

Rick van der Lans over technische en projectaspecten van SOA's

# PERFORMANCE HEET HANGIJZER

Rick van der Lans geeft al sinds vele jaren lezingen en doet consultancy over SOA. In een gesprek licht hij vooral de technische aspecten van SOA toe: waar verschilt het van andere technieken, waar liggen de technische valkuilen en wat zijn de mogelijkheden.

Door Dré de Man

“Ja en nee,” antwoordt Rick van der Lans op de vraag of SOA nu echt zo nieuw is en zo'n geweldig idee is als onder meer Gartner ons wil doen geloven. “Het is natuurlijk niet helemaal nieuw want het erft van alle voorliggende generaties van technologieën, het is alleen technologisch een stap verder.” Van der Lans ziet vijf grote verschillen. Drie daarvan worden vaker genoemd: het is allemaal gebaseerd op internationale standaarden, je kunt loosely koppelen, en het wordt door iedereen geadopteerd. Alleen dat laatste is redelijk nieuw volgens hem.

“Wat ik zelf het grote verschil vind: met alle voorgaande generaties kon je koppelingen mee realiseren, maar de applicaties bleven wel gewoon staan. Bij SOA begin je waarschijnlijk met het inzetten om koppelingen te creëren, maar het leuke is dat als je doorzet je uiteindelijk stap voor stap je hele IT-systeem opnieuw kunt opbouwen. Dan is het niet meer alleen integratietechnologie maar een platform waarop je nieuwe systemen gaat bouwen. Het vijfde verschil ten slotte is de invloed op de organisatie. Doen we SOA met BPM dan is de invloed op zowel de IT- als op de gebruikersorganisatie erg groot. Er verandert dan gewoonweg veel.”

*Die flexibiliteit is de grote belofte van SOA?*

“Ja, dat hoort de belofte te zijn, al denk ik dat sommige bedrijven nu zeggen dat ze aan SOA doen of SOA-projecten

doen, maar in feite gebruiken ze alleen de integratietechnologie van vandaag, bijvoorbeeld SOAP. Dat is zeker ook interessant, maar dan is het niet revolutionair, het is dan gewoon een heel kleine stapje in de goede richting. Ik denk dat als Gartner van die voorspellingen doet dat ze uitgaan van niet alleen integreren maar ook een heel nieuw systeem neerzetten, want dan pas kom je van je silo's af.”

*Er bestaat nog steeds geen algemeen geaccepteerde definitie. Volgens Van der Lans vertroebelt dit ook veel discussies over het onderwerp:*

“W3C begint zijn definitie met ‘het is een set van componenten die je kunt aanroepen’. Dat vind ik wel een heel simpele definitie. Als je kijkt naar een definitie van IBM; daar praten ze expliciet over businessproces, dat vind ik heel interessant. Ik weet dat Gartner in een van zijn definities het eigenlijk een approach noemt, daar vind ik ook wat voor te zeggen, maar het is jammer dat we niet één algemene geaccepteerde definitie hebben.”

*Mijn observatie is dat veel bedrijven die zeggen met SOA bezig te zijn, zich nog in het stadium van de proof of concept bevinden. Klopt dat?*

“De markt is nu verdeeld in vier kwarten. Een kwart dat nog niets doet, een kwart die zitten te lezen en te studeren, een kwart zit te experimenteren, en een kwart heeft wat draaien. Maar ik denk dat weer een heel substantieel deel daarvan iets

heel simpels heeft draaien, waarschijnlijk weer een aantal SOAP services, niet per definitie een SOA. Maar er zijn wel bedrijven die een SOA draaien met business proces lagen daar bovenop en dus voelen wat de voordelen daarvan zijn. Het leek een beetje alsof de Europeanen wat voorliepen op de Amerikanen op dit gebied, maar volgens mij gaan de Amerikanen ons weer inhalen. So be it.”

## Modellen

*De invloed op de organisatie is misschien wel het belangrijkste verschil tussen SOA en voorgaande technologieën.*

“Veel van de technologieën die we gehad hebben waren gewoon puur technologie. Met SOA, mits we wel naar de businessproceslaag toegaan (als we die niet meenemen dan is het wederom gewoon technologie, speelgoed voor de IT-afdeling). Dan ga je echt ingrijpen op de organisatie. Businessgebruikers krijgen vanuit dat soort omgevingen ineens heel veel inzage hoe efficiënt bepaalde stappen nu eigenlijk uitgevoerd worden, en kunnen daarmee de organisatie gaan sturen. Dat is iets wat we met OO of client/server nooit hebben gekregen. We zitten met die bovenlaag van SOA, die businessproceslaag staat toch wel met meer dan één been in de business zelf en nog maar met een half beentje in de technologie.

Daar ligt meteen ook een uitdaging, want je moet mensen hebben die aan de ene kant de business heel goed begrijpen, en daar modellen of systemen voor creëren die wel door de IT-afdeling te bouwen zijn. Daar struikelen we soms ook in de projecten over, omdat we daar niet de mensen voor kunnen vinden, of omdat de organisatie er nog niet helemaal klaar voor is. Ik denk ook dat meer SOA-projecten mislukken door dit aspect, dan door de techniek. Als er een probleem is met technologie, dan krijgen we dat uiteindelijk altijd wel opgelost. Maar als het probleem in de organisatie zit dan kan je dat niet zomaar oplossen door er geld ertegenaan te gooien. Dat moet, soms heeft het met politiek te maken, soms met krachten binnen de organisatie, met visies, met doelstellingen. Dat is nooit echt ons terrein geweest.”

*Sommige architecturen maken weliswaar gebruik van web-services maar zijn toch geen SOA. Van der Lans vindt dat je een SOA zou kunnen maken zonder business proces laag. Daarbij gaat het vooral om de vraag of de nadruk ligt op de (business-) services, of op de bus. Wordt de bus (ESB) alleen maar gebruikt om data of commando's heen en weer te sturen, dan is het volgens Van der Lans geen SOA, maar een BOA (bus-georiënteerde architectuur). Als daarentegen in een project de nadruk ligt op (business) services, dan wel.*

“Het gaat daarbij wel om business services, functionaliteit die voor iemand uit de business wat betekent. Misschien moeten we dat ook in de definities van SOA opnemen, dat de service vanuit het business proces en de wensen van de business zijn ontworpen. Van de interface naar de body toe en niet vanuit

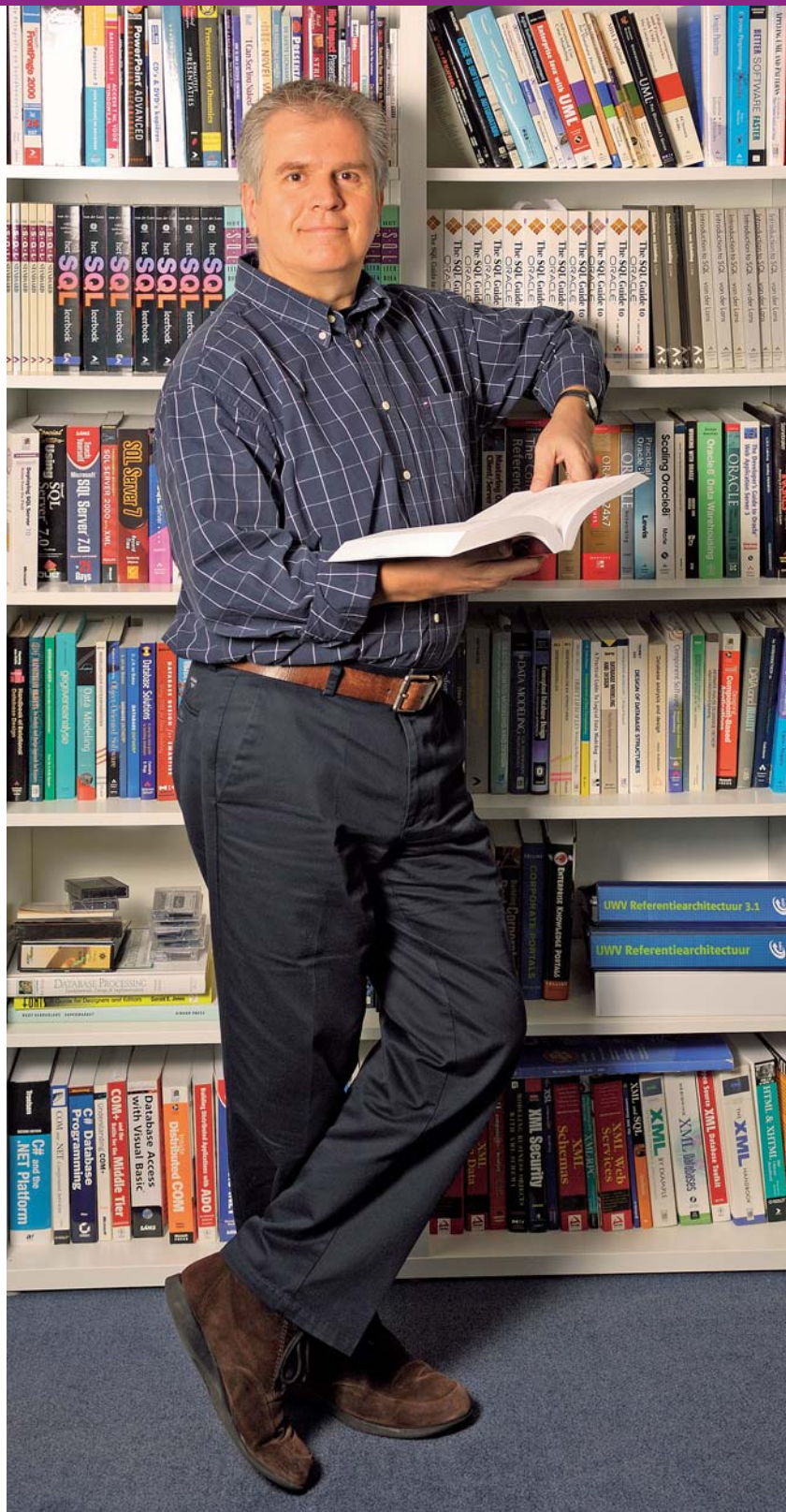


Foto: Dré de Man

de applicaties naar de interface toe, outside in in plaats van inside out.”

*Ook het gebruik van programmeertalen gaat volgens Van der Lans veranderen. Daarbij moet verschil gemaakt worden tussen hoger gelegen en lager (dichter bij de applicatie) gelegen services. Hoger gelegen services zullen waarschijnlijk toch in*



*een BPEL of BPEL-achtige taal geschreven worden, die leent zich ook veel beter voor dingen als a-synchrone calls. Voor de onderste services zullen we waarschijnlijk verschillende oplossingen hanteren.*

*“Als je moet koppelen met een COBOL-systeem dan is het waarschijnlijk toch gemakkelijker om die koppeling ook in COBOL te schrijven. Eigenlijk willen we daar speciale tools voor inzetten om die interfaces te bouwen. Er zijn nu tools, maar die gaan wederom uit van die *inside out* benadering. Daar zouden we eigenlijk veel meer intelligentie willen, om data te kunnen reinigen of filteren en waar je declaratief dingen kan aangeven.”*

*Voor de middenlaag is het antwoord niet zo eenduidig.*

*“Daar zullen we voor een deel Java en C#-achtige talen gaan zien, maar misschien ook BPEL of BPELJ-achtige talen. Het accent gaat toch iets meer liggen op efficiency, want het moet toch allemaal snel gebeuren.”*

*Bij de user interface ten slotte zal het klassiek programmeren waarschijnlijk blijven bestaan.*

*“Als er een prachtige AJAX-gebaseerde user interface gebouwd moet worden die een business proces service opstart, dan moet waarschijnlijk nog wel gewoon geprogrammeerd worden. Daar gebruik je waarschijnlijk gewoon verder JavaScript-achtige talen. Ik denk echter dat het merendeel van deze applicaties niet meer op de gewone manier gebouwd zal worden.”*

*Van der Lans vindt het idee van business rules engines heel mooi past bij een SOA. Wel vraagt hij zich af of de overzichtelijkheid behouden blijft wanneer er te veel business rules in de engine terecht komen, en maakt hij zich zorgen over de vraag of dingen nog trackable en traceable blijven.*

## Performance

*Performance is volgens Van der Lans een bron van grote zorg in dit soort projecten. In een gelaagde architectuur moeten voortdurend XML documenten opgebouwd en verwerkt worden. Die verwerking kost veel performance, veel meer dan het versturen van de documenten over de bus.*

*“Sommige leveranciers bedenken daar wel oplossingen voor. Uiteindelijk is die BPEL-engine gewoon een Java-applicatie die draait op een Java-applicatieserver, en vanuit dat BPEL-proces wordt bijvoorbeeld een service aangeroepen en die service draait uiteindelijk ook op die applicatieserver. Dan zou je natuurlijk kunnen zeggen: ik doe een interne call via RMI. Dit gebeurt nu wel, bijvoorbeeld door Oracle.”*

*Maar dan geef je wel een stuk openheid prijs?*

*“Ja, het zijn natuurlijk proprietary interne optimalisaties, ik denk dat we die sowieso moeten uitvoeren. Dan verandert XML SOAP van een runtime-technologie in een design time technologie. Verplaatsen we dan die service ineens naar de*

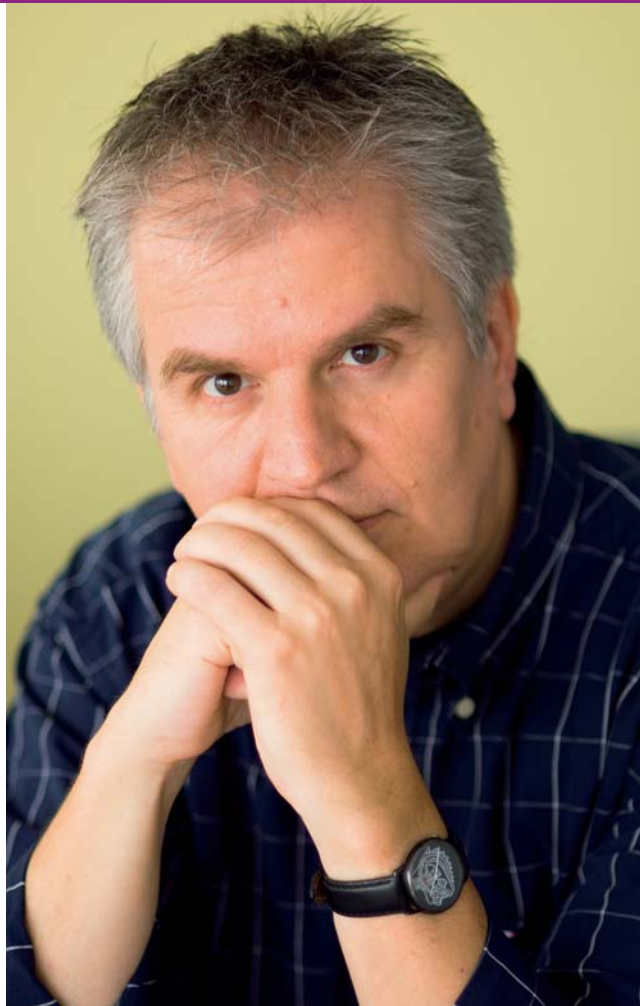


Foto: Dré de Man

*andere kant van de wereld dan moet hij alsnog gewoon XML SOAP gaan uitvoeren. Dat vind ik eigenlijk wel goede verbeteringen, want daar winnen we heel veel performance mee. Waar we voor moeten oppassen, is dat het we het al in design time gaan optimaliseren.”*

*Van der Lans ziet een soort relatie met de ontwikkeling van relationele databases. Die waren in het begin ook erg traag en de leveranciers zijn op een gegeven moment optimalisaties gaan uitvoeren om het sneller te doen gaan. Niettemin rust ook een zware verantwoordelijkheid op de schouders van de ontwerpers van de applicaties. Hij ziet het als noodzakelijk om vanwege de performance een deel van de functionaliteit in de applicaties in plaats van in de services onder te brengen.*

*“Wanneer heel veel communicatie gaat plaatsvinden tussen service en applicatie dan kan de performance zo slecht worden dat een onwerkbaar situatie ontstaat. Zoiets gebeurt bijvoorbeeld wanneer je veel applicaties hebt en die gewoon laat as is en alles daar bovenop zet. Dan krijg je gewoon heel veel services. Van Volkswagen wordt nu wel gezegd dat die meer dan 15.000 services hebben en dat is toch gewoon veel te veel. Dan kun je beter aan een oudere COBOL-applicatie een module toevoegen, puur om een service te kunnen bouwen. In plaats van dat je een service schrijft met een dikke body krijg je eigenlijk een dunne body, en die body verplaatst zich naar de applicatie. Dat soort dingen gebeuren*

al en die gaan we de komende jaren nog meer doen. We hopen natuurlijk dat we in de toekomst zo efficiënt worden dat we dat niet meer hoeven te doen.”

*Een canonical datamodel kan ook een oplossing zijn.*

“Ja, ik denk dat we voor de SOA-wereld toch weer hard moeten gaan nadenken over onze enterprise datamodellen. Wanneer er bij de één een adresgegeven uitkomt en het moet er bij de ander weer in, zouden beide partijen dezelfde definities moeten hanteren. Een grote overheidsinstantie bijvoorbeeld hanteert in dit soort situaties ook de stelregel ‘de vervuiler vertaalt’. Als ze een applicatie aansluiten dan komen er services op te staan die in principe de data opleveren conform de eisen van de SOA. Het betekent ook dat de services die dat doen niet behoren tot de SOA zelf, maar tot de applicaties en dus ook waarschijnlijk eigendom zijn van de applicatie-eigenaar. Dat is wel een principe waarvan ik denk dat iedereen het zou moeten hanteren.

Ik ben er alleen geen voorstander van dat we het totale enterprise data model voor de hele organisatie neerzetten. Dat zijn meestal projecten die tot niets lijden. Het leidt tot wat de Amerikanen ‘analysis paralysis’ noemen: het eindigt nooit, er is altijd wel meer te analyseren. Dat moeten we gewoon niet doen, we moeten praktisch denken en met een paar business processen beginnen.”

Foto: Dré de Man



*Bij performance hoort ook de discussie over grofkorrelige en fijnkorrelige services. Hoe kijkt Van der Lans daar tegenaan?*

“Het antwoord is: de services moeten niet te groot en niet te klein zijn (lacht). Ik doe niet meer mee aan die discussie. Het lijkt ook wel alsof je tussen twee discussies in zit. Aan de ene kant wil je de services tamelijk groot maken om de hoeveelheid interactie met de applicatie tot een minimum te beperken, puur om performance-redenen, aan de andere kant wil je natuurlijk niet je applicaties gaan uitbreiden. Misschien wil je die eigenlijk juist wel uifaseren, dus juist de service in trekken. Je zit met conflicterende belangen, performance en flexibiliteit zitten elkaar dan in de weg.”

*Wanneer je start vanuit je businessprocessen is er volgens Van der Lans vaak helemaal geen discussie over granulariteit.*

*Binnen een model van een businessproces heb je een aantal stappen, en elke stap wordt een service.*

“De discussie die we wel hebben: als je vanuit twee businessprocessen start, dan heb je soms ineens twee services, die bijna hetzelfde zijn. Misschien moet ik die twee stappen in twee verschillende businessprocessen dan samenvoegen tot één, de interface wat generaliseren, en dan kan hij dienst doen voor allebei.”

## Beheer

*De gedistribueerde opzet van een SOA-structuur maakt het beheer moeilijker. Gelukkig zijn er ook oplossingen.*

“Ik denk dat *governance tools* als van AmberPoint, Sonic Software, en IBM in de Tivoli-tak dat redelijk goed invullen. Die producten kunnen *real time* laten zien wat de performance is en die ten opzichte van de SLA in stand houden. Waar ik zenuwachtiger over ben, is dat veel bedrijven in de fase zitten waarin ze een service bus gaan selecteren. Dan kijken ze initieel eigenlijk alleen naar de producten om zo’n SOA te bouwen. Maar stel je nu voor dat je zo’n product hebt dat een optimalisatie uitvoert via de applicatieserver als het ware om de SOAP-call heen gegaan wordt en het governance-tool houdt daar geen rekening mee? Wanneer je een ESB koopt, moet je je ook meteen de vraag stellen: hoe gaan we governance doen? Dat moet gewoon bij elkaar passen. Niet alleen vangen beheeraspecten, maar ook vanwege de security. Mag iedereen wel alle services aanroepen, op een bepaald moment, moet er encryptie worden toegepast? Ik ben geen security-expert, maar alle security-experts zeggen altijd dat je niet moet wachten met security inbouwen tot het einde van het project. Ik denk dat bijna alle analisten het daarover eens zijn: governance moet er gewoon vanaf dag één bij zijn.”

*De keuze voor een bepaalde ESB is op dit moment nog niet zo gemakkelijk. Volgens Van der Lans zijn er nog te veel aanbieders en moet de echte shake out nog plaatsvinden.*

“Als je zo’n ESB-markt bekijkt zou je moeten zeggen, ja, uiteindelijk wordt het gewoon een slag, op een gegeven moment moet je technologisch mee en je moet ook qua

marketing mee. En ik denk dat een aantal op een gegeven moment niet meekan, die gaan gewoon uit de boot vallen.”

*Probleem bij de selectie is ook dat de onderlinge portabiliteit minimaal is.*

“Interoperabiliteit is er uiteraard, maar de portabiliteit is minimaal. Als ik met de spullen van IBM zo’n hele SOA opbouw en daarna zou ik willen migreren naar Microsoft en dan naar Oracle, daar blijft niet veel van over. Er moeten zoveel specificaties opgevoerd worden waarvoor geen standaard bestaat, denk aan een business rules engine. Alle specificaties die daar worden ingevoerd zijn in principe proprietary specificaties. Er zijn heel veel plekken waar je toch allemaal proprietary talen gebruikt.

De standaarden geven interoperabiliteit, en bij portabiliteit kunnen we een groot vraagteken zetten. En ik denk dat dat vraagteken alleen maar groter gaat worden. Je ziet nu bedrijven als Tibco en webMethods die nu met modules bezig zijn om voorspellingen te doen. Die functionaliteit hebben we nu bij een aantal leveranciers, prachtig, maar dat zijn wederom allemaal proprietary talen, daar is helemaal geen standaard voor. Ik denk dat je er vanuit moet gaan dat je een hofleverancier kiest en je zit daar gewoon aan vast.”

*Maar wanneer je nu heel veel functionaliteit bijvoorbeeld al je business rules in zo’n systeem hebt gestopt en het blijkt een doodlopende weg, dan heb je een probleem.*

“Ja, helemaal mee eens. Maar we zeggen ook: je moet klein beginnen. Mensen denken dat ze allemaal low level services moeten bouwen. Dat bedoelen we niet: je moet beginnen met een soort staaftje te bouwen een business proces, wat services daaronder en dan een link naar de applicaties. De reden om het op die manier te doen is: dan krijg je misschien al uit dat staaftje een ROI, want ergens ga je een soort optimalisatie in de business uitvoeren. Zou je beginnen met wat low level services, wat wordt daar dan ooit de ROI van? Waar ga je het geld verdienen? Als dan op een gegeven moment blijkt dat je het verkeerde product gekozen hebt omdat het buiten de shake out valt, dan is het wel een heel verdrietig verhaal.”

*Ik heb het gevoel dat met SOA wel heel veel dingen naar elkaar toegroeien, in het algemeen de business en de IT, maar ook binnen IT-vakgebieden.*

“SOA en BI raken elkaar op een aantal vlakken. Een interessant raakvlak is dat de BI-wereld graag kijkt naar KPI’s, terwijl we moeten bekennen dat we sommige KPI’s die we willen hebben niet kunnen genereren vanuit het productiesysteem. Simpel gezegd: in veel productiesystemen registreren we eigenlijk alleen het startpunt en het eindpunt van een bedrijfsproces. Van het proces zelf, van wat er na een bestelling in al die afdelingen gedaan is en hoe lang ze erover gedaan hebben, zien we weinig terug. Veel van de KPI’s hangen eigenlijk aan dat proces. Als we zo’n business proces

engine inzetten, die verzamelt al dat soort gegevens, dat is een automatisme, die data kunnen we dan oppakken en in een datawarehouse stoppen en de meest fantastische rapporten over draaien. Voor de BI-wereld is dit een geweldige toevoeging, sommige mensen zeggen: BI of balanced scorecards of KPI’s kan eigenlijk niet zonder business processen. Voor het eerst krijgen we een geweldig mechanisme om die businessprocessen echt te implementeren.”

## Fortune 100

“Het is heel logisch om deze werelden wel naast elkaar te gaan zetten. Een ander aspect is: veel van de services die we bouwen halen uiteindelijk toch data uit meerdere systemen. Je wil bijvoorbeeld een service bouwen: geef een klant dossier. Dat betekent: je moet uit vijftien systemen data halen, in ons datawarehouse is dat natuurlijk ook al gebeurd, en daar hebben we vaak heel veel tijd en geld in gestoken om al die data bij elkaar te brengen, te masseren en te filteren, te cleanen en in dat datawarehouse te krijgen. Dan zou je eigenlijk al die logica opnieuw moeten gaan bouwen om het in de service ook voor elkaar te krijgen. Dan bouw je toch gewoon die queryservice op het datawarehouse? Je kunt ook gebruik maken van de kennis van degenen die het datawarehouse gebouwd hebben. Die hebben het al een keer gedaan en kunnen dat vrij eenvoudig herhalen voor de services. Laat ik het zo samenvatten: datawarehousing is data-integratie, SOA is applicatie-integratie, maar applicatie-integratie is uiteindelijk ook data-integratie. Dus dan is het logisch om te zeggen die problemen ga ik niet twee keer oplossen.”

*Hebben kleinere bedrijven bij de invoering van een SOA voordeel van de omgekeerde wet van de remmende voorsprong?*

“SOA is er in ieder geval niet alleen voor de Fortune 100 bedrijven, het is voor iedereen. Ik moet ook bekennen dat we nu al zien dat de kleintjes gedwongen worden. Grote multinationals eisen vaak dat SOAP-services beschikbaar zijn. Ik denk dat het net als een relationele database, een technologie gaat worden die we uiteindelijk allemaal zullen gaan gebruiken.”

*Is Van der Lans nu blij met de huidige ontwikkelingen?*

“Ja! Ik vind het een mooi en goed idee, als we dit op die manier gaan doen dan denk ik inderdaad dat we in staat zullen zijn om flexibeler IT-systemen te gaan bouwen. Dan zullen we ook gemakkelijk business-IT alignment voor elkaar kunnen krijgen. We krijgen dan bouwstenen waarmee we gemakkelijker de business kunnen volgen in plaats van dat we zelf een bepaalde structuur neerleggen. Persoonlijk vind ik het ook leuk dat we het eindelijk hebben over de business en de technologie samen, het wordt gewoon één onderwerp, en dat werd ook tijd.”

Dré de Man is hoofdredacteur van Software Release Magazine.