

Design Document

Sluis Opdracht



Document Naam: Design Document
Auteur(s): Rafal Grasman (Revisies 1 en 2), Ian Seelen (Revisie 1)
Klas: T34
Revisie: 2
Datum: 2017-01-12
Status: Final

Documenthistorie

Revisies

Revisie	Status	Datum	Wijzigingen
1	Final	2017-01-06	Design document gemaakt
2	Final	2017-01-16	Design document layout verbeterd, design zelf uitgebreid (sequence, class, interaction en state diagrammen), en afgemaakt.

Goedkeuring

Dit document heeft de volgende goedkeuringen:

Revisie	Datum goedkeuring	Naam	Functie
1	2017-01-06	Ian Seelen	Programmeur, Designer, Auteur
1	2017-01-06	Rafal Grasman	Programmeur, Designer, Auteur
2	2017-01-16	Rafal Grasman	Programmeur, Designer, Auteur

Distributie

Dit document is verstuurd aan:

Revisie	Datum verzending	Naam	Functie
1	2017-01-06	Peter Dingemans en Ronald Marcelis	Docenten Fontys ICT
2	2017-01-16	Peter Dingemans en Ronald Marcelis	Docenten Fontys ICT

Inhoudsopgave

Documenthistorie	2
Revisies	2
Goedkeuring	2
Distributie	2
Inhoudsopgave	3
Inleiding	4
Use cases	5
Vrijgeen-Invaren	5
Vrijgeen-Uitvaren	6
Alarm	7
Herstel	8
Schutten	9
Klassen Diagram	10
State Diagrammen	12
Main programma	12
Traffic Lights	12
SluiceController	13
Door	14
Sequence Diagrammen	16

Inleiding

Dit document is voortgekomen uit eerder opgestelde requirements in de sluis opdracht van PRC3. Een beknopte versie van deze requirements is hieronder te lezen.

Een sluizencomplex moet computer gestuurd worden. De toezichthouder moet 5 knoppen beschikbaar gesteld krijgen: Schutten, Alarm, Herstel, Vrijgeven voor invaren, Vrijgeven voor uitvaren. Het programma verbindt met het sluis-systeem dat volgens een protocol werkt. De applicatie gebruikt dit protocol om:

- Te schutten; deuren bij hetzelfde waterniveau sluiten, het water wordt naar de andere hoogte gepompt (laag->hoog of hoog->laag), deuren bij hetzelfde waterniveau worden geopend.
- Te alarmeren; bij een alarm stopt alles om voor een veilige omgeving te zorgen
- Te herstellen; vanaf een alarmsituatie wordt teruggegaan naar de vorige situatie en wordt de vorige opdracht gecontinueerd.
- Te vrijgeven; dit beheerst de rode en groene lichten aan beide kanten van de deuren om de boten uit of in te laten varen (in totaal 4 lichten, elk licht bestaande uit 1 rode en 1 groene lamp)

Ook zijn er requirements voor twee verschillende deuren: een sluizencomplex die deuren heeft die om manueel gesloten/geopend moeten worden, en een sluizencomplex die deuren heeft waarbij de motor om de 1 seconde opnieuw aangezet moet worden bij een sluit/open operatie.

Elke deur heeft drie Valves die de toe/afvoer van water beheersen. Water moet stapsgewijs worden opgehoogd/verlaagd (3-2-1/1-2-3, niet alle valves tegelijk).

De applicatie die geschreven wordt verbindt met 1 sluis. Indien meerdere sluizen bestuurd moeten worden, moet voor elke sluis de applicatie met een andere poort worden opgestart (door te geven via commandline parameters: ./applicatie 5555)

Use cases

Dit hoofdstuk beschrijft de usecases van het systeem. Deze usecases zijn afgeleid uit het protocol en requirements document die geleverd zijn door deze opdracht.

Usecase naam	Vrijgeven-Invaren
Doel in context	De sluis wordt vrijgegeven voor invaren
Preconditions	<ul style="list-style-type: none"> - Het systeem staat aan - Heeft verbinding met de sluis - Sluisdeuren zijn geopend bij hetzelfde waterniveau - Systeem is niks anders aan het doen (alarm situatie of schutten)
Successful end condition	<ul style="list-style-type: none"> - Het rode licht (aan de buitenkant bij hetzelfde waterniveau) staat uit - Het groene licht (aan de buitenkant bij hetzelfde waterniveau) staat aan
Failed end condition	Rode licht blijft rood of groene licht blijft uit
Trigger	De operator drukt de knop 'Invaren Vrijgeven' in
Main flow	
<ol style="list-style-type: none"> 1. Applicatie request van de sluis om rood aan de buitenkant bij hetzelfde waterniveau uit te doen 2. Applicatie request van de sluis om groen aan de buitenkant bij hetzelfde waterniveau aan te doen 	
Exceptional flow	
<ol style="list-style-type: none"> 1. Geen ACK ontvangen van de sluis – iets is fout gegaan, onbekend wat, onvoorzien door het protocol, geen mogelijkheid tot herstel of opvragen van missende ACK's 2. Applicatie moet opnieuw gestart worden 	
<ol style="list-style-type: none"> 1. De verbinding valt weg – iets is fout gegaan, onbekend wat, onvoorzien door het protocol 2. Applicatie moet opnieuw gestart worden 	

Usecase naam	Vrijgeven-Uitvaren
Doel in context	De sluis wordt vrijgegeven voor uitvaren
Preconditions	<ul style="list-style-type: none"> - Het systeem staat aan - Heeft verbinding met de sluis - Sluisdeuren zijn geopend bij hetzelfde waterniveau - Systeem is niks anders aan het doen (alarm situatie of schutten)
Successful end condition	<ul style="list-style-type: none"> - Het rode licht (aan de binnenkant bij hetzelfde waterniveau) staat uit - Het groene licht (aan de binnenkant bij hetzelfde waterniveau) staat aan
Failed end condition	Rode licht blijft rood of groene licht blijft uit
Trigger	De operator drukt de knop 'Uitvaren Vrijgeven' in
Main flow	
<ol style="list-style-type: none"> 1. Applicatie request van de sluis om rood aan de binnenkant bij hetzelfde waterniveau uit te doen 2. Applicatie request van de sluis om groen aan de binnenkant bij hetzelfde waterniveau aan te doen 	
Exceptional flow	
<ol style="list-style-type: none"> 1. Geen ACK ontvangen van de sluis – iets is fout gegaan, onbekend wat, onvoorzien door het protocol, geen mogelijkheid tot herstel of opvragen van missende ACK's 2. Applicatie moet opnieuw gestart worden 	
<ol style="list-style-type: none"> 1. De verbinding valt weg – iets is fout gegaan, onbekend wat, onvoorzien door het protocol 2. Applicatie moet opnieuw gestart worden 	

Usecase naam	Alarm
Doel in context	De sluis wordt in een noodsituatie stil gezet
Preconditions	<ul style="list-style-type: none"> - Het systeem staat aan - Heeft verbinding met de sluis
Successful end condition	<ul style="list-style-type: none"> - Indien deuren dicht – valves (kleppen) ook dicht - Indien deuren openen/sluiten – worden deuren direct gestopt
Failed end condition	<ul style="list-style-type: none"> - Indien deuren dicht – valves (kleppen) zijn open - Indien deuren openen/sluiten – deuren gaan door met openen/sluiten
Trigger	De operator drukt de knop 'Alarm' in
Main flow	
<ol style="list-style-type: none"> 1. Kleppen gaan dicht 2. Alle rode lichten aan, alle groene lichten uit <ol style="list-style-type: none"> 1. Deuren stoppen met openen/sluiten 2. Alle rode lichten aan, alle groene lichten uit 	
Exceptional flow	
<ol style="list-style-type: none"> 1. Geen ACK ontvangen van de sluis – iets is fout gegaan, onbekend wat, onvoorzien door het protocol, geen mogelijkheid tot herstel of opvragen van missende ACK's 2. Applicatie moet opnieuw gestart worden <ol style="list-style-type: none"> 1. De verbinding valt weg – iets is fout gegaan, onbekend wat, onvoorzien door het protocol 2. Applicatie moet opnieuw gestart worden 	

Usecase naam	Herstel
Doel in context	De sluis wordt in vanaf een noodsituatie terug gezet naar waar die mee bezig was
Preconditions	<ul style="list-style-type: none"> - Het systeem staat aan - Heeft verbinding met de sluis - Systeem is in alarm staat
Successful end condition	<ul style="list-style-type: none"> - Indien deuren dicht – valves (kleppen) open - Indien deuren open – deuren gaan verder met openen/sluiten
Failed end condition	De sluis doet niet wat er voor de alarmsituatie gedaan werd
Trigger	De operator drukt de knop 'Herstel' in
Main flow	
1. Kleppen gaan dicht 1. Deuren stoppen met openen/sluiten	
Exceptional flow	
1. Geen ACK ontvangen van de sluis – iets is fout gegaan, onbekend wat, onvoorzien door het protocol, geen mogelijkheid tot herstel of opvragen van missende ACK's 2. Applicatie moet opnieuw gestart worden 1. De verbinding valt weg – iets is fout gegaan, onbekend wat, onvoorzien door het protocol 2. Applicatie moet opnieuw gestart worden	

Usecase naam	Schutten
Doel in context	De sluis gaat het waterniveau verhogen (indien laag) of verlagen (indien hoog)
Preconditions	<ul style="list-style-type: none"> - Het systeem staat aan - Heeft verbinding met de sluis - Systeem is niet in een alarm staat
Successful end condition	<ul style="list-style-type: none"> - Water binnen sluis is verhoogd indien deze laag was - Water binnen sluis is verlaagd indien deze hoog was
Failed end condition	<ul style="list-style-type: none"> - De waterstand is niet aangepast - Vergrendeling kapot - Motor kapot
Trigger	De operator drukt de knop 'Schutten' in
Main flow	
<ol style="list-style-type: none"> 1. Alle groene lichten gaan uit 2. Alle rode lichten gaan aan 3. Deuren waar waterniveau hetzelfde is sluiten (indien niet gesloten) 4. De valves (kleppen) gaan open: <ol style="list-style-type: none"> a. Onderste klep indien water hoog is bij begin b. Onderste klep indien water laag is bij begin, daarna middelste klep als waterniveau boven onderste klep is, daarna hoogste klep wanneer waterniveau boven middelste klep is 5. Deuren waar waterniveau hetzelfde is openen 	
Exceptional flow	
<ol style="list-style-type: none"> 1. Geen ACK ontvangen van de sluis – iets is fout gegaan, onbekend wat, onvoorzien door het protocol, geen mogelijkheid tot herstel of opvragen van missende ACK's 2. Applicatie moet opnieuw gestart worden <ol style="list-style-type: none"> 1. De verbinding valt weg – iets is fout gegaan, onbekend wat, onvoorzien door het protocol 2. Applicatie moet opnieuw gestart worden <ol style="list-style-type: none"> 1. De knop alarm wordt ingedrukt, zie use case 'Alarm' 	

Klassen Diagram

Zie het klassendiagram op de volgende pagina. De volgende uitleg hoort bij het klassendiagram:

Omdat de communicatie verloopt over een TCP verbinding met ascii tekst als commando's, en de states en variabelen zoveel mogelijk strings vermijden om data te onhouden, zijn *mappers* nodig die van enum/value naar string en terug kunnen vertalen. Dit is ook te zien in bijlage 1 (het klasse diagram).

Er zijn acties, variabele en objecten afhankelijk van timing en daarvoor is de Timer klasse in het dessin ontstaan.

Er zijn 4 hoofdcomponenten die samenwerken: SluiceLogic, Networking, Sluice en Door. De volgende objecten zijn in de requirements geïdentificeerd: Door, Valve, TrafficLight, Sluice.

De volgende afleidingen zijn in de requirements geïdentificeerd: DoorOneSecondMotor, DoorTwoSecondLock.

Een TrafficLight is een combinatie van 2 Lights: 1x RedLight en 1x GreenLight.

Een Door bestaat uit 3 Valves (voor low, mid en high waterniveau) en 2 TrafficLights (eentje binnen en eentje buiten de sluis).

Een Sluice bestaat uit twee Doors, en kan informatie over de WaterLevel doorgeven.

Omdat alle klassen netwerking commando's moeten kunnen sturen en acknowledgments ontvangen is er een SluiceNetworkingHandler nodig. D.m.v. deze klassen kunnen alle klassen data uitwisselen met de sluis.

Sluicelogic is de hoofd logica van het programma. Hierin wordt één object Sluice aangemaakt. Verder leest de klasse de user input uit en stuurt zo de sluis aan. De status van de sluis wordt geprint naar de console.

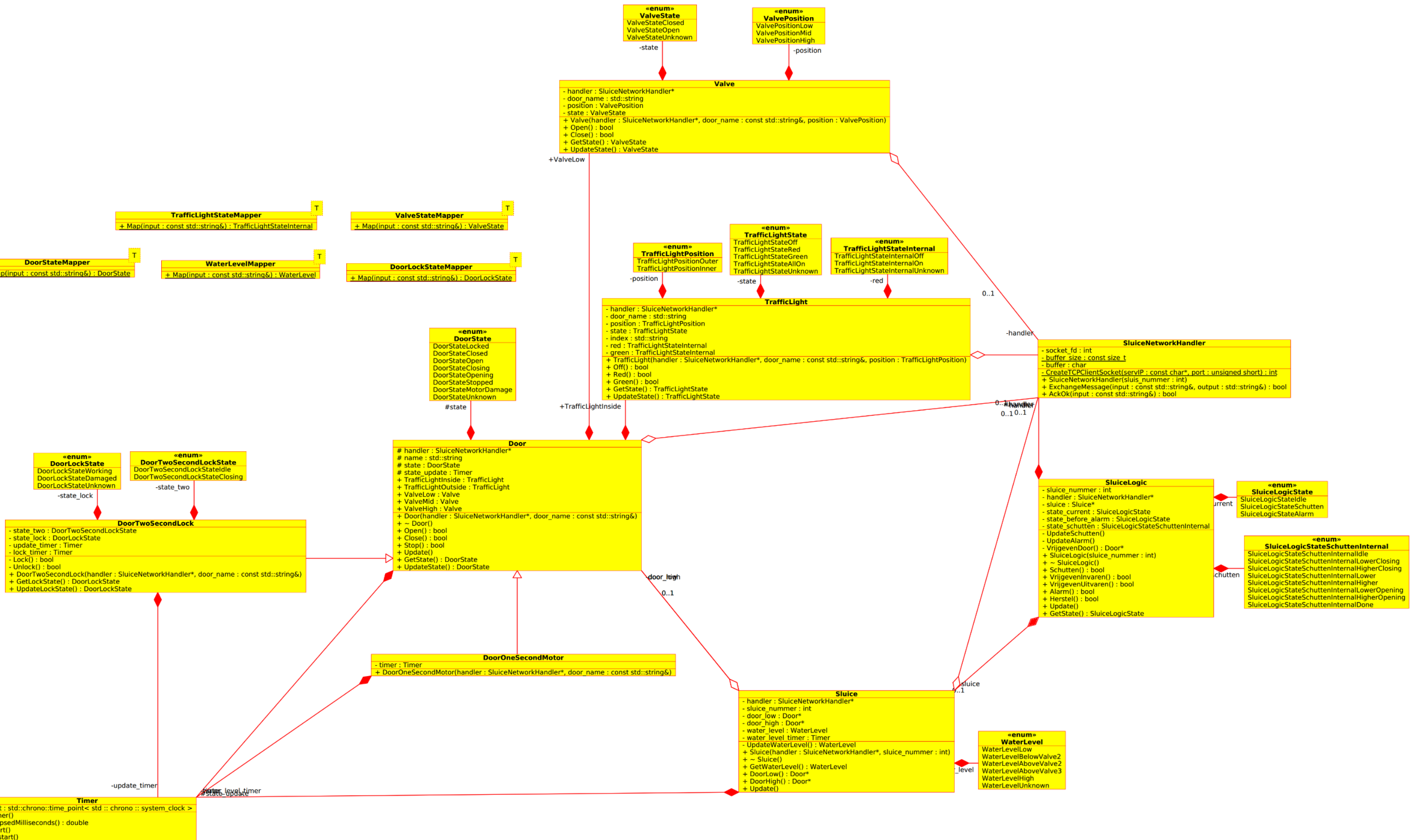
Valve

De klasse Valve staat representatief voor een klep in de deur. Elke Deur heeft drie kleppen. Deze kleppen zitten boven elkaar in de deur. De klasse bestuurt de kleppen door deze te openen en te sluiten.

TrafficLightStateMapper, ValveStateMapper, DoorStateMapper, WaterLevelMapper, DoorLockStateMapper:

Deze klassen worden gebruikt om de enums van de desbetreffende klassen te mappen naar een string om te communiceren.

Het protocol van de Sluis geeft geen mogelijkheid om te identificeren welke acknowledgment bij welke gestuurde actie hoort dus alles moet sequentieel om in een consistente state te kunnen blijven. Indien het protocol bij acknowledgments de acties specificieerde (in chronologische volgorde) zou het op een consistente en robuuste manier mogelijk zijn om een multithreaded of asynchronous oplossing te maken.



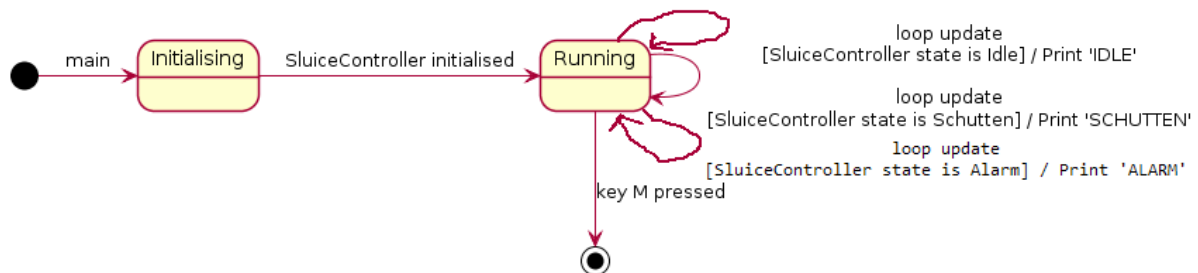
State Diagrammen

Er zijn 3 hoofdstaten:

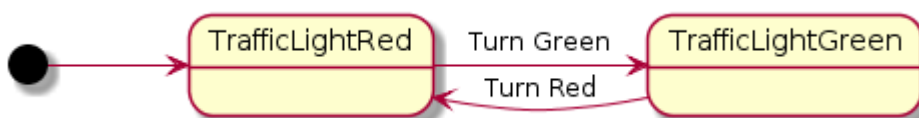
- Idle: er gebeurt niks
- Schutten: er is een schuttingsoperatie bezig (sluiten van deuren daar waar niveau aan beide kanten hetzelfde is, het waterpeil aan gelijk maken met de tegenovergestelde zijde, de deuren openen waar het water nu gelijk staat, vervolgens naar de idle state gaan)
- Alarm: Alles wat er op dit moment gebeurt, stopt. Na herstellen gaat alles weer normaal verder.

Verder zijn er verschillende sub-states voor klassen/procedures die verschillende stappen moeten doorlopen om een geheel proces af te ronden.

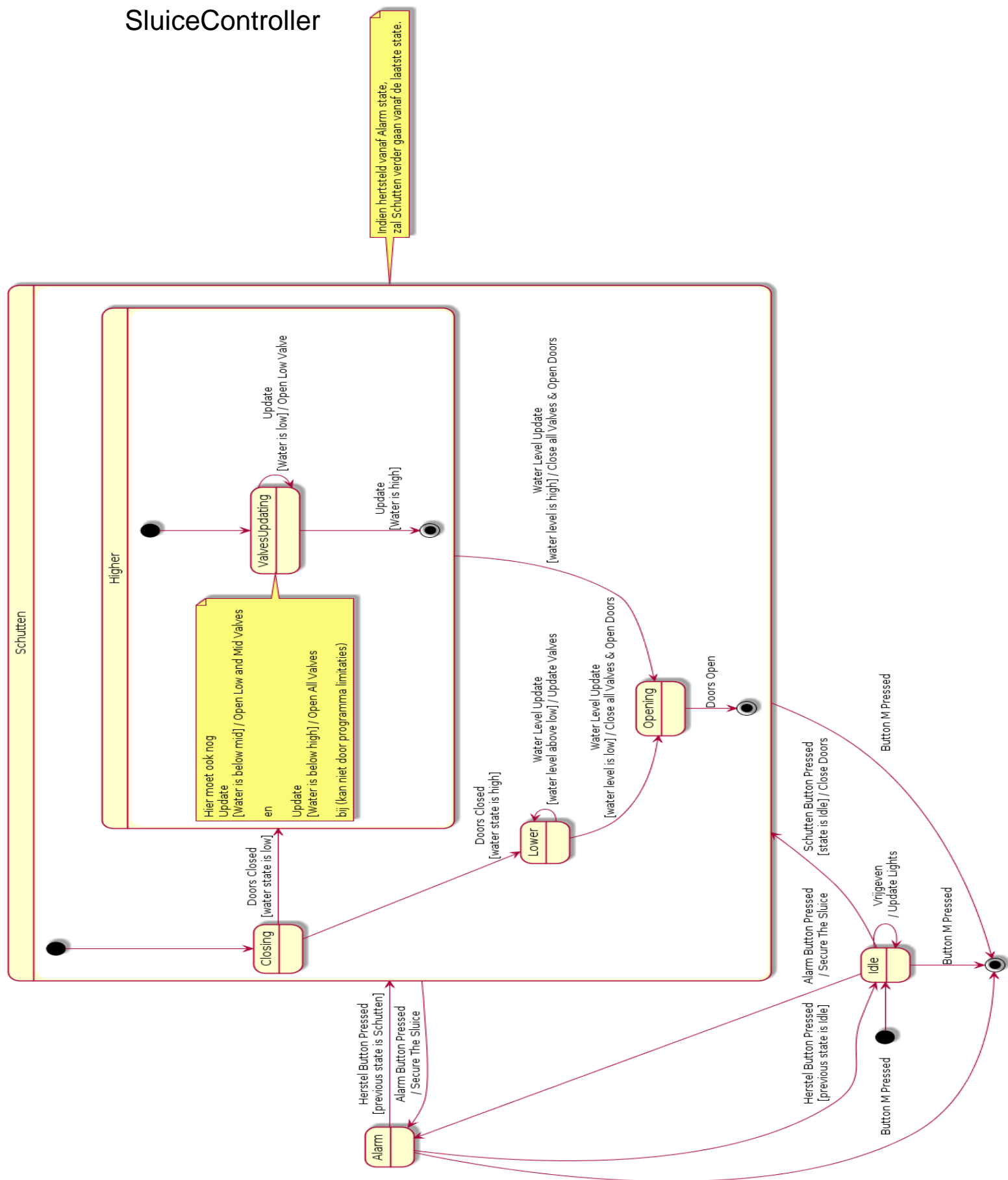
Main programma



Traffic Lights

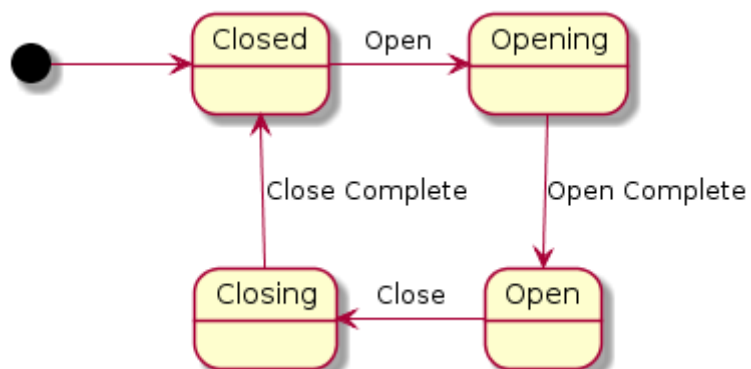


SluiceController

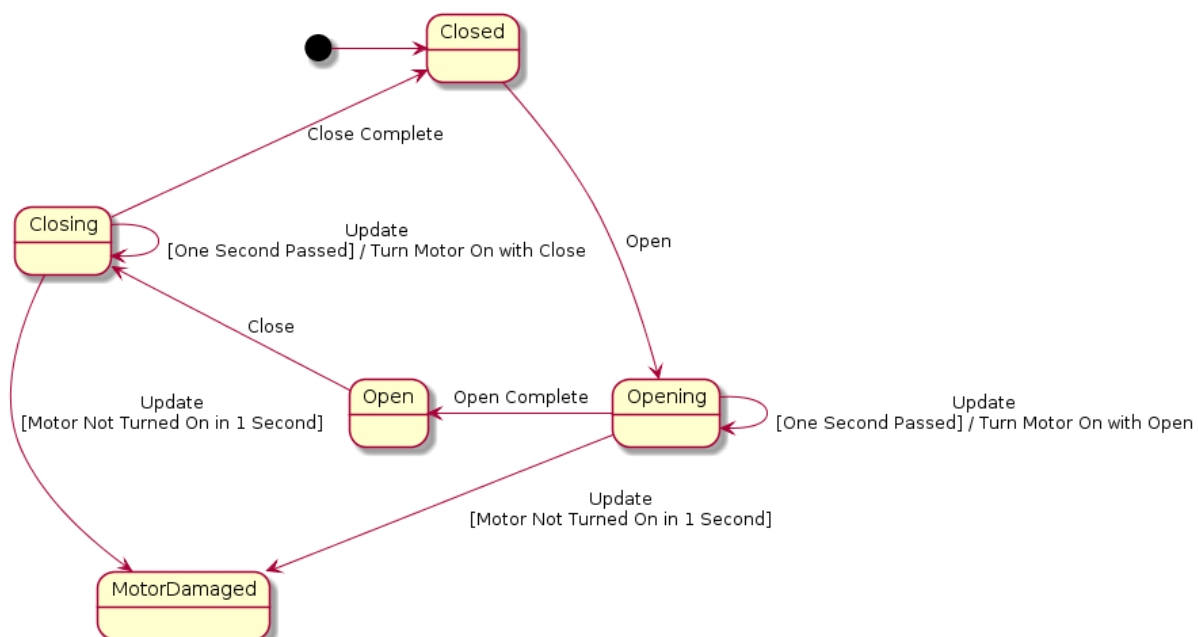


Door

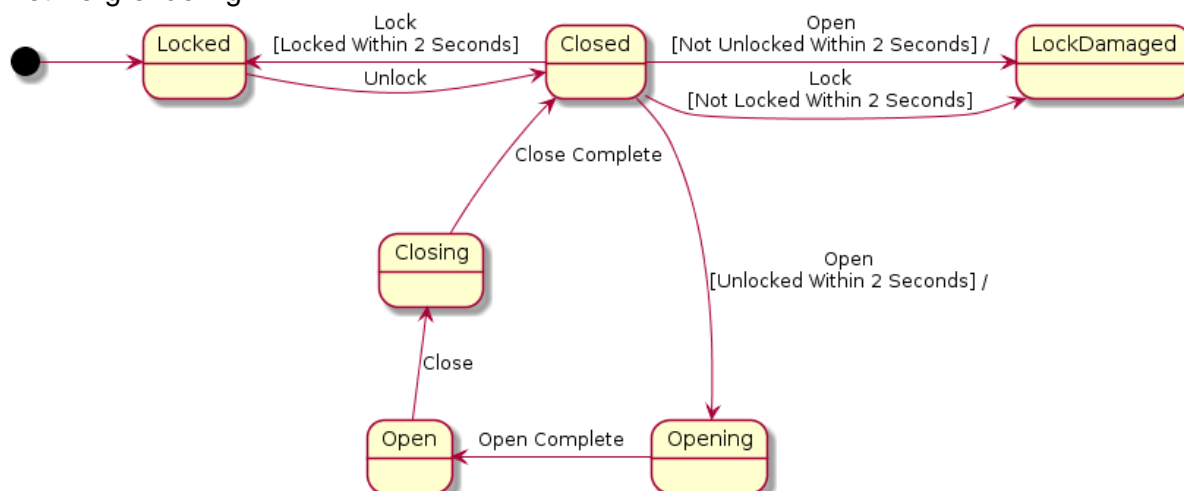
Normaal:



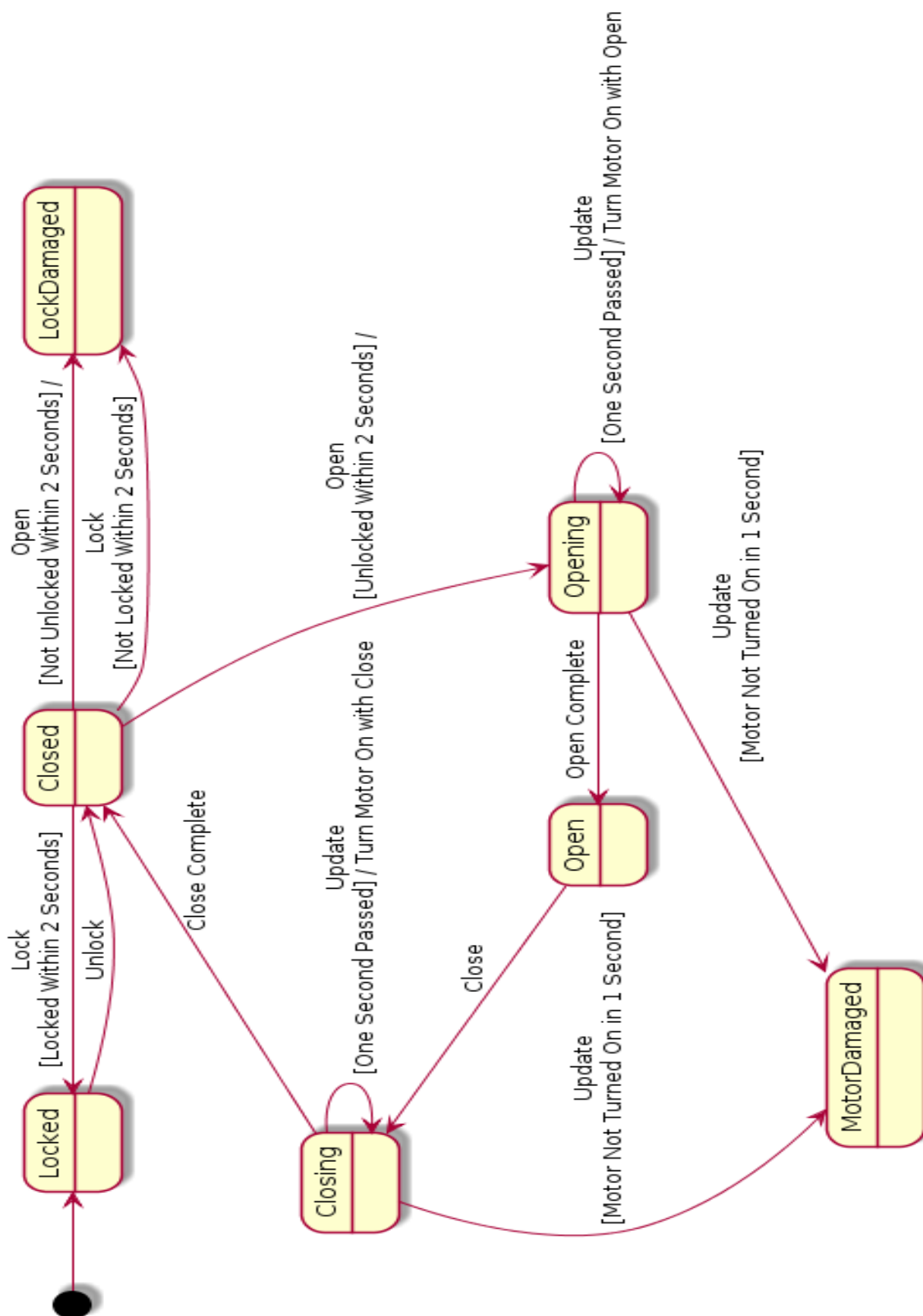
Speciale Motor:



Met Vergrendeling:



Gecombineerd:



Sequence Diagrammen

De volgende klassen hebben een directe interactie met elkaar:

Door ⇔ Timer
Door ⇔ TrafficLight
Door ⇔ SluiceNetworkHandler
Door ⇔ Valve
DoorTwoSecondLock ⇔ Timer
DoorTwoSecondLock ⇔ SluiceNetworkHandler
DoorOneSecondMotor ⇔ Timer
DoorOneSecondMotor ⇔ SluiceNetworkHandler
Sluice ⇔ Timer
Sluice ⇔ Door
Sluice ⇔ SluiceNetworkHandler
SluiceLogic ⇔ Sluice
SluiceLogic ⇔ SluiceNetworkHandler
Valve ⇔ SluiceNetworkHandler
TrafficLight ⇔ SluiceNetworkHandler