

## De rol van een editor

Verbetering van de RuleXpress Business Rules  
Editor

05-02-2010

Jeffrey Schoemaker

Scriptienummer: 626

Supervisor: Dr. Stijn Hoppenbrouwers  
Prof. dr. Ben Dankbaar

**Master script Computer Science**

Management en Toepassing-Track

Radboud Universiteit Nijmegen

LibRT

RuleArts

Auteur: Jeffrey Schoemaker  
E-mail: jeffreyschoemaker@gmail.com  
Datum: 05-02-2010  
Scriptienummer: 626

Supervisors: Dr. Stijn Hoppenbrouwers, Faculteit NWI, Radboud Universiteit Nijmegen  
Prof. dr. Ben Dankbaar, Faculteit NWI, Radboud Universiteit Nijmegen

Drs. Silvie Spreeuwenberg, LibRT  
Drs. Kees Noordsij, RuleArts

## VOORWOORD

---

Afstuderen is een zwaar traject. Het is een traject waarin je een half jaar bezig bent met een onderwerp waar je interesse ligt en waar nog wat te ontdekken valt. Vol goede moed ga je van start en met een open vizier treed je het afstudeertraject binnen. Al snel wordt duidelijk dat het eigenlijk toch best wel een complexe wereld is, waarin je bent beland, en dat oogkleppen toch echt een must zijn om binnen een half jaar een goed verslag te kunnen opleveren. Gedurende het afstuderen loop je vervolgens meerdere keren vast, moet je soms enkele stappen terug doen om veel verder vooruit te komen. Je ziet delen van jouw kostbare werk in de prullenbak verdwijnen omdat inzichten in een later stadium je een hele andere kant opsturen.

Al deze fases heb ik dus ook doorlopen en ik ben ontzettend blij dat er goede begeleiders om mij heen waren die mij daarin stuurden, hielpen en nieuwe inzichten verschaften. In het bijzonder wil ik Stijn Hoppenbrouwers bedanken. Altijd stond de deur open om mijn ideeën te toetsen en ik vertrok na een gesprek met Stijn altijd weer vol goede moed en met een lading extra ideeën. Daarnaast is de begeleiding van Silvie Spreeuwenberg van uitzonderlijke waarde geweest voor mijn afstudeerverslag. Bedankt voor alle tijd en kennis die je met mij hebt gedeeld! Iets verder van de zijlijn misschien, maar zeker niet minder belangrijk waren Ben Dankbaar en Kees Noordsij. Hartelijk bedankt voor alle begeleiding gedurende het afstuderen.

Ook wil ik Kees Koster graag bedanken voor het beschikbaar stellen van zijn natuurlijke taal-parser. Weliswaar is dit onderdeel niet in de uiteindelijke versie van de editor terecht gekomen, maar een proof-of-concept implementatie biedt zeker een mogelijkheden voor verder onderzoek in de toekomst.

Voor mij is deze scriptie de afsluiting van een tijdperk genaamd 'Studententijd'. Terugblikkend weet ik nu al zeker dat dit een uniek tijdsvak in mijn leven is geweest waar ik voor de rest van mijn leven met weemoed aan terug zal denken. Graag bedank ik iedereen die heeft bijgedragen aan deze unieke periode.

Alvast veel leesplezier gewenst,

Jeffrey Schoemaker

Nijmegen, januari 2009

## ABSTRACT

---

In 2007 is er een productstandaard geaccepteerd door de Object Management Group (OMG) met daarin concrete handvatten om bedrijfsregels gestructureerd vast te leggen. Deze productstandaard heet de "Semantics of Business Vocabulary and Rules" (SBVR). SBVR is het resultaat van een unieke samenwerking tussen de werelden van de 'Terminologie en Vocabulaire', 'Formele Logica', 'Organisatie' en 'Linguïstiek en communicatie'.

Een goede bedrijfsregel volgens SBVR bestaat uit termen, die worden verbonden door andere termen en waardoor middel van sleutelwoorden een beperking aan worden opgelegd. Als een regelset conform SBVR is vastgelegd, moet het eenvoudiger zijn om de afdeling IT te laten communiceren met de gebruikers van de regels uit de organisatie. Idealiter beschrijven de gebruikers uit de organisatie zelf hun regels en onderhouden zij deze ook zelf, aangezien de domeinkennis van het bedrijf bij de gebruiker aanwezig is. Echter om een goede bedrijfsregel te schrijven, moet een regel aan een aantal voorwaarden voldoen. Om dit te ondersteunen kan een editor gebruikt worden. Er zijn verschillende editors op de markt die het proces op een verschillende manier benaderen.

In deze scriptie is nader gekeken naar de verschillende rollen die een editor kan spelen in het proces van het opstellen van bedrijfsregels. Volgens deze scriptie kan een editor een viertal verschillende rollen vervullen, namelijk een faciliterende, een ondersteunende, een evaluerende en een coachende rol. Deze verschillende rollen hebben consequenties voor het verloop van het proces van het opstellen van regels en op de gebruiker die de bedrijfsregels opstelt.

Aan de hand van deze rollen is een concrete editor voor het opstellen van bedrijfsregels geïmplementeerd, waarbij alle verschillende rollen tot uitdrukking zijn gekomen. De gedachte achter al de verschillende implementaties is toegelicht. Niet alle rollen zijn volledig geïmplementeerd in de editor, maar ze zijn wel volledig beschreven.

De editor is tevens geevalueerd met verschillende gebruikers. In deze evaluatie kregen de testers een set met regels aangereikt. Met behulp van de geïmplementeerde editor zijn enkele regels opnieuw opgesteld. Deze regels zijn op kwaliteit beoordeeld door drie bedrijfregel-experts. Uit de resultaten is gebleken dat de gemaakte regels van een hoger niveau waren dan de oorspronkelijke regels. De testers hadden geen ervaring in het opstellen van bedrijfsregels.

## INHOUDSOPGAVE

---

Voorwoord .....	3
Abstract.....	4
1. Inleiding.....	7
1.1 SBVR .....	8
1.2 Vastleggen van bedrijfsregels.....	9
1.3 Onderzoeksvragen .....	10
1.4 Methodologie .....	11
2. Bedrijfsregels .....	12
2.1 In natuurlijke taal .....	12
2.2 Semantics of business vocabulary and rules.....	14
2.2.1 Concepten .....	14
2.2.2 Termen .....	15
2.2.3 Feittypen.....	15
2.2.4 Regels.....	15
2.2.5 Notatie .....	16
2.3 Het Totaalplaatje.....	17
2.4 De mensen .....	19
2.5 Het expliciteren van de bedrijfsregels .....	20
2.5.1 Constructie .....	20
2.5.2 Reconstructie .....	20
2.5.3 Kwaliteit van de regels.....	21
2.5.4 Dynamisch proces.....	22
3. De editor.....	23
3.1 Editors voor bedrijfsregels .....	23
3.1.1 Soorten editors.....	23
3.2 RuleXpress.....	24
3.3 Het Proces.....	25
3.4 De gebruiker .....	26
3.5 Verschillende rollen.....	27
3.5.1 Faciliterende rol .....	27
3.5.2 Ondersteunende rol.....	29
3.5.3 Een evaluerende rol.....	31
3.5.4 Coachende rol .....	33

4. Rulespeak.....	35
4.1 De richtlijnen.....	36
4.2 Samenvatting .....	43
5. De nieuwe editor .....	45
5.1 Faciliterende rol .....	46
5.2 Ondersteunende rol.....	47
5.2.1 De interactie met de wizard .....	47
5.3 Evaluerende rol.....	57
5.4 Coachende rol .....	58
5.5 De implementatie.....	59
6. Evaluatie .....	60
6.1 Opzet.....	60
6.2 Resultaten.....	61
6.2.1 Definitieve regels .....	61
6.2.2 Modelleerervaring versus geen modelleerervaring.....	61
6.3 Vragenlijst.....	62
6.3.1 Vragen over de case.....	62
6.3.2 Vragen over de editor .....	63
6.3.3 Open vragen.....	64
6.4 Observaties .....	65
6.5 Reflectie op de evaluatie .....	66
7. Conclusies .....	67
8. Toekomstig werk .....	71
9. Vocabulaire .....	72
Bibliografie.....	73
Appendix A. De editors .....	75
Appendix B. De case.....	79
De oorspronkelijke regels .....	79
De termen .....	81
Het feittypemodel.....	82
Appendix C. Het opstellen van een bedrijfsregel met behulp van de editor .....	83

## 1. INLEIDING

---

Overal in het leven zijn regels aanwezig. Bij het aanvragen van een parkeervergunning, het verkrijgen van een hypotheek of het reizen met het openbaar vervoer. Altijd zijn er regels die aangeven wat wel en wat niet mag. Binnen bedrijven spelen regels een cruciale rol, want zij leggen de basis waarop beslissingen binnen een bedrijf worden genomen.

Hedendaagse bedrijven worden geconfronteerd met een toenemende vraag om hun activiteiten en beslissingen te verantwoorden en om snel te kunnen reageren in een flexibele omgeving. Door de bedrijfsregels duidelijk voor ogen te hebben en deze eenduidig te definiëren kan worden voldaan aan deze vraag

In de praktijk echter zijn deze bedrijfsregels vaak niet eenduidig vastgelegd. Vaak zijn bedrijven afhankelijk van een handvol werknemers die overzicht, inzicht en kennis hebben van de operationele regels. Zij hebben het 'grotere plaatje' voor ogen, maar deze regels liggen enkel vast in hun hoofd. Hierdoor is het voor nieuwe medewerkers, maar ook externe partijen lastig om inzicht te krijgen in regels en de basis waarop beslissingen worden genomen. Daarnaast zijn regels soms vastgelegd in reglementen die van alle kanten kraken en op meerdere manieren zijn uit te leggen.

Het correct vastleggen van regels is niet alleen belangrijk voor de gebruikers zelf. Tevens kan het correct vastleggen belangrijk zijn voor de IT-afdeling. Als regels niet duidelijk zijn vastgelegd, liggen er vrijheden voor de IT-afdeling die zij moet invullen, en dat levert nogal eens onduidelijkheden op. De IT-afdeling en de bedrijfsvloer spreken namelijk van nature niet dezelfde taal. De gebruikers in een organisatie zijn namelijk gewend om zich in natuurlijke taal uit te drukken en hierbij komt geregeld een stukje proza om de hoek kijken. De IT moet echter computers aansturen, en computers luisteren alleen naar instructies, waarin vaagheden niet worden geaccepteerd. De computer heeft behoefte aan een formele taal.

Op de een of andere manier moet er een soort van communicatie plaats vinden tussen de IT en de gebruikers in een organisatie. Decennialang vertelde de gebruikers in wollige natuurlijke taal hun wensen aan de IT, en de IT probeerde dat om te zetten in modellen en programma's. De gebruikers specificeerde meestal de eisen niet volledig, waardoor er open stukken of vaagheden aanwezig waren in deze eisen. Deze moesten vervolgens door de IT worden opgevuld in de modellen.

Nadat de IT zijn modellen en programmacode had gemaakt, werd er getracht de correctheid hiervan te verifiëren. De gebruikers werden vervolgens geconfronteerd met ER-Diagrammen of UML-modellen en met hulp van een IT'er werd getracht deze te verifiëren. Het probleem hiermee was dat de gebruikers eigenlijk niet goed begrepen wat er precies werd geverifieerd, omdat zij zelf niet de modellen konden interpreteren, maar dit in samenspraak met een andere persoon (die deze modellen wel kon lezen) moest gebeuren.

Het gevolg hiervan was dat in een later stadium, bijvoorbeeld tijdens de bouw van een programma, bleek dat de modellen toch niet correct waren. De oorzaak is vaak dat de gebruikers uit de organisatie toch niet bedoelden wat er in de modellen weergegeven werd.

---

## 1.1 SBVR

---

Decennialang is er gewerkt met dit probleem aangezien er geen concrete oplossing voor handen was. Een consortium van een twintigtal bedrijven is in 2003 gestart met het idee om een productstandaard te ontwerpen voor het opstellen van bedrijfsregels. Na een jarenlang proces is deze standaard uiteindelijk in december 2007 aangenomen door de Object Management Group (OMG). De OMG is een internationaal consortium van bedrijven uit de computerindustrie dat IT-gerelateerde standaarden ontwerpt (of laat deze ontwerpen).

In 2007 werd de Semantics of Business Vocabulary and Business Rules (SBVR) geaccepteerd door de OMG. SBVR is een product standaard die ondersteuning biedt op het gebied van het ontwikkelen en vastleggen van bedrijfsvocabulaire en bedrijfsregels. Met behulp van deze standaard is het mogelijk om organisatie zelf haar bedrijfsvocabulaire en bedrijfsregels te laten opstellen zodat het voor de gebruikers binnen de organisatie begrijpelijk is en de IT hier ook mee kan werken.

SBVR is enkel de productstandaard en deze standaard definieert enkel hoe regels gestructureerd moeten worden. Om dit daadwerkelijk te implementeren is er een taal nodig waarmee dit gebeurt. Eén van de talen waarmee dit mogelijk is, is RuleSpeak. Deze taal is ontwikkeld door Ron Ross van Business Rule Solutions. RuleSpeak is een verzameling richtlijnen om bedrijfsregels eenduidig vast te leggen op een bedrijfsvriendelijke en precieze manier. RuleSpeak is geen strikte procedure, maar eerder een leidraad om te helpen met het opstellen van de regels. Een document wat gelieerd is aan RuleSpeak is het Business Rules Manifesto. Dit manifest bestaat uit 10 artikelen waarin wordt beschreven hoe er moet worden omgegaan met bedrijfsregels. Eén van de belangrijke boodschappen die het Business Rules Manifesto uitdraagt is: *“Laat deskundigen uit de bedrijfspraktijk regels opstellen”* en RuleSpeak is een concrete handreiking aan dit credo. RuleSpeak tracht ondersteuning te bieden zodat de gebruikers van de regels ook de opstellers en beheerders van de regels moeten zijn.



---

## 1.2 VASTLEGGEN VAN BEDRIJFSREGELS

---

Er zijn diverse producten op de markt om ondersteuning te bieden bij het opstellen, beheren en uitvoeren van deze regels. Enkele daarvan zijn RMSuite, Haley Systems, ILOG Rules en RuleXpress. Al deze producten benaderen dit proces op een verschillende manier. RuleXpress is een softwaretool ontwikkeld door het bedrijf RuleArts. Het biedt ondersteuning om bedrijfsregels vast te leggen, te analyseren en te onderhouden. Het is de bedoeling dat de business zelf haar regels en gebruikte vocabulaire kan vastleggen en deze ook zelf onderhoudt. Dit is niet een eenmalig proces, maar een doorlopend verbeterproces waarbij 'communicatie' het toverwoord is. Door te communiceren met elkaar kunnen eenduidige regels en definities worden vastgelegd.

Doordat bedrijfsregels in natuurlijke taal worden vastgelegd, heeft de gebruiker alle vrijheid om de regels vast te leggen. Dit heeft zijn voordelen, want doordat regels in natuurlijke taal worden beschreven, kan de business zelf zijn regels opstellen. Maar er zijn ook nadelen, want door het ontbreken van duidelijke regels over het opstellen van regels en door het ontbreken van een vorm van syntaxis, kan van alles worden opgesteld. De kwaliteit van deze regels moet op een bepaalde manier worden gewaarborgd.

RuleXpress probeert de gebruiker te helpen bij het opstellen van de regels, maar de gebruiker daarin niet te beperken in zijn vrijheid. Door achteraf een kwaliteitscontrole over de regels uit te voeren, wordt de gebruiker duidelijk hoe goed zijn regel is.

Op dit moment biedt de editor in RuleXpress voornamelijk een faciliterende rol. Dit houdt in dat regels kunnen worden vastgelegd en worden gewijzigd door middel van een editor. Achteraf kan er vervolgens een controle worden gedaan op de kwaliteit van de regels. Binnen het bedrijf bestaat de wens om de gebruiker meer ondersteuning te bieden bij het opstellen van een kwalitatief goede regel zonder de creatieve vrijheid van de gebruiker hierbij te beperken.

### 1.3 ONDERZOEKSVRAGEN

---

Mijn onderzoek bestond uit twee delen. In het eerste gedeelte is er gekeken naar het proces van het opstellen en beheren van de bedrijfsregels vanuit het oogpunt van de (bedrijfs)gebruiker. Hierbij is voornamelijk gekeken naar de rol van de editor in dit proces.

Vervolgens is er onderzocht welke nieuwe functionaliteiten een meerwaarde kunnen bieden voor RuleXpress. Uitgangspunt hierbij was dat de nieuwe functionaliteit een meer ondersteunende/coachende rol aan de gebruiker moet bieden. Enkele van de gevonden functionaliteiten zijn uitgewerkt in een proof-of-concept applicatie.

De onderzoeksvragen zagen er als volgt uit;

1. Welke rol speelt de editor in het proces van het opstellen van bedrijfsregels?
  - a. Hoe ziet het proces van het opstellen van bedrijfsregels eruit?
  - b. Welke ideeën bestaan er, in de wereld van softwareontwikkeling, over de rol(len) die een editor speelt in dit proces?

Voor elke rol zijn vervolgens de volgende vragen uitgewerkt:

- i. Wat zijn de gevolgen van deze rol voor het proces?
    - ii. Wat zijn de gevolgen van deze rol voor de kwaliteit van de bedrijfsregels?
    - iii. Wat voor eisen worden er aan de gebruiker gesteld in deze rol?
2. Op welke manier kan de editor meer ondersteuning bieden aan de gebruiker?
  - a. Welke mogelijkheden zijn er om meer ondersteuning te bieden aan de gebruiker?
  - b. Hoe worden deze aanpassingen ervaren door de gebruiker?
  - c. Wat zijn de technische consequenties van deze aanpassingen?

Uitgangspunt bij deze vragen was dat de gebruiker uit een organisatie de bedrijfsregels zelf opstelt.

Bij vraag 2 is een vergelijking gemaakt met de huidige editor van RuleXpress. Aan de hand van interviews en/of enquêtes met gebruikers en ontwikkelaars van RuleXpress is onderzocht wat voor verbeteringen er ontworpen en geïmplementeerd zouden kunnen worden. Er is tijdens deze stap een stand-alone applicatie gebouwd om deze ontwerpen in te implementeren. Dit houdt in dat het niet de bedoeling is om de applicatie direct al te integreren in de bestaande RuleXpress-editor. Hiervoor is gekozen omdat dit technisch gezien een te complexe aangelegenheid is, en welke het doel van dit verslag niet dient. In een latere fase zou deze integratie altijd nog kunnen plaatsvinden.

Verder heeft vraag 2b voornamelijk een evaluerende rol in het project. Aan de hand van de editor RuleXpress zijn concrete verbeteringen gedaan aan de editor. Deze verbeteringen vloeien voort uit de resultaten van vraag 1b en 2a. Vervolgens is aan de hand van een testcase bekeken of de wijzigingen ook daadwerkelijk een positieve invloed op het proces hebben.

---

## 1.4 METHODOLOGIE

---

Zoals in het vorige hoofdstuk al geschetst werd, is het project in twee delen te onderscheiden. In het eerste deel is gekeken naar de rollen die een editor kan vervullen, terwijl in het tweede gedeelte er ook daadwerkelijk een prototype geïmplementeerd is. Het eerste gedeelte dat draait om de rol van de editor in het gehele proces, is voornamelijk aan de hand van een literatuurstudie en introspectie gedaan. In de wereld van software is nog niet veel geschreven over de verschillende rollen die een editor kan vervullen. Hier is onderzoek naar gedaan en waar nodig zijn deze door eigen observaties aangevuld. Daarnaast zijn er verschillende bronnen op het gebied van bedrijfsregels en editors geraadpleegd. Deze bevindingen zijn uitvoerig met de afstudeerbegeleiders van zowel het bedrijf als de universiteit bediscussieerd.

Het andere gedeelte, betreffende de daadwerkelijke verbeteringen aan een bestaande editor, is onderzocht binnen het kader van Design Science. Dit paradigma biedt ondersteuning aan het onderzoek naar daadwerkelijke verbeteringen aan een bestaand IT-artefact, in concreto de editor. Verbeteringen worden voorgesteld en ontworpen. Om vervolgens te testen of deze wijzigingen ook daadwerkelijk verbeteringen zijn, is de wijziging geïmplementeerd in een applicatie. Aan de hand van evaluaties met gebruikers is vervolgens gekeken of de verbeteringen ook daadwerkelijk een positieve uitwerking hebben. Het ontwerp van de editor is een iteratief proces waarbij aan de hand van kleine incrementele veranderingen een evaluatie worden gedaan met gebruikers om te kijken of de veranderingen daadwerkelijk van positieve invloed zijn.

## 2. BEDRIJFSREGELS

---

Ieder bedrijf heeft te maken met bedrijfsregels, ieder persoon krijgt te maken met bedrijfsregels en iedereen weet wat ze zijn. De meest gangbare definitie voor een bedrijfsregel is: *“Een statement dat een aspect van het bedrijf definieert of beperkt. Het is bedoeld om de bedrijfsstructuur te bepalen of te controleren of om het gedrag van het bedrijf te beïnvloeden. Bedrijfsregels zijn atomair ~ oftewel, ze kunnen niet verder worden onderverdeeld.”* (Hay & Healy, 2000)

In feite zijn bedrijfsregels een verzameling van regels die in een bedrijf gelden en waar actoren rekening mee moeten houden. Een voorbeeld van bedrijfsregel is: *“Een klant is een ‘vaste klant’ als zijn jaarlijkse besteding meer dan 1000 euro bedraagt.”* Eigenlijk niets nieuws onder de zon. Regels bestaan al net zo lang als bedrijven bestaan en een bedrijf hoeft eigenlijk ook niets essentieel nieuws te doen om bedrijfsregels vast te leggen.

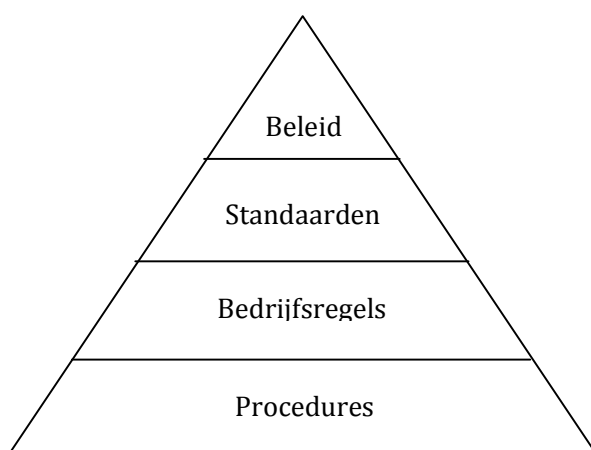
### 2.1 IN NATUURLIJKE TAAL

---

Bedrijfsregels worden opgesteld in natuurlijke taal. Dit wordt gedaan omdat men wil dat de gebruikers uit de organisatie in staat worden gesteld deze regels te begrijpen, aan te passen en uit te voeren.

Er kleven echter zowel voor- als nadelen aan het feit dat dit voor en vooral door gebruikers van de regels wordt gedaan. Het grote voordeel is uiteraard dat een bedrijf de regels zelf kan opstellen en kan begrijpen, aangezien men zich niet hoeft te houden aan een andere syntaxis dan die van de eigen moedertaal. Het grote nadeel is echter ook het ontbreken van een formele syntaxis. Natuurlijke taal is van nature ambigu en moeilijk te formaliseren. Dit houdt in dat de borging van de kwaliteit van de regels een lastig fenomeen is.

Het proces van het opstellen van bedrijfsregels begint vaak aan de hand van algemene regels, die zijn vastgelegd in het beleid van een bedrijf. Dit zijn veelal brede richtlijnen, die verder vertaald moeten worden tot concrete, niet-ambigue regels. In figuur 1 is een schematische weergave hiervan te vinden. (IsecT Ltd.)



Figuur 1 Verschillende lagen van richtlijnen

Beleid: Brede richtlijnen die de filosofie van een bedrijf weergeven.

Standaarden: Gedetailleerdere regels die omschrijven hoe het beleid gerealiseerd dient te worden.

Bedrijfsregels: Concreet geformuleerde beperkingen om te voldoen aan de standaarden.

Procedures: Gedetailleerde stappen om de bovengenoemde regels te implementeren en uit te voeren.

Om van het beleid naar standaarden, bedrijfsregels en procedures te komen, zullen er telkens interpretatiestappen moeten plaats vinden. Deze stappen zullen voornamelijk door de werknemers binnen een bedrijf worden uitgevoerd. Beleid wordt namelijk vaak in algemene termen beschreven en het is aan de werknemers om deze om te zetten in concretere bedrijfsregels en procedures. In deze interpretatiestap zit vaak het probleem. De Business Rules Group omschrijft deze problemen als *business ramblings*. Barbara von Halle definieert *ramblings* als volgt; "Zinnen die soms duidelijk zijn, soms (misschien opzettelijk) ambigu, en meestal meer dan één idee bevatten." Deze ambiguïteit en vaagheden moeten verdwijnen om de IT iets te kunnen laten doen met deze regels. Als dit niet gebeurt, zal de IT namelijk zelf een interpretatieslag moeten maken om het model volledig en 'correct' te maken.

De IT neemt echter lang niet altijd de juiste interpretatiestappen. Vaak wordt dit veroorzaakt doordat de IT simpelweg niet de specifieke domeinkennis van de organisatie heeft. De organisatie zou daarom eigenlijk gedwongen moeten worden om zijn regels zelf eenduidig en niet-ambigu te specificeren, zodat deze interpretatiestappen niet meer nodig zijn.

---

## 2.2 SEMANTICS OF BUSINESS VOCABULAIRE AND RULES

---

In 2003 is een consortium van een twintigtal bedrijven gestart met het idee om een productstandaard te ontwerpen voor het opstellen van bedrijfsregels. Deze productstandaard werd getiteld SBVR; Semantics of Business Vocabulary and Rules. In 2007 is de SBVR-standaard geaccepteerd door het OMG, waardoor voor het eerst concrete handvatten zijn aangegeven om bedrijfsregels gestructureerd vast te leggen.

SBVR is het resultaat van een samenwerking van bedrijven uit verschillende ‘werelden’. Het is een samensmelting van werk uit de werelden ‘Terminologie en vocabulaire’, ‘Formele logica’, ‘Bedrijfswereld’ en ‘Linguïstiek en communicatie’. Nooit eerder is er op zo’n grote schaal samengewerkt om tot een dergelijke standaard te komen.

SBVR is een metamodel om modellen te maken van bedrijfsvocabulaire en –regels. Uitgangspunten hierbij zijn dat

- het bedrijfsperspectief als uitgangspunt wordt genomen (en niet zoals vaak het IT-perspectief);
- de taal van de organisatie wordt gebruikt (en niet zoals vaak de IT-modellen die de communicatie bepalen);
- de regels en het vocabulaire niet per se geautomatiseerd hoeven te worden. Het hoofddoel is namelijk om regels goed vast te leggen en het hoofddoel is niet om de IT er mee te laten werken (alhoewel dit natuurlijk vaak wel een doel is). (Spreeuwenberg, What happened to the B and the M of BRM?, 2009)

De gedachte is dat SBVR enerzijds formeel genoeg is om te dienen als basis voor het genereren van applicaties anderzijds begrijpelijk genoeg is voor de gebruikers uit de organisatie. (Blankena, 2008)

SBVR reikt een metamodel aan met in het achterhoofd de volgende mantra (Business Rules Group, 2003):

*rules build on facts*  
*facts build on concepts*  
*as expressed by terms*

---

### 2.2.1 CONCEPTEN

---

De Business Rules Mantra geeft direct een belangrijk punt aan binnen SBVR, namelijk het verschil tussen de betekenis van een concept en de representatie van een concept. Eenvoudig gezegd is de betekenis van een concept, de betekenis die een ieder in zijn hoofd heeft. De representatie van dit concept is de manier waarop je het concept uitdrukt in bijvoorbeeld natuurlijke taal, een symbool of een bord.

Een voorbeeld is het concept “*dat er hier niet gefietst mag worden*”. Dit concept kan op verschillende manieren tot uitdrukking worden gebracht; bijvoorbeeld door middel van een rond bord met in het midden een fiets en een rode rand erom heen of door middel van een bordje met hierop “Verboden te fietsen” of door middel van een regel.

In een organisatie bestaan er vaak ook verschillende representaties van eenzelfde concept, die niet altijd hetzelfde betekenen. Zo kan het voorkomen dat het concept “klant” voor de afdeling marketing wordt gedefinieerd als *“een persoon die interesse heeft om ons product in de toekomst te gaan kopen”* en door de afdeling verkoop als *“een persoon die een product heeft aangeschaft”*. Het concept klant heeft in deze verschillende werelden een verschillende betekenis. Het is een gegeven dat verschillende afdelingen binnen één enkele organisatie verschillende betekenissen geven aan eenzelfde concept. Dit gegeven wordt door SBVR onderkend en ondersteund.

---

### 2.2.2 TERMEN

---

In SBVR worden concepten vastgelegd door middel van termen. Deze termen kunnen een verschillende betekenis hebben binnen verschillende gemeenschappen (communities). SBVR tracht niet deze termen een gemeenschappelijke betekenis te geven, maar juist handvatten te geven om de termen te expliciteren zodat het duidelijk is wat een gemeenschap ermee bedoelt.

Voorbeelden van termen die een rol kunnen spelen in een organisatie zijn *“klant”, “vaste klant”, “bestelling”* en *“bestelbedrag”*. De termen en de definities van de termen worden samen het vocabulaire genoemd.

---

### 2.2.3 FEITTYPEN

---

Tussen verschillende concepten kan een relatie bestaan en deze relatie dient dan ook te worden geëxpliciteerd. Tussen de concepten “klant” en “bestelling” zal vast de relatie bestaan dat *“een klant een bestelling plaatst”, “een bestelling een totaal bestelbedrag heeft”* en *“een klant een vaste klant is”*.

Deze kennis dient te worden vastgelegd en in SBVR gebeurt dat door middel van feittypen. Feittypen worden altijd uitgedrukt door middel van werkwoord, eventueel in combinatie met een voorzetsel. Feittypen beschrijven het mogelijk voorkomen van een actie tussen twee concepten (Ross, Business Rule Concepts, 2005). Het kan zijn dat er meer dan één relatie bestaat tussen twee concepten. Een klant kan bijvoorbeeld een factuur zowel ontvangen, als betalen.

Als alle feittypen worden samengevoegd in een model ontstaat het feittypemodel. Hierin staan alle gebeurtenissen of situaties die kunnen voorkomen in het door het model beschreven domein. Het model beschrijft op conceptueel niveau alle operaties die binnen een organisatie kunnen plaats vinden. Feittypen zijn gerelateerd aan predicaten in de taal van de logica.

---

### 2.2.4 REGELS

---

Tenslotte worden deze feittypen als bouwstenen gebruikt om regels te maken. Waar een feittype beschrijft dat iets kan gebeuren, leggen regels beperkingen op aan de feittypen. Een voorbeeld is; *“een klant is altijd een vaste klant indien de klant een bestelling heeft geplaatst met een totaal bestelbedrag van meer dan 10.000 euro.”*

Door middel van sleutelwoorden wordt er een beperking opgelegd aan de bestaande feittypen en ontstaan er regels. In het voorbeeld zijn er twee sleutelwoorden te vinden. “*Altijd*” geeft aan dat het hier een structurele regel betreft die ten alle tijden waar moet zijn. “*Indien*” introduceert een voorwaarde waar aan voldaan moet worden. De sleutelwoorden zijn gerelateerd aan logische operatoren in de taal van de logica.

---

### 2.2.5 NOTATIE

---

Aangezien SBVR slechts een metamodel is om vocabulaire, feittypen en regels vast te leggen zal er ook een taal moeten zijn waarmee dit geëxpliciteerd wordt. Deze taal beperkt de natuurlijke taal en stelt een gestructureerd vocabulaire ter beschikking. Door deze beperkte taal te gebruiken wordt er voor gezorgd dat verschillende gebruikers dezelfde woorden en frases gebruiken om dezelfde dingen uit te drukken. In de SBVR-standaard zijn er drie referentienotaties die gebruikt zijn tijdens de ontwikkeling van de standaard.

Eén daarvan is SBVR Structured English. Zoals de naam al doet vermoeden is dit een notatie voor de Engelse taal. Deze notatie is (net als de andere talen) een gecontroleerde natuurlijke taal, oftewel een seminatuurlijke taal waarbij een soort syntaxis een handreiking doet over de manier van gebruik. Tevens wordt geen gebruik gemaakt van het gehele Engelse grammatica maar van een beperkte subset. In SBVR Structured English kunnen alle concepten uit SBVR uitgedrukt worden in natuurlijke taal.

Een andere notatie is RuleSpeak, opgesteld door Ronald G. Ross. Deze notatie was al opgesteld ruim voordat SBVR was geïntroduceerd. In 1996 al was het beschikbaar in het Engels. RuleSpeak sluit meer dan SBVR Structured English aan bij de taal van de werkvloer doordat het eenvoudiger leesbaar is en wat meer vrijheden toestaat. Deze leesbaarheid heeft meerdere oorzaken.

- RuleSpeak gebruikt een infix notatie. SBVR Structured English daarentegen gebruikt een prefix notatie. Door gebruik te maken van een infix notatie komen de zinnen natuurlijker over. Een voorbeeld is:  
*Het is verplicht dat een klant minimaal twee kaartjes koopt.* (SBVR Structured English)  
*Een klant moet minimaal twee kaartjes kopen* (RuleSpeak)
- RuleSpeak staat het gebruik van alternatieve sleutelwoorden toe, als de regel zich daartoe leent. Hierdoor kan afhankelijk van de regel gekozen worden om een woord te kiezen dat beter leesbaar is. SBVR Structured English geeft daarentegen geen mogelijkheden tot het gebruik van alternatieven. Daardoor komen deze zinnen soms gekunsteld over.
- RuleSpeak is specifiek gericht op de taal van de werkvloer. RuleSpeak werd al gebruikt voordat er sprake was van SBVR en is in meerdere project toegepast. Hierdoor heeft het zijn waarde al bewezen als 'taal voor de organisatie'.

Een ander voordeel van RuleSpeak is dat het inmiddels in vier verschillende talen (Engels, Duits, Spaans en Nederlands) beschikbaar is. Dit komt het draagvlak voor deze notatie uiteraard ten goede.

In principe is er een 1-op-1 transformatie mogelijk tussen RuleSpeak en SBVR Structured English, ook al bestaat de beschrijving van deze transformatie op dit moment nog niet. De keuze voor een van beide talen is eigenlijk nooit definitief, maar moet wel consistent gebruikt worden. In de RuleXpress-editor zijn al enkele van de handreikingen van RuleSpeak geïmplementeerd als kwaliteitscontrole. In de editor die voor dit project is ontwikkeld, wordt daarom ook gebruik gemaakt van RuleSpeak.



## 2.3 HET TOTAALPLAATJE

Tot voor kort was het lastig om de gebruikers binnen een organisatie te laten communiceren met de IT en vice versa, terwijl hier grote behoefte aan was. Het liefst zou de IT willen dat de gebruikers de specificatie in (semi-)formele logica aanlevert. Dit mag echter niet worden verwacht van deze gebruikers. Een persoon uit een willekeurige organisatie zou niet weten waar je het over hebt als er wordt gevraagd om specificaties door middel van kwantificaties, conjuncties, disjuncties en logische operatoren, terwijl dit toch allemaal onderdelen zijn van een goede bedrijfsregel. En de informele bedrijfsregel zal in veel gevallen echter wel kunnen worden opgesteld.

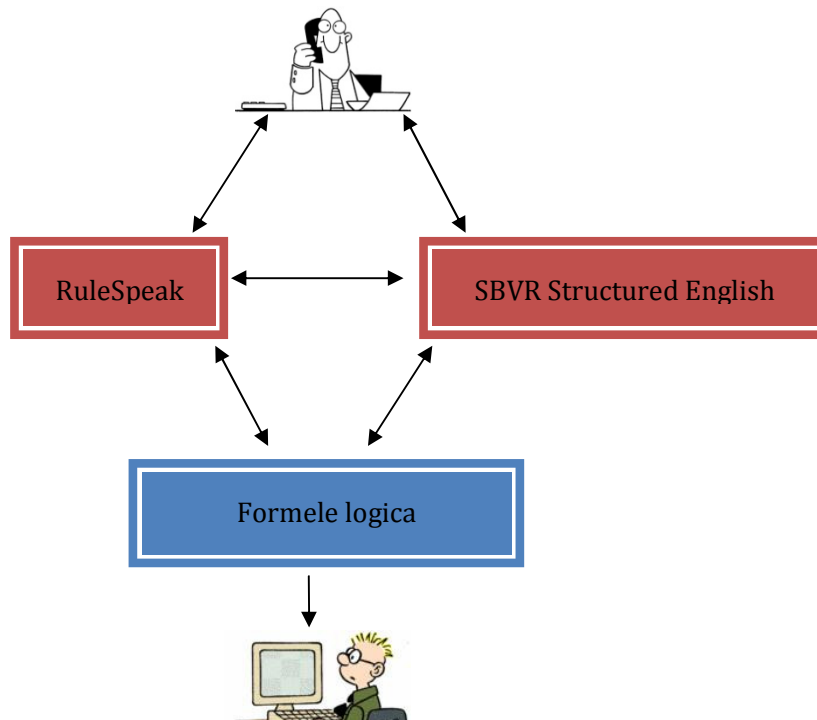
Door RuleSpeak te gebruiken wordt er een beperkt en gestructureerd vocabulaire gebruikt die vervolgens kan worden omgezet in een formelere notatie. Een voorbeeld zou de volgende regel kunnen zijn;

*“Het is verplicht dat een klant minimaal twee kaartjes bestelt”*

Deze regel is leesbaar (en hopelijk op te stellen) door de gebruiker. Deze zin kan vervolgens geformaliseerd worden tot;

*“Voor alle klanten  $K$  geldt, er is een product  $Y$  waarvoor geldt, als  $K$   $Y$  koopt, dan moet het aantal voor  $Y$  minimaal 2 zijn.”*

Deze koppeling is in theorie 1-op-1 te maken, waardoor er geen interpretatie meer nodig is om van de natuurlijke taal naar de formele taal te komen.



Figuur 2 Een mogelijke manier van communiceren tussen het bedrijfsleven en de IT-afdeling

Hierbij is het mogelijk om door middel van gestructureerde stappen van de taal van de werkvloer naar de IT te komen. Helaas is deze schets in figuur 2 nogal kort door de bocht, maar in theorie zouden deze stappen gemaakt kunnen worden (Spreeuwenberg & Healy, 2009). De eerste stap in dit proces is om er voor te zorgen dat de organisatie haar wensen RuleSpeak-conform specificieert. In deze scriptie zal ook op deze stap worden ingegaan omdat dit de eerste en meest belangrijke stap is in het gehele proces. Als de output van dit proces een verkeerd resultaat oplevert, zal het vervolg ook problemen ondervinden.

## 2.4 DE MENSEN

---

Tot nu toe is er telkens geschreven over het “gebruikers” en de “organisatie”. Het liefst zouden we deze groep over één kam scheren en een eenduidig profiel maken. Helaas zit het leven niet zo simpel in elkaar. De mensen die de regels binnen een organisatie opstellen en beheren zijn van diverse plumeau en ook de functies waarin men met bedrijfsregels te maken krijgt zijn uiteenlopend. Betrokken zijn bijvoorbeeld de rollen van bedrijfsjurist, functioneel ontwerper of informatieanalist.

Deze groep heeft echter een aantal belangrijke gemeenschappelijke eigenschappen die van belang zijn voor het opstellen van bedrijfsregels. Ten eerste zijn zij de domein experts. Zij weten hoe de wereld in elkaar zit, hoe de organisatie werkt, welke regels er zijn en kennen deze regels door en door. Het is aan hen om deze regels duidelijk te omschrijven en op te schrijven.

Hierin schuilt direct een andere gemeenschappelijke eigenschap, waarin de noodzaak voor een taal als RuleSpeak is te vinden. Dat is namelijk het feit dat zij moeite hebben met het duidelijk en eenduidig te expliciteren van deze regels. Verschillende redenen liggen hier aan ten grondslag, maar er zijn enkele belangrijke redenen welke problemen opleveren.

De belangrijkste is wel dat de meeste mensen niet geschoold zijn in formele logica en hierdoor de gevolgen van hun redeneringen vaak niet overzien. Hierdoor kunnen er regels worden opgesteld, waarbij de gevolgen niet precies bedoelen wat ze willen. In 1972 hebben Wason en Johnson-Laird het Wason Card-probleem bedacht. Het probleem is als volgt;

- Er zijn vier kaarten
- Aan de ene kant van de kaart staat een letter
- Aan de andere kant van de kaart staat een cijfer
- Je krijgt vier kaarten, namelijk



Vervolgens wordt er gevraagd om een redenering voor deze vier kaarten te controleren door zo min mogelijk kaarten om te draaien. De regel is: “Als op een kaart een klinker staat, moet er een even nummer op de andere zijde staan.”

Het experiment werd gehouden met 128 universitair geschoolde mensen en maar 5% wist het correcte antwoord (A en 7) te geven. (Wason & Johnson-Laird, 1972). Uit dit onderzoek blijkt dat logisch redeneren niet een basiseigenschap is van de mens, terwijl dit voor IT-programmeurs juist een belangrijke eigenschap is. Ondanks dat bovenstaande verklaring voor de meeste mensen eenvoudig te “begrijpen” is, overziet 95% niet wat de gevolgen hiervan zijn.

Een andere reden is dat verschillende (soorten) mensen niet dezelfde woorden gebruiken om hetzelfde uit te drukken. Zo zijn er verschillende woorden te gebruiken om bijvoorbeeld een verplichting vast te leggen. Door mensen te dwingen gebruik te maken van een gecontroleerde natuurlijke taal, worden mensen gedwongen om, weliswaar onbewust, de bedrijfsregels in een enigszins geformaliseerde vorm te noteren. Men gaat dezelfde woorden gebruiken om dezelfde ‘zwaarte’ van een verplichting vast te leggen, en gaat dezelfde woorden gebruiken voor dezelfde concepten.

---

## 2.5 HET EXPLICITEREN VAN DE BEDRIJFSREGELS

---

In de vorige paragrafen is beschreven wat het nut is van bedrijfsregels en wat belangrijke onderdelen zijn in bedrijfsregels. Hier wordt kort ingegaan op het proces van het expliciteren van bedrijfsregels.

Er kunnen grofweg twee processen worden onderscheiden omtrent het opstellen van bedrijfsregels. Dat zijn een vorm van constructie, waarbij er nog niets op papier staat en de regels vanuit “niets” worden opgesteld, en een vorm van reconstructie, waarbij er al regels op papier staan maar deze eerst worden ontleed en vervolgens in verbeterde vorm weer worden opgesteld.

Het is belangrijk om te zien dat het proces rondom bedrijfsregels veelal een continu proces is. Telkens zullen er veranderingen plaatsvinden waardoor regels moeten worden aangepast. Deze veranderingen kunnen intern worden geïnitieerd door middel van een verandering van bijvoorbeeld het beleid, of extern doordat bijvoorbeeld de wet wordt gewijzigd. Een set met regels is eigenlijk nooit af.

---

### 2.5.1 CONSTRUCTIE

---

Zoals al eerder in het hoofdstuk genoemd, worden regels meestal niet verzonnen, maar bestaan ze vaak al lang voordat er over na is gedacht. In sommige bedrijven liggen deze regels vast in programmacode of in wet- en regelgeving. Soms liggen ze alleen vast in de hoofden van de medewerkers. Het probleem is dat deze regels uit de programmacode en uit de hoofden van medewerkers moet worden gehaald en moet worden vertaald naar heldere regels op een centrale plek. Dit is typisch de taak van specialisten, die de juiste vragen weten te stellen en op deze manier de gaten, de uitzonderingen en manco's in en op een regel weten aan te wijzen.

---

### 2.5.2 RECONSTRUCTIE

---

Een ander geval kan zijn dat de regels al op papier staan, maar dat de set van regels incompleet, ambigu en niet duidelijk beschreven staan. In zo'n geval moet er een kwaliteitslag gemaakt worden, om de regels op een betere manier vast te leggen.

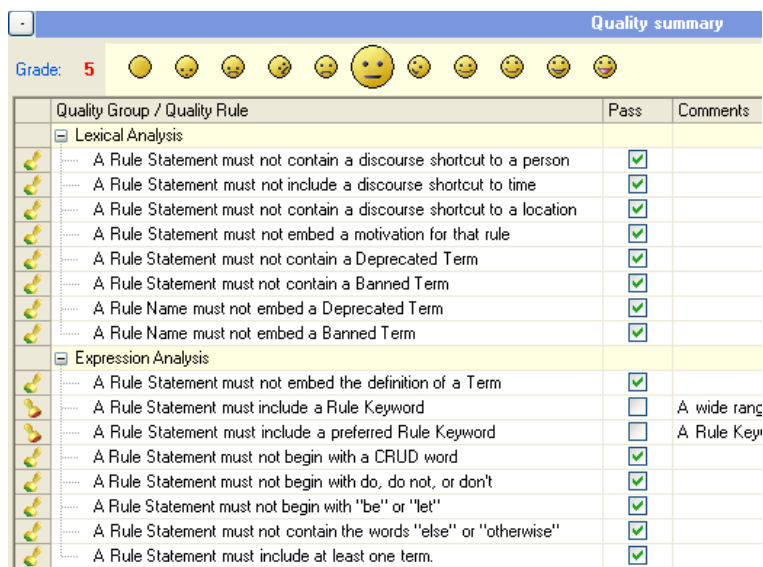
Op zo'n moment moet een regel uit elkaar worden gehaald om vervolgens op een betere manier weer in elkaar te worden gezet. Dit zal gedaan worden nadat een gebruiker meer kennis heeft verworven, of een gebruiker aanwijzingen heeft gekregen van een bedrijfsregel-expert. De gebruiker kan op dat moment opnieuw nadenken over de regel. Is de regel compleet? Zijn alle benodigde termen gedefinieerd? Zijn de voorwaarden goed opgesteld?

Een manier waarop dit proces kan worden ondersteund is om de regel eerst af te laten breken tot de bouwstenen en vervolgens de gebruiker in staat te stellen om met deze bouwstenen de zin opnieuw, maar beter, in elkaar te zetten.

### 2.5.3 KWALITEIT VAN DE REGELS

Bij zowel de constructie als reconstructie van regels, moet er voor gezorgd worden dat er kwalitatief hoogstaande regels worden gevormd. De vraag die dan direct opkomt is uiteraard wat een regel een kwalitatief goede regel maakt.

Een manier die RuleXpress hiervoor gebruikt, is het geven van kwaliteitsfeedback aan de gebruiker aan de hand van de richtlijnen uit RuleSpeak (zie figuur 3). Door middel van een zogenaamde smiley en een cijfer wordt aangegeven hoe goed een regel is. Dit wordt bepaald door te kijken naar enkele kwaliteitscriteria uit RuleSpeak. Het missen van een kwaliteitscriterium levert strafpunten op die van het cijfer worden afgehaald.



Quality Group / Quality Rule	Pass	Comments
<b>Lexical Analysis</b>		
A Rule Statement must not contain a discourse shortcut to a person	<input checked="" type="checkbox"/>	
A Rule Statement must not include a discourse shortcut to time	<input checked="" type="checkbox"/>	
A Rule Statement must not contain a discourse shortcut to a location	<input checked="" type="checkbox"/>	
A Rule Statement must not embed a motivation for that rule	<input checked="" type="checkbox"/>	
A Rule Statement must not contain a Deprecated Term	<input checked="" type="checkbox"/>	
A Rule Statement must not contain a Banned Term	<input checked="" type="checkbox"/>	
A Rule Name must not embed a Deprecated Term	<input checked="" type="checkbox"/>	
A Rule Name must not embed a Banned Term	<input checked="" type="checkbox"/>	
<b>Expression Analysis</b>		
A Rule Statement must not embed the definition of a Term	<input checked="" type="checkbox"/>	
A Rule Statement must include a Rule Keyword	<input type="checkbox"/>	A wide rang
A Rule Statement must include a preferred Rule Keyword	<input type="checkbox"/>	A Rule Key
A Rule Statement must not begin with a CRUD word	<input checked="" type="checkbox"/>	
A Rule Statement must not begin with do, do not, or don't	<input checked="" type="checkbox"/>	
A Rule Statement must not begin with "be" or "let"	<input checked="" type="checkbox"/>	
A Rule Statement must not contain the words "else" or "otherwise"	<input checked="" type="checkbox"/>	
A Rule Statement must include at least one term.	<input checked="" type="checkbox"/>	

Figuur 3 De kwaliteitsbeoordeling in RuleXpress

In RuleXpress zijn lang niet alle richtlijnen van RuleSpeak geïmplementeerd. De kwaliteitscontrole geeft echter wel een goede indicatie voor de kwaliteit van een regel. Sommige bedrijven verplichten de opstellers van regels ook dat hun regelset een minimale score moeten hebben voordat deze mogen worden overgedragen aan de IT.

In deze scriptie zullen we ons niet verder verdiepen in het scoren van de kwaliteit van een regel. Om tot een kwalitatief goede regel te komen, zal RuleSpeak als leidraad worden aangehouden.

---

#### 2.5.4 DYNAMISCH PROCES

---

Het opstellen van bedrijfsregels is een zeer dynamisch proces. De uitgangssituatie verschilt van organisatie tot organisatie. Bij sommige organisaties zijn er al regels op papier aanwezig en bij sommige nog helemaal niet. Bij sommige organisaties zijn de concepten al gedefinieerd door gemeenschappelijke termen en bij andere organisaties is er nog geen afspraak gemaakt over de definities van termen.

Een ander probleem is dat het opstellen van regels nauw samenhangt met het definiëren van nieuwe termen en feittypen. Als een gebruiker een nieuwe regel opstelt, zullen er misschien nieuwe termen of feittypen worden geïntroduceerd die nog niet zijn gedefinieerd. Dit zal dan moeten gebeuren nadat de regel is opgesteld. Op deze manier worden door het opstellen van regels, het feittypemodel en het vocabulaire uitgebreid.

Een andere manier zou kunnen zijn dat een organisatie eerst zijn vocabulaire en feittypemodel definieert en aan de hand hiervan regels gaat opstellen. Als een feittype of term namelijk niet wordt gebruikt in regels, heeft dat feittype of die term echter geen toegevoegde waarde en zou deze buiten het model moeten worden gehouden.

Dit introduceert een kip-en-ei-verhaal rondom het opstellen van een goede set bedrijfsregels. Waar wordt mee begonnen, het opstellen van een vocabulaire, een feittypemodel of de regels. Alle drie de onderdelen grijpen op elkaar in en kunnen onafhankelijk van elkaar worden aangemaakt, maar hebben een zeer nauwe relatie met elkaar.

### 3. DE EDITOR

---

Een belangrijke rol in het vastleggen van de bedrijfsregels is weggelegd voor een editor. Een editor is eigenlijk niets meer dan een programma waarmee je content kunt creëren en wijzigen. Voorbeelden van simpele editors zijn kladblok en Word. Er zijn echter ook specialistischere editors die hulp kunnen bieden bij het proces waar je mee bezig bent. Zo zijn er bijvoorbeeld editors voor programmeeromgevingen, die kunnen controleren of de ingetoetste content voldoet aan de syntaxis van de programmeertaal.

Echter, zoals eerder geschreven is worden bedrijfsregels typisch opgesteld in natuurlijke taal, en om daarvan de syntaxis te controleren is haast ondoenlijk. Laat staan de semantiek! Toch zijn er mogelijkheden genoeg om een editor een belangrijke rol te laten spelen in het proces. Een voorbeeld hiervan is in het vorige hoofdstuk besproken bij de editor van RuleXpress. Hier wordt een kwaliteitscontrole uitgevoerd, door te kijken naar het voorkomen of ontbreken van bepaalde woorden in een regel.

#### 3.1 EDITORS VOOR BEDRIJFSREGELS

---

In de vorige hoofdstukken is beschreven wat bedrijfsregels zijn, waarom bedrijfsregels nuttig zijn en is er kennis gemaakt met RuleSpeak richtlijnen voor het opstellen van goede bedrijfsregels. In dit hoofdstuk wordt kort beschreven welke soorten editors er beschikbaar zijn. Daarnaast zal worden gekeken naar RuleXpress, de editor die het uitgangspunt is in deze scriptie, en naar de verbeterde editor zoals die wordt voorgesteld in deze scriptie.

---

##### 3.1.1 SOORTEN EDITORS

---

In het landschap van business rules editors zijn er grofweg twee soorten te onderscheiden; editor met een Business Rules Management Suite en editors met een Business Rules Engine.

##### **Business Rules Management**

Een business rules management suite (BRMS) ondersteunt de gebruiker in het vastleggen, onderhouden en beheren van bedrijfsregels. De nadruk ligt op de ondersteuning van deze processen.

##### **Business Rules Engine**

Een Business Rule Engine (BRE) is software die kan redeneren en in staat is om ingevoerde gegevens te evalueren tegen de geldende regels. (Coenen, Hermans, Roosmalen, & Spreeuwenberg, 2008) Een software pakket dat een BRE bevat is in staat om bijvoorbeeld hypotheekaanvragen automatisch te evalueren of om een medische aandoening te diagnosticeren. De BRE zorgt er voor dat regels ook daadwerkelijk geëxecuteerd worden.

---

### 3.2 RULEXPRESS

---

RuleXpress is een softwaretool voor het opstellen, analyseren en beheren van bedrijfsregels. De tool is gemaakt door het Amerikaanse bedrijf RuleArts. De tool is ontwikkeld met het credo van het Business Rules Manifesto in het achterhoofd; “*Laat deskundigen uit de bedrijfspraktijk zelf de regels opstellen*”. De tool biedt daarom een gebruiksvriendelijke omgeving om regels en het bijbehorende vocabulaire vast te leggen en te onderhouden.

Dit is niet een eenmalig proces, maar een doorlopend verbeterproces waarbij ‘communicatie’ het toverwoord is. Enkel door te communiceren met elkaar kunnen regels en definities eenduidig geëxpliciteerd worden.

Bedrijfsregels worden in natuurlijke taal opgesteld, waardoor de gebruiker alle vrijheid heeft om de regels te verwoorden op de manier die haar goedgeeft. RuleXpress laat de gebruiker dan ook volledig vrij in het uitdrukken van de regels. Deze vrijheid heeft echter ook nadelen, zoals we hebben gezien. De kwaliteit van de regels kan nogal verschillen, doordat er geen regels voor de syntaxis zijn.

RuleXpress probeert de kwaliteit van de regels te waarborgen door achteraf de mogelijkheid tot een kwaliteitscontrole op de regels te bieden. Door middel van deze kwaliteitscontrole, krijgt een gebruiker inzicht in de kwaliteit van de regel. Daarnaast biedt de tool aanwijzingen om de kwaliteit te verbeteren.

Verder biedt de tool handvatten om het dynamische, iteratieve proces van het opstellen goed te ondersteunen. Als er regels worden opgesteld kunnen er direct nieuwe termen en feittypen worden gedefinieerd. Maar tijdens het opstellen van een feittype kan er ook direct een nieuwe regel worden toegevoegd. De gebruiker wordt vrijgelaten in de manier waarop hij de regelset wil opbouwen.



### 3.3 HET PROCES

---

Het proces van vastleggen van bedrijfsregels zal variëren naarmate er een keuze voor een andere rol voor de editor wordt gemaakt. Een editor kan je bijvoorbeeld de facilititeit bieden om regels vast te leggen en op te slaan. Echter een geavanceerdere editor zal je ook bij kunnen staan in het proces om te komen tot een bedrijfsregel. Tijdens en na het formuleren van de bedrijfsregel kan de editor bijvoorbeeld tips geven die kunnen bijdragen in het proces. Hierbij moet de editor dus een andere rol vervullen en zal het proces van het opstellen van de regels er anders uitzien. In het laatste geval is er namelijk sprake van een getriggerde iteratie, terwijl deze in het eerste geval niet veroorzaakt zal worden.

Om de vergelijking tussen verschillende rollen te trekken, zal er gekeken worden naar enkele aspecten van het proces en de output van het proces (Selker, 2005) (Kletke, Mackay, Barr, & Jones, 2000);

- Kwaliteit van de output. Zoals al eerder beschreven is, is de kwaliteit van een bedrijfsregel lastig te beschrijven. Echter op grond van het proces kunnen er wel daadwerkelijke aanwijzingen aanwezig zijn dat een rol een betere kwaliteit zal opleveren dan een andere rol.
- Vrijheid en creativiteit. Bedrijfsregels worden in natuurlijke taal opgesteld, echter er zijn wel enkele richtlijnen om betere regels op te stellen. Door de editor hierop te laten inspelen kan de vrijheid in het proces (en van de gebruiker) worden beperkt. Dit heeft uiteraard dan ook directe gevolgen voor de creativiteit binnen het proces. Als het proces in een strak keurslijf wordt gegoten, is er weinig ruimte meer voor creativiteit. Terwijl een goede regel vaak ook het resultaat is van een stukje creativiteit. (Spreeuwenberg, Rule Authoring Is a Creative Process, 2009)
- Repetitiviteit. Een editor kan in een bepaalde rol een gebruiker vragen om steeds hetzelfde proces te doorlopen. In dat geval zal er sprake zijn van zeer veel repetitiviteit. Het gevolg is dat de aandacht van een gebruiker op een gegeven moment zal verslappen en dit ten koste van bijvoorbeeld de kwaliteit van de output zal gaan.
- Fasering. In een eerder hoofdstuk is al duidelijk het onderscheid gemaakt tussen de constructie en reconstructie van een regel. Bij de constructie wordt uitgegaan van een nog niet geëxpliciteerde regel, terwijl bij de reconstructie er wordt uitgegaan van een bestaande regel om over deze regel een verbeterslag heen te halen. Het kan per rol verschillen wanneer deze wordt ingezet.

---

### 3.4 DE GEBRUIKER

---

De keuze voor de rol die een editor speelt zal ook verschillende eisen stellen aan de gebruikers en verschillende dingen vragen van de gebruikers. Voor elke rol die een editor kan spelen zal bekeken moeten worden wat er in deze rol verwacht wordt van de gebruiker.

Ten eerste zullen we kijken naar de eisen die aan de gebruiker worden opgelegd, of simpel gezegd; "*Wat moet de gebruiker kunnen*". Een opsteller van bedrijfsregels zal aan verschillende criteria moeten voldoen;

- Ervaring met bedrijfsregels. Verschillende rollen zullen in meer of mindere mate eisen stellen aan de ervaring van de gebruiker. Dit hangt uiteraard samen met de vrijheid die beschikbaar is binnen het proces. Als de rol van de editor de gebruiker volledig vrij laat om zijn regels op te stellen, zal de gebruiker al een hoop ervaring moeten hebben met het opstellen van bedrijfsregel. Als dat niet het geval is, zal dit hoogstwaarschijnlijk leiden tot regels van lage kwaliteit.
- Ervaring in formaliseren. Indien een gebruiker weinig tot geen ervaring heeft in het opstellen van een formalisatie zal de editor hierbij ondersteuning kunnen bieden. Dit verschilt echter per rol in welke mate deze ondersteuning aanwezig is. Een editor kan bijvoorbeeld een gebruiker verplichten om specifieke (sleutel)woorden of frases te gebruiken waardoor de formaliseerbaarheid van een bedrijfsregel wordt verhoogd.
- Domeinkennis. Een gebruiker dient kennis te hebben van het domein waarvoor de bedrijfsregels worden opgesteld. Een editor kan niet snel bijdragen aan deze specifieke domeinkennis en daarom zal in deze scriptie deze factor als een constante worden beschouwd en niet per rol uiteen worden gezet.

Daarnaast heeft de gebruiker natuurlijk ook wensen en eisen aan de editor, of simpel gezegd; *Wat wil de gebruiker*. Ook hier speelt de editor een belangrijke rol. De criteria waaraan elke rol zal worden getoetst zijn de volgende;

- Mogelijkheden. Een gebruiker zal niet altijd op dezelfde manier zijn doel willen bereiken. Het ontstaan van een bedrijfsregel is een creatief proces, waarbij meerdere wegen leiden naar Rome. Hierdoor wil een gebruiker niet altijd vastzitten in hetzelfde patroon om tot zijn doel te komen. De editor kan hierin een belangrijke rol spelen.

### 3.5 VERSCHILLENDE ROLLEN

---

De computer kan een belangrijkere rol vervullen in het proces van het opstellen van bedrijfsregels dan enkel de faciliterende rol. Lubart dichtte de computer al een viertal rollen toe (Lubart, 2005), namelijk de computer

- Als oppasser
- Als schrijfmaatje
- Als coach
- Als collega

Echter Lubart kijkt voornamelijk naar de rol van de gehele computer als faciliterend hulpmiddel. Een editor is echter meer een specifiek hulpmiddel, welke een stap verder kan gaan in de ondersteuning. In deze scriptie zijn vier rollen uiteen gezet, welke vervuld kunnen worden door een editor. Hierbij zal gekeken worden wat de invloed is op het proces van het opstellen van bedrijfsregels, wat de mogelijke invloed is op de kwaliteit van de regels en wat de invloed is op de gebruiker.

---

#### 3.5.1 FACILITERENDE ROL

---

De meest eenvoudige rol van de editor is de faciliterende rol. Deze rol wordt sedert vele jaren geïmplementeerd en is ook het meest eenvoudig te implementeren. Deze rol is eigenlijk de vervanging van de pen en het papier en biedt verder weinig specifieke ondersteuning in het proces van het opstellen van bedrijfsregels. In een faciliterende rol biedt de computer de ondersteuning aan de gebruiker om content vast te leggen en te bewaren.

Het is wel mogelijk dat er algemene functionaliteit wordt ondersteund. Zo kan de editor zorgen dat ingevoerde regels automatisch worden opgeslagen of een bestand iedere vijf minuten wordt opgeslagen. Daarnaast kan de editor ondersteuning bieden in het visueel overzichtelijk maken van bedrijfsregels en bijvoorbeeld feittypen.

---

#### HET PROCES

---

Zoals al geschreven is het proces vergelijkbaar met het opstellen van bedrijfsregels met behulp van pen en papier.

##### **Kwaliteit van de output**

In dit proces zal de kwaliteit van de output volledig afhankelijk zijn van de gebruiker. De editor biedt geen ondersteuning met betrekking tot de inhoud en de uiteindelijke kwaliteit van de uitkomst van het proces.

##### **Vrijheid en creativiteit**

De vrijheid wordt op vrijwel geen enkele manier beperkt als de editor enkel als faciliterend optreedt. De inspiratie en ervaring van de gebruiker zal bepalend zijn voor het verloop van het proces en de uitkomst ervan. Het is volledig aan de gebruiker om tot een bedrijfsregel te komen en de editor dwingt de gebruiker in geen enkele richting.

**Repetitiviteit**

Doordat de vrijheid totaal niet wordt beperkt is de kans op herhaling van telkens hetzelfde proces niet erg groot. Elke regel kan door middel van een verschillende worden beschreven. Er is geen vast verloop voor het proces.

**Fasering**

Deze rol kan de editor in alle fasen van het opstellen en beheren van bedrijfsregels spelen. Zowel in de ontwerp-, de bedenken- als de beheersfase zal deze rol van dienst kunnen zijn. Het is dus niet vereist dat de bedrijfsregel al is opgesteld, voordat deze rol aan de slag kan.

---

**DE GEBRUIKER**

---

**Ervaring**

De gebruiker zal volledig bepalen wat de uiteindelijke uitkomst van het proces is. Daarom zal het ook vereist zijn dat dit wordt gedaan door een gebruiker die zowel ervaring heeft met het opstellen van bedrijfsregels, als ervaring heeft in formaliseren. De gebruiker moet zichzelf bewust moeten zijn van de beslissingen die hij zal moeten nemen om tot het uiteindelijke resultaat te komen. Van een ervaren modelleur mag worden verwacht dat hij verschillende stappen al zo snel in zijn hoofd doet, dat deze niet geëxpliciteerd hoeven te worden op papier, maar voor de minder ervaren gebruiker zal deze opgave problemen opleveren.

**Mogelijkheden**

Alle mogelijkheden voor een gebruiker zijn vrij. De gebruiker bepaalt zelf hoe hij het opstellen van de bedrijfsregel aanpakt. De editor geeft de gebruiker daarin geen enkele sturing.

---

### 3.5.2 ONDERSTEUNENDE ROL

---

Deze rol biedt de gebruiker al meer ondersteuning doordat het een specifieke functionaliteit biedt die de gebruiker helpt bij het opstellen van bedrijfsregels. Deze ondersteuning zal voornamelijk procesinhoudelijk van aard zijn. De gebruiker wordt door middel van een soort van geschakelde invuloefeningen op weg geholpen naar de uiteindelijke regel.

In het vorige hoofdstuk is dieper ingegaan op het proces van het opstellen van regels. Hierin is aangestipt dat een regel niet in één keer ontstaat, maar dat een uiteindelijke bedrijfsregel vaak het product is van een reeks van tussenstappen (Hoppenbrouwers, Bommel, & Järvinen, 2008). Een implementatie van een ondersteunende rol kan juist bijdragen aan het onderkennen van deze tussenstappen. In elk van de tussenstappen kan een ander aspect van de bedrijfsregel aan bod komen.

Een voorbeeld van de implementatie van een dergelijk proces zou als volgt kunnen gaan;

- Benoem de termen die relevant zijn voor de bedrijfsregel;
- Benoem het type bedrijfsregel dat vastgelegd gaat worden (een verplichting, iets is niet toegestaan, of iets is juist toegestaan, etc.);
- Benoem of er voorwaarden aan de regel verbonden zijn en zo ja, welke.

Door al deze elementen van een goede bedrijfsregel expliciet te bevragen, wordt er van de gebruiker geëist dat hij bewust nadenkt over elke element. Een ervaren modelleur zal deze stappen snel in zijn hoofd maken, maar voor de minder ervaren gebruiker is het juist erg belangrijk om deze stappen expliciet en afzonderlijk te maken.

Tenslotte zijn de verschillende elementen van de uiteindelijke bedrijfsregel al ingevuld. Deze zijn ter beschikking gesteld aan de gebruiker. Deze moet hier vervolgens nog een correcte regel in natuurlijke taal van maken. Deze laatste stap kan in meer of mindere mate worden ondersteund. Er kan van de gebruiker expliciet worden verwacht dat hij minimaal alle bouwstenen gebruikt en de zinsvolgorde hierbij ook al wordt aangegeven. Een andere manier zou zijn om dit volledig aan de gebruiker over te laten; hij heeft de elementen en hij is zelf verantwoordelijk in welke mate hij deze gebruikt.

---

### HET PROCES

---

#### **Kwaliteit van de output**

Door de gebruiker bewust na te laten denken over de belangrijkste elementen in de bedrijfsregel zal er minder snel een denkstap in het proces worden overgeslagen. Door als het ware een kladblaadje te hebben waarop deze elementen staan, zal het vormen van de uiteindelijke regel eenvoudiger worden. Aangenomen mag worden dat doordat de gebruiker bewuster nadenkt over de deelstappen, de kwaliteit van het eindproduct beter zal zijn dan bij een faciliterende rol. Tevens wordt de gebruiker niet afgeleid door 'dingen' die onthouden moeten worden. Deze staan genoteerd op het 'kladblaadje' en de gebruiker wordt hier tijdelijk niet door gestoord. (Selker T. , 2005)

### **Vrijheid en creativiteit**

De vrijheid van de gebruiker zal in deze rol enigszins worden beperkt, doordat er gevraagd wordt om expliciet door verschillende deelprocessen heen te lopen. De uiteindelijke beperking is wel afhankelijk van hoe strikt deze processtappen worden afgedwongen. Als er volgens een vast proces wordt gewerkt zal de vrijheid sterker worden beperkt, dan wanneer de gebruiker vrij staat om bepaalde bouwstenen wel of niet te gebruiken en zelf te bepalen in welke volgorde de processen worden doorlopen.

### **Repetitiviteit**

De kans op het herhalen van telkens dezelfde deelprocessen is veel groter dan bij de andere rollen. Elke maal zullen dezelfde deelprocessen moeten worden doorlopen. Voor sommige regels (die veel op elkaar lijken) zal deze exercitie als herhalend en saai worden ervaren en misschien zelfs overbodig zijn voor minder ervaren gebruikers. Het proces in deze rol is vrij statisch en beperkt en daardoor is de kans op repetitiviteit erg hoog.

### **Fasering**

De ondersteunende rol zal voornamelijk gebruikt worden voordat een regel is opgesteld. In de fase waarin wordt bedacht hoe de regel het best geformuleerd kan worden, kan deze rol van belang zijn. Op dat moment moeten er keuzes worden gemaakt voor relevante bouwstenen en de editor kan hierbij een ondersteunende rol spelen. Als de bedrijfsregel eenmaal is beschreven zal de gebruiker minder snel nogmaals alle elementen willen onderscheiden die van belang zijn.

---

## DE GEBRUIKER

---

### **Ervaring**

Doordat de editor de gebruiker aan de hand neemt langs alle deelprocessen zal de ervaring van de gebruiker met bedrijfsregels minder van belang zijn. De editor borgt als het ware dat alle deelprocessen worden doorlopen. Als de gebruiker vervolgens correct alle elementen identificeert, hoeft hij enkel nog de elementen op een juiste manier met elkaar te verbinden. Tevens kan de editor concrete handvatten bieden, in de vorm van sleutelwoorden, die bijdragen aan de formaliseerbaarheid van de bedrijfsregel.

### **Mogelijkheden**

Afhankelijk van hoe de rol precies wordt geïmplementeerd zullen de mogelijkheden veel beperkter zijn dan bij de faciliterende rol. Er wordt verwacht dat de gebruiker alle deelprocessen zal doorlopen. De volgorde waarin deze processen worden doorlopen zal vrij kunnen zijn waardoor hier nog wat gespeeld kan worden met de mogelijkheden om tot het einddoel te komen.

---

### 3.5.3 EEN EVALUERENDE ROL

---

In deze rol zal de editor bekijken wat de gebruiker aan het doen is of heeft gedaan. Hij zal uit de bedrijfsregel extraheren wat hij ziet dat er is gebeurd en zal trachten uit te leggen wat hij ziet dat de gebruiker bedoelt.

Neem als voorbeeld de zin: *“Een klant moet altijd minimaal twee kaartjes bestellen”*

In deze zin kunnen verschillende elementen worden onderscheiden

- De termen *“klant”* en *“kaartje”*.
- Een feittype *“<klant> bestelt <kaartje>”*.
- Een structurele regel *“moet altijd”*.
- Een kwantificatie van *“minimaal twee”*.

Deze elementen kunnen op verschillende manieren worden verkregen, bijvoorbeeld door middel van een lijst met al eerder gedefinieerde termen, een natuurlijke taal parser, etc. Een natuurlijke taal parser is een computerprogramma dat de grammaticale structuur van een invoer volgens een vastgelegde grammatica ontleeft. Dit programma krijgt een natuurlijke taal zin als input en geeft een datastructuur als output met daarin aangegeven welke woorden bijvoorbeeld zelfstandig naamwoord, werkwoord of bijwoord zijn. Op dit moment zal er echter nog geen rekening mee worden gehouden. Op het gebruik van een parser zal verderop in het verslag worden ingegaan als de daadwerkelijke editor wordt beschreven.

Door het geven van deze feedback aan de gebruiker, maakt de editor de gebruiker er van bewust wat hij op dit moment heeft vastgelegd. Zo kan bijvoorbeeld worden aangegeven dat de gebruiker op dit moment een structurele regel heeft vastgelegd. Er kan dan vervolgens door de editor worden gevraagd of de gebruiker zich ervan bewust is dat deze regel nooit aftreden mag worden. Op dat moment wordt de gebruiker aan het denken gezet en zal hij zich beter bewust worden van zijn keuze.

Een andere manier van feedback zou kunnen zijn dat de editor aangeeft dat het feittype-model incompleet is, omdat er een feittype wordt gebruikt dat niet voorkomt in het huidige feittype-model. Dit kan bijvoorbeeld ook gebeuren bij niet gedefinieerde termen.

---

## HET PROCES

---

### **Kwaliteit van de output**

Aangezien de evaluerende rol vooral op basis van een al ingevoerde regel werkt, zal het de bedoeling zijn dat de output van een eerdere rol worden verbeterd door deze rol uit te voeren. Hoogstwaarschijnlijk zal de kwaliteit verbeteren doordat er een reflectie plaats vindt op hetgeen is ingevoerd. Een gebruiker wordt bewust aan het denken gezet over de bedrijfsregel is opgesteld. Door de gebruiker hier nogmaals zelf over te laten nadenken, zal hij zich beter bewust worden van de regel en waar nodig de kwaliteit van de regel kunnen verbeteren.

### **Vrijheid en creativiteit**

De vrijheid van de gebruiker wordt niet beperkt, doordat de editor enkel aanwijzingen probeert te vinden en dit te peilen bij de gebruiker. De gebruiker zal door de vragen aan het denken moeten

worden gezet, maar zal zelf de conclusies moeten interpreteren en eventueel verwerken. De gebruiker wordt in de evaluerende rol dus niet gedwongen om zich iets aan te trekken van de editor.

### **Repetitiviteit**

De kans op repetitiviteit is erg hoog in de evaluerende rol, omdat de editor telkens constateert welke elementen er in een bedrijfsregel aanwezig zijn. Hierdoor moet de gebruiker telkens bekijken of dit inderdaad is wat er bedoeld wordt. Naarmate de regel van hogere kwaliteit wordt, zal deze rol minder toevoegen aan het proces. Op zo'n moment zal de editor in deze rol enkel nog constateren en niet meer bijdragen aan het verbeteren van de bedrijfsregel. Op dat moment is de kans op verslapping van de aandacht erg groot.

### **Fasering**

Deze rol dient te worden ingezet nadat een bedrijfsregel is opgesteld. Aan het begin van het opstellen kan deze rol niet van toegevoegde waarde zijn voor de gebruiker, aangezien er dan simpelweg nog niets valt te evalueren en te constateren.

---

## DE GEBRUIKER

---

### **Ervaring**

De editor neemt een deel van de benodigde ervaring van de gebruiker over. De editor zal zelf aangeven wat hij ziet als termen, feittypen, soorten zinnen en eventuele condities. Op die manier hoeft de gebruiker zich hier minder bewust van te zijn en misschien dit onderscheid niet eens te hoeven kennen om toch te kunnen controleren of een regel correct is. Er zal dus minder ervaring vereist zijn dan in de faciliterende rol.

### **Mogelijkheden**

De gebruiker staat qua mogelijkheden helemaal vrij om te doen en laten wat hij wil. De editor geeft uitsluitend feedback op hetgeen hij kan extraheren uit de bedrijfsregel. Vervolgens geeft de editor alleen de opties om termen of feittypen toe te voegen of te verwijderen uit het model. Of de gebruiker dit inderdaad ook doet, is nog altijd een keuze van de gebruiker zelf.



---

### 3.5.4 COACHENDE ROL

---

De coachende rol is de meeste geavanceerde rol die een editor kan invullen. In deze rol zal de editor proberen als sparringpartner voor de gebruiker op te treden. De editor probeert hierbij aan de hand van een regel aanwijzingen en adviezen te geven aan de gebruiker die kunnen zorgen voor een betere kwaliteit regels. De editor treedt als het ware op als een coach of supervisor.

Voorbeelden van aanwijzingen kunnen zijn:

- Als er “alle” is ingevuld, zal hoogstwaarschijnlijk bedoeld worden “iedere”. *Alle studenten hebben rechten op een basisbeurs*. Hebben alle studenten hierbij recht op één basisbeurs of wordt er bedoeld *“Iedere student heeft recht op een basisbeurs”*.
- Het gebruik van een meervoud zorgt vaak voor verwarring in een regel. De meeste regels zijn ook in een enkelvoudsvorm te noteren.

Een voorbeeld van een goede set met aanwijzingen en best practices is RuleSpeak. Hierin zijn een tal van aanwijzingen te vinden van veelgemaakte fouten die simpel kunnen worden gedetecteerd en worden opgelost. Uiteraard zullen er meerdere aanwijzingen kunnen worden gegeven. Voor een specifieke organisatie zouden er zelfs aanwijzingen kunnen worden gegeven die alleen binnen die organisatie van toepassing zijn.

In de coachende rol zal de editor het meest intelligent moeten zijn aangezien de editor heel specifieke aanwijzingen zal moeten geven. De coachende rol zal vaak samengaan met een evaluerende rol omdat er eerst zal moeten worden gekeken naar wat er staat. Vervolgens kunnen aan de hand van de inhoud van de regel concrete aanwijzingen worden gegeven.

---

### HET PROCES

---

#### **Kwaliteit van de output**

Doordat er gecoacht gaat worden op het gebruik van best practices en er concrete aanwijzingen worden gedaan, zal de kwaliteit van de bedrijfsregels verhoogd worden. Voorwaarde daarbij is uiteraard wel dat de gebruiker begrijpt wat er veranderd moet worden en dit ook daadwerkelijk gebeurt.

#### **Vrijheid en creativiteit**

RuleSpeak bestaat uit een hele set met regels en aanbevelingen, maar uiteindelijk zijn het enkel aanbevelingen. In sommige gevallen kan een regel niet anders worden beschreven, of is het beter om niet de aanwijzing te volgen. De gebruiker heeft in de coachende rol dus zelf nog altijd de vrijheid om te bepalen of hij zich laat sturen door de editor.

#### **Repetitiviteit**

De kans op repetitiviteit is erg laag in de coachende rol. Deze rol geeft enkel aanwijzingen als deze van toepassing zijn. Dit is in tegenstelling tot bijvoorbeeld de evaluerende rol, waarbij enkel wordt geconstateerd wat er in een bedrijfsregel is beschreven. Deze rol geeft directe aanwijzingen die van toepassing zijn op de specifieke bedrijfsregel. Hierdoor wordt telkens de aandacht van de gebruiker gevraagd om de specifieke bedrijfsregel te verbeteren.

### **Fasering**

Deze rol zal voornamelijk worden ingezet als er al een geschreven regel beschikbaar is. Op dat moment kan de editor concrete aanwijzingen doen voor verbeteringen. Er zijn enkele richtlijnen in RuleSpeak die ook voor het opstellen van de bedrijfsregel relevant kunnen zijn. Een voorbeeld is dat een bedrijfsregel vaak met het woord “een” moet beginnen. Deze aanwijzing kan ook al worden gegeven tijdens de constructie van de bedrijfsregel.

---

## DE GEBRUIKER

---

### **Ervaring**

Ook in de coachende rol zal de editor een stukje van de kennis overnemen van de gebruiker. Zonder dat de gebruiker kennis hoeft te hebben van RuleSpeak, zal deze rol een aanvulling zijn bij het opstellen van bedrijfsregels. In deze rol worden er concrete aanwijzingen gegeven aan de gebruiker. De gebruiker zal echter wel wat ervaring en inzicht moeten hebben om de richtlijnen verder te kunnen vertalen naar verbeteringen aan de specifieke bedrijfsregel.

### **Mogelijkheden**

De gebruiker staat vrij om de aanwijzingen op te volgen. Enkele RuleSpeak aanbevelingen geven ook aan dat iets soms niet goed of juist wel goed is. De gebruiker zal dan nog altijd zelf moeten kijken of dit op hem van toepassing is.

---

## 4. RULESPEAK

---

RuleSpeak is een basisset richtlijnen welke oorspronkelijk zijn opgesteld door Ron Ross. De set met richtlijnen moet ondersteuning bieden voor “het helder verwoorden van bedrijfsregels”, “het verbeteren van communicatie tussen mensen uit de bedrijfspraktijk, analisten en IT-ers” en “het behouden van product- en bedrijfskennis in een beheersbaar en herbruikbaar formaat”. (Ross, 2001)

RuleSpeak is een van de drie notaties die opgenomen is in de appendix van de SBVR-specificatie (Object Management Group, 2008). Volgens OMG sluiten de richtlijnen dichter aan bij de bedrijfspraktijk dan de notatie in SBVR Structured English. Dit wordt mede ondersteund door het feit dat RuleSpeak zich al in meerdere projecten in het bedrijfsleven heeft bewezen. Dit komt voornamelijk dat RuleSpeak zich meer aan sluit bij de taal op de bedrijfsvloer. Een voorbeeld in SBVR Structured English is: “Het is verplicht dat *<een klant minimaal twee kaartjes koopt>*”. RuleSpeak lost dat op door het woordje “moet”; “*<een klant> moet <minimaal twee kaartjes kopen>*”. Deze laatste regel zal een regel zijn die eerder op de bedrijfsvloer terug te vinden is dan de eerste regel.

Naast een set van sleutelwoorden, bevatten de RuleSpeak documenten ook een set met richtlijnen voor het opstellen van goede bedrijfsregels. Deze richtlijnen zijn een uitstekende input voor de “intelligentie” van een editor in een coachende rol. Daarom zal er ook verder worden ingegaan op deze richtlijnen en de eventuele automatiseerbaarheid.

---

## 4.1 DE RICHTLIJNEN

---

In deze sectie zullen alle richtlijnen uit RuleSpeak aan bod komen. Per richtlijn zal beschreven worden:

- wat de richtlijn beoogt;
- de gedachte achter de richtlijn;
- een voorbeeld van de toepassing van de richtlijn;
- een manier om de controle van de richtlijn te automatiseren.

Op deze manier kunnen er “kandidaten” gevonden worden om geïmplementeerd te worden in een editor.

### 1. De formulering van een regel moet precies één van de volgende sleutelwoorden bevatten;

- **moet ... of mag niet**
- **alleen ....**
- **etc.**

Door aan een feittype een sleutelwoord toe te voegen, wordt er een regel opgesteld. Zonder dit sleutelwoord zou er geen sprake zijn van een regel, aangezien de vrijheid van het feittype niet beperkt zou worden.

In RuleSpeak zijn er een aantal sleutelwoorden voorgesteld, zodat er duidelijkheid over de lading van een woord en het gebruik van het woord. Een operationele regel kan met talloze woorden worden uitgedrukt, echter RuleSpeak raadt het woord “*moet*” aan en als alternatief het woord “*dient*”.

Een voorbeeld: Neem het feittype “*klant koopt kaartje*”. Door hier een sleutelwoord aan te voegen, zal de lading veranderen en het feittype een regel worden “*een klant moet een kaartje kopen*”.

De controle van deze regel is eenvoudig op te lossen door middel van een eenvoudige zoekoperatie in elke regel op het voorkomen van een sleutelwoord.

### 2. “Kan” is niet goed

Uit de definitie van een bedrijfsregel volgt dat een bedrijfsregel altijd de vrijheid van handelen moet beperken. Het woord “*kan*” geeft echter een vrijblijvendheid aan die in regels niet geaccepteerd dient te worden.

*Het systeem kan twee aanvragen tegelijkertijd verwerken.*

Deze formulering geeft enkel de existentie van de gebeurtenis weer en legt geen beperking van de vrijheid op. Een verbeterde versie zou zijn:

*Het systeem moet twee aanvragen tegelijkertijd kunnen verwerken.*

Een uitzondering geldt uiteraard in het geval van een advies waar wel het woord “*kan*” gebruikt mag worden. Op een dergelijk moment kan er feedback worden gegeven aan de gebruiker om te verifiëren dat een gebruiker daadwerkelijk een advies wil vastleggen en niet een regel.

Deze regel is eenvoudig te automatiseren door het voorkomen van “kan” te bekijken in een bedrijfsregel.

### 3. “Zomaar wat” is niet goed

RuleSpeak schrijft voor dat elke relatie tussen een onderwerp en lijdend voorwerp gebaseerd moeten zijn op een feittype. Als het voorbeeld “*een klant moet een bestelling plaatsen*” wordt bekeken dan zien we het feittype “*klant plaats bestelling*” terugkomen. Dit feittype moeten dan als dusdanig worden onderkend, anders moet de regel veranderd moeten worden.

Hier komt het kip-en-ei verhaal van de termen, regels en feittypen terug. Een gebruiker kan eerst bijvoorbeeld een bedrijfsregel maken. Als er in deze bedrijfsregel een nieuwe feittype wordt gebruikt, moet de gebruiker de optie krijgen om het nieuw feittype snel toe te kunnen voegen aan het feittypemodel. Als dit consequent wordt gedaan, zal het feittype model de regels blijven reflecteren en vice versa.

Het detecteren van feittypen in een regel is een complexe aangelegenheid. Het is zeker mogelijk om feittypen te detecteren, maar daar is wel een krachtige natuurlijke taal parser voor nodig, die feittypen kan extraheren uit een regel.

### 4. Verborgene feittypen zijn niet goed

Kennis dient geëxpliciteerd te worden in regels en feittypen. Er zijn tal van woorden die een relatie geven tussen twee concepten maar niet de lading dekken. Als we de regel “*Een team moet een manager hebben*” bekijken, kunnen daar meerdere interpretaties aan worden gegeven. Wordt hier eigenlijk “geeft leiding aan”, “ondersteunt” of “beoordeelt” bedoeld? Dit is inhoudelijk nogal een verschil en verschillende mensen kunnen deze regel verschillend uitleggen. Het is belangrijk om niet in te algemene woorden een relatie uit te drukken, maar juist werkwoorden en voorzetsels te gebruiken met een duidelijke betekenis.

Het automatiseren van deze regelnaleving is erg complex. Om dat te realiseren zal er meta-informatie over of betekenis van de objecten beschikbaar moeten zijn om “gewicht” aan de woorden te geven. Dit kan namelijk niet automatisch worden afgeleid door middel van een natuurlijke taal parser. Aangezien deze informatie over werkwoorden op dit moment niet aanwezig is, is het niet mogelijk om deze richtlijn te automatiseren.

Een mogelijke oplossing die eenvoudig gerealiseerd kan worden is het bijhouden van een woordenlijst met algemene werkwoorden, zoals bijvoorbeeld het werkwoord “hebben” en “bestaan”. Als één van deze woorden wordt gebruikt, kan er automatisch gevraagd worden of er een specifiek werkwoord gebruikt kan worden. Deze vraag kan hoogstens geopperd worden, aangezien alle bovenstaande gevallen wel een correct feittype zijn en zouden opgenomen kunnen worden in een regel.

### 5. Een onderwerp in meervoud is niet goed en een lijdend voorwerp is soms niet goed

Een regel wordt eigenlijk altijd toegepast op een enkelvoudige instantie van een groep en niet op een gehele groep. Als we kijken naar de regel “bestellingen moeten een status hebben”, wordt eigenlijk bedoeld dat voor elke bestelling geldt dat deze een status moet hebben, oftewel “Een bestelling moet een status hebben”. Als het meervoud zou blijven staan in de regel kan je je afvragen of er bedoeld wordt dat alle bestellingen één status hebben. Waarschijnlijk wordt dat

niet bedoeld. Het is dus altijd fout als het onderwerp in meervoudsvorm staat. Een best practice is dan ook om een regel te beginnen met het woord 'een'. Dit woord zorgt er namelijk voor dat er eigenlijk altijd een enkelvoudsvorm wordt toegepast in het onderwerp.

Bij een lijdend voorwerp kan een meervoudsvorm in een enkel geval wel correct zijn. Een meervoud is namelijk van toepassing als er een kwantificatie wordt gebruikt in een regel, zoals in *"een klant moet minimaal twee kaartjes bestellen"*. In alle andere gevallen is het waarschijnlijk fout.

Het detecteren van een meervoud is relatief eenvoudig te doen met behulp van een natuurlijke taal parser.

## 6. De zin beginnen met "als" is niet goed

IT-ers redeneren graag volgens een "if ... then ... else ..." -patroon, aangezien zij dit veelal gebruiken in programmacode. In het opstellen van regels is dit niet goed, aangezien we regels op declaratieve wijze willen opstellen en niet op een procedurele wijze. In een regel is het dan ook veel beter om te starten met een onderwerp en vervolgens een voorwaarde toe te voegen.

Neem de regel *"Als een medewerker met pensioen is, dan moet hij niet toegewezen worden aan een team"*. Direct is de "if ... then ..." -structuur terug te zien. Deze regel is te herschrijven naar *"Een medewerker mag niet toegewezen worden aan een team indien de medewerker met pensioen is"*. In deze regel wordt begonnen met het onderwerp en vervolgens wordt een voorwaarde gegeven.

Het detecteren van woorden als "wanneer" of "als" aan het begin van de zin is eenvoudig te doen met een zoekactie in de zin. Of er echter wordt begonnen met de kern van de regel is zonder de betekenis van de zin te kennen niet te verkrijgen. Het is eenvoudig te detecteren met een natuurlijke taal parser of er wordt begonnen met een onderwerp, echter of dit onderwerp ook de kern van de zaak is, valt niet te zeggen zonder betekenis.

## 7. Een tijdsbepaling aan het begin van de regel is niet goed

Een richtlijn, die veel weg heeft van de vorige richtlijn, is dat er niet moet worden begonnen met een tijdsbepaling. De regel draait om het onderwerp. De voorwaarden en eventuele tijdsbepalingen kunnen beter achteraan de zin worden geplaatst.

In de regel *"Bij het sluiten van het schooljaar, moet een student minstens twee cursussen hebben afgerond"*, wordt dit niet gedaan. Door een herschrijfactie komt het onderwerp vooraan te staan en staat er *"Een student moet minstens twee cursussen afgerond hebben bij het sluiten van het schooljaar"*.

In dit geval zal er betekenis aan woorden of frases beschikbaar moeten zijn om iets zinnigs hierover te zeggen. Er zou aangegeven moeten kunnen worden of een zinsdeel betrekking heeft op een tijdsbepaling. Deze informatie is niet beschikbaar waardoor het automatisch controleren hiervan niet mogelijk is.

Een alternatief is wederom het werken met woordenlijsten. Als een zin bijvoorbeeld wordt begonnen met de woorden *"Aan het begin van"* of *"Tijdens"*, is dit een goede indicatie dat er begonnen is met een tijdsbepaling. Deze lijst is niet volledig op te stellen, echter kan wel ondersteuning bieden.

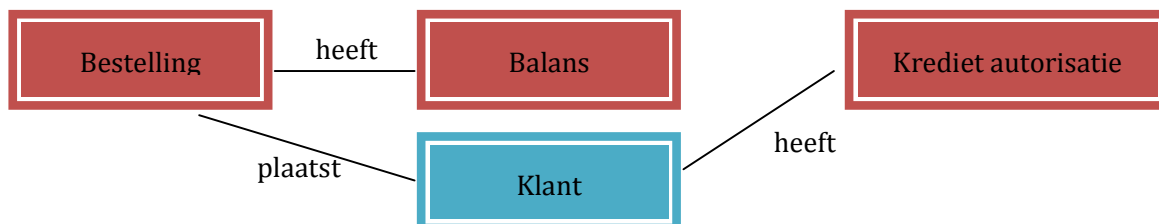
## 8. Overslaan van feittypen is niet goed

In een regel moeten alle feittypen een relatie met elkaar hebben, anders ontbreekt er informatie waarop een bepaald feittype betrekking heeft. Neem als voorbeeld de zin “Een bestelling mag niet afgeleverd worden indien de openstaande balans hoger is dan de krediet autorisatie.” Als we vervolgens het feittypemodel vervolgens zouden maken van deze regel, zou het er uitzien als in figuur 4.



Figuur 4 Een feittypemodel waarbij niet alle feittypen zijn verboden.

Het is duidelijk dat de concepten niet met elkaar te verbinden zijn. Er ontbreekt waarschijnlijk nog een concept om de regel compleet te houden. In dit geval zal dat waarschijnlijk het concept “klant” zijn met de bijbehorende feittypen “klant plaats bestelling” en “klant heeft krediet autorisatie”. In figuur 5 is te zien dat alle feittypen nu onderling verbonden zijn.



Figuur 5 Een verbonden feittypemodel

De regel kan vervolgens aangepast worden tot “Een bestelling mag niet afgeleverd worden indien de openstaande balans op de rekening van de klant die de bestelling heeft geplaatst hoger is dan de kredietautorisatie van de klant.”

Bij een voorgaande richtlijn is al uitgelegd dat het ontdekken van feittypen kan worden ondersteund door een natuurlijke taal parser. In deze groep feittypen is het vervolgens relatief eenvoudig te controleren of alle feittypen verbonden zijn.

## 9. Een persoon als onderwerp is vaak niet goed

Als er een persoon voorkomt als onderwerp in een regel moet er altijd goed nagedacht worden of de persoon wel bedoeld wordt. Neem als voorbeeld de regel “Een klant mag alleen geld opvragen indien de rekening actief is.” De vraag is of het hier wel om de klant gaat, of dat het gaat om het geld dat wordt opgevraagd. Deze regel verbiedt namelijk dat de bank zelf of een gemachtigde van de rekening geld kan opvragen van de rekening.

Een verbeterde versie van deze regel zou zijn “Een geldopvraging van een rekening mag alleen uitgevoerd worden indien de rekening actief is”. In deze regel gaat het om het echte onderwerp van de regel, namelijk de geldopvraging. Het is belangrijk om bij regels met personen als onderwerp nog eens goed na te denken of het onderwerp wel echt een persoon is.

Deze richtlijn is niet af te dwingen, aangezien bijvoorbeeld een natuurlijke taal parser geen informatie kan geven over een woord of dit een persoon betreft. Een alternatief zou kunnen zijn om een woordenlijst op te nemen met veelvoorkomende woorden die duiden op een persoon. Voorbeelden zouden zijn “een klant”, “een adviseur” en “een manager”.

### **10. De gebiedende wijs is niet goed**

Door de gebiedende wijs in een regel op te nemen, neemt een regel de vorm aan van een werkinstructie in plaats van een regel. Neem als voorbeeld de zin

*“Verzend geen bestelling indien de openstaande balans op de rekening van de klant die de bestelling heeft geplaatst hoger is dan de kredietautorisatie”*

Deze zin is duidelijk onderdeel van een procedurebeschrijving. Een bedrijfsregel moet echter declaratief worden opgesteld. Een verbeterde versie van de regel zou zijn:

*“Een bestelling mag niet verzonden worden indien de openstaande balans op de rekening van de klant die de bestelling heeft geplaatst hoger is dan de kredietautorisatie van de bestelling”*

Het detecteren van de gebiedende wijs is niet mogelijk met behulp van een natuurlijke taal parser. Dit komt omdat in het Nederlands en het Engels geen aparte werkwoordsvorm aanwezig is voor de gebiedende wijs (in tegenstelling tot bijvoorbeeld het Latijn). Het is echter wel meestal zo dat bij de gebiedende wijs er een werkwoord op de eerste plaats in de zin staat. Dit kan uiteraard wel gedetecteerd worden met behulp van een natuurlijke taal parser.

### **11. Werkwoorden over een actie zijn niet goed**

Dit is een lastige richtlijn in RuleSpeak. Zoals al een paar keer vermeld mag een regel geen procesbeschrijving geven. Een regel moet een state checken en kijken of hier aan wordt voldaan, maar een regel mag niet beschrijven wat de volgorde van acties is. Bekijk de volgende regel:

*“Een spel mag niet eindigen als de stand gelijk is.”*

Hier wordt een proces beschreven, namelijk de voorwaarden waaronder een spel mag eindigen. Hiervoor zijn bedrijfsregels niet bedoeld. Een verbeterde versie van de regel zou zijn:

*“Een beëindigd spel mag geen gelijke stand hebben.”*

Hier wordt gekeken naar de state van een spel en kan worden bekeken of aan de regel wordt voldaan op elk moment in het systeem.

Het is niet mogelijk om deze richtlijn te automatiseren. Ook al zou er meta-informatie of de betekenis van een woord beschikbaar zijn of een werkwoord gaat over een actie, dan nog zijn niet alle werkwoorden die een actie impliceren per definitie fout in een regel. Deze richtlijn kan dus niet geautomatiseerd worden.

### **12. CRUD woorden zijn nooit goed**

Deze richtlijn is een variant op de vorige richtlijn. De CRUD-woorden (Create, Retrieve, Update, Delete) verwijzen altijd naar een actie en zijn per definitie niet goed. Neem als voorbeeld de volgende regel:



*“Een klant mag niet verwijderd worden indien de klant openstaande bestellingen heeft”*

Deze regel verwijst naar een proces, en niet naar de status van een informatiesysteem. Deze regel kan niet op ieder moment gecontroleerd worden, omdat dit niets zegt over de status waarin het systeem zich bevindt. Dit komt doordat het woord “verwijderen” verwijst naar een actie. Een verbeterde regel zou zijn:

*“Een bestelling moet geplaatst zijn door een klant”*

Hier wordt een regel beschreven, welke op ieder moment gecontroleerd kan worden.

Het automatiseren van deze regel is mogelijk, omdat het eenvoudig is om uit te zoeken over een van de CRUD-woorden voorkomt in de regel. Als dat het geval is, zal deze expliciet naar een actie verwijzen en zal deze regel niet correct zijn.

### **13. Een onduidelijke periode is niet goed**

Zodra er een tijdsperiode in een regel voorkomt, moet er altijd goed gekeken worden of het wel duidelijk is welke periode wordt bedoeld.

*“Een bestelling moet goedgekeurd worden door twee managers tijdens een congres”.*

De vraag is waar de periode “tijdens een congres” naar verwijst. Dit kan namelijk zowel verwijzen naar de bestelling die wordt geplaatst (tijdens een congres) of naar het goedkeuren van de bestelling (tijdens een congres). Om onduidelijkheid te vermijden is het een goede zaak om deze onduidelijkheid te voorkomen en te expliciteren waar deze periode naar verwijst. Een gewijzigde versie zou kunnen zijn:

*“Een bestelling die wordt geplaatst tijdens een congres moet goedgekeurd worden door twee managers.”*

Net als in een van de voorgaande regels, is het een vereiste dat er betekenis van woorden in de regels beschikbaar is. Als de editor uit de betekenis kan halen dat “tijdens een congres” verwijst naar een tijdsperiode, zou hij de vraag kunnen stellen of de tijdsperiode duidelijk verwijst naar een van de twee werkwoorden in de zin (het goedkeuren of het binnenkomen). Echter de editor kan niet detecteren of de regel fout is. Het alternatief is net als bij richtlijn 7 het gebruik van een woordenlijst.

### **14. Mijden van “iedere” en “elke” is niet altijd goed**

In veel gevallen is het gebruik van het woord “iedere” en “elke” niet goed, maar er zijn uitzonderingen waar het juist wel handig is. Een goed voorbeeld is de regel: *“Een team moet een manager hebben vanaf 1 januari 2010.”*. Stel er zijn meerdere teams, betekent het dat in dit geval dat er maar één team een manager moet hebben? Waarschijnlijk niet, dus het gebruiker van “ieder” zou hier juist op zijn plek zijn.

Het is niet mogelijk om deze regel te automatiseren, aangezien het volledig afhankelijk is van de interpretatie van de regel. Strikt gezien is de regel niet fout, echter de regel roept wel vragen op. Er zou hoogstens door middel van een vraag aan de gebruiker geverifieerd kunnen worden of deze regel zo bedoeld is.

## 15. Verbergen van de rekeneenheid is niet goed

Als er een berekening wordt vastgelegd of uitgevoerd in een regel, moet er goed worden gekeken of de regel maar op een enkele manier is vast te leggen. Neem als voorbeeld de regel:

*“Een cursus mag niet meer dan 5 trainers en adviseurs betrekken”*

Men kan zich afvragen of het dan gaat om het feit dat er niet meer dan 5 trainers en 5 adviseurs aanwezig mogen zijn of dat het gaat om het feit dat het gecombineerde aantal van trainers en adviseurs niet meer dan 5 mag zijn. Deze syntactische ambiguïteit wordt veroorzaakt door het woordje “en”.

Het automatiseren van deze richtlijn is niet eenvoudig mogelijk, aangezien deze informatie niet met een natuurlijke taal parser is af te leiden. Er kan enkel een hint worden gegeven als het woord “en” in een regel wordt gebruikt. Er kan worden aangegeven dat er goed moet worden gekeken of de regel en dan met name de kwantificatie goed en helder is beschreven, Het ontdekken van het woordje “en” kan eenvoudig door middel van een string search.

## 16. “Of”, “hetzij” en “zowel” is zelden goed

In een regel kan de verleiding groot zijn om bij een opsomming gebruik te maken van het woordje “of”. “Of” kan echter snel onduidelijkheid creëren. Beschouw de regel *“Een project moet een manager, budget of sponsor hebben”*. Moet het project of een manager, of een budget of een sponsor hebben. Of misschien twee van de drie? Door gebruik te maken met lijsten van voorwaarden kan dit duidelijk geëxpliciteerd worden, is de uitbreidbaarheid eenvoudiger en is er sneller overzicht.

Een herschreven versie zou als volgt er uit kunnen zien. *“Een project moet minimaal aan één van de volgende voorwaarden voldoen:*

- *Het project heeft een manager.*
- *Het project heeft een budget.*
- *Het project heeft een sponsor.”*

De automatisering van deze richtlijn is eenvoudig mogelijk door een string search.

## 17. Samengestelde berekeningen zijn niet goed

Het is mogelijk om met een regel een berekening vast te leggen. Het is echter een gevaar om samengestelde berekeningen in een regel te gebruiken. Neem als voorbeeld de regel:

*“De optelling van alle betaalde bedragen van een bestelling moet groter of gelijk zijn aan het uitstaande bedrag voor de bestelling.”*

Hierbij wordt er twee maal een berekening gedaan, namelijk het optellen van alle betaalde bedragen en de vergelijking hiervan met het uitstaande bedrag voor de bestelling. Eigenlijk zitten hier twee berekeningen verstopt in één regel en deze zouden dus gesplitst moeten worden.

1. *Het betaalde bestelbedrag moet berekend worden als de optelling van alle bedragen die betaald zijn voor de bestelling (definitie van “betaald bestelbedrag”)*
2. *Het betaalde bestelbedrag moet groter of gelijk zijn aan het uitstaande bedrag voor de bestelling. (gebruik van “betaald bestelbedrag”)*

Het detecteren van samengestelde berekeningen is niet mogelijk met behulp van een natuurlijke taal parser. Enkel door betekening aan woorden te geven zou dit gedetecteerd kunnen worden. Door wederom een woordenlijst te gebruiken zouden bepaalde woorden geïdentificeerd kunnen worden als berekening. Voorbeelden van die woorden zijn “*optelling*” en “*groter of gelijk*”. Als er twee of meer van dit soort woorden voorkomen in een regel, is er waarschijnlijk sprake van een samengestelde berekening.

---

## 4.2 SAMENVATTING

---

Het automatisch ondersteunen van de richtlijnen in RuleSpeak is voor een groot gedeelte mogelijk. De benodigheden hiervoor zijn dan als volgt:

- Het gebruiken kunnen maken van een string search op een regel. Dit is in elke programmeertaal mogelijk.
- Een natuurlijke taal parser die feittypen uit een regel kan opleveren en meervoudsvormen kan ontdekken. Deze natuurlijke taal parsers zijn beschikbaar.
- Meta-informatie over en betekenis van bepaalde woorden uit regels, bijvoorbeeld woorden die een tijdsindicatie of –periode aangeven. Dit soort informatie kan niet automatisch afgeleid worden en er mag dan ook aan worden genomen dat dit niet beschikbaar is.

Voor alle richtlijnen waar de betekenis van woorden voor benodigd is, kunnen dus niet volledig geautomatiseerd worden. Er kan wel een goede gok worden gedaan door het gebruik van woordenlijsten met bijvoorbeeld woorden die waarschijnlijk een tijdsindicatie of een persoon aangeven. Deze woordenlijsten kunnen zelf opgesteld en aangevuld worden. Dit zal nooit een complete lijst worden, maar in ieder geval wel de meeste voorkomende gevallen kunnen afdekken. Het aanvullen van woordenlijsten kan ondersteund worden door gebruikers de optie te geven om woorden te kunnen markeren als persoon of als tijdsbepaling.

Feit blijft dat sommige richtlijnen niet volledig automatisch te controleren zijn. Het beste wat een editor kan doen, is het geven van een hint om de aandacht van de gebruikers te trekken en hem nogmaals goed hierover te laten nadenken. Het blijven tenslotte richtlijnen.

In tabel 1 op de volgende pagina staat het overzicht van alle richtlijnen in RuleSpeak. Van elke richtlijn is aangegeven of deze richtlijnen te automatiseren valt en zo ja, met behulp waarvan. Als er betekenis van woorden benodigd is, is er waar mogelijk een alternatief aangegeven.

Richtlijn	Te automatiseren?	M.b.v.	Alternatief
<b>Er moet een sleutelwoord voorkomen in elke regel</b>	Ja	String search	
<b>“Kan” is niet goed</b>	Ja	String search	
<b>“Zomaar wat” is niet goed</b>	Ja	Een natuurlijke taal parser	
<b>Verborgen feittypen zijn niet goed</b>	Nee	Enkel met de beschikbaarheid van betekenis of meta-informatie	Gebruik van een woordenlijst
<b>Een onderwerp en lijdend voorwerp in meervoud</b>	Ja	Een natuurlijke taal parser	
<b>De zin beginnen met “als” is niet goed</b>	Ja	String search	
<b>Een tijdsbepaling aan het begin van de regel</b>	Nee	Enkel met de beschikbaarheid van betekenis of meta-informatie	Gebruik van een woordenlijst
<b>Overslaan van feittypen is niet goed</b>	Ja	Een natuurlijke taal parser en een correct feittype model	
<b>Een persoon als onderwerp is vaak niet goed</b>	Nee	Enkel met de beschikbaarheid van betekenis of meta-informatie	Gebruik van een woordenlijst
<b>De gebiedende wijs is nooit goed</b>	Gedeeltelijk	Een natuurlijk taal parser	
<b>Werkwoorden over een actie zijn niet goed</b>	Nee		
<b>CRUD woorden zijn nooit goed</b>	Ja	Een natuurlijke taal parser	
<b>Onduidelijke periode is niet goed</b>	Nee	Enkel met de beschikbaarheid van betekenis of meta-informatie	Gebruik van een woordenlijst
<b>Mijden van ‘iedere’ en ‘elke’ is niet altijd goed</b>	Nee		
<b>Verbergen van de rekeneenheid is niet goed</b>	Gedeeltelijk	String search	
<b>“Of”, “hetzij” en “zowel” zijn zelden goed</b>	Ja	String search	
<b>Samengestelde berekeningen zijn niet goed</b>	Nee	Enkel met de beschikbaarheid van betekenis of meta-informatie	Gebruik van een woordenlijst

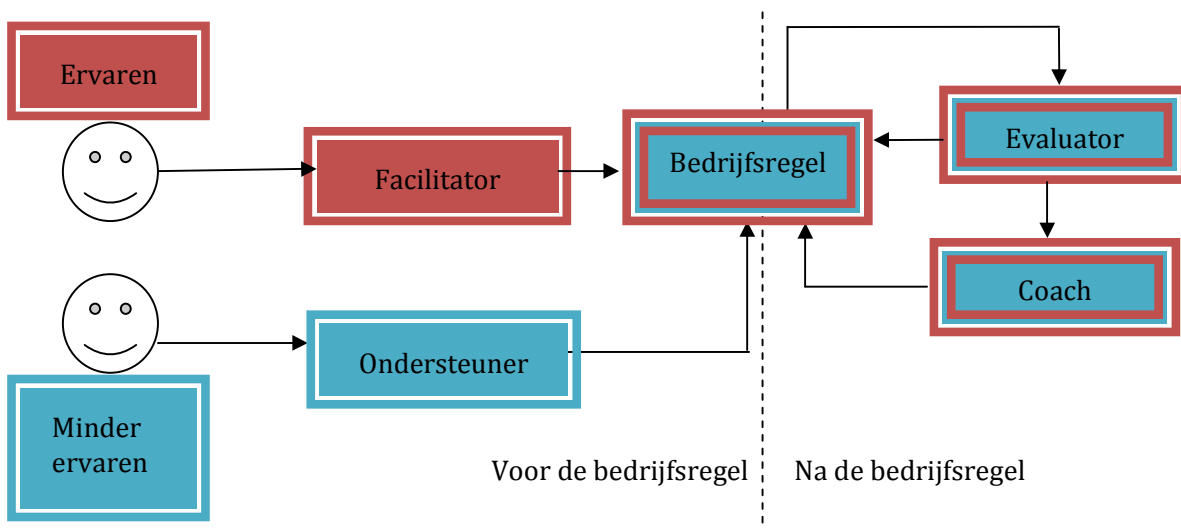
Tabel 1 Een overzicht van RuleSpeak en in hoeverre de richtlijnen te automatiseren zijn

## 5. DE NIEUWE EDITOR

In de vorige hoofdstukken hebben we gezien dat een editor verschillende rollen kan spelen voor verschillende gebruikers. De aanleiding tot deze scriptie was de vraag om concrete verbeteringen aan de RuleXpress-editor.

In een en dezelfde editor kunnen al deze rollen tot uitdrukking komen. Op verschillende momenten in het proces kunnen deze rollen hun waarde bewijzen.

In figuur 6 zijn de rollen schematisch weergegeven ten opzichte van de gebruiker en de bedrijfsregel.



Figuur 6 Een schematisch overzicht van de verschillende rollen

De ondersteunende en faciliterende rol spelen een rol als er nog geen bedrijfsregel bestaat. Op dat moment kan een van de twee rollen worden gekozen om de bedrijfsregel op te gaan stellen. De minder ervaren gebruiker zal gebruik kunnen maken van de ondersteunende rol. Op die manier wordt de gebruiker aan de hand genomen in het modelleringsproces en zullen alle stappen die moeten worden genomen expliciet worden doorlopen. Tevens kan de editor handreikingen doen om de gebruiker op weg te helpen.

De meer ervaren gebruiker zal een gedeelte van de deelstappen al nemen, zonder deze expliciet te onderscheiden. Door gebruik te maken van de faciliterende rol zal een ervaren gebruiker sneller zijn doel bereiken, namelijk het vastleggen van een regel. Hierbij worden de deelstappen overgeslagen.

Als eenmaal een bedrijfsregel is vastgelegd kunnen de andere twee rollen in werking treden. In eerste instantie kan de evaluator zijn rol vervullen. Op dat moment bekijkt de editor wat hij kan onderscheiden in de regel. Er kan bijvoorbeeld worden aangegeven welke termen zijn gebruikt, welk zintype wordt herkend en welke voorwaarden er op van toepassing zijn. Deze kunnen voor de gebruiker een trigger zijn om nog eens goed te kijken of dit daadwerkelijk is wat beoogd is. Tevens kunnen er potentiële termen en feittypen worden aangedragen.

Vervolgens kan de coachende rol worden uitgevoerd. Aan de hand van de resultaten uit de evaluator kunnen er concrete aanwijzingen worden gegeven voor een specifieke bedrijfsregel. Deze coaching leidt, als het goed is, tot een nieuwe versie van de regel. Vervolgens kan het proces opnieuw beginnen door de evaluator zijn werk te laten doen en vervolgens de coachende functie uit te voeren.

Dit proces benadrukt de iteratieve manier van het opstellen van een bedrijfsregel. Een bedrijfsregel wordt zelden in één keer goed opgeschreven. Door de evaluator en coachrol toe te passen kunnen er steeds verbeterlagen worden gemaakt aan de regel wat ten slotte een betere regel tot gevolg moet hebben. Deze rollen en de aanwijzingen van deze rollen activeren de gebruiker om een nieuwe iteratie in te gaan en de bedrijfsregel opnieuw te verbeteren.

In veel organisaties zijn er al regels aanwezig binnen de organisatie. In sommige gevallen zijn deze ook al vastgelegd. Een goede manier om deze te verbeteren is de regels allemaal in te voeren via de faciliterende rol en vervolgens per regel de evaluator en de coach zijn gang te laten gaan.

In de rest van dit hoofdstuk zal worden ingegaan op de manier waarop de verschillende rollen in de gemaakte editor tot uitdrukking komen.

---

## 5.1 FACILITERENDE ROL

---

De faciliterende rol is in de gemaakte editor erg summier aanwezig. In RuleXpress wordt deze rol al volledig ondersteund door middel van verschillende functionaliteiten. Zo is het mogelijk om in RuleXpress termen, regels en feittypen vast te leggen en te beheren. De wijzigingen van termen, regels en feittypen worden bijgehouden en het is mogelijk om samen te werken om regels te maken en te bediscussiëren.

Aangezien het doel van de scriptie niet is om RuleXpress na te bouwen, is deze rol op de meest eenvoudige manier geïmplementeerd. Het is in de editor mogelijk om een regel vast te leggen door middel van een simpel tekstveld. Het bijhouden van historie of collaboratie met anderen wordt niet ondersteund.

---

## 5.2 ONDERSTEUNENDE ROL

---

De ondersteunende rol is daarentegen een stuk uitgebreider. Deze rol wordt in de geïmplementeerde editor ingevuld door een wizard (Hoppenbrouwers, Weigand, & Rouwette, *Setting Rules of Play for Collaborative Modeling*, 2009). Een wizard is een interactief stuk programma dat ten doel heeft om aan de hand van concrete vragen de gebruiker door een complex ontwerpproces te leiden. In het modelleerproces moeten verschillende modelleerkeuzes worden gemaakt. Een ervaren gebruiker is zich bewust van deze keuzes en zal deze automatisch maken en kunnen verantwoorden waarom deze keuzes zijn gemaakt. De minder ervaren gebruiker zal zich veelal niet bewust zijn dat deze keuzes moeten worden gemaakt en zal daarom door de wizard kunnen worden geholpen. Uit een eerdere studie is al gebleken dat een wizard kan bijdragen om minder ervaren gebruikers te ondersteunen in het modelleerproces. (Hoppenbrouwers & Schotten, *A Game Prototype for Basic Process Model Elicitation*, 2009)

De wizard is geïmplementeerd als een vraag en antwoordspel. Aan de hand van de uitkomst van een vraag zal een concrete handreiking worden gegeven in de vorm van te gebruiken woorden, of zal er een vervolgvraag worden gesteld. Elke vraag probeert een ander onderdeel van een bedrijfsregel te expliciteren en de gebruiker te laten nadenken over dat onderdeel. (Hoppenbrouwers, Proper, & Weide, *Formal Modelling as a Grounded Conversation*, 2005)

De woorden die worden aangereikt kunnen worden gezien als bouwstenen welke de belangrijkste onderdelen moeten gaan vormen van de regel. Deze worden bij elkaar gezet zodat de gebruiker deze uiteindelijk eventueel kan gebruiken bij het daadwerkelijk formuleren van de regel. De gebruiker maakt hierbij zelf de keuze of hij bepaalde bouwstenen wel of niet gebruikt.

---

### 5.2.1 DE INTERACTIE MET DE WIZARD

---

In bedrijfsregels zijn grofweg vijf belangrijke onderdelen te onderscheiden;

- De termen. De concepten waar de regel betrekking op heeft.
- De feittypen, oftewel de relatie tussen de verschillende termen.
- Een sleutelwoord. Dit sleutelwoord bepaald de “zwaarte” van de relatie. Hiermee wordt aangegeven of een regel structureel of operationeel van aard is of dat er een advies wordt gegeven.
- Eventuele voorwaarden. Dit kunnen er 0, 1 of meer voorwaarden zijn.
- Een tijdsperiode. Sommige regels hebben betrekking op een specifieke tijd of tijdsperiode.

Door de wizard op elk van deze onderdelen ondersteuning te laten bieden, wordt de gebruiker gedwongen bewust over deze stappen na te denken. Tevens krijgt de gebruiker de bouwstenen aangereikt in vorm van een soort kladblaadje, waardoor het voor de gebruiker eenvoudiger is om alle onderdelen te overzien. Bij elke vraag van de wizard is ook een hulptekst beschikbaar waarin nadere uitleg wordt gegeven en één á twee voorbeelden zijn gegeven over de invloed van het antwoord.

Er is voor gekozen om niet een vast stramien te hanteren om de bouwstenen te selecteren. Een gebruiker kan er voor kiezen om eerst het onderdeel “sleutelwoord” uit te voeren, en daarna de termen pas te selecteren, maar ook vice versa. De volgorde staat de gebruiker vrij.

## DE TERMEN EN FEITTYPEN

In dit deel van de wizard worden alle gedefinieerde termen aan de gebruiker getoond. Deze termen kan de gebruiker vervolgens als “bouwstenen” toevoegen. Zodra er een term wordt geselecteerd, worden de bijbehorende gedefinieerde feittypen getoond. Op die manier weet de gebruiker direct welke relaties deze term heeft met andere termen. Als er een compleet feittypemodel zou zijn, zou de gebruiker altijd hier uit moeten kiezen, echter in de praktijk zal een feittypemodel vaak niet compleet aanwezig zijn. Daarom worden de feittypen enkel als hint aangegeven.

## DE SLEUTELWOORDEN

Door een drietal vragen te doorlopen wordt de gebruiker zich bewust wat voor type regel hij gaan definiëren.

***Wilt u een advies geven of een regel beschrijven?***

- ***Regel***
- ***Advies***

Deze vraag geeft een belangrijke eerste indicatie. Het belangrijkste verschil tussen een regel en een advies, is dat een regel de vrijheid beperkt terwijl een advies dat niet doet. Een regel zal leiden tot worden als “moet”, “altijd”, “mag niet” terwijl een advies zal resulteren in vrijblijvende woorden als “kan” of “mag”.

Een advies beperkt de vrijheid geenszins maar is soms toch erg nuttig om te voegen aan de regelset. Zo kunnen adviezen gedrag en begrip sturen bij gebruikers sturen. In tegenstelling tot een regel geeft een advies expliciet aan dat iets is toegestaan, iets mogelijk is of niet verboden is. Dit kan handig zijn voor onduidelijke situaties. Een voorbeeld van een advies zou kunnen zijn;

*“Twee personen van hetzelfde geslacht mogen trouwen.”*

Tot een aantal jaar geleden was dit niet het geval. Een advies als het bovenstaande kan erg nuttig zijn om expliciet naar de gebruiker te communiceren dat dit nu wel het geval is.

<b>Regel</b>	<b>Advies</b>
Beperkt de vrijheid van handelen	Beperkt de vrijheid van handelen niet
Geeft aan dat iets <ul style="list-style-type: none"> <li>- Verplicht is</li> <li>- Verboden is</li> <li>- Niet toegestaan is</li> <li>- Noodzakelijk is</li> <li>- Onmogelijk is</li> </ul>	Geeft aan dat iets <ul style="list-style-type: none"> <li>- Mogelijk is</li> <li>- Niet onmogelijk is</li> <li>- Toegestaan is</li> <li>- Niet verboden is</li> </ul>
Kan worden eventueel worden overtreden	“Overtredingen” zijn niet van toepassing op adviezen



Als er gekozen wordt om een advies vast te leggen, is dit onderdeel van de wizard klaar. Als er voor wordt gekozen om een regel vast te stellen zal de wizard verder gaan.

Uitkomst	Aangereikte bouwstenen
<b>Regel</b>	Ga verder met de volgende vraag
<b>Advies</b>	... mag ... ... kan ...

### ***Kan deze regel in de praktijk overtreden worden?***

- **Ja**
- **Nee**

Middels deze vraag wordt er verder ingegaan op het “gewicht” van de regel. Op dit moment moet er een modelleerkeuze worden gemaakt of een operationeel of structurele regel vastgelegd gaat worden. In het geval van een structurele regel wordt er vanuit gegaan dat deze regel niet overtreden kan worden en altijd moet gelden. Het gevolg hiervan is dat het sleutelwoord “altijd” of in het geval van een ontkenning “nooit” gebruikt moet worden.

*De leeftijd van de aanvrager is altijd minstens 18.*

Deze regel kan zo worden vastgelegd als het bijvoorbeeld gaat om het aanvragen van een rijbewijs. Deze regel mag hierbij nooit overtreden worden.

Een operationele regel is daarentegen mogelijk om te overtreden en geeft meer een mogelijkheid weer. De keuze voor de operationele regel heeft als gevolg dat er gebruik gaat worden gemaakt van een sleutelwoord zoals “moet”, “mag niet” of “mag ... alleen”.

*De leeftijd van de aanvrager moet minstens 18 zijn. (maar het is mogelijk dat deze wordt overtreden)*

Dit kan bijvoorbeeld het geval zijn in een slijterij, waarbij iemand probeert sterke drank te halen. Normaliter is een aanvrager altijd 18 jaar, maar in de praktijk gebeurt het ook wel dat een aanvrager geen 18 is omdat bijvoorbeeld geen legitimatie wordt gevraagd.

In beide bovenstaande gevallen is de regel hetzelfde en zou het moeten gelden, echter in het geval van de structurele regel moet de regel altijd gelden, terwijl in het tweede geval de regel overtreden kan worden.

Het is belangrijk om er van bewust te zijn dat dit een modelleerkeuze is. Er zal bewust een keuze moeten worden gemaakt voor het een of het ander.

Uitkomst	Aangereikte bouwstenen	Alternatieve bouwstenen
<b>Ja</b>	... moet .. ... mag niet ... ... mag ... alleen ...	... dient ...  ... mag alleen ...
<b>Nee</b>	... altijd ... ... nooit ... ... kan ... alleen ...	... in ieder geval ... ... in geen geval ... ... kan ... alleen indien ...

**Wordt met deze regel een categorie afgeleid, een berekening vastgelegd of een proces beschrijving gegeven?**

- **Nee**
- **Een categorie afgeleid**
- **Een berekening vastgelegd**
- **Een procesbeschrijving gegeven**

Tenslotte wordt er nog gevraagd of de gebruiker een speciaal type regel wil vast leggen. Dit kan gebeuren in sommige gevallen en RuleSpeak beschrijft drie van deze gevallen. In principe zal er meestal een regel worden gedefinieerd en dan zal de uitkomst van de vorige vraag worden aangereikt.

Er kan echter ook een categorie worden afgeleid. Een voorbeeld van zo'n afleiding is:

*“Een aanvraag moet worden beschouwd als “ontvankelijk” indien de aanvrager minstens 18 jaar oud is”*

In deze regel wordt de categorie “ontvankelijk” omschreven.

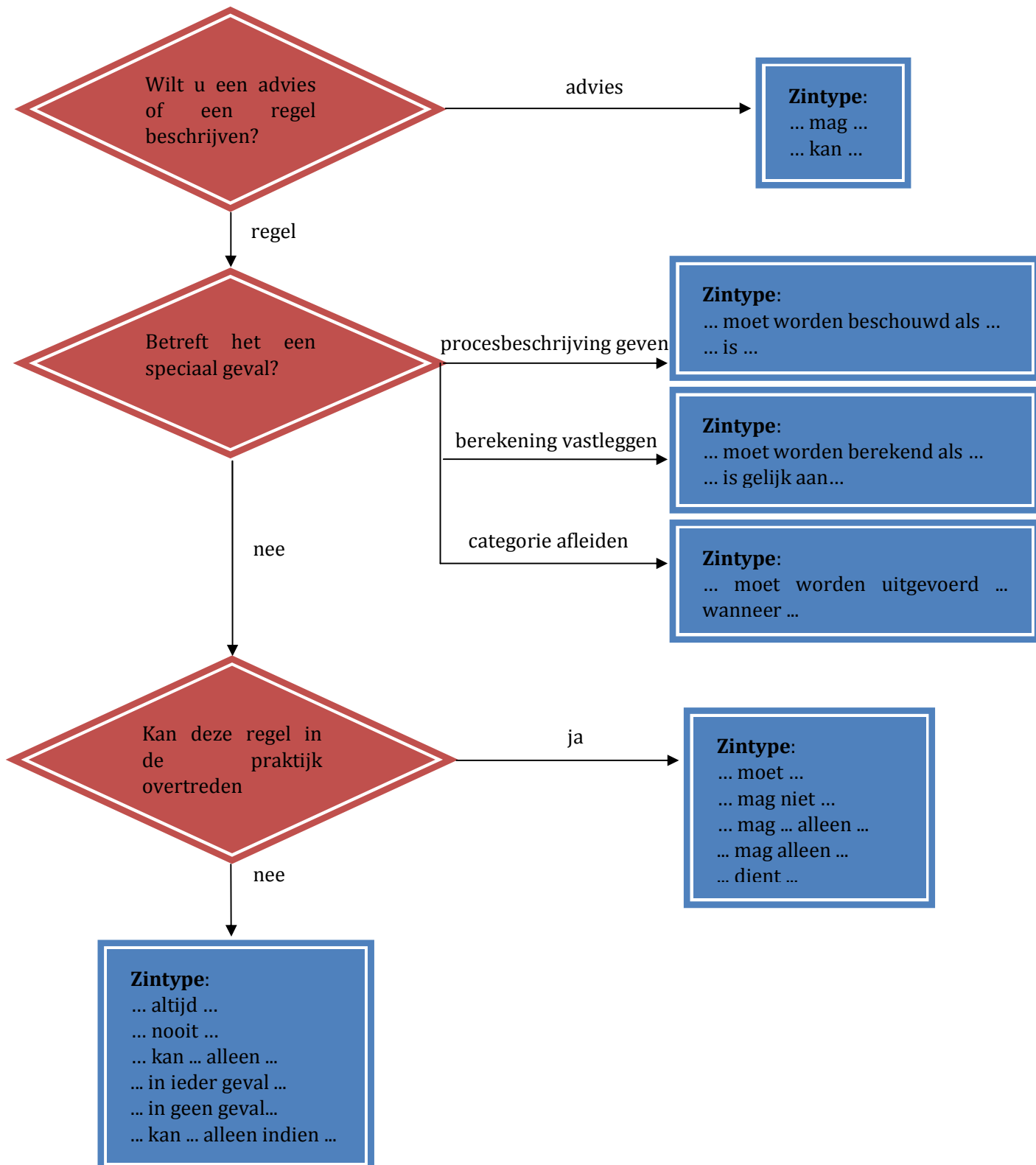
Een andere manier is het vastleggen van een berekening. Dit zal resulteren in het gebruik van de sleutel woorden “moet worden berekend als” of “is gelijk aan”. Een voorbeeldregel is;

*“De premie volksverzekeringen moet worden berekend als de som van ...”*

Tenslotte kan er ook een procesbeschrijving worden vastgelegd. Dit komt voor als er stapjes in een proces gedefinieerd worden. Dit wordt typisch niet vastgelegd door middel van regels, maar er zijn uitzonderingen waarbij dit eventueel van pas komen. Het vastleggen van een procesbeschrijving resulteert in de sleutelwoorden “... moet worden uitgevoerd ... wanneer ...”

<b>Uitkomst</b>	<b>Aangereikte bouwstenen</b>	<b>Alternatieve bouwstenen</b>
<b>Nee</b>	Gebruik de uitkomst van de vorige vraag	
<b>Categorie</b>	... moet worden beschouwd als ...	... is ...
<b>Berekening</b>	... moet worden berekend als ...	... is gelijk aan ...
<b>Proces</b>	... moet worden uitgevoerd ... wanneer ...	

In de onderstaande figuur bevindt zich een schematisch overzicht van dit onderdeel van de wizard.



Figuur 7 Het proces om een zintype te bepalen

## VOORWAARDEN

Op een regel kunnen voorwaarden van toepassing zijn. Deze kunnen aangeven wanneer een regel juist wel of juist niet van toepassing is. Door middel van de wizard wordt boven water gehaald of er voorwaarden zijn en hoe deze zich verhouden tot elkaar.

### ***Zitten er voorwaarden aan de regel?***

- ***Ja***
- ***Nee***

Als een regel in alle gevallen geldt zal dit onderdeel verder niet van toepassing zijn in de regel. Er zal dan ook geen bouwsteen worden aangereikt.

Mochten er wel voorwaarden verbonden zijn aan de regel dan wordt er doorgedaan met de volgende vraag.

Uitkomst	Aangereikte bouwstenen
<b>Ja</b>	Geen
<b>Nee</b>	Ga door met de volgende vraag

### ***Zijn er meer dan één voorwaarde verbonden aan de regel?***

- ***Ja***
- ***Nee***

Aan sommige regels is maar een enkele voorwaarde verbonden. Dit zal dan resulteren in het toevoegen van het sleutelwoord “indien” of “als”. Als er meer voorwaarden aan een regel zijn verbonden zal er beter moeten worden gekeken naar deze voorwaarden.

Uitkomst	Aangereikte bouwstenen	Alternatieve bouwstenen
<b>Ja</b>	Ga door met de volgende vraag	
<b>Nee</b>	... indien ...	... als ...

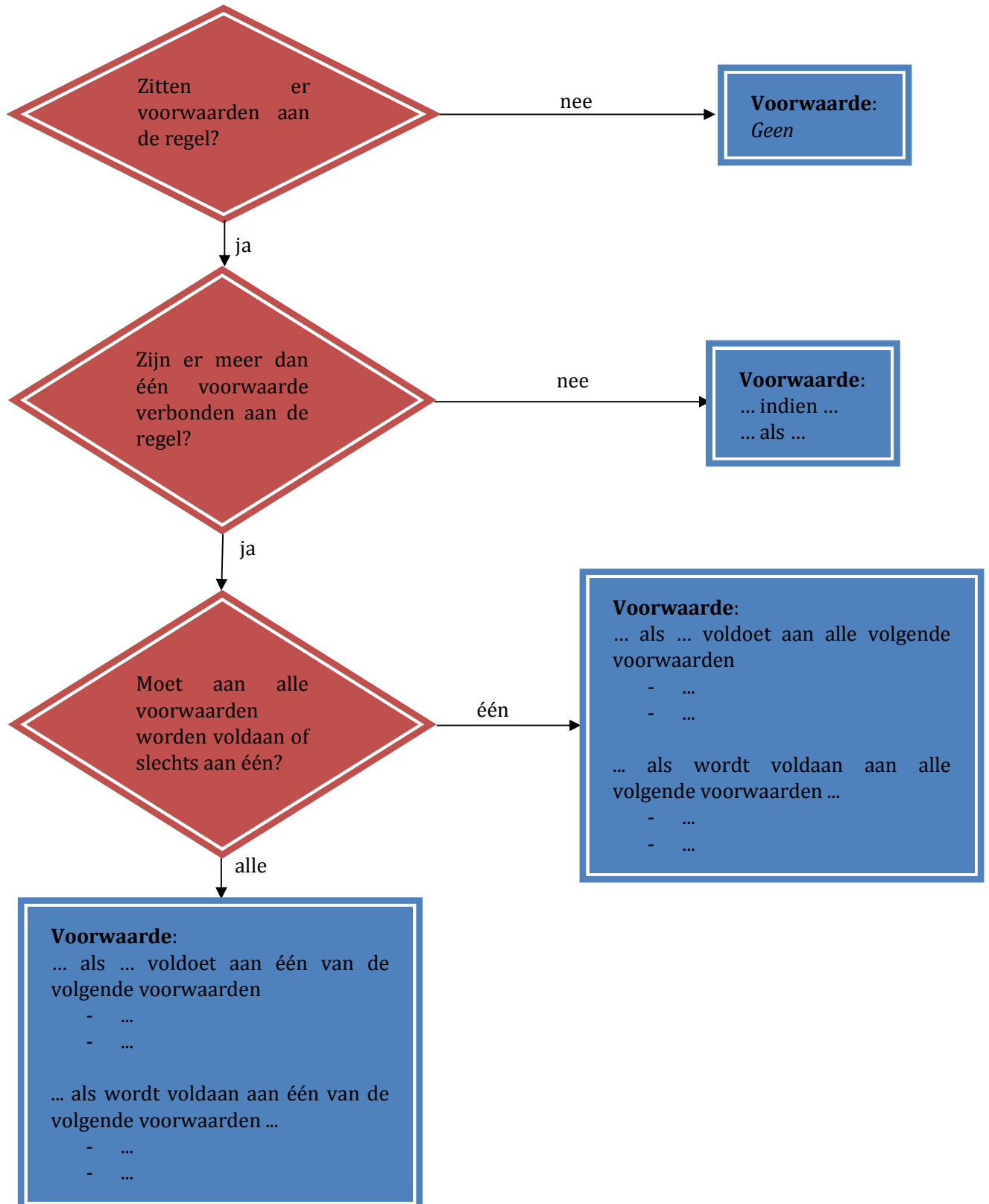
**Moet aan alle voorwaarden worden voldaan of slechts aan één?**

- **Alle**
- **Een**

Zoals al in de RuleSpeak-richtlijnen duidelijk werd, moeten gebruikers er op worden geattendeerd dat in een regel expliciet wordt gemaakt bij voorwaarden sprake is van een conjunctie (en, en, en) of van een disjunctie (of, of, of) of een mix van beide vormen (aan minimaal twee voorwaarden). Vaak laten gebruikers dit in het midden. Het gevolg daarvan zijn onduidelijke regels. Door deze vraag te beantwoorden wordt er een bouwsteen aangereikt dit concreet maakt.

<b>Uitkomst</b>	<b>Aangereikte bouwstenen</b>	<b>Alternatieve bouwstenen</b>
<b>Alle</b>	... als ... voldoet aan alle volgende voorwaarden - ... - ...	... als wordt voldaan aan alle volgende voorwaarden - ... - ...
<b>Een</b>	... als ... voldoet aan een van de volgende voorwaarden - ... - ...	... als voldaan wordt aan een van de volgende voorwaarden - ... - ...

Een schematisch overzicht van dit onderdeel van de wizard is te vinden in figuur 8.



Figuur 8 Het proces om tot een voorwaarde te komen

---

 TIJDSPERIODE
 

---

***Is er sprake van een tijdsindicatie of -periode in de regel?***

- ***Ja***
- ***Nee***

In sommige regels is er sprake van een tijdsaanduiding of een tijdsinterval. Gebruikers zijn zich niet altijd even bewust van het feit dat dit het geval kan zijn, waardoor deze stapt hen daar aan herinnert. Een voorbeeld zou kunnen zijn

*“De begin- en eindtijd van een tentamen moet bekend zijn aan het begin van een semester”*

Door expliciet de vraag te stellen of hier sprake van is, wordt de gebruiker getriggerd hier over na te denken. In de RuleSpeak standaard zijn (helaas) geen concrete handvatten gegeven om dit op te lossen. Daarom zijn de uitkomsten van de vraag zelf gedefinieerd op grond van analyse van verschillende sets met regels.

Uitkomst	Aangereikte bouwstenen	Alternatieve bouwstenen
<b>Ja</b>	... op ...	... voor het begin van ... ... tijdens ... ... voor het eind van ... ... na het sluiten van ... ... voor ... ... na ...
<b>Nee</b>	Geen	

---

## DE REGEL OPSTELLEN

---

Nadat alle bouwstenen zijn geselecteerd kan er begonnen worden met het opstellen van de betreffende regel. De gebruiker heeft de verzamelde bouwstenen bij elkaar staan om daarmee zijn regel op te bouwen.

Op dit moment kunnen er al enkele richtlijnen uit RuleSpeak worden aangereikt als tip om te beginnen. De meeste regels zullen met het woord “een” beginnen, omdat een regel altijd gaat over een willekeurig element uit een set elementen. Zo zal “Een bestelling” gaan over een willekeurige bestelling en met “De bestelling” wordt een specifiek element bedoeld. Door met “een” te beginnen dwing je de gebruiker tevens om het onderwerp in enkelvoud te plaatsen.

RuleSpeak geeft daarnaast enkele handreikingen over de opbouw van een regel; zoals “begin met het onderwerp” en “Plaats de voorwaarden en tijdsbepalingen achteraan”. Een goede opbouw van een regel zou er als volgt uit kunnen zien;

*<term> < sleutelwoord> < feittype-werkwoord> <term> < tijdsindicatie> < voorwaarden>*

Zoals bij de meeste richtlijnen uit RuleSpeak is dit niet in alle gevallen waar, maar door de gebruiker hierop te attenderen, kunnen zij hier veel bewuster mee omgaan. De gebruiker staat nog steeds vrij om de regel op te bouwen hoe hij dat wil, de wizard schrijft geen syntaxisvorm voor aangezien dit simpelweg niet kan. De wizard doet enkel de handreikingen, het is aan de gebruiker om deze handreikingen te gebruiken.



### 5.3 EVALUERENDE ROL

---

Nadat de regel is opgesteld komt deze rol in beeld. De editor bekijkt wat hij “ziet” in de regel en zal de gebruiker vragen om dit te verifiëren of te wijzigen. Deze rol moet ingevuld worden met behulp van een natuurlijke taal parser, aangezien een dergelijke parser daadwerkelijk “ziet” wat er in een regel aan de hand is.

Vooralsnog zal de editor kijken naar twee onderdelen in de zin;

- De termen
- De feittypen

#### **De termen**

Elk zelfstandig naamwoord in een regel is een indicatie van een term en elke term zou gedefinieerd moeten zijn in het vocabulaire van de regels. In de evaluerende rol geeft de editor een lijst van zelfstandige naamwoorden die gebruikt zijn in de regel en daarachter wordt aangegeven of het zelfstandig naamwoord is gedefinieerd als term of niet. Als de term niet is gedefinieerd, kan de editor het volgende vragen aan de gebruiker:

- of de gebruiker de term nu wil definiëren en zodoende wil toevoegen aan het vocabulaire
- of de gebruiker een andere term wil gebruiken
- of de gebruiker niets wil doen met de term.

Door de gebruiker voor deze keuze te stellen krijgt hij weliswaar de vrijheid om te doen en laten wat hij wil, maar wordt de gebruiker zich wel bewust van deze keuze.

#### **De feittypen**

Uit een natuurlijke taal parse kunnen ook de feittypen worden gehaald. Een parse geeft aan welke werkwoorden zijn herkend en op welke zelfstandige naamwoorden deze betrekking hebben. Net als bij de termen kan de editor een set met herkende feittypen geven. Vervolgens kan van elk feittype worden bepaald of deze al is gedefinieerd als feittype. Als dit niet het geval is, is er een keuze voor de gebruiker:

- een gebruiker kan het nieuwe feittype toevoegen aan het model
- een gebruiker kan het huidige niet-bestaande feittype vervangen door een al wel bestaand feittype
- een gebruiker kan niets doen met het feittype.

Het voordeel van deze manier van werken is dat er niet eerst een volledige set met termen hoeft te worden gedefinieerd of een geheel feittype-model moet worden gemaakt. Gedurende het maken van de regels en het evalueren van de regels kunnen termen en feittypen worden toegevoegd. Hierdoor wordt er zeer flexibel omgegaan met het spanningsveld tussen termen, regels en feittypen.

## 5.4 COACHENDE ROL

---

De coachende rol is bedoeld om achteraf aanwijzingen te geven op basis van de regel. De inhoudelijke semantische structuur kan uiteraard niet worden beoordeeld, maar op basis van de RuleSpeak richtlijnen kunnen er wel aanwijzingen worden gegeven. Hierbij zal de “coach” ook enkel aanwijzingen geven, omdat er op de meeste RuleSpeak richtlijnen altijd uitzonderingen zijn te geven. De gebruiker bepaalt uiteindelijk of hij de regel wel of niet wijzigt.

In het vorige hoofdstuk is beschreven welk van de richtlijnen van RuleSpeak te automatiseren is en wat daarvoor nodig is. Deze richtlijnen geven sturing aan de coachende rol.

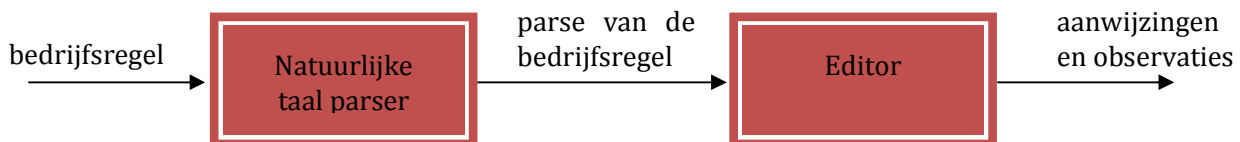
## 5.5 DE IMPLEMENTATIE

---

De implementatie van de editor is gebeurd middels een webgebaseerde applicatie. In deze applicatie kunnen de verschillende onderdelen van bedrijfsregels worden toegevoegd. Er is een wizard die de gebruiker ondersteunt bij het construeren van een regel.

Daarnaast is er een gedeelte waar de evaluerende rol en coachende rol zijn geïmplementeerd. Aan de hand van een bedrijfsregel worden er eventuele opmerkingen en aanwijzingen geplaatst. In appendix C is een voorbeeld weergave vastgelegd van het opstellen van een bedrijfsregel door middel van de wizard om vervolgens de regel verder te evalueren op basis van aanwijzingen.

Helaas kon niet de volledige beschrijving, zoals gegeven in dit hoofdstuk, worden geïmplementeerd in de editor. Er is onderzoek gedaan naar een natuurlijke taal parser zoals deze benodigd is in een groot gedeelte van de evaluerende rol. De natuurlijke taal parser die ter beschikking stond was enkel werkzaam voor de Engelse taal. Daarnaast kon deze niet eenvoudig gekoppeld worden aan de editor, waardoor het niet mogelijk is om 'live' een regel in te voeren, de natuurlijke taal parser een parse te laten genereren en vervolgens aan de hand van deze parse aanwijzingen te geven.



Figuur 9 De voorgestelde implementatie

Er is daarom een proof-of-concept editor gemaakt. Hierbij was de output van de parser al gegenereerd en werd deze vervolgens aangeboden aan de editor. Aan de hand hiervan konden de observaties en aanwijzingen worden gegenereerd. Uit dit proof-of-concept bleek dat deze parser zeer nuttig kan zijn voor een editor. Er is echter niet getest met testers of deze ook deze aanwijzingen als meerwaarde beschouwen.

Daarnaast zijn enkele onderdelen uit dit hoofdstuk pas toegevoegd nadat de evaluatie was uitgevoerd. Gedurende de evaluatie is onder andere gebleken dat gebruikers behoefte hebben aan enkele richtlijnen bij het opstellen van een bedrijfsregel. In de test-editor werden gebruikers vrij gelaten en werd er geen enkele aanwijzing gegeven. Achteraf gaven veel gebruikers aan dat zij dit wel op prijs zouden stellen en zodoende is dit aan de beschrijving toegevoegd. Daarnaast is de tijdsperiode niet onderkend als aparte stap in de test-editor. Aan de hand van observaties bleek echter dat dit wel degelijk een relevante deelstap is, en deze is daarom toegevoegd aan de beschrijving van de editor.

---

## 6. EVALUATIE

---

Wanneer gebruik wordt gemaakt van Design Science speelt de evaluatie een zeer voorname rol. Aan de hand van ervaringen van gebruikers kan bekeken worden of aanpassingen en voorstellen daadwerkelijk een verbetering zijn voor de editor. Aan het eind van het afstudeertraject is de geïmplementeerde editor getest door een gemêleerd gezelschap van gebruikers.

Tijdens deze evaluaties werd gebruik gemaakt van het think-aloud protocol. Hierbij wordt van gebruikers gevraagd om al hun handelingen, hun denken, en hun gevoelens hardop uit te spreken, terwijl ze bezig zijn. Op deze manier kunnen eventuele onduidelijkheden worden gesignaleerd en geëxpliciteerd. Verder is getracht om zo min mogelijk aanwijzingen te geven aan de tester gedurende het proces, zodat de tester volledig op de editor en de aanwijzingen uit de editor moest vertrouwen.

---

### 6.1 OPZET

---

Het doel van de evaluatie was om te testen of het algemene idee achter de editor correct is. Het was niet de bedoeling om de gedetailleerde functionaliteit van de editor te testen; dat is tenslotte slechts een prototype. Het doel was om uit te zoeken of de uiteindelijke set regels van een hogere kwaliteit was dan de originele set door alle verschillende rollen te implementeren in de editor. Fouten in de feitelijke editor vinden was daarbij dus van secundair belang.

Zoals we in een eerder hoofdstuk al hebben gelezen is het lastig om de opstellers en beheerders van bedrijfsregels over een kam te scheren. Daarom is er ook met verschillende soorten mensen getest: mensen met veel ervaring in logica en modelleren, mensen zonder enige ervaring in logica en modelleren. Geen van de testers had vooraf ervaring met het opstellen van bedrijfsregels of SBVR.

De editor is aan de hand van een case getest. Net als veelal in de praktijk was er een oorspronkelijke set met regels beschikbaar. In het geval van de case betrof dit het examenreglement van de opleiding "Informatica en Informatiekunde" van de Radboud Universiteit Nijmegen. Een twintigtal regels hiervan werd voorgelegd aan de tester. De werd gevraagd om enkele regels hiervan opnieuw op te stellen met behulp van de wizard. Van te voren werd een set van termen aangereikt, evenals een (incomplete) feittypemodel. In appendix B zijn de oorspronkelijke regels, de termen en het feittypemodel te vinden. Nadat de tester de wizard had doorlopen kon een eerste versie van de bedrijfsregel worden opgeslagen.

Nadat de regels waren opgeslagen konden de testers de regels eventueel nog verbeteren aan de hand van de aanwijzingen die de editor gaf middels de evaluerende en coachende rol. Zoals aangegeven in het vorige hoofdstuk is deze maar in beperkte mate geïmplementeerd. Enkel de richtlijnen die door middel van een string search konden worden gegeven, zijn terug te vinden in de editor. Hierdoor was van tevoren al duidelijk dat voornamelijk de wizard-functionaliteit getest zou gaan worden.

---

## 6.2 RESULTATEN

---

Met elk van de testers is ongeveer een uur getest. In dit uur had een tester gemiddeld tijd de uitleg door te nemen en vier regels opnieuw op te stellen met behulp van de wizard. Geen van de testers had vooraf ervaring met SBVR-modellering. Gedurende het testen is geprobeerd de testers zo min mogelijk aanwijzingen te geven. Alleen als de tester niet verder kwam, is er een aanwijzing gegeven.

De nieuwe regels zijn later gecontroleerd door drie bedrijfregel-experts op de kwaliteit ervan. Aan elke bedrijfsregel is een cijfer toegekend tussen 0 en 10. Hierbij gold dat de oorspronkelijke regel een 5 kreeg. Een verslechtering van de regel had een cijfer tussen 0 en 4 als gevolg, en een verbetering een cijfer tussen 6 en 10, als de regel niet verbeterd of verslechterd was bleef het een 5. Daarnaast zijn alle regels in RuleXpress gestopt om te kijken welke kwaliteitsscore daaraan is gegeven. Op deze manier is getracht om de resultaten van de editor kwantificeerbaar te maken.

---

### 6.2.1 DEFINITIEVE REGELS

---

In totaal zijn er 27 nieuwe regels geschreven, waarvan 19 regels het resultaat waren van alleen de wizard functie. Nadat deze regels met behulp van de wizard zijn aangemaakt, vervulde de editor de evaluerende en coachende rol. Op grond van de aanwijzingen hiervan zijn nog 8 van deze 19 regels herschreven tot een nieuwe versie.

Als er gekeken wordt naar het gemiddelde cijfer dat gegeven wordt voor alle regels (zowel de regels uit de wizard als de regels na de evaluerende en coachende rol) stijgt het gemiddelde cijfer voor de regels van een 5 (de uitgangspositie voor de niet herschreven regels) naar 5,2. Dit is een zeer kleine verbetering van de regels.

Als er vervolgens ook wordt gekeken naar de definitieve regels, oftewel de regels die na de evaluerende en coachende rol eventueel ook zijn verbeterd, stijgt het gemiddelde cijfer van een 5 naar een 5,9. Dit is een aanzienlijke verbetering.

---

### 6.2.2 MODELLEERERVARING VERSUS GEEN MODELLEEREERVARING

---

Een andere manier om naar de resultaten te kijken was door te kijken naar het gemiddelde cijfer dat de personen zonder modelleerervaring scoorde en het gemiddelde cijfer dat een persoon zonder modelleerervaring scoorde.

Drie personen die aangaven modelleerervaring te hebben en een IT achtergrond scoorden gemiddeld een 5,2. Hierbij is dus niet een grote verbetering aangetroffen.

Drie personen gaven aan geen modelleerervaring te hebben en geen IT achtergrond. Deze drie personen scoorden gemiddeld een 5,9. Het idee van de wizard was ook om mensen zonder IT/modelleerachtergrond door het modelleerproces heen te leiden. Op grond van deze score mag worden aangenomen dat dit inderdaad om een verbetering gaat.

---

## 6.3 VRAGENLIJST

---

Na de test moest de tester een vragenlijst invullen. Hierin werd de tester gevraagd een twaalfstal stellingen te beoordelen met een cijfer 1 tot en met 5, een drietal open vragen te beantwoorden, en drie vragen over de achtergrond van de tester te beantwoorden. Het cijfer 1 betekent dat de tester het zeer oneens was met de stelling en een 5 betekent dat de test het zeer eens is met de stelling.

Ondanks dat het aantal van testers te laag is om statistische analyse te kunnen loslaten op de vragenlijsten, zal toch worden getracht de resultaten te interpreteren. Voor elke vraag wordt gekeken naar de gemiddelde score voor alle zes de testers, voor de drie testers zonder modelleerervaring en de drie testers met modelleerervaring.

---

### 6.3.1 VRAGEN OVER DE CASE

---

#### ***Ik begreep de opzet van de case***

Gemiddeld: 4,2, Modelleerervaring: 4, Niet modelleerervaring 4,3

Deze vraag is gesteld om er zeker van te zijn dat de uitkomst van de regels zo min mogelijk afhankelijk was van de inhoud van de case. Door een set met regels te pakken waarmee een ieder in zijn leven weleens in aanraking is gekomen, is getracht dit te voorkomen. Uit de beoordeling blijkt dit ook zo te zijn.

#### ***Ik ben tevreden met het resultaat***

Gemiddeld: 4,2, Modelleerervaring: 4, Niet modelleerervaring 4,3

Aangezien de wizard een tester een bepaalde kant op probeert te krijgen kan het zijn dat een gebruiker woorden, zinstypen en zinsbouw gebruikt waar de tester zich niet goed bij voelt of waarbij de aanwijzingen niet logisch lijken. Aan de hand van deze vraag is onderzocht of de tester zich ook kon vinden in de opgestelde regel. Dit blijkt inderdaad het geval.

#### ***Ik heb de regels gemaakt aan de hand van de beschrijving van de case en niet aan de hand van het feittypemodel***

Gemiddeld: 4, Modelleerervaring: 4, Niet modelleerervaring 4

Er zijn verschillende manieren om de nieuwe regels op te stellen; aan de hand van de bestaande regels, aan de hand van het feittypemodel of een combinatie van beide. Alle manieren zijn een goede insteek, maar uit de uitkomst blijkt dat alle testers gebruik maakte van het bestaande regels en niet zozeer van het feittypemodel. Dit is ook gebleken uit de observaties gedurende het testen. Verder bleek uit de observaties dat de testers met modelleerervaring wel meer aandacht gaven aan het feittypemodel dan de testers zonder modelleerervaring.

---

### 6.3.2 VRAGEN OVER DE EDITOR

---

***Ik had het gevoel dat ik controle had over de editor***

Gemiddeld: 4,5, Modelleerervaring: 4,3, Niet modelleerervaring 4,7

De editor en met name de wizard begeleidden de gebruiker door het proces. Het is belangrijk dat de gebruiker daarbij controle over de wizard houdt en niet andersom. Het kan bijvoorbeeld zo zijn dat een wizard de tester een kant opstuurt waar de tester helemaal niet naartoe wil. Door de volgorde van de deelprocessen vrij te laten werd die beperkende werking al enigzins opgeheven. Uit de antwoorden blijkt ook dat de testers zelf aangaven controle over de editor te hebben.

***Ik begreep de gevolgen van mijn antwoorden***

Gemiddeld: 4, Modelleerervaring: 3,7, Niet modelleerervaring 4,3

De wizard levert uiteindelijk een zinsbouw op welke bij voorkeur gebruikt moet gaan worden. Het is belangrijk dat de tester zich kan vinden in de antwoorden en dat de tester ook het gevoel heeft dat de voorgestelde zinsbouw een logisch gevolg is van zijn antwoorden.

***Ik had het gevoel dat verschillende keuzes daadwerkelijk tot andere conclusies leidden***

Gemiddeld: 4, Modelleerervaring: 4, Niet modelleerervaring 4

De wizard doorloopt een beslisboom die niet van tevoren bij de gebruiker bekend is. Het is daarom belangrijk dat de gebruiker ook doorheeft dat een verschillend antwoord op een vraag ook daadwerkelijk tot een andere uitkomst leidde. Uit zowel het antwoord als de observaties bleek dat de testers dit duidelijk doorhadden.

***De editor ondersteunt mij in het opstellen van de regels***

Gemiddeld: 3,6, Modelleerervaring: 3,3, Niet modelleerervaring 4

In deze vraag is te zien dat de testers zonder modelleerervaring meer ondersteuning ondervonden dan de testers met modelleerervaring. Dit komt voornamelijk doordat de wizard de gebruiker aan de hand neemt door de deelprocessen heen. Op deze manier worden er al concrete bouwstenen aangereikt waarmee een gebruiker kan beginnen. Voor de tester zonder modelleerervaring was dit erg handig om mee te beginnen.

***Zonder de editor had ik een hele andere set regels gehad***

Gemiddeld: 3,6, Modelleerervaring: 3,3, Niet modelleerervaring 4

Voornamelijk de testers zonder modelleerervaring zouden een hele andere set regels hebben gehad als zij niet een bepaalde richting op waren gestuurd. Uit gesprekken na de test gaven deze testers ook aan dat zij dan voornamelijk de regel zouden hebben overgeschreven aangezien ze in eerste instantie niet begrepen wat er 'mis' was met de bestaande regel. Zij gaven aan dat de editor ze bewust maakte van de sleutelwoorden (*moet, mag niet, etc.*) om een bepaalde 'lading' aan de regel te geven.

***Zonder de editor waren mijn regels van een lager niveau geweest***

Gemiddeld: 3.8, Modelleerervaring: 3.3, Niet modelleerervaring 4.3

Het doel van de editor was om regels van een hoger niveau te maken. De testers zelf wisten niet wat een kwalitatief goede regel was, maar toch kregen zij het gevoel dat ze het met de editor beter doen dan zonder de editor. Uit de interviews bleek dat dit voornamelijk werd veroorzaakt doordat de editor in alle rollen de tester concrete handvatten gaf om een regel op te bouwen.

***De interface was intuïtief en eenvoudig te gebruiken***

Gemiddeld: 4, Modelleerervaring: 3.6, Niet modelleerervaring 4.3

Ook al viel dit buiten de doelen van de test was dit toch een belangrijke vraag. Als de editor niet eenvoudig te gebruiken was zouden misschien de voorgestelde manier van werken verkeerd zijn. De testers gaven echter aan dat dit niet het geval was.

---

### 6.3.3 OPEN VRAGEN

---

Naast de gesloten vragen waren er ook een drietal open vragen. In deze vragen konden de gebruikers duidelijk maken wat zij misten aan de editor en wat de sterke en mindere punten waren van de editor.

***Wat was uw algemene indruk van de editor?***

De algemene indruk van alle testers is dat het tijdens het opstellen van de eerste regel nog een beetje onduidelijk is hoe de editor werkt. Tijdens de eerste regel werd er dan ook veel op de verschillende opties geklikt en de hulpteksten geraadpleegd. Nadat het proces van de wizard eenmaal volledig was doorlopen, hadden de testers door hoe het proces verliep. De regels daarna gingen allemaal veel sneller dan de eerste twee regels. Dit antwoord wordt ook ondersteund door de gemaakte observaties.

De testers gaven aan dat ze de wizard beschouwen als een nuttig hulpmiddel om de deelprocessen te expliciteren en te doorlopen. Bijna alle testers hadden enkele opmerkingen over de usability van de editor, echter deze opmerkingen vallen buiten de scope van de scriptie.

***Wat zou u willen veranderen aan de editor?***

Twee van drie testers zonder modelleerervaring gaven aan graag meer voorbeelden te zien bij de hulpteksten in de wizard. Bij alle vragen waren nu één á twee voorbeelden beschikbaar, maar de testers hadden graag nog wat meer houvast willen hebben in de vorm van nog meer voorbeelden.

Daarnaast zou men nog interactiever termen en feittypen toe willen voegen. Als men halverwege de wizard is en men komt er achter dat er nog een term of feittype mist zouden ze het liefst direct dit willen toevoegen. Op dit moment moest dan eerst de wizard worden afgerond om daarna de term of het feittype toe te voegen.



***Wat zou u willen toevoegen aan de editor?***

Op deze laatste vraag kwam van twee testers het antwoord dat zij graag zouden zien dat feittypen eenvoudiger toegevoegd zouden kunnen worden aan het feittypemodel. Zoals in het vorige hoofdstuk aangegeven is, is dit mogelijk door middel van de evaluerende rol. Helaas was de natuurlijke taal parser niet beschikbaar om directe input van de editor te verwerken. Mocht deze koppeling op termijn wel gerealiseerd kunnen worden dan zou dit verzoek ingewilligd kunnen worden.

Een andere tester gaf aan dat hij graag na afloop van de gemaakte regels het feittypemodel zou willen inzien om te bekijken of het feittypemodel goed leek. Ook dit is op termijn mogelijk in de editor, als er een natuurlijke taal parser beschikbaar komt.

---

#### 6.4 OBSERVATIES

---

Doordat er gebruik werd gemaakt van het think-aloud protocol, waren de sessies erg goed te observeren. Tijdens het testen is getracht zo min mogelijk aan de gebruiker uit te leggen zodat deze zelf zijn weg moest vinden. Enkel als de gebruiker echt vastliep werd er hulp geboden.

De meeste testers hadden wat moeite met het beginnen aan de wizard. Als de wizard wordt geopend is er nergens een uitleg die vertelt hoe er moet worden omgegaan met de wizard en hoe het proces van de wizard gaat verlopen. Dit kan echter eenvoudig worden opgevangen door het toevoegen van een korte uitleg op de pagina. Na een korte aanwijzing bij de eerste regel vonden de testers hun weg vanzelf door de editor en bij latere regels was hier dan ook geen onduidelijkheid meer over.

Tijdens de test bleek ook dat de testers bij de eerste regels erg lang deden over het opstellen van een regel en het doorlopen van de wizard. Na het opstellen van twee regels hadden ze echter door hoe het proces verliep en hoe ze de handreikingen van de wizard moesten interpreteren, zodat de latere regels in een veel hoger tempo gingen.

Wat verder opviel was dat de testers met modelleerervaring graag zo snel mogelijk een compleet feittypemodel wilde maken. Deze testers raakte vrij snel in de war wanneer het feittype dat zij graag wilden gebruiken niet in de aangereikte opties stond. Aan het begin van het testen was aangegeven dat het feittypemodel mogelijk incompleet was. De tester probeerde in eerste instantie de regel toch te vormen op basis van de aangereikte opties, terwijl dit in sommige gevallen onmogelijk was. De testers zonder modelleerervaring daarentegen bekommerden zich niet tot nauwelijks om het feittypemodel en nam de aangereikte opties ter kennisgeving aan en gingen vervolgens hun eigen weg. Voor deze testers is het dus belangrijk dat de editor in de evaluerende rol de gebruikte feittypen nogmaals toont en eventueel andere opties voorstelt.

---

## 6.5 REFLECTIE OP DE EVALUATIE

---

Zoals van tevoren bekend was kon alleen de ondersteunende rol van een editor volledig worden getest. De evaluerende rol en coachende rol konden maar voor een zeer klein gedeelte worden vervuld tijdens de test. Deze twee rollen zijn echter wel verantwoordelijk voor de implementatie van de meeste ‘intelligentie’ en geven de meest concrete handvatten aan een gebruiker. Dit is waarschijnlijk de verklaring waarom de kwaliteit van de regels maar enigzins verhoogd is. Tijdens de interviews met de testers bleek ook dat zij zeker de meerwaarde zagen in de evaluerende rol en de coachende rol en hiermee waarschijnlijk een hoger niveau regels zouden schrijven.

Verder gaven de beoordelaars van de bedrijfsregels aan dat er een drietal factoren aan te wijzen zijn waarom sommige regels niet verbeterd zijn;

- Als er wordt gekeken naar de regels waarbij geen kwaliteitsverbetering optrad, is te zien dat de meeste oorspronkelijke regels al niet zo goed waren. De oorspronkelijke regel was dan al vaag of onduidelijk. Bijvoorbeeld “*de examencommissie raadt elke examiner af om het cijfer 5,5 te geven*”. In SBVR is weinig ondersteuning voor dit type regels. De editor kon hierbij dan ook weinig ondersteuning geven en de nieuwe regels waren dan ook slecht.
- In enkele regels was de relatie tussen verschillende termen niet goed gedefinieerd doordat er een feittypen ontbreekt. Dit is een overtreding van de richtlijn “Het overslaan van feittypen is niet goed” van RuleSpeak. Dit is o.m. op te lossen door het gebruik van een natuurlijke taal parser. Iets minder dan de helft van de verslechterde regels werd veroorzaakt door deze fout.
- Tenslotte werden enkele negatieve oordelen veroorzaakt door een ‘onhandige’ wijziging van de terminologie wat leidde tot een betekeniswijziging. De editor kan echter niet verantwoordelijk worden gehouden voor de betekeniswijziging, aangezien hier de gebruiker altijd voor verantwoordelijk blijft.

Al met al zijn er enkele verklaringen aanwezig waarom de regels maar in beperkte mate zijn verbeterd. Na de implementatie van de evaluerende en coachende rol zal deze evaluatie herhaald moeten worden met een grotere groep testers om harde conclusies te mogen trekken.

---

## 7. CONCLUSIES

---

Aan de hand van zowel het theoretische onderzoek, als de implementatie van editor kunnen de onderzoeksvragen nu afdoende worden beantwoord. Eén voor één zullen de vragen doorlopen en beantwoord worden.

---

### WELKE ROL SPEELT DE EDITOR IN HET PROCES VAN HET OPSTELLEN VAN BEDRIJFSREGELS?

---

Deze vraag is beantwoord aan de hand van de volgende vragen.

---

#### HOE ZIET HET PROCES VAN HET OPSTELLEN VAN BEDRIJFSREGELS ERUIT?

---

Een bedrijfsregel bestaat uit een aantal elementen

- termen
- feittypen
- sleutelwoorden
- voorwaarden
- tijdsperiode

Al deze elementen moeten correct worden geïdentificeerd door de gebruiker om tot een goede bedrijfsregel te komen.

---

#### WELKE IDEEËN BESTAAN ER, IN DE WERELD VAN SOFTWAREONTWIKKELING, OVER DE ROL(LEN) DIE EEN EDITOR SPEELT IN DIT PROCES?

---

Aangezien in de literatuur weinig geschreven is over de rol van een editor, zijn de rollen via introspectie verkregen. Hieruit kwam naar voren dat een editor een viertal rollen kan spelen

- een faciliterende rol
- een ondersteunende rol
- een evaluerende rol
- een coachende rol

Voor elke van de rollen zijn de volgende vragen beantwoord.

---

#### WAT ZIJN DE GEVOLGEN VAN DEZE ROL VOOR HET PROCES?

---

Voor de faciliterende rol geldt dat vrijheid en creativiteit tijdens het proces op geen enkele manier worden ingeperkt. Door deze vrijheid is de kans op repetitiviteit van hetzelfde proces niet erg groot.

In de ondersteunende rol daarentegen wordt het proces aan banden gelegd. De gebruiker wordt aan de hand genomen en doorloopt verschillende deelprocessen. Doordat telkens dezelfde deelprocessen worden doorlopen is repetitiviteit veel hoger dan bij de andere rollen.

In de evaluerende rol wordt de vrijheid in het proces niet erg beperkt. De editor geeft enkel aan wat er geconstateerd wordt in de bedrijfsregel met de bedoeling om de gebruiker aan het denken te zetten. De gebruiker is echter vrij om de resultaten zelf te interpreteren hier wat mee te doen. De kans op repetitiviteit is hoog aangezien de editor telkens enkel constateert wat er in de regel staat. In principe zou de gebruiker daarom elke keer door de constatering moeten gaan wat kan resulteren in een saaie klus.

Als de editor de rol van coach aanneemt zal de vrijheid van de gebruiker niet worden beperkt. Er worden concrete aanbevelingen gedaan echter is het aan de gebruiker zelf om hier wat mee te doen. De kans op repetitiviteit is veel kleiner dan bijvoorbeeld bij de evaluerende rol aangezien er alleen aanwijzingen worden gegeven die op een concrete bedrijfsregel van toepassing zijn.

#### WAT ZIJN DE GEVOLGEN VAN DEZE ROL VOOR DE KWALITEIT VAN DE BEDRIJFSREGELS?

---

In de faciliterende rol is de kwaliteit van de bedrijfsregel volledig afhankelijk van de gebruiker. De gebruiker is vrij om de bedrijfsregel op elk mogelijke manier op te stellen en de editor biedt geen ondersteuning met betrekking tot de inhoud.

De ondersteunende rol van een editor zorgt ervoor dat de gebruiker zich bewust wordt van alle deelprocessen die doorlopen moeten worden tijdens het opstellen van een bedrijfsregel. Door deze expliciet uit te voeren mag worden aangenomen dat de gebruiker bewuster hierover nadenkt en de kwaliteit van het eindproduct dan hoger is dan in de faciliterende rol.

De evaluerende rol heeft als doel de gebruiker opnieuw te laten nadenken over de opgestelde bedrijfsregel. Door aan te geven wat de editor herkent in de regel zal de gebruiker aan het denken worden gezet en zich opnieuw afvragen of dit werkelijk de juiste bedrijfsregel is. De kwaliteit van de bedrijfsregel zal hierdoor ook hoger zijn dan in de faciliterende rol.

De coachende rol geeft concrete aanwijzingen die van toepassing zijn op een specifieke bedrijfsregel. Deze aanwijzingen geven concrete verbeterpunten aan in een regel en dit zal de kwaliteit van de bedrijfsregel dan ook ten goede komen. In de ideale situatie zijn alle richtlijnen uit RuleSpeak geïmplementeerd en zijn in elke regel alle richtlijnen opgevolgd. De kwaliteit van de regels zou daarmee zeer verbeterd zijn.

#### WAT VOOR EISEN WORDEN ER AAN DE GEBRUIKER GESTELD IN DEZE ROL?

---

Aangezien de faciliterende rol geen ondersteuning biedt met betrekking tot de inhoud zal de gebruiker een ervaren opsteller van bedrijfsregels moeten zijn. Deze rol is voor ervaren gebruikers.

De ondersteunende rol daarentegen is voor minder ervaren gebruikers die sturing nodig hebben in het proces van het opstellen van bedrijfsregels. De gebruiker wordt aan de hand genomen door een vraag- en antwoordspel en de gebruiker onderscheidt hier eenvoudiger de verschillende elementen die benodigd zijn voor een bedrijfsregel. Voor de ervaren gebruiker zal dit als vertragend en overbodig zijn.

De editor neemt in de evaluerende rol een deel van de benodigde ervaring over van de gebruiker. De editor geeft al aan welke termen en feittypen in de regel gebruikt worden. Op die manier hoeft de gebruiker dit niet meer zelf te doen.

De gebruiker zal in de coachende rol concrete aanwijzingen krijgen. Echter is het aan de gebruiker om deze aanwijzingen te kunnen vertalen naar verbetering aan de specifieke bedrijfsregel. Dit kan

een stuk ervaring en inzicht vragen van de gebruiker, maar met goede aanwijzingen kan ook de minder ervaren gebruiker hiermee uit de voeten.

### OP WELKE MANIER KAN DE EDITOR MEER ONDERSTEUNING BIEDEN AAN DE GEBRUIKER?

---

Deze vraag is beantwoord door de verschillende rollen in een nieuwe editor te implementeren. Hierbij is de editor RuleXpress als uitgangspunt genomen. Er is aangenomen dat RuleXpress voornamelijk ondersteuning biedt bij de faciliterende rol. Deze rol is daarom slechts zeer summier in de nieuwe editor opgenomen.

---

### WELKE MOGELIJKHEDEN ZIJN ER OM MEER ONDERSTEUNING TE BIEDEN AAN DE GEBRUIKER?

---

De ondersteunende, evaluerende en coachende rol zijn in de editor uitgewerkt. De faciliterende rol wordt al in voldoende mate ondersteund door de RuleXpress editor en viel buiten de scope van het onderzoek.

De ondersteunende rol is vervuld door middel van een wizard. Hierbij werd door middel van een vraag- en antwoordspel de gebruiker handvatten gegeven om een goede bedrijfsregel op te stellen. Aan de hand van de antwoorden werden verschillende sleutelwoorden aangereikt die een gebruiker kon gebruiken bij het formuleren van de bedrijfsregel.

De evaluerende rol is enkel als proof-of-concept geïmplementeerd. Met behulp van een natuurlijke taal parser is aangetoond dat het mogelijk is om de verschillende termen en feittypen in een bedrijfsregel te onderscheiden en aan de hand hiervan de gebruiker aanwijzingen te geven over het toevoegen en verwijderen van termen en feittypen.

De coachende rol is gedeeltelijk geïmplementeerd in de editor. De rol van coach is ingevuld door de richtlijnen van RuleSpeak te implementeren. Voor elk van de richtlijnen is beschreven of de richtlijn te automatiseren is en wat er nodig is om deze richtlijn te automatiseren. Er zijn een drietal benodigdheden voor het automatiseren van de richtlijnen:

- Het gebruiken maken van een string search in een regel
- Een natuurlijke taal parser die feittypen en meervoudsvormen uit een regel kan opleveren
- De betekenis van bepaalde woorden uit regels, bijvoorbeeld woorden die een tijdsindicatie of -periode aangeven.

De richtlijnen waar alleen de string search voor benodigd is zijn in de editor geïmplementeerd. De richtlijnen waarvoor een natuurlijke taal parser benodigd is zijn als proof-of-concept gedemonstreerd. Er is geen natuurlijke taal parser of methode beschikbaar die de betekenis van woorden uit een regel kan geven, daarom kunnen deze richtlijnen niet geïmplementeerd worden. Als alternatief is er voorgesteld om woordenlijsten te gebruiken die een correcte ondersteuning van de richtlijnen in ieder geval kunnen benaderen.

---

### HOE WORDEN DEZE AANPASSINGEN ERVAREN DOOR DE GEBRUIKER?

---

Uit de resultaten van de evaluatie bleek dat de gemaakte regels met de editor van een hoger niveau waren dan de oorspronkelijke regels. Hierbij moet worden opgemerkt dat sommige regels ook verslechterden door de editor te gebruiken, maar gemiddeld genomen waren de regels van een hoger niveau. Daarnaast was enkel de ondersteunende rol volledig geïmplementeerd en de evaluerende en coachende rol maar voor een klein gedeelte.

De testers gaven aan dat ze zich ondersteund voelden door de editor en het gevoel hadden dat ze door de editor te gebruiken betere regels zouden maken dan wanneer ze de editor niet hadden gebruikt.

---

### WAT ZIJN DE TECHNISCHE CONSEQUENTIES VAN DEZE AANPASSINGEN?

---

De ondersteunende rol kan eenvoudig worden geïmplementeerd door middel van een vraag- en antwoordspel. De technische consequenties hiervan zijn verwaarloosbaar.

Om de evaluerende rol te implementeren zal er een natuurlijke taal parser beschikbaar moeten zijn om termen en feittypen in een regel te kunnen herkennen. Deze elementen zijn nodig om de evaluerende rol in te vullen.

Om de coachende rol te implementeren zijn er een drietal benodigdheden:

- Het gebruiken maken van een string search in een regel
- Een natuurlijke taal parser die feittypen en meervoudsvormen uit een regel kan opleveren
- De betekenis van bepaalde woorden uit regels, bijvoorbeeld woorden die een tijdsindicatie of -periode aangeven.

De string search is in elke programmeertaal aanwezig en de technische consequenties zijn daarom verwaarloosbaar. Natuurlijke taal parsers zijn beschikbaar. Deze zullen gekoppeld moeten worden aan de applicatie zodat communicatie tussen beide componenten mogelijk is. In de geïmplementeerde editor kon deze koppeling helaas niet tot stand worden gebracht, waardoor voor de evaluerende en coachende rol enkel een proof-of-concept-implementatie is gemaakt.

Tenslotte mag er aangenomen worden dat er nog geen parser beschikbaar is die de betekenis van woorden kan afleiden. Deze kan dus ook niet worden geïmplementeerd. Als alternatief zouden hiervoor woordenlijsten kunnen dienen. Deze kunnen eenvoudig geïmplementeerd kunnen worden door de eerder genoemde string search en de technische consequenties zijn dan ook verwaarloosbaar.

## 8. TOEKOMSTIG WERK

---

In dit verslag is ingegaan op de verschillende rollen die een editor kan vervullen. Vervolgens is de ondersteunende rol van een editor geïmplementeerd en getest.

Door te experimenteren met een natuurlijke taal parser is ook een proof-of-concept gerealiseerd om feittypen te detecteren, te controleren of feittypen onderling verbonden waren en enkelvoud- en meervoudvormen te ontdekken. Dit proof-of-concept was een succes, en de evaluerende en coachende rol lijken dan ook rijp voor een verdere implementatie. De gebruikte parser was enkel in het Engels beschikbaar en kon niet eenvoudig gekoppeld worden aan de editor. In toekomstig werk is dit absoluut mogelijk en hierdoor zal de evaluerende en coachende rol dan ook daadwerkelijk in de praktijk getest kunnen worden. Door dit te doen wordt het iteratieve proces van opstellen ook beter ondersteund. Na elke aanpassing geeft de editor feedback over eventuele nieuwe verbeteringen en zodoende kunnen regels keer op keer verbeterd worden.

Daarnaast vindt de evaluerende en coachende rol op dit moment plaats op regelniveau. In een andere fase van het project is er ook gekeken om deze rollen op regelset-niveau plaats te vinden. Op deze manier kunnen bijvoorbeeld veel voorkomende zinspatronen herkend worden. Deze zinspatronen kunnen vervolgens worden toegevoegd aan de editor, zodat deze dit soort constructies kan herkennen. Omwille van de tijd liet de scope van het afstudeertraject het niet toe om hier verder onderzoek naar te doen. In de toekomst zou het echter zeer nuttig zijn om te kijken naar de regels als set in plaats van enkel naar de regels als individu.

Ten slotte is het doel van dit afstudeerproject nooit geweest om een perfecte editor te bouwen, maar om het idee van de verschillende rollen binnen een editor te onderzoeken. In toekomstig werk zou de editor echter wel op een betere manier geïmplementeerd kunnen worden, zodat alle foutjes en onzuiverheden uit de editor gehaald kunnen worden om vervolgens nog beter te kunnen testen met gebruikers of er daadwerkelijk betere regels kunnen worden gemaakt aan de hand van een volledige geïmplementeerde editor. Tevens zou er gekeken kunnen worden om sommige functionaliteiten op een andere manier in te vullen. Zo kan er bijvoorbeeld onderzoek kunnen worden gedaan naar de vraag of gebruikers liever drag-and-drop-functionaliteit gebruiken om de bouwstenen van de wizard bij elkaar te zetten.

## 9. VOCABULAIRE

---

Bedrijfsregel	<i>Een statement dat een aspect van het bedrijf definieert of beperkt. Het is bedoeld om de bedrijfsstructuur te bepalen of te controleren of om het gedrag van het bedrijf te beïnvloeden. Bedrijfsregels zijn atomair ~ oftewel, ze kunnen niet verder worden onderverdeeld.</i>
Term	Een woord of zinsdeel met een specifieke betekenis voor het bedrijf in een bepaalde context
Feittype	Een associatie tussen twee of meer termen.
SBVR	Semantics For Business Vocabulary and Rules. Een productstandaard voor het opstellen van bedrijfsregels
RuleSpeak	Een verzameling regelpatronen om SBVR-conforme bedrijfregels op te stellen. De verzameling is ontworpen door Ron Ross.
Operationele regel	Instructie die aangeeft hoe actoren (mensen en systemen) in de bedrijfsvoering in bepaalde situaties moeten handelen
Structurele regel	Regel die een noodzakelijkheid uitdrukt. Een structurele regel kan niet overtreden worden.



---

## BIBLIOGRAFIE

---

- Blankena, F. (2008). Eindelijk: een formele taal voor de 'business'. *Automatisering Gids*.
- Business Rules Group. (2003, november 1). *The Business Rules Manifesto*. Opgeroepen op september 15, 2009, van The Business Rules Manifesto: <http://www.businessrulesgroup.org/brmanifesto.htm>
- Coenen, A., Hermans, L., Roosmalen, M. v., & Spreeuwenberg, S. (2008). *Uw bedrijf geregeld met Business Rule Management*. Den Haag: Sdu Uitgevers.
- Hay, D., & Healy, K. A. (2000). *Defining Business Rules ~ What Are They Really?*
- Hoppenbrouwers, S., & Schotten, B. (2009). *A Game Prototype for Basic Process Model Elicitation*.
- Hoppenbrouwers, S., Bommel, P. v., & Järvinen, A. (2008). *Method Engineering as Game Design: an Emerging HCI Perspective on Methods and CASE Tools*.
- Hoppenbrouwers, S., Proper, H., & Weide, T. P. (2005). Formal Modelling as a Grounded Conversation. *Proceedings of the 10th International Working Conference on the Language Action Perspective on Communication Modelling*, (pp. 139-155). Kiruna, Sweden, EU.
- Hoppenbrouwers, S., Weigand, H., & Rouwette, E. (2009). *Setting Rules of Play for Collaborative Modeling*.
- IsecT Ltd. (sd). *ISO/IEC 27000*. Opgeroepen op februari 1, 2010, van <http://www.iso27001security.com/html/faq.html>
- Kletke, M., Mackay, J., Barr, S., & Jones, B. (2000). *Creativity in the organization: the role of individual creative problem solving and computer support*.
- Lubart, T. (2005). *How can computers be partners in the creative process*.
- Object Management Group. (2008, januari). *Semantics of Business Vocabulary and Business Rules*. Opgehaald van <http://www.omg.org/spec/SBVR/1.0/PDF>
- Ross, R. G. (2001). *Basic RuleSpeak Guidelines* (Versie 2.2 ed.).
- Ross, R. G. (2005). *Business Rule Concepts*.
- Selker, T. (2005). *Fostering motivation and creativity for computer users*.
- Selker, T. (2005). *Fostering motivation and creativity for computer users*.
- Spreeuwenberg, S. (2009, september). *Rule Authoring Is a Creative Process*. Opgeroepen op februari 1, 2010, van Business Rules Community: <http://www.brcommunity.com/b497.php>
- Spreeuwenberg, S. (2009, maart). *What happened to the B and the M of BRM?* Opgeroepen op februari 1, 2010, van Business Rules Community: <http://www.brcommunity.com/b467.php>
- Spreeuwenberg, S., & Healy, K. A. (2009). *SBVR's Approach to Controlled Natural Language*.

Wason, P., & Johnson-Laird, P. (1972). *Psychology of Reasoning: Structure and Content*. Harvard University Press.

## APPENDIX A. DE EDITORS

---

In deze appendix zullen de verschillende versies van de editor worden toegelicht. Kort zal bekeken worden wat voor functionaliteit de editor biedt en waarom deze functionaliteit is aangeboden. Daarnaast wordt er gekeken welke problemen naar voren kwamen bij het evalueren van de versies en oplossingen geboden om deze problemen in een vervolgversie op te lossen.

### EERSTE VERSIE

---

De eerste versie was er vooral opgericht om de constructiefase van bedrijfsregels in te vullen. De opzet was om de ondersteunende rol in te vullen door de gebruiker stapsgewijs enkele “invuloefeningen” te laten doen.

Het proces was als volgt;

- Selecteer termen die in de bedrijfsregel gebruikt moeten worden. Deze werden ingeladen vanuit een bestand met bestaande termen. Er kon een bestaande term worden geselecteerd of een nieuwe term worden toegevoegd.
- Selecteer het zintype. Door het zintype te selecteren kon een voorzet worden gedaan voor de opbouw van de zinstructuur. Deze was gebaseerd op RuleSpeak. RuleSpeak geeft bijvoorbeeld aan dat als er een verplichting wordt aangegeven de zin de vorm “... moet ...” moet hebben. De mogelijkheden waren als volgt;

Zintype	Structuur
Een verplichting aangeven	... moet ...
Aangeven dat iets niet is toegestaan	... mag niet ...
Aangeven dat iets onder voorwaarden is toegestaan	... mag alleen ...
Een advies geven	... mag ...
Aangeven dat iets niet verplicht is	... hoeft niet ...

- Selecteer eventuele voorwaarden aan de regel. Ook deze processtap is ingegeven door RuleSpeak. Als er voorwaarden aan de regel zitten, moet er een gedeelte aan de zin worden toegevoegd met de structuur “... indien ...”.

Alle geselecteerde onderdelen verschenen in afzonderlijke velden. Als laatste stap kon dan in een vrij tekstveld de regel worden ingevoerd, waarbij het uiteraard de bedoeling was om de geselecteerde onderdelen als bouwstenen te gebruiken.

De bedoeling van deze eerste versie was vooral om eerste gedachten te vertalen naar een concrete applicatie om hier daadwerkelijk aan verder te bouwen.

---

### OPENSTAANDE ISSUES EN PROBLEMEN

---

- Het aantal zintypen en -structuren was erg beperkt.
- Geen kwaliteitsmeting over ingevoerde bedrijfsregels
- Geen sturing in het proces

---

## TWEEDE VERSIE

---

De tweede versie was niet zozeer een verbetering van de eerste versie, maar eerder een uitbreiding. Waar de eerste versie enkel keek naar de constructiefase van een bedrijfsregel, schonk de tweede versie aandacht aan de deconstructiefase van een bedrijfsregel.

Ten eerste werden bestaande regels uit een xml-bestand gelezen. Vervolgens werd voor elke bedrijfsregel de gebruikte termen aangegeven. Deze werden al in het xml-bestand aangegeven met tags.

Daarnaast werd geprobeerd door middel van pattern matching te kijken of er bepaalde zinspatronen herkend konden worden. De gedachte hierachter was dat als in een zin geen enkel zinspatroon herkend kon worden er nader naar deze zin moest worden gekeken. Óf er moest een nieuwe zinstructuur worden gedefinieerd, óf de zin moest worden geherformuleerd zodat er wel een patroon werd gebruikt.

Ook deze editor-versie was voornamelijk om wat vage gedachten omtrent de deconstructiefase te concretiseren naar een echte editor. Ook deze versie was verre van volledig, maar in ieder geval voldoende om inzicht te geven in de deconstructiefase.

---

## DERDE VERSIE

---

De derde versie probeerde een evaluerende rol te implementeren die de gehele set met regels evalueerde. Op setniveau probeerde deze editor patronen te herkennen die in meerdere regels voorkwamen maar (nog) niet als zinstructuur was onderkend. Ook gaf deze versie hoe vaak een zinstructuur gebruikt is. Op deze manier kan bijvoorbeeld afgeleid worden dat er in een set met regels nog geen enkele structurele regel is gedefinieerd. Dit zou een aanwijzing kunnen zijn om eens kritisch naar de striktheid van de bedrijfsregels te kijken.

---

## OPENSTAANDE ISSUES EN PROBLEMEN

---

Al snel in het implementatieproces kwamen er moeilijkheden naar voren. Zo steeg de complexiteit van de analyse exponentieel, aangezien alle regels met alle regels vergeleken moesten worden om patronen te herkennen.

Er bleken implementatiealgoritmen voorhanden te zijn die hier beter mee om konden gaan, echter in samenspraak met de begeleiders is besloten dat dit onderwerp buiten de scope van het project valt. Het analyseren van een set van regels is daarom een onderwerp dat in een vervolgtraject nog nader bekeken kan worden.

---

## VIERDE VERSIE

---

Deze versie trachtte de evaluerende versie in kaart te brengen. Via Kees Koster was er een natuurlijke taal parser beschikbaar die op grond van een regel een parse kon geven bestaande uit factoids. Deze factoids waren ongeveer de feittypen die gedefinieerd worden in bedrijfsregels.

Door middel van deze parser konden er dus regels geëvalueerd gaan worden, en konden verschillende RuleSpeak-richtlijnen, waar een natuurlijke taal parser voor benodigd was, gecontroleerd gaan worden.

Met behulp van de parser kon een Proof-of-Concept worden aangetoond van de implementatie van de RuleSpeak richtlijnen.

---

## OPENSTAANDE ISSUES EN PROBLEMEN

---

Er waren twee problemen aanwezig in deze versie van de editor. Enerzijds was de natuurlijke taal alleen beschikbaar voor de Engelse taal. Tot dan toe was er gewerkt aan een Nederlandse taal editor. Kees Koster gaf aan een Nederlandse taal parser beschikbaar te hebben binnen enkele maanden, echter dit was na de geplande afstudeerdatum. Er is daarom besloten om een proof of concept te maken voor enkele Engelstalige zinnen en op die manier aan te tonen dat een evaluerende rol mogelijk moet zijn om te implementeren voor een editor.

Een ander probleem was dat de parser enkel als shell script onder linux beschikbaar was. Deze kon niet eenvoudig geïntegreerd worden met de website, waardoor het 'live' invoeren van data en de evaluatie daarvan niet via de website kon gebeuren. Daarom zijn er zinnen ingevoerd en is de output als tekstbestand opgeslagen. Vervolgens heeft de editor deze tekstbestanden ingelezen, geparseerd en hierop zijn evaluerende rol toegepast. In een volledige geïntegreerde versie zou dit uiteraard automatisch moeten gebeuren.

---

## VIJFDE VERSIE

---

Deze versie was de eindversie van het product. Alle vier rollen werd geïmplementeerd in deze versie. Zo kon een gebruiker eerst zelf kiezen door middel van welke rol hij het proces wil starten. Dit was enerzijds mogelijk via de faciliterende rol, anderzijds door middel van de ondersteunende rol.

Nadat een van deze rollen uitgevoerd was kwamen de evaluerende en coachende rol in actie. Voor het volledig werken van deze twee rollen was het nodig dat er een natuurlijke taal parser beschikbaar is in de editor. Zoals tijdens de implementatie van de vorige versie van de editor was besloten, zou dit niet worden gedaan. Zodoende konden de evaluerende en coachende rol van het product niet volledig worden geïmplementeerd. Enkel de eenvoudige richtlijnen waar alleen een string search voor nodig was, zijn geïmplementeerd.

Aan de hand van deze editor is er vervolgens met verschillende gebruikers getest. De openstaande issues zijn in de evaluatie in hoofdstuk 6 uitgebreid aan bod geweest.

---

## APPENDIX B. DE CASE

---

De testcase bestond uit een set met oorspronkelijke regels, die vervolgens aan de hand van de editor moesten worden aangepast. De tester kreeg een (incomplete) set van termen en een (incomplete) feittypemodel aangereikt.

---

### DE OORSPRONKELIJKE REGELS

---

## 2. Tentamens

### 2.1 Vorm van de tentamens

1. Aan het begin van elke cursus deelt de examinator de vorm van de bijbehorende tentamens mede aan de studenten. De vorm waarin tentamens kunnen worden afgenomen is beschreven in de OER'en.
2. De examinator kan voor 1 of meerdere studenten met instemming van die studenten een andere vorm van het tentamen afspreken. De uitslag van een dergelijk tentamen wordt door de examinator middels een "cijfer verwerken formulier" aan de administratie van de faculteit doorgegeven.
3. Studenten die conform de OER een verzoek indienen voor een andere of aangepaste vorm van 1 of meerdere tentamens dienen dit schriftelijk te doen.

### 2.2 Tentamendata

1. Voor de aanvang van elk studiejaar stelt de examencommissie de data van alle tentamens vast op voorstel van het onderwijs instituut. Het aantal tentamens per onderwijsonderdeel dient aan de vereisten in de OER'en te voldoen. De data dienen zodanig gespreid te zijn dat de studeerbaarheid van elke opleiding gegarandeerd is
2. Wijziging van de in het eerste lid genoemde data vindt uitsluitend plaats in geval van overmacht. Dit dient op geschikte wijze aan de betrokken studenten doorgegeven te worden.
3. De tentamendata dienen zowel voor de regeling van de aanmelding en deelname (zie artikel 2.3) als voor de regeling van tentamenactiviteiten die afhankelijk zijn van de vorm van een tentamen (zie artikel 2.5)

### 2.5 Tentamenactiviteiten

1. Voor cursussen waarvan de tentamens de vorm "schriftelijk" bevatten, wordt ook de begin- en eindtijd van het schriftelijk tentamen vastgesteld voor aanvang van het studiejaar.
2. Voor cursussen waarvan de tentamens de vorm "mondeling" bevatten, wordt het mondeling deel voor de tentamendatum of binnen 8 dagen na de tentamendatum afgelegd. Het tijdstip van het mondeling voor elke student wordt door de examinator na overleg met de student vastgesteld. Hierbij wordt rekening gehouden met de andere studieactiviteiten van de student.
3. Voor cursussen waarvan de tentamens een "practicum" vorm bevatten, dient op de tentamendatum de examinator de beschikking te hebben over alle stukken die nodig zijn om een oordeel over een student te vormen.

4. De examiner kan voor een mogelijk hertentamen in hetzelfde studiejaar van een vak met een “practicum” vorm een student in de gelegenheid stellen voor een of meer “practicum” onderdelen een nieuw of aangepast werkstuk, verslag, computerprogramma, etc. in te dienen. Dit kan alleen wanneer daar geen extra onderwijsfaciliteiten voor nodig zijn zoals bijvoorbeeld begeleiding door een docent, toezicht van ander personeel.

## **2.7 Beoordeling en inzage van de tentamens**

1. Wanneer een tentamen verschillende onderdelen bevat, wordt de manier waarop uit de beoordelingen van elk onderdeel het eindoordeel gevormd wordt, aan het begin van elke cursus door de examiner aan de student meegedeeld.

2. Voor een schriftelijk tentamen geldt dat de relatieve gewichten van de diverse opgaven aan het begin van het schriftelijk bij de studenten bekend dienen te zijn.

3. De beoordeling van een tentamen is zodanig dat de geëxamineerde kan nagaan hoe de uitslag van zijn tentamen tot stand is gekomen.

4. De examiner draagt er zorg voor dat een geëxamineerde gebruik kan maken van het in de OER'en genoemde inzagerecht.

5. De in de OER'en genoemde inzage in het beoordeelde werk en de kennisneming van vragen, opdrachten en normen geschiedt op het secretariaat van het onderwijs instituut, tenzij de examiner besluit dit op andere wijze te regelen.

6. Na het vaststellen van de uitslag van een mondeling tentamen vindt op verzoek van de geëxamineerde dan wel de examiner direct een gesprek plaats waarin de gegeven uitslag toegelicht wordt.

7. De vaststelling en bekendmaking van een tentamenuitslag vindt plaats binnen de termijn en op de wijze die in de OER'en is vastgelegd. Voor tentamens waarvan dit niet in een OER is vastgelegd geldt de OER regeling voor schriftelijke tentamens tenzij de examencommissie op verzoek van de examiner een andere regeling vaststelt.

8. De examiner kan informeel de uitslag van een tentamen schriftelijk, mondeling of per email aan een student meedelen.

9. De uitslag van een tentamen kan bestaan uit de gehele of halve cijfers tussen de 1 en de 10, inclusief de grenzen, “voldaan”, “niet voldaan” of “vrijstelling”. De examencommissie raadt elke examiner af om het cijfer 5,5 te geven.



---

## DE TERMEN

---

De wet: de Wet op het Hoger onderwijs en Wetenschappelijk onderzoek afgekort tot WHW en zoals sindsdien gewijzigd.

Opleiding: de bacheloropleiding of masteropleiding alsmede schakelprogramma's.

Student: hij of zij die is ingeschreven aan de Radboud Universiteit Nijmegen voor het volgen van het onderwijs en/of het afleggen van de tentamens en de examens van de opleiding

Practicum: een praktische oefening als bedoeld in art. 7.13, lid 2 onder d van de wet, in één van de volgende vormen: het maken van een scriptie; het maken van een werkstuk of een proefontwerp; het uitvoeren van een ontwerp- of onderzoekopdracht; het verrichten van een literatuurstudie; het schrijven van een computerprogramma; het verrichten van een stage; het deelnemen aan veldwerk of een excursie; het uitvoeren van proeven en experimenten; of het deelnemen aan een andere onderwijsactiviteit, die gericht is op het bereiken van bepaalde vaardigheden.

Tentamen: een onderzoek naar de kennis, het inzicht en de vaardigheden van de student met betrekking tot een bepaalde onderwijseenheid, alsmede de beoordeling van dat onderzoek door minstens één daartoe door de examencommissie aangewezen examinator.

Tentamdatum: de administratieve datum van een tentamen gebruikt om de handelingen van alle betrokkenen rond een tentamen te regelen. Bij een schriftelijk tentamen is dit de datum van het schriftelijk, bij een practicum de datum waarop alles ingeleverd dient te zijn.

Examen: toetsing, waarbij door de examencommissie wordt vastgesteld of alle tentamens van de tot een examen behorende onderwijseenheden met goed gevolg zijn afgelegd, voor zover de examencommissie niet heeft bepaald dat het examen tevens omvat een door haar zelf te verrichten onderzoek naar de kennis, inzicht en vaardigheden van de examinandus alsmede de beoordeling van de uitkomsten van dat onderzoek. (conform artikel 7.10 van de wet).

Examencommissie: de examencommissie van een opleiding ingesteld conform artikel 7.12 van de wet. Zie ook Structuurregeling RU.

Examinator: degene die door de examencommissie wordt aangewezen ten behoeve van het afnemen van tentamens, conform artikel 7.12 van de wet.

ec: studiepunten conform het European Credit Transfer System.

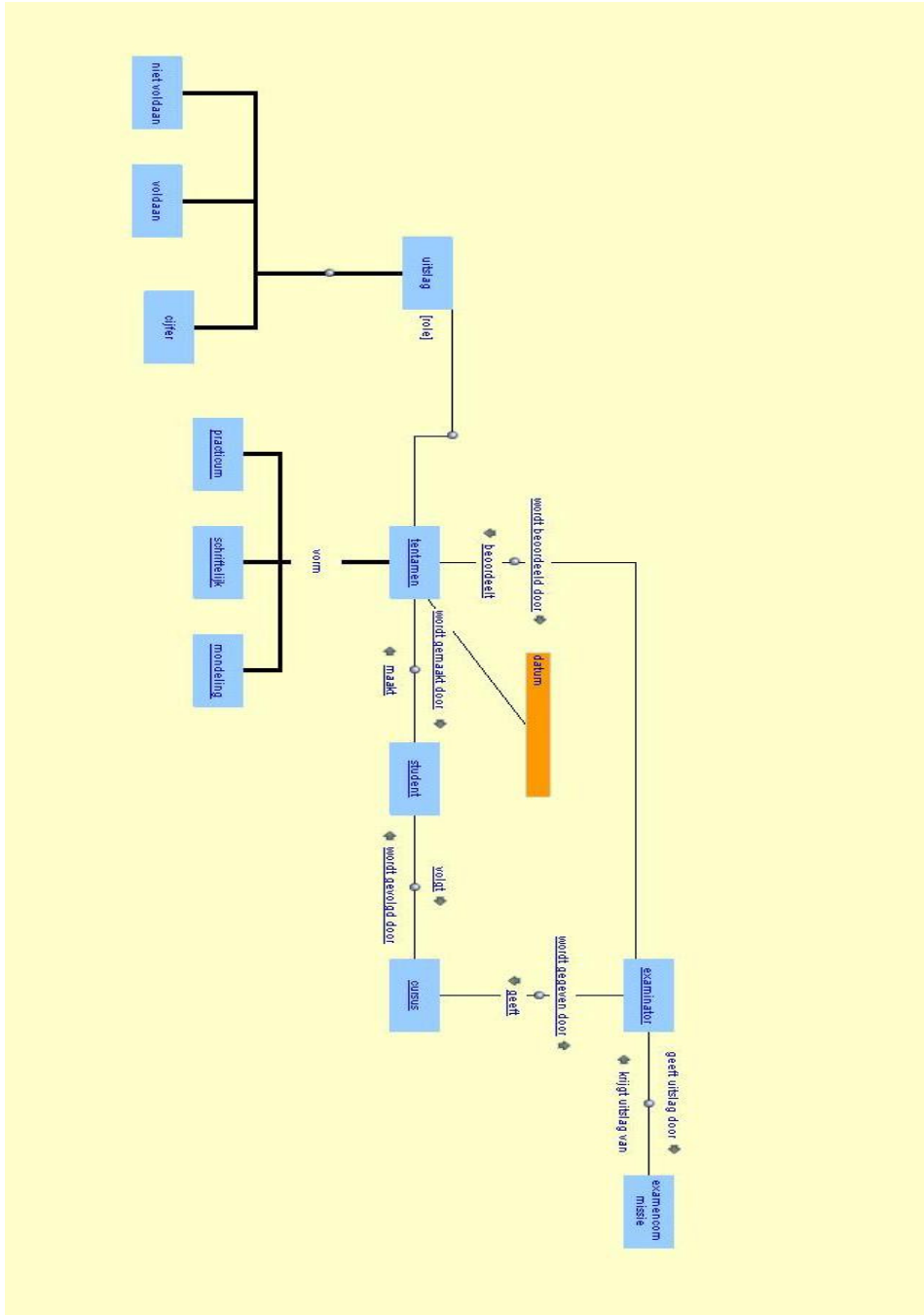
Werkdag: maandag t/m vrijdag m.u.v. de erkende feestdagen.

Instelling: Radboud Universiteit Nijmegen.

OER: de Onderwijs en Examenregeling die voor een student van toepassing is.

## HET FEITTYPEMODEL

Het onderstaande feittypemodel is tijdens het afnemen van de test uitgereikt aan de testers. Hierbij is aangegeven dat het feittypemodel niet compleet is maar een eerste indicatie geeft wat de relatie is tussen de verschillende termen.



## APPENDIX C. HET OPSTELLEN VAN EEN BEDRIJFSREGEL MET BEHULP VAN DE EDITOR

---

In deze appendix zal een voorbeeldregel worden ingevoerd met behulp van de editor.

Als voorbeeld zal de volgende regel opnieuw worden opgesteld door middel van de wizard:

*"Aan het begin van elke cursus deelt de examinerator de vorm van de bijbehorende tentamens mede aan de studenten."*

Als eerste zullen de termen worden geselecteerd die terugkomen in de bestaande regel.

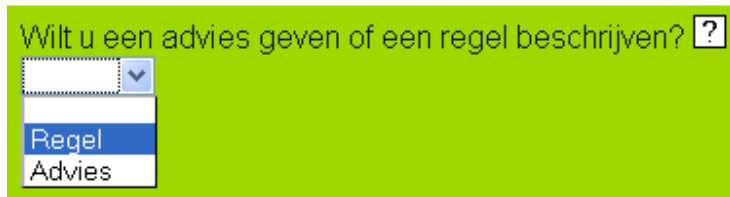


De termen *"cursus"*, *"examinator"*, *"tentamen"* en *"student"* zijn al als termen gedefinieerd en zullen in ieder geval geselecteerd worden. De term *"(tentamen)vorm"* is nog niet gedefinieerd. De gebruiker kan er voor kiezen om *"(tentamen)vorm"* eerst als nieuwe term toe te voegen of dit pas te doen nadat de regel is geformuleerd.

Als de termen zijn geselecteerd worden deze toegevoegd aan het 'kladblaadje' met bouwstenen. Hierbij worden gelijk alle feittypen weergegeven die op dit moment aanwezig zijn in het feittypemodel. Feittypen die voor deze regel niet gebruikt gaan worden, kunnen in verband met de overzichtelijkheid weggeklikt worden door middel van het min-teken voor het feittype.

Termen	Bijbehorende feittypen
student	<input type="checkbox"/> <i>student volgt cursus</i> <input type="checkbox"/> <i>student maakt tentamen</i> <input type="checkbox"/> <i>student stemt in met uitzondering</i>
tentamen	<input type="checkbox"/> <i>examinator beoordeelt tentamen</i> <input type="checkbox"/> <i>student maakt tentamen</i> <input type="checkbox"/> <i>tentamen heeft uitslag</i> <input type="checkbox"/> <i>tentamen sluit af cursus</i>
cursus	<input type="checkbox"/> <i>student volgt cursus</i> <input type="checkbox"/> <i>examinator geeft cursus</i> <input type="checkbox"/> <i>tentamen sluit af cursus</i>
examinator	<input type="checkbox"/> <i>examinator geeft cursus</i> <input type="checkbox"/> <i>examinator beoordeelt tentamen</i>

In de volgende stap zal de gebruiker het zintype moeten gaan selecteren.

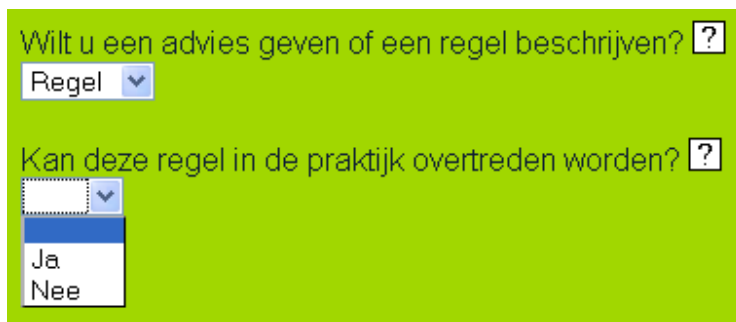


Wilt u een advies geven of een regel beschrijven? ?

Regel

Advies

Als eerste wordt de vraag gesteld of er advies wordt gegeven of een regel wordt beschreven. In dit geval zal de vrijheid worden beperkt (de examiner moet namelijk iets doen) en zal er dus een regel worden beschreven. De keuze voor “regel” is de enige correcte optie. Doordat deze keuze wordt gemaakt, verschijnt een nieuwe vraag.



Wilt u een advies geven of een regel beschrijven? ?

Regel

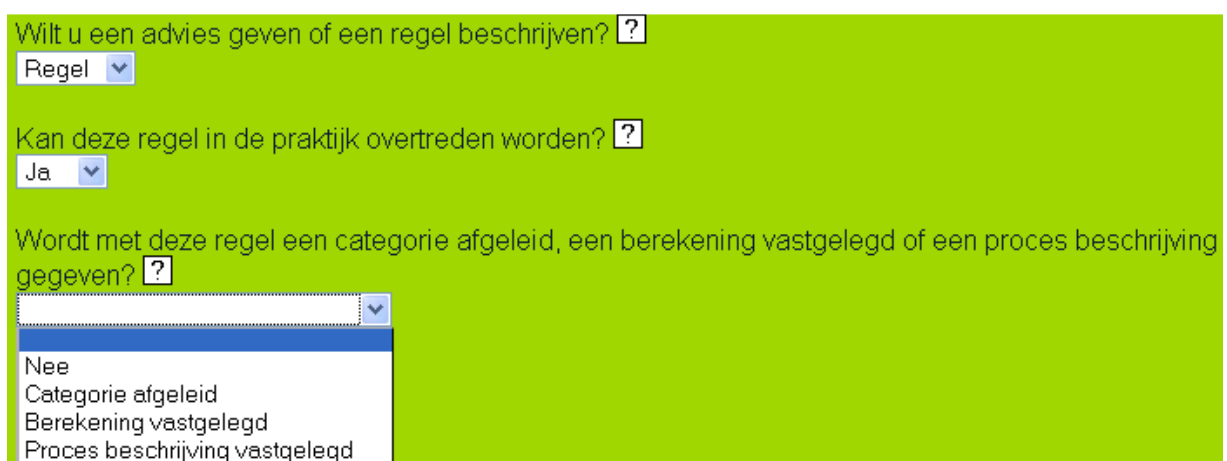
Kan deze regel in de praktijk overtreden worden? ?

Ja

Nee

Deze vraag is een modelleerkeuze en hierbij is dus niet maar één goed antwoord mogelijk. Een keuze voor “Ja” zal resulteren in het gebruik van de woorden “moet”, “mag alleen” of “mag niet”. De keuze voor “Nee” zal resulteren in de ‘zwaardere’ woorden “altijd”, “nooit” of “kan alleen”. In dit voorbeeld gaan we er vanuit dat deze regel in de praktijk wel overtreden kan worden. Er wordt vanuit gegaan dat een examiner ook kan vergeten om dit mede te delen.

Tenslotte wordt er gevraagd of er sprake is van een speciaal geval. Dit zijn drie vooraf gedefinieerde gevallen, die overgenomen zijn uit RuleSpeak.



Wilt u een advies geven of een regel beschrijven? ?

Regel

Kan deze regel in de praktijk overtreden worden? ?

Ja

Wordt met deze regel een categorie afgeleid, een berekening vastgelegd of een proces beschrijving gegeven? ?

Nee

Categorie afgeleid

Berekening vastgelegd

Proces beschrijving vastgelegd

In ons geval betreft het niet een speciaal geval en zal er dus met “nee” worden geantwoord.

Dit heeft als resultaat dat de bouwstenen “moet”, “mag niet” en “mag ... alleen” worden toegevoegd aan het ‘kladblaadje’. Ook de alternatieven “dient” en “mag alleen” worden toegevoegd.

Termen	Bijbehorende feittypen	Zintype
student	<input type="checkbox"/> <i>student</i> volgt <i>cursus</i>	... moet ...
	<input type="checkbox"/> <i>student</i> maakt <i>tentamen</i>	... mag niet ...
	<input type="checkbox"/> <i>student</i> stemt in met <i>uitzondering</i>	... mag ... alleen ...
tentamen	<input type="checkbox"/> <i>examinator</i> beoordeelt <i>tentamen</i>	<i>Alternatieven</i> ... dient ...
	<input type="checkbox"/> <i>student</i> maakt <i>tentamen</i>	... mag alleen ...
	<input type="checkbox"/> <i>tentamen</i> heeft <i>uitslag</i>	
	<input type="checkbox"/> <i>tentamen</i> sluit af <i>cursus</i>	

Tenslotte vraagt de editor of er nog voorwaarden zijn te onderscheiden in de regel. Dit is bij deze regel niet het geval, want deze regel geldt voor alle gevallen.

Zitten er voorwaarden aan de regel?

- Ja
- Nee

Nu alle bouwstenen zijn geselecteerd kan begonnen worden met het opstellen van de regel. Dit gebeurt in het volgende tekstvak.

**Naam regel:**

**Regel:**

*Door gebruik te maken van de verschillende geselecteerde bouwstenen kunt u hier de regel formuleren*

*Probeer de regel te beginnen met het woord "Een" en de volgende structuur voor de regel aan te houden:*

```
<term> <zintype-sleutelwoord> <feittype-werkwoord> <term>
<tijdsindicatie> <voorwaarden>
```

Voordat de gebruiker begint met opstellen, worden er nog enkele handvatten gegeven om te beginnen. De gebruiker zal vervolgens moeten proberen de verschillende termen en het zintype samen te voegen tot een regel. Een voorbeeld van de regel zou kunnen zijn:

*"Een examinerator moet de vorm van het tentamen mededelen aan de studenten aan het begin van de cursus."*

Nadat de regels is opgeslagen gaat de editor over in de rol van evaluator en coach. In de evaluatie-rol geeft de editor feedback over wat er in de regel is geconstateerd. Helaas is deze rol niet functioneel in deze versie van de editor. De verklaring hiervoor is te vinden in hoofdstuk 5.5.

Geconstateerd			
Termen	Bijbehorende feittypen	Soort zin	Structuur
student tentamen cursus examinator		operationele regel	*moet*


De enige onderdelen die geïdentificeerd kunnen worden zijn de termen, die al gedefinieerd zijn in het vocabulaire en het type zin. In volledig werkende editor zouden ook de feittypen uit de regel kunnen worden gehaald, evenals alle termen. Aan de hand van deze feittypen en termen kan de editor vervolgens voorstellen doen om feittypen en/of termen toe te voegen. In dit geval zou bijvoorbeeld *"de vorm"* als voorstel kunnen worden gedaan.

Tenslotte zal ook de coachende rol van de editor gebruikt worden. Omwille van het voorbeeld is de regel veranderd in:

*“De examiner moet de vorm van het tentamen mededelen aan de studenten aan het begin van de cursus.”*

De editor zal vervolgens de volgende hint geven.

#### Opmerkingen

- Een regel moet vaak met "een" beginnen. 

Een gebruiker kan aan de hand van deze opmerking opnieuw nadenken over zijn regel. Eventueel kan via het vraagteken achter de opmerking extra informatie worden opgezocht over deze opmerking.