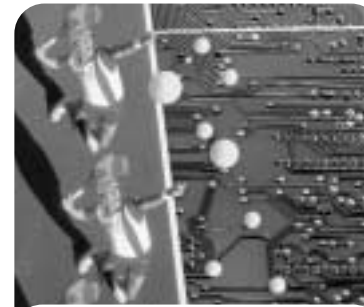


Microsoft-prominent Don Box was afgelopen maart even in Nederland voor een evenement ter ere van de lancering van Visual Studio .Net en de oprichting van de .Net sectie van de Software Developers Group Netherlands (SDGN). Don Box geniet een reputatie als 'goeroe' op het gebied van component based development en als gewaardeerd spreker op "veel te veel" (sic) software-conferenties in de vreemdste uithoeken der aarde. Hij is sinds enige tijd in dienst van Microsoft om het .Net evangelie te verkondigen. De overgang naar Microsoft maakte velen nieuwsgierig of hij zou blijven grossieren in provocerende uitspraken. Jim van Dam nam voor Software Release Magazine polshoogte.



thema

Don Box: 'Meta data is the key thing'

.Net belooft gepassioneerde relatie

Don is mede-oprichter van opleiding- en trainingenbedrijf DevelopMentor en auteur van verschillende boeken over XML en COM. Hij is mede-auteur van de SOAP-specificatie en lid van de W3C Schemas Working Group. Daarnaast is hij redacteur en columnist voor het Microsoft-periodiek MSDN Magazine en (af en toe) de O'Reilly-website XML.com. We waren benieuwd of Don webservices op dit moment eigenlijk 'klaar' acht voor serieuze toepassingen.

Box: 'Als je het web zelf, los van webservices, geschikt vind voor serieuze toepassingen: ja. Webservices zijn gebouwd op HTTP en XML. HTTP wordt al gebruikt voor miljarden dollars aan transacties. Het transport is OK. De core technologie is niet slechter dan wat het web biedt. Het kan uiteraard veel beter, maar dat is geen reden om het nog niet te gebruiken.'

Dan heb je het over Internet, maar wat vind je van het gebruik van webservices over een intern netwerk?

Box: 'De toekomst zal waarschijnlijk gebaseerd zijn op alomtegenwoordige XML berichtenverkeer over voorlopig HTTP. Misschien zal HTTP in de toekomst worden vervangen, maar niemand is daartoe bereid op dit moment. Het XML onderdeel zal waarschijnlijk nog belangrijker worden dan HTTP. Er gebeuren al interessante dingen rond XML en dat wordt alleen maar interessanter.'

BEVEILIGINGSTRUC *HTTP was oorspronkelijk bedoeld voor het GET / POST commando bij het browsen. Hoe bruikbaar blijft dat protocol voor de doelen die worden nagestreefd met Webservices?*

Box: 'Het POST commando is interessant want dat is eigenlijk standaard RPC (Remote Procedure Call). Het enige dat we gedaan hebben met SOAP (Simple Object Acces Protocol) is POST gebruiken om ons doel te bereiken. We maken het daarmee niet minder veilig of veiliger dan wat POST biedt. Veel mensen zeggen over



Don Box is mede-oprichter van opleiding- en trainingenbedrijf DevelopMentor en auteur van verschillende boeken over XML en COM.

SOAP: "Ja, dat is een beveiligingstruc om door de firewall heen te komen". Wij hebben altijd getracht om via de specificaties in de berichten duidelijk te maken dat het om methoden aanroepen over het Internet ging. Daarmee is het eenvoudig om SOAP berichten te herkennen en eventueel te blokkeren.'

Hoe zou je moeten omgaan met de betrouwbaarheid van HTTP berichten? Ze kunnen verloren gaan of veel later als nog beantwoord worden.

Box: 'Mensen starten met rechttoe rechtaan berichtenverkeer en bouwen daar overheen een laag die

'De webservices aanpak zal alomtegenwoordig worden, ook in een gesloten interne omgeving'

betrouwbaarheid garandeert. Dat is ook wat wij willen gaan doen. Microsoft werkt aan betrouwbaar berichtenverkeer met XML. BizTalk ondersteunt 'reliable messaging' en ook de MSMQ hebben specificaties uitgebracht voor betrouwbaar berichtenverkeer. Dat gezegd hebbende, er is nog een heleboel te doen op dit terrein door zowel Microsoft als alle anderen.'

Verwacht je de vooruitgang op dit gebied van zoiets als HTTPR of meer in de richting van producten als MSMQ, BizTalk en WebSphere MQ?

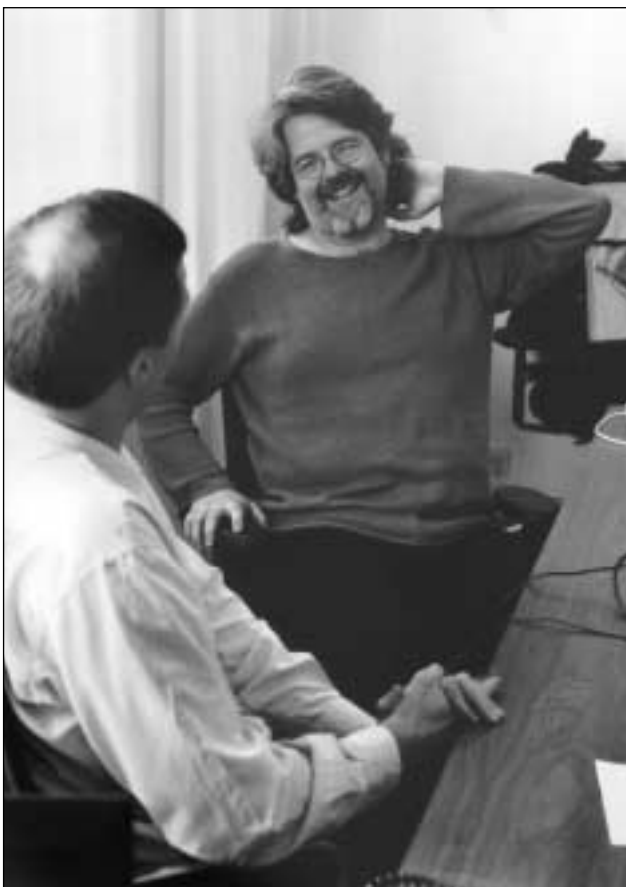
Box: 'Ik verwacht meer van de producten dan van een protocol als HTTPR. HTTPR lost het probleem niet op. Tenminste niet op het juiste niveau. Het gaat niet om het verbeteren van HTTP, maar om het ondersteunen van 'reliable messaging' over willekeurig welk protocol. Aan de andere kant van de wereld bieden anderen de JMS (Java Message Service) mensen oplossingen. Ik weet dan ook niet of bedrijven echt dergelijke producten als MQ of JMS nodig hebben. We moeten in ieder geval een duidelijk beeld krijgen van hoe een dergelijke 'reliable message' eruit ziet als het over de lijn gaat en dat hebben we nog niet. Zodra we dat wel hebben, wordt het ook mogelijk voor andere leveranciers om met eigen oplossingen te komen. Ik verwacht dat dergelijke producten dan gemeengoed worden. Dat is dan juist voor Microsoft wel een voordeel. Zij verdienen hun geld niet voornamelijk met dit soort software. Gespecialiseerde software leveranciers van dit soort producten gaan het moeilijker krijgen om geld te verdienen als het zo gemeengoed wordt en overal al in zit.'

Juist omdat SOAP over HTTP zo simpel is kan iedereen daar parsers voor schrijven. Dat lijkt een groot voordeel.

Box: 'Precies, je kunt je toch niet voorstellen dat bedrijven een businessplan baseren op het verkopen van XML parsers. Gelukkig maar, sommige technologie moet ook gewoon gratis zijn. Over drie jaar zal reliable messaging onderdeel uitmaken van de services van het OS en andere producten.'

TAAL VERSUS FRAMEWORK *Hoe zie je positie van de twee kampen J2EE en .Net?*

Box: 'Vroeger zag ik dat ook als twee kampen die tegenover elkaar staan, maar tegenwoordig niet meer. Webservices heeft dat volledig veranderd. De vraag J2EE of .Net is op zichzelf niet meer zo belangrijk. Ik zie J2EE als Windows DNA maar dan in Java. Een transactioneel Object Management System met een op HTTP gebaseerde toegang en op HTML templates gebaseerde presentatie ervoor. Microsoft had dat al met bijvoorbeeld ASP en MTS. Sun deed dat na met JSP en EJB, maar dat zijn oplossingen voor problemen uit begin jaren negentig gebaseerd op ideeën uit eind jaren tachtig. Het voordeel van het platformafhankelijk zijn is niet zo heel belangrijk meer. Webservices heeft het strijdtoneel verschoven naar spelers als Microsoft met .Net, IBM met WebSphere en BEA met Weblogic. Nu zie je ook de individuele leveranciers langzaam wegdrijven van J2EE.'



Jim van Dam in gesprek met Don Box.

J2EE kiest voor de taalaanpak waarbij de webservices ondersteuning in de taal zelf zit en niet in een framework zoals bij .Net.

Box: 'Zeker, maar dat veronderstelt wel dat alle leveranciers samen overgaan naar een dergelijke aanpak. Mensen vinden het interessant om twee kampen te zien die tegenover elkaar staan, maar er wordt in het Java kamp intern ook flink geknukt. Sun is onder de voet gelopen door IBM en BEA vecht om overeind te blijven. Microsoft heeft goede relaties met IBM. We hebben ze betrokken bij WSDL (Web Services Description Language), we hebben samen SOAP opgezet en we doen nog steeds allerlei dingen samen zoals UDDI. Dat vindt BEA niet zo leuk. Het is dus niet meer Microsoft tegen de rest van de wereld. Daardoor is het ook niet meer bij voorbaat interessant geworden om 'niet Microsoft' te zijn of juist 'gerelateerd aan J2EE'. Dat denk ik te zien gebeuren in ieder geval en dat dacht ik al voordat ik bij Microsoft kwam werken.'

GENERIEKE AANPAK *Mensen die van DCOM applicaties af komen die missen bij webservices nog wel functionaliteit zoals transactie management. Hoe kijk je daar tegenaan?*

Box: 'Webservices kunnen wel transactioneel zijn binnen de service zelf, maar je kunt geen dingen doen zoals 2PC (2-Phase-Commit) over service grenzen heen. De vraag is of mensen dat echt wel nodig hebben. Men denkt vaak dat men dat nodig heeft en daarom zullen we dat gaan ondersteunen. We zullen dat gaan bieden, omdat men daar naar vraagt, maar ik geloof er absoluut niet in dat het veel gebruikt zal worden. Het probleem is dat het locking-model niet schaalbaar is voor een asynchrone langzame wereld als het Internet.'

Toch zal in een wereld die meer gebruik maakt van Webservices de behoefte zijn om een actie die uit het aanroepen van diverse services bestaat te ondersteunen bij het terugdraaien van de gevolgen als het onderweg mis gaat.

Box: 'Ja, het Microsoft antwoord daarop is: gebruik BizTalk voor dergelijke ondersteuning. Wij zullen ons best doen om de BizTalk functionaliteit beschikbaar te maken voor de massa. Op dit moment is het nog een van de duurste producten die we leveren. Het model dat in BizTalk zit zal steeds verder worden verspreid. De reden dat we toch ook 2PC gaan ondersteunen over grenzen heen, is niet dat we denken dat je dat moet doen over organisatiegrenzen heen, maar omdat we zien dat de webservices aanpak alomtegenwoordig zal zijn en dus ook wordt gebruikt in een gesloten interne omgeving. Webservices moet dan de standaard aanpak worden voor alles wat je ontwikkelt.'

Ik ben nog niet overtuigd dat je met een dergelijke generieke aanpak in zijn huidige vorm optimale resultaten



'Het model dat in BizTalk zit zal steeds verder worden verspreid'

behaalt voor zowel intern- als extern-gerichte software.

Box: 'Nee, maar het zou wel leuk zijn om een gemeenschappelijk technisch fundament te hebben dat iedereen kan gebruiken. Op dit moment heeft bijvoorbeeld IBM oplossingen als LU6.2, WebSphere MQ, HTTP voor WebSphere, IIOP (Internet Inter-ORB Protocol), RMI (Remote Method Invocation) en dat zijn allemaal 'stovepipe protocols' om een bericht van A naar B te sturen. Wij willen naar één standaard fundament.'

OK, maar dan willen bedrijven dat ook voor interne oplossingen over en dan missen ze ten opzichte van hun huidige oplossingen faciliteiten als connection en sessiemanagement.

Box: 'Het team waarin ik werk ontwikkelt GXA (Global XML Web Services Architecture) met daarin onder andere WS- Routing en WS-Referral specificaties.

'Java is als een mooie vrouw met slechte adem'

Ook 'reliable messaging' en ondersteuning voor dialoogachtig berichtenverkeer zullen worden uitgewerkt in komende specificaties binnen GXA.'

OVEREENSTEMMING *De uitbreiding van specificaties is hard nodig, want nu moet iedereen het oplossen in de applicatie zelf.*

Box: 'Inderdaad, dat is vervelend. Het moet onderdeel worden van het platform. De WS-Routing specificaties zijn het meest richtinggevend voor waar we naar toe willen. Het tracht een abstractielaag te bouwen om de afhankelijkheid van HTTP weg te halen. WS-Routing heeft eigen adresformaat semantiek en een simpel model voor berichtencorrelatie voor Request/Response.

Het is volledig onafhankelijk van HTTP. Het werkt uitstekend over HTTP, maar als je iets wilt doen waarbij je een bericht uitstuurt en het antwoord komt pas na vijf dagen (en dat is geen fout), dan is HTTP niet geschikt. WS-Routing streeft naar een virtueel netwerk en als dat toevallig HTTP spreekt, mooi, maar dat is niet noodzakelijk. Zo komen er nog meer specificaties van proto-

all self-discovering, self-describing and self-healing". Onzin volgens mij. Ik wil kijken hoe ver ik met de onderliggende technologie kan komen vanuit het gezichtspunt van een ontwikkelaar.'

DOEL VAN DE ONTWIKKELAAR *Als ik nu overga naar de Common Language Runtime, wat doe ik dan met mijn explicit memory management, buffer handling en RTTI (Run-Time Type Information) code?*

Box: 'Die hoeft je niet overboord te gooien. Het verschil met talen als Visual Basic en Java is dat wij de CLR hebben gemaakt om ook onze eigen producten mee te bouwen. Visual Basic is niet ontwikkeld om door ons intern veel gebruikt te worden. We hebben Visual Basic

5 niet gebruikt om Visual Basic 6 te maken. Sun heeft ook geen Java gebruikt om Solaris te herschrijven. De CLR wordt wel degelijk veel door ons gebruikt voor onze eigen producten. De CLR dwingt je niet om al die performancetechnieken af te zweren. We proberen wel het programmeermodel van expliciet naar impliciet geheugenbeheer te migreren. Daardoor kun je voor performance gevoelige software nog steeds optimaal implementeren. Je hebt in C# ook een 'unsafe' mode die je dat voor bepaalde stukken code toestaat zonder runtime kosten zoals bij de Java Native Interface.'

Wat vind je het belangrijkste aan de Common Language Runtime, die samensmelting van talen?

Box: 'Nee, daar wordt wel veel over gesproken, maar ik vind de meta data veel belangrijker. Het zit overal in en niet zoals bij COM en type library's soms wel en soms niet. Het ultieme doel van de meta data is om duidelijk te maken wat het doel van de ontwikkelaar van die code was: hoe moet de XML die jouw type representeert eruit zien en hoe ziet de COM interop eruit? Dat soort dingen kan nu perfect worden vastgelegd in uitbreidbare meta data. Ik heb het altijd gewild en ik verwacht dat anderen zoals BEA dat ook gaan implementeren. Meta data is mijns inziens 'the key thing'.

'Het voordeel van het platformonafhankelijk zijn is niet zo heel belangrijk meer'

collen die je semantiek bieden op applicatie niveau over de diverse netwerkprotocollen die er nu al zijn. Die zijn op zich prima. Ik heb bijvoorbeeld geen problemen met HTTP performance.'

Tenzij je nog aan HTTP 1.0 vastzit of als er ergens een proxy tussen zit die HTTP 1.0 praat.

Box: 'Ja, dan heb je pech. De performance van 1.1 is niet het probleem, maar wel de complexiteit als je iets probeert waarbij bijvoorbeeld het antwoord op jouw verzoek moet binnenkomen op een apart adres. Dat soort functionaliteit ondersteunt WS-Routing. WS-Routing zegt mijns inziens veel over waar wij naar toe willen. Door de specificaties vroeg vrij te geven hebben we bovendien de mogelijkheid om erover te praten met hardwareleveranciers die het in de hardware kunnen bakken.'

WSDL lijkt mij een logische stap: je definieert interfaces zodat je beter bereikbaar bent. UDDI lijkt mij nu nog wat vergezocht.

Box: 'OK, ik wil hier de Schema-Repository-View niet verdedigen. Ik ben het net zelf een beetje gaan accepteren, daarvoor negeerde ik het als volkomen onbelangrijk. WSDL vond ik zo belangrijk dat ik het liefst eerst WSDL had neergezet voor SOAP. Helaas was het nog lastiger om overeenstemming te bereiken over WSDL dan bij SOAP. De vraag met betrekking tot UDDI is: wat betekent 'discovery'. Het meeste doe je tijdens ontwikkeling van de software. Andere soorten 'discovery' doe je tijdens installatie, omdat je dan exact weet met welke adressen je wilt koppelen. Weer een ander gedeelte gebeurt tijdens run-time gedurende een conversatie tussen twee componenten. We hebben als software industrie nog geen goed beeld van wat 'discovery' inhoud en de diverse vormen ervan. Nu roepen mensen wat al te makkelijk: UDDI als oplossing voor alle 'discovery' problemen. Mijn doel voor het komende half jaar is meer inhoud te geven aan UDDI. Persoonlijk had ik ook niets met de mensen die erover spraken in de trant van: "it's



'Het leuke van Microsoft is dat je je eigen onderwerpen kunt uitkiezen'

Die multilanguage ondersteuning is mooi, maar dat je Visual Basic en C# bij elkaar kan brengen verbaast mij niets. Dat we ook in staat zijn om iedere C++ code, ieder MFC of ATL programma kunnen compileren voor de CLR zonder aanpassing, dat vind ik wel knap. Vervolgens kun je stap voor stap over gaan naar meer managed code. Dat was uiteraard ook belangrijk voor Microsoft want we hebben een substantiële investering in C++ code in onze producten.'

Zie je al andere talen die minder gelijksoortig zijn die goed werken binnen de CLR zoals Eiffel en Smalltalk?

Box: 'De eerste release van de tweede versie van Eiffel voor de CLR ziet er al goed uit. Die versie ondersteunt de volledige taal. Wat ik wel jammer vind is dat er nog geen uitgebreide Python release voor de CLR is, want dat lijkt mij heel goed haalbaar en zeker interessant. We hebben niet genoeg ontwikkelaars bij Microsoft om dat zelf te doen. Wij zijn maar een kleine club. Ongeveer 45.000 mensen en volgens mij werken er ongeveer 43.000 bij marketing.'

SUPERONTWIKKELING *Wat kan een software engineer op dit moment met de specificaties uit GXA?*

Box: 'Ze zijn interessant om te lezen, want als ze de problemen adresseren die je hebt dan kun je je beter conformeren aan de richtlijnen die erin staan. Daarmee heb je een grotere kans dat de oplossingen die je bouwt passen binnen de standaarden die zullen ontstaan. Net als bij SOAP. SOAP is niet zo radicaal, maar als je in 1999 de specificaties hebt gelezen en in die richting bent gaan ontwikkelen dan kun je nu meeliften met de producten die uitkomen.'

Als ik die specificaties lees, kan ik mij voorstellen wat voor soort producten daar nuttig bij zouden zijn, maar ik weet ook dat die op dit moment al worden gebouwd door onder anderen Microsoft en IBM en over een jaar zullen uitkomen.

Box: 'Klopt, maar het heeft mij verbaasd hoeveel kleine leveranciers toch bovenop het ontwikkelen van producten rond XML zijn gesprongen en nu marktwaarden in de negen cijfers hebben. Ik begrijp niet dat mensen daar zoveel geld voor zouden willen geven. Ik verwacht dat de meeste bedrijven die dat doen zullen verdwijnen en enkele zullen worden overgenomen. Toen wij begonnen met de SOAP specificaties heb ik ook parsers en marshallers geschreven, maar als weggooiproducten. Ik had nooit verwacht dat men voor dergelijke producten geld over zou hebben. Op dit moment wordt de situatie wel anders want de grote jongens geven die producten nu gratis weg.'

Volg je de ontwikkeling van specificaties bovenop XML zoals MathML en SMIL?

Box: 'Nee, die niet. Ik vind ontwikkelingen als XML Query wel heel interessant. Database leveranciers gaan



'Je ziet individuele leveranciers langzaam wegdrijven van J2EE'

dat soort dingen direct in de database ondersteunen. Het bouwt voort op XPATH 2.0 en biedt een syntax zoals SQL, maar in het formaat van XML. Een XML Query expressie is zelf ook weer XML. Ik vind dat een super ontwikkeling en ben blij dat de mannen in gebouw 35 bij Microsoft daar zoveel energie in steken want ik word een gebruiker. Het leuke van Microsoft is dat je je eigen onderwerpen kunt uitkiezen en je als gebruiker kunt opstellen bij de rest.'

BADSCÈNE *Je website was onlangs uit de lucht, maar in de cache van Google stond nog een pagina met een fraaie vergelijking tussen COM en een lelijke vrouw.*

Box: 'De site was inderdaad even uit de lucht want mijn vrouw deed de verhuizing van Los Angeles naar Seattle en de servers zijn ook mee verhuisd. De vergelijking gaat als volgt:

COM is net een onaantrekkelijke vrouw met een hart van goud: er zijn drempels maar als je die overwint heb je een goede en bevredigende relatie, Java is een mooie vrouw met slechte adem: je voelt je direct tot haar aangetrokken maar de rest van de relatie wordt je steeds met die blijvende tekortkoming geconfronteerd en .Net is net een prachtige vrouw die een passievolle relatie belooft, maar zonder ex-minnaars dus niemand om aan te vragen hoe ze is.'

Vorig jaar trok je op de TechEd conferentie nogal de aandacht door tijdens je presentatie in een badkuip plaats te nemen. Heb je al plannen om bij de volgende presentatie op TechEd die "badscène" te overtreffen?

Box: 'Ik heb hard gewerkt om de mensen niet teleur te stellen die verwachtingen hebben wat dat betreft.'

Jim van Dam is werkzaam bij Ordina Finance SDS (e-mail: jim.van.dam@ordina.nl)

Fotografie: Dré de Man.