

Werkplan

Software Engineering en Gedistribueerde Applicaties F.A.C.H.T.

Chris Bovenschen, Harm Dermois,
Alexandra Moraga Pizarro, Tamara Ockhuijsen en Fredo Tan
6104096, 0527963, 6129544, 6060374 , 6132421

June 14, 2011

Contents

1	Inleiding	1
2	Model	2
3	Deliverables	3
4	Testen	3
5	Modules	4
5.1	Movements	4
5.2	Listener	4
5.3	Reflexmodules	4
5.4	Botsingvermijder	4
5.5	Muurzoeker	4
5.6	Muurvolger	5
5.7	Routeplanner	5
5.8	Mapmaker	5
6	Planning	5

1 Inleiding

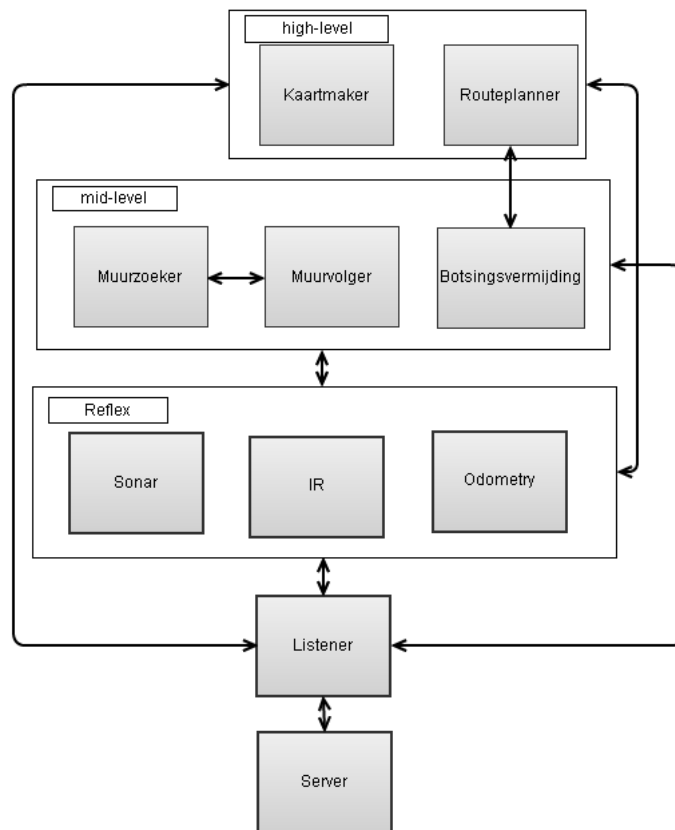
Dit is het werkplan van team F.A.C.H.T. voor het vak Software Engineering en Gedistribueerde Applicaties. In dit document zijn de planning, de deliverables, de modules en het model te vinden van ons project. De voorzitter is Harm Dermois. De secretarissen zijn ingedeeld als volgt:

- Week 1: Fredo Tan
- Week 2: Alexandra Moraga Pizarro

- Week 3: Tamara Ockhuijsen
- Week 4: Chris Bovenschen

Het doel is een robot te maken die zelf rond kan rijden en een map kan maken. Nadat er een map is gemaakt, kan de robot een pad plannen die je van punt A naar B brengt.

2 Model



Alle grijze blokken zijn modules.

De listenermodule is verbonden aan de server via een TCP-verbinding. Deze wordt tevens gebruikt om commando's naar de robot te sturen. De ontvangen sensor data wordt doorgegeven naar de sensormodules, 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 kaartenmakermodule 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 de movementsmodule. De botsingvermijder 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 robuuster te maken. Als de listener het niet doet, werkt het hele programma niet, maar als er één van de sensormodules is uitgevallen, kan het programma nog steeds blijven draaien. Ook hebben we het gedrag van de robot dus opgedeeld in drie verschillende modules, namelijk: muurzoeker, muurvolger en routeplanner. Van deze modules kan er maar 1 tegelijk aanstaan.

3 Deliverables

Aan het eind van het project leveren we een robot 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/sensormodules per soort data.
4. Een muurzoeker.
5. Een muurvolger.
6. Een botsingvermijder.
7. Een routeplanner.
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 is te vinden in sectie 5. Bij de eerste deliverable leveren we de volgende modules: De listener, muurvolger, muurzoeker, reflex/sensor en communicator. Deze staan allemaal op zichzelf en maken hun eigen verbinding met de server. De tweede deliverable is het eindproduct en dan hebben we een werkend geheel.

4 Testen

Alle modules gaan we apart testen. We gaan elke module direct aan de simulator vast zetten en met de directe data de modules testen. Wanneer de modules werken, gaan we ze verder testen op onconventionele omstandigheden. De pre-condities worden aan het begin van de methode gedefinieerd. Wanneer er in de module waarden aankomen die niet voldoen aan de pre-condities, worden die genegeerd. Al deze modules worden later gemodificeerd met de communicator methode, zodat ze met elkaar kunnen praten.

5 Modules

Alle modules worden voorzien van een communicator. Deze communicator leest uit waar alle modules zich bevinden. Wanneer de communicator wordt opgestart, wordt er een thread aangemaakt die verbindingen kan aannemen. Deze thread start op zijn beurt nog een andere thread op die luistert naar alle inkomende requests en handelt deze af. Het precieze protocol voor de communicatie moet nog nader worden bepaald.

Onze definitie van een muur omvat alle obstakels die we zien. Wij gebruiken geen camera, dus het is niet mogelijk om het verschil te zien tussen een muur en een blok die in de weg staat.

Pre-conditions:

5.1 Movements

Dit is een bibliotheek waar 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 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, wordt direct opgestuurd naar de respectievelijke reflexmodules die de data dan verder bewerken. Het is mogelijk om levels toe te voegen voor de urgentie van de commando's.

5.3 Reflexmodules

De reflexmodules krijgen de respectievelijke data van de listener. Deze modules zijn een soort opslagplaats 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 vallen. Deze modules kunnen door de high-level modules aangeroepen worden voor de meest recente data.

5.4 Botsingvermijder

De botsingvermijder hoort bij de routeplannermodule. Deze methode runt alleen wanneer de routeplanner aan het werken is. Deze detecteert dingen die zich niet op de map bevinden. De botsingvermijder wordt dus alleen gebruikt voor dingen die onvoorzien zijn.

5.5 Muurzoeker

De muurzoeker is de mode voor de robot wanneer hij de wereld moet verkennen. De robot begint om zich heen te kijken, de kleinste waarde op te zoeken en zich daarheen 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 muurvolgermodule die dan verder gaat met de gevonden muur te volgen.

5.6 Muurvolger

De muurvolger volgt de muur die door de muurzoeker gevonden is. Bij de gevonden muur draait de robot 90 graden, zodat de kleinste waarde van de sonar in de meest rechtersensor te vinden is. Hierna blijft hij rechtdoor rijden totdat er geen muur meer 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, IR en odometry data waarmee hij een map gaat maken.

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 die is geschreven, maken we in dezelfde week nog een begin aan de sensormodules. De drie modules muurzoeker, muurvolger en botsingsvermijder zijn de volgende componenten die geïmplementeerd 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. De eerste deliverable omvat de listener, bewegingen en (een deel van) de mid-level componenten. In de derde week vindt het inleveren van de tweede deliverable plaats. Op 23 juni hebben we de mid-level componenten en de high-level componenten gereed.

Onze focus en deliverables per week:

Week	Focus	Deliverables
week 1	listener, movements, sensoren en werkplan	listener, movements, sensoren en schetswerkplan
week 2	definitief werkplan, mid-level modules en tests	werkplan en mid-level modules
week 3	mapmaker, routeplanner en tests	mapmaker
week 4	eindverslag, opschonen code en testen	eindcode en verslag