

Hogeschool Rotterdam
COLLEGE OF ENGINEERING, Applied ComputerScience

Dissertation

EINDVERSLAG TINLAB

by

GALVIN BARTES

B.S., Hogeschool Rotterdam University, 2022

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

2022

© 2022 by
GALVIN BARTES
All rights reserved

**Prior to having this page signed by the readers, please have
it reviewed by the Mugar Library staff.**

Approved by

First Reader

Oele W. Last, PhD
Professor of Electrical and Computer Engineering

Second Reader

Abdrasitova E. Last
Associate Professor of ...

*Facilis descensus Averni;
Noctes atque dies patet atri janua Ditis;
Sed revocare gradum, superasque evadere ad auras,
Hoc opus, hic labor est.* Virgil (from Don's thesis!)

Acknowledgments

Het is geweldig om dit vak te herkansen met twee fantastische collega's

Galvin Bartes

Student

Tycho

Student

Koosha

Student

CMI

EINDVERSLAG TINLAB

GALVIN BARTES

Hogeschool Rotterdam, College of Engineering, Applied

ComputerScience, 2022

Major Professors: Oele W. Last, PhD

Professor of Electrical and Computer Engineering

Secondary appointment

Abdrasitova E. Last, PhD

Professor of Computer Science

ABSTRACT

Het ministerie van verkeer en Waterstaat wil in het kader van het klimaatakkoord en onderzoek laten uitvoeren naar de staat van het sluizenpark in Nederland. Het onderzoek moet zich richten op het ontwerpen en ontwikkelen van een geautomatiseerd sluismodel dat geschikt is voor een brede toepassing. In het onderzoek moet naar voren komen wat de huidige staat is van de sluizen met oog op veiligheid, efficiëntie, capaciteit, onderhoud, duurzaamheid en automatisering. Het onderzoek geeft aan hoe een volledig model worden opgeleverd opdat ontwerp van verschillend volledig geautomatiseerde sluizen in de toekomst geautomatiseerd kunnen worden.

Contents

1	Inleiding	1
1.1	Design goals	3
1.2	Leeswijzer	3
2	Methodologie	4
3	Theoretisch kader	5
3.0.1	MODE CONFUSION	5
3.0.2	Wat is automatiseringsparadox	5
3.0.3	Wat is een model	5
3.0.4	World and machine samenvatting	7
3.0.5	SIX Variable model	10
4	Requirements en specificaties	16
4.0.1	Requirements	16
4.0.2	Veiligheidsoverwegingen	18
4.0.3	Afbakning	19
5	Ontwerp	21
5.1	Inleiding	21
5.2	Concept	23
6	Kripke model	26
6.1	Uppaal kripke structuren	26

7	CTL logica	29
7.1	Doel van de test	29
7.1.1	Wat wordt getest en hoe	29
7.1.2	toetsen met queries	29
7.1.3	Operator: AG	29
7.1.4	Operator: EG	29
7.1.5	Operator: AF	29
7.1.6	Operator: EF	29
7.1.7	Operator: AX	29
7.1.8	Operator: EX	29
7.1.9	Operator: $p \text{ U } q$	30
7.1.10	Operator: $p \text{ R } q$	30
8	Testresultaten CTL logica	31
8.0.1	Fairness	32
8.0.2	Liveness	32
8.0.3	safety	33
8.0.4	zeno vrij	33
8.0.5	deadlocks	33
9	Conclusie	34
10	Discussie	35
10.0.1	Future work	35
11	Notulen	36
12	Queries	38
13	Testresultaten	41
14	Reflectie	43

15 Conclusions	45
15.1 Summary of the thesis	45
A Uppaal source	46
A.1	
Appendix B: Model eerste deelname aan cursus 2020	48
A.2	
Appendix B: Model herkansing tweede deelname aan cursus 2020 . .	50
References	56
Mijn profiel	57

List of Tables

List of Figures

List of Abbreviations

The list below must be in alphabetical order as per BU library instructions or it will be returned to you for re-ordering.

CAD	Computer-Aided Design
CO	Cytochrome Oxidase
DOG	Difference Of Gaussian (distributions)
FWHM	Full-Width at Half Maximum
LGN	Lateral Geniculate Nucleus
ODC	Ocular Dominance Column
PDF	Probability Distribution Function
\mathbb{R}^2	the Real plane

Chapter 1

Inleiding

Algemeen

Het ministerie van verkeer en Waterstaat wil in het kader van het klimaatakkoord en onderzoek laten uitvoeren naar de staat van het sluizenpark in Nederland. Het onderzoek moet zich richten op het ontwerpen en ontwikkelen van een geautomatiseerd sluismodel dat geschikt is voor een brede toepassing. In het onderzoek moet naar voren komen wat de huidige staat is van de sluizen met oog op veiligheid, efficiëntie, capaciteit, onderhoud, duurzaamheid en automatisering. Het onderzoek geeft aan hoe een volledig model worden opgeleverd opdat ontwerp van verschillend volledig geautomatiseerde sluizen in de toekomst geautomatiseerd kunnen worden.

Probleemanalyse

Na grondige analyse van het Nederlandse sluizenpark is gebleken dat renovatie van een groot aantal sluizen noodzakelijk is. Uit een eerste verkenning is gebleken dat het gecombineerd renoveren en automatiseren van het Nederlandsesluizenpark een aanzienlijke verbetering kan opleveren t.a.v. Op het ministerie van infrastructuur en waterstaat is helaas onvoldoende kennis van ict en systemen aanwezig om eenen ander uit te voeren

Waarom nu

In het kader van het onlangs afgesloten klimaatakkoord heeft de Nederlandse overheid daarom besloten over te gaan tot een ingrijpende renovatie van de diverse sluizen die ons land rijk is.

Gewenst resultaat

Wij vragen u een model (of een onderling samenhangend aantal modellen) aan te leveren, opdat ontwerpen van verschillende, volledig geautomatiseerde sluizen in de toekomst gerealiseerd kunnen worden. Zoals gesteld in de brief is het de bedoeling dat een sluis gemodelleerd wordt dat bewezen kan worden dat de te bouwen sluis een aantal eigenschappen bezit.

Scope

He gaat om het simuleren van een geautomatiseerde sluis. Wat voor type sluis wordt niet gemeld en ook niet uit welke onderdelen. Belangrijk is dat het model werkt en dat het voldoet aan de eisen die gebaseerd zijn op basis van literatuuronderzoek, observatie, interviews, brainstorming of een andere vorm van requirements elicitation.

Onderzoeksvragen

1. Uit het onderzoek zal moeten blijken welke veiligheidseisen er zijn voor sluizen in Nederland.
2. Daarnaast welke factoren een rol spelen in de duurzaamheid van het sluisenpark.
3. Hoe wordt de routinecontrole op de sluizen uitgevoerd?
4. Welke automatisering is mogelijk met oog op veiligheid, efficiëntie en capaciteit?

5. Welke criteria wegen zwaar in de ontwikkel- en onderhoudskosten van duurzame technologie?

1.1 Design goals

Het systeem moet minimaal aan de volgende prestatie eisen voldoen

1. (a) Ship management,
2. Data
 - (a) Water level management
3. Veiligheid
 - (a) Uitbreidbaarheid

1.2 Leeswijzer

In de methodologie wordt de lezer uitgelegd met welke methoden de onderzoeksvragen zijn beantwoord. In het hoofdstuk Onderzoek worden alle resultaten behandeld die naar voren zijn gekomen bij het deskresearch. De analyse van de verzamelde data wordt gedaan in het hoofdstuk analyse. Hierin wordt behandeld zoekopdracht naar IoT cloud platforms, feature extractie, prijs-berekening en prijs-feature vergelijking. In het ontwerp komen de uml diagrammen en systeemschetsen naar voren. In de de hoofdstukken Prototype, IoT cloud en Firmware wordt de implementatie behandeld van het IoT cloud platform in een bestaand project.

Chapter 2

Methodologie

Chapter 3

Theoretisch kader

3.0.1 MODE CONFUSION

Mode confusion treedt op als geobserveerd gedrag van een technisch systeem niet past in het gedragspatroon dat de gebruiker in zijn beeldvorming heeft en ook niet met voorstellingsvermogen kan bevatten.

3.0.2 Wat is automatiseringsparadox

Gemak dient de mens. Als er veel energie wordt gestoken in de ontwikkeling van hulpmiddelen die taken van werknemers overnemen heeft dat tot resultaat dat veel productieprocessen worden geautomatiseerd. De vraag is dan of vanuit mechanisch wereldpunt de robot niet de rol van de mens overneemt en of de mens nog de kwaliteiten heeft om het werk zelf te doen.

<https://www.dalton.nl/literatuur/item/252-de-automatiseringsparadox> <https://www.debicker.eu/de-automatiseringsparadox/> <https://vse.nl/de-paradox-van-de-industrie/> <https://automatie-pma.com/nieuws/industriële-automatiseringsparadox> <https://blog.xot.nl/2016/11/21/slimme-apparaten-maken-ons-dom-en-kwetsbaar/index.html>

3.0.3 Wat is een model

Conceptueel model

Om duiding te geven aan het grote plaatsje

in vivo model

Levende organismen die in de werkelijkheid of in een laboratorium vergelijkbare eigenschappen bezitten als bestaande fenomenen in de werkelijkheid. Deze objecten zijn vergelijkbaar met werkelijkobjecten en geven vergelijkbare resultaten

in vitro model

Een model dat dezelfde condities biedt buiten het onderzoeksobject om, maar is voldoende vergelijkbaar om vergelijkbare processen te simuleren. Zowel *in vivo* als *in vitro* modellen zijn beperkt door de materialen die beschikbaar zijn voor onderzoek en de arbeidsomstandigheden waaronder ze worden gebruikt. Desondanks zijn het geen werkelijke natuurlijke modellen dus voor een onderzoek kan het geen volledige uitsluitel.

In silico model

Een veelzijdig object. Het verwijst naar simulaties die gebruik maken van wiskundige modellen in computer, een zijn dus afhankelijk van silicon chips. In *in silico* model analyseert wiskundige vergelijkingen om resultaten te geven onder bepaalde omstandigheden. Deze vergelijkingen vertellen iets over de correlatie van verschillende objecten van een wetenschappelijk onderzoek. Om deze modellen te kunnen gebruiken is het noodzakelijk te omschrijven wat de fenomenen in kwestie van onderzoek zijn door middel van getallen. Kwantitatieve relaties kunnen worden geïntegreerd in het model en waar deze relaties complex zijn is een computer noodzakelijk deze op te lossen. Vaak worden hierbij verschillende mechanismen gebruikt. Als je bijvoorbeeld de ontwikkeling van een marsreep in kaart wilt brengen.

in simulacra model

3.0.4 World and machine samenvatting

Waarom zijn wij engineers? Omdat we bruikbare apparaten willen laten functioneren in de wereld waarin we leven. Dat doen we door de machine te beschrijven en deze beschrijving van instructies bieden we aan onze computer opdat deze als de attribuut en gedragingen uitleest zoals wij die hebben omschreven. Dit alles op basis van theoretische funderingen en praktisch inzicht.

Het doel van een machine is om te worden geïnstalleerd en te worden gebruikt. De eisen die we stellen zitten in de omgeving en in de wereld en de machine is slechts de oplossing die we bedenken om aan een eis te voldoen.

De relatie machine-wereld world gecategoriseerd in:

Het modelleer aspect: waar een machine de wereld simuleert

Het interface aspect: waar er fysieke interactie is tussen de machine en de wereld

Het engineering aspect: waar de machine zich gedraagt als een controlemotor gebruikmakend van de gedragingen van de omgeving in de wereld

Het probleem aspect: waar de omgeving in de wereld en de omvang van het probleem invloed heeft op de machine en de oplossing

Het modelleer of simulatie aspect over een deel van de wereld. Er zijn data, object en proces modellen. Het doel van een model is toegang te geven tot informatie over die wereld. Door het opvangen van statische weergaven en gebeurtenissen kunnen wij deze gebruiken van opgeslagen informatie die we kunnen hergebruiken. Een model kan bruikbare informatie bevatten omdat zowel het model als de wereld waarin het model zich bevindt gemeenschappelijke omschrijvingen hebben die waar zijn voor zowel het model als voor de wereld. Daarbij moet gesteld worden dat de interpretatie van een model verschilt met een interpretatie van de wereld.

Omdat zowel de wereld als de machine fysieke realiteiten zijn en niet slechts ab-

stracties, zijn de gemeenschappelijke beschrijvingen slechts een deel van de werkelijkheid van beide objecten. For elk object zijn er meerdere beschrijvingen. Toch maken niet alle omschrijvingen deel uit van het getoonde repertoire. Zoals niet alle eigenschappen van een boek; meer dan een auteur, pseudoniemen, een onderdeel van een reeks, een gerevisiteerde versie, worden gereflecteerd in een database.

Het interface aspect. Een machine kan een probleem in de wereld oplossen als de wereld en de machine phenomena kunnen uitwisselen. Maar de participatie is niet symmetrisch: een status kan als phenomena worden uitgewisseld maar slechts een partij kan er invloed op uitoefenen maar beiden kunnen dezelfde status signaleren.

Het engineering aspect gaat over requirements, specificaties, en programma's. Requirements hebben betrekking op phenomena in de wereld. Een programma heeft alleen betrekking tot de machinale phenomena. Het doel van programma's is om eigenschappen en gedragingen te omschrijven van de machine ten behoeve van de gebruiker. Tussen de requirements en de programma's zitten de specificaties. Omdat programma's dan wel beschrijvingen zijn van een gewenste machine, maar dat moeten beschrijvingen zijn van de machines die de computers kunnen uitvoeren zodanig dat de computer deze beschrijvingen ook zo kan interpreteren. De engineer moet de eigenschappen van de wereld kennen en begrijpen en deze eigenschappen manipuleren en laten werken met als doel het dienen van het systeem.

Het probleem aspect. Het onderscheid tussen specificatie en implementatie. Het probleem zit in de relatie van de machine en de wereld. De machine brengt de oplossing maar het probleem zit in de wereld. Een vertoog over een probleem moet dus gaan over de wereld en over de opvatting die de gebruiker heeft in de wereld. Omdat de wereld veelzijdig is moeten we ervan uit gaan dat er verschillende soorten problemen zijn. Een realistisch probleem wordt dus niet opgelost met een simpele hiërarchische structurele aanpak en een homogene decompositie maar met een parallelle structurele

oplossing waar beide kanten van het probleem worden opgelost.

Ontkenningen

We hebben als engineers de taak om een machine te bouwen aan de hand van de specificaties opgeleverd door de opdrachtgever. Een engineer heeft niet als taak de fitheid voor een doeleind te onderzoeken, maar wel de haalbaarheid naar een doeleind aan de hand van kennis, tijd, resources, budget en ontwikkelmethodiek. Daaruit komt naar voren dat een engineer zich richt op: elicitation (schetsen van een requirement), description (omschrijving) en analyse van de requirements waaraan het systeem moet voldoen. Vertaalt naar de volgende vragen: Wat is precies de klantwens? Wat is de precieze omschrijving van het probleem? Voor welke doelen wordt het systeem gebouwd? Welke functies moet het systeem hebben?

Denial by hacking: obsessief bezig zijn met een systeem omdat het de gebruiker veel macht geeft. Een uitgebreidheid van een systeem zorgt er soms voor dat mensen niet meer geprikkeld zijn na te denken over probleemstellingen, domein beschrijvingen en analyse.

Denial by a abstraction. Wiskundige benaderingen van werkelijke problemen is een belangrijke intellectuele strategie om problemen te formuleren. Een software ontwikkelaar moet een probleem kunnen omschrijven in zo min mogelijk woorden, maar de complexiteit ligt in de oplossing.

Denial by vagueness. De vaagheid van een omschrijving is terug te vinden in:

Von Neumann's principe

Principe van reductionisme

Shanley principe

Montaignes's principe

Von Neumann principe

Voor een vocabulair moet een grondslag zijn ontwikkeld waarmee gesproken kan

worden over de wereld en de machine. Belangrijke fenomenen moeten geïdentificeerd worden, door middel van een grondregel of ‘herkenningsregel’ moet een fenomeen worden herkend, en vervolgens het fenomeen een formele term geven die gebruikt wordt als duiding van een bepaalde omschrijving. Dan moet voor de formele term een symbool gevonden worden. Samen vormen de grondregel en het symbool een designatie.

Principe van reductionisme

Simpelweg het openbreken van termen met een weerlegbare definitie totdat alle begrippen die worden gebruikt om iets te duiden niet meer te herconstrueren zijn in hun definitie.

Shanley principe

Er bestaan volgens dit principe geen scherpe verdelingen in de wereld zoals wetenschappers soms denken. Een strenge opvatting over de wereld waarin een individu geclassificeerd kan worden als een onsamenvattend geheel. Maar dat is slechts een opname van een beeld. De werkelijkheid staat soms toe dat een elementair individueel object in verschillende classificaties verschillend getypeerd kan worden in een andere setting of view.

Montaignes principe

De indicatieve mood; gaat over wat we beweren waar te zijn.

De optitatie mood; gaat over wat we willen dat waar is

3.0.5 SIX Variable model

Optitatieve statements omschrijven de omgeving zoals we het willen zien vanwege de machine.

Indicatieve statements omschrijven de omgeving zoals deze is los van de machine.

Een requirement is een optitatief statement omdat ten doel heeft om de klantwens uit te drukken in een softwareontwikkel project.

Domein kennis bestaat uit indicatieve uitspraken die vanuit het oogpunt van software ontwikkeling relevant zijn.

Een specificatie is een optitatief statement met als doel direct implementeerbaar te zijn en ter verondersteuning van het natreven van de requirements.

Drie verschillende type domeinkennis: domein eigenschappen, domein hypothesen, en verwachtingen.

Domein eigenschappen zijn beschrijvende statementen over een omgeving en zijn feiten. Domein hypothesen zijn ook beschrijvende uitspraken over een omgeving, maar zijn aannames.

Verwachtingen zijn ook aannames, maar dat zijn voorschrijvende uitspraken die behaald worden door actoren als personen, sensoren en actuators.

Het verschil tussen essentie en incarnatie van een systeem. Een essentie bevestigt de mogelijkheden dat een systeem moet hebben om te voldoen aan de eisen, ongeacht hoe het systeem is geïmplementeerd. De incarnatie bevestigt of omvat de mogelijkheden die te maken hebben met details omtrent implementatie. Een heuristiek voor het identificeren van de essentie van een systeem is de aanname van perfecte technologie, ofwel de aanname dat de technologie binnen een systeem perfect is. Om essentie te identificeren nemen we aan dat technologie buiten de machine om perfect is. Zouden we incarnatie overwegen dan wordt de aanname van perfecte machine-externe technologie opgeheven.

Voor de documentatie van contextuele beslissingen en opties/alternatieven wordt de OVM (Orthogonale variability Model) gebruikt. Oorspronkelijk was deze methode bedoeld om de variatiepunten en de variant van een productlijn samen met hun variabele afhankelijkheden (mandatory, optional, alternative) en beperkende afhankelijkheden (requires en excludes) te omvatten. De variant kan worden gerelateerd aan een ontwikkelartefact zoals een requirement of een diagram als een zogenoemde arte-

fact dependency. Een artefact is dan gedefinieerd als variabele. Voor de documentatie van de keuzen die we maken is een selectie model gemaakt. We gebruiken het OVM voor de documentatie van contextuele beslissingen die moeten worden genomen, opties en alternatieven die selecteerbaar zijn, en de afhankelijkheden tussen hen. met behulp van de artefact dependency relateren we de alternatieven aan variabele elementen van de AND/OR graaf. Voor documentatie van de keuzes gebruiken we ook een selectiemodel. De kracht van het OVM model en de voornaamste reden deze methode te gebruiken is dat deze is in staat is om een variant te relateren aan een geheel model, een model element, of een selectie van een model.

AND/OR graaf wordt gebruikt voor de documentatie van refinement/decompositie of requirements. De AND/OR graaf is een directe, asyclische graaf met nodes knopen die requirements voorstellen en lijnen die AND-decomposities voorstellen en OR-decompositiestussen de requirements. Een decompositie van een requirement in een set van subrequirements R_1, \dots, R_n is een OR-decompositie iff die dusdanig aan een subrequirement voldoet en daarmee voldoet aan requirement R . Wat moet worden gedocumenteerd met betrekking tot de AND/OR graaf is de abeargumentering waarom elkeAND/OR-decomopositie voldoende is.

Conceptueel model

System requirement: uitspraak over wereld fenomenen (gedeeld of niet) of doelen die bereikt moeten worden. met enige regelmaat informeel, niet precies geformuleerd.
Software requirement/speci

catie: uitspraak over gedeelde fenomenen of doelen die de machine moet bereiken middels de onderdelen waar die machine uit bestaat of middels de fenomenen waar de machine controle over heeft. doorgaans preciezer, meetbaar, exact geformuleerd.

Systemen gaan een zekere interactie aan met hun omgeving: Sensoren: meten fenomenen uit de omgeving (temperatuur, druk, licht, geluid, etc.) actuatoren: ve-

randeren iets in de omgeving (mechanische, electrisch, pneumatisch, etc.) Software: Kan niet direct communiceren met de buitenwereld. Snapt derhalve niets van de buitenwereld. Kan alleen maar bestaan in en communiceren met het systeem.

Requirement elicitation technieken

Requirements elicitation technieken zijn methoden die een onderzoeker kan gebruiken om de behoeften van de stakeholders in kaart te brengen. De stakeholders vormen de belangrijkste groep die de doelstelling van een project vastlegt.

Enkele voorbeelden van requirement elicitation technieken zijn:

1. Interviews
2. Brainstorming sessions
3. Use case approach
4. Document analysis
5. Observation
6. Prototyping
7. Joint applicationdevelopment
8. Reverse engineering
9. Survey/ Questionnaire
10. Focus groups
11. Interface analysis
12. Stakeholder analysis

13. Card sorting laddering

14. Open ended-interview

Functionele en niet-functionele eisen specificaties

Het vier variabelen model

Systemen (met daarin software) en de bijbehorende vier variabelen:

Monitored variabelen

: door sensoren gekwanti-
ceerde fenomenen uit de omgeving

Controlled variabelen

door actuatoren bestuurde fenomenen uit de omgeving

Input variabelen

Output variabelen

Literatuuronderzoek

Literature Review

S.no	Author	Title	Findings	Gap in literature
S.no	Author	wanrooy _vab1991a.pdf	Findings	Gap in literature
S.no	Author	wa3300-bezuien2000(1).pdf	Findings	Gap in literature
S.no	Author	Title	Findings	Gap in literature
S.no	Author	Title	Findings	Gap in literature
S.no	Author	rapport-veiligheid-van-op-afstand-bediende-burggen.pd	Findings	Gap in literature
S.no	Author	pronk.pdf	Findings	Gap in literature
S.no	Author	Olieman1987a.pdf	Findings	Gap in literature
S.no	Author	richtlijnen-vaarwegen-2020.pdf	Findings	Gap in literature
S.no	Author	richtlijnen-vaarwegen-2017_tcm21-127359(1).pdf	Findings	Gap in literature
S.no	Author	Olieman1987a.pdf	Findings	Gap in literature
S.no	Author	Meijer1980b.pdf	Findings	Gap in literature
S.no	Author	Meijer1980c.pdf	Findings	Gap in literature
S.no	Author	kst-31200-A-80-b2.pdf	Findings	Gap in literature
S.no	Author	duurzaamheid _bij _de _ontwikkeling _van _reevesluis.pdf	Findings	Gap in literature
S.no	Author	De _deltawerken _Cultuurhistorie _ontwerpgeschiedenis _web-A.pdf	Findings	Gap in literature
S.no	Author	wa3300-Bezuijen2000.pdf	Findings	Gap in literature
S.no	Author	Sander van Alphen Haalbaarheidsstudie naar grote sluisdeuren uitgevoerd in hogesterktebeton.pdf	Findings	Gap in literature
S.no	Author	Dalmeijer1994a.pdf	Findings	Gap in literature
S.no	Author	Dalmeijer1994b.pdf	Findings	Gap in literature
S.no	Author	Dalmeijer1994c.pdf	Findings	Gap in literature
S.no	Author	ceg _pruijssers _1982.pdf	Findings	Gap in literature
S.no	Author	Capaciteitsanalyse _van _de _prinses_margrietsluis _in _lemmer _- _Marc _Lamboo.pdf	Findings	Gap in literature
S.no	Author	Boer1979a.pdf	Findings	Gap in literature

Chapter 4

Requirements en specificaties

4.0.1 Requirements

Requirements zijn alleen die eisen die gesteld worden aan het gedrag of de kwaliteit van het systeem om te voorzien in de behoeften van een belanghebbende uit de business.

- Een schip okmt aanvaren en geeft een signaal aan de sluis.
- Indien er meer dan twee schepen in de sluis zitten dan wordt het ship geplaatst in de wachrij.
- Een schip kan pas naar binnenrijden als de sluisdeuren open zijn, het stoplicht is op groen en er zijn minder dan 2 schepen in de sluis.
- Eenmaal in de sluis zal het schip moeten wachten op de sluis en de pomp.
- Een schip mag alleen uitvaren als de pomp klaar is, de sluisdeuren open.
- Een sluis ontvangt een aankomst signaal van een schip en bestuurt de sluisdeuren en de pomp.
- De sensor is een onderdeel van de sluis en ontvangt signalen van naderende schepen.
- De sluisdeur voor boven en beneden kunnen beiden open en dicht. De sluisdeur wordt aangestuurd door de sluis.

- Een pomp begint met pompen bij een signaal van de kluis. Een kluis io zijn beert geeft alleen een signaal aan de pomp als de sleudeuren dichtzijn
- Geen deadlock
- Voor geen enkel pad geldt dat als de deuren gesloten zijn volgens de kluis dat er een deur openstaat om een schip naar buiten te laten.
- Voor alle paden geldt dat als een sluis aan het voorbereiden is, dan zijn alle duren dcht.
- Voor alle paden geldt dat als een deur dicht is het aantal schepen in de kade gelijk is aan nul
- Voor een enkel pad geldt dat als het binnenstoplicht op groen staat dat het niet toegestaan in naar binnen te varen
- Voor alle paden geldt dat de globale tijd langer is dan 30 tijdseenheden
- Er is een pad waarvoor geldt dat als een schip wilt stoppen dat er meer dan 5 schepen in de sluis zitten.
- Voor alle paden geldt als schip vrtrekt is sluisdeur dicht
- Voor alle paden geldt als stoplicht op rood sluisdeuren dicht en schip vertrollen dan is de nivelleermachine uit
- Er is geen pad waarop een schip vertrekt vanuit de rechtersluisdeur en de linker-sluisdeur is open en linkeruitvaartstoplicht en linkeruitvaartsoplicht opgroen en nibelleermachine is aan
- Er is een pad waarvoor geldt dat linkersluisdeuren dicht zijn, rechtersluisdeuren dicht zijn rechteruitvaartstoplicht is rood en rechteruitvaartstoplicht is rood terwijl eer geen schip in de sluis licht

- Een stopluch staat altijd op groen als de deuren open staan en de pomp niet bezig is.
- In geen enkele staat van de sluis behalve tussen de lowergate en uppergate en uppergate en lowergate en de staten `AtArrivalLow` en `AtEnteringHigh` is de wachttijd langer dan 5 tijdseenheden
- Voor alle paden in een pomp geldt dat als water level lager is dan waterlaag pompwaterweg is altijd false
- Voor alle paden geldt dat als water level hoger is dan waterhoog dan is pompwater altijd false
- Het zal nooit gebeuren dat een pomp water toevoegt als deuren open zijn, geen schip in sluis en stoplicht op groen
- Het kan gebeuren dat bij pompr het stoplicht op rood staat, het schip in de sluis en deur is dicht, en waterstand gelijk aan waterlaag
- Er is een mogelijkheid dat vanuit pomp get stoplicht op rood wordt gezet en waterlevel gelijk is aan waterlaag
- Het kan voorkomen dat bij state pompaan het waterniveau gelijk is aan waterlaag
- Voor alle paden geldt dat er een mogelijkheid is dat deur is open/dicht en sluis nivelleert omhoog/omlaag

4.0.2 Veiligheidsoverwegingen

- Ik wil zeker zijn dat mijn schip niet tegen de sluisdeuren aanvaart als een stoplicht op groen is

- Ik wil er zeker van zijn dat mijn schip niet tegen de tweede deur vaaart als het eerste stoplicht op groen is
- Ik wil er zeker van zijn dat als mijn schip de sluis op hoog binnentreedt dat het waterniveau in de sluis gelijk is aan hoog.
- Ik wil er zeker van zijn dat als mijn schip de sluis op een laag waterpeil binnenvaart dat het waterniveau in de sluis gelijk is aan laag.
- Ik wil er zeker van zijn dat als mijn schip de sluis op laag binnenvaart dat het waterniveau in de sluis gelijk is aan laag.
- Ik wil een signaal wanneer er een schip in de sluis zit als sluisbediening
- Ik wil als sluiscontrollor een signaal als de deuren openstaan en een schip komt aanvaren en er is tegelijk een schip in de sluis.
- Ik wil max 2 schepen in de sluis
- Ik wil dat een schip de sluis pas na 5 seconden in de arrival state kan binnentreden
- Ik wil dat mijn stoplicht alleen bedient kan worden door de sluis
- Ik wil dat de deuren alleen bedient kunnen worden door de sluis
- Ik wil dat sensoren alleen bedient kunnen worden door de sluis
- Een schip moet een route kunnen aflaggen

4.0.3 Afbakning

- Wat doet de sluis niet.

- De sluis houdt geen rekening met links of rechtsrijdend verkeer vanuit de zeevaart
- De sluis heeft geen queue met daarin een id gekoppeld aan de sluis.
- De waterpomp wordt alleen aan en uitgezet
- De waterpomp houdt geen rekening met waterstand
- Houdt geen rekening met een schip in de sluis dat is blijven hangen.

$$D = \{ x \in \mathbb{N} \mid 1 \leq x \leq 100 \}$$

$$D = \{ x \in \mathbb{N} \mid 1 \leq x \leq 100 \}$$

$$D = \left\{ x \in \mathbb{N} \mid 1 \leq x \leq 100 \right\}$$

$$D = \left\{ x \in \mathbb{N} \mid 1 \leq x \leq 100 \right\}$$

$$D = \left\{ x \in \mathbb{N} \mid 1 \leq x \leq 100 \right\}$$

$$D = \left\{ x \in \mathbb{N} \mid 1 \leq x \leq \frac{200}{2} \right\}$$

Sisasetoffinitestates

$S0 \subseteq Sisdasetvaninitielestatess$

$S0 \subseteq SxSiseentranstiterelatiedietotaalmoetzij, datbetekent, datvoorelkestates \in Sereenstatsiss'$

$SzodatR(s,s') \ L \leftarrow \Box a$

$\forall x \exists y \implies$

$\forall x \exists y \cap \subset \in \vee \Diamond \dashv \exists \pm$

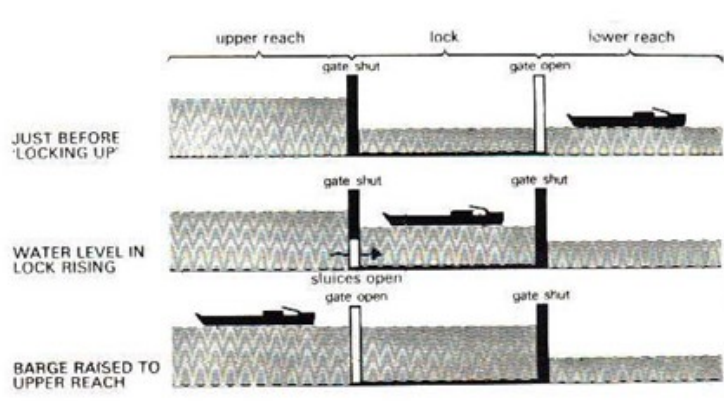
Chapter 5

Ontwerp

5.1 Inleiding

Aankomst, uitvoering, vrijgave

ontwerp



Onderdelen

Op basis van de schets kunnen we vaststellen dat een sluismodel uit de volgende onderdelen bestaat.

1. Een tweetal sluisdeuren.
2. Een sluiskolk waarin de schepen in- en uitvaren
3. een stoplicht om een signaal af te geven voor invaren en uitvaren.
4. Een nivelleermachine zorgt ervoor dat het water in de sluis op het gewenste niveau wordt gebracht

5. Een control-system dat ervoor zorgt dat de opdrachten van de sluisbeheerder (geautomatiseerd) worden uitgevoerd

Werking

Een schip komt aanvaren en meld zich aan bij de sluismeester. De sluismeester geeft een signaal aan het controlsysteem voor het openen van de sluisdeuren, nadat gecontroleerd is of de nivelleermachine al klaar is. Als er ruimte is voor een invarend schip mag het schip dat zoich heeft aangemeld en toestemming heeft in de sluis varen. Op het moment dat de sluis vol is gaan de sluisdeuren dicht. Eenmaal afgesloten kan de nivelleermachine beginnen om het water in de sluiskolk op het gewenste waterpeil te brengen. Als dit nivelleerprces is afgerond geeft het controlsysteem daan da de sleusdeuren open kunnen. Als de sleusdeuren open zijn en het uitvaarsignaal is op groen dan moet het schip in de sluis de sluis uitvaren.

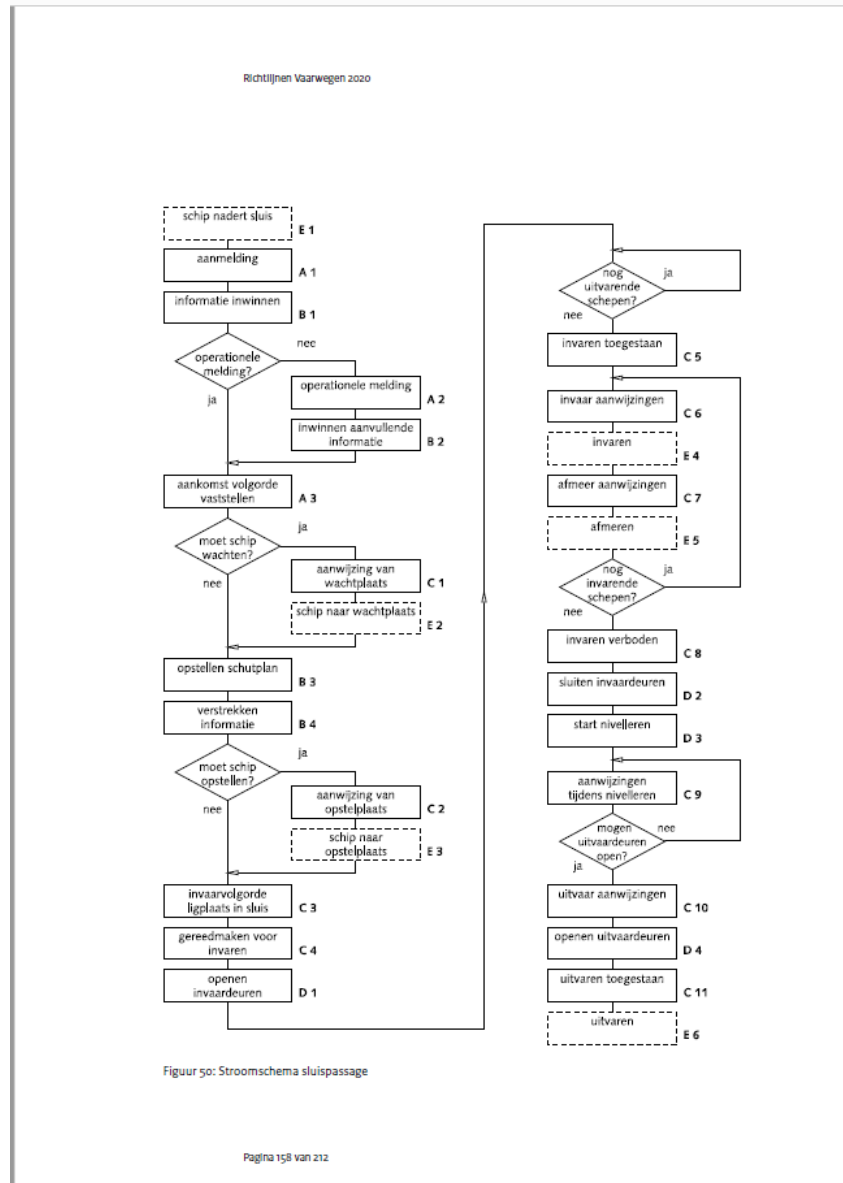
extra cases Uit het zojuist genoemde scenario valt het volgende op te maken.

1. Een schip geeft een signaal aan een sluismeester.
2. Er wordt gekeken of er wel plek is in de sluis .
3. Er wordt gekeken of de nivelleermachine is afgerond.
4. Er wordt gekeken wat het niveo van de waterpeil in de sluiskolk is.
5. Er wordt gekeken of de sluisdeuren gereed zijn voor invarende schepen.

Aandachtspunten

1. Voorrang tussen schepen onderling in de sluis?
2. Hoe lang mag een schip zich in de sluis bevinden?

5.2 Concept



Vooraanmelding

informatie inwinnen

operationele melding

aankomst volgorde

aanwijzen wachtplaats

verstrekken informatie

aanwijzen opstelplaats

opstellen schutproces

verstrekken informatie

invaarvolgorde en ligplaats in sluis

gereedmaken voor invaren

openen invaardeuren

invaren toegestaan

aanwijzingen voor invaren

aanwijzingen tijdens afmeren

invaren verboden

sluiten invaardeuren

start nivelleren

aanzwijzingen voor uitvaren

openen uitvaardewuren

uitvaren toegestaan

brainstorm 22-5-2022

invaardeuren en uitvaardeuren Gaan we uit van binnendeuren en buitendeuren? Er ontstaat dan een extra ruimte in de sluis. Hoeveel schepen kunnen in deze ruimte? Wat is de maximale wachtreij in deze ruimte en wat zijn de verkeersregels in deze ruimte?

invaarstoplicht en uitvaarstoplicht Als invaren is toegestaan hoe wordt dit dan doorgegeven aan de schepen in de sluis? moeten zij dan uit zichzelf wachten of krijgen zij een signaal dat zij wel/niet mogen uitvaren? En moeten zij dan kiezen voor links, midden of rechts? Of maakt dat allemaal niets uit?

invaarwachtrij en uitvaarwachtrij Als er meerder schepen in een sluiskolk zitten moet het systeem dan rekening houden met het schip dat als eerste is ingevaren en/of het langst in de sluis zit?

Chapter 6

Kripke model

6.1 Uppaal kripke structuren

templates

Schip

Sluis

Aanvoer

Afvoer

Pomp

Pompbediening

Stoplicht

Deur

case Als een schip van rechts binnen komt en sluisdeuren zijn dicht dan moet het stoplicht op rood, de pomnp in transitie van laag naar hoog en niet andersom

case Voor invaren geldt altijd: waterlevel, pomp uit, sluisdeuren open en stoplicht op groen

case uitvarenden hebben voorrang op invarenden

case Voor invarenden geldt pomp uit, sluisdeur open en stoplicht op groen

case voor nivelleren geldt pomp is aan, sluisduren zijn doicht en het stoplicht is op rood

case Als een schip vertrekt dan zijn altijd, sluisdeuren open, waterlevel gereed op niveau 5 of 0 en stoplicht direct op groen

case urgent locations; het is niet mogelijk om hier te wachten

case urgent syn; een synchronisatie moet direct worden uitgevoerd als de guards geldig zijn

case als een schip binnen is, en er zijn wachtende schepen, dan moet het stoplicht via oranje naar rood

case committed; als deze staat actief is dan wordt de eerst volgende transitie uitaande van deze state

case als een schijp binnen vaart mnoiet hij ook eft binnen zijn en niet binnen-varen, dit geldt ook voor sluisdeuren en pompen dus deze zijn committed.

case

case

case

Parallele compositie

Parallele compositie

Modeleigenschappel

Parallele compositie Om een sluispark te kunnen modelleren meerdere templates die de verschillende abstracties van het systeem aantonen.

Synchronisatie Zorgt ervoor dat een transitie die genomen worden in de ene kripke structuur op hetzelfde moment wordt opgenomen in een andere kripke structuur.

Chapter 7

CTL logica

7.1 Doel van de test

7.1.1 Wat wordt getest en hoe

7.1.2 toetsen met queries

7.1.3 Operator: AG

Voor alle paden

7.1.4 Operator: EG

Uiteindelijk geldt er een pad waarvoor geldt

7.1.5 Operator: AF

Voor alle paden/richtingen vroeg of laat

7.1.6 Operator: EF

Er is een pad

7.1.7 Operator: AX

Alle opvolgende toestanden

(?)

7.1.8 Operator: EX

Er bestaat vanaf de volgende minstens 1 state waarvoor geldt

7.1.9 Operator: $p \cup q$

Er geldt p tot q (?)

7.1.10 Operator: $p \mathbf{R} q$

q moet waar zijn totdat en inclusief de situatie dat p voor het eerst waar is, als p niet geldig is, dan moet q vooraltjd geldig zijn

Chapter 8

Testresultaten CTL logica

1	$A[] \text{!deadlock}$	TRUE
2	$A[] \text{ not (Sluis.Tussenstop5 \&\& Deur.Klaar_voor_uitvaart)}$	Disconnected
3	$A[] (\text{Sluis.Vorbereiden} \text{ imply } \text{Deur.Dicht})$	TRUE
4	$A[] (\text{Deur.Dicht} \text{ imply } \text{Counter}==0)$	TRUE
5	$A[] (\text{Buitenstoplicht.Groen} \text{ imply } \text{invaren_allowed}==\text{true})$	TRUE
6	$A[] \text{!} (\text{Binnenstoplicht.Groen} \text{ imply } \text{invaren_allowed}==\text{false})$	FALSE
7	$A[] (\text{globale_tijd} \leq 30)$	FALSE
8	$E[] (\text{Schip.Stoppen} \text{ and } (\text{Counter} \leq 5))$	Ship not a structure
9	$A[] (\text{Schip.Vertrekken} \text{ imply } \text{Sluisdeur.Dicht})$	-

resultaat

verklareing

8.0.1 Fairness

$AG(AF(p))$

In welke staat de automaat zich ook bevindt, in alle richtingen kom je vroeg of laat

8.0.2 Liveness

Altijd en overal geldt: Als p geldt dan geldt vroeg of laat q
 Ookal treedt p nooit p volgens de logica klopt het dan dat q volgt uit p.
 In een situatie, waarin p nooit optreedt, spreekt men van een
 vacuous truth.

1.8233,1.8233,1.8233,0.9243,0.8651,0.9013,0.3217,0.3377,34.4858
 0.2753,0.2753,0.2753,0.5383,0.5038,0.5249,0.3217,0.3377,8.4552
 0.0898,0.0898,0.0898,0.2804,0.2625,0.2734,0.3217,0.3377,1.5514
 0.4689,0.1193,0.0417,0.8046,0.2227,0.1795,0.6413,0.3307,6.4488
 0.339,0.8068,0.0936,0.4335,0.8036,0.2434,0.3046,0.624,20.8422
 0.2162,0.133,0.8711,0.1707,0.1503,0.8215,0.1562,0.0692,2.3365
 0.5187,0.9138,1.0432,0.4332,0.8028,0.8406,0.2269,0.3404,22.8164
 0.58,0.2096,0.9086,0.808,0.2134,0.8294,0.333,0.1596,8.4349
 0.8237,0.9378,0.0855,0.8654,0.8096,0.2487,0.4338,0.5187,26.9923

8.0.3 safety

8.0.4 zeno vrij

Geen enkele state kan oneindig een transitie uitvoeren. Elke state heeft een uitgaande transitie.

8.0.5 deadlocks

Chapter 9

Conclusie

Chapter 10

Discussie

10.0.1 Future work

Hoogte waterniveau

type deuren naar waterniveau

De sluis kan ook rekening houden met waterniveau van hoog naar laag. Als een schip naar binnen vaart moet de sluis weten welk schip ook weer naar buiten vaart en aan welke kant.

voorrang uitvarend op invarend

Als een schip uitvaart komt er een moment dat een sluis ruimte vrij heeft. Voordat de sluisdeur sluit nadat een schip is vertrokken kan er nog een polling worden gedaan naar alle schepen in de buurt om te zien of deze willen en kunnen invaren.

stoplicht invarend en stoplicht uitvarend

Een handige functionaliteit is dat voor invarende schepen er een stoplicht is en voor uitvarende schepen. Anders ontstaat er een probleem van collision.

Volgorde

Kunnen aantonen dat schepen kunnen worden behandeld met voorkeur, wie het eerst komt die het eerste in behandeling wordt genomen.

Chapter 11

Notulen

pgf /form field1/.store in=, field2/.store in=,

Actielijst			
Onderwerp	Besluit	Wie	Gereed
Orange	Fruit	Vitamin C	It is fruit, which is full of nutrients and low in calories. They can promote clear, healthy skin and also lowers the risk for many diseases. It reduces cholesterol and also helps in building a healthy immune system.
Cauliflower	vegetable	B-Vitamins	It is the vegetable, which is high in fiber and B-Vitamins. It also provides antioxidants, which help in fighting or protect against cancer. It enhances digestion and has many other nutrients.

Chapter 12

Queries

Queries

```
Sluis.Draining-->Deuren.laag_open
Deuren.laag_open-->Stoplicht.Green
E<> (Ship.ship_can_move&&Stoplicht.Green)
A[] not (Stoplicht.Green && not (Deuren.hoog_open||Deuren.laag_open||Deuren.stopga
A[] not ((Deuren.hoog_open||Deuren.laag_open||Deuren.Opening_laag||Deuren.Opening_
Sensor.Wait-->Sensor.Wait
Stoplicht.Green-->Stoplicht.Green
(Deuren.hoog_open||Deuren.laag_open)-->(Deuren.laag_open||Deuren.hoog_open)
Deuren.laag_open-->Deuren.Closed
Deuren.hoog_open-->Deuren.Closed
Deuren.Closed-->Stoplicht.Red
Ship.ship_can_move-->Deuren.Closed
Deuren.hoog_open-->Stoplicht.Green
Ship.ship_can_move-->Stoplicht.Green
A[] not (Deuren.laag_open && Deuren.hoog_open)
Ship.ship_can_move-->Ship.ship_can_move
A[] not (Deuren.laag_open && Sluis.water != Sluis.water_laag)
A[] not (Deuren.hoog_open && Sluis.water != Sluis.water_hoog)
A[]not deadlock
```

Project declaraties

//Declarations

```
chan boot_hoog;
chan boot_laag;
```

```

chan changedoor_low;
chan changedoor_high;
chan ship_moves;
chan ship_abletomove;
chan changelight;

```

```

\\Sluis declaraties
const int water_laag=0;
const int water_hoog=10;
const int water_median=(water_hoog+water_laag)/2;
int[water_laag,water_hoog] water=water_median;
clock x;
\\Stoplicht declaraties

```

```

\\Ship declaraties
clock x;
\\Sensor declaraties

```

```

\\Deuren declaraties
bool stoplicht_hoog=false;
bool stoplicht_laag=false;
clock x;

```

```

\\System declaraties
system Deuren,Sensor,Sluis,Ship,Stoplicht;

```

Uitleg

Als het schip boven is, dan is waterlvel gelijk aan hoog, filling valve is dicht,
 Schip is in waterlock, waterlevel is hoog, filling valve is dicht, lower gates ges
 Schip is dan laag, waterlevel gelijk aan laag, filling valve is dicht, lowergates
 AtArrivalHigh

AtArrivalLow

Als schip beneden is dan is waterlevel gelijk aan laag, filling valve is dicht, lo
Schip is in water lock, waterlevel is laag, flilling valve is open, lower gates zi
Schip is dan hoog, waterlevel is gelijk aan hoog, filling valve is dicht, uppergat

Chapter 13

Testresultaten

onderdeel van de test

project name						
Test case ID				Test designed by		
test priority (low/medium/high)				Test design date		
Module name				Test executed by		
Test title				Test execution date		
Description						
Pre condition						
Dependencies						
Step	Test steps	Test data	expected result	Actual result	(pass or fail)	notes

Chapter 14

Reflectie

Ik heb erg veel geleerd van het analyseren van de verschillende requirements en specificaties en het opzetten van een model in Uppaal. Een dergelijk model opzetten had ik namelijk nog nooit gedaan. Het uitvoeren van onderzoek heb ik eerder gedaan. Ook de toetsing van het model met behulp van proposities heb ik nog nooit gedaan. Verder heb ik de kennis die had van programmeren/ design patterns gebruikt om de verschillende templates in mijn Uppaal model van elkaar te onderscheiden. Het leukste onderdeel van het project vond ik hoe mijn templatemodel deadlockvrij werkte. Voor de verificatie van het model heb ik veel achtergrondinformatie opgezet, en het is mooi om te zien dat je met enkele duidelijke zinnen kan aantonen of een propositie geldig is of niet. Verder had ik moeite met het opstellen van de juiste veiligheidseisen bij het model. Ik had aangenomen dat ik het project niet zou halen omdat ik de opdracht niet in teamverband heb uitgevoerd. Ik ben toch blij dat ik een concept heb opgeleverd dat ik kan toetsen aan de doormijzelf opgestelde eisen en dat ik met mijn huidige kennis de proposities uit de requirements kan toetsen.

```
pgf /form field1/.store in=, field2/.store in=,  
/form field1 = G. Wales, field2 = Mathematics
```

Naam vergadering	Datum en plats	Namen aanwezigen
Opening en goedkeuring		
Ingekomen stukken en rondvraag		
Sluiting		

Chapter 15

Conclusions

15.1 Summary of the thesis

Time to get philosophical and wordy.

Important: In the list of references at the end of thesis, abbreviated journal and conference titles aren't allowed. Either you must put the full title in each item, or create a List of Abbreviations at the beginning of the references, with the abbreviations in one column on the left (arranged in alphabetical order), and the corresponding full title in a second column on the right. Some abbreviations, such as IEEE, SIGMOD, ACM, have become standardized and accepted by librarians, so those should not be spelled out in full.

Appendix A

Uppaal source

```
// Place global declarations here.
```

```
// knop
chan klik;
```

```
// alle lampen tegelijk
//broadcast chan klik;
```

```
// max aantal lampen
const int MAX=4;
```

```
//declare voor een lamp
typedef int[0,MAX-1] id_l;
```

```
// lichtintensiteit in lu
typedef int[0,10] lumen;
```

```
lumen bright;
```

```
// locale variabelen voor de template lamp: geef de template de volgende paramete
```

```
void functie()
```

```
{
```

```
}
```

```

lumen sterkte;

// wacht 5 tijdseenheden
clock x;

// een constraint op een bepaalde variabele
bool isForMe()
{
// return false;

if(lampnr < 2 && bright<6) return true;
else if(lampnr>1 && bright>=6) return true;
else return false;

}

//verschillende tijdseenheden voor even en oneven lampnummers

// Place local declaraties voor de knop

clock x;

```

A.1

Appendix B: Model eerste deelname aan cursus 2020

Text of Appendix B is Here

Queries

```

Sluis.Draining-->Deuren.laag_open
Deuren.laag_open-->Stoplicht.Green
E<> (Ship.ship_can_move&&Stoplicht.Green)
A[] not (Stoplicht.Green && not (Deuren.hoog_open||Deuren.laag_open||Deuren.stopga
A[] not ((Deuren.hoog_open||Deuren.laag_open||Deuren.Opening_laag||Deuren.Opening_
Sensor.Wait-->Sensor.Wait
Stoplicht.Green-->Stoplicht.Green
(Deuren.hoog_open||Deuren.laag_open)-->(Deuren.laag_open||Deuren.hoog_open)
Deuren.laag_open-->Deuren.Closed
Deuren.hoog_open-->Deuren.Closed
Deuren.Closed-->Stoplicht.Red
Ship.ship_can_move-->Deuren.Closed
Deuren.hoog_open-->Stoplicht.Green
Ship.ship_can_move-->Stoplicht.Green
A[] not (Deuren.laag_open && Deuren.hoog_open)
Ship.ship_can_move-->Ship.ship_can_move
A[] not (Deuren.laag_open && Sluis.water != Sluis.water_laag)
A[] not (Deuren.hoog_open && Sluis.water != Sluis.water_hoog)
A[]not deadlock

```

Project declaraties

//Declarations

```

chan boot_hoog;
chan boot_laag;
chan changedoor_low;
chan changedoor_high;
chan ship_moves;
chan ship_abletomove;

```

```
chan changelight;
```

```
\\Sluis declaraties
```

```
const int water_laag=0;
```

```
const int water_hoog=10;
```

```
const int water_median=(water_hoog+water_laag)/2;
```

```
int[water_laag,water_hoog] water=water_median;
```

```
clock x;
```

```
\\Stoplicht declaraties
```

```
\\Ship declaraties
```

```
clock x;
```

```
\\Sensor declaraties
```

```
\\Deuren declaraties
```

```
bool stoplicht_hoog=false;
```

```
bool stoplicht_laag=false;
```

```
clock x;
```

```
\\System declaraties
```

```
system Deuren,Sensor,Sluis,Ship,Stoplicht;
```

A.2

Appendix B: Model herkansing tweede deelname aan cursus 2020

Text of Appendix B is Here

```
// Place global declarations here.
```

```
/*
```

```
Project working
```

```
AtArrival
```

```
StoplightRed
```

```
DoorOpen
```

```
StoplightGreen
```

```
Startmove
```

```
Sensor
```

```
SchipEntered
```

```
Doorclosed
```

```
StoplightRed
```

```
-----
```

```
Nivelleer started
```

```
Nivelleer stopped
```

```
Waterlevel equilibrium
```

```
-----
```

```
AtLeaving
```

```
Stoplightred
```

```
Dooropened
```

```
Stoplightgreen
```

```
StartMove
```

```
Sensor
```

```
SchipHasLeft
```

```
Doorclosed
```

```
StoplightRed
```

Uitleg

Als het schip boven is, dan is waterlvel gelijk aan hoog, filling valve is dicht,
 Schip is in waterlock, waterlevel is hoog, filling valve is dicht, lower gates ges
 Schip is dan laag, waterlevel gelijk aan laag, filling valve is dicht, lowergates
 AtArrivalHigh

AtArrivalLow

Als schip beneden is dan is waterlevel gelijk aan laag, filling valve is dicht, lo
 Schip is in water lock, waterlevel is laag, flilling valve is open, lower gates zi
 Schip is dan hoog, waterlevel is gelijk aan hoog, filling valve is dicht, uppergat

*/

```
const int N = 2;          // # trains
typedef int[0,N-1] id_t;
```

```
chan      appr[N], stop[N], leave[N];
urgent chan go[N];
```

```
// waterniveau in in meter van 0 tot 10
typedef int[3,10] waterniveau;
```

```
waterniveau level;
```

```
//doors
```

```
chan lower_gate;
```

```
chan upper_gate;
```

```
//filling
```

```
chan emptying_valve;
```

```
chan filling_valve;
```

```
bool nivelleer_sessie_bezig;
```

```
// water level
```

```
chan high_water_level;
```

```

chan low_water_level;
//sluices
chan signal_sluis_low[N];
chan signal_sluis_high[N];
//
chan move[N];
//
chan groen;
chan rood;

clock central;

\\geef de schip parameter const id_t id
// Place local declarations here.
clock schip_clock;

\\sensor declaraties
clock x;

\\sluis declaraties

const int water_laag=3;
const int water_hoog=10;
const int water_median=(water_hoog+water_laag)/2;
int[water_laag,water_hoog] water=water_median;
// level wordt gelijk gezet met temp
// temp is gelijk aan waterniveau
clock sluis_clock;
id_t list[N+1];
int[0,N] len;
bool contentHigh, contentLow;
// Put an element at the end of the queue
void enqueue(id_t element)

```

```

{
list[len++] = element;
}

// Remove the front element of the queue
void dequeue()
{
int i = 0;
len -= 1;
while (i < len)
{
list[i] = list[i + 1];
i++;
}
list[i] = 0;
}

// Returns the front element of the queue
id_t front()
{
return list[0];
}

// Returns the last element of the queue
id_t tail()
{
return list[len - 1];
}

\\stoplicht declaraties
clock stoplicht_clock;

```



```
\\pomp declaraties
```

```
const int water_laag=3;
const int water_hoog=10;
const int water_median=(water_hoog+water_laag)/2;
int[water_laag,water_hoog] water=water_median;
clock pomp_clock;
// waterniveau van de sensor voor de sluis is gelijk aan level
waterniveau depth;
```

```
// een constraint op een bepaalde variabele
```

```
bool isForLow()
```

```
{
```

```
// return false;
```

```
if( level>=3) return true;
```

```
else return false;
```

```
}
```

```
bool isForHigh()
```

```
{
```

```
// return false;
```

```
if( level>=6) return true;
```

```
//else if(level>=6) return true;
```

```
else return false;
```

```
}
```

//verschillende tijdseenheden voor even en oneven lampnummers

References

- Debreuve, E., Barlaud, M., Aubert, G., Laurette, I., and Darcourt, J. (2001). Space-time segmentation using level set active contours applied to myocardial gated SPECT. *IEEE Trans. Med. Imag.*, 20(7):643–659.
- Lamport, L. (1985). *TEX—A Document Preparation System—User’s Guide and Reference Manual*. Addison-Wesley.

Mijn werkzaamheden

Galvin Bartes

Basically, this needs to be worked out by each individual, however the same format, margins, typeface, and type size must be used as in the rest of the dissertation.