deze functie kunnen goede simulaties worden gemaakt.

6.1.4 Animatie en "testen en efficiëntie"

In de onderzoeksvraag kwam de animatie al naar voren als één van de belangrijkste criteria, daarom krijgt dit criterium hier een relatief hoog gewicht van veertig.

De laatste vier criteria (aanschafkosten daargelaten) staan onder twee items, de eerste kop is "testen en efficiëntie", met als criteria multitasking en stapfunctie. Multitasking wil zeggen dat tijdens het runnen ook andere dingen kunnen worden gedaan, zoals het wijzigen van code. Dit criterium krijgt twintig punten, net als de mogelijkheid om stap voor stap door het programma te lopen. Deze stapfunctie maakt het eenvoudiger om fouten uit de simulatie te halen.

6.1.5 Uitvoer

De uitvoer is onderverdeeld in rapportage en grafieken, met rapportage wordt de leesbaarheid en het gemak van wegschrijven van de uitvoer bedoeld. Beide criteria zijn belangrijk om een goed inzicht te krijgen in het gedrag van het model om deze reden hebben beide criteria een gewicht van 30 punten gekregen.



6.2 Multi criteria analyse

Arena	Rockwell simulation software automation	1	slecht
Automod	Brooks automation	2	matig
eM-plant	Tecnomatix	3	voldoende
Tomas/Delphi	H.P.M. Veeke en J. A. Ottjes	4	goed
Prosim	Prosim B.V.	5	zeer goed

		Arena		Automod		eM-Plant		Tomas		Prosim	
Criteria	Gewicht	Score		Score		Score		Score		Score	
Software											
Softwarebedrijf	40	4	160	4	160	4	160	2	80	3	120
Documentatie	50	4	200	5	250	3	150	1	50	3	150
Ondersteuning	30	4	120	4	120	3	90	2	60	3	90
Model en input											
Grafisch bouwen van modellen	30	5	150	5	150	4	120	1	30	1	30
Invoer	30	4	120	3	90	5	150	2	60	3	90
Statistische verdelingen	40	5	200	3	120	4	160	3	120	3	120
Mogelijkheid tot invoer code	50	3	150	2	100	5	250	5	250	5	250
Wachtrij systemen	40	3	120	4	160	4	160	3	120	3	120
Ingebouwde variabelen	30	4	120	2	60	5	150	3	90	3	90
Debugmogelijkheden	30	3	90	2	60	5	150	4	120	3	90
Draaien van het programma											
Aanpassen snelheid	30	4	120	3	90	5	150	2	60	2	60
Executable	20	3	60	3	60	3	60	5	100	1	20
Animatie	40	4	160	5	200	3	120	2	80	2	80
Testen en efficiëntie											
Multitasking	20	1	20	1	20	4	80	2	40	4	80
Uitvoer											
Rapportage	30	3	90	3	90	3	90	3	90	3	90
Grafieken	30	4	120	4	120	3	90	4	120	4	120
Totaal score	590		2000		1850		2130		1470		1600

Tabel 6.2.1 Multi criteria analyse softwarepakketten



6.3 Toelichting weging

In de onderstaande paragrafen zullen de beoordelingen van de verschillende pakketten voor de criteria uitgewerkt worden. De eerste drie paragrafen beschrijven de kop "software" uit de multi criteria analyse. De paragrafen vier tot en met tien beschrijven "Model en input", de overige paragrafen zullen op dezelfde wijze aan de mca gekoppeld worden. De beoordelingen zijn gedaan naar aanleiding van Simulation Evaluation review van E. Valentin, de evaluatie van T.W. Tewoldeberhan en interviews met simulatie-programmeurs.

6.3.1 Software

Onder de kop "softwarebedrijf" wordt de algemene indruk van het bedrijf gewaardeerd. De drie grote softwareproducenten: Rockwell software (Arena), Brooks Automation (Automod) en Tecnomatix Technologies (eM-Plant) worden hier met een vier gewaardeerd, omdat het grote bedrijven zijn waarvan het zeer waarschijnlijk is dat ze in de toekomst blijven bestaan. De twee resterende producenten zijn veel kleiner en werken vooral op nationaal niveau. Prosim BV (Prosim) heeft van deze twee bedrijven de hoogste waardering gekregen, omdat het programma Prosim al lang op de markt is en Tomas nog in de ontwikkelingsfase verkeerd.

De manier waarop de documentatie door de bedrijven wordt verzorgd verschilt aanzienlijk. Bij Tomas is de documentatie nog niet beschikbaar (nieuw programma) en voor Automod bestaat zowel een handboek als een leerboek (soort stappenplan om Automod te leren). Bij Tomas worden wel voorbeeld modellen geleverd, zodat op deze wijze het programma kan worden geleerd. De documentatie van Arena is goed maar minder uitgebreid dan Automod, dit komt vooral door het duidelijke handboek van Automod. eM-Plant scoort minder dan Arena hoewel de helpfiles van eM-Plant wel vrij snel inzicht in het programma verschaffen. Prosim staat tussen eM-Plant en Tomas in er bestaat een handleiding, maar deze is niet van dezelfde kwaliteit als de helpfiles van eM-Plant.

Ondersteuning wordt in principe door alle bedrijven wel geleverd alleen zit er een verschil in de service. De grote bedrijven Automod, Arena en eM-Plant beschikken allemaal over online vraagbaken waar de meest voorkomende vragen op staan en er kunnen vragen op gesteld worden. Bij de kleinere bedrijven kunnen vragen aan de programmeurs van de software worden gesteld.

6.3.2 Model en input

Zowel Automod als Arena zijn beide gebaseerd op een grafische omgeving waar met behulp van het zogenaamde "drag en drop"-principe elementen in gesleept kunnen worden. Op deze manier kan een simulatie worden opgebouwd, die de minimale eisen stelt aan de programmeerkwaliteiten van de gebruiker. eM-Plant beschikt ook over een "drag en drop"-omgeving, deze is alleen niet van hetzelfde niveau als eerder genoemde programma's. Zowel Prosim als Tomas beschikken niet over deze functie.



De invoer van data in de programma's Arena, eM-Plant en Automod kan door middel van het kopiëren van data in tabellen die aanwezig zijn in de programma's. Deze tabellen vertonen grote gelijkenis met Excel, wat het kopiëren vanuit Excel makkelijk maakt. Verder kunnen in de bovengenoemde programma's gegevens ingevoerd worden vanuit databases en vensters. In Tomas en Prosim kunnen de gegevens met behulp van Excel bewaard worden als tekstbestand. Het nadeel hiervan is dat wanneer de gebruiker de data wil veranderen er twee mogelijkheden zijn:

5 de data moet in het tekstbestand veranderd worden.

6 de tabel moet behulp van Excel veranderd worden.

Alleen Tomas beschikt over vensters waar waarden wel in veranderd kunnen worden. eM-Plant krijgt de hoogste waardering, omdat naast de bovenstaande invoermogelijkheden ook de mogelijkheid bestaat om de invoer via webpagina's te laten verlopen. Hierna komt Arena, die alle andere mogelijkheden bezit. Automod bezit dezelfde mogelijkheden als Arena, maar scoort minder omdat het gebruik van de tabellen ingewikkelder is dan bij Arena of eM-Plant. Tomas en Prosim beschikken niet over de interne tabellen en scoren daarom lager.

Arena heeft veruit de beste mogelijkheden om statistische verdelingen in te voeren. Het programma beschikt over een invoer "analyser" waar ruwe (ongeordende) data in ingevoerd kunnen worden, waarna een verdelingsfunctie wordt uitgerekend. Het programma berekent ook de gemaakte afwijking. De overige pakketten zijn gerangschikt naar het aantal standaard aanwezige verdelingen van de pakketten.

Mogelijkheid tot de invoer van code: deze functie heeft sterke connecties met het grafisch bouwen van modellen. De programma's die beschikken over sterke grafische kwaliteiten zijn over het algemeen zwakker in de invoer van code. In dit geval komen Prosim, eM-Plant en Tomas als beste naar voren. In Arena is het ook mogelijk om door middel van code zelf programmablokken te schrijven. Dit wordt alleen niet echt aangeraden. Automod beschikt niet over deze functie. Dit betekent, dat wanneer er geen standaard blokken moeten worden gebruikt, de softwareproducent moet worden aangeschreven om deze blokken te programmeren. Deze functie moet zeker met het oog op Havensimulatie niet worden onderschat, het geeft de software een grotere flexibiliteit.

De wachtrij systemen FIFO en LIFO kunnen in Automod en eM-Plant beide standaard worden ingevoerd. Arena beschikt alleen over FIFO maar dit zou omgebouwd kunnen worden naar andere systemen, alhoewel dit geen makkelijke opgave is. In Tomas en Prosim kunnen alle systemen worden gebruikt, maar deze moeten wel zelf worden geprogrammeerd.

Ingebouwde variabelen zijn de standaard aanwezige variabelen definities zoals integer, real of andere minder bekende date, time, speed. Hiervan heeft eM-Plant de meeste en ook de makkelijkste, zoals bijvoorbeeld datetime waarmee direct naar een bepaalde dag in het programma verwezen kan worden (bijvoorbeeld 19-4-2000 10:00 uur). Arena beschikt ook over een groot aantal basis variabelen. Automod heeft minder van dit soort variabelen en Prosim en Tomas hebben alleen de standaard definities zoals bijvoorbeeld integer en real.



Het gemak waarmee fouten uit de programmacode kunnen worden gehaald (debuggen) is bij eM-Plant het grootst. Bij dit programma kunnen willekeurige variabelen in een apart venster worden bijgehouden. Er kunnen stops in de programmacode worden ingelast, zodat het programma stopt op het moment dat de regel in kwestie gelezen wordt. Tijdens het debuggen is het ook mogelijk een entiteit tijdens het draaien van het programma te volgen. Deze laatste functies is het grote voordeel van eM-Plant boven de andere programma's. Tomas (Delphi) en Prosim hebben ongeveer dezelfde debug mogelijkheden. Het verschil tussen deze twee is het gemak waarmee in Tomas de variabelen bijgehouden kunnen worden. Om deze reden heeft Tomas een hogere score dan Prosim en Arena. Arena heeft een ander systeem om de entiteiten door het model te laten lopen en dit is volgens de een van de gebruikers redelijk goed te debuggen. In Automod is de code bedoeld om de entiteiten aan te sturen niet om nieuwe processen of entiteiten te programmeren. Dit is waarschijnlijk de reden dat de debug mogelijkheden ook wat minder zijn.

6.3.3 Draaien van het programma

Het aanpassen van de snelheid kan zowel bij Arena, Automod als eM-Plant. Het nadeel van Automod is dat iedere keer dat het programma opnieuw wordt opgestart ook de snelheid weer terugvalt op een bepaald referentie niveau. Bij Arena is dit niet het geval waardoor deze hoger scoort. eM-Plant krijgt hierbij de hoogste score omdat hier direct de werkelijke tijd kan worden afgelezen. De scores van Prosim en Tomas zijn het laagst, omdat deze programma's altijd op de maximale snelheid werken.

Het maken van een "executable" of met andere woorden het maken van een zelf-draaiend programma wat direct kan worden opgestart is bij vier van de vijf programma's mogelijk. Deze functie geeft alleen wel een vertekend beeld. Het betekent namelijk niet dat de simulatie zonder het pakket (met licentie) op de computer draait. Dit is maar bij één pakket mogelijk, namelijk Tomas. Vandaar ook de score verdeling Tomas vijf, Arena, Automod en eM-Plant drie en Prosim één.

6.3.4 Animatie

De animatie van de pakketten is gerelateerd aan de grafische omgeving (genoemd in de eerste alinea), waaruit volgt dat Automod en Arena grafisch het sterkst zijn. Er hoeft in deze twee talen het minst geprogrammeerd te worden om grafische uitvoer te krijgen. Hierna volgt eM-Plant waarin een geïntegreerde 2d omgeving zit. Tomas heeft de zwakste grafische applicaties, alle grafische uitvoer zal handmatig moeten worden geprogrammeerd.

6.3.5 Testen en efficiency

Multitasking betekent het tegelijkertijd uitvoeren van meerdere taken. Bij het simuleren kunnen we hierbij denken aan:

- Het openen van meerdere vensters zodat de reeds getypte modules vergeleken kunnen worden met de nieuwe modules.
- Het controleren van het model tijdens het runnen. In sommige gevallen duurt het runnen van een simulatie vrij lang. In dit geval is het makkelijker om tijdens het simuleren de code op fouten te controleren.



De bovengenoemde functies kunnen beide door eM-Plant uitgevoerd worden, het is alleen belangrijk dat tijdens deze controles niets opgeslagen wordt anders is er kans dat het model vastloopt. In Tomas is het mogelijk meerdere vensters te openen maar het is niet mogelijk om tijdens het runnen het model dat op dat moment runt te bekijken. In Prosim kan er wel gekopieerd en geplakt worden tussen de verschillende vensters maar de vensters kunnen niet op hetzelfde moment "open" op het scherm staan. Bij Arena is er niet echt sprake van verschillende vensters. Het programma beschikt over een werkgebied waarin de simulatie gemaakt wordt. Er kan niet in het model gewerkt worden als deze bezig is met runnen. Bij Automod is het rungedeelte totaal afgeschermd van het ontwerpgedeelte. Dit wil zeggen dat wanneer men werkt in het ontwerp gedeelte en men het model wil runnen automatisch een apart programma opstart.

6.3.6 Uitvoer

De rapportage is in de multi criteria analyse genoemd omdat dit zeker behoort tot één van de criteria. Het nadeel is alleen dat er weinig bekend is over de rapportage van de verschillende pakketten. In een vervolgonderzoek zou dit criterium verder uitgezocht kunnen worden. In deze multi criteria analyse is de score voor alle pakketten gelijk gekozen op drie punten.

In ieder programma is het mogelijk om grafieken uit te draaien. De uitvoer van eM-Plant is minder overzichtelijk dan bij de overige programma's, vandaar een mindere score.

6.3.7 Aanschafkosten

De aanschaf van de verschillende pakketten, kan als volgt worden ingedeeld: eM-Plant is van de vier pakketten het duurst (50.000 euro), daarna komen Automod (± 17.000 euro), Arena (± 15.000 euro) en Prosim (± 11.000 euro). Tomas is veruit de goedkoopste optie met ongeveer 300 euro (dit zijn aanschafkosten van Delphi).

6.4 Definitieve pakketkeuze

Automod en Arena zijn vooral gebaseerd op een grafische omgeving waar objecten in geslept kunnen worden. Tomas is een taal die vooral op code is gericht eM-Plant zit tussen deze twee uitersten in. Door deze grafische omgeving zijn zowel Automod als Arena relatief makkelijk te leren. Er zitten echter nadelen aan dit gebruikersgemak. Eén van deze nadelen is, wanneer er een model moet worden ontworpen dat niet standaard aanwezig is. Zoals bijvoorbeeld een havenmodel, waarbij nieuwe blokken moeten worden geprogrammeerd, of de blokken moeten zodanig gemanipuleerd worden dat er een havenmodel ontstaat. Het programmeren van nieuwe blokken kan in deze “voorgeprogrammeerde” omgevingen problemen geven. In Arena bestaat de mogelijkheid tot het maken van nieuwe blokken. Dit is echter geen makkelijke opgave. Bij Automod is deze functie niet aanwezig en zal er met behulp van de aanwezige blokken een model moeten worden gemaakt. Bij de andere twee talen zal het programmeren van nieuwe code aanzienlijk minder problemen geven, omdat deze talen gebaseerd zijn op het schrijven van code. Het nadeel van de op code gebaseerde talen is de gebruiksonvriendelijkheid. De talen zullen alleen door een expert gebruikt kunnen worden hetgeen de gebruikersgroep binnen een organisatie aanzienlijk verkleint.

In de documentatie over de software zijn ook een aantal verschillen. Automod levert de beste documentatie. Daarna Arena waar voor iedere module een aparte helpfile aanwezig is. eM-Plant komt als derde met redelijke helpfiles en als laatste Tomas, waar alleen voorbeeldmodellen bijgeleverd worden.

Er zijn twee methoden om op basis van de mca tot een definitieve keuze te komen:

- 1) De eerste methode is de methode die vaak wordt toegepast bij de faculteit Civiele Techniek. Deze methode is onder andere geschikt voor de kiezen tussen alternatieven van grote civieltechnische projecten. Bij deze methode wordt het totale puntenaantal, dat in de mca toegekend is gedeeld door de aanschafkosten van het pakket. Paragraaf 6.4.1 geeft de keuze weer op basis van de civiel technische methode.
- 2) De tweede methode is de methode waarmee op de faculteit Techniek, Bestuur en Management de verschillende software pakketten gekozen worden. Bij deze methode worden de aanschafkosten als een van de criteria meegenomen in de mca. Deze methode wordt in paragraaf 6.4.2 gegeven.

6.4.1 Civieltechnische methode

In de mca (paragraaf 6.2) worden de totaalscores van de verschillende pakketten berekend. De uitkomsten zijn weergegeven in Tabel 6.4.1. Zoals in de bovenstaande tekst reeds vermeld is zullen de totaalscores gedeeld worden door de aanschafkosten van het pakket. Dit is weergegeven in Tabel 6.4.1. Hieruit volgt dat Tomas per punt de goedkoopste is en eM-Plant de duurste.

	Automod	Arena	eM-Plant	Tomas	Prosim
Totaalscore	2000	1850	2130	1470	1600
Aanschafkosten (euro)	17,000	15,000	50,000	300	11,000
Kosten per punt (euro)	8.50	8.11	23.47	0.20	6.88

Tabel 6.4.1 Overzicht civieltechnische methode.



6.4.2 De methode van de faculteit TBM

De methode van TBM neemt de aanschafkosten van de verschillende pakketten mee als één van de criteria in de mca. Net als de andere criteria krijgt ook het criterium kosten een gewicht mee. Dit gewicht is relatief ten opzichte van de andere criteria. Het criterium kosten krijgt een gewicht van 40 (op een schaal van 10 tot 50). Er zijn in de lijst twee criteria die hoger scoren dit zijn "de mogelijkheid tot de invoer van code" en "documentatie". Deze twee scoren hoger omdat:

- de programmeerkosten vele malen hoger zullen uitvallen wanneer de documentatie slecht is.
- wanneer tijdens het programmeren van de simulatie blijkt dat een bepaald element niet standaard in het programma aanwezig is en deze niet zelf kan worden geprogrammeerd moeten tijdrovende kunstgrepen bedacht worden. In dit geval moeten standaard elementen gemanipuleerd worden om de benodigde elementen te vervangen.

Bij het laatste punt kunnen we bijvoorbeeld denken aan de vervoersstroom door een waterweg, die gesimuleerd moeten worden als een baan waarover AGV's (automated guided vehicles) rijden. De AGV's zullen niet dezelfde eigenschappen hebben als schepen (denk aan de afhankelijkheid van de waterstand in geval van getijbewegingen). In Tabel 6.4.2 is een overzicht gegeven van de methode zoals deze op TBM toegepast wordt. Hieruit blijkt dat eM-Plant de hoogste score heeft met net daaronder Arena. Tomas eindigt met de laagste score.

Criteria	Gewicht	Arena		Automod		eM-Plant		Tomas		Prosim	
		Cijfer	Score	Cijfer	Score	Cijfer	Score	Cijfer	Score	Cijfer	Score
Totaal score			2000		1850		2130		1470		1600
Aanschafkosten	40	3	120	3	120	1	40	5	200	3	120
Nieuwe totaal score			2120		1970		2170		1670		1720

Tabel 6.4.2 Methode TBM

6.4.3 Keuze van methode

In de bovenstaande twee paragrafen zijn twee mogelijkheden weergegeven voor het bepalen van de definitieve pakketkeuze. De eerste is de civieltechnische methode waarbij de totale score door de aanschafkosten gedeeld worden. De tweede mogelijkheid is de methode van de faculteit TBM waarbij de kosten meegenomen worden als één van de criteria in de mca. De keuze tussen deze twee methoden is gedaan op basis van ervaring van de faculteiten op het gebied van het vergelijken van verschillende simulatiepakketten. Op dit gebied heeft de afdeling systeemkunde van de faculteit TBM meer ervaring. Ook zijn er binnen TBM in recent verleden pakketkeuzes gemaakt op basis van de methode zoals deze op TBM gehanteerd wordt. Aangezien in dit rapport eveneens een afweging tussen verschillende pakketten behandeld wordt is er voor gekozen om de TBM methode toe te passen. Hieruit volgt dat eM-Plant met een zeer kleine voorsprong op Arena de hoogste score behaald heeft. Hierna volgen Automod, Prosim en Tomas. De scores van de eM-Plant en Arena liggen zo dicht bij elkaar, dat er op basis van een kleine analyse verder wordt gekozen.



Eén van de belangrijkste functies van het pakket is de flexibiliteit, met andere woorden de mogelijkheid de programmastructuur en de code aan te passen aan de te realiseren haven. Uit het voorgaande blijkt dat eM-Plant flexibeler is dan Arena. Om tijdens het maken van de simulatie niet tegen de beperkingen van het pakket aan te lopen is gekozen voor eM-Plant.

De onderzoeksvraag of er een programma buiten Prosim bestaat waarmee havensimulaties gedaan kunnen worden, lijkt in dit stadium van het onderzoek positief beantwoord te kunnen worden. Het tweede deel van de onderzoeksvraag of het betreffende programma grafisch krachtiger is dan Prosim kan ook positief beantwoord worden. Na het programmeren van de simulatie moeten deze twee deelvragen echter opnieuw worden bekeken.

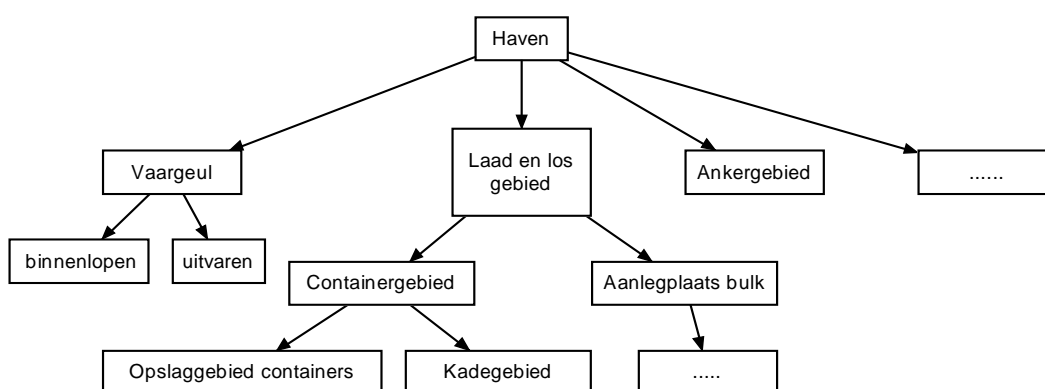


7 Modeleren in eM-Plant

Om beter inzicht te krijgen in de werking van eM-Plant zal in dit hoofdstuk een korte handleiding voor het gebruik van het programma gegeven worden. Als bron is de eM-Plant "reference manual" gebruikt. Dit boek is een van de handleidingen van eM-Plant.

7.1 Opbouw van de modellen

Een van de belangrijkste eigenschappen van eM-Plant is de hiërarchische opbouw van de modellen in het programma. We kunnen dit laten zien door middel van een voorbeeld in de onderstaand boomstructuur, hierbij is een gedeelte van een fictieve haven van graf naar fijn uitgewerkt.



Figuur 7.1.1 Voorbeeld stroomdiagram willekeurige haven

Ieder blok in de boomstructuur stelt een frame voor in het programma. Het hoogste frame is in dit geval de haven. In dit frame kunnen de frames die één niveau lager liggen worden aangeklikt. Wanneer men in dit voorbeeld naar het opslaggebied containers wil gaan moet in het frame haven, eerst het laad- en losgebied worden aangeklikt. In dit frame kan vervolgens het containergebied geselecteerd worden. Tot slot kan in het frame "containergebied" het opslaggebied havens geselecteerd worden.

Deze hiërarchische opbouw van het model maakt het eenvoudiger om bij grote en complexe modellen het overzicht te behouden. Ook kan er op deze manier voor worden gekozen om de onderliggende frames eerst eenvoudig te modelleren. De frames kunnen dan later alsnog worden uitgebreid. Het voordeel hiervan is dat het model al in een vroeg stadium gerund kan worden, dit "kleine" model is minder complex waardoor de fouten er eenvoudiger uit gehaald kunnen worden. Het "kleine" model kan vervolgens steeds verder uitgebreid worden tot de beoogde complexiteit bereikt is. Om het debuggen eenvoudiger te maken is het aan te raden om na iedere uitbreiding het model opnieuw te debuggen.

7.2 Bouwen van modellen

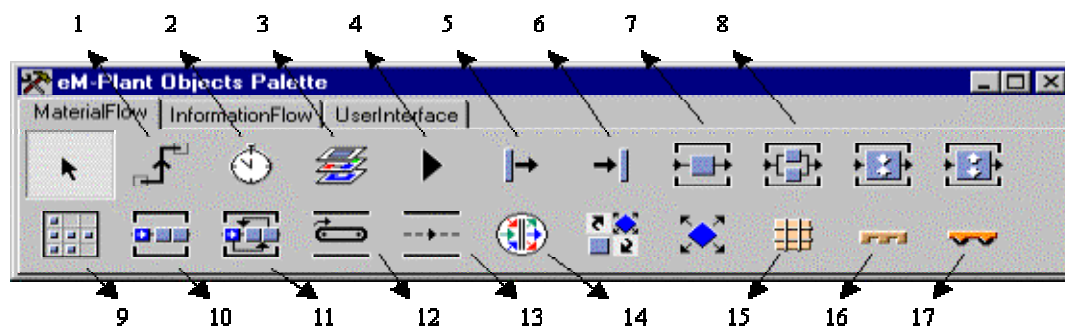
In eM-Plant kan men op verschillende manieren een model maken:

- 1) Men kan er voor kiezen om de vaste objecten van eM-Plant niet te gebruiken en het model in zijn geheel zelf te programmeren.
- 2) Er kan gekozen worden om de vaste elementen te gebruiken.

Door deze keuze kan voor ieder model afzonderlijk een strategie bepaald worden. In het model dat in de vervolghoofdstukken verder is uitgewerkt is gekozen om de standaard elementen samen met de code toe te passen. Om deze reden zal in dit hoofdstuk deze manier van modelleren verder worden uitgewerkt.

De vaste objecten zijn bij elkaar geplaatst op een zogenaamde "object palette". Het object palette is onderverdeeld in drie tabbladen in Figuur 7.2.1 wordt een schermvoorbeeld gegeven.

7.2.1 Tabblad 1



Figuur 7.2.1 Object palette eM-Plant (tabblad materialflow)

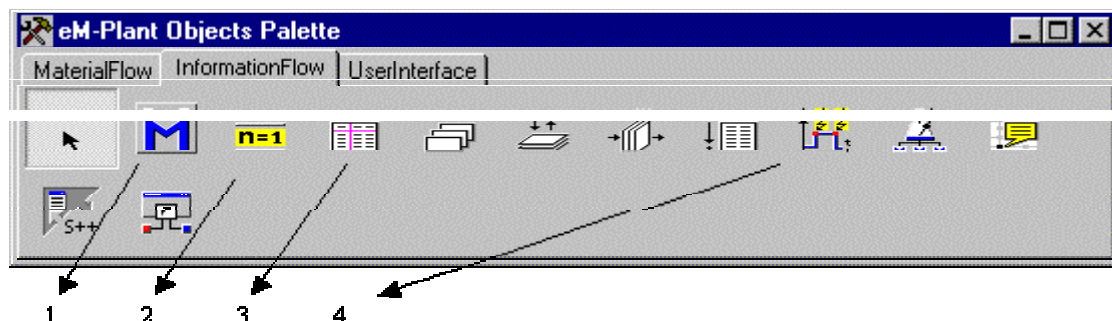
Het eerste tabblad "materialflow" bevat 19 objecten, de belangrijkste objecten zullen hieronder behandeld worden.

1. Connector: met de connector kunnen de elementen die connecties met elkaar hebben met elkaar verbonden worden.
2. Event controller: de "event controller" is het object waarin de simulatie onder andere gestart, gestopt en gereset kan worden. Dit object regelt ook de snelheid van de simulatie.
3. Frame: dit object is in de vorige paragraaf behandeld.
4. Interface: dit object verzorgt de onderlinge verbinding tussen de frames.
5. Source: de "source" is vertaald: de bron van het model, hier worden de entiteiten in aangemaakt.
6. Drain: dit object is het tegenovergestelde van de bron hierin worden de objecten vernietigd.
7. Enkel proces: het object "enkel proces" staat zoals de naam al zegt voor één proces. De entiteiten kunnen hierin één bewerking ondergaan.
8. Meerdere processen: in dit object kunnen de entiteiten meerdere processen ondergaan, het kan gezien worden als een verzameling van 7.
9. Opslag, in deze opslagruimte kunnen meerdere entiteiten vastgehouden worden.
10. Wachtrij, de wachtrij is een plaats waar de entiteiten kunnen verblijven als er geen plaats is om het volgende proces te doorlopen. Een voorbeeld hiervan is een wachtrij bij een postkantoor.

11. Wachtrij, in deze wachtrij kan echter de volgorde van de wachtenden aangepast worden. In sommige gevallen komt het voor dat de ene entiteit prioriteit krijgt boven de andere.
12. Lopende band.
13. Weg voor een AGV (Automated Guided Vehicle), hierover kunnen de AGV's rijden.
14. Richting controller, met deze richting controller kan de richting van de entiteiten gestuurd worden. Er kan bijvoorbeeld aan de linkerkant van het object een "connector" aangekoppeld zitten waardoor de entiteiten binnen komen, aan de rechterzijde kunnen meerdere "connectors" zitten, die de controller koppelen aan een ander object. De controller bepaalt nu (na ingave door de gebruiker) hoeveel entiteiten naar welke "connector" worden gestuurd. Er kunnen verschillende systemen gehanteerd worden zoals wanneer het object, waar de entiteit naar toegestuurd wordt bezet is wordt een ander vrij object gekozen. De entiteiten kunnen ook volgens een bepaalde verdeling verdeeld worden.
15. Entiteit met een bepaalde afmeting
16. Entiteit container
17. Entiteit AGV

7.2.2 Tabblad 2

Figuur 7.2.2 geeft een schermvoorbeeld van het tweede tabblad uit de "objects palette" genaamd "information flow". "Information flow" is een verzameling objecten die te maken hebben met de informatie stroom in het model. Hieronder zullen de belangrijkste elementen behandeld worden.



Figuur 7.2.2 Objects palette eM-Plant (tabblad informationflow)

1. Method, in dit object kan de alle benodigde code geprogrammeerd worden. De ene method kan bijvoorbeeld de andere method aanroepen of een method kan worden aangeroepen als een entiteit een proces aandoet of verlaat.
2. Variabele, dit is een variabele zoals deze ook voorkomt in andere programmeertalen. Aan de variabele kunnen verschillende types gekoppeld worden zoals integer of datumtijd. Wanneer de variabele als object in het programma wordt ingevoerd is deze overal in het programma geldig. Er kan ook gebruik gemaakt worden van lokale variabelen die alleen geldig zijn in een "method".

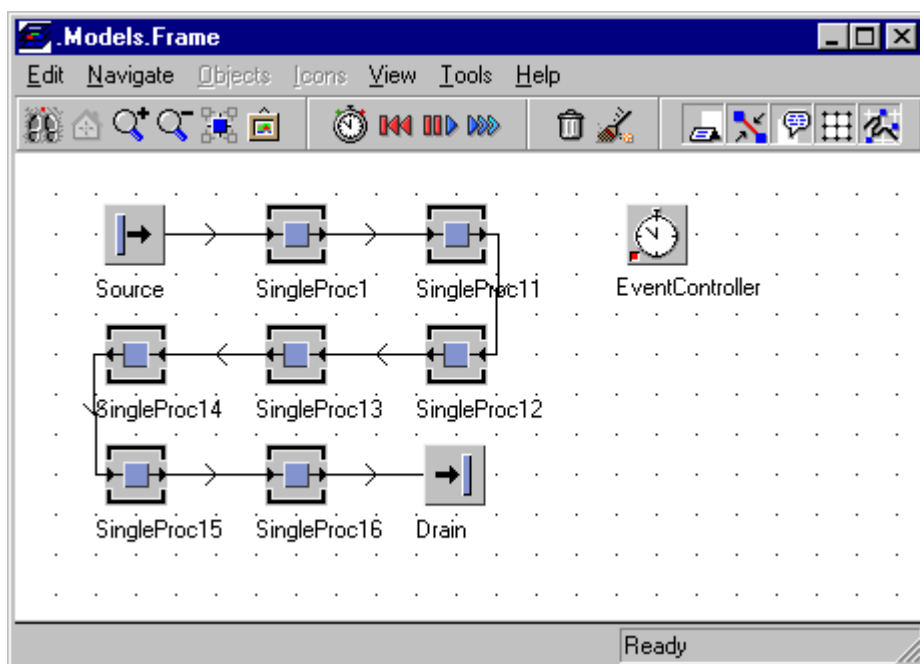
3. Tabel, de tabel is een object waarin de invoer of uitvoer opgeslagen kan worden de presentatie van de tabel vertoont grote gelijkenis met de tabellen in Excel.
4. Trigger, de “trigger” zorgt ervoor dat op een bepaald moment in de tijd iets gebeurt. Een voorbeeld hiervan is dat op een bepaalde datum een aantal schepen de haven binnen moeten lopen. De “trigger” zorgt er voor dat op die bepaalde datum het vastgestelde aantal schepen de haven binnenloopt.

7.2.3 Tabblad 3

Het laatste tabblad is de user interface hierin staan de objecten waarmee grafieken gemaakt kunnen worden.

7.3 Eenvoudig voorbeeld

De hiervoor genoemde begrippen geven een abstract beeld van de opbouw van het programma. Om de werking van de verschillende objecten duidelijk te maken is gekozen voor de uitwerking van een eenvoudig model. Dit model is ontleend aan de reference manual van eM-Plant. Wanneer een van de begrippen niet helemaal duidelijk is wordt verwezen naar de vorig paragraaf.



Figuur 7.3.1 Voorbeeldmodel eM-Plant

Bij het op starten van eM-Plant wordt eerst gekozen voor een nieuw model, vervolgens start het programma een frame op. In dit frame kunnen de objecten uit de “objects palette” gesleept worden. In dit geval is gekozen voor één bron met vijf “single processes” de entiteiten eindigen in een “drain”. De objecten worden stuk voor stuk in de frame gesleept vervolgens worden ze met de “connectors” gekoppeld. In de bron wordt ingesteld hoeveel en welke entiteiten moeten worden aangemaakt. Om het systeem te kunnen draaien moet ook een “Event controller” aan het model worden toegevoegd. In Figuur 7.3.1 is het complete systeem weergegeven.



Dit eenvoudige model is op dit moment klaar voor gebruik. De entiteit zal ontstaan in de bron en zal vervolgens proces na proces doorlopen, om tot slot in de “drain” te verdwijnen. Dit model kan naar inzicht van de gebruiker verder uitgebreid worden, hierbij kunnen we denken aan, extra processen of het uitwerken van een van de processen. De entiteiten kunnen ook bij het binnenlopen van de processen verschillende methods aanspreken die bijvoorbeeld aangeven dat het proces niet toegankelijk is of dat de attributen van de entiteit veranderd moeten worden. Dit zijn slechts een paar van de mogelijkheden, in een van de latere hoofdstukken zal een deel van een havenmodel uitgewerkt worden. Hierbij worden ook de andere objecten behandeld.



8 Project beschrijving

De te modelleren haven is een haven met een totale doorvoer van 3 miljoen ton. Van deze 3 miljoen ton is 1.9 miljoen ton export en 1.1 miljoen ton import. Het grootste gedeelte van deze doorvoer wordt gerealiseerd met 3 producten, kolen, ijzererts en klinker. Door de economische situatie van de afgelopen jaren is er weinig aandacht besteed aan onderhoud en vernieuwing, wat geleid heeft tot een aantal problemen:

- Grote bulk schepen kunnen slechts gedeeltelijk van de haven gebruik maken vanwege de diepgang.
- De capaciteit van de haven voldoet niet meer aan de huidige en toekomstige eisen.
- De afmetingen van de moderne schepen zorgen voor gevaarlijke situaties in het havenbekken.

In Figuur 7.3.1 wordt een overzichtskaat van de betreffende haven gegeven.



Figuur 7.3.1 Kaart haven gebied

8.1 Producten

De drie belangrijkste producten die op dit moment worden overgeslagen zijn kolen, ijzererts en klinker. Kolen en ijzer worden geëxporteerd en klinker wordt geïmporteerd. Voor hoeveelheden zie Tabel 8.1. Deze producten genereren meer dan 70% van de totale handel in de haven. In het totaal hebben in 2000 530 schepen de haven aangedaan waaronder 200 bulkschepen.

Import	Hoeveelheid	Export	Hoeveelheid
Klinker	736.000 ton	Kolen	504.000 ton
		ijzererts	904.000 ton

Tabel 8.1 Import en export overzicht



8.2 Lay-out

De haven is ingesloten tussen twee golfbrekers respectievelijk met een lengte van 2000 en 1450 meter, de ingang van de haven is naar het noorden gericht, heeft een diepte van 10.4-12.8 meter en een breedte van 180 meter (zie Figuur 8.2.1). In het totaal worden er 3 aanlegplaatsen gebruikt voor de overslag van de belangrijkste producten. De kolenkade wordt gebruikt voor de overslag van kolen, kade 1 wordt gebruikt voor ijzererts en boei 1 wordt zowel voor ijzererts, kolen als klinker gebruikt. Voor het vervoer van de producten naar de boei wordt gebruik gemaakt van bakken.



Figuur 8.2.1 Lay-out Haven

In Tabel 8.2 wordt de diepte restrictie van de verschillende aanlegplaatsen vermeld.

Naam	Diepte
Kade 1	8,5 meter
Kolen kade	9,1 meter
Boei 1	10,3 meter

Tabel 8.2 Aanlegplaatsen

8.3 Alternatieven

In het begin van dit hoofdstuk is al naar voren gekomen dat er bij de te simuleren haven een aantal problemen spelen. Deze problemen hebben vooral te maken met de capaciteit en de efficiëntie van de haven. Om deze problemen op te lossen zijn door Boskalis een aantal alternatieven opgesteld:

- Nul alternatief, bij dit alternatief wordt de huidige situatie behouden. Op deze manier kan de huidige situatie makkelijk vergeleken worden met de ander alternatieven.
- Alternatief 2, alternatief 2 geeft twee afzonderlijke aanpassingen. De eerste aanpassing is de realisatie van een extra aanlegplaats. Deze aanlegplaats (boei 3) komt naast boei 1 te liggen en heeft dezelfde functie als boei 1. De tweede aanpassing is het verdiepen van de aanlegplaatsen boei 1 en boei 3 tot 11 met onder CD, alsmede het verdiepen van de aanvaarroute. Door de extra diepte kunnen de schepen meer lading meenemen wat een reductie in het aantal vaarbewegingen geeft (minder schepen nodig om dezelfde vracht te vervoeren). In Figuur 8.3.1 is boei 3 en het te verdiepen gedeelte van de haven aangegeven.
- Alternatief 3, Alternatief drie heeft dezelfde lay-out als alternatief alleen zijn de aanlegplaatsen boei 1, boei 3 extra verdiept tot 13 meter, daarnaast is de aanvaarroute op deze diepte gebracht. Op deze manier kunnen de schepen die de haven aandoen tot de maximale capaciteit geladen kun worden, zodat er minder schepen nodig zijn om de totale vracht te vervoeren.



Figuur 8.3.1 Visualisering alternatieven

9 Conceptualisatie

9.1 Inleiding

Om de complexe werkelijkheid te kunnen modeleren moeten er grenzen aan de grootte van het model gesteld worden. In dit hoofdstuk zullen de verschillende processen met de daarbij behorende vereenvoudigingen en grenzen worden beschreven. Het hoofdstuk zal beginnen met de verschillende grenzen om het model af te bakken. Vervolgens zullen de verschillende processen verder worden uitgediept om beter inzicht te krijgen in het systeem.

9.2 Fysische grens

De haven binnen de golfbrekers, de klinkersteiger en het ankergebied vallen binnen de fysische grenzen van het model. De overige havens en de aanvaarroute door open water vallen hier niet onder, dit heeft tot gevolg dat de aankomsttijden en het soort schip uit een random trekking volgen. De fysische grens wordt weergegeven in Figuur 9.2.1. In dit geval staat de fysische grens voor de fysische grens op vaargebied, de processen aan de kade worden niet meegenomen in de simulatie. Zoals te zien is in Figuur 9.2.1 is ook de uitbreiding met boei 3 weergegeven.



Figuur 9.2.1 Overzicht fysische grens

9.3 Overige grenzen

Om de overige grenzen vast te stellen zijn de verschillende processen op een rij gezet, zodat voor ieder proces een grens kan worden gesteld.

De schepen komen binnen volgens een bepaalde verdeling, na aankomst zal het schip toestemming vragen aan de havenautoriteiten om de haven binnen te lopen. De havenmeester controleert op dat moment een aantal zaken om vast te stellen of het schip door mag varen of moet wachten in het ankergebied. Het ankergebied zal worden geschematiseerd tot een wachtrij waarin een oneindig aantal schepen kan verblijven. De havenmeester zal het volgende controleren:

- Is de vaargeul vrij?
- Is er voldoende waterdiepte?
- Is de aanlegplaats vrij?
- Zijn er voldoende sleepboten aanwezig?

De vaargeul is geschematiseerd tot een aantal rechte stukken waar een bepaalde tijd over wordt gevaren met een draaicirkel. De waterdiepte volgt uit een vooraf berekende tabel, hierin staan de verschillende condities per dag genoemd. Bij het controleren van de aanlegplaats moet ook gecontroleerd worden of er op dat moment geen andere schepen naar de betreffende aanlegplaats onderweg zijn. Voor de sleepboten wordt een aparte rij gemaakt waar ze zullen wachten op werk, dit is ook de rij die gecontroleerd wordt door de havenmeester .

Bij aankomst bij de aanlegplaats wordt het schip aangemeerd, dat zal worden geschematiseerd tot een bepaalde wachttijd. Na het afmeren moet gekeken worden of er bakken nodig zijn om het schip te lossen of dat het schip rechtstreeks op de kade kan lossen. Bij het gebruik van bakken moet worden gekeken of er bakken aanwezig zijn in de wachtrij voor de bakken, als dit het geval is moeten de eerste bakken uit de wachtrij tussen het schip en de wal varen net zo lang tot het schip geladen c.q. gelost is. Het laden of lossen van de bakken aan de wal wordt geschematiseerd door een bepaalde wachttijd. Na deze procedure kunnen de bakken terug varen naar de wachtrij. In het geval dat er geen bakken transport nodig is, zal alleen de laadtijd van de installatie in rekening worden gebracht in de vorm van een wachttijd aan de kade. Na het laden en lossen vragen de schepen opnieuw toestemming aan de havenmeester voor vertrek. Als er toestemming wordt gegeven kunnen de schepen vertrekken en het systeem verlaten.



10 Specificatie

In dit hoofdstuk worden de processen en overige gegevens die een belangrijke rol spelen in het havensysteem beschreven. Er zal uitgelegd worden hoe deze processen in het model geïmplementeerd zijn. In de laatste paragraaf worden de stroomdiagrammen van de verschillende processen gegeven. De overige invoergegevens voor het model zullen ook in dit hoofdstuk behandeld worden.

10.1 Scheepsgegevens

In deze paragraaf worden de scheepsgegevens van de verschillende producten geanalyseerd. De indeling van de paragraaf zal evenals de voorgaande indeling beginnen met klinkerschepen gevolgd door kolenschepen en tot slot zullen de ijzerertsschepen worden behandeld.

In het totaal zijn er in 2000 530 schepen geweest die de haven hebben aangedaan, hiervan vervoerde 93 schepen een van de beschouwde producten, van deze 93 schepen waren 45 klinkerschepen, 14 kolenschepen en 34 ijzerertsschepen. Door de namen van de schepen te koppelen aan een database konden 67 van de scheepsbewegingen worden geïdentificeerd. Er wordt hier over scheepsbewegingen gesproken omdat sommige schepen de haven meerdere keren hebben aangedaan. Over de geïdentificeerde scheepsbewegingen is bekend wat de lengte, de breedte, de geladen diepgang en het tonnage is. In de onderstaande analyse is iedere scheepsoort onderverdeeld in vier klassen, om op deze manier een eenduidige invoer te krijgen.

Door het analyseren van de gegevens kan de invoer van de simulatie nauwkeuriger worden uitgevoerd. Hier moet echter wel een kanttekening bij worden geplaatst: er zijn namelijk eigenlijk te weinig gegevens beschikbaar om een optimale analyse uit te voeren.

10.1.1 Analyse

Om een beeld te krijgen van de aankomstverdelingen, de tonnageverdelingen en de scheepsdiepten zijn al deze gegevens met behulp van Excel in tabellen geplaatst. De verschillende tabellen worden hieronder gegeven.

De drie klassen zijn ingedeeld naar aanleiding van het tonnage van de schepen, voor iedere tonnageklasse zijn daarna de andere gegevens gemiddeld over het aantal schepen in de desbetreffende klasse. In Tabel 10.1.1, Tabel 10.1.2 en Tabel 10.1.3 staan de indelingen zoals deze in de simulatie zijn gevoerd.

Klasse	Aantal	Cumulatief	Gem. DWT	Gemiddelde diepgang	Lengte
I	14	39%	38300	11.1	183
II	14	78%	43950	11.3	187
III	8	100%	46880	11.4	187

Tabel 10.1.1 Klasse indeling Klinkerschepen



Klasse	Aantal	Cumulatief	Gem. DWT	Gemiddelde diepgang	Lengte
I	4	36%	33800	11.5	177
II	4	73%	36600	11.5	179
III	3	100%	41850	11.2	184

Tabel 10.1.2 Klasse indeling Kolenschepen

Klasse	Aantal	Cumulatief	Gem. DWT	Gemiddelde diepgang	Lengte
I	8	40%	18700	9.1	152
II	8	80%	29650	10.2	180
III	4	100%	42100	11.2	187

Tabel 10.1.3 Klasse indeling ijzerertsschepen

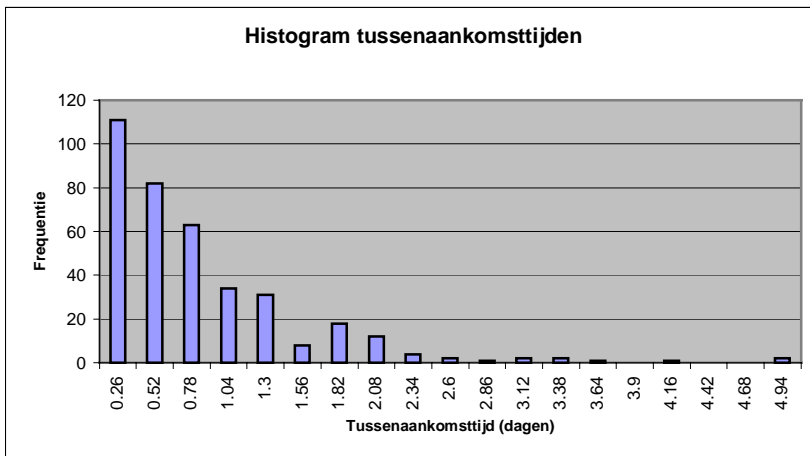
10.2 Toekomstige scheepsafmetingen

De simulaties zijn doorgerekend op de huidige scheepsgrootten. Er is aangenomen dat de scheepsgrootten tot 2010 niet zozeer zullen verschillen. Na 2010 zou de haven graag panamax schepen toe willen laten, dit is echter niet meegenomen in de simulatie.

10.3 Tussenaankomsttijden

Een tussenaankomsttijd is de tijd die tussen de aankomst van twee schepen zit. In de simulatie worden schepen in de aankomstgenerator gegenereerd, op deze manier moet een betrouwbare weergave van de werkelijkheid worden gerealiseerd. De aankomstgenerator creëert door middel van een verdelingsfunctie schepen. Om deze verdelingsfunctie op te stellen moet voldoende data beschikbaar zijn. Vanwege dit dilemma is er voor gekozen om niet ieder scheepstype afzonderlijk te analyseren. De tussenaankomsttijden van de eerste 374 schepen zijn met behulp van een input analyser geanalyseerd met als resultaat dat bij de exponentiële verdeling de kleinste fout optreedt. Voor het output scherm en verdere gegevens zie Bijlage 3. Om toch voor ieder scheepstype tussenaankomsttijden te kunnen genereren zullen eerst "lege" schepen worden aangemaakt met de bovengenoemde exponentiële verdeling. Deze schepen zullen later volgens een uniforme verdeling opgedeeld worden in de verschillende scheepstypen (kolen ijzererts enz.). In Figuur 10.3.1 is de histogram van de tussenaankomsttijden van de schepen gegeven. De bovenstaande methode is een vereenvoudiging van de werkelijkheid, het gevaar bestaat dat er meerdere schepen van dezelfde klasse achter elkaar in de haven aankomen. Vanwege het gebrek aan gegevens is dit echter de meest accurate methode.



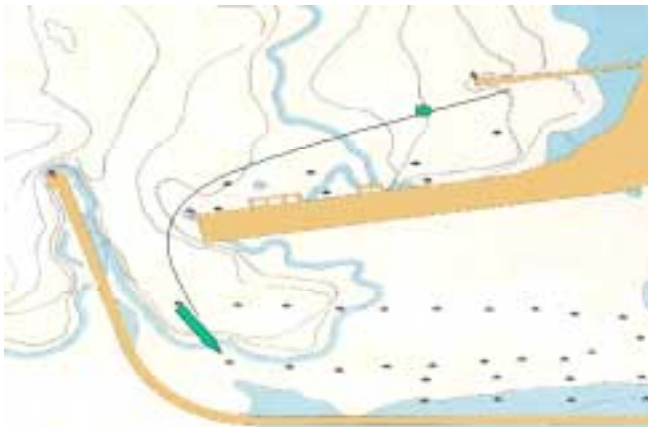


Figuur 10.3.1 Histogram tussenaankomstijden

10.4 Klinker

10.4.1 Laad en los cyclus

Klinker is een import product, de schepen die de klinker komen lossen, meren aan bij boei 1, waarna met behulp van bakken de klinker naar de klinkersteiger wordt vervoerd. Bij deze steiger wordt door middel van twee vaste kranen met grijpers de klinker op een lopende band gestort. Er wordt aangenomen dat de schepen voor 85 % vol geladen zijn als ze de haven binnenlopen (vanwege diepterestricties). Deze hoeveelheid wordt bij de simulaties van alternatief 2 verhoogd naar 90 % (diepte van 11 meter) en bij de simulaties van alternatief 3 waarbij de diepte 13 meter is zullen alle klinkerschepen 100% vol aankomen. Verder zal 50 % van de meegenomen lading in de beschouwde haven worden gelost. De overige lading wordt verscheept naar een andere haven.

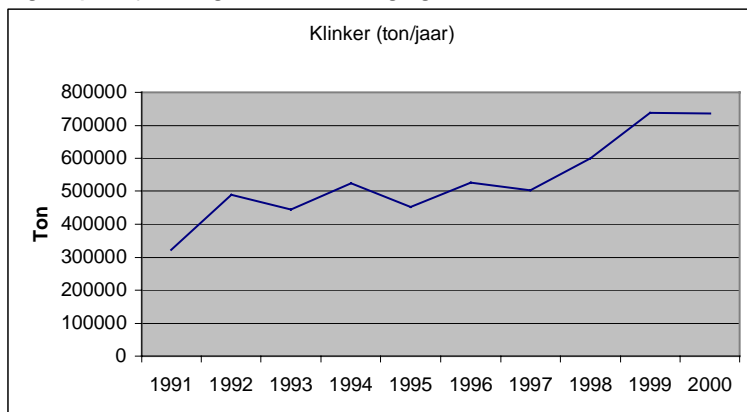


Figuur 10.4.1 Vaarroute klinker



10.4.2 Huidige import

De laatste jaren is de import van klinker gestegen van 324.000 ton in 1991 naar 730.000 ton in 2000, dit is een stijging van ongeveer 8 % per jaar. Het laatste jaar is echter een stabiliteit van de import opgetreden. In Figuur 10.4.2 wordt de import van klinker over de afgelopen jaren grafisch weergegeven.



Figuur 10.4.2 Klinkerimport 1999-2000

10.4.3 Toekomstige doorvoer

De prognoses voor de import van klinker zijn niet optimistisch. De verwachtingen zijn dat er een daling optreedt tot een constant peil van 700.000 ton per jaar in 2010.

10.5 Kolen

10.5.1 Laad en los cyclus

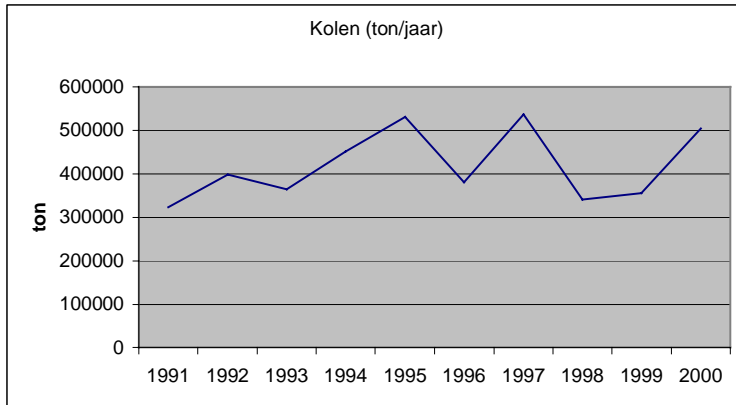
De kolen worden met de trein naar de haven vervoerd, waar ze worden gestort op de kolendump ten noorden van de korte golfbreker. Hiervandaan worden de kolen met een lopende band naar de klinkersteiger vervoerd, waar ze met bakken naar afgemeerde schepen worden gebracht. De schepen worden eerst bij de kolenkade afgemeerd tot ze hier niet verder kunnen worden geladen, vervolgens worden ze bij boei 1 verder gevuld (zie Figuur 10.5.1).



Figuur 10.5.1 Vaarroute kolen

10.5.2 Huidige export kolen

De export van kolen fluctueerde de afgelopen jaren tussen de 324.000 en 531.000 ton met een gemiddelde van 420.000 ton, zoals weergegeven in Figuur 10.5.2.



Figuur 10.5.2 Kolenexport 1991-2000

10.5.3 Prognose kolen export

De prognose voor de export van kolen is goed, de verwachting is dat er een groei optreedt van ongeveer 8% per jaar tot 1.2000.000 ton in 2010.

10.6 ijzererts

10.6.1 Laad en los cyclus

De ijzererts wordt ook met een trein achter de ijzerertskade gestort. De schepen meren eerst aan bij de ijzerertskade waarna ze als ze te diep komen te liggen versleept worden naar boei 1.

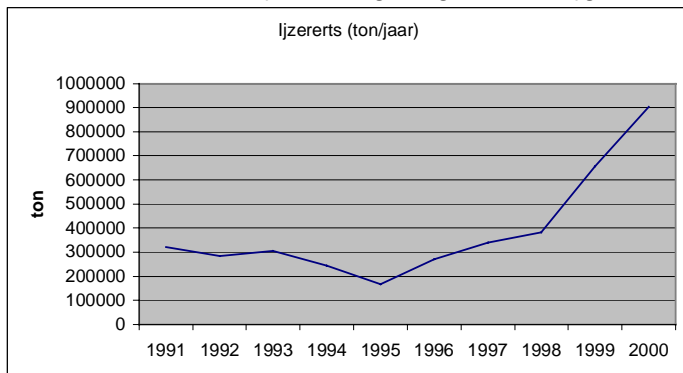
(zie Figuur 10.6.1).



Figuur 10.6.1 Vaarroute ijzererts

10.6.2 Huidige export ijzererts

In de begin jaren 90 bleef de export van ijzererts nagenoeg gelijk op 300.000 ton, pas aan het eind van de jaren negentig is een stijgende trend te zien.



Figuur 10.6.2 IJzererts export 1991-2000

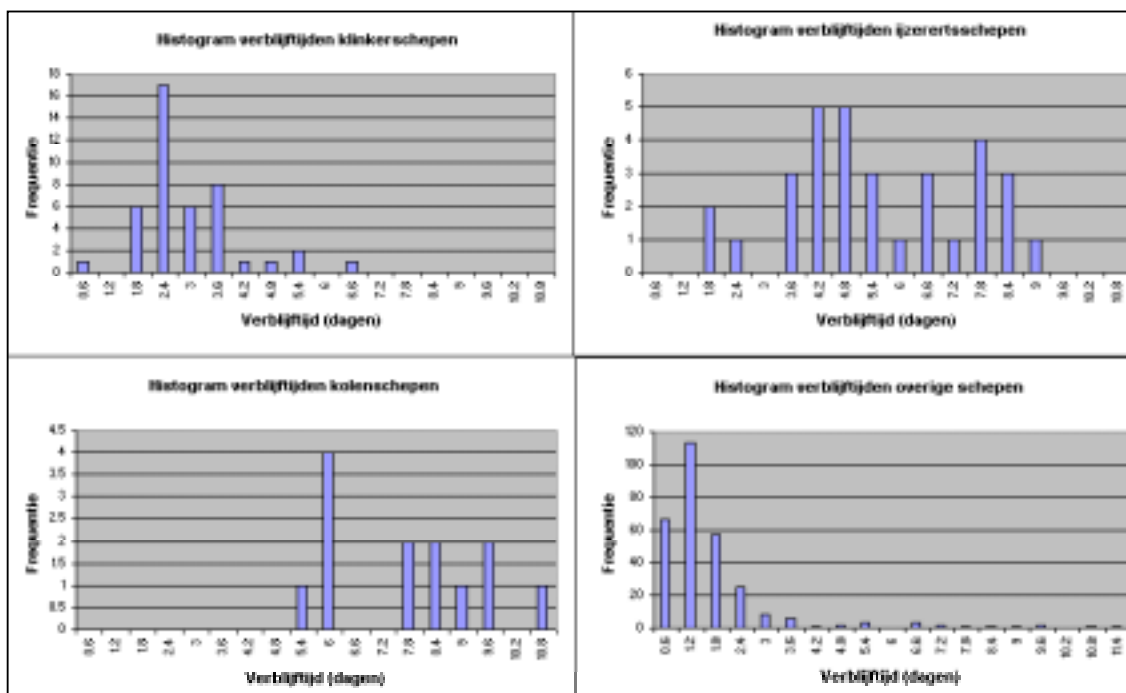
10.6.3 Prognose export ijzererts

De verwachting is dat de stijging van de laatste jaren doorzet in de toekomst, tot een export van 1.300.000 ton in 2010.

10.7 Verblijftijden

De schepen die geen klinker, kolen of ijzererts vervoeren worden geschematiseerd als schepen die volgens de bovengenoemde verdeling aan komen en een bepaalde periode in de haven verblijven. Door analyse van de verblijftijden van de overige schepen is een verdelingsfunctie opgesteld. De verdelingsfunctie is een log normale functie (zie bijlage 3). Voor de histogram van de verblijftijden zie Figuur 10.7.1, in dit figuur staan de histogrammen voor alle scheepstypen weergegeven.





Figuur 10.7.1 Histogrammen verblijftijden schepen

10.8 Bakkentransport

Het bakkentransport wordt op twee plaatsen in de haven ingezet, bij boei 1 en bij de kolensteiger. Bij boei 1 worden de klinkerschepen met de bakken gelost en de kolen en ijzerertschepen worden door middel van de bakken geladen. Bij de kolensteiger worden de bakken ingezet omdat de aanlegsteiger te fragiel is om op te rijden met zwaar materieel. Er zijn twee soorten bakken, bakken van 400 ton en bakken van 200 ton. Het bakkentransport is gemodelleerd als voorraad van 7, hiervan zijn 4 bakken 400 ton en 3 bakken 200 ton,. Iedere keer als een schip bakkentransport nodig heeft zullen vijf bakken naar het schip varen en heen en weer pendelen totdat het schip de bakken niet meer nodig heeft.

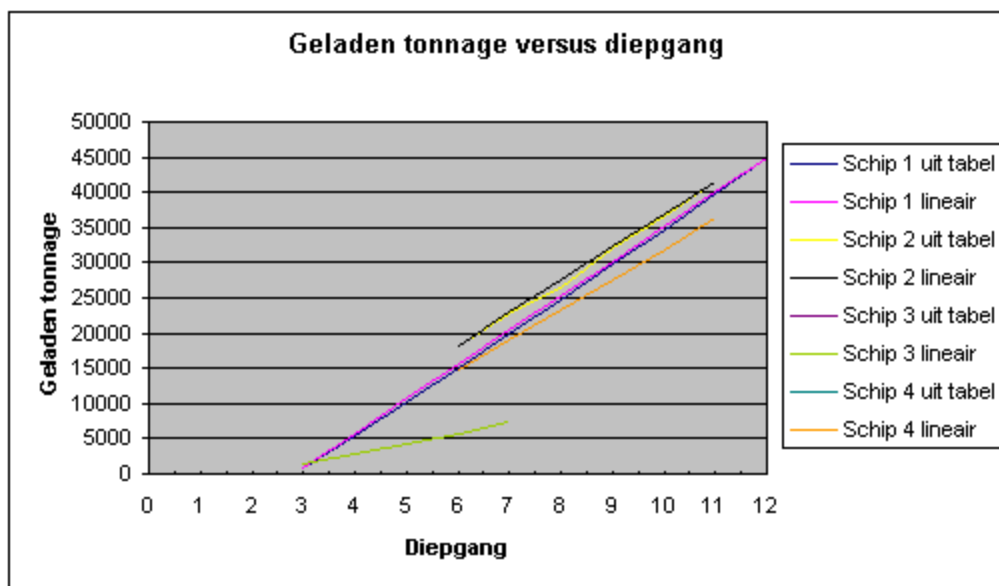
10.9 Relatie diepgang en tonnage

In de te modelleren haven is met name de diepte een belangrijk gegeven. Bij een vergroting van de diepte zullen de kolen en ijzerertschepen in staat zijn meer lading mee te nemen. De klinkerschepen kunnen in dit geval meer klinker invoeren. De vraag is echter hoeveel lading de schepen extra mee kunnen nemen bij een bepaalde dieptetoename. De tabel die de relatie tussen de diepgang en het geladen tonnage weergeeft is in de scheepvaartwereld bekend als "Deadweight scale". Deze relatie is voor ieder schip anders, omdat de scheepsvorm voor ieder schip anders is. In het model is het onmogelijk om voor ieder schip afzonderlijk een tabel in te voeren. Daarom is gekozen om voor ieder schip een vaste relatie aan te nemen tussen diepgang en geladen tonnage. Een lineair verloop van de minimum diepgang tot de maximum diepgang bleek hierbij de beste fit te geven. In onderstaande grafiek is voor vier schepen



het verschil tussen het lineair verloop en het verloop uit de “deadweight scale” tabellen weergegeven. Formule lineair verloop:

$(\text{Diepte vol} - \text{Diepte leeg}) / \text{maximaal tonnage} = \text{meter per ton}$ (extra diepgang van het schip per extra geladen ton).



Figuur 10.9.1 Relatie tussen diepgang en geladen tonnage



10.10 Getij

Vanwege de relatief ondiepe haven heeft het getij een grote invloed op de processen in de haven. Zowel het binnenlopen als het laden dan wel lossen van de schepen wordt sterk beïnvloed door het getij. De klinker schepen kunnen vanwege de geringe diepte niet volgeladen binnenlopen, of moeten ondanks dat de schepen slechts voor 85 % geladen zijn wachten op hoog water. Ook de export schepen hebben te lijden onder de waterdiepte en zijn om deze reden ook afhankelijk van het getij. De schepen kunnen slechts laden tot de waterdiepte ten opzichte van C.D. (minus de overdiepte) plus de waterdiepte die het getij die dag genereert.

Het getij zal worden onderverdeeld in vijf referentie niveaus:

- < CD+0
- < CD+0.5
- < CD+1
- < CD+1.5
- > CD+1.5

Met behulp van het programma tidepred is een tabel gegeneerd waarin iedere zes minuten de waterdiepte is uitgezet. Met deze tabel is voor ieder referentie niveau gekeken wanneer het waterpeil lager was dan het referentieniveau. Op deze manier zijn er intervallen gegeneerd waarin het waterpeil lager is dan het referentieniveau. Tijdens de simulatie wordt voor het schip wat op dat moment binnen wil lopen berekend welke tabel gebruikt moet worden.

Rekenvoorbeeld:

Diepgang schip 11 meter
Ruimte onder de kiel 10 % (1,10 meter)
Diepte 11.4 meter

De diepgang plus de ruimte onder de kiel is in dit geval 12,10 meter, met een diepte van 11,40 meter geeft dit een diepte tekort van 0.70 meter. De tabel van <CD+1meter wordt in dit geval gebruikt. In deze tabel staan de tijden waarin het water onder de CD+1 meter is.

Het programma kijkt of de tijd dat het schip toestemming vraagt om de haven binnen te lopen, in het interval van "niet varen" valt. Tevens wordt gekeken wat de verwachte doorlooptijd van het toegangskanaal is. Deze tijd wordt opgeteld bij de eerste tijd om vervolgens ook te verifiëren dat deze tijd niet binnen het interval valt. Tijdens de laatste controle wordt gekeken af het interval niet binnen de tijd valt van het binnenlooproces.

Tijdens het laden van de schepen (kolen en ijzererts) moet ook rekening gehouden worden met het getij. In dit geval is met behulp van tidepred gekeken naar de minimale waterstanden van de desbetreffende dag deze zijn in een tabel verwerkt. Het programma berekent de minimale waterstand per dag, de schepen controleren iedere dag opnieuw de waterstand waarmee ze de maximale laadcapaciteit kunnen berekenen.



10.11 Processen en Stroomdiagrammen

In deze paragraaf zullen voor de belangrijkste onderdelen uit de simulatie stroomdiagrammen worden gegeven. Op deze manier kan op een relatief eenvoudige en schematische manier een beeld van de simulatie worden verkregen. Het eerste stroomdiagram behandelt de havenmeester deze zal een aantal controles uitvoeren voordat de schepen door mogen varen. Het tweede, derde en vierde diagram behandelt de verschillende scheepstypen, als eerste de klinker schepen vervolgens de kolenschepen en als laatste de ijzerertsschepen. In de laatste twee diagrammen wordt het bakken transport schematisch weergegeven.

De stroomdiagrammen behandelen een aantal processen die nog niet eerder genoemd zijn. Om deze reden worden de processen eerst beschreven voordat de stroomdiagrammen worden gegeven. Het eerste proces is de havenmeester. Dit proces is het controleproces voor de haven, de schepen kunnen namelijk niet altijd de haven binnenlopen. De vaarroute moet bijvoorbeeld vrij zijn (van zowel binnenlopende als wegvarende schepen). Figuur 10.11.1 geeft het stroomdiagram van de havenmeester weer.

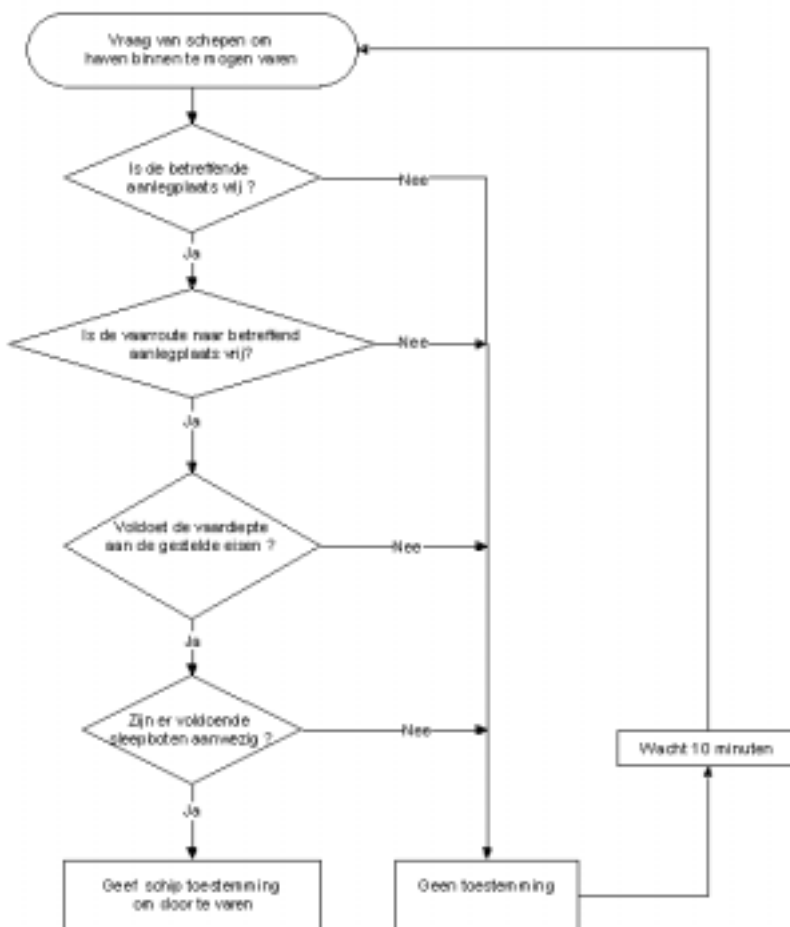
Het tweede stroomdiagram geeft het proces van de klinkerschepen weer. Op het moment dat de schepen de haven aandoen waarschuwen ze de havenmeester. De schepen zijn op dat moment geladen met klinker dat in de haven moet worden gelost. Er is aangenomen dat de schepen op het moment dat ze de haven binnenlopen voor 85 % volgeladen zijn. De klinkerschepen zijn de enige schepen die volgeladen aankomen en leeg wegvaren. Om deze reden hebben de klinkerschepen bij binnenkomst het meeste last van lage waterstanden. Nadat er toestemming van de haven meester is gekregen zal het schip gebruik maken van de sleepboten. Om vervolgens met deze sleepboten naar boei 1 te varen. Hier zal het schip wachten tot de bakken gearriveerd en deze bakken het schip hebben gelost. Nadat de bakken het schip hebben gelost moet het schip 12 uur wachten totdat de papieren in orde zijn. Hierna wordt toestemming aan de havenmeester gevraagd om weg te mogen varen. Het schip moet bij de boei blijven wachten totdat de toestemming is verleend. Als het schip toestemming heeft gekregen zullen de sleepboten het schip vergezellen tot buiten de haven. Het stroomdiagram van de klinkerschepen wordt weergegeven in Figuur 10.11.2.

De kolenschepen zullen net als de klinkerschepen toestemming vragen aan de havenmeester bij het binnenlopen van de haven. Deze schepen zullen alleen minder diep liggen dan de klinkerschepen. Het getij zal daarom niet van belang zijn voor deze schepen. Nadat de schepen toestemming hebben gekregen om de haven binnen te lopen zullen de schepen naar de kolenkade varen. Na het aanleggen wordt met behulp van bakken het schip geladen. Het schip zal worden geladen tot de maximale diepgang of de maximale capaciteit bereikt is. De diepte bij de kolenkade is over het algemeen niet voldoende om de schepen tot het maximum vol te laden. Wanneer de diepte niet voldoende is zal het schip aan de havenmeester toestemming vragen om naar boei 1 te varen. Hier zal het schip verder worden geladen tot de maximale capaciteit of de maximale diepgang bereikt is. Voor het stroomdiagram zie Figuur 10.11.3.



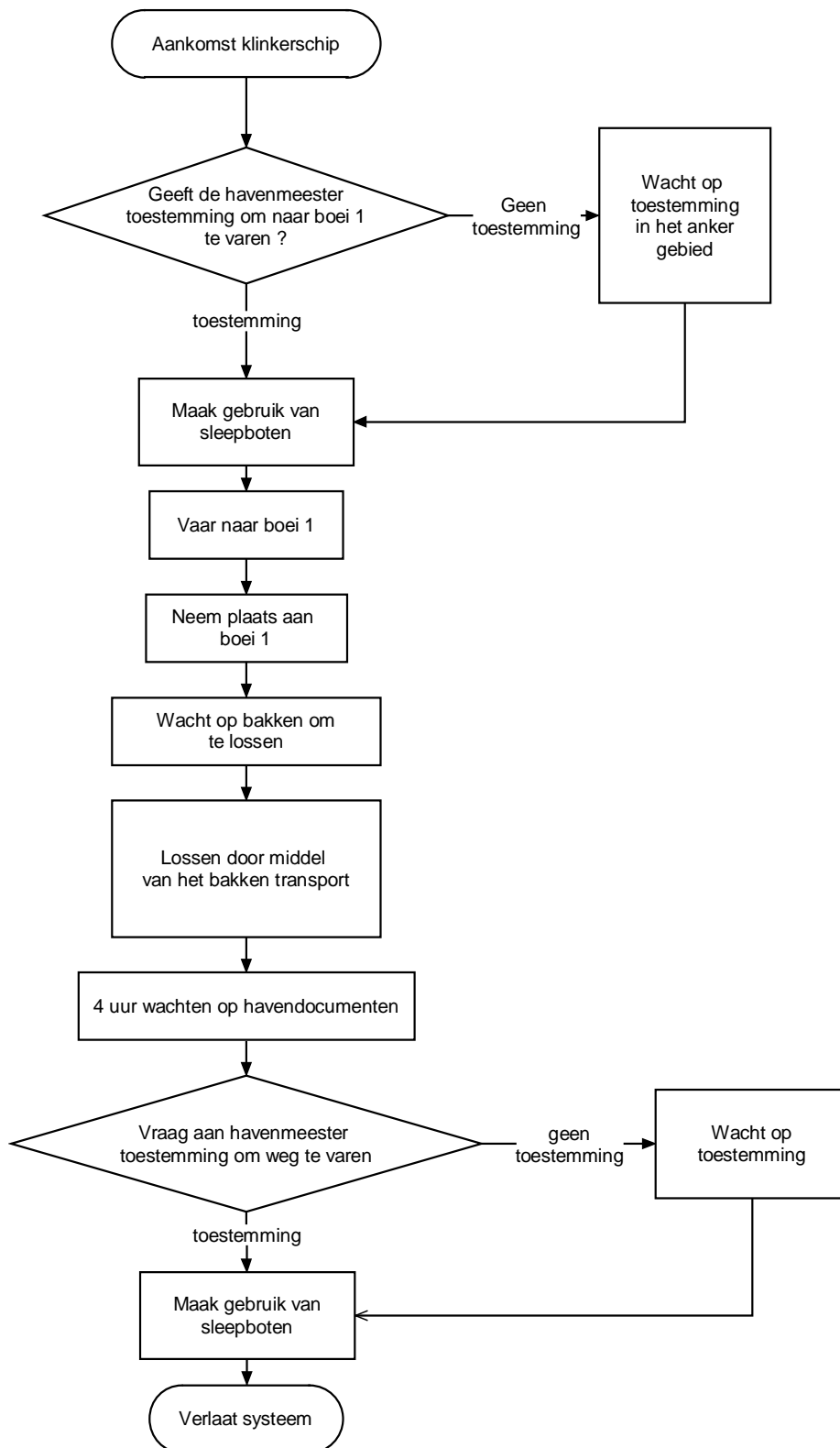
Het begin van het proces van de ijzerertsschepen verloopt hetzelfde als dat van de klinkerschepen. De schepen zullen de havenmeester om toestemming vragen en als ze toestemming krijgen zullen ze naar de ijzerertskade varen. Bij de ijzerertskade wordt het schip echter geladen door middel van een laadinstallatie in plaats van met bakken. De laadinstallatie zal het schip laden tot de maximale capaciteit of de maximale diepgang. Wanneer de maximale capaciteit nog niet bereikt is het schip zal worden versleept naar boei 1 waar het verder zal worden geladen tot ook hier de maximale capaciteit of diepgang bereikt is. Na dit proces zal het schip toestemming vragen om de haven te mogen verlaten. Als de toestemming is gegeven kan het schip wegvaren. Het stroomdiagram van de ijzerertsschepen is weergegeven in Figuur 10.11.4.

De bakken hebben twee verschillende taken, namelijk het laden en het lossen van de schepen. Voor beide taken is een stroomdiagram gegeven in Figuur 10.11.5.



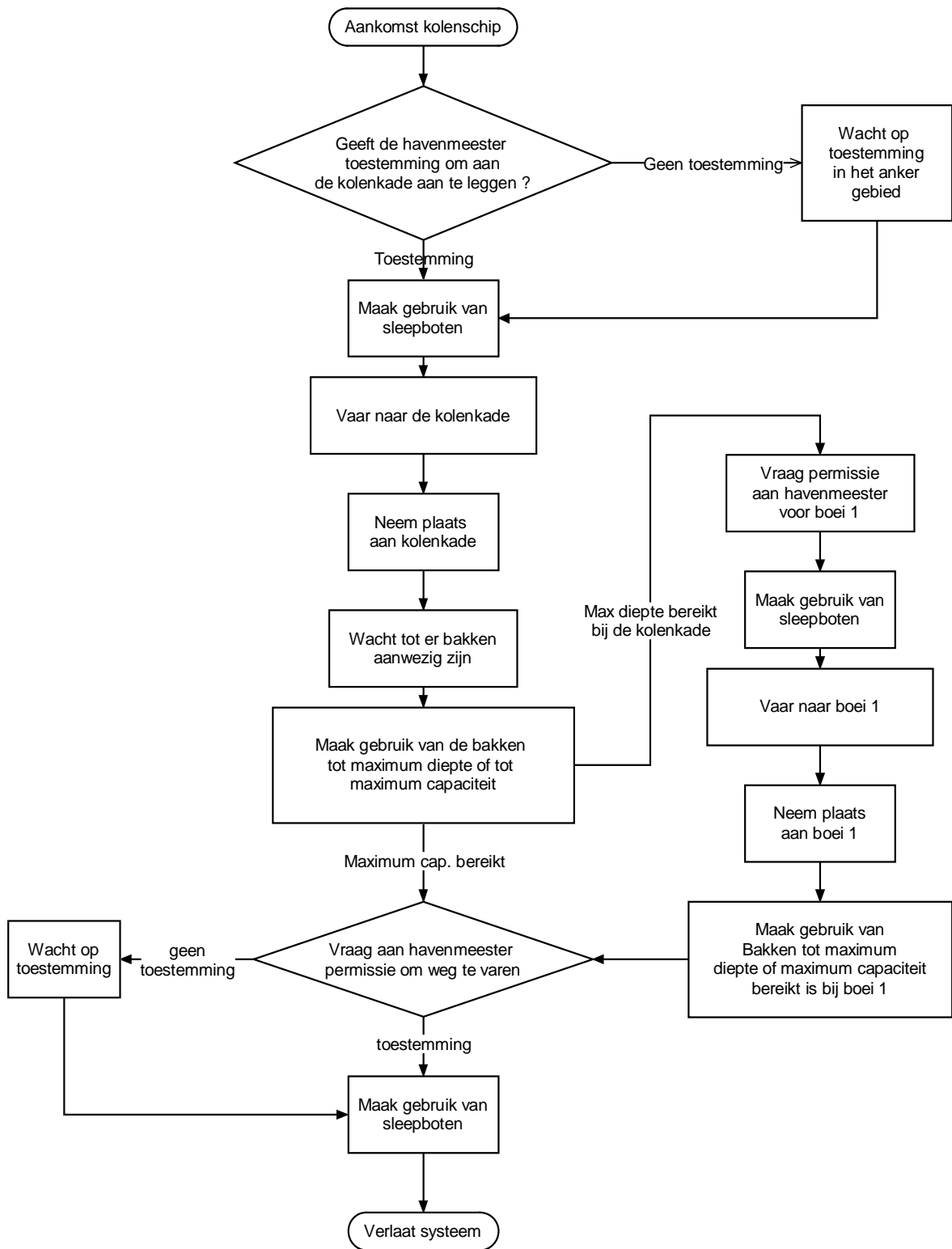
Figuur 10.11.1 Stroomdiagram havenmeester





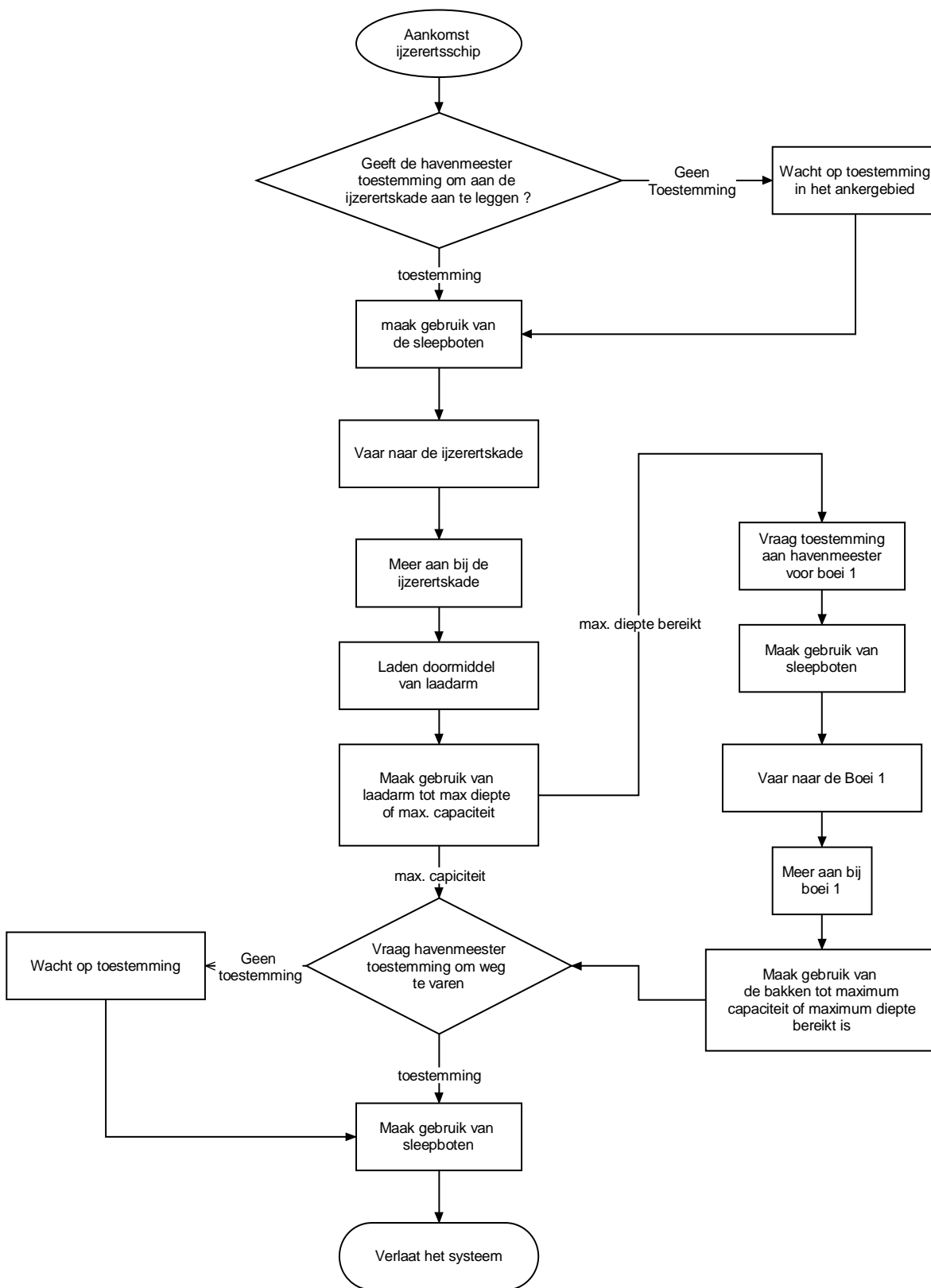
Figuur 10.11.2 Stroomdiagram klinkerschepen





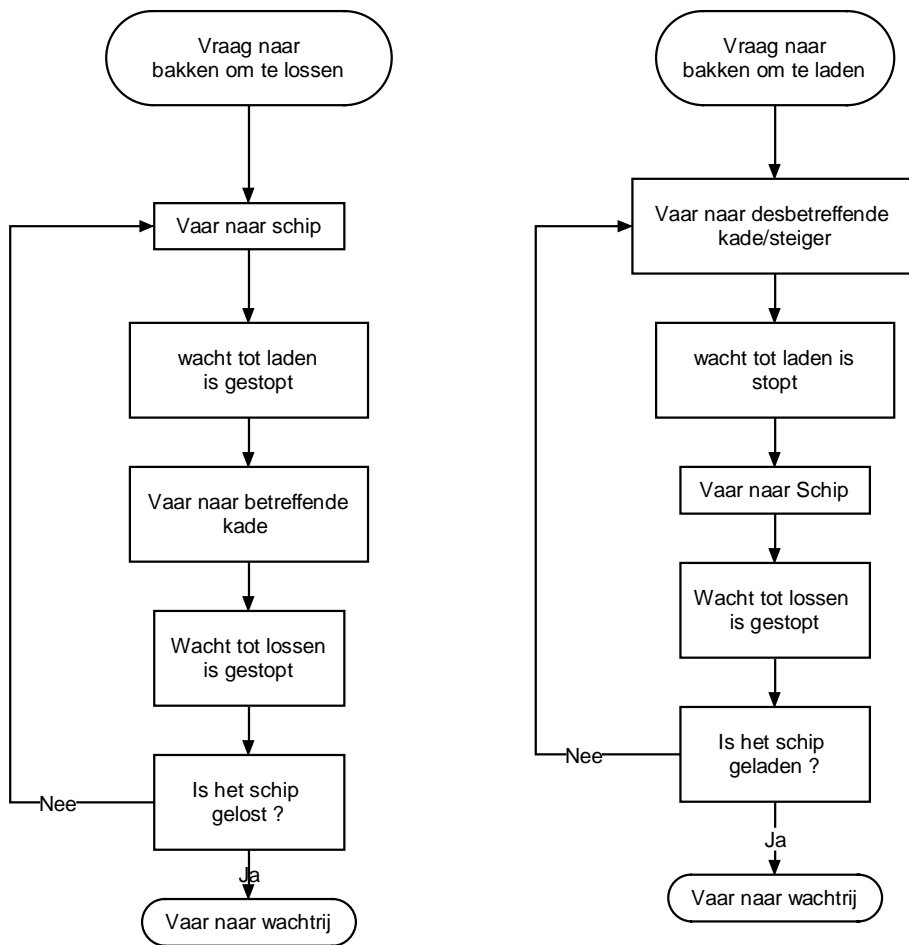
Figuur 10.11.3 Stroomdiagram kolenschepen





Figuur 10.11.4 Stroomdiagram ijzerertsschepen





Figuur 10.11.5 Stroomdiagrammen bakkentransport

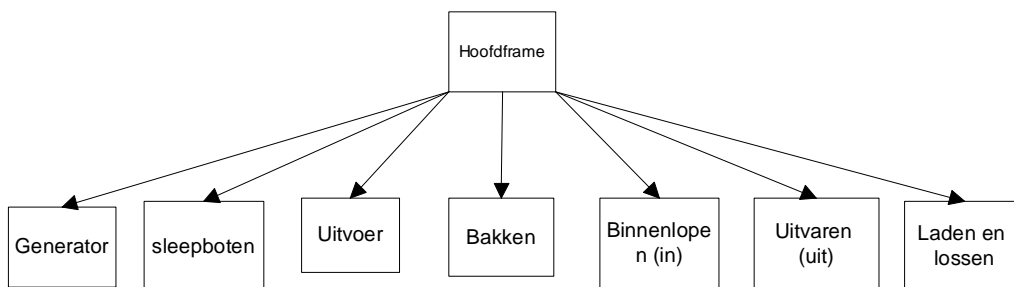


11 Modelbeschrijving eM-Plant

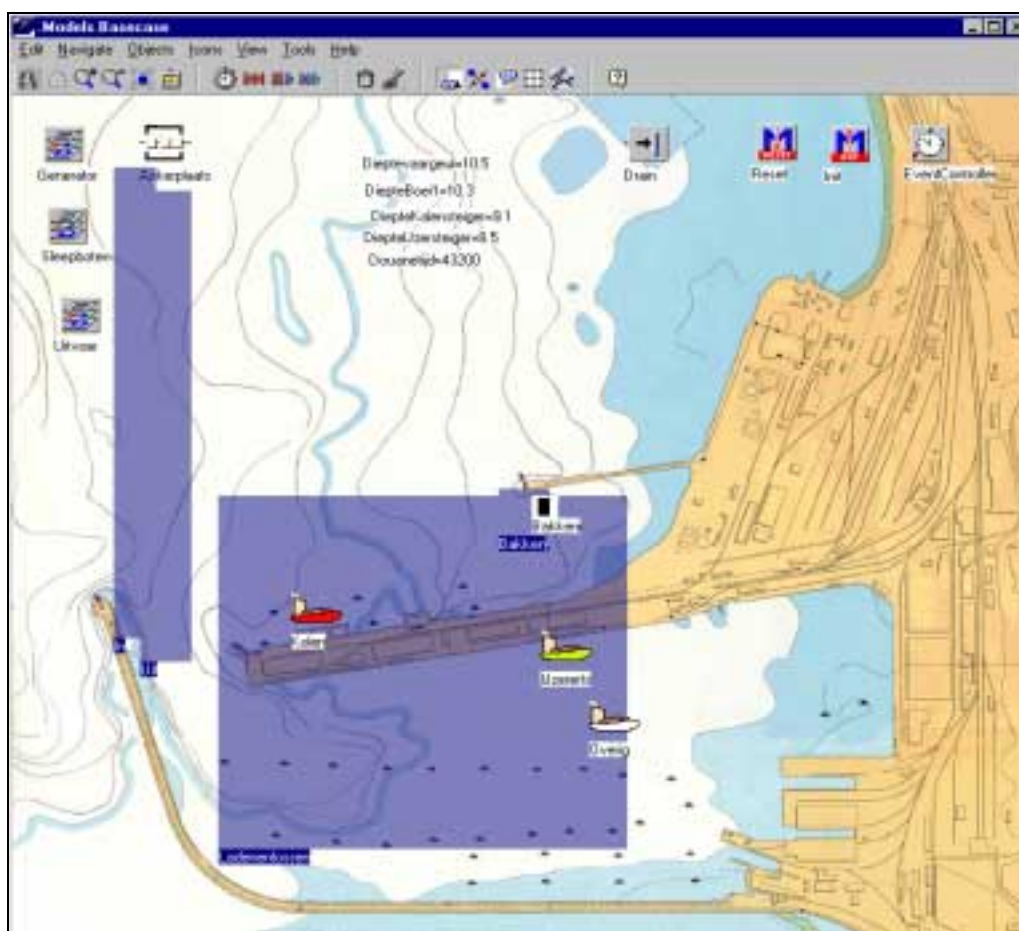
In de vorige twee hoofdstukken zijn achtereenvolgens de conceptualisatie en de invoergegevens behandeld. In dit hoofdstuk zal de modelbeschrijving van het in eM-Plant gerealiseerde model gegeven worden. Deze modelbeschrijving is teruggekoppeld aan de vorige twee hoofdstukken en met name op de paragraaf processen en stroomdiagrammen. In de eerste paragraaf wordt het hoofdframe van het simulatiemodel beschreven. In de tweede paragraaf zal het stroomdiagram van de havenmeester gekoppeld worden aan het simulatiemodel, zodat de werking van eM-Plant en de werking van het model duidelijk wordt. De overige schermafdrucken zijn weergegeven in de bijlage.

11.1 Beschrijving hoofdframe

In Figuur 11.1.2 is het hoofdframe van het model weergegeven. In het hoofdframe zijn alle frames die in de hiërarchie één niveau lager staan gepositioneerd. De hiërarchie is in de vorm van een boomstructuur terug te lezen in Figuur 11.1.1. We kunnen vrij makkelijk de frames in Figuur 11.1.2 van de generator, de sleepboten, de uitvoer, de bakken, het binnenlopen (in), het laden en lossen en het uitvaren (uit) onderscheiden. De blauwe frames zijn in werkelijkheid doorzichtig maar vanwege duidelijkheid zijn ze in het figuur blauw weergegeven.



Figuur 11.1.1 Stroomdiagram van het hoofdframe van het model



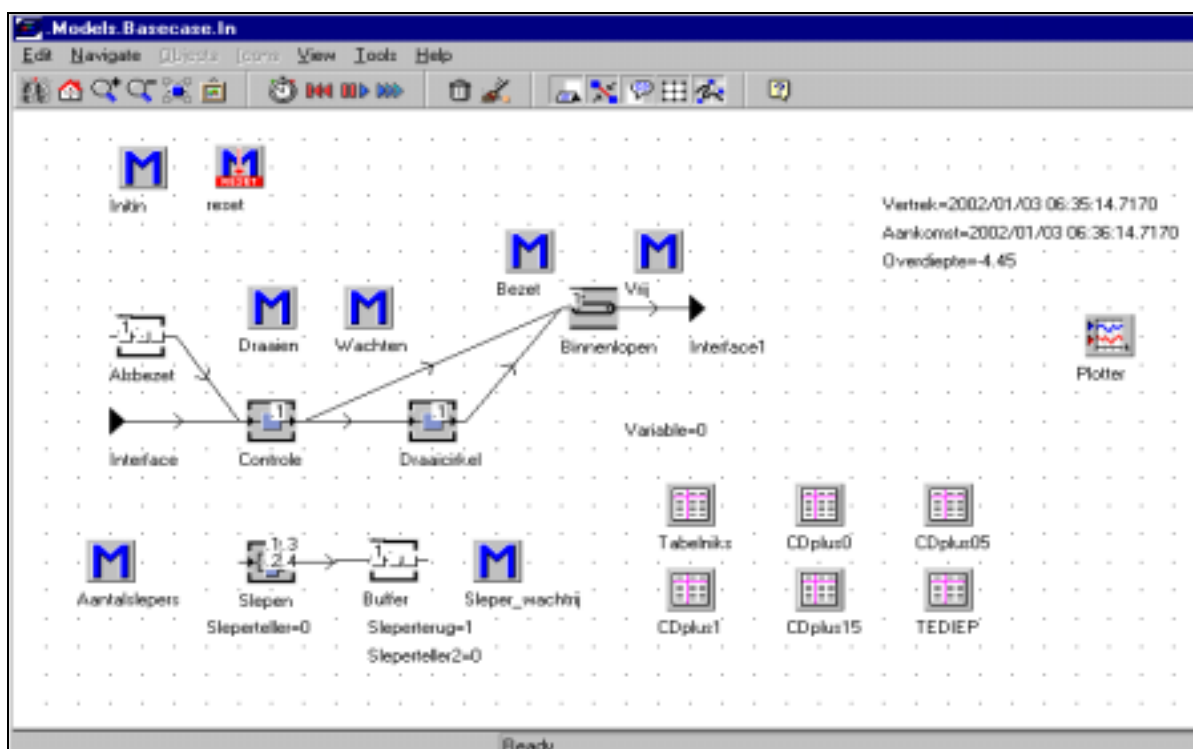
Figuur 11.1.2 Hoofdframe simulatiemodel

11.2 Koppeling stroomdiagram aan simulatiemodel

Het in het vorige hoofdstuk besproken stroomdiagram van de havenmeester speelt zich af in het frame "in". Wat staat voor het binnenvaren van de schepen in de haven. In Figuur 11.2.1 is een grafische weergave gegeven van het frame "in". Dit frame is een schematische weergave van wat er zich afspeelt bij het binnenvaren van het schip. Aan de linkerzijde van het figuur is "interface" te lezen, dit driehoekje is de aansluiting van het frame op het hoofdframe. De schepen varen vervolgens de controle in waar de method "wachten" wordt aangesproken. In deze method worden de controles zoals deze in het stroomdiagram staan uitgevoerd. De controles waren:

- Is de betreffende aanlegplaats vrij?
- Is de vaarroute naar de betreffende aanlegplaats vrij?
- Voldoet de diepte aan de gestelde eisen?
- Zijn er voldoende sleepboten aanwezig?

De laatste controle wordt in het model niet meegenomen, omdat aangenomen is dat er voldoende sleepboten aanwezig zijn. Dit zou eventueel later nog aangepast kunnen worden wanneer men wil kijken wat de invloed is van een beperkt aantal sleepboten op het systeem.



Figuur 11.2.1 Schermafdruck van het frame "In"

In Figuur 11.2.2 staat het eerste deel van de method "wachten" afgebeeld, in het begin van de method staat de eerste controle, de variabelen `root.in.variable`, de variabele `root.uit.variable` en de variabele `root.Ladenenlossen.Boei1verplaatsvar` geven de staat van de vaarroute aan (1 is bezet 0 is vrij), wanneer een deze drie variabelen 1 is moet het schip wachten. De eerste variabele staat voor het binnenvaren, de tweede voor het uitvaren en de derde voor het varen tussen twee aanlegplaatsen. De overige drie variabelen (alle eindigend op aangifte uitvaren) geven aan of er op dat moment schepen weg willen varen, de schepen in de haven hebben voorrang op de schepen buiten de haven. Wanneer een van de drie variabelen op 1 staat (als er een ander schip wil vertrekken) moet het schip dat naar binnen wil varen wachten tot het andere schip vertrokken is.

De tweede controle die het model uitvoert is de dieptecontrole, is de diepte toereikend voor het schip om de haven binnen te lopen. Het eerste deel van de code is bestemd om te bepalen hoe diep het schip ligt, eerst wordt onderscheid gemaakt tussen klinkerschepen en andere schepen. De klinkerschepen zijn de enige schepen van de drie scheepsklassen die in de haven komen lossen, om deze reden zullen alleen de klinkerschepen bij binnenkomst last hebben van te laag water. In de code regel "if soortschip = "klinker"" wordt vastgesteld dat het schip een klinkerschip is daarna wordt bepaald wat de diepte van het schip is.



```

1: 1:
2: 2: Beest : integer;
3: 3: Diepteschip : real;
4: 4: DiepteschipCD : real;
5: 5: Dieptetekort : real;
6: 6: Tabel : object;
7: 7: Tijdsraam, Tijdsinstellingen : Time;
8: 8: Starttijd, StopTijd : DateTime;
9: 9: teller, Bij, Kolom, Toonhoog, v : Integer;
10: 10:
11: 11:
12: 12: if root.In-Variable=1
13: 13:   or root.Hil-Variable=1
14: 14:   or root.Ladeninlassen.Ereil.Aangifteuitvaren=1
15: 15:   or root.Ladeninlassen.Klinkade.Aangifteuitvaren=1
16: 16:   or root.Ladeninlassen.IJzerertakade.Aangifteuitvaren=1
17: 17:   or root.Ladeninlassen.Aangifteuitvaren=1
18: 18:   or root.Ladeninlassen.Ereil.Vergisover=1
19: 19: then until root.In-Variable=0
20: 20:   and root.Hil-Variable=0
21: 21:   and root.Ladeninlassen.Ereil.Aangifteuitvaren=0
22: 22:   and root.Ladeninlassen.Klinkade.Aangifteuitvaren=0
23: 23:   and root.Ladeninlassen.IJzerertakade.Aangifteuitvaren=0
24: 24:   and root.Ladeninlassen.Aangifteuitvaren=0
25: 25:   and root.Ladeninlassen.Ereil.Vergisover=0
26: 26: goto 11
27: 27:
28: 28: DiepteschipCD:=root.DiepteschipCD;
29: 29: if %scort="Klinker" then
30: 30:   Diepteschip:=%Diepteschip;
31: 31: elseif %scort="IJzererts" or %scort="Ereil" then
32: 32:   Diepteschip:=%Diepteschip;
33: 33: else
34: 34:   Diepteschip:=0.1*(Diepteschip+Diepteschip-DiepteschipCD);
35: 35: Overdiepte:=Diepteschip;
36: 36: if Diepteschip<=0.04 then
37: 37:   Tabel:=Tabel1;
38: 38: elseif Diepteschip<=0.05 then
39: 39:   Tabel:=CDplus0;
40: 40: elseif Diepteschip<=0.06 then
41: 41:   Tabel:=CDplus1;
42: 42: elseif Diepteschip<=0.07 then
43: 43:   Tabel:=CDplus2;
44: 44: elseif Diepteschip<=0.08 then
45: 45:   Tabel:=CDplus3;
46: 46: else
47: 47:   Tabel:=TABEL;
48: 48: %Tabel:=Tabel;
49: 49: Toonhoog:=0;
50: 50: Tabel.CursorR:=1;
51: 51: Tabel.CursorT:=1;
52: 52: Starttijd:=Eventstartteller.observatietijd;
53: 53: Vertrekt:=Starttijd;
54: 54: Tijdsinstellingen:=GET_TIME("1:00");
55: 55: Tijdsinstellingen:=GET_TIME("1:00");
56: 56: if %Drainage=true then
57: 57:   Starttijd:=Eventstartteller.observatietijd;

```

Figuur 11.2.2 Frame in.wachten deel 1

Vervolgens wordt het dieptetekort berekend door de diepte van het schip te vermenigvuldigen met 1,1 (diepte schip + ruimte onder de kiel) en hier de diepte van het scheepvaartkanaal ten opzichte van "chart datum" vanaf te trekken. We houden nu het dieptetekort ten opzichte van "chart datum" over. Wanneer het waterpeil onder (CD + het dieptetekort) valt zal het schip niet meer kunnen varen. Voor deze situaties is een aantal tabellen opgezet, de tabellen zijn vernoemd naar het dieptetekort, bijvoorbeeld CD+1 waarin 1 het dieptetekort is. In de tabellen staan de tijden waartussen het desbetreffende schip niet kan varen.

Voorbeeld:

Een klinkerschip wil de haven binnenvaren, het schip steekt 10 meter diep (aanvaarroute is 10,4 meter). Het model identificeert eerst het schip als klinkerschip. Volgens wordt de diepte van het schip op 10 meter gezet (diepgang) en wordt het diepte tekort bepaald (1,1 * diepgang (10)- diepte aanvaarroute t.o.v. CD(10.4)= 0,6 meter). Het dieptetekort valt in de range groter dan 0.5 en kleiner dan 1, hieruit volgt dat tabel CD +1 wordt gebruikt (in deze tabel staan de tijden waartussen niet gevaren mag



worden). Op dit moment wordt gekeken wat de tijd en de datum is b.v. 10-1 2002 12:00 uur. Bij deze tijd wordt de te verwachten vaartijd opgeteld (b.v. 1,5 uur). Tussen 12:00 uur en 13:30 mag het water niet te laag staan, anders zou het schip aan de grond lopen. Er worden nu drie checks uitgevoerd:

1. Valt de eerste datumtijd in een van de periodes uit de tabel?
2. Valt de tweede datumtijd in een van de periodes uit de tabel?
3. Valt een van de periodes uit de tabel in de periode van de verwachte vaartijd?

Als een van de bovenstaande vragen met ja beantwoord wordt krijgt het schip geen toestemming om door te varen en wordt naar de wachtrij "Alsbezet" gestuurd. In deze wachtrij moet het schip 30 minuten wachten voordat opnieuw method "Wachten" wordt aangeroepen.

```

Medele Bascam In Wachten
De 100 800000 Yes No 100 100
57: Datum1jgd:=Datum1jgd+TijdInWachten;
58: elseif $.Draaicirkel=False then
59: Datum1jgd:=Datum1jgd+TijdInWachten;
60: end;
61: Assumer:=datum1jgd;
62: Kolom:=Tabel.CursorT;
63: Rij:=Tabel.CursorR;
64: for q:=1 to Tabel.ydim loop
65:   if Datum1jgd>Tabel[Kolom,Rij] and Datum1jgd<Tabel[Kolom+1,Rij] then
66:     Teondiep:=1;
67:     Datum:=Tabel[Kolom,Rij];
68:     Datum2:=Tabel[Kolom+1,Rij];
69:   end;
70:   if Datum1jgd<Tabel[Kolom,Rij] and Datum1jgd>Tabel[Kolom+1,Rij] then
71:     Teondiep:=1;
72:     Datum:=Tabel[Kolom,Rij];
73:     Datum2:=Tabel[Kolom+1,Rij];
74:   end;
75:   if Datum1jgd<Tabel[Kolom,Rij] and Datum1jgd>Tabel[Kolom+1,Rij] then
76:     Teondiep:=1;
77:     Datum:=Tabel[Kolom,Rij];
78:     Datum2:=Tabel[Kolom+1,Rij];
79:   end;
80:   Rij:=Rij+1;
81: next;
82: Bezet:=0;
83: if Teondiep=0 then
84:   if $.Root="Wlaker" then
85:     if root.Ledeenlussen.Noril.Nevenbezet=1 then
86:       Bezet:=1;
87:     end;
88:     root.Ledeenlussen.Noril.Nevenbezet:=1;
89:   elseif $.Root="LJzeren" then
90:     if root.Ledeenlussen.LJzerenbezet.Nevenbezet=1 then
91:       Bezet:=1;
92:     end;
93:     root.Ledeenlussen.LJzerenbezet.Nevenbezet:=1;
94:   elseif $.Root="Eisag" then
95:     if root.Ledeenlussen.Eisagbezet.Nevenbezet=1 then
96:       Bezet:=1;
97:     end;
98:     root.Ledeenlussen.Eisagbezet.Nevenbezet:=1;
99:   end;
100: end;
101: if Bezet=1 then
102:   $.move(root.In.AlsBezete);
103: end;
104: if Teondiep=1 then
105:   $.move(root.In.AlsBezete);
106: end;
107: if Bezet=0 and Teondiep=0 then
108:   root.Sleephout.Varini:=0;
109:   --Toewijzing verschillende schepen aan verschillende ankerroutes
110:   if $.Draaicirkel=True then $.move(root.In.Draaicirkel);
111:   elseif $.Draaicirkel=False then $.move(root.In.Binnengaan);
112:   end;
113: end;
114: end;

```

Figuur 11.2.3 Frame in.wachten deel 2



De derde controle is de controle of de aanlegplaatsen bezet zijn, dit kunnen we zien in regel 85 hierin wordt gekeken of de aanlegplaats bezet is. Als de aanlegplaats bezet is (`root.ladenenlossen.boei1.bezet=1`) dan wordt de lokale variabele bezet op 1 gezet. Bij regel 101 wordt het schip naar de wachtrij "alsbezet" gestuurd, als de lokale variabele bezet op 1 staat.

Wanneer de controles doorlopen zijn en goed zijn gekeurd (`bezet=0` en `teondiep=0`) dan kan het schip zijn weg vervolgen.



12 Behandelingsopzet, verificatie en validatie

In dit hoofdstuk wordt het model geverifieerd (paragraaf 12.2) en gevalideerd (paragraaf 12.3). Verificatie is het bekijken van het programma op eventuele fouten in de code of met andere woorden het kijken of het programma juist is geprogrammeerd. Bij de validatie wordt gekeken of het model een juiste weergave van de werkelijkheid geeft. Het bovenstaande kan pas gecontroleerd worden, nadat bepaald is hoelang de simulatie moet draaien, wat de opstarttijd is en hoeveel replicaties er gedraaid moeten worden. Dit laatste wordt de behandelingsopzet genoemd (zie paragraaf 12.1). Veel van de in dit hoofdstuk gebruikte termen zijn overgenomen uit het dictaat van het vak TB232 een modelleer vak op de faculteit TBM.

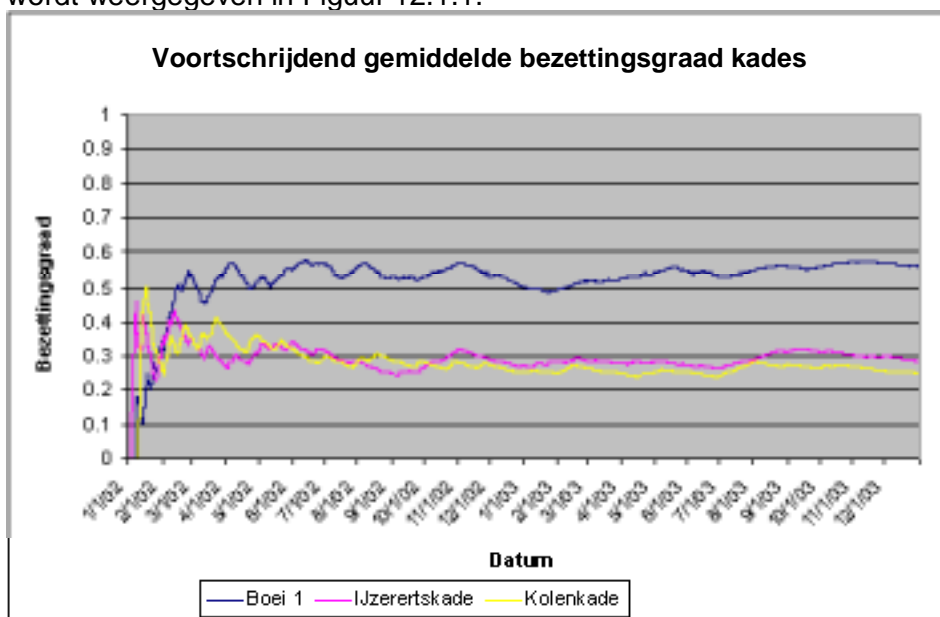
12.1 Behandelingsopzet

De behandelingsopzet bestaat, zoals in de inleiding van dit hoofdstuk al genoemd wordt uit een aantal verschillende stappen:

- 1) Opstarttijd
- 2) Runlengte
- 3) Bepalen aantal replicaties

12.1.1 Opstarttijd

Als eerste zal de opstarttijd van het systeem worden behandeld. De opstarttijd is een bepaalde periode na het starten van de simulatie waarin geen gegevens worden verzameld. De simulatie begint namelijk met lege wachtrijen en aanlegplaatsen, terwijl in werkelijkheid de aanlegplaatsen en wachtrijen nooit helemaal leeg zijn. De opstarttijd wordt bepaald door tijdens de gehele simulatie de belangrijkste variabelen weg te schrijven en deze in grafiekvorm weer te geven. Een van de maatgevende grafieken wordt weergegeven in Figuur 12.1.1.



Figuur 12.1.1 Opstarttijd systeem



In Figuur 12.1.1 is duidelijk te zien dat de bezettingsgraden er een aantal maanden over doen om stabiel te worden. Om aan de veilige kant te zitten is een opstartperiode van zes maanden gekozen.

12.1.2 Runlengte

De volgende stap is het bepalen van de runlengte, de runlengte is de periode waarin waarnemingen aan het systeem verricht worden. De opstartperiode hoort dus niet bij de runlengte. Als vuistregel wordt aangehouden dat de runlengte minimaal drie keer de langste cyclustijd is. De cyclustijden van de kolenschepen kunnen oplopen tot ongeveer een maand. Een runlengte van 20 maanden zou in dit geval meer dan voldoende moeten zijn. Er is echter gekozen voor een runtijd van 40 maanden om het aantal replicaties beperkt te houden (zie volgende paragraaf).

12.1.3 Bepalen aantal replicaties

De laatste stap is het bepalen van het aantal replicaties. Replicaties zijn de herhalingen van een simulatierun. Iedere keer dat de simulatie gedraaid wordt zal er een uitvoer gegenereerd worden. Deze uitvoer is afhankelijk van de verdelingen van de invoervariabelen. Om een juist beeld te krijgen van de uitvoervariabelen zal de simulatie een aantal keer herhaald moeten worden met dezelfde invoerwaarden. Het is belangrijk om de maatgevende uitvoer te kiezen voor deze replicatiebepaling. Aangezien de kolenschepen ten opzichte van de andere schepen de grootste cyclus tijd hebben zijn de uitvoerwaarden van deze schepen gekozen om het aantal replicaties te bepalen. Met behulp van de student t toets is een betrouwbaarheidsinterval voor de verwachte uitvoer berekend. In Tabel 12.1.1 staan de uitkomsten voor de gemiddelde verblijftijd van de kolenschepen. Uit deze gegevens volgt dat met een zekerheid van 95% (t waarde 9,975), dat de gemiddelde verblijftijd van de kolenschepen tussen de 7 dagen 15 uur en 7 minuten en 7 dagen 22 uur en 55 minuten ligt. Het verschil tussen de ondergrens en de bovengrens van bijna 8 uur op een totaal van ongeveer 8 dagen geeft een afwijking van 4,2 %. De overige runs zullen vijf keer gerepliceerd worden, omdat de afwijking van ruim vier procent als een voldoende kleine afwijking wordt aangenomen.

Replicaties	Verblijftijd kolenschepen (dd hh:mm)
1	07 18:06:55
2	07 22:50:24
3	07 16:10:34
4	07 13:53:52
5	08 00:02:03
Gemiddelde waarde	07 19:00:46
Aantal replicaties	5
Afwijking	00 03:52:11
Gemiddelde afwijking	00 01:43:50
t _{9.975}	2.262
h	00 03:54:53

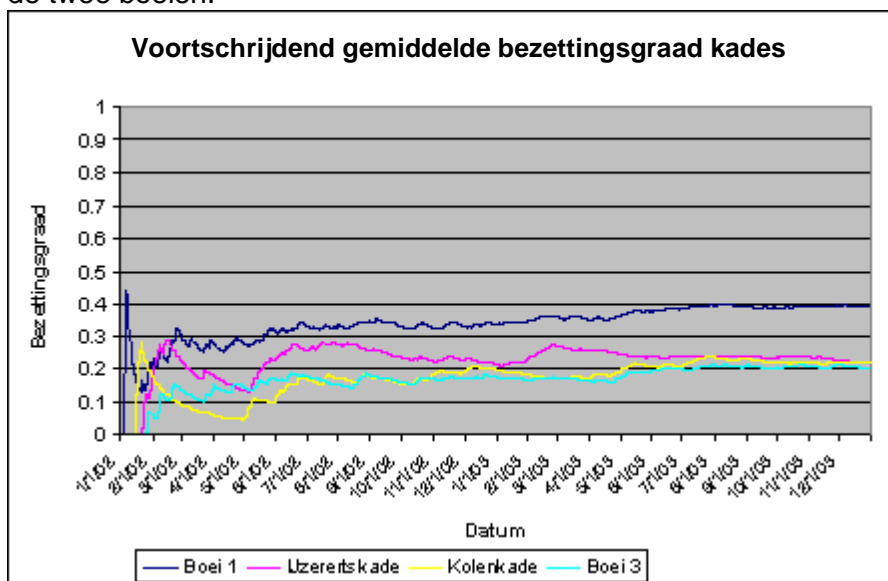
Tabel 12.1.1 Replicatie bepaling



12.2 Verificatie

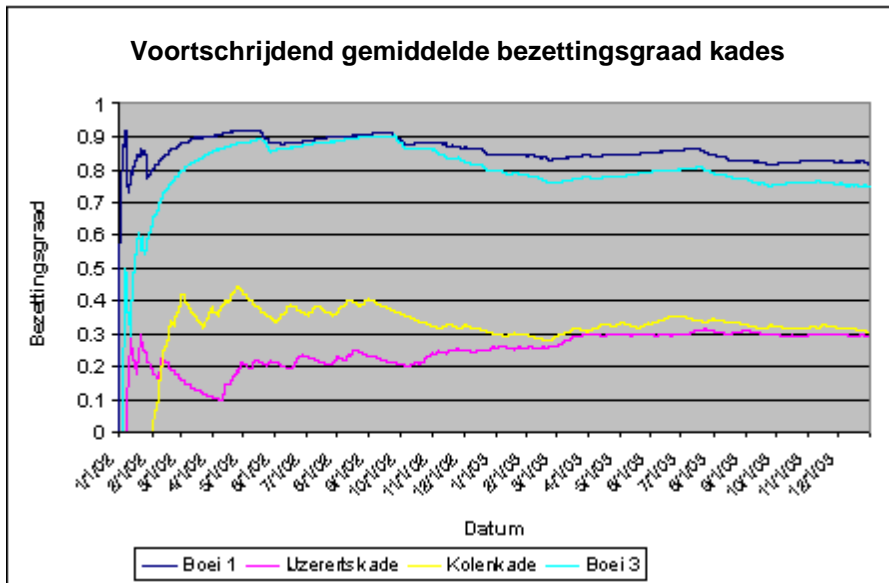
Nu de opstarttijd, de runlengte en het aantal replicaties bekend zijn kan worden gekeken of de het model juist gecodeerd is. Deze controle wordt de verificatie genoemd. De controle kan geschieden door een visuele check van de animatie, deze geeft over het algemeen een goed beeld van mogelijke fouten in het model en door de uitvoerwaarden op fouten te controleren. Tijdens deze verificatie wordt ook naar de autocorrelatie gekeken.

Door het extreem verhogen van de invoerwaarden kunnen de uitvoerwaarden op fouten worden gecontroleerd. Bij de casestudie van de haven is een van de mogelijkheden, extreme waarden te kiezen voor de verschillende scheepstypen. Door eerst voor de klinkerschepen, vervolgens voor de kolenschepen en tenslotte voor de ijzerertschepen extreme waarden te kiezen kan in de grafiek voor de bezettingsgraden gecontroleerd worden of de betreffende bezettingsgraad stijgt. Een voorbeeld zal in deze tekst worden behandeld. Voor verdere verificatie wordt verwezen naar bijlage 3, hier worden de overige scheepstypen geverifieerd. In Figuur 12.2.1 staan de bezettingsgraden met standaard invoerwaarden. Wanneer de hoeveelheid klinker extreem wordt verhoogd zullen de bezettingsgraden van zowel boei 1 als boei 3 sterk moeten stijgen. Boei 1 zal de sterkste stijging vertonen, omdat in het model eerst wordt gekeken of boei 1 vrij is en pas als boei 1 bezet is zal boei 3 worden aangesproken. Ook de bezettingsgraden van de andere aanlegplaatsen zullen stijgen, omdat de schepen die de ijzererts-kade of de kolenkade aandoen vanwege de geringe diepgang verlegd moeten worden naar een van de twee boeien.



Figuur 12.2.1 Bezettingsgraad aanlegplaatsen met standaard invoerwaarden



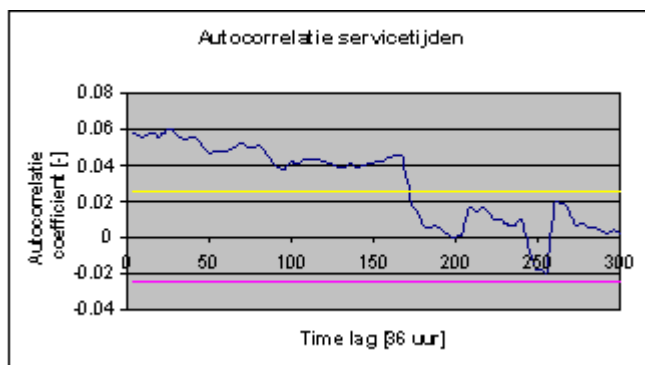


Figuur 12.2.2 Bezettingsgraden met extreem aantal klinkerschepen

De bovengenoemde processen kunnen ook in de grafiek worden waargenomen waaruit opgemaakt kan worden dat de code voor de bepaling van de bezettingsgraden voldoet.

12.2.1 Autocorrelatie

Autocorrelatie is een correlatiecoëfficiënt. Alleen in plaats van de correlatie tussen twee variabelen te berekenen wordt de correlatie tussen twee waarden van dezelfde variabele op twee verschillende momenten in de tijd berekend. Wanneer de correlatiecoëfficiënt tussen de acceptatiebanden ligt van 0.025 en -0.025 is er sprake van een random trekking. In Figuur 12.2.3 is te zien dat de servicetijden van de schepen voldoen aan deze norm, de servicetijden zijn niet gecorreleerd.



Figuur 12.2.3 Autocorrelatie van de servicetijden



12.3 Validatie

Door een model op te stellen van de huidige situatie kunnen de uitvoerwaarden van het model vergeleken worden met de werkelijkheid. De uitvoerwaarden zullen in ieder geval van dezelfde orde grootte moeten zijn om de simulatieruns van toekomstige situaties te kunnen vergelijken met de huidige situatie. Voor de gemiddelde verblijftijd van de verschillende scheepstypen is in Tabel 12.3.1 een vergelijking opgesteld tussen het model en de werkelijkheid. In het model is de gemiddelde verblijftijd van de klinkerschepen 6 uur korter (7 %) dan uit de werkelijkheid kan worden afgelezen.

Voor kolenschepen is dit verschil 5 uur (2.5 %) en voor ijzererts 2 uur (1.5 %). Deze waarden zijn acceptabel, de verschillen zijn procentueel vrij klein. Voor de overige vergelijkingen wordt verwezen naar Bijlage 3.

In onderstaande tabel zijn de waarden uitgedrukt in dd hh:mm:ss

	sim 1	sim 2	sim 3	sim 4	sim 5
Gemiddelde verblijftijd klinkerschip	03 01:26:21	03 04:08:39	03 00:13:48	03 02:48:42	03 03:05:58
Gemiddelde verblijftijd kolenschip	07 18:00:07	07 13:25:04	07 20:25:38	07 21:37:08	07 19:37:46
Gemiddelde verblijftijd ijzerertschip	05 03:26:55	05 00:53:27	05 10:05:00	05 07:53:22	05 07:19:51

	uit simulatie	uit gegevens
Gemiddelde verblijftijd klinkerschip	03 02:20:41	3:08:32:00
Gemiddelde verblijftijd kolenschip	07 18:37:09	7:23:43:00
Gemiddelde verblijftijd ijzerertschip	05 05:55:43	5:08:14:00

Tabel 12.3.1 Validatie van de verblijftijden



13 Experimenten

De experimenten die in dit hoofdstuk worden behandeld zijn opgesteld ten opzichte van de "base case" of in het Nederlands het nulalternatief. In hoofdstuk 7 worden de problemen in de gemodelleerde haven reeds genoemd. Voor de duidelijkheid zullen ze hier herhaald worden. De belangrijkste problemen zijn:

- De beperkte diepgang.
- De capaciteit van de haven voldoet niet aan de huidige en toekomstige eisen.

In het verleden is door Boskalis Westminster een aantal mogelijke alternatieven ter verbetering van de haven opgesteld. Dit resulteerde in drie mogelijke scenario's:

- Nul alternatief (het behouden van de huidige situatie).
- Het verdiepen van de haven tot -11 meter CD plus het aanleggen van een extra boei (boei 3). In figuur Figuur 12.3.1 is de positie van boei 3 weergegeven.
- Het verdiepen van de haven tot -13 meter CD plus het aanleggen van een extra boei (boei3).



Figuur 12.3.1 Overzicht haven inclusief boei 3

Met behulp van het eerder uitgewerkte model zullen de bovengenoemde scenario's met elkaar vergeleken worden. Een goede manier om de verschillende cases te vergelijken is door middel van kosten vergelijking. Hierbij wordt verschil gemaakt tussen twee kostensoorten,

- De kosten voor het verblijf in de haven.
- De kosten van het schip op zee.

13.1.1 Havengerelateerde kosten

De kosten van het verblijf in de haven worden met de volgende formule bepaald (deze formule volgt uit eerder onderzoek door Haskoning in samenwerking met Boskalis Westminster):

$$C_t = 167.4DWT^{0.372}$$

Hierin is C_t de kosten van het schip per dag (US dollar per dag) en DWT is het "DeadWeight" tonnage. (Deze formule geldt voor bulkschepen)

13.1.2 Vaargerelateerde kosten

De kosten van het schip op zee zijn afhankelijk van de afstand die het schip moet afleggen en volgen uit de volgende formule:

$$C_v = 7.7DWCT^{-0.3} d^{0.36} e^{0.69}$$

Hierin is C_v de kosten van de reis per ton (US dollar per ton), d is de afstand (km) en DWCT staat voor de Deadweight Cargo tons. Om DWT om te rekenen naar de maximale DWCT wordt een factor 0.9 gehanteerd. C_v wordt in het model berekend met het geladen aantal ton. (Deze formule geldt voor bulkschepen). De benodigde afstanden zijn gegeven in Tabel 12.3.1

	Gemiddelde afstand (km)
klinker	5250
kolen	4250
ijzererts	7250

Tabel 12.3.1 Gemiddelde vaarafstanden

In de volgende paragrafen zal eerst de uitvoer per alternatief worden besproken, daarna zal er een paragraaf worden gewijd aan de verschillen tussen de drie alternatieven. De simulaties zijn voor een periode van 3,5 jaar gerund. De grootste verschillen tussen de alternatieven zijn te verwachten in de reductie van het aantal schepen. De reductie is goed waar te nemen door uit de invoerwaarden te bepalen wat de hoeveelheid materiaal is die in 3 jaar vervoerd moet worden. Het programma zal vervolgens bijhouden wanneer de hoeveelheid materiaal vervoerd is en hoeveel schepen hiervoor nodig zijn. Voor ieder alternatief zijn twee tijdstippen doorgerekend namelijk 2000 als referentie en 2010 als toekomstverwachting.

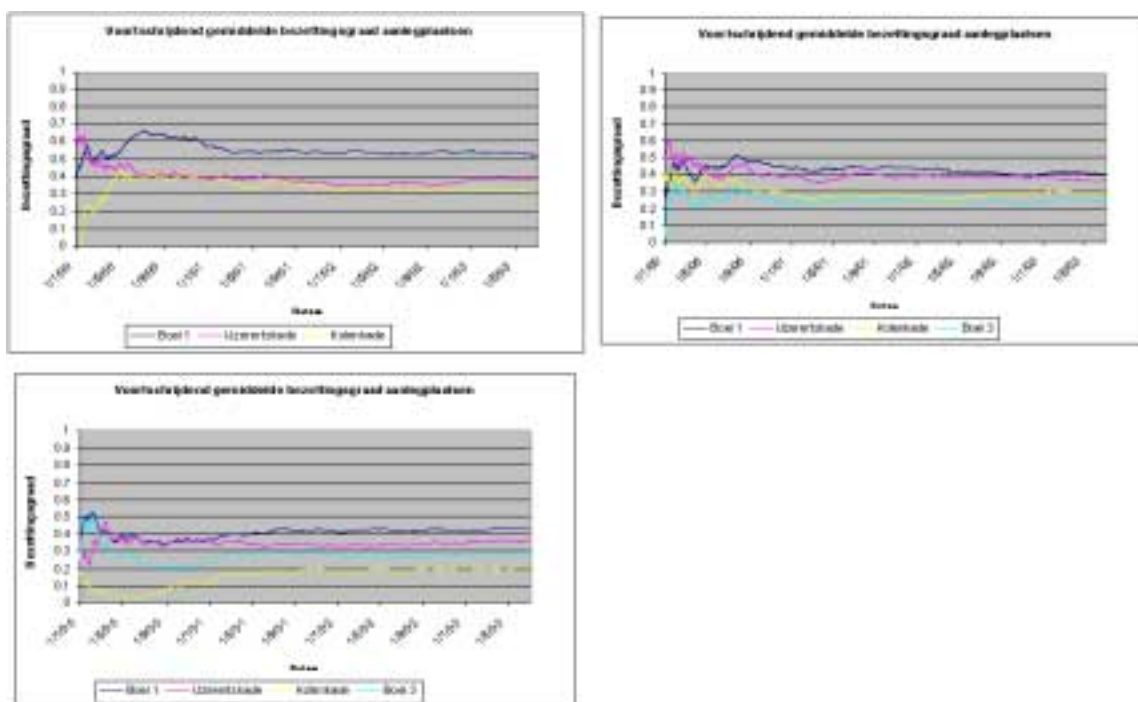


13.2 Uitvoerwaarden

In deze paragraaf zullen de verschillende uitvoerwaarden van de runs naast elkaar worden gezet om de verschillen tussen de alternatieven weer te geven. In het totaal zijn er 6 runs uitgevoerd. De eerste drie runs beschrijven de situatie van het jaar 2000, deze runs staan in Tabel 13.2.1. In de eerste drie kolommen staan de uitkomsten van per drie jaar weergegeven, de tweede drie kolommen geven de hoeveelheid schepen en de hoeveelheid kosten per jaar weer. In de tweede tabel zijn de uitvoergegevens van het jaar 2010 gegeven.

13.2.1 Uitvoer 2000

Uit de gegevens blijkt dat wanneer er een extra boei wordt aangelegd (boei 3) en de haven verdiept wordt naar 11 meter het benodigd aantal schepen afneemt. Dit heeft vooral te maken met de toegenomen diepgang. Er zullen minder schepen nodig zijn om dezelfde vracht te vervoeren, omdat per schip meer vracht meegenomen kan worden. Vooral voor de kolenschepen heeft dit grote gevolgen, aangezien kolenschepen de grootste diepgang hebben (zie Figuur 13.2.1).



Figuur 13.2.1 Bezettingsgraden van de verschillende runs in 2000

Legenda:
 Linksboven: Nulalternatief
 Rechtsboven: Extra aanlegplaats en verdiept tot CD-11
 Linksonder: Extra aanlegplaats en verdiept tot CD-13

De kosten direct gerelateerd aan de tijd dat een schip aan een aanlegplaats ligt stijgen voor klinker, dit wordt waarschijnlijk veroorzaakt doordat de klinkerschepen langer moeten wachten op ander verkeer in de haven. Wanneer er schepen naar boei 1 worden gesleept kunnen de schepen van boei 3 niet wegvaren en andersom is dit ook niet mogelijk. Door deze extra boei kunnen de ijzererts en de klinkerschepen sneller



naar een van de boeien worden versleept, hierdoor worden voor deze twee bulkgoederen de aanmeerkosten lager. De kosten van het totale verblijf in de haven zijn de kosten die gerelateerd zijn aan de totale tijd dat het schip de haven en de ankerplaats bezoekt. Het aanleggen van de extra boei en het verdiepen tot 11 meter heeft ook hier een positief effect op de kosten. Deze kostenbesparingen worden grotendeels veroorzaakt door de aanleg van de extra aanlegplaats (boei 3). Hierdoor neemt de bezettingsgraad van boei 3 af en hoeven de schepen minder lang te wachten. De laatste drie variabelen zijn de kosten die gemaakt worden voor het varen van de schepen. Deze kosten komen direct uit de vaarafstand en de hoeveelheid vracht per schip (Bij een scheepsgrootte toename van 10 procent zullen de kosten met 3 procent afnemen). Aangezien er bij een vergroting van de diepte meer kan worden vervoerd door de schepen zullen de verschepingskosten afnemen. Zoals ook te zien is in de tabel.

De derde kolom geeft het benodigd aantal schepen en de daaraan verbonden kosten weer bij een extra verdieping van twee meter bij boei 1, boei 3 en de aanvaarroute. Door de grotere diepte zullen de schepen meer lading per schip mee kunnen nemen, waardoor er minder schepen per jaar nodig zijn. In plaats van 122 klinkerschepen zullen er nog 111 nodig zijn om de totale vracht van drie jaar te vervoeren. Ook voor de kolen en ijzerertschepen zal dit tot afname van het aantal schepen leiden (zie Tabel 13.2.1). De verblijftijd aan de haven zal nagenoeg gelijk blijven aangezien nog steeds dezelfde hoeveelheid lading overgeslagen moet worden, dat dit door minder schepen wordt gedaan heeft weinig invloed. De verblijftijd van de schepen in omgerekend naar geld in de onderstaande tabel. De totale verblijftijden van de schepen zijn van elkaar afhankelijk. Wanneer de diepte vergroot wordt zullen er per jaar minder schepen nodig zijn. De schepen zullen echter wel langer bij boei 1 en boei 3 liggen, omdat dit de enige twee aanlegplaatsen met voldoende diepgang zijn. Het is niet eenvoudig in te schatten welke effecten zwaarder wegen (dit is een van de redenen om te kiezen voor simulatie). Uit de uitvoer tabel (Tabel 13.2.1) blijkt dat de kosten gerelateerd aan de totale verblijftijd voor kolen en klinker dalen en voor ijzererts stijgen. Deze vergelijking is gemaakt tussen de 11 en de 13 meter varianten. De vaarkosten zullen voor zowel klinker, kolen als ijzererts lager uitvallen, omdat deze uitgerekend zijn naar scheepsinhoud (hoe meer inhoud hoe lager de kosten).

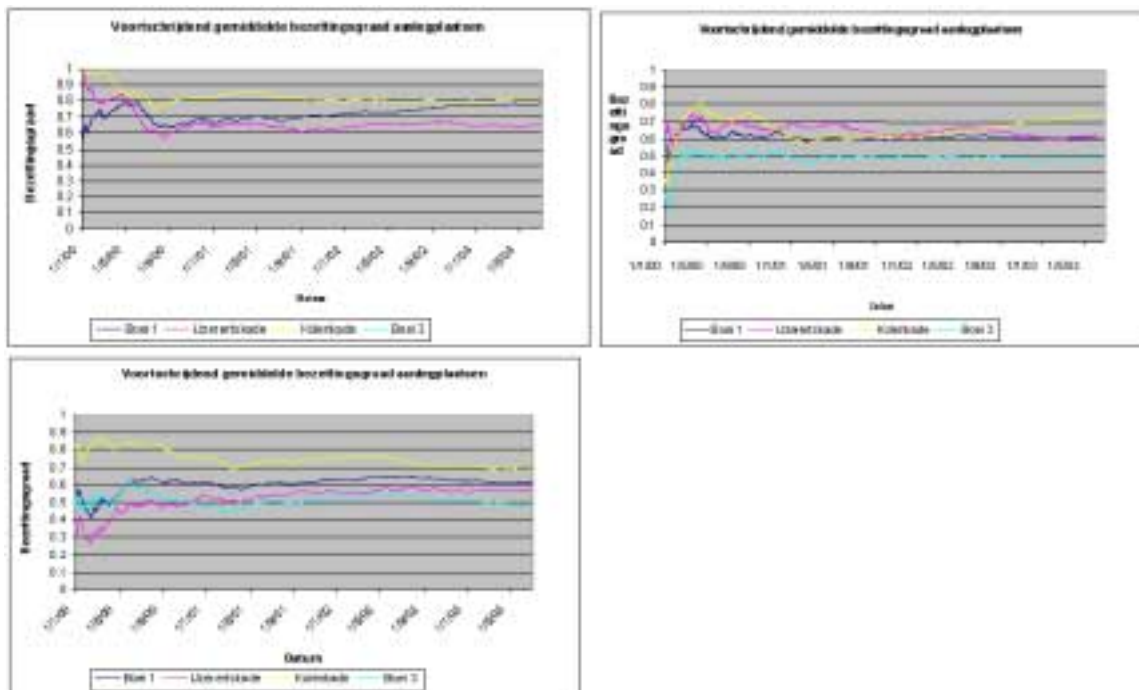
2000	Per 3 jaar			Per jaar		
	0-alternatief	Boei 1 en 3 (11 meter diep)	Boei 1 en 3 (13 meter diep)	0-alternatief	Boei 1 en 3 (11 meter diep)	Boei 1 en 3 (13 meter diep)
Aantal klinker schepen	130	122	111	43	41	37
Aantal kolen schepen	42	38	35	14	13	12
Aantal ijzerertsschepen	105	103	100	35	34	33
Kosten klinker (aan aanlegplaats)	\$3,639,849	\$4,043,281	\$3,910,335	\$1,213,283	\$1,347,760	\$1,303,445
Kosten kolen (aan aanlegplaats)	\$2,643,964	\$2,604,269	\$2,550,214	\$881,321	\$868,090	\$850,071
Kosten ijzererts (aan aanlegplaats)	\$4,341,377	\$4,073,620	\$4,185,652	\$1,447,126	\$1,357,873	\$1,395,217
Kosten totaal verblijf klinker	\$6,114,911	\$4,702,266	\$4,520,720	\$2,038,304	\$1,567,422	\$1,506,907
Kosten totaal verblijf kolen	\$3,228,920	\$3,086,953	\$2,870,930	\$1,076,307	\$1,028,984	\$956,977
Kosten totaal verblijf ijzererts	\$6,040,327	\$5,158,116	\$5,308,641	\$2,013,442	\$1,719,372	\$1,769,547
Vaarkosten klinker	\$62,601,993	\$61,431,471	\$59,723,296	\$20,867,331	\$20,477,157	\$19,907,765
Vaarkosten kolen	\$16,909,700	\$16,317,685	\$15,949,632	\$5,636,567	\$5,439,228	\$5,316,544
Vaarkosten ijzererts	\$45,292,726	\$45,067,527	\$44,591,233	\$15,097,575	\$15,022,509	\$14,863,744
Totaal	\$140,188,578	\$135,764,018	\$132,964,452	\$46,729,526	\$45,254,673	\$44,321,484

Tabel 13.2.1 Uitvoerwaarden 2000



13.2.2 Uitvoer 2010

Wanneer we de uitvoer van 2010 analyseren, blijkt dat de bezettingsgraad van de aanlegplaatsen tijdens alle runs hoog is. Door deze hoge bezettingsgraad moeten de schepen lang wachten, wat lange wachttijden en kosten tot gevolg kan hebben. De run met het nulalternatief heeft van de drie runs de hoogste bezettingsgraad, de aanleg van een extra aanlegplaats zorgt zoals verwacht kan worden voor een afname van de bezettingsgraden.



Figuur 13.2.2 Bezettingsgraden van de verschillende runs in 2010

Legenda:
 Linksboven: Nulalternatief
 Rechtsboven: Extra aanlegplaats en verdiept tot CD-11
 Linksonder: Extra aanlegplaats en verdiept tot CD-13

In Tabel 13.2.2 staat de uitvoer van de verschillende runs. We zien dat wanneer er een extra aanlegplaats gerealiseerd wordt en boei 1 en 3 tot 11 meter verdiept wordt het benodigd aantal schepen omlaag gaat (net als bij de runs in 2000). Vooral voor klinker- en kolenschepen geeft dit een reductie.

De kosten gerelateerd aan de verblijftijd in de haven zullen bij de aanleg van een extra boei waarschijnlijk stijgen, omdat er meer scheepvaart in de haven is en de klinkerscheepen langer moeten wachten voordat ze uit kunnen varen. De aan de verblijftijd gerelateerde kosten blijven voor de kolenschepen ongeveer gelijk. Dit is waarschijnlijk een evenwicht tussen een snellere doorvaart naar boei 1 of 3 maar een minder snelle afvaart van een van deze twee boeien. Voor de ijzerertscheepen worden de kosten wel lager door de mindere wachtrij naar een van de twee boeien.

De totale verblijftijd van alle scheepsoorten aan het systeem wordt lager wat resulteert in lagere kosten. Door de extra aanlegplaats zullen de schepen minder lang hoeven te wachten voordat de haven binnengelopen kan worden.



2010	Per 3 jaar			Per jaar		
	0-alternatief	Boei 1 en 3 (11 meter diep)	Boei 1 en 3 (13 meter diep)	0-alternatief	Boei 1 en 3 (11 meter diep)	Boei 1 en 3 (13 meter diep)
Aantal klinker schepen	130	122	110	43	41	37
Aantal kolen schepen	129	117	109	43	39	36
Aantal ijzerertsschepen	166	163	157	55	54	52
Kosten klinker (aan aanlegplaats)	\$4,778,961	\$5,116,378	\$4,949,960	\$1,592,987	\$1,705,459	\$1,649,987
Kosten kolen (aan aanlegplaats)	\$8,157,123	\$8,238,546	\$8,324,247	\$2,719,041	\$2,746,182	\$2,774,749
Kosten ijzererts (aan aanlegplaats)	\$7,516,471	\$7,308,149	\$7,350,246	\$2,505,490	\$2,436,050	\$2,450,082
Kosten totaal verblijf klinker	\$10,436,771	\$6,334,044	\$5,945,168	\$3,478,924	\$2,111,348	\$1,981,723
Kosten totaal verblijf kolen	\$17,208,060	\$13,630,408	\$13,627,861	\$5,736,020	\$4,543,469	\$4,542,620
Kosten totaal verblijf ijzererts	\$18,072,236	\$10,960,733	\$10,862,338	\$6,024,079	\$3,653,578	\$3,620,779
Vaarkosten klinker	\$62,547,759	\$61,549,815	\$59,596,916	\$20,849,253	\$20,516,605	\$19,865,639
Vaarkosten kolen	\$51,877,668	\$50,337,678	\$49,463,984	\$17,292,556	\$16,779,226	\$16,487,995
Vaarkosten ijzererts	\$71,352,961	\$70,273,813	\$70,039,719	\$23,784,320	\$23,424,604	\$23,346,573
Totaal	\$231,495,456	\$213,086,492	\$209,535,986	\$77,165,152	\$71,028,831	\$69,845,329

Tabel 13.2.2 Uitvoerwaarden 2010



14 Conclusies en aanbevelingen

In dit hoofdstuk worden aan de hand van de resultaten van het onderzoek conclusies getrokken. Het eerste deel van de conclusies is gericht op de keuze van de verschillende simulatie software pakketten. In het tweede deel worden de resultaten en de daaraan te verbinden conclusies van de casestudie uitgewerkt. Vervolgens worden zowel het antwoord op de onderzoeksvraag als de begrenzing van het model gegeven. Het hoofdstuk wordt afgesloten met de aanbevelingen.

14.1 Conclusies eerste deel

Het eerste deel van het rapport is een afweging tussen vijf simulatiepakketten, deze simulatiepakketten zijn:

- Arena (Rockwell simulation software automation)
- Automod (Brooks automation)
- eM-Plant (Tecnomatix)
- Tomas (H.P.M. Veeke en J.A. Ottjes)
- Prosim (Prosim B.V.)

De pakketten zijn door middel van een multi criteria analyse vergeleken. Uit deze analyse is geconcludeerd dat eM-Plant met de gestelde criteria het meest geschikte pakket is om de casestudie mee uit te voeren. De pakketkeuze zoals deze is uitgevoerd in dit onderzoek heeft geleid tot een aantal conclusies en opmerkingen:

1) Aangezien de pakketten niet specifiek geprogrammeerd zijn voor het simuleren van de vervoersstromen in havens is het belangrijk dat het pakket flexibel is. De keuze van een grafisch pakket dat niet flexibel is zou kunnen leiden tot problemen bij het creëren van een simulatiemodel. Geconcludeerd kan worden dat er voorzichtigheid geboden is bij de keuze van een pakket als deze voor een ander doel gebruikt wordt dan waarvoor het gemaakt is. Van de onderzochte pakketten is Automod het minst flexibel, bij dit pakket is het namelijk voor de gebruiker niet mogelijk eigen elementen aan de simulatie toe te voegen.

2) Ieder simulatiepakket heeft zijn eigen interessante kenmerken enkele van deze kenmerken zijn uniek voor dat simulatiepakket. Geen enkel van de onderzochte pakketten beschikt over alle interessante kenmerken.

3) Door de hoge investeringskosten in zowel de aanschaf van het pakket als de investering in de cursus voor de programmeur is het voor bedrijven die relatief weinig simulatiemodellen maken niet rendabel deze modellen intern te modeleren.



14.2 Conclusies tweede deel

In het tweede deel van het onderzoek is een case studie verricht deze studie betreft een haven waarin een aantal problemen spelen. Deze problemen worden vooral veroorzaakt door capaciteitsproblemen en een te geringe diepte. De alternatieven zijn:

- 1) Behouden van de huidige situatie (nulalternatief).
- 2) De aanleg van een extra aanlegplaats in combinatie met een vergroting van de diepte naar CD -11 meter bij twee van de vier aanlegplaatsen.
- 3) De aanleg van een extra aanlegplaats in combinatie met een vergroting van de diepte naar CD -13 meter bij twee van de vier aanlegplaatsen.

Uit het tweede deel kunnen de volgende conclusies getrokken worden:

1) Door de aanleg van een extra aanlegplaats en het verdiepen van de haven tot CD -11 dalen de kosten ten opzichte van het 0 alternatief met ongeveer 1,3 miljoen dollar per jaar in 2000 en 5,4 miljoen dollar in 2010.

2) De aanleg van een extra aanlegplaats en het verdiepen van de haven tot CD -13 geeft een besparing ten opzichte van het nulalternatief van 2,3 miljoen dollar in 2000 en 7 miljoen dollar in 2010.

3) De te verwachten vervoersstromen van 2010 zijn op de grens van de maximaal toelaatbare vervoersstromen. Dit geldt vooral voor het nul alternatief, maar ook voor de andere alternatieven.

-4) Door het verdiepen van de haven kunnen de schepen per schip meer lading meenemen Dit leidt tot een reductie van het totaal aantal schepen.

14.3 Beantwoording onderzoeksvraag

Wanneer we de twee delen van het rapport samenvoegen is kan geconcludeerd worden dat eM-Plant vanwege de flexibiliteit in combinatie met een grafische omgeving zeer geschikt is voor het simuleren van de vervoersstromen in de hedendaagse havens.

14.4 Begrenzing

De resultaten van het onderzoek zijn geldig binnen een bepaald kader. De onderstaande beperkingen geven de rand van dit kader aan.

- Het onderzoek is uitgevoerd voor vijf simulatiepakketten er zijn echter veel meer pakketten op de markt. De mogelijkheid bestaat dat een ander pakket betere specificaties heeft dan de in dit rapport onderzochte pakketten.

- De in dit rapport gebruikte criteria uit de multi criteria zijn op basis van de criteria van Nikoukaran opgesteld. Het is mogelijk dat toekomstige gebruikers andere eisen stellen aan het pakket. Hierdoor kan de uitslag van de multi criteria analyse beïnvloed worden.



14.5 Aanbevelingen

Hieronder worden een aantal aanbevelingen gedaan voor eventuele vervolgstudies.

- Dit onderzoek heeft aangetoond dat het mogelijk is om met eM-Plant de vervoersstromen in hedendaagse havens te modelleren. Als vervolgonderzoek zouden de minder flexibele programma's zoals Arena en Automod getest kunnen worden voor vervoersstromen in havens.
- Ondanks de verbeteringen ten opzichte van het nulalternatief zal de haven in 2010 tegen zijn maximale capaciteit aan zitten. Het verdient aanbeveling om nieuwe alternatieven te genereren om de wachttijden acceptabel te houden.
- Eén van de begrenzings van het model is het aantal onderzochte pakketten. Er is reeds opgemerkt dat er een groot aantal vergelijkbare softwarepakketten op de markt is. Een toekomstig onderzoek kan zich richten op de bruikbaarheid van deze pakketten voor het simuleren van havens.



15 Afstudeercommissie

De afstudeercommissie ziet er als volgt uit

Prof Ir H. Ligteringen (TU Delft faculteit Civiel Techniek en Geowetenschappen)

Tel:

E-mail: h.ligteringen@citg.tudelft.nl

Ir R. Groenveld (TU Delft faculteit Civiele techniek en Geowetenschappen)

Tel:

E-mail: [R. Groenveld@citg.tudelft.nl](mailto:R.Groenveld@citg.tudelft.nl)

Ir J.J. van de Looij (Boskalis)

Tel. 078-6969685

E-mail:

Ir T.W. Tewoldeberhan (TU Delft faculteit Technische bestuurskunde en Management)

Tel:

E-mail: tamratt@tbm.tudelft.nl

Algemene informatie

15.1.1.1 Afstudeerder

P.N. van Zijl

Wateringsevest 10

2611 AV Delft

Tel werk: 078-6969579

Tel prive: 06-41268053

Email: Pvzjl@hotmail.com

15.1.1.2 Bedrijf

Boskalis Westminster nv

Rosmolenweg 20, postbus 43,

3350 AA Papendrecht, The Netherlands.

Telefoon 078 6969000. Fax +31 (0)78 6969555.

E-mail: royal@boskalis.nl

15.1.1.3 Vakgroep

Faculteit der Civiele Techniek

Vakgroep waterbouwkunde

Sectie Havens en Scheepvaartwegen

Stevinweg 1

2628 CN Delft



16 Literatuurlijst

Boeken en dictaten

- Dr.ir. Verbraeck, A., Valentin, E.C., Prof. Dr. Sol H.G., *Simulation Masterclass*, 2001 (dictaat TB9309)
- Ir. Versteegt, C., dr.ir.Verbreack, A., drs. Duin van R., ir. Saanen Y., *Simulation of logistic Systems*, 2001 (dictaat TB9427)
- Prof.dr. Sol H.G., dr. ir. Verbreack A., ir. van Til K.P., *Discete modellen*, 1999 (dictaat TB232)
- Shannon, Robert E., *Systems simulation*, 1975
- Banks J, *Handbook of simulation*, 1998
- Nikoukaran J. et al., *Simulation practice and theory* 7 (1999) 219-231
- 1 Banks J, *Getting started with Automod*, 2000
- Tecnomatix, *eM-Plant Product discription*, 2001
- Tecnomatix, *eM-Plant reference manual*, 2000

Internetsites

- Montlick T, <http://catalog.com/softinfo/object.html>, What is Object-Oriented software, Software Design Consultants.
- Hommes B.J., <http://is.twi.tudelft.nl/~hommes/toolsub.html>, Business Process Modelling Tools (by Technique)
- <http://user.cs.tu-berlin.de/~gebert/Simulation/simtab.html>, Marktubersicht uber simulationssysteme
- <http://www.interduct.tudelft.nl/Pltools>, Softwaretools for process integration
- Valentine E, <http://www.tbm.tudelft.nl/webstaf/edwinv/simulationsoftware/>, Discrete event simulation software packages
- Tecnomatix, <http://www.emplant.de/Frames/eMPlant/default.html>, Product discription



Bijlage 1 Simulatie pakketten

Arena

Software

Softwarebedrijf

Algemene indruk van het bedrijf

Het bedrijf is begin 2000 overgenomen door

Rockwell software

Het bedrijf beschikt over een groot aantal produkten,

Door deze grote variëteit is de kans op continuïteit groot

Documentatie

Bestaat er goede en duidelijke documentatie van de software ?

De documentatie is goed binnen iedere module kan een helpmenu worden opgestart

Worden er voorbeeldmodellen bij de software geleverd ?

Ja

Ondersteuning

Wordt er telefonische of andere ondersteuning door het bedrijf geleverd ?

Volgens de website wordt er aan studenten geen telefonische ondersteuning geleverd

Zijn er op de Technische universiteit Delft mensen aanwezig die ondersteuning kunnen/willen geven ?

Er zijn mensen die ondersteuning kunnen geven

Model en input

Grafisch bouwen van modellen

Is het model vooral op code of vooral grafisch georiënteerd ?

De modules kunnen in het programma worden geslept dus vooral grafisch georiënteerd

Invoer

Op wat voor soort manieren kan de invoer worden ingelezen ? (interactief uit file of uit een ander systeem)

de gebruiker kan in vensters invoer inlezen, uit een database, uit tekstfiles en tijdens het runnen kan nog data worden ingevoerd

Statistische verdelingen

welke verdelingen zijn mogelijk ?

Beta, continu, discreet, erlang, exponentieel, gamma Johnson, lognormaal, normaal, Poisson, driehoekig, uniform en Weibull

Mogelijkheid tot invoer code

Is het mogelijk zelf code in te voeren of zelfs de sourcecode te veranderen ?

Ja, het pakket is er alleen niet speciaal voor ontworpen het invoeren van code blijft daarom lastig

Wachtrijen

Kan het programma verschillende wachtrij-systemen zoals FIFO en LIFO aan ?

De rijen zijn standaard FIFO maar kunnen verandert worden naar een andere. Het veranderen is alleen geen makkelijke opgave

Draaien van het programma

Aanpassen snelheid

Kan de programma snelheid worden aangepast ?

Ja

Executable

Kan het programma van een model een zelfdraaiend model maken (executable) ?

Het kan wel maar het programma moet wel op de computer geïnstalleerd zijn

Animatie

Geïntegreerd animatiepakket

Is het animatiepakket geïntegreerd in de software of wordt het er apart bijgeleverd ?

Het animatiepakket is geïntegreerd

Iconen

Worden er iconen bij het pakket geleverd ?

Ja

Is het mogelijk om nieuwe iconen aan te maken ?

Ja

Kunnen er iconen uit andere software ingevoerd worden (zoals bijvoorbeeld CAD of bitmap) ?

Ja CAD kan worden gebruikt

Testen en efficiëntie

Multitasking

Is het mogelijk om tijdens het bewerken van een model op hetzelfde moment een model te "runnen" ?

Nee

Stapfunctie

Kan het model stap voor stap worden gerunt ?

Ja

Terugdraaien van de tijd

Kan het model ook terug in de tijd rekenen

Nee

Uitvoer

Rapportage

Welke rapporten komen standaard voor in het programma ?

Data over de verschillende wachtrijen en servicetijden

Kunnen er zelf rapporten worden opgesteld ?

?

Uitvoer mogelijkheden

Kan het programma zijn uitvoer rechtstreeks in andere programma's zoals excel onderbrengen ?

Arena kan zijn bestanden wegschrijven naar een tekstbestand waarna het later in andere programma's kan worden geopend

Grafieken

Kunnen er met het programma ook histogrammen grafieken en staafdiagrammen worden gemaakt ?

Ja

Gebruiker

Financieel

Kan de software tegen een redelijke prijs voor een bepaalde periode gehuurd worden en wat zijn de kosten van het gehele pakket ?

De kosten van het pakket komen op ongeveer 15000 euro

Automod

Software

Softwarebedrijf

Wat is de algemene indruk van het bedrijf

Het bedrijf (brooks automation) is een grote simulatie software producent dus de continuïteit is waarschijnlijk gewaarborgd.

Documentatie

Bestaat er goede en duidelijke documentatie van de software ?

Ja zeer duidelijk

Worden er voorbeeldmodellen bij de software geleverd ?

Ja

Ondersteuning

Wordt er telefonische of andere ondersteuning door het bedrijf geleverd ?

Ja

Zijn er op de Technische universiteit Delft mensen aanwezig die ondersteuning kunnen/willen geven ?

Ja

Model en input

Grafisch bouwen van modellen

Is het model vooral op code of vooral grafisch georiënteerd ?

Vooraf grafisch

Invoer

Op wat voor soort manieren kan de invoer worden ingelezen ? (interactief uit file of uit een ander systeem)

Invoer van data is niet echt makkelijk

Statistische verdelingen

welke verdelingen zijn mogelijk ?

Continu, exponentieel, gamma, lognormaal, normaal, driehoekig, uniform en weibull

Mogelijkheid tot invoer code

Is het mogelijk zelf code in te voeren of zelfs de sourcecode te veranderen ?

Alle "entities" worden bestuurd door zogenaamde bron bestanden deze bestaan uit zelf in te voeren code, niet bekend of de source code ook kan worden verandert

Wachtrijsystemen

Kan het programma verschillende wachtrij-systemen zoals FIFO en LIFO aan ?

Ja

Draaien van het programma

Aanpassen snelheid

Kan de programma snelheid worden aangepast ?

De snelheid kan worden aangepast, maar springt bij het opnieuw opstarten weer terug op het oude niveau (dus kan wel maar is niet makkelijk)

Executable

Kan het programma van een model een zelfdraaiend model maken (executable) ?

Er kan een exe bestand worden aangemaakt maar deze kan alleen gerund worden als Automod op de computer geïnstalleerd is

Animatie

Geïntegreerd animatiepakket

Is het animatiepakket geïntegreerd in de software of wordt het er apart bijgeleverd ?

Geïntegreerd

Iconen

Worden er iconen bij het pakket geleverd ?

Ja

Is het mogelijk om nieuwe iconen aan te maken ?

Ja, het nadeel is in dit geval de 3 d omgeving aangezien de objecten 3 d moeten worden getekend.

Kunnen er iconen uit andere software ingevoerd worden zoals bijvoorbeeld CAD of bitmap) ?

Ja maar de objecten moeten wel 3 dimensionaal ingevoerd worden.

Testen en efficiëntie

Multitasking

Is het mogelijk om tijdens het bewerken van een model op het zelfde moment een model te "runnen" ?

Nee de omgevingen zijn van elkaar gescheiden

Stapfunctie

Kan het model stap voor stap worden gerund ?

Ja

Terugdraaien van de tijd

Kan het model ook terug in de tijd rekenen ?

?

Uitvoer

Rapporten

Welke rapporten komen standaard voor in het programma ?

Er kan een overzicht worden opgevraagd van de verschillende objecten

Kunnen er zelf rapporten worden opgesteld ?

?

Uitvoer mogelijkheden

Kan het programma zijn uitvoer rechtstreeks in andere programma's zoals excel onderbrengen ?

?

Grafieken

Kunnen er met het programma ook histogrammen grafieken en staafdiagrammen worden gemaakt ?

Ja met behulp van Autostat kan dit worden gedaan

Gebruiker

Financieel

Kan de software tegen een redelijke prijs voor een bepaalde periode gehuurd worden en wat zijn de kosten van het gehele pakket ?

De kosten van Automod zijn ongeveer 17000 euro

eM-plant

Software

Softwarebedrijf

Wat is de algemene indruk van software en bedrijf ?

Het bedrijf Tecnomatix bestaat ongeveer 10 jaar en heeft een aantal verschillende tools die beginnen met em-... De software wordt vooral verkocht aan de auto-industrie in Asia en Europa. Tecnomatix is een vrij groot bedrijf waardoor de kans op continuïteit groot is.

Documentatie

Bestaat er goede en duidelijke documentatie van de software ?

Er zijn helpfiles in het programma aanwezig het gebruik ervan is alleen niet optimaal

Worden er voorbeeldmodellen bij de software geleverd ?

Ja maar ze worden slecht uitgelegd.

Ondersteuning

Wordt er telefonische of andere ondersteuning door het bedrijf geleverd ?

Ja

Zijn er op de Technische universiteit Delft mensen aanwezig die ondersteuning kunnen/willen geven ?

Er zijn mensen aanwezig die bekend zijn met de software

Model en input

Grafisch bouwen van modellen

Is het model vooral op code of vooral grafisch georiënteerd ?

Het kan beide er zijn een aantal vaste objecten die in het model gesleept worden en de overige objecten moeten zelf worden geschreven.

Invoer

Op wat voor soort manieren kan de invoer worden ingelezen ? (interactief uit file of uit een ander systeem)

Data kan ingevoerd worden via: gebruiker, uit bestanden, uit databases of zelfs met behulp van webpagina's

Statistische verdelingen

welke verdelingen zijn mogelijk ?

beta, binominal, continu, discreet erlang, exponentieel, gamma, geometrisch, hypergeometrisch, lognormaal, normaal, Poisson, driehoekig, uniform en Weibull. De overige verdelingen kunnen met behulp van een soort datafit ook worden ingevoerd

Mogelijkheid tot invoer code

Is het mogelijk zelf code in te voeren of zelfs de sourcecode te veranderen ?

Ja, er kan zelf code worden ingevoerd

Wachtrijen

Kan het programma verschillende wachtrij-systemen zoals FIFO en LIFO aan ?

Zowel FIFO , LIFO zijn mogelijk

Draaien van het programma

Aanpassen snelheid

Kan de programma snelheid worden aangepast ?

Ja, ook werkelijke tijd

Executable

Kan het programma van een model een zelfdraaiend model maken (executable) ?

Kan alleen als er een werkende versie van eM-plant op de computer aanwezig is

Animatie

Geïntegreerd animatiepakket

Is het animatiepakket geïntegreerd in de software of wordt het er apart bijgeleverd ?

Het 2d model is geïntegreerd, maar werkt moeizaam. Het 3d model is niet geïntegreerd.

Iconen

Worden er iconen bij het pakket geleverd ?

Ja

Is het mogelijk om nieuwe iconen aan te maken ?

Ja

Kunnen er iconen uit andere software ingevoerd worden zoals bijvoorbeeld CAD of bitmap) ?

Ja

Testen en efficiëntie

Multitasking

Is het mogelijk om tijdens het bewerken van een model op het zelfde moment een model te "runnen" ?

Ja, kan alleen niet direct geïmplementeerd worden

Stapfunctie

Kan het model stap voor stap worden gerund ?

ja

Terugdraaien van de tijd

Kan het model ook terug in de tijd rekenen

Er is een venster waar dit aangeklikt kan worden het werkt alleen niet goed

Uitvoer

Rapporten

Welke rapporten komen standaard voor in het programma ?

De gegevens van de standaard objecten kunnen worden verzameld. Deze gegevens blijven bij het object en kunnen moeilijk in een rapport ondergebracht worden

Kunnen er zelf rapporten worden opgesteld ?

?

Uitvoer mogelijkheden

Kan het programma zijn uitvoer rechtstreeks in andere programma's zoals Excel onderbrengen ?

kan maar moet zelf worden geprogrammeerd

Grafieken

Kunnen er met het programma ook histogrammen grafieken en staafdiagrammen worden gemaakt ?

Ja met een apart object

Gebruiker

Financieel

Kan de software tegen een redelijke prijs voor een bepaalde periode gehuurd worden en wat zijn de kosten van het gehele pakket ?

De kosten voor de aanschaf van het hele pakket bedragen ongeveer 50.000 euro

Tomas/delphi

Software

Softwarebedrijf

Wat is de algemene indruk van de software en het bedrijf ?

De software is erg nieuw en zelfs niet geheel uitontwikkeld, het is dus moeilijk aan te geven of de continuïteit is gegarandeerd

Documentatie

Bestaat er goede en duidelijke documentatie van de software ?

?

Worden er voorbeeldmodellen bij de software geleverd ?

Ja

Ondersteuning

Wordt er telefonische of andere ondersteuning door het bedrijf geleverd ?

?

Zijn er op de Technische universiteit Delft mensen aanwezig die ondersteuning kunnen/willen geven ?

Ja

Model en input

Grafisch bouwen van modellen

Is het model vooral op code of vooral grafisch georiënteerd ?

Vooraf code, standaard verdelingen end wel aanwezig)

Invoer

Op wat voor soort manieren kan de invoer worden ingelezen ? (interactief uit file of uit een ander systeem)

Zelf invoeren makkelijk maar uit file moeilijk,

Statistische verdelingen

welke verdelingen zijn mogelijk ?

zit in Tomas neg exp uniform normaal erlang (zelf programmeren.)

Mogelijkheid tot invoer code

Is het mogelijk zelf code in te voeren of zelfs de sourcecode te veranderen ?

ja alleen code

Wachtrijsystemen

Kan het programma verschillende wachtrij-systemen zoals FIFO en LIFO aan ?

Moet je zelf programmeren

Draaien van het programma

Aanpassen snelheid

Kan de programma snelheid worden aangepast ?

Nee

Executable

Kan het programma van een model een zelfdraaiend model maken (executable) ?

Ja, automatisch

Animatie

Geïntegreerd animatiepakket

Is het animatiepakket geïntegreerd in de software of wordt het er apart bijgeleverd ?

Nee, alle animatie zelf programmeren

Iconen

Worden er iconen bij het pakket geleverd ?

Nee

Is het mogelijk om nieuwe iconen aan te maken ?

Kan waarschijnlijk wel geprogrammeerd worden.

Kunnen er iconen uit andere software ingevoerd worden zoals bijvoorbeeld CAD of bitmap) ?

Testen en efficiëntie

Multitasking

Is het mogelijk om tijdens het bewerken van een model op het zelfde moment een model te "runnen" ?

Nee, cpu volledig benut

Stapfunctie

Kan het model stap voor stap worden gerund ?

Ja stap voor stap in de listing

Terugdraaien van de tijd

Kan het model ook terug in de tijd rekenen

Nee

Uitvoer

Rapporten

Welke rapporten komen standaard voor in het programma ?

Wachttijden en standaarddeviatie zijn standaard

Kunnen er zelf rapporten worden opgesteld ?

Ja maar niet aan te raden

Uitvoer mogelijkheden

Kan het programma zijn uitvoer rechtstreeks in andere programma's zoals Excel onderbrengen?

Kan maar niet heel makkelijk

Grafieken

Kunnen er met het programma ook histogrammen grafieken en staafdiagrammen worden gemaakt ?

Ja die worden door Tomas gemaakt

Gebruiker

Financieel

Kan de software tegen een redelijke prijs voor een bepaalde periode gehuurd worden en wat zijn de kosten van het gehele pakket ?

Tomas is gratis en de studenten versie van Delphi kost ongeveer 600 gulden

Prosim

Software

Softwarebedrijf

Wat is de algemene indruk van de software en het bedrijf ?

De softwareproducent Prosim B.V. bestaat al een aantal jaar, het is alleen een klein bedrijf de vraag blijft dan ook of de continuïteit gegarandeerd is.

Documentatie

Bestaat er goede en duidelijke documentatie van de software ?

Er bestaat een handleiding,

Worden er voorbeeldmodellen bij de software geleverd ?

Ja

Ondersteuning

Wordt er telefonische of andere ondersteuning door het bedrijf geleverd ?

Het is mogelijk contact op te nemen met Prosim B.V.

Zijn er op de Technische universiteit Delft mensen aanwezig die ondersteuning kunnen/willen geven ?

Ja

Model en input

Het bouwen van het model

Is het model vooral op code of vooral grafisch georiënteerd ?

Geheel op code gebaseerd

Invoer

Op wat voor soort manieren kan de invoer worden ingelezen ? (interactief uit file of uit een ander systeem)

Kan uit tekstfiles worden gelezen

Statistische verdelingen

welke verdelingen zijn mogelijk ?

Uniform, normaal, exponentieel, gamma en beta en de overige kunnen zelf geprogrammeerd worden

Mogelijkheid tot invoer code

Is het mogelijk zelf code in te voeren of zelfs de sourcecode te veranderen ?

Ja, alleen code

Wachtrijen

Kan het programma verschillende wachtrij-systemen zoals FIFO en LIFO aan ?

Moet je zelf programmeren

Draaien van het programma

Aanpassen snelheid

Kan de programma snelheid worden aangepast ?

De snelheid kan gesynchroniseerd worden (x maal de tijd)

Executable

Kan het programma van een model een zelfdraaiend model maken (executable) ?

Nee altijd Prosim nodig

Animatie

Geïntegreerd animatiepakket

Is het animatiepakket geïntegreerd in de software of wordt het er apart bijgeleverd ?

Nee, alle animatie zelf programmeren

Iconen

Worden er iconen bij het pakket geleverd ?

Nee

Is het mogelijk om nieuwe iconen aan te maken ?

Animatie kan geprogrammeerd worden

Kunnen er iconen uit andere software ingevoerd worden zoals bijvoorbeeld CAD of bitmap) ?
?

Testen en efficiëntie

Multitasking

Is het mogelijk om tijdens het bewerken van een model op het zelfde moment een model te "runnen" ?

Ja, er kan gewoon door worden gewerkt

Stapfunctie

Kan het model stap voor stap worden gerund ?

Ja stap voor stap in de listing, maar niet gebruiksvriendelijk

Terugdraaien van de tijd

Kan het model ook terug in de tijd rekenen

Nee

Uitvoer

Rapporten

Welke rapporten komen standaard voor in het programma ?

Wachttijden en standaarddeviatie zijn standaard

Kunnen er zelf rapporten worden opgesteld ?

Ja waarden moeten weggeschreven worden naar Tekstfiles en in een spreadsheet geopend worden

Uitvoer mogelijkheden

Kan het programma zijn uitvoer rechtstreeks in andere programma's zoals excel onderbrengen

Nee, eerst naar een tekstfile en daarna in excel importeren

Grafieken

Kunnen er met het programma ook histogrammen grafieken en staafdiagrammen worden gemaakt ?

De standaard rapporten kunnen makkelijk in grafiekvorm worden weergegeven, voor de overige uitvoer is dit moeilijker.

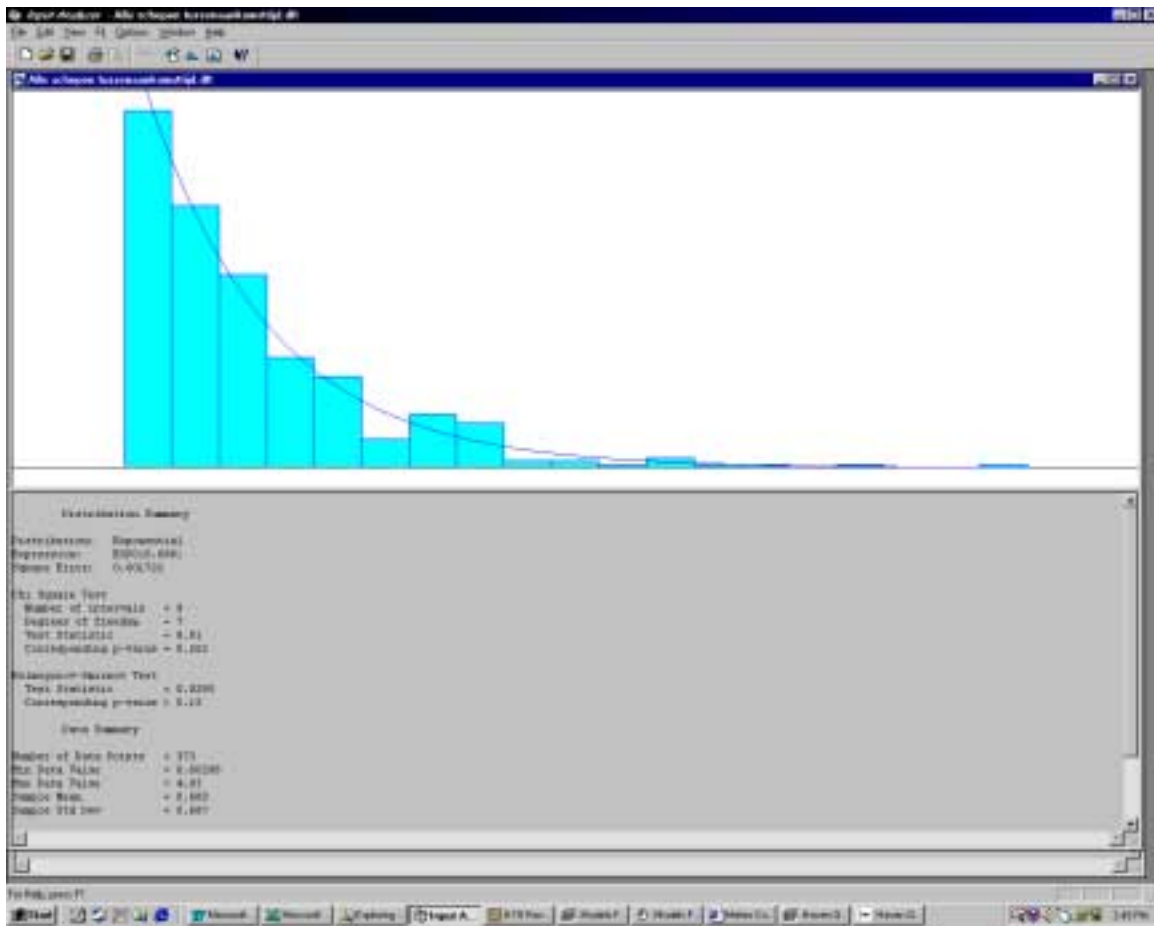
Gebruiker

Financieel

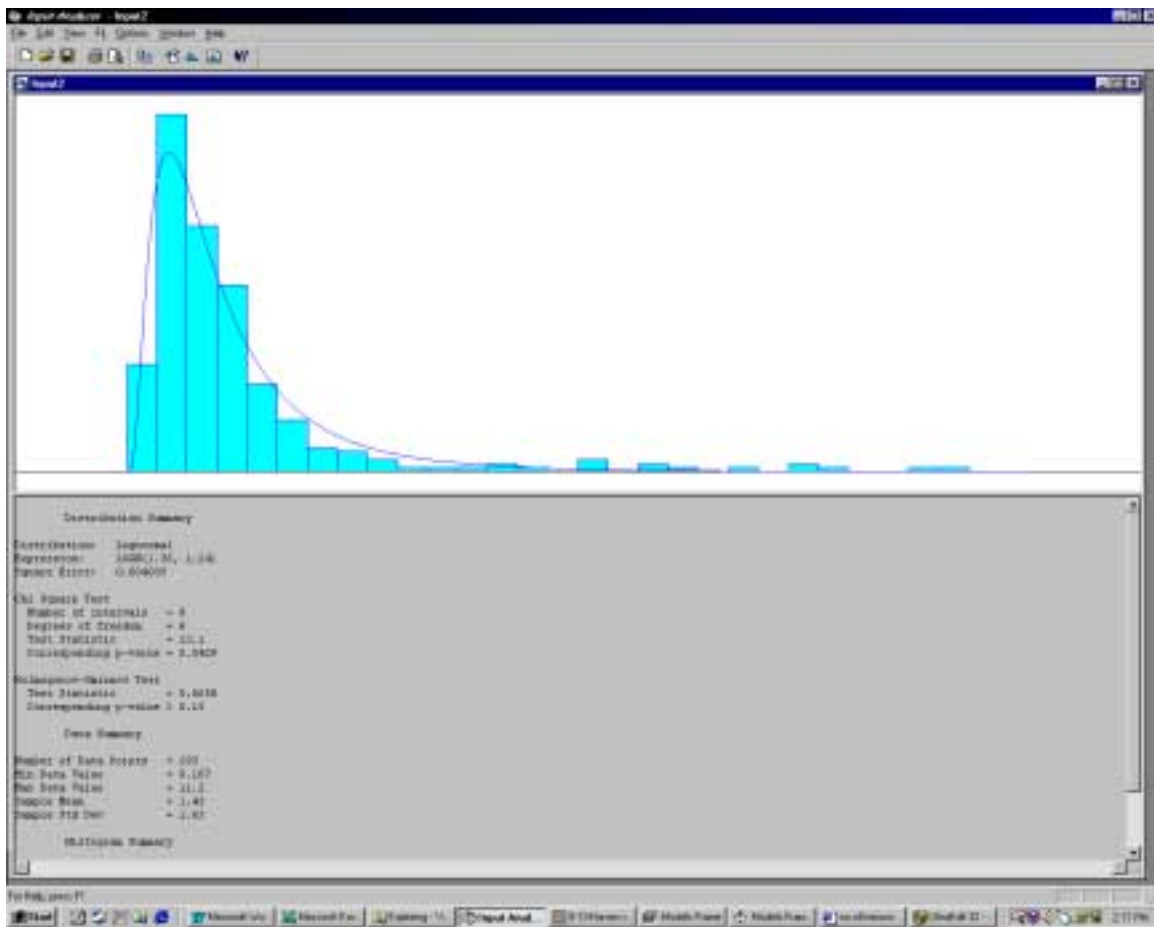
Kan de software tegen een redelijke prijs voor een bepaalde periode gehuurd worden en wat zijn de kosten van het gehele pakket ?

11000 euro

Bijlage 2 Datafit



Datafit tussenaankomsttijden alle schepen



Datafit verblijftijden overige schepen

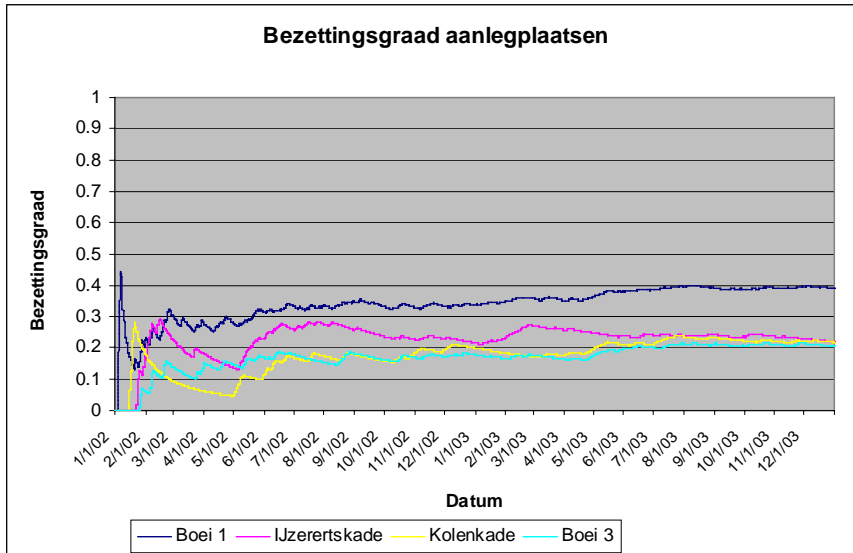
Bijlage 3 Verificatie en validatie

In deze bijlage worden de uitkomsten van de simulatie geverifieerd en gevalideerd.

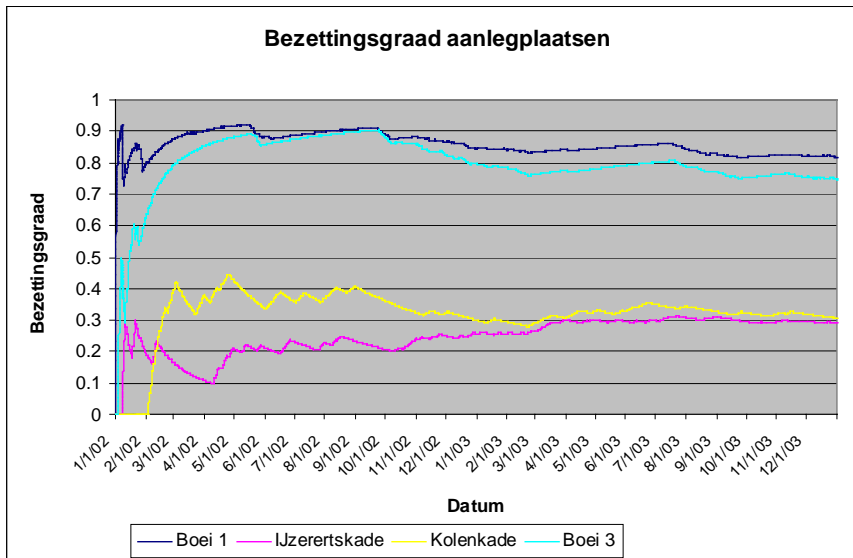
Verificatie 1

Bij de eerste verificatie worden er gekeken wat de effecten zijn van een extreme hoeveelheid klinkerschepen op het systeem. De verwachting is :

- Bezettingsgraden van boei 1 en boei 3 gaan naar 1
- Bezettingsgraden van de ijzererts-kade en kolenkade zullen iets stijgen



Zelfde aantal schepen maar een extreem hoog percentage klinkerschepen



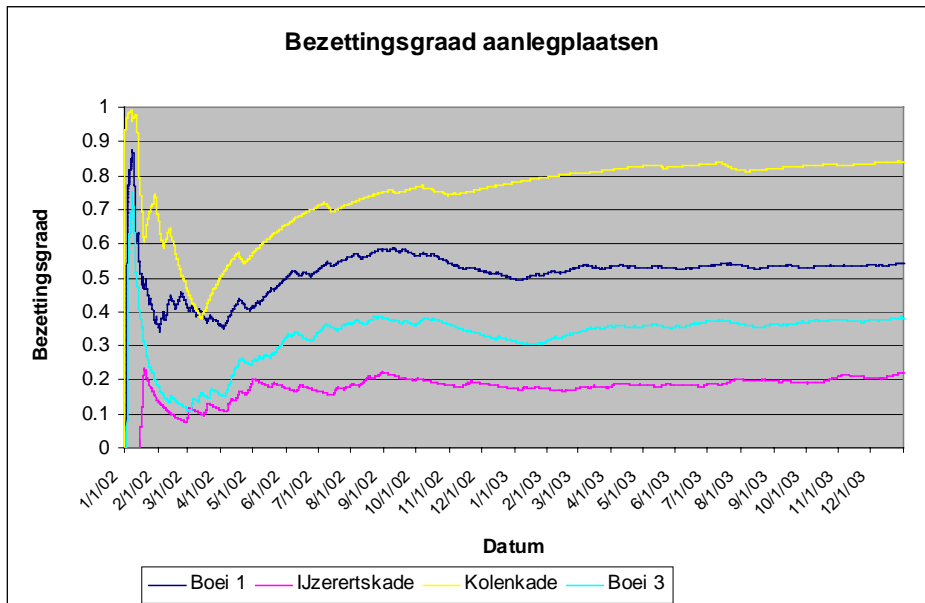
Zoals verwacht gaan de bezettingsgraden van boei 1 en boei 3 extreem omhoog, ook de bezettingsgraden van de kolenkade en de ijzererts-kade gaan omhoog. De schepen die hier afmeren moeten namelijk veelal verplaatst worden naar of boei 1 of boei 2 en vanwege de grote drukte bij deze aanlegplaatsen moet er gewacht worden.

Verificatie 2

In de tweede verificatie wordt het aantal kolenschepen extreem verhoogd. De verwachtingen zijn:

- Bezettingsgraad van de kolenschepen stijgt
- Bezettingsgraad boei 1 en boei 3 stijgt iets
- Bezettingsgraad ijzererts blijft gelijk

Zelfde aantal schepen maar een hoger percentage kolenschepen



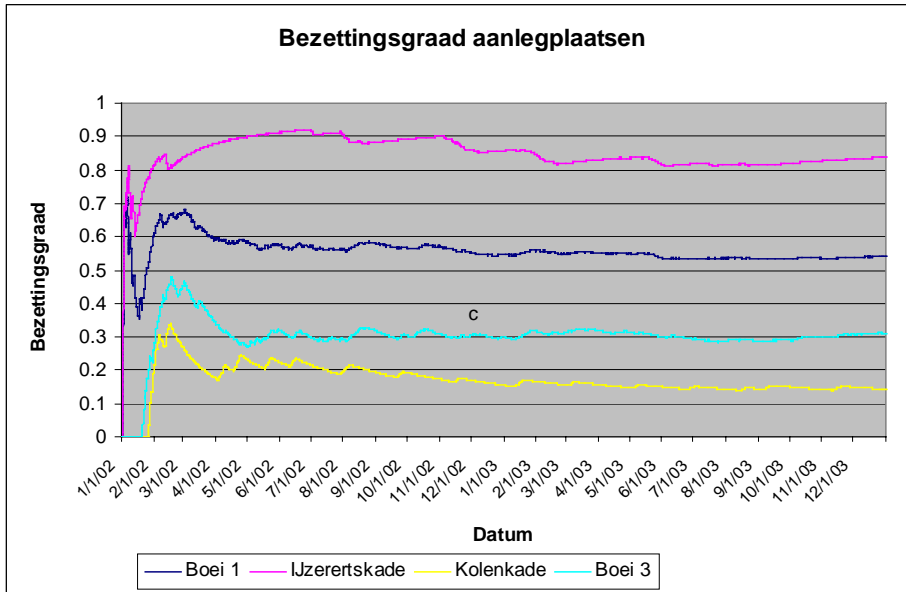
Wanneer het percentage kolenschepen aanzienlijk wordt verhoogd, dan zal ook de bezettingsgraad van de kolenkaide stijgen ook bij de beide boeien zal er een grotere drukte ontstaan vanwege het doorschuiven van de kolenschepen. De bezettingsgraad bij de ijzerertskaide blijft ongeveer gelijk. De capaciteit van boei 1 en boei 3 is voldoende om het groter aantal kolenschepen op te vangen, zonder merkbare gevolgen voor de ijzerertsschepen.

Verificatie 3

Bij deze verificatie zijn het aantal ijzerertsschepen relatief hoog gekozen. Verwacht wordt dat:

- De bezettingsgraad van de ijzerertsschepen sterk stijgt
- De bezettingsgraden van zowel boei 1 als boei 3 stijgen
- De bezettingsgraad van de kolenkade nagenoeg gelijk blijft

Zelfde aantal schepen maar een hoger percentage ijzerertsschepen



Ook deze verificatie geeft de verwachte waarden, door de stijging van de ijzererts schepen zal het drukker worden bij de ijzerertskade. Hierdoor zal de bezettingsgraad van de ijzerertskade stijgen. Doordat de meeste schepen verder worden geladen bij boei 1 of boei 3 zijn de bezettingsgraden van beide aanlegplaatsen ook gestegen. Alleen de bezettingsgraad van de kolenkade is nagenoeg gelijk gebleven, blijkbaar heeft de grotere drukte bij boei 1 en boei 3 geen gevolgen voor de bezettingsgraad van de kolenkade.

Validatie 1 Tonnage

Door de gesimuleerde hoeveelheid bulk te vergelijken met de werkelijke gegevens kan een extra validatie verkregen worden. In het totaal zijn 5 replicaties van een run met de gegevens uit het jaar 2000 gedaan. Uit deze 5 replicaties is een gemiddelde berekend en deze is vergeleken met de werkelijkheid. De simulatie had in dit geval een runtijd van 3.5 jaar, om deze om reden zijn de andere getallen ook met 3.5 vermenigvuldigd. De gegevens staan vermeld in onderstaande tabel

	sim 1	sim 2	sim 3	sim 4	sim 5
Totaal klinker gelost (ton)	2,901,033	2,455,497	2,564,070	2,238,275	2,115,435
DWCT klinker (ton)	5,802,066	4,910,994	5,128,139	4,476,551	4,230,871
Totaal kolen geladen (ton)	1,410,338	1,508,686	1,176,875	1,608,110	1,034,409
DWCT kolen (ton)	1,689,075	1,803,510	1,413,900	1,923,120	1,223,955
Totaal ijzererts geladen (ton)	2,756,836	3,268,421	3,108,644	2,853,141	2,270,882
DWCT ijzererts (ton)	2,914,065	3,442,860	3,269,160	3,029,085	2,381,670

	Gem. Waarde	Analyse schepen	Rapport
Totaal klinker gelost (ton)	2,454,862	2,497,930	2,450,000
DWCT klinker (ton)	4,909,724		
Totaal kolen geladen (ton)	1,347,684		1,750,000
DWCT kolen (ton)	1,610,712	1,632,062	
Totaal ijzererts geladen (ton)	2,851,585		3,150,000
DWCT ijzererts (ton)	3,007,368	2,885,590	

Uit de analyse blijkt dat de waarden relatief goed kloppen met de analyse van de schepen die de haven hebben aangedaan. De cijfers die door de haven zelf gehanteerd worden liggen voor kolen en ijzererts beduidend hoger dan kan worden berekend met behulp van de scheepslijsten. Waarschijnlijk komt dit doordat de lijsten met schepen niet altijd de juiste materiaal vermelding heeft. Om de simulatie nauwkeuriger te maken zal er verder onderzoek moeten worden verricht naar: welke schepen de haven aandoen en welk materiaal er wordt vervoerd.

Validatie 2 Scheepsverdelingen

In deze validatie worden de werkelijke scheepsverdelingen vergeleken met de scheepsverdelingen die uit de simulatie volgen. In de onderstaande tabel is een weergave van de vergelijking.

	Klinker	Kolen	IJzererts	Overig	Totaal
Sim 1	39	15	34	442	529
Sim 2	42	15	44	448	548
Sim 3	46	16	35	447	543
Sim 4	48	13	38	447	546
Sim 5	46	17	33	438	533
Gemiddelde	44	15	36	444	540
Werkelijke aantallen	45	14	34	437	530

De waarden uit de simulatie zitten dicht bij de waarden uit de werkelijkheid, de validatie van de scheepsverdelingen valt daarom positief uit.