



Drone meshnetwerk simulatie

Software Design Document

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

1 mei 2019

Inhoudsopgave

1	Inleiding	2
1.0.1	leeswijzer	2
1.1	Begrippenlijst	2
2	Architectural Overview	3
2.1	Componenten	5
2.2	Interfaces	8
3	Detailed Design Description	9
3.1	Deployment Diagram	9
3.1.1	Design Decisions related to deployment	9
3.2	Design Sub-Systeem Drone Simulatie	9
3.2.1	Design Class Diagram	9
3.2.2	Sequence Diagrams	10
3.2.3	Activity and State Diagrams	10
3.2.4	Design decisions made for the sub-system	10
3.3	Design Sub-Systeem Meshnetwerk	10
3.3.1	Design Class Diagram	10
3.3.2	Sequence Diagrams	10
3.3.3	Activity and State Diagrams	10
3.3.4	Design decisions made for the sub-system	10
3.4	Design Sub-System Wireless Simulatie	10
3.4.1	Design Class Diagram	10
3.4.2	Sequence Diagrams	10
3.4.3	Activity and State Diagrams	10
3.4.4	Design decisions made for the sub-system	10
	Literatuur	11
A	Appendix 1	12

1 — Inleiding

Het volgende verslag betreft de Software Requirements Specification voor de afstudeerstage van Maurice Berentsen (hierna: student). Dit document volgt het document: *"Software Design Description 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.

1.0.1 leeswijzer

Eerst zal een korte beschrijving gegeven worden van het doel van dit document. Daarna wordt er door middel van een tabel een begrippenlijst toegelicht. 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.

1.1 Begrippenlijst

Term	Omschrijving
term	Omschrijving

Tabel 1.1: Begrippenlijst

2 — Architectural Overview

In het component diagram [Figuur 2.1](#) op de volgende pagina is te zien dat bepaalde componenten voorzien zijn van een andere kleur. De groene kleur betekend dat het component voorzien is door gazebo. De rode en blauw kleur zijn gegeneerd door ROS waarbij de rode een rosservice zijn en de blauwe een rostopic. Op het hoogste niveau is het diagram te verdelen in vier groepen.

Communication is het component die de beslissingen neemt voor de communicatie. Het heeft de intelligentie om het netwerk in kaart te brengen en te routeren. Daarnaast is het in staat om een beslissing te nemen om een verzoek te sturen om de drone te verplaatsen. Tenslotte bevat het meerdere berichten die gebruikt om het netwerk te kunnen onderhouden en op te bouwen.

Drone is op dit moment een leeg component waar alleen een high level interface aanwezig is die de mogelijkheid biedt om de huidige locatie van een drone terug te geven of een nieuwe locatie als doel te geven aan de drone.

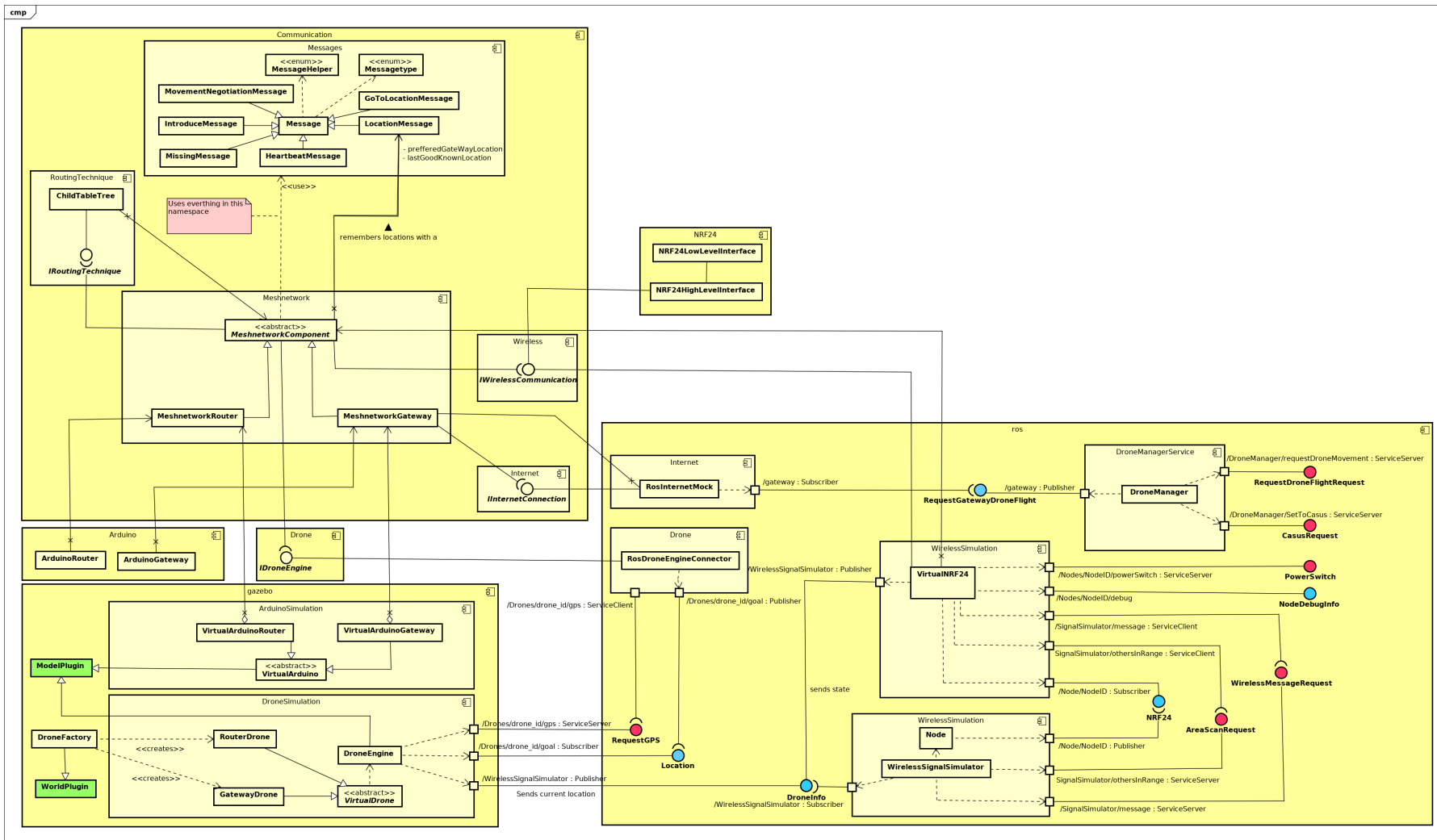
Gazebo is de plek waar de virtualisatie van de drone plaats vindt. Er kunnen virtuele router of gateway drones gemaakt worden die voorzien zijn van een virtuele arduino ingeladen met de juiste software en een virtuele drone motor. Deze worden als model plug-in gekoppeld aan de drone in een gesimuleerde wereld waar Gazebo vervolgens physics op de drones kan uitvoeren.

ROS is het component waar de simulatie van de communicatie plaats vindt. Er is een virtuele NRF24 om te communiceren met andere nodes in het netwerk. De aanwezige WirelessSignalSimulator zorgt dat dit realistisch gebeurt door alleen communicatie uit te voeren als dit volgens de voorwaarden mag. Dit bepaald de simulator op basis van informatie die het krijgt van Gazebo en de NRF. Tenslotte is er een DroneManager aanwezig die een interface aanbiedt via ros voor de ontwikkelaar om verbinding te maken met de gateways uit het netwerk.

Arduino is het component die zich onderscheid in twee rollen, een Arduino kan de rol van gateway hebben of de rol van router in de het meshnetwerk.

NRF24 is het gebruikte component om draadloze communicatie mogelijk te maken. Het is voorzien van een driver in de vorm van een low level interface en bied zich aan via een high level interface voor communicatie.

Na het component diagram zal elke subcomponent kort toegelicht worden



Figur 2.1: package diagram drone meshnetwork

2.1 Componenten

Internet Dit component bevat alleen een high level interface om verbinding te maken en te verbreken met het internet

Meshnetwork Dit component bevat de basis componenten voor het opbouwen van een meshnetwork.

MeshnetworkComponent Een MeshnetworkComponent is de basis van elk component in het mesh netwerk. Het vereist routingstechniek via de [IRoutingTechnique](#), een aansluiting naar een drone via [IDroneEngine](#) en een vorm van draadloze communicatie via [IWirelessCommunication](#). Hij maakt gebruik van het component [Messages](#) om te communiceren met andere MeshnetworkComponenten.

MeshnetworkRouter Een meshnetwork router is een [MeshnetworkComponent](#) die in staat is verbinding met andere nodes op te bouwen. Zijn doel is om altijd verbinding te hebben met een [MeshnetworkGateway](#). Als hij dit te lang niet heeft kan hij zich verplaatsen door aanspraak te maken op de DroneEngine. Wanneer hij zich in een groep van andere Routers bevindt zal hij eerst onderling onderhandelen wie er zich moet verplaatsen.

MeshnetworkGateway De gateway is een [MeshnetworkComponent](#) die in staat is een verbinding op te bouwen naar een punt buiten het meshnetwork. In de huidige situatie kan dit via een internetverbinding die loopt via de interface [IInternetConnection](#).

Messages Dit component betreft een verzameling van bericht samenstellingen die gebruikt worden voor de communicatie van het meshnetwork.

Message Een Message is de basis van elke bericht en bevat tenminste de volgende informatie: Maker, zender, berichttype, ontvanger, geadresseerde.

GoToLocationMessage Dit bericht bevat een locatie waar een drone zich naartoe moet verplaatsen

HeartbeatMessage Dit bericht wordt gebruikt om de verbinding met anderen te onderhouden. Dit bericht maakt een hop per keer dat deze doorgestuurd wordt

IntroduceMessage Een introductie bericht wordt gebruikt door een node om schik voor te stellen aan alle andere nodes die dichtbij zijn.

LocationMessage Dit bericht wordt gebruikt om de huidige locatie van een node door te sturen naar een ander.

MissingMessage Zodra een [MeshnetworkComponent](#) de verbinding verliest met een ander gebruikt hij dit bericht om andere daarover te informeren

MovementNegotiationMessage Om onderling te onderhandelen tussen de nodes wie er actie moet ondernemen wordt dit bericht gebruikt.

RoutingTechnique Dit component voorziet het meshnetwerk van een routing techniek wat dus inhoudt dat dit het component is die de communicatieroutes opbouwt naar andere punten in het netwerk.

ChildTableTree Deze techniek is een hybride meshnetwerk routing techniek waarbij een node de burens die deze heeft ziet als een kind. Als een node een nieuw kind heeft vertelt hij dit aan zijn burens. Omdat een node door zijn burens ook gezien wordt als kind registreren deze zijn nieuwe kind dus als kleinkind. Hierdoor hoeft een node alleen maar te zoeken aan wie welk kind hij een bericht hoeft door te geven. De complexiteit van de routingtechniek neemt per stap in het netwerk af bij nodes met meerdere kinderen. Als een drone zich verplaatst heeft wist deze het de opgebouwde geheugen van kinderen en kleinkinderen.

Wireless Dit component voorziet een [MeshnetworkComponent](#) van een high level interface voor draadloze communicatie. Als er een draadloos component wordt aangesloten moet deze hierop aangesloten worden.

Drone Dit component representeert een drone. Omdat er op dit moment geen fysieke drone aanwezig is bezit deze component alleen een high level interface. Deze interface is geschikt voor het ophalen van de huidige locatie en het ontvangen van een doellocatie

ArduinoRouter Dit component is een Arduino die voorzien is van een sketch om zich te gedragen als een router in het meshnetwerk

ArduinoGateway Dit component is een Arduino die voorzien is van een sketch om zich te gedragen als een gateway in het meshnetwerk

ArduinoSimulation In de dit component wordt een arduino gesimuleerd, deze simulatie is puur functioneel en betreft alleen een gelijk gedrag met betrekking tot de Arduino sketch. Het heeft geen effect op de clock snelheid deze is gelijk aan de computer waarop de software draait.

VirtualArduino Dit abstracte component verplicht elke virtuele Arduino tot het aanmaken van een setup en loop functie zoals dit ook is in Arduino sketches. Omdat deze Arduino altijd wordt toegevoegd aan een ander model erft dit component over van modelplugin.

VirtualArduinoRouter Deze [VirtualArduino](#) is voorzien van een sketch die de arduino zich laat gedragen als in router in het meshnetwerk. Dit component representeert het component [ArduinoRouter](#) in gazebo.

VirtualArduinoGateway Deze [VirtualArduino](#) is voorzien van een sketch die de arduino zich laat gedragen als in gateway in het meshnetwerk. Dit component representeert het component [ArduinoGateway](#) in gazebo.

DroneSimulation Dit component is verantwoordelijk voor het simuleren van drones.

DroneEngine Elke [VirtualDrone](#) is voorzien deze drone engine. Dit component maakt het mogelijk voor een drone om zich in een rechte lijn door lucht zich te verplaatsen waarbij het opstijgen en de landing verticaal wordt uitgevoerd. Daarnaast biedt dit component via een rosservice het deel van de interface [IDroneEngine](#) aan om de huidige locatie op te vragen. Via een rostopic kan er een doel gestuurd gestuurd worden om het andere deel van de net genoemde interface te voorzien. Tenslotte stuurt dit component bij elke verplaatsing de huidige locatie door naar de [WirelessSignalSimulator](#).

VirtualDrone Een virtuele drone is een abstract component die alle variabelen bevat die nodig zijn om een drone in de wereld te injecteren. Deze variabelen betreffen een drone id, locatie en verwijzing naar de gazebo wereld.

GatewayDrone Deze drone is een [VirtualDrone](#) die in de sdf omschrijving wordt voorzien van een [VirtualArduinoGateway](#) plugin.

RouterDrone Deze drone is een [VirtualDrone](#) die in de sdf omschrijving wordt voorzien van een [VirtualArduinoRouter](#) plugin.

DroneFactory Dit component is verantwoordelijk voor het produceren van drones. Het is een WorldPlugin die aan de hand van meegegeven parameters gateway en router drones aanmaakt.

RosDroneEngineConnector Dit component is een adapter tussen de [DroneEngine](#) en de interface [IDroneEngine](#). Die het mogelijk maakt om ros te gebruiken om aanspraak te maken op de virtuele drone.

DroneManagerService Dit component is de toegangspoort voor de ontwikkelaar tot de [MeshnetworkGateway](#). Op dit moment is het component alleen geschikt om verplaatsingverzoeken te versturen via de gateway naar de Drones. Hiervoor biedt het twee interfaces in de vorm van rosservices aan waarbij er een voor locaties is en de ander voor casussen.

DroneManager Dit subcomponent realiseert de rosservices en publiceert verplaatsingverzoeken naar de gateways.

RosInternetMock Deze internetmock laat de [DroneManager](#) zich voordoen als internetpunt zodat er geen daadwerkelijke TCP/IP implementatie hoeft worden gemaakt

WirelessSimulation Het WirelessSimulation component is verantwoordelijk voor het simuleren van de NRF24 en het draadloze signaal hiervan.

WirelessSignalSimulator Deze simulator bepaald of twee nodes met elkaar mogen communiceren. Het doet dit op basis van informatie die het continue ontvangt via het drone informatie topic.

Node Dit component wordt alleen gebruikt door de [WirelessSignalSimulator](#) om te registreren welke Nodes bestaan, waar ze zijn en naar welk topic ze luisteren.

VirtualNRF24 Dit component is de virtuele versie van de NRF24, het is in staat om payloads te verwerken van 32 byte. Deze kan de NRF24 naar een direct adres versturen of zenden naar alle NRF24 nodes binnen bereik.

2.2 Interfaces

IIInternetConnection Deze high level interface wordt gebruikt voor het leggen van een verbinding naar een extern punt buiten het netwerk. Omdat de gateway zich alleen hoeft te verbinden met een punt bestaat deze interface alleen uit een connect() en een disconnect().

IWirelessCommunication In deze interface wordt de functionaliteit gesteld waar een draadloos communicatiemiddel aan moet voldoen. Het bevat functies om een antenne te starten en te stoppen. Het versturen van een bericht naar een specifiek punt of het zenden naar elke punt in de buurt. Er is een functie aanwezig om wanneer mogelijk debugging te gebruiken. Tenslotte moet een aansluitend component met deze interface moeten kunnen teruggeven of deze aan of uit staat.

NRF24HighLevelInterface Deze high level interface is de aangeboden interface van de NRF24 en voldoet aan de interface [IWirelessCommunication](#). Het praat direct met de driver van de NRF24.

NRF24LowLevelInterface Deze low level interface is de driver van de NRF24

IDroneEngine Een drone engine interface representeert de aansluiting met een drone. Hierin moet het mogelijk zijn om een doel coördinaat te sturen naar de drone om zich naartoe te verplaatsen. Daarnaast moet het mogelijk zijn voor een drone om de huidige positie terug te geven.

IRoutingTechnique Deze interface bevat functies voor het uitvoeren van een routingstechniek. Het heeft functies voor het starten en onderhouden van het netwerk. Daarnaast zijn er functies hoe gereageerd moet worden op het vinden en verliezen van andere netwerkpunten. Er zijn functies aanwezig om een adres op te halen waar naartoe gezonden moet worden om een punt te bereiken. Ook kan er opgehaald worden welke punten zijn

aangesloten, hoeveel punten dit zijn en hoeveel directe aansluitingen er zijn. Tenslotte is er functie aanwezig die aangeroepen wordt zodra een drone zich verplaatst zodat hier adequaat op gereageerd kan worden.

3 — Detailed Design Description

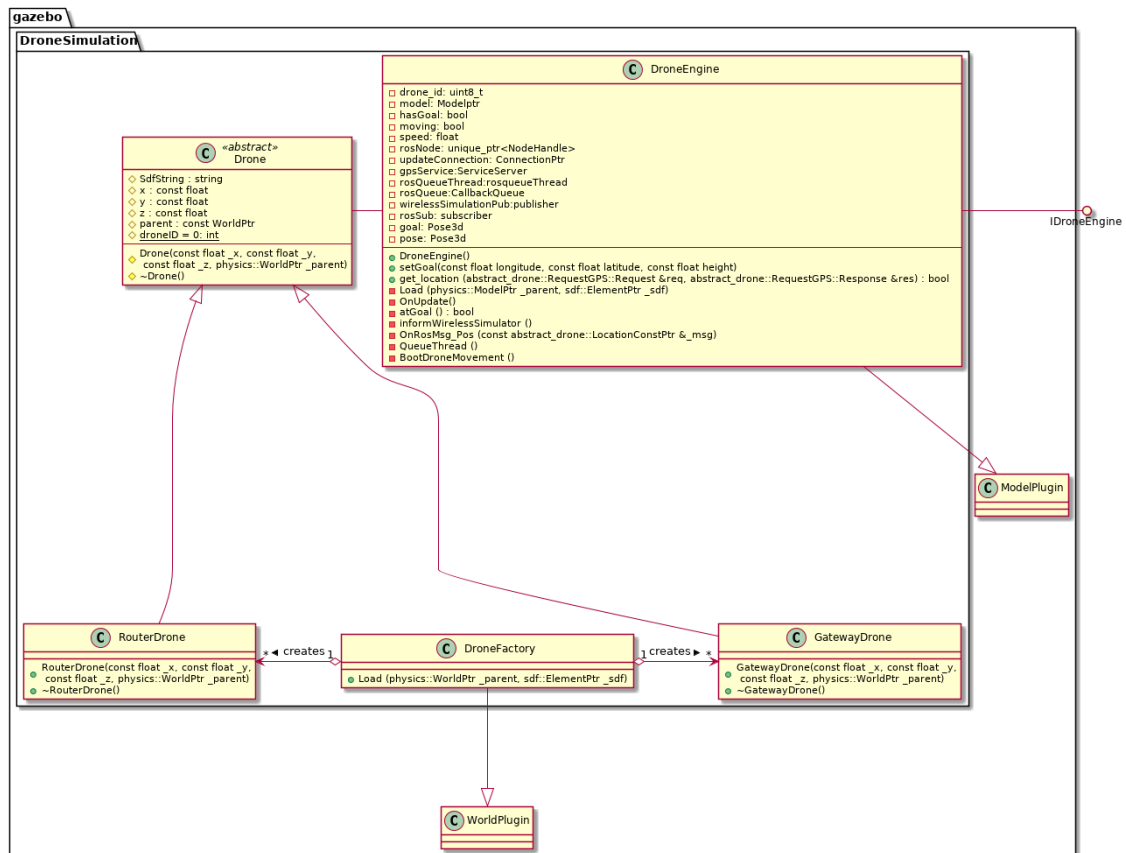
3.1 Deployment Diagram

De volgende paragraaf behandelt het deployment diagram

3.1.1 Design Decisions related to deployment

3.2 Design Sub-System Drone Simulatie

3.2.1 Design Class Diagram



Figuur 3.1: class diagram Drone Simulatie

3.2.2 Sequence Diagrams

3.2.3 Activity and State Diagrams

3.2.4 Design decisions made for the sub-system

3.3 Design Sub-System Meshnetwork

3.3.1 Design Class Diagram

3.3.2 Sequence Diagrams

3.3.3 Activity and State Diagrams

3.3.4 Design decisions made for the sub-system

3.4 Design Sub-System Wireless Simulatie

3.4.1 Design Class Diagram

3.4.2 Sequence Diagrams

3.4.3 Activity and State Diagrams

3.4.4 Design decisions made for the sub-system

Literatuur

Van Heesch, U. (2016, 23 september). *Software design description template*.

A — Appendix 1