



BDI in games

AfstudeerScriptie



Andrax vd Bogaard

Rotterdam, Augustus 2012

Colofon

■ Bedrijf / Opdrachtgever

Ranj Serious Games

Lloydstraat 21m
3024 EA Rotterdam
Nederland

Tel +31 (0) 10 21 23 101
Fax +31 (0) 10 21 23 104
E-mail info@Ranj.nl

Begeleider **Ivo Swartjes**
E-mail ivo@Ranj.nl

■ Afstudeerder

Andrax van den Bogaard

Zouavenlaan 40
5037MX Tilburg

Tel +31 (0) 634 24 72 54
E-mail AndraxB@gmail.com / a.vandenbogaard@student.fontys.nl
Studentnummer 2139954
Opleiding Software Engineering | Game Design & Technology
Afstudeerrichting Game Design & Technology

■ School

Fontys Hogeschool

Rachelsmolen 1
5612 MA Eindhoven
Nederland

Tel 0877 877 299
Begeleider Jan Oonk
E-mail j.oonk@fontys.nl

Voorwoord

In deze afstudeerscriptie leest u het onderzoek dat ik heb gedaan naar het ontwikkelen van een BDI Framework voor (serious) games. BDI (Beliefs, Desires, Intentions) valt binnen het gebied van AI (Artificial Intelligence). BDI is nog niet in gebruik genomen in games en heeft de potentie het leerproces van serious games goed te ondersteunen. Dit BDI Framework zal bij Ranj Serious Games gebruikt worden om verder onderzoek op dit gebied te ondersteunen. Als programmeur heb ik me hiermee mijn kennis van AI verder kunnen ontwikkelen.

Het afstudeerprofiel voor deze scriptie is Game Design and Technology, maar hiermee zal ik als HBO-ICT student op Fontys Hogeschool Eindhoven ook voor mijn major Software Engineering afstuderen.

Ranj Serious Games wil ik graag bedanken voor de kans om hier af te studeren, in het bijzonder Ivo Swartjes (mijn bedrijfsbegeleider bij Ranj) voor de goede begeleiding van mijn onderzoek en scriptie. Daarnaast wil ik ook Miranda Plomp bedanken voor de goede begeleiding en vrijheid in mijn manier van werken bij Ranj, die ik zo nu en dan nodig had.

Mijn docentbegeleider op het Fontys, Jan Oonk, wil ik ook graag bedanken voor de begeleiding van het proces en de feedback op mijn beroepsproducten.

Van alle docenten bij Game Design weet ik dat ze kritisch kijken naar mijn werk en dat ik een eerlijk antwoord kan verwachten waar ze de nodige tijd voor hebben genomen. Dat waardeer ik enorm en heeft me de afgelopen jaren gemotiveerd om steeds harder te werken.

Andrax van den Bogaard.

Inhoud

Colofon	2
Voorwoord.....	3
Inhoud	4
Samenvatting	6
Summary	7
Verklarende woordenlijst	8
1. Inleiding	9
2. Het bedrijf	10
Visie	10
Geschiedenis	10
Vestiging	10
Producten	10
Werkplek stagiair	10
Organigram	11
3. De Opdracht	12
3.1. Achtergrond	12
3.2. Doel.....	12
3.3. Vraagstelling.....	12
3.4. Aanpak	12
4. Oriëntatie	14
4.1. Aanpak vooronderzoek	14
4.2. BDI	14
4.3. GGR	15
4.4. JADEX	16
4.5. AS3 / Framework.....	16
4.6. AI.....	17
4.7. Bestaande libraries.....	18
5. Onderzoek	20
5.1. AI.....	20
5.2. JADEX	20
5.3. GGR	21
5.4. Deelvraag: “Welke onderdelen van BDI (/het JADEX framework) zijn van belang voor het bedrijf. “	22
5.5. Deelvraag: “Dit BDI framework zal uiteindelijk door Game Designers goed/gemakkelijk bruikbaar moeten zijn, hoe kan hier voor gezorgd worden?”	23

5.6.	Deelvraag: “Voor welke games bij het bedrijf is dit BDI framework van belang. “	23
5.7.	BDI	25
	BDI Software Model	26
6.	Ontwikkeling - BDI Framework.....	30
6.1.	BDI Model	30
6.1.1.	Execution Model.....	30
6.2.	Reasoning System.....	31
6.3.	External libraries	34
6.4.	Tests.....	35
6.5.	Ontology	35
7.	Conclusie(s) en aanbeveling(en)	37
	Evaluatie	38
	Bibliografie	39
	Lijst van figuren.....	41
	Lijst van tabellen	41
	Bijlagen.....	42
I.	Project Initiation Document	43
II.	Blog.....	60
III.	Requirements	66
IV.	Technische Implementatie	69

Samenvatting

Deze scriptie gaat over het afstudeerproject van Andrax van den Bogaard bij Ranj Serious Games in Rotterdam, als Software Engineering student op Fontys Hogeschool Eindhoven met als afstudeerroute Game Design & Technology.

Het doel is een AI (Artificial Intelligence) Framework te maken gebaseerd op het BDI(Beliefs, Desires, Intentions) Framework van JADEX. JADEX is in Java geprogrammeerd en is enorm uitgebreid, hier is de kern uit gehaald en opgebouwd in de programmeertaal die RANJ voor hun games gebruikt: AS3(ActionScript 3).

BDI is een model oplossing voor vraagstukken uit het gebied van Artificial Intelligence(AI) en het staat voor Beliefs, Desires, Intentions. Dit is een stuk software dat er voor gaat zorgen dat een Non-Player Character(NPC) meer menselijk lijkt.

Om dat even toe te lichten:

Beliefs: dingen waar een NPC in gelooft: "Water is blauw".

Desires: een doel dat de NPC graag zou willen halen: "Rijk worden!".

Intentions: dingen die een NPC doet om zijn doel te halen: "Werken als visser"(ooit wordt hij vast rijk!).

Klinkt relatief simpel, maar om dit in een werkend software model te vatten en uit te werken bleek enorm complex. Het Artificial Intelligence gebied is niet voor niets een gebied waar vele onderzoekers in werken!

Ranj Serious Games specialiseert zich in het samenwerken met domeinexperts en de klant om een creatief en leerzaam game concept te ontwikkelen.

Ranj wil in de toekomst haar wereldwijd bekende positie in het creëren van serious games blijven versterken. Vandaar dat Ranj altijd nieuwe mogelijkheden onderzoekt om dit te doen.

BDI is daar één van.

Summary

This is the thesis of Andrax van den Bogaard, a Fontys University of Applied Sciences bachelor student in Software Engineering and Game Design & Technology, finalizing his study with an internship at Ranj Serious Games in Rotterdam, by researching the possibilities of creating a BDI Framework.

The goal is to create an AI (Artificial Intelligence) Framework, based on the BDI (Beliefs, Desires, Intentions) Framework of JADEX. JADEX was programmed in Java and is elaborate, so the core concepts were taken from this and programmed into a new AS3 (ActionScript 3) BDI framework, so Ranj can implement it in their games.

BDI is a model solution for some problems in Artificial Intelligence (AI), it's a piece of software that will make a Non-Player Character(NPC) seem more humane.

BDI stands for:

Beliefs: anything an NPC believes to be true: "Water is blue"

Desires: A goal that an NPC would like to achieve: "Become rich!".

Intentions: actions an NPC has to take in order to achieve his goals: "Working as a fisherman" (someday he'll be rich...!).

This still sounds relatively simple, but to capture this into a working software model and develop the corresponding software is extremely complex. Artificial Intelligence has many people researching it for a reason!

Ranj Serious Games specializes in collaborating with domain experts and the client to develop a creative and educational game concept. Ranj would like to strengthen her already known worldwide position in serious games. This is why Ranj always looks into ways to achieve this. Researching BDI is one of these ways.

Verklarende woordenlijst

NPC Non-player Character

Een character dat gestuurd wordt door een computer, door middel van AI. Een NPC die gebruik maakt van een Agent-Model als het BDI-model is een IA (Intelligent Agent).

IA Intelligent Agent [1]

Een Intelligent Agent is een autonome entiteit die observeert door middel van sensoren, handelingen uit voert op een omgeving, door middel van actuatoren en zijn activiteiten rationeel uit voert om doelen te bereiken.

Agent

Een autonome entiteit die handelingen uitvoert die er in geprogrammeerd zijn.

AI Artificial Intelligence

De studie en het design van Intelligent Agents.

BDI Beliefs Desires Intentions (Model)

Zie: BDI Software Model (in short) op pagina 14.

AS3 ActionScript 3.

Een Object-georiënteerde programmeertaal die gebruikt wordt om voor Flash te programmeren.

NLG Natural Language Generation

Het genereren van zinnen door een machine, die voor de mens als realistisch ervaren (zouden) moeten worden.

Multi-Agent systeem [2]

Simpel gezegd is dit een systeem dat meerdere Agents ondersteund, maar in het geval van Agents betekent dit dus dat ze met elkaar kunnen communiceren. Hoe en wat ze precies communiceren is natuurlijk afhankelijk van het specifieke systeem.

FSM Finite State Machine [3]

Een model voor het gedrag van een machine, bestaande uit een eindig(finite) aantal states, overgangen tussen deze states en acties. In game design wordt dit vaak gebruikt voor AI van NPC's.

BT Behavior Tree

Een Behavior Tree wordt in game design vaak gebruikt om complexe systemen overzichtelijk te houden. Vaak wordt dit gebruikt als Finite State Machines te onduidelijk (zouden) worden.

1. Inleiding

Al vanaf het ontstaan van de informatica(ook wel computerwetenschappen genoemd), probeerden wetenschappers een centrale eigenschap van de mens, intelligentie, zo vast te leggen dat het gesimuleerd kan worden door een machine. Deze tak van informatica wordt Artificial intelligence(AI) genoemd. Deze tak is onderverdeeld in vele divisies, wetenschappers komen namelijk met aanpakken en oplossingen voor specifieke problemen binnen dit veld, maar nog niemand is er in geslaagd dit alles te bevatten in een model dat ook nog eens door een machine gesimuleerd kan worden.

We bevinden ons hier dus op een gebied dat voorlopig nog geen oplossing heeft voor alle problemen die er zijn en misschien komen die er ook wel nooit.

Dit feit heb ik mezelf in mijn hoofd moeten prenten in de loop van mijn onderzoek, omdat ook BDI een enorme achtergrond en onderbouwing heeft. De onderzoekers die het JADEx framework (een implementatie van BDI) ontwikkeld hebben, werken hier al meer dan 10jaar aan en hebben inmiddels een goed werkende aanpak. Maar een eind is ook daar nog niet aan gegeven.

Hoe pak je iets aan waarvan je niet zeker weet wanneer een eind in zicht is? Wanneer is het bruikbaar voor RANJ?

RANJ wilde graag weten of het mogelijk was om uit het enorme framework van JADEx te halen wat ze nodig hebben, zonder afhankelijk te zijn van de oplossingen voor problemen waar RANJ niet eens naar zocht. Zo hoeft RANJ niet de massa aan documentatie en research papers door te lopen om te werken met het stukje dat ze nodig hebben.

In mijn stageperiode heb ik een werkende basis opgezet van een BDI framework, dat bestaat uit de onderdelen die nodig zijn om BDI werkende te krijgen als systeem en de onderdelen om aan de eisen van RANJ te voldoen.

Hoofdstuk 2 vertelt wat meer over het bedrijf, RANJ. Hoofdstuk 3 beschrijft de opdracht die is samengesteld en wordt verteld hoe de opdracht aangepakt gaat worden.

Vervolgens is de Oriëntatie gestart in Hoofdstuk 4 met daarin een vooronderzoek. In Hoofdstuk 5 is het daadwerkelijke onderzoek behandeld, met in Hoofdstuk 6 de Ontwikkeling - BDI Framework.

Met de Conclusie(s) en aanbeveling(en) in Hoofdstuk 7 wordt afgesloten.

2. Het bedrijf

Ranj Serious Games is een wereldwijde autoriteit in de ontwikkeling van serious games. Het maken van serious games bij Ranj is een collaboratief proces tussen gamedesigners en domeinexperts. Sinds 1999 hebben ze al meer dan 400 games ontwikkeld en hebben ze een goed begrip opgebouwd voor het vertalen van leerdoelen naar pakkende gameplay. Ranj blijft ook de creatieve grenzen opzoeken en blijft daardoor vernieuwend, wat al meerdere malen beloond is met awards als bijv. de European Innovative Game Awards.

Visie

Wij willen mensen helpen zichzelf te ontwikkelen op een effectievere, efficiëntere en leukere manier, met serious games. [4]

In de visie van Ranj is de essentie van serious gaming het creëren van ervaringen waarin spelers worden uitgedaagd om real world problems op te lossen. In een spel kunnen ze experimenteren, waardoor ze kennis en vaardigheden opbouwen en bewust worden van hun gedrag in de betreffende situatie. Daarmee worden ze beter in staat om deze uitdaging in de werkelijkheid aan te gaan.

Geschiedenis

Naast de feiten die hierboven al vermeld staan, kijk ik hier ook elke dag uit op een schilderij van Purno de Purno [5]. 2 van de oprichters van Ranj, Marcus Vlaar en Michael Bas, werkten voor Ranj al samen aan deze tv-show die ik vroeger als kind erg leuk vond. Nu dat ik de show terugkijk als volwassene zie ik dat er een hele dubbele betekenis aan de animaties zit, die ik als kind (haast) niet zag; Geweldig!

Vestiging

De studio van RANJ is gevestigd in Rotterdam in de Lloydstraat. Het zit in een oud pakhuis en heeft alles op dezelfde verdieping zitten. De 1^e kant zitten alle werkplekken verdeeld in open lange tafels waar de tafels min of meer ingedeeld zijn naar specialisatie: Projectmanagement, Marketing, Programming, Game Design, Visual Design en de CEO's.

Producten

Een van de meest recente producten van Ranj is Juf in a Box [6] .

De vraagstelling "Gaming voor Zorg - Een op de vier basisschoolleerlingen heeft moeite met schrijven. Hoe maak je leren schrijven leuker?"

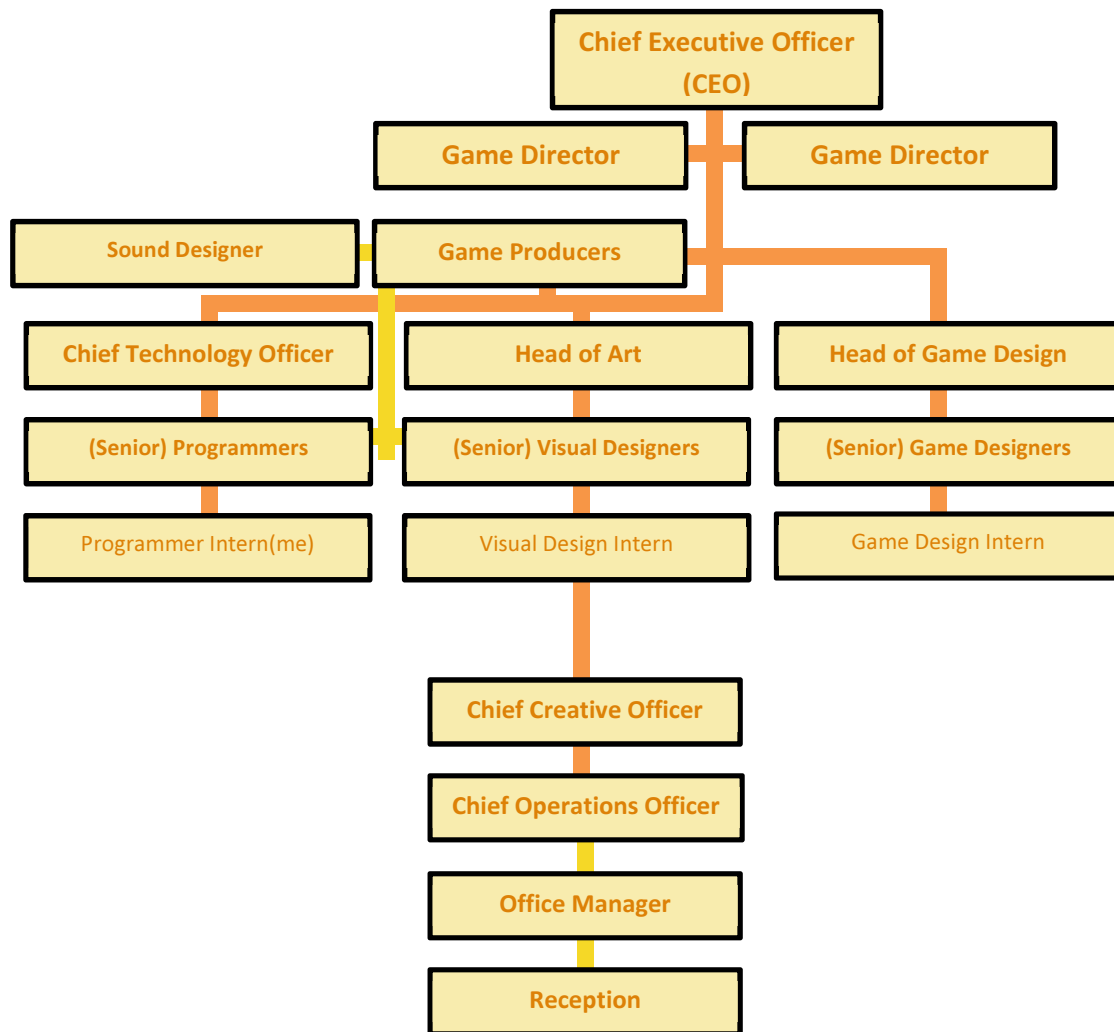
De uitkomst: het product "Juf in a Box". Zoals Ranj vaak doet bij het ontwikkelen van games, werd er samengewerkt met experts uit het vakgebied.

Meestal komen opdrachtgevers met soortgelijke problemen bij Ranj aan en helpt Ranj een duidelijke onderzoeksvraag neer te zetten waar aan gewerkt kan worden.

Werkplek stagiair

Zelf werk ik aan het eilandje(/de werktafel) van de Programmeurs, naast mijn stagebegeleider. Met mijn eigen Workstation met 2 beeldschermen, waar zo'n beetje alle software op zit die ik nodig heb. En ook genoeg ruimte voor zowel een laptop als aantekeningen te maken.

Organigram



3. De Opdracht

3.1. Achtergrond

Momenteel onderzoekt Ranj de mogelijkheden en onmogelijkheden van het gebruik van AI voor het voeren van dialogen met NPC's. In het bijzonder gebruiken ze hiervoor het BDI (beliefs, desires, intentions) modelleerparadigma. In dit paradigma wordt wat een NPC zegt en doet niet bepaald door een script, maar door regels over wat hij weet, wat hij wil en hoe hij dit kan bereiken. Het framework dat Ranj hiervoor gebruikt is JADEX, wat voor een proof of concept prima is maar voor daadwerkelijk gebruik in de toekomst niet geschikt is omdat (1) het een mate van complexiteit bevat die veel verder gaat dan nodig en (2) het een Java framework is terwijl veel van de Ranj games in Flash zijn.

Glenngary Glenn Ross (GGR in het kort), naar de gelijknamige film [7], is een project dat is opgezet door RANJ en TNO¹ om de mogelijkheden van BDI te onderzoeken.

In het GGR project worden de, bij RANJ gebruikelijke, gescipte scenario's vervangen door een BDI-Agent. Waardoor alle NPC's in een serious game redeneren over hun eigen *beliefs*, *desires* en *intentions*. Dit wordt gebruikt om een realistische conversatie te hebben tussen speler en Agent [8].

Hierna wilde Ranj zien of het haalbaar is hier een eigen Framework van te bouwen in AS3, waarin enkel de nodige functionaliteit aanwezig is.

3.2. Doel

Het opzetten van een kernachtig en simpel BDI framework in AS3, gebaseerd op de principes van JADEX. Welke aspecten van JADEX zijn hiervoor essentieel en welke zijn overbodig?

3.3. Vraagstelling

Hoofdvraag:

"Hoe ga ik een BDI(beliefs, desires, intentions) framework in AS3 maken waar Ranj baat bijheeft, gebaseerd op de principes van JADEX."

Deelvragen:

"Voor welke games bij het bedrijf is dit BDI framework van belang. "

"Dit BDI framework zal uiteindelijk door Game Designers goed/gemakkelijk bruikbaar moeten zijn, hoe kan hier voor gezorgd worden?"

"Welke onderdelen van BDI (/het JADEX framework) zijn van belang voor het bedrijf. "

3.4. Aanpak

Iteraties

De aanpak die Scrum [9] hanteert is toegepast om 4 relatief korte iteraties aan het onderzoek en de ontwikkeling te geven, om de haalbaarheid van het project te waarborgen. Aangezien er geen zekerheid van slagen was bij aanvang, is het goed om tussentijds te kunnen evalueren wat wel/niet haalbaar is. De iteraties zijn als richtlijn gebruikt om de tijd tussen onderzoek en realisatie te balanceren.

¹ Nederlandse Organisatie voor Toegepast Natuurwetenschappelijk Onderzoek. (<http://www.tno.nl>)



Onderzoek

Het onderzoek document zal van week 1 t/m 16 uitgewerkt en uitgebreid worden.

Het TSP (Tien Stappen Plan) [10] is gebruik om de globale richtlijnen vast te stellen. “Het ontwerpen van een onderzoek” [11] is gebruikt om het onderzoek in de diepte uit te werken.

Het is voornamelijk een deskresearch en er is gebruik gemaakt van de volgende items:

- **Gesprekken met betrokken partijen**
Er zijn gesprekken geweest met de bedrijfsbegeleider (die ook opdrachtgever is) Ivo Swartjes, over BDI en het Framework.
- **Literatuur**
Boeken en papers zijn geraadpleegd om kennis op te doen van het domein en het bouwen van het Framework.
- **Internet**
Websites en online artikelen zijn geraadpleegd om kennis op te doen van het domein en het bouwen van het Framework.

Het onderzoek is gedaan naar de volgende items:

- **AS3 Framework** - Hoe bouw je een modulaair framework en houdt je een ingewikkeld framework overzichtelijk voor andere programmeurs
- **BDI** - Hoe zit BDI in elkaar en wat wil Ranj hiervan gebruiken
- **JADEX + GGR** - Ondersteunend om te bepalen wat Ranj van BDI wil gebruiken, mede omdat ze al bekend zijn met JADEX d.m.v. het GGR project.
- **Ontology** - Waar BDI het mechanisme is, staat Ontology voor de kennis van een Agent, zonder enige kennis draait het mechanisme niet

Framework

Om het framework te verwezenlijken zullen de volgende onderdelen uitgewerkt worden over de loop van de gehele stage (week 1 t/m 19):

- **Requirements**
- **Class diagrams**
- Het **Framework** zelf, bestaande uit:
 - **BDI Engine**
 - **Externe libraries implementeren** – Om het Framework uit te breiden zijn er libraries geïmplementeerd van Ranj (met name de Simulation Engine) en AS3-Signals.
 - **Documentatie** - ASDoc zo goed mogelijk bij proberen te houden, zodat andere programmeurs gemakkelijker inzicht kunnen krijgen in de code
 - **Test base** - Om gemakkelijk te kunnen testen of nieuwe onderdelen van het Framework werken
 - **Scenario's** - Om mee te testen

In het PID (zie: Bijlage I Project Initiation Document) is de uitgebreide beschrijving van de initiële doelstellingen te zien.

4. Oriëntatie

De oriëntatie stond op de planning voor de weken 1-3 en omvat de “Oriënterende activiteiten” en “Analyse” [10], hiermee werd kennis opgedaan over het onderwerp en kon het PID (Bijlage I : Project Initiation Document) opgesteld worden.

In de weken hierna viel de “Terugkoppeling/contractering” [10] plaats.

4.1. Aanpak vooronderzoek

Aangezien het een redelijk ingewikkeld onderwerp is, is er voor gekozen om te beginnen met een kort vooronderzoek, om te kunnen bepalen wat belangrijk gaat zijn in het hoofdonderzoek.

Onderzoeksvragen

Om de Hoofd- en deelvragen van het onderzoek beter te kunnen beantwoorden zijn er nog een aantal onderzoeksvragen opgesteld:

“Wat is nodig om een BDI Framework werkend te maken?”

“Hoe zit JADEX in elkaar?”

“Welke onderdelen van JADEX/BDI worden in GGR gebruikt?”

“Zijn er dingen over AS3 / Frameworks bouwen in AS3 die ik moet weten voor dit project?”

“Hoe zorg ik dat het BDI Framework goed bruikbaar/uitbreidbaar gaat zijn voor andere programmeurs? (om tooling op te maken uiteindelijk)”

Dit vooronderzoek zal geen volledige informatie over de onderwerpen bevatten enkel een basiskennis om uit te leggen wat ze zijn en waarvoor het gebruikt gaat worden in het volledige onderzoek.

4.2. BDI

BDI Software Model (in short)

Het model wordt gebruikt om een **IA** te ontwikkelen, die gebruik maakt van de BDI principes.

Beliefs: van een Agent representeren zijn Beliefs(geloof) alles in de wereld dat bij hem bekend is. Het wordt expliciet niet “kennis” genoemd maar “geloof(/belief)”, omdat het niet perse harde feiten zijn, maar het is wel wat een Agent ziet als de waarheid. Deze Beliefs kunnen in de toekomst dus ook veranderen.

Beliefs zijn het informatieve component van een Agent [12].

Desires: representeren de motivationele toestand van de Agent. Dit zijn verlangens die een Agent graag zou willen bereiken of teweegbrengen.

Intentions: representeren de overwogen toestand van de Agent. Dit is wat de Agent gekozen heeft om te doen. In feite zijn dit de Desires die een Agent actief vervolgd.

Deze concepten(Beliefs, Desires, Intentions) worden gebruikt om de volgende 3 activiteiten uit te voeren binnen het BDI Model:

Een mechanisme om activiteiten op te splitsen of een Plan te selecteren (vanuit een Plan library) uit de executie van momenteel actieve plannen.

Om het met een mens te vergelijken: Als je een tafel in elkaar aan het zetten bent(je **plan**), zijn daar meerdere activiteiten voor nodig. Mocht er ondertussen iemand op je aflopen en vragen met iets(een

nieuw plan!) te helpen kan je beslissen je eigen plan(en dus **huidige activiteit**) even stil te leggen. *Of je huidige activiteit* (bijv. : de spijker er volledig in slaan, die nu halverwege is) nog even af te ronden en daarna te helpen(**nieuwe activiteit**). *Of* natuurlijk “nee” te zeggen.

Voor ons is keuzes maken vanzelfsprekend, maar een Agent heeft hier een mechanisme voor nodig.

Ten tweede, het managen van tijd die gespendeerd wordt aan: “delibereren over plannen” en “het uitvoeren van plannen”.

Om hier een beeld van de schetsen: “Als mens denk je over het algemeen ook even na voordat je iets doet, de ene keer wat langer dan de andere keer. Maar af en toe moet je gewoon een knoop doorhakken en gaan voor de best mogelijke oplossing.” Dat is de activiteit die hiermee bedoeld wordt.

De 3^e activiteit, de plannen maken, wordt niet door het model uitgevoerd maar door de systeem designer.

Het beeld dat in de vorige activiteit geschetst werd voeden wij als mensen met onze eigen kennis, een BDI Agent heeft in feite nog geen kennis dus zal hij hiermee gevoed moeten worden door een designer. Dit geldt overigens ook voor het voorbeeld van de 1^e activiteit: “Keuzes baseer je op bepaalde kennis en je huidige staat van wezen”, deze dingen zal je ook in zekere mate aan de Agent moeten toevoegen.

Deze laatste activiteit: het voeden van kennis aan de agent, blijkt een flink onderdeel te zijn. Dit zal verder onderzocht moeten worden in het hoofdonderzoek. Dit onderdeel staat bekend als **Ontology**.

4.3. GGR

GlennGarry Glenn Ross, naar de gelijknamige film [7], is een project dat is opgezet door RANJ en TNO² om de mogelijkheden van BDI te onderzoeken.



Figuur 1 Screenshot GGR

² Nederlandse Organisatie voor Toegepast Natuurwetenschappelijk Onderzoek. <http://www.tno.nl>

“Binnen het GGR project is een game ontwikkeld, waarin een speler de rol aan neemt van een makelaar en de conversationele BDI-Agent, de *buyer Agent*, neemt de rol aan van potentieel koper, met zijn eigen knowledge, goals, properties, strategieën en een persoonlijkheid. Het doel van het GGR project was om BDI karakters te creëren die wezenlijk verschillend zijn, intern consistent zijn en op een geloofwaardige manier converseren.

Het spel begint in het makelaarskantoor van de speler, aan tafel met de klant, de *buyer agent*. Het doel voor de speler is om een gesprek te hebben met de potentiële koper dat tot een goed resultaat leidt. De speler moet proberen inzicht te krijgen in de wensen van de *buyer agent* en proberen uit te vinden met wat voor persoon hij aan de tafel zit. Tegelijkertijd moet de speler er voor zorgen dat het huis aantrekkelijk over komt bij de *buyer agent* in de hoop het te verkopen. Daarnaast moet de speler ook zorgen dat het gesprek lekker verloopt en hij een goede relatie op bouwt met de *buyer agent*. De *buyer agent* heeft al naar de informatie en foto's van het huis gekeken, dus hij heeft al enige kennis van het huis, maar zijn doel(goal) is om voldoende te weten te komen over het huis, om te zien of het iets voor hem is. De speler is in staat om de *buyer agent* vragen over zichzelf of zijn wensen te stellen, hem informatie(subjectief of objectief) te verschaffen over het huis of om hem een opinie of voorbeeld over bepaalde aspecten van het huis te geven.

De *buyer agent* gebruikt vervolgens de BDI en Ontology(knowledge en persoonlijkheid) om te redeneren over de nieuwe informatie die de speler hem gaf en hier een realistisch antwoord op te geven, terwijl hij zelf ook proactief probeert te handelen.” – vertaald uit [8].

Dit is een belangrijk item om in mijn onderzoek naar te gaan kijken: hier kan ik uit halen wat RANJ ongeveer nodig denkt te gaan hebben van BDI. Door dit project te analyseren kan ik tot een initiële requirements lijst komen, die bespreekbaar is met de opdrachtgever.

4.4. JADEx

JADEx is een BDI Agent Systeem dat ontwikkeld is door de Computer Science Department van de University of Hamburg [13].

Dit systeem is gebruikt in GGR en is in de ontwikkeling van BDI Software één van de best ontwikkelde oplossingen. Naast het GGR project te helpen analyseren is het ook slim om de principes van BDI hiermee beter leren te begrijpen. Dit is natuurlijk maar één manier om BDI aan te pakken, maar wel een veel gebruikte manier. Er zijn ook redelijk veel documentatie en research papers voor, om het concept BDI en de JADEx implementatie te ondersteunen.

4.5. AS3 / Framework

AS3 was voor mij een haast nieuwe taal, mijn stagebegeleider had daarom ook aangeraden voordat ik begon aan dit project al wat AS3 ervaring op te doen. Dit heb ik deels met wat oefening gedaan en deels door een bootcamp samen met Flavour³ in opdracht van het AMC⁴. Naar mijn gevoel was dit voldoende om van start te kunnen gaan met dit project.

“Zijn er dingen over AS3 / Frameworks bouwen in AS3 die ik moet weten voor dit project?”

Een framework is simpelweg een verzameling van softwarecomponenten. Wat er precies binnen het framework valt(en dus binnen deze opdracht) zal bepaald worden in het PID.

³ Flavour playful branding. <http://www.flavour.nl/>

⁴ Academisch Medisch Centrum. <http://www.amc.nl/>

Wat het dus betekent om een BDI-Framework te maken is het volgende:

Er wordt een Framework gebouwd met als Engine het BDI-software model. Overige functionaliteiten bovenop de Engine behoren tot het Framework.

RANJ maakt geen .swc van elke library/framework maar heeft gewoon één groot framework waar de code files direct in staan. De packages moeten dus beginnen met “Ranj.bdi.*” en de nodige referenced/external libraries moeten beschikbaar zijn.

AS3 is Single-threaded.

Dit is een belangrijk verschil met bijna alle programmeertalen waarmee ik heb gewerkt en JADEX is bijvoorbeeld ook in Java (Multi-threaded) geschreven. Dit is een belangrijk item om rekening mee te houden bij de ontwikkeling.

4.6. AI

Om te achterhalen waar BDI vandaan komt ben ik eens gaan kijken naar de achtergrond hiervan in AI. Dit deels om te kunnen achterhalen of er misschien ook wel simpelere/betere oplossingen zijn die tot min of meer de zelfde doeleinden leiden als BDI.

De keuze voor BDI is bij Ranj echter al gemaakt samen met TNO en er is dus weinig tijd gestoken in het kijken naar andere aanpakken. Toch bleek het interessant om hier wat meer over te weten te komen.

BDI, een korte achtergrond in AI

Het BDI model is een Multi-Agent systeem [2], een gebied dat ontstaan is in begin jaren 90.

Opvallend is dat het vakgebied van ontwikkeling voornamelijk **wetenschappelijk** is geweest, terwijl dit Framework gebruikt gaat worden in het vakgebied van **Game Development**.

Ook opvallend is dat door de wetenschappers vaak de vorm van een game wordt gebruikt om het BDI model op te testen.

Alternatieve oplossingen binnen AI

Het doel voor RANJ is om een *efficiëntere* oplossing te vinden dan gebruikelijk binnen haar vakgebied. De alternatieve oplossingen zullen daarom vergeleken worden met gebruikelijke oplossingen uit het vakgebied Game Development.

Daarnaast zijn er in het onderzoeksgebied van AI veel mensen die onderzoek doen naar Agents en in dit geval dus naar Goal-based Agents. Als hier een volledig onderzoek naar gedaan zou worden, ben je goed een jaar bezig om de beste oplossing te vinden. De kans is groot dat je dan na het afwegen van de pro's/con's van mogelijke oplossing, tot de conclusie komt dat je net zo goed een eigen model op kan stellen dat dit weer net even beter aan pakt.

Als we dus naar het gebied Game Development gaan kijken is de meest gebruikelijke aanpak de *finite state machine(FSM)* [3] of de *behavior tree(BT)* [14]. Er wordt nog weinig van deze 2 welbekende aanpakken afgeweken bij games: FSM wordt vaak voor de wat simpelere AI gebruikt en BT's voor de ingewikkeldere AI.

In feite hebben ze allebei hetzelfde doel: de structuur van de AI overzichtelijk te houden en volledig de controle te hebben over elke mogelijke stap die je uit wil werken.

Hier wijkt BDI enigszins af van deze 2 oplossingen: je hebt niet per se de volledige controle over de mogelijke stappen die de Agent zal gaan volgen. Je kan hem wel zo instellen dat de kans groot is dat hij zal kiezen voor hetgeen jij wil, maar het principe achter BDI is dat je de Agent een aantal opties voert en hij zelf bepaalt op basis van zijn Beliefs wat hij uit gaat voeren.

Naar mate de omgeving (waar de Agents zich in bevind) ook wijzigt, zal de Agent hier ook rekening mee houden. In het geval van BDI zit dit ook verwoven in het model zelf, bij een FSM of BT zal je weer een laagje op de juiste plek in de tree moeten plaatsen.

De complexiteit van FSM's neemt gemiddeld kwadratisch toe naar mate het aantal *behaviours* van de Agent stijgt, waar dat bij BDI in een lineaire lijn ligt [14].

Zo splits je in feite de activiteiten en de keuzes van een Agent, i.p.v. dit bijvoorbeeld in één enorme BT te stoppen. Hierdoor kan je bijvoorbeeld psychologische profielen ontwikkelen voor Agents die in talloze games toegepast kunnen worden. En kan je de activiteiten aanpassen per game.

Afsluitend

Eigenlijk was de keuze voor BDI al van te voren vastgelegd, omdat het voortraject samen gestart is met de onderzoekers van TNO. Maar is er nog niet veel onderzoek gedaan naar de alternatieven binnen Game Development. Vandaar een korte kijk hierop, om de keuze te ondersteunen.

Daadwerkelijk onderzoek naar allerlei mogelijkheden binnen de wetenschappelijke kant is niet realistisch om uit te voeren binnen mijn stageperiode, omdat een oplossing gevonden zou kunnen worden in het gehele veld van AI.

JADEX is een goede implementatie van BDI, met name omdat het gebaseerd is op het JADE framework. Het JADE framework [17] is 1 van de weinige Multi-Agent Frameworks die voldoet aan de eisen van de FIPA [18].

4.7. Bestaande libraries

RANJ Libraries

Er staan wat handige dingen in de RANJ library, vooral als je gewend bent aan C#/Java en mis je een aantal dingen in AS3. Hashtables/Enumerators/Uitgebreide Arrays en meer van dit soort classes.

Om ook rekening te houden met het feit dat Game Designers/andere Programmeurs ook met het Framework aan de gang zullen gaan zal er gekeken worden naar libraries in de RANJ library die daar mee te maken hebben. Tijdens het vooronderzoek bleek ook een item van interesse de *Simulation Engine*. In het daadwerkelijke onderzoek zal gekeken moeten worden hoe/of dit in het framework verwerkt kan worden.

AS3-Signals

Het BDI Framework lijkt Events als een redelijk centraal stuk te gaan bevatten (kijkende naar JADEX), vandaar dat ook even het Event Systeem van AS3 kort onderzocht wordt.

Mijn begeleider gaf als tip eens te kijken naar AS3-Signals. Wat me een erg goed alternatief leek voor de standaard AS3 Messaging. Gezien het meer lijkt op C# Events (waar ik beter bekend mee ben) en het er voor zorgt dat Listeners zich op Objecten aanmelden, ipv een Channel gebaseerd op een String. Bovendien kan er altijd nog heel gemakkelijk gewitcht worden naar AS3 Events, omdat Signals deze ook ondersteund.

Daarnaast heb ik even kort een paar performance testjes gedaan met praktisch lege EventDispatchers en Listeners, om te zien of het niet trager was dan het AS3 Event Systeem. En bleek dat Signals gemiddeld zo'n 20% sneller was dan het AS3 Event systeem. Niet veel verschil dus, maar toch ook weer een klein pluspuntje.

5. Onderzoek

Het viel meteen op waarom AI een enorm gebied van onderzoek is: in eerste instantie kan je denken “dit moet toch niet moeilijk zijn?” maar alle complexe handelingen die komen kijken hierbij maken het een omvangrijk geheel. Gelukkig is BDI al een bestaand iets en valt er dus wel redelijk wat informatie over te vinden, maar om het werkend te maken in AS3 en naar de eisen van Ranj, vergde nog flink wat onderzoek.

5.1. AI

“Waar komt BDI vandaan?”

Simpel gezegd komt BDI voort uit de wil en vraag naar meer realistische game ervaringen, voornamelijk bij dialogen met NPC's. Dit is op zich geen belangrijke onderzoeksvraag, maar deze vraag was belangrijk voor de begripsvorming bij de ontwikkeling van het framework.

Zo is er gebruik gemaakt van alle artikelen over AI en dialoog gerelateerde AI in game development, in boeken als de Game Programming Gems series [17] [18] en AI Game Programming Wisdom [19]. Deze boeken hebben enige houvast gegeven in het onderzoekstraject van de relatief simpele omschrijving van BDI, zoals eerder aangegeven in *Hoofdstuk 4.2 BDI Software Model (in short)*, naar een model dat het framework gaat vormen.

5.2. JADEX

Het gebruik van een bestaand framework heeft zo zijn voor en nadelen:

Je kan gebruik maken van de ideeën van JADEX om dit in code om te zetten , deze is echter in JAVA gebouwd met al een bestaand AI Framework als basis, genaamd JADE. Belangrijkste verschil tussen JAVA en AS3: Java is multi-threaded en AS3 is single-threaded. Dit brengt wat grote verschillen in de opbouw van het project.

JADEX is een project dat al een jaar of 10 draait, dit heeft dus enorm veel iteraties doorgaan en is vele male te ingewikkeld om binnen een paar maanden na te bouwen. Om ook de kern uit zo'n enorme massa code te halen is niet fatsoenlijk te doen, zonder volledig te begrijpen wat de kern is die je nodig denkt te hebben.

Het blijft echter nuttig, omdat Ranj gebruikt heeft gemaakt van dit Framework voor hun GGR project. Zo viel er uit te halen wat RANJ belangrijk vond in JADEX.

“Hoe zit JADEX in elkaar?”

De User Guide van JADEX over BDI [20] is relatief kort, maar vertelt goed hoe de concepten werken en ook hoe de basis van de code in elkaar steekt.

Hier uit is een hoop nuttige informatie kunnen halen, wat bij het programmeren altijd aangehouden is als mogelijk aanpak.

Om in detail te gaan over de structuur van JADEX zelf is onnodig bevonden, omdat dit onderzoeksproject gebaseerd is op JADEX zullen de dingen globaal overeen komen. De details over JADEX zelf zijn te vinden op [20].

De grootste afwijking van JADEX zal natuurlijk zijn dat het niet meer multi-threaded is. Er zal dus rekening gehouden moeten worden binnen het BDI model dat dingen op volgorde gebeuren, i.p.v. mogelijk tegelijkertijd.

Uiteindelijk bleek dit het zelfs te vergemakkelijken qua inzicht bij het maken van de Engine. Er gebeuren namelijk nooit dingen tegelijkertijd en je houdt dus gemakkelijker overzicht bij het testen in je ontwikkeling. Bij multi-threading is dit overzicht moeilijker te houden.

Het gebruik van single-threading bij het bouwen van ingewikkelde engines is in bijna alle gevallen aan te raden, multi-threading is iets wat je vaak relatief gemakkelijk kan implementeren als je precies weet hoe je engine draait.

5.3. GGR

“Welke onderdelen van JADEX/BDI worden in GGR gebruikt?”

Na de JADEX onderdelen uit de documentatie beoordeeld te hebben, kon er gekeken worden wat hiervan in GGR gebruikt werd. En zo ook meteen welke BDI onderdelen in GGR gebruikt zijn.

Om het gemakkelijk te maken is een lijstje opgesteld van onderdelen om vast te leggen tot op welk niveau de onderdelen gebruikt werden. De niveaus van gebruik (te zien als *Usage* in de tabel) zijn als volgt:

Volledig gebruikt
Deels gebruikt (mogelijk dus niet essentieel)
Niet gebruikt

What?	Usage	Description/elaboration
Beliefs		
Beliefsets		
Goals - Deliberation - Unique		
Perform Goal <i>Enkel in BuyerAgent</i>		Perform Goals zijn bedoeld om activiteiten uit te voeren.
Achieve Goal		Met Achieve Goals kan een gewenste world state behaald worden.
Query Goal		Query Goals zijn bedoeld om informatieve te vergaren (over andere dingen in de wereld).
Maintain Goal <i>Enkel in BuyerAgent</i>		Maintain Goals monitoren een bepaalde toestand, als deze toestand geschonden wordt zal dit goal proberen de gewenste toestand weer te behalen.
Meta Goal		Meta Goals worden gebruikt voor het meta-niveau van de reasoning. Als er een goal of event uitgevoerd wordt en er is bepaald dat meta-niveau reasoning toegepast moet worden (als er bijvoorbeeld meerdere mogelijke plannen beschikbaar zijn bij dit goal) dan zal gekeken worden naar het bijbehorende Meta Goal.
Plans - name		Een plan bestaat uit de concrete acties die een agent uitvoert om zijn goals te bereiken.
- priority		Priority kan aan een plan meegegeven worden om te bepalen waar voorkeur aan gegeven moet worden bij de



		reasoning.
Body		Bedoeld om te specificeren welke implementatie-class aan dit Plan behoort. Deze class is bedoeld om het plan daadwerkelijk uit te voeren.
Trigger Possibly triggered by:		
- Goal - Messageevent - Factchanged - Factadded - Internalevent		
Precondition		De preconditionie wordt geëvalueerd voor de initiatie van het plan. Als er niet aan de preconditionies voldaan wordt zal het niet toegevoegd worden aan de lijst van toepasselijke plannen.
Contextcondition		De contextcondition wordt geëvalueerd tijdens de uitvoering van een plan. Wanneer deze overtreden wordt zal de uitvoering van het plan gestopt worden.
Waitqueue		When an event occurs, and triggers an execution step of a plan, it may take a while, before the plan step is actually executed, due to many plans being executed concurrently inside an agent. Therefore, it is sometimes possible, that a subsequent event, which might be relevant for a plan, is not dispatched to that plan, because it still has to execute previous plan step, and does not yet wait for the event. To avoid this, each plan has a waitqueue to collect such events.
Parameters/Parameter Sets		Soortgelijk aan belief(sets) Bedoeld om lokale waarden vast te leggen, die nodig of gegenereerd zijn tijdens de executie van het plan.
- value - bindingoptions		
- mapping (internalEvent/ messageEvent/ Goal) <i>Goalmapping is de enige gebruikt in GGR.</i>		Vaak worden parameters gebruikt om parameters vast te leggen die komen uit het goal of event dat dit plan getriggerd heeft. Door middel van mapping leg je deze relatie tussen event/goal en plan expliciet vast.
Events		Over het algemeen is er een plan dat uitgevoerd wordt als een Event ontvangen wordt.
MessageEvent		Events voor communicatie tussen verschillende Agents . Kan een eigen format aan gegeven worden.
InternalEvent		Events die intern binnen 1 Agent voorkomen.

Tabel 1 JADEX BDI Usage in GGR (short table)

5.4. Deelvraag: “Welke onderdelen van BDI (/het JADEX framework) zijn van belang voor het bedrijf. “

Simpel gezegd waren er 2 belangrijke onderdelen ontstaan bij deze vraag:

1. **De lijst van onderdelen die is samengesteld in Tabel 1 JADEX BDI Usage in GGR (short table)**
Dit is met name bedoeld als een uitgangspunt van wat RANJ verwacht van de BDI Engine.
2. **De minimaal nodige onderdelen om de BDI Engine draaiende te krijgen.**
Dit stuk is simpelweg nodig om het geheel draaiende te krijgen, maar staat past volledig vast aan het eind van het onderzoek. Om zo veel mogelijk van deze eisen bij te houden is er het volgende document bijgehouden in Bijlage III Requirements .

Dit zijn de eisen die samen met de opdrachtgever vast te stellen waren na de eerste weken van het onderzoek en waar dus qua realisatie een afspraak over te maken viel. Omdat het op dit moment van het onderzoek(+week 7 van het project) nog niet duidelijk was of het realistisch was om dit ook als Framework uit te bouwen.

In de loop van het project waren deze steeds beter vast te leggen in de vorm van requirements, te zien in Bijlage III Requirements.

5.5. Deelvraag: “Dit BDI framework zal uiteindelijk door Game Designers goed/gemakkelijk bruikbaar moeten zijn, hoe kan hier voor gezorgd worden?”

Dit punt heeft in de ontwikkeling minder aandacht nodig gehad, omdat het voor de ontwikkeling van het Framework eigenlijk neer kwam op de volgende vraag:

“Hoe zorg ik dat het BDI Framework goed bruikbaar/uitbreidbaar gaat zijn voor andere programmeurs? (om tooling op te maken uiteindelijk)”

De tooling is uiteindelijk namelijk waar de Game Designers mee aan de slag gaan, maar om het Framework dus zo op te stellen dat de programmeurs uiteindelijk zo min mogelijk tijd aan ontwikkelen van Tools moeten besteden, werd even gekeken naar beide vragen.

Note: Voor deze onderzoeksvraag was niet veel tijd beschikbaar, maar het was wel handig voor toekomstig gebruik om hier naar te kijken. Kleine dingen kunnen soms ingrijpend zijn voor het resultaat van een framework. Liever van te voren er dus over na denken dan achteraf!

Een handige tool waar Ranj al over beschikt is de Simulation Editor, hiermee kunnen allerlei verschillende properties gemakkelijk door Game Designers ingevoerd worden. Het format van de Simulation Engine, die achter deze Editor zit, is daarom ook al zo vroeg mogelijk verwerkt.

Daarnaast moet het mogelijk zijn om een betere tool te kunnen bouwen voor Ontology (of een extensie op de Simulation Editor), dus de code moet goed leesbaar en te begrijpen zijn. Relatief standaard dingen om dus rekening mee te houden zijn:

- Goede comments in de code
- ASDocs goed bijhouden in classes
- Documentatie van het project bijhouden

5.6. Deelvraag: “Voor welke games bij het bedrijf is dit BDI framework van belang. “

Dit valt onder te verdelen in 2 categorieën:

- **Bestaande games / formats**
- **Toekomstige games**

Om kort proberen te vatten waarvoor het BDI Framework belang zou kunnen hebben is het belangrijk om een aantal *key selling points* te noteren. Deze punten zullen van toepassing zijn op **toekomstige games**. En kunnen ook bepalen op welke **bestaande games/formats** dit van toepassing is.

Key selling points:

1. Psychologische profielen

Beoogd gebruik:

BDI zou bij RANJ voornamelijk toegepast worden op dialogen die NPC's voeren met de speler.

Het zou ook gebruikt kunnen worden voor NPC's die een AI nodig hebben. Denk bijvoorbeeld aan management games, waar je meerdere mensen(NPC's) moet vertellen wat ze doen. Je zou dit kunnen ondersteunen met BDI.

Voorbeeld case:

Een management game in een kantoor. Waarbij je een flink aantal NPC's moet managen, maar waar ze bijvoorbeeld ook zelf taken gaan doen als het wachten ze te lang duurt. Dit zou je normaal simpelweg met een timer doen, maar nu kan je ook echte karaktereigenschappen aan de NPC's hangen. Geduldige NPC's wachten misschien wel 15minuten netjes op je instructies, waar een ongeduldige NPC na enkele minuten wachten zegt "Ik ga wel even wat anders doen hoor". En vervolgens een pauze neemt van 5 minuten, waarin je hem als speler geen directe orders kan geven. Als je de NPC's leert kennen als speler, weet je een beetje wat je van hun kan verwachten en kan je daar dus op in spelen.

Zo heb je misschien wel meerdere psychologische profielen die je aan NPC's zou willen hangen. Aan de hand van BDI kan je deze profielen ook daadwerkelijk implementeren in je spel. Een voordeel is dat deze profielen ook weer te behouden zijn voor andere games.

Key selling point: Psychologische profielen koppelen aan NPC's en NPC's hier naar laten handelen.

2. Feedback op je keuzes

Beoogd gebruik:

Nadat een speler het resultaat ziet van zijn keuzes, wil een speler vaak weten waar en waarom het goed/fout is gegaan. Elke keuze kan gemakkelijk geregistreerd worden, wat feedback mogelijk maakt.

Voorbeeld case:

Neem GGR weer als voorbeeld case, je probeert als speler de klant(een NPC) tevreden te houden en hem een huis te verkopen. Als je de klant vaak dingen probeert te vertellen die niet waar zijn, loop je geheid een keer tegen iets aan dat de klant al wist, dan weet hij dus dat je liegt! -Dit soort momenten zouden een flinke dip in de sfeer van het gesprek op leveren, misschien loopt de klant wel meteen de deur uit. En dit is dus precies het soort data dat prima uit te lezen is.

Naar aanleiding van de data kan hier dus feedback op geven worden. Het kan simpelweg feedback zijn in de vorm van: "Liegt niet te veel tegen de klant, dat vind hij niet leuk en dan vertrekt hij misschien!". Of zelfs een grafische weergave van de sfeer in het gesprek, zodat je als speler precies kan zien waar het mis gaat/ging.

Key selling point: Feedback op je keuzes, zo kan je als speler leren van je fouten.

3. Keuze van gespreksstrategie

Beoogd gebruik:

Als speler kan je bij salesgames getraind worden om producten te verkopen, enkel kennis van je product is daarbij niet voldoende. Elke potentiële klant valt over te halen om iets te kopen, zelfs al hebben ze het niet nodig, daarbij kan de juiste strategie een grote rol spelen.

Voorbeeld case:

Neem GGR weer als voorbeeld case, je probeert als speler de klant(een NPC) tevreden te houden en hem een huis te verkopen. Als je de NPC vaak dingen verteld die hij graag wil horen, ook al zijn ze misschien iets mooier dan de waarheid, dan raakt hij enthousiaster over het huis. Maar vertel je te veel mooie verhalen, dan begint een klant te twijfelen aan de waarheid hiervan. De juiste strategie hierachter te vinden vergroot de kans op verkoop.

Key selling point: Trainen van spelers op de juiste keuze van gespreksstrategie.

Bestaande Games/Formats

Changing the dialog

Changing the Dialog is een interactieve training game voor raadgevende verkopen.

Mogelijke toepassing van BDI hierin:

- Voorkooptechnieken afhankelijk van **Psychologische profielen**.
- **Keuze van gespreksstrategieën:** De speler kan verschillende verkoopstrategieën oefenen om te kijken wat het beste werkt bij de BDI Agent. Afhankelijk van het profiel en ook de Ontology van de Agent kunnen er heel uiteenlopende strategieën ontstaan vanuit de speler, om een verkoop succesvol te maken.
- **Feedback op keuzes:** Aan de hand van BDI en Ontology kan er ook gemakkelijk feedback gegeven worden aan de speler op hoe succesvol de speler was in het overtuigen/aardig zijn/etc...

5.7. BDI

In het vooronderzoek is al een korte intro gegeven op het BDI Software Model (in short). Hier zal in detail ingegaan worden op het volledige BDI Model.

“Wat is nodig om een BDI Framework werkend te maken?”

Na het vooronderzoek dacht ik duidelijk te hebben wat het BDI Model inhoudt. Maar de volgende stap is dit naar code te gaan vertalen. Na hier even goed over na te denken begin je aan jezelf te twijfelen. Snapte ik nou wel echt wat daar stond? Als ik om hulp vraag bij mijn stagebegeleider of op het internet kijk lijkt dat ik het echt begrijp. En toch lijkt het omzetten in code haast onmogelijk om “even” in je hoofd te krijgen. Tot de realisatie komen dat dit misschien wel niet zomaar te doen is voor mij heeft me wat tijd gekost. Gelukkig versimpel ik in dat proces dingen voor mezelf naar behapbare stukken. Maar na een paar weken ploeteren leek ik er nog niet te zijn. De concepten van JADEX en de andere bronnen die ik hierbij door loop geven me nog enige grip op het geheel. Dit is niet iets dat je even doet. Mensen komen hier na jaren lang werken tot een redelijke oplossing voor hun probleem, geen perfecte.

BDI Software Model

Uiteindelijk heb ik duidelijk kunnen krijgen welke onderdelen van het BDI Model nou de belangrijkste dingen zijn die ik nodig ga hebben bij de ontwikkeling. Die zijn in dit stuk van het onderzoek beschreven.

Dit is een voortgang op het BDI Software Model (in short) dat gesteld werd in 4.2 BDI.

Architecture

Bij de vertaling van de concepten naar een model veranderen de Desires en Intentions enigszins. Beliefs zijn echter nog steeds het informatieve component van de Agent [12] .

Desires en Intentions worden vertaald naar **Goals**. En de nodige **Plans** om de bijbehorende activiteiten van je Intentions actief na te kunnen volgen.

Als laatste heb je ook nog de **Events**, deze kunnen zorgen voor veranderingen van Beliefs, het triggeren van Plans, of het veranderen van de Goal state (actief/niet actief nastreven).

Ter opsomming:

Een BDI Agent bestaat nu dus uit de volgende stukken, met voorbeelden die je kan vergelijken met een echt persoon:

Beliefs: het informatieve component van de Agent. Bijv: "De oceaan is blauw".

Desires: de dingen die een Agent graag zou willen bereiken. Bijv: "Rijk worden".

Goals: een Desire dat de Agent actief na streeft. Bijv: "Om rijk te worden, ga ik werken"

Intentions: representeren de overwogen toestand van de Agent. Dit is wat de Agent gekozen heeft om te doen. In feite zijn dit de Desires die een Agent actief vervolgd. Bijv: "Om rijk te worden, ga ik werken bij een makelaarskantoor."

Plan: een sequentie van acties die de Agent uit voert, om één of meerdere Intentions te behalen.

Bijv: *Bij het makelaarskantoor komt er een klant binnen* "Ik zag een klant binnen komen, ik ga hem overtuigen een huis te kopen" → het overtuigen lukt "De klant tekent voor het huis" en uiteindelijk wordt de Agent uitbetaald en is hij een stap dichterbij zijn Intention om rijk te worden.

Events: deze kunnen zorgen voor veranderingen van Beliefs, het triggeren van Plans, of het veranderen van de Goal state (actief/niet actief nastreven).

BDI Execution Model

Vervolgens moet je verder gaan kijken hoe je van de architectuur een werkend geheel kan maken.

Om een idee te geven wat een Agent in feite zou moeten doen, hierbij een voorbeeldje in de vorm van pseudo-code:

```
while true
  observe the World;
  update internal world model;
  deliberate about what intention to achieve next;
  use means-ends reasoning to get a plan for the intention;
  execute the plan
end while;
```

Figuur 2 Simple Agent Control Loop

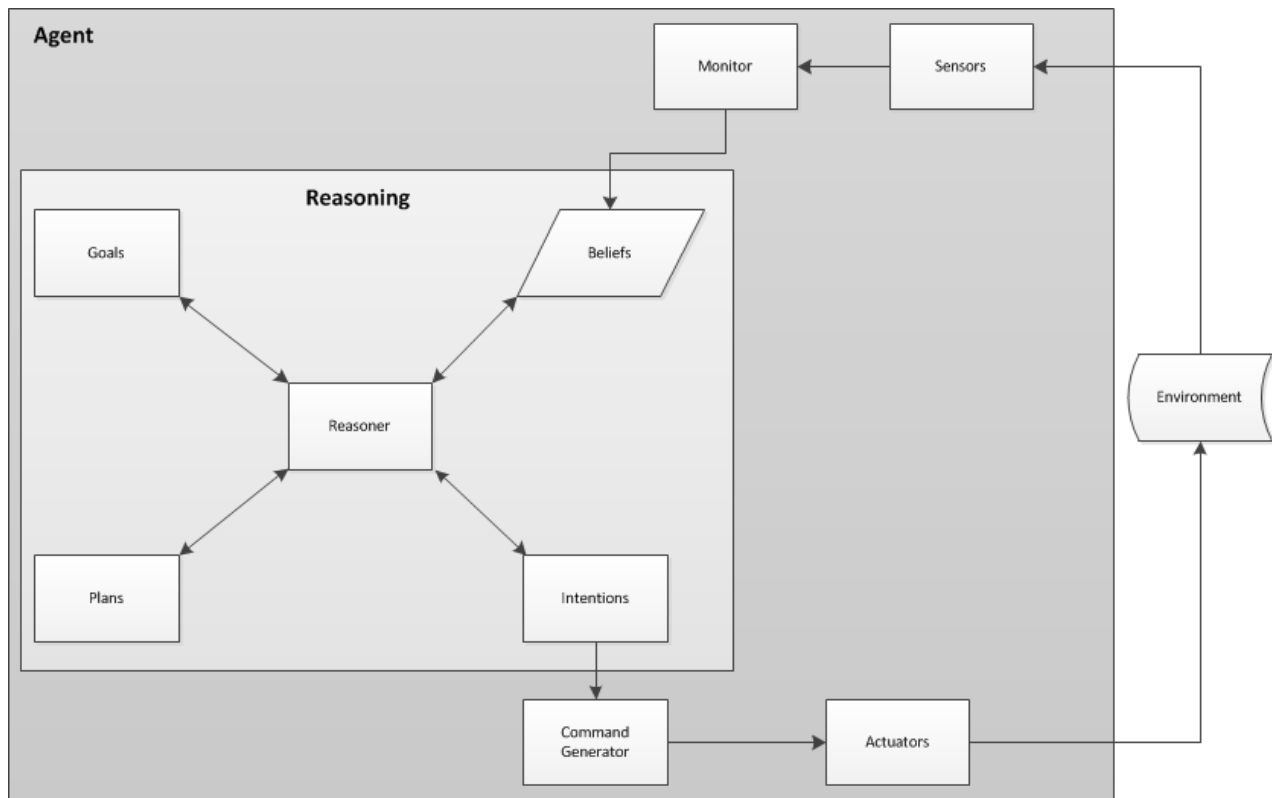
Dit vertaalt zich in een algemene BDI Execution Model, ook wel een Interpreter genoemd.

1. initialize-state
2. repeat
 3. options: option-generator(event-queue)
 4. selected-options: deliberate(options)
 5. update-intentions(selected-options)
 6. execute()
 7. get-new-external-events()
 8. drop-unsuccessful-attitudes()
 9. drop-impossible-attitudes()
10. end repeat

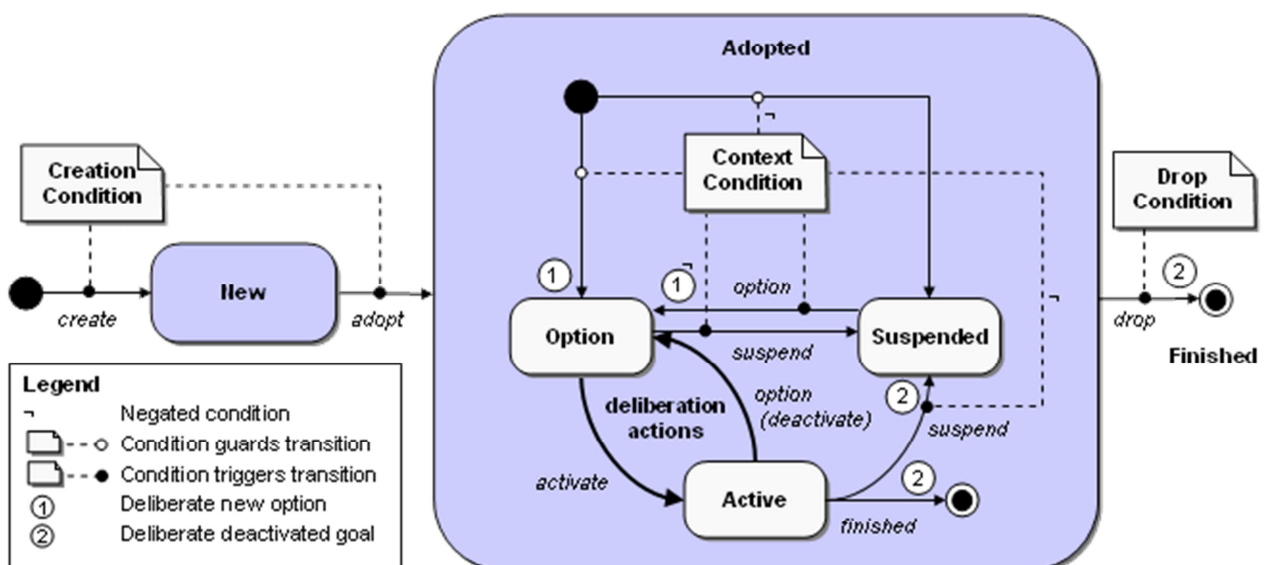
Tabel 2 BDI Interpreter (Execution model)

Dit is het basis Execution model dat bepalend is voor de **abstracte architectuur**, gebruikt in het PRS(Procedural Reasoning System) zoals te zien hieronder in Figuur 3 Procedural Reasoning System (PRS). Dit geeft een wat schematischer beeld van wat er net geschetst werd in Figuur 2 samen met het Execution Model.

Dit is een goede manier om de concepten systematisch aan te pakken en zal waarschijnlijk ook als de abstracte architectuur voor dit project gelden.



Figuur 3 Procedural Reasoning System (PRS)



Figuur 4 Goal Lifecycle (Easy Goal Deliberation)

Reasoning

Nu moet enkel de Reasoning hier nog in verwerkt worden!

Waar voornamelijk de reasoning op gebeurd zijn de Goals. Om het hier kort te houden zal ik verwijzen naar [22], voor een gedetailleerde uitleg over de Easy Deliberation Strategy die gekozen is als aanpak. Simpel gezegd zorgt dit systeem voor de transities van een Goal tussen Desire en Intention, zoals te zien hierboven in Figuur 4. Waar Desire staat voor een Goal dat nog niet in het Adopted stuk beland is en een Intention wel in het Adopted gedeeld valt.

Nu hebben we een manier om Goal deliberation aan te pakken, maar als je goed kijkt naar Figuur 3, is er nog een hoop informatie die de Reasoner binnen komt, van zowel Goals(en Intentions), Plans, als Beliefs(die ook van buitenaf veranderd kunnen worden door de Events).

Hier is eigenlijk een hele simpele aanpak voor, bedacht door de makers van JADEX, elke verandering die het systeem teweeg brengt, wordt door een EventManager aangepakt. De aanpak hiervan zal behandeld worden in het volgende hoofdstuk in de implementatie van het model.

6. Ontwikkeling - BDI Framework

In dit hoofdstuk zal de nodige uitleg gegeven worden op de resultaten van de ontwikkeling.

Uiteindelijk bestaat het **BDI Framework** uit de *BDI Engine*, de *externe libraries* (Simulation Engine Format en Signals) en een *test project* (dat ook als voorbeeld dient voor gebruik van het BDI Framework).

In de loop van de ontwikkeling is het Requirements document uit Bijlage III aangevuld met de ontwikkelingen van het framework in de vorm van Bijlage IV Technische Implementatie.

Het technische aspect zal ik hier niet volledig aanpakken, voor een gedetailleerde uitleg van de Technische Implementatie, is te kijken in Bijlage IV. Daarbij zit ook het volledige klassendiagram van het huidige Framework.

6.1. BDI Model implementatie

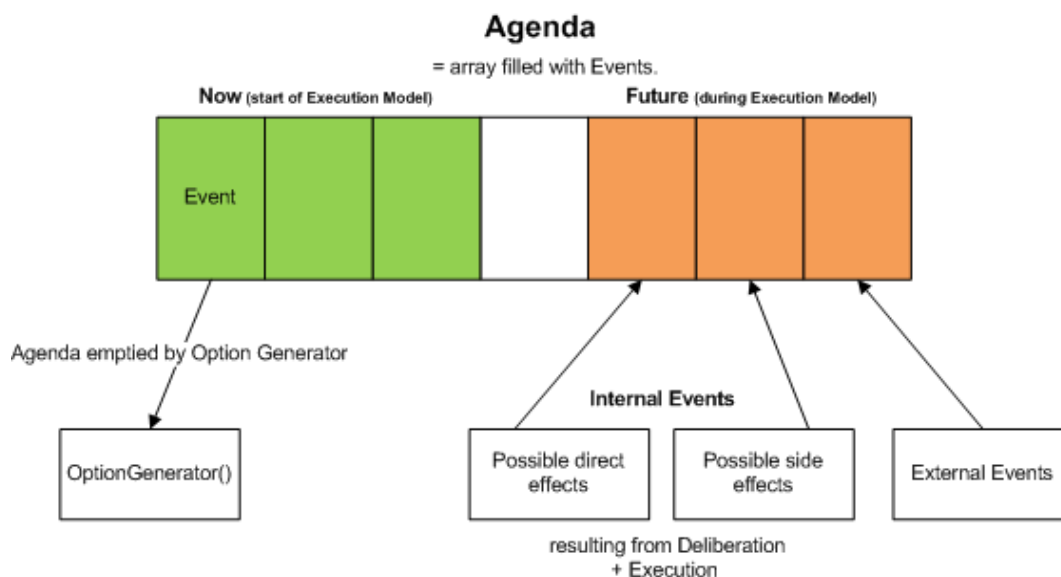
De daadwerkelijke implementatie van het BDI Model in de Engine zal hier behandeld worden

Al de concepten gebruikt in dit model zijn gebaseerd op de concepten van JADEX [13] en de standaard concepten voor dit model te vinden op [23].

De interpretatie die ontstaan is tijdens het onderzoek is wat afwijkend van JADEX zelf, maar maakt veel gebruik van zowel de ideeën/modellen van JADEX als de algemene concepten over BDI [24].

6.1.1. Execution Model

Omdat het execution model een vitaal onderdeel is van het gehele systeem, wordt dat hierbij nog iets verder uitgelicht, te beginnende met de Agenda:



Figuur 5 Agenda

Agenda

De Agenda is een belangrijk onderdeel van het Execution Model. Dit is waar de mogelijke opties voor een Agent in worden opgeslagen en uitgehaald om zijn komende doel(Goal) en actie (Plan) te kunnen bepalen door hier over te beraadslagen(Deliberate).

Execution Model(Interpreter)

1) Agent init

Simpelweg de initiation van de Agent en zijn Reasoning System. Wordt gedaan samen met het initiëren van het SimulationModel.

De Agent met Reasoner(met Agenda/ GoalDeliberator/ MeansEndReasoner) + EventManager + Belief-/ Goal-/ Planbase worden initialized.

Alle Signals(Events) die door de Goals/ Plans gecreëerd zijn, worden direct behandeld door de EventManager. Belief Signals(Events) hebben meestal *listeners* in de vorm van Goals/ Plans, die elk Signal zelf weer af handelen.

De Reasoner bevat in feite de rest van het Execution model:

Begin loop.

2) Option generator

Deze haalt alle mogelijke opties uit de Agenda.

Over het algemeen zullen dit Signals(Events) zijn die met Plans/Goals te maken hebben.

3) Deliberation

De nieuwe staat van Goals/Plans worden overwogen op basis van de gegeven Signals(Events). Alles wat betrekking heeft tot deze Signals wordt bekeken, andere dingen niet (Geen Goal/Plan Signals == geen deliberation).

4) Update intentions

Uit de Deliberation is een keuze van Goals/Plans gekomen die doorgevoerd worden.

5) Execute

Plan steps worden uitgevoerd.

Hieruit kunnen weer veranderingen in de Agent plaatsvinden, die weer Signals(Events) veroorzaken.

6) Drop succesful-attitudes

Goals/plans die klaar zijn met hun taak, worden geëvalueerd om te zien of ze dropped moeten worden. De verandering veroorzaakt een Signal(Event).

7) Drop impossible-attitudes

Goals/ Plans die niet(/onmogelijk) hun doel kunnen bereiken / aan hun dropcondities voldoen, worden gedropped. Deze verandering veroorzaakt ook een Signal(Event).

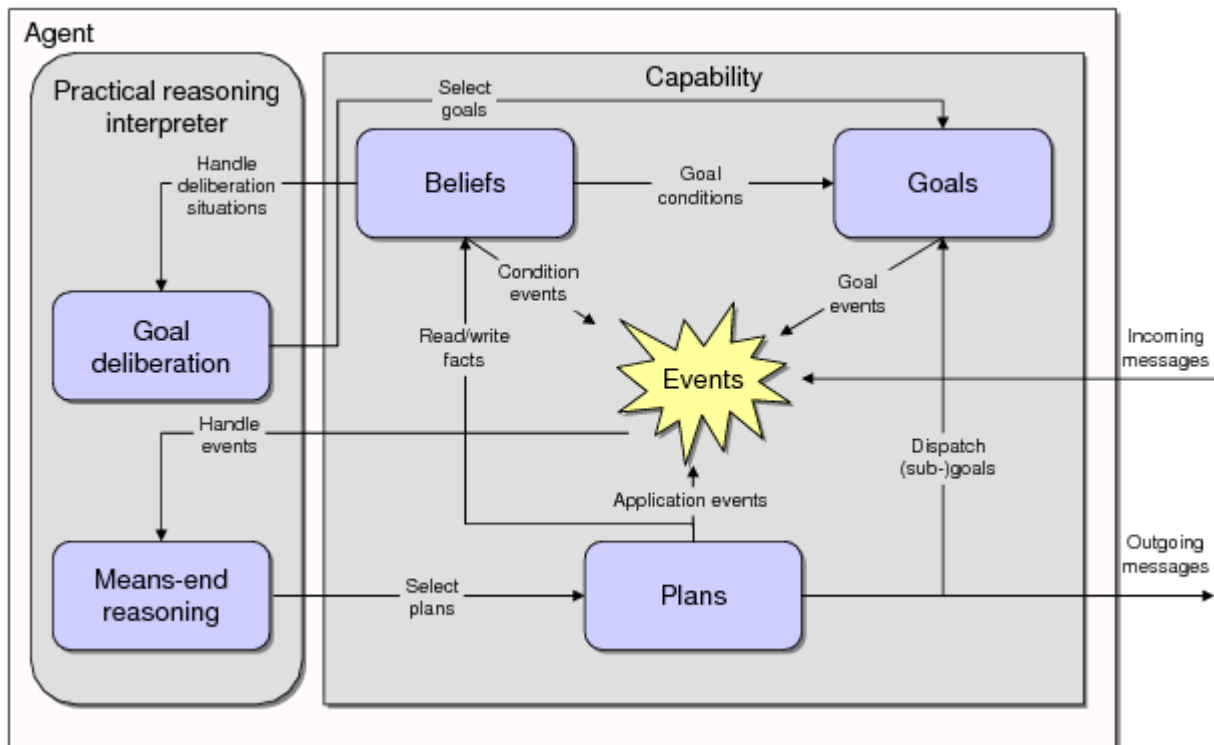
Einde loop.

6.2. Reasoning System implementatie

Om ook het Reasoning Systeem iets verder toe te lichten, hierbij de Reasoner die er voor zorgt dat de Agent na denkt.

Het grootste deel van het Execution Model wordt hierin uitgevoerd (de loop: 2 t/m 7).

Hier is eigenlijk een hele simpele aanpak voor, elke verandering die het systeem teweeg brengt, wordt door een EventManager aangepakt. Zo wordt het PRS, gesteld in Figuur 3 Procedural Reasoning System (PRS) (pagina 28), aangevuld met een Event handling systeem, waardoor het algemene PRS meer begint te lijken op de architectuur van JADEX te zien hieronder in Figuur 6.

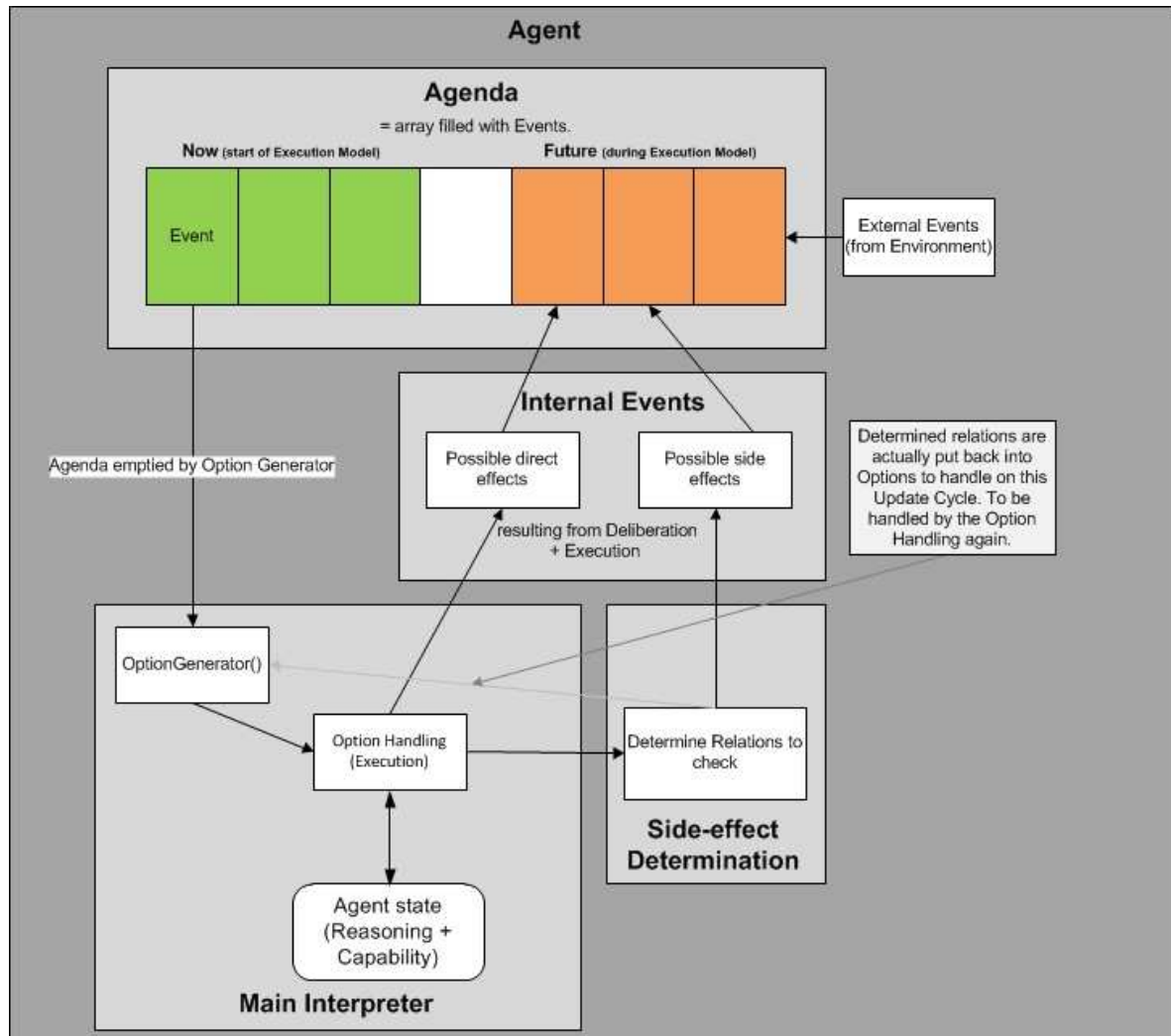


Figuur 6 JADEX abstract architecture

Hier komt echter het issue terug dat AS3 heeft t.o.v. JAVA: het is Single-threaded i.p.v. Multi-threaded. De JADEX architectuur in Figuur 6 is daarom niet in gebruik genomen.

Maar zo is er wel een structuur ontwikkeld die voornamelijk komt uit de Easy Goal Deliberation Strategy. Deze maakte ook gebruik van een Agenda om alle Events in weg te zetten tijdens stap 2 t/m7 van het Execution Model in 6.1.1. Aan het begin van de loop in dit execution model, kunnen dan weer de Events behandeld worden, die tijdens de stappen 2 t/m7 ontstaan zijn, zonder dat elke verandering weer direct effect zal hebben op de staat van de Agent.

Zoals je hieronder in Figuur 7 kan zien, is dit de uiteindelijke aanpak geweest hiervoor, waar het Reasoning gedeelte van de Agent verder nog hetzelfde blijft als in Figuur 3 Procedural Reasoning System (PRS).



Figuur 7 Reasoner(Interpreter) abstract Architecture

6.2.1. Goal-deliberation

Binnen Goal-deliberation bepalen we welke goals actief nagestreefd worden en welke goals (voor nu) *niet* nagestreefd gaan worden.

Hiervoor is de Easy-Goal-Deliberation strategie [22] gebruikt, de lifecycle van een goal is te zien hierboven in Figuur 4 Goal Lifecycle (Easy Goal Deliberation)_op pagina 28.

De status van deze goals wordt bepaald door een aantal condities.

CreateConditions: moeten aan voldaan worden om creatie en adoptie van van dit goal toe te staan.

ContextConditions: bewaken de transitie tussen de 3 geadopteerde goal stati: *Option*, *Active* en *Suspended*. Als aan ContextConditions voldaan wordt, betekent dat een goal in de *Option* state kan zijn en de *Active* state toelaatbaar is (toelaatbaar door de deliberation), het niet voldoen aan de ContextConditions betekent dat het goal *Suspended* wordt. De *ContextConditions* worden beschikbaar na adoptie van het goal.

DropConditions: worden aan deze voldaan, zal de Goal *dropped* worden, waarmee het goal *Finished* wordt.

6.2.2. Means-end Reasoning

Hier gebeurt eigenlijk hetzelfde als bij de Goal-deliberation, maar dan voor plannen.

“Welke acties gaan actief ondernomen worden?” is dus het uitgangspunt van de Means-end Reasoner.

Hierbij is voor de Plans ook simpelweg dezelfde aanpak gekozen als bij de Goals, met enkele uitzonderingen:

- De **CreateCondition** wordt vervangen door een **PreCondition**, aangezien er meerdere instanties van een Plan aanwezig kunnen zijn.
- In deze **PreConditions** kunnen Goals zitten: Zo ja horen deze plans bij dit Goal en kunnen ze enkel getriggered worden als dit Goal actief is.
Zitten er géén Goals in de PreConditions: dan worden deze in de Means-end Reasoning meegenomen als een Plan dat potentieel uitgevoerd moet worden.

6.3. External libraries

6.3.1. Simulation Framework (SimulationModel + Controller.)

Het Simulation Model is verwerkt in de BDI Engine, zoals gemeld in 5.5, is dit een belangrijk onderdeel voor toekomstige tooling. Het Simulation Model is overigens een Framework van RANJ zelf.

In het BDI Framework wordt een Simulation Model gebruikt als een laagje bovenop de Agent zelf, het Simulation Model houdt alle entity's binnen de Agent bij (inclusief de Agent zelf).

Zo kan er gezorgd worden dat elke entity zijn unieke entityID heeft, zodat 1) entiteiten elkaar kunnen vinden, 2) er niet per ongeluk duplicaten ontstaan en 3) is het gemakkelijk hanteerbaar, omdat RANJ hier al een editor voor heeft!

Daarnaast geldt dit ook als een centraal punt voor initialisatie en updaten van de Agent. Door middel van de SimulationController wordt de update aangestuurd met een timer.

6.3.2. Signals Framework

Signals is een Framework gemaakt door Robert Penner [25].

Signals is in feite een variant op het eventsysteem van AS3 zelf. De voornaamste rede om dit in gebruikt te nemen voor het BDI Framework was het feit dat er een lijst met listeners direct

beschikbaar is, wat het stukken gemakkelijker maakt om listeners te managen. Maar het ook de optie open liet om nog van de functionaliteiten van het eventsysteem van AS3 zelf gebruik te maken.

6.4. Tests

Er zijn hele simple test cases gemaakt om bepaalde functionaliteiten te kunnen testen tijdens het ontwikkelen. Om hier dieper op in te gaan in de scriptie is niet van belang.

Final Test Project

Om uiteindelijk het gehele project goed te kunnen testen op de requirements en het representabel te maken naar RANJ toe heb ik nog een testproject ontwikkeld die een vervolg op GGR voorstelt. De case zelf staat er, op het moment van deze scriptie schrijven, maar het bijbehorende project zal nog ontwikkeld worden in de laatste weken van de stage.

Hiermee kunnen de programmeurs van Ranj ook goed zien hoe er met het framework gewerkt kan worden.

In 4.3 GGR, op pagina 15 viel te zien wat GGR precies in houdt, dit speelde zich af in het kantoor van de makelaar. De volgende logische stap hierop (als een klant het huis interessant genoeg vond) is dat de klant samen met de makelaar het huis gaat bezichtigen.

In feite blijven de principes van de game hetzelfde, de setting verandert simpelweg. I.p.v. voorbeelden te geven over het huis, kan de klant ze nu daadwerkelijk zien. Voordat de klant (de agent) in het huis komt krijgt de speler nog even de tijd om in het huis rond te kijken en bijvoorbeeld wat vlekken op de houten vloer te verstoppen onder een stoel. Zo kan de speler fysiek dingen verstoppen voor de klant of ze verbaal beter gaan verwoorden dan ze zijn, in de hoop dat de klant (de agent) je woorden aan neemt.

6.5. Ontology

Ontology representeert kennis, als een set van concepten binnen een bepaald domein en de relaties tussen deze concepten [26].

De Ontology zal één van de eerste uitbreidingen op het BDI Framework zijn, omdat dit staat voor de kennis die een BDI Agent zal hebben en wat voor eigenschappen de Agent zal hebben het domein zal dan afhankelijk zijn van de game. Maar er zal hiervoor wel een basis opgezet moeten worden om dit handig te kunnen managen.

Zo kan bijvoorbeeld je de persoonlijke eigenschappen (psychologische profielen) van een Agent die in een Ontology gemaakt worden bewaren voor meerdere games. En kan de domeinkennis aangepast worden naar het domein.

De eerdere kennis die is opgedaan door RANJ en TNO m.b.t. Ontology is bij RANJ bekend [8] (deze thesis is overigens niet beschikbaar buiten RANJ/TNO).

In het onderzoek dat door RANJ en TNO gedaan is, is gebruik gemaakt van Protégé [27]. Een open-source tool voor Ontology. Deze tool kan in feite Ontology's representeren, bestaande uit classes, instances, properties en de karakteristieken van deze properties.

Het enige nadeel is dat dat in Java is gemaakt en ook bedoeld is om geëxporteerd te worden in Java code in een door JADE [16] geaccepteerd format. De ondersteuning van JADE maakt het een product

met een sterke ondergrond, omdat JADE voldoet aan de eisen van FIPA [17], een organisatie die standaarden voor Agent-Based technologie ontwikkelt.

Maar dit is helaas dus niet fatsoenlijk bruikbaar in AS3. Het schrijven van een plugin voor Protégé is helaas ook niet aan te raden, want zelfs al zou je het kunnen exporteren naar een AS3 format, mis je de nodige ondersteuning van het JADE framework.

Een logische vervolgstap voor RANJ op het BDI Framework is om een eigen Ontology Editor te bouwen. De kennis in Ontology die bij RANJ is opgedaan door Ivo Swartjes / TNO is hierbij een pré.

7. Conclusie(s) en aanbeveling(en)

Dit onderzoek werd ingegaan, met de hoofdvraag *“Hoe ga ik een BDI(beliefs, desires, intentions) framework in AS3 maken waar RANJ baat bij heeft, gebaseerd op de principes van JADEX.”*, met geen zekerheid of dit haalbaar zou zijn binnen de stageperiode. Nu na het onderzoeken en ontwikkelen van het Model en het Framework bleek dat toch realiseerbaar.

De requirement van de **“minimaal nodige onderdelen om de BDI Engine draaiende te krijgen”** uit Hoofdstuk 5.4 was het primaire onderdeel dat voltooid moest worden voor de hoofdvraag, dat lijkt in ieder geval gelukt. De andere requirement die keek naar de eisen van Ranj bij JADEX zijn hier redelijk gemakkelijk aan toe te voegen mocht dit nodig zijn. We hebben voor nu in ieder geval vast kunnen stellen hiermee welke onderdelen van BDI van belang zijn voor Ranj.

Ook is er rekening gehouden met de toekomst van dit framework door het Simulation Model te implementeren, hiermee kan het BDI Framework gemakkelijk toegankelijk gemaakt worden voor de Game Designers. Het is waarschijnlijk nog wel zo dat er een Ontology editor/extensie hierbij gemaakt moet worden, om de abstracte BDI begrippen te veranderen in iets meer begrijpelijks.

Toekomst

Ivo's kennis m.b.t. Ontology zou wel een must zijn om nut te halen uit het BDI Framework. Ik zou adviseren een Ontology editor/extensie te maken die aan sluit op de Simulation Editor / binnen de Simulation Editor past. Hierna zou je namelijk het vervolgonderzoek kunnen voortzetten door BDI met Ontology in een game te implementeren.

Er zit ook potentie in het gebruik van dit BDI Framework in de toekomst, voor spellen bij Ranj als *“Changing the Dialog”*. In het vervolgonderzoek zal kunnen blijken wat de effecten hier van zijn.

Evaluatie

Ik vond het erg leuk dat ik bij Ranj heb mogen werken, de mensen zijn daar heel gezellig, zelfs al praat ik zelf af en toe wat weinig (Dat komt omdat iedereen zo veel interessante dingen verteld!). Ik baal wel dat ik fysiek niet altijd gezond was en voelde me alsof ik me daardoor niet kon bewijzen, maar naar mate ik meer het gevoel kreeg dat ik lekker aan het werk was ging alles een stuk beter.

Mijn voornaamste leerpunten waren: eens dieper in de code duiken en ook aan performance improvement van de code doen. Met name in de code duiken is goed gelukt tijdens de stage, performance improvement iets minder, maar dat was simpelweg niet zo toepasselijk op de opdracht in het huidige stadium.

Hierbij heb ik wel enorm veel over onderzoeken geleerd binnen een werkveld dat (mogelijk) geen duidelijk uitzicht heeft vanaf het begin. Een duidelijk eind is iets waar we op school vaak wel voor proberen te plannen, omdat opdrachten over het algemeen al een vast einddoel hebben. Dit gaf me een hele andere kijk op onderzoeken in het algemeen. Gelukkig is mijn begeleider Ivo goed bekend met het schrijven van scripties en onderzoek doen, ik heb enorm veel kunnen leren van hem in een korte tijd.

Wat heb ik me hier in een overvolle zee aan informatie gegooid... na een week voor mijn stage nog een game te hebben gemaakt voor Laparoscopische Cholecystectomy binnen een weekje, waarvan het domein prima behapbaar was, dacht ik dit ook wel even in mijn hoofd te kunnen krijgen. Boy was I wrong! Op tijd een stapje terug weten te nemen is hierbij vitaal, maar na overmoed door de week voor mijn stage, maakte ik deze stap net wat laat.

Desondanks mijn misstap heb ik in deze fase een goed resultaat neer kunnen zetten. En had ik al een groot deel van het onderzoek gestart.

De ontwikkeling en het onderzoeken van de mogelijkheden liep voor een groot deel samen, dat was gewoon de aard van dit project. Toch voelde het onwennig, simpelweg omdat het nieuw voor me was. Uiteindelijk ben ik toch weer enorm blij om iets nieuws aan te hebben gepakt.

Na het aanpakken van een pittig onderzoek leek het schrijven van de scriptie niet meer zo moeilijk. Toch bleek wel dat ik dit absoluut niet had mogen onderschatten. Mijn eerste paar dagen heb ik puur besteed aan een opzet maken van hoofdstukken en nadenken wat ik er in zou kunnen schrijven. Op een gegeven moment ben ik gelukkig gewoon maar begonnen met schrijven. Uiteindelijk heb ik verschillende aanpakken en uitgangspunten gebruikt om deze scriptie te schrijven. Waar ik begon met het idee dat ik haast een "research paper" moest opstellen, ben ik geëindigd met een combinatie van datgeen en een procesverslag.

Een groep van de stagiaires(waaronder ik) is aangeboden om aanpak te verzinnen voor een opdracht die RANJ ons graag uit zou zien werken tijdens een zomervakantie-baan. Ik zie er naar uit om dat daadwerkelijk uit te kunnen voeren, als de opdracht aangenomen wordt natuurlijk.

Of ik daarna bij RANJ kan en wil blijven is voor mij zelf ook nog een vraag. Ik heb ook de ambitie om zelf iets te ondernemen in de Game industrie, het is echter niet gemakkelijk om zomaar succes te boeken in deze industrie.

Één ding is zeker, ik wil enorm graag in de Game industrie aan de slag gaan!

Bibliografie

- [1] Wikipedia, „Wikipedia on "Intelligent agent",” 2012. [Online]. Available: http://en.wikipedia.org/wiki/Intelligent_agent. [Geopend 2012].
- [2] „Wikipedia on "Multi-agent system",” mei 2012. [Online]. Available: http://en.wikipedia.org/wiki/Multi-agent_system. [Geopend mei 2012].
- [3] N. I. o. S. a. Technology(NIST), „finite state machine,” 12 mei 2008. [Online]. Available: <http://xlinux.nist.gov/dads//HTML/finiteStateMachine.html>. [Geopend mei 2012].
- [4] „Ranj Serious Games Website,” [Online]. Available: <http://www.Ranj.com/>. [Geopend Mei 2012].
- [5] „Purno de Purno,” 2007. [Online]. Available: http://www.purnodepurno.nl/index.php?option=com_content&view=article&id=4&Itemid=5. [Geopend 2012].
- [6] Ranj, „Juf in a Box,” Ranj, 2012. [Online]. Available: <http://www.Ranj.com/nl/content/werk/juf-in-a-box>. [Geopend Mei 2012].
- [7] J. Foley, Regisseur, *Glengarry Glen Ross*. [Film]. 1992.
- [8] A. Brandenburgh, *Influence of Personality on the Behavior of Conversational Agents*, Not yet Final red., 2012.
- [9] P. Verschuren en H. Doorewaard, *Het ontwerpen van een onderzoek.*, vol. 4e druk, Nijmegen: Boom Lemma uitgevers, 2007, august.
- [10] H. Kniberg, *Scrum and XP from the Trenches - How we do Scrum*, C4Media Inc, 2007.
- [11] P. Kempen en J. Keizer, *Competent afstuderen en stagelopen - Een advieskundige benadering*, vol. 4e druk, Groningen/Houten: Noordhoff Uitgevers, 2010, oktober.
- [12] „Wikipedia on BDI Software Model,” 18 april 2012. [Online]. Available: http://en.wikipedia.org/wiki/Belief%E2%80%93desire%E2%80%93intention_software_model. [Geopend 4 juni 2012].
- [13] A. S. Rao en M. P. Georgeff, „BDI Agents From Theory to Practice,” 1995.
- [14] L. Braubach en A. Pokahr, „Jadex BDI Agent System - BDI User Guide,” University of Hamburg, 2002-2012. [Online]. Available: <http://jadex-agents.informatik.uni-hamburg.de/xwiki/bin/view/BDI+User+Guide/01+Introduction>. [Geopend 2012].
- [15] R. Dromey, *Formalizing the transition from requirements to design*, Nathan, Brisbane,

AUSTRALIA, 2006.

- [16] A. Bartish en C. Thevathayan, BDI agents for game development, New York, NY, USA: ACM, 2002.
- [17] „Java Agent Development Framework,” Telecom Italia Lab, 18 nov 2011. [Online]. Available: <http://jade.tilab.com/>. [Geopend mei 2012].
- [18] „Foundation for Intelligent Physical Agents,” IEEE Foundation for Intelligent Physical Agents, 2012. [Online]. Available: <http://www.fipa.org/>. [Geopend mei 2012].
- [19] S. Jacobs, Game Programming Gems 7, Boston: Course Technology PTR, 2008.
- [20] A. Lake, Game Programming Gems 8, Boston: Course Technology, 2011.
- [21] S. Rabin, AI Game Programming Wisdom 4, Boston: Charles River Media, 2008.
- [22] L. Braubach en A. Pokahr, „JADEX BDI User Guide - Chapter2. Concepts of the Jadex BDI Reasoning Engine,” 2002-2012. [Online]. Available: <http://jadex-agents.informatik.uni-hamburg.de/xwiki/bin/view/BDI+User+Guide/02+Concepts>. [Geopend Mei 2012].
- [23] A. Pokahr, L. Braubach en W. Lamersdorf, A Goal Deliberation Strategy for BDI Agent Systems, Berlin Heidelberg New York: Springer-Verlag, 2005.
- [24] M. E. Bratman, Intention, Plans, And Practical Reason, 1999 red., Cambridge, Massachusetts: Harvard University, 1987.
- [25] R. Penner, „AS3 Signals on Github,” 2011. [Online]. Available: <https://github.com/robertpenner/as3-signals>. [Geopend april 2012].
- [26] „Wikipedia on Ontology in Information Science,” 29 apr 2012. [Online]. Available: [http://en.wikipedia.org/wiki/Ontology_\(information_science\)](http://en.wikipedia.org/wiki/Ontology_(information_science)). [Geopend jun 2012].
- [27] „Protégé,” Stanford Medical Informatics, 2012. [Online]. Available: <http://protege.stanford.edu/>. [Geopend 24 mei 2012].

Lijst van figuren

Figuur 1 Screenshot GGR.....	15
Figuur 2 Simple Agent Control Loop.....	26
Figuur 3 Procedural Reasoning System (PRS).....	28
Figuur 4 Goal Lifecycle (Easy Goal Deliberation).....	28
Figuur 5 Agenda.....	30
Figuur 6 JADEX abstract architecture	32
Figuur 7 Reasoner(Interpreter) abstract Architecture	33

Lijst van tabellen

Tabel 1 JADEX BDI Usage in GGR (short table).....	22
Tabel 2 BDI Interpreter (Execution model)	27

Bijlagen

I. Project Initiation Document	43
II. Blog.....	60
III. Requirements	66
IV. Technische Documentatie.....	69

De bijlagen zullen ook digitaal beschikbaar zijn, vooral voor bijlage IV Technische Documentatie is het praktischer om deze digitaal te bekijken.

I. Project Initiation Document

BDI IN GAMES

Project Initiation Document



Auteur:

Andrax vd Bogaard

Status:

v1.1

Datum voltooid:

13-3-2012

Documenthistorie

Revisies

Versie	Status	Datum	Wijzigingen
0.1	Concept	19-2-2012	Voornamelijk een opzet gemaakt in Achtergrond en Projectdefinitie + Initiële Business Case.
0.2	Concept	27-2-2012	Projectdefinitie en Initiële Business Case aangepast aan de hand van gesprek met Opdrachtgever.
0.3	Concept voltooiing	1-3-2012	Invulling geven aan alle onderdelen (idealiter een v1.0 zonder goedkeuring)
0.4	Concept	5-3-2012	Projectdefinitie improvement. Naar aanleiding van review door Opdrachtgever(Ivo).
1.0	Final	13-3-2012	Na gesprek met docentbegeleider voor het grootste deel al goedgekeurd. Maar TSP er nog in verwerkt bij punt 3.2 + planning iets aangepast om dit er ook duidelijk in te maken. + Managementsamenvatting

Goedkeuring

Dit document behoeft de volgende goedkeuringen:

Versie	Datum goedkeuring	Naam	Functie	Paraaf
0.1				

Distributie

Dit document is verstuurd aan:

Versie	Datum verzending	Naam	Functie
0.3	1-3-2012	Ivo Swartjes	Bedrijfsbegeleider
0.4	5-3-2012	Jan Oonk	Docentbegeleider
1.0	13-3-2012	Jan Oonk	Docentbegeleider
1.0	13-3-2012	Ivo Swartjes	Bedrijfsbegeleider

Managementsamenvatting

Doel van dit document

De projectscope in beeld te krijgen en de basisinformatie en uitgangspunten van het project vast te leggen. Het project moet hiermee gedefinieerd kunnen worden door het projectmanagement.

Daarbij wordt een richtlijn gegeven aan het onderzoek, wordt het beoogde en realistisch haalbare resultaat vastgesteld en ook de mogelijke vervolgstappen.

Dit dient ook als basisplanning voor het gehele project, praktijk kan hier natuurlijk enigszins van afwijken.

Aanleiding

Momenteel onderzoekt Ranj de mogelijkheden en onmogelijkheden van het gebruik van AI voor het voeren van dialogen met NPC's. In het bijzonder gebruikt Ranj hiervoor het BDI (beliefs, desires, intentions) modelleerparadigma.

In samenwerking met TNO is er het afgelopen jaar een onderzoeksproject geweest naar BDI, door gebruik te maken van JADEX (een implementatie van BDI). JADEX is echter een enorm breed en ingewikkeld platform, een groot deel van de functionaliteit is in dat onderzoeksproject nooit gebruikt.

Globale aanpak

Scrum om binnen sprints kort en krachtig onderdelen van het project aan te pakken. TSP wordt hiernaast gebruikt om een algemene richtlijn te geven aan het project.

De doelstelling is om een BDI Framework in AS3 te maken, waar Ranj baat bij heeft. Dit zal deels uit het onderzoek blijken en deels uit het te ontwikkelen framework. JADEX wordt hierbij als basis gebruikt om gemakkelijk te kunnen definiëren welke onderdelen van het JADEX framework ook van belang zijn bij dit project. Het onderzoek zal overigens voor een groot deel nog samen met het ontwikkelen van het framework gaan, omdat het een vrij ingewikkeld onderwerp is.

Globale kosten en doorlooptijd

De kosten van dit project zullen puur bestaan uit manuren van de student en studentbegeleider.

Het project zal voor 14-16 weken lopen.

Inhoudsopgave

1.	Inleiding	47
1.1.	Doel van dit document	47
2.	Achtergrond	48
3.	Projectdefinitie.....	49
3.1.	Projectdoelstellingen	49
3.2.	Gekozen oplossing of aanpak	50
3.3.	Scope van het project.....	52
3.4.	Producten c.q. eindresultaat	52
3.5.	Stakeholders	53
3.6.	Uitsluitingen.....	53
3.7.	Beperkingen	53
3.8.	Afhankelijkheden	53
3.9.	Aannames	53
4.	Projectorganisatiestructuur	54
4.1.	Opdrachtgever	54
4.2.	Seniorgebruiker.....	54
4.3.	Projectborging.....	54
4.4.	Projectmanager.....	55
4.5.	Projectsupport	55
Bijlage A:	Communicatieplan	56
Bijlage B:	Project planning	57
Bijlage C:	Initiële Business Case	58
Bijlage D:	Initieel Projectplan	59

1. Inleiding

1.1. Doel van dit document

De projectscope in beeld te krijgen en de basisinformatie en uitgangspunten van het project vast te leggen. Het project moet hiermee gedefinieerd kunnen worden door het projectmanagement.

Daarbij wordt een richtlijn gegeven aan het onderzoek, wordt het beoogde en realistisch haalbare resultaat vastgesteld en ook de mogelijke vervolgstappen.

Dit dient ook als basisplanning voor het gehele project, praktijk kan hier natuurlijk enigszins van afwijken.

Het PID behandelt de volgende fundamentele aspecten van het project:

- Wat beoogt men met het project te bereiken?
- Waarom is het belangrijk om deze doelstellingen te bereiken?
- Wie zijn er betrokken bij het managen van het project en wat zijn hun rollen en verantwoordelijkheden?
- Hoe en wanneer zullen de maatregelen die in dit PID besproken worden gerealiseerd worden?

Het document wordt gebruikt:

- Om er zeker van te zijn dat het project een gezonde basis heeft voordat de Stuurgroep gevraagd wordt het project te lanceren.
- Om te dienen als basisdocument op grond waarvan de stuurgroep en het projectmanagement de voortgang en wijzigingen kunnen toetsen en bewaken en vragen omtrent geldigheid van het project tijdens de uitvoering ervan kunnen beoordelen.

1.2. Opbouw van dit document

Om aan te geven welke onderdelen worden bijgewerkt en dus nieuwe versies zullen krijgen tijdens de voortgang van het project is dit AfstudeerScriptie verdeeld in twee secties: een statisch gedeelte en een dynamisch gedeelte:

Het statische deel bestaat uit de hoofdstukken en bijlagen:

- Achtergrond (Hoofdstuk 2)
- Projectdefinitie (Hoofdstuk 3)
- Projectorganisatiestructuur (Hoofdstuk 4)
- PID Bijlage A: Communicatieplan

Het dynamische deel bestaat uit de bijlagen:

- PID Bijlage B: Project planning
- PID Bijlage C: Initiële Business Case
- PID Bijlage D: Initieel Projectplan

2. Achtergrond

Ranj Serious Games is een wereldwijde autoriteit in de ontwikkeling van serious games. Sinds de start in 1999 hebben ze meer dan 400 games gemaakt. In de loop der jaren hebben ze een diep begrip opgebouwd voor het vertalen van leerdoelen naar pakkende gameplay. Het bedrijf wordt geleid door Michael Bas en bestaat uit meerdere producers, programmeurs, game designers, artists en freelancers, over het algemeen zo'n 20-30 man totaal.

Momenteel onderzoekt Ranj de mogelijkheden en onmogelijkheden van het gebruik van AI voor het voeren van dialogen met NPC's. In het bijzonder gebruikt Ranj hiervoor het BDI (beliefs, desires, intentions) modellerparadigma. In dit paradigma wordt wat een NPC zegt en doet niet bepaald door een script, maar door regels over wat hij weet, wat hij wil en hoe hij dit kan bereiken. Het framework dat Ranj hiervoor gebruikt is JADEX, wat voor een proof of concept prima is maar voor daadwerkelijk gebruik in de toekomst niet geschikt is omdat (1) het een mate van complexiteit bevat die veel verder gaat dan nodig en (2) het een Java framework is terwijl veel van de games bij Ranj in Flash zijn.

3. Projectdefinitie

3.1. Projectdoelstellingen

Het opzetten van een kernachtig en simpel BDI framework in AS3, gebaseerd op de principes van JADEX. *Welke aspecten van BDI/JADEX zijn hiervoor essentieel en welke zijn overbodig voor Ranj?*

In feite moet er dus een framework gebouwd worden dat NPC's(dialogen met NPC's) van beliefs, desires en intentions voorziet:

Ze moeten over komen als een echt persoon: Je zal haast nooit 2keer precies dezelfde dialoog doorlopen met de NPC ook al vraagt en antwoord de speler dezelfde dingen.

De **2 hoofd doelstellingen** van dit project zijn dus:

- **Onderzoeksvraag beantwoorden**
- **BDI framework opzetten**

De **onderzoeksvraag** luidt als volgt:

Hoofdvraag: Hoe ga ik een BDI(beliefs, desires, intentions) framework in AS3 maken waar Ranj baat bij heeft, gebaseerd op de principes van JADEX.

Deelvragen:

- Voor welke games bij het bedrijf is dit BDI framework van belang. (definitie van belangen van dit project voor het bedrijf)
- Dit BDI framework zal uiteindelijk door Game Designers goed/gemakkelijk bruikbaar moeten zijn, hoe kan hier voor gezorgd worden?
- Welke onderdelen van BDI (/het JADEX framework) zijn van belang voor het bedrijf.

De **opzet van het BDI framework** zal bestaan uit:

Een BDI framework in AS3 met een testproject waarin het volgende verwerkt zit:

- Simpele BDI architectuur voor één Agent:
 - o Beliefs
 - Beliefset
 - o Desires
 - Goals
 - o Intentions
 - Plans
 - o Events
- Uitbreidbaar in complexiteit
Bijvoorbeeld:
 - o Multiple Events synchronously
 - o Multiple Agents
 - o Communicatie tussen Agents
- Testproject heeft als doelstelling de functionaliteit en performance te testen. Waarbij functionaliteit een *must have* is en performance een *could have*.

Waarom moet dit project gerealiseerd worden?

Om een basis framework te creëren van BDI in AS3 en te zien of het realistisch is, dat dit framework commercieel inzetbaar gaat worden.

Welke mogelijkheden (capabilities, facilities) biedt het product of het projectresultaat?

En masse kan je NPC's genereren die afwijkende BDI's kunnen hebben. Zelfs al maak je dezelfde beliefset, goals en plannen aan voor een NPC, dialogen met deze NPC kunnen alsnog afwijkend zijn. Je keuzes in eerdere communicatie met deze NPC kunnen hier ook een invloed op hebben.

De NPC's zullen voornamelijk gebruikt worden voor dialogen, dus hier zal het BDI framework op gericht zijn.

De bedoeling is dat het framework (uiteindelijk) ook gemakkelijk bruikbaar wordt voor iemand die weinig tot geen verstand van BDI heeft. Er zal uiteindelijk ook tooling omheen gebouwd moeten kunnen worden die goed begrijpbaar is voor Game Designers.

3.2. Gekozen oplossing of aanpak

Onderzoeksstrategie: Bureauonderzoek.

Algemene en beknopte beschrijving van de aanpak:

Het project zal verdeeld worden in meerdere sprints van 2 a 3 weken, waarbinnen onderzoek wordt gedaan en de code wordt ontworpen / gerealiseerd.

Overview van de aanpak (niet specifiek in volgorde):

1. Definieer het gebruik en de benodigdheden van BDI voor Ranj (PID Vastleggen)
 - a. Omschrijving
 - b. Scenario's/simpel UML model van komende BDI systeem
 - c. Interviews met stakeholder(s)
2. Research (met name bestaande uit research naar BDI (gefocust op JADEX))
 - a. Research document + de nodige example projects (voor begrip van Agent en dialoog systemen)
 - b. Scenario's/simpel UML model van komende BDI systeem verder uitwerken
 - c. Simplified Framework base
 - d. Test Project
3. (optioneel) Jadex implementatie van BDI
 - a. Omschrijving
 - b. Flowchart op basis van een voorbeeld scenario
 - c. Test project
 - d. (optioneel) gegenereerd klassendiagram van het Jadex framework (of delen er van)
4. BDI Framework creëren
 - a. Verschillende niveaus van uitwerking definiëren (en deze in sprints indelen)
 - i. Benodigde documentatie erbij maken (requirements / issue tracking)
 - ii. Class Diagram
 - b. Demo's creëren op basis van Scenario's, om bruikbaarheid te testen en meten.

Scrum (sprints)

Met name omdat er voor korte perioden, redelijk duidelijke doelen te stellen zijn, is scrum naar mijn idee een goede manier om dit aan te pakken. Het project zal dus ook onderverdeeld worden in meerdere sprints, om zo iteraties op te kunnen bouwen van het project. De sprints zullen 2 a 3 weken duren.

TSP (Tien Stappen Plan)

Om invulling te kunnen geven aan de sprints, zal het TSP als een rode lijn door het hele project heen lopen. Hier onder een korte samenvatting van wat TSP inhoudt voor dit project:

I Oriënterende fase (in de weken van 13feb - ~ 12mrt)

1. Externe oriëntatie Oriëntatie op het bedrijf en evt. de afdeling. Hoofddoel: context en drijfveren/belangen van de opdrachtgever in kaart brengen.
2. Intake gesprek Gesprek met de opdrachtgever met als doel het preciseren van de behoefte van de opdrachtgever. Hoofdvraag: Wat zijn de wensen en verwachtingen van de opdrachtgever?
3. Oriënterende Activiteiten Inlezen/Bestuderen/Gesprekken om de bestaande situatie in kaart te brengen. Hoofddoel: voldoende info verzamelen om een eigen mening te kunnen vormen over de opdrachtspecificatie.
4. Analyse Opstellen van het Project Initiation Document met daarin de eigen interpretatie van de opdrachtspecificatie en een plan van aanpak met een globale planning.
5. Terugkoppeling/contractering Vergadering met bedrijfsbegeleider(s) en schoolbegeleider met als doel goedkeuren van PID /PvA en bijbehorende afspraken over facilitering van afstudeeropdracht.

II Ontwikkelings- en oplossingsfase (in de weken van 12mrt - ~14mei)

6. Werkplanning en projectorganisatie Gedetailleerd uitwerken van de organisatie en planning van het project uit het PID.
7. Ontwikkeling Uitvoering van het project (ontwikkeling en/of onderzoek). Zal voor veel opdrachten in ieder geval bestaan uit de gebruikelijke engineering activiteiten: specificatie, ontwerpen, realiseren, testen.
8. Oplossing(s)plan Oplevering van de voorgestelde oplossing/onderzoeksresultaat aan de opdrachtgever samen met de verantwoording van de oplossing. Kan gecombineerd worden met een presentatie binnen het bedrijf.

III Invoeringsfase (in de weken van 14mei – 18jun)

9. Overdracht/invoering Zorg voor behoud van het product of de opgedane kennis voor het bedrijf als de afstudeeropdracht afgelopen is. Indien van toepassing zorg voor een adequaat beheer van opgeleverde software.
10. Afronden en afstuderen Afronden van de opdracht bij het bedrijf, afronden van de scriptie en voorbereiden op de afstudeervoordracht op school.

Doelstellingen aanpakken

Hoofdvraag: *Hoe ga ik een BDI(beliefs, desires, intentions) framework in AS3 maken waar Ranj baat bij heeft, gebaseerd op de principes van JADEx.*

D.m.v. de deelvragen te beantwoorden.

Deelvragen:

- *Voor welke games bij het bedrijf is dit BDI framework van belang. (definitie van belangen van dit project voor het bedrijf)*

Dit zal grotendeels voortkomen uit dit PID, verder zal er hier meer duidelijk over worden zodra dit project afgerond is en het getest kan worden in draaiende projecten.

- *Dit BDI framework zal uiteindelijk door Game Designers goed/gemakkelijk bruikbaar moeten zijn, hoe kan hier voor gezorgd worden?*

Tijdens het ontwikkelproces van het framework zal hier rekening mee gehouden worden. Door te denken aan de volgende punten:

- Er zullen tools gemaakt worden op basis van dit framework, die vervolgens door Game Designers gebruikt gaan worden. De tools zullen niet binnen de scope van dit project vallen, maar het is wel handig om hier al vast rekening mee te houden. Het maken van de tools moet dus gemakkelijk vertaalbaar zijn van het framework naar de tools zelf. *In short:* Het framework moet gemakkelijk begrijpbaar zijn voor de makers van de tooling.
- Er kunnen templates gemaakt worden die gemakkelijk herbruikbaar zijn in toekomstige projecten. Denk dan aan templates voor: dialogen, Agents, taal templates. etc. Templates kunnen in de vorm van code of bijvoorbeeld XML.

- Welke onderdelen van BDI (/het JADEX framework) zijn van belang voor het bedrijf.

Dit zal deels onderzocht worden in het vooronderzoek. En deels terug komen bij de requirements vast te leggen. Door middel van communicatie met de opdrachtgever en senior gebruikers.

Om deze deelvragen te beantwoorden zal er dus voornamelijk onderzoek gedaan moeten worden naar BDI, waarbij je zou kunnen denken aan de volgende dingen:

- Wat houdt BDI in. (zal met name voorkomen in het vooronderzoek, met als doeleinde een "basic understanding" van BDI)
- Hoe zit JADEX in elkaar.

Om het **BDI framework op te zetten** zal rekening gehouden moeten worden met een aantal dingen, waaronder natuurlijk de onderzoeksvraag, maar ook de volgende dingen:

- Hoe zet je een modulair AS3 Framework op?
- Hoe zorg je voor een goede performance van je AS3 framework? (Wat zijn de do's en dont's van AS3 m.b.t. performance)
- Welke bestaande software heeft Ranj tot zijn beschikking, die het framework kunnen ondersteunen.

3.3. Scope van het project

Direct betrokken bij dit project zijn enkel de bedrijfsbegeleider en ik.

Overige betrokkenen kunnen vastgesteld worden puur om het onderzoeksgebied uit te breiden, maar zijn niet direct betrokken bij het project zelf. Overige betrokkenen kunnen enkel bestaan uit medewerkers van Ranj.

De doelgroep bestaat uit de Game Designers en Programmeurs van Ranj zelf.

3.4. Producten c.q. eindresultaat

Onderzoek producten

Het **PID** wordt in de eerste 3 weken van de stage vastgelegd, om een globale aanpak voor het onderzoek te bepalen.

Het **onderzoek document** zal van week 1 t/m 16 uitgewerkt en uitgebreid worden. Dit zal voornamelijk de volgende punten bevatten:

- AS3 Framework (Hoe bouw je een modulair framework en houdt je een ingewikkeld framework overzichtelijk voor andere programmeurs)
- BDI (Hoe zit BDI in elkaar en wat wil Ranj hiervan gebruiken)
- JADEX (ondersteunend om te bepalen wat Ranj van BDI wil gebruiken, mede omdat ze al bekend zijn met JADEX)

Framework

Naast het onderzoek document zal ook het daadwerkelijke **BDI Framework** gebouwd moeten worden in de periode van week 1 t/m 16.

Om het framework te verwezenlijken zullen de volgende onderdelen uitgewerkt worden:

- Requirements
- Class diagrams
- Het Framework zelf, bestaande uit:
 - o Code
 - o Documentatie (ASDoc zo goed mogelijk bij proberen te houden, zodat andere programmeurs gemakkelijker inzicht kunnen krijgen in de code)
 - o Test base (om gemakkelijk te kunnen testen of nieuwe onderdelen van het Framework werken)
 - Scenario's (om mee te testen)

In het Requirements document zullen de daadwerkelijke eisen aan het Framework beschreven worden, bestaande uit een harde eis voor een basis en mogelijke uitbreidingen.

3.5. Stakeholders

Ivo Swartjes (Ranj Serious Games), Opdrachtgever en voornaamste stakeholder. Ivo wil graag zien of het mogelijk is BDI te gaan gebruiken binnen toekomstige projecten. Hij heeft al voorkennis van BDI en werken met het JADEx framework. Dus kan met Ivo het beste overlegd worden wat haalbaar en realistisch is.

3.6. Uitsluitingen

TNO heeft niet te maken met het eindresultaat van dit project. Dit BDI project is wel voort gekomen uit een project dat Ranj in samenwerking met TNO heeft gedaan, maar dit project is bedoeld enkel voor gebruik door Ranj zelf.

3.7. Beperkingen

Beschikbaar Budget A	Beschikbaar Budget B	Beschikbare resources	Gewenste opleverdata	Beschikbare Doorlooptijd
560 – 640 uur	-	Ranj	14-juni-2012	14-16weken

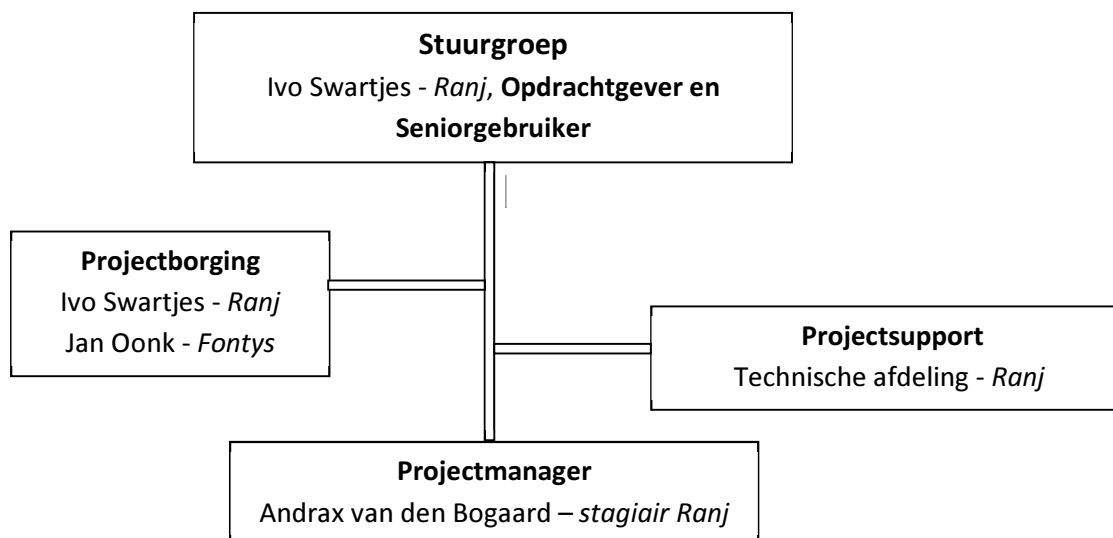
3.8. Afhankelijkheden

De enige afhankelijkheid is kennis van BDI en Agent systemen, bij de uitvoerder van het project. Zonder de nodige kennis zal het zo goed als onmogelijk zijn een BDI Framework neer te zetten.

3.9. Aannames

Een simpel BDI Framework moet te realiseren zijn. De mate van complexiteit die er in aangebracht kan worden zal bepalen of dit ook aan de projectdoelstellingen kan voldoen. Voor feedback en advies kan er, mits het geen hinder oplevert, altijd aangeklopt worden bij collega's. Bestaande libraries van Ranj zelf mogen gebruikt worden in dit project.

4. Projectorganisatiestructuur



4.1. Opdrachtgever

Rolbeschrijving

De opdrachtgever is de eigenaar van de opdracht en het project, hij is de voornaamste belanghebbende bij het eindproduct.

Project gerelateerde taken

Bij aanvang het project mee vastleggen. Controleert en geeft feedback op de voortgang van het project. De opdrachtgever zal het PID keuren, om het project van start te laten gaan.

Specifieke verantwoordelijkheden

In de loop van het project de specifieke details van het BDI Framework vastleggen. Daarbij samen met de Projectmanager de onderdelen bepalen en hun prioriteit vastleggen.

4.2. Senioregebruiker

Rolbeschrijving

De senioregebruiker, in dit geval de Technische afdeling van Ranj met als aanspreekpunt Ivo Swartjes, is degene die verantwoordelijk is voor de realisatie van de geplande baten.

Project gerelateerde taken

De technische afdeling zal zorgen voor functionele specificaties van het BDI Framework.

Specifieke verantwoordelijkheden

De senioregebruiker zal aan het begin van het project zorgen voor de algemene functionele specificaties. En ook in de loop van het project zal hij helpen de gedetailleerde specificaties van de software mee vast te leggen.

4.3. Projectborging

Rolbeschrijving

Projectborging zorgt voor controle op de status en kwaliteit van het project. Kwaliteit van werk wordt waar geborgen aan de hand van de projectdefinitie en requirements.

Project gerelateerde taken

Controleren van de status van het project en in de gaten houden of er geen afspraken of requirements worden overgeslagen. Daarmee de kwaliteit van het eindproduct te waarborgen.

4.4. Projectmanager

Rolbeschrijving

Zorgt voor de uitvoering van het gehele project. De projectmanager zal hierbij rekening houden met de eisen van de stuurgroep.

Daarbij zal de projectmanager zorgen voor de communicatie tussen alle betrokkenen.

Project gerelateerde taken

Dit bestaat uit een projectplanning en doelstelling opstellen, onderzoek uitvoeren en het project te realiseren.

Specifieke verantwoordelijkheden

De projectmanager is verantwoordelijk voor het bewaken van de planning en controleert op de kwaliteit van de op te leveren producten.

4.5. Projectsupport

Rolbeschrijving

De technische afdeling van Ranj kan hulp bieden aan de uitvoerende, indien nodig. Mocht de uitvoerende niet voldoende kennis beschikken kan hij hen inschakelen. Dit heeft voornamelijk betrekking tot coding quality/ diagrammen/ scenario's.

Project gerelateerde taken

Ondersteuning van de uitvoerende, wanneer deze daar om vraagt.

PID Bijlage A: Communicatieplan

Inleiding

Dit communicatieplan benoemt alle partijen die een (positief of negatief) belang hebben bij het project en de wijze waarop zij bij het project zullen worden betrokken en welke communicatievormen daarbij gebruikt worden. Het gaat hierbij om partijen en communicatie buiten de formele projectmanagementstructuur zoals beschreven in het PID.

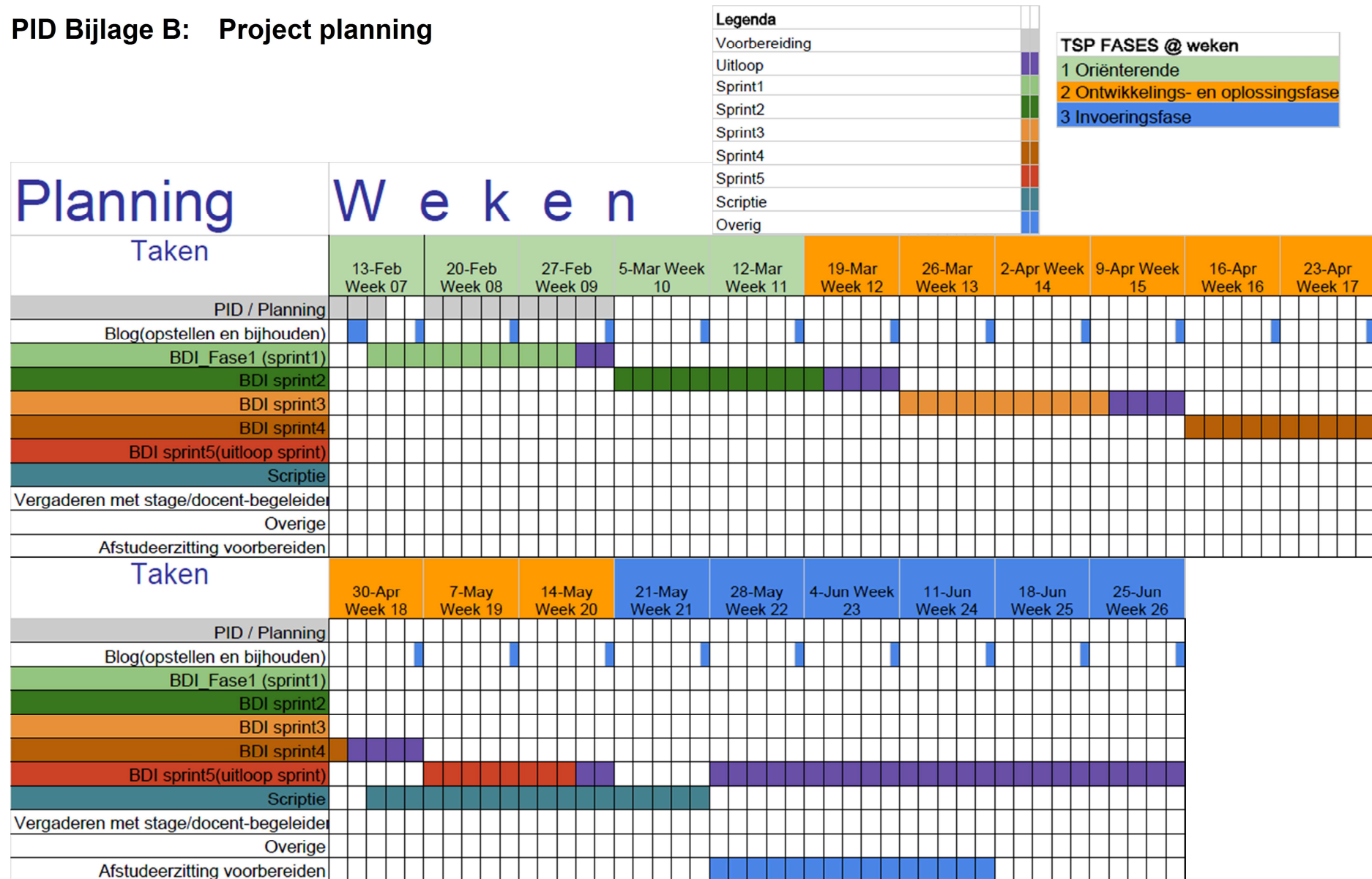
Belanghebbenden bij het project

Wie	Namens	Belang	Communicatievorm
<i>persoon</i>	<i>groep, afdeling</i>	<i>Welk belang bij project?</i>	<i>Hoe wordt deze persoon betrokken bij het project?</i>
Andrax van den Bogaard	Uitvoering, Programmeur	Onderzoek/Realisatie	Mondeling, chat en email
Ivo Swartjes	Opdrachtgever/ Programmeur/ Bedrijfsbegeleider	Advies, sturing, controle, begeleiding	Mondeling, chat en email
Jan Oonk	Stagebegeleider	Begeleiding, controle	Email, stage bezoeken
Collegae (voornamelijk Technische afdeling & Game Designers)	Ranj	Mogelijk toekomstige gebruiker	Mondeling, chat en email

Communicatiekanalen

Van	Naar	Informatie	Medium	Frequentie of data
<i>Persoon of groep</i>	<i>Persoon of groep</i>	<i>Soort informatie</i>	<i>Email, telefoon, memo, rapport</i>	
Andrax van den Bogaard (hierna genoemd "Uitvoerende")	Alle betrokkenen	Advies vragen (project support)	Allen	Irregulier
Uitvoerende	Stagebegeleider	Proces rapportage (projectborging)	Blog, email	Wekelijks
Uitvoerende	Bedrijfsbegeleider	Proces/inhoudelijke rapportage (projectborging)	Mondeling, blog	Wekelijks
Uitvoerende	Technische afdeling	Inhoudelijk overleg	Mondeling	Aan het eind van elke sprint

PID Bijlage B: Project planning



PID Bijlage C: Initiële Business Case

Inleiding

Doel van deze business case is om de waarde van dit project te kunnen valideren. Door middel van het doel helder te beschrijven en onderbouwen.

Redenen en voordelen

Momenteel onderzoekt Ranj de mogelijkheden en onmogelijkheden van het gebruik van AI voor het voeren van dialogen met NPC's. In het bijzonder gebruikt Ranj hiervoor het BDI (beliefs, desires, intentions) modelleerparadigma.

BDI zorgt er voor dat dialogen gegenereerd kunnen worden op basis van de BDI van een NPC. De maker van deze dialogen heeft natuurlijk wel invloed op het onderwerp, de BDI van de NPC en de inhoud van de dialogen. Maar de daadwerkelijke verloop van een dialoog is afhankelijk van BDI. Zo krijgt de speler van een spel het gevoel dat hij met een echt persoon praat. Wat helemaal duidelijk zal worden als hij dezelfde dialoog nogmaals zal voeren: de dialoog zou goed heel anders kunnen verlopen, ook al kies de speler voor dezelfde antwoorden of vragen.

Hiermee kunnen ook gesprekken en masse gesimuleerd worden, wat nuttig kan zijn om gesprekken te simuleren, die je anders met verschillende mensen daadwerkelijk zou moeten voeren.

Door een simpele basis te hebben om mee te kunnen testen in projecten, kan het gebruik van AI in dialogen met NPC's getest worden. En kan daardoor beoordeeld worden of dit voor toekomstige projecten nuttig gaat zijn.

De Game Designers zijn nog enigszins sceptisch over het gebruik van BDI voor dialogen, omdat ze dan niet meer de volledige controle hebben over het verloop hiervan. Mogelijk kan het niveau van controle ook nog verwerkt worden in het framework.

Tijd en Kosten

De kosten van dit project zullen puur bestaan uit manuren van de student en studentbegeleider.

~800 uur door de student om het project te realiseren van begin tot eind. Binnen deze tijd valt de oriëntatie, het PID schrijven, onderzoek doen, uitvoeren van het project en de scriptie schrijven. Wat neer komt op ongeveer 1000euro in de vorm van stagevergoeding.

~40-60 uur aan begeleiding door de bedrijfsbegeleider.

Risico's

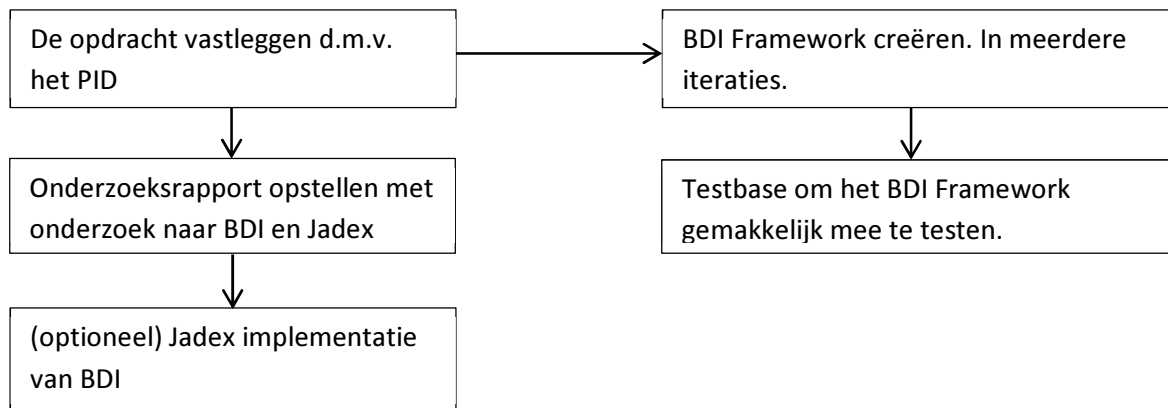
BDI is redelijk ingewikkeld, er bestaat de kans dat het opzetten van een framework niet ver genoeg gerealiseerd kan worden. Echter valt dit in het begin van het project al in te zien.

Investeringsvoorstel

De investering zal voornamelijk in de tijd van de stagiair zijn, daarom zullen de kosten aanzienlijk lager zijn dan dit door een werknemer te laten doen.

Het framework dat ontwikkeld wordt door de stagiair kan als een goede basis dienen binnen toekomstige projecten waar het van pas kan zijn. Dit biedt het bedrijf een breder scala aan oplossingen die een klant geboden kan worden, met deze potentieel kost efficiënte oplossing.

PID Bijlage D: Initieel Projectplan



II. Blog

Afkomstig van: <http://www.andrax.nl/Ranj-afstudeerstage-blog/>

Week 15

Posted on **2012 May 29**

Scriptie is deze week al weer een stuk verder gekomen qua inhoud nu ook. Wat goede feedback gehad van Ivo op mijn eerste opzet ook. Ziet er dus goed uit, maar de scriptie moet komende week echt afgemaakt worden zodat ik nog een beetje feedback kan krijgen vóór het inleveren over 2 weekjes!

Begin vorige week zat ik mezelf vooral in de weg met de scriptie schrijven: Mijn insteek was alles volledig zelf schrijven in eigen woorden. Maar ik moest accepteren dat dat niet altijd slim is, vooral niet als je een beperkte tijd (~3weken) hebt om een scriptie te schrijven.

Goed gebruik maken van referenties en ook blijven nadenken over wat je schrijft en aan neemt van iemand anders = succes.

Test case ligt uitgeschetst met mogelijke subjects en dergelijke. Maar daar begin ik pas weer aan als mijn scriptie compleet is! Dat is een werkje waar ik naar uit kijk, dus dat motiveert me om de scriptie sneller klaar te krijgen 😊

Week 14

Posted on **2012 May 21**

Deze week lekker aan mijn scriptie gewerkt en een test case om mijn project mee te gaan testen.

Scriptie heb ik qua koppen nu goed uitgedacht en ook al wat inhoud gegeven aan een aantal dingen. Gaat tot nu toe best lekker heb ik het idee! Ook naar mijn begeleiders opgestuurd om te kijken of ze dat ook vinden 😊.

Ter afwisseling ben ik ook gaan kijken naar het bedenken van een test project waarmee het BDI Framework goed getest kan worden. Het zal een vervolg op het GGR project worden. In het GGR project ging het om als makelaar de klant ver genoeg geïnteresseerd te krijgen in een huis dat ze het wel willen bekijken en er dus potentie in zien om het te kopen. Mijn idee met de testcase is dus vrij simpel: we gaan naar de volgende stap: de bezichtiging van het huis.

Hierbij kan je als makelaar van te voren het huis bezichtigen, even bedenken hoe je bepaalde mankementen beter kan verwoorden. Of eventueel meubilair op een rot plekje houten vloer plaatsen. Een kleine slaapkamer is "knus"... etc 😊 En misschien is het huis wel echt perfect voor de klant! Bepaalde eigenschappen van de klant zijn na het eerste gesprek namelijk al beter bepaald, weten dat de klant van klussen houdt bijvoorbeeld. Hiermee lijkt het spel alsnog veel op het GGR project en is het dus herkenbaar binnen de studio.

Ik had naar de mogelijkheid gekeken van Unity die nu Flash enigszins ondersteund. Dit werkt nog niet optimaal, maar ik heb er mijn AS3 Framework in draaiende gekregen zonder al te veel moeite. Mijn plan is dus om de test case uit te werken in AS3 als een extensie op het Framework. Het Framework geeft zelf een hoop trace(debug) data al die voldoende zou moeten zijn om aan te tonen hoe het werkt en wat er gebeurd. Maar dit alles is dus puur in de vorm van text en kan vrij onduidelijk zijn om uit te leggen.

Vandaar de keuze dat, mocht ik de tijd er voor hebben, ik Unity als GUI ga gebruiken om de case te

2012 Andrax vd Bogaard, student te Fontys Hogeschool Eindhoven

In opdracht van RANJ

verbeelden. Aangezien ik in Unity vrij bekend ben met een wereld snel op te zetten leek me dit een ideale gelegenheid om Flash in Unity meteen even te proberen.

Week 13

Posted on [2012 May 14](#)

De tijd gaat hard! Afgelopen week begonnen met mijn scriptie. Al meteen lekker mee bezig, ik heb in mijn hoofd al zo'n 80% aan invulling voor de scriptie zitten, waarvan zo'n 40% gedaan is momenteel. Daarnaast eens gaan kijken in Unity of Flash daar fatsoenlijk in werkt + of ik het BDI framework daarin werkend kreeg: wat beide gelukt is. Daar liggen misschien wat leuke mogelijkheden voor de visuele weergave van een GGR test-case → Als makelaars → in een 3d huis rondlopen met een klant en over het huis praten. (aka een vervolgstap op het GGR project in AS3)

Over de missende weken in mijn blog zal ik binnenkort posten! (was het blog een beetje uit het oog verloren 😊)

Week 7-12 – Onderzoek + Ontwikkelingsweken

Posted on [2012 June 6](#)

Onderzoeksvraag: “Wat is nodig om een BDI Framework werkend te maken?”

Dit stukje zal ook in de scriptie te vinden zijn uiteindelijk:

Na het vooronderzoek dacht ik duidelijk te hebben wat het BDI Model in houd. Maar de volgende stap is dit naar code te gaan vertalen. Na hier even goed over na te denken begin je aan jezelf te twijfelen. Snapte ik nou wel echt wat daar stond? Als ik om hulp vraag bij mijn stagebegeleider of op het internet kijk lijkt dat ik het echt begrijp. En toch lijkt het omzetten in code haast onmogelijk om “even” in je hoofd te krijgen. Tot de realisatie komen dat dit misschien wel niet zomaar te doen is voor mij heeft me wat tijd gekost. Gelukkig versimpel ik in dat proces dingen voor mezelf naar behapbare stukken. Maar na een paar weken ploeteren leek ik er nog niet te zijn. De concepten van JADEX en de andere bronnen die ik hierbij door loop geven me nog enige grip op het geheel. Dit is niet iets dat je even doet. Mensen komen hier na jaren lang werken tot een redelijke oplossing voor hun probleem, geen perfecte.

Door het gevoel van twijfel dat ik bij mezelf creëerde ging ik ook weer op zoek naar de basis van dit alles te begrijpen. Waarom is BDI ontstaan? Wat voor problemen hadden mensen, waar BDI als een goede oplossing uit ontstond? Ik ging zoeken naar artikelen over Agents en Dialogen in games die enigszins betrekking op BDI hadden. Zo vond ik met name in Game Programming Gems wel een aantal interessante artikelen.

“Waar komt BDI vandaan?”

Was de vraag die ik mezelf meerdere malen heb gesteld om een beter begrip van BDI te krijgen, dit is op zich geen belangrijke onderzoeksvraag. Maar puur omdat het mezelf eigen maken van de vraag “Waarom is BDI ontstaan?”, mij een goed beeld geeft wat de achterliggende gedachte-gangen zouden kunnen zijn, kan ik gemakkelijker het BDI model verwerken in code.

Week 6

Posted on [2012 March 26](#)

Vanaf dinsdag ziek geweest eigenlijk de hele week... Hopelijk de laatste keer tijdens mijn stage.

Maandag nog wel een lekker begin gemaakt met requirements/onderzoeksvragen. En een initieel class diagram met daaruit weer een start van het BDI framework v0.2.

2012 Andrax vd Bogaard, student te Fontys Hogeschool Eindhoven

In opdracht van RANJ

Sprint 2 eindigt zo helaas met een weekje minder dan ik had gehoopt: ik had gehoopt de hele week met het framework en de requirements aan de slag te gaan en hier de 2e iteratie op te maken. Dat zal echter in **week7** moeten gaan gebeuren binnen Sprint 3.

Sprint 3 zal simpelweg een continuatie van Sprint 2 zijn, bestaande uit: Requirements / Static Class Diagram / Domain Model / Framework iteraties.

Benieuwd hoe ver ik met de Engine binnen week 7 kan komen!

Week 5

Posted on [2012 March 19](#)

Na de requirements veel te ingewikkeld aan te pakken, besloot ik dat er een moment van overleg met Ivo nodig was. Gelukkig had hij de tijd en hebben we meteen even naar mijn requirements gekeken. Blijkbaar had ik iets te ver door zitten denken al in de requirements en had ik het meer in substappen moeten verdelen. Ik kwam ook al een beetje in de knoop met het bedenken er van en dat komt omdat ik simpelweg meerdere vragen op rij aan het beantwoorden was en niet eens al deze vragen duidelijk in mijn hoofd had. Ik wilde de requirements in feite zo ver uitwerken dat het meteen een compleet software model was waarbij het klassendiagram in feite ook al uitgewerkt is. Zelf had ik al een poging gedaan een stapje terug te nemen en het klassendiagram als 2e stap naar te zetten, maar dit bleek ook niet te werken. Dus vandaar het overleg met als resultaat een gestructureerde aanpak waarmee in een vroeg stadium van requirements overlegd kan worden hierover. En ook al beslissingen gemaakt kunnen worden wat er nou wel en wat er nou niet nodig gaat zijn.

Side-note: Nu dat ik terug kijk op mijn sprint task post-its, lees ik de aantekening aan mezelf: "Houdt rekening met de onderzoeksvragen"... dat is precies wat ik niet gedaan had dus... Do I feel stupid now or what! Toch fijn om te weten dat ik van mezelf weet dat ik dit soort dingen wel eens vergeet en daar dus juist een aantekening voor maak... Just have to read it next time 😊

Hier had ik gelukkig maar 3 dagen van de week aan besteed dusver. De laatste 2 dagen had ik dus de tijd om naar mijn onderzoeksvragen te kijken en daarmee de requirements invulling te geven! En mezelf even voor te bereiden op de open dag @ Fontys Hogeschool Eindhoven van zaterdag. Leuk om eens de tijd te hebben wat meer games van Ranj te zien 😊 vooral de rappende hamster techneuten blijven erg hangen.

Na dit weekend ook weer even met mijn docenten en collegae te hebben gepraat heb ik ook weer het idee dat het presenteren van mijn project aan de mensen bij RANJ ook belangrijk kan zijn. Tussentijdse korte presentatie(s) kunnen misschien interessant zijn, om zowel het belang van het project over te brengen, als simpelweg "Wat doet deze student hier nou eigenlijk!".

Week 6 dan weer om de onderzoeksvragen en requirements in iteraties te gaan bespreken en verder uit te werken.

Het liefst in week 6 ook ruw vast te leggen wat ik ga maken, zodat ik aan de slag kan met het framework te bouwen. We gaan er namelijk van dat de details nogal kunnen veranderen tijdens het ontwikkelen, vandaar in het begin ook de keuze voor een iteratief proces.

Week 4

Posted on [2012 March 12](#)

Beginnetje gemaakt met een simpel statisch klassendiagram. Na overleg met mijn begeleider leerde ik gelukkig vrij snel dat hij ook een kijk wou op een iets algemener vlak: de engine + het framework en

2012 Andrax vd Bogaard, student te Fontys Hogeschool Eindhoven

In opdracht van RANJ

de mogelijke tools/apps/template projecten, etc die op basis van het framework(met de engine) gemaakt kunnen worden.

Na dat uitgebreid te beschrijven, vormde dit een mooie inleiding op het Requirements document! Wat nu mijn volgende stap in dit project is. Goed requirements en een klassendiagram gaan het werk gemakkelijk maken. Ik merk nu al dat ik een groot deel van de invulling hierdoor in me hoofd begin te krijgen.

Helaas halverwege de week toch ook griepig geworden en de woensdag weer een dagje in bed besteed. (begint al goed met een 2e ziekte dag :/)

Donderdag heb ik nog een meeting bij mogen wonen van het TNO project (waar mijn project uit ontstaan is). Handig om te zien wat de ideeën waren bij het ontwikkelen daarvan en wat ze dachten er verder mee te kunnen. En natuurlijk helemaal belangrijk: wat voor mogelijkheden liggen hierin commercieel en wat is nou dus echt belangrijk om in mijn project te hebben.

Daarnaast ook weer leuk om te merken dat het game design wereldje in Nederland relatief klein is: Annerieke (van TNO) werkte ook mee aan dit project en was er bij. Dat is weer iemand die ik via de Games for Health beurs door Menno Deen heb leren kennen!

On to **week5** om deze verder uit te werken! Daarnaast is dinsdag in week 5 het eerste stagebezoek van mijn stagebegeleider, ben benieuwd wat ie over het PID te vertellen heeft 😊

Week 3

Posted on [2012 March 6](#)

Na het research gedeelte met Ivo te bespreken ben ik mijn PID eindelijk verder invulling gaan geven.

Mijn initiële planning bestond uit een sprint van week 1-3, om een kleine basis van het BDI framework te maken, naast het PID en beginnend research. Na week1/2 leek het gebrek aan kennis dit onmogelijk te maken. Maar na een stevige 2 weekjes kennis op doen van het onderwerp leek het toch mogelijk om een begin te maken.

En dat is ook gelukt! v0.1 van het framework is gemaakt op basis van een simpel scenario en te kijken naar de bestaande game (Glengarry Glenn Ross). Iets wat voor een nog beter beeld zorgde op de beoogde werking van het framework, de benodigde onderdelen en de potentiële tijd die deze onderdelen gaan kosten. En omdat dit ook het uitgangspunt van v0.1 was, is de code relatief slordig opgezet. **Note to self:** "Do not use this project as a code base! It'll only get messier."

Hiermee heb ik het PID deze week voor het grootste deel rond kunnen krijgen, tijd om het te overhandigen aan mijn bedrijfsbegeleider.

Week 4 wordt het tijd de requirements en klassendiagrammen eindelijk rond te krijgen. (*eindelijk*, als in: statische versies van het klassendiagram kwam ik nog niet ver genoeg mee zonder de v0.1 van het framework, wat dus nog één rede was om dit te maken)

De algemene planning in het PID is gemaakt en het is tijd om Sprint2 planning vast te stellen. Voor nu bestaat deze uit het Requirements document en de blogs weer eens bijwerken! *Which is hereby done, on to week 4!*

Week 2

Posted on [2012 March 6](#)

Week 2 vergeten te bloggen, maar here it is alsnog!

Begonnen aan het PID deze week, maar ook vooral aan het research. Zonder research kon ik namelijk moeilijk ergens komen in het PID. Ik heb het research mogelijk iets te breed aan willen pakken, maar uiteindelijk vind ik het toch altijd prettig even zo breed mogelijk te beginnen om te weten wat er allemaal bestaat en wat er allemaal bij dit onderwerp hoort. Ivo gaf gelukkig al aan dat menig onderzoeker over deze hoeveelheid stof goed een jaar doet, puur in onderzoek, niet eens realisatie. Dus dat moest beperkt worden tot puur BDI voor dit project.

Helaas was het een kort weekje, want donderdag heb ik me ziek gemeld. Iets te veel nieuwe info in mijn hoofd + gemiddeld 5 uur slaap per dag gingen blijkbaar niet helemaal samen 😊 Gelukkig gaf een dag rust mij de tijd om al deze stof eens in mijn hoofd na te lopen en vrijdags werk ik sowieso thuis, dus heb ik die dag lekker de tijd genomen om mijn kennis van AI in de richting van BDI en Agents in het algemeen nog wat te verbreden.

Al met al een productief weekje qua research!

Week 1

Posted on [2012 February 14](#)

Afstudeerstage gestart!

Het eerste weekje is natuurlijk wennen aan het bedrijf, de mensen en mijn opdracht duidelijk proberen te krijgen.

Na al een idee te hebben gekregen welke stukken van BDI belangrijk zouden zijn voor Ranj (na een eerste gesprek met mijn bedrijfsbegeleider), is mijn eerste plan om BDI te gaan onderzoeken. En tegelijkertijd om meer over AS3 te leren deze eerste week.

Om zo vervolgens de scope en complexiteit van het project vast te kunnen stellen. De vraag “of BDI wel de beste manier van aanpak is voor Ranj” is niet echt te beantwoorden, omdat BDI één aanpak is van meerderen. Maar het is een keuze die voor de hand liggend is, gezien dit de aanpak is die het afgelopen jaar in een project is gebruikt en het toekomstig gebruik sowieso experimenteel is. Vandaar dat ik mij ook bij BDI zal houden.

Daarnaast ben ik begonnen een planning te maken, dit stukje blog klaar te zetten, pc indelen en wat andere praktische dingen.

Getting started...

Het eerste wat ik merk is dat BDI complexer is dan ik in eerste instantie voor ogen had. De laparoscopische cholecystectomie van het bootcamp vorige week was prima behapbaar, maar dit is toch flink wat abstracter...

Als ik er over na denk klinkt BDI nog best makkelijk... Maar hoe ga je dat in code brengen.

Aanpak 1: Keep It Short and Simple

Na begonnen te zijn aan de code en documentatie van JADEX kom ik er achter dat deze stof met moeite blijft hangen. Dus besloot ik toch al in AS3 een poging te wagen mijn 1e versie van het BDI project te gaan bouwen.

Dit creëert bij mezelf meteen vragen en zo ook een beter begrip op het denkproces van de makers van JADEX. Wij mensen zijn nou eenmaal egoïstisch ingesteld en door het probleem in codevorm mezelf meer eigen te maken staan mijn hersenen beter open voor oplossingen. Plus het feit dat het nu in wat kleinere delen komt, maakt het gemakkelijker te behappen. “Hoe breng je nou eigenlijk een Desire, een Intention, of een Belief om in code.”

Aan de slag met het volgende korte stappenplan:

stap1: Probeer BDI in zijn essentie, zo kort en duidelijk mogelijk uit te leggen.

1.1 Wat heeft Ranj van deze essentie nodig.

stap2: Hoe kan je BDI in code brengen.

2.1 Kijk naar bestaande aanpak : JADEX

2.2 Probeer een eigen aanpak te vinden die past binnen het kader van Ranj

De pogingen om het huidige project werkend te krijgen in Flash en Java(JADEX), met de SmartFox server die als communicatiemiddel tussen de 2 draait, gaan al met veel moeite. Meteen wordt de rede duidelijk dat er liever een AS3 library is dan deze aanpak... Ik heb het aan het eind van de week namelijk nog niet volledig werkend gekregen.

Aanpak 2: Het volledige onderzoek

Na het huidige project van Ranj te proberen, zelf een project te bouwen(met beperkt onderzoek) en enigszins de JADEX code door te werken... Merk ik dat dit niet short & simple (als bij gamejams en dergelijke) te doen is. Maybe I got a bit overconfident from the bootcamp last week, where the subject was some complicated matter as well (Laparoscopische Cholecystectomy).

Week1 bood gelukkig de tijd om een beetje te experimenteren. Ik heb daardoor een breed idee gekregen van de volgende dingen:

- wat BDI betekent
- wat BDI voor Ranj gaat betekenen
- hoe ik het project globaal aan ga pakken (en mijn planning dus verder uit kan werken)
- hoe ik met Flash/AS3 moet werken en wat de beperkingen zijn (multi-threading is niet daadwerkelijk beschikbaar bijvoorbeeld)

Nu dus aan mij om al de dingen hierboven in documenten te gaan verwerken op een nette manier, mijn PID(Project Initiation Document) samen te stellen en te gaan beginnen met een 1e sprint.

Week 1 samengevat

Een hoop geprobeerd, jammer genoeg niet al een 1e iteratie van een project meteen kunnen maken. Maar des al niet te min een hoop informatie binnen gekregen. (Mijn plan van aanpak is overigens al wel bij de 4e iteratie inmiddels)

Ik had deze blogpost nog wat uitgebreider willen doen, maar dit is wel meer dan genoeg informatie. Time to get started on week 2!

III. Requirements

What?	Description/elaboration
Beliefs	
Elk Belief bestaat uit een feit dat het omschrijft, dit moet zicht- en wijzigbaar zijn voor het Framework.	
Een Belief zijn feit, moet kunnen laten weten aan andere objecten wanneer deze veranderd.	Listeners moeten mogelijk zijn / Event dispatcher.
Beliefsets	
Bevat simpelweg een lijst van Beliefs die tot een zelfde soort / categorie behoren.	
Goals	
Een Goal heeft een tijdsspan. Dit kan variëren tussen een vast tijdsspan en (nog) onbepaald tijdsspan.	
Aan een Goal zitten pre-condities.	Als er aan de pre-condities voldaan wordt, kan het Goal geadopteerd worden.
Aan een Goal zitten contextcondities waar op gecontroleerd worden als het Goal actief/suspended is.	Afhankelijk daarvan kan het Goal in een suspended state gezet worden, waardoor het later weer hervat kan worden zodra weer aan de contextcondities voldaan wordt.
Aan een Goal zitten dropcondities die bepalen wanneer de Goal beëindigd wordt.	Droppen van de Goal betekent dat het niet meer gereactiveerd kan worden.
Een Goal moet verschillende states kunnen hebben na adoptatie : active / suspended / option	Option staat voor inactief, maar de optie staat open. Vergelijkbaar met een media speler: "play / pause / stop"
Een Goal heeft een lijst met mogelijke plannen tot zijn beschikking.	
Een Goal moet kunnen laten weten wanneer hij veranderd van <i>state</i> .	Een Goal moet aan de hand van <i>state-changes</i> en <i>finished</i> zijn, een Event Dispatchen. Dit is zodat bijbehorende Plans/Goals/andere Agents weten van de state-change en ook zelf daar naar kunnen handelen.
Een Goal moet de mogelijkheid hebben opnieuw geïnitieerd(recurring) te kunnen worden	Heeft een bepaald delay op het terugkeren. En er moet vastgelegd worden binnen het goal of dit kan of niet.
Een goal moet opnieuw geprobeerd (retryable) kunnen zijn.	Als het doel niet direct gehaald kan worden, moet het wel (met een mogelijke delay) opnieuw geprobeerd kunnen worden, totdat het bereikt is of er geen plannen meer beschikbaar zijn.
Een Goal moet "unique" kunnen zijn	
? Er moeten meerderen van dezelfde Goal kunnen bestaan?	
Goals moeten Parameters kunnen bevatten	Parameters zijn in feite referenties of properties. Soortgelijk aan een Belief, al is dit niet een property die een Belief van een agent is, maar een property die bij het Goal hoort.
Aan een Goal zit altijd een stukje Deliberation dat bepaald wat hij moet doen.	
Easy Goal Deliberation strategy	<i>conceptual limitations:</i> – The strategy does only consider bilateral relationships. Hence, it is impossible to specify e.g. that two goals together are more important than another single goal.

	– Conflicts between subgoals cannot always be resolved optimally, e.g. when a conflict between subgoals could be resolved by replacing one of the subgoals with another non-conflicting subgoal [14]. – Conflicts at plan level are not considered, which means that inconsistencies between plans e.g. because of access to conflicting resources are not detected. – Positive interactions between goals are not considered, which means that the strategy cannot identify and exploit potentially common subgoals.
Een Goal moet andere Goals kunnen “inhibiten” (parenting) voor deliberation	Dit betekent dat een parent goal in feite eerst uitgevoerd wordt, en de children pas daarna. Met name handig voor bijv een “maintain goal”.
Perform Goal	Puur bedoeld om activiteiten uit te voeren.
Achieve Goal	Bedoeld om een doel staat te bereiken.
Er zit een bepaalde conditie aan dit type goal die bereikt geacht te worden.	Vaak is dit een Belief met een beoogde waarde bijv 0.6, die huidig op 0.4 staat. Dit type goal zal dan proberen 0.6 te halen.
Query Goal	Bedoeld om informatie te achterhalen
Maintain Goal	Heeft het doel om een gewenste status te monitoren en deze status te bewaken, indien hij af wijkt.
Meta Goal	
Plans	Een plan bestaat uit de concrete acties die een agent uit voert. Vaak om zijn Goals te bereiken.
Plan moet kunnen failen ?	
- priority	Om een prioriteit te kunnen geven aan bepaalde plannen, bij het reasoning proces.
Body	Bedoeld om te specificeren welke implementatie-class aan dit Plan behoort. Deze class is bedoeld om het plan daadwerkelijk uit te voeren.
Trigger Possibly triggered by: - Goal - Messageevent - Factchanged - Factadded - Internalevent	Trigger, door welke Goals/Plans/Events mag dit Plan triggered worden.
Precondition	
Contextcondition	Dit zijn condities waar tijdens het uitvoeren van het plan op gecontroleerd wordt, als niet aan deze condities wordt voldaan zal het plan suspended worden. Zodra weer wordt voldaan aan de contextconditions zal het weer hervat worden.
Dropcondition	
Waitqueue	
Parameters/Parameter Sets - value	

- bindingoptions	
- mapping (internalEvent/ messageEvent/ Goal)	Kan in een wat simpelere variant gedaan worden (simpelweg parameters meegeven aan het plan)
Events	Events zijn bedoeld om Beliefs te updaten, Plans te triggeren, of Goals aan te passen.
MessageEvent	Events voor communicatie tussen verschillende Agents . Kan een eigen format aan gegeven worden.
InternalEvent	Deze events komen van binnen een Agent. Dit kan ook opgelost worden simpelweg met function calls.
Deliberation	



IV. Technische Implementatie

Software libraries used in the framework

Flex SDK version:	Flex SDK 4.6.0
Framework created in:	Flex Builder 4.6
Tested in Flash Player version:	Flash 11.1
XML usage for Ontology input	Not implemented
Signals Framework:	v0.8-217

Used existing RANJ Software libraries

Ranj Simulation library	last revision update on SVN: 1099 (JSON compatibility) @ 2012-apr-25
-------------------------	--

Met hierbij een lijst van Usage in de implementatie, aan de hand van de Requirements.

Note: De “Notes on implementation” bevatten ook suggest naar Ranj toe m.b.t. doorontwikkeling van het BDI Framework.

What?	Usage	Description/elaboration	Notes on implementation
Beliefs			
Elk Belief bestaat uit een feit dat het omschrijft, dit moet zicht- en wijzigbaar zijn voor het Framework (zowel interne als externe bronnen). <i>Extern is over het algemeen d.m.v. sensoren die de Agent monitort in zijn omgeving.</i>			In de vorm van Fact geïmplementeerd, dit Fact kan een Object van allerlei types zijn.
Een Belief zijn feit, moet kunnen laten weten aan andere objecten wanneer deze veranderd.		Listeners moeten mogelijk zijn / Event dispatcher.	
Een Belief heeft een benaming om te beschrijven wat het omvat.			
Beliefsets			
Bevat simpelweg een lijst van Beliefs die tot een zelfde soort / categorie behoren.			Niet volledig geïmplementeerd in het huidige Framework, omdat de implementaties relatief klein zullen zijn bij RANJ is het niet nodig om deze

			in een categorie te plaatsen. Daarnaast zal de Simulation Editor van RANJ zelf al voor een mate van overzicht zorgen. Het is dus niet aan te raden hier gebruik van te maken, tenzij de nood voor overzicht hoog is. Maar dan moeten de bases alsnog volledig geïmplementeerd worden binnen de Engine.
Goals			
Een Goal heeft een tijdsspan. Dit kan variëren tussen een vast tijdsspan en (nog) onbepaald tijdsspan.			
Aan een Goal zitten pre-condities.		Als er aan de pre-condities voldaan wordt, kan het Goal geadopteerd worden.	
Aan een Goal zitten contextcondities waar op gecontroleerd worden als het Goal actief/suspended is.		Afhankelijk daarvan kan het Goal in een suspended state gezet worden, waardoor het later weer hervat kan worden zodra weer aan de contextcondities voldaan wordt.	
Aan een Goal zitten dropcondities die bepalen wanneer de Goal beëindigd wordt.		Droppen van de Goal betekent dat het niet meer gereactiveerd kan worden.	
Een Goal moet verschillende states kunnen hebben na adoptatie : active / suspended / option		Option staat voor inactief, maar de optie staat open. Vergelijkbaar met een media speler: "play / pause / stop"	
Een Goal heeft een lijst met mogelijke plannen tot zijn beschikking.			
Een Goal moet kunnen laten weten wanneer hij veranderd van <i>state</i> .		Een Goal moet aan de hand van <i>state-changes</i> en <i>finished</i> zijn, een Event Dispatchen. Dit is zodat bijbehorende Plans/Goals/andere Agents weten van de state-change en ook zelf daar naar kunnen handelen.	
Een Goal moet de mogelijkheid hebben opnieuw geïnitieerd(recurring) te kunnen worden		Heeft een bepaald delay op het terugkeren. En er moet vastgelegd worden binnen het goal of dit kan of niet.	To be done later; But is a very important 1: if it can't be achieved straight away, it needs "recur conditions" to define under what conditions it will be pursued again.
Een goal moet opnieuw geprobeerd (retryable) kunnen zijn.		Als het doel niet direct gehaald kan worden, moet het wel (met een mogelijke delay) opnieuw geprobeerd kunnen worden, totdat het bereikt is	

		of er geen plannen meer beschikbaar zijn.	
Een Goal moet “unique” kunnen zijn			Can be done later; error thrown now, if it already exists
Er moeten meerderen van dezelfde Goal kunnen bestaan			Wordt nog niet implemented for now...
Goals moeten Parameters kunnen bevatten		Parameters zijn in feite referenties of properties. Soortgelijk aan een Belief, al is dit niet een property die een Belief van een agent is, maar een property die bij het Goal hoort.	
Aan een Goal zit altijd een stukje Deliberation dat bepaald wat hij moet doen.			
Easy Goal Deliberation strategy		<i>conceptual limitations:</i> – The strategy does only consider bilateral relationships. Hence, it is impossible to specify e.g. that two goals together are more important than another single goal. – Conflicts between subgoals cannot always be resolved optimally, e.g. when a conflict between subgoals could be resolved by replacing one of the subgoals with another non-conflicting subgoal [14]. – Conflicts at plan level are not considered, which means that inconsistencies between plans e.g. because of access to conflicting resources are not detected. – Positive interactions between goals are not considered, which means that the strategy cannot identify and exploit potentially common subgoals.	
Een Goal moet andere Goals kunnen “inhibiten” (parenting) voor deliberation		Dit betekent dat een parent goal in feite eerst uitgevoerd wordt, en de children pas daarna. Met name handig voor bijv een “maintain goal”.	Note: Jadex has: “When_active” and “when_in_process”, but we won’t be making this distinctive difference.
Perform Goal		Puur bedoeld om activiteiten uit te voeren.	
Achieve Goal		Bedoeld om een doel staat te bereiken.	



Er zit een bepaalde conditie aan dit type goal die bereikt geacht te worden.		Vaak is dit een Belief met een beoogde waarde bijv 0.6, die huidig op 0.4 staat. Dit type goal zal dan proberen 0.6 te halen.	
Query Goal		Bedoeld om informatie te achterhalen	
Maintain Goal		Heeft het doel om een gewenste status te monitoren en deze status te bewaken, indien hij af wijkt.	
Meta Goal			
Plans		Een plan bestaat uit de concrete acties die een agent uit voert. Vaak om zijn Goals te bereiken.	
Plan moet kunnen failen ?			
- priority		Om een prioriteit te kunnen geven aan bepaalde plannen, bij het reasoning proces.	
Body		Bedoeld om te specificeren welke implementatie-class aan dit Plan behoort. Deze class is bedoeld om het plan daadwerkelijk uit te voeren.	
Trigger Possibly triggered by: - Goal - Messageevent - Factchanged - Factadded - Internalevent		Trigger, door welke Goals/Plans/Events mag dit Plan triggered worden.	
Precondition			
Contextcondition		Dit zijn condities waar tijdens het uitvoeren van het plan op gecontroleerd wordt, als niet aan deze condities wordt voldaan zal het plan suspended worden. Zodra weer wordt voldaan aan de contextconditions zal het weer hervat worden.	
Dropcondition			
Waitqueue			
Parameters/Parameter Sets			



- value - bindingoptions			
- mapping (internalEvent/ messageEvent/ Goal)		Kan in een wat simpelere variant gedaan worden (simpelweg parameters meegeven aan het plan)	
Events		Events zijn bedoeld om Beliefs te updaten, Plans te triggeren, of Goals aan te passen.	
MessageEvent		Events voor communicatie tussen verschillende Agents . Kan een eigen format aan gegeven worden.	
InternalEvent		Deze events komen van binnen een Agent. Dit kan ook opgelost worden simpelweg met function calls.	
Deliberation			
Advanced Technical Requirements			
Agent			
<i>SOME GENERALLY USED FUNCTIONS</i>		This makes it easier to call these functions from a custom Class or a custom Plan class. This way you don't have to find the "right place" to locate those functions	Should be done in later versions.
<i>BeliefBase / GoalBase / PlanBase</i>		Dit zijn voor de Agent de locaties waar alle Beliefs/Goals/Plans worden (in BDI vaak ook genoemd de Capabilities) gemanaged.	Elke call/change/removal van Beliefs/Goals/Plans worden dus in hun bijbehorende Base gedaan.
IElement		Base interface voor alle Beliefs, Plans, Goals en Events.	
Name			
Destructor functie.			
Means-End Reasoner		Picks the plan to perform after goal-deliberation is done.	
Goal-deliberation		First, a goal can become an option either when a new goal is adopted or when the context of a suspended goal becomes valid again. In these cases the deliberation process needs to decide whether the new option can be activated and additionally what the consequences of the	It is done upon the Goal that triggered deliberation. All goals with inhibition relationships to the triggering goal will have to be considered.



		activation are, i.e. which other active goals need to be deactivated to avoid having conflicting goals (1: Deliberate new option). Second, an active goal can become inactive when it gets suspended, finished or dropped. In this case, the deliberation has to determine which options(e.g. "retryable") have been possibly inhibited by the deactivated goal. For each of these options it needs to be checked whether it can be reactivated (2: Deliberate deactivated goal).	
Inhibition		Basically a way of determining which Goal should always come above others.	Keep in mind that this brings a lot of changes in the code, possibly not only in the Goals + Goal deliberation itself.
Priority		Cardinality number to determine which Goal should be prioritized over another, if they're both inhibited by another Goal (and are on the same level)	
Plan Availability		Check if there's a Plan available to perform the Goal, if not remain an Option instead of going to Active.	Note: Currently counting on this to run well, but this could be the only possible Deliberation step I'd foresee as a possible Must, if it doesn't run as planned.
Goals			
Every Goal is Unique and initiated from the init of the Agent (which means it's in OPTION state straight away)			Issue: creation/adoption is skipped due to this: To solve the creation condition issue: ADOPTED property (false/true) → true is meets creation conditions and can be adopted.
Creation conditions			Currently ignoring these. Should be implemented later. Probably best to do creation at Execute phase(5) and adoption in option generation phase(2).
Context/drop condition checking			Zouden gechecked moeten worden bij elke execution round... of zouden getriggered moeten worden door property changes zelf (die events naar de eventmanager sturen, die vervolgens weer kan kijken welke Goals/Plans hier op



			listenen en aan de hand daarvan Goal/Plan state changes triggeren → die weer een event/option veroorzaken in de Agenda)
onFinished		Actually finish up the Goal, checks if it's re-runnable, etc..	
Goal Finish		Goal gets a list of possible plans, any dropped plans are removed from this list. Running out of possible plans = finish. Having no currently use-able plans = turn it back into an option.(having only suspended plans / no plans that meet creation condition, etc...)	
Goal Creation			Possible TODO: Upon Goal creation also create new (copies) of a Plan, if the plan has multiple triggers(?)
Reasoner (holding the Execution Model)			Going to limit the option generation to Goals for now...
1 Agent-init		Not done here...	
2 Option generator		Get options from Agenda	Skipping this for now...
3 Deliberate		Deliberate on goals/plans based on current active ones...	Is split into deliberation of options... and doing goal + plan deliberation. For starters: just doing goal + plan delib... in a simple form.(no goal linkage is done, so can only check the 1 Goal itself)
4 Update-intentions		And update the goals/plans accordingly	
5a Execute		Execution of plans.	
5b Condition Evaluation		Evaluate the conditions of the plans/goals	We don't have Listeners on any Conditions, so Goal state changes will also have to be re-checked in here.
6 Drop succesful-attitudes		After evaluation → drop successful goals	
7 Drop impossible-attitudes		After evaluation → drop impossible goals / termination of plans	
Agenda (in Reasoner)			
Array with agenda items			
Agenda item (probably Internal / external Event)			
Plan			



PlanSteps (should be implemented once Goal-deliberation is finished)			Preferably the plan steps (which are at 3-4-2012: functions), will be made into classes with their own conditions.
On finished		finishes up the plan, checks if it's re-runnable, etc..	
On fail: throw exception / event		Bijv, een waarde die opgehaald wordt is null. Cleaner example: wastebin is nog niet gevonden → opruimen plan niet mogelijk.	
Multiple Instances of a Plan		Allowing multiple instances of 1 Plan to be created	This will probably require some time of PlanFactory, as Goals also create a link with their Plans. (On Goal creation the PlanFactory would be called.)
Event-handler		Message handling for internal and external messages. Sends them out to the right internal section of the Agent.	
Parameter			
Value			Can be any type, including custom Classes.
Binding			Not quite sure what this is... Mapping should be fine for now.
Mapping		Mapping on Parameters that are contained by other objects(Goals / Events) that could have activated this plan.	May not be needed to handle this part straight away, because technically mapping is inherent to saving an object value.
ParameterSet		Set of multiple parameter values	Won't implement these any time soon... can be done with single values as well.
Conditions			Possible TODO Simplify into a more broadly use-able Condition Interface that can also make use of Ranj.logic.LogicalExpression Or simply replace it with that.



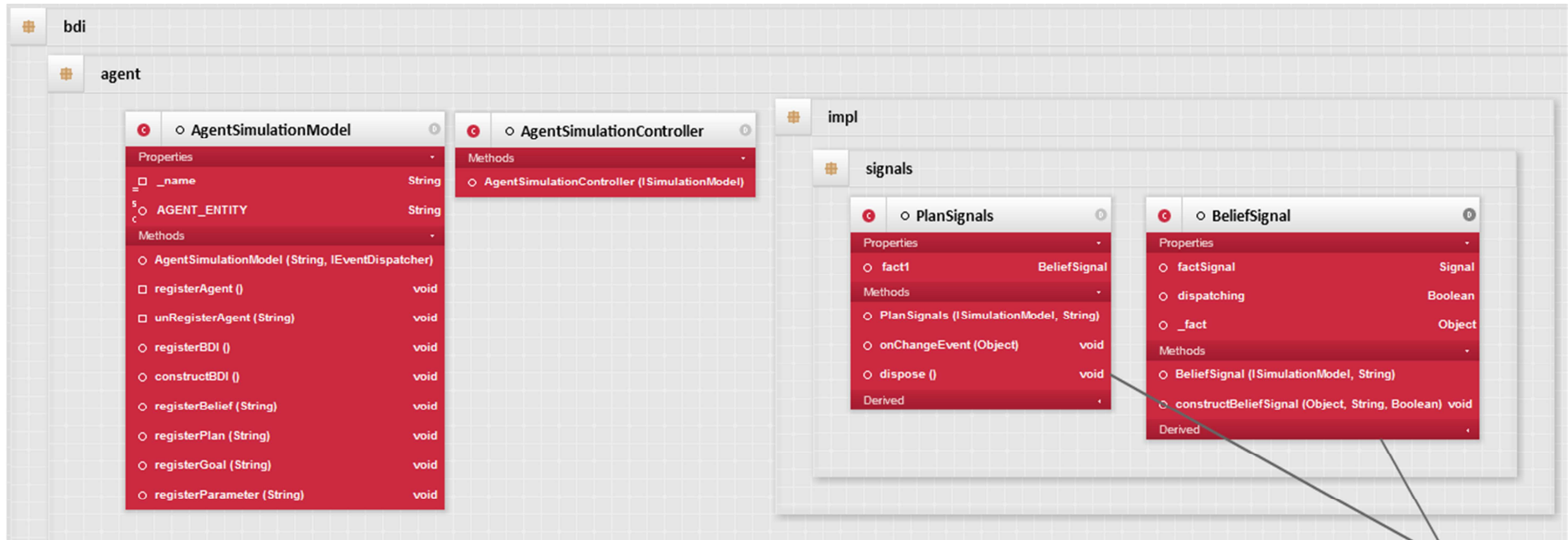
Klassendiagram

Het klassendiagram is helaas te groot om leesbaar op 1 pagina te tonen, om toch een idee te geven de implementatie zal het toegevoegd worden in delen.

Overzicht package structuur van BDI package:

- 1. agent
 - 1.1. impl
 - 1.1.1.signals
 - 1.2. model
 - 1.2.1.base
 - 1.2.1.1. events
 - 1.2.1.1.1. plan
 - 1.2.1.1.2. goal
 - 1.2.1.2. plan
 - 1.2.1.3. reasoning
 - 1.3. interfaces

Op de eerst volgende pagina is een overzicht te zien, dat niet leesbaar is maar wel toont hoe de losse stukken een geheel vormen.



interfaces

