

Afstudeerscriptie: Simulation of VOIP Traffic with Agent Technology

Naam:	Rick van Rijn
Studentnummer:	1581388
Docentbegeleider:	Dr. ir. Ander de Keijzer
Tweede examiner:	Dr. Leo van Moergestel
Bedrijfsbegeleider	ing. Bram de Boer
Datum:	16-12-2014
Versie:	1.0

Voorwoord

Dit document beschrijft de progressie van mijn afstudeer periode bij BasiCall BV te Houten. Deze afstudeerperiode is van Augustus 2014 tot en met December 2015.

De keuze voor deze afstudeeropdracht is vanwege die uitdaging die de implementatie van agent technologie, VOIP en simulatie welke aan bod komt. Omdat agent technologie en simulatie een vorm van artificiële intelligentie zijn en agent technologie een vernieuwende en innovatie is in de ICT heeft dit mijn aandacht getrokken. Alle hiervoor genoemde onderwerpen zijn een interesse van mij en dat maakt dat ik geloof in een goede afwikkeling en volledige inzet kan tonen.

Ik heb veel geleerd over agent technologie, simulatie technieken en veel over de omgang met hardware in het besturingssysteem Linux. Door middel van het JADE platform is de mogelijkheid er om alle ideeën te realiseren door middel van onder andere agent technologie. Het idee om agent technologie te verwerken in combinatie met het simuleren van bestaande software en VOIP gesprekken staat zowel mij als het bedrijf erg aan. Gezien de simulatie en analyse mogelijkheid vereist dit een nauwkeurige en accurate aanpak die beter door een software agent kan worden toegepast vanwege dat deze exacter zijn en sneller.

Voorafgaand dit document is een voorstel en plan van aanpak gemaakt. De opdracht was een zeer pittige opdracht en had een hoop tijd en inzet nodig om tot een goede uitkomst te leiden. Graag wil ik hier nog een aantal mensen bedanken namelijk Ander de Keijzer als mijn begeleider en reviewer, Bram de Boer van BasiCall vanwege zijn technische ondersteuning en Rinus Versteeg vanwege zijn operationele kennis en ondersteuning.

Rick van Rijn

Houten, 15 Oktober 2014

Management-samenvatting

Voor het bedrijf BasiCall BV wat zich specialiseert in software voor contactcenters is een tool gemaakt. Een contactcenter is een bedrijf dat zich bezighoudt met het telefonisch verkopen of ondersteunen van consumenten. De gemaakte tool is bedoeld om medewerkers van een contactcenter te simuleren. Dit gebeurt met behulp van Agent Technology. Dit zijn op zichzelf staande entiteiten maar dan softwarematig.

Er worden contactcenter medewerkers gesimuleerd om in staat te zijn de hardware performance te meten van de server(s) met de daarvoor geïmplementeerde Analyse Agent. Deze Analyse Agent maakt gebruik van het Linux systeem pakket Sysstat. De Agent Technology wordt gerealiseerd door de implementatie van het JADE Framework. De tool is met de programmeertaal JAVA gerealiseerd en wat op ieder besturingssysteem kan worden gedraaid. Hierdoor is het mogelijk om vanuit een Windows omgeving met een Linux omgeving te communiceren. De standaard server-set bestaat uit een opstelling van twee Linux servers, waarvan een Asterisk server en een webserver welke met elkaar communiceren door middel van de door BasiCall BV gemaakte software.

De Asterisk server is de server welke verantwoordelijk is voor de VOIP communicatie, door middel van een softwarematige telefoon in de simulatie te verwerken is het mogelijk om automatisch aan te melden bij de Asterisk server waardoor er audio streaming kan plaatsvinden.

Het software programma van BasiCall BV voor de medewerker zal een simulatiemodus bevatten. Deze simulatiemodus zal wanneer deze actief is door een belscript lopen. Door middel van vooraf gedefinieerde commando's wordt het mogelijk om door een belscript te lopen. Er wordt een Machine Learning algoritme gebruikt om een vorm van intelligentie in de simulatie te brengen. Het belscript wordt door het eerder genoemde algoritme in verschillende volgordes en hierdoor zal de analyse tot verschillende eindresultaten komen.

Op de webserver worden voor het gebruik van de tool simulatie accounts aangemaakt. Hierdoor is het mogelijk om tijdens de simulatie met accounts in te loggen welke verder geen effect hebben op de rapportages. De resultaten die worden geboekt tijdens een simulatie zullen dan ook niet worden weggeschreven in de database en er zullen geen daadwerkelijke bevestigingsmails worden verzonden. De simulaties worden gestart vanuit een centrale plek, welke gerepresenteerd wordt door een grafische beheer applicatie. Vanuit een simulatie worden twee processen gestart:

1. Softwarematige VOIP telefoon;
2. De BasiCall software in simulatie modus;

De softwarematige VOIP telefoon dient als eerste te worden gestart omdat de BasiCall software pas door het belscript kan gaan wanneer er een daadwerkelijk telefoongesprek is opgezet. Aan de serverkant zal er een software Agent aanwezig zijn die de analyse kan maken. Deze hiervoor genoemde Agent zal actief worden op het moment dat de gebruiker de analyse start in de grafische beheer applicatie. Indien de analyse klaar is zullen de resultaten worden verzonden naar de beheer applicatie en zal er een PDF analyse rapport worden opgeslagen. Een analyse wordt gestart voordat er simulaties worden gestart en zal nadat de laatste simulatie is gestopt of handmatig eerder wordt gestopt de analyse resultaten verwerken. In dit rapport valt te lezen waar eventueel performance problemen liggen.

Versie Beheer

Versie nummer	Wie	Wanneer	Wat
0.1	Ricky van Rijn	15-10-2014	Document aangemaakt en structuur aangemaakt
0.2	Ricky van Rijn	23-10-2014	Document gevuld met onderzoeksgegevens
0.3	Ricky van Rijn	25-10-2014	Inleiding, Voorwoord en Samenvatting tekstueel invulling gegeven
0.4	Ricky van Rijn	07-11-2014	Structuur document herzien
0.5	Ricky van Rijn	10-11-2014	Voorblad en Inleiding inhoudelijk aangepast.
0.6	Ricky van Rijn	15-11-2014	Inleiding tekstueel aangepast.
0.7	Ricky van Rijn	25-11-2014	Hoofdstuk 2 tekstueel aangepast
0.8	Ricky van Rijn	01-12-2014	Hoofdstukken 3 en 4 tekstueel en inhoudelijk aangepast.
0.9	Ricky van Rijn	09-12-2014	Hoofdstuk 5 en 6 tekstueel en inhoudelijk aangepast.
1.0	Ricky van Rijn	11-12-2014	Hoofdstuk 7,8, Bronvermelding en de Management-samenvatting tekstueel en inhoudelijk aangepast.

Inhoudsopgave

1. Inleiding	6
1.1. Doelstelling	6
1.2. Probleemstelling	6
1.3. Huidige situatie	6
1.4. Gewenste situatie	7
1.5. Onderzoeksvragen	8
1.6. Risico's en randvoorwaarden.....	8
1.7. Project organisatie	9
1.8. Begrippen.....	9
2. Analyse.....	13
2.1. Methodieken	13
2.2. Project aanpak.....	14
2.3. Gebruikte methoden en tools	15
2.4. Onderzoek.....	15
2.5. Testen	26
3. Functionele beschrijving	27
3.1. Globaal ontwerp	27
3.2. Simulation Agent.....	27
3.3. Sim Agent	28
3.4. Analyse Agent	29
4. Implementatie	29
4.1. De Agent Technologie	29
5. Resultaten.....	34
5.1. Voltooide producten	34
5.2. Overdracht van project	34
5.3. Demonstratie	35
6. Conclusie.....	36
6.1. Antwoorden van onderzoeksvragen	36
6.2. Evaluatie.....	37

7. Aanbevelingen	37
8. Vervolgonderzoek	38
9. Bronvermelding	39
Bijlage A: Plan van Aanpak	41
Bijlage B: Zelfevaluatie	15
Bijlage C: UML Klassendiagram	17

1. Inleiding

Dit hoofdstuk beschrijft de achtergrond informatie van de afstudeer opdracht. De projectorganisatie en de mensen die hierbij betrokken zijn worden genoemd. De ideeën achter het project zullen worden toegelicht en de sleutel punten zullen tot in detail worden toegelicht.

1.1. Doelstelling

Het project wordt uitgevoerd om de mogelijkheid om een medewerker te simuleren en/of de bijbehorende technische aspecten van de server tijdens een of meerdere gesprekken te monitoren. Het afstudeerbedrijf is specifiek gericht op contactcenters welke telefonisch producten verkopen en op bedrijven welke een klantenservice hebben. Het telefonische verkeer wordt gerealiseerd door middel van VOIP technologie.

De doelstelling van dit project is een of meerdere medewerkers simuleren en de server diens hardware prestaties te analyseren ten behoeve van een betere probleemoplossing bij storingen en/of defecte hardware.

Doel

Het doel van het project is om een correcte en gedetailleerde analyse te maken van de hardware om in kaart te brengen wat de oorzaak is van terugvallende kwaliteit bij simultane VOIP. Hiervoor is het benodigd om een of meerdere actieve contactcenter medewerkers te simuleren om een correcte analyse te kunnen maken.

De Simulatie Agent zal een VOIP simulatie gesprek opzetten via de Asterisk server en met behulp van programmeerbare commando's door het script lopen van een contactcenter medewerker. De Simulatie Agent zal vervolgens een instantie starten van de BasiCall Agent software en daarbij inloggen met simulatie gegevens bij een vooraf aangegeven project.

Een contactcenter heeft meerdere projecten lopen voor bijvoorbeeld de verkoop van een catalogus of een tijdschrift. Omdat er meerdere projecten tegelijkertijd kunnen lopen is het van belang om de Simulatie Agent in te kunnen loggen bij een specifiek project. Zo kan er een realistisch beeld worden geschept van meerdere Simulatie Agent instanties die onder verschillende projecten zijn ingelogd.

Middels een analyse proces op de servers zal er een analyse gemaakt worden op basis van de harde schijf lees en schrijf operaties, gebruikt RAM geheugen, processoren gebruik, vrije harde schijfruimte en wisselbestand geheugen. Hier zal een advies uitkomen in de vorm van een tekst bestand waarbij het probleem duidelijk zichtbaar is indien het probleem hardware matig is.

1.2. Probleemstelling

Het probleem is het niet goed kunnen analyseren van de terugvallende kwaliteit bij simultane VOIP gesprekken bij het vastlopen van het systeem. Met name de vraag of het defect ligt aan de hardware of de software. Het systeem dient op diepgaand niveau een analyse op te leveren waarbij een analyse van de hardware wordt gemaakt waaruit te halen moet kunnen zijn hoe deze heeft gepresteerd.

1.3. Huidige situatie

In dit hoofdstuk wordt een beschrijving gemaakt van de infrastructuur welke op dit moment aanwezig is. Hierin wordt een beeld geschetst hoe de huidige contactcenter infrastructuur op basis van de BasiCall software werkt en welke software verantwoordelijk is voor welke activiteit. De huidige situatie bestaat uit een Asterisk server en een webserver. Op de webserver is de BasiCall Server software geïnstalleerd. De Asterisk server is een Linux systeem waarop Asterisk draait. Asterisk is een programma voor VOIP communicatie, zie hiervoor bron 20 in hoofdstuk 9.

1.3.1. Asterisk Server

De Asterisk server heeft de volgende extensies heeft geïnstalleerd:

1. Asterisk-Java;
2. SIP-Realtime;

De Asterisk-Java extensie maakt het mogelijk om door middel van een JAVA API de Asterisk met JAVA code aan te sturen, zie hiervoor bron 17 in hoofdstuk 9. De SIP-Realtime extensie maakt het mogelijk om de SIP instellingen te beheren in een MySQL database waaronder ook de SIP accounts toe behoren, zodoende kunnen de SIP accounts beheerd worden door MySQL queries uit te voeren.

Op de Asterisk server is ook Apache, PHP en MySQL geïnstalleerd. Echter de MySQL installatie wordt alleen voor de SIP-Realtime extensie gebruikt, zie hiervoor bron 21 in hoofdstuk 9.

1.3.2. Webserver

Op de webserver zijn de volgende programma's geïnstalleerd:

1. Apache;
2. PHP;
3. MySQL;
4. BasiCall Server software;
5. Screen;

De Apache, PHP en MySQL omvat het geheel voor de belscripts. De belscripts worden gemaakt in PHP en de gegevens worden in de MySQL database opgeslagen. De BasiCall Server software draait continu in een screen sessie en is een JAVA programma. De Sun JAVA software is ook geïnstalleerd op de webserver om de BasiCall Server software te kunnen draaien. De BasiCall Server software ontvangt de JAVA commando's van de BasiCall Agent software welke de communicatie regelt met de Asterisk server.

Het programma screen is een programma dat de output van een programma in een virtueel scherm start en opvangt. Zelfs als de ssh-sessie gesloten wordt blijft het virtuele scherm en de daarin gestarte programma's in deze screen-sessie doorlopen. Zodoende kan er wanneer de BasiCall Server software draait de output alleen worden getoond indien de screen sessie wordt geopend.

Doordat de zogenoemde heartbeats die controleren of de verbinding met een instantie van de BasiCall Agent software er nog is en verdere debug informatie worden geprint op het scherm is het handig om de output in een apart scherm vast te houden. Er wordt nu een grove inschatting gemaakt op basis van eerdere ervaringen hoeveel connecties de Asterisk Server en webserver aan kunnen. Echter verschilt het aantal connecties wat een server-set aankan en stabiel blijft. Onder stabiel wordt het volgende verstaan:

- Een belscript opent binnen 2 seconden;
- Er ontstaan geen haperingen in het geluid;
- Het geluid vervormt niet, het ontstaan van een "Robot" stem blijft achterwege;
- Het inloggen gebeurt binnen 2 seconden;

Indien een server-set dreigt onstabiel te worden wordt er hardwarematig een component vervangen of geplaatst om de performance te vergroten.

1.4. Gewenste situatie

Door een aantal contactcentermedewerkers te simuleren en vervolgens een analyse te starten kan inzicht worden verkregen wanneer een server-set onstabiel dreigt te worden. Indien de analyse klaar is kan het analyserapport opgevraagd worden met de resultaten. Dit rapport moet inzichtelijk maken waardoor het systeem bij het starten van een vooraf bepaald aantal Simulatie Agents goed of slecht loopt.

Indien een server-set onstabiel dreigt te worden kan hierdoor worden bepaald welke factoren problematisch zijn voor de performance. Hierdoor moet het mogelijk worden om tot een goed onderbouwde oplossing te komen om de performance te verbeteren. Door een beheer applicatie moet het mogelijk zijn om een vooraf bepaald aantal Simulatie Agents te starten en te stoppen en het kunnen opvragen van de resultaten van een analyse.

Zo moet het mogelijk worden om een server-set vooraf te testen alvorens de server-set naar een klant gaat en kan het maximaal aantal accounts wat in kan loggen op de server-set bepaald worden.

1.5. Onderzoeksvragen

De hoofdvraag van dit project luidt: “Welke direct inzetbare tool dient geïmplementeerd te worden voor het maken van een analyse van de actieve processen die de oorzaak zijn voor de terugvallende kwaliteit bij simultane VOIP gesprekken?”

De bijbehorende deelvragen zijn:

1. Welke actieve processen hebben effect op de kwaliteit van de VOIP gesprekken?
2. Welke methodes zijn er om van de actieve processen meetgegevens te verkrijgen?
3. Wat is de definitie van de kwaliteit van een VOIP gesprek en wat is hierbij de minimaal geaccepteerde kwaliteit?
4. Welke criteria zijn nodig om uit de meetgegevens tot een onderbouwde analyse te komen?

1.6. Risico's en randvoorwaarden

Randvoorwaarden:

- Het analyse-proces wordt vermeden zelf te worden meegenomen in de meting omdat dit anders kan leiden tot foutieve en onrealistische metingen .
- De bestaande BasiCall software dient volledig functioneel te zijn zonder netwerk time-outs tussen een set van 2 servers.
- Het analyse rapport dat wordt opgeleverd dient voor een persoon zonder achtergrond in de ICT duidelijk te zijn. Het moet aangeven waar het probleem in zit met behulp van een duidelijk bepaald referentiekader.
- De simulatie agent moet een aantoonbare eigen rol hebben in het geheel.
- Het VOIP gesprek dient te kunnen worden meebeluisterd door middel van de BasiCall Control Center software waardoor aangetoond wordt dat het gesimuleerde VOIP gesprek actief is.
- Er kunnen meerdere Simulatie Agents inloggen wat inherent is aan het simuleren van simultane VOIP gesprekken.

Risico's:

- Er draait een oudere JAVA versie op de BasiCall servers waardoor mogelijk sommige functies niet of incorrect werken.
- Indien er sprake is van routers die door derden worden beheerd kan er niet gegarandeerd een perfecte berichten uitwisseling tussen agents zijn.
- Er kunnen mogelijk memory leaks voorkomen in externe libraries welke kunnen leiden tot het crashen van de JAVA Runtime, waardoor alle JAVA functionaliteiten worden gestaakt.

1.7. Project organisatie

Dit project is mogelijk gemaakt door BasiCall BV. Dit bedrijf is een software-bedrijf wat een eigen contactcenter applicatie heeft ontwikkeld. Daarnaast worden er ook andere IT-diensten aangeboden. Het bedrijf is gevestigd op De Haag 6, Houten, Nederland.

1.7.1. Betrokken personen

De betrokken personen met betrekking tot dit project en hun rol worden in een tabel op de volgende pagina uitgelegd.

Rol	Persoon	Taak
Begeleider	Ander de Keijzer	- Feedback op resultaten en documentatie
Project supervisor	Ander de Keijzer Rinus Versteeg Bram de Boer	- Feedback op resultaten - Bewaking van de business case - Technische ondersteuning
Project deelnemer	Rick van Rijn	- Ontwerp - Implementatie proof-of-concept - Overdracht van project

1.8. Begrippen

1.8.1. Software Agent

Een Software Agent is een software entiteit welke een soort vrijheid heeft om zichzelf te uiten. Om als Agent te kunnen uiten moet een Agent de omgeving kunnen interpreteren.

Een Software Agent is een intelligente entiteit met input en output welke een Agent kan gebruiken om te observeren en acties te ondernemen. De manier waarop intelligente Agents handelen is erg interessant omdat deze ook ineens een negatief antwoord kunnen terugsturen wat niet verwacht was. Dit komt door de toepassing van het BDI model, Believe, Desire, Intention. Het BDI model wordt hier later toegelicht.

De definitie van een Software Agent zoals deze gebruikt wordt in dit document is de definitie zoals Wooldridge hem heeft gedefinieerd. Wooldridge definieert een Agent als volgt: “An agent is a computer system, that can act autonomously in a certain environment to achieve its assigned goals.”

1.8.2. BasiCall contactcenter Agent

De contactcenter Agent simuleert een contactcenter medewerker welke door middel van BasiCall BV Software inlogt op een project en door middel van deze software door het script manoeuvreert en resultaten afboekt als zijnde negatief zowel als positief.

1.8.3. BasiCall analyse Agent

De analyse Agent zal een analyse maken aan de server kant en deze informatie doorsturen naar de client welke de beheerapplicatie heeft openstaan en de analyse heeft gestart.

1.8.4. Multi-Agent systemen

Er werken meerdere Agents samen in een multi-agent systeem om een resultaat te bereiken. Een sociale interactie en samenwerking zijn van groot belang in een multi-agent systeem. De combinatie van de acties van meerdere Agents kan tot veel geavanceerdere en complexere systemen leiden zelfs als het gedrag van de Agent erg basaal is.

Dit fenomeen is bekend als emergent gedrag, het multi-agent systeem als geheel is niet langer te beschrijven vanwege de individueel opererende agents. Voor een persoon welke multi-agent systemen ontwerpt is het van groot belang om te kijken naar het individuele gedrag van elke agent, hoe deze samenwerken en hoe deze samenwerking zou moeten gaan.

1.8.5. Simuleren

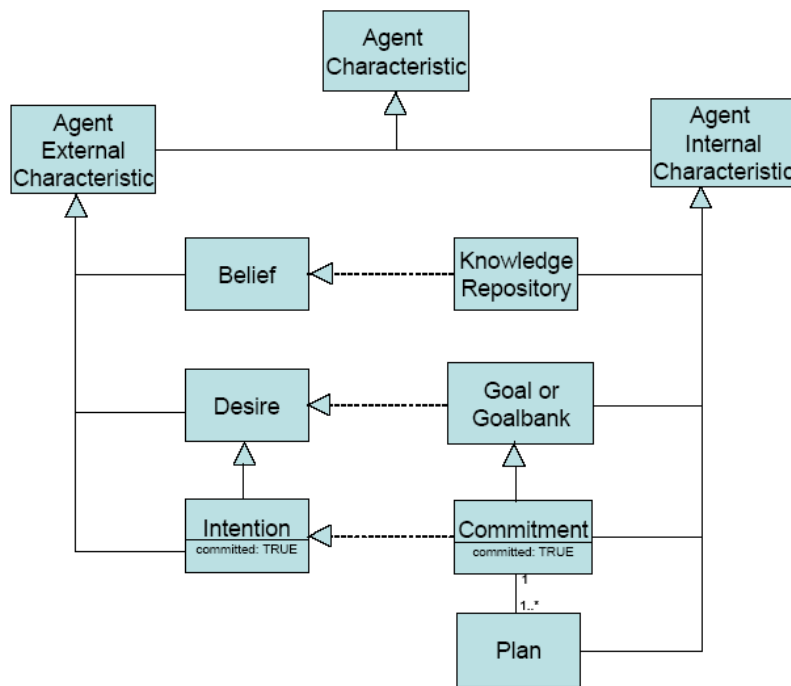
Het simuleren van een proces is het kunnen nabootsen van een situatie zonder dat deze daadwerkelijk wordt ingezet op volledige schaal.

1.8.6. BDI- model

Het BDI model staat voor Believe, Desire en Intention, ofwel naar het Nederlands vertaald: Geloof, Verlangen, Intentie. Dit model is geïmplementeerd aan de hand van Michael Bratman's "theory of human practical reasoning", zie hiervoor bron 6 in hoofdstuk 9 . De architectuur van een BDI model is als volgt:

- **Beliefs:** Beliefs of ook wel geloof in het Nederlands, representeert de informele staat van de Software Agent, ofwel zijn geloof in de wereld wat inhoudt dat wat hij gelooft niet noodzakelijk waar is.
 - **Beliefset:** Een database van Beliefs, dit is echter een implementatie keuze.
- **Desires:** Desires of ook wel verlangen in het Nederlands representeert de motivatie van de Software Agent wat deze leuk vindt of wil bereiken.
 - **Goal:** Het doel van de Software Agent, het doel van de software agent dient consistent te zijn en bijvoorbeeld niet om naar een feest te gaan en om thuis te blijven hoewel dit beide een verlangen kan zijn.
- **Intentions:** Intentions of ook wel intenties in het Nederlands representeert de overwegende staat van de Software Agent.
 - **Plans:** De plannen zijn reeksen van acties die de Software Agent kan uitvoeren om een van zijn intenties uit te voeren. Plannen kunnen ook andere plannen bevatten, zoals voor het plan om auto te gaan rijden kan ook het plan om de sleutels te zoeken inhouden.

Dan zijn er nog de Events of ook wel acties in het Nederlands welke de reacties representeren op basis van een sensor, een waarde of de uitkomst van een plan waardoor het model gewijzigd kan worden en er vervolgens een nieuwe situatie ontstaat met andere Beliefs, Desires en/of Intentions.



Figuur 1: BDI Model

De knowledge repository geldt als database voor opgedane kennis wat leidt tot Belief. De goal of goalbank is een database van doelen die gesteld worden, wat leidt tot Desire. Commitment leidt tot intention, waarbij de Agent committed moet zijn, vandaar dat comitted op true ofwel waar staat.

Een plan bestaat uit de knowledge repository, Goalbank en Commitment. De knowledge repository, Goalbank, Commitment en een plan vormen samen de interne karakteristieken van een Agent. Intention, Desire en Belief vormen de externe karakteristieken van een Agent. De interne en externe karakteristieken vormen samen de Agent karakteristieken, zie hiervoor bron 25 in hoofdstuk 9.

1.8.7. Ontologie

Een ontologie representeert een kennisbank als een hiërarchie van concepten binnen een domein door gebruik te maken van een gedeelde woordenschat om typen en eigenschappen en overige vormen van kennis te delen.

1.8.8. ACL

ACL is de afkorting voor Agent Communication Language en is voorgesteld door de Foundation for Intelligent Physical Agents (FIPA) als standaard taal voor de communicatie tussen Agents, zie hiervoor bron 10 in hoofdstuk 9. Een ACL bericht bestaat uit een aantal onderdelen waarvan een verplicht is, namelijk de performative. De performative houdt het type van bericht in. Dit kan onder andere een inform bericht zijn.

De volgende onderdelen kunnen gebruikt worden in een ACL bericht:

1. Frame: de verplichte naam van de entiteit die elke instantie van deze klasse representeert.
2. Ontology: dit is de naam van de ontologie welke het domein specificeert en beschrijft welke parameters er dienen te worden meegegeven.
3. Parameter: de parameter identificeert ieder component van een frame.
4. Description: de beschrijving van de desbetreffende parameter.
5. Reserved Values: dit is een lijst met door FIPA vooraf gedefinieerde constanten geassocieerd met iedere parameter.

1.8.9. MTP

Ten behoeve van de communicatie tussen Agents die op verschillende platformen aanwezig zijn is er het Message Transfer Protocol. Dit protocol is in staat om middels MTP FIPA-ACL berichten te verzenden tussen verschillende platformen, dit maakt communicatie mogelijk voor een Agent met een andere Agent welke aanwezig is op een extern platform, ongeacht of deze onder hetzelfde Framework draait of niet.

MTP bestaat uit drie modules:

1. HTTP MTP: MTP gebaseerd op webserver technologie;
2. IIOP MTP: MTP gebaseerd op een door CORBA gedefinieerd transport protocol;
3. SMTP MTP: MTP gebaseerd op Simple Mail Transfer Protocol;

1.8.10. Asterisk

Asterisk is een open source framework om communicatie applicaties te maken. Asterisk heeft de mogelijkheid om te dienen als onder andere Voip gateway, conferentie server en is uitermate geschikt voor een zelf bedachte applicatie. Asterisk is de open source standaard voor realtime geluid en video, zie hiervoor bron 20 in hoofdstuk 9.

1.8.11. Java Remote Method Invocation

De Java Remote Method Invocation is een service van de Sun Java software dat de Object georiënteerde gelijke van de RPC (Remote Procedure Calls) uitvoert met ondersteuning voor een directe overdracht van versie benoemde Java Objecten en de gedistribueerde garbage collection. Deze service draait standaard op poort 1099. Omdat het JADE Framework ook standaard op poort 1099 draait zal er een andere poort gekozen moeten worden om zonder problemen te kunnen draaien. Als gevolg van de keuze om op een andere poort te werken is dat de firewall aangepast moet worden om ingaand en uitgaand verkeer toe te laten op deze poort.

2. Analyse

2.1. Methodieken

2.1.1. SCRUM

De SCRUM methodiek wordt in dit project op kleine schaal gebruikt. SCRUM is een software ontwikkeling framework voor het beheren van software applicaties en ontwikkeling. Er wordt een SCRUM master toegewezen in een groep indien nodig. De SCRUM master is verantwoordelijk voor het houden van SCRUM vergaderingen die elke dag gehouden dienen te worden om te zien hoe de project groep het doet tijdens een zogenoemde sprint.

De ontwikkeling van een software applicatie is verdeeld in zogenoemde sprints, iedere sprint heeft taken aan welke voldaan moeten zijn, voor het einde van de sprint. Elk lid van de project groep kan de verantwoordelijkheid opeisen van een van deze taken en deze taak oppakken. In een dagelijkse SCRUM vergadering vertelt ieder lid aan de project groep wat het lid heeft gedaan en wat het lid deze dag gaat doen en tegen welke problemen deze aanloopt. Alles wat wordt besproken gedurende de SCRUM vergaderingen zal worden vastgelegd in een zogenoemde backlog.

2.1.2. GAIA

De Gaia methodologie is een poging om een complete en generieke methodologie te definiëren, welke specifiek is afgestemd op de analyse en het ontwerp van Multi-Agent systemen. Een Multi-Agent systeem wordt door Gaia gezien als een aantal autonome interactieve Agents welke in een georganiseerde samenleving leven. Hierbij heeft iedere Agent een of meer specifieke rollen.

Gaia definieert de structuur van een Multi-Agent systeem in een rollenmodel. Dit model geeft de rol aan welke de Agents zullen uitoefenen binnen in het Multi-Agent systeem en zal de interactie tussen de verschillende rollen specificeren. Het doel van het Gaia analyse proces is, de identificatie van de rollen en de modellering van interacties tussen de gevonden rollen. Een rol houdt vier attributen in:

1. verantwoordelijkheden;
2. bevoegdheden;
3. activiteiten;
4. protocollen;

De verantwoordelijkheden zijn de sleutel attribuut welke de functionaliteit bepaald van de rol. Er zijn twee typen verantwoordelijkheden:

1. levendigheid;
2. veiligheid;

De levendigheid is bedoeld om een positieve toevoeging aan de rol te geven en de veiligheid is bedoeld om de rol te behoeden voor slechte uitwerkingen en het niet toestaan dat er iets slechts gebeurt.

Om verantwoordelijkheden te kunnen realiseren zijn bevoegdheden nodig. Bevoegdheden representeren wat de rol mag doen en welke informatiebronnen deze mag gebruiken. De activiteiten zijn taken die uitgevoerd worden door de Agent zonder interactie met andere Agents. Protocollen zijn de uitgewerkte blauwdrukken voor de interactie tussen de Agents. Bijvoorbeeld een verkoper kan meerdere veiling protocollen hanteren. Gaia heeft formele operatoren en sjablonen voor het representeren van rollen en hun attributen. Zo heeft Gaia ook schema's die gebruikt kunnen worden voor de representatie van interacties tussen de verscheidene rollen in een Multi-Agent systeem. Gaia wordt hier verder niet gebruikt gezien er geen sprake is van een Multi-Agent systeem. De rollen zijn zodanig toegewezen dat een centrale Agent en een of meerdere Agents een activiteit uitvoeren en automatisch stoppen nadat de activiteit klaar is, zie hiervoor bron 24 in hoofdstuk 9.

2.2. Project aanpak

Het project houdt de volgende activiteiten in welke voltooid dienen te worden om tot een werkende proof-of-concept te komen:

1. Een analyse Agent.
2. Een simulatie Agent.
3. Een applicatie om de hierboven genoemde Agents makkelijk mee te kunnen starten en stoppen in de vorm van een grafische toepassing;

2.2.1. Een analyse Agent

Een analyse Agent bestaat uit de volgende activiteiten:

1. De mogelijkheid om een bericht te ontvangen, start en stop bericht.
2. IOstat uitvoeren met vooraf gespecificeerde parameters, hoeveel rapporten en met welke tijdsinterval deze gemeten wordt.
3. Gegevens naar de client sturen.
4. Eindrapport opmaken.
 - a. Legenda opmaken.
 - b. Referentiekader opmaken.
5. Eindrapport tonen.

2.2.2. Een simulatie Agent

Een simulatie Agent bestaat uit de volgende activiteiten:

1. Sip account registreren.
2. Inloggen van een BasiCall account bij een project.
3. Verbinding met de SipPhone maken, automatisch opnemen.
4. BasiCall account op beschikbaar zetten.
5. Een project script openen.
6. Door het script manoeuvreren.
 - a. Algoritme voor het willekeurig door het script laten lopen.
 - b. Een parser voor het parsen van HTML om de formulieren te bevestigen.
7. Na de laatste pagina van een project weer op beschikbaar zetten van de BasiCall account.

2.2.3. Beheer applicatie

Deze beheer applicatie bestaat uit de volgende activiteiten:

1. Het maken van een grafische omgeving.
2. De knoppen en velden in deze omgeving koppelen aan acties.
3. Communiceren met het JADE Framework.

2.3. Gebruikte methoden en tools

Een opsomming van de gebruikte methodieken zijn:

- SCRUM;
- GAIA;

Een opsomming van de gebruikte tools zijn:

- Eclipse Kepler;
- Notepad ++;
- SUN JAVA software;
- Command line/terminal;
- Putty;
- JADE Framework;
- NETSTAT;
- Ipconfig/ifconfig.
- Extra geïnstalleerde Linux pakketten:
 - Sysstat;
 - Iostat;
 - Screen;
- Batch/Shell/Bash scripting Language;
- jSoup HTML Parser;
- Encog Artificial Intelligence Framework for Java;

2.4. Onderzoek

2.4.1. Agent Technologie

Agent Technologie houdt een aantal computer programma's in welke kunstmatige intelligentie gebruiken om te 'leren' en daarmee een proces of een aantal procedures automatiseert. Omdat in dit project een aantal procedures dient te worden geautomatiseerd is er gekozen voor Agent Technologie. Het simuleren van een contactcentermedewerker welke door een belscript manoeuvreert, is een reeks van procedures waarbij er afwisseling kan zitten in het manoeuvreren door het belscript en de verscheidene projecten. Hierdoor is de keuze gemaakt om Agent Technologie, welke de reeks procedures kan automatiseren, te gebruiken bij de implementatie.

2.4.2. Performance meten

2.4.2.1. IOstat

Het programma IOstat wat onder het pakket sysstat is genesteld wordt gebruikt op een Linux systeem om de harde schijf en wisselgeheugen te meten op gedetailleerd niveau. Waaronder het schrijf en lees gedrag van de harde schijf, door deze waardevolle informatie kan er achterhaald worden of er probleem optreed met betrekking tot wachttijden op de harde schijf wanneer meerdere contactcenter medewerkers gebruik maken van de BasiCall software of Simulation Agents actief zijn op de desbetreffende server.

IOstat heeft de mogelijkheid om parameters op te geven voor de tijdsinterval en een aantal rapporten. Indien er specifiek voor een apparaat een performance gemeten dient te worden, kan de naam van het apparaat zoals dit in Linux gebruikt wordt als parameter voor IOstat dienen, om alleen de gegevens van dit apparaat te meten.

2.4.2.2. Top

Top is het standaard programma wat op elke Linux Ubuntu machine aanwezig is, om real-time de huidige processen en beknopte systeeminformatie te weergeven. Deze informatie over apparaten is over

de statische informatie als harde schijfruimte en gebruikte harde schijfruimte. Deze wordt aangegeven in procenten en geeft verder niet de real-time informatie over het gebruik van de apparaten weer.

2.4.2.3. IOPing

IOPing is een programma wat de I/O latency realtime weergeeft. De I/O latency wordt getoond in dezelfde weergave als waarin netwerk latency wordt getoond.

2.4.2.4. Atop

Atop laat de apparaten zien met de meeste memory load wanneer het commando wordt uitgevoerd. Afhankelijk van deze output zal er een lijst worden geretourneerd met daarin de apparaten op volgorde gesorteerd op basis van de meeste load.

2.4.3. Software Agent Platform

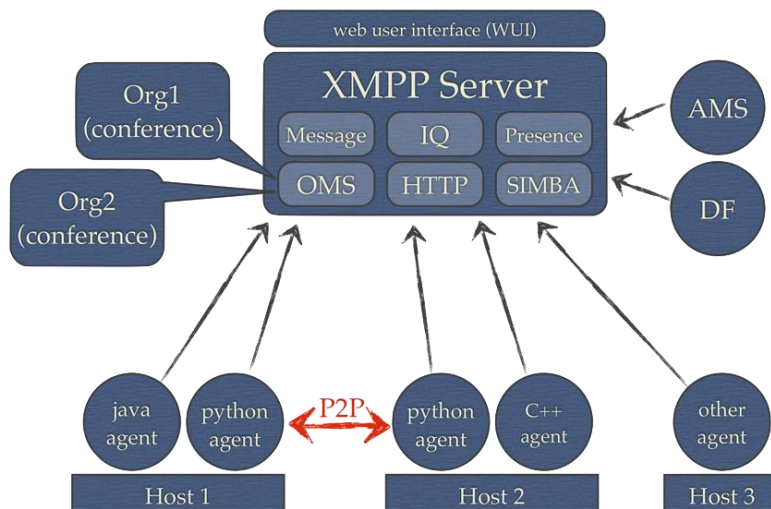
Er zijn meerdere Agent Frameworks beschikbaar om te gebruiken als Agent Platform. Het framework bestaat uit een stabiele omgeving welke een of meerdere Agents kan bevatten. Een Agent in deze omgeving is een software agent, welke autonoom kan handelen en kan communiceren met andere Agents.

Er zijn een aantal Frameworks welke als stabiel worden aangemerkt. Onderstaande tabel is gemaakt met verdere specificaties van deze gekozen Frameworks. Deze Frameworks zijn gekozen omdat er voorkennis aanwezig is van deze Frameworks.

	<u>SPADE</u>	<u>JADE</u>
URL	http://pypi.python.org/pypi/SPADE	http://jade.tilab.com/
Programmeertaal	Python	Java
Laatste update	24-11-2012	26-6-2012
Onder ontwikkeling	Ja	Nee
Stabiel	Ja	Ja
Cross OS support	Ja	Ja
Migratie tussen platforms	Nee	Ja (met plugin)
Communicatie tussen platforms	Ja	Ja
Statisch Framework	Ja	Ja

De BasiCall Software is gemaakt in JAVA. Omdat de BasiCall Software dient te worden aangestuurd door de programmatuur en de compatibiliteit hiervoor zo groot mogelijk te houden is gekozen voor het Agent platform JADE, om te gebruiken in dit project. De kans dat er commando's worden gegeven welke niet ondersteund worden door een andere Framework gebaseerd op een andere programmatuur, is aannemelijk en wordt daardoor niet gekozen.

2.4.3.1. SPADE



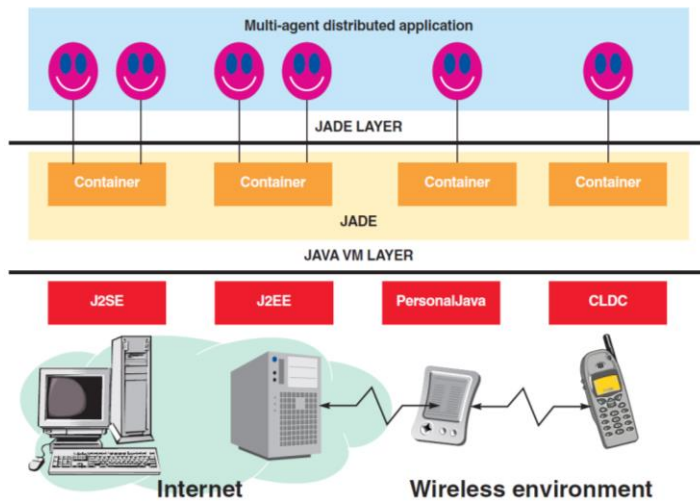
Figuur 2: SPADE Architectuur

SPADE is een agent platform voor meerdere besturingssystemen gezien SPADE meerdere communicatieprotocollen hanteert. De programmeertaal die gebruikt is om dit platform te ontwikkelen is Python. De grafische omgeving van SPADE is HTTP gebaseerd, wat het makkelijk toegankelijk maakt voor meerdere apparaten en besturingssystemen. SPADE is in staat om te communiceren met andere agent platforms waaronder JADE, 2APL en Magentix. Enkele onderdelen van SPADE zijn de ingebouwde webserver functionaliteit, het P2P Framework, de berichten architectuur en het gedragsmodel.

SPADE is gebaseerd op XML gebaseerde protocollen waaronder Jabber, een open protocol gebaseerd op XML wat gebruikt wordt voor “instant messaging” en “presence notification” en zo ook het XMPP protocol wat de opvolger is van Jabber.

Bij SPADE worden de threads ingepland door het besturingssysteem zelf en heeft iedere behaviour van een Agent een eigen mailbox, als zijnde een vorm van een wachtrij. Zo maakt SPADE ook gebruik van Interactie protocollen, door middel van vooraf gedefinieerde sjablonen te gebruiken, zie hiervoor bron 1 in hoofdstuk 9.

2.4.3.2. JADE

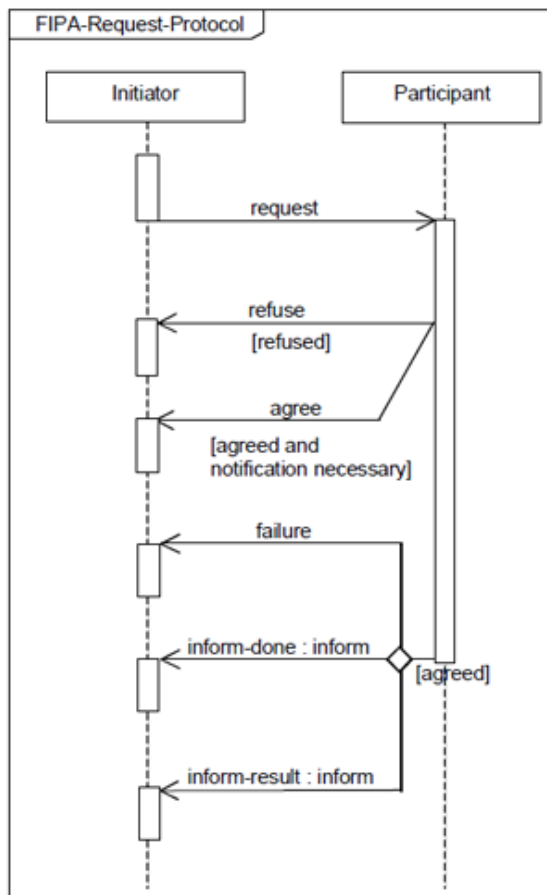


Figuur 3: JADE Architectuur

JADE is een agent platform voor een breed aantal besturingssystemen omdat JAVA op bijna ieder systeem aanwezig is. Het JADE platform heeft een grafische gebruikers interface welke volledig is ontwikkeld in JAVA. De Software Agents zijn in staat om te communiceren met elkaar door middel van de ACL taal, of MTP berichten welke XMPP gebaseerd zijn. De JADE omgeving kent een groot aantal add-ons, welke van pas kunnen komen bij uitbreiding van de ondersteuning naar andere systemen en mobiele platformen als Android. Doordat JADE is gebaseerd op JAVA heeft deze een hoge compatibiliteit met de JAVA gebaseerde software van BasiCall wat het mogelijk maakt om een contactcenter medewerker te simuleren.

JADE is gebouwd gebaseerd op het FIPA protocol wat voor Agents als standaard protocol kan worden beschouwd. De FIPA staat voor “Foundation for Intelligent Physical Agents” en houdt een collectie van standaarden in voor Agent en Multi-Agent systemen zowel als voor de Ontologie en Bericht protocollen.

FIPA wordt ingezet als communicatieprotocol in Multi-Agent systemen. Gezien dit project een monolithisch Agent systeem is, zal de FIPA standaard niet geheel gebruikt worden. Enkel een gedeelte van het communicatieprotocol zal worden gebruikt, zie hiervoor bron 2 en 3 in hoofdstuk 9.



Figuur 4: FIPA Request Protocol

Hierboven het FIPA request protocol waarbij een initiatiefnemer en een participant er aan deelnemen. De initiatiefnemer stuurt een request bericht naar de participant, waarna de participant kan reageren met een Refuse of een Agree.

Indien de participant reageert met een Agree bericht, wordt er een tweede Request bericht verzonden waarop 3 mogelijke antwoorden mogelijk zijn:

1. failure;
2. inform-done;
3. inform-result;

Een failure, geeft aan dat het bericht wel is aangekomen, maar de actie die goedgekeurd was om uit te voeren niet gelukt is.

Een inform-done, zal worden verzonden naar de initiatiefnemer als de actie succesvol is afgerond.

Een inform-result, zal worden verzonden naar de initiatiefnemer indien er een antwoord verwacht wordt. Zodoende zal dit inform bericht de desbetreffende informatie bevatten.

De twee inform berichten, zijn varianten van elkaar met beide het doel om te informeren, echter geeft de inform-result variant ook een resultaat mee wat kan zijn van een berekening.

De berichtgeving is volgens de FIPA standaard.

In dit project wordt bij de communicatie naar de Simulatie Agent gebruik gemaakt van een twee weg systeem. Er wordt een verzoek verzonden en vervolgens is een antwoord bericht genoeg om te weten of de actie voltooid is of niet. Er wordt geen derde bericht verzonden of ontvangen.

Bij de Analyse Agent zal de berichtgeving volgens de FIPA standaard zijn, zie ook bron 10 in hoofdstuk 9.

2.4.4. Artificiele Intelligentie Algoritmen

Voor het manoeuvreren door het belscript van de contactcenter medewerker zal er een algoritme moeten plaatsvinden, dat op basis van een aantal vooraf bepaalde parameters zal worden ingesteld. Belangrijk is dat er door het script heen wordt geklikt om de activiteit te kunnen meten. Door een vooraf ingestelde tijd te gebruiken, zal deze tot deze tijd is verlopen, door het script manoeuvreren. Dit zal ook in moeten houden dat dit niet een belscript is maar dat meerdere achter elkaar ook goed mogelijk moet zijn.

Door een nader te bepalen algoritme te gebruiken wordt een contactcenter medewerker gesimuleerd en kunnen de acties van het belscript worden gemeten aan de hand van de analyse Agent.

2.4.4.1. Neuroph Neural Network Framework

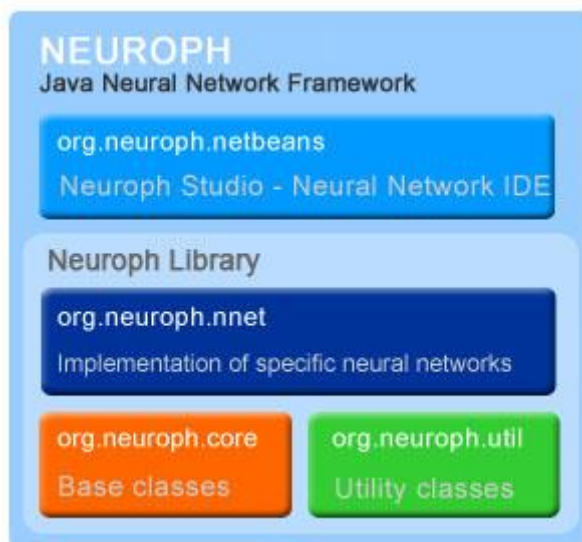
Neuroph is in het leven geroepen om met de meest gebruikte neurale netwerk architecturen te kunnen ontwikkelen. De open source JAVA bibliotheek houdt een klein aantal basis classes in. Deze corresponderen met basis concepten van neurale netwerken. Er is een grafische omgeving beschikbaar om een neuraal netwerk in elkaar te zetten vanuit de aanwezige componenten.

Neuroph maakt gebruik van een wizard welke te starten is middels een .exe bestand waarna verwacht wordt dat er handmatig het .jar bestand wordt toegevoegd aan het desbetreffende project. Er is echter weinig mogelijk zonder grafische elementen omdat deze verwerkt zitten in de open source JAVA bibliotheek. De neurale netwerk elementen die in de open source JAVA bibliotheek zitten, zijn basis en niet voldoende om mee te werken in de scope van dit project.

In Neuroph is het gemakkelijk om een trainingsset te maken en te trainen waarna vervolgens de resultaten te zien zijn in de grafische omgeving van Neuroph. Er zijn veel documenten en diagrammen beschikbaar wat het erg makkelijk maakt om mee te werken, zie bron 18 in hoofdstuk 9.

Pluspunten: Ondersteuning voor een aantal algoritmen, veel documentatie en diagrammen. Makkelijk toepasbaar. Professioneel opgebouwd.

Minpunten: Heeft een klein aantal AI algoritmen, bij veel algoritmen is er een overlappende GUI gebouwd.



Figuur 5: Neuroph Framework Architectuur

2.4.4.2. Heaton Research Encog Machine Learning Framework

Encog is een machine learning framework waarin een breed aantal algoritmes wordt aangeboden voor zowel het normaliseren van data als het verwerken van data. Het Encog Machine Learning Framework heeft onder andere de volgende algoritmen: Support Vector Machines, Artificiële Neuronale Netwerken, Genetisch Programmeren, Bayesian Networks, Hidden Markov Modellen en Genetische algoritmen. Het machine learning framework ondersteund multithreading en kan schalen bij een multicore systeem. De rekenkracht kan ook uit de grafische processor worden gehaald wat desbetreffende algoritme sneller kan verwerken. Er is ook een grafische omgeving aanwezig om de algoritmen te testen. Alle algoritme zitten in een JAVA bibliotheek bestand, bedoeld voor programmeurs. Dit is omdat er geen grafische elementen in het JAVA bibliotheek bestand zijn verwerkt.

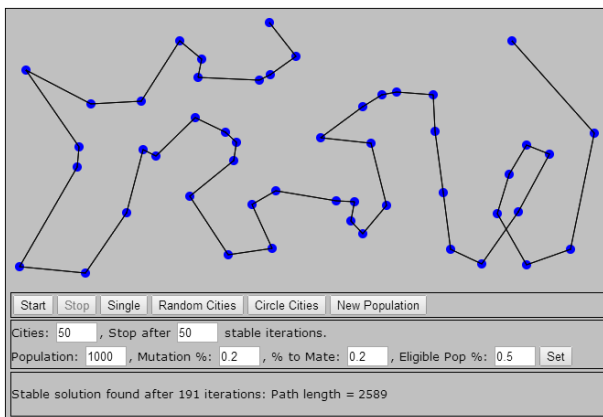
Het Encog Machine Learning Framework is er voor JAVA, .NET en C/C++.

Met name om de vele geavanceerde en gevarieerde algoritmen die het Encog Machine Learning Framework biedt, zal de aanwezigheid van een passend algoritme groot zijn. Er zijn twee algoritmen gevonden welke in de buurt komen:

1. Simulated Annealing;
2. Genetic Algorithm;

Beide zijn een vorm van het “Traveling Salesman Problem” wat een oplossing zoekt door middel van een pad te vinden van een soort coördinaten, zie hiervoor bron 7 in hoofdstuk 9. Hierdoor kan het mogelijk zijn om een pad van afboekcodes en formulierknoppen te creëren, binnen een bepaald tijdsbestek of een aantal klikken.

Het Traveling Salesman Problem is gebaseerd op de volgende vraag: “Geef een lijst van steden en de afstanden tussen ieder paar steden, wat is dan de kortst mogelijke route dat elke stad bezoekt en vervolgens weer terug bij de eerste stad is? “. Zodoende dient ieder script afgesloten te worden en gaat daarna door naar hetzelfde beginpunt in het bescript.



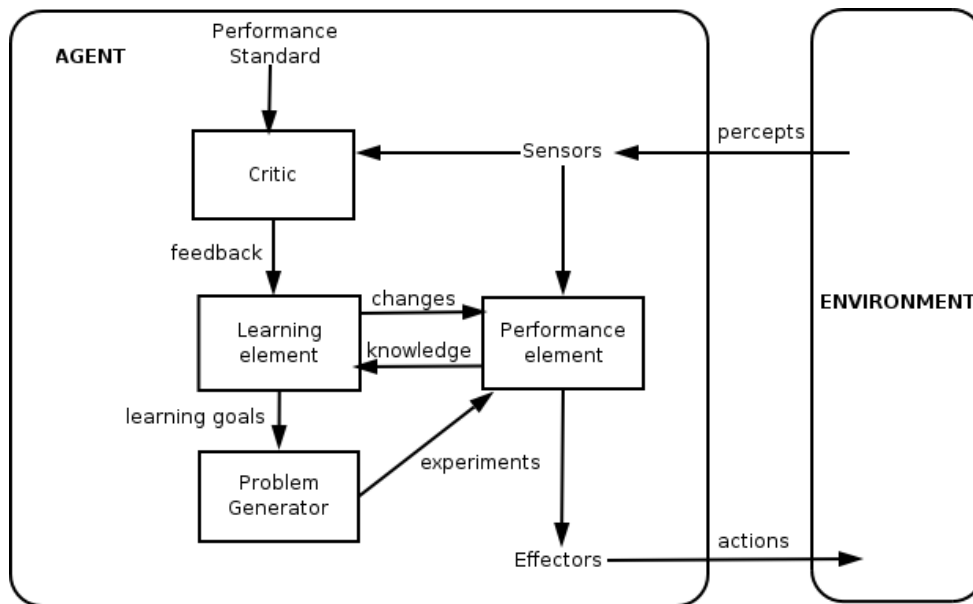
Figuur 6: Traveling Salesman Problem

Pluspunten: Brede ondersteuning en makkelijk toe te passen door middel van de JAVA bibliotheek.

Minpunten: Bij het importeren van de JAVA bibliotheek worden alle algoritmen meegenomen, maar doordat er gebruik gemaakt kan worden van de grafische processor kan de snelheid aanzienlijk sneller zijn bij grotere berekeningen.

2.4.5. Multi-Agent Systemen

Een Multi-Agent Systeem is een systeem dat uit meerdere intelligente software Agents bestaat, zie bron 3 in hoofdstuk 9.



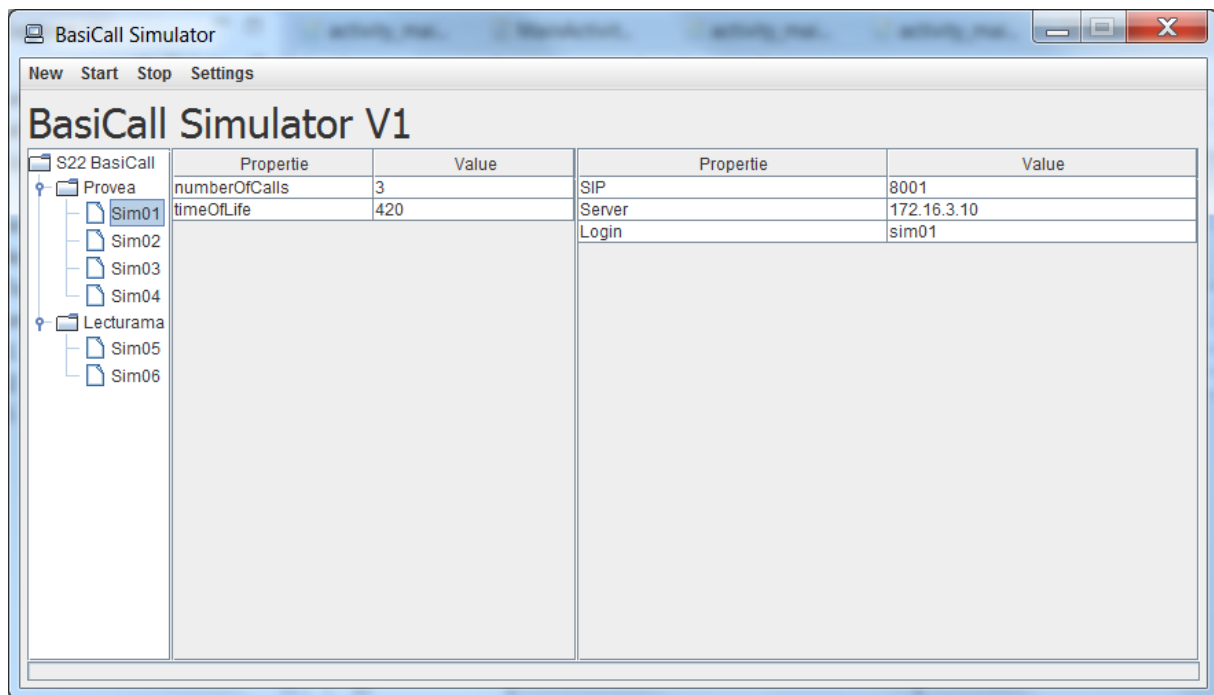
Figuur 7: Intelligente Software Agent Architectuur

De Software Agents dienen een zekere mate van autonomie te hebben en geen van de Agents heeft overzicht op het systeem. Ook dient er in het geval van een Multi-Agent systeem geen aansturing door een centrale Agent te zijn, want zou dit wel het geval zijn dan is het een monolithisch systeem.

In het kader van dit project wordt het wel degelijk aangestuurd door een centrale Agent en dat maakt dit project tot een monolithisch systeem. De centrale Agent kan op verzoek van de gebruiker een Agent starten. Dit kan zowel een Analyse Agent als een Sim Agent zijn en kan ook de analyse resultaten opvragen. Een Multi-Agent systeem kan zelf organisatie vertonen en in het kader van dit huidige project wordt de organisatie geregeld door de gebruiker door middel van de centrale Agent.

2.4.6. Agent GUI Programming

Het is mogelijk om een Agent uit te voeren met een bovenliggende grafische applicatie. Deze applicatie wordt door de Agent uitgevoerd en heeft middels Events de communicatie met de onderliggende JADE Agent. Hieronder afgebeeld een voorbeeld van een grafische applicatie met onderliggend een JADE Agent. De Agent kan middels een uitklapmenu de optie selecteren om een nieuwe Simulation Agent te starten.



Figuur 8: Voorbeeld GUI

Door middel van de JADE bibliotheek te importen heeft de applicatie de mogelijkheid om de functie `onGpouiEvent()` aan te roepen, welke een actie van de applicatie doorgeeft aan de JADE Agent. Hierdoor is het mogelijk voor de JADE Agent om verscheidene acties te definiëren en uit te voeren.

De communicatie van de JADE Agent naar de GUI werkt twee kanten op. De mogelijkheid is er om van de JADE Agent naar de GUI te communiceren, als ook van de GUI naar de JADE Agent.

2.4.7. Softphone

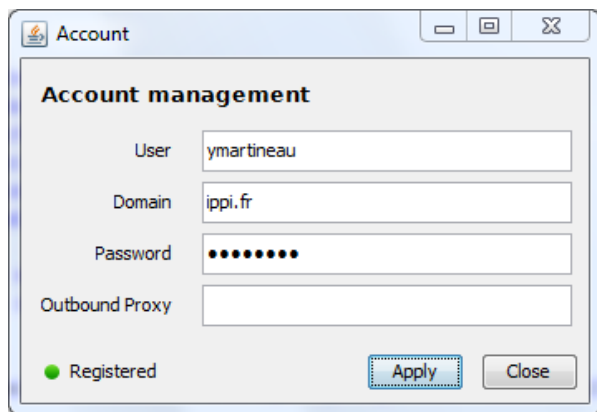
Voor de Simulation Agent is het van belang dat er een programmaticaal toegevoegde softphone in het project zit. Een Softphone is een applicatie voor de computer waarmee bij een Asterisk-server een verbinding kan worden gemaakt. De Asterisk centrale heeft een verbinding geconfigureerd met een VOIP-provider waardoor het mogelijk is om naar bestaande nummers te bellen.

De uitdaging in dit project, is om bij iedere gesimuleerde contactcenter medewerker een SIP-Account te registreren bij de Asterisk centrale. Doordat veel op internet beschikbare softphones werken met configuratie bestanden met de mogelijkheid tot maximaal een SIP-account per programma, is er slechts een geschikte open source softphone gevonden.

2.4.7.1. SIP Peers

Peers JAVA SIP Softphone maakt het mogelijk om een softphone in een JAVA applicatie te implementeren. Er is een gelimiteerde documentatie beschikbaar en de broncode is matig gevuld met commentaar. Echter de broncode is duidelijk gestructureerd. Door middel van het implementeren van de desbetreffende JAVA bibliotheek kan een EventManager voor SIP activiteiten worden aangemaakt, zie bron 12 in hoofdstuk 9.

Door de implementatie mogelijkheid van de JAVA config klasse uit de SIP Peers API is het mogelijk om dynamisch een configuratie per SIP-Account te maken. Zo wordt het mogelijk om meerdere connecties vanuit dezelfde computer te hebben met een Asterisk server. De configuratie houdt minimaal een sip-account naam, een wachtwoord en domein in. Er zijn meerdere configuratie mogelijkheden indien er achter een proxyserver wordt gewerkt of als er handmatig poorten zijn gewijzigd.



Figuur 9: JAVA SIP Softphone GUI

De basis GUI van de JAVA SIP Softphone, geeft de gebruiker de ruimte om een gebruikersnaam, domein, wachtwoord en een outbound proxy server in voeren, om connectie met een SIP server te maken.

2.4.8. HTML parser

Om door het belscript te manoeuvreren is het van belang dat het script kan worden uitgelezen op een manier zodat er een werkbare situatie ontstaat. Dit dient dan te gebeuren door een HTML Parser. Het belscript bestaat uit PHP pagina's waarvan de HTML output zal worden verwerkt door de HTML Parser. Hierdoor kunnen er keuzes in het belscript gemaakt worden en gebaseerd op die keuzes, kan een stap terug of een stap verder worden gezet. Doordat HTML 4.0 wordt gebruikt in de BasiCall Software is het niet mogelijk om HTML 5 Parsers te gebruiken, vanwege dat niet alle elementen herkent zullen worden omdat deze deprecated zijn.

2.4.8.1. HTML Validator

De HTML Validator is een implementatie van het HTML5 parsing algoritme gemaakt in de programmeertaal JAVA, zie bron 19 in hoofdstuk 9. De parser is ontworpen om te dienen als vervanging voor de al aanwezige XML parser, welke ondersteuning biedt voor het parsen van XHTML 1.x scripts. Er is sprake van een basale functionaliteit, voor applicaties welke een eigen mogelijkheid hebben om te lezen en te schrijven.

De HTML Validator is een uitbreiding op het al aanwezige SAX, DOM of XOM API en vereist, dat de gekozen API wordt geïmporteerd. De HTML Validator is echter niet in staat om te werken met formulieren en kan deze niet bevestigen. Omdat de belscripten gemaakt zijn in PHP waarbij de waarden via een formulier worden doorgegeven, kan de HTML Validator niet gebruikt worden in dit project.

2.4.8.2. jSoup

De jSoup bibliotheek is een API gebaseerd op de DOM API, CSS en jQuery methoden. Het is mogelijk om met jSoup door middel van een pad het bestand of URL als parameter op te geven en de HTML te parsen. De parser kan met de elementen uit het bestand of via de URL werken om vervolgens waarden aan te passen en formulieren te bevestigen. Dit maakt jSoup tot een zeer geschikte API om te gebruiken in dit project, omdat er veel met formulieren zal worden gewerkt, zie hiervoor bron 9 in hoofdstuk 9.

2.4.9. PDF Library

Om de analyse in rapport-vorm te kunnen tonen is er gekozen om het rapport in PDF formaat beschikbaar te stellen. Er is gekozen voor PDF omdat dit formaat cross-platorm ondersteund wordt.

2.4.9.1. Apache PDFBox

De Apache PDFBox is een open source library die beschikbaar is gesteld voor derden om er PDF bestanden mee te creëren als manipuleren. De Apache PDFBox is nog in ontwikkeling en heeft nog geen volledige ondersteuning voor alle functionaliteiten. De Apache PDFBox is verder een vrij grote library die de JAVA Applicatie behoorlijk zal doen toenemen. Op de langere termijn worden er wel meer opties voor formulieren, tabellen en gesigioneerde bestanden aangeboden die erg nuttig kunnen zijn. Deze opties zijn echter niet op dit moment beschikbaar of in ontwikkeling, zie hiervoor bron 15 in hoofdstuk 9.

2.4.9.2. iText Programmable PDF

De iText PDF library is een kleine maar krachtige library die functies heeft voor het creëren, manipuleren en muteren van een PDF. Deze PDF library is gericht op onder andere JAVA ontwikkelaars, zie hiervoor bron 4 in hoofdstuk 9. De iText library is inmiddels qua ontwikkeling stabiel en wordt wel onderhouden en mogelijk nieuwe functies. De basis is echter al stabiel en kan gebruikt worden in applicaties. Omdat de iText library ook gebruikt kan worden op Android, een mobiel platform, neemt de iText library weinig schijfruimte in gebruik.

2.4.10. Systeem meetpakketten

Om de analyse gegevens te verzamelen van de actieve processen moeten de al aanwezige mogelijkheden voor het verzamelen hiervan onderzocht worden.

2.4.10.1. HypericSigar

HypericSigar is een opensource javabibliotheek bestand wat het mogelijk maakt om via de JAVA programmeertaal een Linux systeem de I/O apparaten te benaderen. Dit heeft als doel om de apparaat gegevens te kunnen uitlezen en het kunnen meten van I/O apparaten zoals het aantal schrijfacties op de harde schijf.

Doordat HypericSigar de mogelijkheid heeft om specifieke gegevens op te vragen is dit een zeer interessante en goede optie om te gebruiken. De javabibliotheek wordt nog geregeld vernieuwd met verbeteringen en extra opties, zie hiervoor bron 11 in hoofdstuk 9.

2.4.10.2. Slamd

Slamd is een opensource javabibliotheek die het in staat stelt om programma's aan te sturen via een server/client constructie die standaard geïmplementeerd dient te worden. Zo zal er een aparte server en client applicatie voor het gebruik hiervan te worden geïmplementeerd. Via Slamd kan er dan meetgegevens opgevraagd worden.

Deze meetgegevens worden geretourneerd als zijnde de output van het programma. De output dient nog gefilterd te worden voor er specifieke meetgegevens uitgehaald kunnen worden om mee te werken. Dit project is echter al enige tijd ongemoeid gelaten en lijkt erop dat het project niet meer in ontwikkeling is, zie hiervoor bron 23 in hoofdstuk 9.

2.5. Testen

2.5.1. Tests omgaande Analyse Agent

De analyse Agent zal actief blijven aan de hand van vooraf meegegeven parameters en zal gestart worden via het JADE platform om te testen of deze correcte waarden krijgt. De Linux Ubuntu server editie computer bevat het pakket sysstat en heeft voldoende schijfruimte om een 2 MB aan data op te slaan. Indien succesvol is er geen Analyse Agent meer actief en is er een bestand achtergelaten met daarin de verzamelde data op de computer waar de Analyse Agent is gestart.

Het referentiekader wordt hierbij afgebakend door de waarden die verzameld worden door iostat, die worden opgeslagen en die leidend zijn voor de rest van de analyse. Indien een berekening een afwijkende waarde toont, kan door middel van deze verzamelde gegevens te bekijken worden herleid of dit correct is, of de berekening correct is uitgevoerd en de analyse juist is.

2.5.2. Tests omgaande Simulation Agent

De test procedure is op een BasiCall server welke een JADE omgeving bevat en er zal een Simulatie Agent gestart worden met de parameter hoeveel clicks en hoeveel seconde deze actief moet blijven. Deze zet een nepgesprek op met de Asterisk server en simuleert actie op de webserver. Indien dit goed gaat is er een SIP account ingelogd op de asterisk server en is er Apache en/of MySQL activiteit op de webserver.

Er zal een softphone worden gebruikt om een telefoon te laten inloggen op de server, welke benodigd is voor een Simulatie Agent om in te loggen op een project.

2.5.3. Tests omgaande de grafische agent applicatie

De grafisch applicatie zal worden getest door middel van het activeren van de knoppen en vervolgens het uitvoeren van de aangegeven actie. Een knop dient een logische benaming te hebben wat de actie goed omschrijft. Teksten dienen binnen de kaders te passen. De grafische applicatie zal ook groot genoeg moeten zijn om de inhoud te kunnen weergeven, maar indien objecten over andere objecten vallen is dit niet voldoende. Naarmate de berekeningen vorderen en ingevoerd zijn zullen deze door middel van het draaien van de applicatie goed bevonden worden, of eventueel in een later stadium, waarin er meer tekst hoort te komen, worden aangepast. Indien de applicatie gesloten wordt zal de applicatie dit ook moeten doorvoeren naar de overige actieve Agents welke correct afgesloten dienen te worden.

De Simulation Agent wordt gerepresenteerd door een grafische klasse, de ControlGUI. Door middel van de Agent Technologie wordt er gecommuniceerd tussen de Simulatie Agent en de grafische applicatie. Deze grafische weergave biedt een overzicht aan de gebruiker met knoppen en tabellen. In de grafische weergave kunnen acties worden gestart. Een simulatie of analyse kan dan worden gestart en de resultaten vervolgens worden opgevraagd van een analyse. Indien op de knop Quit wordt gedrukt zullen de actieve Agent instanties worden gestopt en vervolgens het JADE Framework worden gestopt.

3.2.1. ControlGUI

De ControlGUI is de grafische weergave van de SimulatieAgent. De ControlGUI biedt de gebruiker een overzichtelijke weergave aan. De grafische weergave bevat de statusinformatie van de actieve Sim Agents die in dit project gedefinieerd zijn en de mogelijke opties die ter aansturing van het geheel dienen.

De ControlGUI zal de acties doorgeven aan de SimulatieAgent. De SimulatieAgent dient deze actie uit te voeren. Om vanuit een grafische weergave acties aan een JADE Agent door te geven is een functie onGuiEvent benodigd. Indien de gebruiker gegevens nodig heeft kan de gebruiker via de grafische weergave, de SimulatieAgent direct aanspreken. De SimulatieAgent kan alle actieve Agents aanspreken. De ControlGUI heeft een ingebouwde timer om het overzicht van alle actieve Agents te vernieuwen door de SimulatieAgent direct aan te spreken.

3.3. Sim Agent

1. Start een BasiCall instantie in simulatie modus op de client;
2. Gebaseerd op een vooraf bepaald aantal click's en tijdsinterval aangegeven in seconden zal deze door script heengaan door middel van JAVA commando's in combinatie met machine learning;
3. Er is een Software Agent in JADE gemaakt welke deze aanstuurt tot het moment dat deze stopt met werken, zowel programmaticaal als op basis van een foutmelding.
4. Maakt een verbinding met de Asterisk server op basis van een verkregen SIP account met een programmaticaal ingebouwde softphone.

De BasiCall Agent instantie is verantwoordelijk voor de connectie met de BasiCall server applicatie. De SimAgent zal zich eerst met de ingebouwde softphone registreren bij de Asterisk server waarna door JAVA commando's naar de BasiCall instantie door het belscript gemanoeuvreed zal worden totdat het aantal clicks of de tijd verlopen is.

Indien de SimAgent een stop bericht ontvangt zal deze de BasiCall instantie stoppen, de softphone uitloggen en stoppen waarna de gehele SimAgent zal worden gestopt.

De SimAgent roept een softphone instantie aan en registreert deze met een automatisch verkregen nummer. Dit nummer wordt gegenereerd aan de hand van het aantal SimAgents wat al actief is. De SimAgent zal dan een BasiCall instantie starten. De BasiCall instantie zal met een simulatie account inloggen en met de configuratie optie 'simulatie' op ja. Indien de configuratie optie simulatie op ja staat zal er een BasiCall instantie worden gestart zonder vorm van grafische elementen.

De BasiCall instantie communiceert met de BasiCall server waar het belscript bevindt en zal daar met vooraf gedefinieerde JAVA commando's formulieren doorlopen. Die formulieren worden door de HTML parser gevonden om het mogelijk te maken om door het script heen te manoeuvreren.

De vooraf gedefinieerde commando's houden het volgende in:

1. endOutboundCall();
2. startRecording();
3. stopRecording();
4. volgendeGesprek();
5. beschikbaar();
6. navigeerVorige();
7. callHold(boolean);

De softphone zal zich registreren maar geen geluidskanaal op de client openen. Doordat er meerdere geluidskanalen worden geopend en afgespeeld zal de computer en de geluidskaart dit niet kunnen bewerkstelligen. De audio wordt alsnog wel naar de softphone gestuurd om de datastroom over de netwerkverbinding te laten lopen en zo de realiteit van de simulatie te bevorderen. Zo kan er ook een accuratere berekening worden gemaakt.

3.4. Analyse Agent

De Analyse Agent is alleen op de server aanwezig. Op de server is het alleen nodig om een analyse te starten en de resultaten te kunnen versturen via een of meerdere FIPA-ACL berichten. De resultaten zullen na afloop van een analyse naar de client worden verzonden. De resultaten worden opgeslagen in een data bestand op de client om de performance niet te beïnvloeden op de server ter bevordering van de meting om de meting zo realistisch mogelijke waarden te laten bevatten.

Het data bestand zal na het ophalen ervan worden uitgelezen op de client door middel van het opvragen van een analyse rapport via de grafische weergave. Dit rapport zal door middel van berekeningen een pdf opleveren met daarin de resultaten, de referentiekaders en duidelijk weergegeven welke performance gedurende de analyse is gemeten. Ook zal er een advies in het rapport komen te staan. Dit advies is gebaseerd op de resultaten. Dit advies beschrijft het meest voor de hand liggende knelpunt met betrekking tot de hardware om eventuele hardware performance problemen te kunnen oplossen.

De Analyse Agent zal een iostat programma starten met de door de client aangegeven parameters: Het aantal rapporten en de tijdsinterval aangegeven in seconden. Nadat het programma iostat klaar is en de gegevens verzameld zijn zal de Analyse Agent met een of meerdere FIPA-ACL berichten de resultaten sturen. De gegevens worden op de client ontvangen zodat op de client de gegevens berekend zullen worden. De resultaten van deze berekening kunnen opgevraagd worden in de vorm van een pdf bestand.

4. Implementatie

4.1. De Agent Technologie

Het JADE Framework is op zowel client als server geïnstalleerd en actief. Het JADE Framework wordt gestart met een parameter die staat voor een vooraf ingestelde poort welke vrij is. De standaard poort die wordt meegegeven is 1099. Deze poort is op de meeste computers al in gebruik vanwege de Java Remote Method Invocation service welke standaard op poort 1099 actief is. Omdat JADE gebaseerd is op JAVA zal deze service altijd mee starten. Hierdoor moet deze poort gewijzigd worden indien er communicatie naar andere platformen dient te worden gebruikt.

Hiervoor is netstat een programma wat standaard op zowel UNIX gebaseerde systemen als ook Windows gebaseerde systemen aanwezig is. Netstat wordt gebruikt om een statusoverzicht van poorten te tonen op dat moment. Er kunnen verschillende parameters worden meegegeven aan netstat. Zo geven de parameters -ano de output welke poorten er op dit moment in gebruik zijn van de computer. Zodoende kan er een keuze gemaakt worden welke poort te gebruiken voor de JADE omgeving van de applicatie. In dit project is gekozen om de JADE omgeving te starten op poort 2500.

Er is gekozen om de communicatie tussen platformen op poort 2500 te draaien. Deze poorten zijn op beide servers zowel als op de meeste systemen beschikbaar. Er moet een platform aanwezig zijn op de server en de client. Iedere JADE omgeving zal de computernaam en platform-id mee moeten geven om zo de communicatie op te zetten. Door de host en platform-id parameter mee te geven weet een ander platform als er een verzoek komt waar het antwoord naar gericht zal moeten worden.

Indien de poort al in gebruik is of wordt geblokkeerd in de firewall zal er respectievelijk een timeout optreden. Dit heeft als gevolg dat de gewenste communicatie zal mislukken en er dan geen simulatie en/of analyse mogelijk is.

Omdat de JADE omgeving niet werkt met streams zal dan ook het TCP-IP protocol worden gebruikt en niet het UDP protocol. De VOIP functionaliteit zal wel het UDP-protocol gebruiken ten behoeve van de audio-stream.

Het JADE Framework heeft een parameter genaamd `-gui`. Deze start de standaard grafische omgeving van JADE op om verscheidene test tools aan te bieden. Deze standaard grafische omgeving zal niet worden gebruikt in dit project behalve als test-omgeving. Er wordt handmatig gemaakte grafische applicatie zal worden gestart middels de JADE omgeving. De JADE omgeving zal het volledige pad naar `jade.jar`, `jadeLeap.jar` en `BasiCallAgent.jar` nodig hebben zowel als het pad naar de map die de `.class` bestanden en de bibliotheekbestanden die worden gebruikt.

Hiervoor is een batch script gemaakt welke door middel van de JAVA Runtime de JADE omgeving opstart met de eerder genoemde parameters. In dit batch script wordt standaard de gemaakte grafische applicatie gestart welke het overzicht biedt over de actieve simulaties en instellingen.

4.1.1. Hosts bestand

Het hosts bestand is een bestand op iedere windows/linux computer waarmee je domeinnamen kunt converteren naar een IP-adres. Deze conversie gaat buiten het DNS systeem om. Voor de JADE omgeving is het benodigd dat de hosts bestand aangepast zal moeten worden omdat er communicatie tussen platformen zal plaatsvinden en de conversie van de IP-adressen door JADE puur op basis van het hosts bestand is. Omdat de JADE omgeving gebruik maakt van het hosts bestand als adres conversie van computernaam naar IP-adres zal hierin de desbetreffende computernaam en IP-adres komen te staan.

Het IP-adres van de localhost zal aangepast moeten worden omdat het IP-adres van de localhost wordt verzonden naar de ontvanger en dit mag niet 127.0.0.1 zijn gezien anders de communicatie staakt omdat er geen antwoord wordt verkregen. Het IP-adres 127.0.0.1 komt altijd bij de lokale computer zelf uit. Er wordt geen gebruik gemaakt van NetBIOS namen binnen in de JADE omgeving omdat de JADE omgeving ondersteund wordt op meerdere besturingssystemen waarin UNIX gebaseerde systemen geen NetBIOS kennen zonder aparte programma's daarvoor te installeren.

SUN Java Runtime versie 1.6 moet vooraf geïnstalleerd zijn op de server om de JADE omgeving en de BasiCall Software te kunnen draaien. De BasiCall software is gemaakt in SUN Java versie 1.6 wat dit als afhankelijkheid maakt om deze software zonder problemen te kunnen draaien.

Het hosts bestand is dan wel niet een dynamische aanpak, maar maakt het wel mogelijk om zonder extra benodigdheden een communicatie op te zetten tussen verschillende besturingssystemen.

4.1.2. Simulation Agent

De Simulation Agent is de aansturing van de simulatie applicatie. Deze klasse wordt standaard opgestart en geeft aansturing aan de Agents. Via de grafische klasse is er ook inzage op de status en instellingen van de Agents en de applicatie zelf.

De Simulation Agent bestaat uit een grafische JAVA klasse en een JADE Agent. De grafische klasse is gekoppeld aan de JADE Agent. De JADE Agent wordt meegegeven in de constructor van de grafische klasse `ControlGUI` om deze te kunnen aansturen. Er worden drie verschillende perspectieven weergegeven in de grafische klasse. Deze zijn voor het projecten overzicht, Agent overzicht, VOIP overzicht. Ieder overzicht zal worden gevuld met informatie over de live status van de simulatie als ook de instellingen waarmee de Agent is gestart.

Er is voor deze communicatie tussen deze klassen een aparte klasse gemaakt met daarin vooraf gedefinieerde constanten. Deze constanten worden in beide klassen gebruik. Indien er een wijziging optreedt die effect heeft op de communicatie tussen deze klassen dan hoeft dit alleen in deze klasse gewijzigd te worden. Voor de communicatie van de grafische omgeving naar de Jade Agent is een methode gemaakt in de Agent klasse die de events van de grafische klasse afvangt. Deze methode kan

vervolgens worden aangeroepen vanuit de grafische klasse en zodoende acties uitvoeren in het actieve Jade platform.

Op deze manier kan de grafische klasse actie ondernemen die betrekking hebben op de Agents. Zo kan er ook een Analyse Agent gestart worden op basis van instellingen die in de grafische omgeving kunnen worden ingevoerd. Er wordt op basis van actienaam gekeken welke actie een knop uitvoert. De Simulation Agent zal de iText PDF library gebruiken om de analyse resultaten in een PDF op te slaan in een door de gebruiker in te stellen locatie op de harde schijf.

4.1.3. Analyse Agent

Aan de server kant zal er een JADE Framework en Sun JAVA Runtime actief zijn welke op dezelfde poort luistert als de client. Via deze poort zal de Analyse Agent kunnen communiceren met de client. Wanneer de Analyse Agent opstart staat deze in een standby modus. Met de Standby modus wordt bedoeld dat de Agent wacht op een signaal dat de analyse kan beginnen.

Indien het bericht binnen is dan zal de Agent de cyclus in gang zetten door middel van parameters die uit het bericht kunnen worden gehaald. Deze communicatie wordt afgehandeld door de CommunicationBehaviour waar ook de andere communicatie in verwerkt zit volgens de FIPA standaard. De grafische beheer applicatie zal hierbij een FIPA-ACL bericht naar een ander platform sturen. Dit gebeurt door middel van het format: agentnaam@platformnaam. Waarbij de platformnaam zich ook in het hosts bestand moet bevinden om communicatie mogelijk te maken met andere platformen.

De servers hebben de JADE omgeving ingesteld als dusdanig dat deze mee opstart met de computer. Deze configuratie word gedefinieerd in het bestand init.d van de Linux distributie. Door het toevoegen van een lijn code aan het opstart script zal het JADE Frame ook starten tijdens het opstarten van de server. Het opstart script zal zich in de Scripts map bevinden onder de map Linux in de Dropbox account. Het commando waarmee de JADE omgeving wordt gestart in een aparte screen sessie. Doordat dit in een aparte screen sessie wordt afgevangen is het makkelijk om te beheren. Indien er problemen voorkomen of de applicatie herstart dient te worden heeft dit geen effect op andere processen.

4.1.4. Sim Agent

De Sim Agent is verantwoordelijk voor de simulatie van de BasiCall software, de SoftPhone en het doorlopen van een of meerdere belscripts. De Sim Agent zal beginnen met het maken van een VOIP connectie met de Asterisk Server. Er dient een VOIP connectie te zijn waaraan de BasiCall server kan zien dat een contactcentermedewerker is ingelogd en beschikbaar is voor een gesprek.

Er wordt via de CommunicationBehaviour een start bericht gegeven welke een aantal benodigde gegevens meestuurt. Het aantal clicks, het desbetreffende project en aantal seconden dat de Sim Agent actief is wordt hierbij door de Simulation Agent per FIP-ACL bericht doorgegeven.

Voor een VOIP connectie wordt een softphone gebruikt. Deze wordt ingesteld met een gebruikersnaam, wachtwoord, SIP-account en een ip-adres van de Asterisk server. Een library SIPPeers wordt gebruikt om de softphone in de Sim Agent te gebruiken. De SIP account wordt berekend op basis van het aantal actieve simulaties dat actief is en zal het eerstvolgende beschikbare account gebruiken om de connectie te maken.

Nadat een VOIP connectie is gemaakt zal er een BasiCall Agent software instantie worden geïnitieerd. Dit gebeurt op basis van gebruikersnaam, wachtwoord en project. Zo ook dient er in het configuratie bestand het ip-adres van de BasiCall server en de simulatie-optie op aan te staan. Indien dit IP-adres en/of de simulatie-optie niet correct is ingevuld zal er ook geen simulatie mogelijk zijn. Voor de simulatie zijn speciaal accounts aangemaakt in de MySQL database die niet worden meegenomen in rapportages en niet zichtbaar zijn voor de contactcenters zelf. Indien de Sim Agent deze stappen heeft gevolgd zal de Simulatie pas echt beginnen. De BasiCall Agent software zal inloggen op een project met een simulatie account en het algoritme starten. Dit Handelsreizigersprobleem algoritme zal een pad creëren wat deze gaat volgen, dit pad zal verschillen naarmate het doel anders is.

Het einddoel dit algoritme is niet alleen om elke simulatie tot een succesvol einde te laten leiden. Een belscript kan ook eindigen in een terugbelafsprake of geen-interesse status. Hierdoor wordt er per gesprek een eindpagina willekeurig gekozen en als einddoel meegegeven aan het algoritme. Indien het belscript begint zal er een bandje op de VOIP connectie beginnen te lopen met een voorbeeld gesprek dat continu doorloopt om data over de netwerkverbinding te laten lopen. Dit ter bevordering van de analyse van het systeem.

Het belscript bestaande uit PHP pagina's worden allemaal uitgelezen door een HTML parser jSoup. Deze HTML parser is in staat om formulieren uit te lezen en deze te bevestigen waardoor deze pagina naar een volgende pagina gaat. Doordat de Sim Agent met deze HTML parser in staat is door deze formulieren heen te gaan en met behulp van de BasiCall Agent commando's is het mogelijk om door het belscript heen te lopen met het handelsreizigerprobleem algoritme.

Indien het algoritme klaar en het belscript zowel als het gesprek ten einde is, wordt er een nieuw gesprek gestart. Een nieuw gesprek wordt op hetzelfde project gestart maar het algoritme en het voorbeeld bandje wordt opnieuw aangeroepen. De Sim Agent wordt getoond in een overzicht van actieve Agents indien deze succesvol gestart is. Indien een of meer acties mislukt is zal de Sim Agent niet starten. De Sim Agent zal na een succesvolle start ten alle tijden gestopt kunnen worden en zich verwijderen uit het platform. Na het verwijderen zal er niks meer aan informatie over deze Sim Agent beschikbaar zijn.

De Softphone

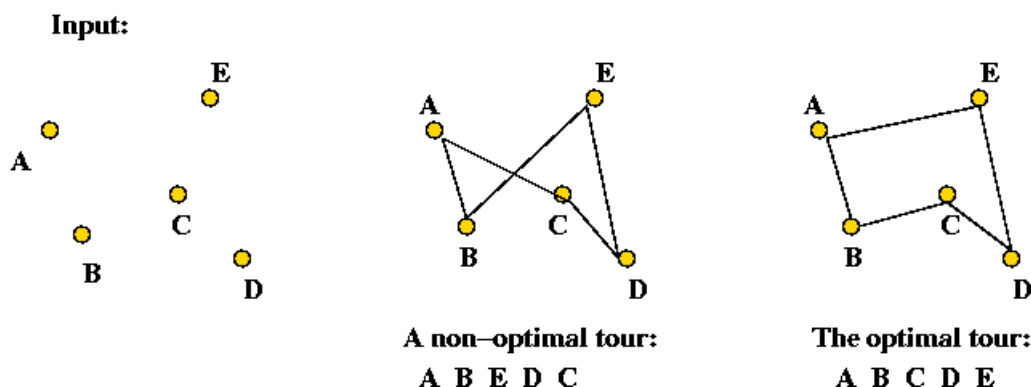
De Softphone zal een berekening uitvoeren om de SIP-account te berekenen, dit houdt het volgende in: Aantal actieve Agents +8001. Het getal 8001 is de eerste SIP-account die gebruikt wordt om te registreren. Vervolgens zal de softphone zich kunnen registreren bij de Asterisk server.

De BasiCall Agent instantie

De BasiCall Agent instantie zal automatisch een gebruikersnaam berekenen om mee in te loggen. Het Sim Agent nummer, wat begint bij 0, wordt aan de tekst "sim" gevoegd om zo een gebruikersnaam te berekenen. Zodoende kan er bijvoorbeeld worden ingelogd met de gebruikersnaam sim0.

4.1.5. Handelsreizigerprobleem algoritme

Het handelsreizigerprobleem algoritme bestaat uit een lijst van punten. Deze punten vergen allemaal een afstand naar elkaar toe. De totaallengte van dit doel zal zo min mogelijk moeten zijn om alle punten langs te gaan. Dit is de simpele uitwerking van het handelsreizigerprobleem algoritme.



Figuur 11: Handelsreizigerprobleem

In dit projecten is er sprake van een vast eindpunt en een drietal aan eindpunten welke aan het begin van ieder gesprek zullen worden bepaald. Zodoende zal er iedere keer een andere weg worden bewandeld indien er ook verschillende waardes worden gegeven aan de tussenliggende pagina's om zo het algoritme te beïnvloeden om andere uitkomsten te produceren.

4.1.6. CommunicationBehaviour

De CommunicationBehaviour is de klasse waarin al de onderlinge communicatie in wordt gedefinieerd. De communicatie is gebaseerd op de FIP-ACL standaard. Deze standaard is gebaseerd op minimaal een vraag, antwoord en bevestiging structuur. Ook wel bekend als een drie-weg. Alle JADE Agents implementeren deze klasse omdat de communicatie hierop gebaseerd is. Om te vermijden dat er andere afwijkingen ontstaan is de communicatie voor alle JADE Agents in een klasse geïmplementeerd.

De communicatie tussen alle JADE Agents zijn in deze klasse gespecificeerd. Alle functionaliteiten worden via een FIPA-ACL bericht aangevraagd en daarop zal worden gereageerd. Op het moment dat het bericht succesvol ontvangen is dan zal desbetreffende actie uitgevoerd worden en een bevestiging terugsturen. Indien een bericht wordt ontvangen die niet conform de FIPA-ACL standaard is of de inhoud is niet bekend dan zal er automatisch een bericht worden gestuurd. Dit bericht zal naar de verzender worden verstuurd met de melding dat het desbetreffende commando niet is geïmplementeerd.

4.1.8. Gebruikte Hardware

Er worden 2 servers gebruikt van BasiCall voor het testen, maar gezien het wisselen van specificaties per server-set voor een klant van BasiCall dient er rekening gehouden te worden met wisselende specificaties.

Server 1: Ubuntu Linux Server Edition

Doel van deze server is het draaien van de webserver waar de scripts op staan en de BasiCall BV server software welke continu aanstaat om inlog verzoeken te beantwoorden van de BasiCall BV Agent software.

Server 2: Ubuntu Linux Server Edition

Doel van deze server is het draaien van Asterisk en de server dient ter opslag van de geluidsopnames. De beide servers hebben geen specifieke hardware eisen, een minimale systeem eis van Intel Core i3 met 16 GB intern geheugen en 1 TB 7200rpm harde schijf en Gigabit Ethernet is aan te houden. Server 1 wordt het meeste belast door de webserver en database welke alle data van de gesprekken opslaat. Onder deze data wordt onder andere naam, adres, woonplaats mee bedoeld als mede de uitslag van het gesprek. Indien een gesprek positief is dan wordt vaak bestempeld met de term 'Sale'. Zo kan er ook sprake zijn van een Terugbelafsprake, Geen interesse of bijvoorbeeld Voicemail als eindstatus.

Deze data wordt realtime weggeschreven, hierdoor wordt het aantal webverzoeken aan de webserver behoorlijk groot. De database heeft bij grotere ophaal-opdrachten last van het locking mechanisme waarmee andere gegevens niet beschikbaar worden indien deze grote opdracht nog bezig is.

Doordat er veel verzoeken zijn naar Server 1 zal dan ook de hardware flink belast worden. De netwerkkaart lijkt voor de hand liggend maar vooral ook de harde schijf wordt belast door het aantal lees/schrijf acties. Server 2 wordt voornamelijk belast door de streaming van de audio. Deze streaming verloopt allemaal via VOIP wat via het netwerk loopt. De audio streaming wordt echter ook opgeslagen. Zowel het gehele gesprek als alleen de voicelog wordt realtime opgeslagen op de harde schijf. De harde schijf en de netwerkkaart zal enorm belast worden en het intern geheugen zal belast worden door de Asterisk server. De Asterisk server zal zijn VOIP verkeer continu door blijven routeren.

5. Resultaten

In dit hoofdstuk wordt een opsomming gegeven van de afgeronde producten welke aan het einde van het project worden opgeleverd. Een beknopte omschrijving is gegeven bij de onderdelen.

5.1. Voltooide producten

- Een Simulatie Agent omvattende:
 - Het ontwerp;
 - De benodigde software om te kunnen draaien;
 - Een implementatie van een Proof-of-concept.
- Een Analyse agent welke:
 - De harde schijf, wisselgeheugen, processor gegevens meet;
 - Deze data verzameld in een tekst bestand;
 - Deze data kan meten afzonderlijk van de eigen analyse Agent instantie;
- Een grafische applicatie welke:
 - Een Simulatie Agent kan starten/stoppen;
 - Een Analyse Agent kan starten/stoppen;
 - Een analyserapport opvragen;

5.2. Overdracht van project

Alle code als ook de uitvoerbare bestanden bevinden zich in een SVN repository. De documentatie en diagrammen bevinden zich in een Dropbox folder. Doordat deze gegevens toegankelijk kunnen worden gemaakt voor andere partijen wordt het op deze wijze overgedragen voor verdere ontwikkeling of mogelijke betrekking tot nieuwe projecten. De documenten en de code zijn vanzelfsprekend en intuïtief ingericht. Er wordt door middel van een overdrachtmoment de code en uitvoerbare bestanden overgedragen aan het afstudeerbedrijf. Dit overdrachtmoment biedt gelijk de mogelijkheid om vragen te stellen en een uitgebreide demonstratie te geven.

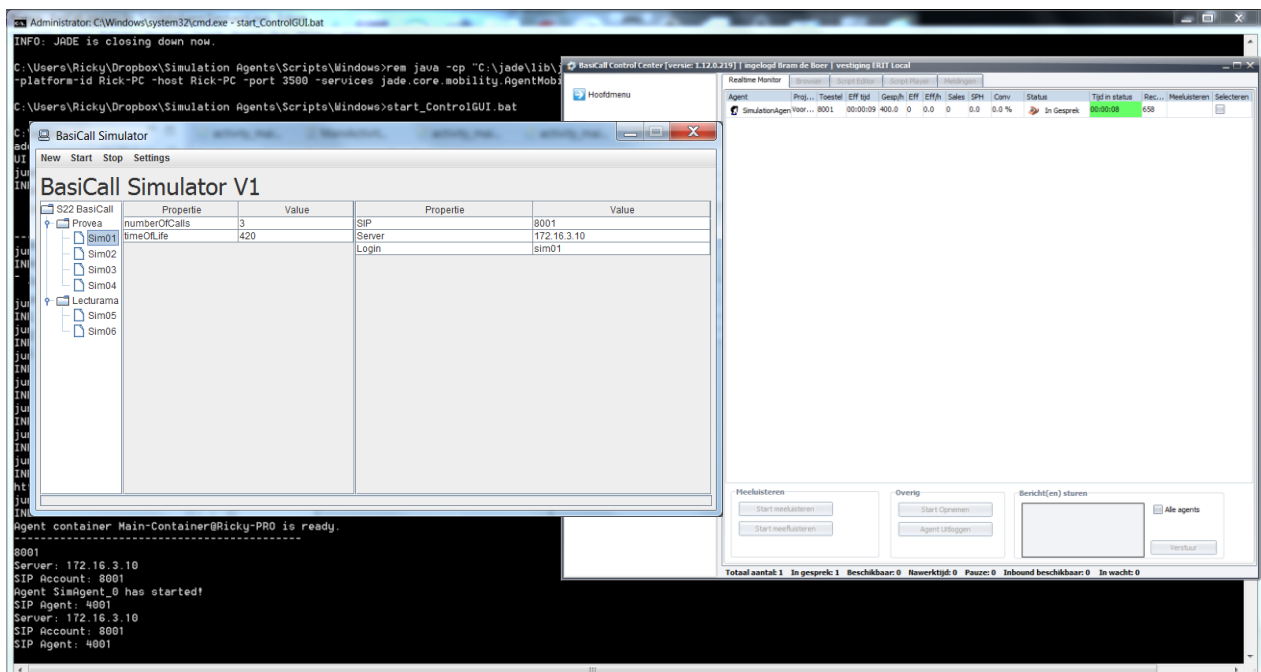
5.3. Demonstratie

Er is nog tijd over voor de eindpresentatie en daarin zal het proof-of-concept nog worden verbeterd en perfectioneert worden. De overdracht van het project zal in deze tijd ook worden voorbereid om dit zo makkelijk mogelijk te maken. De presentatie en demonstratie zal in deze tijd ook worden voorbereid.

De demonstratie op de eindpresentatie zal een video inhouden omdat het lastig is om een server-set van BasiCall neer te zetten en deze volledig werkzaam binnen het schoolnetwerk te implementeren. Dit zou enorm veel tijd vergen wat ten kosten zou gaan van de presentatie zowel als de demonstratie. De demonstratie laat zien hoe de Simulatie Agent zijn werk doet en de Analyse Agent zijn analyse maakt.

Het BasiCall ControlCenter zal op de video naast de gemaakte applicatie open staan. Dit is met name om aan te tonen dat de Sim Agent daadwerkelijk een gesprek simuleert. In het BasiCall ControlCenter is namelijk te zien dat het gesprek actief is en eventuele statuswijzigingen worden hier realtime in een overzicht getoond aan de gebruiker.

Deze demonstratie zal feitelijk aantonen of er een hardware probleem is waardoor de Simulatie Agent of actieve contactcenter medewerker(s) een traagheid ervaren of geluidsproblemen. De demonstratie bewijst ook dat de proof-of-concept werkzaam is volgens de gestelde eisen.



Links in beeld is de beheer applicatie en rechts in beeld is het BasiCall Control Center met de Realtime monitor geopend welke de ingelogde contactcentermedewerkers toont.

6. Conclusie

Het project is door middel van een implementatie van de BasiCall software, Softphone, Machine Learning, JADE, iText PDF library, HypericSigar library en HTML parser software tot een goed eindproduct geleid. Er waren een aantal knelpunten welke het project tot een uitdaging maakte. Er is een simulatie modus in de BasiCall Agent software gebouwd vanwege compatibiliteitsproblemen en de vele debug informatie.

Door een Machine Learning algoritme wordt er succesvol door een script gemanoeuvreerd terwijl er een bandje wordt afgespeeld om de audiostream te kunnen simuleren. Door deze belasting is het nu mogelijk een reële analyse te kunnen maken en een analyse rapport op te stellen om een mogelijk hardwarematig knelpunt te kunnen voorzien. Door de aanwezigheid van vele beschikbare bibliotheek bestanden met name voor Machine Learning, het meten van systeemprocessen en de Softphone was het mogelijk om de simulatie tot een geheel te brengen.

6.1. Antwoorden van onderzoeksvragen

De hoofdvraag van dit project luidt: “Welke direct inzetbare tool dient geïmplementeerd te worden voor het maken van een analyse van de actieve processen die de oorzaak zijn voor de terugvallende kwaliteit bij simultane VOIP gesprekken?”

De bijbehorende antwoorden op de deelvragen zijn:

1. Welke actieve processen hebben effect op de kwaliteit van de VOIP gesprekken?
De netwerkbelasting en de harde schijf lees- en schrijfoperaties hebben het meeste effect op de kwaliteit van de VOIP-gesprekken. Omdat de harde schijf gelijk opneemt wil deze bij een te groot aantal gelijknamige opnames onderbrekingen vertonen omdat de harde schijf moet wachten tot deze de data weg kan schrijven. Indien de netwerkbelasting te hoog wordt dan verliest de VOIP stream te veel pakketten waardoor het gesprek weg kan vallen.
2. Welke methodes zijn er om van de actieve processen meetgegevens te verkrijgen?
Er zijn veel Linux pakketten beschikbaar die de mogelijkheid bieden om de meetgegevens te verkrijgen. De pakketten iostat, systat en top maken het mogelijk om de meetdata van I/O apparaten uit te lezen en als resultaat te retourneren. Dit is exact wat nodig is om de juiste data te verkrijgen om tot een goede analyse te komen.
3. Wat is de definitie van de kwaliteit van een VOIP gesprek en wat is hierbij de minimaal geaccepteerde kwaliteit?
De functionele kwaliteit van een VOIP gesprek moet minimaal voldoen aan de eis dat de audio goed te horen moet zijn voor de versturende als ook de ontvangende kant. De technische kwaliteit moet voldoen aan de eis dat deze minimaal 80% van zijn stream moet behouden om een goede verbinding te zijn.
4. Welke criteria zijn nodig om uit de meetgegevens tot een onderbouwde analyse te komen?
Het aantal actieve Sim Agent instanties, de verschillende projecten waarop is ingelogd en de meetgegevens van de netwerkverbinding, RAM geheugen en harde schijf zijn benodigd om tot een goed onderbouwde analyse te kunnen komen.

Er wordt een direct inzetbare tool gebouwd welke inzicht geeft op de actieve processen die de oorzaak zijn voor de terugvallende kwaliteit van de simultane VOIP gesprekken.

6.2. Evaluatie

Er was wat traagheid met het opzetten van een testsysteem, verder is er altijd iemand in de buurt geweest om operationele vragen te stellen. De technische vragen konden per telefoon af en vaak was hier ook iemand voor aanwezig. Er is voor zowel documentatie als code om feedback gevraagd. Dit was erg fijn en leverde een fijne werksituatie op.

Voor de BasiCall Agent software liep ik tegen een aantal knelpunten op voordat ik de implementatie van de simulatie goed kon voorbereiden, waarbij het grootste probleem de grafische omgeving van de software was. Gezien er in de oude situatie per ingelogde simulatie account een grafische weergave getoond werd van de BasiCall Agent software. Bovendien werden de grafische mogelijkheden niet volledig ondersteund doordat het JADE framework in JAVA versie 1.4 is gemaakt en de BasiCall Agent software in JAVA versie 1.6.

De grafische modus wordt ook niet ondersteund op een commandline, gezien hier geen grafische shell aanwezig is. Vervolgens kwam de debug informatie van de BasiCall Agent software in beeld waarbij de computer vast loopt indien er teveel BasiCall Agent instanties actief waren. Door deze niet te tonen indien de simulatie modus aan staat is dit probleem verholpen.

Indien er teveel simulatiegesprekken open staan waarbij de stream wordt afgespeeld via de geluidskaart, dan wordt de geluidskaart na een aantal Sim Agents al zwaar belast en loopt de gehele computer vast. Deze aanpassingen zijn door Bram verholpen die de GUI elementen en system.out commando's heeft uitgesloten indien de simulatiemodus op aan staat. Het gehele project is dan ook een goede mogelijkheid om diepere kennis op te doen van VOIP systemen en Linux. Het is goed mogelijk door een goede grens afbakening en goede ondersteuning en voorkennis.

7. Aanbevelingen

In dit project zijn een aantal verbeterpunten mogelijk welke een toegevoegde waarde kunnen hebben voor de werking en nut van dit project. De aanbevelingen die eventueel geen onderzoek vereisen voor de implementatie, maar buiten de scope van dit project vallen zijn:

- De netwerkpoort waarop gecommuniceerd wordt dient te worden doorgezet omdat deze standaard geblokkeerd wordt. Er wordt geadviseerd om hier een mogelijkheid op te maken waardoor deze de firewall applicatie aanspreekt om een pop-up weer te geven om deze automatisch toe te staan;
- Behalve een hardware analyse ook een software analyse kunnen maken, zodat er betere overweging gemaakt kan worden waar een probleem ligt;
- Een benchmark applicatie welke automatisch zonder beheer applicatie een aantal Sim Agents start en een analyse start en vervolgens het analyse rapport toont;
- Een analyse per project, indien er meerdere contactcentermedewerkers op verschillende projecten ingelogd zijn heeft dat een negatief effect op de belasting op de server-set;

8. Vervolgonderzoek

Er is de mogelijkheid om na de afronding van dit project nog vervolgonderzoek te doen naar de hieronder genoemde mogelijkheden. Door een van de vervolgonderzoeken te initiëren zal het een positieve wending geven aan het doel van dit project en de distributie ervan.

Er kan vooral vervolgonderzoek worden gepleegd naar andere algoritme die beter of sneller werken dan het huidige algoritme. De manier waarop de meetgegevens worden opgehaald kan misschien door middel van een andere procedure worden opgevraagd. Op dit moment gebeurt dat via de Agent zelf maar dit kan evengoed via een SNMP protocol wat wellicht sneller een resultaat biedt.

Het algoritme voor de simulatie is nu gebaseerd op Machine Learning maar kan wellicht door onderzoek worden vervangen door een neurale netwerk. De distributie gebeurt nu handmatig en onderzoek naar de mogelijkheden om dit proces te automatiseren is te adviseren.

Door gebruik te maken van een hosts file kan het ook voordelig zijn om nader te kijken naar een andere naam translatie methode. Met name een methode die de mogelijkheid biedt om JADE dynamisch te kunnen installeren.

Het is mogelijk om een vervolgonderzoek te initiëren om voor de analyse een verschillende algoritme te onderzoeken om te implementeren. Hierdoor is het mogelijk om te wisselen van doel voor de analyse en wordt dit project multifunctioneel.

9. Bronvermelding

- 1) Aranda, G., & Palanca, J. (2013, January). *SPADE Smart Python multi-Agent Development Environment*. Retrieved from <https://code.google.com/p/spade2/>
- 2) Bellifemine, F. L. (1999). *Jade Agent Technology Framework*. Retrieved from <http://jade.tilab.com/>
- 3) Bellifemine, F. L. (2007). *Developing Multi-Agent Systems with JADE*. Wiley Series.
- 4) Binstock, A. (2006). *iText Programmable PDF Software*. Retrieved from <http://itextpdf.com/>
- 5) Boer, B. d. (2008). *Basicall BV*. Retrieved from <http://www.basicall.nl>
- 6) Bratman, M. (1999). *Intention, Plans, and Practical Reason*. CSLI Publications.
- 7) Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm.
- 8) Heaton, J. (2008). *Encog Machine Learning Framework*. Retrieved from <http://www.heatonresearch.com/encog>
- 9) Hedley, J. (2009). *jsoup: Java HTML Parser*. Retrieved from <http://jsoup.org/>
- 10) Huget, M.-P. (2005). *FIPA Organization*. Retrieved from <http://www.fipa.org/>
- 11) MacEachern, D. (2002, September). *SIGAR - System Information Gatherer And Reporter*. Retrieved from <https://support.hyperic.com/display/SIGAR/Home>
- 12) Martineau, Y. (2007, November). *Peers Java SIP SoftPhone*. Retrieved from <http://peers.sourceforge.net/>
- 13) Miller, J. (1999, January). *The XMPP Standards Foundation*. Retrieved from <http://www.xmpp.org/>
- 14) Nauman, M. (2011, December). *Computer Science Research And Development Unit, Peshawar, Pakistan*. Retrieved from <http://www.csrdp.org/nauman/2011/12/29/agent-mobility-with-jade-and-jipms/>
- 15) Nichols, A. (2011). *Apache PDFBox | A Java PDF Library*. Retrieved from <https://pdfbox.apache.org/>
- 16) O'Hare, G. (1998). *AFAS::Protocols - Agent Factory*. Retrieved from <http://www.agentfactory.com/index.php/AFAS::Protocols>
- 17) Reuter, S. (2006, August). *Asterisk-Java | The free Java library for Asterisk PBX ...* Retrieved from <https://blogs.reucon.com/asterisk-java/>
- 18) Sevarac, Z. (2008, September). *Neuroph Java Neural Network Framework*. Retrieved from <http://neuroph.sourceforge.net/>

- 19) Sivonen, H. (2005). *The Validator.nu HTML Parser*. Retrieved from <http://about.validator.nu/htmlparser/>
- 20) Spencer, M. (1999). *Asterisk.org*. Retrieved from <http://asterisk.org/>
- 21) Spencer, M. (2008, April). *Asterisk RealTime - voip-info.org*. Retrieved from <http://www.voip-info.org/wiki/view/Asterisk+RealTime>
- 22) Telgen, D., van Moergestel, L., Puik, E., & Meyer, J. (2012). Agile Manufacturing Possibilities with Agent Technology.
- 23) Wilson, N. (2005, May). *Slamd — Project Kenai - Java.net*. Retrieved from <https://java.net/projects/slamd>
- 24) Wooldridge, M. (2000). The Gaia Methodology for Agent-Oriented Analysis and Design. *Autonomous Agents and Multi-Agent Systems*.
- 25) Wooldridge, M. (2009). *An introduction to MultiAgent Systems* (2e ed.). John Wiley & Sons.

Bijlage A: Plan van Aanpak

Plan van Aanpak: Simulation of VOIP Traffic with Agent Technology

Auteur	Rick van Rijn
Studentnummer	1581388
Datum	29-08-14
Versie	0.1
Bedrijf	BasiCall BV
Bedrijfsbegeleider	ing. Bram de Boer
Afstudeerbegeleider	Dr. ir. Ander de Keijzer

1. Inhoudsopgave

2. Inleiding	3
3. Versiebeheer	3
4. Beschrijving van het bedrijf	4
5. Huidige situatie	4
5.1. BasiCall software structuur	4
5.2. Werking BasiCall Agent Applicatie	6
6. De opdracht	6
6.1. Aanleiding voor de opdracht	6
6.2. Onderzoeksvragen	6
6.3. Gewenste situatie	7
7. Op te leveren producten	8
8. Projectactiviteiten	9
9. Fasering/planning	10
10. Te gebruiken methoden en methodieken	11
11. Risico's	12
12. Organisatie van het project	13
12.1 Projectdeelnemer	13
12.2 Afstudeerbegeleider	13
12.3 Bedrijfsbegeleider	13
13. Communicatie	14
14. Bronvermelding	15

2. Inleiding

Dit document beschrijft de afstudeeropdracht “Simulation of VOIP Traffic with Agent Technology”. Deze opdracht wordt uitgevoerd bij BasiCall. Dit bedrijf levert diensten op het gebied van VOIP en heeft behoefte aan software waarmee de belasting van hun servers en systemen getest kan worden.

Door de wisselende kwaliteit van VOIP gesprekken en verschillende serverconfiguraties is het lastig te herleiden waar het probleem van de slechte VOIP kwaliteit ontstaat. Door de secundaire ictactiviteiten en -randvoorwaarden in kaart te brengen zal er een direct inzetbare tool worden gebouwd.

Door middel van meetgegevens zal er een analyse worden opgesteld. De analyse zal dan weer de basis vormen voor een gedegen advies op basis van de meetgegevens welke softwarematig zullen plaatsvinden.

Het in kaart brengen van deze secundaire ictactiviteiten en -randvoorwaarden en de oplevering van deze direct inzetbare tool is het hoofddoel van deze afstudeeropdracht.

3. Versiebeheer

Wat	Wanneer	Versienummer
Aanmaken document	29-08-14	0.1

Voor de broncode van de op te leveren tool wordt het versiebeheer via een SVN repository bijgehouden.

4. Beschrijving van het bedrijf.

BasiCall is een op Asterisk PBX (telefooncentrale) gebaseerde contactcenter applicatie. Het BasiCall platform biedt een totaaloplossing voor in- en outbound contactcenters voor zowel traditionele als VoIP telefonie. De applicatie draagt zorg voor een hoog effectiviteitspercentage van de contactcentermedewerkers zonder complexe of kostbare benodigdheden. Dit hoge effectiviteitspercentage wordt bereikt door middel van een geautomatiseerd proces om gesprekken op te zetten.

Er zijn vier man in dienst welke de support dragen zowel als de werkzaamheden verrichten aan de applicatie. Het bedrijf is gevestigd aan De Haag 6 te Houten.

5. Huidige situatie

In deze sectie beschrijven we de huidige situatie van de software en enige informatie over de gebruikte hardware. Dit dient als uitgangspunt voor de beschrijving van de afstudeeropdracht.

5.1. BasiCall software structuur

Figuur 1 geeft een overzicht van het systeem dat bij BasiCall in gebruik is. Het doel van dit systeem is onder meer het faciliteren van contactcenter medewerkers. Deze contactcenter medewerker voert gesprekken met externen en hanteert hierbij een zogenoemd script dat het gesprek moet sturen. Hiervoor gebruikt deze medewerker een werkstation waar de software BasiCall agent op draait. Via de vaste telefoonverbinding worden uitgaande gesprekken opgezet en wordt het script met de externe opgeroepene doorgenomen. De resultaten van het script worden in de database opgeslagen terwijl de VOIP gesprekken via de telefooncentrale gebaseerd op Asterisk werken. De rol van BasiCall Medewerker heeft een controlerende en beheersfunctie.

De BasiCall software is speciaal geschreven voor gebruik in een contactcenter. Er zijn drie applicaties nodig van BasiCall om een werkend geheel te krijgen:

1. BasiCall Agent
2. BasiCall Server
3. BasiCall Control Center

De structuur is een server-client constructie welke bestaat uit twee servers en minimaal een client en het maximum aantal clients is afhankelijk van de hardware.

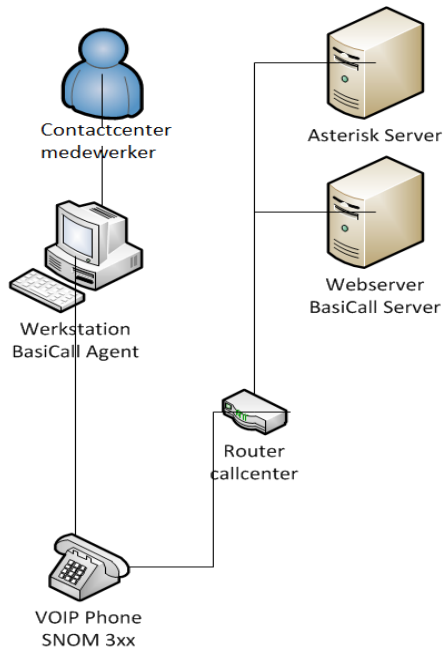
De client applicatie, BasiCall Agent of BasiCall Control Center, komt zoals al eerder is opgemerkt op een werkstation te staan van een contactcenter medewerker. De twee servers bestaan uit:

1. Een Linux Ubuntu 12.0.4 Server Edition met daarop Asterisk geïnstalleerd voor de VOIP communicatie, opslag voor de opgenomen audio en een opensource PPTPD installatie voor de VPN verbinding met het desbetreffende contactcenter te maken.
2. Een Linux Ubuntu 12.0.4 Server Edition met daarop Apache, PHP en MySQL geïnstalleerd om de scripts en rapportage data op te slaan en een BasiCall Server installatie om de informatie van desbetreffende server en de bijbehorende Asterisk server te kunnen doorgeven aan de BasiCall Agent en BasiCall Control Center applicatie.

Er zijn drie type gebruikers van BasiCall:

1. Contactcenter medewerker

- a. Maakt gebruik van BasiCall Agent applicatie voor het voeren van callcenter gesprekken
2. Contactcenter manager
 - a. Maakt gebruik van de BasiCall Control Center. Voert vanuit het contactcenter een monitoring functie uit om overzicht te krijgen over de medewerkers van het contactcenter.
3. BasiCall medewerker
 - a. Beheert de servers en heeft inzage in alle broncode.



Figuur 12: BasiCall Agent overzicht structuur

Op de webserver staat:

1. Project Informatie
2. Beheer scripts
3. Rapportage scripts
4. BasiCall Server
5. Project specifieke uitbreiding scripts

Op de asterisk staat:

1. Asterisk
2. PPTPD (open source VPN daemon)
3. Voicelogs (samenvatting van gemaakte afspraken op audio)
4. Volledige recordings (volledige gesprekken)

5.2. Werking BasiCall Agent Applicatie

De BasiCall Agent applicatie is browser-based en in een JAVA omgeving gebouwd. Na het inloggen komt er een scherm wat Internet Explorer gebruikt met daar omheen een applicatie gemaakt in JAVA. Dit scherm toont de desbetreffende projectinformatie van het project waar de contactcenter medewerker op is ingelogd vanaf de webserver. Wanneer een contactcenter medewerker aangeeft in de applicatie dat deze beschikbaar is zal de asterisk server worden aangesproken om een VOIP gesprek te initiëren.

Indien dit gesprek succesvol tot stand is gebracht zal het script vanaf de webserver getoond worden en kan de contactcenter medewerker beginnen aan zijn gesprek. Tijdens het gesprek zal de contactcenter medewerker door het script manoeuvreren en zullen de gegevens hiervan opgeslagen worden in de database op de webserver en het volledige gesprek zowel als de samenvatting van gemaakte afspraken, ook wel voicelog genoemd, worden opgenomen en opgeslagen op de asterisk server.

6. De opdracht

6.1. Aanleiding voor de opdracht

BasiCall wil graag meten of de door hen toegezegde performance inderdaad geleverd kan worden. Hiervoor moeten tests uitgevoerd worden die zoveel mogelijk overeenkomen met de praktijksituatie. Hiertoe moeten er software onderdelen gemaakt worden die de taken van een contactcenter medewerker simuleren. Voor een dergelijke simulatie wil BasiCall agent technology inzetten.

6.2. Onderzoeksvragen

Hoofdvraag:

Welke direct inzetbare tool dient geïmplementeerd te worden ten behoeve van de analyse van de secundaire ict-activiteiten en -randvoorwaarden die de oorzaak zijn voor de terugvallende kwaliteit bij simultane VOIP gesprekken?

Deelvragen:

5. Welke secundaire ict-activiteiten en -randvoorwaarden hebben effect op de kwaliteit van de VOIP gesprekken?
6. Welke methodes zijn er om van de secundaire ict-activiteiten en -randvoorwaarden meetgegevens te verkrijgen?
7. Wat is de definitie van de kwaliteit van een VOIP gesprek en wat is hierbij de minimaal geaccepteerde kwaliteit?
8. Welke criteria zijn nodig om uit de meetgegevens tot een onderbouwde analyse te komen?

6.3. Gewenste situatie

Om de tests succesvol uit te kunnen voeren moet het systeem een monitoring en testsysteem toegevoegd zijn, waarmee de performance ook daadwerkelijk getest kan worden. Dit moet transparant in het gehele systeem verwerkt worden.

7. Op te leveren producten

De volgende producten dienen te worden opgeleverd:

- Een Linux server met aangepaste BasiCall software;
- Implementatie van een Simulatie Agent voor op de server. Dit moet zo worden uitgevoerd dat het eenvoudig mogelijk is meer agents toe te voegen.
- Front-end applicatie welke communiceert met de server;
- Analyserapport van een Simulatie Agent;

8. Projectactiviteiten

De activiteiten opgesomd in chronologische volgorde:

- Oriëntatie
 - Oriëntatie m.b.t. het onderwerp Agent Technology en VOIP
- Analyse
 - Analyse van het onderwerp VOIP in combinatie met Agent Technology
- Definitie
 - Definiëren van doelstelling en onderzoeksvragen
- Ontwikkelingsfase
 - Ontwikkelen / ontwerpen van een prototype
 - Ontwikkelen van testplan
- Bouwfase
 - Bouwen van een prototype
 - Vastleggen welke afwegingen er zijn gemaakt en de argumenten waarom keuzes zijn gemaakt.
- Test fase
 - Testen van het prototype
- Opleveringsfase
 - Uiteindelijke oplevering prototype en documenten/scriptie

Hierbij zal in onderstaand diagram naar voren komen welke documenten worden opgeleverd en in welke fasering dat zal gebeuren.

Projectactiviteiten				
	Oriëntatie / analyse / definitie fase	Ontwikkel / ontwerp fase	Bouw fase	Test fase
	Acceptatie / implementatie / oplever fase			
Fase	Plan van Aanpak Planning (voorlopig)	Functioneel ontwerp Technisch ontwerp	Prototype Testplan	

9. Fasering/planning

Fase	Datum Deadline	Weeknummer deadline	Aan
Plan van Aanpak Concept	05-09-14	36	Ander de Keijzer
Plan van Aanpak & Planning	12-9-2014	37	Ander de Keijzer, Bram de Boer, Hogeschool Utrecht
Scriptie startdocument	19-09-14	38	Bram de Boer
Scriptie startdocument	26-09-14	39	Ander de Keijzer
Scriptie concept	10-10-14	41	Bram de Boer
Scriptie concept	17-10-2014	42	Ander de Keijzer
Testplan	24-10-2014	43	Ander de Keijzer, Bram de Boer
Testfase	7-11-2014	45	Bram de Boer
Prototype	14-11-2015	46	Ander de Keijzer, Bram de Boer
Opleveren scriptie	28-11-2014	48	Hogeschool Utrecht
Uitloop	16-12-2014	51	Hogeschool Utrecht

Documenten worden per e-mail in bewerkbaar formaat verzonden zodat er opmerkingen geplaatst kunnen worden in het document ten behoeve van de feedback. Er zal een Dropbox folder worden aangemaakt genaamd “Simulation of VOIP with Agent Technology” ten behoeve van het afstudeerproject met daarin de afstudeerbegeleider en bedrijfsbegeleider als begunstigde en projectdeelnemer als eigenaar van de folder.

10. Te gebruiken methoden en methodieken

Voor dit project wordt een groot aantal technieken toegepast. Voor de projectmatige aanpak wordt gebruik gemaakt van scrum. Om met de technische aspecten om te kunnen gaan is kennis van Linux en Windows nodig, alsmede virtualisatie, agent technology en VOIP (zoals geïmplementeerd in Asterisk). Onderstaande tabel geeft meer specifiek weer hoe de agent omgeving geïmplementeerd gaat worden.

Onderdeel	Simulatie Agent	Front-end Applicatie
<i>Programmeertaal:</i>	JAVA	JAVA
<i>Ondersteuning door:</i>	JADE en Java Runtime Engine	Java Runtime Engine
<i>Versie:</i>	JAVA 1.7.x Sun JDK	De nieuwste JRE

11. Risico's

m = minor

Een minor risico is een matig risico wat weinig vertraging oplevert of weinig risico geeft tot falen van het project.

M = Major

Een Major risico is een redelijk risico welke tot behoorlijke vertraging kan opleveren en wellicht de mogelijkheid geeft tot falen van een gedeelte van het project.

G = Giga

Een Giga is een groot risico welke het project enorm kan vertragen en de kans groot is dat het project zal falen.

Risico omschrijving	Beoordeling	Reden beoordeling
Het fout inschatten van tijd	M	Hierdoor kan het gehele project uitlopen en indien niet tijdelijk bijgestuurd wordt dit een G
Kwaliteit van informatie	m	Doordat er vaak op algemeen niveau wordt gesproken over Agent Technology in combinatie met Machine Learning en er weinig code en architecturen hiervan diepgaand worden besproken kan dit tijd kosten.

12. Organisatie van het project

12.1 Projectdeelnemer

Ricky van Rijn,

Student

12.2 Afstudeerbegeleider

Dr. ir. Ander de Keijzer,

Hogeschool Utrecht

12.3 Bedrijfsbegeleider

Dhr. ing. Bram de Boer,

BasiCall B.V.

Bram de Boer is aangewezen als bedrijfsbegeleider omdat hij in bezit is van een Bachelor diploma op het gebied van Informatica en veel werkervaring heeft.

13. Communicatie

Er zijn de volgende afspraken gemaakt met betrekking tot de onderlinge communicatie:

- Documenten en code worden gedeeld door middel van een Dropbox folder genaamd "Simulation of VOIP with Agent Technology".
- 24 uur van tevoren melden van ziekte/afwezigheid bij beide begeleiders.
- Minimaal een keer in de week de voortgang melden en eventueel knelpunten aangeven bij beide begeleiders.

14. Bronvermelding

Machine Learning	Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar (2012). Foundations of Machine Learning , The MIT Press. ISBN 9780262018258.	01-08-12
JADE Agent platform	http://jade.tilab.com/	22-08-14
Linux Performance and Tuning Guidelines	http://www.redbooks.ibm.com/redpapers/pdfs/redp4285.pdf	01-07-07
Linux Ubuntu Server Edition	http://www.ubuntu.com/download/server	22-08-14
BasiCall B.V.	http://www.basicall.nl/	22-08-14
VOIP	http://www.ilocus.com/2007/07/the_10_that_established_voip_p_2.html	01-07-07

Bijlage B: Zelfevaluatie

Ik heb veel geleerd over Machine Learning en artificiële intelligentie omdat er veel informatie over beschikbaar is op het internet en er waren veel kant en klare bibliotheek bestanden die ik kon inzetten in dit project die voor mij in staat hebben gesteld dat ik Machine Learning kan inzetten in dit project. Vlak na het begin heb ik geschakeld van JAVA versie 1.7 naar JAVA versie 1.6 in het begin nadat ik er achter kwam dat JADE niet helemaal compatibel was met de nieuwere java runtime versie 1.7.

Door de vertraging heb ik beter geleerd te plannen, wat ik ook zeker zou doen in de toekomst. Veel dingen die klein lijken die kunnen een enorme klus worden en ook andersom. Dit is een valkuil die ik niet goed had voorzien.

Ik vond het een leuk project om te doen, de kant van kunstmatige intelligentie, Agent Technologie en communicatie protocollen ligt me erg goed.

Ik vind dat ik zelf een actievere rol moet gaan aannemen als het gaat om mijn zwakke kanten aan te pakken, wat onder andere het tijdsmanagement is en ik laat vooral mijn nieuwe ideeën op de vrije loop wat erg leuk is maar niet relevant voor het project. Ik moet hier voortaan actiever mee bezig zijn en mijn eigen inschatting vermenigvuldigen met een getal zodat ik een dichterbij een goede en reële inschatting kom qua tijdsbesteding.

Bijlage C: UML Klassendiagram

