

# Oracle SOA Suite Bèta Test (2)

## Beheer en deployment

**In het vorige artikel (Optimize nr. 4, 2006) hebben we vooral aandacht besteed aan de componenten uit de Oracle SOA Suite 10.1.3. die een belangrijke rol spelen tijdens het ontwikkelen van functionaliteit Dit tweede en laatste artikel in de reeks over de bevindingen met het Oracle SOA Suite Beta Program is geheel gewijd aan de componenten die beschikbaar zijn voor het beheer en deployment.**

Het zal voor iedereen duidelijk zijn dat het werken in een SOA omgeving makkelijk leidt tot een toename van de beheer inspanning. Om te beginnen hebben is er sprake van meerdere omgevingen (onder andere BPEL PM, AS Container(s), Database(s), OID, Portal, ...). Nog veel meer impact komt voort uit het feit dat er meer componenten zijn (waaronder veel services). Alles wat bijdraagt aan het vereenvoudigen van het beheerproces is derhalve meegenomen. In de SOA Suite zijn een aantal nieuwe onderdelen opgenomen die in dit artikel ieder besproken worden:

- Oracle Web Service Manager (OWSM)
- Oracle Registry
- Oracle Enterprise Service Bus (ESB)

### Oracle Web Service Manager

Binnen een Service Oriented Architecture (SOA) zijn webservices de voor de hand liggende (maar niet enige) manier om services te realiseren, waarmee verschillende business functionaliteiten en processen worden verzorgd. Het gebruik van webservices brengt nieuwe uitdagingen voor het beheer met zich mee. Binnen een SOA zijn vaak meerdere services actief die allemaal hun specifieke eisen hebben op het gebied van beschikbaarheid, loggen van in- en uitgaande berichten en beveiliging. De Oracle Webservice Manager (OWSM) is dé tool binnen de Oracle SOA Suite om deze beheertaken in te vullen. Hiermee is het mogelijk om alle services binnen een SOA op één centrale plek te beheren.

De OWSM legt als het ware een transparante laag over de webservices heen, waarmee al het in- en uitgaande data van de

webservices gecontroleerd en beheerd kan worden. Het beheer van de webservices wordt geregeld aan de hand van policies. De OWSM bevat een uitgebreid policy management systeem om policies te creëren en te beheren. Policies kunnen vervolgens worden toegewezen aan een enkele webservice of aan een groep van webservices. Daarnaast bevat de OWSM functionaliteit om webservices en de bijhorende policies te monitoren vanuit een centrale plek.

Het policy management systeem scheidt operational policy's van de applicatielogica. Bijvoorbeeld: security-aspecten worden

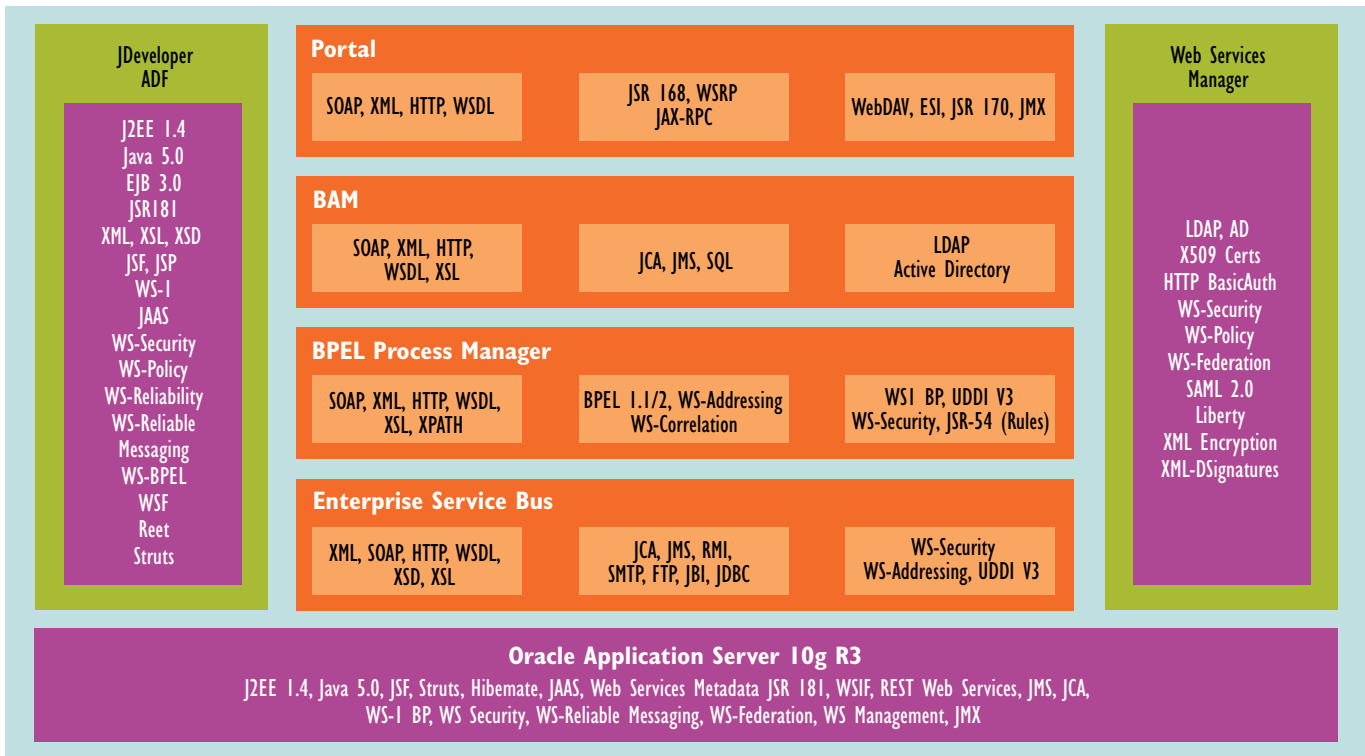
**Binnen een SOA zijn webservices de voor de hand liggende manier om services te realiseren**

in de OWSM met een policy gedefinieerd die op zijn beurt weer wordt toegewezen aan de juiste webservices. Op deze manier is de OWSM verantwoordelijk voor het policy management in plaats van gecodeerde policy's in de webservice implementatie zelf. Deze scheiding zorgt ervoor dat de OWSM een 'non-intrusive' webservice management tool is. Verandering aan de policies en toevoegingen van policies kunnen run-time doorgevoerd worden, zonder dat het gedrag van de onderliggende service verandert. De OWSM bestaat uit verschillende componenten:

#### Policy Enforcement Points

De OWSM past policies toe op webservices via Policy Enforcement Points (PEPs). De OWSM onderscheidt twee verschillende typen PEPs: gateways en agents.

Agents zijn SOAP interceptors die berichten opvangen en de gedefinieerde policies erop toepassen. Agents komen ook weer voor in twee smaken: client en service agents. De client agents



Figuur 1. De Oracle SOA Suite (met vermelding van de ge-supporte standaarden).

'leven' in de container waar de client applicatie is gedeployed en service agents 'leven' in de container waar de webservice is gedeployed en kunnen afhankelijk van de SOAP interception framework als een SOAP handler of servlet filter geïmplementeerd worden. Een agent kan maar een enkele policy toepassen op de webservices, namelijk die aan de agent is gekoppeld. Agents pakken automatisch alle inkomende berichten op naar

gateway kunnen ook webservices van buiten het domain van de organisatie beheerd worden.

#### OWSM Policy Manager

De policy manager beheert alle policies op een centrale plek. De OWSM Policy Manager regelt de levens-cyclus van de policies, de registratie van agents, gateways en webservices.

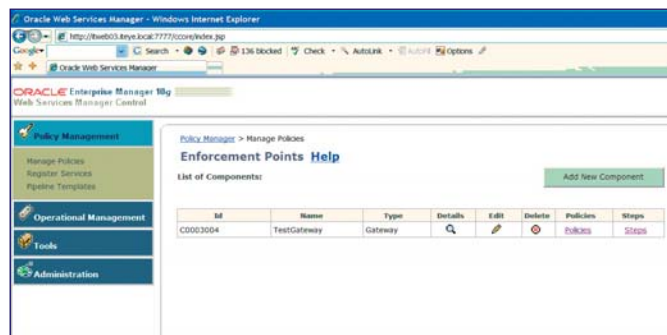
#### OWSM Monitor

De OWSM Monitor beheert het dataverkeer van en naar de webservices en stelt de beheerder in staat om rapporten te maken om overzichtelijk inzicht te krijgen in de data die OWSM Monitor heeft verzameld. Bijvoorbeeld, gateways en agents sturen ruwe data naar de OWSM Monitor omtrent verschillende aspecten, zoals beschikbaarheid van webservices,

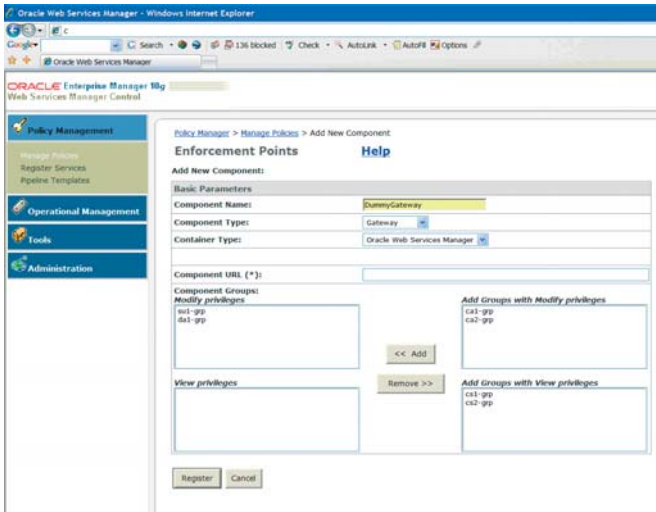
## Het policy management systeem scheidt operational policy's van de applicatielogica

een webservice, waardoor de webservice niet geregistreerd hoeft te worden bij de agent.

Gateways zijn SOAP/XML intermediaries (proxy) waar alle request en response berichten van een webservice heengaan. Webservices moeten bij de gateway worden geregistreerd, waarna de webservices alleen nog via de url van de gateway te benaderen zijn. Hierdoor worden de interne urls afgeschermd van de buitenwereld. Waar een agent een enkele policy op meerdere webservices toepast, kan een gateway een specifieke policy toepassen op elk geregistreerde webservice. Met een



Figuur 2. Aanmaken PEP.



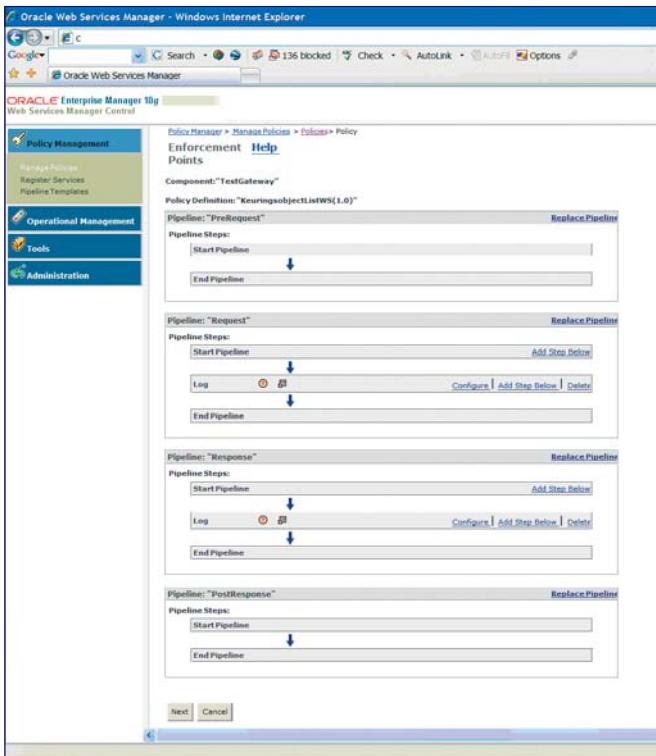
Figuur 3. Aanmaken gateway.

schending van security regels enz.. De beheerder kan met de OWSM Monitor inzicht krijgen in deze ruwe data.

#### Webservice Manager Control

Dit is de console (web-based) waarmee een beheerder alle componenten van OWSM kan beheren en gebruiken.

Met de positionering van deze componenten is er voldoende context om een praktijkvoorbeeld te volgen aan de hand van een case bij de NAK (zie kader vorige uitgave). Dit voorbeeld



Figuur 4. Aanmaken policy.

beschrijft hoe het policy management van de OWSM past in een SOA omgeving.

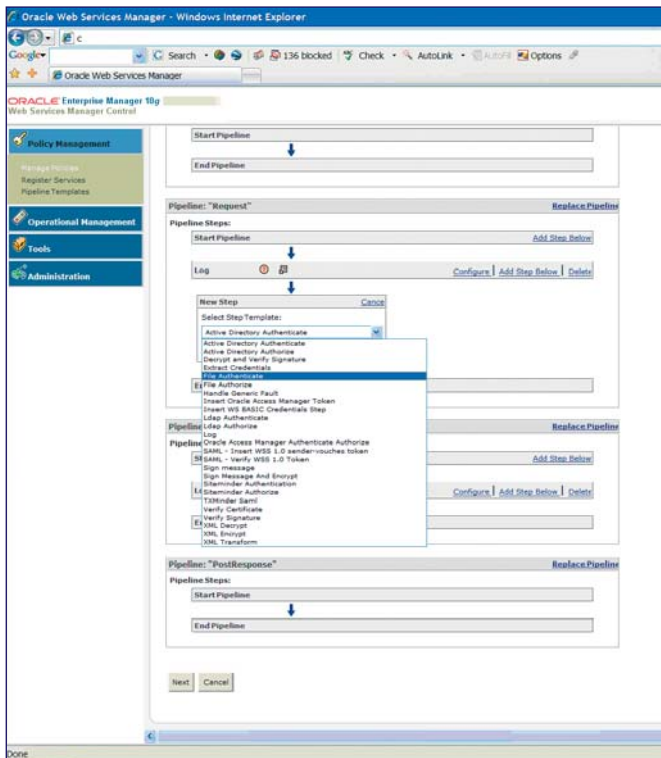
## Eerste stappen met de OWSM

In de Webservice Manager Console zien we aan de linker kant de opties, en in het midden de bestaande PEPs, waar we middels een wizard een nieuwe PEP kunnen aanmaken (figuur 2). We kiezen er voor om een nieuwe gateway toe te voegen, waar we een webservice bij kunnen registreren. Het aanmaken van een gateway gebeurt middels een eenvoudige wizard (figuur 3), waarin kan worden aangegeven welke personen/groepen de gateway kunnen aanpassen. Oracle adviseert om gateways (en agents) altijd een representatieve naam te geven, zodat ze eenvoudig zijn te identificeren. Een goede naamconventie is hierbij van belang!

Nu we een eenvoudige gateway aangemaakt hebben, kunnen we een webservice bij de gateway registreren. In het hoofdmenu kiezen we voor "Register services", waarna we in het vol-

***De Registry is een pure UDDI,  
een Gouden Gids voor alle  
beschikbare webservices***

gende scherm de zojuist aan gemaakte gateway selecteren om een webservice bij te registreren. Het registreren van een webservice verloopt ook weer middels een wizard. Wat opvalt is dat naast het HTTP(s) protocol ook webservices aangemeld kunnen worden die via andere protocollen communiceren (bijv. JMS). Tevens kan er bij het HTTP(s) protocol aangegeven worden of er een custom error pagina getoond dient te worden wanneer een HTTP service een fout code retourneert. Zoals al eerder is aangegeven kan een gateway aan iedere geregistreerde webservice een specifieke policy toewijzen. Vanwege dit feit, wordt een policy in de console ook aan een geregistreerde webservice toegewezen. Wanneer we kiezen om een policy te definiëren voor de zojuist aan gemaakte webservice verschijnt het scherm zoals in figuur 4 te zien is. Een policy in de OWSM bestaat uit vier pipelines (pre-request, request, reponse en post-response). Aan iedere pipeline kunnen dan afzonderlijke policy steps toegevoegd worden om zaken als encryptie, logging of authenticatie en autorisatie te regelen (figuur 5). De OWSM biedt tevens de mogelijkheid om pipeline templates aan te maken die op deze manier herbruikt kunnen worden in verschillende policies. Op deze manier kunnen business policies op een eenduidige manier uitgedragen worden. Nadat de policy is gedefinieerd en definitief is opgeslagen, is de webservice middels de gateway beheerd door de OWSM.



Figuur 5. Policy-steps.

De OWSM is een prima tool om het groeiende aantal webservices binnen een SOA te beheren en de geldende business policies, zoals security en encryptie, op een eenduidige manier uit te dragen. Het policy management systeem ziet er veelbelovend uit en biedt veel opties. Daarnaast biedt de OWSM veel controle en reportingen mogelijkheden (alarms, statistische rapporten), om alle webservices en de daarvoor geldende policies te monitoren.

## Oracle Registry

De Registry is een pure UDDI (Universal Description Discovery and Integration), als het ware een Gouden Gids voor alle beschikbare webservices. Naarmate de functionaliteit in een SOA omgeving groeit, neemt het aantal services (enorm) toe. Om hergebruik van services te kunnen faciliteren is het noodzakelijk dat iedereen (iedere service-consumer: ontwikkelaars en beheerders) zicht heeft op de aanwezige services, de werking ervan, de beschikbare interfaces en de status ervan. Op dit moment is het zo, dat het merendeel van de services gerealiseerd wordt met webservices. Hoewel de UDDI veel meer functionaliteiten bevat, is de UDDI vooral een 'naslagwerk'. Aan alle beschikbare services kunnen 'keywords' worden toegevoegd, waarmee ze eenvoudig kunnen worden getypeerd. Evenzeer is van belang, dat in de Registry niet alleen de beschrijving zichtbaar is, maar ook de berichten die uitgewisseld worden. Hiermee ontstaat één plaats waar alle relevante informatie van de aanwezige services beschikbaar is.

Waar een UDDI ook een belangrijke rol kan spelen, is in het gebruik van services van derden. Deze kunnen hier, voorzien van alle benodigde autorisatie informatie, worden gedefinieerd, waarna ze vanuit andere services, of vanuit BPEL, eenvoudig kunnen worden aangeroepen. Hierbij is het ook mogelijk om gevoelige services af te schermen, waardoor deze alleen voor geautoriseerde gebruikers zijn in te zien.

Door de UDDI centraal te plaatsen in het ontwikkelproces, worden consequenties van veranderingen sneller inzichtelijk. Ontwikkelaars en beheerders kunnen zicht krijgen op veranderingen in interfaces. Voorwaarde hiervoor is, dat alle services daadwerkelijk worden geregistreerd.

## Oracle Enterprise Service Bus

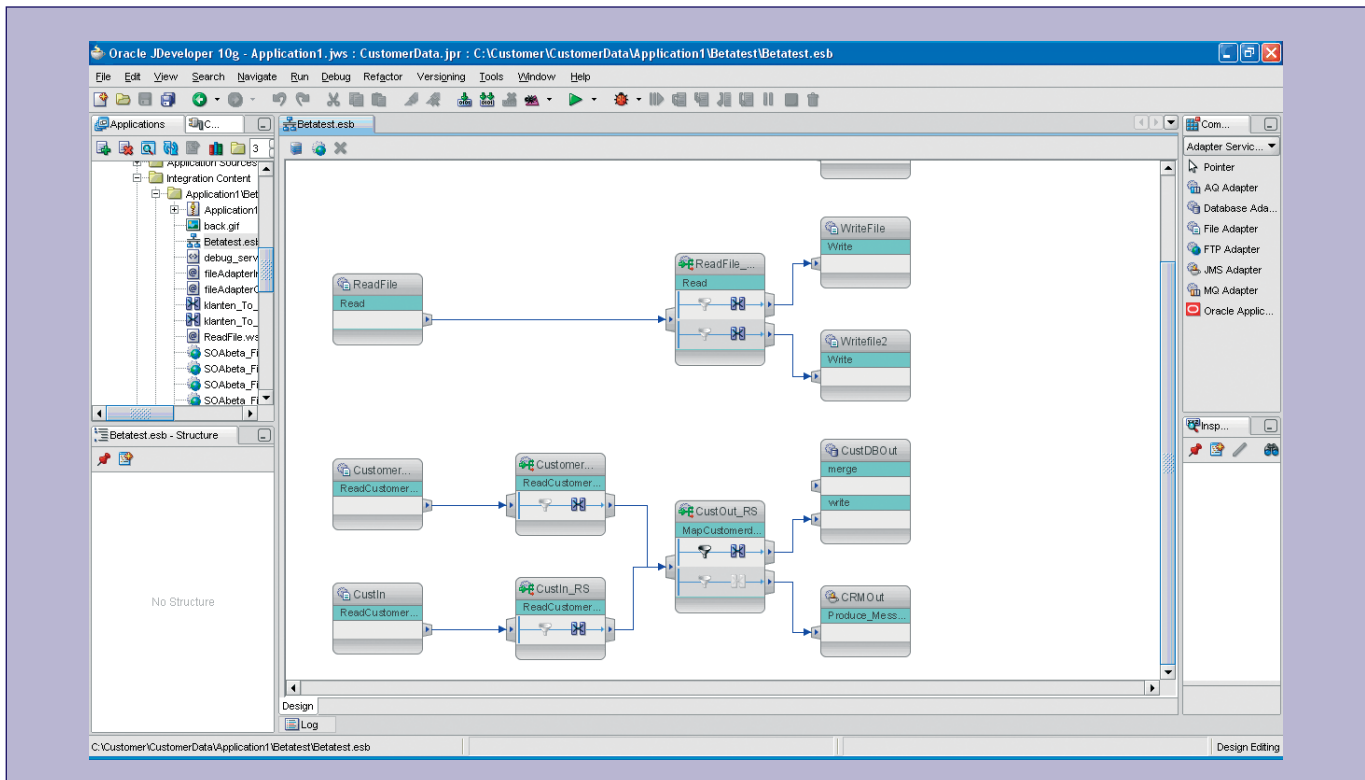
In veel organisaties komen vele soorten van technische infra-structuren voor. Vrijwel geen enkele organisatie beschikt over slechts één platform. De Enterprise Service Bus (ESB) maakt interactie tussen alle verschillende service componenten die op de verschillende platformen voorkomen mogelijk. Daarmee is

## De keuzes van IT-eye

Wat ons betreft is het vrij simpel. De SOA Suite 10.1.3 biedt een vrijwel complete set hulpmiddelen waarmee alle mogelijke vormen van servicegerichte architecturen kunnen worden gerealiseerd. We zien graag verdergaande ontwikkelingen ten aanzien van integratie, maar zijn erg tevreden over de vorderingen die Oracle in de afgelopen twee jaar heeft geboekt. Uiteraard blijft het noodzakelijk om met een kritische blik te blijven kijken. Wat zijn de hulpmiddelen die je op welk moment inzet? In een homogene omgeving is een ESB niet per sé nodig, maar deze kan wel een bijdrage leveren aan de betrouwbaarheid van de omgeving. Een Web Services Manager is in een lokale omgeving niet direct nodig, maar kan toch een zinvolle toevoeging zijn. Zo zijn er voor alle vormen voor- en nadelen te bepalen, die situationeel ingevuld moeten worden. Het maken van deze keuze(s) is dan onderdeel van de voorbereidingsfase die wij bij al onze projecten doen. In één van onze lopende projecten kiezen we bewust voor het niet inzetten van de ESB, omdat het een zeer procesgestuurde en homogene omgeving is. Het gebruik van webservices met BPEL processen is hierbij ruimschoots afdoende.

Op grond van onze ervaring in de afgelopen twee jaar, durven wij te stellen dat een SOA omgeving niet langer een theoretische mogelijkheid is, maar praktisch zeer uitvoerbaar is. De beloofde toename in flexibiliteit tegen lagere kosten is daarbij direct aantoonbaar.

De Oracle SOA Suite 10.1.3 is een goede basis voor een service gerichte architectuur!



Figuur 6. Transformaties ten bate van ESB.

een ESB feitelijk een 'enabler' voor een SOA omgeving in een hybride technologische infrastructuur.

Een zeer bekend voorbeeld van een service bus is CORBA, wat feitelijk niet meer is dan een communicatie infrastructuur voor individuele object instanties. De CORBA infrastructuur stelt een object in staat een ander object op de bus te lokaliseren en operaties van dat object aan te roepen. In die zin is CORBA in essentie een symmetrische bus: geen verschil tussen clients en servers. CORBA bestaat al geruime tijd, maar heeft nooit een groot draagvlak gekend. De Oracle ESB daarentegen is meer hybride van aard. De OESB kan ingezet worden als 'symmetrische bus', maar vooral ook als een 'hub-spoke' platform. Hier herkennen we direct de functionaliteit zoals we die kennen uit InterConnect.

Wat heeft de ESB te bieden? De basisfuncties uit de ESB zijn:

- het aanbieden van connectiviteit via een serie van mechanismen, zoals JCA, JMS, SOAP en WSIF
- het uitvoeren van message transformation
- het uitvoeren van message routing

Het is opvallend, dat de door de ESB gebruikte adapters exact dezelfde zijn als die gebruikt worden in BPEL. Dit geeft aan dat er een zekere mate van overlap is tussen de ESB en BPEL. Naast de (volledige) uit InterConnect bekende functionaliteit in de ESB, zoals het maken van een centraal datamodel, het verzorgen van transformaties van en naar het centrale datamodel

en het vaststellen van routes tussen verschillende nodes, bevat de ESB interessante mogelijkheden. Zo kunnen niet alleen koppelingen worden gemaakt op basis van adapters, maar alle services (WSIF, webservices, EJB, enzovoort) kunnen worden geregistreerd in de ESB. Zodra ze zijn geregistreerd, wat over het algemeen niet meer is dan een simpele muisklik in JDeveloper,

**Door de UDDI centraal te plaatsen in het ontwikkelproces, worden consequenties van veranderingen sneller inzichtelijk**

kunnen ze in het hele patroon van transformaties en routeringen worden opgenomen (zie figuur 6). Hierdoor wordt de ESB dé spil in een hybride SOA omgeving (zie kader Orchestratie van Services).

Voor de gebruiker van een service (consumer) is het derhalve niet meer nodig om te weten waar en hoe een service is geïmplementeerd: alle services komen voor in de ESB. Door slim gebruik te maken van logische verwijzingen ontstaat een omgeving die het gehele traject van ontwikkelen, testen en exploiteren ondersteunt.



## Orchestratie van Services

De toepassing van een service gerichte architectuur wordt ervaren als een voorwaarde voor het implementeren van procesbesturing op basis van BPEL. Uitgaande van de juistheid van deze stelling is het tegenovergestelde echter niet waar. Het inzetten van BPEL is op geen enkele manier een voorwaarde voor het succesvol implementeren van functionaliteit in een SOA omgeving.

Het al dan niet inzetten van BPEL maakt wel uit voor de manier waarop services worden georchestreerd. In een SOA omgeving is er vrijwel altijd sprake van orchestratie van services: aan het samenstellen van nieuwe services (composite services) op basis van beschikbare andere services (basic services). Dit samenstellen wordt door verschillende componenten van de Oracle SOA Suite ondersteund en zodra iets op verschillende manieren kan is het van belang de juiste keuze te maken:

### *BPEL Process Manager*

Oracle BPEL PM biedt een prachtige grafische interface voor orchestratie in het algemeen. Voor alle versies tot aan 10.1.3. ook de enige efficiënte tool voor orchestratie. Hoewel de naam een nadruk op procesmodellering doet vermoeden kan BPEL PM prima worden ingezet om ook services te orchestreren. Door de goede support in BPEL PM op het inzoomen op de verschillende niveau's van orchestratie kan snel inzicht verkregen worden in de hele compositie van services. Een aanrader als BPEL PM wordt ingezet in een homogene omgeving.

### *Enterprise Service Bus*

Biedt een meer administratieve toegang tot de services en is beperkter in de mogelijkheden om de composities grafisch weer te geven. Daar staat tegenover dat de ESB je wel in staat stelt om direct ook de relatie naar externe adapters op verschillende niveaus vast te leggen. Indien het inzetten van BPEL niet vereist is een prima hulpmiddel voor het orchestreren van services. Daarnaast de meest verstandige keuze indien in een heterogene omgeving gewerkt wordt.

## Deployment

Behalve de nieuwe en gewijzigde tools, is er ook het nodige veranderd in de wijze van exploitatie. Zo is er een hele grote verandering op het gebied van BPEL, door de mogelijkheid om alle vormen van BPEL's geautomatiseerd uit te rollen naar een of meer omgevingen. In de huidige vorm moeten BPEL's die voorzien zijn van zgn 'usertaken' nog handmatig, of via complexe scripts worden gedeployed. Een bekend probleem hierbij

is de harde verwijzing naar url's die is opgenomen in de BPEL-suitecase (bpel.xml).

BPEL is in de 10.1.3 versie volledig te configureren. Niet alleen is er een suitcase, die het gehele package met zijn afhankelijkheden (waaronder testscripts e.d.) beschrijft, maar ook een build-file, waarin omgevingsafhankelijke variabelen (verwijzingen naar services e.d.) parametrizeerbaar zijn. Het zal niet verbazen dat dit gehele proces eenvoudigweg te besturen is met behulp van ANT.

## Productie

Als we kijken naar de BPEL Process Manager valt eigenlijk op dat er relatief weinig wijzigingen zijn aangebracht (zie vorige artikel). Ten aanzien van exploitatie is de belangrijkste mogelijkheid om vanuit de BPEL Console de historische data van sensoren ('meters' in de BPEL processen) te wijzigen. Gezien het feit dat het gebruik van sensoren leidt tot een stevige groei van de hydratatie database, is dit een zeer wenselijke optie.

Op zich zijn we ook niet verrast door het beperkte aantal wij-

***Er is een duidelijke overlap in functionaliteit tussen BPEL en de ESB, maar ook tussen de Registry en de WSM***

zigingen. De BPEL Process Manager is in de huidige 10.1.2.0.2 versie al erg stabiel en betrouwbaar. Waar het helaas wel aan ontbreekt is uitgebreide functionaliteit voor het onderhoud van de hydratatie database, waarin BPEL de stand van zaken (en alle historie) van een BPEL proces bewaart. Deze heeft nogal de neiging om erg snel te groeien. Om deze groei beheersbaar te houden zal iedere gebruiker zijn eigen onderhoudsscripts moeten maken. Buiten het feit dat deze via het BPEL forum redelijk eenvoudig te krijgen zijn (zie o.a. het BPEL Cookbook) vinden wij dat dit soort functionaliteit tot het standaard functiepalet van de BPEL Process Manager behoort.

Evenzeer is er nog steeds geen goede Administration Guide. De laatste versie is nog steeds een 'draft' versie die nog niet publiekelijk beschikbaar is, terwijl deze informatie is die van wezenlijk belang is voor het tunen van de BPEL omgeving. Iedereen die gebruik maakt of gaat maken van deze omgeving kan ik aanraden om onze weblog te blijven volgen (<http://www.it-eye.nl/weblog>). Daar kun je al onze ervaringen, met bijbehorende oplossingen, terugvinden.

## Conclusies

Met de SOA Suite 10.1.3 heeft Oracle een grote stap gezet in de richting van een volledige set tools, hulpmiddelen en infrastructuur waarmee het realiseren van een SOA architectuur

volledig kan worden ondersteund. Hierbij is niet alleen gekeken naar infrastructurele componenten (application server, process manager), maar ook naar ondersteuning en inrichting van het ontwikkelproces (subversion, BPEL Test Framework, ANT integration).

Met de OWSM zijn we nu ook in staat om beveiliging meer inhoud te geven. Met name als het gaat om webservices die buiten de lokale infrastructuur beschikbaar gesteld (gaan) worden, is de OWSM van groot belang. Hiermee kan de gegevensuitwisseling beveiligd worden (zowel autorisatie als authenticatie). Wat de ESB betreft: BPEL biedt al vele mogelijkheden ten aanzien van transformatie van gegevens. Routing van gegevens is in een SOA niet direct van belang, maar speelt vooral een rol in het aansluiten van een SOA met de buitenwereld en/of de overige (nog-niet-SOA) lokale infrastructuur. Daarnaast is ook duidelijk geworden dat Oracle nog druk bezig

is met de integratie van de verschillende hulpmiddelen. Er is een duidelijke overlap in functionaliteit tussen BPEL en de ESB, maar ook tussen de Registry en de WSM. Dit betekent dat je goede keuzes moet maken over hoe je ieder hulpmiddel inzet. Deze keuzes zijn in die zin arbitrair, dat er niet een éénduidige antwoord bestaat voor deze keuzes. Dit is veelal afhankelijk van de specifieke omgeving, maar wordt ook bepaald door voorkeuren (zie kader De keuzes van IT-eye).

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

---