



Software Requirement Specification

studentnummer: 1527782

opleiding: HBO-ICT / Afstuderen

profiel: ESD

docenten: Joost Kraaijeveld, Chris van Uffelen

bedrijfsbegeleider: Jeroen Veen

versie: 2.1

Agit Tunç

12 maart 2025

INHOUDSOPGAVE

1	Revisie Historie	3
2	Introductie	4
2.1	Algemene beschrijving	4
2.2	Gebruikersklassen en kenmerken	4
2.3	Operationele omgeving	4
2.4	Ontwerp- en implementatiebeperkingen	5
2.5	Product Functies	5
3	Domeinmodel	7
4	Use-case Beschrijvingen	9
4.1	Use case: Aanzetten robot	9
4.1.1	Fully-dressed use case beschrijving	9
4.2	Use case: Detecteren van aanwezigheid	9
4.2.1	Fully-dressed use case beschrijving	9
4.3	Use case: Volg gezicht	10
4.3.1	Fully-dressed use case beschrijving	10
4.4	Use case: Begroet gebruiker bij naderen	11
4.4.1	Fully-dressed use case beschrijving	11
4.5	Use case: Afscheid nemen bij verlaten	11
4.5.1	Fully-dressed use case beschrijving	11
5	Requirements	13
5.1	Overzicht	13
5.2	Categorisatie van requirements	13
5.3	Overige Functionele requirements (F)	13
5.4	Usability requirements (U)	14
5.5	Reliability requirements (R)	14
5.6	Performance requirements (P)	14
5.7	Supportability requirements (S)	14

1 REVISIE HISTORIE

Versie	Datum	Aanpassing	Auteur
0.1	2024-09-25	Initiele versie WORD	Agit
1.0	2024-09-26	Domeinmodel (hoofdstuk 4) use case nr 1 + 2 uitgewerkt en de requirements hiervoor. Vision systeem. Overige requirements m.b.t. andere subsystemen alleen de must requirements beschreven. WORD	Agit
2.0	2024-11-22	Scenario beschrijvingen WORD	Agit
2.1	2025-03-10	Nieuwe SRS herschreven in latex. We werken nu alleen met radar; camera; LCD-schermen en speaker.	Agit

2 INTRODUCTIE

2.1 ALGEMENE BESCHRIJVING

Dit document beschrijft de softwarevereisten voor een functioneel prototype van een sociale buddy robot. De aanleiding, achtergrond en doelstelling van het project blijven ongewijzigd en zijn te vinden in het Plan van Aanpak. Echter, de afgesproken functionaliteiten binnen de opdracht zijn aangepast en dus ook welke hardware, sensoren en actuatoren gebruikt worden voor de robot in dit project. Gedurende het document wordt het prototype ook gerefereerd als “BuddyBot”. In het plan van aanpak werd het prototype “Open Haru” genoemd.

Kort samengevat moet de software voor BuddyBot ervoor zorgen dat de robot enkele interacties met een gebruiker kan uitvoeren. In deze uitvoering van het project richt BuddyBot zich op het aantrekken van de aandacht van een gebruiker door middel van visuele en auditieve output. Daarnaast is het gewenst dat software modulair en opensource wordt gebouwd, zodat toekomstige ESE studenten hierop voort kunnen bouwen.

2.2 GEBRUIKERSKLASSEN EN KENMERKEN

De gebruikers van de software zijn in feite twee groepen:

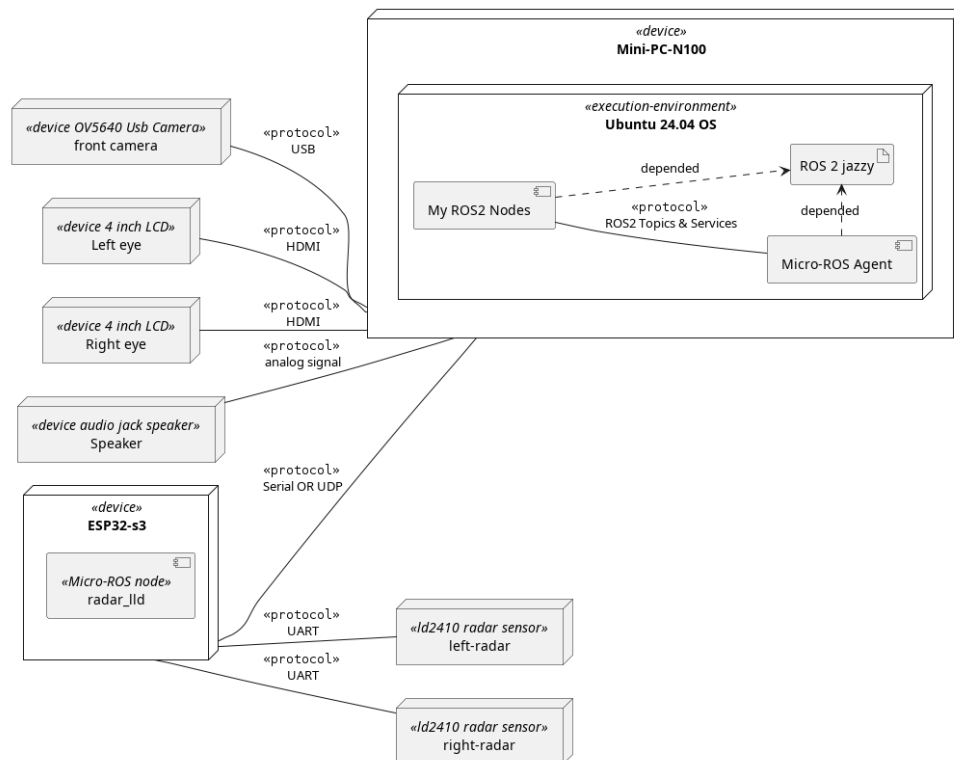
- **Gebruikers in een zorgomgeving:** De robot fungeert als een sociaal hulpmiddel voor bewoners. In het document refereren we naar deze actor als “bewoner”.
- **ESE studenten & Afstudeerders:** De software wordt modulair gebouwd en open source beschikbaar gesteld. De modulariteit van de software wordt gerealiseerd door gebruik te maken van het “algemene robotmodel” architectuur i.c.m. ROS2.

2.3 OPERATIONELE OMGEVING

De geschreven software zal een gedistribueerde applicatie zijn verdeeld over verschillende hardware componenten gebruik makend van diverse sensoren en actuatoren. In het onderstaande lijst is een opsomming hiervan te zien.

- **Besturingssysteem:** Ubuntu 24.04 (voor op de mini-pc)
- **Hardware:**
 - Intel N100 mini-pc
 - ESP32-S3 met micro-ROS voor radar sensoren
 - LCD-schermen voor oogvisualisatie
 - Camera voor gezichtsdetectie
 - Speaker voor audio-uitvoer
- **Communicatie:**
 - USB-verbinding tussen ESP32-S3 en mini-pc (Wi-Fi optioneel)
 - HDMI-verbinding voor de LCD-schermen en mini-pc

In **Figuur 2.1** is een deployment diagram weergegeven om duidelijk de relatie tussen de hardware en software componenten van BuddyBot visueel te weergeven.



Figuur 2.1: Deployment diagram BuddyBot.

2.4 ONTWERP- EN IMPLEMENTATIEBEPERKINGEN

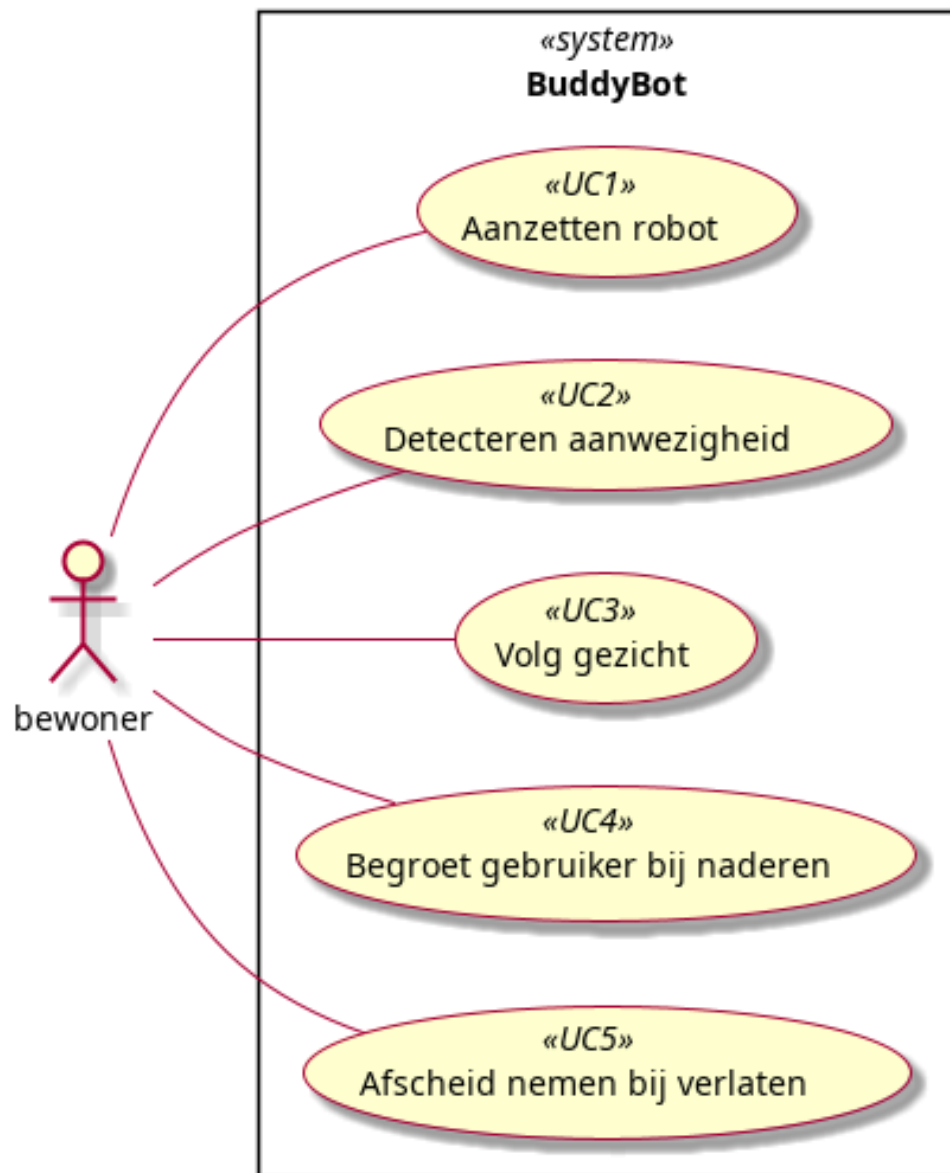
Voor dit project wordt er code geschreven in C++ in combinatie met het ROS2 (jazzy jalisco) framework. Code die geschreven wordt voor de esp32 zal een mix zijn van C++ en C code, omdat er gebruik gemaakt zal worden van micro-ros framework. De micro-ros framework en de API hiervan is geschreven/beschikbaar in C.

2.5 PRODUCT FUNCTIES

In dit paragraaf wordt op hoog niveau weergegeven wat de robot functioneel moet kunnen doen. In de onderstaande tabel wordt er verteld wat de robot moet kunnen doen en welke use case hierbij hoort.

Wat moet de robot kunnen	Use Case Nummer
Zich zelf in operationele toestand zetten wanneer het aangezet wordt.	UC1
Een gebruikers aanwezigheid kunnen detecteren.	UC2
Een gebruiker te volgen met zijn ogen.	UC3
Een gebruiker te begroeten wanneer die recht voor hem staat stil staat.	UC4
Afscheid te nemen wanneer er niemand zich bevindt binnen een straal van 2 meter.	UC5

Tabel 2.1: Overzicht van de product functionaliteiten en bijbehorende use cases.

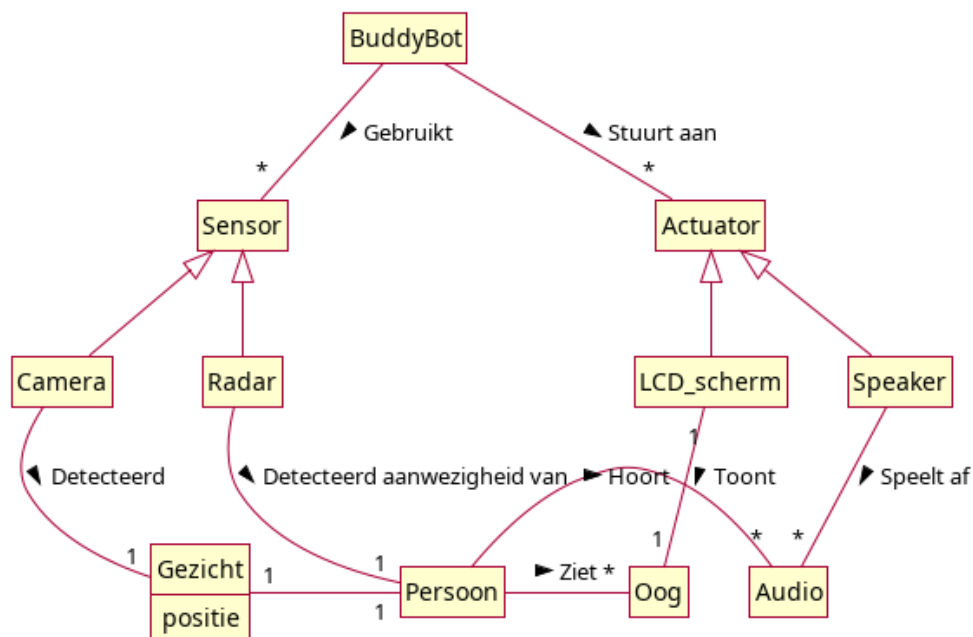


Figuur 2.2: Usecase diagram BuddyBot.

3 DOMEINMODEL

Het domeinmodel biedt een abstracte representatie van de belangrijkste concepten en hun onderlinge relaties binnen het BuddyBot-systeem. Het helpt bij het begrijpen van de structuur en hoe de verschillende componenten samenwerken.

In **Figuur 3.1** is het domeinmodel te vinden. Daaronder zijn de toelichtingen van de concepten te vinden.



Figuur 3.1: Domeinmodel BuddyBot.

Concept	Beschrijving
BuddyBot	De sociale robot die interactie heeft met de gebruiker en de input van sensoren verwerkt om actuatoren aan te sturen.
Sensor	Een hardwarecomponent die gegevens verzamelt. In dit systeem omvatten sensoren twee radaraanwezigheidssensoren en een camera.
Camera	Een sensor die gezichten detecteert en de positie van de gebruiker bepaalt.
Radar	Een sensor die de aanwezigheid van een gebruiker detecteert binnen een bepaalde straal. De radar kan ook bepalen of gedetecteerde gebruiker stil staat of beweegt.
Gezicht	De locatie van het gezicht van de gebruiker, bepaald door de camera. De coördinaten (x, y, z) van het gezicht ten opzichte van de camera.
Persoon	De gebruiker die door de robot wordt waargenomen en waarop gereageerd wordt.
Oog	Een visuele representatie van ogen op de LCD-schermen, die de gebruiker volgen op basis van gezichtspositie zoals gedetecteerd door de camera.
Actuator	Een hardwarecomponent die uitvoer genereert op basis van de input van de sensoren. In dit systeem omvatten actuatoren twee LCD-scherm en een speaker.

LCD-scherm	Een actuator die visuele feedback toont, zoals ogen die de gebruiker volgen.
Speaker	Een actuator die audio afspeelt, zoals begroetingen en afscheidsgeluiden.
Audio	De geluiden die via de speaker worden afgespeeld.

Tabel 3.1: Interface beschrijving tussen componenten

4 USE-CASE BESCHRIJVINGEN

4.1 USE CASE: AANZETTEN ROBOT

4.1.1 FULLY-DRESSED USE CASE BESCHRIJVING

Tabel 4.1: Aanzetten robot

Use Case ID: UC1	
Primary Actor:	Afstudeerder
Stakeholders:	Bewoner,
Description:	De gebruiker kan de robot aanzetten, zodat de robot operationeel is. Voor een gebruiker is van buiten af te zien dat de ogen van de robot weer-geven worden op de display schermen van de robot. Voor een ontwikkelaar kan de OS via ssh benaderd worden voor volledige controle (bij-voorbeeld software matig herstarten van robot).
Precondition:	<ol style="list-style-type: none"> 1. De robot is uit. 2. De robot en alle onderdelen zijn voorzien van stroom. 3. De besturingssoftware van de robot is aanwezig op de robot.
Postcondition:	De robot is operationeel.
Main	
Actor	System
1) De gebruiker drukt op de aan knop.	<ol style="list-style-type: none"> 2) Het systeem start de OS op. 3) Het systeem maakt een lokaal netwerk aan. 4) Het systeem start een ssh server op. 5) Het systeem configureert de LCD-schermen. 6) Het systeem start de besturingssoftware op.

4.2 USE CASE: DETECTEREN VAN AANWEZIGHEID

4.2.1 FULLY-DRESSED USE CASE BESCHRIJVING

Tabel 4.2: Detecteren van aanwezigheid

Use Case ID: UC2	
Primary Actor:	Bewoner

Stakeholders:	Bewoner
Description:	De robot detecteert de aanwezigheid van een gebruiker met radarsensoren. Wanneer een aanwezigheid binnen twee meter wordt gedetecteerd opent de robot zijn ogen.
Precondition:	<ol style="list-style-type: none"> 1. De display schermen zijn aangesloten. 2. De robot is aan. 3. De radarsensoren zijn verbonden.
Postcondition:	Twee mogelijke postcondities: <ol style="list-style-type: none"> 1. "[Een gebruiker is binnen twee meter aanwezig]" De robot heeft zijn ogen geopend. 2. "[Er is niemand aanwezig binnen twee meter]" De robot heeft zijn ogen gesloten.
Main	
Actor	System
2) De gebruiker loop richting de robot.	<ol style="list-style-type: none"> 1) De radarsensoren scannen continue zijn gebied. 3) [Robot detecteert een aanwezigheid binnen 2 meter] De robot opent zijn ogen.
Alternative	
Actor	System
	3A) [Robot detecteert geen enkel aanwezigheid binnen 2 meter] De robot sluit zijn ogen.

4.3 USE CASE: VOLG GEZICHT

4.3.1 FULLY-DRESSED USE CASE BESCHRIJVING

Tabel 4.3: Volg gezicht

Use Case ID: UC3	
Primary Actor:	Bewoner
Stakeholders:	Bewoner
Description:	De robot detecteert een gezicht met zijn camera en volgt een gezicht met zijn ogen.
Precondition:	<ol style="list-style-type: none"> 1. De camera is verbonden. 2. UC2 is uitgevoerd.
Postcondition:	De ogen van de robot volgt een gezicht.

Main	
Actor	System
1) De gebruiker loopt in het gezichtsveld van de robot.	2) De robot detecteert met de camera een gezicht en berekent de middelpunt coördinaten van het gezicht. 3) De robot kijkt met zijn ogen naar richting van het berekende middelpunt.

4.4 USE CASE: BEGROET GEBRUIKER BIJ NADEREN

4.4.1 FULLY-DRESSED USE CASE BESCHRIJVING

Tabel 4.4: Begroet gebruiker bij naderen

Use Case ID: UC4	
Primary Actor:	Bewoner
Stakeholders:	Bewoner
Description:	Wanneer een gebruiker voor de robot staat, wordt de gebruiker altijd begroet.
Precondition:	1. UC3 wordt uitgevoerd.
Postcondition:	De robot begroet de gebruiker.
Main	
Actor	System
1) De gebruiker staat recht voor de robot.	2) De robot zegt "Hallo ik ben BuddyBot, leuk dat je voor me staat".

4.5 USE CASE: AFSCHEID NEMEN BIJ VERLATEN

4.5.1 FULLY-DRESSED USE CASE BESCHRIJVING

Tabel 4.5: Afscheid nemen bij verlaten

Use Case ID: UC5	
Primary Actor:	Bewoner
Stakeholders:	Bewoner
Description:	De robot neemt afscheid en sluit zijn ogen wanneer een gebruiker eerder is gevolgd en nu langer niemand in zijn omgeving aanwezig is.

Precondition:	1. UC3 wordt uitgevoerd.
Postcondition:	De robot heeft afscheid genomen en zijn ogen gesloten.
Main	
Actor	System
1) De gebruiker verlaat het detectiebereik van de robot (>2 meter).	2) [De robot detecteert geen enkel aanwezigheid] De robot zegt "Doei doe tot de volgende keer" en sluit zijn ogen.
Alternative	
Actor	System
	2A) [De robot detecteert een aanwezigheid in zijn omgeving (<2 meter)] De robot doet niks.

5 REQUIREMENTS

5.1 OVERZICHT

In dit hoofdstuk worden de functionele en niet-functionele requirements van BuddyBot beschreven. De requirements definiëren de verwachtingen waaraan de robot moet voldoen om correct te functioneren binnen de vastgestelde context.

5.2 CATEGORISATIE VAN REQUIREMENTS

De requirements zijn ingedeeld volgens de FURPS-methode, waarbij de functionele eisen verder worden gespecificeerd met de MoSCoW-methode:

- **Functionality (F)**: Functionele vereisten die beschrijven wat het systeem moet doen.
- **Usability (U)**: Gebruiksvriendelijkheid en interactie met gebruikers.
- **Reliability (R)**: Betrouwbaarheid en robuustheid van het systeem.
- **Performance (P)**: Systeemprestaties en snelheidseisen.
- **Supportability (S)**: Onderhoudbaarheid en uitbreidbaarheid van het systeem.

5.3 OVERIGE FUNCTIONELE REQUIREMENTS (F)

Een deel van de functionele requirements zijn in use case vorm beschreven en zijn requirements die gerealiseerd moeten worden (MUST requirement). In dit paragraaf worden de overige functionele requirements beschreven. De overige functionele requirements die niet door de use cases worden afgedekt, zijn ingedeeld volgens de MoSCoW-methode en worden weergegeven in [Tabel 5.1](#).

Code	Beschrijving	Prioriteit
F-001	De aanwezigheid detectie moet plaatsvinden via radaraanwezigheidssensoren.	Must-have
F-002	De dichtstbijzijnde afstand-detectie van de radarsensoren wordt beschouwd als een persoon .	Must-have
F-003	De communicatie tussen de ESP32 en de mini-PC moet via USB verlopen.	Must-have
F-004	De gezichtsdetectie moet minimaal tot 1 meter afstand werken.	Must-have
F-005	De robot moet een opgegeven audiobestand (mp3) kunnen afspelen en stoppen.	Must-have
F-006	De robot moet een opgegeven audiobestand (mp3) kunnen pauzeren en hervatten.	Should-have
F-007	De robot moet een opgegeven audiobestand (mp3) kunnen pauzeren en hervatten.	Should-have
F-008	De ogen van de robot moeten periodiek knipperen.	Should-have
F-009	De pupillen van de ogen moeten afhankelijk van de afstand tot de ogen vernauwen of vergroten. Dichterbij = vergroten, Verderaf = vernauwen.	Should-have
F-010	De robot moet zijn nek kunnen bewegen met servo-motoren om de gebruiker fysiek te volgen.	Could-have

F-011	De robot moet emotionele expressies kunnen tonen op de LCD-schermen.	Could-have
F-012	De robot moet een animatie kunnen tonen op de LCD-schermen.	Could-have
F-013	De robot moet voorzien zijn van een microfoon en de opgepikte audio hiervan realtime kunnen laten horen.	Could-have
F-014	De robot kan een opgegeven text input afspelen als audio.	Could-have
F-015	De robot zal geen spraakherkenning ondersteunen om een interactie of dialoog met een gebruiker aan te gaan.	Won't-have

Tabel 5.1: Functionele requirements van BuddyBot.

5.4 USABILITY REQUIREMENTS (U)

Code	Beschrijving
U-001	De robot moet voor ontwikkelaars via ssh beschikbaar zijn.
U-002	Audio-uitvoer moet verstaanbaar zijn.

Tabel 5.2: Usability requirements van BuddyBot.

5.5 RELIABILITY REQUIREMENTS (R)

Voor deze uitvoering van het project zijn er geen Reliability requirements vereist.

5.6 PERFORMANCE REQUIREMENTS (P)

Voor deze uitvoering van het project zijn er geen performance requirements vereist.

5.7 SUPPORTABILITY REQUIREMENTS (S)

Code	Beschrijving
S-001	De software moet modulaair gebouwd zijn zodat toekomstige studenten makkelijk uitbreidingen kunnen aanbrengen.
S-002	De software moet gedocumenteerd zijn.
S-003	De software moet gebouwd zijn voor ROS2 jazzy.

Tabel 5.3: Supportability requirements van BuddyBot.

BIBLIOGRAFIE

[Object Management Group (OMG), 2015] Object Management Group (OMG) (2015). OMG Unified Modeling Language, Version 2.5. OMG Document Number formal/2015-03-01 (<http://www.omg.org/spec/UML/2.5>).