

2013•2014
FACULTEIT BEDRIJFSECONOMISCHE WETENSCHAPPEN
*master in de toegepaste economische wetenschappen:
handelsingenieur in de beleidsinformatica*

Masterproef

De waarde en maturiteit van The Decision Modeler en Operational Decision Manager

Promotor :
Prof. dr. Koenraad VANHOOF

Tim Vanbrabant
*Proefschrift ingediend tot het behalen van de graad van master in de toegepaste
economische wetenschappen: handelsingenieur in de beleidsinformatica*

2013•2014

FACULTEIT BEDRIJFSECONOMISCHE WETENSCHAPPEN

*master in de toegepaste economische wetenschappen:
handelsingenieur in de beleidsinformatica*

Masterproef

De waarde en maturiteit van The Decision Modeler en
Operational Decision Manager

Promotor :
Prof. dr. Koenraad VANHOOF

Tim Vanbrabant

*Proefschrift ingediend tot het behalen van de graad van master in de toegepaste
economische wetenschappen: handelsingenieur in de beleidsinformatica*

WOORD VOORAF

Deze masterproef vormt het sluitstuk van mijn opleiding master Handelsingenieur in de beleidsinformatica ICT. Het onderwerp van deze masterproef 'De waarde en maturiteit van The Decision Modeler en Operational Decision Manager' vormt een mooi afsluitstuk op deze opleiding. Als handelsingenieur in de beleidsinformatica werd er steeds verwacht dat wij als studenten probeerden zowel kennis op te nemen van de businesswereld als van de informaticawereld. Dit opdat we zelf een schakel konden vormen tussen deze twee 'werelden'. Via deze masterproef werd nogmaals de waarde benadrukt van tools en omgevingen die het mogelijk maken voor businessmensen en informatici om elkaar beter te begrijpen en makkelijker met elkaar te kunnen samenwerken.

Graag zou ik van deze gelegenheid gebruik willen maken om de personen te bedanken die me geholpen hebben bij het verwezenlijken van deze eindverhandeling. In het bijzonder denk ik aan mijn promotor Prof. dr. Koen Vanhoof en de heer Frank Vanhoenshoven van het bedrijf AGServ. Ze hebben me begeleid, gemotiveerd en begrip getoond zodat ik deze thesis tot een goed einde kon brengen. Het bedrijf AGServ heeft de Operational Decision Manager omgeving voor me opengesteld. En dhr. Vanhoenshoven heeft me de omgeving geïntroduceerd en me er doorheen begeleid.

Tim Vanbrabant

Hasselt, mei 2014

SAMENVATTING

In deze masterproef werd onderzocht op welke manier de tool The Decision Modeler (TDM) van Bizdesign en het Business Rules Management Systeem (BRMS) Operational Decision Manager (ODM) van IBM best gebruikt en gecombineerd kunnen worden om aan decision management te doen. Daarnaast werd er onderzocht voor welke bedrijfsfuncties TDM en ODM geschikt zijn en wat de maturiteit van beiden is. Om deze aspecten te onderzoeken werd er een literatuurstudie gedaan rond business proces- en decision management. Daarnaast werd er met behulp van een voorbeeld rond autoverhuur nagegaan welke functionaliteiten, best practices en beperkingen TDM en ODM hebben.

In de huidige competitieve bedrijfswereld zijn business process management en business decision management niet meer weg te denken. Bedrijven dienen mee te evolueren om niet achterop te geraken. Daarom dienen bedrijven hun processen zo flexibel mogelijk te maken. Dit om aanpassingen in het beleid, in de wetgeving of de omgeving te kunnen opvangen op een vlotte manier. Deze flexibiliteit kan verkregen worden door de business logica uit de bedrijfsprocessen te destilleren. Op die manier hebben veranderingen een minder sterk effect op de processen zelf, maar kunnen ze opgevangen worden door een onafhankelijk business logicamodel. Bedrijven kunnen hiervoor gebruik maken van business decision management. De logica neemt dan de vorm aan van beslissingen. Deze zijn makkelijker aan te passen en kunnen ook hergebruikt worden in verschillende processen. Ze worden namelijk zo opgesteld dat ze technologieonafhankelijk zijn.

Voor deze masterproef werd er specifiek gekeken naar het nog 'jonge' 'The Decision Model' dat ontwikkeld werd door B. Von Halle en L. Goldberg in hun gelijknamige boek: The Decision Model. Deze methode maakt het mogelijk beslissingen logisch te structureren met behulp van een beslissingsmodel en ondersteunende rule families die verschillende condities bevatten die samen leiden tot een bepaalde actie. De methodologie van 'The Decision Model' is gebaseerd op 15 principes en 3 normen waaraan voldaan moet worden, om een correct model te creëren.

Voor 'The Decision Model' werd er specifiek binnen de Architect omgeving van Bizdesign een plug-in ontwikkeld die volledig gebaseerd is op het gelijknamige boek. De TDM plug-in vormt een modelleringomgeving en maakt het mogelijk beslissingen uit te werken met behulp van een decision model en rule families. Binnen Architect kan het decision model gelinkt worden met andere modellen zoals een business process model. Deze integratie maakt het makkelijk te zien waar beslissingen genomen worden en waar dit invloed zal hebben. De integratie kan wel nog beter aangezien de link wel bestaat, maar het nog geen effect heeft. Zo zullen uitkomsten van bepaalde beslissingen geen invloed hebben op de gevolgde weg in het business process model terwijl dit in realiteit wel zou moeten zijn. Dankzij het gebruik van TDM kan een gewone business user de gewenste business logica al ruw schetsen, wat de complexiteit voor de ontwikkelaar kan verlagen.

Het Business Rules Management Systeem ODM van IBM vormt een ontwikkeling- en executieomgeving voor business logica. Een ontwikkelaar formuleert in een Eclipse-gebaseerde console de juiste klassen en attributen die nodig zijn voor het creëren van de rules. Omdat business users vaak pas betrokken worden bij de business rules wanneer ze al gecreëerd zijn, zou

TDM hier verandering in kunnen brengen. Door in TDM decision models, rule families en UML klassen te creëren, kan de gebruiker zijn wensen en verwachtingen al aan de ontwikkelaar tonen. De ontwikkelaar dient op basis van die UML klassen, Java klassen aan te maken die samen een eXecution Object Model (XOM) vormen. Op basis van de XOM wordt er een Business Object Model (BOM) ontwikkeld. De BOM-vocabulary kan gebruikt worden om rules, decision tables en decision trees te creëren. Eenmaal dit voltooid is, dient de ontwikkelaar de rules te debuggen, te controleren, te testen en uit te voeren. Vanaf dat moment zijn de rules beschikbaar voor de business user. Indien de rules toch nog niet volledig naar wens van de business user zijn, dan kan hij/zij binnen ODM gebruik maken van de Business Console die toelaat rules te creëren of te wijzigen en hierover met andere gebruikers te communiceren in een web 2.0. omgeving.

De hierboven aangehaalde tools kunnen een meerwaarde bieden voor het business decision management domein. Iedere tool op zijn eigen manier. Toch bieden de tools niet enkel apart meerwaarde, ook gecombineerd kunnen ze extra waarde creëren. Dit werd in deze masterproef onderzocht door gebruik te maken van de Unified theory van V. Venkatesh, M. Moris, G.B. Davis en F. D. Davis. die betrekking heeft op de het accepteren van nieuwe technologieën. De combinatie van beide tools laat business users toe om tijdens het hele ontwikkelingsproces van de business rules actief bij te dragen. Dit zorgt er niet enkel voor dat de business users zich meer betrokken voelen bij het project en de technologie daarom sneller zullen accepteren, het zorgt er ook voor dat het werk van de ontwikkelaar in ODM sneller er correcter kan gebeuren.

Naast de waarde van TDM en ODM werd ook hun maturiteit onderzocht. Voor TDM werd er gebruik gemaakt van het Business Decision Maturity Model. TDM verkreeg onlangs een level 2 certificaat. Level 2 (agile) maturiteit houdt in dat er flexibiliteit in de tool mogelijk is. Veranderingen in het beleid die invloed hebben op de business logica kunnen binnen een bepaald project snel aangepast worden dankzij de transparantie met betrekking tot de beslissingen die in TDM gemodelleerd zijn. Daarnaast houdt het ook in dat rules herbruikbaar zijn binnen een project, maar niet overheen verschillende projecten. Ook dit is correct voor TDM daar rules gelinkt kunnen worden aan meerdere modellen binnen één project, maar nog niet aan meerdere projecten tegelijk. Eenmaal deze feature in TDM wordt opgenomen kan TDM evolueren naar een level 3 maturiteit.

Ook voor ODM werd de maturiteit onderzocht. Voor dit BRMS werd gebruik gemaakt van de hype cycle en priority matrix van Gartner. ODM wordt geplaatst in de derde fase van deze cyclus, met name de 'trough of disillusionment'. De elementen die zich in deze fase bevinden, bestaan al even en zullen ofwel matuur worden ofwel stilaan verdwijnen. Dit is afhankelijk van het succes van de use cases en de kracht van de BRMS aanbieders. Omdat verwachtingen van gebruikers in het begin meestal onrealistisch zijn, en deze niet allemaal ingelost kunnen worden, leidt dit er toe dat gebruikers voorzigtiger verwachtingen zullen stellen.

In de Priority matrix kreeg het BRMS, de 'high benefit rating'. Dit omdat het nieuwe wegen biedt om applicaties te sturen en te beheren, wat kan leiden tot kostenreductie en flexibiliteit. Het maturiteitslabel dat het BRMS van Gartner verkreeg is 'Adolescent', dit houdt in dat de functionaliteiten van de technologie matuur worden. De systemen zijn meestal al de tweede of derde generatie waardoor er steeds minder aangepast wordt aan het aanbod.

INHOUDSOPGAVE

Woord vooraf.....	I
Samenvatting.....	III
Inhoudsopgave	V
Hoofdstuk 1: Probleemstelling	1
1.1. Nood aan verandering.....	1
1.2. Tools die de oplossing kunnen bieden	2
Hoofdstuk 2: Literatuurstudie.....	3
2.1. Business process management	3
2.1.1. Business Process Management life cycle.....	3
2.2. Decision modeling & Management.....	5
2.2.1. Decision management.....	5
2.2.2. Decision modeling	7
2.2.3. Business rules:	9
2.2.4. Het business rules management systeem.....	11
2.2.5. The decision model	12
Hoofdstuk 3: The decision modeler & Operational Decision Manager	17
3.1. The decision Modeler by Bizzdesign.....	17
3.2. Operational decision manager by IBM	20
3.2.1. De rule designer.....	22
3.2.2. Het gebruiken van de business console (business users)	29
3.2.3. De rollen binnen ODM.....	31
Hoofdstuk 4: Toepassen van de tools	33
4.1. Stap 1.4. Het ontwikkelen van de procesmodellen in TDM.....	34
4.2. Stap 1.5. Het ontwikkelen van de businessrules op basis van de procesmodellen en beslissingsmodellen in ODM.....	46
4.2.1. Eligibility.....	47
4.2.2. Car Group availability.....	49
4.2.3. Pricing	51
4.2.4. Rental agreement registrering.....	53
4.2.5. Scenario testing	53

Hoofdstuk 5: De waarde en maturiteit van The decision modeler & Operational decision Manager	57
5.1. Waarde van The decision modeler (Architect) by bizzdesign	57
5.1.1. Usability	58
5.1.2. Functional features	59
5.1.3. Standards compliance/Interoperability	62
5.1.4. Support/Team modeling	63
5.1.5. Financial value	63
5.1.6. Algemene beoordeling	64
5.2. Waarde van Operational Decision Manager by IBM	64
5.2.1. Technical assessment	65
5.2.2. Commercial assessment:	67
5.3. Waarde van de combinatie van TDM en ODM	68
5.4. De maturiteit van TDM	71
5.5. De maturiteit van ODM	73
Hoofdstuk 6: Conclusie	77
Lijst van de geraadpleegde werken	79
Bijlage	81
Bijlage A: overzicht figuren en tabellen	81
Overzicht figuren:	81
Overzicht tabellen:	82
Bijlage B: De principes en normaalvormen van The Decision Model	83
Structurele simpliciteit	83
Declaratieve opbouw	85
Optimale integriteit	86
Bijlage C: Uitwerken van praktijkvoorbeeld	87
Stap 1.1 Het modelleren van het functionele model:	87
Stap 1.2. Het modelleren van het structurele model	90
Stap 1.3. Het modelleren van het gedragsmodel	91
Bijlage D: The business decision maturity model	92

HOOFDSTUK 1: PROBLEEMSTELLING

Het eerste hoofdstuk zal de masterproef inleiden en de vooropgestelde probleemstelling gedetailleerder toelichten. Met name heerst er op het gebied van business rules en meer specifiek, het decision model, een tekort aan gebruiksvriendelijke, efficiënte tools. Daar business rules vaak veranderd moeten worden en deze makkelijker aan te passen zijn dan de processen zelf, kan het modelleren en managen ervan een voordeel opleveren. De voorbije maanden en jaren boden steeds meer bedrijven tools aan die dit tekort konden opvullen. Daar de markt nog steeds relatief jong is, is een grondige analyse van de bestaande tools een must. Dit om de waarde en maturiteit van deze tools aan het licht te brengen.

1.1. NOOD AAN VERANDERING

In de huidige bedrijfswereld zorgt de correcte aaneenschakeling van processen dat het vooropgestelde doel makkelijker bereikt wordt. Omdat de markt niet stilstaat en de concurrenten ook steeds mee veranderen, is het noodzakelijk dat het bedrijf ook zelf steeds veranderingen en verbeteringen aanbrengt in de eigen processen om niet achterop te geraken.

Vaak moeten deze veranderingen in de bestaande processen geïmplementeerd worden. Vaak zijn processen te complex of te uitgebreid omdat er te veel logica in de processen vervat zit. Het probleem wat zich dan stelt, is een beperkte flexibiliteit. Dit maakt het moeilijk veranderingen correct door te voeren. Daarnaast kan een bepaalde verandering in een proces invloeden hebben op diverse aspecten binnen het proces. Wanneer zulk proces dan zeer complex is, wordt het na een verandering verwarrend om het proces nog correct en logisch uit te voeren.

Bij het doorvoeren van de veranderingen, worden vaak IT mensen ingeschakeld omdat zij kennis hebben van de achterliggende systemen. Het probleem wordt al snel duidelijk. Indien het bedrijf steeds bij iedere verandering de IT specialisten dient in te schakelen, wordt dit al snel zeer kostelijk. Daarnaast is het voor de IT specialist niet altijd even duidelijk wat er precies veranderd moet worden omdat IT specialisten en businessmensen een andere omgangstaal gebruiken. De taal van de business en de taal van IT. Dus ook daar schuilt er een inefficiëntie die mogelijk opgelost kan worden door het destilleren van de logica uit de processen. Dit dan onder de vorm van onafhankelijke business rules. Dit maakt niet enkel de processen minder complex, maar maakt een aanpassing van een business rule ook makkelijker door deze onafhankelijkheid. Op die manier zouden businessmensen zelf in staat moeten kunnen zijn om bij kleine veranderingen in het beleid of de regelgeving, de business rules aan te passen zonder hiervoor steeds de IT specialisten in te roepen. Om dit mogelijk te maken dienen gebruiksvriendelijke tools op het juiste niveau beschikbaar gemaakt te worden.

1.2. TOOLS DIE DE OPLOSSING KUNNEN BIEDEN

Deze thesis zal specifiek onderzoeken of de aangeboden tools, met name Architect met bijbehorende TDM (decision modeling en simulatie) plug-in en ODM (decision modeling en execution) daadwerkelijk de flexibiliteit van processen ondersteunen door een gebruiksvriendelijke omgeving aan te bieden waarin zowel businessmensen als ontwikkelaars de processen en voornamelijk de businessrules kunnen managen.

De keuze voor TDM en ODM heeft meerdere redenen. Zo diende er binnen het gegeven tijdsbestek een keuze gemaakt te worden daar iedere tool toch de nodige tijd vraagt om er mee te leren werken. Daarnaast heerst er het fenomeen dat er vaak tools bestaan voor grotere ondernemingen die zich dure en zeer uitgebreide tools kunnen veroorloven. Dit is voor KMO's minder het geval. Zij geraken vaak achterop door het beperkte aanbod aan tools die betaalbaar en efficiënt zijn. Grote bedrijven hebben vaak IT departementen waarop ze kunnen terugvallen wanneer er veranderingen doorgevoerd worden. KMO's daarentegen hebben vaak weinig of geen echte IT mensen in hun bedrijf. Daardoor is het voor hen nog belangrijker dat er tools bestaan die businessmensen kunnen gebruiken. Bizzdesign, dat de tool Architect (en TDM) ontwikkelde rond de methodologie 'The decision model', is één van de eerste bedrijven die een betaalbare tool uitbrengt voor KMO's. Daarom ook de interesse binnen deze masterproef om deze tool grondiger te analyseren. Daar Bizzdesign meldt dat het werkt aan een mogelijke link van hun tool naar het Business rules Management systeem 'Operational Decision Manager' van IBM, leek het interessant deze link dan ook zelf te analyseren. Dit om te achterhalen in welke mate een combinatie van deze twee tools een voordeel kan opleveren.

De probleemstelling en topics waarrond deze masterproef verduidelijking zal proberen te brengen, hebben dan ook betrekking op de waarde die de module TDM (Architect) kan bieden in het modelleren en simuleren van de processen en de bijbehorende business rules. Daarnaast wordt nagegaan welke functies binnen een bedrijf van deze tool gebruik zouden kunnen maken.

Volgens de auteurs van het boek 'The decision model' waarop de TDM module gebaseerd werd, kan dit model helpen business logica van de processen te scheiden voor een makkelijker onderhoud en automatisering van de logica zelf. De beslissingen worden hierdoor ook technologieonafhankelijk. Dit zal het makkelijker maken om toekomstige technologieën te implementeren.

Naast de mogelijkheden van iedere tool, dienen ook de maturiteit en de limieten van iedere tool onderzocht te worden. De maturiteit zal beperkt aan bod komen, daar deze tools nog redelijk nieuw zijn, waardoor wetenschappelijke modellen voor deze soort tools nog in ontwikkeling zijn.

Ook voor ODM van IBM zal een dergelijke analyse gebeuren. Het grote verschil met TDM is het grotere geheel wat ODM vormt. Binnen ODM kunnen rules gemodelleerd en getest worden, maar daarnaast omvat het ook een volledige executieomgeving wat ODM een compleet BRMS maakt. Het is dan ook onmogelijk de tools op gelijke hoogte met elkaar te vergelijken, daar deze immens verschillen in omvang en mogelijkheden. Toch is het interessant om na te gaan hoe deze tools op elkaar kunnen inwerken en hoe bedrijven het beste uit deze tools kunnen halen wanneer ze gecombineerd worden.

HOOFDSTUK 2: LITERATUURSTUDIE

In dit hoofdstuk worden de begrippen die betrekking hebben op deze masterproef gedetailleerd beschreven. De eerste paragraaf behandelt business proces management en vloeit geleidelijk over in de bespreking decision modeling en decision management.

2.1. BUSINESS PROCESS MANAGEMENT

In deze paragraaf wordt de eerste stap richting een flexibel, beheersbaar proces verduidelijkt. Wanneer een bedrijf de businesslogica snel en correct mogelijk wil aanpassen, dienen de processen hiervoor aangepast te worden. Een proces dient namelijk zo gemodelleerd te worden dat de businesslogica die eruit gedestilleerd wordt, toegepast kan worden in de business rule management systemen. Daarom is het zo belangrijk hier in het begin al rekening mee te houden in het business process management.

2.1.1. BUSINESS PROCESS MANAGEMENT LIFE CYCLE

Processen vormen de sequentie van activiteiten die uitgevoerd worden om bepaalde producten of diensten te leveren. Zoals eerder aangehaald dient het bedrijf deze processen te modelleren en te managen, dit omdat doorheen de tijd, de markt, de technologie, etc. , verandert. Daarmee dient ook het bedrijf zijn processen goed op te volgen en aan te passen om competitief sterk te staan. Business Process Management kan dan ook gezien worden als het systeem dat het mogelijk maakt de processen op een hoger niveau te analyseren om zo verbeteringen in prestaties zoals kostvermindering, snelheid en service mogelijk te maken. (Reijers H. 2004)

Business Process Management (BPM) life cycle:

Planning	Het ontwikkelen van een plan en strategie met betrekking tot de processen. Het plan dient in overeenstemming te zijn met de strategieën, processen en systemen. Daarnaast worden rollen, verantwoordelijken, doelen en verwachtingen vastgelegd, samen met de prestatie maatstaven.
Analyse	Het begrijpen van de huidige processen met betrekking tot de gewenste doelen en objectieven. Dit houdt in: het analyseren van de procesmodellen, het evalueren van de prestatie maatstaven en het doorgronden van externe veranderingen die een impact kunnen hebben op het bedrijf.
Design & modellering	Het creëren van representaties van bestaande of toekomstige businessprocessen door de sequentie van activiteiten te documenteren en te bepalen wat bepaalde processen inhouden. Ook bevat deze stap het creëren van nieuwe specificaties voor businessprocessen die reeds bestonden.
Implementatie	Het uitvoeren van de activiteiten die werden opgesteld voor de processen.

Monitoren & controleren	Het verschaffen van voldoende informatie aan de beslissingsnemers zodat aanpassingen met betrekking tot resources gemaakt kunnen worden. Vaak is dit prestatiegerelateerde informatie gebaseerd op de maatstaven verbonden met de doelstellingen van de organisatie.
Verbetering	Het implementeren van de resultaten die verkregen worden door iteratieve analyse en design. Dit omvat het management van veranderingen en het verbeteren van de activiteiten. Met name de re-engineering en re-designing op basis van de geanalyseerde prestaties.

Tabel 1: De business proces management cyclus.

Uit [Vertaald en verwerkt] "Business process management, a systemic approach," door M. Segatto, S.I. Dallavalle de Padua en D.P. Martinelli, 2013, Business Process Management Journal, 19,4, p. 706

Het is zeer belangrijk dat deze levenscyclus ook werkelijk als een cyclus gebruikt wordt. Business Process Management is een continu proces dat de prestaties van een bedrijfsproces voortdurend opvolgt en probeert te koppelen aan de vooropgestelde doelen. Het is niet de sequentie van deze fases die het allerbelangrijkste is, maar wel het inzicht in het management van businessprocessen dat het mogelijk maakt de prestaties van het bedrijf te verbeteren. Het voornaamste binnen BPM is dan ook het procesdenken, met name het identificeren, beheren, uitvoeren en bijsturen van de processen. (Reijers H. 2004)

Het voorzien van de nodige tools op het niveau van het middelmanagement zou processen manageable moeten maken zodat ze makkelijker te verbeteren zijn. Dit houdt in dat processen continu gemonitord en gecontroleerd moeten worden om tijdig te kunnen bijsturen. Wanneer processen namelijk niet manageable zijn, maar hard-coded, is het zeer moeilijk de nodige veranderingen door te voeren.

Om processen manageable te maken is het noodzakelijk tijdens het procesdesign er rekening mee te houden dat uit de procesmodellen de beslissingslogica gedestilleerd wordt. Het gebruiken van beslissingen, zowel de automatische als de niet-automatische, zorgt voor die nodige flexibiliteit waardoor er een lerend proces ontstaat. Het doel van deze flexibiliteit is het drukken van kosten en het verhogen van de proceskwaliteit door het maken van tijdige aanpassingen en verbeteringen. Een proces is een lerend proces wanneer het wordt aangepast op basis van leerervaringen die bepaalde activiteiten sterker benadrukken of onderdrukken afhankelijk van het gekozen pad.

Het gebruik van beslissingen heeft invloed op de hele Business Process Management cyclus. Het gebruiken van onafhankelijke beslissingen heeft betrekking op de eerste design en modellering fase, maar ook op de verbetering en aanpassing fase éénmaal het proces geïmplementeerd is. Dit zijn de vijfde en de zesde fase, met name 'Monitoren & Controleren' en 'Verbeteren'.

Om meer inzicht te krijgen in de hierboven vermelde beslissingen en bijbehorende business rules, wordt de volgende paragraaf hieraan gewijd.

2.2. DECISION MODELING & MANAGEMENT

De tweede paragraaf biedt een gedetailleerd overzicht over het modelleren en managen van beslissingen. Wanneer processen gemodelleerd worden, dient er nagegaan te worden in welke delen van het proces beslissingen verborgen zitten. Eenmaal deze geëxtraheerd kunnen worden, kunnen deze beslissingen apart gemodelleerd worden om ze onafhankelijk te maken. Wel dient de link met het proces duidelijk te blijven zodat de beslissingen invloed hebben waar nodig.

2.2.1. DECISION MANAGEMENT

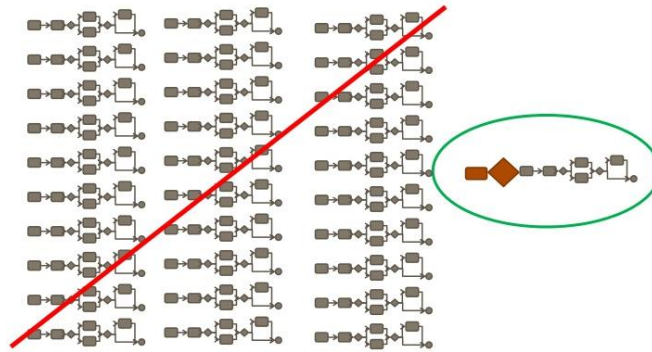
Om processen flexibel en managebaar te maken, kan er gebruik gemaakt worden van decisions. Hierbij is het belangrijk de noodzakelijke beslissingen te identificeren en te managen. Sommigen van deze beslissingen kunnen geautomatiseerd worden, maar toch moet de mogelijkheid blijven bestaan aanpassingen door te voeren. Zowel procesmanagement als decision management is hierbij van belang om de complexiteit van veranderingen te verminderen. Daarnaast geeft het de business users meer controle over de processen en kritieke beslissingen. (Taylor,2012)

Business Decision Management is de kunde om intelligente en snelle beslissingen te managen.

Drie elementen vormen hierbij de basis (Von Halle, Goldberg, 2010):

- ✚ **Business Motivation:** Het business plan met de objectieven dat met behulp van business decisions geïmplementeerd moeten worden.
- ✚ **Business metrics:** Metingen van de resultaten van de business decisions in vergelijking met de vooropgestelde objectieven.
- ✚ **Business Logic:** Dit is de logica die de business decisions ondersteunt en geïmplementeerd wordt om de objectieven te halen.

Belangrijk hierbij is te onthouden dat decisions geen processen zijn, zij ondersteunen het proces. Een business process is namelijk een serie van herhaalbare gedefinieerde activiteiten die in een bepaalde sequentie worden uitgevoerd door actoren (individueen of systemen) binnen een bepaalde scope. Zo dient er gebruik gemaakt te worden van decision-aware processen die onderscheid maken tussen verwerkingstaken en beslissingstaken, waarbij conclusies getrokken worden op basis van de businesslogica. Het is daarbij belangrijk te beseffen dat een business proces procedureel is en een business decision declaratief. Procedureel wil zeggen dat stap voor stap wordt getoond hoe iets verwezenlijkt wordt, met de nadruk op **hoe**. Declaratief wil zeggen dat het specificeert wat gedaan moet worden, ongeacht de volgorde van de businesslogica, met de nadruk op **wat**. Verbeteringen in het proces zullen leiden tot een verhoogde werk-efficiëntie terwijl verbeteringen in de business decisions leiden tot intelligentere business logica. (Von Halle, Goldberg, 2010)



Figuur 1: Omvormen complexe naar simpele processen
 Uit [Overgenomen uit] "increasing-bpm-agility-and-effectiveness-with-decision-management", door J. Taylor, 2011, Decision Management solutions, presentatie

Vaak zitten zulke beslissingen in het proces verrat. Het probleem bij embedded decisions in een proces is dat ze de complexiteit van een proces sterk verhogen. Daarom is het belangrijk de beslissing onafhankelijk van het procesmodel te maken met behulp van het decision model. Dit valt dan onder de noemer decision management. (Taylor, 2011)



Figuur 2: Voordelen Onafhankelijk management van beslissingen
 Uit [aangepast uit] "increasing-bpm-agility-and-effectiveness-with-decision-management", door J. Taylor, 2011, Decision Management solutions, presentatie

Wanneer een bedrijf zijn bestaande processen wil verbeteren met behulp van decision management, dienen er drie fases doorlopen te worden. De eerste fase is de Decision Discovery waarbij de analist de beslissingen moet proberen te identificeren. Dit om de beslissingspunten die in het proces verrat zaten te destilleren. Deze beslissingen worden in de tweede fase verder uitgewerkt tot onafhankelijke beslissingsprocessen. Fase 3 omvat het toepassen van analyses en metingen op de resultaten om mogelijke verbeteringen aan te kaarten (Taylor, 2011)

In deze masterproef ligt de nadruk op de tweede fase. Met name het uitwerken van de beslissingen met behulp van decision modeling. Dit houdt in het aanmaken van business rules die afhankelijk van verschillende voorwaarden tot bepaalde acties leiden.

2.2.2. DECISION MODELING

Het modelleren van beslissingen vormt een kritisch punt, daar de ideeën en doelen zo nauwkeurig mogelijk uitgewerkt moeten worden. Het belang van business rules uit zich in kostreductie, betere beslissingen en hogere reactiesnelheden. Ze is vooral belangrijk wanneer beslissingen vaak veranderen, complex zijn en het bedrijf er analyses op wil toepassen.

Kenmerken van een beslissing (Taylor, 2012):

- ✚ **Repeatable:** Dit betekent dat iedere beslissing steeds een bepaalde plaats in het proces heeft. Daarnaast is er voor iedere beslissing steeds een zelfde consistente set informatie beschikbaar om de beslissing te maken. Ook de mogelijke acties dienen consistent te blijven overheen verschillende beslissingen.
- ✚ **Niet-triviaal:** Een beslissing moet een bepaalde complexiteit bevatten, met name rekening houden met het beleid en de reguleringen.
- ✚ **Meetbare business impact:** Voor iedere beslissing moet het mogelijk zijn om de kost te zien van slechte beslissingen en de waarde van goede beslissingen.
- ✚ **Kandidaat voor automatisatie:** Naast de drie vorige voorwaarden, dient de beslissing ook automatiseerbaar te zijn. Als er steeds een menselijk oordeel nodig is, dan is de beslissing niet geschikt voor een Decision Management System.

Types van beslissingen (Taylor, 2012):

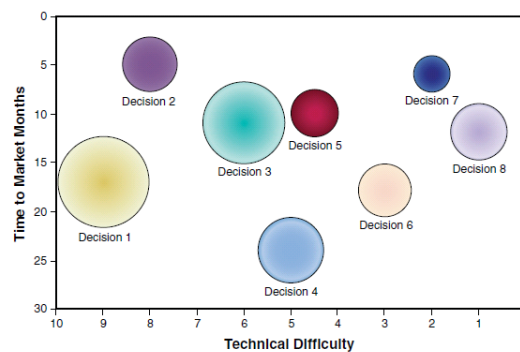
- ✚ **Eligibility (toelating):** Deze beslissingen bepalen of een bepaalde persoon of organisatie de toestemming krijgt iets te doen of te verkrijgen. Deze zijn meestal zeer geschikt voor automatisatie.
- ✚ **Validation (Valideren):** Meestal vormen deze beslissingen een barrière voor een proces om verder te gaan. Pas wanneer de validatie gebeurd is, wordt het proces verder afgehandeld. Ook deze beslissingen zijn meestal geschikt voor automatisatie.
- ✚ **Calculation:** Het berekenen van een bepaalde prijs kan geautomatiseerd worden. Het kan soms afhankelijk zijn van persoon, product en tijd en kan dus zeer complex zijn, wat het non-triviaal maakt.
- ✚ **Risk decisions:** Vaak zijn deze beslissingen zeer belangrijk omdat er een groot verschil is tussen een foute en een goede beslissing. Deze beslissingen kunnen geautomatiseerd worden op basis van de kennis en ervaring van experts.

De bedoeling is uiteindelijk om een organisatiebreed systeem te bouwen dat alle beslissingen omvat. Toch kan het soms efficiënter zijn te focussen op een bepaald proces om zo het hele systeem stap voor stap op te bouwen. Om te vinden welke beslissingen van invloed zijn op een bepaald proces, kunnen er vergaderingen gehouden worden met de verschillende belangenpartijen. Daarnaast kunnen zowel processen en events geanalyseerd worden. Eénmaal de mogelijke beslissingen duidelijk zijn, dient er nagegaan te worden wat deze beslissingen precies inhouden. Met name welke eigenschappen ze hebben, hoe verschillende beslissingen gelinkt zijn en welke informatie beschikbaar moet zijn voor een bepaalde beslissing. Daarnaast dient het duidelijk te zijn welke conclusies een bepaalde beslissing als resultaat kan hebben.

Eigenschappen van een beslissing waarmee rekening gehouden dient te worden (Taylor, 2012):

- ✚ **Volume:** Hoe vaak dient een bepaalde beslissing gemaakt te worden.
- ✚ **Timeliness:** Hoe snel dient een bepaalde beslissing gemaakt te worden.
- ✚ **Consistentie:** Zijn er vaak nieuwe regulaties of blijven ze relatief onveranderlijk.
- ✚ **Value range:** Het verschil in waarde tussen een goede beslissing en een slechte beslissing.
- ✚ **Degree of freedom:** Hoe gelimiteerd is een bepaalde beslissing.

Afhankelijk van het soort beslissing en de eigenschappen hiervan, wordt er een bepaalde prioriteit toegekend voor het modelleren van de beslissing. De prioriteit wordt bepaald met behulp van een hotspot analyse. Dit houdt in dat er gekeken wordt, welke beslissingen het dichtst aansluiten bij de bedrijfsdoelen en key performance indicators. Daarnaast wordt de voorkeur gegeven aan goed meetbare beslissingen, die een grote business impact hebben. Zo wordt hieronder een voorbeeld gegeven van een mogelijk prioriteitenvoorstel. (Taylor, 2012)



Figuur 3: Hotspot-analyse

Uit [Overgenomen uit] Decision management systems(p.113), door J. Taylor, 2012, Indiana: International Business Machines Corporation.

De grootste cirkels het dichtst bij de rechterbovenkant verkrijgen de voorkeur vanwege drie redenen. Deze beslissingen vereisen het minste technische kennis, hun effect kan het snelst waargenomen worden en hun waarde is het grootst.

Wanneer de afhankelijkheden van verschillende beslissingen gekend zijn, kan de analist het afhankelijkheidsnetwerk in een decision flow transformeren. Een decision flow bestaat uit een serie van beslissingtaken die gelinkt worden met conditionele takken. Verschillende subdecisions worden gelinkt met hogere beslissingen. Beslissingen die het meest doorslaggevend zijn, dienen als eerste gezet te worden. De reden hiervoor is dat wanneer de eerste beslissing negatief zou zijn, de volgende beslissingen niet meer genomen moeten worden. Ook de data die beschikbaar moet zijn tijdens de decision flow dient gespecificeerd te worden. De decision flow vormt de basis voor de individuele decisions die hierna dienen uitgewerkt te worden. Dit houdt in dat er gezocht wordt naar bepaalde sets van business rules die deze beslissing ondersteunen. Een rule set is een set van regels die samen geëvalueerd en uitgevoerd moet worden. (Taylor, 2012) Om een bepaalde ruleset te kunnen uitvoeren kan het zijn dat een andere rule set eerst uitgevoerd moet worden. Dit is zichtbaar in de decision flow.

2.2.3. BUSINESS RULES:

Voor de elementen van een business rule werden er specifieke concepten uitgewerkt om een standaard te hebben om hierover te communiceren (Von Halle, 2001):

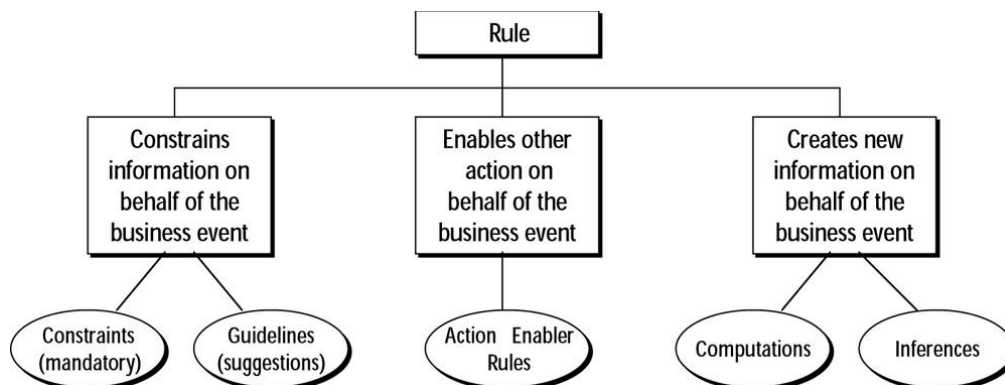
Term: Termen moeten consistent gedefinieerd worden om te zorgen dat gebruikte woorden een standaardbetekenis hebben die voor iedereen duidelijk is. Samen worden ze opgenomen in een glossary. Een term kan een concept, een eigenschap van een concept, een waarde of een set van waardes voorstellen.

Fact: Een fact is een statement dat termen met elkaar linkt via voorzetsels en werkwoorden tot een business relevant feit. Mogelijke facts worden hieronder opgesomd (zie ook figuur 4).

- ✚ **Een verplichte constraint:** Een compleet statement dat aangeeft wat moet of niet mag alvorens een business event voltooid kan worden.
- ✚ **Een richtlijn:** Een compleet statement dat een waarschuwing geeft of iets (niet) mag.
- ✚ **Action enabler:** Een compleet statement dat bepaalde voorwaarden nagaat. Afhankelijk van de waarde van de voorwaarden, wordt er een ander event of activiteit uitgevoerd.
- ✚ **Berekening:** Een compleet statement dat een algoritme voorziet om de waarde van bepaalde termen te berekenen. Dit kan door middel van een som, verschil, product, ...
- ✚ **Conclusie:** Een compleet statement dat de waarden voor bepaalde condities nagaat om de waarde van het conclusiefact te bepalen.

De termen en facts vormen samen de semantiek achter de regels. Ze vormen de basis voor het logische datamodel, de fysieke database en vaak ook het business object model.

Rule: Een rule is een declaratief statement dat logica of berekeningen toepast op informatie en nieuwe informatie of een bepaalde actie als resultaat heeft. Een rule kan gezien worden als uitvoerbare logica die informatie als input gebruikt en informatie of acties als output genereert.



Figuur 4: Verschillende types van rules

Uit [Overgenomen uit] Business Rules applied (p.37), door B. Von Halle, 2001, New York: John Wiley & Sons, Inc.

Wanneer zulke regels gedefinieerd worden, is het belangrijk dat ze voldoen aan het ABCDU principe (Bauer, 2009):

- ✚ **Atomic:** Ze kunnen niet verder opgedeeld worden zonder info te verliezen.
- ✚ **Business related:** Enkel termen en facts van het factmodel mogen gebruikt worden.
- ✚ **Consistent:** Er mogen geen contradicties tussen business rules bestaan.
- ✚ **Declarative:** Een rule is geen procedurele beschrijving, slecht 1 enkele beslissing.
- ✚ **Unambiguous:** Per rule is er slechts één interpretatie mogelijk.

Meer specifiek voor deze masterproef is er interesse voor de business rule. Deze vormt een beknopt statement over een aspect van de business. De business rule definieert wat moet of niet mag gedaan worden in een bepaald geval. Deze rules worden gebaseerd op het beleid, wetgevingen, ... Eénmaal voldoende rules verzameld zijn, is het belangrijk deze om te zetten in executable business rules die ondersteund kunnen worden door een Business Rule Management Systeem. Een executable business rule bestaat uit een set condities en acties die genomen kunnen worden op basis van de condities. (Taylor, 2011)

Belangrijk is hierbij het gebruik van de **business rules approach**:

"A formal way of managing and automating an organization's business rules so that the business behaves and evolves as it leaders intended." (Von Halle, 2001).

Dit houdt in dat er duidelijk gedefinieerde processen, taken en rollen zijn waarbij het management dankzij verzamelde informatie, business rules kan evalueren en controleren. Daarnaast moeten business rules operationeel gemaakt worden zodat een business application er gebruik van kan maken.

Deze benadering is gebaseerd op vier principes, met de name de STEP principes (Von Halle en Goldberg, 2010):

(S) Separate: Scheiden houdt in dat de rules gescheiden worden van de andere aspecten en systemen. Dit maakt het mogelijk om regels opnieuw te gebruiken en ze aan te passen onafhankelijk van andere aspecten zoals procesmodellen.

(T) Trace: Ondanks de scheiding, is een connectie van iedere rule in twee richtingen noodzakelijk. Met name een link met business missies, doelen, objectieven, strategieën en beleiden. De tweede link is de link met de implementatie van de rule, zodat de impact van veranderingen in rules kan nagegaan worden met behulp van de vooropgestelde metrics.

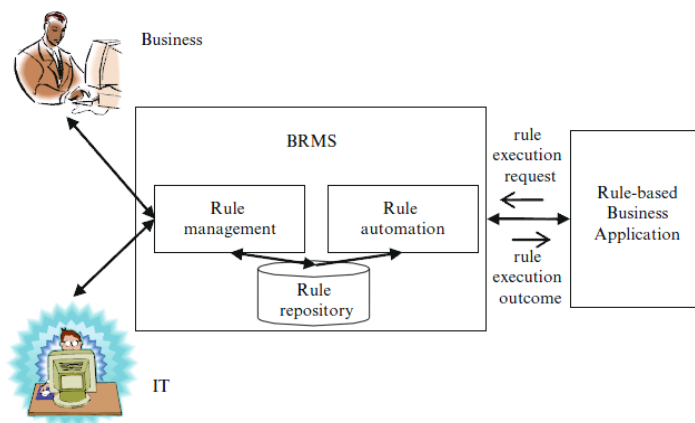
(E) Externalize: Het derde principe wijst erop dat een rule moet worden uitgedrukt in een formaat dat begrijpbaar is voor de business users. Daarnaast dient de rule ook voor hen beschikbaar te zijn zodat ze weten welke rules er zijn, wat ze betekenen en hoe ze gebruikt kunnen worden.

(P) Position: Het laatste principe maakt duidelijk dat een rule altijd klaar moet zijn voor verandering. Belangrijk is dat die veranderingen snel en eenvoudig doorgevoerd kunnen worden.

Het moet mogelijk zijn een vlottende impactanalyse te doen wanneer een rule verandert, zodat geweten is welke events, beslissingen, processen hiervan invloed zullen ondervinden.

2.2.4. HET BUSINESS RULES MANAGEMENT SYSTEEM

Om de business rules te beheren dient er een toolset te bestaan om de rules te managen en uit te voeren, met name een Business Rule Management Systeem. Het BRMS ondersteunt het managen van de regels door een opslag te voorzien voor de rule artifacten. Het BRMS zorgt dat alle verzamelde business rules opgeslagen, gemanaged en verwerkt worden. Het BRMS is één van de belangrijkste pijlers van de business rule approach. (Bauer, 2009)



Figuur 5: DE BRMS-omgeving

Uit [Overgenomen uit] Agile Business Rule Development (p.14), door J. Boyer en H. Mili, 2011, New York: Springer.

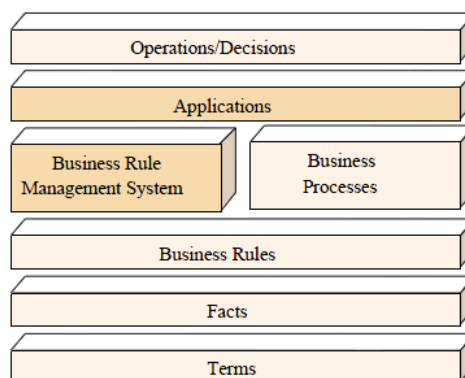
Een Business Rule Management system heeft twee componenten, een management en een execution component die samen een ruleopslag delen. Deze opslag kan gelezen en aangepast worden door de rule management component, maar enkel gelezen door de rule engine die zorgt voor rule execution. De rules die een applicatie nodig heeft, worden dus buiten de applicatie uitgevoerd. Dit vergemakkelijkt het onderhoud, omdat de regels aangepast kunnen worden onafhankelijk van de applicatie, en zo de applicatie minder belast wordt. (Boyer & Mili, 2011)

Business rules worden op verschillende manieren opgeslagen. Zowel in een meer technische als meer business-taal zodat aanpassingen van business rules ook door business users gedaan kunnen worden. Business users worden zo dan zelf verantwoordelijk voor het creëren en managen van de business rules. Een applicatieontwikkelaar is hierdoor dus minder vaak nodig om het gedrag van de applicatie aan te passen.

Belangrijk hierbij is dat er een keuze gemaakt wordt of het rule management synchroon of asynchroon wordt toegepast. (Bauer, 2009) Beiden hebben hun eigen voordelen en nadelen. Bij de synchrone keuze, is er sprake van een groot business rule management geheel dat diverse business applicaties voorziet van regels. De rules management organisatie is dan verantwoordelijk voor het verzamelen, coderen, valideren en publiceren van de business rules. Iedere applicatie heeft dan nood aan een subset van de business rules. In het geval van het asynchrone rule

management, worden de rules voor iedere applicatie apart ontwikkeld. De regels worden wel samen opgeslagen in de rules repository, maar zijn specifiek verbonden met de applicaties.

Het grootste voordeel van de synchrone versie is de organisatiebrede zichtbaarheid van de regels, waarbij er een consistente en coherente ruleopslag is. Wel vraagt het een complex en uitgebreid management om deze repository zo algemeen mogelijk te maken. Bij de asynchrone versie is er dus minder complexiteit en zijn de regels zeer specifiek aangepast aan iedere applicatie. Het nadeel hiervan is de inconsistentie tussen de verschillende rules en de duplicatie door het meermaals gebruiken van dezelfde regels door verschillende applicaties. (Bauer, 2009)



Figuur 6: Positie van het BRMS in het gehele systeem
Uit [Overgenomen uit] the business rules approach (p.3), door E. Bauer, 2009, seminar.

Het BRMS heeft zowel voordelen als nadelen, deze worden hieronder opgelijst:

Voordelen:

Centralisatie leidt tot homogene business rules die slechts eenmaal opgenomen moeten worden en door verschillende applicaties gebruikt kunnen worden. Veranderingen aan een business rule hebben daarom ook real-time effect op de applicatie. Het vergemakkelijkt het change management.

Nadelen:

Door de grote centralisatie zijn alle applicaties voor hun business rules afhankelijk van de functionaliteit van één BRMS. Eén access point betekent één groot mogelijk faalpunt. Hierbij wordt een groot vertrouwen geschonken aan de aanbieder van het BRMS. Deze aanbieder kan updates uitvoeren, crashes verhelpen. Daarom is het belangrijk dat deze aanbieder dit ook doet.

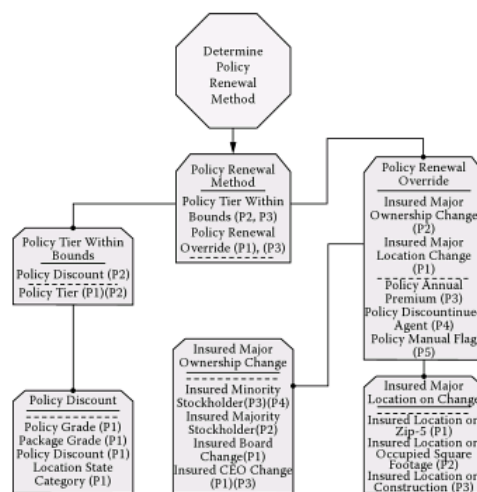
2.2.5. THE DECISION MODEL

Doorheen de tijd werden processen meer en meer geautomatiseerd. De opeenvolging van de stappen in een proces werden samen met de bijbehorende businesslogica omgezet in programmeercode. Het probleem van deze snelle evolutie in dit domein zorgde ervoor dat de business logica vast kwam te zitten in de systemen en dus ook moeilijk aan te passen was. Doorheen de tijd is dit complexe geheel stap voor stap gesplitst in onderdelen zoals een database

component, beveiligingscomponent, transactiecomponent, ... wat zorgde voor meer duidelijkheid. Ondanks die opsplitsing, bleef de business logica wel nog steeds vastzitten aan die componenten. (Von Halle, Goldberg, 2010)

Business logica is een set van business rules die voorwaarden representeren en leiden tot bepaalde conclusies. Het beschrijft de manier waarop belangrijke businessbeslissingen genomen worden. Tegelijkertijd ondersteunt het de integriteit, innovatie en intelligentie van de onderneming. Omdat deze business logica vaak verspreid zit over verschillende documenten, spreadsheets, databases en programmeercode, is het moeilijk deze logica toe te passen waar nodig.

Daarom is er nood aan het Decision model wat veronderstelt dat de business logica op zichzelf bestaat, onafhankelijk van hoe het uitgevoerd wordt. Het model heeft onderstaande specifieke structuur.



Figuur 7: The decision model

Uit [Overgenomen uit] The decision model (p.9), door B. von Halle B. & L. Goldberg, 2010, New York: CRC Press.

Deze vorm van modelleren is een technologieonafhankelijke manier om business logica te managen. Het model is puur gebaseerd op een goed gedefinieerde structuur op basis van logica, in samenhang met integriteit en normalisatieprincipes. Ondanks dat het onafhankelijk is van technologie, is het toch makkelijk te implementeren. Het helpt de business logica en rules efficiënt te managen in een onafhankelijk model. Dit maakt het mogelijk voor business users die niet technisch aangelegd zijn om intuïtief te werken met hun eigen business logica. (Von Halle, Goldberg, 2010)

Samen met opkomende technologiegebieden zoals SOA (Service oriented Architecture, Business Process Management (BPM) en Business Decision management (BDM) maakt deze tool het gemakkelijker om de flexibiliteit, snelheid en IT governance te verbeteren. Om ervoor te zorgen dat al deze domeinen optimaal werken, dient de business logica optimaal en herbruikbaar te zijn.

Het Decision model vormt een template voor het organiseren en managen van de business logica die business decisions ondersteunt. Die decisions zijn een conclusie van de business logica die

betekenis en waarde heeft voor de business. Het is niet een simpele opsomminglijst, maar een model dat de structuur van de logica weerspiegelt. Om zo een model te kunnen bouwen dient er eerst inzicht verkregen te worden in wat precies de business logica inhoudt. Dit kan neergeschreven worden in tekstvorm. Deze tekst dient daarna omgezet te worden in business facts die condities vormen voor een bepaalde conclusie. Dit gebeurt door tweedimensionale structuren te creëren, met name de rule families. (Von Halle, Goldberg, 2010)

Een rule family wordt als volgt opgebouwd (zie figuur 8): de rule patterns, de condities (bruine kolom) en de conclusie (gele kolom). De condities zijn de facts die getest worden en die samen leiden tot een bepaalde conclusie. Iedere fact-kolom wordt nogmaals opgesplitst in twee andere kolommen, één daarvan is de operator (IS, =, <, >) en de andere kolom de operand (Yes, No).

Eligibility									
Regel	Regel-patroon	Blacklisted		Another reservation at date		Drivers age		Eligibility	
1	4	is	no	is	no	is	Accepted	is	yes
2	1					is	Rejected	is	no
3	2	is	Yes					is	no
4	3			is	Yes			is	no

Figuur 8: Een rule family

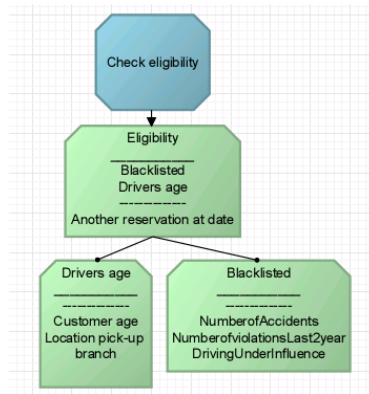
Belangrijk is dat voor ieder fact geweten is wat de mogelijke waarden kunnen zijn zodat iedereen akkoord kan gaan met de conclusie wanneer aan bepaalde voorwaarden voldaan is.

Een regel in een rule family kan als volgt gelezen worden:

Als conditie Blacklisted = No EN 'Another reservation at date' = No EN 'Drivers age' = 'accepted', dan conclusie Eligibility = Yes

Soms is een bepaalde conditie afhankelijk van andere feiten. Deze conditie wordt dan opgenomen als conclusie van een andere rule family. (zie figuur 9) De waarde van de conclusie vormt de basis voor de conditie in andere rule family. Voor een beknopt overzicht van de hele structuur, werd het Decision model bedacht.

Een decision model diagram (zie figuur 9) bevat een aantal rule family tables en één eindbeslissing die wordt voorgesteld door een octagonale vorm. Deze eindbeslissing wordt dan gerelateerd aan een taak binnen het procesmodel. De rule family die verbonden is met de octagonale vorm, is de decision rule family. De naam van iedere rule family wordt bepaald door de conclusie van deze rule family. In het symbool van een rule family worden twee lijnen opgenomen, een volle lijn en een stippellijn. Het fact boven de volle lijn is de conclusie, de facts tussen de volle lijn en de stippellijn zijn een conditie voor de rule family, maar vormen een conclusie voor een andere gerelateerde rule family. De facts onder de stippellijn zijn condities die niet berusten op een andere rule family. De waarde van deze facts is dan ook onmiddellijk beschikbaar.



Figuur 9: Een voorbeeld van een decision model diagram

Het Decision model is gebouwd rond 3 basisbegrippen die bestaan uit 15 verschillende principes en 3 normaalvormen. Deze drie begrippen zijn: structurele simpliciteit, declaratieve opbouw en optimale integriteit. Ze zorgen er samen voor dat het decision model duidelijk, begrijpbaar en consistent is. De verschillende principes en normaalvormen kunnen in bijlage B geraadpleegd worden.

De waarde van een bepaald decision model, dat vorm, functie en een visuele representatie geeft aan de businesslogica, wordt bepaald door 3 karakteristieken (Von Halle, Goldberg, 2010):

- ✚ **De operationele context:** Dit bepaald hoe het Decision Model een probleem of een opportuniteit oplost.

Simple operationele context: De fact types en hun waarden zijn bekend, het is dus zeer makkelijk, de facts, de condities en conclusies af te leiden.

Het decision model kan ook helpen om een **complexe context** om te vormen tot een simpele context door het toevoegen van extra condities (en rule families). Mogelijke waarden van een bepaalde conditie kunnen stapsgewijs aan het licht gebracht worden. Dit op basis van ervaring, advies, ... van experts.

- ✚ **De volumegebaseerde economische impact:** Dit geeft inzicht in de financiële waarde van het business model.

De economische impact van een businessbeslissing is afhankelijk van hoe vaak deze beslissing genomen dient te worden en hangt ook af van zijn individuele economische waarde. Het business model probeert ondersteuning te bieden bij operationele businessbeslissingen. Dit omdat operationele beslissingen vaak genomen worden en ze voor de klant het meest zichtbaar zijn. Veel van deze beslissingen worden best geautomatiseerd nadat de business logica duidelijk is. Daarnaast is ook het meten en controleren van deze beslissingen een competitief voordeel.

- ✚ **De Business logica complexiteit:** Dit helpt de kost van de ontwikkeling en het managen van het Decision Model mee te bepalen.

De complexiteit is afhankelijk van het aantal rule families, fact types en logic statements. Bij lage complexiteit, zorgt het decision model dat logica gedocumenteerd, gedeeld, gestandaardiseerd en veranderd kan worden. Als het complexer wordt, kan het decision

model gebruikt worden om de complexiteit te verlagen en bepaalde delen simpeler te maken voor automatisatie.

Ondanks dat businesslogica onafhankelijk moet zijn, is het soms noodzakelijk dat bepaalde businesslogica al uitgevoerd wordt alvorens bepaalde daaropvolgende beslissingen genomen kunnen worden. Om dit op te lossen dient er een opeenvolging van meerdere decision models gecreëerd te worden waarbij elke decision model gelinkt wordt aan een bepaalde process taak. Deze decisions samen vormen dan de eerder besproken decision flow. Indien een proces gelinkt wordt aan meerdere decision models, dient de analist na te gaan of deze niet samengenomen kunnen worden, indien wel, dan wordt het geheel kleiner en minder complex. Toch is het soms noodzakelijk deze scheiding te behouden wanneer bepaalde beslissingen door andere departementen genomen dienen te worden en/of wanneer de sequentie echt belangrijk is. (Von Halle & Goldberg, 2010)

OVERZICHT VAN HET OPSTELLEN VAN HET DECISION MODEL

Voordat een finaal Decision model optimaal is, dienen er verschillende stappen gevolgd te worden. Zo beginnen de businessarchitecten meestal met het opstellen van de processen. Voor een bepaald proces kan de architect bepaalde statements opstellen in de natuurlijke taal. Deze dienen gedetailleerd omschreven te zijn om misverstanden te voorkomen. Omdat het vaak niet specifiek genoeg is voor een bepaalde context, wordt er ook gebruik gemaakt van business use cases.

Een business use case is een stap voor stap verhaal dat geformuleerd wordt door een individuele actor die interacties doet met het systeem. Een complete use case omvat het pad dat gevolgd wordt wanneer alles goed gaat en alternatieve paden die gevolgd kunnen worden wanneer bepaalde zaken anders gaan. Samen vormen meerdere use cases het business use case diagram. De link tussen de Business use cases en het decision model wordt gecreëerd wanneer bepaalde stappen in de case als decision steps worden geïdentificeerd. (Von Halle, Goldberg, 2010)

Wanneer de links gelegd zijn, kan de architect nagaan of bepaalde beslissingen een volgorde hebben of dat deze ook parallel genomen kunnen worden en dus als één beslissing kunnen worden opgenomen. Verschillende modellen kunnen gecreëerd worden van deze use cases. Belangrijk hierbij is de link tussen de verschillende modellen. Dit omdat een verandering van het decision model ook invloed kan hebben op de rest van de modellen. Zo is het business process model en het Decision model gelinkt via de decision points in het procesmodel. De termen en facts die gebruikt worden in het decision model worden best steeds opgenomen in een Glossary om ervoor te zorgen dat het Decision model juist geïnterpreteerd kan worden. (Von Halle, Goldberg, 2010)

Finaal kunnen het decision model en de bijbehorende rule families getest worden. Afhankelijk van het aantal regels en het aantal kolommen, wordt er gekozen voor het testen van alle mogelijke waarden of enkel de grenswaarden of samples. Dit is belangrijk om de correctheid en volledigheid van het decision model te testen. (Von Halle, Goldberg, 2010)

HOOFDSTUK 3: THE DECISION MODELER & OPERATIONAL DECISION MANAGER

3.1. THE DECISION MODELER BY BIZZDESIGN

Daar veranderingen in reguleringen, beleid en markten een steeds frequenter fenomeen worden, is een flexibel decision/rule management van uitermate belang. Zowel grote bedrijven als kleine en middelgrote ondernemingen (KMO's) krijgen hiermee te maken. Grote ondernemingen hebben meestal een eigen IT departement. Dit maakt het makkelijker voor hen om bijvoorbeeld Business Rules Management Systemen te implementeren. Voor KMO's daarentegen is het veel moeilijker om op die veranderingen te reageren. Dit door het gebrek aan kostefficiënte, volledige tools. De kapitaalcrachtige bedrijven maakten meestal gebruik van Sapiens Decision, maar voor de KMO's was er niet dadelijk een complete tool te vinden. Daarom gebruikten de kleinere bedrijven vaak een combinatie van Visio, Excel en Word templates voor het modelleren van beslissingen samen met OpenRules BDMS voor het testen en uitvoeren van de rules. Dit gaf kleinere bedrijven toch de mogelijkheid om met beslissingsmodellen te werken, maar bij deze aanpak bleek er een gebrek aan cohesie en traceerbaarheid. Het kostte namelijk veel manueel werk en bood een grote kans op fouten. (<http://blog.luxmagi.com/2013/06/new-tool-to-change-the-face-of-business-decision-modellingmanagement/>)

Om dit probleem op te lossen ontwikkelde Bizzdesign in 2013 een kostefficiënte module 'The decision modeler' voor hun bestaande Architect tool. Er bestond tot dan geen kostefficiënte tool die het eerder besproken The Decision Model ondersteunde. De module is namelijk volledig gebaseerd op het decision model. De voordelen van het decision model werden eerder in deze masterproef duidelijk gemaakt.



Figuur 10: De evolutie binnen business rules
Uit [Overgenomen uit] The Decision Modeler powered by The Decision Model,
door KPI en Bizzdesign, webseminarie.

The Decision Modeler kan gezien worden als een visuele editor van decision models en rule families, waarbij binnen de Architect tool een link gemaakt kan worden met UML Class Diagrammen, BPMN procesdiagrammen en ArchiMate modellen. Het laat de gebruiker toe modellen te creëren, rekening houdend met de principes van het gelijknamige boek. Wanneer het model gecreëerd is, kan dit getest worden met behulp van de ingebouwde checks. Daarnaast kunnen scenario's getest worden met behulp van Excel en kunnen business rules eventueel geëxporteerd worden naar OpenRules®.

De geplande functionaliteiten die in de roadmap van Architect terug te vinden zijn: extra validatie mogelijkheden, waarbij een meer geavanceerd model gebouwd kan worden. Een andere nieuwe functionaliteit wordt het exporteren naar andere rulestandaarden en tools zoals Drools, IBM ILOG/ODM, Oracle Policy Automation, ... Daarnaast ontbreekt tot nu toe een simultane analyse van de processen en beslissingen zoals het in de praktijk ook gebeurt. Processen en decision models kunnen wel gelinkt worden, maar dit heeft nog geen effect. Dit zou in de nabije toekomst wel mogelijk moeten zijn. (Bizzdesign, 2013)

Het bedrijf Bizzdesign dat de tool aanbiedt, werd zelf in het Gartner Magic Quadrant voor Enterprise Architecture tools als visionair gepositioneerd in oktober 2013. Dit Quadrant houdt rekening met de internationale marktpositie, vakkennis en kracht van de enterprise architecture propositie. Volgens Gartner zijn de beste EA tools, tools die informatie opvangen, opslaan, structureren, analyseren en presenteren. Ze maken het mogelijk de business strategy te implementeren rekening houdend met de resultaten en risico's. Die resultaten moeten voor de verschillende stakeholders begrijpbaar zijn. (<http://www.persberichten.com/persbericht/75319/BiZZdesign-gepositioneerd-als-Visionair-in-Magic-Quadrant-voor-Enterprise-Architecture-Tools-2013>)

Via de Architect toolsuite kunnen verschillende stakeholders uit de business en IT wereld met elkaar communiceren. Om dit te kunnen verwezenlijken worden het Business Model Canvas®, ArchiMate®, TOGAF®, BPMN™ (process modeling), TDM (business logic modeling), OIAm (infrastructure methode), UML, ERD en Lean ondersteund. Het bedrijf Bizzdesign heeft als voornaamste sterke punten de sterke internationale groei, de ondersteuning voor de meest bekende open standaarden en hun geregelde innovaties die ze uitwerken samen met universiteiten, researchers en partners.

Zwakheden die werden aangekaart waren ontbrekende functionaliteiten op het gebied van portfolio analyse en portfoliomanagement. Daarnaast ook een beperking wat betreft data-invoercontrole, restricties en geavanceerde rapportering.



Figuur 11: BizZdesign in het Gartner Enterprise architecture tools quadrant
 Uit [Overgenomen uit] Magic quadrant for enterprise architecture tools,
 door Gartner, webartikel.

4 belangrijke features van The Decision Modeler module zijn (Jonkers et al, 2013):

- ✚ De mogelijkheid om relaties tussen business decisions en artifacts of andere architectuurmodellen te creëren. TDM biedt namelijk de mogelijkheid ook proces-, informatie-, actoren-, business- en motivatiemodellen te ontwerpen. Deze kunnen dan ook geassocieerd worden met de beslissingen, wat het een coherent geheel maakt. Zo kan de gebruiker dus linken maken met de taken waar de beslissingen gebeuren en door wie ze genomen worden. Het biedt een geïntegreerd geheel van business logic, processen en informatie. Zoals eerder aangehaald dienen deze links nog uitgewerkt te worden om er het praktisch nut uit te halen.
- ✚ Het managen van business vocabulary: TDM biedt een glossary management systeem aan dat toelaat aan businessgebruikers om nieuwe businesstermen en fact types te definiëren. Dit kunnen zowel liniaire datatypes (numerical, boolean, date), list types als opsommingen zijn die geformuleerd kunnen worden zonder de hulp van IT. Het systeem maakt een makkelijke navigatie tussen de glossary en andere TDM artifacten mogelijk.
- ✚ Model testing: naast het modelleren van decision models, kunnen ze ook getest worden dankzij de interactieve interface die automatisch meldingen geeft over ontbrekende facts. Daarnaast kan een Excel spreadsheet met test cases ook gebruikt worden als input om het model en de rules te testen.
- ✚ Output naar BDMS/BRMS: Op dit moment bestaat de mogelijkheid artefacten direct te exporteren naar Open Rules. Daarnaast zit de support voor de commerciële en open-source BRMS in de lift. Zo gaf BizZdesign te kennen dat een export naar Drools en ILOG (ODM) voor de nabije toekomst is.

De grootste voordelen zijn dus het relatief goedkope prijskaartje en de integratie van de complete proces/informatie/logica/executie van de beslissingen. Het maakt het namelijk makkelijker decisions te linken met BPMN decision activiteiten en de glossary te linken met de UML klassen en attributen.

3.2. OPERATIONAL DECISION MANAGER BY IBM

Zoals eerder werd aangehaald is een toekomstige functionaliteit van The Decision Modeler het exporteren van de decision modellen naar Operational Decision Manager (ODM). ODM is een Business Rules Management Systeem (BRMS), beter bekend onder zijn vorige naam: IBM websphere ILOG Jrules, JRules in het kort. Het bevat een reeks functionaliteiten die het mogelijk maken voor zowel business users als ontwikkelaars business rules te creëren, te testen, te implementeren en te managen. Dit gebeurt onafhankelijk van de applicaties. Het laat toe business users en IT experts samen te werken rond business rules. Omdat niet iedereen dezelfde capaciteiten heeft, biedt ODM verschillende tools aan voor de verschillende gebruikers.

IBM vormde JRules BRMS in oktober 2011 om tot WebSphere Operational Decision Management. Deze rebranding was gebaseerd op de integratie van Jrules met IBM's WebSphere Business Events product. Op die manier werd er een compleet business rules en events management platform gecreëerd.

De bespreking in de volgende bladzijden met betrekking tot de werking en functies van ODM werden verkregen via de documentatie verschaft in het IBM Operational Decision Manager Version 8.0 information center en door het zelf toepassen van de tools dankzij de ter beschikking stelling van de ODM omgeving via het bedrijf AGServ.

IBM ODM bestaat uit de volgende twee grote onderdelen:

Decision center:

Het Decision center vormt een opslag en management component voor het creëren en onderhouden van de business logica. Het kan gezien worden als een centrale hub waarin de levenscyclus van beslissingen wordt gecoördineerd. Het houdt zowel rekening met business rules als business events. Het Decision center omvat de Enterprise en business Console.

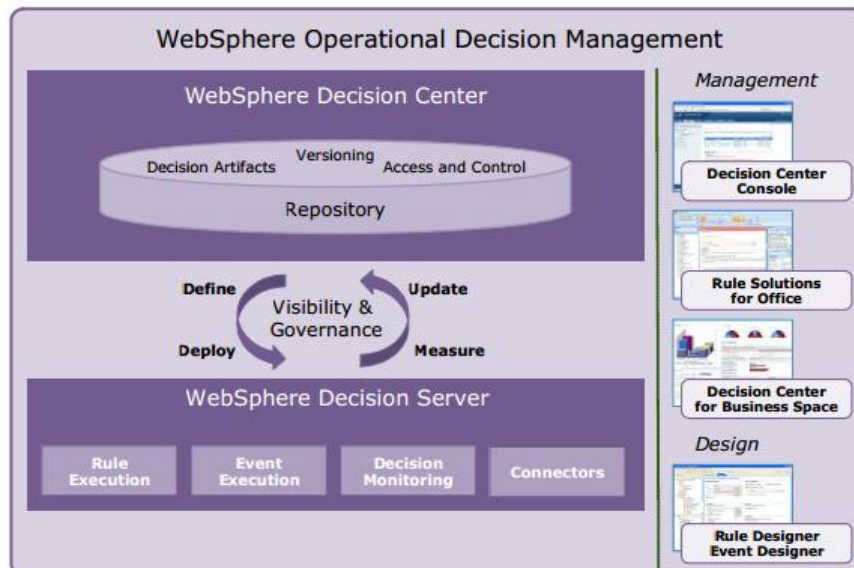
Decision server:

De decision server omvat de runtime onderdelen die nodig zijn voor business rules en events. Het maakt het mogelijk om beslissingen en events uit te voeren, te monitoren en op te volgen. De twee grote deelcomponenten zijn de decision server for rules en decision server for events.

De **Decision server for rules** omvat een groep tools die gebruikt wordt voor het ontwerpen, opstellen, testen en uitvoeren van de rules en de geautomatiseerde beslissingen. Voor de verschillende gebruikers zijn er verschillende tools. De tool waar in

deze masterproef voornamelijk de focus op ligt, is de Rule Designer, een Eclipse gebaseerde applicatie voor het ontwikkelen van de rule omgeving.

De **Decision server for events** omvat een geïntegreerde set componenten voor het uitvoeren van events. Afhankelijk van de gebruiker, wordt deze server gebruikt voor het ontwerpen, ontwikkelen, testen, uitvoeren en onderhouden van de business events. Een voorbeeld hiervan is de events designer, ook een eclips gebaseerde applicatie.

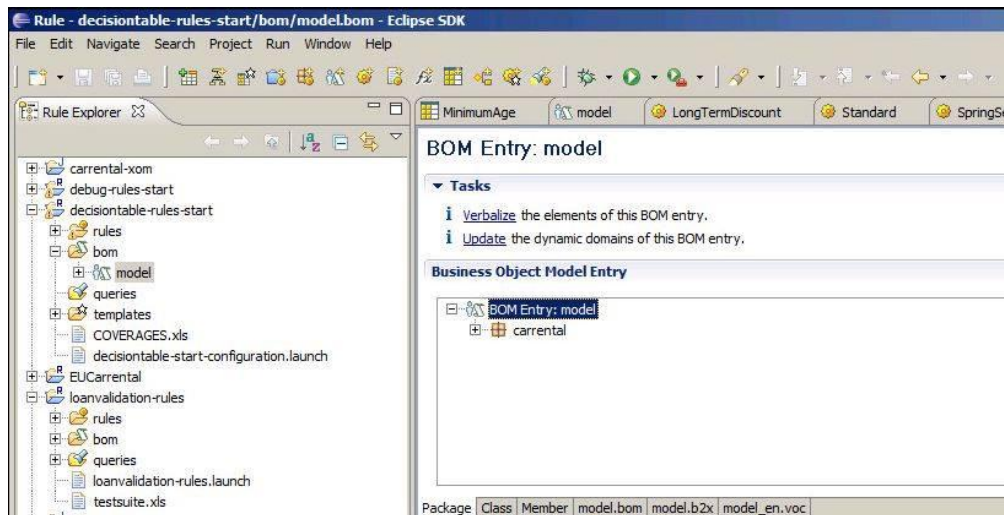


Figuur 12: De webSphere operational decision management omgeving
Uit [Overgenomen uit] Making Better Decisions Using IBM WebSphere Operational Decision Management (p.3), door D. Clark, P. Berlandier, 2012,IBM

Zoals eerder vermeld werd, ligt de focus in deze masterproef op de rule designer. Deze wordt gebruikt om regels te creëren. Dit kunnen zowel objectmodellen, ruleprojecten, ruleflows en andere business rule artifacten zijn. Het object model bestaat uit twee lagen, de logische laag (Business Object Model, BOM) die de woordenschat omvat voor de rules en een fysieke laag (eXecutable Object Model, XOM) die zorgt voor de implementatie van de objecten in Java of XML binnen Eclipse. Rule designer wordt gebruikt door rule ontwikkelaars, rule architecten, softwareontwikkelaars en businessanalysten. Wanneer nieuwe inhoud dient toegevoegd te worden door niet technische mensen, dienen de ontwikkelaars rule projecten in het Decision center beschikbaar te maken. Zo kunnen ontwikkelaars en business users samenwerken op dezelfde rules via een verschillende omgeving. Belangrijk is dat aanpassingen zo snel mogelijk worden doorgegeven aan het Decision center. Binnen dit center werken alle business users in een gedeelde werkruimte en gebruiken ze locks om te voorkomen dat andere gebruikers toegang hebben tot de resources die op dat moment worden aangepast.

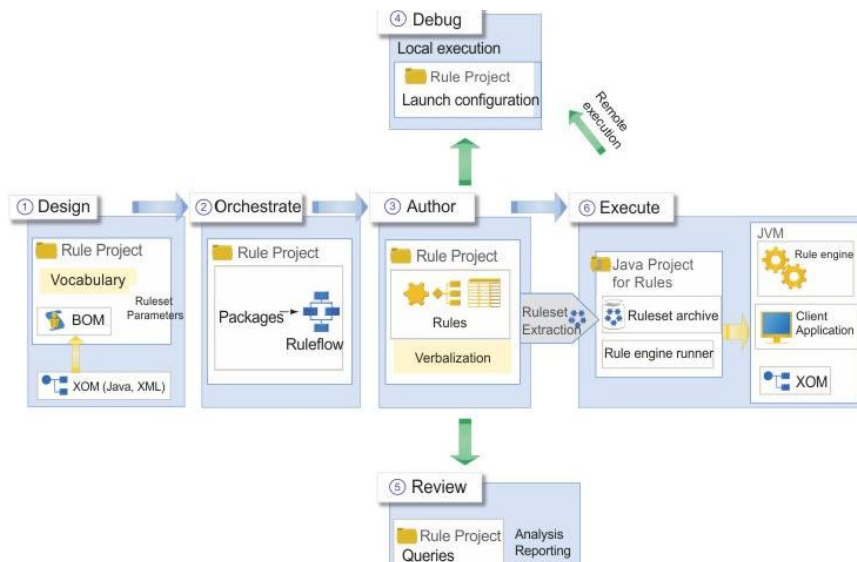
3.2.1. DE RULE DESIGNER

Rule architecten gebruiken deze tool om een rule project te creëren, het is een eclipse project voor de creatie van business rules. Een rule project bestaat uit verschillende rule artifacts: het Extensible Object Model (XOM), Business Object Model (BOM, een businessview van de XOM), de hiërarchie van rule packages, decision tables, decision trees en de rule flow.



Figuur 13: De rule designer interface

Onderaan in het scherm is er een rule project map view die helpt de user een rule project te voltooien door hem door de verschillende stappen te loodsen.



Figuur 14: De rule project map binnen Rule Designer

Uit [Overgenomen uit] WebSphere Operational Decision Management V8.0, door IBM

De verschillende stappen die doorlopen dienen te worden bij het creëren van een Rule Project wordt hieronder kort opgelijst en verderop uitgebreid besproken:

- ✚ **Stap 1 Design:** Een rule project wordt aangemaakt. Dit zal dienen als opslag voor de rule artifacten die nodig zijn om de business logic uit te drukken. In deze stap creëert en importeert de rule architect of ontwikkelaar het Execution Object Model (XOM) en wordt het Business Object Model (BOM) daarvan afgeleid. Daarnaast wordt er ook een vocabulary aangemaakt die de gebruikte termen duidelijk maken voor de gewone business users. Als laatste in deze stap worden de ruleset parameters gedefinieerd. Deze parameters maken het mogelijk data met applicaties uit te wisselen.
- ✚ **Stap 2 Orchestrate:** Eenmaal de eerste stap voltooid is, creëert de ontwikkelaar de rule packages voor de verschillende beslissingen die genomen dienen te worden. Rule packages zijn groepen van rules die tot een bepaalde eindbeslissing leiden. Met deze rule packages wordt er een RuleFlow gevormd die de volgorde vastlegt.
- ✚ **Stap 3 Author:** Binnen de verschillende rule packages worden specifieke business rules uitgewerkt. Dit door gebruik te maken van de Business Action Language (BAL). Dit kan onder verschillende vormen zoals rules, decision tables of trees.
- ✚ **Stap 4 Debug:** Vooraleer de rules uitgevoerd kunnen worden, dient het rule project door de ontwikkelaar gedebugd te worden.
- ✚ **Stap 5 Review:** Wanneer het debuggen voltooid is, kunnen de rules nog een laatste keer nagekeken worden op inconsistenties en mogelijke conflicten.
- ✚ **Stap 6 Execute:** Wanneer het hele rule project compleet en correct is, kunnen de rules getest en uitgevoerd worden.

STAP 1: HET ONTWERPEN VAN DE RULE PROJECT STRUCTUUR (DESIGN)

Een rule project is een container van rule artifacts. Binnen een rule project zijn rules georganiseerd in packages, meestal stelt een bepaald package een bepaald beslissingspunt voor. Belangrijk is dat een bepaald beslissingspunt gelinkt is met een bepaalde ruleset en die ruleset met een bepaald ruleproject. Rule projecten bevatten de business logica. De Java projecten die de classes bevatten worden afzonderlijk opgeslagen.

Verschillende Rule projecten kunnen naar elkaar verwijzen, dit maakt het mogelijk een gemeenschappelijk model voor verschillende projecten te creëren. Een project dat naar een ander refereert, kan alle elementen van dat project zelf gebruiken.

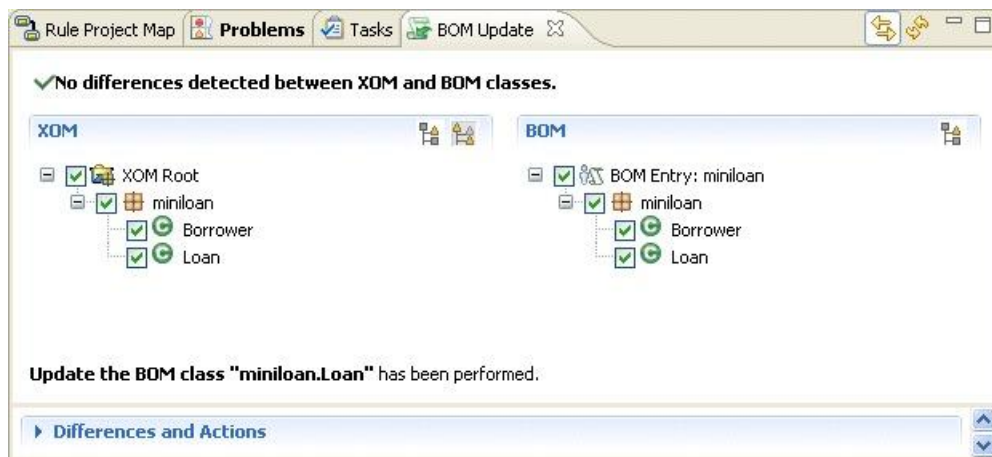
Het ontwerpen van het Extensible object Model

De XOM bevat de verschillende Java classes die nodig zijn om de rules op te stellen.

Het ontwerpen van het Business Object Model

De BOM bestaat uit drie lagen:

- ✚ Het **BOM data model** waarin de definities van de klassen, hun publieke attributen en functies opgenomen zijn in een Java-achtige syntax.
- ✚ De **vocabulary**: Om de BOM begrijpbaar te maken voor business users probeert de ontwikkelaar natuurlijke taal te linken met de verschillende klassen en methodes in de BOM. Al deze verwoordingen samen vormen de vocabulary. Het gebruiken van een vocabulary laat business gebruikers toe zelf business rules te schrijven.
- ✚ De **BOM to XOM mapping** toont hoe de elementen van het BOM datamodel gelinkt worden met de onderliggende XOM/Java klassen. Wanneer alle klassen overeenkomen worden er geen verschillen gemeld.



Figuur 15: BOM to XOM mapping check

Uit [Overgenomen uit] WebSphere Operational Decision Management V8.0, door IBM

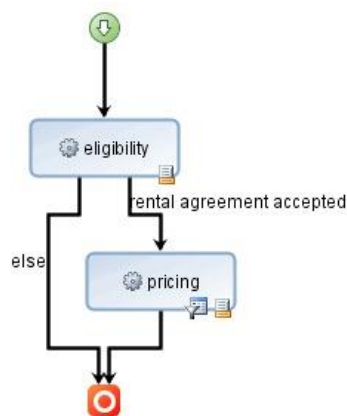
Belangrijk hierbij is dat het model, 'sound' (geen errors), 'complete' (elke belangrijk aspect is gemodelleerd), 'specific' (terminologie model moet aansluiten bij de business terminologie), 'abstract' (abstractieniveau dient juist te zijn voor het beheren van de business rules) en 'relevant' (model dient enkel relevante informatie te bevatten) is.

De approach die toegepast wordt in deze thesis is de bottom up approach, waarbij gestart wordt met een XOM java project. Dit wordt dan uitgebreid door het toevoegen van een BOM. Bij deze mapping wordt voor iedere XOM klasse een BOM klasse aangemaakt. Het BOM design is een kritieke taak van de business rule ontwikkeling, daar het belangrijk is, een intuïtieve gemakkelijke business rule vocabulary te bouwen zodat 'rule authoring', 'reviewing' en 'maintenance' eenvoudiger is. Om ervoor te zorgen dat ook business users rules zullen kunnen maken, kan de ontwikkelaar tijdens het BOM design, rule templates creëren. Deze 'rules' zijn al in de correcte vorm en dienen enkel nog aangevuld te worden met condities of conclusies. Dit maakt het makkelijker voor de rule author om rules aan te maken. Daarnaast biedt het ook zekerheid dat business rule authors de business logica respecteren.

Voor zowel de XOM als de BOM, dient er een goede kennis te zijn van de Java taal en de ILOG rules technical language. Een rule architect dient namelijk de BOM, XOM, templates voor rule authors en het executiegedrag uit te werken. Dit geheel wordt de rule entry infrastructuur genoemd. Deze infrastructuur wordt geüpload naar het decision center.

STAP 2 : CREATIE RULE PACKAGES EN RULEFLOW (ORCHESTRATE)

De verschillende rule artifacten worden meestal gegroepeerd in rule packages. Deze verschillende rule packages worden dan als rule tasks opgenomen in een rule flow die de volgorde van uitvoering van de regels bepaalt. Deze rule flow wordt meestal gecreëerd door ontwikkelaars en architecten. Binnen een ruleflow wordt er een routing logic opgemaakt die de te doorlopen rule tasks met elkaar verbindt binnen de rule set. De routing logic wordt dan ook de transition genoemd. Sommige transitions kunnen ook condities vooropstellen waarbij het systeem afhankelijk van de waarde van een bepaalde klasse het ene of het andere pad kiest. Als voorbeeld hieronder een ruleflow voor de EU-car rental company. Wanneer een bepaalde klant volgens de rules eligible verklaard wordt, dan wordt de pricing rule task doorlopen, anders eindigt de rule flow meteen.



Figuur 16: Ruleflow EU-car rental

STAP 3: HET CREËREN VAN DE RULES (AUTHOR)

Voor het creëren van rules, kan de ontwikkelaar binnen Rule designer gebruik maken van action rules, decision tables en decision trees. Deze worden door de architect of ontwikkelaar geschreven in de Business Action Language en worden daarna door het systeem automatisch vertaald naar ILOG® Rule Language (IRL), een uitvoerbare vorm van een rule artifact. Zo wordt bijvoorbeeld ook voor iedere rij van een beslissingstabel een rule in IRL gegenereerd.

Een BAL wordt automatisch vertaald in een IRL, zo krijg je bijvoorbeeld volgend resultaat:

If

The age of the customer of 'the current rental agreement' is less than 25

Then

Reject 'the current rental agreement'

De uitvoerbare vorm in IRL wordt dan automatisch gecreëerd in de rule designer en kan teruggevonden worden onder de IRL tab in de rule editor.

```
package eligibility {  
    rule minimum_age {  
        property status = "new";  
        when {  
            customer.age() from customer;  
            evaluate (customer.age < 25);  
        } then {  
            rentalagreement.reject();  
        }  
    }  
}
```

Indien de complexe logica die een ontwikkelaar wil opnemen in het project niet mag aangepast worden door businessgebruikers, kan de ontwikkelaar meer technische rules gebruiken. Deze worden geschreven in ILOG® Rule Language (IRL) en staan in directe verbinding met de BOM elementen. Het gebruik van technische rules komt voornamelijk voor bij het programmeren van loops en uitzonderingen.

Wanneer de business users ook zelf graag rules zouden aanpassen, kan de architect templates creëren waarbij makkelijker rules gecreëerd en aangepast kunnen worden dankzij de voorgeprogrammeerde elementen. Een template is een gedeeltelijk ingevulde action rule of decision table waarbij bepaalde elementen nog aangevuld dienen te worden. Dit maakt het makkelijk een series van gelijkaardige rules te creëren. Daarnaast zorgt het ook voor een veiligheid omdat de business users dan enkel veranderingen kunnen aanbrengen aan de elementen die mogen veranderen.

Voor business users is het van groot belang dat de taal waarin ze de rules kunnen schrijven voor hen natuurlijk en begrijpbaar overkomt. Dit kan onder de vorm van de eerder aangehaalde Business Action Language dat een simpele (if-then) syntax en constructs voor business rules biedt. Zo kan een gebruiker rules creëren als volgt:

Er bestaan drie types business rules waarbij BAL wordt toegepast:



Action rules: Action rules zijn zinnen die statements uitdrukken door een set van condities te laten volgen door bepaalde acties indien aan de condities wordt voldaan.

- Definitions: Het aanmaken van lokale variabelen.
- Conditions (if): voorwaarden & Actions (then, else):acties
- 'IF' een bepaalde conditie is TRUE, 'THEN' een set van acties kan uitgevoerd worden.



Decision tables

Decision tables stellen de beslissingslogica voor als een tabel waarbij iedere rij een action rule bevat. Rules met dezelfde condities en acties worden in een tabel weergegeven, waarbij de kolommen de voorwaarden en acties zijn. De rijen vormen de individuele regels.

Deze zijn vaak zeer duidelijk voor business users omdat ze reeds Excel kennen en dit hier mee linken. (Dit is ook vergelijkbaar met de rule families in The Decision Modeler')



Decision trees

Decision trees maken het makkelijk beslissingslogica grafisch voor te stellen. De taken van een decision tree vertegenwoordigen de condities terwijl de bladeren de acties voorstellen.

Zoals eerder aangehaald is het van uitermate belang dat de 'rulemanagement' en 'authoring' omgeving die door de ontwikkelaar en architect opgezet wordt, voldoet aan de noden van de business users. Zo is het beter de rule packages te laten overeenstemmen met de beslissingspunten in een proces. Daarnaast dient de ontwikkelaar ervoor te zorgen dat de vocabulary duidelijk is. Wanneer de automatische verbalisatie van de klassen en elementen niet natuurlijk klinkt, worden deze ook best aangepast zodat het voor businessgebruikers duidelijk is.

STAP 4: HET DEBUGGEN VAN HET PROJECT (DEBUG)

Het debuggen van een ruleproject dient om fouten op te sporen en te verbeteren die er voor zouden zorgen dat rules niet uitgevoerd worden zoals de gebruikers zouden verwachten. Daarom is het van uitermate belang dat de ontwikkelaar na het schrijven van de rules in de Rule Designer en voor het uitvoeren van de rules in de applicatie, de rules debugd.

Hiervoor dient er een launch configuration gecreëerd te worden die vastlegt welk type sessie de ontwikkelaar wil starten. Bij het starten van een project in debug mode maakt de tool een connectie tussen de debugger client en het rule project dat gedebugd dient te worden.

Er bestaan verschillende types van debugging binnen de Rule Designer:

-  Het testen van rules in een rule project
-  Het testen van rules in een rule project en het java project dat de rule engine aanroept
-  Het testen van een Java applicatie die de rules van een rule project uitvoert.

Wanneer het rule project in debug mode uitgevoerd wordt, kunnen klassen, objecten, rules, decision tables, trees en ruleflows apart getest worden met behulp van breakpoints. Deze breakpoints vormen punten waarbij de tester kan nagaan of alles tot aan het breakpoint nog foutloos was. Indien er dan fouten gevonden worden, is het veel makkelijker terug te vinden, waar precies de fout zit.

De debugger monitort de uitvoering van de business rules als volgt: het controleert de ruleset, stuurt de ruleset naar de engine en voert de regels uit met behulp van de engine. Het debuggen zelf kan gebeuren voor of na het uitvoeren van de rules.

STAP 5: HET CONTROLEREN VAN HET RULE PROJECT (REVIEW)

Bij de laatste controle van het project wordt er gezocht naar mogelijke ambigüiteiten en ontbrekende regels. Dit biedt zekerheid dat de regels consistent zijn en dat het rule project compleet is.

Rule Designer omvat drie tools die de gebruiker helpen het ruleproject en de daarin bevatte rules na te kijken:

- ✚ **Queries:** Met behulp van queries kan de ontwikkelaar een ruleproject doorzoeken naar de aanwezigheid van bepaalde rules.
- ✚ **Analyse:** Het uitvoeren van een analyse helpt ambiguïteiten en ontbrekende rules te identificeren.
- ✚ **Report:** Het generen van een rapport over de verschillende rule artifacten in een rule project.

Queries:

Queries verschaffen de mogelijkheid doorheen het ruleproject op zoek te gaan naar rule artifacts die voldoen aan de zoekcriteria die de gebruiker vastlegt. Zo kan een verandering die van toepassing is op een bepaald type rule makkelijker doorgevoerd worden wanneer er een query wordt gedaan naar dat type rules. Daarnaast kan een query gebruikt worden om een bepaalde set van rules te selecteren waarbij de reviewer enkel de geselecteerde rules wil uitvoeren. Queries worden op dezelfde manier geschreven als rules. Zo bestaat een query uit een conditiegedeelte dat de criteria beschrijft waaraan de matches moeten voldoen. De actie omvat hetgeen gedaan moet worden als aan de criteria voldaan werd, meestal het weergeven van de resultaten van de query.

Rule analyse:

Een analyse wordt door een gebruiker uitgevoerd om na te gaan of alles in het project opgenomen werd, wat er gepland was. Dit kan zowel tijdens de ontwikkeling of na de ontwikkeling van de rules. De analyse maakt het mogelijk te zoeken of bepaalde rules op verschillende manieren begrepen kunnen worden of dat er conflicten in of tussen regels bestaan. Voorbeeld van een conflict: twee regels hebben invloed op hetzelfde object, maar geven het object twee verschillende waarden.

Reporting:

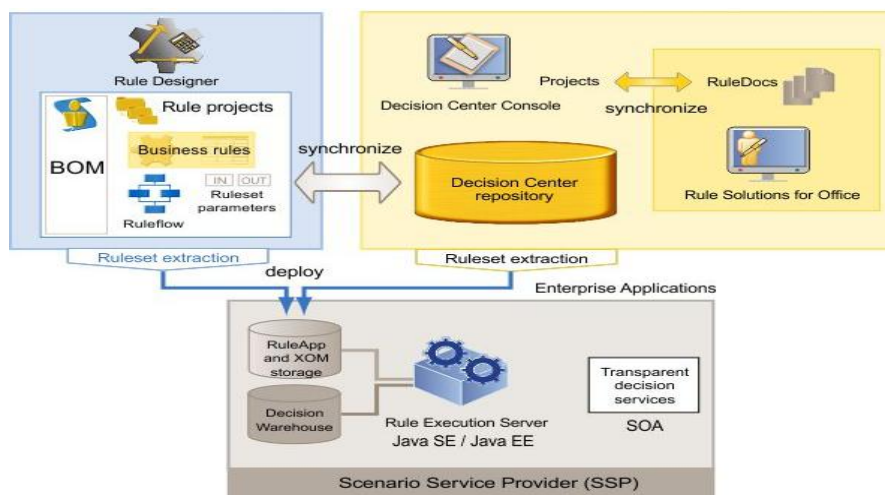
Rapporten worden voornamelijk gecreëerd voor audits en voor het communiceren van informatie naar groepsleden die samen op hetzelfde project zitten. Binnen rule designer bestaan er templates zodat de reviewer makkelijker de consistentie en compleetheid van rulesets kan controleren.

STAP 6: HET TESTEN EN UITVOEREN VAN HET RULE PROJECT (EXECUTE)

Wanneer alle rules en ruleflows opgemaakt zijn, kan de rule testing plaatsvinden, wat binnen ODM **Decision Validation** genoemd wordt. Om rules te testen kan de tester gebruik maken van Decision Validation Services om rulesets te testen en te simuleren. Het maakt het mogelijk voor gebruikers om scenario's te simuleren om de correctheid van rulesets na te gaan. Scenario's worden als input meegegeven aan de Scenario Service Provider (SSP). De output is een rapport met de resultaten van de uitvoering van de verschillende scenario's.

Het testen van rules in de DVS omgeving kan belangrijk zijn om de correctheid en efficiëntie van de rulesets na te gaan. Het geeft de business users de zekerheid dat bepaalde veranderingen die gedaan werden, ook werkelijk het gewenste resultaat hebben. Rulesets worden hierbij uitgevoerd op een grote hoeveelheid realistische 'historische' scenario's. Dit om te vermijden dat de rulesets tot foutieve resultaten leiden in de applicatie zelf. Vooral bij de veranderingen in datasets blijken simulaties van grote waarde.

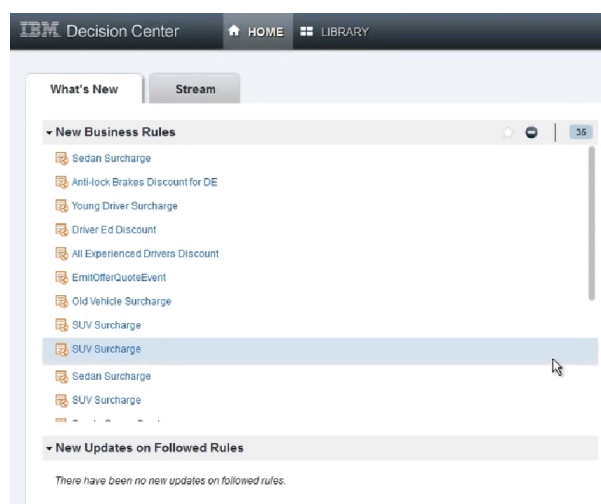
Decision Validation Services integreert de Rule Designer, die testen en simulaties mogelijk maakt met het Decision Center, dat testsuites en simulaties uitvoert en rulesets valideert en de Rule Execution Server, die de audit doet van het resultaat van uitgevoerde rulesets.



Figuur 17: Decision Validation Services
 Uit [Overgenomen uit] WebSphere Operational Decision Management V8.0, door IBM

3.2.2. HET GEBRUIKEN VAN DE BUSINESS CONSOLE (BUSINESS USERS)

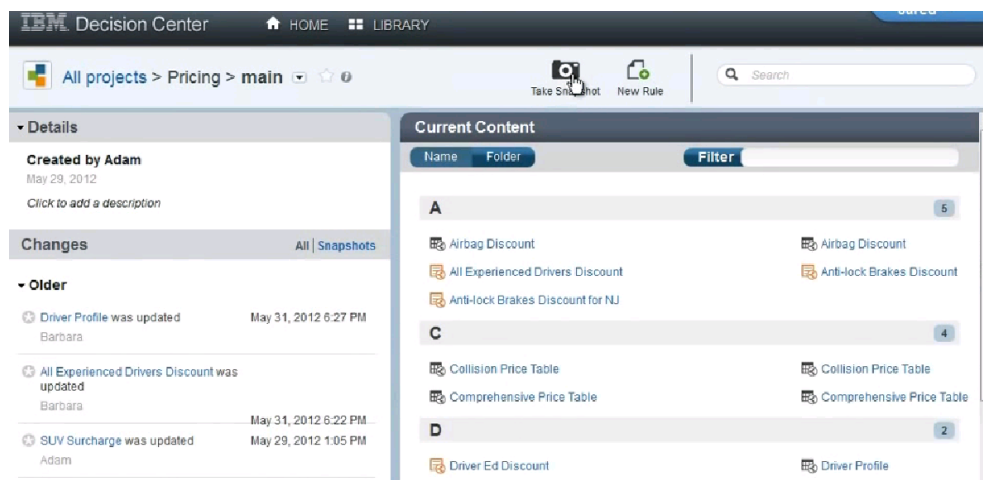
Zoals eerder vermeld werd, worden rule projecten uiteindelijk beschikbaar in het Decision Center. dit is de omgeving aangepast aan business users. Van hieruit kunnen de business users rules, decision tables en events bekijken, creëren en aanpassen.



Figuur 18: Home (business console)

In de business console in het Decision Center verkrijgt de business user bij het inloggen een home scherm waarbij wordt weergegeven wat de laatste nieuwe business rules zijn die door andere business users of ontwikkelaars werden toegevoegd. Omdat business users vaak verantwoordelijk zijn voor slechts een beperkte groep rules, kan een user zich dan ook op de updates van die rules abonneren. Veranderingen in die rules komen dan onder het tabblad 'New updates on Followed Rules'.

Onder het hoofdtabblad 'Library' krijgt de business user een overzicht van de verschillende rule packages. Deze packages groeperen rules die bij een bepaald proces of aspect horen. Indien de gebruiker een specifieke rule package zoekt, kan hij/zij de filterbalk gebruiken.



Figuur 19: De rule package pricing (Business console)

In bovenstaand scherm kan de business user zien welke recente veranderingen er in dit package gemaakt zijn. In het kader 'Current Content' krijgt de business user een overzicht van de rules en tables die in dit package werden opgenomen. Ook hier kan de gebruiker een filter toepassen.

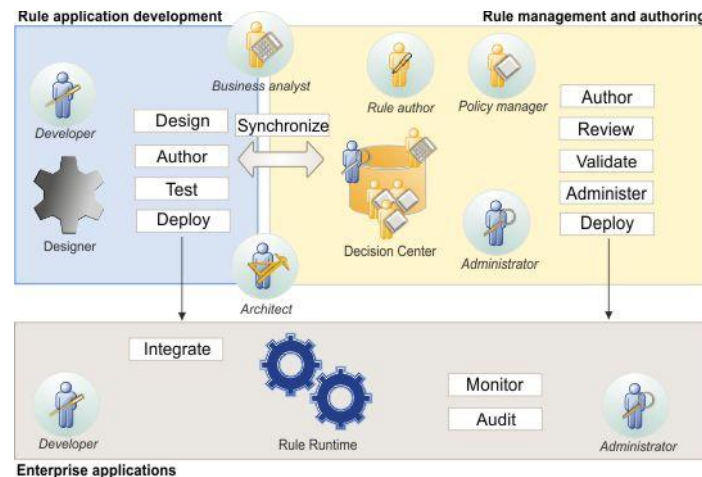
Een handige functie binnen de business console is de snapshot functie (fototoestel-icoon). Dit maakt het mogelijk voor de gebruiker om de huidige toestand van het project te bewaren. Dit kan handig zijn wanneer de business user regels aanpast, maar dit uiteindelijk niet het gewenste resultaat heeft. Via de gemaakte snapshot kan de gebruiker dan gewoon terugkeren naar de toestand voor de veranderingen.

De business user kan zowel rules als decision tables creëren en aanpassen dankzij het werk van de ontwikkelaar die alle benodigde elementen in de rule designer ontwikkelde. Op die manier kan de business user vlotter en gemakkelijker met de rules en tables aan de slag. Wanneer een gebruiker een rule creëert, krijgt hij/zij automatische suggesties die hij/zij kan gebruiken om de rule te vervolledigen. Al deze elementen zijn daar dankzij het werk van de ontwikkelaar in de rule designer.

Omdat business users niet allemaal even sterk zijn in het creëren en aanpassen van rules, kan de gebruiker raad vragen aan andere business users via de stream. Dit is een soort forum-achtige stijl waarmee gebruikers over een bepaalde rule of tabel kunnen posten en andere gebruikers hierop kunnen reageren om te helpen. De combinatie van de rule designer voor de ontwikkelaar en de

business console voor de business user maakt het mogelijk veel correcter en flexibeler om te gaan met veranderingen in rules en tables.

3.2.3. DE ROLLEN BINNEN ODM



Figuur 20: De rollen binnen ODM

Uit [Overgenomen uit] WebSphere Operational Decision Management V8.0, door IBM

Binnen ODM zijn er verschillende rollen voor verschillende gebruikers. Voor de gehele ODM omgeving kan er onderscheid gemaakt worden tussen twee grote categorieën van gebruikers.

De IT users: Architecten, ontwikkelaars, administrators die de rule applicaties ontwikkelen en onderhouden.

De business users: Business analisten, beleidsmanagers en rule authors die de decision logica ontwerpen en onderhouden.

In de tabel hieronder worden deze rollen kort toegelicht:

Rol	Beschrijving
 Architect	Rule Architecten werken vooral in de Rule Designer . Ze proberen te zorgen dat de rules op een juiste manier worden ontwikkeld, beheerd en uitgevoerd. Zij denken ook na over de link tussen de applicaties en de business rules.
 Developer	Ontwikkelaars werken voornamelijk in de Rule Designer en houden zich voornamelijk bezig met het ontwikkelen, testen en debuggen van de business rules.

Rol	Beschrijving
 Administrator	Systeemadministrators houden zich bezig met de achterliggende servers om ervoor te zorgen dat ze correct werken. Zij zetten de servers op en zorgen voor de configuratie ervan. Ze zijn verantwoordelijk voor de tracking en monitoring van de rules en hun executie.
 Business analyst	Business analisten werken zowel in de Rule designer als het decision center . Zij vormen de connectie tussen het business en IT departement. Vanaf het ontwerp van de rules tot de integratie met de applicatie. Zij ontwikkelen de specificaties, vocabulary en proberen regels te formuleren. Zij werken verder op hetgeen de ontwikkelaar ontwikkeld heeft.
 Policy manager	Beleidsmanagers zijn de eigenaars van de beslissingen en werken voornamelijk in het decision center . Zij denken mee bij het ontwerp van de specificaties en vocabulary samen met de analisten. Binnen de business console creëren en updaten ze rules. Daarnaast doen ze ook testen en simulaties zodat ze zeker zijn dat ze correct geschreven zijn.
 Rule author	Rule authors zijn de gewone business users en werken net zoals de beleidsmanagers meestal in het decision center (Business Console) . Daarin proberen de users rules te updaten en te creëren. Van tijd tot tijd kunnen users met behulp van zoekopdrachten en rapporten de rules en events nakijken.

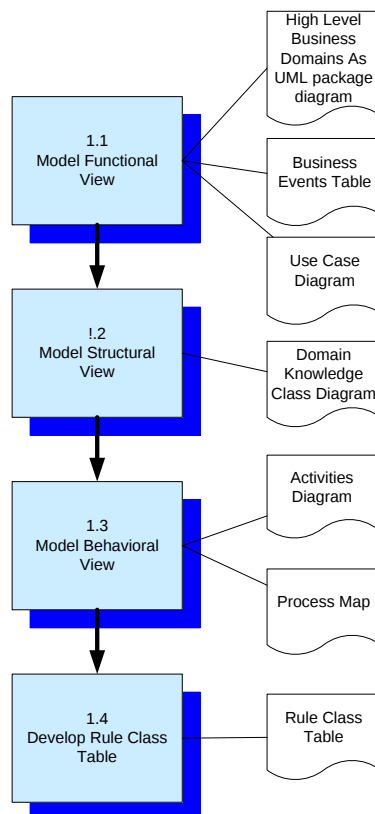
Tabel 2: Rollen in Operational decision manager

HOOFDSTUK 4: TOEPASSEN VAN DE TOOLS

Wanneer een proces gemodelleerd wordt, heeft dit als doel, het beschrijven van hoe het event gebeurt. Belangrijk hierbij is het begrip van de organisatie waarin het proces uitgevoerd wordt. Er dient steeds rekening gehouden te worden met mogelijke problemen en eventuele opportuniteiten. Vanuit de visie van het bedrijf worden er processen, rollen en verantwoordelijkheden gedefinieerd in een bedrijfsarchitectuur. Daarnaast gebeurt er ook de creaties van een use case model en een domein object model.

In deze praktijktoepassing zal de focus liggen op het business rule model dat meestal niet zo standaard is als een data- of procesmodel. Business rules maken het mogelijk om beslissingen te nemen bij een bepaalde beslissingspunten in het procesmodel. Belangrijk hierbij is een mogelijke link tussen het beslissingspunt en een bepaalde ruleset. Rulesets moeten niet specifiek gelinkt zijn aan 1 beslissingspunt of activiteit, ze hebben namelijk de meeste waarde wanneer ze ergens anders opnieuw gebruikt kunnen worden. Rulesets moeten dus als externe onafhankelijke sets van regels gezien worden.

Onderstaande figuur geeft weer hoe de eerder besproken business rules approach precies benaderd dient te worden en hoe deze ook verder uitgewerkt zal worden in het praktijkvoorbeeld met betrekking tot het EU-Rent autoverhuurbedrijf. Daar stappen 1 tot 3 minder betrekking hebben op het onderzoek van de masterproef, maar wel uitgewerkt werden, zijn deze in bijlage C te vinden. De eerste paragraaf begint dan ook bij stap 4 van de business rules approach.

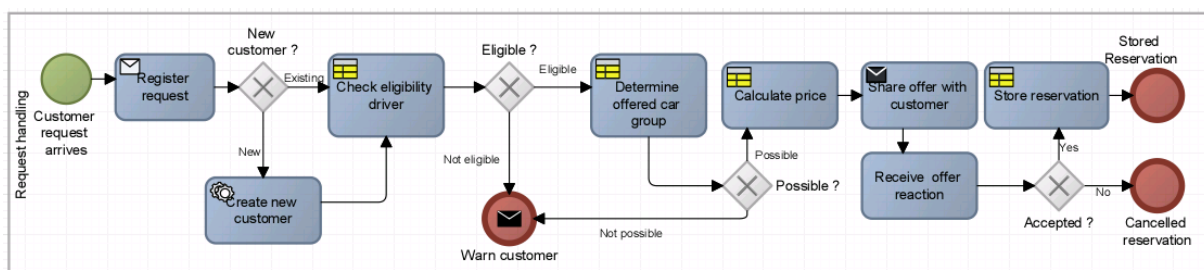


Figuur 21: Business rules approach
Uit [Overgenomen uit] Business modeling: how to guide (p.11), door IBM

4.1. STAP 1.4. HET ONTWIKKELEN VAN DE PROCESMODELLEN IN TDM

Voor het fictieve autoverhuurbedrijf EU-Rent is het autoverhuurproces een core activiteit. In de autoverhuurbusiness staat de tijd niet stil en dienen bepaalde regels vaak aangepast te worden. Het zou daarom makkelijk zijn indien de businesslogica in de mate van het mogelijke uit de processen werd gehaald en werd vertaald naar business rules. Deze vertaling zal worden getoond met behulp van de tool Architect en de module The Decision Modeler. De Business processen worden gemodelleerd via de Business Process Modeling Notation (BPMN) standaard. De rules worden opgenomen in het eerder besproken 'The Decision Model'.

BE01 – Request handling:



Figuur 22: BPM Request handling

Het eerste event, Request handling start wanneer een klant een rental request stuurt. Afhankelijk of de klant reeds in het systeem opgenomen is of niet, wordt er een nieuwe klantenfile gecreëerd. Op basis van die gegevens wordt er nagegaan of de klant wel degelijk een wagen bij EU-Rent kan huren (= Eligible). Eligibility wordt op basis van leeftijd, schadegevallen, ... bepaald. Indien de klant niet eligible is, dan krijgt de klant een melding. Wanneer de klant wel eligible is, dan wordt er gekeken of de gevraagde Car Group die periode beschikbaar is of niet. Indien niet, wordt er gekeken of het filiaal een andere mogelijkheid kan aanbieden. Indien er een beschikbare Car Group gevonden wordt, dan wordt de prijs geschat. De klant krijgt dan de kans om het aanbod te nemen of te weigeren. Indien de klant akkoord gaat, dan wordt het request in het systeem vastgelegd.

Voor dit event werd er per beslissing een decision model en de bijbehorende rule families gebouwd. De beperking van TDM was voornamelijk dat er enkel rule family tables aangemaakt konden worden en geen aparte business rules, wat vaak leidde tot omslachtige tabellen. Daarnaast waren beperkingen zoals leeftijd berekenen niet mogelijk, zo diende er vaak input meegegeven te worden die normaal berekend zou moeten worden door het systeem. Een uitgebreidere evaluatie hierover wordt gegeven in hoofdstuk 5. Ondanks deze beperkingen was het toch een goede tool om dit proces vast te leggen. Hetgeen uitgewerkt kon worden vormde een duidelijke basis voor een ontwikkelaar die in een complexere omgeving deze rules uitvoerbaar dient te maken.

In de volgende bladzijden wordt de glossary getoond die de basis vormt voor de beslissingen. Deze glossary maakt het makkelijker voor de ontwikkelaar daar hij kan zien voor iedere variabele wat het domein is, wat ze betekent en tot welke UML klasse ze behoort. Zo kan dit later in de Java file overgenomen worden.

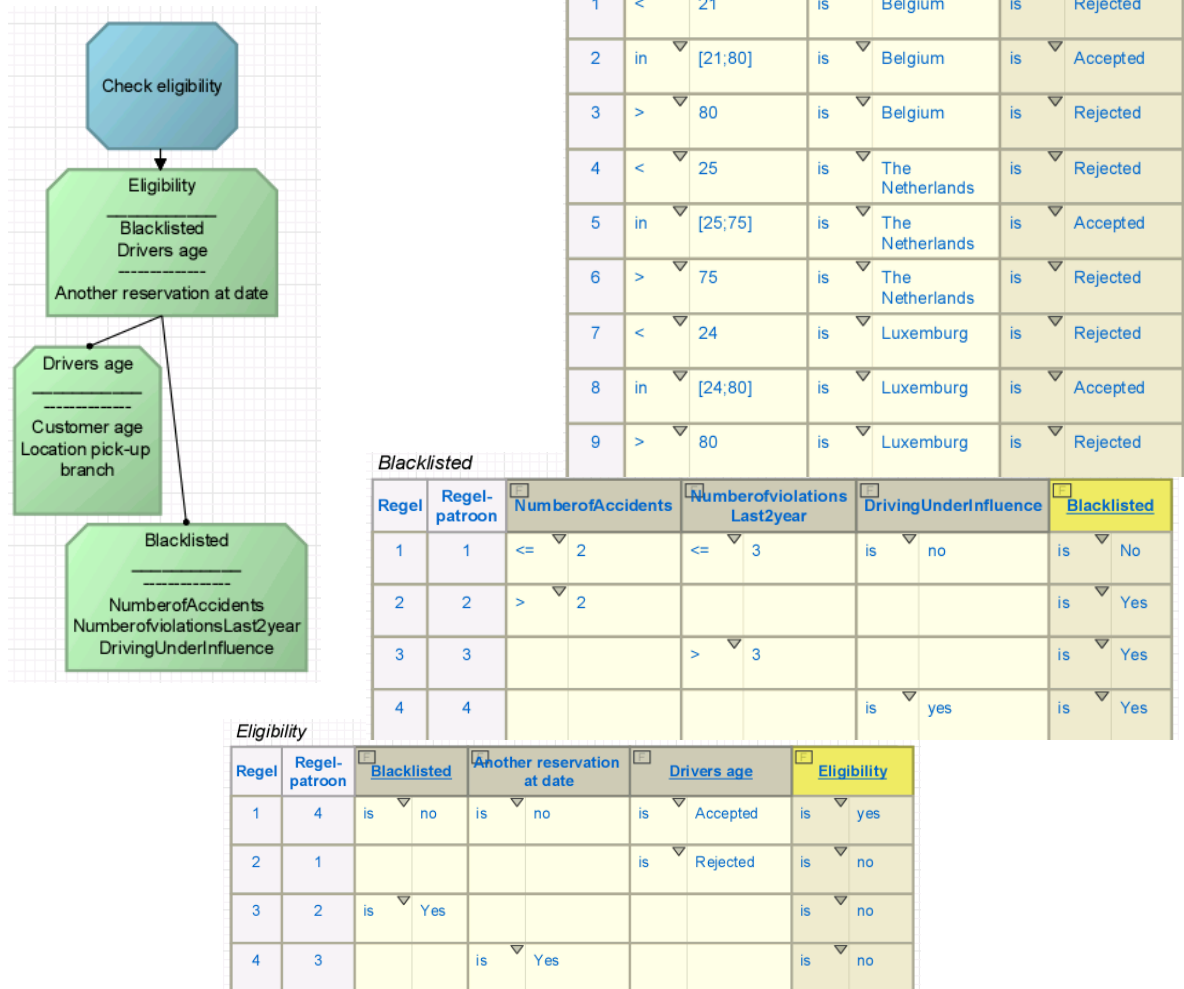
Glossary

Feitttype	Huidige	Domein	Beschrijving	Entiteittype	Entiteitattribuut
Location pick-up branch		Belgium, The	In which country is the pick-up branch located	Branch	Location pick-up branch
Customer age		integer	How old is the customer ?	Customer	Customer age
<u>Drivers age</u>		Accepted, Rejected	Is the customer old enough ?	Customer	Drivers age
DrivingUnderInfluence		boolean	Has the customer been driving a rental car under influence ?	Customer	Driving under influence
NumberOfViolationsLast2years		integer	How many violations did the customer do with a rental car	Customer	NumberOfViolationsLast2year
NumberOfAccidents		integer	How many accidents was the customer involved in with a rental	Customer	NumberOfAccidents
OfferAcceptance		boolean	Does the customer accept the offer ?	RentalAgreement	OfferAcceptance
Another reservation at		boolean	Does the customer have another reservation at that date ?	Customer	Another reservation at
<u>Blacklisted</u>		boolean	Has the customer appeared on the blacklist ?	Customer	Blacklisted
<u>Reservation</u>		tekst	The summary of the rental agreement	RentalAgreement	Reservation
<u>Eligibility</u>		boolean	Is the customer eligible to rent a car ?	Customer	Eligibility
days		getal	How many days does the customer want to rent ?	RentalAgreement	days
Startdate rental period		datum	When does the rental period start ?	RentalAgreement	Startdate rental period
<u>Price</u>		geld	What is the price of the offer ?	RentalAgreement	price
Requested car group		economy, compact,	What was the car group the customer requested ?	RentalAgreement	requested car group
<u>Possible car group</u>		economy, compact,	What is the possible car group that can be offered ?	RentalAgreement	Possible car group
number of cars waiting		integer	How many cars in the group in a certain branch need maintenance	Branch	number of cars waiting for
number of cars of		integer	How many cars are in the upgraded group in a certain branch	Branch	number of cars of downgraded
number of cars of		integer	How many cars are in the downgraded group in a certain	Branch	number of cars of upgraded car
number of cars of		integer	How many cars are in the requested group in a certain branch	Branch	number of cars of requested car
<u>offered car group</u>		do w upgrade, downgrad	Will the branch be able to reserve the requested, upgraded or	RentalAgreement	Offered car group

Figuur 23: Glossary request handling

De glossary bevat alle elementen die gebruikt werden in de decision models en rule families voor het proces 'request handling'.

Beslissingsmodel 1: Check Eligibility



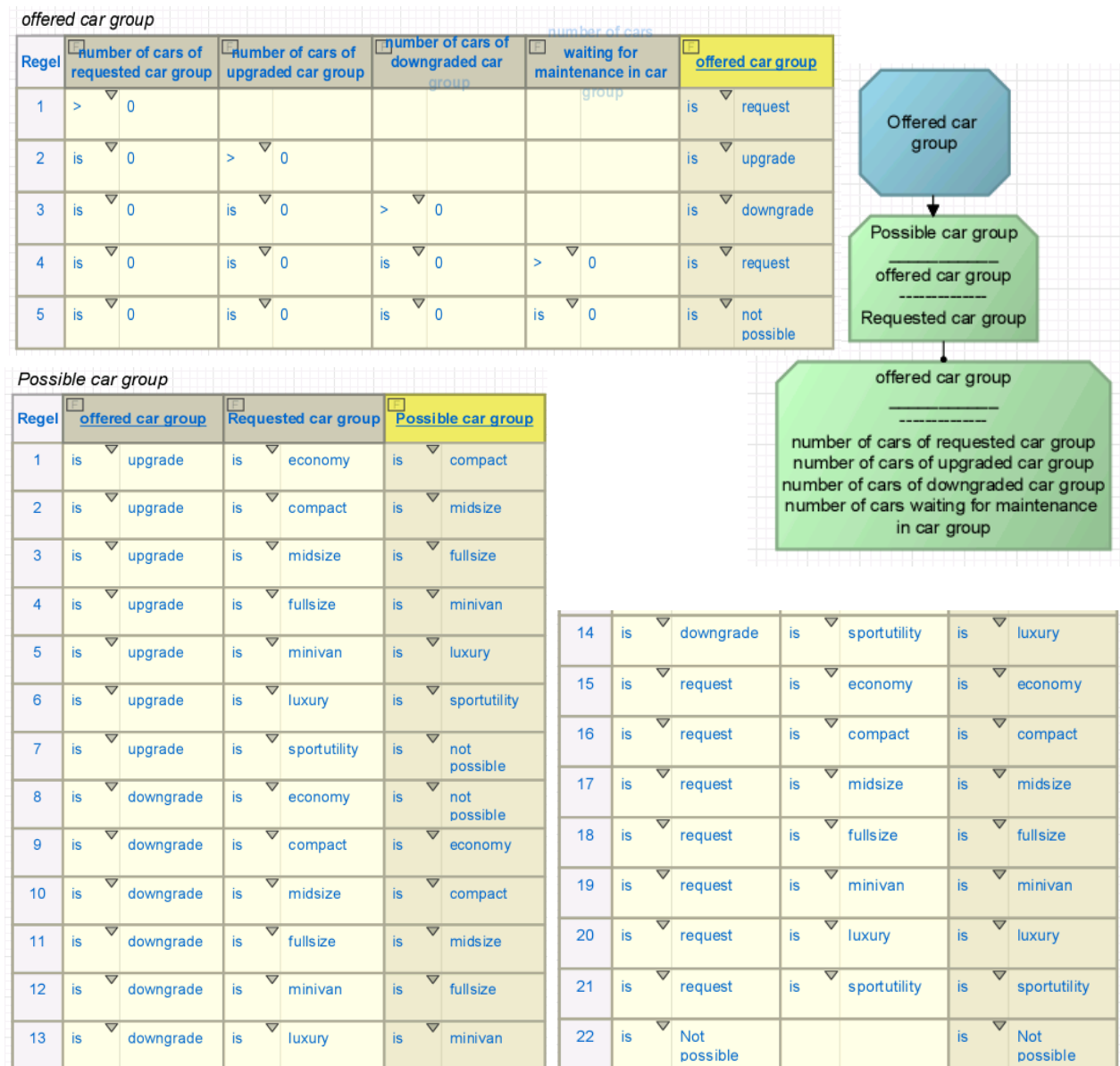
Figuur 24: The decision model en rule families (check Eligibility)

De allereerste beslissing wordt gemaakt op basis van de gegevens van de klant. Dit leidt tot het ofwel accepteren of weigeren van deze klant. Zoals uit het decision model toont, is 'eligibility' gebaseerd op 3 fact types: Drivers Age, Blacklisted en Another reservation at date.

Customer Age dient in deze tool als een getal ingegeven te worden, het gebruik van het verschil tussen geboortedatum en de huidige datum werd door de tool niet aanvaard. Afhankelijk van het land waarin de wagen opgepikt wordt, verschilt de minimumleeftijd. Het fact type 'Another reservation at date' dient in Operational Decision Manager (ODM) uitgewerkt te worden, daar de ontwikkelaar met behulp van business rules kan nagaan of een klant al een bepaalde wagen op die datum gereserveerd heeft. Deze logica kan niet binnen TDM gemodelleerd worden. Daarmee dient de ontwikkelaar hiervoor de glossary te gebruiken om na te gaan wat het fact type 'Another reservation' at date betekent en hoe dit in ODM geprogrammeerd dient te worden.

Het condition fact type 'Blacklisted' is afhankelijk van een ondersteunende rule family: Blacklisted. Dit houdt in dat er voor iedere klant nagegaan wordt hoeveel ongevallen en overtredingen de klant reeds gedaan heeft met een wagen van EU-Rent. Ook wordt er opgezocht of de klant al eerder betrapt werd op het rijden onder invloed in een gehuurde wagen. Afhankelijk van de 3 fact types: Drivers Age, Blacklisted en Another reservation at date, kan er dan beslist worden of een klant een wagen zal kunnen huren of niet.

Beslissingsmodel 2: Determine car group



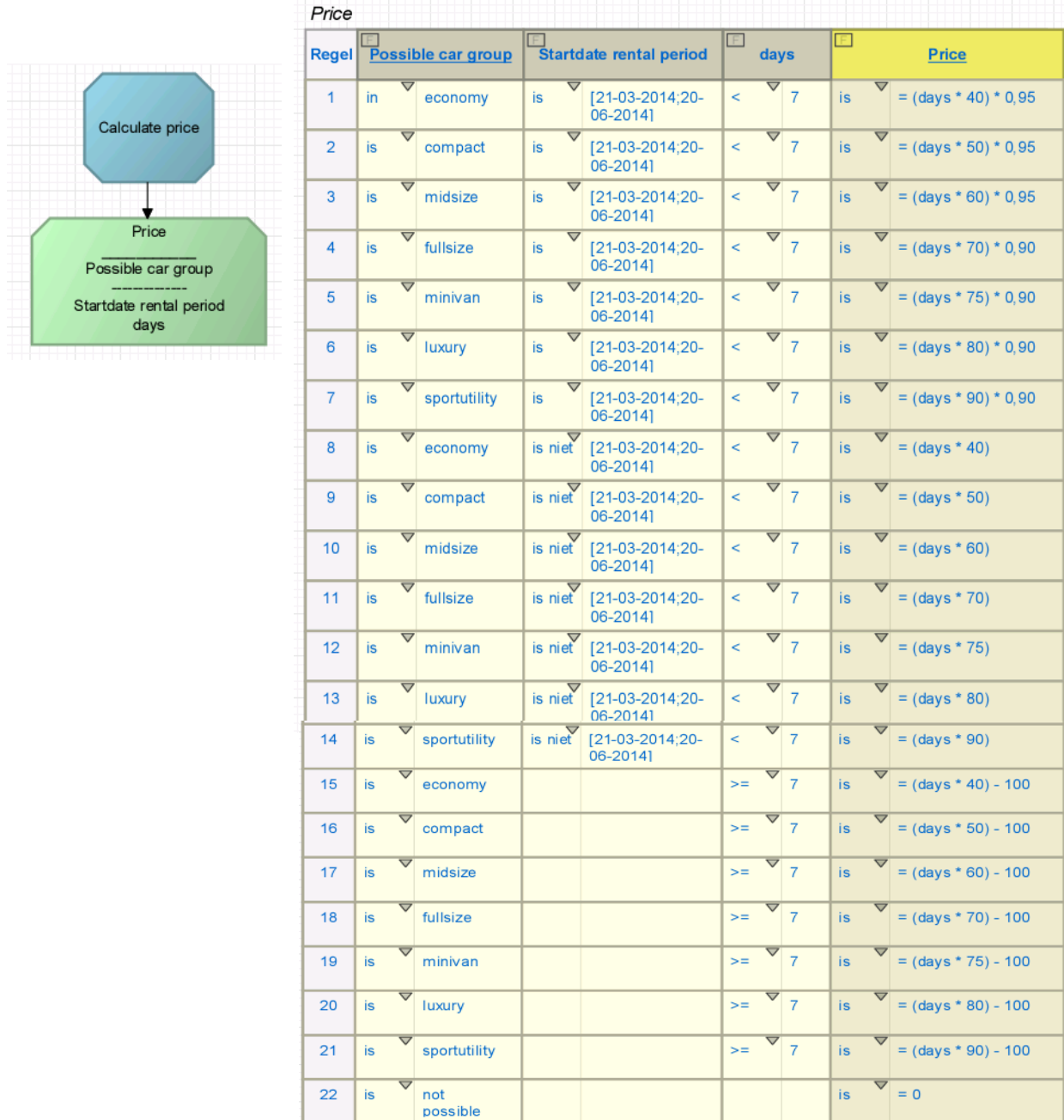
Figuur 25: The decision model en rule families (Determine car group)

Voor de tweede beslissing wordt er nagegaan hoeveel wagens er nog van iedere Car Group beschikbaar zijn. Wanneer er nog voldoende wagens zijn in de gewenste Car Group is er geen probleem. Wanneer dit niet het geval is, moet er nagegaan worden of er eventueel in de hogere Car Group of de lagere Car Group een wagen beschikbaar is. Indien ook dit niet het geval is, wordt er nagegaan of er nog een wagen binnen de gewenste Car Group beschikbaar was die binnenmoest voor onderhoud, maar nu eenmalig toch nog verhuurd kan worden.

Afhankelijk van die beschikbare aantallen wordt dus de gevraagde Car Group gereserveerd of biedt EU-Rent een downgrade of een upgrade aan. Indien niets van al deze mogelijk is, dan wordt dit aan de klant gemeld.

Omdat binnen TDM enkel met rule families gewerkt kan worden en niet gewoon met business rules, krijgt de gebruiker vaak omslachtige tabellen voor redelijk simpele logica zoals te zien is.

Beslissingsmodel 3: Calculate price



Figuur 26: The decision model en rule families (Calculate price)

Afhankelijk van de wagen die aangeboden wordt, wordt er een prijs berekend. Deze is afhankelijk van de fact types: Possible Car Group, Start date rental period, days (lengte verhuurperiode).

Eu-Car rental biedt een lentekorting aan, dit kan gezien worden in de kolom 'Start date rental period'. Wanneer een klant dus in de lente een wagen komt huren, krijgt deze afhankelijk van de Car Group een korting. Ook bestaat er nog de long term korting die de klant krijgt wanneer hij/zij 7 dagen of meer dan 7 dagen een wagen wenst te huren. Een klant kan nooit van twee kortingen tegelijk genieten, daarom worden er steeds meerdere prijzen berekend, in dit geval, eenmaal met de lentekorting en eenmaal met de 'long term' korting. Het systeem kiest dan het goedkoopste aanbod, en dit wordt aan de klant voorgelegd. Deze logica kon niet in TDM gemodelleerd worden, maar zal wel in ODM aan bod komen.

Beslissingsmodel 4: Store reservation



Figuur 27: The decision model en rule families (Store reservation)

Wanneer de driver eligible is, er een mogelijke Car Group en prijs gevonden werd, dan kan EU-Rent dit voorstel aan de klant voorleggen en vragen of hij akkoord gaat of niet. Dit kan dan leiden tot een reservatie.

Omdat de decision models gelinkt zijn aan het procesmodel via de beslissingspunten, werd er verwacht dat de uitkomst van een bepaalde beslissing ook effect zou hebben op de gevolgde weg in het proces. In realiteit zou dit moeten, maar in TDM bleek deze functionaliteit nog te ontbreken. De gebruiker kan het procesmodel wel linken met de decision models, maar dit heeft nog geen effect. Na het versturen van een mail naar TDM werd duidelijk dat dit in de volgende versie zou opgenomen worden.

Dit zou inhouden dat wanneer een bepaalde klant niet eligible is, de andere beslissingen niet meer doorlopen hoeven te worden, wat zowel duidelijkheid schept als tijd en rekenkracht bespaart.

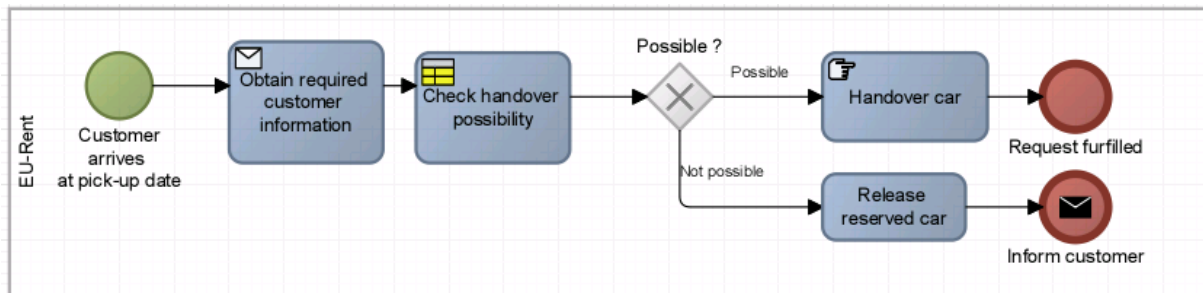
Omdat dit nog niet het geval is, werd voor deze praktijkoefening geopteerd om deze beperking toch te omzeilen door een extra decision model (figuur 27) te creëren dat de gebruiker in staat stelt om het resultaat van alle vorige beslissingen te combineren in 1 regel. Om dit aan te kunnen bieden, werd er gebruik gemaakt van een 'list fact type', dit houdt in dat zo'n fact type niet 1 waarde kan bevatten, maar een hele lijst van waardes. Zo kan de gebruiker in één oogopslag zien of de klant eligible is, welke wagen EU-Rent kan aanbieden, wat de prijs van de verhuur is en of de klant met het aanbod akkoord gaat. Wanneer dit model niet gecreëerd werd, moest de gebruiker zelf gaan zoeken in de lijst wat de uitkomst van de verschillende beslissingen was.

Via de scenario analyse die uitgevoerd kan worden binnen TDM met behulp van Microsoft Excel, kan de gebruiker nagaan wat deze verschillende modellen hierboven als resultaat hebben wanneer hij/zij een bepaald scenario invoert. Deze scenario analyse wordt op de volgende pagina getoond. Zoals uit figuur 28 zal blijken, werden er vier scenario's getest. De onderste regel van ieder scenario is de uitkomst die werd verkregen door het toevoegen van het laatste model in figuur 27 en geeft de resultaten samengevat weer.

	Fact type								
simple fact types	Another reservation at date	no	yes	no	no	no			
	Customer age		27	35	48				27
	days		10	14	9				5
	DrivingUnderInfluence	no	no	no	no				
	Location pick-up branch	Belgium	Luxemburg	The Netherlands	Belgium				
	number of cars of downgraded car group		5	4	7				0
	number of cars of requested car group		4	7	0				0
	number of cars of upgraded car group		6	8	6				4
	number of cars waiting for maintenance in car group		7	2	2				0
	NumberofAccidents		0	1	2				2
	NumberofviolationsLast2year		7	1	2				1
	OfferAcceptance	yes	yes	yes	no				
	Requested car group	fullsize	minivan	luxury	economy				
Derived fact types: expected values (optional)	Startdate rental period		21/06/2014	14/09/2014	21/04/2014				21/04/2014
	Blacklisted								
	Drivers age								
	Eligibility								
	offered car group								
	Possible car group								
	Price								
	Reservation								
	Blacklisted	ja	nee	nee	nee				
	Drivers age	accepted	accepted	accepted	accepted				
	Eligibility	nee	nee	ja	ja				
	offered car group	request	request	upgrade	upgrade				
	Possible car group	fullsize	minivan	sportutility	compact				
Derived fact types: actual derived values	Price		600	950	710				250
	Reservation								
		Driver is not eligible	Driver is not eligible	Driver is not eligible	Driver is Eligible, EUR 710.00, Offer is accepted, sportutility	Driver is Eligible, EUR 250.00, Offer is not accepted, compact			

Figuur 28: Excel scenario batch test

BE02 – Car handover:



Figuur 29: BPM Car handover

Het tweede event, Car Handover start wanneer de klant de wagen oppikt. Op dat moment wordt de klant gecontroleerd en nagekeken of de wagen in orde is. Afhankelijk daarvan, kan de wagen overhandigd worden of niet. Wanneer het niet in orde is, dan wordt de wagen terug vrijgegeven. Indien alles wel in orde is, dan wordt de autosleutel en de wagen aan de klant overhandigd.

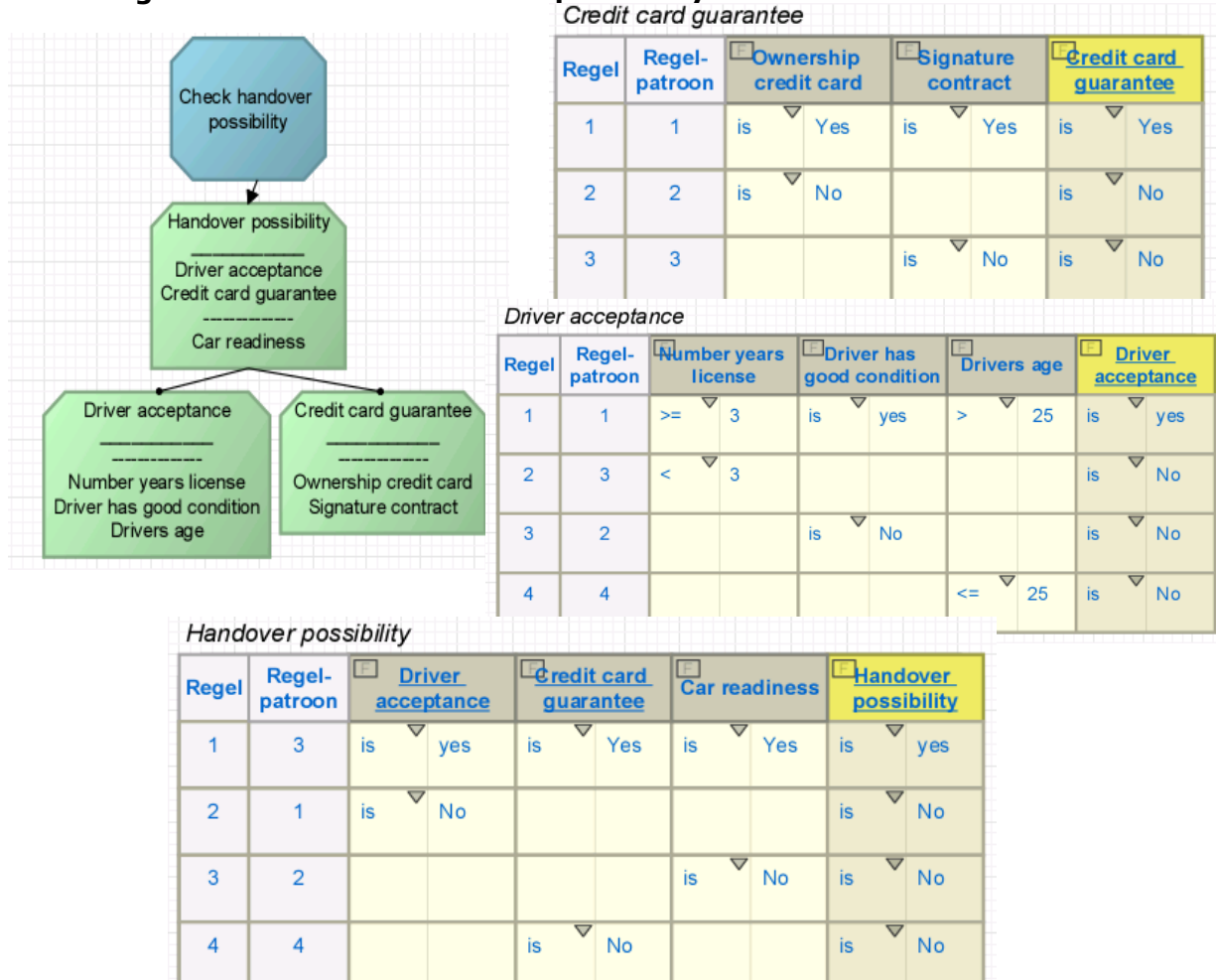
Binnen het decision model gebouwd rond de handover possibility wordt er veel gewerkt met booleans, dit omdat veel fact types door een filiaalwerknemer zelf nagekeken dienen te worden. Zo moet de werknemer in de vestiging zelf nakijken of de wagen klaar is, of de persoon in goede staat verkeert en of de klant zijn handtekening op het contract heeft geplaatst.

Glossary

Feittype	Huidige	Domein	Beschrijving	Entiteitstype	Entiteitattribuut
Car readiness		boolean	Checked by agent if car has been cleaned, has full tank of fuel, oil and	Car group	Car readiness
Signature contract		boolean	Has the customer signed the contract ?	Car rental	Signature contract
Ownership credit card		boolean	Does the customer own the credit card ?	Customer	Ownership credit card
<u>Credit card guarantee</u>	ja	boolean	Does the credit card cover the rental ?	Customer	Credit card guarantee
Drivers age		integer	Age of the customer, must be more than 25	Customer	Age
Driver has good condition		boolean	Is the customer not under the influence of alcohol or drugs, and physically able	Customer	Driver Condition
Number years license		integer	Number of years a customer has a drivers license	Customer	Years of license
<u>Handover possibility</u>	ja	boolean	Can the car be handed over to the customer ?	Car rental	Handover possibility
<u>Driver acceptance</u>	ja	boolean	Is the driver accepted to drive with the car ?	Customer	Driver acceptance

Figuur 30: Glossary car handover

Beslissingsmodel 1: Check handover possibility



Figuur 31: The decision model en rule families (Check handover possibility)

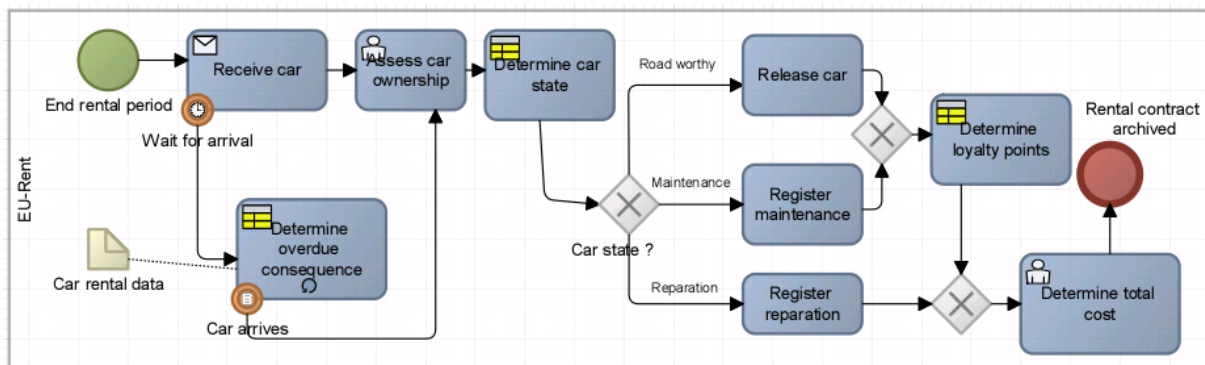
Wanneer de klant aankomt dient de bediende in de vestiging de credit card garantie na te kijken. Dit is enkel in orde wanneer de credit card van de chauffeur zelf is en hij/ zij een handtekening op het contract heeft gezet. Daarnaast dient de chauffeur meer dan drie jaar een rijbewijs te bezitten, 25 of meer dan 25 jaar te zijn en dient de chauffeur in goede staat te zijn, dit houdt in: niet onder de invloed van alcohol of drugs en fysiek in orde om een wagen te besturen.

Ook voor dit event werd een scenario analyse gedaan om te zien of de regels het gewenste resultaat gaven.

	Fact type	Domain				
Simple fact types	Car readiness	boolean	yes	yes	yes	yes
	Driver has good condition	boolean	yes	no	yes	yes
	Drivers age	integer	29	29	21	27
	Number years license	integer	14	14	4	1
	Ownership credit card	boolean	yes	yes	yes	yes
	Signature contract	boolean	yes	yes	yes	yes
Derived fact types: expected	Credit card guarantee	boolean				
	Driver acceptance	boolean				
	Handover possibility	boolean				
Derived fact types: actual	Credit card guarantee	boolean	ja	ja	ja	ja
	Driver acceptance	boolean	ja	nee	nee	nee
	Handover possibility	boolean	ja	nee	nee	nee

Figuur 32: Excel scenario batch test

BE03 – Car return:



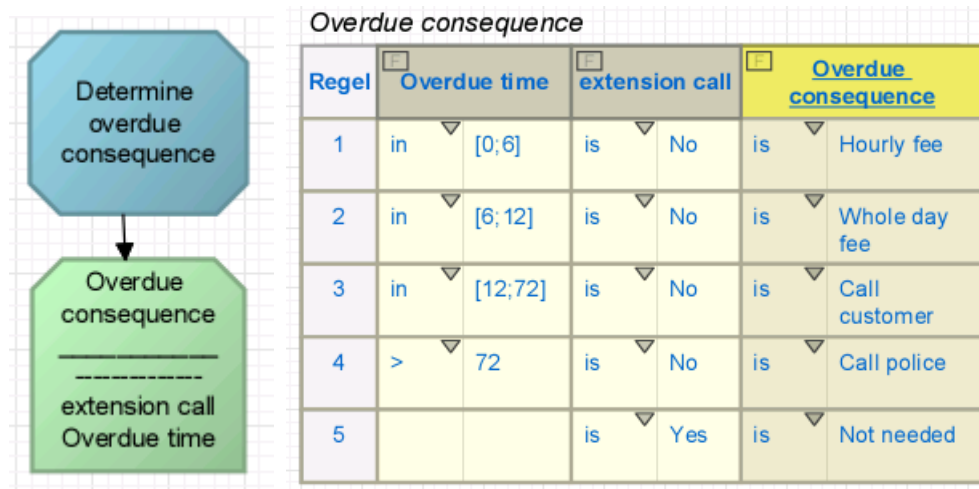
Figuur 33: BPM car return

Het derde event, Car return begint wanneer de verhuurperiode eindigt. Wanneer de wagen terug binnengebracht wordt, wordt de normale sequentie gevolgd. Wanneer de wagen niet op tijd of niet wordt teruggebracht, wordt het gevolg hiervan bepaald. Wanneer een wagen toekomt in een filiaal krijgt dit filiaal de eigendom. Dit omdat een wagen niet moet teruggebracht worden in het filiaal waar hij afgehaald werd. De wagen wordt dan onderzocht op schade en wordt ofwel terug vrijgegeven, gereserveerd voor onderhoud of reparatie afhankelijk van de staat van de wagen. De loyaliteitspunten worden berekend en de totale kosten voor de klant worden bepaald op basis van de prijs die berekend werd in het request handling proces.

Glossary					
Feittype	Huidige	Domein	Beschrijving	Entiteittype	Entiteitattribuut
Offered car group		economy, compact,	The car the customer rented	Car group	Offered car group
Number car rentals		integer	The number of rentals a customer did with our	Customer	Number of car rentals
<u>Overdue consequence</u>		Hourly fee, whole day fee,	What is the punishment for a late return	Car rental	Overdue consequence
Free rental		boolean	Was the rental done by a free rental coupon	Car rental	Free rental
<u>Loyalty points</u>		integer	Number of loyalty points of a customer	Customer	Loyalty points
extension call		boolean	Did the customer call for an extension of the rental	Car rental	Extension call
Overdue time		getal	How much too late is a return	Car rental	Overdue time
Car damage		boolean	Is the car damaged ?	Car	Car damage
Car wear		boolean	Is there wear seen on the car	Car	Car wear
<u>Car state</u>		tekst	What is the general state of the car ?	Car	Car state

Figuur 34: Glossary car return

Beslissingsmodel 1: Determine overdue consequence



Figuur 35: The decision model en rule families (Determine overdue consequence)

Wanneer de wagen op tijd wordt teruggebracht is er geen probleem, maar wanneer dit niet het geval is, wordt afhankelijk van de overschreden tijd bepaald wat het gevolg is. Wanneer de klant verwittigd heeft dat hij/zij te laat ging zijn via een telefoontje, dan is er geen probleem, anders dient er wel gevolg aan gegeven te worden. Zo wordt eerst het aantal uren extra aangerekend wanneer de klant minder dan 6 uur te laat is. Wanneer de klant tussen de 6 en 12 uren na het afgesproken uur terugkomt wordt hem/haar een hele dag aangerekend. Eenmaal de klant na 12 uur nog steeds niet terug is, wordt de klant gebeld en na 72 uur wordt de politie gebeld om de wagen op te sporen.

Beslissingsmodel 2: Determine car state



Figuur 36: The decision model en rule families (Determine car state)

Als de wagen terug in het filiaal is, wordt er nagekeken of er schade is en/of dat er bepaalde onderdelen vervangen dienen te worden. Afhankelijk van de fact types 'car wear' en 'car damage' wordt bepaald of de wagen onderhouden, hersteld of gewoon terug vrijgegeven dient te worden.

Beslissingsmodel 3: Determine loyalty points



Figuur 37: The decision model en rule families (Determine loyalty points)

Wanneer de wagen niet te laat werd binnengebracht, kan het systeem de andere fact types nakijken om na te gaan of de klant in aanmerking komt voor loyaliteitspunten die hij/zij kan gebruiken om een gratis rental te verkrijgen. Een klant komt pas in aanmerking voor loyaliteitspunten vanaf de vierde rental. Wanneer de huidige rental een free rental was, dan worden er geen punten toegekend. Als laatste conditie wordt de car Group nagekeken, hoeveel te hoger de klasse, hoeveel te meer punten een klant toegekend krijgt.

Ook voor dit derde proces werden scenario's uitgevoerd door middel van een Excel file die TDM via een batch test zelf invult.

	Fact type				
Simple fact types	Car damage	no	no	no	yes
	Car wear	no	yes	no	no
	extension call	no	yes	no	no
	Free rental	no	no	no	no
	Number car rentals	7	5	2	6
	Offered car group	compact	luxury	minivan	midsize
	Overdue time	45	45	78	2
Derived fact types:	Car state	Road worthy	Maintenance	Road worthy	Reparation
	Loyalty points	0	300	0	0
	Overdue consequence	call customer	not needed	call police	hourly fee
Derived fact types:	Car state	Road worthy	Maintenance	Road worthy	Reparation
	Loyalty points	0	300	0	0
	Overdue consequence	call customer	not needed	call police	hourly fee

Figuur 38: Excel scenario batch test

4.2. STAP 1.5. HET ONTWIKKELEN VAN DE BUSINESSRULES OP BASIS VAN DE PROCESMODELLEN EN BESLISSINGSMODELLEN IN ODM

Een ander deelaspect in deze masterproef was het onderzoeken van de inspanning die het met zich meebrengt wanneer modellen uit 'The Decision Modeler', in de Operational Decision Manager (ODM) gemodelleerd en uitgevoerd zouden moeten worden. Om dit te kunnen bepalen, werd er binnen ODM, meerbepaald met behulp van de Rule designer tool een rule project gebouwd gebaseerd op de Car Rental modellen in TDM. Voor dit deel werd er gefocust op een verdere uitwerking van het eerste business event van de Car rental company, namelijk de 'request handling'. Daar Operational Decision Management een grote en complexe omgeving is die gebruikt wordt door zowel ontwikkelaars, analisten, architecten en business users was het niet zo simpel om met deze omgeving te leren werken. Vooral het opstellen van uitgebreide java klassen die alle nodige elementen omvatten voor de rules, was redelijk complex. Daarom werd de basis voor de java files gelegd via de tutorial files van ODM en kon er ook gerekend worden op de ondersteuning van de heer Vanhoenshoven van AGServ.

Om een ruleproject te maken dient er eerst een XOM gecreëerd te worden zoals aangehaald in de werking van de Rule designer in hoofdstuk 3. De XOM voor dit project had de volgende Java klassen nodig: Branch, Cargroup, Customer, DateUtil, Model, Modelfeeder, Rentalagreement en Session. Hieronder worden kort de belangrijkste java klassen besproken die nodig zijn om het vervolg te begrijpen.

De Branch class omvat de mogelijkheid tot het creëren van een specifieke vestiging (bepaald door locatie en land), het aantal beschikbare wagens binnen een bepaalde Car Group en de mogelijkheid tot het registreren van een pick-up en return van een wagen.

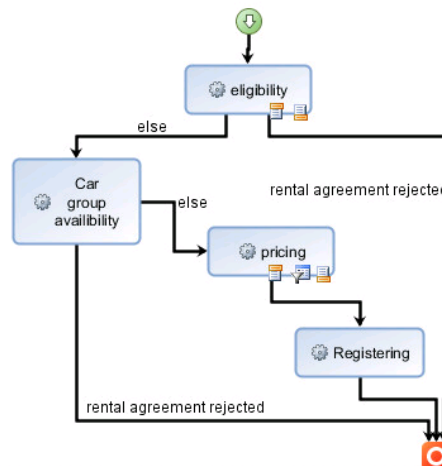
De CarGroup class omvat de mogelijkheid tot het creëren van een specifieke Car Group die een bepaalde dagprijs en weekprijs heeft. Daarnaast maakt de javafile het ook mogelijk een upgrade en een downgrade voor een bepaalde klasse te bepalen.

De Customer class omvat de mogelijkheid tot het creëren van een specifieke klant met een bepaalde voornaam, achternaam, geboortedatum en of deze klant deel uitmaakt van de bijgehouden blacklist.

De RentalAgreement class vormt de link tussen de andere Javaklassen en omvat de mogelijkheid tot het creëren van een specifiek rental agreement voor een bepaalde klant die een bepaalde Car Group, pick-up datum, return datum, pick-up branch, return branch en verzekering verkiest.

Op basis van deze XOM werd er een BOM gecreëerd voor het rule project. Binnen deze BOM kwamen de hiervoor toegelichte javaklassen terug onder de vorm van BOM elementen. Voor de verschillende BOM elementen werd er een verbalisatie toegekend die het mogelijk maakt voor business users de elementen beter te begrijpen. Op deze manier wordt het makkelijker rules te creëren omdat deze beter bij de natuurlijke taal aansluiten.

Voor het testen van de Rule Designer werd er gekozen de request handling uit te werken. Dit hield in: het nakijken of een klant eligible is, het nagaan of de gevraagde wagen beschikbaar is. Indien dit niet het geval is, een andere mogelijkheid aanbieden. Het berekenen van de geschatte prijs en het eventueel reserveren van de wagen. Dit houdt in dat de voorraad van een bepaalde klasse gedurende de rental period van die klant met 1 zal verminderen.



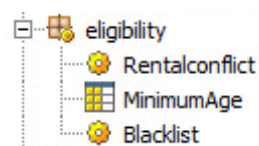
Figuur 39: Rule flow request handling

Zoals figuur 39 weergeeft worden de beslissingspunten die in TDM gemodelleerd werden, terug opgenomen. Het grote voordeel in ODM is de mogelijkheid tot het afbreken van de ruleflow wanneer aan een bepaalde voorwaarde niet wordt voldaan. Dit is tot nu toe in TDM niet mogelijk waardoor steeds alle beslissingen doorlopen worden.

Binnen deze ruleflow zal er eerst nagekeken worden of een klant eligible is. Indien niet, dan wordt de flow meteen beëindigd. Indien wel, dan wordt er gekeken of er nog wagens beschikbaar zijn op de aangevraagde datum. Eenmaal er een wagen beschikbaar is, wordt de voordeligste prijs voor de klant berekend. Het eindigt met een registratie die zorgt dat het aantal beschikbare wagens van de gekozen groep in het filiaal met 1 verminderd gedurende de verhuurperiode.

De verschillende rule packages ('Eligibility', 'Car Group availability', 'pricing' en 'registering') uit de bovenstaande ruleflow worden hieronder gedetailleerd besproken.

4.2.1. ELIGIBILITY



Figuur 40: Decision package eligibility (decision modeler)

Voor de eligibility beslissing worden drie factoren in rekening gebracht: er wordt gekeken naar de leeftijd van de klant binnen een bepaald land, naar de aanwezigheid/afwezigheid op de blacklist en of de klant reeds een wagen gereserveerd heeft op die datum. Dit om te voorkomen dat een klant twee wagens reserveert op een zelfde moment.

Action Rule 1: Bepalen eligibility afhankelijk van andere rental reservations

```
definitions
set 'Previous Agreement' to a rental agreement in 'the agreement history'
where the customer of this rental agreement is the customer of 'the current rental agreement' ;

if the pickup date of 'the current rental agreement' is after or the same as the pickup date of
'Previous Agreement' and the pickup date of 'the current rental agreement' is before or the same as
the return date of 'Previous Agreement'

then
reject 'the current rental agreement' ;
in 'the current session' , display the message :
'the current rental agreement' description +
'is rejected. Customer already has reservation at that moment" ;
```

Met behulp van bovenstaande action rule wordt er nagegaan of een bepaalde klant een nieuwe reservering wil doen met een startdatum (pickup date) die valt in een periode waar de klant reeds een wagen gereserveerd heeft. Wanneer dit het geval is, dan wordt de request verworpen en krijgt de gebruiker het bericht dat de klant reeds een reservering heeft op dat moment.

Beslissingstabel 1: Bepalen eligibility op basis van leeftijd en land

Branch location	Age of the customer		Rental rejected	Output
	Minimum Age	Maximum Age		
Belgium	0	21	true	This customer is younger than 21
	21	80	false	
	80	200	true	This customer is older than 80
The Netherlands	0	25	true	This customer is younger than 25
	25	75	false	
	75	200	true	This customer is older than 75
Luxemburg	0	24	true	This customer is younger than 24
	24	80	false	
	80	200	true	This customer is older than 80

Branch location (condition)

```
the state of the pickup branch of 'the current rental agreement' is <an object>
```

Age of customer (condition)

```
the customer's age of 'the current rental agreement' is at least <min> and less than <max>
```

Rental rejected (action)

```
make it <a boolean> that 'the current rental agreement' is rejected
```

Voor de beslissing met betrekking tot de minimum en maximum leeftijd voor het huren van een wagen wordt deze beslissing nagegaan door middel van een beslissingstabel. Afhankelijk van het land waarin de klant de wagen oppikt, zijn deze leeftijdsgrenzen anders.

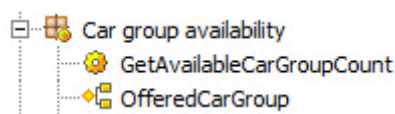
Action Rule 2: Bepalen eligibility op basis van de aanwezigheid/afwezigheid op de blacklist

```
if the blacklist of the customer of 'the current rental agreement' is "Blacklisted"
then
reject 'the current rental agreement';
in 'the current session' , display the message : 'the current rental agreement' description +
"is rejected. Customer cannot be Blacklisted" ;
```

Als de klant op de blacklist staat, dan kan deze persoon geen wagen huren.

Op basis van de bovenstaande rules en table wordt er beslist of de klant eligible is of niet. Indien wel, dan wordt de 'Car Group availability' geëvalueerd. Anders wordt het proces afgebroken.

4.2.2. CAR GROUP AVAILABILITY



Figuur 41: Rule package car group availability (decision modeler)

Wanneer de klant eligible is, dan worden de rules met betrekking tot de Car Group availability uitgevoerd. Deze houdt rekening met het aantal beschikbare wagens op de gevraagde datum. Wanneer dit voldoende is, is er geen probleem. Slechts wanneer dit niet het geval is, dient de vestiging een oplossing te zoeken door middel van een upgrade, downgrade of door een wagen gereserveerd voor onderhoud terug beschikbaar te stellen.

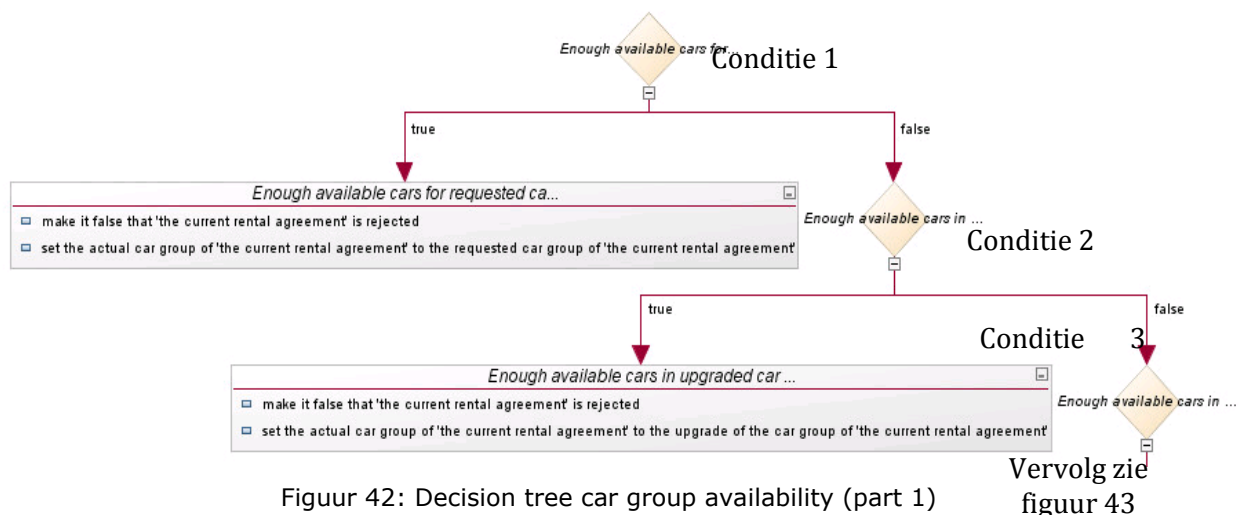
Action Rule 1: Bepalen beschikbaarheid op basis van het aantal beschikbare wagens voor de gevraagde Car Group

```
if
the pickup branch of 'the current rental agreement'
.getAvailableCarNumber ( the requested car group of 'the current rental agreement' ,
the pickup date of 'the current rental agreement' ) is at least 1
then
in 'the current session' , display the message : "Enough cars of the requested group available";
else reject 'the current rental agreement' ;
in 'the current session' , display the message : "Not enough cars of the requested group available";
```

Wanneer er nog minstens 1 wagen beschikbaar is op de gevraagde datum, wordt er geen probleem gemeld. Indien er geen wagen meer vrij is, dan dient er een oplossing gezocht te worden.

Decision tree 1: Bepalen beschikbaarheid gevraagde wagen of toestaan upgrade, downgraded, car reserved for maintenance

Voor deze mogelijke oplossingen werd een decision tree opgesteld. Dit houdt in dat wanneer er niet voldoende wagens zijn in de gevraagde groep, er eerst gekeken wordt of er een upgrade of een downgrade mogelijk is. Als ook dit niet mogelijk is, dan wordt er als laatste optie gezocht of er in de Car Group die aangevraagd werd door de klant, nog wagens aan de kant staan voor onderhoud. Deze mogen dat eventueel nog gebruikt worden voor verhuur. Als geen enkele van deze oplossingen mogelijk is, dan wordt de request verworpen.



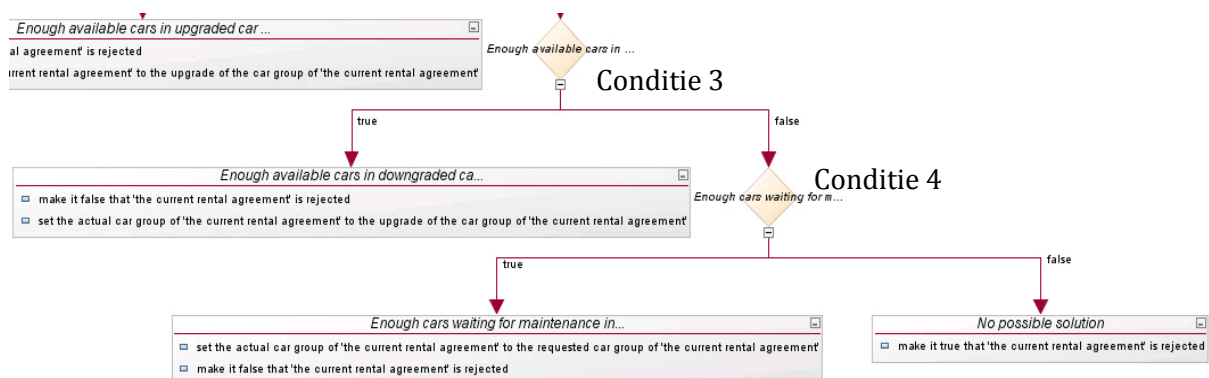
Conditie 1 Enough available cars requested car group:

```
the number of available the requested car group of 'the current rental agreement' on the pickup date of 'the current rental agreement' for the pickup branch of 'the current rental agreement' is at least 1 is <a boolean>
```

Conditie 2 Enough available cars in upgraded car group:

```
the number of available the upgrade of the requested car group of 'the current rental agreement' on the pickup date of 'the current rental agreement' for the pickup branch of 'the current rental agreement' is at least 1 is <a boolean>
```

Bovenstaande condities en actions (zie grijze blokken figuur 42) hebben betrekking op het nagegaan van het aantal beschikbare wagens in de Requested en Upgraded car Group.



Conditie 3 Enough available cars in downgraded car group:

```
the number of available the downgrade of the requested car group of 'the current rental agreement' on the pickup date of 'the current rental agreement' for the pickup branch of 'the current rental agreement' is at least 1 is <a boolean>
```

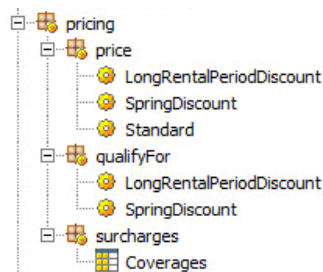
Conditie 4 Enough cars waiting for maintenance in requested car group:

```
the number of available maintenance the requested car group of 'the current rental agreement' on the pickup date of 'the current rental agreement' for the pickup branch of 'the current rental agreement' is at least 1 is <a boolean>
```

Bovenstaande condities en actions hebben betrekking op het nagegaan van het aantal beschikbare wagens in de downgraded Car Group en het aantal wagens klaar voor onderhoud.

Indien er een wagen beschikbaar is voor de klant, dan kan hiervoor de prijs berekend worden.
Indien er geen wagen beschikbaar is op die datum, wordt het proces afgebroken.

4.2.3. PRICING



Figuur 44: Rule package pricing (decision modeler)

Voor iedere wagen is er een standaard dagprijs en weekprijs. Deze prijs kan vermeerderen of verminderen naargelang de klant gebruik maakt van bepaalde acties (de lenteactie of lange periode korting) en hij/zij bepaalde verzekeringen erbij wil.

Action Rule 1: Bepalen standaardprijs beschikbare Car Group

```

definitions
  set 'current group' to the requested car group of 'the current rental agreement' ;

  then
    'the current rental agreement' qualifies for the "standard" offer ;
    set the price of the "standard" offer of 'the current rental agreement'
    to (the number of weeks of 'the current rental agreement' * the weekly rate of 'current group') +
    ( the number of days of 'the current rental agreement' * the daily rate of 'current group' );
    in 'the current session' , display the message :
    | set price of standard offer for the" + 'the current rental agreement' description ;
  
```

Voor de bekomen Car Group wordt de standaardprijs berekend op basis van dag – en weekprijzen.

Action Rule template 1: Bepalen of klant een wagen huurt voor een lange periode

```

definitions
  set group to the requested car group of the current rental agreement
if
  the number of weeks of the current rental agreement [±] is at least ▼ 1 [±]
  and group is one of { MidSize , FullSize , Luxury , SportUtility , Minivan , Compact , Economy [±] }
  ↴
then
  the current rental agreement qualifies for the long term offer
  in the current session , display the message : the current rental agreement description + ▼ qualifies for the long term offer [±]
  
```

In ODM kan de ontwikkelaar de action rules business-vriendelijk maken door er templates van te maken. Zo werd er hier voor EU-Rent een template gemaakt waarbij de gebruiker kan specificeren vanaf wanneer een rental als lange termijn wordt beschouwd, hier is dit 1 week. Daarna kan bepaald worden voor welke Car groups dit van toepassing is. Een template maakt het makkelijk voor zowel de ontwikkelaar als de business user. Dit omdat de gebruiker enkel hoeft te denken over specifieke variabelen en de ontwikkelaar er zeker van is dat de gebruiker enkel de toegelaten waarden kan veranderen en niet de logica van de rule kan verstoren.

Action Rule 2: Bepalen korting indien klant voor lange termijn huurt

```
definitions
  set 'the standard offer' to the "standard" offer of 'the current rental agreement' ;
  set 'the long term offer' to the "long term" offer of 'the current rental agreement' ;
  set 'rebate' to 100;

  then
    set the price of 'the long term offer' to the price of 'the standard offer' - rebate ;
    set the discount of 'the long term offer' to rebate / the price of 'the standard offer'
    - the price of 'the long term offer' * 100;

  in 'the current session' , display the message :
  | Reduce the price for the" + 'the current rental agreement' description ;
```

Voor ieder request wordt er nagegaan hoeveel dagen de klant de wagen wenst te huren. Indien dit zeven of meer dagen betreft, dan krijgt de klant 100 euro korting, ongeacht de categorie van de wagen hij huurt. Dit om de klant te danken voor zijn lange termijn huurperiode.

Action Rule template 2: Bepalen of klant een wagen huurt in de kortingsperiode

```
definitions
  set group to the requested car group of the current rental agreement
  if
    the state of the pickup branch of the current rental agreement [≠] is Belgium [≠]
    and the pickup date of the current rental agreement is after ▼ 3/21/2014 5:20:43 PM and before ▼ 6/20/2014 12:00:39 AM
    and group is one of { MidSize , SportUtility , Minivan , Compact , Economy , FullSize , Luxury [≠] }
    ↴
  then
    the current rental agreement qualifies for the Spring Season offer
    in the current session , display the message : the current rental agreement description + ▼ qualifies for the Spring Season offer [≠]
```

EU-Rent belooft klanten in de lente door hen extra korting te geven. Wanneer dus een rental periode start in de lente en de wagen behoort tot de opgenomen Car Groups, dan is het toegelaten een extra lentekorting te berekenen

Action Rule 2: Bepalen lentekorting voor een bepaald request

```
definitions
  set 'car group' to the requested car group of 'the current rental agreement' ;
  set 'small discount' to 5 %;
  set 'big discount' to 10 %;
  set 'the standard offer' to the "standard" offer of 'the current rental agreement' ;
  set 'the Spring Season offer' to the "Spring Season" offer of 'the current rental agreement' ;

  if 'car group' is one of { Economy , Compact , MidSize }

  then set the discount of 'the Spring Season offer' to 'small discount' ;
    set the price of 'the Spring Season offer' to the price of 'the standard offer' * (1 - 'small discount' );
    in 'the current session' , display the message :
    "Allocating the small discount to the Spring Season offer for" +
    'the current rental agreement' description ;

  else set the discount of 'the Spring Season offer' to 'big discount' ;
    set the price of 'the Spring Season offer' to the price of 'the standard offer' * (1 - 'big discount' );
    in 'the current session' , display the message :
    "Allocating the big discount to the Spring Season offer for" +
    'the current rental agreement' description ;
```

Afhankelijk of het rental agreement in aanmerking komt voor de lentekorting en de bekomen Car Group, wordt er een kleine korting, grote korting of geen extra korting toegekend.

Decision table 1: Bepalen extra kosten voor verzekeringen

Coverages	Car Group	Surcharge	Comment
CDW	Compact,Economy,MidSize	15	Add Collision damage Waiver charge of 15
	FullSize,Luxury,Minivan,SportUtility	25	Add Collision damage Waiver charge of 25
LDW	Compact,Economy,MidSize	21	Add Loss damage Waiver charge of 21
	FullSize,Luxury,Minivan,SportUtility	21	Add Loss damage Waiver charge of 21
PAI		23	Add Personal Accident Insurance charge ...
PEI		2	Add Personal effects Insurance charge of 2
ALI	Compact,Economy,MidSize	11	Add Additional Liability Insurance charge ...
	FullSize,Luxury,Minivan,SportUtility	12	Add Additional Liability Insurance charge ...

Een klant kan er voor opteren enkele verzekeringen af te sluiten die effect hebben bij schade, diefstal, persoonlijke schade, ... Afhankelijk van de opties die de klant kiest, en de Car Group wordt er een extra kost (surcharge) aangerekend. Deze rule table werd niet opgenomen in The Decision Modeler omdat de pricing rule family anders te uitgebreid zou worden. In Operational Decision Manager kan hier makkelijker mee omgesprongen worden en werd het wel opgenomen.

4.2.4. RENTAL AGREEMENT REGISTERING

Wanneer de klant akkoord gaat met de gegeven Car Group en prijs dan kan het Rental Agreement geregistreerd worden. Dit houdt in dat de voorraad van een bepaalde Car Group op bepaalde datums met 1 wagen verminderd wordt.

```
if 'the current rental agreement' is accepted
then the pickup branch of 'the current rental agreement'
.registerPickup ( the actual car group of 'the current rental agreement',
the pickup date of 'the current rental agreement' ) ;

print 'the current rental agreement' description + "is registered";
```

4.2.5. SCENARIO TESTING

Nadat het volledige 'request handling' proces van TDM uitgewerkt was in ODM, was het ook belangrijk om te testen of de gecreëerde regels ook wel werkelijk het gewenste resultaat hadden. Hiervoor werd er een configuration file aangemaakt en werden er vier scenario's als input meegegeven. Hieronder wordt de input weergegeven en daaronder het resultaat van de testen.

Input data voor het testen van scenario's :

```
carrental.ModelFeeder feeder = new carrental.ModelFeeder();

feeder.initCalendar();

carrental.Model currentModel = new carrental.Model();
feeder.populateCarGroups(currentModel);
feeder.populateBranches(currentModel);

carrental.Customer jan = new carrental.Customer("Jan", "Peeters", carrental.DateUtil.makeDate(1970,
java.util.Calendar.MAY, 20), "NotBlacklisted");
```

```

currentModel.addCustomer(jan);

currentModel.addRentalAgreement(new RentalAgreement(
    currentModel.getCustomer("Jan", "Peeters"),
    currentModel.getBranch("Hasselt"), DateUtil.addDays(Session.getStartDate(), 12),
    currentModel.getBranch("Hasselt"), DateUtil.addDays(Session.getStartDate(), 20),
    CarGroup.MidSize, new String[] { "CDW", "LDW", "PEI", "PAI" }));

currentModel.addCustomer(new Customer("Kees", "Wiensma", DateUtil
    .makeDate(1974, Calendar.AUGUST, 19), "NotBlacklisted"));

    Calendar cal = new GregorianCalendar();
    int currentYear = cal.get(Calendar.YEAR);

    currentModel.addCustomer(new Customer("Greet", "Vangeel", DateUtil
        .makeDate(currentYear-20, Calendar.SEPTEMBER, 12), "NotBlacklisted"));

    currentModel.addCustomer(new Customer("Aagje", "Hoekstra", DateUtil
        .makeDate(1974, Calendar.JULY, 14), "Blacklisted"));

currentModel.addRentalAgreement(new RentalAgreement(
    currentModel.getCustomer("Kees", "Wiensma"),
    currentModel.getBranch("Amsterdam"), DateUtil.addDays(Session.getStartDate(), 12),
    currentModel.getBranch("Amsterdam"), DateUtil.addDays(Session.getStartDate(), 20),
    CarGroup.FullSize, new String[] { "CDW", "ALI", "PEI" }));

    currentModel.addRentalAgreement(new RentalAgreement(
        currentModel.getCustomer("Greet", "Vangeel"),
        currentModel.getBranch("Antwerpen"), DateUtil.addDays(Session.getStartDate(), 1),
        currentModel.getBranch("Antwerpen"), DateUtil.addDays(Session.getStartDate(), 5),
        CarGroup.FullSize, new String[] { "CDW", "PEI" }));
        CarGroup.FullSize, new String[] { "CDW", "PEI" }));

    currentModel.addRentalAgreement(new RentalAgreement(
        currentModel.getCustomer("Aagje", "Hoekstra"),
        currentModel.getBranch("Utrecht"), DateUtil.addDays(Session.getStartDate(), 1),
        currentModel.getBranch("Utrecht"), DateUtil.addDays(Session.getStartDate(), 5),
        CarGroup.SportUtility, new String[] { "CDW", "LDW" }));

note("##### RESERVATIONS #####");

RentalAgreement[] rentals = currentModel.rentalAgreements;

for (int r=0; r<rentals.length; r++) {
    carreservation(rentals[r]);
}

```

Dit zijn vier fictieve personen die een bepaalde wagen willen huren voor een bepaalde termijn.

Input die per persoon ingevoerd wordt: Voornaam, Naam, geboortedatum, aanwezigheid op blacklist, pick-up vestiging, pick-up datum, return vestiging, return datum, gewenste Car Group en de gekozen extra verzekeringen.

Resultaat scenario testing:

```
##### RESERVATIONS #####

New Request

Eligibility task : Initial action for Rental Agreement for Jan Peeters (May 20, 1970)
Eligibility task : Final action for Rental Agreement for Jan Peeters (May 20, 1970)
-----
Availability task :
Enough cars of the requested group available
-----
Pricing task : Initial action for Rental Agreement for Jan Peeters (May 20, 1970)
  set price of standard offer for the Rental Agreement for Jan Peeters (May 20, 1970)
Add Personal effects Insurance charge of 2
Add Personal Accident Insurance charge of 3
Add Loss damage Waiver charge of 21
Add Collision damage Waiver charge of 15
  Rental Agreement for Jan Peeters (May 20, 1970) qualifies for the long term offer
  Reduce the price for the Rental Agreement for Jan Peeters (May 20, 1970)
  Rental Agreement for Jan Peeters (May 20, 1970) qualifies for the Spring Season offer
Allocating the small discount to the Spring Season offer for Rental Agreement for Jan Peeters (May 20, 1970)
  * Offer: standard price: 928.00
  * Offer: Spring Season price: 881.60 (5.00 % discount)
  * Offer: long term price: 828.00
  # Best Offer: long term price: 828.00
Pricing task : Final action for Rental Agreement for Jan Peeters (May 20, 1970)
-----
Registering task
  Rental Agreement for Jan Peeters (May 20, 1970) is registered
-----

New Request

Eligibility task : Initial action for Rental Agreement for Kees Wiensma (Aug 19, 1974)
Eligibility task : Final action for Rental Agreement for Kees Wiensma (Aug 19, 1974)
-----
Availability task :
Enough cars of the requested group available
-----
Pricing task : Initial action for Rental Agreement for Kees Wiensma (Aug 19, 1974)
  set price of standard offer for the Rental Agreement for Kees Wiensma (Aug 19, 1974)
Add Additional Liability Insurance charge of 12
Add Personal effects Insurance charge of 2
Add Collision damage Waiver charge of 25
  Rental Agreement for Kees Wiensma (Aug 19, 1974) qualifies for the long term offer
  Reduce the price for the Rental Agreement for Kees Wiensma (Aug 19, 1974)
    * Offer: standard price: 802.00
    * Offer: long term price: 702.00
    # Best Offer: long term price: 702.00
Pricing task : Final action for Rental Agreement for Kees Wiensma (Aug 19, 1974)
-----
Registering task
  Rental Agreement for Kees Wiensma (Aug 19, 1974) is registered
-----

New Request

Eligibility task : Initial action for Rental Agreement for Greet Vangeel (Sep 12, 1994)
This customer is younger than 21
Eligibility task : Final action for Rental Agreement for Greet Vangeel (Sep 12, 1994)
-----

New Request

Eligibility task : Initial action for Rental Agreement for Aagje Hoekstra (Jul 14, 1974)
  Rental Agreement for Aagje Hoekstra (Jul 14, 1974) is rejected. Customer cannot be Blacklisted
Eligibility task : Final action for Rental Agreement for Aagje Hoekstra (Jul 14, 1974)
```

Voor iedere persoon wordt de eligibility nagegaan, indien dit niet in orde is, dan stopt de rule engine en wordt de volgende persoon onderzocht. Indien de klant wel eligible is, wordt er gekeken of er nog voldoende wagens beschikbaar zijn in de gekozen groep. Wanneer dit het geval is, dan

wordt de geschatte prijs berekend en kiest het systeem de prijs die het voordeligst is voor de klant. Dit kan ofwel de standaardprijs, de prijs met lentekorting of de 'long term' prijs zijn.

HOOFDSTUK 5: DE WAARDE EN MATURITEIT VAN THE DECISION MODELER & OPERATIONAL DECISION MANAGER

Deze masterproef omvat het onderzoek naar de waarde van The Decision Modeler en Operational Decision Manager. In de hoofdstukken twee en drie werd voornamelijk de theoretische waarde van decision modeling via een literatuurstudie onderzocht. In hoofdstuk vier werd voornamelijk de praktische waarde van de decision modeling tools via de praktijktoepassing rond het fictieve autoverhuurbedrijf EU-Rent getoond. Hoofdstuk 5 bouwt hierop verder en probeert voor TDM en ODM een algemeen beeld van de waarde en maturiteit te schetsen.

5.1. WAARDE VAN THE DECISION MODELER (ARCHITECT) BY BIZZDESIGN

Zoals eerder aangehaald, bestaan er nog geen specifieke frameworks om de waarde van business decision modeling tools te bepalen. Daarom werd er geopteerd om het evaluatieframework van BPMNtools.com te gebruiken. Dit framework wordt gebruikt om business process modeling tools te evalueren. Daarom is er voor het beoordelen van TDM een selectie gemaakt over welke criteria op deze tool van toepassing kunnen zijn.

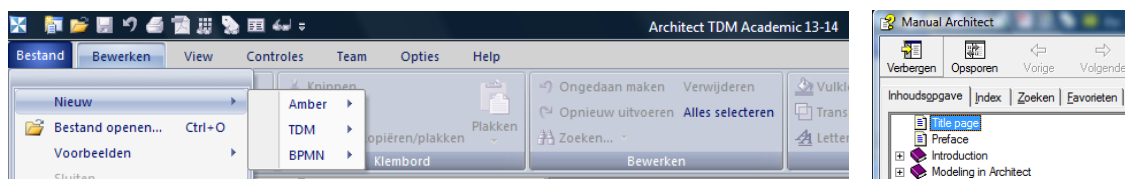
The decision modeler werd onderworpen aan de volgende criteria:
(<http://www.bpmntools.com/article/bpmn-tool-evaluation-criteria/>)

- ✚ **Usability:** Een beoordeling van de user interface, ease-to-use, responstijd van de tool, interactieve hulp en documentatie
- ✚ **Functional features:**
 - **Modeling features:** het creëren van modellen, diagrammen, relaties. Het aanpassen van elementen en relaties, definiëren business rules, relatietabellen. Model management, ondersteunende functies, inspanning met betrekking tot installatie en administratie
 - **Simulation/execution features:** het simuleren en uitvoeren van de business proces flows, en business rules.
- ✚ **Standards compliance/Interoperability:** Het gebruik van standaarden voor het modeleren en de integratie binnen en buiten de tool.
- ✚ **Support/Team modeling:** Installatie, updates, technische support. De mogelijkheid tot het samenwerken op projecten met betrekking tot het bijhouden van de historie en user permissions.
- ✚ **Financial value:** Vergelijking uitgebreidheid features ten opzichte van de kostprijs.

5.1.1. USABILITY

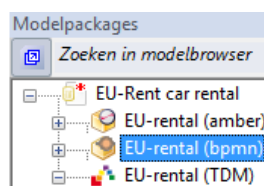
The Decision Modeler (TDM) vormt één module uit de overkoepelende tool Architect. Dit maakt dat de module niet losstaat van andere modellen zoals Business Process Modeling (BPM) en Amber, maar automatisch geïntegreerd wordt in het geheel.

De lay-out van TDM is intuïtief opgebouwd. Het neemt de interface aan van de welbekende Officepakketten. Hierdoor kan de gebruiker redelijk vlot met de tool aan de slag zonder dat er werkelijk technische ervaring vereist is. De tool wijst zichzelf uit met behulp van het bijbehorende helpdocument. Indien bepaalde knoppen of functies niet duidelijk zijn, zijn ze meestal snel op te zoeken via de help-knop. Een minder punt dan wel is de beperktheid aan hulp buiten dit document. Daar het programma nog redelijk nieuw is, zijn er op internet weinig of geen extra tutorials te vinden. Ook het aantal voorbeelden binnen het document blijft eerder beperkt.



Figuur 45: Interface TDM, Architect (links) | Helpdocument Architect (rechts)

Ondanks de integratie, blijft de scheiding tussen beslissingen, processen, ... toch duidelijk zichtbaar door het gebruik van de verschillende packages. Dit benadrukt de onafhankelijkheid die belangrijk is binnen dit domein. De integratie tussen de packages verloopt vlot. Er kan snel geschakeld worden van het procesmodel naar het decision model via de gecreëerde link. De responstijd van het systeem is daarbij zeer kort. Enkel wanneer er een redelijk uitgebreide simulatie uitgevoerd werd, kon de responstijd toch wel een langere tijdspanne in beslag nemen. Maar hierbij moet de kanttekening gemaakt worden, dat dit ook afhankelijk kan zijn van de gebruikte laptop en niet zozeer de tool.

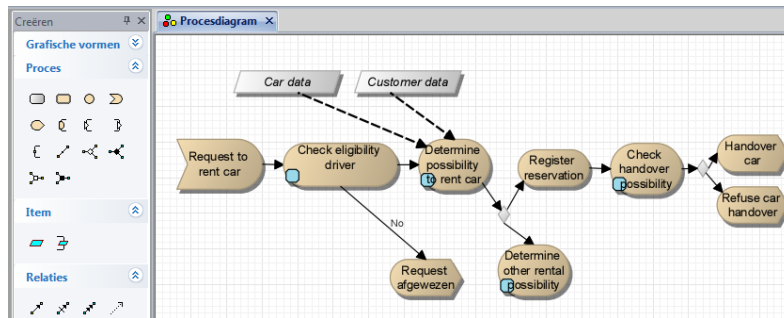


Figuur 46: Modelpackages binnen Architect

Een negatief punt met betrekking tot de integratie is de bruikbaarheid ervan. Wanneer namelijk een procesmodel en de bijbehorende beslissingsmodellen worden gesimuleerd, wordt er geen rekening gehouden met de uitkomst van deze beslissingen in het procesmodel. In realiteit is dit wel belangrijk. Afhankelijk van een bepaalde beslissing wordt er meestal binnen een procesmodel een bepaald pad gevolgd. Dit is ondanks de integratie in Architect nog niet mogelijk. Wel werd er meegedeeld dat dit in een volgende versie zou verwerkt zijn.

5.1.2. FUNCTIONAL FEATURES

MODELING FEATURES



Figuur 47: Drag & Drop voor het opstellen van diagrammen en modellen

Bij het creëren van de decision modellen worden de elementen op het scherm gedragd en gedropt (dit geldt ook voor procesmodellen) en dient de gebruiker links te creëren tussen de overkoepelende bedrijfsbeslissing (octagon) en de onderliggende rule families. Voor een bepaalde rule family, kan de gebruiker kiezen welke condition en action facts in de kolommen opgenomen worden. Wanneer extra kolommen (nieuwe facts) toegevoegd worden, dan kan de gebruiker ofwel bestaande facts kiezen of automatisch nieuwe aanmaken waarbij de facts dadelijk aan de glossary worden toegevoegd. Dit wijst nogmaals op het sterk geïntegreerde geheel.

Zoals hierboven vermeld werd, is het aanmaken van nieuwe facts makkelijk. Het aanpassen en verwijderen van facts is wat moeilijker. Bij het veranderen van een fact dat eerst een ondersteunende rule family had, naar een fact zonder ondersteunende rule family wordt er een probleem opgemerkt. Dit kan namelijk niet rechtstreeks. Zo zal de gebruiker eerst dit fact moeten verwijderen om er daarna een nieuw te creëren dat niet gelinkt is aan een ondersteunende rule family. Ook bij het verwijderen van een fact uit een rule family, mag de gebruiker niet vergeten dit te verwijderen uit de glossary, anders blijft dit daar bestaan.

Om er voor te zorgen dat alle ongebruikte elementen verwijderd worden, kan de gebruiker deze optie uitvoeren. Dit maakt het makkelijk om ongebruikte decisions of rule families te verwijderen, jammer genoeg kunnen de ongebruikte facts hier niet mee verwijderd worden.

De tool kan overweg met verschillende fact types, zowel gewone (boolean, integer, string) als list fact types zoals een bepaalde opsomming. Fact Values kunnen dus bestaan uit zowel waarden als formules.

Overdue consequence	Hourly fee, whole day fee,
Car group	A, B, C

Figuur 48: List fact types

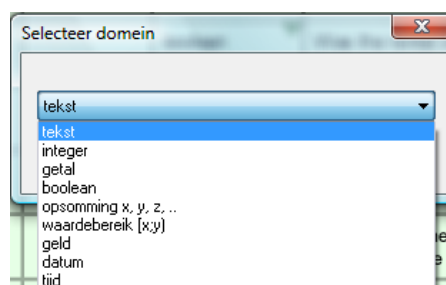
De glossary:

Het fact type overzicht, de glossary, is goed geïntegreerd met de rule families zoals eerder aangehaald. Het ondersteunt verschillende domeintypes en linken met datamodellen. De glossary vormt een zeer belangrijk onderdeel van de tool. Belangrijk is dat alle gebruikers precies weten wat de verschillende termen inhouden zodat ze ook de juiste rule families kunnen maken.

Glossary					
Feittype	Huidige waarde	Domein	Beschrijving	Entiteittype	Entiteitattribuut
Car readiness		boolean	Checked by agent if car has been cleaned, has full	Car	Car readiness
Number car rentals		integer	The number of rentals a customer did with our	Customer	Number of car rentals
<u>Overdue consequence</u>		Hourly fee, whole day fee,	What is the punishment for a late return	Car rental	Overdue consequence
Car group Class		A, B, C	Group of cars, with A the lowest and C the highest	Car group	Car group class
Free rental		boolean	Was the rental done by a free rental coupon	Car rental	Free rental
<u>Loyalty incentive</u>		boolean	Does the customer have a loyalty incentive scheme	Customer	Loyalty incentive scheme

Figuur 49: Voorbeeld van een glossary

In de glossary worden attributen opgenomen die geassocieerd zijn met ieder fact type zoals het domein, beschrijving, de relatie met een entiteit en attribuut uit het datamodel. Het domein kan verschillende vormen aannemen zoals tekst, integer, reële getallen, een opsomming en een range.



Figuur 50: Domeintypes

Principe validatie:

Een andere belangrijke modeling feature is de validatie van het Decision model en de bijbehorende rule families. Het Decision Model is gebaseerd op 15 principes en 3 normaalvormen. Het is dan ook zeer belangrijk dat een tool die hierop gebouwd werd, rekening houdt met deze principes. Zo zal TDM waarschuwen voor verschillende situaties waarbij de principes verbroken worden. Zo waarschuwt TDM voor overlappende rules met een zelfde input, maar verschillende conclusies. Ook wanneer niet elke regel een conclusie heeft, zal dit een melding opleveren. Wanneer er circulaire relaties bestaan waarbij een decision rule family zowel direct als indirect afhankelijk is van een zelfde rule family, krijgt dit ook een melding omdat dit volgens de principes niet is toegelaten.

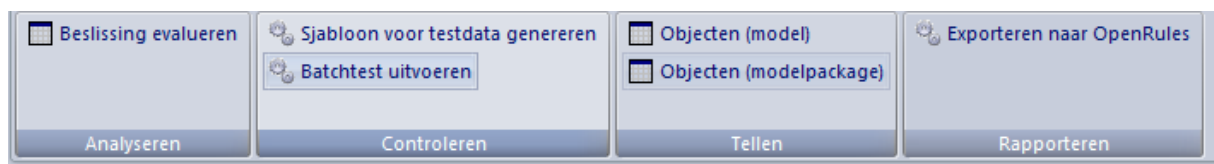
TDM verplicht correcte fact type domeinwaarden in te geven. Zo zal het een foutmelding geven wanneer de ingevoerde waarden geen deel uitmaken van het domein. Toch kwam het tijdens de praktijktoepassing meermaals voor dat er een foutmelding gegeven werd bij het uitvoeren van de

scenario analyse. Hierbij ontdekte de tool pas op het laatste moment dat er een fout in het domein was.

Er kan dus geconcludeerd worden met betrekking tot de validatie dat TDM overweg kan met alle 15 principes en ze ook grondig nagaat. Het minpuntje is de waarschuwing die verkregen wordt, met name het tijdstip waarop de waarschuwing weergegeven wordt.

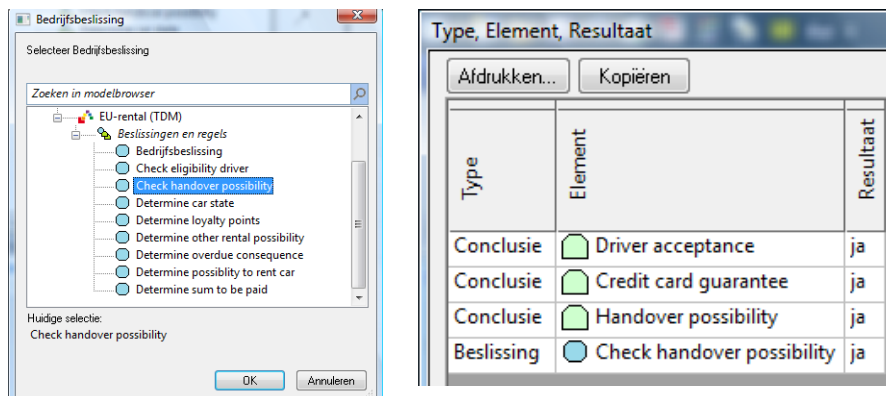
SIMULATION/EXECUTION FEATURES

Eenmaal het model op punt staat, kan de gebruiker binnen TDM iedere beslissing apart evalueren of opteren voor een volledige simulatie. De executie zal niet binnen TDM zelf kunnen gebeuren, daarvoor is er een Business Rules Management Systeem nodig. In de huidige versie kan er enkel geëxporteerd worden naar de OpenRules executieomgeving. In volgende versies zullen de exportmogelijkheden uitgebreider worden.



Figuur 51: Stappen binnen TDM

De eerste stap is het evalueren van de aparte beslissingen. Dit valt onder de noemer analyseren.



Figuur 52: Evalueren individuele bedrijfsbeslissing

Per bedrijfsbeslissing kan de gebruiker de waarden van de onafhankelijke fact types ingeven en zien wat het resultaat is. Afhankelijk van de correctheid van het model, heeft de beslissing het (on)gewenste resultaat. Als de gebruiker hier een fout op merkt, weet hij/zij dat de fout in dit beslissingsmodel zit. Dit is een stuk makkelijker dan wanneer de tool enkel toeliet het hele project als geheel te evalueren.

Indien alle beslissingen geëvalueerd zijn, kan de gebruiker de tool een Excel file (zie figuur 53) laten creëren. Deze Excel file kan via Microsoft Excel ingevuld worden. De gebruiker is hierbij verplicht de simple fact types in te vullen. Deze gebruikt de tool om later een batchtest te kunnen uitvoeren. Bij de expected derived fact types, kan de gebruiker invullen welke waarden hij/zij verwacht. De laatste groep, de actual derived fact types, wordt door de tool zelf aangevuld. Deze feature geeft de tool extra waarde omdat dit toelaat het gehele project te simuleren in enkele

klikken. Zo is het makkelijk een model op te zetten, te analyseren en te testen met behulp van een Excel file om te weten of de verschillende decision models voor een bepaald proces werken.

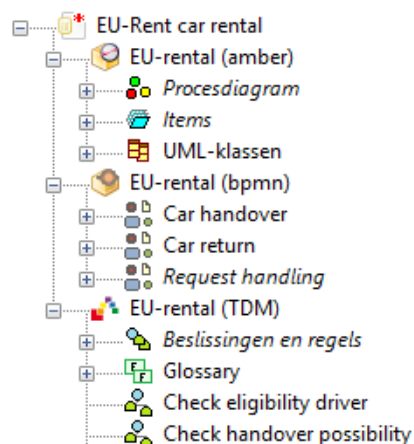
Simple fact types	Driver has good condition	boolean							
	Drivers age	integer							
	extension call	boolean							
	Free rental	boolean							
	Number car rentals	integer							
Derived fact types: expected values (optional)	Driver acceptance	boolean							
	Eligibility driver	boolean							
	Handover possibility	boolean							
	Loyalty incentive scheme	boolean							
	Loyalty points	integer							
	Other rental possibility	Reserved car, upgrade group, downgrade group, returned car, car for service, not needed							
	Overdue consequence	Hourly fee, whole day fee, call customer, call police, not needed							
Derived fact types: actual derived values	Driver acceptance	boolean							
	Eligibility driver	boolean							
	Handover possibility	boolean							
	Loyalty incentive scheme	boolean							
	Loyalty points	integer							
	Other rental possibility	Reserved car, upgrade group, downgrade group, returned car, car for service, not needed							
	Overdue consequence	Hourly fee, whole day fee, call customer, call police, not needed							

Figuur 53: Sjabloon voor testdata binnen excel (batchtest)

5.1.3. STANDARDS COMPLIANCE/INTEROPERABILITY

De overkoepelende Architect tool werkt met de bekende standaarden voor de verschillende modelleermogelijkheden. Zo biedt Architect interoperabiliteit met Business Model Canvas®, ArchiMate®, TOGAF®, BPMN™ (process modeling), TDM (business logic modeling), OIAm (infrastructure methode), UML, ERD en Lean. Het gebruiken van deze standaarden verhoogt de bruikbaarheid en begrijpbaarheid van de tool.

Veel van deze standaarden worden binnen de tool met elkaar geïntegreerd. Zo kunnen de Decision points van het procesmodel gelinkt worden met business decisions, en de glossary termen kunnen verbonden worden met het logische datamodel dat voorgesteld wordt door het UML Class diagram.



Figuur 54: Overzicht modellen binnen architect

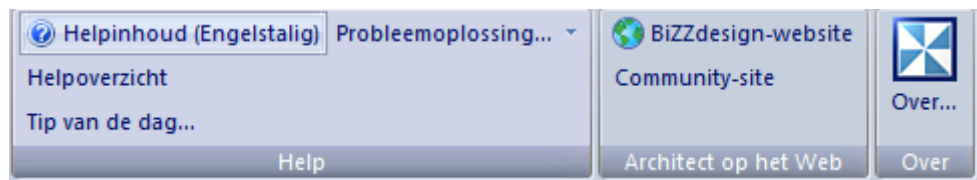
Naast het gebruik van standaarden is het ook mogelijk andere tools met Architect te linken.

Input: XML en Visio XML bestanden kunnen als input voor Architect gebruikt worden.

Output: Rapporten over gemodelleerde processen in Architect kunnen geëxporteerd worden naar Word, Excel en Powerpoint. De business rules in The Decision Modeler kunnen geëxporteerd worden naar OpenRules voor de executie ervan.

5.1.4. SUPPORT/TEAM MODELING

De ondersteuning van de tool bestaat uit een helpdocument waarin de gebruiker de functionaliteiten van de tool kan opzoeken. Vragen die niet opgelost kunnen worden met behulp van het helpdocument, kunnen gesteld worden op de community site. Daarnaast wordt er door Bizdesign ook de mogelijkheid geboden om enkele dagen les te volgen om zo de tool beter onder de knie te krijgen. Dankzij deze support stijgt de waarde van de tool.



Figuur 55: Support binnen Architect

Voor het aspect team modeling, bestaat er de tab 'Team Revisies'. Deze tab kon in deze thesis niet gebruikt worden, daar hiervoor nog een extra licentie nodig is. Maar met behulp van het helpdocument kon dit toch kort onderzocht worden.



Figuur 56: Team tabblad binnen architect

Om van de team functies gebruik te kunnen maken, dient een model package opgeslagen te worden als een 'Shared model package'. Het omvat dezelfde informatie als een normaal package, maar het omvat nog extra multi-user en revisie informatie. Dankzij het Shared package, wordt het mogelijk voor verschillende gebruikers tegelijk aan het model te werken. Elke verandering wordt dan opgeslagen in een model package revision. Voor elke revision wordt er bijgehouden, wanneer en wie de verandering deed.

5.1.5. FINANCIAL VALUE

De financiële waarde van de plug-in The Decision Modeler en de overkoepelende tool Architect is op dit moment moeilijk te bepalen. Omdat deze decision methodologie nog redelijk nieuw is, zijn er weinig of geen andere spelers op de markt die een gelijkaardig geheel aanbieden. Dit maakt het

moeilijk in te schatten of het werkelijk die prijs kan hebben, omdat de vraag wel groot is, maar het aanbod nog klein en dus de prijs nog niet in evenwicht is.

Ondanks deze beperking, kan gezegd worden dat het bedrijf de kost van de tool zeker kan terugverdienen bij een correct gebruik ervan. De betaling gebeurt per gebruiker en kost enkele duizenden euro's voor een privébedrijf. Wanneer deze gebruikers wensen samen te werken, dient hiervoor een extra licentie per gebruiker te worden aangeschaft.

Architect (+The Decision Model) zou voor de kleine en middelgrote ondernemingen dus waarde kunnen bieden omdat de prijzen nog betaalbaar zijn ten opzichte van andere bekendere, grotere tools en omgevingen.

5.1.6. ALGEMENE BEOORDELING

Daar de tool nog redelijk jong is, is de tool functioneel en gebruiksvriendelijk. Ook de link met het Decision model komt zeer sterk naar voor, wat toch zeker wel een pluspunt is.

De user interface is zeer intuïtief en de navigatie doorheen de tool is gebruiksvriendelijk. De verschillende modelleeraspecten zijn in de tool aanwezig en de validatie houdt rekening met de verschillende principes. De support tot nu toe is redelijk beperkt, daar de tool nog jong is. Hetgeen het meest gemist werd, was het nut van de integratie van de procesmodellen en de decision modellen. Zoals gezegd, bestaat de link wel, maar worden toch alle beslissingen getest ondanks dit niet meer nodig zou zijn volgens het proces.

5.2. WAARDE VAN OPERATIONAL DECISION MANAGER BY IBM

De markt van de Business rules management systemen (BRMS) heeft een steeds grotere invloed op de samenhang van IT en de business. De BRMS technologie moet bedrijven helpen de transitie van proces-centrische naar decision-centrische applicatieontwikkeling te vergemakkelijken. (Craggs, 2012)

Business rules worden steeds belangrijker door de verhoging van het aantal beslissingen dat er genomen dient te worden. Doordat beslissingen steeds nauwkeuriger genomen kunnen worden, gescheiden van de acties, kunnen ze steeds beter identificeren welke acties best genomen worden. Wanneer een bedrijf probeert business rules op te stellen en deze te linken aan de nodige acties, zorgt dit ervoor dat er ook een grondiger begrip is van de business. Dit op zich leidt dan tot een betere en consistentere applicatieontwikkeling. Organisaties die BRMS technologie gebruiken, kunnen op die manier een competitief voordeel halen uit hun betere afstemming van IT op de business.

Om de waarde van ODM te onderzoeken, werd verkozen het onderzoek van Lustratus rond de waarde van het BRMS te gebruiken. Dit omdat er rond dit domein nog weinig onderzoek verricht werd, maar Lustratus een onderzoek deed om de bestaande BRMS met elkaar te kunnen vergelijken.

5.2.1. TECHNICAL ASSESSMENT

Volgens Lustratus is de waarde van een BRMS waar Operational Decision Manger (ODM) onder valt, afhankelijk van drie categorieën, met name functionaliteit, eigenschappen en integratie:

Functionality	Characteristics	Solution Extensions
• Rules development • Events specification • Collaborative design • Save/Browse/Load of decision components • Test and simulation • Deployment • Integration with other components • Operational monitoring and feedback	• Usability • Range of supported inputs • Time-to-value • Import / export • Samples • Scalability • Performance • Security • Versioning • Dynamic deployments • Governance	• Business Analytics • BPM support • Ruleset templates • Professional services • Partner ecosystem • Collaboration community

Figuur 57: Categorieën waarde BRMS

Uit [Overgenomen uit] competitive review of operational decision management (p. 7), door S. Craggs, 2012, lustratus research.

Functionaliteiten:

Zoals eerder vermeld werd, is IBM opgesplitst in twee grote delen: de Decision Server (de design en runtime tool voor het creëren en uitvoeren van ODM projecten) en het Decision Center dat meer gericht is op de businessgebruiker. Het decision center bevat een business console die gebruikt kan worden om rulesets en eventsets te creëren, aan te passen of te beheren. Deze heeft een web 2.0 uiterlijk en voelt dus herkenbaar aan voor een businessgebruiker.

De designer tool die in voorgaande hoofdstukken grondig besproken werd, maakt deel uit van de IBM Decision Server. Met behulp van deze eclips gebaseerde tool legt de ontwikkelaar klassen vast die in de rules gebruikt worden. Binnen het project kan er reeds een rule flow vastgelegd worden via een drag en drop stijl. Wanneer het project binnen de designer tool voltooid is, kan dit beschikbaar gemaakt worden in het Business Center dat dus niet-technisch en webgebaseerd is. Op die manier heeft iedere gebruiker zijn omgeving wat de ease-of-use verhoogt. Alle rules en events worden opgeslagen in hetzelfde Decision Center Repository wat het delen en bewerken makkelijker maakt.

ODM biedt voor het Decision Center ook een add-on, IBM rule solutions for office. Dit laat toe regels en events te creëren en aan te passen via microsoft office pakketten zoals Word en Excel. Rulesets worden hiervoor in Ruledocs file format opgeslagen zodat action rules in Word en decision tables in Excel kunnen worden aangepast. De office omgeving is meestal zeer bekend bij de

meeste businessmensen wat het creëren van rules voor hen makkelijker maakt. Daar Microsoft Office niet beschikbaar was op de ODM server die ter beschikking werd gesteld voor deze masterproef, kon deze functionaliteit hier niet gedetailleerder besproken worden.

De business console werd ontwikkeld voor business users. Het is meer toegankelijk en gebruiksvriendelijker dan de Rule Designer voor de niet-technische gebruiker. In plaats van de hiërarchisch opgebouwde Explorer in de Rule designer, wordt de gebruiker in de business console de mogelijkheid geboden om rules te zoeken via keywords. Daarnaast krijgt de gebruiker een overzicht van de pas veranderde rules. De business user kan in deze omgeving zelf snel business rules creëren en aanpassen. Indien hij/zij samenwerkt met andere business users, kunnen zij met elkaar communiceren via het stream-bord.

Ook testen en simulatie van regels kunnen door de ontwikkelaar in ODM uitgevoerd worden. Op deze manier kan de ontwikkelaar nagaan wat een bepaalde regel als resultaat heeft wanneer deze wordt toegepast op historische of realistische data in vergelijking met voor de aanpassing van de regel. Op die manier hebben businessgebruikers een snelle validatie van hun veranderingen.

Metingen en feedback worden verkregen op twee niveaus. Aan de ene kant heeft de gebruiker de Decision Warehouse tool dat gebruikt kan worden voor gedetailleerde queries uit te voeren met betrekking tot de uitvoering van de regels. Het omvat welke regels aangeroepen en uitgevoerd werden, welke versienummer de regel had en wat het resultaat was. De andere tool 'Business monitor' geeft eerder hogere niveau feedback via Executive dashboards en business metrics.

Eigenschappen:

De business console maakt het mogelijk voor business users om op een niet-technische manier aan decision management te doen. Het zorgt voor de empowerment van de businessgebruikers. Zij kunnen zelf hierdoor meer specificeren wat business rules betreft. Dit is wat vaak een probleem was, omdat regels werden gecreëerd door ontwikkelaars die niet zozeer de precieze context van de rule begrepen. Doordat het makkelijker is samen te werken, zullen users ook sneller leren van elkaar en zullen gebruikers elkaar kunnen aanvullen. Ook de eerder besproken natuurlijke (if-then) taal en de ruledocs zorgen ervoor dat de gebruiker steeds meer de controle zelf in handen kan nemen. Op die manier is de Time-to-Value korter dan wanneer rules opgemaakt werden door ontwikkelaars, nagekeken door businessgebruikers, terug naar ontwikkelaars gestuurd voor verbetering en zo eventueel nog een paar keer heen en weer.

Ook het gebruiken van rollen die een gebruiker toegewezen krijgt, maakt het makkelijker gebruikers autonomie te geven binnen hun domein terwijl de controle toch bewaard blijft.

Integratie:

IBM Business Process manager omvat business process management en biedt de mogelijkheid om ODM artifacten te gebruiken in de business process flows. Ook SPSS kan met ODM geïntegreerd worden. Op deze manier kan een predictive analysis gebruikt worden bij het nemen van beslissingen.

Het aantal ODM samples en templates is nog beperkt, maar stijgt geleidelijk aan dankzij de IBM Blueworks live community. Ondanks deze beperking, kunnen bedrijven toch rekening op hulp bij de implementatie van ODM. Zo zijn er voldoende professionele service teams en consultancybedrijven die hulp kunnen bieden.

Functionaliteit	Eigenschappen	Integratie
Decision center bevat rol gebaseerde rules en event ontwikkeling	Business Console is begrijpbaar en hanteerbaar voor business users. Ze kunnen microsoft office als input gebruiken en samenwerken in eenzelfde repository	Integratie met IBM Business Process manager om beslissingen in processen te integreren.
Business users kunnen natuurlijk IF/THEN taal, beslissingstabellen en –bomen gebruiken voor het bouwen en aanpassen van rules en events	Rules en events worden op eenzelfde manier bewerkt waardoor time-to-value verbetert.	Integratie met SPSS voor decision analytics
De business console heeft een web 2.0. look herkenbaar voor business users.	De onderliggende IBM Websphere applicatie server engine zorgt voor schaalbaarheid en beveiliging.	Veel erkende professionele services die helpen bij het implementeren van de tool door het partner ecosysteem
Mogelijkheid om te testen en te simuleren		Mogelijkheid om ODM elementen als web services te gebruiken maakt het makkelijker integreerbaar met andere omgevingen en applicaties

Tabel 3: Waarde van Operational decision manager

5.2.2. COMMERCIAL ASSESSMENT:

Deze assessment is gebaseerd op vier criteria die het onderzoek van Lustratus aanbood:

- ✚ **Time-to-value:** Hoe lang duurt een ODM project alvorens het functioneel is. Vanaf wanneer zijn de voordelen zichtbaar ?
- ✚ **TCO efficiency:** Wat zal de Total cost of ownership zijn van het BRMS ? Hoeveel tijd en geld kan het kosten om ODM projecten te creëren, aan te passen en te onderhouden ?
- ✚ **Risk mitigation:** Kunnen vooropgestelde service levels gerespecteerd worden ? Zullen projecten voldoen aan de vooropgestelde businessdoelen ?
- ✚ **Value potential:** Wat kan allemaal gedaan worden met het ODM platform ? Hoe breed is de range van scenario's die door de tool verwerkt kunnen worden ? Bestaat er voldoende integratie ?

De key bij **Time-to-value** is het betrekken van business users om zo waarde te creëren. Binnen ODM kunnen business users via een herkenbare interface zelf rules opstellen, aanpassen en controleren. Technische ondersteuning is nodig bij ODM van IBM, maar de werkelijke Time-to-Value wordt gemeten aan de hand van het gebruiksgemak voor business users. Dit omdat vaak daar problemen en tijdverlies ontstaan doordat business users niet weten hoe ze de omgeving precies moeten gebruiken wanneer dit te complex is. Hierin biedt IBM zowel de technische tools voor decision ontwikkeling als de meer gebruiksvriendelijke interface voor business users.

TCO efficiency focust voornamelijk op training en skill kosten, het gemak van onderhoud en updates binnen ODM en het bestaan van templates om tijd en kosten te reduceren. Wanneer verschillende interfaces beschikbaar zijn voor de verschillende gebruikers, kunnen de training kosten gedrukt worden. Het onderhoud van de gehele ODM omgeving vraagt dan wel weer het nodige personeel. Daarom dat de ODM omgeving ook meestal enkel gebruikt wordt door grote, kapitaalkrachtige bedrijven.

Risk mitigation omvat het evalueren van risico's. Dit kunnen risico's zijn op operationeel vlak, zoals de verschillen tussen de werkelijke resultaten van het BRMS en de verwachtingen, het halen van de service levels. Rule testing, verification en simulatie maakt het mogelijk risico's te minimaliseren, dit omdat vaak user errors de fout zijn van crashes. Daarom is de ease-of-use hier sterk mee verbonden.

Value potential biedt een overzicht van de gehele waarde bovenop de basis ODM functionaliteit. De waarde van ODM stijgt dankzij de integratie die mogelijk is met de eerder besproken Business Process manager en SPSS. Daarnaast kan ODM ook geïntegreerd worden met de IBM case manager waarbij de case manager ODM kan aanroepen om beslissingen te nemen over relevante gebeurtenissen.

5.3. WAARDE VAN DE COMBINATIE VAN TDM EN ODM

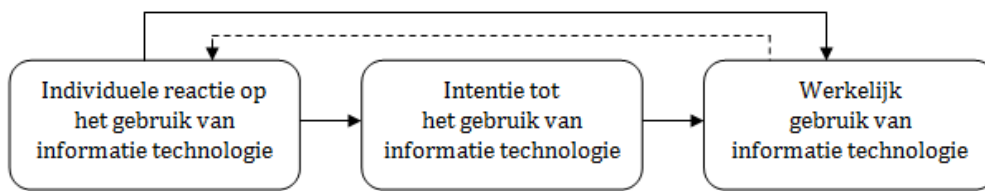
Naast de bespreking van de aparte waarde van TDM en ODM in bovenstaande pagina's is het zeer belangrijk dat niet alleen de aangeboden tools efficiënt werken, ook de gebruiker zelf moet ze willen gebruiken en ook werkelijk gebruiken. Een tool of systeem kan wel geïmplementeerd worden, maar wanneer de gebruikers er geen aandacht aan schenken of het verkeerd gebruiken dan zal de verwachte waarde zeker niet gerealiseerd worden.

Om na te gaan of TDM en ODM werkelijk ook waarde heeft voor de gebruikers, meerbepaald de business users, worden TDM en ODM verderop onderworpen aan de Unified Theory of Acceptance and Use of Technology (UTAUT), een vervolg op het reeds bekende Technology Acceptance Model. Deze theorie onderzocht de verschillende Acceptance methodologieën en modellen en combineerde deze samen tot een overkoepelend model. Omdat dit model het meest recent is en deze de integratie van eerdere modellen bevat, werd dit verkozen als het te gebruiken model. De Unified theory werd ontwikkeld door V. Venkatesh, M. Moris, G.B. Davis en F. D. Davis.

UTAUT kan gebruikt worden door managers om het succes en de waarde te bepalen van een nieuwe technologie, tool of systeem. Decision modeling en de tools die daar rond gebouwd zijn, zijn nog redelijk jong. Zowel het modelleren van beslissingen als het gebruiken van de tools moeten nog algemeen aanvaard worden door het grote publiek. Het is daarom ook belangrijk te bepalen hoe de gebruikers TDM en ODM zullen aanvaarden.

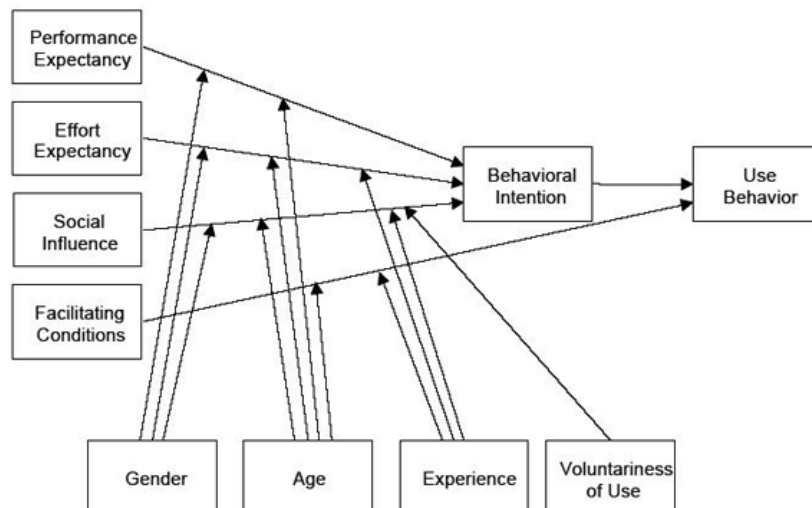
Naast de tool zelf, zijn andere aspecten zoals training en marketing van een tool ook van belang. Maar wanneer de tool zelf niet business user gericht is, dan zullen training en marketing ook minder effect hebben. Alle aspecten spelen namelijk een grote rol.

De Unified Theory of Acceptance and Use of Technology wordt hieronder kort geschetst.



Figuur 58: Basisconcepten Unified theory Uit [vertaald uit] User acceptance of Information technology (p.447), door Venkatesh, V., Morris, M.G., Davis, F.D., and Davis, G.B., 2003, MIS Quarterly, 27.

Het basis concept onderliggend aan de theorie houdt in dat de individuele reactie van gebruikers op TDM en ODM invloed heeft op de intenties tot het gebruiken van de tools en deze intenties dan zelf invloed hebben op het werkelijk gebruik van de tool. Zelf heeft de individuele reactie ook rechtstreeks effect op het werkelijke gebruik en dit in beide richtingen, want wanneer de business user een tool gebruikt zal dit zijn mening veranderen in positieve of negatieve zin.



Figuur 59: Unified view of the acceptance model
Uit [Overgenomen uit] User acceptance of Information technology (p.447), door Venkatesh, V., Morris, M.G., Davis, F.D., and Davis, G.B., 2003, MIS Quarterly, 27.

Bovenstaand model bestaat uit meerdere onafhankelijke en afhankelijke variabelen. De belangrijkste variabelen worden hieronder kort geschetst en toegepast op TDM en ODM.

Performance expectancy is de graad waartoe de individuele business user gelooft dat het systeem of de tool hem/haar kan helpen om zijn/haar prestaties te verbeteren. (Venkatesh et al., 2003) Door de combinatie van TDM en ODM te gebruiken kan de Performance expectancy stijgen wanneer de tools goed gebruikt worden. Dit houdt in dat wanneer een middlemanager de processen analyseert waarvoor hij verantwoordelijk is en deze logica via TDM uitschets, dit voor de ontwikkelaar tijds winst kan opleveren. Wanneer de business user zelf de logica al grotendeels heeft kunnen modelleren in TDM, weet de gebruiker ook wat hij uiteindelijk ongeveer van ODM kan verwachten waardoor zijn geloof in de ondersteuning van het systeem sterk toeneemt.

Niet enkel voor de business user kan de performance expectancy toenemen, ook geldt dit voor de ontwikkelaar omdat hij/zij nu duidelijker weet wat er precies in het rulesysteem moet worden opgenomen. De businestaal werd namelijk via TDM in modellen en tabellen uitgedrukt waardoor er een soort van vertaling plaatsvindt. Bij deze opmerking moet wel een kanttekening gemaakt worden. Op dit moment is het namelijk nog niet mogelijk om de decision models uit TDM naar ODM te exporteren. Dit houdt in dat het handig is dat er een TDM model bestaat, maar dat het werk en tijd zou besparen mocht dit rechtstreeks exporteerbaar zijn zodat niet alles van nul af aan terug opgebouwd dient te worden in ODM. In de Roadmap van Bizzdesign, de eigenaar van de TDM tool, werd wel vermeld dat die export in de nabije toekomst mogelijk wordt. Dit zou dit probleem kunnen oplossen.

Uit de theory bleek dat het effect van Performance expectancy sterk gemodereerd werd door geslacht en leeftijd, namelijk mannen en jonge individuen ondervonden dit effect sterker.

Effort Expectancy is de graad van ease-of-use van de tool of het systeem. (Venkatesh et al., 2003) Toegepast op TDM en ODM houdt dit in dat het leerproces redelijk vlot verloopt omdat de tools intuïtief en grafisch opgebouwd zijn. Een gebruiker wordt bij het hele proces betrokken, dankzij de combinatie van TDM en ODM. Dankzij het gebruiksgemak kan de business user de beslissingsmodellen zelf aanmaken. Deze beslissingsmodellen worden dan in de Rule designer van ODM door de ontwikkelaar uitgebouwd. Hierna kan de business user via de business console in ODM zelf de rules aanmaken, aanpassen en controleren omdat de nodige elementen door de ontwikkelaar in het systeem werden gebracht. Ook de business console is dankzij zijn web 2.0. uiterlijk herkenbaar en makkelijker te gebruiken voor business users.

Op deze factor hebben zowel geslacht, leeftijd en ervaring effect. Het effect van ervaring is logisch verklaarbaar. Namelijk wanneer een gebruiker meer ervaring heeft dankzij training of door gewoon de tool te gebruiken, dan zal dit de ease-of-use verhogen. Qua leeftijd en geslacht bleek dit effect sterker te zijn bij vrouwen, oudere individuen en zij die nog weinig ervaring met de tool hadden.

Social influence is de graad waartoe een bepaald individu waarneemt dat belangrijke andere personen geloven dat hij/zij het nieuwe systeem moet gebruiken. (Venkatesh et al., 2003) Deze factor heeft minder betrekking op de tools zelf, maar wordt eerder bepaald door de bedrijfscultuur en -structuur. Wanneer ondergeschikten kunnen overtuigd worden door hun verantwoordelijken dat de tool werkelijk waarde kan bieden bij het uitvoeren van een bepaalde job, dan zal het individu dit ook sneller doen. Wanneer er dus gepland wordt om TDM en ODM in het bedrijf te implementeren, moeten alle mensen in dezelfde richting denken en ondergeschikten gemotiveerd worden voor het gebruik ervan. Enkel zo kan de optimale waarde uit een bepaalde tool of systeem gehaald worden.

Deze invloed bleek afhankelijk te zijn van geslacht, leeftijd, vrijwilligheid en ervaring. Sociale invloed bleek vooral effect te hebben op vrouwen en oudere gebruikers met weinig ervaring. Daarnaast was het effect sterker bij individuen waar de tool niet vrijwillig werd opgelegd, maar waar de gebruiker verplicht was de tool te gebruiken.

Facilitating conditions is de graad waartoe een individu gelooft dat de organisationele en technologische infrastructuur er zich toe leent om het gebruik van het systeem te ondersteunen. (Venkatesh et al., 2003) Voor TDM mag de ondersteunende infrastructuur beperkter zijn omdat het minder processing power vraagt dan bijvoorbeeld ODM. Dit omdat TDM vooral dient om de beslissingen te modelleren en eventueel te testen, maar niet uit te voeren. Voor ODM dient er zowel financieel als technologisch een veel grotere ondersteuning te zijn. ODM is een zeer dure tool die enkele grote ondernemingen zich kunnen aanschaffen. Vaak hebben zij ook een IT departement die de nodige servers en infrastructuur op poten kan zetten. Kleine en middelgrote ondernemingen zouden dan eerder een andere omgeving dienen te zoeken waar ze de modellen van TDM zouden kunnen uitvoeren. Dit kan bijvoorbeeld OpenRules zijn. Dit is een heel stuk goedkoper en wordt al ondersteund door TDM.

Volgens de Unified theory heeft deze factor geen significante invloed op de intentie tot het gebruik, maar wel op het gebruik zelf. Dit omdat business users vaak niet op de hoogte zijn van wat precies de kracht en beperkingen zijn van onderliggende infrastructuren en systemen. Wanneer daarentegen gebruikers reeds met de tool werken, is deze infrastructuur wel van groot belang daar dit de ervaring met de tool sterk beïnvloedt. Trage servers zullen bijvoorbeeld het gebruik negatief beïnvloeden.

Behavioral intention: Uit het onderzoek bleek dat de intentie tot het gebruik een positieve invloed heeft op het gebruik van de nieuwe tools en systemen. Wanneer de intentie groter is, zal de gebruiker sneller geneigd zijn de tool te proberen. (Venkatesh et al., 2003) Als de tools, in dit geval TDM en ODM, gemakkelijk zijn om in te stappen en mee te leren werken en daarnaast het ook de efficiëntie verhoogt, dan zal de business user de tool sneller accepteren en steeds meer gebruiken.

Naast het bespreken van de waarde van TDM, ODM en de combinatie van beiden, wordt in de volgende paragraaf aandacht geschonken aan de maturiteit van de tools.

5.4. DE MATURITEIT VAN TDM

De kwaliteit van business processen hangt af van de kwaliteit van de beslissingen. Dus zowel de business decision maturiteit als de business procesmaturiteit zijn van belang. Maturiteit van processen worden gemeten door Capability Maturity Model Integration (CMMI), de maturiteit van beslissingen door het Business Decision Maturity Model (BDMM) dat opgesteld werd door Knowledge Partner International (KPI).

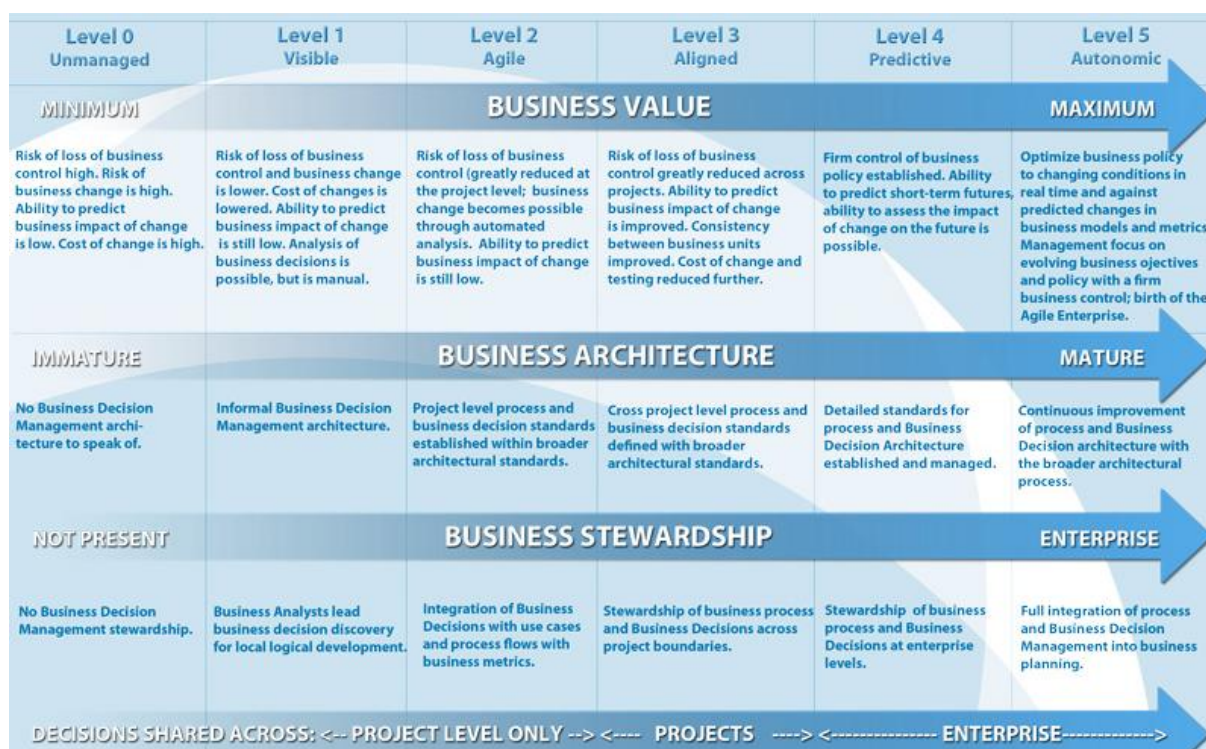
Er zijn 6 maturiteitsniveaus: 0 (Unmanaged), 1 (Visible), 2(Agile), 3 (Aligned), 4(Predictive), 5 (Autonomic). De naamgeving heeft betrekking op het decision model dat gecreëerd werd op dat niveau. De gedetailleerde beschrijving van het model is terug te vinden in bijlage D.

Het model bevat 3 vectoren voor ieder niveau:

De Business Value vector: De waarde en invloed van een bepaald maturiteitsniveau. Wanneer een bedrijf op een bepaald niveau zit, kan er bepaald worden naar welk niveau gestreefd wordt. De andere vectoren zijn afhankelijk van deze vector en bepalen de eigenschappen en criteria van dit niveau. Business waarde zit niet specifiek in een bepaalde regel, maar in het geheel van de bedrijfsbeslissingen die rekening houden met de businessobjectieven, business proces management en andere businessperspectieven. Hiervoor is het van belang te weten wat de impact van de businessbeslissingen is op business en IT.

De Business Architecture vector geeft de maturiteit van het Business Decision Management proces binnen de business architectuur weer.

De Business Governance vector geeft de maturiteit van de governance van het Business Decision Management proces weer.



Figuur 60: Het Business Decision Maturity Model (BDMM)
Uit [Overgenomen uit] business decision maturity model, door KPI, webartikel

TDM kreeg onlangs door KPI de Business Decision Maturity model 'level 2' certificatie. Dit is het niveau 'Agile'. Voor deze masterproef werd onderzocht waarom TDM precies de level 2 certificatie kreeg.

De business value: Het focuspunt bij level 2 is het verkrijgen van flexibiliteit. Business decisions moeten duidelijk en snel aan te passen zijn. Dit is bij TDM het geval. Rule families binnen de tool zijn snel aan te passen door cellen, rijen of kolommen te veranderen, toe te voegen of te verwijderen. Daarnaast is op dit niveau de businesslogica reeds geëxtraheerd uit het proces. Dit kan met behulp van TDM door de link van de BPMN processen met de verschillende decision

models te leggen. Zo kan de gebruiker de logica uit de processen extraheren maar het er toch mee linken.

Op niveau 2 heeft de gewenste flexibiliteit voornamelijk betrekking op een individueel project. Binnen TDM kan de gebruiker namelijk ook rules hergebruiken binnen een project maar nog niet overheen verschillende projecten.

Dankzij deze tool is het mogelijk om beslissingen meer transparant te maken in de processen. De gebruikers kunnen precies zien wat de invloed van de business logica is op de processen. Dankzij die transparantie is het makkelijker aanpassingen te maken bij wijzigingen in het beleid of de doelen van de onderneming. Het risico op controleverlies over de business wordt dus op niveau 2 sterk gereduceerd.

De business architecture: projecten en tools die niveau 2 behalen, dienen de mogelijkheid te bieden om op een consistente manier business decisions te creëren, te managen en te onderhouden zodat ze vlot kunnen aangepast worden wanneer doelen of condities veranderen. De business logica moet namelijk duidelijk gescheiden zijn van de andere dimensies. Ook hier kloppen deze beweringen voor TDM aangezien de tool een intuïtieve interface biedt die het toelaat business decisions vlot te creëren, te managen, te veranderen en te onderhouden. Het decision model kan gelinkt worden met een glossary en andere modellen zoals het procesmodel, maar is er toch duidelijk van gescheiden.

De business governance: processen dienen het mogelijk te maken de juiste businessbeslissingen te nemen, ze te optimaliseren en te controleren. Meer specifiek betekent dit dat business decisions gebaseerd moeten zijn op business objectieven en de bijbehorende metrics. Binnen TDM is het mogelijk de nodige decision models en rule families hiervoor te creëren.

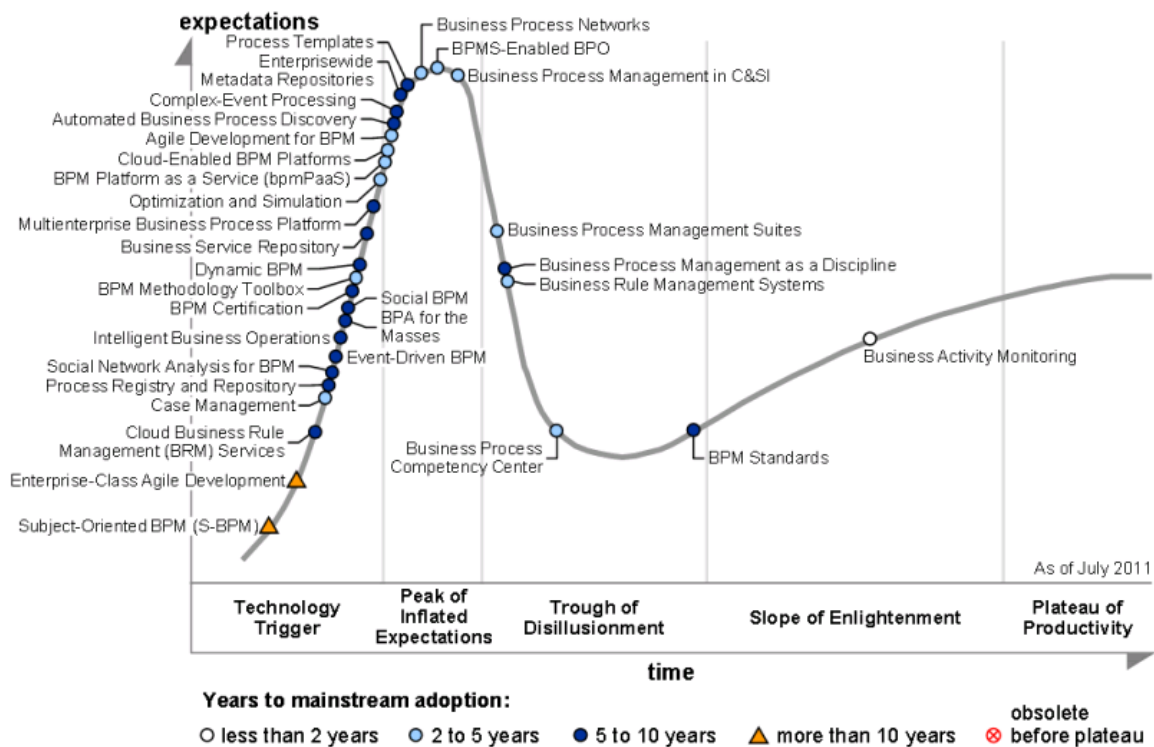
De reden waarom TDM nog niet in aanmerking komt voor de level 3 certificatie is dus het gebrek aan integratie tussen verschillende projecten. De gebruiker kan rules creëren en hergebruiken binnen een bepaald project waardoor een sterke integratie binnen een bepaald project zeker bestaat. Maar eenmaal de gebruiker verschillende projecten maakt, is die integratie tussen de verschillende projecten niet aanwezig. Zolang de tool toelaat slechts rules te hergebruiken binnen 1 project, kan TDM geen level 3 certificaat krijgen, dit heeft namelijk betrekking op het hergebruiken van rules overheen meerdere projecten. Dit is één van de belangrijke schakels die TDM nog mist.

5.5. DE MATURITEIT VAN ODM

Gartner definieerde in de paper rond de Hype cycle for business process management een maturiteitsmodel dat gebruikt kan worden om zowel een bepaalde technologie als een tool te evalueren.

De hype cycle en de bijbehorende Priority Matrix hebben als doel inzicht te verschaffen in de relatieve maturiteit van bepaalde technologieën, IT methodologieën en managementdisciplines. Het geeft weer op welk deel van de cyclus een bepaalde technologie zich bevindt en hoe lang het zal duren eerdad een bepaalde technologie of tool een stabiele maturiteitsgraad zal behalen. De hype

cycle vormt een gefilterd beeld van het Business process management gebied. Dit omdat de nadruk vooral gelegd wordt op nieuwe concepten en niet op de reeds mature concepten die al de plateau van productiviteit bereikt hebben. (Amelior, 2011)

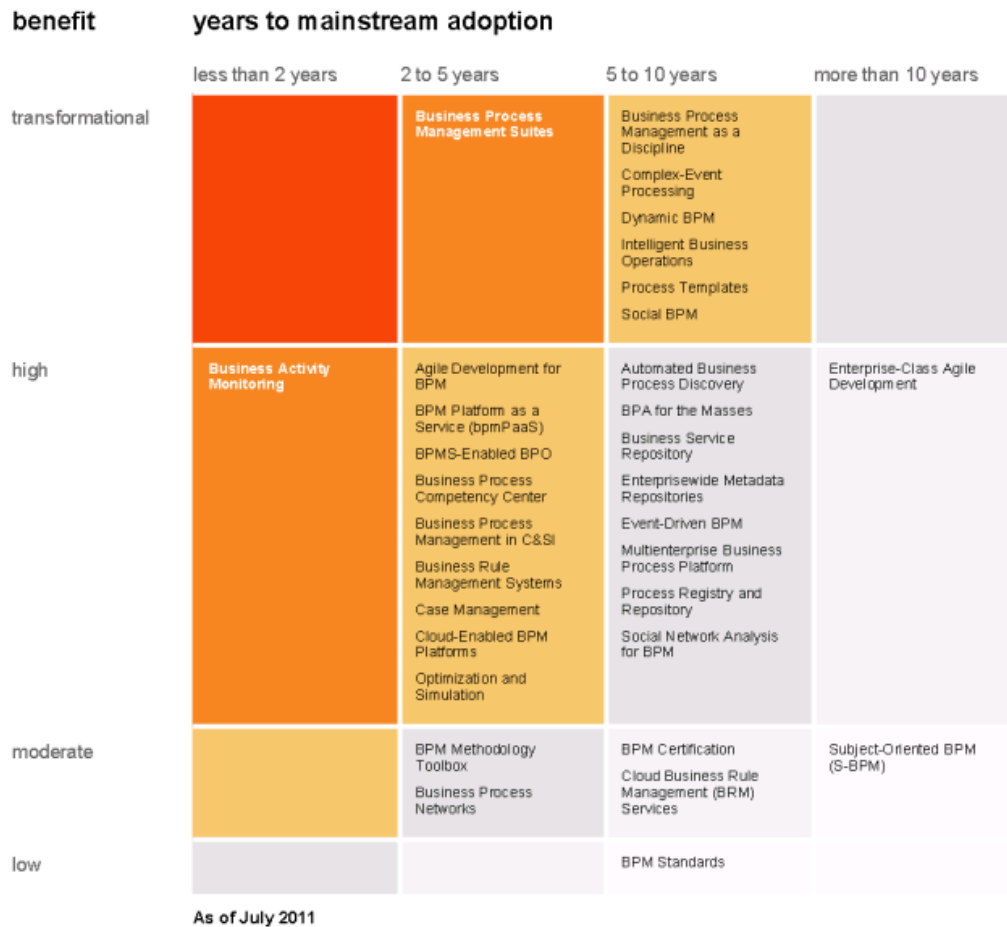


Figuur 61: Hype cycle for business process management
 Uit [Overgenomen uit] Hype cycle for business process management, door J. Dixon, T. Jones, 2011, Gartner research.

De hype cycle bestaat uit 5 fases: de '**Technology trigger**', wat kan beschouwd worden als een doorbraak, een productlancering, een publieke demonstratie. De tweede fase '**Peak of inflated expectations**', waarbij bedrijven en gebruikers vaak overenthousiaste of onrealistische verwachtingen over de tool stellen, vormt een kantelpunt voor het succes. Successen worden in deze fase meestal enkel geboekt door grote leidende bedrijven. Meestal zijn er veel successen, maar nog meer falingen. De derde fase '**trough of disillusionment**' houdt in dat vanwege de onrealistische verwachtingen die niet ingelost konden worden, de aandacht van het publiek en de media verzwakt waarbij de gebruikers terug voorzichtiger verwachtingen stellen. '**Slope of enlightenment**' is de vierde fase en houdt in dat bedrijven door middel van experimenteren en intensief gebruik van een bepaalde technologie, de toepasbaarheid, risico's en meerwaarde waarnemen. Wanneer er ook commerciële off-the-shelf producten aangeboden kunnen worden, wordt deze fase versneld. Zo zorgt TDM er bijvoorbeeld voor dat de grote BRMS systemen die steeds zelf toegankelijker worden, nog toegankelijker worden bij het begin van het modelleren. De laatste fase is het '**Plateau of productivity**' waarbij de voordelen van de technologie niet enkel via studies, maar werkelijk in een bedrijfsomgeving bewezen zijn. Tools worden in deze fase ook steeds stabielier omdat ze al een tweede of derde versie zijn. Meer en meer organisaties zullen zich in deze fase aangesproken voelen om deze technologie ook een kans te geven. Dit omdat de

technologie zich heeft kunnen bewijzen en de risico's om er in te investeren steeds minder groot zijn. (Amelior, 2011)

Naast de hype cycle creëerde Gartner ook een Priority matrix, deze toont hoe Gartner gelooft dat een bepaalde technologie die opgenomen werd in de hype cycle een bedrijf het mogelijk maakt binnen een bepaalde tijdsframe er voordelen uit te halen. Business rules management systemen, zoals ODM werden opgelijst bij de in de cel 'high', '2 tot 5 jaar'. Dit houdt in dat het nog 2 à 5 jaar zal duren eerdat deze technologie algemeen aanvaard wordt. Deze studie was van 2011. Om dit moment is het reeds 2014 en valt het ook op dat meer en meer bedrijven BRMS adopteren.



Figuur 62: Priority Matrix for business process management
Uit [Overgenomen uit] Hype cycle for business process management, door J. Dixon, T. Jones, 2011, Gartner research.

Gartner definieerde BRMS als een business rule management omgeving die het mogelijk maakt business rules te creëren, registreren, classificeren, verifiëren, opzetten en uit te voeren. De belangrijkste onderdelen die volgens Gartner in een BRMS moeten zitten: een execution engine, repository, rule templates, een geïntegreerde ontwikkelings- en simulatieomgeving, management, administratie en analysemogelijkheden.

Gartner, analyseerde dit ook in 2013, maar deze paper kon niet worden bemachtigd. Wel werden via mail de posities verkregen waar de verschillende elementen zich in 2013 op de Gartner Hype

cycle bevonden. Volgens deze bron bevonden Business rule management systemen zoals ODM, zich op dit moment al in het dal van de derde fase 'Sliding into the trough'.

De positie van ODM: Omdat er reeds heel wat bedrijven BRMS platformen aanbieden, is de hype gereduceerd en verkrijgen bedrijven steeds meer ervaring door de implementaties ervan. Op dit moment bevindt ODM zich in de 'trough', wat inhoudt dat afhankelijk van de ondersteuning en successen van deze systemen, het ofwel geleidelijk aan werkelijk matuur wordt of juist geleidelijk aan verdwijnt. Dit is afhankelijk van de markt, de kracht van de BRMS aanbieder en het succes van de use cases. Zo zou een tegenslag binnen business rule management ook even de groei van de tools kunnen tegenhouden.

De business impact van ODM: BRMS geven de mogelijkheid tot enkele grote voordelen. BRMS krijgt dan ook de 'high benefit rating'. Het biedt nieuwe wegen om applicaties te sturen en te beheren, wat kan leiden tot kostenreductie en flexibiliteit. Die flexibiliteit maakt het mogelijk sneller te reageren op opportuniteiten. Ook prestaties kunnen verhogen en fouten kunnen gereduceerd worden. De gebruikers die de grootste impact hiervan ondervinden zijn de business process owners, business analisten en ontwikkelaars. De tools moeten het mogelijk maken IT en Business dichter bij elkaar te brengen door het delen van een rule-based communicatienetwerk. Daarnaast kan het de automatische gevoeligheid voor problemen verhogen (een bepaalde rule wordt getriggerd bij het overtreden van een bepaalde beleidsregel). Ook worden processen transparanter door de mogelijkheid van het uitvoeren van audit trails op de rule repository.

Het maturiteitslabel dat BRMS van Gartner verkreeg in 2011 is 'Adolescent', dit houdt in dat de functionaliteiten van de technologie meer matuur worden en dat de early adopters de technologie een kans geven. De systemen zijn meestal dan al een tweede of derde generatie en er wordt steeds minder aangepast aan het aanbod. Ondertussen is het BRMS zoals ODM al werkelijk in de 'trough' beland. Dit houdt in dat BRMS door meer bedrijven zal worden aangeboden en dat bedrijven het ook een kans willen geven. Het maturiteitslabel is daarom in 2014 ook geëvolueerd in 'Early mainstream' wat inhoudt dat de technologie zich al in meerdere bedrijven bewezen heeft en dat verschillende aanbieders zelf een systeem of tool ontwikkelen die hierop betrekking heeft. Vaak bereiken de grote tools nu al hun derde of vierde versie en verschijnen er out-of-the-box pakketten zoals bijvoorbeeld TDM van Bizzdesign.

HOOFDSTUK 6: CONCLUSIE

Voor deze masterproef werd er nagegaan hoe de plug-in 'The Decision Modeler' (TDM) in de Architect tool van Bizzdesign en het Business Rules Management Systeem 'Operational Decision Manager' (ODM) van IBM best gebruikt en gecombineerd kan worden om aan decision management te doen. Hiervoor werden zowel de functionaliteiten, beperkingen als best practices onderzocht die samen de waarde van de tool bepalen. Naast het analyseren van iedere tool apart, werd een mogelijke combinatie van de tools ook onderzocht. Door als handelingenieur in de beleidsinformatica deze tools zelf toe te passen door middel van een praktijkvoorbeeld, kon worden de waarde ingeschat worden en werd het duidelijk welke bedrijfsfuncties met deze tools zouden kunnen werken. Als laatste werd er bepaald wat de maturiteit van TDM en ODM op het moment van schrijven was.

De The Decision Modeler (TDM) plug-in van de Architect tool werd gebaseerd op het gelijknamige boek 'The Decision Model'. TDM vormt een modelleringomgeving om beslissingen uit te werken met behulp van een decision model en rule families. Omdat de Architect tool verschillende standaarden ondersteunt, kan het decision model binnen de tool gelinkt worden met andere modellen zoals een business process model. Dit maakt het makkelijk om tussen modellen te switchen via een simpele klik. Het maakt duidelijk waar beslissingen genomen worden en waar dit invloed zal hebben. Deze integratie kan wel nog beter aangezien de link wel bestaat, maar het nog geen effect heeft. Zo zullen de uitkomsten van beslissingen geen invloed hebben op de gevolgde weg in het business process model terwijl dit in realiteit wel zou moeten zijn. Dankzij het gebruik van TDM kan een gewone business user de gewenste business logica al ruw schetsen, wat de complexiteit voor de ontwikkelaar kan verlagen. Op dit moment ontbreekt wel nog de exportmogelijkheid naar Operational Decision Manager (ODM) waardoor de ontwikkelaar toch nog handmatig alles van scratch moet programmeren. Dit zou volgens de roadmap van Bizzdesign wel mogelijk moeten zijn.

Het Business Rules Management Systeem ODM van IBM vormt een ontwikkeling- en executieomgeving voor business logica in de vorm van business rules, rule tables, decision trees en events. Allereerst formuleert de ontwikkelaar in een Eclipse-gebaseerde console de klassen en attributen die nodig zijn voor het creëren van de rules. Omdat business users vaak pas betrokken worden bij de business rules wanneer ze al gecreëerd zijn, zou TDM hier verandering in kunnen brengen. Door in TDM decision models, rule families en UML klassen te creëren, kan de gebruiker zijn wensen en verwachtingen al aan de ontwikkelaar tonen. De ontwikkelaar kan dan op basis van die UML klassen, Java klassen aan te maken die samen een eXecution Object Model (XOM) vormen. Op basis van de XOM wordt er een Business Object Model (BOM) ontwikkeld. De BOM-vocabulary kan gebruikt worden om rules, decision tables en decision trees te creëren. Eenmaal dit voltooid is, dient de ontwikkelaar de rules te debuggen, te controleren, te testen en uit te voeren. Vanaf dat moment zijn de rules beschikbaar voor de business user. Indien de rules toch nog niet volledig naar wens van de business user zijn, dan kan hij/zij binnen ODM gebruik maken van de Business Console die toelaat rules te creëren of aan te passen en hierover met andere gebruikers te discussiëren in een web 2.0. omgeving.

De hierboven aangehaalde tools kunnen een meerwaarde bieden voor het business decision management domein. Iedere tool op zijn eigen manier. Toch bieden de tools niet enkel apart meerwaarde, ook gecombineerd kunnen ze extra waarde creëren. Dit werd in deze masterproef onderzocht door gebruik te maken van de Unified theory van V. Venkatesh, M. Moris, G.B. Davis en F. D. Davis. die betrekking heeft op de het accepteren van nieuwe technologieën. De combinatie van beide tools laat business users toe om tijdens het hele ontwikkelingsproces van de business rules actief bij te dragen. Dit zorgt er niet enkel voor dat de business users zich meer betrokken voelen bij het project en de technologie daarom sneller zullen accepteren, het zorgt er ook voor dat het werk van de ontwikkelaar in ODM sneller er correcter kan gebeuren.

Naast de waarde van TDM en ODM werd ook hun maturiteit onderzocht. Voor TDM werd er gebruik gemaakt van het Business Decision Maturity Model. TDM verkreeg onlangs een level 2 certificaat. Level 2 (agile) maturiteit houdt in dat er een grote flexibiliteit in de tool bestaat. Veranderingen in het beleid die invloed hebben op de business logica kunnen binnen een bepaald project snel aangepast worden dankzij de transparantie in de beslissingsmodellen in TDM. Daarnaast houdt het ook in dat rules herbruikbaar zijn binnen een project, maar niet overheen verschillende projecten. Ook dit is correct voor TDM daar rules gelinkt kunnen worden aan meerdere modellen binnen één project, maar nog niet aan meerdere projecten tegelijk. Eenmaal deze feature in TDM wordt opgenomen kan TDM evolueren naar een level 3 maturiteit.

Ook voor ODM werd de maturiteit onderzocht. Voor dit BRMS werd er gebruik gemaakt van de hype cycle en priority matrix van Gartner. ODM wordt geplaatst in de derde fase van deze cyclus, met name de 'trough of disillusionment'. De elementen die zich in deze fase bevinden, bestaan al even en zullen ofwel matuur worden ofwel stilaan verdwijnen. Dit is afhankelijk van het succes van de use cases en de kracht van de BRMS aanbieders. Omdat verwachtingen van gebruikers in het begin meestal onrealistisch zijn, en deze niet allemaal ingelost kunnen worden, leidt dit er toe dat gebruikers voorzichtiger hun verwachtingen zullen stellen.

In de Priority matrix kreeg het Business Rules Management Systeem (BRMS), de 'high benefit rating'. Deze rating werd verkregen omdat het BRMS nieuwe wegen biedt om applicaties te sturen en te beheren. Dit zou kunnen leiden tot kostenreductie en flexibiliteit. Het maturiteitslabel dat het BRMS van Gartner verkreeg is 'Adolescent', dit houdt in dat de functionaliteiten van de technologie matuur worden. De systemen zijn meestal al de tweede of derde generatie waardoor er steeds minder aangepast wordt aan het aanbod.

LIJST VAN DE GERAADPLEEGDE WERKEN

- Amelior. (2011). Hype cycle for business process management. Geraadpleegd op 9 mei 2014 via <http://www.amelior.be/ndl/artikels/artikel.asp?a=358&tc=1>
- Bauer, E. (2009) The Business Rule Approach. Geraadpleegd op 14 maart 2014 via <http://is.uni-paderborn.de/fileadmin/Informatik/AG-Engels/Lehre/WS0809/SE/Sonstiges/Seminar/Version1.0/Seminar.NAQ.Eduard.Bauer.v1.0.pdf>
- Bizzdesign. (2013). Bizzdesign gepositioneerd als Visionair in Magic Quadrant voor Enterprise Architecture Tools 2013. Geraadpleegd op 4 maart 2014 via <http://www.persberichten.com/persbericht/75319/BiZZdesign-gepositioneerd-als-Visionair-in-Magic-Quadrant-voor-Enterprise-Architecture-Tools-2013>
- Boyer, J., & Mili, H. (2011). Agile Business Rule Development. Berlijn: Springer-Verlag.
- Bpmntools.com. (2011). Bpmn tool evaluation criteria. Geraadpleegd op 17 mei 2014 via <http://www.bpmntools.com/article/bpmn-tool-evaluation-criteria/>
- Business Rules Group, (2000). Defining Business Rules ~What Are They Really? Geraadpleegd op 5 februari 2014 via http://www.businessrulesgroup.org/first_paper/BRG-whatBR_3ed.pdf
- Clarck, D., Berlandier, P. (2012). Making Better Decisions Using IBM WebSphere Operational Decision Management. Geraadpleegd op 20 maart 2014 via <http://www.redbooks.ibm.com/redpapers/pdfs/redp4836.pdf>
- Craggs, S. (2012). Competitive review of operational decision management. Geraadpleegd op 18 april 2014 via ftp://public.dhe.ibm.com/software/websphere/dm/pdf/ODM_Competitive_Review_Oct2012.pdf
- Dixon, J., Jones, T. (2011). Hype cycle for business process management. Geraadpleegd op 10 mei 2014 via http://www.adeptia.com/products/Hype_cycle_BPM_2011.pdf
- Fenn, J. , & Raskino, M. (2013). Gartner's Hype Cycle special report for 2013. Geraadpleegd op 7 mei 2014 via <http://www.gartner.com/doc/2574916?ref=sd>
- Gartner. (2013). Magic Quadrant for enterprise architecture tools. Geraadpleegd op 4 maart 2014 via <http://www.gartner.com/technology/reprints.do?id=1-1LDKC9U&ct=131008&st=sb>
- Hinkelmann, K. (2014). Business decision management – Business Decision Maturity Model. Geraadpleegd op 17 april 2014 via http://knut.hinkelmann.ch/lectures/bpm2013-14/08_BDMM.pdf
- IBM. (2009). Business modeling: how to guide. Verkregen via Professor Koen Vanhoof.
- IBM. (2013). IBM operational decision manager v8.5. information roadmap. Geraadpleegd op 8 april 2014 via http://www.ibm.com/developerworks/bpm/roadmaps/roadmap_odm85.html
- IBM. (n.d.). IBM WebSphere® Operational Decision Management Version 8.0 information center. Geraadpleegd op 2 april 2014 via <http://pic.dhe.ibm.com/infocenter/dmanager/v8r0/index.jsp>
- Jonkers, H. Quartel, D., Van den Berg, H., Franken, H., (2013). Integration of Business Decision Modeling in Organization Design. Geraadpleegd op 10 februari 2014 via <http://logon.bg.wordpress/wp-content/uploads/2013/10/Whitepaper-Integration-of-Business-Decision-Modeling-in-Organization-Design.pdf>
- KPI & Bizzdesign. (2013). KPI & BiZZdesign proudly present: The Decision Modeler powered by The Decision Model. Geraadpleegd op 4 februari 2014 via <http://www.bizzdesign.com/assets/Downloads/Presentations/Webinar-Slides-TDM-Webinar-BiZZdesign-KPI-Final.pdf>

- KPI. (2013). Business Decision Maturity Model. Geraadpleegd op 17 april 2014 via <http://kpiusa.com/index.php/The-Decision-Model/business-decision-maturity-model-bdmm.html>
- Purchase, J. (2013, juni 23). New tool to change the face of business decision modelling management [Web log post]. Geraadpleegd op 7 maart 2014 via <http://blog.luxmagi.com/2013/06/new-tool-to-change-the-face-of-business-decision-modellingmanagement/>
- Reijers, H.A., Van Hee, K.M. (n.d.). Business proces management in een notendop. Geraadpleegd op 5 november 2012 via <http://www.win.tue.nl/~hreijers/H.A.%20Reijers%20Bestanden/notendop.pdf>
- Robertson, B., (2013) Hype cycle for business process management 2013. Geraadpleegd op 14 mei 2014 via <https://www.gartner.com/doc/2565319>
- RoyalCyber. (2012). Best practices for developing rules in IBM ODM. Geraadpleegd op 5 april 2014 via <http://dms.royalcyber.com/wp-content/uploads/2012/03/Best-Practices-for-Developing-rules-in-IBM-ODM.pdf>
- Segatto, M., Dallavalle de Pádua, S.I., Martinelli, D.P. (2013) Business process management: a systemic approach?, *Business Process Management Journal*, 19(4), p.698 – 714. Doi: 10.1108/BPMJ-Jun-2012-0064
- Short, J. (2013) Magic quadrant for enterprise architecture. Geraadpleegd op 18 april 2014 via http://www.sap.com/bin/sapcom/en_us/downloadasset.2014-03-mar-05-21.magic-quadrant-for-enterprise-architecture-tools-pdf.html
- Taylor, J. (2011). Increasing BPM agility and effectiveness with Decision Management. Geraadpleegd op 8 november 2013 via <http://www.slideshare.net/jamet123/increasing-bpm-agility-and-effectiveness-with-decision-management>
- Taylor, J. (2012). Decision management systems: a practical guide to using business rules and predictive analysis. Indiana: Pearson.
- Venkatesh, V. (n.d.). Theoretical Models. Geraadpleegd op 6 mei 2014 via http://www.vvenkatesh.com/it/organizations/Theoretical_Models.asp#Con=structdefs
- Venkatesh, V., Morris, M.G., Davis, F.D., and Davis, G.B., (2003). User acceptance of Information technology (p.447), *MIS Quarterly*, 27(3), p. 425-478.
- Von Halle, B. (2001). Business Rules Applied: Building Better Systems Using the Business Rules Approach. New York: John Wiley and Sons.
- Von Halle, B., & Goldberg, L. (2010). *The Decision Model: A Business Logic Framework Linking Business and technology*. New York: CRC Press.

BIJLAGE

BIJLAGE A: OVERZICHT FIGUREN EN TABELLEN

OVERZICHT FIGUREN:

Figuur 1: Omvormen complexe naar simpele processen.....	6
Figuur 2: Voordelen Onafhankelijk management van beslissingen.....	6
Figuur 3: Hotspot-analyse.....	8
Figuur 4: Verschillende types van rules.....	9
Figuur 5: DE BRMS-omgeving.....	11
Figuur 6: Positie van het BRMS in het gehele systeem.....	12
Figuur 7: The decision model.....	13
Figuur 8: Een rule family.....	14
Figuur 9: Een voorbeeld van een decision model diagram.....	15
Figuur 10: De evolutie binnen business rules.....	17
Figuur 11: Bizzdesign in het Gartner Enterprise architecture tools quadrant.....	19
Figuur 12: De webSphere operational decision management omgeving.....	21
Figuur 13: De rule designer interface.....	22
Figuur 14: De rule project map binnen Rule Designer.....	22
Figuur 15: BOM to XOM mapping check.....	24
Figuur 16: Ruleflow EU-car rental.....	25
Figuur 17: Decision Validation Services.....	29
Figuur 18: Home (business console).....	29
Figuur 19: De rule package pricing (Business console).....	30
Figuur 20: De rollen binnen ODM.....	31
Figuur 21: Business rules approach.....	33
Figuur 22: BPM Request handling.....	34
Figuur 23: Glossary request handling.....	35
Figuur 24: The decision model en rule families (check Eligibility).....	36
Figuur 25: The decision model en rule families (Determine car group).....	37
Figuur 26: The decision model en rule families (Calculate price).....	38
Figuur 27: The decision model en rule families (Store reservation).....	39
Figuur 28: Excel scenario batch test.....	40
Figuur 29: BPM Car handover.....	41
Figuur 30: Glossary car handover.....	41
Figuur 31: The decision model en rule families (Check handover possibility).....	42
Figuur 32: Excel scenario batch test.....	42
Figuur 33: BPM car return.....	43
Figuur 34: Glossary car return.....	43
Figuur 35: The decision model en rule families (Determine overdue consequence).....	44
Figuur 36: The decision model en rule families (Determine car state).....	44
Figuur 38: Excel scenario batch test.....	45
Figuur 37: The decision model en rule families (Determine loyalty points).....	45
Figuur 39: Rule flow request handling.....	47
Figuur 40: Decision package eligibility (decision modeler).....	47

Figuur 41: Rule package car group availability (decision modeler)	49
Figuur 42: Decision tree car group availability (part 1).....	50
Figuur 43: Decision tree car group availability (Part 2).....	50
Figuur 44: Rule package pricing (decision modeler).....	51
Figuur 46: Modelpackages binnen Architect	58
Figuur 45: Interface TDM, Architect (links) Helpdocument Architect (rechts)	58
Figuur 47: Drag & Drop voor het opstellen van diagrammen en modellen.....	59
Figuur 48: list fact types.....	59
Figuur 49: Voorbeeld van een glossary	60
Figuur 50: Domeintypes.....	60
Figuur 51: Stappen binnen TDM.....	61
Figuur 52: Evalueren individuele bedrijfsbeslissing	61
Figuur 54: Overzicht modellen binnen architect.....	62
Figuur 53: Sjabloon voor testdata binnen excel (batchtest)	62
Figuur 55: Support binnen Architect.....	63
Figuur 56: Team tabblad binnen architect	63
Figuur 57: Categorieën waarde BRMS	65
Figuur 58: Basisconcepten Unified theory	69
Figuur 59: Unified view of the acceptance model.....	69
Figuur 60: Het Business Decision Maturity Model (BDMM)	72
Figuur 61: Hype cycle for business process management.....	74
Figuur 62: Priority Matrix for business process management.....	75

OVERZICHT TABELLEN:

Tabel 1: De business proces management cyclus	4
Tabel 2: Rollen in Operational decision manager.....	32
Tabel 3: Waarde van Operational decision manager.....	67

BIJLAGE B: DE PRINCIPES EN NORMAALVORMEN VAN THE DECISION MODEL

STRUCTURELE SIMPLICITEIT

Hiervoor bestaan 7 principes (Von Halle, Goldberg, 2010):

Principe 1: Het tabulaire principe

Dit principe houdt in dat er gewerkt wordt met de standaardvorm van een decision model, met name de 2 dimensionale rule families tabel (zie figuur 11) die intuïtief zijn en door de meeste business users gekend zijn.

Principe 2: Het heading principe

Elk label in de heading van de rule family moet een fact type zijn, waarbij een fact een stukje informatie is, bijvoorbeeld een bepaalde waarde en een fact type dus een algemene classificatie van een stukje informatie.

Belangrijk voor elk fact type is het beschrijven van de definities en domeinen. Dit om de interpretatie uniform te maken en de eenheid van een bepaald fact type vast te leggen. Deze definities en domeinen worden dan best opgenomen in een glossary. Om geen verwarring te veroorzaken dient ieder label in de rule family naar maximaal 1 fact type te verwijzen. Dit fact type neemt ofwel de rol aan van conditie of van conclusie.

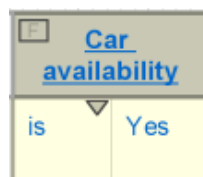
Een standaard benoemingsconventie wordt aangeraden voor het benoemen van fact types:

Business context (business object, data-entiteit, businessconcept) + beschrijvend woord + type van waarde (datum, naam, code)

Voorbeeld: Car Group Availability. Het gebruik van business context (Car) als eerste maakt het makkelijker om een alfabetische lijst te maken en vlot iets terug te vinden.

Principe 3: Het cel principe

Elke cel is een atomische logische uitdrukking 'operand (heading) + operator + operand (inhoud cel)' waarbij de inhoud van de cel dient te passen bij de heading



Cel uit de rule family

Veranderingen in de cellen zullen over het algemeen vaker gebeuren dan in de headings. Dit omdat de fact types minder snel veranderen dan fact values. Aanpassingen in de headings hebben doorgaans ook een grotere impact.

Principe 4: Het rij-principe:

Het is de correlatie van de condition fact types met de conclusie fact type. Er is namelijk sprake van functionele afhankelijkheid waarbij de waarde van een set van cellen een unieke waarde van de conclusiecel bepaalt.

Regel	Regel-patroon	 <u>Blacklisted</u>	 <u>Another reservation at date</u>	 <u>Drivers age</u>	 <u>Eligibility</u>
1	4	is ▼ no	is ▼ no	is ▼ Accepted	is ▼ yes

Rij uit de rule family table

Dit principe heeft een sterke invloed op de normalisatie, voornamelijk op de **First normal form**, wat betekent dat de rule family op slechts 1 manier kan worden gerepresenteerd en geïnterpreteerd. Dit houdt specifiek in dat een bepaalde rij niet meer kan onderverdeeld worden in meerdere rijen om die conclusie te bereiken. Indien dit wel het geval zou zijn, dient deze rij opgesplitst te worden in meerdere rijen met verschillende conclusies. Om dit te bereiken dient principe 5 en 6 toegepast te worden.

Principe 5: Het conclusie principe:

Een rule family mag slechts 1 conclusie fact type bevatten (zie figuur 14). Het zorgt dat elke rij een conclusie vormt over maximaal 1 fact type. Dus geneste conclusie fact types dienen in een andere rule family gezet te worden, dit vergemakkelijkt het aanpassen van de rule family bij een wijziging in het model.

Principe 6: Het conditie principe

Om een bepaalde conclusie te bekomen, dienen alle fact values in dezelfde rij waar te zijn. Er mag dus geen OR in vervat zitten, enkel AND's. Het geheel van rijen met een gemeenschappelijke set van conditiecellen wordt een rule pattern genoemd.

Principe 5 en 6 leiden samen tot het vormen van de **Second normal form**. Een conclusie moet functioneel afhankelijk zijn van alle condities, anders verwoord, de populatie van een cel die irrelevant is voor de conclusie moet verwijderd worden. Anders zou dit leiden tot duplicatie van business logica.

Principe 7: Het connectie principe

Wanneer er een link gelegd wordt tussen de conclusie van een bepaalde rule family en een conditie van een andere rule family, hebben ze beiden invloed op elkaar. Daarom kan een rule family geen lege conclusiecellen hebben, daar dit kan leiden tot een probleem in een andere rule family. Wanneer er wel lege conclusiecellen zouden zijn, kan de conclusie op basis van de businesslogica niet met zekerheid succesvol voltooid worden.

DECLARATIEVE OPBOUW

Een decision model dient declaratief te zijn om logisch onafhankelijk te zijn van de technologie en de processen zodat het een maximale stabiliteit (bruikbaar in elke omgeving) en flexibiliteit (snel aanpasbaar) heeft voor het uitwerken en managen van de business logica.

Dit punt werd opgebouwd rond 3 principes (Von Halle, Goldberg, 2010):

Principe 8: Het declaratieve heading principe

Er is geen geïmpliceerde volgorde in de kolommen in een rule family. Het zijn declaratieve kolommen, daar de volgorde geen betekenis heeft en dus op verschillende manieren getest kan worden. Wanneer de gebruiker dit principe wenst na te kijken, dient er nagegaan te worden of er geen verborgen conditiesequenties in de rule family vervat zijn. Wanneer dit wel het geval is, moet dit uitgewerkt worden in het procesmodel en verwijderd te worden uit het decision model. Ook andersom kan het gebeuren dat er bepaalde fact types opgenomen worden in het proces model terwijl deze in het decision model horen opgenomen te worden.

Principe 9: Het declaratieve body principe

Er is geen geïmpliceerde volgorde in de rijen in een rule family. De rijen dienen onafhankelijk van elkaar te zijn, zodat deze individueel geüpdatet kunnen worden. Dit houdt ook in dat er geen duplicatie en contradictie in de rijen mag voorkomen. Iedere rij dient uniek te zijn met behulp van zijn condition keys.

Principe 10: Het declaratieve afgeleide relatie principe:

Er is geen geïmpliceerde volgorde bij de rule families onderling. Ondanks dat bepaalde rule families input nodig hebben van andere rule families, hoeven ze niet precies in die volgorde uitgevoerd te worden.

Wanneer namelijk gestart wordt met de fact values die reeds bekend zijn zonder input van andere rule families en zo verder de rule families uitwerkt tot een conclusie bekomen wordt in de Decision rule family, wordt dit Forward chaining genoemd.

Maar ook andersom is het mogelijk, namelijk backward-chaining houdt in dat er vertrokken wordt vanaf een bepaalde eindconclusie, en zo dieper in de rule families de paden afgegaan wordt tot alle fact values overeenkomen met de waarden die waargenomen werden. Dit principe houdt dus in dat het de gehele decision model structuur geëvalueerd kan worden in welke volgorde dan ook.

Uit deze principes blijkt dat sequentie geen rol heeft, toch moet dit voorzichtig benaderd worden. Het klopt voor de business logica en het Decision model dat sequentie geen rol speelt. Toch kan een bepaalde volgorde bij het navigeren door het decision model een sterke impact hebben op de

prestaties van de technologie die dit model gebruikt. Ondanks deze opmerking, is het zo dat deze technologie niet in rekening wordt gebracht bij het opstellen van het decision model, zodat het zo onafhankelijk mogelijk is.

OPTIMALE INTEGRITEIT

Integriteit kan gedefinieerd worden als de meting van de bruikbaarheid en correctheid van een decision model. Op drie manieren moet het correct zijn, op vlak van structuur (minimale redundantie), logica (consistent en compleet) en businesswaarde (positieve beïnvloeding businessprestaties).

Om een optimale integriteit te bekomen dient normalisatie nagestreefd te worden. Dit houdt in dat een decision model geanalyseerd en verdeeld dient te worden in verschillende structuren die proces en technologie onafhankelijk zijn (Von Halle, Goldberg, 2010):

Principe 11: Rule pattern Transitive conditions principle

Er mogen geen functionele afhankelijkheden zijn tussen de condities in een rule pattern ten opzichte van de conclusie. Dit wordt ook wel de **third normal form** genoemd. Meer specifiek houdt het in dat geen condities leiden tot een conclusie over een andere conditie. Er mag dus geen geneste conclusie vervat zitten in een conditie. Wanneer dit wel het geval is dient het rule pattern opgesplitst te worden in meerdere rule patterns.

Principe 12: Rule family and role pattern consistency principle

In een Rule family mogen geen inconsistenties voorkomen binnen en tussen de rule patterns. Dit principe houdt in dat iedere uitvoering van een rule pattern moet leiden maximaal 1 conclusie. Daarnaast dienen de condities van een rule pattern enkel die waarden op te nemen die de gebruiker als input kan verwachten. De verschillende waarden binnen een bepaald fact type mogen niet overlappen omdat dit kan zorgen voor meer dan 1 conclusie.

Principe 13: Rule family transitive conditions principle

Rule families die zowel direct als indirect met een andere rule family verbonden zijn, hebben een redundante relatie. Dit mag niet volgens principe 13 en zal dus dan ook veranderd moeten worden zodat ze enkel nog indirect een relatie hebben.

Principe 14: Inferentiele integriteit principe

Alle mogelijke conclusies in een ondersteunende rule family dienen opgenomen te worden als fact values in de afhankelijke rule family. Wanneer dit niet het geval is, kan het proces onmogelijk verder wanneer er bij de conclusie een waarde wordt bekomen die niet in de condities van de afhankelijk rule family vervat zit.

Principe 15: Business alignment principe:

Een decision model moet in overeenkomst staan met de business direction en metrics voor het meten van de vooruitgang. Het moet duidelijk zijn welke impact een decision model heeft op de prestaties van de onderneming.

Om het mogelijk te maken dit principe te volgen, dient de gebruiker eerst te zoeken naar de businessopportunity waarbij het gebruiken en managen van het decision model, de oplossing is. Daarna dienen metrics opgesteld te worden om te meten hoe groot de vooruitgang is. Hierna dienen er procesmodellen gemaakt te worden en dient de link tussen het decision model en het procesmodel duidelijk gemaakt te worden.

BIJLAGE C: UITWERKEN VAN PRAKTIJKVOORBEELD

STAP 1.1 HET MODELLEREN VAN HET FUNCTIONELE MODEL:

Het doel van deze stap is het beschrijven van de functies die door de verschillende bedrijfsactoren uitgevoerd worden. De output van deze stap zijn high-level use cases, een stakeholders tabel en een business events tabel.

Voor deze thesis wordt er gebruik gemaakt van de fictieve autoverhuurbedrijf case. Om deze redenen wordt hieronder dan ook even een schets gemaakt van de organisatie:

De visie van EU-Rent:

“Het autoverhuurmerk zijn, dat geprefereerd wordt door bedrijfsmensen in de regio’s waarin we actief zijn”

Het objectief van EU-Rent dat de visie ondersteund:

“EU-rent moet industry-leading klantenservice aanbieden door voertuigen beschikbaar te stellen voor verhuur wanneer en waar klanten ze verwachten.”

Algemene informatie over EU-Rent:

EU-Rent is een autoverhuurbedrijf met filialen in vijf landen waar ze autoverhuurdiensten aanbieden.

EU-Rent behoort tot de EU-Corporation waarvan ook EU-Fly en EU-Stay dochterondernemingen zijn.

De klant kan een wagen huren in één van de 1000 filialen via het web of via het filiaal zelf. Voertuigen behoren tot een bepaalde voertuiggroep. Ieder filiaal heeft een manager en een boekingsbediende die de verhuring afhandelt.

Allereerst dienen de verschillende actoren opgelijst te worden in een stakeholderstabel:

Naam actor	Organisatie	Rol
Boekingsbediende	Filiaalkantoor	Registreert informatie voor het boeken en verhuren van een voertuig.
Filiaalmanager	Filiaal	Managet het personeel en legt een strategie vast voor het filiaal binnen de beleidslijnen van het overkoepelende EU-Rent.
Klant	Privé persoon of organisatie	Reserveert wagen voor persoonlijk of voor bedrijfsgebruik.
Servicedepot	Service filiaal	Repareert en onderhoudt de voertuigen van EU-Rent

Voor de thesis werden drie grote use-cases uitgewerkt. Namelijk, de Rent-a-car use-case, Return-a-car use-case en customer loyalty use-case.

Case 1: Rent-a-car use-case:

Use case naam:	Het huren van een voertuig in het filiaal.
Beschrijving:	Een klant wenst een wagen te huren.
Actoren:	Boekingsbediende, klant, 'Klantendatabase', 'wagendatabase'
Voorwaarden:	De boekingsbediende dient toegang te hebben tot het systeem.
Geslaagde eindcondities:	De klant vertrekt met de wagen met een getekend contract OF De klant vertrekt zonder wagen doordat aan bepaalde voorwaarden niet werd voldaan
Niet geslaagde eindcondities:	Het verhuurverzoek van de klant kan niet worden afgehandeld wegens bepaalde problemen.
Geslaagd scenario	✓ De boekingsbediende registreert de nodige informatie in het systeem met betrekking tot de reservatie (voertuiggroep, start- en einddatum, pick-up en return filiaal) en met betrekking tot de klant. ✓ Het systeem dient dan de beschikbaarheid van de wagen en mogelijkheid tot verhuring aan de specifieke klant na te gaan. ✓ Het systeem toont beschikbaarheid en prijs voor de verhuring.
Alternatief	
Input	Klantinformatie en reservatie-informatie
Output	Een contract of een weigering

Case 2: Return-a-car use-case:

Use case naam:	Het terugkrijgen van een wagen na verhuring
Beschrijving:	Een klant levert de wagen terug binnen op het einde van de huurperiode en betaalt de verschuldigde som
Actoren:	Klant, boekingsbediende, 'klantdatabase', 'wagendatabase'
Voorwaarden:	De boekingsbediende dient ingelogd te zijn in het

	systeem.
Geslaagde eindcondities:	De betaling is voltooid, het contract kan worden gearhiveerd en de wagen is terug beschikbaar voor verhuur.
Niet geslaagde eindcondities:	De betaling, de registratie van het voertuig, of de archivering van het contract kan niet worden voltooid.
Geslaagd scenario	✓ De boekingsbediende registreert de teruggave van de wagen in het systeem. ✓ Het systeem registreert de betaling. ✓ Het systeem archiveert het contract.
Alternatief	
Input	Betalingsbewijs
Output	Bevestiging afhandeling verhuurcontract.

Case 3: Customer loyalty use-case:

Use case naam:	Het toekennen van loyaliteitspunten aan klanten
Beschrijving:	Wanneer een klant meer dan vier maal een wagen gehuurd heeft, kan hij loyaliteitspunten sparen
Actoren:	Boekingsbediende, 'klantdatabase'
Voorwaarden:	De boekingsbediende dient toegang te hebben tot het systeem.
Geslaagde eindcondities:	Het nieuwe puntenaantal van de klant is geregistreerd.
Niet geslaagde eindcondities:	Het nieuwe puntenaantal werd niet geregistreerd.
Geslaagd scenario	✓ Het systeem berekent aantal verdiende punten na voltooide betaling klant. ✓ Het systeem voegt punten toe aan de klant-record.
Alternatief	
Input	Verhuurgegevens, betalingsgegevens, klantgegevens
Output	Bevestiging registratie loyaliteitspunten.

Ten derde dienen de Business events geïdentificeerd te worden in deze use cases. Business events triggeren een reeks activiteiten. Deze events kunnen zowel intern als extern zijn. Een business process is een reeks van activiteiten die getriggert wordt door een extern event.

Hieronder wordt de Business event tabel voor EU-Rent weergegeven.

Referentie	Business event	Beschrijving
BE01	Het huren van voertuig in een filiaal of via het web.	Het reserveren van het voertuig.
BE02	Het overhandigen van de huurauto	De klant komt de wagen ophalen. Er moet een controle gebeuren of dit ook werkelijk kan.
BE03	Het ontvangen van de huurauto na de verhuurperiode	Het terug in ontvangst nemen van de wagen. Het regelen van de betaling en loyaliteitsbonus.

Voor elk business event wordt er een activiteitentabel uitgewerkt.

Business event (BE01) Het huren van voertuig in een filiaal of via het web:

Business		event BE01
Activiteit-ID	Activiteit Naam	Beschrijving
BE01-1	Register reservation	Het request wordt in de database geregistreerd.
BE01-2	Check eligibility driver	De ingevoerd gegevens van de klant worden gecontroleerd om na te gaan of de klant een wagen kan huren.
BE01-3	Determine offered Car group	De gevraagde Car Group wordt nagegaan op beschikbaarheid, een alternatief kan gegeven worden.
BE01-4	Calculate price	Op basis van de aangeboden wagen, wordt er een prijs berekend.
BE01-5	Store reservation	Het request wordt geregistreerd.

Business event (BE02) Het overhandigen van het voertuig:

Business		event BE01
Activiteit-ID	Activiteit Naam	Beschrijving
BE02-1	Obtain required customer information	Extra benodigde gegevens worden gevraagd.
BE02-2	Check handover possibility	Wanneer de klant de wagen komt halen, moet de klant en de wagen eerst nog gecontroleerd worden.
BE02-3	Handover car	De autosleutels worden aan de klant overhandigd.

Business event (BE03): Het ontvangen van de huurauto na de verhuurperiode:

Business		event BE03
Activiteit-ID	Activiteit Naam	Beschrijving
BE03-1	Receive car	De wagen wordt terug ontvangen in een filiaal. Dit filiaal krijgt de eigendom.
BE03-2	Determine car state	De staat waarin de wagenterug wordt gebracht, wordt nagekeken.
BE03-03	Release car	De wagen wordt terug beschikbaar gesteld voor reservering. Indien er schade of onderhoud nodig is, wordt dit eerst gedaan.
BE03-04	Determine loyalty points	Het aantal punten dat hij verdiend heeft, wordt toegekend aan de klant.
BE03-05	Determine total cost	Bepaal de uiteindelijke kost die de klant moet betalen.

STAP 1.2. HET MODELLEREN VAN HET STRUCTURELE MODEL

In deze stap dient nagegaan te worden welke businessentiteiten een rol hebben in de processen en hoe zij zich ten opzichte van elkaar verhouden. Belangrijk hierbij is dat iedere entiteit wordt gedefinieerd vanuit het business perspectief.

Voor EU-rent leidt dit tot volgend overzicht:

Rentals: De meeste rentals gebeuren via reservatie, waarbij de periode van verhuur en de autogroep vastgelegd wordt.

Returns: Wagens geleend van een bepaald filiaal kunnen terug binnengebracht worden in een ander filiaal. Het filiaal waar de wagen vertrokken is, blijft de verantwoordelijkheid en eigendom hebben tot de wagen wordt binnengebracht in het andere filiaal, dan krijgt deze de eigendom.

Cliënteel: Een klant kan meerdere reservaties hebben, maar slechts één wagen tegelijk huren. EU-rent houdt data bij over de verhueringen aan deze klant. Daarnaast ook eventuele slechte ervaringen en loyaliteitsschema's.

STAP 1.3. HET MODELLEREN VAN HET GEDRAGSMODEL

Business event (BE01) Het ontvangen en verwerken van de rental request:

Business		event BE01
Activiteit-ID	Activiteit Naam	Beslissingen
BE01-1	Register reservation	
BE01-2	Check eligibility driver	✓ Verifieer of klant op de Blacklist staat ✓ Verifieer gegevens klant
BE01-3	Determine possibility to rent car	✓ Verifieer de Auto(groep) beschikbaarheid ✓ Verifieer andere verhuurmogelijkheden
BE01-4	Calculate price	
BE01-5	Store reservation	

Business event (BE02) Het overhandigen van het voertuig:

Business		event BE02
Activiteit-ID	Activiteit Naam	Beschrijving
BE02-1	Obtain required customer information	
BE02-2	Check handover possibility	✓ Verifieer of klant de wagen mag huren (leeftijd, rijbewijs, verzekering, probleemgeval, schadecategorie) ✓ Verifieer de credit kaart garantie ✓ Verifieer de gereedheid wagen (benzine, schade)
BE02-3	Handover car	

Business event (BE03) Het terug in ontvangst nemen van de wagen en afhandelen verhuuring:

Business		event BE03
Activiteit-ID	Activiteit Naam	Beschrijving
BE03-1	Receive car	De wagen wordt terug ontvangen in een filiaal. Dit filiaal krijgt de eigendom.

BE03-2	Determine car state	✓ Verifieer eventuele schade aan de wagen ✓ Verifieer eventuele aansprakelijkheid klant bij schade ✓ Verifieer datum teruggave
BE03-03	Release car	
BE03-04	Determine loyalty points	✓ Verifieer aantal verhuringen in dat jaar aan bepaalde klant ✓ Verifieer soort rental (Indien gratis → geen punten) ✓ Bereken aantal punten op basis van autogroep
BE03-05	Determine total cost	✓ Bereken de uiteindelijke kost die de klant moet betalen.

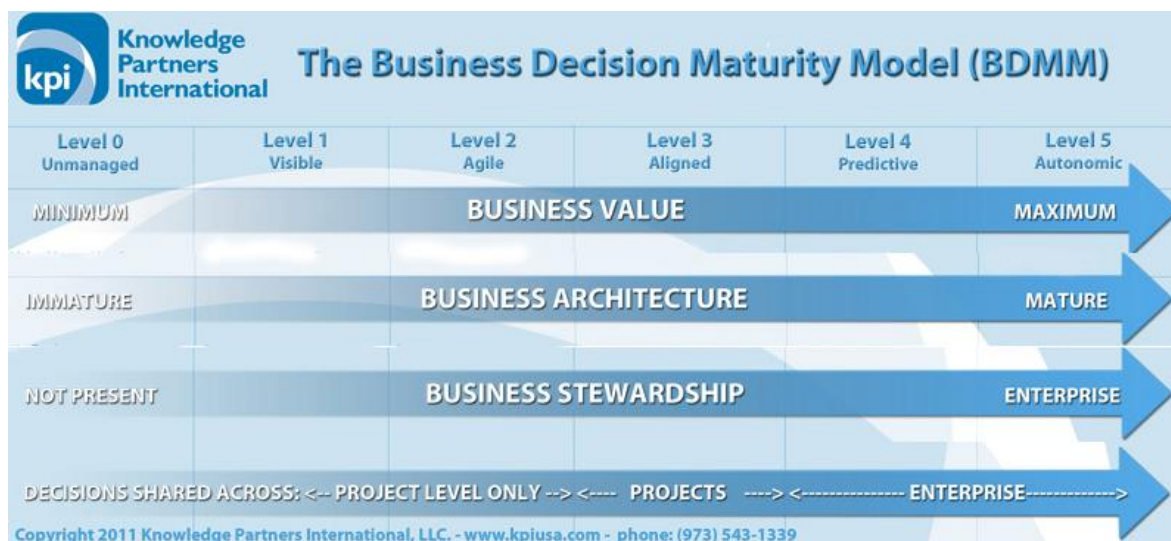
BIJLAGE D: THE BUSINESS DECISION MATURITY MODEL

Het model bevat 3 vectoren voor ieder niveau.

Business Value: De waarde en invloed van een bepaald maturiteitsniveau. Wanneer het bedrijf op een bepaald niveau zit, kan het bedrijf bepalen welk niveau het nastreeft. De andere vectoren zijn afhankelijk van deze vector en bepalen de eigenschappen en criteria van dit niveau. Business waarde zit niet specifiek in een bepaalde regel, maar in het geheel van de bedrijfsbeslissingen die rekening houden met de businessobjectieven, business proces management en andere businessperspectieven. Hiervoor is het van belang te weten wat de impact van de businessbeslissingen op business en IT.

De Business Architecture: vector geeft de maturiteit van het BDM proces binnen de business architectuur weer.

De Business Governance: vector geeft de maturiteit van de governance van het BDM proces weer.



De zes niveaus:

Level 0: Beslissingsmodellen worden niet gebruikt. Beslissingen worden niet gezien als iets dat managebaar is. Business logica zit gewoon vervat in de processen.

Business waarde: Er is dus geen belang aan BDM. Risico's en kosten van verandering zijn hier maximaal.

Business architectuur: Niet aanwezig

Business governance: Niet aanwezig

Level 1: zichtbare geëxternaliseerde beslissingsmodellen en businesslogica op lokaal niveau (bijvoorbeeld: bepaalde project, proces, organisatie-eenheid)

Business waarde: business beslissingen en logica wordt ontdekt, gedocumenteerd en gemodelleerd.

Business architectuur: Zichtbaar, maar nog geen formeel management of controleproces om bedrijfsbeslissingen in de architectuur te integreren.

Business governance: Weinig formele business management aandacht voor het bedrijfsbeslissingen ontdekkingsproces.

Level 2: Flexibel formeel beslissingsmodel waarbij de business beslissingen gekend zijn en waarbij snelle aanpassingen kunnen gebeuren, maar dan wel op lokaal niveau (project, proces, organisatie-eenheid). Businessbeslissingen worden geïmplementeerd in de technologie opdat business analisten en gebruikers ze snel kunnen aanpassen.

Business waarde: Snelle aanpassingen aan het beleid kunnen doorgevoerd worden. Daarnaast is er al een integratie van het beleid, de objectieven en businessbeslissingen. Het risico van het verlies van controle over de business is al minder op lokaal niveau.

Business architectuur: Er bestaat reeds een consistente mogelijkheid om business beslissingen te creëren, managen en te onderhouden. Verschillende stakeholders hebben hier toegang toe.

Business governance: Businessbeslissingen worden gebaseerd op businessobjectieven en hun bijbehorende maatstaven. Bedrijfsbeslissingen worden continu opgevolgd en aangepast.

Level 3: link beleid en decision models is transparant en duidelijk voor de verschillende stakeholders. Er is een hergebruik van beslissingsmodellen. aanpassingen kunnen snel gebeuren. Ditmaal wordt er gewerkt overheen de organisatiegrenzen (projecten, processen, eenheden) .

Business waarde: Bedrijfsbeslissingen worden samengevoegd overheen de processen. Het risico op verlies van businesscontrole is sterk gereduceerd. Ook de mogelijkheid tot het bepalen van de business impact is verbeterd.

Business architectuur: De architectuur wordt geïmplementeerd overheen verschillende processen en businesseenheden. Business planning is gelinkt met consistente business intelligence toepassingen op business unit niveau. Bedrijfsbeslissingen worden ontwikkeld en gemanaged in een goed consistent proces.

Business governance: Helpt mee conflicten op te lossen tussen verschillende bedrijfsbeslissingen die door verschillende stakeholders opgesteld worden. Het zorgt voor consistentie met business objectieven en maakt het mogelijk te reageren op veranderende businesscondities en maatstaven. Ook het senior management wordt hierbij betrokken. Prestatieagenten worden verantwoordelijk gesteld voor het nagaan van de prestaties met betrekking tot de businessbeslissingen.

Level 4: Voorspellend opstellen om veranderingen te anticiperen. Bedrijfsbeslissingen worden ontwikkeld als reactie op geanticiperde of veronderstelde business events en voorwaarden. Er is een brede business scope.

Business waarde: Toekomstige risico's worden verlaagd door het anticiperen. Dit houdt in dat er scenarioplanning en business Intelligence tools gebruikt worden om de optimale business logica te creëren. Hierbij worden bestaande bedrijfsbeslissingen aangepast om organisatiebrede veranderingen op te vangen.

Business architectuur: Biedt de mogelijkheid toekomstige events te modelleren door gebruik te maken van scenarioplanning, decision models en business intelligence tools.

Business governance: Gebruik processen om business scenario's te onderhouden voor toekomstige events. Prestatieagenten moeten simulaties uitvoeren om de juiste logische veranderingen te ontdekken voor een gegeven scenario. Senior management wordt nog sterker betrokken.

Level 5 – Onafhankelijke snelle reactie op veranderingen binnen een brede business scope. Dit gebeurt automatisch zonder menselijke tussenkomst. De reactie is gebaseerd op voorgaande planning en directe aanpassingen gebaseerd op algoritmes en voorspellingsmodellen die de logica begrijpen.

Business waarde: Business risico wordt gereduceerd en business opportuniteiten geoptimaliseerd door de planning en reactie op events te versnellen en te verbeteren. Er is al een zeer goede kennis over decision management die constant verbeterd wordt.

Business architectuur: Geautomatiseerde algoritmes en formules vergelijken de business beslissingen met de veranderende business resultaten. Er vind daarnaast ook een automatisatie plaats van business modellen en scenario's die gecreëerd en getest werden op niveau 4.

Business governance: Er is een constant controle van de accuraatheid, relevantie en optimalisatie van de automatische business beslissingen. Er is op dit niveau een zeer nauwe samenwerking tussen senior management, prestatie-agenten en de governanceraad.

Auteursrechtelijke overeenkomst

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:

De waarde en maturiteit van The Decision Modeler en Operational Decision Manager

Richting: **master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica**

Jaar: **2014**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Voor akkoord,

Vanbrabant, Tim

Datum: **3/06/2014**