



Student:

Thomas Kok, 1627579

thomas.t.kok@student.hu.nl

HBO-ICT, richting Technische Informatica

Begeleider:

Richard Verhaaf

richard.verhaaf@sogeti.com

SCRIPTIE

Agent Oriented Programming in domotica-omgevingen

Lange Dreef 17
4131 NJ Vianen

+31 886 606 600

info@sogeti.nl
www.sogeti.nl

Versiebeheer

Datum	Versie	Wijziging
07-03-16	0.1	Opzet document
20-05-16	0.2	Invulling opzet van onderzoek
28-05-16	0.3	Onderzoeksresultaat APL's toegevoegd
10-06-16	0.3	Onderzoeksresultaat agent communicatie standaarden
20-06-16	0.3	Onderzoeksresultaat interview Wouter van Rooij toegevoegd
24-06-16	0.4	Aanwijzingen/opmerkingen docent verwerkt
27-06-16	0.5	Onderzoeksresultaat gateways toegevoegd
30-06-16	0.6	Onderzoeksresultaat interview Leo van Moergestel toegevoegd
01-07-16	0.7	Feedback bedrijfsbegeleider verwerkt
02-07-16	0.8	Invulling gegeven aan het antwoord op de hoofdvraag
03-07-16	1.0	Document afgerond en managementsamenvatting toegevoegd

Managementsamenvatting

“Domotica is een ontzettend mooie richting qua hardware en om voor te programmeren”, aldus Steven van Dalen. Echter heeft ieder domotica-device een probleem; het is fabrikantgebonden. Dit leidt tot slechte interoperabiliteit. Sogeti gaat hier verandering in aanbrengen en gaf Technisch Informatica afstudeerder Thomas Kok de opdracht om hier onderzoek naar te verrichten en een proof of concept hiervoor te maken.

Dit onderzoek wordt uitgevoerd om antwoord te krijgen op de vraag hoe agent-oriented programming is in te zetten voor een betere samenwerking tussen devices in een domotica-omgeving. De onderzoeksvraag is opgebroken in deelvragen. Om antwoord te krijgen op deze vragen is literatuuronderzoek uitgevoerd en zijn interviews gehouden met Leo van Moergestel en Wouter van Rooij.

Uit literatuuronderzoek is gebleken dat Jade het beste programmeerframework is om het proof of concept te realiseren omdat het geschikt is voor embedded systemen. Dit valt samen met het antwoord op de deelvraag over welke communicatiestandaard gebruikt kan worden. De FIPA ACL standaard wordt gebruikt omdat deze niet gebonden is aan een programmeerframework. Tevens wordt FIPA ACL standaard aangeboden in Jade. Er is één vergelijkbaar project gevonden dat is uitgevoerd door een onderzoeksteam onder leiding van Leo van Moergestel. De resultaten van dit project suggereren de Raspberry Pi te gebruiken voor het proof of concept. Literatuuronderzoek naar een alternatief voor een domotica-gateway is op niets uitgelopen. Exploratief onderzoek liet echter zien dat het principe van het Zigbee netwerk kan worden ingezet voor het vraagstuk van dit onderzoek. Met dit principe kan ieder domotica-device optreden als gateway om zo fabrikantgebonden speciale gateways buiten spel te zetten en interoperabiliteit te verbeteren.

Voor de resterende deelvragen zijn interviews gehouden. Allereerst met een autoriteit op het gebied van domotica en agent technologie, Leo van Moergestel. Ook is een interview gehouden met Wouter van Rooij, om kennis in te winnen over beveiliging van programmeerwerk. Volgens Leo van Moergestel heeft het toekennen van prioriteit aan inputs met gezond verstand te maken, eventueel met verder onderzoek per situatie. Volgens Wouter van Rooij zijn er negen aspecten waar rekening mee gehouden moet worden om een applicatie veilig te programmeren: clientside controls, authenticatie, sessiemanagement, toegang en de controle hiervan, inputvalidatie, functie input, logica fouten, omgeving en tot slot server. Volgens van Rooij is het voor domotica met name belangrijk om te letten op toegangsvalidatie, inputvalidatie en logicavalidatie. Het derde punt is bevestigd in een onderzoek van Jacobsson (Jacobsson et al, 2016). Deze aspecten zullen worden meegenomen in het proof of concept.

Agent oriented programming is in te zetten om interoperabiliteit tussen domotica-devices te verbeteren en biedt meerdere mogelijkheden tot uitbreiding en doorontwikkeling in de toekomst.

Het proof of concept wordt gemaakt met een drietal Raspberry Pi 3 B's en een Android telefoon, om het antwoord op de onderzoeksvraag te demonstreren.

Inhoudsopgave

I.	Inleiding	1
II.	Context	2
	Bedrijf	2
	Werkwijze.....	2
III.	Opdracht	3
	Kwestie	3
	Formulering	3
	Opdrachttype.....	4
	Doelstelling.....	4
IV.	Op te leveren producten	5
V.	Onderzoek en methodiek.....	6
	Hoofdvraag.....	6
	Deelvragen	6
VI.	Resultaten	10
	Onderzoeksdeelvragen.....	10
	Agent Oriented Programming frameworks	10
	Communicatiestandaard	18
	Soortgelijke projecten	20
	Domotica-gateways.....	22
	Inputprioriteit	23
	Veiligheidsaspecten	24
VII.	Conclusie.....	28
	Hoofdvraag.....	28
VIII.	Proof of concept.....	31
IX.	Aanbevelingen & toekomstbeeld	32
X.	Bibliografie	33
XI.	Bijlagen.....	35
	Bijlage A: Plan van aanpak.....	35

Inleiding

Voor iedere afstudeerstudent een afstudeeropdracht en onderzoek. Zo ook voor Thomas Kok. Hij sluit zijn opleiding Technische Informatica af met een afstudeerstage bij Sogeti. Sogeti is een ICT-dienstverlener en heeft tal van interessante projecten lopen voor een Technisch Informatica student als Thomas. Eén van deze projecten is het onderzoeken hoe smart devices in een domotica-omgeving beter met elkaar kunnen communiceren. Vrijwel ieder huishouden heeft tegenwoordig een smart device in huis. Zij het in de vorm van een Nest apparaat, een Toon thermostaat of misschien zelfs een IP-camera voor de deur zodat op de televisie zichtbaar gemaakt kan worden of de pizzabezorger er al is.

Vrijwel alle apparaten nu op de markt, noem het Philips HUE of bijvoorbeeld KlikAanKlikUit, hebben allen één groot nadeel: ze zijn fabrikantgebonden. Hierdoor is geen sprake van interoperabiliteit.

Hier komt het onderzoek van deze afstudeeropdracht om de hoek kijken. Voor Sogeti is het uitermate interessant en belangrijk om deze interoperabiliteit te vergroten, indien mogelijk met het achterwege laten van gateways en controllers. Zachariah stelde dat het Internet of Things een probleem tegemoet gaat met betrekking tot gateways (Zachariah, 2015). In zijn woorden:

“In much the same way that it would be difficult to imagine requiring a new web browser for each website, it is hard to imagine our current approach to IoT connectivity scaling to support the IoT vision.”

In de ogen van de opdrachtgever, Steven van Dalen, is dit probleem ook aanwezig voor domotica. Dit komt mede doordat apparaten in een domotica-omgeving op dezelfde manier werken, voor iedere toepassing is dikwijls een nieuw gateway nodig. Ook zijn apparaten zeer beperkt met betrekking tot communicatie met andere apparaten wanneer deze niet van dezelfde fabrikant komen.

Door Thomas zal onderzoek uitgevoerd worden om middels het programmeerprincipe Agent Oriented het probleem van slechte interoperabiliteit de wereld uit te helpen.

In dit document zal allereerst een schets worden gemaakt van de omgeving waarin de opdracht uitgevoerd gaat worden en een beschrijving van Sogeti en haar werkwijze. Vervolgens wordt de opdracht beschreven, waarna het onderzoek en de te gebruiken methoden daarbij worden toegelicht. In het aansluitende hoofdstuk zullen de behaalde resultaten gepresenteerd worden. De resultaten van de deelvragen zullen worden samengevoegd in een antwoord op de hoofdvraag, welke wordt beschreven in de conclusie. Aansluitend volgt een beschrijving van het proof of concept en hoe deze in elkaar steekt. Tot slot is er ruimte voor de discussie en aanbevelingen.

Context

Bedrijf

Sogeti is een leverancier van ICT-diensten en onderdeel van Capgemini. Sogeti werd opgericht in 1967 door een Fransman, totdat het in 1975 fuseerde met CAP en Gemini Computer Systems tot CAP Gemini Sogeti. In 2002 werd Sogeti losgemaakt en geïntroduceerd als onafhankelijk bedrijf. Jaarlijks maakt Sogeti een omzet tussen de 200 en 300 miljoen, met in totaal meer dan 2500 medewerkers verdeeld over alle vestigingen wereldwijd. Sogeti is een relatief plat bedrijf. Organisatorisch gezien zijn er vier lagen, echter is het de bedoeling dat iedereen elkaar kan aanspreken en dit ook doet. Er heerst een relatief informele werksfeer met veel ruimte voor eigen input.

Sogeti detacheert haar werknemers naar bedrijven aan wie zij haar diensten verleent. Enkele grote partners waar op vaste basis medewerkers aan gedetacheerd worden zijn Microsoft, IBM en banken zoals ING en Rabobank.

Het bedrijf is onderverdeeld in divisies en business lines. Voor stagiairs en afstudeerders is er een aparte unit met professionele begeleiding en coaches. Ook biedt Sogeti haar medewerkers toegang tot alle mogelijke bijscholingen en cursussen. Onderdeel van deze cursussen is een scala aan certificeringen voor bijvoorbeeld Microsoft of Cisco.

Werkwijze

De werkwijze van Sogeti bestaat grotendeels uit het detacheren van haar medewerkers. Echter biedt zij ook mogelijkheden om binnen het bedrijf zelf projecten uit te voeren. Voor het uitvoeren van deze interne projecten beschikt Sogeti over een aantal eigen flexwerkplekvestigingen in onder andere Capelle aan de IJssel, Amersfoort en Amsterdam. Voor afstudeerders is er ruimte in deze vestigingen, alsmede voor alle andere medewerkers die behoefte hebben aan een werkplek wanneer het bedrijf waar zij gedetacheerd zijn bijvoorbeeld geen werkplekken beschikbaar hebben.

Deze werkwijze resulteert in een divers aanbod van kennis, variërend van dag tot dag. Uiteraard zijn er communicatiesystemen aanwezig zodat medewerkers elkaar kunnen contacteren zonder dat zij op dezelfde vestiging hoeven te zijn. Skype for Business en Lync worden hier met name voor gebruikt, maar medewerkers reageren zeer goed en snel op e-mails.

Opdracht

Kwestie

Steeds meer bedrijven beginnen domotica apparaten te leveren. Toon, Nest, KlikAanKlikUit, Philips HUE, de lijst met spelers blijft groter worden. Al deze producten creëren een nieuwe dimensie op het gebied van gebruikersgemak. Zo is het met de een mogelijk je verlichting vanaf je telefoon te regelen waarin de ander je juist voorziet in een slimme thermostaat die op zijn beurt ook weer aan is te sturen vanaf de telefoon dan wel een tablet. Wat de producten echter allemaal in gemeen hebben is dat deze gebonden zijn aan een en dezelfde fabrikant. Onderlinge communicatie en samenwerking van dergelijke producten is helaas nog uit den boze en daar zit het probleem. Want wat als alle apparaten direct met elkaar kunnen communiceren, zonder dat zij van dezelfde fabrikant moeten zijn? Als een willekeurige smart televisie kan communiceren met je op afstand bestuurbare verlichting, dat zodra je een film op zet de verlichting hiervoor dimt?

Dit doel is exact wat Sogeti voor ogen heeft. Alle smart devices gemakkelijk met elkaar laten communiceren. Onderlinge consensus en bewustzijn van de mogelijkheden van de apparaten om zich heen. Daarom streeft Sogeti naar een algemene standaard voor communicatie tussen smart devices, zodat uiteindelijk het geen probleem meer is dat apparaten van verschillende fabrikanten zijn wanneer zij deze standaard implementeren. Het achterliggende programmeren is dan volledig aan de fabrikant.

Gedurende de afstudeeropdracht van Thomas zal er onderzocht worden hoe Agent Oriented Programming hiervoor gebruikt kan worden en de resultaten van het onderzoek worden besproken in dit document.

Formulering

Na enige gesprekken met Steven van Dalen is vastgesteld dat bij Sogeti de behoefte leeft aan meer duidelijkheid over hoe smart en connected devices in een domotica-omgeving betere interoperabiliteit kunnen bereiken. Het doel van Sogeti is hierbij de ontwikkeling van een standaardprotocol binnen de domotica wereld. Dit standaardprotocol dient ervoor te zorgen dat apparaten betere interoperabiliteit bereiken en Sogeti wil deze standaard dan ook laten integreren in de industrie. In de woorden van Steven van Dalen:

"Domotica is een ontzettend mooie richting van hardware en programmeren. De mogelijkheden zijn eindeloos, maar momenteel binnen de beperkingen opgelegd door de fabrikant. Dit is absoluut niet wenselijk en Sogeti wil hier verandering in brengen. Aan jou de opdracht om te onderzoeken hoe devices in domotica-omgevingen zich volgens een standaard kunnen gedragen die betere samenwerking met zich mee draagt, waarbij geen gateway vereist is. Demonstreer de werking van deze standaard middels een proof of concept. "

Na verder sparren met Steven werd bovendien de wens geuit dat men de voorkeur heeft voor een oplossing geheel zonder het gebruik van gateways. Reden hiervoor is dat Steven zo min mogelijk randapparatuur wil introduceren. Niet alleen de extra randapparatuur wordt als drijfveer gebruikt, een eventueel gateway is namelijk te gemakkelijk om te verbinden aan een leverancier. Dit werkt interoperabiliteit tegen.

Opdrachttype

De verworven afstudeerplek is hoofdzakelijk een onderzoeksopdracht, waarbij ook een proof of concept opgeleverd gaat worden. Dit heeft als toegevoegde waarde dat de student de onderzoeksvaardigheden die zijn opgedaan gedurende de opleiding nu in de praktijk kan toepassen.

Doelstelling

De afstudeeropdracht wordt uitgevoerd met een aantal hoofddoelen voor ogen:

- Het opleveren van een proof of concept;
- Het vaststellen van een protocol dat voor devices in een domotica-omgeving ingezet kan worden;
- Het opleveren van een onderzoeksrapport waar de bevindingen in beschreven staan;
- Een onderzoek tot een goed einde leiden.

Op te leveren producten

In dit onderzoek is veel informatie vergaard. Deze informatie wordt gebruikt om een drietal producten op te leveren. Allereerst is er het onderzoeksrapport, een document wat voor de opdrachtgever bedoeld is en compact de nieuwe kennis noteert. Ten tweede zal er een proof of concept geprogrammeerd worden, om de werking van het communicatieprotocol te demonstreren. Tot slot zal er tijdens de verdediging een bundel beschikbaar worden gesteld waar in groter detail wordt beschreven hoe het proof of concept werkt. Er is bewust gekozen voor het inleveren van een bundel tijdens de verdediging, daar wegens onvoorziene omstandigheden de afstudeerder aanzienlijke vertraging heeft opgelopen. De afstudeerder heeft geen invloed kunnen uitoefenen op deze omstandigheden en had deze niet kunnen voorkomen, waardoor het niet mogelijk was om de werking van het proof of concept in detail te beschrijven in dit document. Wel zal beschreven worden hoe het proof of concept op wordt gezet en hoe verwacht wordt dit uit te voeren.

Het onderzoeksrapport voor de opdrachtgever zal bestaan uit de antwoorden op de onderzoeksvragen van dit onderzoek. In het rapport zal de volgende structuur gehanteerd worden: Opzet en uitvoering, resultaten, resultaten bespreken en concluderen.

Op deze wijze is voor de opdrachtgever helder en compact te lezen wat de resultaten van het onderzoek zijn.

Onderzoek en methodiek

Om het onderzoek gedegen uit te kunnen voeren is de hoofdvraag onderverdeeld in deelvragen. Door een antwoord te vergaren op de deelvragen is het mogelijk om de hoofdvraag van een onderbouwd antwoord te voorzien. Hierover meer in dit hoofdstuk, met een beschrijving van de opzet van het onderzoek per deelvraag.

Hoofdvraag

"Hoe kan Agent Oriented Programming ingezet worden om de interoperabiliteit van connected devices in domotica-omgevingen zonder gateways te verbeteren?"

Deze hoofdvraag zal hoofdzakelijk beantwoord worden middels de vergaarde informatie bij het onderzoek van de deelvragen. De methode voor het beantwoorden van deze vraag betreft een som van de methoden die per deelvraag worden beschreven, in onderstaande koppen, omdat de antwoorden op de deelvragen gebruikt worden om het model vorm te geven wat het antwoord op de hoofdvraag is.

Doelstelling van het onderzoek is de aanzet voor een standaardprotocol waarmee devices in een domotica-omgeving met elkaar kunnen communiceren zonder dat hiervoor een controller of gateway nodig is.

Deelvragen

De zojuist besproken hoofdvraag zal beantwoord worden met behulp van een zestal deelvragen. Per deelvraag zal een plan geformuleerd worden met de verwachting vergaring van een antwoord.

Agent Oriented Programming frameworks

"Welke programmeertaal heeft een Agent Oriented Programming framework en is geschikt voor het uitvoeren van de opdracht?"

Deze deelvraag zal beantwoord gaan worden door middel van een exploratief onderzoek dat als doel heeft helder te krijgen welke programmeertalen geschikt zijn voor Agent Oriented Programming. Er is voor deze methode gekozen omdat het antwoord op de vraag een sommatie is van mogelijke programmeertalen, waarbij een exploratief onderzoek een goede match is.

Uit de mogelijke programmeertalen zal vervolgens een besluit worden genomen voor een programmeertaal die het meest geschikt is om het proof of concept in op te zetten. Deze keuze zal worden gemaakt op basis van de volgende criteria:

- Is de taal geschikt om te draaien op een embedded systeem, rekening houdende met de beschikbare rekenkracht op embedded systemen?
- Is de community van de taal levendig genoeg om support te verlenen?

Voor de doeleinden van dit onderzoek verstaan we onder de term embedded systeem, een systeem wat draait op een Cortex architectuur van ARM, vergelijkbare architecturen, of betere architecturen.

Als startpunt van het onderzoek zal bestaande kennis van de student gebruikt worden om bekende programmeertalen onder de loep te nemen, om vervolgens bij gerelateerde programmeertalen terecht te komen. Meer hierover in het volgende hoofdstuk, de resultaten.

Het verwachte resultaat van deze deelvraag is een onderbouwde keuze voor een programmeerframework dat gebruikt kan worden om het communicatieprotocol mee op te bouwen.

Communicatiestandaard

"Welke standaard kan gebruikt worden om agents in een domotica-omgeving onderling te laten communiceren?"

Deze deelvraag zal beantwoord gaan worden door middel van een exploratief onderzoek dat als doel heeft helder te krijgen welke communicatiestandaarden er zijn. Er is gekozen voor deze methode omdat het antwoord op de vraag een sommatie is van de beschikbare standaarden, wat goed aansluit bij een exploratief onderzoek.

Om een goede en onderbouwde keuze te kunnen maken, is de volgende subvraag nodig:

"Welke informatie willen agents in een domotica-omgeving uitwisselen met elkaar?"

Het antwoord op deze deelvraag zal worden vergaard door middel van een interview met een autoriteit op het gebied domotica, Leo van Moergestel. Leo van Moergestel heeft een groot aantal publicaties gedaan in de richting van Agent Oriented Programmeren, alsook in de richting van Technische Informatica.

Uit de mogelijke communicatiestandaarden zal vervolgens een besluit worden genomen voor een standaard die het meest geschikt is om gebruikt te worden bij het beantwoorden van de hoofdvraag. Deze keuze zal worden gemaakt op basis van de volgende criteria:

- Is de standaard gebonden aan een set programmeertalen of niet?
- Biedt de standaard de mogelijkheid om alle informatie te communiceren die een domotica-agent zou willen uitwisselen?

Het verwachte resultaat van deze deelvraag is een onderbouwde keuze voor een communicatiestandaard die gebruikt kan worden voor het communicatieprotocol.

Soortgelijke projecten

"In welke soortgelijke toepassingen wordt Agent Oriented Programming al toegepast?"

Deze deelvraag zal beantwoord gaan worden door middel van een literatuuronderzoek. Er zal worden gezocht naar soortgelijke projecten en hun documentatie. Er is voor deze methode gekozen omdat uit de resultaten van het onderzoek naar deze vraag informatie uit publicaties/documentatie gehaald moet worden. Dit sluit goed aan bij een literatuuronderzoek.

Ter vergaring van deze onderzoeken zal een exploratief en relationeel onderzoek worden uitgevoerd, om op die wijze bij projecten terecht te komen en relationeel gezien door te hoppen naar gerelateerde projecten.

Uit de vergaarde onderzoeken/publicaties zal de volgende subvraag beantwoord worden:

"Welke aspecten van deze projecten zijn toepasbaar op dit onderzoek?"

De antwoorden op deze vragen zullen bijdragen aan het antwoord op de overkoepelende deelvraag.

Qua resultaat van de deelvraag wordt er verwacht dat overkomen uitdagingen van de andere projecten bijdragen om hetzelfde te realiseren voor dit onderzoek.

Domotica-gateways

"Wat wordt verstaan onder een domotica-gateway en welke alternatieven zijn hiervoor?"

Deze deelvraag zal beantwoord gaan worden door middel van een literatuur- en exploratief onderzoek. Er zal worden gezocht naar een definitie van het begrip gateway, en qua alternatief zal er om te beginnen gekeken worden naar de netwerkarchitectuur van ZigBee. Vanuit dit uitgangspunt zal relationeel gezien verder worden gezocht. Er is gekozen voor deze methoden omdat de benodigde informatie om antwoord te geven op deze vraag hoofdzakelijk een definitie is, dit is goed te documenteren in een publicatie en daarom sluit dit goed aan bij literatuuronderzoek. Exploratief onderzoek zal aan bod komen indien er geen publicaties/artikelen gevonden kunnen worden.

Het verwachte resultaat van deze deelvraag is dat er een heldere beschrijving is van een domotica gateway, en er een mogelijk alternatief wordt besproken.

Inputprioriteit

"Hoe dient een domotica-agent prioriteit toe te kennen aan zijn verscheidene inputs?"

Deze deelvraag zal hoofdzakelijk beantwoord worden door middel van een literatuuronderzoek. Een onderdeel van deze vraag is vast te stellen in de volgende subvraag:

"Welke verschillende inputs heeft een domotica-agent?"

Deze vraag zal beantwoord worden middels een literatuuronderzoek en een interview met een autoriteit op het gebied, Leo van Moergestel.

De methoden voor deze deelvraag en zijn subvragen zijn gekozen omdat de verwachting is dat informatie over de inputs van agents goed gedocumenteerd en reeds onderzocht is, waardoor op basis van publicaties een antwoord op de vraag gegeven kan worden. Daarom sluit een literatuuronderzoek goed aan.

Het verwachte resultaat van deze deelvraag is dat er bekend is hoe bepaald kan worden welke prioriteit een input toegekend moet worden in het communicatieprotocol.

Veiligheidsaspecten

"Met welke veiligheidsaspecten hebben agents in een domotica-omgeving rekening te houden?"

Deze deelvraag zal hoofdzakelijk beantwoord worden door middel van een interview met een autoriteit op het veld van veiligheid en security binnen Sogeti, Wouter van Rooij. Er zal qua veiligheid slechts worden gekeken naar informatie en gebruikersauthenticatie, niet op hardwareniveau.

Wouter is een Sogetist en werkt sinds 2013 bij Sogeti als Operational Manager Center of Excellence Informatiebeveiliging, en heeft onder meer een grote bijdrage geleverd aan het testen van de veiligheid van de ING Bankieren applicatie. Hierbij was het zijn taak om vast te stellen dat de beveiliging van deze applicatie op orde was (persoonlijke mededeling van Wouter van Rooij aan Thomas, 16 juni 2016).

Aan de hand van de informatie aangereikt door Wouter van Rooij, kan er een opsomming worden gemaakt van veiligheidsaspecten waar agents rekening mee dienen te houden.

Naast het interview wordt een literatuuronderzoek uitgevoerd. Uit dit literatuuronderzoek zal blijken welke risicofactoren/veiligheidsaspecten reeds bekend zijn.

Het verwachte resultaat van deze deelvraag is dat er bekend is met welke veiligheidsaspecten er rekening gehouden moet worden voor agents in een domotica-omgeving.

Resultaten

In dit hoofdstuk zullen de vergaarde resultaten aangeboden worden. Allereerst zullen de antwoorden op de deelvragen behandeld worden, aangezien deze antwoorden gebruikt worden bij het beantwoorden van de hoofdvraag. Na de deelvragen wordt er ingegaan op de hoofdvraag.

Onderzoeksdeelvragen

Agent Oriented Programming frameworks

De resultaten die aangeboden worden in dit hoofdstuk zijn gekoppeld aan de volgende deelvraag:

Welke programmeertaal heeft een Agent Oriented Programming framework en is geschikt voor het uitvoeren van de opdracht?

Programmeertalen voorzien van een dergelijk framework worden doorgaans een Agent Programming Language (APL) genoemd.

Uit het exploratieve onderzoek is de volgende lijst van de meest prominente APL's voortgekomen welke behandeld zullen worden in dit onderzoek:

- AgentSpeak
 - o Jason (uitbreiding/extensie van AgentSpeak)
 - o JaCa (uitbreiding/extensie van AgentSpeak)
- JADE
- 3APL
- CLAIM
- Jadex
- Jack
- DEFACTO
- ARTIMIS

Bij het onderzoek naar iedere taal is naar voren gekomen dat Jack, DEFACTO en tot slot ARTIMIS inmiddels dusdanig verouderd en niet meer ondersteund zijn, dat deze niet beschreven zullen worden in de resultaten. De meest recente publicatie van deze drie talen is van DEFACTO door Nathan Schurr in 2005.

AgentSpeak

AgentSpeak is een simpele en krachtige programmeertaal voor op het belief-desire-intention (BDI) principe gebaseerde Agents. BDI is een softwaremodel gemaakt voor intelligente agents, waarbij een agent bekend is met zijn omgeving (belief), een verlangen heeft om een bepaald doel te behalen (desire) en tot slot het gemaakte besluit voor een actie gebaseerd op zijn beliefs en desires (intention). De taal werd door Anand Rao gemaakt in 1996, werd in eerste instantie AgentSpeak(L) genoemd, en baseerde zijn werk grotendeels op het Procedural Reasoning System (PRS). Dit PRS werd door SRI International eind jaren '80 ontwikkeld als framework voor realtime rationele systemen, systemen die besluiten maken op basis van een geloof van de omgeving.

AgentSpeak werd vervolgens veelal geciteerd in andere onderzoeken, met name in de richting van Multi-Agent Systems (MAS), hoewel het zelf niet per definitie geschikt is voor MAS programmeren door zijn gebruik aan een mogelijkheid om agents met elkaar te laten communiceren. Uiteindelijk werd AgentSpeak als basis gebruikt voor andere programmeertalen, en zelf niet meer om mee te programmeren.

De programmeertaal ziet er uit zoals in het volgende Hello World voorbeeld beschreven:

```
started.+started <- .print("Hello World!").
```

Is de taal geschikt om te draaien op een embedded systeem, rekening houdende met de beschikbare rekenkracht op embedded systemen?

Rahwan heeft samen met een groep onderzoekers in 2004 een applicatie in AgentSpeak(L) geprogrammeerd voor een PDA (Rahwan, 2004). Als rekening wordt gehouden met Moore's Law (Beal, 2016), is af te leiden dat een PDA uit 2004 onmogelijk meer rekenkracht tot zijn beschikking zou kunnen hebben dan de in dit onderzoek gestelde richtlijn voor een embedded systeem.

Is de community van de taal levendig genoeg om support te verlenen?

Voor deze taal is geen community gevonden. Voor AgentSpeak zal geen support gevonden kunnen worden.

Jason

Als een van de eerste en belangrijkste programmeertalen die naar voren is gekomen uit AgentSpeak, is Jason. Jason is ontwikkeld met de bedoeling om compleet te zijn, verscheidene libraries zijn beschikbaar en ook zijn er debugtools. Tevens is het idee achter Jason om snel "up and running" te zijn. In tegenstelling tot AgentSpeak, is Jason wel geschikt voor MAS programmeren en zeer instelbaar. Dit heeft de taal bereikt door als een schil over

AgentSpeak heen geplaatst te worden, waarbij het mogelijk is gemaakt om agents te laten communiceren. Dit is gerealiseerd door Jason een interpreter te maken die over AgentSpeak heen wordt geplaatst, geprogrammeerd in Java. Ook deze taal stelt je in staat om agents te programmeren volgens het BDI-principe. Jason maakt gebruik van het Agent Communication Language protocol door FIPA vastgesteld, en biedt de gebruiker ook KQML. Meer over KQML in het hoofdstuk over de communicatiestandaard van dit onderzoek.



Figuur 1 - Jason logo. Bron: Jason Sourceforge

Aangezien Jason voort is gebouwd op AgentSpeak, ziet het Hello World voorbeeld er hetzelfde uit:

```
started.+started <- .print("Hello World!");
```

Is de taal geschikt om te draaien op een embedded systeem, rekening houdende met de beschikbare rekenkracht op embedded systemen?

Jason is gebouwd op Java, in onderzoek van de heer van Moergestel is vastgesteld dat de Raspberry Pi voldoende rekenkracht heeft om Java-applicaties te draaien (Moergestel, 2013). Er kan daarom gesteld worden dat Jason geschikt is om te draaien op een embedded systeem.

Is de community van de taal levendig genoeg om support te verlenen?

Er is geen levendige community gevonden, echter is er naslagwerk van 294 pagina's beschikbaar (Wooldridge, 2007).

JaCa

JaCa is een programmeermodel in essentie een samensmelting is van Jason, de zojuist besproken taal, en CARtAgO. Aangezien er qua documentatie en informatie van CARtAgO nagenoeg niets te vinden is, heb ik deze taal niet opgenomen in dit onderzoek. JaCa wordt door het framework JaCa-Android gebruikt om agent oriented programmeren naar het Android platform te brengen. Daarbij wordt Jason gebruikt om entiteiten van agents te programmeren. CARtAgO wordt gebruikt om gemakkelijk te communiceren met en gebruik te maken van aanwezige Android apparaatgebonden componenten. Hier kan worden gedacht aan bijvoorbeeld de gps-module, de accelerometer en andere sensoren. Aangezien JaCa voortbouwt op onder meer Jason, is het mogelijk om agents te programmeren middels het BDI-principe.



*Figuur 2 - JaCa-Android logo.
Bron: Mattia Minotti*

Het volgende voorbeeld lijkt daarom heel erg op een Java applicatie voor Android:

```
public class HelloWorldGUIArtifact extends GUIArtifact {  
    protected void init (JaCaActivity activity, Bundle savedInstanceState) {  
        super.init(activity, savedInstanceState);  
    }  
    @OPERATION void print () { System.out.println("Hello World!") }  
}
```

Is de taal geschikt om te draaien op een embedded systeem, rekening houdende met de beschikbare rekenkracht op embedded systemen?

JaCa is ontworpen voor Android. Android is een platform voor embedded systemen, als gesteld door Matthew Mombrea (Mombrea, 2013). JaCa is daarom geschikt voor embedded systemen.

Is de community van de taal levendig genoeg om support te verlenen?

Na exploratief onderzoek op fora en via zoekmachine Google Scholar, is gebleken dat er geen community is voor JaCa. Echter bestaat er een zeer uitgebreide gids, gemaakt door de developers achter JaCa waardoor genoeg informatie beschikbaar is (Asanti, 2016).

Jade

Het Java Agent DEvelopment Framework, JADE, is het eerste grote framework dat niet op AgentSpeak voortborduurt. JADE is compleet in de taal Java gerealiseerd, en heeft een ingebouwde oplossing voor Multi-Agent Systems, MAS. Dit wordt geïmplementeerd als het zogenaamde netwerkmanagementtool. Tevens ondersteunt JADE zowel communicatie over draadloze media als ethernet. Momenteel



Figuur 3 - Jade logo. Bron: Telecom Italia

heeft JADE de meest levendige community en zijn er een groot aantal systemen in gerealiseerd. Het is met JADE mogelijk om zowel enkele agents te programmeren in een eigen omgeving, als het programmeren van meerdere agents die onderling communiceren middels de Agent Communication Language gedefinieerd in de FIPA-standaard. Het Jade framework beschikt over het zogenaamde Message Transport Protocol (MTP), een module die de gebruiker in staat stelt om de FIPA ACL berichten over het netwerk te verzenden naar een ander platform middels het HTTP- of IIOP-protocol (TI-lab, 2010). Dit andere platform hoeft niet per definitie op Jade gebouwd te zijn, het hoeft alleen de FIPA ACL standaard geïmplementeerd te hebben (TI-lab, 2010). Tevens stelt de taal je in de gelegenheid om op eigen manier het BDI-principe te implementeren.

Het volgende voorbeeld van een Hello World applicatie is geschreven in Java:

```
import jade.core.Agent;
public class HelloWorldAgent extends Agent {
    protected void setup() {
        System.out.println("Hello World! My name is "+getLocalName());
    }
}
```

Is de taal geschikt om te draaien op een embedded systeem, rekening houdende met de beschikbare rekenkracht op embedded systemen?

Van Moergestel heeft in zijn onderzoek vastgesteld dat Jade geschikt is om op embedded systemen, zoals de Raspberry Pi, gebruikt te worden. Er kan daarom gesteld worden dat Jade geschikt is voor embedded systemen (Leo van Moergestel, 2013).

Is de community van de taal levendig genoeg om support te verlenen?

Na kort exploratief onderzoek is vastgesteld dat er allereerst door JadeWorld een forum beschikbaar wordt gesteld om Jade projecten te bespreken. De meest recente reactie op forums.jadeworld.com is op moment van schrijven nog geen drie dagen geleden, 28 juni 2016. Ook biedt TI-lab een groot aantal papers en onderzoeken aan als naslagwerken. Tot slot is er op Stackoverflow een levendige community, waarop in de afgelopen 24 uur van moment van schrijven, 8 topics actief zijn geweest.

3APL

3APL is de samentrekking van "An Abstract Agent Programming Language", en is een experimentele tool voor het ontwikkelen, implementeren en testen van agents middels het BDI-principe. De taal is door een team onderzoekers van de Universiteit Utrecht ontwikkeld als een simpele logica taal. De taal beschikt over logische operatoren zoals bijvoorbeeld AND, OR, en NOT. Wanneer een 3APL agent wordt opgezet, worden zijn BDI-aspecten geprogrammeerd door gebruik te maken van een syntax die op PROLOG lijkt. Aangezien het een experimentele tool betreft, en grotendeels draait op PROLOG, heeft deze taal geen ingebouwd Hello World voorbeeld. Hoewel experimenteel, de taal is aangevuld met een wijze om agents met elkaar te laten communiceren. Daarbij is voor de Agent Communication Language standaard gekozen.

Er bestaat een implementatie die het mogelijk maakt om deze taal aan te sturen binnen Java 2 Micro Editie, zoals bijvoorbeeld de programmeertaal 3APL-M, waar een Hello World voorbeeld er als volgt uit zou zien:

```
public class HelloWorldExample {  
    public void startApp () {  
        Agent ag = new Agent("hello");  
        ag.addCapability ("{} Print(X) { GUI(print, X)}");  
        ag.addPlanRule (" <- TRUE | Print('hello world')");  
        ag.addGoal ("print");  
    }  
}
```

Is de taal geschikt om te draaien op een embedded systeem, rekening houdende met de beschikbare rekenkracht op embedded systemen?

3APL is te gebruiken met Java 2 Micro editie, een platform specifiek voor embedded en mobiele systemen bedoeld. 3APL is geschikt om op embedded systemen te draaien.

Is de community van de taal levendig genoeg om support te verlenen?

De meest recente informatie over 3APL is te vinden op www.cs.uu.nl/3apl/. Deze website is voor het laatst geüpdatet op 16 juni 2006. Daarom is te stellen dat er geen community voor de taal aanwezig is. Er zal geen ondersteuning te verkrijgen zijn voor 3APL.

CLAIM

CLAIM is de samentrekking van Computational Language for Autonomous Intelligent and Mobile agents. Zoals de naam al suggereert, is CLAIM een taal bedoeld voor het programmeren van autonome intelligente agents alsook mobiele agents. Deze taal is ook geschikt voor MAS, waarbij agents in een boomstructuur ondergebracht worden, vergelijkbaar met een ZigBee netwerk. Dit houdt voor de agent in dat hij een ouder agent opslaat, een lijst van subagents en natuurlijk een lijst van lokale processen/functionaliteiten. Onderdeel van deze functionaliteiten is de mogelijkheid om volgens het BDI-principe te opereren. Wanneer een agent aangemaakt wordt, moet dit gedaan worden aan de hand van een template, wat te vergelijken valt met een struct uit de C-taal. Vervolgens wordt een nieuwe agent aangemaakt met dit template. Zodra een CLAIM-agent is aangemaakt, wordt deze gedistribueerd binnen de zogenaamde SyMPA omgeving. SyMPA is te zien als een container, gemaakt in Java, welke de functiecalls van CLAIM afhandelt. Hierbij is het mogelijk om CLAIM gebruik te laten maken van Java functionaliteit.

Een Hello World applicatie zou aanzienlijk meer setup vereisen dan de vorige talen, omdat er een compleet template gedefinieerd dient te worden waarna een nieuwe agent geïntantieerd wordt binnen de SyMPA distributieomgeving.

Is de taal geschikt om te draaien op een embedded systeem, rekening houdende met de beschikbare rekenkracht op embedded systemen?

Een CLAIM-agent kan alleen draaien wanneer het SyMPA platform actief is. Het gaat hier om een volwaardige desktop Java container, dit zal functioneel zijn op de grotere Cortex-A architectuur, de kleinere Cortex-R en Cortex-M zullen de benodigde container niet altijd goed laten draaien.

Is de community van de taal levendig genoeg om support te verlenen?

Voor zover het exploratieve onderzoek toe heeft gestaan, is er geen community gevonden. Enkel via de Google Scholar zoekmachine, IEEE-website, de zoekmachine van de HU genaamd Lucas, en de databank ScienceDirect zijn publicaties gevonden met betrekking tot de taal. Het is daarom mogelijk om de nodige informatie over de taal te verzamelen, echter is het onwaarschijnlijk dat er steun verleend kan worden door fora.

Jadex

Jadex is voortgebouwd op Jade en kan gezien worden als extensie. Het biedt een van de FIPA-pakketten ingebakken en wel, zodat direct gebruik gemaakt kan worden van XML berichten onderling versturen tussen agents. Jadex biedt agents de mogelijkheid gebruik te maken van het BDI-principe, en draaien geheel op zogenaamde events. De agent neemt events waar, redeneert aan de hand van zijn BDI's welke actie ondernomen dient te worden en handelt accorderend. Deze evenementen komen in drie verschillende vormen. Bericht events, goal events en interne events. Entiteiten die het BDI-model van een agent voorstellen zijn zeer plat opgebouwd. Het zijn enkele feiten opgeslagen in de entiteit zelf, waardoor het gemakkelijk is om deze te doorlopen tijdens besluitvorming. Wanneer een agent wordt aangemaakt, moet hierbij gebruik gemaakt worden van een XML-schema.



Figuur 4 - Jadex logo. Bron: Universität Hamburg

Een Hello World applicatie ziet er erg ingewikkeld uit, omdat alle pakketten opgestart en geïntantieerd dienen te worden. Denk hierbij aan zaken als de Jadex Starter welke verantwoordelijk is voor het opbouwen van een agent. Wanneer deze opstartcode versimpeld wordt, komt het volgende voorbeeld:

```
public static void main(String[] args){
    Starter.createPlatform(args).addResultListener(
        new DefaultResultListener<IExternalAccess>(){
            public void resultAvailable(final IExternalAccess{
                System.out.println("Hello World!");
            }
        });
}
```

Is de taal geschikt om te draaien op een embedded systeem, rekening houdende met de beschikbare rekenkracht op embedded systemen?

Na een exploratief onderzoek om naslagwerken te verzamelen, is gebleken dat op moment van schrijven Jadex nog niet gebruikt is op mobiele platformen. Aangezien Jadex draait op Jade en hier extra functionaliteit aan toevoegt, onder meer het pakket om gebruik te maken van het BDI-principe, is de verwachting dat het op de Cortex-A architectuur haalbaar is om gebruikt te worden. Na een Raspberry Pi 3 klaar te hebben gemaakt voor een test, lijkt de volledige Hello World applicatie zoals in vorige alinea besproken te functioneren. Echter biedt dit geen garantie voor uitgebreidere applicaties, waarbij meer rekenkracht gebruikt kan worden.

Is de community van de taal levendig genoeg om support te verlenen?

Via <https://sourceforge.net/p/jadex/discussion/> is het mogelijk om Jadex gerelateerde vragen te stellen. Op moment van schrijven hebben de afgelopen 24 uur drie hulp topics een reactie gekregen. Het gemiddelde aantal views binnen dezelfde periode is afgerond 20. Tevens is er een open discussie mogelijk met de developers, bijvoorbeeld om nieuwe features aan te vragen of bugs te rapporteren. De community is aanwezig, tevens wordt het framework actief doorontwikkeld.

Framework

Met deze resultaten in het achterhoofd, kan er een keuze gemaakt worden. Ter illustratie de volgende tabel, waarbij met embedded maximaal een Cortex-A architectuur bedoeld wordt. Per criteria kan de taal 0 tot en met 2 scoren, waarbij 0 niet compatibel of geen ondersteuning betekent en 2 goede compatibiliteit of goede ondersteuning.

	Embedded compatibel?			Ondersteuning/community?			Totaalscore
	0	1	2	0	1	2	
AgentSpeak			X	X			2
Jason		X			X		2
JaCa		X			X		2
Jade		X				X	3
3APL			X	X			2
CLAIM	X			X			0
Jadex	X					X	2

Tabel 1 - Score programmeerframework.

Zoals uit tabel 1 op te maken valt, is de uiteindelijke keuze op basis van de criteria gevallen op Jade.

Communicatiestandaard

De resultaten die aangeboden worden in dit hoofdstuk zijn gekoppeld aan de volgende deelvraag:

Welke standaard kan gebruikt worden om agents in een domotica-omgeving onderling te laten communiceren?

Om goed uit te zoeken welke standaard geschikt is voor het uitwisselen van informatie, is het nodig om eerst vast te stellen welke informatie er gedeeld moet worden. Dit wordt gedaan middels een subvraag.

Uit te wisselen informatie

Welke informatie willen agents in een domotica-omgeving uitwisselen met elkaar? Om een antwoord te geven op deze vraag, is een interview met Leo van Moergestel gehouden. Alle volgende informatie is gebaseerd op het interview dat gehouden is met Leo van Moergestel (persoonlijke mededeling Leo van Moergestel aan Thomas Kok, 30 jun 2016).

Na het interview met Leo van Moergestel is gebleken dat agents vrijwel iedere denkbare informatie moeten kunnen uitwisselen. Daarom is het belangrijk om agents niet te beperken in hun communicatieve vaardigheden, maar ze juist te faciliteren. Een van de mogelijke oplossingen hiervoor heeft Van Moergestel zelf gebruikt in een van zijn projecten, namelijk de communicatiestandaard Agent Communication Language (ACL) zoals vastgesteld door de Foundation for Intelligent Physical Agents (FIPA). Meer over het project van de heer Leo van Moergestel in de volgende deelvraag "Soortgelijke projecten". Volgens Van Moergestel is deze standaard probleemloos in te zetten voor domotica, daar het een reeks regels en standaarden betreft. Deze kunnen in worden gezet in welke programmeertaal dan ook. Het programmeerframework Jade biedt ondersteuning voor ACL, een eigenschap die mee heeft gespeeld in de keuze voor dit framework. Meer over deze keuze in hoofdstuk "Agent Oriented Programming frameworks". Ook stelt Van Moergestel dat het middels ACL mogelijk is om op meerdere manieren een bericht op te bouwen. In zijn onderzoek zijn vier manieren onderzocht, waaruit is gebleken dat het gebruik van JSON om een ACL bericht op te bouwen optimaal is (Moergestel, 2013). Tot zo ver de informatie gebaseerd op de persoonlijke mededeling.

Uit verder exploratief onderzoek is gebleken dat er slechts één andere taal is met enige significantie, beschreven door Tim Finin in zijn voortkoming "KQML as an agent communication language" van de derde internationale conferentie Information and knowledge management. KQML is na 2008 in de publicatie van Miroslav Prymek niet meer opgenomen in voortkomingen uit conferenties of publicaties.

Is de standaard gebonden aan een set programmeertalen of niet?

De FIPA ACL standaard is te implementeren in welke programmeertaal gewenst is. Er is door FIPA een standaard opgesteld, welke door de programmeur te implementeren is in de taal van keuze. Echter zijn er kant en klare pakketten waar de standaard reeds in geïmplementeerd is, bijvoorbeeld voor JSON en XML. Voor beide pakketten is ondersteuning beschikbaar, met de laatste reactie in een Stackoverflow post minder dan 24 uur geleden op moment van schrijven.

De KQML-standaard is te implementeren in welke programmeertaal gewenst is. Er is een standaard opgesteld welke dicteert hoe een bericht opgebouwd moet zijn, maar er is slechts één kant en klaar pakket beschikbaar, KQML Java. Dit pakket is gedocumenteerd, echter is er geen ondersteuning voor beschikbaar. Het is mogelijk om zelf een pakket te produceren voor een andere programmeertaal, echter is dit een tijdrovend proces (Moergestel, 2013).

Biedt de standaard de mogelijkheid om alle informatie te communiceren die een domotica-agent zou willen uitwisselen?

Zowel de FIPA-standaard als de KQML-standaard beschrijven enkel informatie die voor zender en ontvanger belangrijk zijn om elkaar te identificeren en duidelijk te krijgen wat de intentie is van het bericht. Verder is het mogelijk om deze berichten aan te vullen met key-value paren, om gewenste informatie over te brengen.

De eerste subvraag geeft een doorslaggevend resultaat qua voorkeur voor een communicatiestandaard. De FIPA-standaard heeft meerdere kant en klare pakketten om de standaard over meerdere programmeertalen direct te implementeren, wat het mogelijk maakt om andere devices die niet communiceren middels JSON of XML, alsnog gemakkelijk te kunnen koppelen aan het systeem.

Soortgelijke projecten

De resultaten die aangeboden worden in dit hoofdstuk zijn gekoppeld aan de volgende deelvraag:

In welke soortgelijke toepassingen wordt Agent Oriented Programming al toegepast?

Na een exploratief onderzoek te hebben uitgevoerd, is slechts één project opgedoken dat naar hetzelfde doel van modulaire compatibiliteit werkt en daarvoor waardevol is om te onderzoeken, mede door de persoonlijke mededeling van Leo van Moergestel is dit onderzoek aan het licht gekomen. Dit project is gedocumenteerd in de publicatie "Agents in domestic environments", en zal besproken worden in het volgende hoofdstuk.

Agents in domestic environments

Een publicatie door een team onderzoekers van de Universiteit Utrecht, met als hoofdauteur Leo van Moergestel, waarin een mogelijke architectuur beschreven wordt voor agent-based domotica-omgevingen.

Het onderzoek van Leo van Moergestel is uitgevoerd om een agent-gebaseerde architectuur te beschrijven voor domotica.

Welke aspecten van dit project zijn toepasbaar op dit onderzoek?

Het onderzoek van de heer van Moergestel sluit op meerdere fronten aan op dit onderzoek. Op pagina 2 van de publicatie wordt het volgende gesteld:

"A device in a domotica system is acting in an environment and its actions are influencing that same environment. The devices should have the possibility to communicate and cooperate to achieve system wide goals. Looking at these requirements and properties of the devices, they are also found in the definitions of agents for example the definition given by Wooldrigde (Wooldrigde, 2009). "

Hierin vinden we de bevestiging dat de richting van Agent Oriented Programming de goede is.

Ook is in dit onderzoek vastgesteld dat de Raspberry Pi een goede keuze is voor de hardware, waarbij een goede balans is tussen prijs en de beschikbare rekenkracht, aangezien de Raspberry Pi een erg goedkoop hardware platform biedt. Een gedetailleerdere uitleg over de keuze voor een Raspberry Pi in het hoofdstuk voor het proof of concept. De aanwezige rekenkracht is zeer belangrijk in de keuze voor programmeertaal, waarbij het resultaat van deelvraag 1 bevestigd wordt. Ook maakt het onderzoek van de heer Leo van Moergestel et al de keuze voor een Raspberry Pi om het proof of concept in op te bouwen een zeer aantrekkelijke keuze (Moergestel, 2013).

Van Moergestel heeft gedurende zijn onderzoek gesteld dat Jade het te gebruiken programmeerframework is (Moergestel, 2013). Een opsomming van de redenering achter deze keuze:

- Jade biedt multi-agent-based systemen een aantal kant en klare oplossingen voor onder meer communicatie en andere requirements voor een MAS.
- Jade is gebaseerd op Java, een zeer breed inzetbare en krachtige programmeertaal.
- Aangezien Jade op Java gebaseerd is, biedt het een makkelijke leercurve voor Java programmeurs.
- Agents kunnen migreren, termineren, en er kunnen nieuwe agents ontstaan.
- Indien nodig, kan het platform gemakkelijk geüpgraded worden naar bijvoorbeeld Jadex.

Bijkomend voordeel aan het Jade framework is dat het ingebouwde support heeft voor de FIPA-standaard. Mede hierdoor, vinden we de bevestiging dat de keuze voor Jade de goede is.

Domotica-gateways

De resultaten die aangeboden worden in dit hoofdstuk zijn gekoppeld aan de volgende deelvraag:

Wat wordt verstaan onder een domotica-gateway en welke alternatieven zijn hiervoor?

Een domotica-gateway is een apparaat dat sensoren en actuatoren middels een protocol met elkaar kan verbinden, met het oog op een domotica-omgeving op te zetten. Deze apparaten zijn speciale hardware puur en alleen gericht op de communicatie, maar bieden zelf geen verdere functionaliteit. Een voorbeeld van een dergelijk apparaat is de Popp Hub (Popp & Co, 2015). Dit apparaat is geschikt voor enkel het Z-Wave protocol, een protocol dat niet relevant is voor dit onderzoek en daarom verder niet behandeld wordt, waardoor er relatief veel apparaten beschikbaar zijn maar nog steeds een beperkt aanbod. De controller van Popp, alsmede die van andere merken, hebben allen het probleem dat ze te weinig protocollen ondersteunen wat de compatibele apparaten erg beperkt. Daarom is het beter om voor een alternatief te kiezen, om zowel de compatibiliteit met andere apparaten als interoperabiliteit te vergroten.



Figuur 5 - Popp Hub. Bron: Popp & Co

Op het moment van schrijven bestaat er nog geen out-of-the-box alternatief voor domotica-gateways. Daarom wordt een analyse gedaan van de bestaande standaard Zigbee welke is opgenomen in de IEEE-norm 802.15.4. Wat Zigbee interessant maakt, is de manier waarop een netwerk op dit protocol tot stand komt.

Een Zigbee netwerk is opgebouwd uit drie verschillende nodes. Allereerst is er de coördinator node die het netwerk operationeel maakt. Vervolgens sluiten routers zich bij het netwerk aan, welke het mogelijk maken om theoretisch gezien maximaal 65536 nodes totaal te verbinden en packets te routeren. Deze maximale waarde is een direct resultaat van de wijze waarop nodes elkaar adresseren. Zij doen dit middels een 16 bits adres, in 16 bits bestaat de ruimte om 65536 unieke waarden mee te creëren. Het unieke aan Zigbee is dat het protocol eenieder apparaat toestaat om zowel als coördinator, router, en als node te laten fungeren.

Ieder component van de domotica-omgeving zou namelijk kunnen optreden als zo'n gateway. Agents gedragen zich dan als een node in een Zigbee netwerk, waarbij tijdens opstarten op zoek wordt gegaan naar een bestaand netwerk. Indien er een netwerk is, dan sluit de agent zich hierbij aan. Indien er nog geen netwerk is, zet dit op en treed op als coördinator.

Kort samengevat kan het principe van Zigbee worden ingezet als een alternatief voor een speciaal aangewezen gateway, waarbij ieder component deze rol op zich kan nemen of gewoonweg bij het netwerk aansluiten. Na een kort exploratief onderzoek te hebben uitgevoerd, is gebleken dat een dergelijk project nog niet eerder is uitgevoerd. Hier zijn op moment van schrijven nog geen standaarden voor, wat inhoudt dat het proof of concept de geweldige mogelijkheid heeft om hier een nieuwe standaard voor te definiëren. Het gebruik van dit concept houdt in dat ieder domotica device uitgerust moet worden met geschikte hardware om de programmeertaal van keuze te faciliteren van voldoende rekenkracht.

Inputprioriteit

De resultaten die aangeboden worden in dit hoofdstuk zijn gekoppeld aan de volgende deelvraag:

Hoe dient een domotica-agent prioriteit toe te kennen aan zijn verscheidene inputs?

Interview Leo van Moergestel

Om antwoord te vergaren op deze vraag, is een interview gehouden met Leo van Moergestel. De volgende informatie is gebaseerd op het interview dat gehouden is met Leo van Moergestel (persoonlijke mededeling van Leo van Moergestel aan Thomas Kok, 30 juni 2016).

Wanneer het aankomt op priorisering van communicatie tussen agents, is er eigenlijk vrijwel geen onderzoek gedaan naar welke inputs de grootste input dienen te krijgen. Het is daarom per project afhankelijk welke prioriteit er aan inputs wordt gegeven, waarbij gekeken moet worden naar de situatie van de agents en welke rol zij vullen. Wanneer er wordt gekeken naar een deurslot, is het vrij duidelijk dat een agent van de thermostaat hier niet bij mag. Gezond verstand is de richtlijn.

Naar aanleiding van deze informatie zal er voor het proof of concept, waarover meer in het hoofdstuk "Op te leveren producten", een prioriteitssysteem in gebruik worden genomen waarbij fictionele waarden worden gebruikt om een agent hogere prioriteit toe te kennen over andere agents ten opzichte van elkaar, om te bewijzen dat het mogelijk is om prioriteit toe te kennen. Er is voor fictionele waarden gekozen omdat er volgens Van Moergestel meer onderzoek uitgevoerd moet worden naar priorisering, waardoor momenteel per project een besluit moet worden gemaakt met betrekking hiertoe.

Veiligheidsaspecten

De resultaten die aangeboden worden in dit hoofdstuk zijn gekoppeld aan de volgende deelvraag:

Met welke veiligheidsaspecten hebben agents in een domotica-omgeving rekening te houden?

Interview Wouter van Rooij

Om tot een antwoord op deze deelvraag te komen, is er een interview met Wouter van Rooij gehouden (persoonlijke mededeling van Wouter van Rooij aan Thomas Kok, 16 juni 2016).

Wanneer naar de veiligheid van een systeem wordt gekeken, zijn er negen aspecten waar rekening mee gehouden moet worden en waar in black box testen op wordt gecontroleerd volgens Wouter van Rooij (persoonlijke mededeling, 16 juni 2016). Daarbij is het irrelevant of het nu een systeem van agents is, een webservice, een interne applicatie of wat voor architectuur dan ook. Dit template is op iedere situatie van toepassing aldus de heer van Rooij van Sogeti (persoonlijke mededeling, 16 juni 2016). De speerpunten van het template en de uitleg per speerpunt zijn allemaal opgesteld door het interview van Wouter van Rooij te gebruiken als referentie:

- Clientside controls
- Authenticatie
- Sessiemangement
- Toegang en de controle hiervan
- Inputvalidatie
- Functie input
- Logica fouten
- Omgeving
- Server

Per speerpunt zal er enige uitleg worden verschaft.

Clientside controls

Een groot aspect van veiligheid berust op de veiligheid van de computer van de client of het systeem waar de applicatie op draait. Daarbij ligt de grootste focus op de werking van scripts die op de client draaien en met name de controle op hun functionaliteit en werking. Een makkelijke manier om vast te stellen dat een client systeem naar behoren functioneert en veilig is, is door een zogenaamd security policy in te stellen.

Authenticatie

Om bij gebruik van een applicatie vast te stellen dat de gebruiker ervan daadwerkelijk geautoriseerd is om handelingen uit te voeren, dient de gebruiker een authenticatieprocedure te doorlopen. Het veiligheidsaspect authenticatie is er puur en alleen op gericht om te controleren dat de module die authenticatie regelt goed functioneert. Een voorbeeld van een authenticatieprocedure die als veilig wordt beschouwd, is een combinatie tussen een gebruikersnaam en wachtwoord met een bevestigingscode bijvoorbeeld per sms. Een dergelijke procedure wordt tweetrapsauthenticatie genoemd.

Sessiemangement

Indien een applicatie functioneert met een sessie, is dit een onderdeel om rekening mee te houden. Waarop wordt de sessie gebaseerd? Hoe wordt de sessie onderhouden? Vindt er sessieoverdracht plaats? Dit zijn slechts een paar vragen waar rekening mee gehouden moet worden wanneer er wordt gekeken naar sessiemangement. Daarom dient er, indien er gebruik gemaakt wordt van sessies, een module aanwezig te zijn welke de sessies managet en updatet waardoor er hier geen risico ontstaat.

Toegang en de controle hiervan

Bij vrijwel iedere applicatie heeft de gebruiker een optie om in te loggen. Dit inloggen heeft hoofdzakelijk als doel voor ogen om onderscheid te maken tussen gebruikers, waarbij gemakkelijk functionaliteit beschikbaar gemaakt kan worden voor gebruikers die hier recht toe hebben of er gebruik van moeten maken. Echter is dit een zeer belangrijk aspect van de beveiliging. Er moet zeer bewust om worden gesprongen met de rechten en functionaliteiten die een applicatie aan zijn gebruikers geeft. In het geval van een domotica-systeem is het voor een eindgebruiker die slechtst geïnteresseerd is in het openen van zijn deur niet bijzonder interessant om te weten hoeveel cycli het slot al doorlopen heeft. Dit wordt pas relevant op het moment dat het slot aan onderhoud toe is, eerder niet. Deze informatie, en natuurlijk alle andere beschikbare informatie, kan gemakkelijk afgeschermd worden voor de gebruiker door de toegang te controleren aan de hand van een inlog.

Inputvalidatie

Wanneer de gebruiker ergens in een applicatie zelf input mag aanleveren, is het cruciaal dat deze informatie fatsoenlijk afgehandeld wordt. Het is bijvoorbeeld mogelijk om een zogenaamde SQL-injectie aanval uit te voeren. Volgens de beschrijving op de website van W3schools (2016), is een SQL-injectie simpelweg slim omgaan met beschikbare input van een gebruiker die in een dynamische query wordt geplaatst waarbij het mogelijk is om de uitkomst van een query te beïnvloeden. Omdat op deze wijze een gebruiker bij gegevens uit een database kan komen die niet voor hem bedoeld zijn, is het cruciaal om de gebruikersinput goed te "escapen". Dat houdt in dat een puntkomma bijvoorbeeld wordt vervangen door zijn HTML-entiteitsnummer, 59. Op deze manier zal een server "&59" als input in een query zien, in plaats van een puntkomma.

Functie input

Vervolgens is er de functionele inputvalidatie. Bij deze validatie dient er gelet te worden op de wijze waarop input gecontroleerd wordt. In een situatie waarbij de inputvalidatie controleert dat een puntkomma goed wordt verwerkt, is de functionele inputvalidatie de module die controleert dat er daadwerkelijk volledig en compleet gevalideerd wordt. Functionele inputvalidatie kan dus gezien worden als een overkoepelende inputvalidatie

Programmatuurlogica

Om de veiligheid te vergroten, is het uiteraard nodig om de logica van de programmatuur te controleren. Zit er een fout in de logica, of werkt deze in bepaalde scenario's niet naar verwachting? Het is de bedoeling dat iedere module met logica gevalideerd wordt, om veilig te stellen dat er geen onverwachte resultaten zijn. Thomas Kok is tijdens zijn oriënterende stage verteld door zijn toenmalige begeleider, Pepijn Sitter, dat hiervoor Unit Testing gebruikt kan worden (persoonlijke communicatie, september 2014 - februari 2015). Unit Testing is een wijze van programmeren waarbij iedere functie als een unit wordt gezien, waarbij iedere unit gevalideerd wordt door een test.

Omgeving

Met de omgeving van een applicatie wordt de fysieke omgeving bedoeld. Waar staat het systeem? Is dit een publieke omgeving, waar mensen rondlopen die geen directe gebruikers van het systeem zijn? In het geval van een publieke omgeving, is het belangrijk om duidelijk gebruikers af te bakenen en ervoor te zorgen dat alleen die gebruikers de mensen zijn die gebruik kunnen maken van een systeem. De vergelijking is gemakkelijk getrokken met een geldautomaat. Het gaat dan om een apparaat wat in een zeer openbare omgeving staat, waarbij het de bedoeling is om één persoon door middel van een fysieke kaart en een pincode te autoriseren in het systeem. Per toepassing of project is het goed om na te denken over welke omgeving de applicatie terecht gaat komen, om op die manier van tevoren vast te stellen welke maatregelen hiertoe getroffen dienen te worden.

Server

Uiteraard maakt ook de server uit van het veiligheidsgeheel. Het allerbelangrijkst is de configuratie van de server. Een hele simpele aanval die een groot veiligheidsrisico met zich meebrengt, is de zogenaamde poortscanningaanval. Bij deze aanval wordt door een programma iedere mogelijke poort van een server afgetast, om op deze wijze te achterhalen welke poorten in gebruik zijn en dus mogelijk te kraken zijn. Echter is het op deze wijze voor de eigenaar/beheerder van een server ook mogelijk om te achterhalen waar een lek in de veiligheid van de server zit.

Wanneer bovenstaande negen aandachtspunten rekening wordt gehouden tijdens het ontwikkelen, is het met zekerheid te zeggen dat een applicatie veilig is en geen veiligheidsrisico's heeft.

Volgens Wouter van Rooij is het voor dit onderzoek met name van belang om de focus te leggen op toegangs- en inputvalidatie (persoonlijke mededeling van Wouter van Rooij aan Thomas Kok, 16 juni 2016). Op deze wijze is het zeker te stellen dat slechts de bewoners van het huis met de domotica devices toegang hebben tot alle apparaten aanwezig in hun woning. De inputvalidatie zal er zorg toe dragen dat een gebruiker niet per ongeluk of expres het systeem ongedefinieerd gedrag kan laten vertonen. Andere aspecten van de beveiliging zijn over het algemeen opgenomen in andere lagen van het OSI-model waar geen focus op ligt in dit onderzoek.

Literatuuronderzoek

Uit het literatuuronderzoek is gebleken dat onder meer Jacobsson et al (2016) uitvoerig onderzoek heeft uitgevoerd in de richting van risicofactoren/veiligheidsaspecten. In de daaruit voortgekomen publicatie "*A risk analysis of a smart home automation system*", worden de resultaten van een risicoanalyse beschreven op domotica apparaten met een verbinding met het internet. Aanleiding voor dit onderzoek is het zogenaamde Smart Home Automation System (SHAS), wat gedurende zijn ontwikkeling gaandeweg de beveiliging adequaat wilde maken en de risico's helder te krijgen. Het doel van SHAS ligt in lijn met het doel van dit onderzoek: systemen van verschillende fabrikanten met elkaar laten samenwerken. In het geval van SHAS is dit samenwerken om informatie real-time uit te wisselen om een gebruiker inzicht te geven in het stroomverbruik (Jacobsson et al, 2016).

Gedurende het onderzoek van Jacobsson et al werden 32 mogelijke risico's onderzocht. Uit dit onderzoek is gebleken dat slechts vier van de onderzochte risico's geclassificeerd werden als een uitermate hoog risico. Deze risicofactoren hadden allen te maken met de menselijke factor of de software. Van de resterende risico's werden negen geclassificeerd als een laag risico, het restant als middelmatig.

Deze 32 risico's werden in vijf groepen gecategoriseerd met betrekking tot aard van het risico (Jacobsson et al, 2016). Deze groepen zijn software, hardware, informatieverwerking, communicatiekanalen, en gebruikersrisico's. Qua software stellen de onderzoekers vast dat de grootste dreigingen komen uit de applicatie en de API van het SHAS, beiden in het authenticatieveld. Qua hardware stellen de onderzoekers vast dat het enige risico in verlies/diefstal van hardware ligt.

Wanneer naar de informatie van het interview wordt gekeken, en vergeleken wordt met het onderzoek van SHAS (Jacobsson et al, 2016), is af te leiden dat de focus op de volgende drie aspecten moet liggen:

- Toegangsvalidatie
- Inputvalidatie
- Logicavalidatie

Logicavalidatie is het derde aspect met betrekking tot de gevonden risico's in de software en API-omgeving, zoals reeds vastgesteld (Jacobsson et al, 2016).

Conclusie

Hoofdvraag

Sogeti is hard op zoek naar een standaard binnen de domotica wereld. Zo wil het bedrijf ervoor zorgen dat producten niet meer fabrikant gebonden zijn als het komt op aansturing. Om tot deze standaard te komen is Thomas gevraagd hiernaar onderzoek te verrichten. Voor de uitvoer van dit onderzoek is de volgende hoofdvraag opgesteld:

"Hoe kan Agent Oriented Programming ingezet worden om de interoperabiliteit van connected devices in domotica-omgevingen zonder gateways te verbeteren?"

De vergaarde kennis in de vorm van antwoorden op de deelvragen, geeft een beeld voor een communicatieprotocol wat domotica-agents in staat stelt met elkaar te communiceren zonder hiervoor een gateway te plaatsen.

Jade

Allereerst de programmeertaal. Uit de resultaten van dit onderzoek is gebleken dat Jade de beste keuze is om het protocol te implementeren. Achter deze taal schuilt een levendige community welke bereid is om naar een probleem te kijken en tot een oplossing te komen. Tevens biedt Jade de mogelijkheid om gebruik te maken van zowel een ethernetkabel als draadloze communicatie om agents aan een netwerk te verbinden, door de aanwezigheid van de netwerkmanagementtool. Andere programmeertalen zijn mogelijk indien gebruik wordt gemaakt van dezelfde netwerkstructuur zoals beschreven in de alinea "Zigbee structuur", en deze taal gebruik maakt van de FIPA ACL standaard.

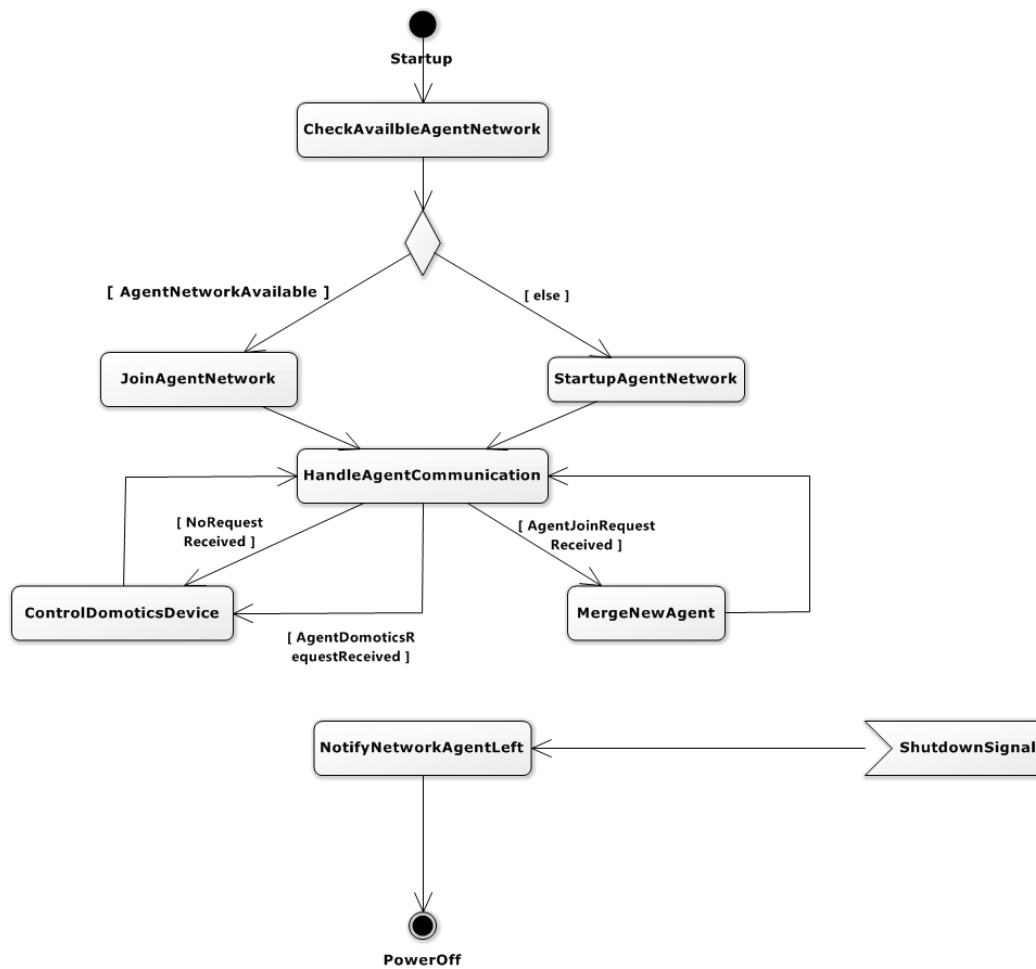
FIPA ACL

Om agents te laten communiceren, is de keuze gemaakt voor de standaard van FIPA, namelijk Agent Communication Language. Zoals in de resultaten van de deelvraag communicatiestandaard aan bod is gekomen, is de FIPA-standaard het meest gebruikte protocol. Deze keuze is gemaakt met het oog op samenwerking en interoperabiliteit. Deze standaard biedt de mogelijkheid om iedere wenselijke informatie uit te wisselen, waardoor een agent hier geen beperkingen door ondervindt. Tevens is het uitwisselen van een FIPA ACL bericht niet gebonden aan een programmeertaal. De standaard beschrijft enkel de opbouw van een bericht, het dicteert niet in welke taal dit geschreven dient te zijn. Door gebruik te maken van de MTP-module, ingebouwd in Jade, is het mogelijk om zelfs domotica-agents welke niet in Jade geprogrammeerd zijn te laten communiceren met domotica-agents die dit wel zijn.

Zigbee structuur

Een groot obstakel van domotica is het buiten spel zetten van controllers en gateways, het doel van dit onderzoek. Het belangrijkste aspect van het antwoord op de hoofdvraag rust dan ook in de kennis die naar voren is gekomen uit de deelvraag over domotica-gateways. De netwerkstructuur door Zigbee gehanteerd en geïmplementeerd in de 802.15.4 standaard is te gebruiken in de 802.11 standaard. Dit is te realiseren door agents over ethernet en wifi met elkaar in verbinding te zetten waarbij onderling een eigen netwerk wordt onderhouden waarover gecommuniceerd wordt. Dit wordt gedaan door een nieuw aangesloten domotica-device zijn netwerkomgeving te laten controleren op een bestaand agentnetwerk. Wanneer een netwerk wordt aangetroffen, zal de domotica-agent zich hierbij aansluiten. Indien er geen

agentnetwerk wordt aangetroffen, zal de domotica-agent een nieuw netwerk opzetten en op deze wijze de taak van controller en gateway aan zichzelf toe-eigenen om hierdoor werkelijk onafhankelijk te worden. Deze structuur is weergegeven in het STD uit figuur 6.



Figuur 6 - Communicatieprotocol. Eigen afbeelding.

Communicatieprotocol

Uit deze kennis volgt een antwoord op de hoofdvraag. Om interoperabiliteit van domotica-devices te verbeteren is het allereerst nodig om controllers en gateways, apparaten zoals beschreven in het hoofdstuk "Domotica-gateways", overbodig te maken. Dit is te bereiken door agentnetwerken op te zetten middels het Zigbee principe, zoals in de vorige alinea beschreven en weergegeven in Figuur 6. Dit is te beschrijven in het volgende stappenplan:

- Een agent start op en controleert zijn netwerkomgeving op reeds aanwezige agentnetwerken
 - o Indien aangetroffen, sluit de agent zich aan bij dit netwerk
 - o Indien geen agentnetwerk aangetroffen, zal de agent zelf optreden als coördinator en toegangspunt voor nieuwe agents.
- De agent stuurt het domotica-device aan
- De agent gaat onderlinge communicatie afhandelen.
 - o Indien een aansluitverzoek ontvangen is, wordt de agent aangesloten op het netwerk
 - o Indien een agent een domotica-gerelateerd verzoek heeft ontvangen, wordt dit afgehandeld volgens het BDI-principe.
 - o Indien een agent geen verzoek heeft ontvangen, zal het op basis van zijn BDI-principe een actie voor zichzelf creëren.

Om agents met elkaar te laten communiceren, zal in het communicatieprotocol gebruik gemaakt worden van de FIPA ACL standaard. Op deze manier is het mogelijk voor domotica devices welke niet geprogrammeerd zijn met Jade, om toch deel te nemen aan het netwerk met minimale aanpassing van de broncode zolang berichten de FIPA-standaard hanteren. Hiervoor kan gebruik worden gemaakt van de MTP-module ingebouwd in het Jade framework, zodat een FIPA ACL bericht over HTTP verzonden kan worden naar andere programmeertalen.

Kort samengevat bestaat het communicatieprotocol uit een richtlijn voor de programmeertaal van de agent. Indien deze programmeertaal niet gehanteerd kan worden is het mogelijk om de agent met minimale broncode aanpassingen de agent alsnog te laten deelnemen aan het netwerk, mits hierbij gebruik wordt gemaakt van de FIPA ACL standaard. Vervolgens beschrijft het communicatieprotocol een wijze van agentnetwerk opzetten, waarbij het principe van het Zigbee protocol wordt gebruikt. Dit houdt in dat ieder domotica device tijdens opstarten zijn netwerkomgeving controleert op de aanwezigheid van agentnetwerken. Indien een agent aanwezig, zal de agent zich hierbij aansluiten. Indien niet aanwezig, zal de agent een nieuw netwerk opzetten en andere agents in de gelegenheid stellen om zich bij dit netwerk te voegen. Deze manier van een netwerk opzetten is de oplossing voor het probleem wat domotica-gateways met zich meebrachten. Apparaten gemaakt met dit protocol zijn niet afhankelijk van een gateway; zij kunnen deze functionaliteit zelf voor hun rekening nemen. Tot slot beschrijft het protocol dat de agents dienen te communiceren middels de FIPA ACL standaard.

Proof of concept

Aan de hand van de resultaten uit het onderzoek, zal een opstelling worden gemaakt waarin de werking van het communicatieprotocol wordt aangetoond.

Hardware & scenario's

Voor het realiseren van het proof of concept wordt er gebruik gemaakt van een drietal Raspberry Pi 3 B's alsmede een Android telefoon. Deze apparaten zullen in een netwerkomgeving op worden gezet om een aantal scenario's te demonstreren:

- Eén Raspberry Pi, geen reeds aanwezige domotica-agents. De nieuwe agent zet een netwerk op.
- Twee Raspberry Pi's, één reeds aanwezige domotica-agent. De nieuwe agent voegt zich bij dit netwerk.
- Twee Raspberry Pi's, één reeds aanwezige domotica-agent. De agents demonstreren de aansturing van een led.
- Twee Raspberry Pi's, twee reeds aanwezige domotica-agents. De agents demonstreren het uitwisselen van een bericht en aan de hand van het BDI-principe wordt bepaald of de agent de aanwezige led aanstuurt of niet.
- Eén Raspberry Pi en één Android telefoon, twee reeds aanwezige domotica-agents. De agents demonstreren het uitwisselen van een bericht middels de MTP-module.

Tevens zal er een systeem geprogrammeerd worden om prioriteit toe te kennen aan verscheidene inputs. Hiermee wordt gedemonstreerd dat een agent de keuze heeft om berichten van een bepaalde agent voorrang te verlenen over berichten van een andere agent of de keuze heeft hierin zijn geheel niet op in te gaan.

Werking proof of concept bundel

In overleg met de opdrachtgever en docentbegeleider is besloten om het tot stand doen komen van het proof of concept uit te stellen. Reden voor dit besluit is de beperkte tijdsduur die beschikbaar was voor dit onderzoek. Een volledig uitgewerkt proof of concept om de beschreven scenario's uit te kunnen voeren zal tijdens de afstudeerzitting worden overhandigd aan de gegadigden.

Aanbevelingen & toekomstbeeld

Tijdens het uitvoeren van het onderzoek is zeer relevante kennis opgedaan. Voor de scope van dit onderzoek en de afstudeeropdracht is echter alleen gekeken naar het opzetten van een nieuw communicatieprotocol, echter biedt het protocol de gelegenheid om bijvoorbeeld een module te creëren welke te koppelen is aan bestaande domotica-devices om deze geschikt te maken voor integratie in een systeem dat gebruik maakt van dit protocol. Deze module maakt het in principe niet uit welke onderliggende hardware het mee communiceert, hier is een generiek interface afdoende voor.

Een dergelijke module zou het mogelijk maken om meer domotica apparaten met elkaar te laten communiceren dan nu mogelijk is. Een mooi toekomstbeeld: de kamerplant beschikt over een bloempot met de eerder vermelde module. De bloempot detecteert dat de kamerplant meer zonlicht nodig heeft. De bloempot communiceert dit aan de module, welke op zijn beurt op zoek gaat naar agents in de buurt met de mogelijkheid om zonlicht te creëren. Er wordt een agent gevonden die in verbinding staat met actuatoren om de gordijnen automatisch te laten openen. De agent van de bloempot stuurt een verzoek naar de agent welke verantwoordelijk is voor de gordijnen.

Tevens is uit het interview met Leo van Moergestel gebleken dat op moment van schrijven nog vrijwel geen onderzoek is uitgevoerd naar hoe een domotica-agent prioriteit moet toekennen aan zijn verscheidene inputs, waardoor er per case naar dit aspect onderzoek uitgevoerd moet worden. Zoals het toekomstbeeld van de kamerplant en de gordijnen. Zal de agent verantwoordelijk voor de gordijnen luisteren naar de agent verantwoordelijk voor de bloempot of niet? Het is een aanbeveling om per toepassing van het communicatieprotocol onderzoek uit te voeren hoe de prioriteit van berichten toegekend kan worden.

Bibliografie

- Asanti. (2016). *JaCa-Android*. Opgehaald van Sourceforge: <http://jaca-android.sourceforge.net/>
- Beal, V. (2016). *Moore's Law*. Opgehaald van Webopedia: http://www.webopedia.com/TERM/M/Moores_Law.html
- Finin, T. (1994). KQML as an agent communication language. *Proceedings of the third international conference on Information and knowledge management* (pp. 456-463). Gaithersburg: ACM.
- Hogeschool Utrecht. (2015). *Afstudeerleidraad Bacheloropleiding HBO-ICT 2015 - 2016*. Utrecht: Hogeschool Utrecht.
- Jacobsson, A. (2016). A risk analysis of a smart home automation system. *Future Generation Computer Systems*, 719-733.
- Moergestel, L. v. (2013). *Agents in Domestic Environments*. Utrecht: Universiteit Utrecht.
- Mombrea, M. (2013, september 26). *Android OS as an embedded platform*. Opgehaald van IT-World: <http://www.itworld.com/article/2833253/consumerization/android-os-as-an-embedded-platform.html>
- Popp & Co. (2015). *Popp Hub - POPP*. Opgehaald van Popp & Co: <http://www.popp.eu/products/controllers/popp-hub/>
- Prymek, M. (2008). Optimizing KQML for usage in power distribution network simulator. *Communications, Control and Signal Processing, 2008*. (pp. 846-849). St. Julians: IEEE.
- Rahwan, T. (2004). *Agent-based Support for Mobile Users using AgentSpeak(L)*. Melbourne: University of Melbourne.
- Schurr, N. (2005). *The Future of Disaster Response: Humans Working with Multiagent Teams using DEFACTO*. Los Angeles: University of Southern California.
- Singh, M. P. (1994). *Multiagent Systems: A theoretical framework*. Raleigh, North Carolina, Verenigde Staten: Springer-Verlag.
- W3school. (2016). *SQL Injection*. Opgehaald van W2schools: http://www.w3schools.com/sql/sql_injection.asp
- Wikia. (2008). *Domotics - Future - Wikia*. Opgehaald van Wikia: <http://future.wikia.com/wiki/Domotics>
- Wooldridge, M. (2007). *Programming multi-agent systems in AgentSpeak using Jason*. John Wiley & Sons .

Wooldrigde, M. (2009). *An Introduction to MultiAgent Systems, Second Edition*. Sussex, UK: Wiley.

Zachariah, T. (2015). *The Internet of Things Has a Gateway Problem*. Ann Arbor: University of Michigan.

Bijlagen

Bijlage A: Plan van aanpak



Student:

Thomas Kok, 1627579

thomas.t.kok@student.hu.nl

HBO-ICT, richting Technische Informatica

Begeleider:

Richard Verhaaf

richard.verhaaf@sogeti.com

PLAN VAN AANPAK

Agent Oriented Programming in domoticaomgevingen

Versiebeheer

Datum	Versie	Wijziging
07-03-16	0.1	Opzet document
14-03-16	0.2	Verdere invulling document
05-04-16	0.3	Onderzoek veranderd, document opnieuw ingevuld
14-04-16	0.4	Theoretisch kader aangevuld
03-05-16	0.5	Onderzoek gefinaliseerd, planning aangemaakt en risico's geanalyseerd
10-05-16	0.6	Na afwijzing vorig onderzoek, nieuw onderzoek opgezet
17-05-16	0.7	Planning toegevoegd
18-05-16	1.0	Laatste feedback verwerkt, afstudeercontract opgesteld

Lange Dreef 17
4131 NJ Vianen

+31 886 606 600

info@sogeti.nl
www.sogeti.nl

Inhoudsopgave

I.	Inleiding	3
II.	Context	4
	Bedrijf	4
	Werkwijze.....	4
III.	Opdracht	5
	Kwestie	5
	Formulering	5
	Opdrachttype.....	5
	Doelstelling.....	6
IV.	Onderzoek.....	7
	Hoofdvraag.....	7
	Theoretisch kader.....	7
	Deelvragen	9
	Aanpak.....	10
V.	Producten	11
	Onderzoeksrapport.....	11
	Proof of concept.....	11
	Scriptie.....	11
VI.	Projectplanning	12
	Methoden	12
	Planning	12
VII.	Projectorganisatie en risicoanalyse.....	13
	Communicatie	13
	Risicoanalyse	13
VIII.	Bedrijfs-/persoonsgegevens	15
IX.	Bibliografie	16
X.	Bijlagen.....	17

Inleiding

Bij iedere opleiding hoort een afstudeeropdracht. Zo ook bij Technische Informatica. Dit document heeft als doel om helder te stellen hoe Thomas Kok zijn vergaarde afstudeeropdracht bij Sogeti verwacht uit te voeren. Er zal meer informatie gegeven worden over de volgende aspecten:

- Het bedrijf Sogeti;
- De aanwezige begeleiding;
- Het onderzoek;
- De op te leveren producten;
- De planning voor het uitvoeren;
- Een risicoanalyse en een plan ter voorkoming van enkele risico's;
- Tot slot de gegevens van betrokken personen.

Door over deze onderdelen van het plan van aanpak na te denken alvorens aan de opdracht te beginnen, kan een beter beeld geschetst worden van de verwachtingen en op te leveren producten. Dit document heeft tevens als doel om de betrokkenen te voorzien van een beter idee van de werkzaamheden van de student die de opdracht uitvoert.

Bedrijf

Sogeti is een leverancier van ICT-diensten en onderdeel van Capgemini. Sogeti werd opgericht in 1967 door een Fransman, totdat het in 1975 fuseerde met CAP en Gemini Computer Systems tot CAP Gemini Sogeti. In 2002 werd Sogeti losgemaakt en geïntroduceerd als onafhankelijk bedrijf. Jaarlijks maakt Sogeti een omzet tussen de 200 en 300 miljoen, met in totaal meer dan 2500 medewerkers verdeeld over alle vestigingen wereldwijd. Sogeti is een relatief plat bedrijf. Organisatorisch gezien zijn er vier lagen, echter is het de bedoeling dat iedereen elkaar kan aanspreken en dit ook doet. Er heerst een relatief informele werksfeer met veel ruimte voor eigen input.

Sogeti detacheert haar werknemers naar bedrijven aan wie zij haar diensten verleent. Enkele grote partners waar op vaste basis medewerkers aan gedetacheerd worden zijn Microsoft, IBM, en banken zoals ING en Rabobank.

Het bedrijf is onderverdeeld in divisies en businesslines. Voor stagiairs en afstudeerders is er een aparte unit met een professionele begeleiding en coaches. Het bedrijf biedt ook toegang tot alle mogelijke bijscholing en cursussen. Onderdeel van deze cursussen is een scala aan certificeringen voor bijvoorbeeld Microsoft of Cisco.

Werkwijze

De werkwijze van Sogeti bestaat grotendeels uit het detacheren van haar medewerkers. Echter biedt zij ook mogelijkheden om binnen het bedrijf zelf projecten uit te voeren. Voor het uitvoeren van deze interne projecten beschikt Sogeti over een aantal eigen flexwerkplekvestigingen in onder andere Capelle aan de IJssel, Amersfoort en Amsterdam. Voor afstudeerders is er ruimte in deze vestigingen, alsmede alle andere medewerkers die behoefte hebben aan een werkplek wanneer het bedrijf waar zij gedetacheerd zijn bijvoorbeeld geen werkplekken beschikbaar hebben.

Deze werkwijze resulteert in een divers aanbod van kennis, variërende van dag tot dag. Uiteraard zijn er communicatiesystemen aanwezig zodat medewerkers elkaar kunnen contacteren zonder dat zij op dezelfde vestiging hoeven zijn. Skype wordt hier met name voor gebruikt, maar medewerkers reageren zeer goed en snel op e-mails.

Opdracht

Kwestie

Steeds meer bedrijven beginnen domotica apparaten te leveren. Toon, Nest, KlikAanKlikUit, Philips Hue, de lijst met belangrijke spelers blijft groter worden. Al deze producten zorgen voor een nieuwe dimensie van gebruikersgemak met de doelen die de producten vervullen. Echter hebben al deze apparaten momenteel hetzelfde probleem: ze zijn gebonden aan producten van dezelfde fabrikant. Maar wat als alle apparaten direct met elkaar kunnen communiceren, zonder dat zij van dezelfde fabrikant moeten zijn? Als een willekeurige smart televisie kan communiceren met je op afstand bestuurbare verlichting, dat zodra je een film op zet de verlichting hiervoor dimt?

Dit doel is exact wat Sogeti voor ogen heeft. Alle smart devices gemakkelijk met elkaar communiceren. Onderlinge consensus, bewustzijn van de mogelijkheden van de apparaten om zich heen. Daarom streeft Sogeti naar een algemene standaard voor communicatie tussen smart devices, zodat uiteindelijk het geen probleem meer is dat apparaten van verschillende fabrikanten zijn wanneer zij deze standaard implementeren. Het achterliggende programmeren is dan volledig aan de fabrikant.

Gedurende de afstudeeropdracht van Thomas zal er onderzocht worden hoe Agent Oriented Programming hiervoor gebruikt kan worden.

Formulering

Na enige gesprekken met Steven van Dalen is vastgesteld dat Sogeti de behoefte heeft aan meer duidelijkheid over hoe smart en connected devices betere interoperabiliteit kunnen bereiken. Sogeti wil de komende tijd hiervoor een standaard ontwikkelen en integreren in de industrie.

"Domotica is een ontzettend mooi veld. De mogelijkheden zijn eindeloos, maar momenteel binnen de beperkingen opgelegd door de fabrikant. Dit is absoluut niet wenselijk en Sogeti gaat hier verandering in brengen. Aan jou de opdracht om te onderzoeken hoe deze devices zich volgens een standaard kunnen gedragen die betere samenwerking met zich mee draagt, waarbij geen gateway vereist is. Demonstreer de werking van deze standaard middels een proof of concept. "

Na verdere gesprekken is gebleken dat de voorkeur voor een oplossing zonder gateways bestaat omdat er zo weinig mogelijk afhankelijkheden aan randapparatuur moeten zijn, om een werkelijke algemene standaard en interoperabiliteit te waarborgen.

Opdrachttype

De opdracht is een onderzoeksopdracht. Er zal onderzoek gedaan worden naar Agent Oriented Programming, om bovenstaande formulering te realiseren. Tevens zal er een proof of concept op worden geleverd om het resultaat te demonstreren. Meer over deze onderwerpen in het hoofdstuk over producten, hoofdstuk **Error! Bookmark not defined..**

Doelstelling

De afstudeeropdracht wordt uitgevoerd met een aantal hoofddoelen voor ogen:

- Het opleveren van een proof of concept,
- Het opleveren van een onderzoeksrapport waar de bevindingen in beschreven staan,
- Een onderzoek tot een goed einde leiden,

Ook zijn er nog enkele persoonlijke doelstellingen, zoals geformuleerd in bijlage B. Deze doelstellingen zullen geëvalueerd worden in een zelfreflectie en als bijlage bij de scriptie gevoegd worden.

Hoofdvraag

Uit de behoefte van Sogeti om meer smart en connected devices met elkaar te kunnen laten communiceren, met name apparaten die niet van dezelfde fabrikant afkomstig zijn, resulteerde in de volgende onderzoeksvraag:

Hoe kan Agent Oriented Programming ingezet worden om de interoperabiliteit van connected devices in domotica omgevingen zonder gateways te verbeteren?

Deze vraag valt onder een groot aantal technische velden, waaronder een programmeer principe genaamd Agent Oriented Programming, domotica en bijvoorbeeld Internet of Things. Om tot de te vullen gaten in kennis te komen, meer informatie over alle relevante onderwerpen in het theoretisch kader.

Theoretisch kader

In dit hoofdstuk zullen er een aantal belangrijke termen toegelicht worden, en middels een klein vooronderzoek uitgediept worden om zo vast te stellen waar de gaten in kennis op dit moment zich bevinden.

Agent Oriented Programming

Agent Oriented Programming is een programmeerstandaard die sinds 1989 zijn naam draagt (Shoham, 2010, hoofdstuk 13). Volgens Leo van Moergestel heeft een agent de volgende eigenschappen: een agent is autonoom, een agent heeft doelen, een agent heeft rollen, een agent heeft plannen, een agent heeft een beeld van zijn omgeving, een agent kan acties ondernemen en reageren op andere invloeden (Moergestel, 2015, slide 4). Qua intelligente agents wordt er onderscheid gemaakt tussen vijf typen (Stuart J Russell, 2009, p. hoofdstuk 2). Allereerst is er de simpele reflex agent. Deze agent zal altijd reageren op puur en alleen het begrip van zijn omgeving op dat moment, zonder rekening te houden met het verleden of eventueel zijn omgeving die momenteel niet zichtbaar is. Ten tweede zijn er de modelgebaseerde reflex agents. Deze agents houden een vorm van geschiedenis van zijn omgeving bij en hebben hier een begrip van. Ze houden als het ware een model van de wereld bij, waar de agent zijn naam van leent. Vervolgens zijn er doel gebaseerde agents. Dit type agent bouwt verder op het concept van een modelgebaseerde agent, en houdt rekening met een wenselijke situatie, namelijk zijn doel. Ook is er een zogenaamd utiliteit gebaseerde agent. Deze agent is vrijwel compleet hetzelfde als doel gebaseerde agents, met het verschil in een graadmeting van hoe "blij" de agent is met de staat waarin het zich verkeert. Deze graadmeting heeft extra invloed tijdens het besluitvormen. Tot slot is er de lerende agent. Deze agent is in alle opzichten gelijk aan de voorgaande agent, maar kan in eerste instantie opereren/functioneren in een voor hem compleet onbekende omgeving en heeft de mogelijkheid om over zijn omgeving te leren. Hierbij heeft het als doel om competenter te worden en efficiënter in het bereiken van zijn doel. Al met al zijn er met deze typen agents zeer diverse situaties te creëren en realiseren.

Domotica

Het woord domotica is een samentrekking van twee Latijnse woorden. Allereerst domus, wat huis betekent, en een toevoeging -tica dat toegepaste wetenschap lijkt te betekenen. Domotica is een automatisering op kleine schaal voor bijvoorbeeld woonkamers, of zelfs hele huizen. Voor de hand liggende voorbeelden van domoticatoepassingen zijn bijvoorbeeld het draadloos aanstuurbaar maken van licht, het draadloos aanstuurbaar maken van gordijnen of jaloezieën, of bijvoorbeeld het koppelen van temperatuursensoren en een tijdklok aan een thermostaat. Het veld van domotica ontwikkelt zich steeds verder, momenteel wordt de stap gemaakt naar Internet of Things toepassingen voor smart devices in een domotica omgeving. Een voorbeeld van deze toepassing is een beacon wat in verbinding staat met een sensor in een deurslot, wat op basis van nabijheid bepaalt of de deur op slot moet of niet.

Internet of Things

Kopetz (2011, chapter 13) stelde dat fysieke devices die middels het internet met elkaar verbonden zijn, het door deze verbinding mogelijk maken om informatie met elkaar uit te wisselen en hierdoor het mogelijk maken om meer voor elkaar te krijgen dan dat het geïsoleerde fysieke device zou kunnen. Dit is de meest compacte definitie van het internet of things, synergetische devices die informatie met elkaar uitwisselen om een groter geheel te realiseren dan de som der delen. Doorgaans wordt een internet of things omgeving opgezet met communicatie over bijvoorbeeld wifi, ook wel middels een ethernetkabel. Uiteraard zijn er tal van mogelijkheden om deze draadloze communicatie te realiseren. Wanneer wordt gekeken naar de lijst die te vinden is op de website van Postscapes (2016), is de volgende, niet per definitie complete, lijst van protocollen voor de fysieke en MAC laag uit het OSI-model te zien: WirelessHart, DigiMesh, ISA100.11a, IEEE 802.15.4, NFC, ANT, Bluetooth, ZigBee, EnOcean, Dash7, Thread, Weightless, WiMax, LoRaWAN, en uiteraard ook middels het cellulaire netwerk.

Smart devices

Volgens Kopetz (2011, chapter 13) zijn smart devices de bouwstenen van het internet of things, en slechts een andere benaming voor een embedded systeem dat verbonden is met het internet. Embedded devices zijn diep betrokken bij de opleiding Technische Informatica, onderzoek naar een mogelijke verbeterde communicatie onderling tussen dit soort devices is een zeer interessante en logische stap voor een Technisch Informaticus. Op de Wikipediapagina over smart devices wordt gesproken over een mogelijk autonome en interactieve operatie van smart devices. Het Slimme Pakket valt grotendeels onder de autonoom opererende smart devices, aangezien de voornaamste functie een simplex verbinding van het apparaat naar de eindgebruiker toe is. Echter heeft het ook eigenschappen van een interactief opererend smart device, met betrekking tot de mogelijkheid om randvoorwaarden van de sensorwaarden in te stellen voordat een Slim Pakket wordt verstuurd. De Wikipediapagina spreekt ook over een set karakteristieken die vaak voor smart devices gelden.

Een smart device:

- Heeft een hardware set en ICT-resources, vaak vastgelegd tijdens het ontwerpproces.
- Is dynamisch georiënteerd met betrekking tot component extensie en af en toe plug-and-play hardwarecomponenten.

- Biedt dikwijls op afstand toegang tot de service.
- Opereert op lokaal niveau geheel autonoom.
- Biedt dikwijls een externe omgeving om menselijke interactie af te handelen en vaak ook interactie met de fysieke wereld te monitoren.
- Deelt vaak eigenschappen met ubiquitous computing.

Deelvragen

Onder andere door het theoretisch kader, alsmede de hoofdvraag, is duidelijk geworden waar gaten in kennis aanwezig zijn. Om deze gaten op te kunnen vullen en de hoofdvraag te beantwoorden, is het belangrijk om deelvragen te formuleren.

Deelvraag 1. Welke standaard kan gebruikt worden om agents in een domoticaomgeving onderling te laten communiceren?

Agents zijn op zich staande entiteiten met een doel. Dit doel kan vaak bereikt worden door middel van samenwerken met andere agents, in een domotica omgeving is dit vrijwel altijd het geval. Daarom is het belangrijk om helder te krijgen welke standaard er gebruikt kan worden.

Deelvraag 2. Welke programmeertaal heeft een Agent Oriented Programming framework en is geschikt voor het uitvoeren van de opdracht?

Agent Oriented Programming is een programmeer paradigma, niet een op zich staande programmeertaal. Daarom is het belangrijk uit te zoeken welke taal het beste geschikt is voor de opdracht.

Deelvraag 3. In welke soortgelijke toepassingen wordt Agent Oriented Programming al toegepast?

Is ergens al informatie ingewonnen over soortgelijke problemen waarop verder gebouwd kan worden, of kan er geleerd worden van reeds gemaakte fouten?

Deelvraag 4. Wat wordt verstaan onder een domotica gateway en welke alternatieven zijn hiervoor?

Om een gatewayloze oplossing te creëren, is het wenselijk om helder te stellen wat een gateway voor domoticaomgevingen precies is.

Deelvraag 5. Hoe dient een domotica agent prioriteit toe te kennen aan zijn verscheidene inputs?

Een domotica apparaat krijgt te maken met meerdere vormen van input, zelfs in een normale omgeving. Wanneer dit gekoppeld wordt met andere agents, kan een apparaat overspoeld worden met input. Welk soort berichten kunnen dit zijn, waar zijn deze van afkomstig, en aan welke input dient voorrang verschaft te worden zijn daarom belangrijke vragen waar een antwoord op dient te komen.

Deelvraag 6. Met welke veiligheidsaspecten hebben agents in een domoticaomgeving rekening te houden?

Domoticaoplossingen zijn veelal draadloos aan te sluiten. In de richting van de hoofdvraag is het zeker wenselijk om draadloos te blijven, wat veiligheidsrisico's met zich meebrengt. Wanneer een nieuwe agent zich in de omgeving wil aansluiten, hoe kunnen de agents vaststellen dat het daadwerkelijk een agent is en geen kwaadwillende.

Aanpak

In dit hoofdstuk zal in worden gegaan op de wijze waarop de vragen beantwoord gaan worden.

Hoofdvraag

Voor het beantwoorden van de hoofdvraag zullen allereerst de antwoorden van de deelvragen gebruikt gaan worden. Verder zal er kennis vergaard worden door het uitvoeren van deskresearch, en een kleinschalig fieldresearch in de vorm van interviews met een tweetal autoriteiten in het veld, namelijk Leo van Moergestel en Steven van Dalen.

Deelvragen

Deelvraag 1 zal beantwoord worden door middel van een exploratief onderzoek, waarbij zo veel mogelijk verschillende communicatiemethoden achterhaald zullen worden en tegen elkaar afgewogen worden, om zo te beantwoorden welke methode voor de opdracht het beste is.

Deelvraag 2 zal beantwoord worden door middel van een kwalitatief literatuuronderzoek, om zo de mogelijke talen te achterhalen welke te gebruiken zijn voor de opdracht.

Deelvraag 3 zal beantwoord worden door het uitvoeren van casestudies, waarbij soortgelijke cases vergeleken worden om zo te achterhalen welke fouten makkelijk voorkomen kunnen worden.

Deelvraag 4 zal beantwoord worden door middel van een kwalitatief literatuuronderzoek, om zo te achterhalen wat nou daadwerkelijk een gateway is en dit goed te kunnen beschrijven.

Deelvraag 5 zal beantwoord worden door het houden van interviews. Door met de opdrachtgever, en tevens autoriteit binnen domotica en internet of things, te spreken over de case, kan achterhaald worden welke varianten van input het meest belangrijk zijn.

Deelvraag 6 zal beantwoord worden door een combinatie van observaties alsmede het houden van interviews. Op die wijze is vast te stellen welke veiligheidsrisico's Agent Oriented Programming in een domoticaomgeving met zich meebrengen.

Producten

Onderzoeksrapport

Bij een onderzoek hoort een rapport. De bevindingen met betrekking tot het onderzoek zullen in een onderzoeksrapport vastgesteld worden, en als bijlage van de scriptie opgeleverd worden. In dit onderzoeksrapport zal ook beschreven staan hoe de uitvoering van het onderzoek is verlopen.

Proof of concept

Het proof of concept zal zich richten op het demonsteren van de resultaten die naar voren zijn gekomen uit het onderzoek. Er zal een gesimuleerde domoticaomgeving op een Raspberry Pi draaien, waarbij meerdere agents worden aangemaakt om op die manier te demonstreren hoe deze met elkaar onderling communiceren.

Afbakening

Het proof of concept zal beperkt worden tot het demonsteren van drie agents, zodat er wel onderscheid gemaakt kan worden tussen verschillende soorten input, naar aanleiding van het antwoord op deelvraag 5. De demonstratie zal bestaan uit het demonsteren van functionaliteit van een led die aangestuurd wordt door een agent, het zichtbaar maken van de onderling verstuurd berichten tussen de agents en natuurlijk menselijke input.

Scriptie

Voor het afstuderen is het een vereiste dat aan het eind van de stage een scriptie wordt opgeleverd. Dit product zal documenteren hoe het onderzoek is gelopen en wat de resultaten zijn en een beschrijving van het proces totstandkoming van het proof of concept.

Projectplanning

Methoden

Er zal een literatuuronderzoek uitgevoerd worden om tot de beste methode te komen hoe Agent Oriented Programming gebruikt kan worden om de interoperabiliteit van smart devices te vergroten. Vervolgens zal er een proof of concept op worden gezet om te methode te bewijzen en demonstreren.

Meer over de data van afronding zijn te vinden in bijlage A.

Planning

Om de voortgang van het onderzoek en project te borgen, en de inmiddels dringende tijd, zal het project uitgevoerd worden volgens een planning. Allereerst zijn er drie belangrijke data waar boven alles rekening mee gehouden moet worden.

13 mei, het inleveren van het plan van aanpak met uitloop van maximaal één week.

5 juli, het inleveren van de definitieve scriptie.

23-26 augustus, het demonstreren van het proof of concept en eindpresentatie.

Verdere onderdelen van de uitvoering zijn opgenomen in de uitgebreide planning als bijlage A.

Projectorganisatie en risicoanalyse

Communicatie

Gedurende de uitvoering van het project zullen er wekelijkse meetings zijn om de bedrijfsbegeleider, stagiaires, en eventueel andere geïnteresseerden, een update te geven over de vorderingen van de opdracht. Deze meeting wordt wekelijks in de vestiging van Sogeti in Eindhoven gehouden.

Ook wordt er een wekelijks meeting georganiseerd op iedere dinsdag, te Vianen. Hier zullen andere projectgroepen aanwezig zijn die onder de businessline Internet of Things vallen binnen Sogeti.

Verdere communicatie zal incidenteel op locatie plaatsvinden maar hoofdzakelijk per email, WhatsApp en Skype.

Risicoanalyse

Aan ieder project zijn risico's verbonden. Ook dit project, echter is de verwachting dat deze risico's bij risico's blijven en geen werkelijkheid worden. Toch is het belangrijk om van tevoren enkele risico's in kaart te brengen, omdat het belangrijk is om voorbereid te zijn.

Tijdsnood

Allereerst bestaat de mogelijkheid dat het project meer tijd in beslag neemt dan in eerste instantie verwacht, wat zou inhouden dat ik het product niet kan opleveren binnen de gedefinieerde afstudeerperiode. Dit zou op te lossen zijn met een verlengd contract, waardoor ik na de afstudeerzitting alsnog werkzaam blijf bij Sogeti om het product af te maken en op te leveren. Echter dient uitermate goed onderbouwd te worden bij de afstudeerzitting waarom er meer tijd nodig is geweest.

Onvoorziene complexiteit

Ook bestaat de mogelijkheid dat de opdracht complex dan ingeschat blijkt, wat inhoudt dat ik hem niet zou kunnen uitvoeren met mijn huidige kennis. Een makkelijke manier om dit te verhelpen, is gebruik te maken van de in het theoretisch kader gedefinieerde literatuur. Op die manier zou mijn kennis uitgebreid kunnen worden, met als resultaat dat het project alsnog afgerond kan worden.

Afwezigheid begeleiding

Gedurende het uitvoeren van het project, wordt ervan uit gegaan dat begeleiding aanwezig is of enigszins een vraagbaak beschikbaar. Echter bestaat de mogelijkheid dat dit niet de situatie is, bijvoorbeeld door langdurige afwezigheid van de begeleider of een detachering van een vraagbaak. Mocht deze situatie zich voor doen, is het wenselijk dat de student allereerst proactief op zoek gaat naar een vervangende vraagbaak. Mocht dit niet succesvol lukken, rest de mogelijkheid om de docentbegeleider op de hoogte te stellen van de afwezigheid. Vervolgens kan er samen gekeken worden naar een geschikte oplossing.

Hardware falen

Voor het proof of concept dient gebruik gemaakt te worden van hardware. Het is daarom belangrijk om een terugval te hebben wanneer de hardware tussentijds defect raakt. In het geval van deze opdracht wordt er gebruik gemaakt van een Raspberry Pi. Dit stuk hardware is tegenwoordig zeer goed beschikbaar voor een relatief lage prijs. Wanneer de beschikbare Raspberry Pi kapot gaat, kan er bij wijze van spreken de volgende dag al een nieuwe aanwezig zijn.

Bedrijfs-/persoonsgegevens

Sogeti

Lange Dreef 17
4131 NJ Vianen
+31 88 6606600
info@sogeti.com

Bedrijfsbegeleider

Richard Verhaaf
+31 6 50856781
richard.verhaaf@sogeti.com

Internet of Things & Domotica vraagbaak

Steven van Dalen
+31 6 53424492
steven.van.dalen@sogeti.com

Docentbegeleider

Gerald Ovink
+31 88 4813383
gerald.ovink@hu.nl

Student

Thomas Kok
+31 6 53511040
thomas.t.kok@student.hu.nl
thomas.a.kok@sogeti.nl
thomasjkok1990@gmail.com

Bibliografie

Hogeschool Utrecht. (2015). *Afstudeerleidraad Bacheloropleiding HBO-ICT 2015 - 2016*. Utrecht: Hogeschool Utrecht.

Kopetz, H. (2011). *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Springer Science+Business Media.

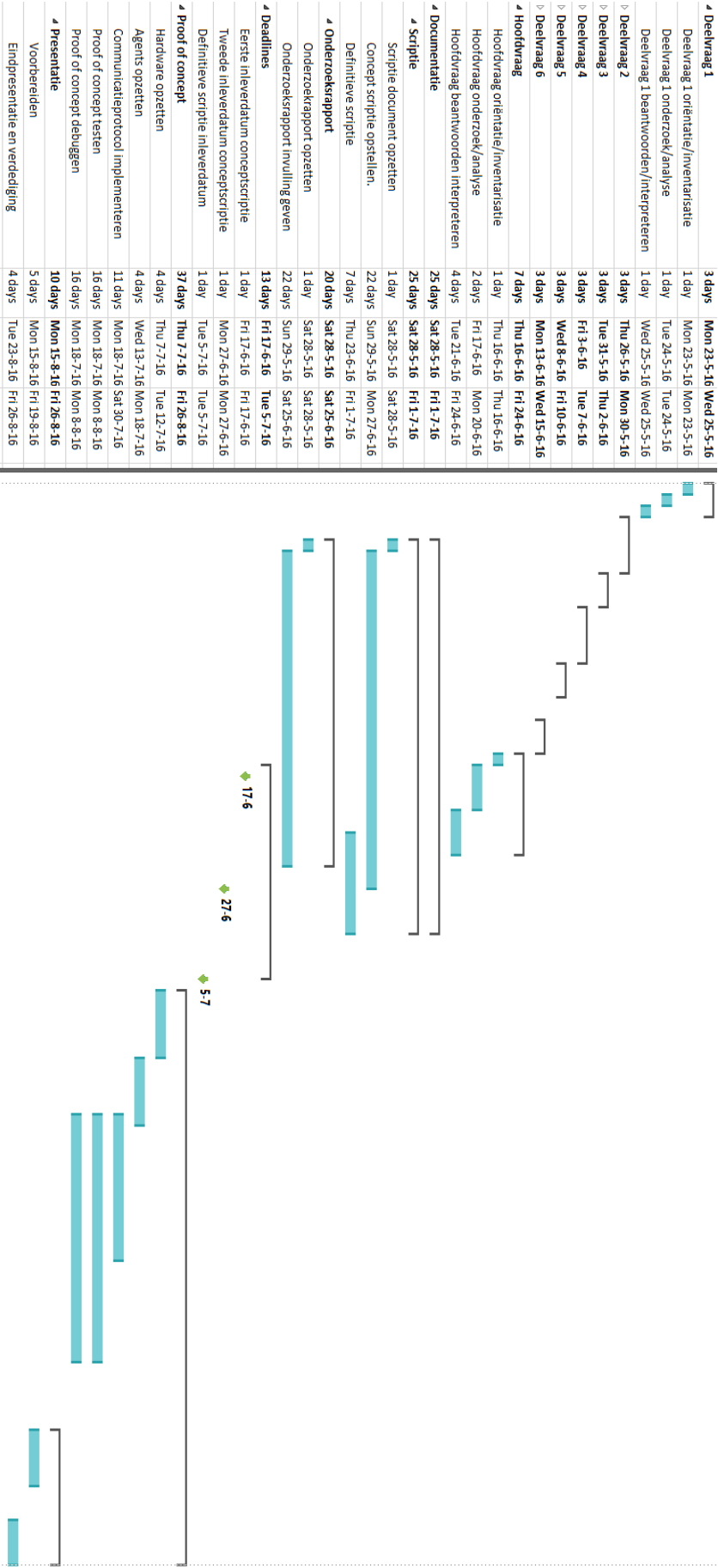
Moergestel, L. v. (2015, 4 23). Agents [PowerPoint slides].

Shoham, Y. (2010). *Multiagent Systems. Algorithmic, Game-Theoretic, and Logical Foundations*. Stanford: Cambridge University Press.

Stuart J Russell, P. N. (2009). *Artificial Intelligence: A Modern Approach, 3rd edition*. Upper Saddle River: Pearson.

Bijlagen

Bijlage A: Strokenplanning



Bijlage B: Persoonlijke doelstellingen

- Assertiever worden in communicatie,
- Duidelijk/helder mijn verwachtingen en wensen aankaarten,
- Zelfverzekerd overkomen in communicatie, met name met betrekking tot twijfel over mijn kennis