

Oracle SOA Suite Beta Test

Architectuur, BPEL en OBR

Oracle heeft korte tijd geleden het Fusion Middleware concept gelanceerd. Onderdeel hiervan is de SOA Suite waarvan aan het eind van 2006 versie 10.1.3. beschikbaar komt. Door deelname aan Oracle's Customer Advisory Board van de BPEL Process Manager is IT-eye rechtstreeks uitgenodigd om deel te nemen aan het Oracle Application Server 10g (10.1.3.1) SOA Suite Beta Program. Deze deelname stelde het bedrijf in staat om kennis op te doen van de nieuwe technologieën in de Oracle tools en tevens gevolgen en mogelijkheden van die nieuwe technologieën voor lopende en nieuwe projecten bij klanten te bepalen.

In een tweetal artikelen wil ik laten zien wat de veranderingen zijn die de nieuwe SOA Suite met zich meebrengt. Gezien de omvang van de SOA Suite beperk ik me tot de meest opvallende zaken. In de twee artikelen leggen we graag de link met de praktijk, en dat doen we op basis van ervaringen bij de Nederlandse Algemene Keuringsdienst (zie kader 'De NAK') uit Emmeloord. IT-eye heeft daar in voorjaar 2006 een toepassing in productie gebracht, gebaseerd op een SOA-architectuur en de Oracle-producten die nu zijn opgenomen in de SOA Suite. In dit eerste artikel ligt de nadruk op architectuur, Oracle BPEL en op Oracle Business Rules (OBR), nieuwe mogelijkheden die vooral tijdens het ontwerpen en ontwikkelen van belang zijn. In het volgende nummer zal aandacht worden besteed aan de infrastructuur componenten Oracle Enterprise Software Bus (ESB), Oracle Web Service Manager (OWSM) en Oracle Registry, nieuwe features die tijdens deployment van belang zijn.

SOA Suite 10.1.3

De SOA Suite die we getest hebben zal naar verwachting eind 2006 beschikbaar komen. De producten in de SOA Suite zijn op veel onderdelen vernieuwd en verbeterd. Daarbij is er veel aandacht besteed aan de integratie van de producten. Officieel bestaat de SOA Suite nu uit de volgende componenten:

- Oracle JDeveloper 10g

- Oracle BPEL Process Manager
- Oracle Web Services Manager
- Oracle Business Rules
- Oracle Business Activity Monitoring
- Oracle Enterprise Service Bus

Wat is een business rule?

Binnen de kaders van de bespreking van Oracle BRE hanteren we de volgende afbakening van het begrip business rule: een business rule bestaat uit drie componenten:

If condition involving facts then action

De condition is een expressie van het type Boolean betreffende facts. Als de expressie true oplevert volgt de actie. Deze actie kan van alles zijn: een foutboodschap, stoppen van de transactie, een mail versturen, een resultaat teruggeven aan de aanroepende service, een status zetten in een tabel. Hierbij maken we onderscheid in twee categorieën rules:

Procesgericht, zoals in:

- Als vastgesteld vochtpercentage < 14% dan niet drogen
- Als vastgesteld vochtpercentage tussen 14% en 17% dan 1 dag drogen
- Als vastgesteld vochtpercentage > 17% dan 2 dagen drogen

Datagericht

- Attribuut, zoals in: Vochtpercentage ligt tussen 0 en 100 %
- Tuple, zoals in: Een aanvraag van het type C-analyse betreft maximaal 15000 kg
- Entity, zoals in: Zuiverheidsonderzoek wordt in duplo uitgevoerd. De twee resultaten (2 tuples) mogen maximaal 3% van elkaar afwijken.
- Inter-Entity, zoals in: De klasse van een perceel is niveau lager dan de klasse van het laagste uitgangsmateriaal op dat perceel

De eerste opvallende zaken komen direct al naar voren tijdens de installatie. Geïntroduceerd in de 10.1.3 Java Editie geeft de Installer de keuze tussen een basis en geavanceerde installatie. In alle varianten verloopt de installatie prettig en snel. De basisinstallatie geeft, zoals verwacht, een aantal standaardkeuzes waarna de installatie begonnen wordt. De geavanceerde optie geeft de mogelijkheid diverse instellingen nader te bepalen. Het belangrijkste verschil tussen de basis en geavanceerde optie uit zich in de manier van deployen van applicaties. De basisoptie plaatst alle applicaties in een home instance, waarbij de geavanceerde optie alle administratieve applicaties, AS Control & JSSO, in de home instance en de overige componenten in een OC4J_SOA instance plaatst.

Architectuur

De toevoeging door Oracle van nieuwe componenten aan de SOA Suite en het verder doorontwikkelen van bestaande componenten kan zeker in een SOA/BPEL omgeving gevolgen hebben voor de gekozen architectuur.

De architectuur bij de NAK (zie figuur 1) is bepaald in het voorjaar van 2005 en gaat uit van basisregistraties die met behulp van services worden ontsloten. Dit zijn zowel basic services (atomaire/CRUD-dataservices) als functionele services, veelal gebaseerd op gegevensregels. Deze services worden tot composite services geaggregeerd met behulp van BPEL. Dit zijn de domein services waarmee een specifieke functie of dienst wordt gerealiseerd. Tot slot hebben we dan de process services die zorgdragen voor de uitvoering van het proces teneinde een

bepaalde dienst of product te kunnen leveren aan de klant. Destijds is bij de opzet van de architectuur nadrukkelijk overwogen om een rule engine in te zetten. De achterliggende reden hiervoor was het feit dat de regelgeving nogal aan veranderingen onderhevig is en deze wijzigingen snel en goedkoop

De services worden tot composite services geaggregeerd met BPEL

moeten kunnen worden aangebracht. Binnen de Oracle stack was destijds echter nog geen rule engine beschikbaar en er is ook niet gekozen een rule engine van een andere leverancier in te zetten.

De samenstelling en mogelijkheden van de SOA Suite 10.1.3 zijn in eerste instantie een bevestiging van de architectuur zoals opgezet ten behoeve van het NAK-project en zoals weergegeven in figuur 1. Maar tegelijkertijd vult Oracle ook een aantal witte vlekken in die ten goede komen aan de architectuur. Zo is er nu wel een eigen rule engine beschikbaar, zijn er meer varianten om services te deployen en zijn er faciliteiten voor geautomatiseerd testen. De SOA Suite als geheel is een heel complete set van functionaliteiten die benodigd zijn om daadwerkelijk een servicegerichte architectuur te kunnen invullen.

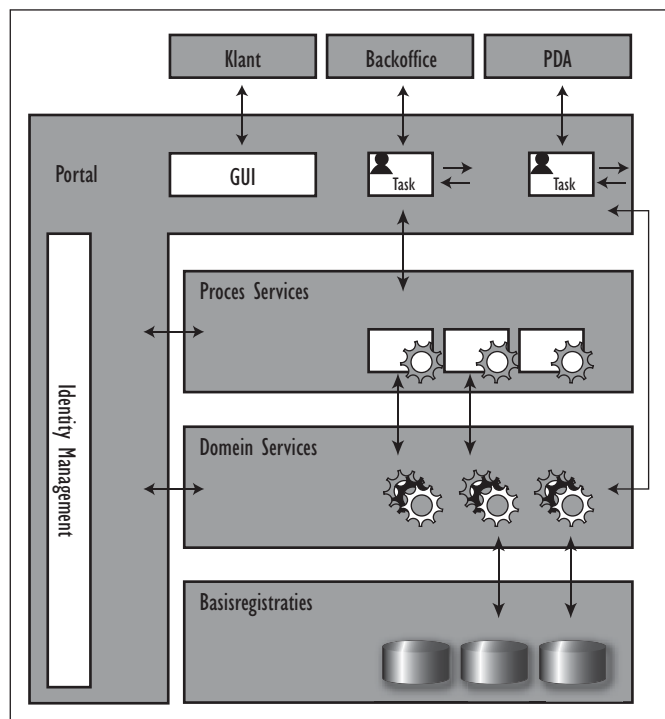
Oracle BPEL

Oracle BPEL bestaat eigenlijk uit twee delen: een *Process Designer* om processen en composite services mee te definiëren en een *Process Manager* om de processen mee te kunnen uitvoeren. Deze keer richten we ons dus op de nieuwe features die tijdens het ontwerpen en ontwikkelen worden benut.

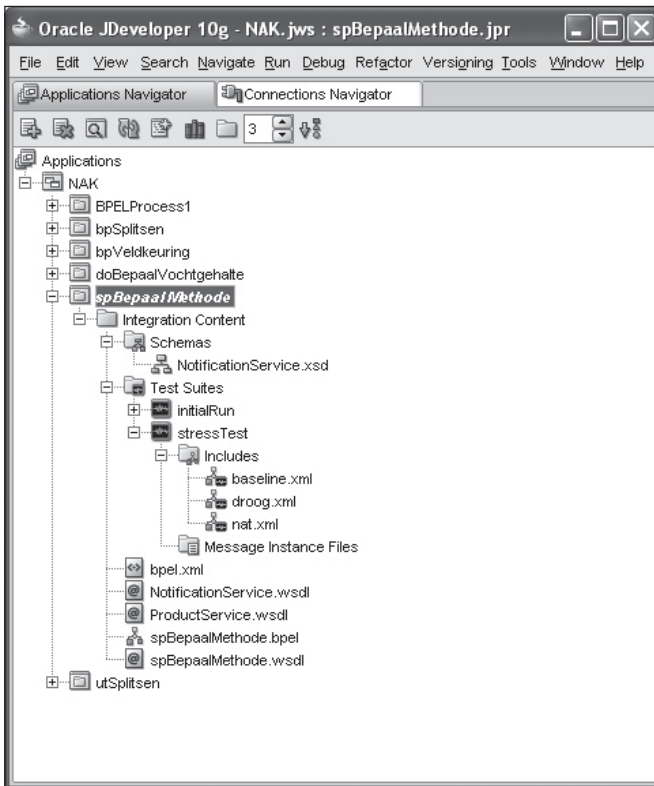
BPEL Process Manager (PM)

De PM is het hart van de SOA/BPEL-architectuur. Deze draagt zorg voor het bewaken van de uitvoering van alle BPEL-processen, het versturen en ontvangen van alle berichten en het vastleggen van de audit-trail. Grote veranderingen zijn hier niet direct waarneembaar. Het meest zichtbare onderdeel van de Process Manager is de BPEL Console. Nieuwe mogelijkheden zijn:

- *Ondersteuning Test Framework*: in de paragraaf BPEL Test Framework zal hier nader op worden ingegaan.
- *Verslaggeving*: voor klanten die niet beschikken over een BAM optie, is het toch mogelijk om rapportages aan te maken van uitgevoerde processen
- *Beveiliging*: BPEL processen kunnen worden voorzien van WS-Security opties (overigens is nu ook WS-Addressing is mogelijk)



Figuur 1. Voor het NAK-project werd deze architectuur opgezet.



Figuur 2. Per BPEL proces kunnen één of meerdere testgevallen worden gespecificeerd.

- **Filtering:** BPEL biedt de mogelijkheid om processen te filteren op basis van datum-uitvoering. Het lijkt een kleinigheid, maar het is wel verschrikkelijk handig bij het traceren van problemen in een omgeving waar duizenden instanties van processen voorkomen.

Wat we graag hadden gezien, maar helaas niet is gerealiseerd:

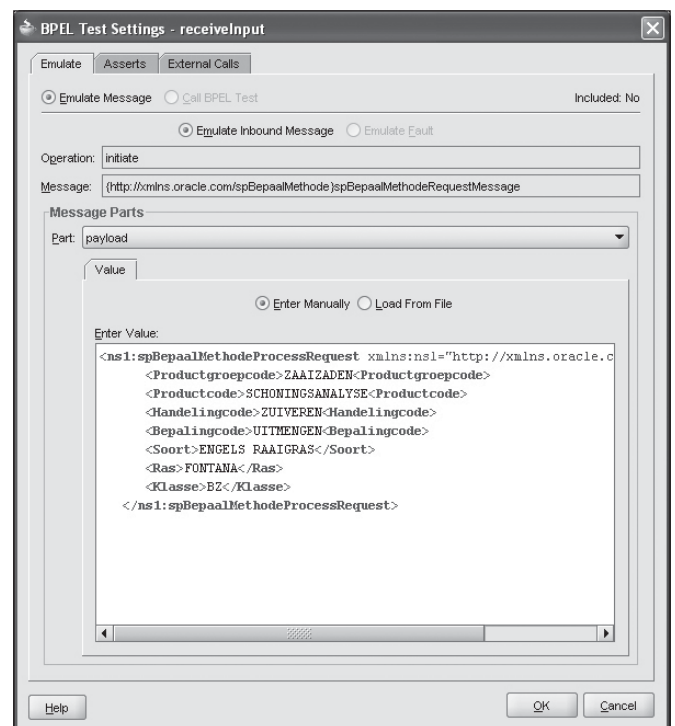
- Het kunnen voorkomen dat een proces 'per ongeluk' wordt overschreven bij het deployen van een aangepaste versie van een procesdefinitie, waardoor alle processen op 'stale' komen te staan en lopende processen worden gestopt. Deze optie is nog niet beschikbaar en dat betekent dat aanpassingen in processen met grote voorzichtigheid moeten worden gedeployed.
- Aanvullende functionaliteiten voor het beheer van de hydratie-database. Deze bevat een berg aan informatie over lopende processen die je goed kunt gebruiken om audit-trail te doen. We blijven op dit vlak nog even gebruik maken van onze eigen aanvullende scripts.

BPEL Process Designer (PD)

De BPEL Process Designer is een plug-in voor JDeveloper (voor Eclipse in juli nog niet beschikbaar). Bij het opstarten van JDeveloper valt direct op dat de Designer flink 'gepimpt' is. Het begin al met het tonen van 'tips' bij het opstarten. Daarnaast veel nieuwe icoontjes, soepele lijnen enzovoort, waardoor het

er allemaal net iets gelijker uitziet dan voorheen. De meest opvallende verschillen:

- **Assign:** de assign is een van de meest gebruikte functies. Hiermee worden variabelen geïnitialiseerd en gekopieerd. Helaas is dit een nogal muisintensieve actie, die in de nieuwste versie minimaal één extra muisclick vereist. Hier is een kans blijven liggen om tot een handiger functie te komen. Wel is het erg prettig dat behalve Copy ook de Append, de Insert-Before en Insert-After nu zichtbaar zijn.
- **Annotations:** in de 10.1.2 versie van de Process Designer was documentatie alleen mogelijk als commentaar in het XML-bronbestand. Hier is nu de 'Annotation' tab bij ieder object aan toegevoegd.
- **Uitgebreide Notificatie mogelijkheden:** het was al mogelijk om vanuit BPEL berichten te sturen middels E-mail, nu is deze selectie aanzienlijk uitgebreid met vrijwel alle mogelijkheden die de hedendaagse technologie ons biedt. Dit alles uiteraard voorzien van wizards die het configureren ervan vergemakkelijken.
- **Business Rules:** het is nu mogelijk om een Decision Service op te nemen die gebaseerd is op Oracle Business Rules of op de Ilog Rules Engine. Dit is een belangrijke optie die bijdraagt aan het integreren van producten binnen de SOA Suite en wordt later in dit artikel verder uitgewerkt.
- **Test Framework:** deze uitbreiding is zo relevant, dat we er in het navolgende verder op in zullen gaan.



Figuur 3. Met behulp van 'Emulate' kunnen inkomende berichten worden gesimuleerd.

De NAK

De NAK keurt in opdracht en onder toezicht van de Minister van Landbouw, Natuur en Voedselkwaliteit zaaizaad en pootgoed van landbouwgewassen. Het werk van de NAK bestaat uit het registreren van de teelt voor het lopende oogstjaar, het uitvoeren van veld- en laboratoriumkeuringen en het uitgeven van certificaten.

Uitdagingen

- De NAK kende afzonderlijke applicaties voor elk gewas, terwijl het proces voor elk gewas veel overeenkomsten heeft
- Vanwege veranderende regelgeving moeten jaarlijks veel applicaties herzien worden
- De applicaties draaiden op verouderde Oracle-software die aan vernieuwing toe was
- De keuringmedewerkers wilden on-line met de centrale systemen verbonden zijn
- Klanten wilden sneller over digitale informatie beschikken

Oplossing

- De gewas georiënteerde werkwijze is vervangen door een procesgestuurde werkwijze, waardoor er geen overlap meer tussen applicaties is
- Een service oriented architecture op basis van Oracle zorgt dat veranderingen en koppelingen met andere partijen en applicaties makkelijker te realiseren zijn
- Met Oracle BPEL zijn de processen gestroomlijnd en is informatie beter ontsloten
- Klanten hebben via het internet toegang tot applicaties, met lagere kosten als gevolg
- De buitendienst heeft voortaan vanaf elke locatie middels PDA's toegang tot de centrale systemen

BPEL Test Framework (TF)

Het Test Framework is een onderdeel van de Process Designer dat erop gericht is om BPEL-processen zoveel mogelijk geautomatiseerd te testen. Het framework stelt je in staat om:

- input naar partnerlinks te valideren
- proceslogica te controleren
- te bepalen of foutafhandeling volledig is

Het framework is opgenomen in het JDeveloper project. Per BPEL proces kunnen één of meerdere testgevallen worden gespecificeerd. Deze zijn daarmee integraal onderdeel van het project en worden daarom ook met het project gedeployed (zie figuur 2).

Er zijn twee manieren om testgevallen aan te maken. Dit kan op basis van functionele specificaties gebeuren, maar er is ook

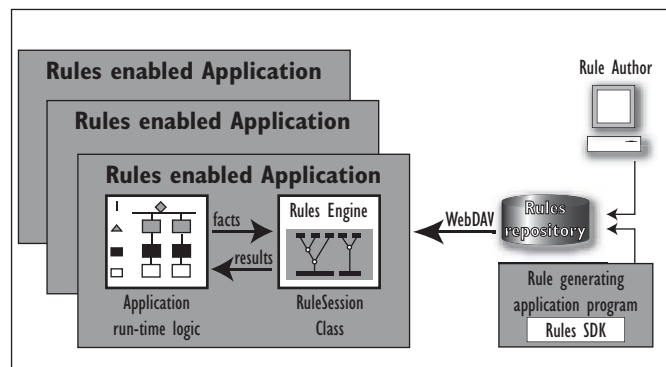
een 'snelle' manier. Het is namelijk mogelijk om de inputs van een uitgevoerd proces te bewaren. In de BPEL Console is hiervoor een specifieke functie opgenomen. Hierbij kan een proces-instance worden bewaard als testgeval. Het mooie daarbij is, dat het ook mogelijk is om een gehele cyclus in samenhang op te slaan. Hiermee kan dan ook de integratie over verschillende BPEL's (en services!) worden getest. Het aldus bewaarde bestand kan worden geïmporteerd in JDeveloper, waarna het als test kan worden hergebruikt.

Als het testgeval in JDeveloper wordt geopend, wordt in één oogopslag duidelijk wat er al dan niet wordt getest. Daarmee kan ook eenvoudig worden bepaald of er voldoende testen zijn gedefinieerd om alle gewenste functionaliteit te testen. Hiermee is een testscenario voorhanden, maar moeten de invoer en uitkomst nog bepaald worden. Met behulp van 'Emulate' kunnen inkomende berichten worden gesimuleerd. Uitgaande berichten (naar andere services) worden middels externe calls gesimuleerd. Daarmee kan het hele proces worden nagebootst, (zie figuur 3). Het naspelen alleen is niet voldoende. Per gespecificeerd testgeval zal ook nog bepaald moeten worden of de uitkomst voldoet. Hiervoor is een zogenaam-

Bij het opstarten van JDeveloper valt direct op dat de Designer flink 'gepimpt' is

de 'Assertion' beschikbaar. Hier kan per element, per bericht een controle voor worden opgenomen.

Als eenmaal alle testgevallen zijn gespecificeerd, moeten ze worden gedeployed. Hiervoor is een speciale functie beschikbaar, de BPEL Test Deployer. Eenmaal gedeployed, kunnen de testen net zo vaak worden uitgevoerd als wenselijk, zonder dat er daadwerkelijk berichtverkeer hoeft plaats te vinden.



Figuur 4. Oracle Business Rules bestaat uit drie componenten: de Rule Engine, de Rule Author en een SDK.

In onze aanpak bij het NAK-project maken we al gebruik van Continuous Integration. Alle sources worden regelmatig en automatisch getest. Hiervoor gebruiken we JUnit en ANT. Het zal niet verbazen, dat het BPEL Test Framework hier naadloos in past. In de build.xml die standaard bij ieder BPEL project aanwezig is, kan gespecificeerd worden welke test(en) bij het deployen, maar ook bij de Continuous Integration worden uitgevoerd. Een mooie combinatie dus!

Oracle Business Rules (OBR)

Oracle Business Rules is een nieuw onderdeel van de SOA Suite. Indien goed gebruikt heeft een rule engine het voordeel dat op één plek de rules worden vastgelegd. Bovendien is het definiëren in een engine meer declaratief dan procedureel, wat aanpassingen eenvoudiger maakt. Met OBR kunnen zowel procesgerichte als datagerichte rules worden geïmplementeerd (zie kader 'Wat is een business rule?'). Door het gebruik van een GUI op het aanmaken van condities zouden ook Business Analisten het beheer (moeten?) kunnen uitvoeren.

De rule engine is meer dan een library met een grote set rules (zoals functions in een package). Zo kan de engine ook afhankelijkheden tussen rules evalueren. Bijvoorbeeld:

Rule 1: als persoon langer dan 5 jaar klant dan wordt hij categorie 1

Rule 2: als persoon > 65 dan wordt hij categorie 1

Rule 3: als persoon is categorie 1 dan geef korting van 10%

Via het aanroepen van rule1 of rule2 wordt rule 3 geëvalueerd.

Voor datagerichte rules zijn in de database een groot aantal standaard componenten aanwezig waarmee de rules (eenvoudig) aangemaakt kunnen worden. Voor (inter) entity rules kan bijvoorbeeld gebruik worden gemaakt van db-triggers of materialized views. Anders wordt het wanneer de data in meerdere (heterogene) bronnen is opgeslagen en over deze data inter-entity rules zijn gedefinieerd. Dan volstaat controle op database niveau niet meer en zal een extra laag moeten worden geïmplementeerd bovenop deze databronnen. Zie ook het kader 'OBR vs. CDM Ruleframe'.

Voor procesgerichte rules is OBR uitstekend geschikt om zorg te dragen voor implementatie. Deze rules kunnen dan worden gebruikt in bijvoorbeeld BPEL als decision service, zoals uitgewerkt in het hoofdstuk 'OBR en BPEL'.

Oracle Business Rules bestaat uit drie componenten zoals weergegeven in figuur 4: de Rule Engine, de Rule Author en een SDK. Rules worden gedefinieerd in een eigen taal: de Rule Language. Deze taal heeft een java-achtige syntax. Code in Rule Language kan direct worden ingevoerd, maar beter is het om deze taal te genereren via de wizards in de Rule Author of in de SDK.

Rule Engine

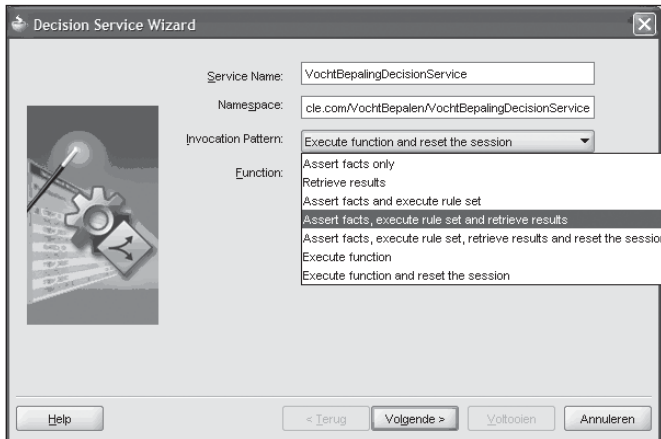
De engine voert de rules uit die zijn gedefinieerd in de repository. De engine is gebaseerd op het product Jess (Sandia Lab). Jess wordt wereldwijd veel gebruikt en is geschikt om zeer veel rules efficiënt uit te voeren. De rule engine is geïmplementeerd als een set java classes. Hierdoor worden de rules uitgevoerd in dezelfde JVM als het Java-programma dat de rule aanroept. De mogelijkheid bestaat om meerdere rule-engines aan te maken op dezelfde repository.

Oracle Business Rules vs. CDM Ruleframe?

Oracle Business Rules is een product dat database onafhankelijk is en als extra laag kan worden ingezet over meerdere (heterogene) bronnen heen. Het is onderdeel van de SOA Suite, dus is het niet verwonderlijk dat het zeer geschikt is voor procesgerichte rules in een BPEL omgeving. De rules die je definieert gaan evenwel niet automatisch af bij het opvoeren, wijzigen of verwijderen van data. Voor bepaalde controles is in de BPEL omgeving access naar de database niet vereist. De aanroep naar de rule-engine moet wel expliciet geprogrammeerd moeten worden als onderdeel van een proces of anderszins. Verder biedt (op dit moment) het tool nog slechts beperkte mogelijkheden voor documentatie in de tool zelf (zo zijn er bijvoorbeeld nu geen comment-velten bij de rules).

CDM-Ruleframe is niet beter of slechter, maar anders. Het is puur gericht op de Oracle database om business rules te definiëren, die op commit-tijd worden afgevangen. Je kunt wel de code hergebruiken om op veldniveau controles te laten afgaan. De bedrijfsregels worden heel dicht bij de data afgevangen: als je eenmaal een bedrijfsregel hebt gedefinieerd en geïmplementeerd, dan hoeft je je daar geen zorgen meer over te maken, de controle gaat vanzelf af. CDM Ruleframe is beperkt inzetbaar want de relatie met Oracle Designer is wel erg intens, hoewel we moeten constateren dat daarmee de mogelijkheden voor documentatie wel weer sterk zijn ingevuld.

Oracle Business Rules is niet (bedoeld als) de opvolger van CDM Ruleframe. Indien alle data in een Oracle database is opgeslagen dan zijn er echter wel goede argumenten om de datagerichte rules behalve in OBR (ook?) in de database te implementeren (bijvoorbeeld met CDM-Ruleframe). Hierdoor ontstaat een robuustere database waarbij controles ook afgaan als data *niet* via de rule engine wordt benaderd. Waar je de knip legt zal als vraagstuk bij het bepalen van de architectuur beantwoord moeten worden.



Figuur 5. De Decision verifiëert de facts, valideert de rule en geeft het resultaat terug.

Rule Author

Dit is een GUI waarmee de rules declaratief kunnen worden vastgelegd in de repository. Met name de conditions zouden hier door business analisten kunnen worden aangemaakt en onderhouden. Dit heeft het voordeel dat een grenswaarde van 14% en 17% uit een van de voorbeelden door de deskundige zelf in de rule kan worden gewijzigd. Aanpassing van een conditie in de repository heeft geen gevolgen voor de services die de rule aanroepen. Deze hoeven niet opnieuw te worden gecompileerd. Het aanmaken van actions kan ook via de GUI. Echter vanwege het technische karakter hiervan zal dit voornamelijk door de (Java)-programmeur gedaan gaan worden.

SDK

Dit is een API waarmee direct de repository kan worden benaderd en rules kunnen worden uitgevoerd. Hier is niet in detail naar gekeken, we hebben ons beperkt tot de interactie met de componenten binnen de SOA Suite. De SDK is wel nuttig om vanuit bestaande applicaties OBR te kunnen gebruiken.

OBR en BPEL

Natuurlijk zijn we bijzonder benieuwd naar de integratie van de rule engine met de BPEL component uit de SOA Suite. Deze integratie zal bepalend zijn voor vraag of het inzetten van de rule engine de moeite waard is of niet. We bespreken de mogelijkheden aan de hand van een voorbeeld van de NAK, de procesgerichte business rule zoals opgenomen in het kader 'Wat is een business rule'.

De rule wordt in de OBR opgenomen en wordt daarna gebruikt in een BPEL proces welke als taak heeft het bepalen van het aantal dagen dat er gedroogd moet worden. Indien later de reglementen ten aanzien van de percentages veranderen hoeven deze alleen in de rule engine doorgevoerd te worden, zonder dat het BPEL proces wat hier gebruikt van maakt wordt gewijzigd. Om de regel uit ons voorbeeld te implemen-

teren is er eerst een zogenaamde *xml fact* aangemaakt. Deze bevat onder andere het vochtpercentage-attribuut. Hierna kan de daadwerkelijke ruleset aangemaakt worden.

Om nu deze ruleset te kunnen koppelen aan ons BPEL proces moet er eerst in JDeveloper een connectie aangemaakt worden om een verbinding te kunnen maken met de rule engine.

Immers, in JDeveloper kun je nu kiezen uit twee Rule Engine Providers: OBR en Ilog.

Om deze repository te kunnen gebruiken in het BPEL proces is er in de Process Designer een nieuw BPEL component toegevoegd: de Decision Service. Er wordt een wizard gestart indien deze component in het BPEL proces gesleept wordt. Met deze wizard kan, naast het opgeven van een naam en de te gebruiken namespace ook een invocation pattern geselecteerd worden. In dit voorbeeld kiezen we voor het verifiëren van de facts, het valideren van de rule en het teruggeven van het resultaat, zoals weergegeven in figuur 5.

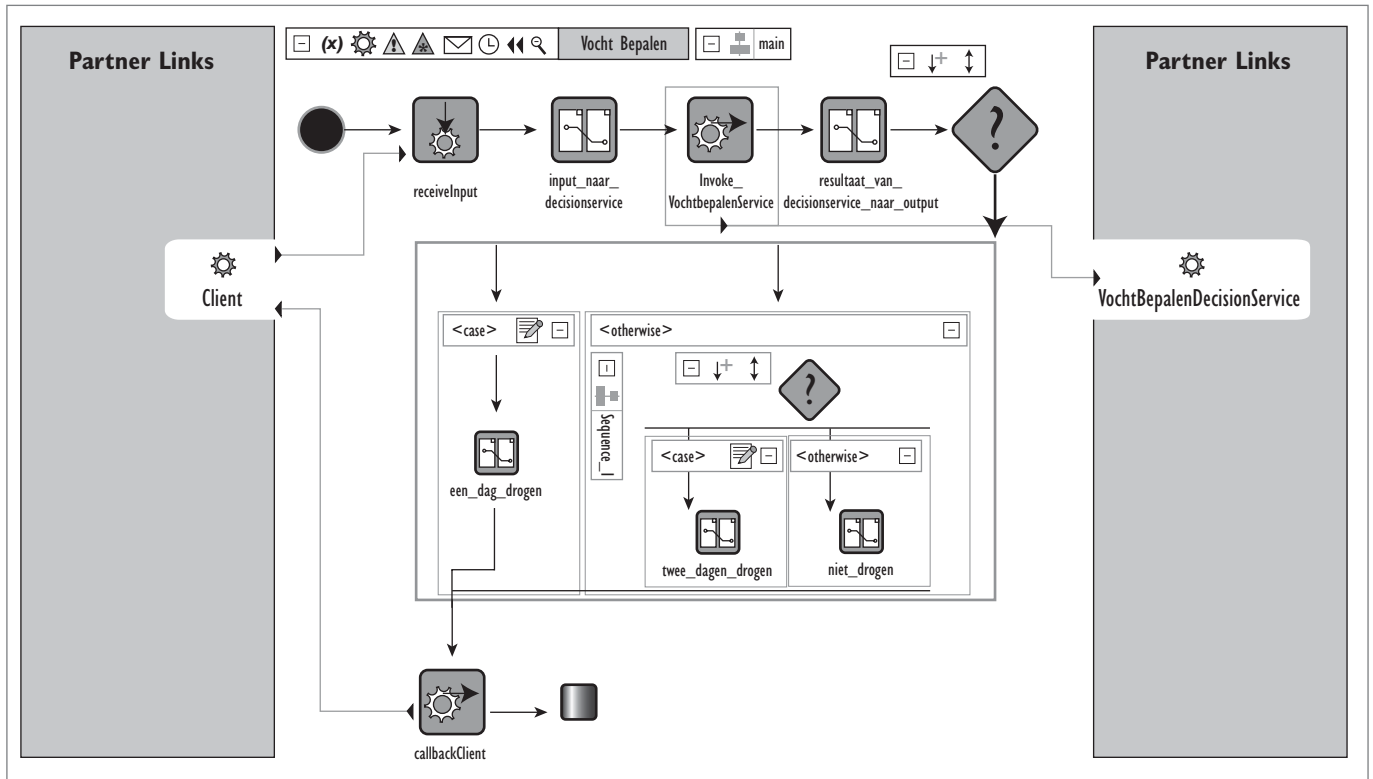
Na het doorlopen van deze wizard is er een nieuwe *partner* aangemaakt, de 'VochtBepalenDecisionService'. Het BPEL proces krijgt als input variabele van de client een object met een

Voor procesgerichte rules is OBR uitstekend geschikt om zorg te dragen voor implementatie

vochtpercentage attribuut. Door middel van de gebruikelijke assign wordt deze variabele in de payload van de decision service gezet. Vervolgens kan de DecisionService aangeroepen worden met een 'Decide' component die eveneens nieuw is toegevoegd in 10.1.3.

Deze service zal dus de rule engine aanroepen met de facts die in de payload zijn gezet. Het resultaat zal een getal 1, 2 of 3 zijn. Dit resultaat wordt verder verwerkt door een standaard BPEL switch component. De resulterende expressie wordt aan de aanroepende client teruggegeven. Hiermee is een eenvoudige procesgerichte rule geïmplementeerd in de rule engine en wordt deze toegepast in een BPEL-proces. Het resultaat in het BPEL-proces is weergegeven in figuur 6.

Een rule engine zou je het liefst willen inzetten om eindgebruikers of business analisten in staat te stellen zelf hun business rules te onderhouden zonder dat je daar zelf specifieke functionaliteiten voor hoeft te realiseren. De functionaliteit zoals die nu beschikbaar komt is een stap in de goede richting, maar nog niet geschikt voor de 'echte eindgebruiker'. Wel betekent dit dat business-logica en proces-logica nog beter van elkaar gescheiden kunnen worden.



Figuur 6. Het resultaat in het BPEL-proces van een eenvoudige procesgerichte rule geïmplementeerd in de rule engine

Samenvattend

De componenten die tijdens de ontwerp en ontwikkelfase worden ingezet zijn op belangrijke punten verbeterd en uitgebreid. De vernieuwingen richten zich niet uitsluitend op de functionaliteit, maar zoals het Test Framework ook op een betere ondersteuning voor het ontwikkelteam. De integratie van de nieuwe technologieën is direct al op een behoorlijk niveau en biedt mogelijkheden om de verschillende componenten effectief in te zetten.

Hebben we in dit artikel vooral gekeken naar de wijzigingen die de nieuwe versie voor de ontwikkelaar met zich meebrengt, in ons volgende artikel zullen we vooral kijken naar de consequenties voor beheer en exploitatie. Met name zullen we dan kijken hoe de ESB, WSM en de Registry passen in de gehele architectuur. Belangrijke vraag daarbij is: welke rol speelt ieder van deze hulpmiddelen daarin. Waar worden processen en (web)services gedeployed? Waar vindt beveiliging plaats? Welke functionaliteit draagt zorg voor transformatie en routing? Al deze vragen zijn van belang om een goede implementatie van de Oracle SOA Suite te kunnen uitvoeren.

Mike van Alst is werkzaam als Senior IT Architect bij IT-eye (mike.van.alst@it-eye.nl). Met dank aan het SOA Beta Program Team van IT-eye voor de geleverde testinspanningen en tekstbijdragen.