



Mark Madsen ziet noodzaak voor nieuwe BI-tools

# SOA moet zich op data richten, niet op code

Teus Molenaar

**De Service Oriented Architecture beweging heeft haar blik strak gericht op de code in plaats van op de data, terwijl daar toch de informatie valt te halen. Op dat vlak moet het roer om, vindt de Amerikaanse analist Mark Madsen. Die benodigde nieuwe aanpak ziet hij niet snel van de gevestigde orde komen, hoewel MicroStrategy, de kleinste BI-leverancier, het volgens hem begint te begrijpen.**

Na ruim een uur geanimeerd praten over Business Intelligence ("Er is de afgelopen tien jaar niets veranderd!"), webservice en data, wacht Mark Madsen een volgende verplichting. "En dan hebben we het nog niet eens over Open Source gehad", stelt hij spijtig vast bij het afscheid. Het is geen boude veronderstelling dat hij daar zinnige zaken over heeft te melden, want zijn betoog verraadt een kundige, onafhankelijke geest. Bijzonder goed ingewijd in de mysteries die met datawarehousing en Business Intelligence te maken hebben.

Zijn verhaal begint bij de ETL-tools (Extraction, Transformation, Load). "Zo'n zeven jaar geleden waren die erg gebrekkig en erg duur. Bedrijven kochten die tools voor een bedrag met zes of zeven cijfers, maar eigenlijk was je toen beter af om zelf dergelijke gereedschappen te maken. Ik gaf toen les over dit soort tools aan het Datawarehouse Institute en had me er goed in verdiept. Pas vijf jaar geleden zijn de ETL-tools van de leveranciers een goed alternatief gebleken. De kern van de zaak is data-integratie. Daar is iedereen eigenlijk voortdurend mee bezig. Omdat het zo complex is en daarmee voor mij een

tamelijk zekere bron van inkomen, heb ik mij in dit onderwerp vastgebeten", vertelt Madsen openhartig.

"Iedereen moet data-integratie 'doen'. Dat is geen gemakkelijke taak, omdat er tal van verschillende complexiteiten zijn. En over de jaren heen is het mij opgevallen dat het er niet makkelijker op wordt. Het wordt eigenlijk steeds ingewikkelder. De tools zijn dan wel sterk verbeterd, maar, omdat het complexer is geworden data te integreren, is de uitkomst eigenlijk nog steeds hetzelfde. De komende decennia wordt dit een probleem", meent Madsen.

## SOA is te complex

Hierbij vindt Madsen een aanknopingspunt bij de discussies over Service Oriented Architecture. "Laten we beginnen met de praktische kant van de zaak: data-integratie. SOA richt zich op de code en laat de data links liggen. Dus het blijft een uitdagend onderwerp en voorlopig ook nog wel een bron van inkomsten voor mij", glimlacht hij.

SOA en services zijn de laatste vijf jaar vooral aangeprezen door IT-leveranciers als IBM, Microsoft en Oracle (die begin 2008 BEA overnam). De aandacht ligt daarbij vooral op standaarden als SOAP (Simple Object Access Protocol) en WS-\* (spreek uit: double U S Star). De laatste is een set protocollen dat de beveiliging van webservices moet regelen. Het is gezamenlijk door IBM en Microsoft aan de standaardcommissie OASIS voorgelegd. "Erg complex allemaal", vindt Madsen.

Het is een onderwerp dat vooral door applicatie-ontwikkelaars is ingepalmd; overigens ook omdat de IT-industrie zich steeds op deze groep heeft gericht, zo verklaart Madsen.

"Als ik met bedrijven praat die aan de slag zijn gegaan met SOAP en WS-\*, dan hoor ik zelden dat het goed gaat. Het meeste komt van de applicatie-ontwikkelaars. Zij coderen webservices, data-services en behandelen het onderwerp alsof het een codeerprobleem is, maar dat het is niet. Het is simpel om de

## Spreker op Array congres

Dag twee van het congres Datawarehousing & Business Intelligence dat Array Publications, uitgever van Database Magazine, begin oktober hield, is afgesloten door Mark Madsen. Zijn thema 'Clues to the Future of Business Intelligence' boeide het publiek mateloos.

De dag erna gaf Madsen voor een uitgelezen groep toehoorders een seminar onder het motto 'Beyond ETL: Data Integration Technologies for the Future Data Warehouse'. Hier raakten zij onder meer bekend met technieken voor het aanpakken van real-time en on-demand data.



Mark Madsen: "SOA richt zich op de code en laat de data links liggen".

code te schrijven, om webservices te maken, maar de infrastructuur om dit alles te beheren is uiterst moeilijk, en de data erachter vormen een uitdaging."

### Aandacht op data

Madsen vertelt dat hij genoeg kreeg van BI; er gebeurde nauwelijks iets nieuws en hij raakte verveeld. Zo heeft hij zeven jaar gewerkt bij opstartende webbedrijfjes in Silicon Valley. Toen de internetzeepbel uiteenspatte, is hij weer het BI-pad opgegaan. Maar hij heeft wel veel geleerd in de zeven jaar bij de start-ups. "Ik zag hoe de webbedrijven omgaan met code en data", legt Madsen uit. "Veel van het gedachtegoed over webservices komt trouwens uit deze wereld. Zij zien services als een platform om zaken te doen. Denk aan Amazon, eBay, Google en dergelijke. Zij maakten zich nauwelijks druk om het schrijven van code om Service Oriented Architecture te bouwen. Ze besteedden geen miljoenen dollars bij IBM om dat voor elkaar te krijgen. Je hebt WebSphere zus en zo, WS-standaard dit en dat, UDDI-directory zo. Daar waren Google en consorten helemaal niet mee bezig. IT-medewerkers hebben hun handen vol met webservices opschalen, zowel vanuit het ontwikkelingsoogpunt als vanuit het perspectief van code en data. Amazon en eBay hebben helemaal geen schalingsproblemen. Ik weet niet precies welk percentage, maar ik geloof dat zo'n dertig procent van de omzet van eBay helemaal niet via de website loopt, maar via web-

services. Dan heb je het over 2,5 miljard dollar. Zij hebben zich vooral geconcentreerd op het beschikbaar stellen van data. Waarschijnlijk gaat het om dertig procent transactionele services, waarbij je probeert iets te laten gebeuren, en zeventig procent puur toegang verlenen tot data via een service. De aandacht gaat erg zwaar naar data in deze wereld. Maar als je kijkt naar de bedrijven die SOA naar voren schuiven, dan zie je dat ze complexe standaarden, complexe en dure tools en complexe architecturen promoten bij programmeurs, terwijl zeventig procent van wat werkelijk geleverd wordt data betreft. Toch is er helemaal geen focus op data-architectuur, datakwaliteit, data-transport en data-integratie; al die zaken die nodig zijn om de systemen in de back-end aan elkaar te binden om ze beschikbaar te krijgen aan de front-end. Daarom mist men met SOA de boot; omdat het op code is gericht in plaats van op data."

### De tijd ontbreekt

De relatie tussen code en data noemt Madsen 'tricky'. "De database slaat data op een bepaalde manier op, registreert transacties en doet van alles voor de automatisering van bedrijfsprocessen. Het idee met services was dat het mogelijk is sneller applicaties te bouwen die dit model volgen, maar er is nog steeds een ont koppeling tussen het model van een object dat een applicatie volgt en het idee van een verzameling van objecten binnen een database. Het opslaan van objecten in een database is geen goed idee. Dat heeft nooit goed gewerkt; alle objectgebaseerde databases zijn inmiddels alweer verdwenen. Applicaties hebben hun eigen specifieke aanpak om met data om te gaan en ze zijn altijd geïnteresseerd in één ding: één klant, één transactie, één adres, ga zo maar door. Terwijl de database juist al die objecten verzamelt; niet alleen voor die ene applicatie, maar ook voor andere toepassingen, voor rapportages en dergelijke. Er is een tussenlaag waar je die twee modellen aan elkaar koppelt. Niemand heeft daar nog een goede oplossing voor. Java-ontwikkelaars proberen het met zaken als Hibernate om die vertaling tussen applicatie-object en database te maken. Dat werkt ook wel voor bepaalde problemen, maar bij complexe database-problemen faalt het.

### Alle objectgebaseerde databases zijn inmiddels alweer verdwenen

Het probleem met tools als Hibernate is dat ze veel aannames maken over de database die mank gaan. Neem het controleren van een adres. Dat lijkt eenvoudig; het is gewoon één object in een applicatie. Maar in een database zitten meerdere tabellen met adressen: woonadres, afleveradres, betaaladres om er een paar te noemen. En dat kunnen allemaal verschillende objecten zijn. In Hibernate definieer je dan een applicatiemodel en geef je aan wat jouw data zijn en vervolgens maakt het een koppeling

### Holistische aanpak

Toen Mark Madsen besloot zich te vestigen als zelfstandig consultant en analist zocht hij een naam voor zijn bedrijf. Het werd Third Nature. Hij wilde beslist geen naam met 'Business Intelligence' erin. Te veel voor de hand liggend. "Bovendien zal de term Business Intelligence gaan verdwijnen", voorspelt hij.

In die periode las hij een boek over 17e eeuwse landschapsfilosofie. Daarin werd het wilde, ongerepte, natuurlijke landschap beschreven als 'First Nature'. Het aangeharkte, strak onderhouden landbouwlandschap dat daarop volgende, staat te boek als 'Second Nature'. Gedurende de Renaissance wilde de gegoede burgerij vanuit de esthetiek een huwelijk tussen de wilde natuur en de strakke lijnen van ingezaaide akkers. Bovendien hadden ze er het geld voor om prachtige paleistuinten (met Versailles als hoogtepunt) aan te leggen. Deze landschappen staan bekend als de 'Third Nature'.

"Mij sprak die filosofie wel aan", verklaart Madsen. "Veel van wat ik doe in de IT-wereld is het ontwerpen van systemen. Ik probeer onderdelen samen te smelten tot een holistisch geheel in plaats van ze te scheiden. Dat sluit wel aan op de Third Nature."

Bovendien is het altijd wel een gespreksonderwerp, geeft hij aan. "Er zijn ook mensen die een verklaring zoeken in sociologie, psychologie en IT vervolgens als derde natuur." Ook dat sluit wel bij hem aan, want oorspronkelijk is hij een afgestudeerd bioloog. Maar begin jaren tachtig kon hij in dat vak geen baan vinden, behalve dan een onderwijs carrière, maar dat trok hem allerm minst aan. "Dan zit je eigenlijk het grootste deel van je leven op de universiteit; niks voor mij."

Vervolgens ging hij computerwetenschappen studeren aan de Universiteit van Pittsburgh. "Ik programmeerde al van alles tijdens mijn biologiestedie om collegegeld te kunnen betalen. Het PC-tijdperk was net begonnen en mensen wilden iets doen met die apparatuur. Dat 'iets' schreef ik voor ze. Dus de overstap naar IT was voor mij heel natuurlijk."

Uiteindelijk heeft Madsen nóg een IT-studie gevolgd: software engineering aan de beroemde Carnegie Mellon universiteit, eveneens in Pittsburgh, Pennsylvania.

met de database met al die tabellen. En dan moet het ingewikkelde SQL-statements genereren om al die verschillende tabellen te leiden naar één object in het geheugen. Als je zelf SQL zou schrijven, dan zou je één of twee statements schrijven om de tabellen op een slimme manier te koppelen, je stelt outer-joins vast enzovoort. Maar elke SQL-opwekking is gelimiteerd. Hibernate zal al de verschillende tabellen apart ondervragen en die data apart zetten om die later aan elkaar te koppelen. Dit kost allemaal heel erg veel tijd en die heb je eigenlijk niet bij webapplicaties; daar moet het allemaal snel gebeuren."

### Verschillende werelden

Het probleem is volgens Madsen dat op maar heel weinig universiteiten te leren is hoe je tegen een database moet

programmeren. De Java-, .NET-, of PHP-ontwikkelaar schrijft een webapplicatie, maar moet er dan weer een SQL-expert bij halen om de benadering van de database te regelen. En dan heb je natuurlijk ook nog de 'gewone' applicatie-ontwikkelaars. Allemaal verschillende experts die nauw met elkaar zouden moeten samenwerken, maar dat in de praktijk nauwelijks doen. "Het zijn ook verschillende werelden. De 'web-jongens' willen lichte codes gebruiken in plaats van de zware frameworks. Ze lachen de applicatie-ontwikkelaars uit als die zeggen dat je BEA of WebSphere nodig hebt om een applicatie op te schalen. Zij hebben dat niet nodig, omdat ze een andere aanpak hebben. Met op http gebaseerde diensten heb je alleen maar een *load balancer* nodig om de applicatie over verschillende diensten uit te smeren."

Overigens zijn er wel mogelijkheden om bij de data te komen en SQL-procedures op te slaan, maar dat zijn volgens Madsen allemaal omwegen. "Maar ja, zo gaat het altijd in de IT."

### Real-time data-integratie

In het verlengde hiervan ligt zijn opmerking dat ontwikkelaars geen SQL kennen, waardoor webwinkels in de problemen kunnen komen. Hij komt met een zelfbedacht voorbeeld van een televisie-uitzending van het filmfestival in Cannes waarbij een filmster een bepaald type schoenen draagt. Er is altijd karige vraag geweest naar die schoenen, maar nu komen er plotseling duizend aanvragen tegelijk binnen.

"We zijn begonnen de interne data beschikbaar te stellen aan partnerbedrijven. Het kan zoiets simpels zijn als een webportal voor bijvoorbeeld het volgen van bestellingen. Vroeger gebruikten we daar EDI voor, maar dat is duur en complex en bovendien gericht op transactionele data. Maar dat is niet genoeg, je wilt ook context-informatie overdragen. Dergelijke informatie 's nachts uitwisselen voldoet niet meer als je webwinkels hebt, dan moet de informatie meteen beschikbaar zijn."

Die schoenenfabrikant die plotseling een order krijgt van duizend paar specifieke schoenen, heeft die niet op voorraad. Hij moet meteen kunnen nagaan in welke winkels er nog voorraad ligt om een transactie te kunnen afronden en het vervoer te regelen. "Je moet heel snel kunnen reageren op de vraag van een klant", onderstreept Madsen. "Dit voert de vraag op naar real-time data-integratie. Dat betekent dat je de data daar moet zien te krijgen waar ze nodig zijn. De data moeten veel sneller het datawarehouse inkomen dan een verversing van één keer per nacht. De andere kant hiervan is dat je de webontwikkelaar niet Hibernate wilt laten gebruiken in een webapplicatie die honderdduizend gelijktijdige sessies heeft richting jouw datawarehouse. Want een set query's, vermenigvuldigd met honderdduizend, is niet bevorderlijk voor jouw website, noch voor het datawarehouse. Teradata en HP (met Neoview) zijn al begonnen te werken met concepten van actieve en dynamische datawarehouses, waarbij je data *near real-time* invoert en in staat bent om ze tijdig te verwerken."

## Een grote kloof

"Het interessante van de discussies die ik met die bedrijven heb, gaat over de vraag hoe je de data weer snel uit het datawarehouse krijgt. Want zij blijven denken in termen van datawarehouses. Zij denken 'database en SQL'. Het probleem is dat SQL niet het pad is dat je moet volgen. BI-leveranciers en BI-tools spreken SQL. Applicatie-ontwikkelaars, webtools, Java-frameworks spreken allemaal geen SQL. En dat willen ze ook helemaal niet. De ontwikkelaars willen 'services' spreken. Dit probleem los je alleen op als je het datawarehouse voorziet van een service-laag. Zodat je de data in het datawarehouse – of een datamart – kunt benaderen op een manier waarop de ontwikkelaars dat willen doen. De uitdaging daarbij is dat applicatie-ontwikkelaars heel gericht services bouwen: precies voor wat ze nodig hebben. Zij zijn erg gefocust op het probleem dat ze moeten oplossen. Maar het datawarehouse daarentegen heeft een heel brede functie, want het moet aan meervoudig gebruik voldoen. Maar datawarehouses hebben geen ervaring met services, service-frameworks, Java-code, .NET-code, of wat dan ook. Daarom is er nu een grote kloof tussen datawarehouses en (web)applicaties. De tools die bijvoorbeeld IBM en Oracle bieden om services mogelijk te maken, zijn zo complex en lastig te gebruiken dat ze eigenlijk in de praktijk niet voldoen."

Als je naar het datawarehouse kijkt dan zal het niet een kwestie zijn van nog meer BI-tools en SQL, maar services die data benaderen, dus dataservice bovenop het datawarehouse. Of BI-services waarbij je de gebruikelijke user interface van de BI-tool weggooit,

zodat je niet de data ontvangt zoals die in een rapport zouden zijn weergegeven, maar de data in een XML-bestand, zodat de applicatie de data op die manier kan ontvangen.

De toekomst van de architectuur van BI-tools ziet Madsen als het integreren van de benodigde data, om ooit een rapport te genereren, maar je moet je niet druk maken om de formats waarin een rapport moet worden opgebouwd, alleen maar om de data die je nodig hebt. Die data moeten vervolgens als een service worden gepubliceerd en aan de applicatie worden aangeboden.

"Dat is de toekomst van de BI-tools, maar de meeste leveranciers bekijken het nog niet op deze manier; ook al zullen ze wel moeten op termijn. Het is wel interessant om te zien dat de kleinste BI-leverancier op de markt, MicroStrategy, tegenwoordig een CTO heeft die heel goed in de gaten heeft wat het betekent om BI-services te verlenen.

Vier tot vijf jaar geleden hebben de BI-leveranciers hun architectuur aangepast, volgens Madsen aan het web 1.0-model met een sterk gecentraliseerde aanpak. "Dat volstaat niet meer in de huidige praktijk met meerdere servers. Dat zullen ze moeten veranderen. Het mooiste web 2.0-product is een Open Source product van Jaspersoft. Het is het nog niet helemaal; maar gaat wel de goede kant op." Maar aan een verder gesprek over Open Source zijn we niet toegekomen.

**Teus Molenaar** is freelance journalist.



# BI-ware

## De harde en de zachte kant van Business Intelligence

BI-initiatieven mislukken nog veel vaker dan andere projecten.

De BI-initiatieven moeten van de harde en de zachte kant komen. En als de harde kant van BI al praat met de zachte, spreken ze niet elkaars taal.

Het boek BI-ware is een boek voor ICT'ers en voor managers en vertelt in gewoon Nederlands wat er allemaal fout kan gaan en wat daaraan kan worden gedaan.

BI-ware bevat een bundeling van artikelen van Karien Verhagen en is een nieuwe uitgave in de reeks van DB/M Essays. De artikelen zijn gepubliceerd in de periode 2002 – 2006.

Wilt u weten hoe u Business Intelligence kunt laten slagen?

Dan kunt u niet zonder deze uitgave!

Ga snel naar [www.array.nl](http://www.array.nl) en bestel BI-ware!

Deze uitgave is mogelijk gemaakt door: **Getronics** **PinkRocade**

**DB/M**

**Array** PUBLICATIONS