

# Werkplan

Software Engineering en Gedistribueerde Applicaties  
Team 8

Chris Bovenschen, Harm Dermois, Alexandra Pizarro, Tamara Ockhuis  
Fredo Tan 6132421  
6104096, 0527963, 6129544, 6060374

June 14, 2011

## Contents

<b>1</b>	<b>Inleiding</b>	<b>1</b>
<b>2</b>	<b>Model</b>	<b>2</b>
<b>3</b>	<b>Deliverables</b>	<b>2</b>
<b>4</b>	<b>Testen</b>	<b>3</b>
<b>5</b>	<b>Modules</b>	<b>3</b>
5.1	Movement . . . . .	3
5.2	Listener . . . . .	3
5.3	Reflex modules . . . . .	3
5.4	Botsing ontwijker . . . . .	4
5.5	Muurzoeker . . . . .	4
5.6	Muurvolger . . . . .	4
5.7	Routeplanner . . . . .	4
5.8	Mapmaker . . . . .	4
<b>6</b>	<b>Planning</b>	<b>4</b>

## 1 Inleiding

Dit is het werkplan van team FACHT voor het vak Software Engineering en Gedistribueerde Applicaties. In dit document is de planning de deliverables, de modules, model en planning te vinden van ons project. De project leider is Harm Dermois. De secretarissen zijn ingedeeld als volgt:

- Week 1: Fredo
- Week 2: Alexandra
- Week 3: Tamara
- Week 4: Chris

Het uiteindelijke doel is een robot te maken die zelf rond kan rijden en een map kan maken. Nadat er een map is gemaakt een pad kan plannen die je van punt a naar b brengt.

## 2 Model

De module listener is verbonden aan de server via een TCP connectie. Deze wordt tevens gebruikt om commandos naar de robot te sturen. De ontvangen comando's worden doorgegeven naar de sensormodulen die kijken of de data daadwerkelijk geldig is en binnen de parameters valt, zodat we de data van de robot kunnen gebruiken. We hebben drie sensormodules waar we informatie vandaan halen: sonar, laser en odometry. Vanuit deze laag gaat de data naar de laag waar de mid-level modules zich bevinden die dienen voor de muurzoeker en muurvolger. De data voor de module kaartenmaker komt rechtstreeks uit de sensormodules. Berekende kaarten uit de kaartenmaker kunnen gebruikt worden voor de routeplanner. De routeplanner geeft zijn berekende pad tenslotte door aan module bewegingen. De botsing ontwijker is een module die botsingen kan detecteren en aan de andere modules meldt wanneer er een botsing zal gaan plaatsvinden. We hebben gekozen voor dit model om het wat robuster te maken. Als de listener niet werkt het hele programma niet, maar als er 1 van de sensormodulen kan het programma nog steeds blijven draaien. Ook hebben we het gedrag van de robot dus opgedeeld in 3 verschillende modules namelijk: muurzoeken, muurvolgen en de routeplanner. Van deze modules kan er maar 1 tegelijk aanstaan.

## 3 Deliverables

Het plan is om aan het eind van het project een robot te leveren die zich kan voortbewegen door middel van berekeningen die gebaseerd zijn op data afkomstig van de sensoren. Vervolgens is het de bedoeling dat de robot diens omgeving in kaart kan brengen met behulp van de informatie van de sensoren. De robot moet in staat zijn om: in een vrij volle ruimte gevuld met diverse obstakels een pad langs een wand te kunnen afleggen. Dus wij leveren de volgende modules:

1. De movements
2. Een listener.
3. Reflex/sensor modules per soort data.
4. Een muurzoeker.
5. Een muurvolger.
6. Een botsing vermijder.
7. Een routplanner.
8. Een mapmaker.
9. Een communicator die tussen modules kan praten.

10. Een werkend geheel die alle modules met elkaar combineert.

Uitleg over de modules te vinden in section 5. Bij de eerste deliverable leveren de volgende modules: De listener, de muurvolger, de muurzoeker, de reflex/sensor en communicator. Deze zijn allemaal standalone en maken hun eigen connectie met de server. De tweede deliverable is het eind product en hebben we een werkend geheel.

## 4 Testen

Alle modules hebben we apart getest. Elke module moet standalone kunnen werken en de verschillende We hebben elke module direct aan de aan de simulator vast gezet en met de echte data de modules getest. Wanneer we denken dat de modules werken gaan we ze verder testen op onconventionele omstandigheden. De pre-condities worden aan het begin van de methode gedefinieerd. Wanneer je in de module komt met waardes die niet voldoet aan de pre-condities worden genegeerd. Al deze modules worden later gemodificeerd met de communicator methode zodat ze met elkaar kunnen praten.

## 5 Modules

Alle geteste modules worden voorzien van een communicator. Deze communicator leest uit waar alle modules zich bevinden. Wanneer de communicator wordt opgestart wordt een thread gestart die connecties kan aannemen. Deze thread start op zijn beurt nog een andere thread op die luistert naar alle inkomende request en handelt deze af. Het preciese protocol voor de communicatie moet nog nader bepaald worden. Onze definitie van een muur zijn alle obstakels die we zien. Wij gebruiken geen camera dus het moeilijk een verschil te maken tussen een muur een blok die in de weg staat.  
precondities:

### 5.1 Movement

Dit is een bibliotheek waarin alle bewegingen in zijn gedefinieerd. Deze kan geïmporteerd worden door alle modules die de robot willen veranderen.

### 5.2 Listener

De listener is de module die direct aan de server vast zit. Deze haalt de informatie binnen van die de robot opstuurt, maar stuurt ook commando's naar de robot. Dit maakt de commando's naar de robot in ieder geval consistent. De data die wordt ontvangen worden direct opgestuurd naar de respectievelijke reflex modules die de data dan verder bewerken. Er moeten misschien nog levels worden toegevoegd voor de urgency van de commando's.

### 5.3 Reflex modules

De reflex modules krijgen de respectievelijke data van de listener. Deze modules zijn een soort opslag plaats van voor de meest recente data die de listener heeft

gekregen. Deze modules kunnen de robot ook stoppen wanneer de data die ze binnen krijgen niet binnen de verwachte parameters zijn. Daarom zijn het de reflex modules. Deze modules kunnen door de hogere level modules aangeroepen worden voor de meest recente data.

## 5.4 Botsing ontwijker

De botsing ontwijker wordt hoort bij de routeplanner module. Deze methode runt alleen wanneer de routeplanner aan het werken is. Deze detecteert dingen die niet op de map zit, dus de botsing ontwijker wordt alleen gebruikt voor dingen die onvoorzien zijn.

## 5.5 Muurzoeker

De muurzoeker is de mode waarin het zich bevindt wanneer de robot de wereld moet verkennen. De robot begint met om zich heen te kijken de kleinste waarde op te zoeken en daar heen te bewegen. Wanneer er een bepaalde drempelwaarde is bereikt gaan we ervan uit dat er een muur is gevonden en gaan we over naar de muurvolger module die dan verder gaat met de gevonden muur te volgen.

## 5.6 Muurvolger

De muurvolger volgt de muur die in de muurzoeker gevonden is. Hier draait het 90 graden zodat de kleinste waarde van de sonar in een van de buitenste sensoren te vinden is. Hierna blijft hij door gaan totdat er geen muur is om te volgen en gaat dan weer over naar de muurzoeker.

## 5.7 Routeplanner

De Routeplanner kan alleen gebruikt worden wanneer er al een map is gemaakt en kan dus alleen een route plannen binnen een bekend domein. Hierin probeert hij een zo kort mogelijk pad te vinden naar de bestemming.

## 5.8 Mapmaker

De mapmaker krijgt een constante stroom van sonar en IR data waarmee hij dan een map gaat maken met waarschijnlijk wat uitbreidingen.

# 6 Planning

Er zijn vier weken beschikbaar voor dit project. In de eerste week van het project wordt de listener als eerst gerealiseerd. Daarnaast wordt de module die de methoden bevat voor de bewegingen geschreven. Wanneer dat gelukt is, maken we in dezelfde week nog een begin aan de sensormodules. De drie modules muurzoeker, muurvolger en botsingsvermijding zijn de volgende componenten die gecomplementeerd moeten worden. Dat willen we graag in de tweede week doen. Wanneer dat voltooid is, maken we de high level componenten aan: de routeplanner en de kaartenmaker. We denken dat deze twee onderdelen vrij gecompliceerd zijn, hier willen we dan ruim de tijd voor nemen.

17 juni is de dag wanneer de eerste deliverable gereed moet zijn. Naar verwachting zal de eerste deliverable de listener, bewegingen en (een deel van) de mid level componenten omvatten. In de derde week vindt het inleveren van de tweede deliverable plaats. Op 23 juni willen we mid level componenten en de high level componenten gereed hebben.

Week	Focus	Deliverables
week 1	listener,bewegingen,sensor en werkplan	listener, beweging, sensoren en schetswerkplan
week 2	echte werkplan en mid-level modules en tests	werkplan en mid-level modules
week 3	mapmaking, routeplannen en tests	mapmaking
week 4	eindverslag, opschonen code testen	eind code en verslag