

In deze .NET-special besteden we natuurlijk niet alleen aandacht aan de Visual-tooling, maar ook naar de door Redmond standaard aanbevolen middleware. Naast bijvoorbeeld IIS en SQL Server is dat in toenemende mate ook BizTalk. In dit artikel kijken we met name naar de positionering en de vergelijking met zowel niet-Microsoft .NET-producten als de Java-stack.

thema

# BizTalk 2006 in de integratie-markt

## *Positionering en de praktijk*

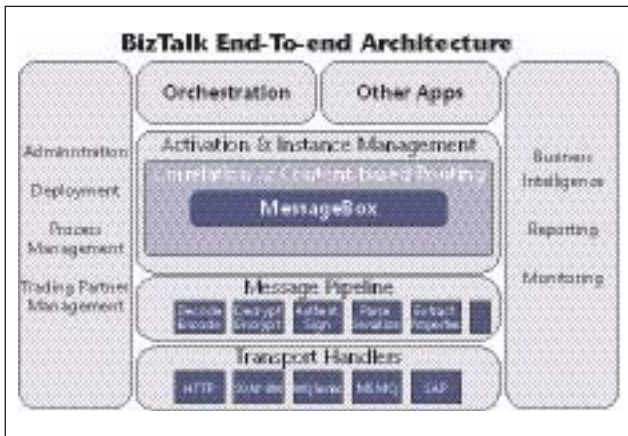
BizTalk is de Microsoft-oplossing voor 'automating and managing business processes'. Niet dat zo iets onmogelijk is met gewone Visual Studio 2005-programmering, maar er zijn heel wat situaties te bedenken waar het procesmatige gebeuren een complex en veelgebruikt pattern gaat worden dat liefst ook nog deels door non-ICT-experts onderhouden kan worden. Voor patterns gebruiken we regelmatig 4GL-achtige ontwikkeltools of standaard-middleware, net zoals een relationeel DBMS in zekere zin ook een pattern of een include-library is van miljoenen regels code die direct op het filesysteem opereren. Dit procesbeheer zit in de software-markt veelal op de EAI (Enterprise Application Integration) en B2B (Business-to-Business) terreinen. BizTalk is, kortom, zo'n 4GL-achtig platform dat specifiek bedoeld is voor procesbeheer.

In tegenstelling tot SQL Server is BizTalk vanaf het prille begin in de Microsoft-laboratoria ontwikkeld, en dat begon met versie 2000. Die was al wel gebaseerd op XSLT en XML, beide redelijk gevestigde standaards in de communicatiewereld, maar nog niet op BPEL4WS en echte standaard-webservices. Die laatste twee kwamen pas een paar jaar terug tot wasdom. Ook integratie met Visual Studio en intern gebruik van .NET-technologie is pas in latere versies volgroeid. Versie 2004 is op moment van schrijven in volle productie, maar BizTalk 2006 is reeds lang feature-frozen en in gevorderde bètatest dus kan niet lang meer op zich laten wachten. BizTalk gebruikt overigens onder de motorkap SQL Server, maar dat is relatief onzichtbaar voor de ontwikkelaar.

De belangrijkste verbeteringen in BizTalk Server 2006 zijn evolutionair. We praten vooral over het beter uitvoeren van de bestaande 2004-functionaliteit, en niet over fundamenteel nieuwe loten aan de stam. Een kort overzicht:

- Simpeler installatie, met name van de productie-runtimes. Zowel de BizTalk-componenten als onze eigen transformatiologica kunnen nu in een enkele MSI-file worden samengebracht, en een nieuwe server kan simpel unattended worden bij-geprovisioneerd.
- De procesbewaking, oftewel Business Activity Monitoring, is verrijkt met extra logica waaronder real-time subscribing en alerting.
- Ontwikkelen van transformatiologica is weer iets verder ontwikkeld, met onder andere een 'flat file schema creation wizard'.
- De nieuwste Server System-zusters worden ondersteund, zoals SQL Server 2005 en met name Visual Studio 2005. Dat laatste beperkt zich echter vooral tot code-integratie; de architectuur- en deployment dimensies van Visual Studio Team System, en diens samenwerking met tools zoals SMS, ontbeert BizTalk nog.

Voor volgende versies heeft Microsoft maar één vooruitblik gegeven, maar wel een belangrijke. Windows Workflow Foundation, dat samen met Vista ontwikkeld wordt maar er in principe los van staat, wordt een rules engine en dialogen-generator die enigszins naast BPEL staat. Deze WWF vult nu nét de lacune op in de markt-



AFBEELDING 1. BizTalk 2004 hoofd-architectuur.

standaarden rondom procesflow en BizTalk. WWF gaat ook de motor vormen in andere Microsoft-tools zoals Identity Integration Server en de CRM-applicaties, en door deze synergie zal op termijn BizTalk ook beter met die Microsoft-omgevingen kunnen samenwerken.

**PRODUCTGROEPEN IN EAI-TOOLMARKT** Om een idee te geven waar BizTalk inzetbaar is, en hoe hij zich tot de Windows-concurrentie binnen én buiten de

.NET-stack verhoudt, lopen we twee keer bijgaande tabel langs. De eerste keer kijken we naar de kolom 'productgroepen' en zien we, niet gerelateerd aan de .NET- en Java-stacks, welke 'toolniveaus' (tweede kolom) onder die groepen vallen. In de tweede ronde kijken we juist wel naar de verschillen tussen de stacks, dus de twee rechterkolommen; en naar de keuzen om binnen de .NET-stack al dan niet BizTalk in te zetten.

We beginnen, enigszins verrassend misschien, in het midden; de tabel toont namelijk vooral het continuüm van real-time integratie, de ETL-rij onderin is slechts een niet onbelangrijk aanhangsel. Dat continuüm vinden we in een functionaliteits-stack die begint bij het operating system, en eindigt bij zwaargewicht tools zoals BizTalk.

In de onderste helft van dat continuüm zit dan de tooling die als verzamelkop 'ESB' (Enterprise Service Bus) kan dragen. Dat zijn primair gewone bouwtools zoals Visual Studio of Eclipse. Onder dit hoofdje scharen zich ook een aantal pakketten die zich 'ESB' noemen en zoveel mogelijk code-loos claimen te werken, onder andere van Sonic Software, maar daar gaan we

Productgroep	Niveau	.NET	J2EE (bijv. WebLogic of JBoss)
Process Manager ESB	Process Management (BPEL, BAM)	Corefunctie van BizTalk, voor beperkte BPEL ook 3rd party tools	· Basis-BPEL ingebouwd in de application server · Zwaarder BPEL en BAM eist Integration-versie van het AS-tool
	Dialog-workflow	Third-party plugin	Soms third-party soms (o.a. WebSphere en Weblogic) eigen plugin
	Datatransformaties	Indien 1-op-1:  XSLT etc. patterns binnen VS2005. Geen grafische transformation editor  Indien complexer: BizTalk (n-op-m flows, editor)	Breed scala:  Basis-XSLT zit in de applicatie-server, complexe n-op-m flows zitten in de Integration-versie van het tool, ook veel third party tooling die tussensmaken levert.
	Integratie-drivers voor o.a. pre-SOAP pakketten	BizTalk Server (o.a. de iWay set) of third-party	Met J2EE-platform gebundeld of soms losse extra's, via JCA
	Basis-runtime en drivers	· Runtime: .Net framework  · Drivers: via extra MS tools, gratis (IIS) of betaald (o.a. MSMQ)	J2EE server. Basisdrivers (werkend met JCA) altijd inbegrepen.
	Operating System & DBMS	Communiceren met file systems en SQL drivers	File systems / SQL drivers
ETL	Nodig voor ETL	Ofwel .NET en third-party tooling, ofwel SQL 2005 Integration Services, ofwel BizTalk.  De laatste niet geoptimaliseerd voor deze taak.	DBMS (zoals Oracle of DB2) en dan gestripte J2EE-server plus JCA-drivers. Geoptimaliseerd voor ETL.

TABEL 1. Vergelijking Integratie-stacks van J2EE en .NET op Windows.

later dit jaar in Release Magazine eens naar kijken; voor BizTalk zouden ze de vergelijking compliceren dus laten we ze even weg.

Binnen de ESB-productgroep onderkennen we dan, van onder naar boven:

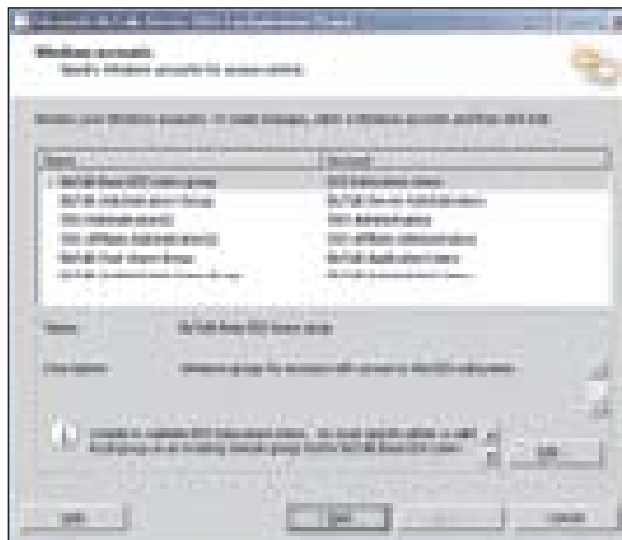
- Het OS, Windows in dit geval, plus de database waarin we zaken willen opslaan.
- De runtime voor onze applicaties, die zelf geschreven zijn in bijvoorbeeld C# of Java. Bij die runtime zijn dan ook een aantal 'connectivity-drivers' nodig, om bijvoorbeeld via HTTP-SOAP, MQSeries of allerlei midframe-protocollen te kunnen communiceren.
- Bij veel leveranciers bevat dit basisoniveau echter onvoldoende drivers voor alle gewenste koppelingen. Regelmatig moeten we extra betalen om bijvoorbeeld standaardpakketten die geen SOAP kennen native aan te kunnen roepen. Dit is het niveau 'integratiedrivers'.
- Het bovenste ESB-niveau heet 'datatransformaties'. Maar dit blijkt een gemeleerde groep: het kan lopen vanaf simpele 1-op-1 record-omzettingen tot complexe n-op-m flows met weggefilterde en toegevoegde velden. We zouden kunnen zeggen dat de onderhelft van de transformatie-groep nog onder de ESB-vlag valt, en dat de bovenste helft al onder de zwaargewicht productgroep ressorteert.

Die zwaargewicht groep noemen we 'Process Managers'; dit zijn commercieel gezien min of meer de opvolgers van het uitstervende ras der 'Integration Brokers'. Het verschil tussen beide heeft vooral te maken met eveneens naar een niche verbannen termen zoals batch-integraties en proprietary EDI. Ook voor deze groep wandelen we de tabel door, van onder (basisbouwsteen) naar boven (integrerende laag):

- De bovenste helft van de transformaties, met zaken zoals n-op-m flows waar de nodige 'intelligentie' in de omzetting gestoken wordt.
- Dialoog-workflow: een proces-voortgang die tijdelijk gestopt wordt totdat een gebruiker een actie genomen heeft. Hierbij horen ook concepten zoals worklists of tasklists.
- Samengestelde flows ofwel het echte procesmanagement. Dat wordt gekenmerkt door twee hype-woorden:
  - a) Ondersteuning van BPEL4WS, ook aan geduid als BPEL, Business Process Execution Language for Web Services. BPEL is een XML-taal die definieert hoe individuele service-calls 'aan elkaar geregen worden', mede acterend op de resultaatcodes van de calls die eerder in de keten zitten. Tevens is BPEL in staat om een actie te parkeren totdat triggers binnengekomen zijn of bijvoorbeeld een (non-BPEL) task uitgevoerd is. BPEL zelf standaardiseert dat parkeren overigens niet, maar leveranciers-extensies voor dat doel zijn soepel in te pluggen.

b) BAM ofwel Business Activity Monitoring. Dit is het verkrijgen van management-informatie, bij voorkeur via mooie drilldown-dashboards, over onze processen. Bijvoorbeeld 'hoe lang doet een order er gemiddeld over van entry tot aflevering, en hoe is dat over de procesfasen verdeeld?' Of 'hoe groot zijn de werklijsten en parkeer-statussen van overwerkbriefjes, en welke afdelingen vormen flessenhalzen?'

De onderste rij is zowel een productgroep als een niveau, en beiden heten ETL. Dat staat voor Extract-Transform-Load, en is de gangbare methode om vanuit productiegegevens de gewenste data naar een datawarehouse over te brengen. Bij de Transform-fase van het ETL-proces komen we vraagstukken tegen die sterk lijken op die van EAI-datatransformatie, en ook hier zijn elementen zoals XSLT-stylesheets erg nuttig. Op dit vlak gedragen de .NET- en de Java-wereld zich vergelijkbaar. Er zijn de nodige misverstanden rondom BizTalk en ETL, mede veroorzaakt door de stack-invulling van Microsoft; vandaar dat we straks de Java- en .NET-keuzen ook voor ETL bekijken.



FIGUUR 2. Configuratiescherm accounts voor BizTalk 2004

**BIZTALK POSITIONERING** We kunnen nu wederom de tabel doorlopen, dit keer van boven naar onder, en nu specifiek kijkend naar de Microsoft- en Java-invullingen. De reden om de tabel te gebruiken is dat we een antwoord willen vinden op de vraag: voor welke taken is BizTalk wél en voor welke minder geschikt?

Een zeer algemeen antwoord is: overal waar langerlopende processen gevolgd moeten worden of meer complexe datatransformaties in beeld zijn. Denk dan vooral aan de Process Manager-categorie. Microsoft positioneert BizTalk op deze wijze als evenknie van SQL Server die bijna even vaak in totale systeem-oplossingen voor-

komt als de Windows Server-zuster. Bij een aantal van die langer lopende processen zijn ook nog third-party componenten nodig. Zolang WWF er nog niet is, moeten dialogen immers met non-BizTalk plugins gebouwd worden, gebruik makend van de Human Workflow Services-bouwstenen die BizTalk zelf al biedt.

Uit de tabel zien we, voor de Process Manager-niveaus, nog een aantal interessante punten zowel binnen .NET als in de Java-vergelijking:

- BPEL hoeft niet zo 'high-end' te zijn op de manier waarop het vrij dure BizTalk wordt neergezet. Bij de Java-concurrentie is basaal BPEL al aanwezig in standaard JBoss, WebSphere en WebLogic - dus de basis-runtime van het platform. Alleen als we extra's zoals BAM willen verwijzen ook onder andere IBM en BEA ons naar de zwaardere 'Integration editions'. Ook voor .NET zijn basale BPEL-engines te koop die met Microsoft concurreren.
- Ook voor datatransformaties, op de grens van Process Manager en ESB, kent de Java-wereld veel gradaties. Bij Microsoft zijn de smaken primair 'simpel dan gewone Visual Studio 2005 en complex dan BizTalk', bij J2EE-platforms zijn naast die extremen ook allerlei simpeler transformatie-plugins leverbaar.

In het ESB-gedeelte van de tabel zien we nog een interessant punt: de enige manier waarop Microsoft zelf, in een standaard ESB-achtige applicatie met Visual Studio, integratiedrivers aanbiedt is wederom via BizTalk - de 'Transport Handlers'! Dat is enigszins het paard achter de wagen spannen: de handler-kant van BizTalk is misschien maar vijf tot tien procent van diens functionaliteit, en dan wordt het een erg dure driver-set. Drivers vergelijkbaar met de iWay-set, die recentelijk door Microsoft ingebouwd is, zijn ook van third-party leveranciers te kopen en zeker de moeite waard om direct met Visual Studio 2005 en .NET te implementeren. Overigens ontbeert .NET een generieke API om dit soort drivers aan te sturen; hoewel het pricing-model van J2EE-servers min of meer vergelijkbaar is met .NET heeft men in Java-land tenminste de generieke Java Connectivity Architecture (JCA) interface. Verder ontlopen de platform-stacks elkaar niet zoveel. Ook in de cellen 'basis-runtime en drivers' zien we dat Java een klein voordeeltje heeft met de JCA-interface, maar Microsoft zorgt wél voor aanwezigheid van alle essentiële drivers via vrijelijk beschikbare tools zoals IIS.

Onderin de tabel komen we weer het ETL-'buitenbeentje' tegen. In de Java-kant blijkt men deze categorie goed te kunnen afhandelen met de J2EE-code die meer en meer tegen de databases wordt aangeplakt; zowel IBM als Oracle doen hun datatransformatie en bulk import/export namelijk meer en meer met JCA-drivers.

Aan de .NET-kant zou het ook veruit de beste manier

zijn om de batchverwerking rondom, veelal, SQL Server óók aan losse drivers en speciale third-party tooling over te laten. Of aan het nieuwe SQL 2005 Integration Services, de opvolger van SQL 2000 Data Transformation Services. Wat blijkt echter? Vanuit hun verknochtheid aan een Microsoft-only oplossing, of simpelweg bekendheid met BPEL en onbekendheid met SQL Server, zetten Microsoft-sites geregeld BizTalk voor deze ETL-taken in. Dat levert regelmatig een slechte reputatie op, omdat BizTalk duidelijk geoptimaliseerd is voor realtime-flows en niet voor dit massale batchverkeer. Het product is dus simpelweg te langzaam en te duur voor ETL.

**EEN PAAR CONCLUSIES** We kunnen na deze vergelijkingen wel een paar conclusies formuleren over BizTalk Server, geldig voor de 2004- maar ook de nieuwe 2006-versie.

- BizTalk is een uitstekende 'Process Manager', direct vergelijkbaar met de zwaargewicht Integration-edities van de applicatieservers van onder andere BEA en IBM.
- Voor taken die lichter zijn dan complexe transformaties en flows bevat het productgamma van Microsoft een gat: de basistaken van Visual Studio en de CLR omvatten wél transformaties binnen moderne formaten zoals XML en CSV maar geen integratie-drivers. Dat soort drivers biedt Microsoft zelf alleen via de Transport Handlers van BizTalk.
- Hoewel BizTalk niet gepositioneerd wordt voor datawarehouse ETL-batchtaken, en daarin ook niet preformt, zetten gebruikers hem daar toch wel eens voor in. Zou dat niet te maken kunnen hebben met de onbekendheid met third-party transformatiedrivers, gerelateerd aan het eerder genoemde gat in het Microsoft-productgamma?

Dat doet echter geen afbreuk aan de opkomst van Process Managers binnen Windows-totaaloplossingen of het Microsoft-beleid om BizTalk Server haast even vaak in totale systeem-oplossingen te laten voorkomen als SQL Server. BPEL is een grote toekomst beschoren, en BizTalk helpt om die toekomst vorm te geven.

*Erik de Ruijter RI is software architect.*