



Drone meshnetwork simulatie

Software Requirements Specification

HAN Arnhem

561399

MWJ.Berentsen@student.han.nl

Versie 1

Alten Nederland B.V.

Docent: J. Visch, MSc

Assessor: ir. C.G.R. van Uffelen

M.W.J. Berentsen

29 april 2019

Inhoudsopgave

1	Inleiding	3
1.1	Algemene beschrijving	3
1.1.1	leeswijzer	3
1.2	Doel van dit document	3
1.3	Actoren en hun eigenschappen	3
1.4	Werkomgeving	3
1.4.1	Arduino UNO	4
1.4.2	NRF24	4
1.4.3	Drone	4
1.4.4	ROS	4
1.4.5	Gazebo	4
1.5	Ontwerp en implementatie beperkingen	5
1.6	Product Functies	5
2	Domein Model	6
3	Use-case omschrijvingen	8
3.1	Use case	8
3.1.1	Fully-dressed use case description	8
3.1.2	System Sequence Diagram (optional)	8
3.1.3	Operation Contracts (optional)	8
3.2	Use case 2 (and so on)	8
3.2.1	Fully-dressed use case description	8
3.2.2	System Sequence Diagram (optional)	8
3.2.3	Operation Contracts (optional)	8
3.3	Use case 2 (and so on)	8
4	Other functional requirements (optional)	9

5	Non-functional Requirements	10
5.1	Performance Efficiency	10
5.2	Security	10
5.3	Reliability (and so on)	10
	Literatuur	11
A	Appendix 1	12

1 — Inleiding

1.1 Algemene beschrijving

Het volgende verslag betreft de Software Requirements Specification voor de afstudeerstage van Maurice Berentsen (hierna: student). Dit document volgt het document: *"Software Requirements Specification Template"* (Van Heesch, 2016) Het beschrijft de hoe de uiteindelijke applicatie eruit zal zien en wat de functionaliteit hiervan zal zijn. Op de manier is het voor de student maar ook betrokken partijen duidelijk wat er gerealiseerd zal worden. Het verduidelijkt de werking van de sub-componenten en de onderlinge relaties.

1.1.1 leeswijzer

Eerst zal een korte beschrijving gegeven worden van het doel van dit document. In het tweede hoofdstuk is als eerste een component diagram te vinden waarin de functionaliteit van de verschillende componenten te vinden is en de manier waarop deze componenten met elkaar communiceren. Vervolgens is de algemene flow van het programma te zien, hierin wordt duidelijk welke stappen er onder water worden genomen als er een bepaalde actie wordt uitgevoerd. In hoofdstuk drie worden alle subsystemen duidelijk uitgewerkt en worden de ontwerpen van deze systemen uitgewerkt.

1.2 Doel van dit document

Het doel is dat de lezer aan het einde van dit document begrijpt hoe de functionaliteit geïmplementeerd zal worden in het op te leveren product. Het document laat zien hoe de flow van het programma loopt en welke classes in de software worden gebruikt. Een lezer van dit document zou met de juiste technische kennis de volledige applicatie moeten kunnen maken zoals het bedacht is door de ontwerper.

1.3 Actoren en hun eigenschappen

In deze paragraaf worden de actoren van het systeem omschreven. Elke actor wordt kort omschreven per sub-paragraaf

1.4 Werkomgeving

Deze paragraaf omschrijft zowel de hardware- als softwareomgeving waarin dit project wordt uitgevoerd.

1.4.1 Arduino UNO

De Arduino UNO is gekozen als ontwikkelbord die gebruikt wordt in het opzetten van de onderlinge communicatie. Deze Arduino is gekozen omdat deze gemakkelijk leent voor het bouwen van prototype. Dit doet de Arduino door zijn aanbod van 14 I/O poorten waarvan er zes bruikbaar zijn als analoge poort en het aanbieden van zowel 3.3 als 5 volt lijnen Hierop kan de [NRF24](#) aangesloten worden waarna er nog genoeg poorten overblijven voor andere hardware zoals sensoren of een internetverbinding.

De aanwezigheid van een 16MHz kristaloscillator geeft de UNO de eigenschap om accuraat genoeg een verschil in tijd te schatten. Dit belangrijk voor het systeem aangezien het beslissingen moet kunnen nemen op basis van een verstreken tijd.

De Arduino voorzien van een ATmega328P die snel genoeg is voor het doorgeven van het communicatieverkeer en het nemen van beslissingen. De ATmega328P heeft 32KB flash geheugen en 1KB EEPROM beschikbaar.

Verder is de Arduino gekozen omdat deze via een USB aansluiting makkelijk geflasht kan worden en gedeeltelijk ondersteuning heeft voor C++

1.4.2 NRF24

De NRF24 is gekozen door zijn prestaties ten opzichte van afstand. De mogelijkheid om een snelheid aan te kunnen bieden van minimaal 250 kbps op een afstand van 500 meter maakt deze module het meest geschikt voor dit project. Daarnaast kan de NRF24 tot zes kanalen tegelijk open houden die kunnen schakelen tussen zenden en ontvangen. De NRF24 is in staat om per payload tot 32 bytes te versturen. Tenslotte is de NRF24 een transceiver die werkt op een voltage van 3.3 volt waarbij de I/O pinnen 5 volt tolerant zijn wat het compatibel maakt met de [Arduino UNO](#).

1.4.3 Drone

Hoewel in dit project drones een essentieel onderdeel zijn wordt er niet gesproken over een specifiek merk of type drone. De reden hiervoor is omdat er geen focus ligt op een specifieke drone en de student ook niet gecertificeerd is om te vliegen met zakelijke drones. Er zal dus gewerkt worden met gesimuleerde drones. Deze hebben een interface die in staat is om een drone te laten vliegen naar een specifiek coördinaat en kan de huidige locatie aangeven.

1.4.4 ROS

ROS is middleware software die gebruikt wordt voor de aansturing en simulatie van robotica. In het geval van dit project wordt ROS gebruikt voor de communicatie naar de [Drone](#) toe, maar ook voor het simuleren van de [NRF24](#) communicatie tussen de drones.

1.4.5 Gazebo

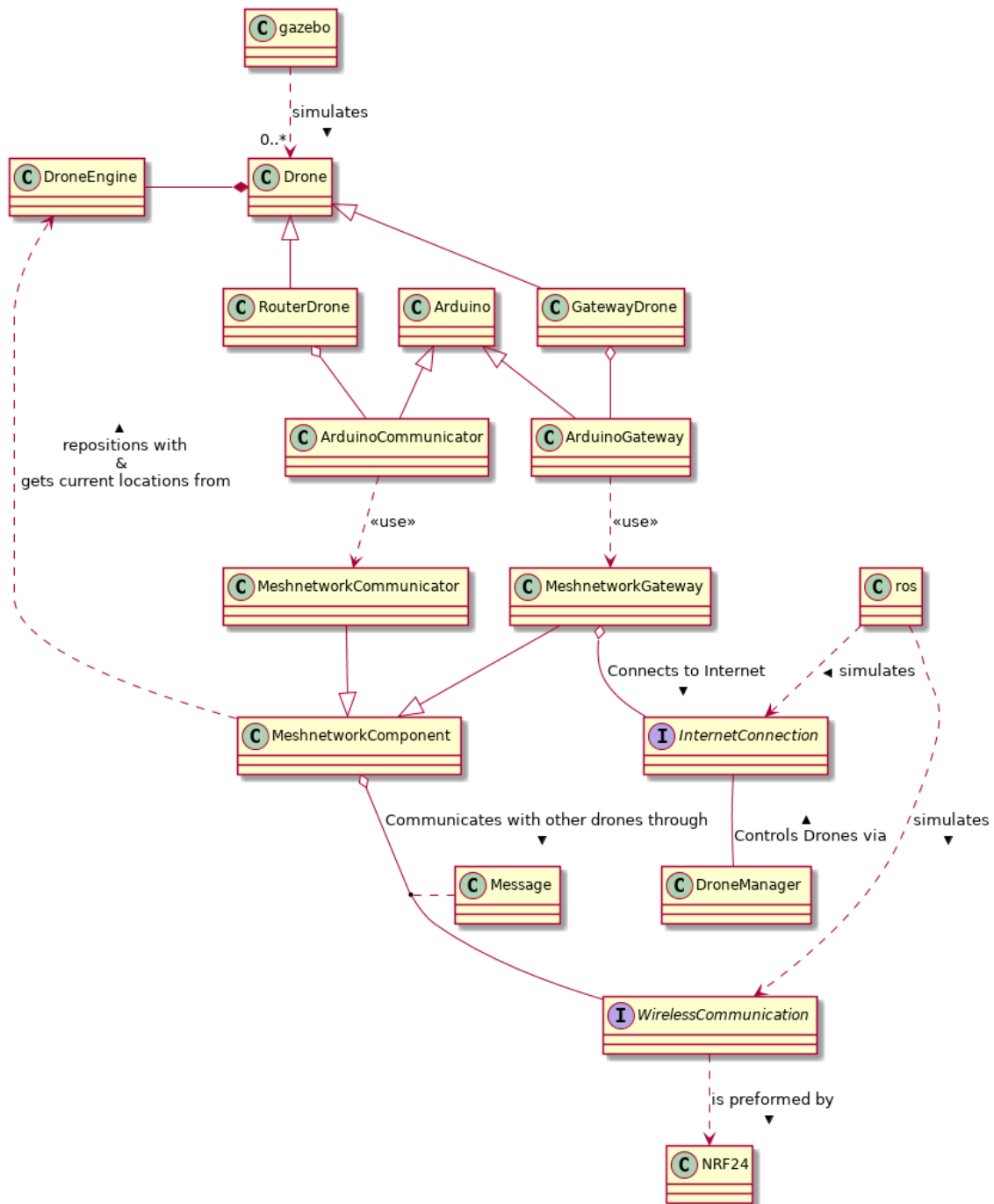
[Gazebo](#) is een opensource robot simulatie framework bijzonder geschikt voor het simuleren van robotica in outdoor omgevingen door de uitgebreide Physics Engine Support. In dit project wordt momenteel geen gebruik gemaakt van de physics engine maar doordat

de onderdelen als virtuele onderdelen beschikbaar zijn in gazebo kunnen ze zo aangesloten worden op een beter gesimuleerde drone. Deze mogelijkheid was daarom ook een hoofdreden om Gazebo te gebruiken.

1.5 Ontwerp en implementatie beperkingen

1.6 Product Functies

2 — Domein Model



Figuur 2.1: Domeinmodel

3 — Use-case omschrijvingen

3.1 Use case

3.1.1 Fully-dressed use case description

3.1.2 System Sequence Diagram (optional)

3.1.3 Operation Contracts (optional)

3.2 Use case 2 (and so on)

3.2.1 Fully-dressed use case description

3.2.2 System Sequence Diagram (optional)

3.2.3 Operation Contracts (optional)

3.3 Use case 2 (and so on)

4 — Other functional requirements (optional)

5 — Non-functional Requirements

5.1 Performance Efficiency

5.2 Security

5.3 Reliability (and so on)

Literatuur

Van Heesch, U. (2016, 21 september). *Software requirements specification template*.

A — Appendix 1