



## Oracle Ace Lonneke Dikmans:

# ‘SOA niet afdwingen vanuit IT’

**Lonneke Dikmans is tot Oracle Ace benoemd. Zij was al Oracle Fusion Middleware Regional Director en daarmee heeft ze dezelfde Oracle eretitels als Lucas Jellema (zie Optimize nr. 1/2006). De titel vormde de aanleiding voor een gesprek dat al snel niet meer over Aces ging maar over Oracle, JDeveloper, SOA en meer.**

Lonneke Dikmans studeerde cognitiewetenschappen, een

Oracle Ace, zie  
[http://www.oracle.com/technology/community/oracle\\_ace/index.html](http://www.oracle.com/technology/community/oracle_ace/index.html)

Oracle Fusion Middleware regional director, zie  
[http://www.oracle.com/technology/community/ofm\\_directors/index.html](http://www.oracle.com/technology/community/ofm_directors/index.html)

bovenbouwstudie tussen psychologie en informatica. Ze begon daarna freelance als useability engineer. Na extra scholing in de Verenigde Staten, onder meer op het gebied van Java, begon ze bij Transfer Solutions als Java-ontwikkelaar en zo kwam ze ook in contact met Oracle-technologie. (Over JDeveloper: ‘JDeveloper verdient echt veel meer aandacht dan het krijgt binnen de Java community.’) Ze werkte daarna enige tijd bij Xebia en heeft sinds een half jaar een eigen bedrijf, dat deel uitmaakt van Approach Alliance, een samenwerkingsvorm voor bedrijven met als belangrijkste speerpunten SOA en BI en een focus op Oracle-oplossingen.

Dikmans: ‘We zijn wel allemaal redelijk gefocuseerd op de business. BI gaat uiteraard sterk uit van de business, en dat is bij SOA ook zo. Het is echter niet alleen zo dat we vanuit de business denken en een *road map* uitstippelen. We doen ook daadwerkelijk projecten om ervoor te zorgen dat dit soort dingen gerealiseerd kunnen worden.’

*Hoe ben je Oracle Ace geworden?*

Dikmans: ‘Ik ben voorgedragen door het OTN-team, dus niet door een specifieke productgroep voor zover ik dat begreep, vanwege mijn bijdrage op forums en de artikelen die ik heb geschreven en vanwege de bijdrage aan de community en de *technical proficiency* die ik gedemonstreerd heb (zie ook de beide URL’s, red.).

Ik ben ook niet zo heel erg verbonden met één product, ik heb me eerst heel erg gericht op Business Components for Java (nu ADFBC), daarna ben ik me meer met TopLink en webservices en wat meer Java-achtige dingen gaan bezighouden. Tegenwoordig houd ik me veel meer bezig met BPEL en ESB’s en de dingen die daarbij horen. Ik heb ook over veel verschillende onderwerpen gepubliceerd.’

*Nu we het over ESB’s hebben: wat zie je van SOA in de praktijk, ik hoor zelf bijna alleen van proof of concept projecten?*

Dikmans: ‘Heel veel bedrijven hebben alleen *proofs of concept* en technische infrastructuurprojecten. Enterprise SOA kom ik ook heel weinig tegen. Wat ik wel zie, is dat er een IT-afdeling vanuit hun applicatiearchitectuur heel erg afdwingt dat alles een service moet zijn, terwijl er vanuit de business helemaal geen relatie mee is. Het idee is dan: ‘wij hebben een service oriented

**‘JDeveloper verdient echt veel meer aandacht dan het krijgt binnen de Java community’**

architecture, want alles is een webservice’. Maar niemand *maakt* dat op de business processen. Zelf ben ik erg voor een *meet in the middle*-aanpak. Daarbij denk je vanuit de business, ga je processen analyseren en van daaruit je services destilleren, maar je doet ook *bottom up* dingen. Als je alleen maar met de business praat en mooie plannen zit te maken komt er volgens mij ook

niks van. Je moet wel degelijk klein beginnen in stapjes, en dan niet met een *proof of concept* maar met een echt project. Vaak zijn *proof of concepts* ook te simpel, zonder security of zonder dat ze ooit in productie komen.'

*Kost een SOA niet veel performance?*

Dikmans: 'Je hebt enige overhead, maar op de plekken waar je die overhead hebt, zou dat als het goed is niet zo erg moeten zijn. Wanneer je van iedere update en delete in een tabel een service maakt, ja dan wel, maar als je wat grovere korrels maakt dan is dat een meer afgerond geheel en dan zou het niet zo'n probleem moeten zijn. Met een deel van de performance heeft de gebruiker op zich ook niets te maken, dat is gewoon een business proces. In plaats van dat je het met de hand overtypt of met een documentje werkt ga je het wat beter aan elkaar koppelen. Op die plekken is de overhead niet zo'n probleem. Ik kom ook vaak tegen dat er allerlei performance-eisen zijn, maar dat ik niet of nauwelijks terug kan vinden waarop die

gebaseerd zijn. Alle schermen moeten binnen zoveel milliseconden terug zijn. Ja, hoezo? Voor performance in een SOA moet je echt aansluiten bij de processen en kijken naar waar het belangrijk is. Per project heb je altijd overhead, je moet wat extra dingen testen, dat is zo, maar je gebruikt het niet design-

**Voor performance in een SOA moet je echt aansluiten bij de processen en kijken naar waar het belangrijk is**

time maar runtime. Het is er al. Je wilt een kaartje in je applicatie, je roept *Google Maps* aan en het staat er. Intern zou je dat ook met je services kunnen doen. Hoe vaak moet je niet bepaalde klantgegevens hebben? Dan kun je iedere keer opnieuw gaan definiëren, maar als je het al één keer hebt gedaan kun je het natuurlijk gaan hergebruiken.'

*Wat vind je van de combinatie van business rules en SOA?*

Dikmans: 'Ik ben min of meer opgegroeid met OO, dus ik heb geleerd: data en logica horen bij elkaar in een component. Bij SOA ga je dat weer heel erg scheiden en daar moet ik nog steeds aan wennen. Ik denk dat je dat weer op de goede granulariteit moet bekijken. Een business rule is ook weer een component waar data bij komen kijken. De structuur die je ziet is sterk afhankelijk van je gezichtspunt, van het stukje in die structuur dat je als uitgangspunt neemt. Als ik naar mijn eigen webapplicatie kijk dan heb ik misschien OO in mijn domeinmodel. Als ik kijk naar de service die ik aanroep, dan krijg ik gewoon een databericht binnen, en dat is een heel andere manier van denken. Wanneer ik een *business rule engine* aanroep om te kijken of een bepaalde hypotheek verstrekt mag worden, dan kijk ik weer tegen een logical service aan, maar die applicatie kan nog steeds wel een OO-applicatie zijn bijvoorbeeld.'

*Dikmans ziet SOA en agile wel goed bij elkaar passen:*

Dikmans: 'Agile ontwikkelmethodes gaan ook heel erg vanuit dat je naast de business user gaat zitten en je software maakt. Hoe meer vertalingen er tussen staan, requirementsdocumenten en business analyses, hoe verder je je verwijderd van wat de business user nu eigenlijk wil. Hoe langer je erover doet: hetzelfde probleem. De wereld verandert terwijl je zit te ontwikkelen. Dat vind ik ook het aardige van SOA, dat het heel erg gaat om services, om brokjes, niet om heel grote dingen. Het is ook een evolutionair model, je begint met een bepaald proces en je gaat dingen die al bestaan als services aanbieden. Dat hoeft niet in een *big bang*. Je kunt beginnen bij de belangrijkste



business case, of bij de gemakkelijkste.'

#### *Ook SOA heeft haken en ogen*

Dikmans: 'Menselijke organisatorische dingen als het *not invented here syndrome* heb je bij SOA net zo goed. Ik lees ook wel eens dat beheer gemakkelijker zou worden. Ik denk dat beheer van de code gemakkelijker is, en dat je het ook gemakkelijker kunt veranderen omdat het niet zo in elkaar zit. Ik denk echter dat het in de operaties wel wat ingewikkelder wordt. Het is belangrijk dat daar heel goede tooling voor komt, anders wordt het een soort zwaard van Damocles dat boven organisaties hangt. Naarmate we meer een SOA-omgeving krijgen wordt het steeds belangrijker dat het goed blijft draaien. Als we het beheer niet goed inrichten dan ontstaat echt een groot probleem. Dat geldt ook voor beveiliging, governance, dat soort dingen. Daar komt nog bij dat veel standaarden nog heel erg aan het veranderen zijn.'

*Veel bedrijven en overheden hebben honderden applicaties in de*

*grootst mogelijke technische verscheidenheid. Is het nu echt verstandig om dat allemaal te willen verbinden?*

Dikmans: 'Nu hebben ze dat probleem ook, alleen kopiëren ze alles en worden veel dingen overgetikt. Fouten herstellen is vaak nog het duurste. Denk aan alle klachten die je binnenkrijgt over dossiers die te laat zijn en termijnen die je niet haalt, waardoor hoge kosten ontstaan omdat je niet binnen een bepaalde tijd hebt gereageerd. De meeste business cases die je ziet zijn fusies, veranderende wet- en regelgeving of markten die heel snel veranderen. Dat zijn de drie hoofdkenmerken van organisaties die er profijt van kunnen hebben. Je ziet dat bij telco's, verzekeringsmaatschappijen, energiemaatschappijen, en bij de overheid; in al die situaties heb je heel veel veranderingen. Het zijn de typische kandidaten voor deze oplossingen, omdat ze het probleem echt voelen.'

**Tekst en fotografie: Dré de Man.**

---

---

#### Advertentie

---

OraVision bouwt Oracle-oplossingen waarin documenten, transacties en bestaande systemen samenwerken.  
OraVision staat bekend als *the mid-office company*.  
OraVision bouwt vanuit haar geheel eigen visie: kwaliteit staat centraal.



#### **Kwaliteit in kennis**

OraVision beschikt over enorme ervaring in Oracle-, Java- en integratietechnologieën. Bij ons staat de techniek echter nooit op zichzelf. Juist bij mid-office en document-integratie toepassingen laten we de technologie tot volle bloei komen.

#### **Kwaliteit in werk**

Klanten geven OraVision al jaren het vertrouwen om geavanceerde ICT-toepassingen te realiseren die tegelijk gebruikersvriendelijk zijn. Onze mid-office oplossingen bevinden zich immers in het hart van elke bedrijfsvoering.

#### **Kwaliteit in samenwerking**

Bij OraVision staat niet alleen technische kwaliteit hoog in het vaandel, ook onze stijl is onderscheidend. Vanuit onze Limburgse basis investeren we nadrukkelijk in persoonlijke relaties en genieten van het goede leven.

Geïnteresseerd in de visie van OraVision op Oracle, Java, integratie en mid-office? Bezoek [www.oravision.com](http://www.oravision.com) en abonneer u gratis op de OraVisionair.

