

# Fit to rule

Verschillen, eigenschappen, voordelen en gebruik van rule engines voor het .NET framework

---

SmartRules Studio  
2010.08

NOBRE

DROOLS.NET

---



# Fit to rule

## Verschillen, eigenschappen, voordelen en gebruik van rule engines voor het .NET framework

### Afstudeerder:

Tom Dillen (2121444)  
HBO Informatica - Software Engineering

### Met dank aan:

Docent-begeleider: P. vd Broek (Fontys Hogeschool Eindhoven)  
Proces-begeleider: C. van Santen (Sogyo B.V.)  
Technisch-begeleider: A. den Uijl (Sogyo B.V.)

### Fontys Hogeschool Eindhoven

Rachelsmolen 1  
5612 MA, Eindhoven  
[www.fontys.nl](http://www.fontys.nl)  
tel: 877-877877

### Sogyo B.V.

Utrechtseweg 301  
3731 GA, De Bilt  
[www.sogyo.nl](http://www.sogyo.nl)  
tel: 030 220 2216



## Voorwoord

Dit document bevat mijn hbo informatica afstudeerscriptie m.b.t. het toepassen van rule engines in het .NET framework. De inhoud van dit document heeft betrekking op mijn afstudeerstage bij Sogyo B.V. gevestigd in De Bilt. De afstudeerperiode is ingegaan op 1-sept-2010 en zal eindigen op 28-jan-2010.

Dit document is bedoeld om inzicht te geven in het gebruik van rule engines in een .NET omgeving. Ook zal dit document het begrip “rule engine” toelichten. Aangezien er meerdere rule engines voor .NET zijn, worden ook de eigenschappen van deze rule engines vergeleken.

In dit document staan ook de conclusies die getrokken konden worden na het implementeren van drie verschillende rule engines.

Dit onderzoek naar de verschillen tussen .NET rule engines is gedaan, zodat Sogyo meer inzicht krijgt m.b.t. het toepassen van .NET rule engines. Hierbij is het belangrijk dat het duidelijk is wanneer een rule engine toegepast kan worden, welke problemen de diverse rule engines op kunnen lossen en welke rule engines gebruiksvriendelijk zijn.

Tijdens dit onderzoek ben ik uitstekend begeleid door mijn bedrijfsbegeleiders Arno den Uijl en Christa van Santen. Ook gaat mijn dank uit naar de docent Peter van den Broek van Fontys Hogeschool Eindhoven die mij heeft begeleid met diverse documentatie en procedures.

Tom Dillen  
De Bilt, 6-jan-2010

## Inhoudsopgave

Voorwoord .....	3
Inhoudsopgave .....	4
Samenvatting.....	7
Begrippenlijst.....	8
1. Inleiding .....	9
2. Het bedrijf.....	10
2.1 Missie.....	10
2.2 Activiteiten .....	10
2.2.1 Detachering & Consultancy .....	10
2.2.2 Development .....	10
2.2.3 Sogyo Academy .....	11
2.3 Markt .....	11
3. De opdracht .....	12
3.1 Probleemstelling.....	12
3.1.1 Begin situatie .....	12
3.1.2 Voordelen voor de organisatie.....	12
3.1.3 Bijdrage aan de bedrijfsdoelstelling .....	13
3.2 Doelstellingen .....	13
3.2.1 Hoofddoelstelling .....	13
3.2.2 Subdoelstellingen .....	13
3.2.3 Bereiken van de doelstellingen .....	13
3.3 Planning .....	14
3.3.1 Globale planning.....	14
3.3.2 Uiteindelijke verloop van de planning.....	14
3.4 Werkwijzen en methoden .....	15
3.4.1 Scrum.....	15
3.4.2 Deskresearch .....	15
3.4.3 Grounded Theory .....	16
4. Onderzoek: Wat zijn Rule Engines? .....	17
4.1 Proces van een rule engine .....	17
4.1.1 Knowledgebase .....	17
4.1.2 Execution core .....	17



4.2 Hulp van een rule engine.....	18
4.2.1 De voordelen .....	18
4.2.2 Rule engines als oplossing.....	19
4.3 No Silver Bullet .....	19
4.4 Rule engines en Artificial Intelligence .....	19
5. Onderzoek: De verschillen in .NET Rule Engines .....	20
5.1 Algemene resultaten .....	20
5.2 Rule engine selectie.....	21
5.3 Wijziging .....	21
6. Toepassen van .NET Rule Engines .....	22
6.1 Planning .....	22
6.2 Drools .NET .....	23
6.2.1 Implementatie .....	23
6.2.2 Werken met Drools .NET .....	24
6.2.3 Bevindingen .....	25
6.3 NxBRE 3.2.0 .....	26
6.3.1 Implementatie .....	26
6.3.2 werken met NxBRE .....	27
6.4 SmartRules.....	28
6.4.1 Implementatie .....	28
6.4.2 Werken met SmartRules .....	29
6.4.3 Bevindingen .....	30
6.5 Resultaat van de implementaties.....	31
6.5.1 Drools .NET 3.0 .....	31
6.5.2 NxBRE .....	31
6.5.3 SmartRules.....	31
6.5.4 Aanbeveling .....	31
7. Conclusie .....	32
Evaluatie .....	33
Overwinningen .....	33
Planning .....	33
Werkzaamheden .....	33
Implementatie .....	33
Uitdagingen .....	33
NxBRE .....	33



Wat heb ik geleerd? .....	34
Discipline .....	34
Opstellen van deelvragen.....	34
Prettige ervaringen.....	34
Invloeden m.b.t. mijn functioneren .....	34
Onvoldoende werkgeheugen .....	34
Literatuurlijst en bronvermelding .....	35
Internetpagina's .....	35
Overige bronnen .....	35
Bijlagenoverzicht .....	36

## Samenvatting

Onderzoek naar de verschillen van diverse .NET rule engines is de afstudeeropdracht die ik bij Sogyo B.V. te De Bilt heb uitgevoerd.

Sogyo is een software development bedrijf die zich ook richt op detachering. Het bedrijf bestaat sinds 1995 en heeft inmiddels ongeveer 75 medewerkers.

Een van de klanten van Sogyo heeft een .NET applicatie m.b.t. verzekeringen. Dit is een zeer complex systeem en maakt daarom gebruik van een rule engine. Deze rule engine kan echter niet meer gebruikt worden en dus zal er onderzoek gedaan moeten worden naar een alternatieve rule engine.

Dit is begonnen met het specificeren van de afstudeeropdracht en vervolgens het opstellen van een plan van aanpak. Bij het opstellen van het plan van aanpak zijn er verschillende taken omschreven en ingepland.

De eerste taak van dit afstudeerproject is een onderzoek naar het begrip “rule engines” en zal op basis van een deskresearch worden uitgevoerd. Er is hier voor de deskresearch methode gekozen, omdat er informatie verzameld moest worden die door specialisten geschreven is. De bedoeling van dit onderzoek is het omschrijven van wat een rule engine kan, hoe deze werkt, wat de voordelen zijn en wanneer het voordelen heeft om een rule engine toe te passen. Uit dit onderzoek is gebleken dat een rule engine een hulpmiddel is om complexe situaties en problemen op te lossen. Ook kan een rule engine gebruikt worden om het beheer en onderhoud van een applicatie gemakkelijker te maken. De rule engine kan dit doen door business rules toe te passen op data uit het werkgeheugen en deze zo te beïnvloeden.

Nu het bekend is wat een rule engine is kunnen de kenmerken van verschillende rule engines onderzocht worden en daarna met elkaar vergeleken worden. Bij dit onderzoek is de grounded theory methode toegepast. Deze methode is toegepast, omdat de verzamelde data constant met elkaar vergeleken moest worden. Nadat de kenmerken van verscheidene rule engines vergeleken waren kon er geconcludeerd worden dat Drools .NET 3.0, NxBRE en SmartRules het meest geschikt zijn om verder te onderzoeken.

Om de implementatie van de rule engines te kunnen plannen is er gekozen voor de scrum methode. Met de scrum methode is er wekelijks een planning gemaakt, zodat er overzicht gehouden kon worden op de werkzaamheden die gedaan zijn, maar ook op de werkzaamheden die nog moesten volgen. Bij het toepassen en gebruiken van de drie rule engines bleek al snel dat Drools .NET 3.0 een goede rule engine is, maar er geen gebruiksvriendelijke manier bestaat om business rules op te stellen voor deze rule engine. Na de implementatie van NxBRE is het niet gelukt om business rules op te stellen. SmartRules is echter een totaal pakket en komt samen met een rule editor. Deze werken perfect samen en de implementatie van deze rule engine was ook vrij eenvoudig.

SmartRules is dus zeker aan te bevelen als rule engine. Ook de rule editor (SmartRules Studio) die meegeleverd wordt is zeer gebruiksvriendelijk.

## Begrippenlijst

Begrip	Omschrijving
Business rule	Zie 4.1.1 “knowledgebase” voor een concrete uitleg.
Ruleflow	Een flow met acties en rules die uitgevoerd worden door een rule engine.
Ruleset	Een groep business rules.
Business proces	Regels en procedures gehanteerd door het betreffende bedrijf.
Facts	Data, in het werkgeheugen geplaatst door de applicatie.
Rule editor	Grafische omgeving om business rules op te stellen (synoniem aan BRMS).
BRMS	Business Rule Management System



## 1. Inleiding

Complexe software systemen hoeven niet zo complex te zijn.

Sogyo B.V. heeft ruim 15 jaar ervaring en kennis opgedaan op gebied van software ontwikkeling. Het bedrijf heeft ongeveer 75 medewerkers die constant de gelegenheid krijgen zich te blijven ontwikkelen. Sogyo houdt zich onder meer bezig met detachering en development.

Bij een van deze development projecten is een rule engine toegepast. Deze rule engine kan niet meer gebruikt worden en is dus aan vervanging toe.

Het afstudeerproject kreeg zo als doel om de beste vervanger voor deze rule engine te zoeken. Zo zijn er 20 weken ingepland voor onderzoek en implementatie. Tijdens het onderzoek was het van belang om het begrip “rule engine” en de verschillen tussen de diverse .NET rule engines duidelijk te krijgen.

Na een drietal rule engines te hebben geselecteerd die voldoen aan de eisen van de klant is het gebruik van deze rule engines onderzocht door alle drie de rule engines toe te passen op een testapplicatie.

In hoofdstuk 2 is meer informatie te vinden over het bedrijf, in hoofdstuk 3 zijn de details van de opdracht beschreven, in hoofdstuk 4 staat omschreven wat een rule engine is, in hoofdstuk 5 staat een overzicht van de verschillen tussen diverse .NET rule engines, in hoofdstuk 6 wordt het gebruik van de drie geselecteerde rule engines omschreven en in hoofdstuk 7 wordt de conclusie van het afstudeerproject beschreven.

## 2. Het bedrijf

### 2.1 Missie

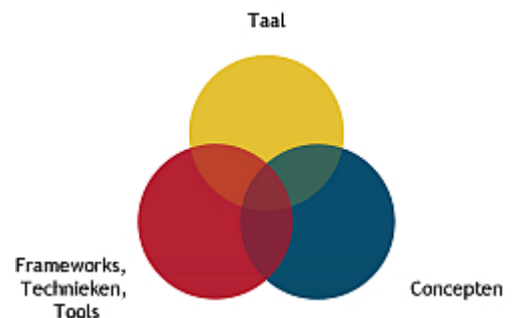
Sogyo vindt het belangrijk dat men het concept achter zowel technieken, talen als frameworks snel en goed doorzien. De Sogyo academy helpt professionals om dit inzicht te bereiken en te vergroten.

Sogyo brengt een balans aan tussen:

- Concepten
- Talen
- Technieken, tools en platformen.

Met deze balans streeft Sogyo naar:

- Beter inzicht in ontwikkelingen in de toekomst.
- Overzicht in wat nu gebeurt en waarom.
- Kennis en kunde om het geleerde snel in de praktijk te brengen.



### 2.2 Activiteiten

Sogyo is een breed IT bedrijf en is in te delen in drie activiteiten.

#### 2.2.1 Detachering & Consultancy

Sogyo biedt ontwikkelaars en software architecten op projectbasis die hun expertise en vaardigheden bij externe bedrijven inzetten. Dit zijn zowel starters als senioren. Sogyo vindt de kwaliteit van haar medewerkers zeer belangrijk en daarom investeert Sogyo in scholing van de medewerkers begeleid door een personal coach.

De consultancyactiviteiten van Sogyo richten zich op het ontwikkelen van IT-organisaties. Senior consultants met ruime ervaring helpen bedrijven met zaken zoals:

- Het inrichten van een ontwikkelingstraat
- Het uitvoeren van code reviews
- Het inrichten en uitvoeren van requirements engineering
- Coaching van developers en software architecten
- Componentenselectie
- Het inrichten van het architectuurproces binnen een organisatie

#### 2.2.2 Development

Sogyo ontwikkelt software op projectbasis. Zij geven hierbij de voorkeur aan intensieve klantgerichte ontwikkeltrajecten, waarin de klant nauw betrokken wordt en krijgt wat er gevraagd wordt. Sogyo hanteert binnen de projecten een domeingedreven ontwikkelaanpak, ondersteund met een agile proces gebaseerd op SCRUM.

Door deze combinatie van structuur en proces binnen onze projecten zijn ze in staat om in korte tijd een overdraagbare oplossing te realiseren.

Deze projecten worden zowel op de boerderij (het kantoor van Sogyo) als op locatie bij de klant uitgevoerd.



### 2.2.3 Sogyo Academy

Starters bij Sogyo stromen in op de Sogyo Academy. Hier worden zij getraind en gecertificeerd om vervolgens met voldoende kennis en inzicht ingezet te kunnen worden op een intern project of gedetacheerd worden bij een klant.

Bij een opleiding horen natuurlijk ook docenten. Alle docenten bij Sogyo zijn werkzaam als software ontwikkelaar/architect en besteden een gedeelte van hun tijd aan het overdragen van hun kennis door het verzorgen van trainingen, seminars, studiegroepen en publicaties.

## 2.3 Markt

Sogyo richt zich op een brede markt en heeft onder meer de volgende klanten:

#### **Finance:**

ABN Amro, Delta Lloyd, Fortis Bank, ING, Nationale Nederlanden, Rabobank

#### **Non profit:**

Agis, Belastingdienst, Politie Amsterdam

#### **Industrie:**

Frieslandcampina, G-Star, KLM, Transavia, Remia

#### **Software producten & diensten:**

Addsup, AG 5, Anva, Brainforce, Casus.com, Chemunity, CHS

#### **Zakelijke dienstverlening:**

ANWB, Hewlett Packard, IBM, Winvision

#### **Media:**

24access Solutions, BNR Nieuwsradio, Telegraaf Media Groep, Tijd Beursmedia

#### **Onderwijs:**

Hogeschool Utrecht, Hogeschool Leiden, Fontys, Universiteit van Amsterdam, Simac Onderwijs

## 3. De opdracht

### 3.1 Probleemstelling

#### 3.1.1 Begin situatie

Sogyo B.V. ontwikkelt en onderhoudt softwaresystemen van diverse klanten. Bij het ontwikkelen en onderhouden van deze systemen is Sogyo constant op zoek om de juiste technieken toe te passen. Eén van deze onderdelen is een business rule engine. Sogyo doet al enige tijd onderzoek naar welke rule engines toepasbaar zijn op de systemen die Sogyo ontwikkelt. Bij dit onderzoek wordt gekeken naar talen, technieken, frameworks en het type probleem dat opgelost kan worden m.b.v. een rule engine. Sogyo heeft deze afstudeeropdracht opgezet om meer inzicht te krijgen in de mogelijkheden m.b.t. rule engines voor het .NET platform.

Dit onderzoek wordt gerelateerd aan een praktijkcase uit één van de lopende projecten binnen Sogyo. Op deze manier wordt het onderzoek concreet en direct toetsbaar.

Eén van de klanten van Sogyo heeft een complex systeem m.b.t. verzekeringen. Dit systeem is ontwikkeld in .NET met behulp van C#. Het systeem maakt gebruik van een rule engine. Deze rule engine regelt onder meer de validatie van data en het updaten van diverse GUI elementen. De rule engine kan door verschillende redenen niet meer gebruikt worden. Er zal nu onderzocht moeten worden of de rule engine vervangen moet worden en indien dit zo is, door welke rule engine moet de oude rule engine dan vervangen worden?

Er is nog niet veel kennis aanwezig m.b.t. .NET rule engines. Afgelopen jaar is er echter wel een afstudeeronderzoek bij Sogyo gedaan naar rule engines in java (zie bron B2). Deze scriptie is wel beschikbaar, maar er staat niet veel bruikbare informatie in voor het toepassen van .NET rule engines. Wel is het een geschikt document om algemene eigenschappen van rule engines te achterhalen.

#### 3.1.2 Voordelen voor de organisatie

Na afloop van het project zal er meer kennis over rule engines aanwezig zijn bij Sogyo B.V. Er zullen documenten geschreven worden waarin uitgelegd wordt wat een rule engine is en hoe deze werkt (zie bijlage I & bijlage II). Ook zullen er verschillende rule engines met elkaar vergeleken worden (zie bijlage III). De verschillende eigenschappen van deze rule engines zullen vastgelegd worden en vervolgens afgewogen worden tegen de wensen van de klant (zie bijlage X). Op deze manier zal Sogyo een overzicht krijgen van de beschikbare .NET rule engine met de daarbij horende eigenschappen.

Met deze bevindingen zal de selectie van onderzochte rule engines gereduceerd worden tot een drietal rule engines die het meest overeenkomen met de wensen van de klant (zie bijlage X).

Deze drie rule engines zullen op een testapplicatie worden toegepast, zodat er een proof-of-concept geleverd wordt. Zo krijgt Sogyo ook ervaring m.b.t. het gebruik van .NET rule engines.

Uiteindelijk zal er na het onderzoek en het toepassen van de rule engines een advies (zie bijlage X) opgesteld worden. Dit advies zal omschrijven wat de beste keuze zou zijn voor het verzekeringen systeem van de klant en waarom dat de beste keuze is.

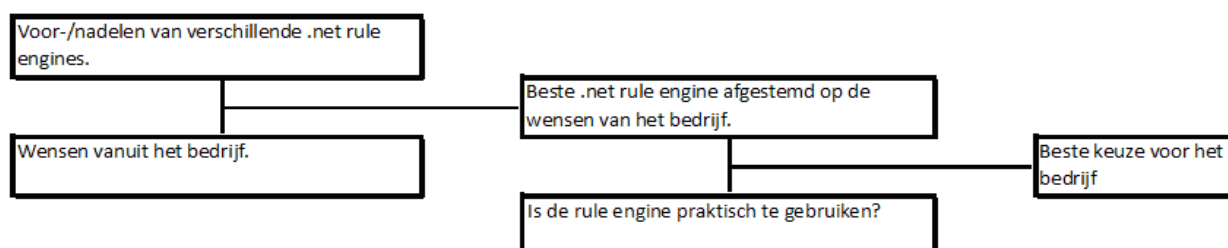
### 3.1.3 Bijdrage aan de bedrijfsdoelstelling

Met dit onderzoek zal Sogyo meer inzicht krijgen in de diverse rule engines met de daarbij horende eigenschappen. Op deze manier krijgt Sogyo een bredere algemene kennis. Zo kunnen zij voortaan overwegen om bij andere projecten rule engines te gaan gebruiken of kunnen ze het gebruik van rule engines adviseren.

## 3.2 Doelstellingen

### 3.2.1 Hoofddoelstelling

Welke rule engine zou er op dit moment het meest geschikt zijn om toe te passen op de Microsoft .NET applicatie van de klant van Sogyo, rekening houdend met de wensen van de klant van Sogyo?



### 3.2.2 Subdoelstellingen

1. Welke voordelen heeft het toepassen van een rule engine binnen een .NET omgeving tot systemen die geen gebruik maken van rule engines maar dit wel zouden kunnen?
2. Wat zijn de verschillende voor- en nadelen van diverse .NET rule engines en welke rule engine zou het beste zijn voor het systeem en welke voor de gebruikers?
3. Wat zijn de minimale eisen en belemmeringen vanuit het bedrijf m.b.t. het integreren van rule engines in het .NET systeem?
4. Welke 3 rule engines van de in subdoelstelling 2 onderzochte rule engines zijn het meest geschikt voor de klant van Sogyo?

### 3.2.3 Bereiken van de doelstellingen

Om tot een antwoord op de doelstellingen te komen zal er eerst uitgezocht worden welke .NET rule engines er beschikbaar zijn. Bij de rule engines van deze longlist zullen de eigenschappen onderzocht worden. Na het vergelijken van de kenmerken zullen de rule engines beoordeeld kunnen worden. Door de drie rule engines met de beste beoordeling te selecteren blijft er een shortlist over. Als deze rule engines in theorie voldoen aan de eisen en wensen van de klant zullen de rule engines toegepast worden in de praktijk. Op basis van het gebruik van de rule engines zal er een advies gegeven kunnen worden. Dit advies zal omschrijven welke onderzochte rule engine het beste gebruikt kan worden voor het verzekeringen systeem en waarom.

#### Opmerkingen:

- Aangezien de doelstelling en subdoelstellingen tijdens het project bijgesteld zijn, wijken deze af van het Plan van Aanpak (zie bijlage IV).
- Voor meer details m.b.t. de opdracht zoals deze voorafgaande het project gedefinieerd is, zie de opdrachtschrijving (zie bijlage V).
- Voor meer details m.b.t. de opdracht zoals deze uiteindelijk gevormd is, zie de Business Case (zie bijlage VI).



### 3.3 Planning

#### 3.3.1 Globale planning

Week:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Omschrijving van de afstudeeropdracht																						
Plan van aanpak																						
Onderzoek: "Wat zijn Rule Engines"																						
Onderzoek: "Mogelijk toe te passen rule engines"																						
Blog: "Wat zijn Rule Engines"																						
Onderzoek: "Wensen vanuit het bedrijf"																						
Toepassen van de rule engine																						
Advies m.b.t. rule engines																						
Presentie "Advies m.b.t. rule engines"																						
Scriptie (Eindverslag)																						
Eindpresentatie Sogyo																						
Eindpresentatie school																						

#### 3.3.2 Uiteindelijke verloop van de planning

Week:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Omschrijving van de afstudeeropdracht																						
Plan van aanpak																						
Onderzoek: "Wat zijn Rule Engines"																						
Onderzoek: "Mogelijk toe te passen rule engines"																						
Blog: "Wat zijn Rule Engines"																						
Onderzoek: "Wensen vanuit het bedrijf"																						
Toepassen van de rule engine																						
Advies m.b.t. rule engines																						
SmartRules Handleiding																						
Business Case																						
Scriptie (Eindverslag)																						
Eindpresentatie maken																						
Eindpresentatie school & Sogyo																						

## 3.4 Werkwijzen en methoden

### 3.4.1 Scrum

De scrum methode is toegepast tijdens de praktijkfase (week 8 t/m 16) van het afstudeerproject (zie 3.3). Door het toepassen van scrum is het mogelijk om per week een planning te maken. Dit is een groot voordeel op het moment dat een project veel onverwachte elementen bevat. Ook is het gunstig om de scrum methode toe te passen om overzicht op de globale planning (zie 3.3.1) te behouden.

Door het feit het team slechts uit één persoon bestaat zijn er een aantal onderdelen overbodig.

Dit houdt in dat tijdens de sprint meetings werd afgesproken welke taken er uitgevoerd zouden gaan worden, maar dat de prioriteit door het scrumteam bepaald werd. Daarnaast zijn de daily standup meetings komen te vervallen.

Wel is er elke week een sprint meeting gehouden. Tijdens deze meeting zijn de product backlog items voor de komende print besproken. Deze product backlog items zijn vervolgens op het project scrumboard geplaatst zodat er een duidelijk overzicht in de werkzaamheden kwam.

Begrippen m.b.t. de scrum methode zijn terug te vinden in het document "Scrum begrippen" (zie bijlage VII).

### 3.4.2 Deskresearch

De deskresearch onderzoeksmethode is toegepast bij het onderzoek "Wat is een Rule Engine?" (zie bijlage I & bijlage II). Het onderzoek gaat in op bronnen geproduceerd door specialisten en gebruikers. Verder heeft het onderzoek niet de praktische toepassing van een rule engine nodig om tot conclusies te komen. Aangezien de deskresearch onderzoeksmethode de volgende kenmerken heeft, is deze geschikt voor een dergelijk onderzoek.

Kenmerken:

1. Bestaand materiaal wordt gebruikt.
2. Er is geen direct contact met het onderzoeksobject.
3. Het materiaal wordt vanuit een ander perspectief gebruik dan waarmee het werd geproduceerd.

De onderzoeksmethode is toegepast door eerst de benodigde informatie te zoeken bij de deelvragen van het onderzoek m.b.v. het internet. Deze informatie is vervolgens samengevoegd. De overbodige informatie is vervolgens verwijderd. Vervolgens zijn er met de beschikbare informatie conclusies getrokken bij de deelvragen van het onderzoek. Om een antwoord te krijgen op de hoofdvraag is er een conclusie getrokken uit de conclusies van de deelvragen.

Bij gebruik van de deskresearch onderzoeksmethode worden er twee documenten geschreven. Deze documenten zijn een werkplan en een onderzoeksverslag.

Het werkplan bevat de volgende elementen:

- Situatieschets
- Doelstelling
- Onderzoeksmodel
- Deelvragen
- Onderzoeksstrategie
- Planning

Het onderzoeksverslag bevat de volgende elementen:

- Projectkader
- De uitgewerkte deelvragen
- Conclusie m.b.t. de hoofdvraag

### 3.4.3 Grounded Theory

De Grounded theory onderzoeksmethode is toegepast op het onderzoek “.NET Rule Engine vergelijking” (zie bijlage III). Het onderzoek is gericht op het zoeken en vergelijken van gegevens. Aangezien de grounded theory onderzoeksmethode de volgende kenmerken heeft, is deze geschikt voor een dergelijk onderzoek.

Kenmerken:

1. Een zorgvuldige en consequente toepassing van gegeven procedures en technieken.
2. Een zoekende houding van de onderzoeker.
3. Een voortdurend onderling en met elkaar vergelijken van empirische gegevens en theoretische concepten.

De onderzoeksmethode is toegepast door eerst na te gaan welke .NET rule engines er beschikbaar zijn. Deze diverse .NET rule engines zijn de onderzoeksobjecten. Vervolgens worden er onderzoeksaspecten opgesteld die bij alle onderzoeksobjecten onderzocht gaan worden. Deze onderzoeksaspecten kunnen nog veranderen tijdens het onderzoek. Alle benodigde informatie zal nu gezocht worden bij alle onderzoeksaspecten van elk onderzoeksobject. Op deze manier kunnen de aspecten van alle onderzoeksobjecten vergeleken worden met elkaar. Uit deze vergelijkingen kunnen conclusies getrokken worden.

Bij gebruik van de deskresearch onderzoeksmethode kan alles in één verslag worden gedocumenteerd. Dit verslag bevat de volgende elementen:

- Voorbereiding
- Data Collection (alle verzamelde data gesorteerd per onderzoeksobject)
- Analyse
- Conclusie



## 4. Onderzoek: Wat zijn Rule Engines?

### 4.1 Proces van een rule engine

Een rule engine bestaat uit twee delen en maakt gebruik data uit het werkgeheugen.

#### 4.1.1 Knowledgebase

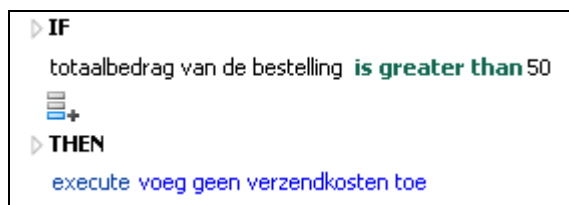
Een knowledgebase (ook wel een business rule repository genoemd) is een centrale omgeving waar business rules in zijn gedefinieerd. Business rules zijn digitaal vertaalde bedrijfsregels. Een bedrijfsregel zou bijvoorbeeld kunnen zijn “bij elke bestelling van 50 euro of meer worden er geen verzendkosten berekend”. Business rules bestaan altijd minimaal uit twee onderdelen.

Het eerste onderdeel is een vergelijking tussen een vooraf ingestelde waarde en data uit het werkgeheugen (fact). In het vorige voorbeeld is de vooraf ingestelde waarde “50”. De data uit het werkgeheugen is het totaalbedrag van de bestelling. Ook moet hier worden aangegeven hoe de waarden met elkaar vergeleken worden. Dit gebeurt in de condition. De condition is in dit geval “is groter dan”. Aangezien dit een bewering is waaraan voldaan moet worden kunnen we “als” voor de bewering toevoegen en “dan” na de bewering. Zo krijgen we dus: als “het totaalbedrag van de bestelling” + “is groter dan” + “50” dan ...

Het tweede deel van een business rule is de actie die uitgevoerd moet worden op het moment dat er aan de vergelijking voldaan wordt. Als we op het vorige voorbeeld ingaan zou de actie zijn “voeg geen verzendkosten toe”.

De business rule wordt compleet als de actie wordt toegevoegd:

als “het totaalbedrag van de bestelling” + “is groter dan” + “50” dan “voeg geen verzendkosten toe”.



In de knowledgebase kan prioriteit aan rules worden toegekend, zodat er een volgorde ontstaat waarin de business rules worden uitgevoerd.

#### 4.1.2 Execution core

De execution core is het onderdeel van de rule engine waar de business rules opgehaald worden uit de knowledgebase en toegepast worden op de data die door de applicatie in het werkgeheugen geplaatst is. De business rules kunnen op 3 verschillende manieren worden toegepast:

#### Inference Forward Chaining Engine

De inference forward chaining engine is de meest gebruikte manier om rules toe te passen op facts. De inference forward chaining engine doorloopt alle business rules uit een ruleset die van toepassing zijn op de data in het werkgeheugen. Op het moment dat een van de rules de data verandert zal de engine de ruleset opnieuw gaan doorlopen. De engine houdt bij welke rules er geactiveerd zijn en zal bij het nogmaals doorlopen niet meer de rules activeren die al toegepast zijn. De engine zal op deze manier door blijven gaan tot alle rules geactiveerd zijn of alle rules doorlopen zijn zonder dat de data aangepast is.

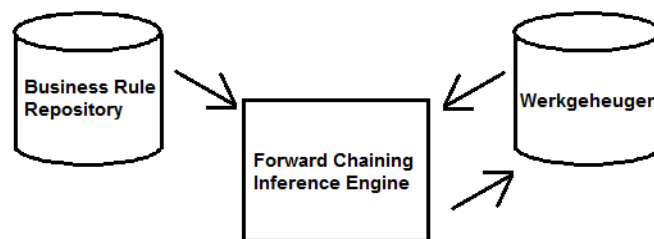
### Inference Backward Chaining Engine

De inference backward chaining engine wordt gebruikt voor het verkrijgen van antwoorden op vragen die gesteld worden aan de engine.

De engine maakt ook hier gebruik van facts. De engine doet niks met de facts totdat er een specifieke vraag aan de engine gesteld wordt. De engine probeert nu een antwoord op de vraag te vinden door alle rules die betrekking hebben op de vraag te doorlopen. Dit levert nieuwe facts op. Met deze facts kan de rule engine steeds andere rules doorlopen en zal bij een geldige vraag uiteindelijk bij een antwoord kunnen komen. Tijdens dit hele proces wordt er geen data gewijzigd. Ook alle acties van de rules zullen niet worden uitgevoerd.

### Flow Engine

De flow engine is een engine die gebruikt wordt voor voorspellingen en detecteren van facts. Door facts constant met business rules te vergelijken kan de flow engine patronen ontdekken en nieuwe facts creëren. Hierbij kan bijvoorbeeld gedacht worden aan het achterhalen van een familiestamboom. Zo kan de flow engine op het moment dat twee jongens dezelfde ouders hebben concluderen dat deze twee jongens broers van elkaar zijn. De flow engine kan hier een fact van maken en deze weer gebruiken bij diverse business rules die vervolgens ook weer facts opleveren.



## 4.2 Hulp van een rule engine

### 4.2.1 De voordelen

Met behulp van rule engines kan de logica van een applicatie gescheiden worden van de applicatie zelf. De functies blijven door de applicatie zelf uitgevoerd worden, maar berekeningen, validatie en toepassen van logica afkomstig van bedrijfsregels kunnen door de rule engine overgenomen worden. Op deze manier worden alle statische gegevens die benodigd zijn voor deze acties vastgelegd op een centrale plaats. Onderhoud wordt op deze manier overzichtelijker en kan bovendien gedaan worden door niet-technisch personeel. De code van de applicatie hoeft nu ook niet meer aangepast te worden als er een bedrijfsregel verandert.

Bij goed gebruik van de rule engine kunnen er zeer complexe problemen opgelost worden. De rule engine zal altijd een keuze kunnen maken en een oplossing geven.

#### 4.2.2 Rule engines als oplossing

In de Drools .NET documentatie wordt antwoord gegeven op de vraag “wanneer een rule engine toe te passen?”. Dit antwoord luidt: “*Wanneer er geen oplossing gevonden kan worden m.b.v. traditioneel programmeren*” (zie bron B1).

Er zijn verschillende redenen waardoor er geen goede oplossing gevonden kan worden, zoals:

- Als er geen stabiele oplossing gemaakt kan worden voor het probleem.
- Het probleem heeft geen logische algoritmes.
- Het probleem wordt niet volledig begrepen door de programmeur.
- De logica van de applicatie verandert vaak.
- De business analisten zijn beschikbaar, maar zijn niet technisch.

Een rule engine toepassen op een applicatie kan ook nadelig zijn. Op het moment dat het beheer en onderhoud van een applicatie erg gemakkelijk is, is het geen oplossing om een rule engine te gaan gebruiken. Omgang met een rule engine duurt even om te leren. Ook zou een rule engine voor vertraging van het systeem kunnen zorgen.

#### 4.3 No Silver Bullet

Het concept achter de “No Silver Bullet” theorie is gebaseerd op een vergelijking tussen weerwolven en software.

Weerwolven zijn normaal gesproken doodgewone mensen en functioneren normaal binnen de maatschappij. Eens in de maand echter veranderen deze doodgewone mensen in monsters en kunnen tot rust gebracht worden met één algemeen voorwerp, zilveren kogels.

Dit wordt vergeleken met software die normaal gesproken goed functioneert en veel voordelen kan opleveren, maar deze software zou zomaar kunnen veranderen in een systeem dat veel ellende bezorgt. In dit geval is het niet altijd mogelijk om één zilveren kogel (een rule engine) toe te passen en daar de problemen mee op te lossen. Rule engines kunnen dus niet voor alle problemen met software ingezet worden.

(Zie bron W2 & bron W3)

#### 4.4 Rule engines en Artificial Intelligence

Bij rule engines wordt vaak direct aan Artificial Intelligence (AI) gedacht. AI is een zelflerend systeem dat op de manier effectiever kan functioneren. Een rule engine is echter niet per definitie AI. Een rule engine is alleen AI als er gebruik wordt gemaakt van een RETE algoritme (zie bron W1). Dit heeft met het feit te maken dat het RETE algoritme eerder berekende waardes onthoudt.

Op het moment dat het RETE algoritme facts doorloopt, onthoudt de rule engine business rules die gedeeltelijk overeenkomen. Op deze manier weet de rule engine de eerst volgende keer dat de ruleset doorlopen wordt of de business rule in aanmerking kan komen om geactiveerd te worden.



## 5. Onderzoek: De verschillen in .NET Rule Engines

### 5.1 Algemene resultaten

Dit onderzoek richt zich op de verschillen tussen de diverse beschikbare .NET rule engines. Bij dit onderzoek is het uitzoeken welke .NET rule engines er beschikbaar zijn de eerste stap geweest. De tweede stap was het opstellen van verschillende onderzoeksaspecten per rule engine. Om deze onderzoeksaspecten uit te zoeken was het van belang om veel informatie te verzamelen m.b.t. deze onderzoeksaspecten.

Deze informatie is gesorteerd gedocumenteerd. M.b.v. deze gegevens is het volgende overzicht gemaakt.

	NxBRE	.NET Drools	InRule	RuleLab .NET BRE	Simple Rule Engine	FlexRule	Windows Workflow Foundation	BizTalk Server 2010
Licence	LGPL	Freeware	Betaald	Betaald	LGPL		Freeware	Betaald
Free (reliable) Support	+	-			-		Forums	
Inference Forward Chaining	+	+		+	+	+	+	+
Inference Backward Chaining	+	+		-	-	+	-	-
Flow engine	+	-		-	-	+	+	-
Rete Algorithm		+				+	-	+
Timer for firing rules		+				+		+
Debugging rules	+	+					+	
Documentatie	+	+			-		+	+
Dynamic Rules	+	+		+			+	+
Rule editor	+	+		+		+	+	+
Totaalscore	8	8	???	3	1	6	6	6

#### Opmerkingen

- De grijze vakken geven aan dat er geen informatie over het betreffende element is gevonden.
- De gegevens van InRule zijn tot het moment van betalen afgeschermd.
- FlexRule is een rule engine die nog niet is vrijgegeven. De release date is nog niet bekend.

## 5.2 Rule engine selectie

M.b.v. het overzicht (zie 5.1 “Algemene resultaten”) kan geconcludeerd worden dat NxBRE en Drools .NET 3.0 de rule engines zijn met de meeste mogelijkheden. Aangezien het project gericht is op het onderzoeken van drie rule engines, zal er nog één rule engine meer uitgezocht moeten worden. De drie geselecteerde rule engines zullen vervolgens op een testapplicatie worden toegepast (zie hoofdstuk 6 “Toepassen van .NET rule engines”) voor een praktijk onderzoek.

Voor het selecteren van de derde rule engine is er gekeken naar de één na hoogste score. De één na hoogste score bedraagt een 6. Dit betekent dat er één rule engine gekozen moet worden uit een selectie van drie rule engines.

Op dit moment worden de nadelen van de verschillende rule engines vergeleken. FlexRule heeft het grote nadeel dat deze nog niet beschikbaar is en de release datum nog niet is vrij gegeven. Het nadeel van de BizzTalk Server 2010 rule engine is het feit dat deze niet gratis is.

Op het moment dat deze drie rule engines vergeleken worden met de wensen van de klant (zie bijlage IX) kan er vastgesteld worden dat alle drie de rule engines aan deze eisen en wensen voldoen.

Een volgende vergelijking wordt gedaan tussen de informatie op de websites van, opmerkingen op forums en de eigenschappen van de betreffende rule engines. Uit deze vergelijking zien we dat het om drie verschillende types rule engines gaat.

.NET Drools is een gebruiksvriendelijke rule engine.

NxBRE is een zeer complexe en technische rule engine.

Windows Workflow Foundation is een open source rule engine.

## 5.3 Wijziging

Na het toepassen van Drools .NET 3.0 en NxBRE bleek het dat er nog een .NET rule engine bestond die niet meegenomen was in het onderzoek. SmartRules is de naam van deze rule engine. Aangezien SmartRules ook een rule editor (SmartRules Studio genaamd) heeft, was het zeer aantrekkelijk om als nog de gegevens van SmartRules te vergelijken met de gegevens van de andere rule engines.

	SmartRules
Licence	Freeware
Free (reliable) Support	+
Inference Forward Chaining	+
Inference Backward Chaining	-
Flow engine	+
Rete Algorithm	+
Timer for firing rules	+
Debugging rules	+
Documentatie	+
Dynamic Rules	+
Rule editor	+
Totaalscore	9

Het blijkt dus dat deze rule engine het hoogste scoort van alle vergeleken rule engines. Op dit moment is er besloten om Windows Workflow Foundation te vervangen door SmartRules.

Wat SmartRules aantrekkelijker maakt dan Windows Workflow Foundation is het feit dat SmartRules een totaal pakket is. De rule editor is afgestemd op de rule engine en zal dus niet veel problemen op leveren bij het uitvoeren van rules die zijn opgesteld m.b.v. SmartRules Studio.

## 6. Toepassen van .NET Rule Engines

Dit hoofdstuk bevat praktische bevindingen m.b.t. het gebruik van de geïmplementeerde rule engines. Het praktische onderzoek bestond uit het implementeren van de rule engine en vervolgens het gebruiken van de rule engine. Het gebruik van de rule engine bestaat voornamelijk uit het schrijven van business rules en het wegschrijven van de data naar het werkgeheugen. Bij het uitvoeren van deze twee onderdelen zullen er een aantal bevindingen worden. Deze bevindingen leiden tot een conclusie m.b.t. het aanbevelen van een specifieke rule engine.

### 6.1 Planning

Het traject om de rule engines te implementeren (week 8 t/m 16) wordt ondersteund door de scrum methode. Van de sprints is een agenda bijgehouden:

Prioriteit	Waarde	Correctie	Omschrijving	Done	Opmerking
<b>Sprint 1: Drools .NET</b>					
1	2	8	Drools .Net 3.0 downloaden	+	Moeilijk te vinden
7	3	3	Drools .NET tutorial downloaden	+	
8	5	10	Tutorial Drools .NET uitvoeren	+	Configuratie fouten
2	12	10	Eigen mini applicatie maken om de rule engine te testen	+	Was zeer snel gebeurd (geen documentatie/unittesten)
9	8	8	Drools loskoppelen van de tutorial software	+	
3	20	40	Drools koppelen aan de mini applicatie	+	Veel onbekende fouten die zeer moeilijk op te lossen waren.
4	8	10	Testen	+	Testen was in combinatie met Drools + Mini applicatie
6	5	5	Rules toevoegen/wijzigen/verwijderen.	+	
5	12		De toepassing van Drools .NET evalueren		Helaas niet meer gelukt binnen deze sprint
<b>Sprint 2: Drools .NET Uitbreiding</b>					
5	5	5	Huidige (console) applicatie opschonen en inchecken bij SVN	+	Afkomstig van sprint 1
1	25	20	Huidige (console) applicatie uitbreiden naar Windows Forms	+	
2	40	50	Graphische rule editor bij Drools betrekken		Er kon nog geen goede BRMS worden toegepast, dit wordt uitgesteld.
3	20	20	De toepassing van Drools .NET evalueren (advies)		Niet alle aspecten van Drools zijn volledig uitgewerkt.
4	5	5	Demo Drools .NET geven	+	
<b>Sprint 3: Losse projecten &amp; NxBRE Toepassen</b>					
4	20	20	Drools Win Form project splitsen voor rule engine interface	+	
1	5	5	NxBRE Downloaden	+	
2	40	20	NxBRE Toepassen	+	
3	5	5	Rules toevoegen/wijzigen/verwijderen		
5	5	5	Unittest scenario's voor de rule engine app		
6	15	15	Aanpassingen NxBRE doorvoeren		Eventuele aanpassingen n.a.v. prioriteit 3 of 5
7	5	5	Demo NxBRE		
<b>Sprint 4: RuleML Rules voor NxBRE</b>					
4	15	10	Zoeken naar een RuleML editor	+	
1	5	5	Opstellen van RuleML rules	+	
2	25	25	De RuleML rule werkend krijgen in NxBRE		
3	10	10	NxBRE toepassen op de rule engine applicatie		
5	5	5	Demo		
<b>Sprint 5: SmartRules</b>					
4	25	15	Onderzoek naar SmartRules mogelijkheden	+	
1	50	30	Rules opstellen	+	
2	20	15	Rule Engine implementeren	+	
3	5	5	Rules testen	+	
5	5	5	Demo SmartRules	+	
<b>Sprint 6: Werken met SmartRules</b>					
1	10	10	GUI updaten via SmartRules	+	
2	15	70	Data Collectie doorlopen met SmartRules		Om dit werkend te krijgen moest ik met adapters gaan werken.
5	15	15	Testscenario's toevoegen		Door de adapters ben ik niet meer aan dit onderdeel toegekomen.
3	10	10	Unittests opstellen		Door de adapters ben ik niet meer aan dit onderdeel toegekomen.
4	20	20	Unittests realiseren		Door de adapters ben ik niet meer aan dit onderdeel toegekomen.
6	25	15	Verslaglegging SmartRules		Door de adapters heb ik dit onderdeel helaas niet af kunnen ronden.
<b>Sprint 7: Testing</b>					
1	15	60	Data Collectie doorlopen met SmartRules	+	
4	15	15	Testscenario's toevoegen	+	
2	10	5	Unittests opstellen	+	
3	20	25	Unittests realiseren	+	
5	25	25	Verslaglegging SmartRules		
<b>Sprint 8: Afronden mini app + Proof-of-Concept op Columbus</b>					
3	20	15	Drools repareren	+	
1	20	50	Columbus project werkend krijgen	+	Ontbrekende Files en slechte references
2	30	30	SmartRules toepassen op Columbus + 1 test rule	+	
4	25	25	Verslaglegging SmartRules		
<b>Sprint 9: Proof-of-Concept op Columbus: Designing the Rules and Adapters</b>					
3	20	15	CalculateRulesTest Analyseren	+	
2	15	20	InputAdapter aanpassen n.a.v. de te verzenden gegevens	+	
1	40	40	Rules uitwerken in SmartRules		
4	25	25	Verslaglegging SmartRules		

Zoals het overzicht hierboven aangeeft is het regelmatig voorgekomen dat een product backlog item niet afgerond kon worden. Door in elke sprint de meest belangrijke product backlog items uit te voeren zijn de meest essentiële taken afgerond voor het einde van de sprint. Hierdoor liep het doel van het project weinig risico.

## 6.2 Drools .NET

### 6.2.1 Implementatie

De implementatie van Drools .NET 3.0 is vrij eenvoudig en er zijn slechts een aantal regels code nodig. Deze code is opgesteld m.b.v. de Drools .NET 3.0 handleiding (zie bron B1 & bron W5).

-----  
Implementatie code van Drools .NET 3.0.  
-----

```
private RuleBase ruleBase = null;
public Drools() //Constructor
{
    PackageBuilder builder = new PackageBuilder();
    Stream stream = Assembly.GetExecutingAssembly().GetManifestResourceStream("namespaceNaam.rulesetNaam.drl");
    builder.AddPackageFromDrl("namespaceNaam.rulesetNaam.drl", stream);
    Package pkg = builder.GetPackage();
    this.ruleBase = RuleBaseFactory.NewRuleBase();
    ruleBase.AddPackage(pkg);
}

public void ExecuteRules(IList objectList)
{
    WorkingMemory workingMemory = this.ruleBase.NewWorkingMemory();

    //Put the objects in the workingMemory
    foreach (Object object in objectList)
    {
        workingMemory.assertObject(object);
    }
    workingMemory.fireAllRules();
}
```

-----

Deze code is afhankelijk van 5 .dll bestanden:

1. drools-3.0.dll
2. drool-dep.dll
3. drools.dotnet.dll
4. IKVM.GNU.Classpath.dll
5. IKVM.Runtime.dll

Bij de implementatie van Drools .NET moet er echter wel rekening gehouden worden met een aantal aspecten (zie 6.1.3).

### 6.2.2 Werken met Drools .NET

Na de implementatie van Drools kunnen de business rules opgesteld worden. Drools .NET is gebaseerd op Drools voor java. Voor Drools voor java is er een Business Rule Management System genaamd "Guvnor" beschikbaar. Helaas werkt deze niet om business rules op te stellen voor Drools .NET. Dit betekent dat de business rules handmatig opgesteld moeten worden.

Een business rule voor Drools .NET ziet er ongeveer zo uit:

```
package RuleEngineApp_Drools

rule "Spoed"
    when
        order : RuleEngineApp_Domain.Order( Spoed == true )
    then
        RuleEngineApp_GUI.Form1.UpdateResultLog("Spoed: Ja");
    end
```

Deze business rule controleert of bij een bepaalde bestelling spoed vereist is.

Eerst wordt de packagenaam (namespace) aan de rule engine doorgegeven, zodat deze methoden uit de betreffende namespace kan uitvoeren (houd hierbij rekening met: 6.1.3 Static methoden).

Dan wordt de naam van de rule geregistreerd zodat deze eventueel door een andere rule in dezelfde ruleset aangeroepen kan worden. Daarna wordt er aangegeven dat het om het type object "order" gaat en dat de eigenschap "Spoed" uit de classe "Order" de waarde "true" moet bevatten om de actie uit te kunnen voeren.

Als vervolgens de actie uitgevoerd mag worden wordt er naar een namespace, class en methode verwezen. Deze methode krijgt in dit geval de parameter "Spoed: Ja" mee.

Ook kan er worden aangegeven in een business rule dat er vanuit de rule engine data gemuteerd moet worden. Dit kan met behulp van de modify methode.



### 6.2.3 Bevindingen

Bij het implementeren en werken met Drools .NET zijn er een aantal aspecten geconstateerd. Deze aspecten zijn belangrijk om te weten bij de implementatie of het werken met Drools .NET. Sommige van deze aspecten zouden eventueel ook verbeterpunten van de rule engine kunnen zijn.

#### Static methoden

Tijdens het werken met Drools .NET traden er fouten op bij het uitvoeren van de business rules. Bij de eerste implementatie was de rule engine toegepast op een console applicatie. Hier traden de fouten niet op. Vervolgens was het de bedoeling om de rule engine toe te passen op een Windows forms applicatie. Dit gaf echter problemen. De rule engine geeft niet duidelijk aan wat het probleem is. Er kon alleen geconstateerd worden dat er een onbekende fout optrad in de ruleset. Na veel onderzoek bleek het dat er vanuit de ruleset een methode werd aangeroepen die bij de console applicatie static was en bij de Windows forms applicatie niet meer. Er moeten dus static methods worden aangeroepen vanuit Drools .NET 3.0.

#### Embedded ruleset

Om de ruleset te kunnen gebruiken zal de ruleset file (.drl) moeten worden toegevoegd aan het project. Na het toevoegen van dit bestand moet de "Built action" eigenschap van de ruleset file als "Embedded Resource" ingesteld worden.

#### Beschikbaarheid

Het downloaden van Drools .NET 3.0 heeft wat problemen opgeleverd. Er zijn een aantal download links die aangeven dat het Drools .NET 3.0 is, maar in feite is het Drools .NET 2.5. Dit is ook de versie die op veel sites staat gedefinieerd als de laatste release. Drools .NET 3.0 is te vinden via de "downloads" link van [codehaus.org](http://codehaus.org) (zie bron W4).

## 6.3 NxBRE 3.2.0

### 6.3.1 Implementatie

De implementatie van NxBRE is **niet** eenvoudig.

Deze code is opgesteld m.b.v. de NxBRE handleiding (zie bron B3 & Bron W6).

Implementatie code van NxBRE

```
-----
IInferenceEngine ie = new IEImpl();

//For using MS Visio
/*
String ruleFile = "../../.vdx";
ie.LoadRuleBase(new Visio2003Adapter(ruleFile, FileAccess.Read));
*/

//For using .HRF
/*
String ruleFile = "../../gedcom-relations.hrf";
Stream ruleFileStream = new FileStream(ruleFile, FileMode.Open);
IRuleBaseAdapter rba = new NxBRE.InferenceEngine.IO.HRF086Adapter(ruleFileStream,
FileAccess.Read);
ie.LoadRuleBase(rba);
*/

//For using RuleML
String ruleFile = "../../TestRuleML.ruleml";
ie.LoadRuleBase(new RuleML09NafDataLogAdapter(ruleFile, FileAccess.Read));
ie.Process();
System.Console.WriteLine("Rules Fired!");

//Retreiving facts
Fact f = ie.GetFact("discount5");
List<Object> pred = new List<object>();
try
{
    pred = (List<Object>)f.PredicateValues.ToList();
}
catch(Exception e)
{
    System.Console.WriteLine("0 Facts retrieved from Working Memory!");
}
-----
```

Deze code is afhankelijk van slechts één .dll bestand. Dit .dll bestand is "NxBRE.dll".

De implementatie is niet eenvoudig door het feit dat NxBRE zeer foutgevoelig is. Er zitten zelfs fouten in voorbeeld rulesets waardoor er fouten optreden in de rule engine. Deze fouten zijn meestal niet goed omschreven en onduidelijk.

### 6.3.2 werken met NxBRE

NxBRE heeft geen eigen Business Rule Management System. Het wordt op deze manier lastig om de business rules correct op te stellen. NxBRE ondersteund 3 soorten ruleset bestanden.

1. .HRF (Human Readable File)
2. RuleML (Afgeleide van XML voor business rules)
3. .vcd (MS visio)

NxBRE levert voorbeelden met .hrf bestanden. Deze voorbeelden werken niet. Er treden fouten op tijdens het uitvoeren van de applicatie. Ook is de syntax van de .hrf bestanden niet te achterhalen met de voorbeelden.

De demo die gebruikt maakt van RuleML bestanden geeft ook fouten bij het uitvoeren. Er zijn verschillende stand-alone RuleML editors (zoals Translator(zie bron B4)). Ook m.b.v. deze editors is het niet gelukt om een correcte RuleML file te creëren die de rule engine kan gebruiken.

Na de poging om deze twee bestandstype toe te passen (ruim 1 week) bleef er geen tijd meer over in de planning om nog verder in te gaan op het opstellen van business rules voor NxBRE.

Aangezien het nu duidelijk werd dat NxBRE geen gebruiksvriendelijke rule engine is en dus niet meer aan de wensen voldoet, is er gekozen om niet verder te gaan met NxBRE.

## 6.4 SmartRules

### 6.4.1 Implementatie

De implementatie van de SmartRules rule engine is uitgebreider dan de implementatie van Drools .NET of NxBRE. De code is echter wel overzichtelijk en goed te begrijpen. De volgende code is opgesteld m.b.v. SmartRules documentatie (zie bron W7).

De code m.b.t. de SmartRules Rule Engine implementatie.

```
public class SmartRulesEngine : RuleEngineApp_RuleEngineInterface.IRuleEngine
{
    private IRuleEngineSession session;

    public SmartRulesEngine()
    {
        this.session = SmartRulesFactory.GetRuleEngineSession();
    }

    public void ExecuteRules(IList objectList)
    {
        try
        {
            IList facts = new ArrayList();
            foreach (Object o in objectList)
            {
                facts.Add(o);
            }

```

Gebruik deze code voor activatie van een ruleflow

```
facts = this.session.InvokeRuleflow(
"SmartRulesStudioProjectName", "nameOfTheRuleflow", facts);
```

Gebruik deze code voor activatie van een ruleset

```
facts = this.session.InvokeRuleset(
"SmartRulesStudioProjectName", "nameOfTheRuleset", facts);
```

```
    }
    catch (Exception ex)
    {
        System.Console.WriteLine(ex.Message);
    }
}
```

De code is afhankelijk van slechts één .dll bestand. Dit .dll bestand is "SmartRules.RuleEngine".

Bij de implementatie van SmartRules moet rekening gehouden worden met een waarde die handmatig aan de settings moet worden toegevoegd. Deze waarde is de "Properties.Settings.Default.ProjectPath" waarde en moet verwijzen naar de locatie van het SmartRules Studio project. Deze waarde wordt gebruikt voor het verkrijgen van de rulesets.

Het document "SmartRules" (zie bijlage VIII) geeft een gedetailleerde beschrijving van SmartRules en SmartRules Studio.

### 6.4.2 Werken met SmartRules

SmartRules heeft een eigen Business Rule Management System. Dit maakt het een stuk gemakkelijker om rules op te stellen.

Het opstellen van de business rules kan in de grafische omgeving van SmartRules Studio. SmartRules Studio heeft de mogelijkheid om .NET assemblies in te laden. Met deze assemblies wordt het mogelijk om bij het opstellen van de business rules direct de methoden van de geladen assemblies te gebruiken.

In SmartRules Studio ziet dit er als volgt uit:

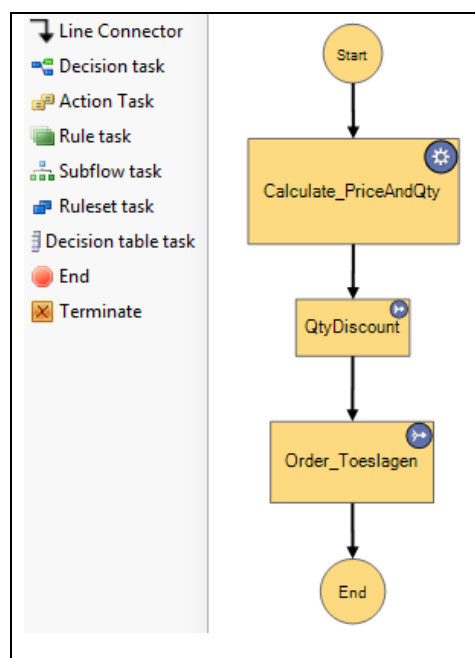
```

> IF
  InputAdapter.GetOrderSpoed is equal to true
  +
  > THEN
    execute InputAdapter.SetOrderPrice ' InputAdapter.GetOrderPrice + highPriorityOrderCosts '
    execute OutputAdapter.sendMsgToResultLog ' highPriorityOrderCostsMsg '

```

De business rules worden in SmartRules Studio dus opgesteld op een manier die iedereen grotendeels kan lezen. De mate waarin de business rules leesbaar zijn is ook zeer sterk afhankelijk van de naamgeving van de methoden en classes.

In SmartRules Studio is het niet alleen mogelijk om rulesets op te stellen, maar ook ruleflows. Een ruleflow is een flow van acties, beslissingen en business rules. Ook kunnen vanuit de ruleflow rulesets worden geactiveerd.



Op het moment dat de rulesets en ruleflows zijn opgesteld biedt SmartRules Studio de mogelijkheid om deze rulesets en ruleflows te compileren. De rule engine kan de nieuwe bestanden nu inlezen zonder te herstarten.

Het document "SmartRules" (zie bijlage VIII) geeft een gedetailleerde beschrijving van SmartRules en SmartRules Studio.

### 6.4.3 Bevindingen

Bij het implementeren en werken met SmartRules en het gebruik van SmartRules Studio zijn er een aantal aspecten geconstateerd. Deze aspecten zijn belangrijk om te weten bij de implementatie van de SmartRules rule engine of het werken met SmartRules Studio. Sommige van deze aspecten zouden eventueel ook verbeterpunten van de rule engine of de rule editor kunnen zijn.

#### Inladen van assemblies

Het laden van .NET assemblies in SmartRules Studio is zeer fout gevoelig. Het is daarom verstandig om de assemblies te baseren op een ClassLibrary project.

#### Herladen van assemblies

Het herladen van assemblies kan problemen opleveren. Een oplossing die bijna altijd werkt is het verwijderen en opnieuw toevoegen van de assembly. Om een assembly te verwijderen moeten wel eerst alle verwijzingen naar deze assembly uit het project worden verwijderd.

#### Afhankelijke classes

SmartRules Studio 2010.08 heeft geen mogelijkheid tot het inlezen van datacollecties en deze data vervolgens te gebruiken.

Bijvoorbeeld:

Een simpel order systeem is opgebouwd uit:

Order:

- date (DateTime)
- orderlines (OrderLine[])

OrderLine

- qty (int)
- product (Product)

Product

- price (double)
- ProductDescription (String)

Het is nu niet mogelijk om via de Order een OrderLine uit te lezen en de gegevens van een Product te achterhalen. Om dit op te lossen zal er een methode gemaakt moeten worden die een index als parameter heeft en deze zal gebruiken om de product informatie te achterhalen.

#### Subversion

Aangezien er fouten op kunnen treden binnen een SmartRules Studio project die ervoor zorgen dat er heel moeilijk op te lossen fouten ontstaan, is het aan te raden om consequent gebruik te maken van een versiebeheer tool.

Het document "SmartRules" (zie bijlage VIII) geeft een gedetailleerde beschrijving van SmartRules en SmartRules Studio.

## 6.5 Resultaat van de implementaties

Na alle drie de rule engines te hebben geïmplementeerd kan het volgende worden geconcludeerd.

### 6.5.1 Drools .NET 3.0

Drools .NET is een vrij eenvoudig te implementeren rule engine. Het gebruik van Drools .NET is echter niet zo eenvoudig. Drools .NET heeft geen eigen business rule editor. Hierdoor wordt het lastig om business rules op te stellen. Ook wordt er zo voorbij gegaan aan het doel dat mensen zonder technische kennis de business rules kunnen opstellen.

Ook blijkt Drools .NET geen (dynamic rules) te hebben wat inhoudt dat de applicatie steeds opnieuw compiled moet worden op het moment dat de ruleset gewijzigd wordt.

### 6.5.2 NxBRE

NxBRE is moeilijk te implementeren. NxBRE heeft ook geen business rule editor. Ondanks het feit dat NxBRE met drie business rule file formaten om kan gaan is het schrijven van de business rules erg moeilijk. Dit maakt NxBRE een zeer gebruiksonvriendelijke rule engine. Ook lijkt het er op dat NxBRE niet verder ontwikkeld wordt.

### 6.5.3 SmartRules

De SmartRules implementatie is niet moeilijk. SmartRules heeft een eigen business rule editor. Dit maakt het gebruik van SmartRules een stuk gemakkelijker. Op het moment dat de .NET assembly is ingeladen bij de SmartRules rule editor kan er gebruik gemaakt worden van een grafische omgeving om de business rules op te stellen. Nadat de business rules opgesteld zijn kunnen deze runtime door de rule engine worden toegepast. De SmartRules rule editor kan soms wel voor fouten zorgen. Bij deze fouten wordt geen omschrijving gegeven, waardoor het niet duidelijk is wat er fout gaat.

### 6.5.4 Aanbeveling

De implementatie van SmartRules is uitgebreider dan die van Drools .NET en NxBRE, maar wel overzichtelijk.

NxBRE en Drools .NET zijn geen gebruiksvriendelijke rule engines. Dit ligt vooral aan het feit dat beide rule engines geen eigen business rule editor hebben. In het onderstaande overzicht wordt een score toegekend aan de factoren "implementatie" en "gebruiksvriendelijkheid" van de drie verschillende rule engines.

	implementatie	Gebruiksvriendelijkheid
Drools.NET 3.0	+	-
NxBRE 3.2.0	-	--
SmartRules 2010.08	+	++

SmartRules is dus vrij eenvoudig te implementeren. Door gebruik te maken van de SmartRules rule editor wordt het gebruik van SmartRules stukken eenvoudiger. Na het opstellen van business rules in een grafische omgeving kunnen de rules runtime door de rule engine worden toegepast op de applicatie. Bij het gebruik van de SmartRules rule editor moet er wel consequent gebruik gemaakt worden van een versiebeheer tool.

## 7. Conclusie

Om te beoordelen welke rule engine het meest geschikt is om toe te passen op de .NET applicatie van de klant van Sogyo moesten er een aantal acties uitgevoerd worden.

Met het onderzoek “Wat is een Rule Engine?” (zie bijlage II) wordt aangetoond dat een rule engine een hulpmiddel is voor complexe software systemen. Met behulp van een rule engine kunnen de business processen gescheiden worden van de applicatie. De business processen staan dan in een knowledgebase. De rule engine is een krachtige tool die al deze business processen kan toepassen op data die wordt aangeleverd vanuit de applicatie. Deze data kan gemanipuleerd en geanalyseerd worden. Op het moment dat alle processen die van toepassing waren zijn doorlopen wordt de data teruggestuurd naar de applicatie.

Het onderzoek “.NET Rule Engine vergelijking” (zie bijlage III) geeft een overzicht van de verschillende eigenschappen van de diverse onderzochte .NET rule engines. De 3 rule engines met de hoogste score zijn vervolgens vergeleken met de wensen van de klant van Sogyo. Deze 3 rule engines voldoen alle 3 aan de wensen van de klant van Sogyo.

Tijdens het proces is er nog een extra rule engine ter sprake gebracht. Van deze rule engine zijn dezelfde onderzoeksaspecten onderzocht. Uit dit onderzoek kon geconcludeerd worden dat deze rule engine eigenlijk op nummer 1 zou moeten staan en dus is Windows Workflow Foundation vervangen door de rule engine SmartRules.

In theorie is het nu duidelijk dat SmartRules de beste keuze is, gevolgd door Drools .NET 3.0 en NxBRE. Om ook de praktische ervaring mee te kunnen nemen in het advies (zie bijlage X), zijn deze drie rule engines toegepast in een test omgeving.

Hieruit is gebleken dat Drools .NET en SmartRules vrij eenvoudig te implementeren zijn en NxBRE heel moeilijk. De business rules van Drools .NET en NxBRE zijn moeilijk op te stellen. Dit komt door het feit dat deze rule engines beide geen business rule editor hebben. Dit maakt Drools .NET en NxBRE gebruiksonvriendelijke rule engines. Daar in tegen heeft SmartRules wel een eigen business rule management system genaamd SmartRules Studio. Met SmartRules studio kunnen business rules opgesteld worden in een grafische omgeving en runtime worden toegepast door de rule engine op de applicatie.

Van alle onderzochte .NET rule engines heeft SmartRules de meeste mogelijkheden en voordelen. Ook in de praktijk is SmartRules eenvoudig te implementeren en zeer gebruiksvriendelijk. Zowel de theoretische mogelijkheden en voordelen van SmartRules als de bevindingen bij de implementatie en het gebruik van SmartRules voldoen aan de eisen en wensen van de klant. Dit maakt SmartRules op dit moment de beste keuze voor de klant van Sogyo.



## Evaluatie

### Overwinningen

#### Planning

Uit het overzicht van 3.3.1 is te herleiden dat er veel vertraging is opgetreden tijdens de implementatie fase van het project. Er is veel vertraging ontstaan door fouten en complicaties die optraden binnen het project. Om deze fouten en complicaties op te lossen moest er soms voor een hele andere aanpak gekozen worden. Het gevolg hiervan kon zijn dat een deel van de testapplicatie herschreven moest worden of een SmartRules Studio project opnieuw gemaakt moest worden.

**Desondanks is het toch gelukt om aan het einde van deze fase alle taken te hebben voltooid.**

#### Werkzaamheden

Bij het opstellen van het plan van aanpak werd mij verteld dat ik vrij enthousiast was geweest met het inplannen van de hoeveelheid taken. Op deze manier zou het mij teveel tijd gaan kosten om alle taken te voltooien. Door consequent te reflecteren op de planning en het stellen van prioriteiten is het proces snel verlopen.

**Uiteindelijk is het mij gelukt om alle taken te voltooien plus extra taken zoals de SmartRules handleiding, de nieuwe business case, het opstellen van de scrumbegrippen, het uitzoeken van de relatie tussen AI en rule engines en het verdiepen in de “No Silver Bullet” theorie.**

#### Implementatie

De implementatie fase van één rule engine had ik op ongeveer 2 weken geschat. Ik wist dan bij deze schatting nog niet wat mij te wachten stond en hoe ingewikkeld het toepassen kon zijn. Door mij goed in de documentatie te verdiepen is het gelukt om dit sneller te voltooien.

**De rule engines zijn alle drie binnen 2 dagen per rule engine geïmplementeerd.**

### Uitdagingen

#### NxBRE

Na het toepassen van de NxBRE rule engine moesten er rules geschreven worden om aan te tonen dat NxBRE ook daadwerkelijk werkt. Het schrijven van deze rules kon in drie formaten (ruleml, .hrf, .vcd). Ruim één week lang heb ik pogingen gedaan om deze rules werkend te krijgen, maar NxBRE gaf steeds andere fouten. Een van de voorbeeld rule files werd echter wel door NxBRE geaccepteerd. Aan de hand hiervan kon ik concluderen dat de implementatie van SmartRules succesvol was, maar de rule files nog steeds niet correct opgesteld konden worden. Om deze rule files wel correct op te stellen heb ik tutorials gevolgd en tools gebruikt. Helaas waren de tutorials en tools niet specifiek gericht op NxBRE en accepteerde de rule engine deze files dus nog niet.

Na ruim een week hebben we besloten dat NxBRE niet gebruiksvriendelijk genoeg is en hebben daarom besloten om verder te gaan met SmartRules.

## **Wat heb ik geleerd?**

### **Discipline**

In het begin van de stageperiode heb ik aangegeven dat ik graag één dag in de week thuis wil werken. Deze dag heb ik gebruikt voor het schrijven van documenten met de nadruk op deze scriptie. Ik heb mezelf er iedere dag die ik thuis gewerkt heb toe kunnen zetten om mijn taken te doen en mij niet te laten verleiden om andere dingen te gaan doen i.p.v. de taken die waren ingepland.

### **Opstellen van deelvragen**

Tijdens het schrijven van het plan van aanpak ben ik door mijn bedrijfsbegeleider gewezen op een methode om effectief deelvragen op te stellen. Bij deze methode wordt er vanuit de hoofdvraag terug geredeneerd welke stappen er nodig zijn om tot een antwoord te komen. Op deze manier weet je zeker dat de conclusies van de deelvragen tot een antwoord op de hoofdvraag leiden.

### **Prettige ervaringen**

Ik vond het zeer prettig dat mij veel vrijheid is gegeven. We hebben over het algemeen een keer per week een bijeenkomst gehad om de voortgang te bespreken. Tijdens deze gesprekken hebben we even besproken wat ik de afgelopen week gedaan had en wat ik de komende week zou gaan doen. Hier werden eventueel ideeën meegegeven waar ik me op kon richten.

Verder heb ik veel thuis mogen werken. Elke vrijdag kreeg ik hier gelegenheid voor en als dit ooit op een andere dag beter uitkwam was dat ook geen probleem. Aangezien mijn stageperiode precies in de winter valt, is het een aantal dagen moeilijker geweest om naar de boerderij te komen. In dit geval was het ook geen probleem als ik thuis mijn werk voortzette.

## **Invloeden m.b.t. mijn functioneren**

### **Onvoldoende werkgeheugen**

Mijn stageperiode (en dan vooral de implementatie fase) had effectiever kunnen verlopen met een krachtigere pc. De pc die ik heb gebruikt om mijn taken op uit te voeren had te weinig werkgeheugen. Hierdoor is het regelmatig voorgekomen dat een programma niet meer reageerde, de hele pc herstartte of het gewoon lang duurde voor een bepaalde taak voltooid was.

Nadat ik de situatie besproken had, werd mij verteld dat ik een andere pc mocht kiezen. Helaas zag ik het niet zitten om mijn gehele werkomgeving over te zetten naar een andere pc en heb dus de rest van mijn stage periode veel geduld moeten hebben.

## Literatuurlijst en bronvermelding

### Internetpagina's

Id	Website
W1	<a href="http://geekswithblogs.net/cyoung/articles/56488.aspx">http://geekswithblogs.net/cyoung/articles/56488.aspx</a>
W2	<a href="http://en.wikipedia.org/wiki/No_Silver_Bullet">http://en.wikipedia.org/wiki/No_Silver_Bullet</a>
W3	<a href="http://www.cs.nott.ac.uk/~cah/G51ISS/Documents/NoSilverBullet.html">http://www.cs.nott.ac.uk/~cah/G51ISS/Documents/NoSilverBullet.html</a>
W4	<a href="http://dist.codehaus.org/droolsdotnet/">http://dist.codehaus.org/droolsdotnet/</a>
W5	<a href="http://droolsdotnet.codehaus.org/Drools.Net+-+3.0+User+and+Installation+Guide">http://droolsdotnet.codehaus.org/Drools.Net+-+3.0+User+and+Installation+Guide</a>
W6	<a href="http://www.docstoc.com/docs/42784923/NxBRE-Documentation-3_2_0">http://www.docstoc.com/docs/42784923/NxBRE-Documentation-3_2_0</a>
W7	<a href="http://www.kontac.net/help/SmartRules_Help.htm">http://www.kontac.net/help/SmartRules_Help.htm</a>

### Overige bronnen

Id	Titel	Type
B1	Drools NET-3.0 guide.doc	Handleiding
B2	Scriptie Tim de Jager.pdf	Scriptie
B3	NxBRE_Manual.pdf	Handleiding
B4	TRANSLATOR_Manual.pdf	Handleiding

## Bijlagenoverzicht

Id	Titel
I	Onderzoek werkplan - Rule Engines
II	Onderzoek verslag - Rule Engines
III	Grounded theory_comparing rule engines
IV	Plan van Aanpak
V	Afstudeerstage opdracht omschrijving
VI	Business Case
VII	Scrum begrippen
VIII	SmartRules
IX	Eisen en wensen van de klant
X	Adviesrapport