

Tijdens JavaOne 2007 was een aparte track gereserveerd voor 'Services and Integration'. Twee interessante standaarden binnen dit gebied zijn JBI en SCA. Er waren dan ook verschillende sessies die dit onderwerp behandelden.

JavaOne 2007: Service Oriented Architecture

Onlangs werd aangekondigd dat Service Component Architecture (SCA) versie 1.0 gereed was om overgedragen te worden aan OASIS. Zij zullen de standaard gaan beheren en verder uitwerken. Aan de totstandkoming van versie 1.0 hebben onder andere IBM, BEA Systems, Oracle, Sun, Interface21, SAP, RedHat en SAP meegewerkt binnen Open SOA (<http://www.osoa.org/>). Deze bedrijven zijn ook sponsors van het committee dat binnen OASIS de standaard gaat beheren.

Een paneldiscussie over dit onderwerp ("SCA/SDO and Java Technology: Complementary Technologies That Drive Open SOA Environments") begon met een ultrakorte uitleg van de belangrijkste SCA concepten (zie ook afbeelding 1):

- * SCA Assembly Model. Dit model beschrijft de configuratie van systemen in termen van service-componenten die services gebruiken of implementeren en de connecties (wires) die beschrijven hoe de componenten communiceren.
- * SCA Bindings: Bindings zorgen ervoor dat services aangeboden kunnen worden, en referenties gevonden via specifieke transportmiddelen. Er zijn verschillende bindings; onder andere voor webservices, JMS, stateless session beans.
- * Java-annotaties & API's, Spring, BPEL, EJB, C++ clients. Deze modellen beschrijven hoe verschillende talen en frameworks gebruikt kunnen worden om SCA-componenten te bouwen en SCA-services aan te roepen.

Na deze korte inleiding, mochten de verschillende panelleden uitleggen waarom SCA voor hun bedrijf van belang was. Wat duidelijk werd is dat voor BEA, IBM, Oracle, SAP, en Tibco SCA van

strategisch belang is. De architect van Sun gaf aan dat Sun voornamelijk deelneemt om te zorgen dat de integratie met Java goed geregeld is. Over de vraag of Service Data Objects (SDO) noodzakelijk zijn bij een SCA implementatie verschilden de heren van mening. Beide kunnen onafhankelijk van elkaar gebruikt worden, maar hebben een soort natuurlijke fit: SCA is een manier om composite applicaties in een heterogene SOA omgeving te construeren, SDO een manier om in zo'n heterogene omgeving data te transporteren.. Er zijn verschillende bedrijven die op dit moment een SCA of SDO-implementatie leveren of die het aangekondigd hebben. Een volledige lijst is te vinden op de Open SOA site: <http://www.osoa.org/display/Main/Early+Implementation+Examples+and+Tools>

Java Business Integration (JBI)

Ook op het vlak van Java Business Integration was er nieuws: Sun heeft JSR 312 aangekondigd, beter bekend als JBI 2.0. De bedoeling van deze specificatie is om JBI te verbeteren. Een greep uit de punten die genoemd werden: verbeterde ondersteuning voor clustering, alignment van JBI met SCA, verbeterde ondersteuning voor WS-policy en verbeterde ondersteuning voor Web 2.0. De specificatie wordt niet gesteund door BEA, IBM en Oracle. De eerste twee hebben tegen gestemd, de laatste heeft niet gestemd. Eén van de sessies op JavaOne besteedde aandacht aan BPEL en JBI. De sessie "Why Do I Need Java Business Integration (JBI) When We Have BPEL?" ging uitgebreid in op SOA, BPEL en ten slotte JBI.

SOA - composite applicaties

De introductie begon met SOA. SOA is een archi-

Lonneke Dikmans

(lonneke.dikmans@approach-alliance.com) is managing partner bij Approach Alliance en richt zich voornamelijk op Service Oriented Architecture met daarbij een speciale focus op Oracle/JEE oplossingen

tectuurprincipe waarbij componenten van elkaar zijn losgekoppeld. Nieuwe services kunnen gemaakt worden met behulp van bestaande services. Deze (nieuwe) services vormen composite applicaties. Nu kun je op twee manieren naar deze verzamelingen van services kijken: vanuit het hele IT-landschap binnen een organisatie, of via een specifiek business proces.

Definitie van de applicaties – WS-CDL, BPEL en BPSS

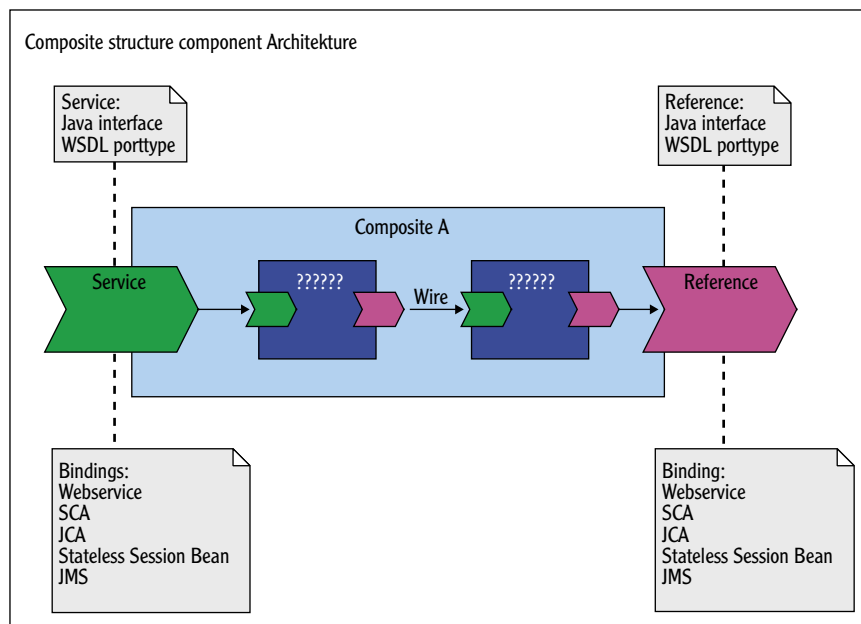
Wanneer je naar het hele landschap kijkt, is WS-CDL een geschikte standaard. WS-CDL beschrijft hoe verschillende services met elkaar samenwerken, hoe de choreografie van de services in elkaar zit. Als de focus op een specifiek business proces ligt, is BPEL een geschikte keuze. In een BPEL-applicatie wordt de volgorde (Orchestration) van aanroepen van specifieke stappen en services (partnerlinks) beschreven. Een derde perspectief is dat van samenwerking: BPSS is een specificatie die samenwerking (collaboration) beschrijft tussen organisaties.

Integratie infrastructuur

De lijm voor deze composite applicaties is de integratietechnologie: in dit geval JBI. JBI definieert een standaard om containers aan elkaar te koppelen, voor business logica en transformaties. OpenESB is een open source implementatie van Sun (<https://open-esb.dev.java.net/>) De integratie is message-based. De messages gebruiken het 'abstracte gedeelte' van een WSDL.

Naast deze sessie, die JBI positioneert in SOA, waren er ook een aantal andere sessies en BOF's over dit onderwerp. Een greep uit het aanbod: JBI 2.0, AJAX en JBI, JEE en JBI.

Afbeelding 1. Service Component Architecture (SCA).



SOA Governance - SGF

Naast ontwikkelen van services en het aan elkaar knopen van verschillende componenten is er nog een aspect van belang bij SOA: governance. De sessie van Michael Wheaton ("Decorating your SOA services with governance enforcement contracts") behandelde het belang van governance in een SOA, hoe deze ontworpen kan worden, de roadmap en ten slotte het framework dat Sun heeft ontwikkeld voor OpenESB: het SOA Governance Framework, ofwel SGF.

Governance is de mogelijkheid om de service-interacties in een bedrijf te organiseren, toe te zien op naleving van afspraken, en opnieuw te configureren. Dit heeft een aantal voordelen voor de business, maar ook voor de IT-afdelingen van bedrijven. Een goede SOA governance oplossing heeft een aantal kenmerken die onder te verdelen zijn in design time, runtime en operations. Design time is een artefact dat de volgende zaken beschrijft:

- * versionering van de service
- * 'lookup and discovery'
- * metadata management (taxonomie en classificaties van de services)

Op runtime-niveau spreekt Michael Wheaton van een governance contract dat de volgende aspecten beschrijft:

- * naleving
- * onderschepping
- * dynamische routing
- * validatie
- * foutcompensatie

Op operations niveau gaat het om:

- * Monitoring en meten
- * Systeem management en beheer
- * BAM en rapportages
- * Dynamisch linken

Er worden drie rollen onderscheiden voor het toepassen van een governance-oplossing: de SOA developer die verantwoordelijk is voor het ontwikkelen van de services en de aspecten van het contract (autorisatie en authenticatie, throughput, lease, throttling et cetera). De operations analyst is verantwoordelijk voor het creëren van SLA-gebaseerde governance contracten en tenslotte de 'governance officer' die 'business policy' toepast op de services. Uiteraard kunnen deze rollen door een en dezelfde persoon uitgevoerd worden, het is analoog aan de J2EE-rollen die Sun onderscheidt.

De manier om een governance contract te implementeren is door het gebruik van aspecten. Deze

kunnen generiek zijn, industrie-specifiek of specifiek voor een bedrijf. Een aantal generieke aspecten zijn: throttling (limiteren van het aantal berichten per minuut), lease (controleren van de expiratedatum van services), monitoring, versioenering, security et cetera.

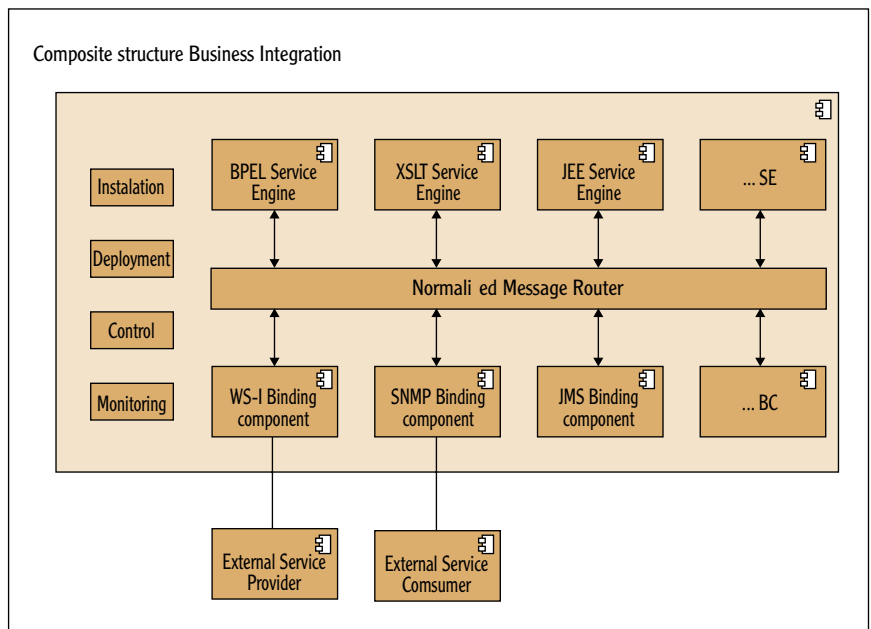
Er werden drie manieren genoemd om de transitie te maken naar een governance oplossing in een SOA omgeving:

1. Service Gateway Model. Alle services in een domein gaan langs deze gateway, en op iedere service wordt een bepaald contract toegepast. Dit is een eenvoudige manier om de contracten te beheren. Het is niet gemakkelijk om bepaalde aspecten uit te breiden, en de contracten worden allemaal op hetzelfde niveau toegepast.
2. Service governance intermediaries. Op verschillende plaatsen en met verschillende granulariteit worden contracten toegevoegd. Dit heeft als voordeel dat er specifieke contracten per service gedefinieerd kunnen worden. Het nadeel is dat het lastiger te beheren is.
3. Standaard gebaseerd. Dit is een benadering waarbij standaarden als JBI/SCA/WCF en policy worden gecombineerd om een 'pluggable' governance infrastructuur te creëren. Dit klinkt vrij vaag, maar Sun en Accenture zijn bezig met een implementatie van een governance-oplossing voor OpenESB. Tijdens de presentatie werd een aantal screenshots van het product getoond. Het heeft een console, houdt rekening met de verschillende rollen, bevat API's en een repository van aspecten voor het creëren van contracten. Het was me niet helemaal duidelijk of het framework straks als open source te downloaden is, of dat het verkocht gaat worden als add-on bij de OpenESB-implementatie. Het product zag er goed doordacht en compleet uit, zeker iets om in de gaten te houden.

Het zou interessant zijn om te kijken wat er aan de kant van SCA op dit gebied geïmplementeerd wordt; SCA kent ook een Policy-specificatie. Die is evenals het SGF gebaseerd op WS-Policy. Helaas heb ik op de conferentie geen sessies gezien over dit specifieke onderwerp.

Conclusie

Naast de besproken sessies waren er sessies over webservice-technologieën (SOAP, JAX-WS, RESTfull webservices API), security, interoperabiliteit, specifieke implementaties (denk aan Salesforce.com, open source SOA implemented) en koppelingen met portals en rich clients. Het enige dat ik echt gemist heb, is een sessie die JBI en SCA naast elkaar zet en vergelijkt. Het blijft



een beetje onduidelijk waar SCA ophoudt en JBI begint en in hoeverre de standaarden complementair zijn. Het lijkt er wel op, dat SCA meer steun geniet onder de belangrijke (Java-)spelers in de markt dan JBI. Opvallend is wel de afwezigheid van Microsoft: zij hebben hun eigen 'standaard': WCF, oftewel Windows Communication Foundation.

SCA of JBI, het is in ieder geval een domein waar Java ontwikkelaars direct (ontwikkelen van aspecten voor governance, ontwikkelen van services en composite applicaties) of indirect (deployen van een SCA component, JBI component, koppelen met een SCA composite) mee te maken krijgen in hun dagelijks werk.

Afbeelding 2. Java Business Integration: OpenESB.