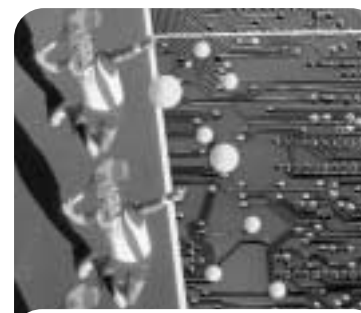


Nu gaan we het krijgen. Nee, deze keer écht. De poorten van Informatica-Nirvana staan wagenwijd open, binnen ligt interoperabiliteit op een bedje van nieuwe standaarden lonkend op ons te wachten... alleen wordt voor de deur nog een dapper robbertje gevochten. Het gevecht om 'DE' architectuur in de middleware markt neemt mythische vormen aan. Het boze oog met een leger Orcs aan de ene zijde, aan de andere kant het gezellige amalgaam van goedmenende Mensen, liederlijke Elfen, vuurvaste Dwerfen, rusteloze Enten, machtige Wijzen en vooral verrassende Hobbits. Van uw oorlogsverslaggever ter plaatse, een overzicht van stellingname en positie spel...



thema

Webservices oorlog tussen J2EE en .NET

Overzicht van stellingname en positie spel

Na het puur op netwerkniveau interconnecteren van zowat alles wat computer heet op deze aardkloot mogen we ons stilaan opmaken voor het koppelen op een meer 'zingevend' applicatieniveau. Dat dit geen vanzelfsprekende operatie wordt mag blijken uit de als mislukt gecatalogeerde pogingen uit het verleden zoals CORBA, DCOM. Niet dat die gedistribueerde component modellen niet aan het werk te krijgen waren, ze gingen alleen nog net iets teveel uit van het grote gelijk. Alles werkt namelijk samen met gedistribueerd systeem X in zoverre alles gedistribueerd systeem X is. Niet de netwerkstandaards maar de gebruikte verschillende data-representatie technieken blijken in de hele applicatie-integratie beweging het grootste zeer.

INFORMATIE KIDNAPPEN De onlangs opduikende term data-hijacking (vrij vertaald als kidnappen van informatie) toont eigenlijk aan waar we als gebruikersgemeenschap heel lang aan hebben meegewerkt: onze eigen bedrijfsgegevens en operationele data vertrouwen we toe aan systemen waaraan we nadien vriendelijk moeten gaan vragen of we de opgeslagen gegevens alstublieft opnieuw mogen inkijken, verder bewerken en uitwisselen met iemand die helaas over een ander systeem beschikt. Ook op het meest triviale niveau gaat dit op: Van elk MS-Word document dat we per dag schrijven leveren we het recht op toekomstig gebruik ervan zonder nadenken af aan het Microsoft Word programma. De afhankelijkheid wordt nooit zo aangevoeld,

maar zonder dat programma (met licentie graag, en straks jaarlijks) in de buurt doe je niet veel meer dan nakijken of de file er nog wel is of niet¹. Door de algemene aanwezigheid en de democratische prijs van het programma in kwestie kun je in de praktijk nauwelijks van een probleem spreken natuurlijk. De denkpiste zou ons alleen wel een iets meer geëmancipeerde kijk op het IT gebeuren moeten geven. Zeker in tijden als deze waar men onder het mom van totale onthechting en openheid toch weer een strijd gaat voeren omtrent 'het grote gelijk'.

MICROSOFT BEPAALT DE AGENDA Waar gaat het nu eigenlijk om? Volgens Microsoft gaat het om 'interoperabiliteit'. Een woord dat dan vooral wordt uitgesproken als tegenstelling tot 'porteerbaarheid' wat als marketingterm in de mond van Sun wordt gelegd. Dit is met voorsprong de meest rake diagnose van de laatste jaren en Microsoft heeft er zich terecht meteen de aandacht van de corporate wereld mee toegeëigend. Sans

1 Er is trouwens goed nieuws over die beruchte niet standaard Office formaten. Op <http://jakarta.apache.org/poi> leeft sinds enkele maanden het POI project wat een Java API aanbiedt voor het behandelen (lezen en schrijven) van deze binaire formaten. Allen daarheen. Dankzij de porteerbaarheid van de onderliggende Java openen we eindelijk de zelf gecreëerde legacy van Office documenten naar niet Windows platformen. Een aantal demo's binnen het Cocoon project (<http://xml.apache.org/cocoon>)

rancune, en weg die herinneringen aan ooit als monopoliedrang omschreven daden. De zo gevreesde TCO (Total Cost of Ownership) van onze IT infrastructures is namelijk niet zozeer een functie van het aantal verschillende systemen dan wel van het aantal verschillende interfaces die zich in die infrastructuur bevinden. Zolang er per aangekocht systeem een of meerdere specifieke interfaces meekomen blijft het drukken van de TCO natuurlijk een dweilen met de kraan open². Dus Microsoft heeft het principe helemaal door, en door het ook nog eens met luider stem te verkondigen geven ze de rest van de wereld het nakijken. De agenda andermaal bepaald door de gevreesde marktspeeler. Na jarenlange strijd op de velden van eer die 'features' en 'productiviteit' mochten heten, en nog zijdelings voorbijgaand aan 'betrouwbaarheid en stabiliteit' als hogere waarden waarvoor wat bloed (marketinginkt) mag vloeien gaan de marketingmachines elkaar nu openlijk te lijf met het aantonen van bij de tegenpartij aanwijsbare gebrekkige interoperabiliteit met de nieuwe referentie: webservices.

2 De stelling volgt uit het tegen elkaar afzetten van de bevindingen van Robert Metcalfe en die van Frederik P. Brooks.

In de vroege jaren 70 al verblijdde Robert Metcalfe ons niet alleen met de nodige wachtrij theorie om de performantie van packet-switched netwerken op te krikken (en zo Ethernet effectief te ontwikkelen), hij bedacht ook de meer economisch gerichte 'Metcalfe's law' die luidt: "De (economische) potentiële waarde van een netwerk (van gebruikers, computers, telefoons of zelfs wagens) verhoudt zich als het kwadraat van het aantal intergeconnecteerde elementen in het netwerk". Voor het klassieke voorbeeld van het GSM netwerk merk je deze verhouding meteen: de tiende GSM in het netwerk kan naar negen andere bellen, en verhoogt ook de waarde van die negen bestaande doordat zij met een extra partij kunnen gaan bellen en sms'en. De honderdste GSM in het netwerk heeft een analoog effect op 99 GSM's. Kortom bij elk toegevoegd netwerkelement krijg je een meerwaarde van het totale netwerk die in verhouding staat tot de waarde die het tot dan had. Het effect is door de netwerk operatoren goed gekend, en door hun aandeelhouders bijzonder gesmaakt. Anderzijds schreef Frederik P. Brooks na jaren ervaring als manager van de ontwikkeling van het OS360 besturingsysteem voor IBM in zijn essay 'the mythical man-month' dat in een netwerk van ontwikkelaars de complexiteit en dus de kost zich evenzeer verhoudt als een exponent van het aantal deelnemers. Developers hebben nu eenmaal hun eigen unieke interface. Toegepast op het voorbeeld van de GSM kun je meteen begrijpen dat de winst van de netwerkoperator meteen door de vingers glipt mocht er niet zoiets zijn als een standaard GSM. Dan zou de introductie van de honderdste bijkomende GSM vereisen dat 99 conversiesystemen worden gedesigned, ontwikkeld, getest en onderhouden. De minstens kwadratische kosten nekken dan je gehoopte winst. De stelling wordt dus al snel dat het aantal interfaces bepalend is voor de kosten, terwijl het aantal spelers blijft instaan voor de waarde.

'INTEROPERABILITEIT' Zo heet het nieuwe slagveld, en 'interoperabiliteit' is daarin helaas een excuus geworden. Want het gekibbel gaat ondertussen natuurlijk over 'van wie' webservices zijn. In die discussie is het alvast leuk te merken dat het W3C alsnog niet helemaal is bezweken onder de beukende hamerslagen van de industriespelers zelf. Het aan hun zijde krijgen van de standaardisering is slechts een detail in de strijd, en als het moet wordt en passant wel een eigen standaardiseringorganisatie gecreëerd. Wat niet zo vlot loopt in de W3C arena loopt al een stuk meer vanzelf in het selecte WS-I clubje van waaruit de echte strijd, namelijk die voor de gebruikerscommunities, in alle hevigheid kan gaan losbarsten. We zitten nu eenmaal in een markt die graag toegeeft aan zijn natuurlijke drang naar monopolies. Wars van alle standaardiseringorganisaties zijn het helaas de 'de facto' standaarden die echt meetellen.

ONTBREKENDE SESSIELAAG XML en HTTP zijn er zo twee en daar komen we als we de zogenaamde REST³-fractie binnen het W3C mogen geloven al een heel eind mee. REST staat voor REpresentational State Transfer, en is zowat de achteraf-het-succes-verklarende theorie die de design principes en grondslagen van HTTP weet te vatten. De essentie daarin is eenvoud. Die eenvoud ligt vooral besloten in het ontbreken van de sessielaag waardoor de benodigde sessie informatie voor de correcte verwerking van de uitkomst continu heen en weer gestuurd moet worden. Het hiertoe uitgedachte URL formaat is wat hen betreft afdoende om zowel de gevraagde service aan te duiden als de nodige parameters mee door te sturen. Het 'dure' webservice woord wordt zo herleidt tot zomaar een webapplicatie die XML retourneert in plaats van HTML. Voor de REST mensen binnen het W3C is het opleggen van de SOAP envelope dan ook een soort toevoegen van een nodeloze complexiteit die daarenboven, via de aangeboden vrijheid tot specifiek gebruik van de SOAP Headers, veel te veel kansen tot nieuwe vormen van data-hijacking levert aan de software-leveranciers. In hun wereldbeeld dienen die laatste in deze trouwens een beetje tegen hun eigen voortvarendheid te worden beschermd. De door henzelf groots aangekondigde droom van interoperabiliteit is vermoedelijk iets wat in essentie niet aan een te kleine en te economische gedreven groep spelers overgelaten kan worden. Alleen jammer dat de nuchterheid van de REST principes zo moeilijk te verkopen zijn als 'nieuw', 'baanbrekend' en mijn favoriet: 'paradigm shift'.

3 Het REST model is beschreven door Roy T. Fielding (zie <http://www.ebuilt.com/fielding>). De man is 'chief scientist' bij eBuilt, Inc. Dr. Fielding is de co-auteur van verschillende Internet standaarden waaronder HTTP and Uniform Resource Identifiers (URI), en de mede oprichter van een aantal open-source software projecten waaronder het Apache HTTP Server Project.

FAUT LE FAIRE Onterecht natuurlijk, want tegen de verwachtingen van de opbouw van dit artikel tot nu krijgt Microsoft van mij de prijs voor baanbrekend werk op het gebied van het REST model, en wel voor de ontwikkeling van ASP.Net. Heen en weer in de URL zit een afzichtelijke parameter die niet minder is dan de serialisering van de server side componenten die nodig zijn voor de verwerking van de request. Als belangrijkste effect realiseren ze hiermee het waterdichte antwoord op de back-button en proxy-caching problematiek waar elke hardcore webdeveloper al eens een nachtje slaap voor heeft gelaten. Toegegeven, ze bewijzen hier niet voor de eerste keer een behoorlijk hack niveau te halen, en je kunt het moeilijk standaard noemen, maar het combineert voortreffelijk een echt objectgeïntereerde ontwikkelomgeving met de meer remote procedure stijl die HTTP met zich meebrengt. Faut le faire.⁴ En ze zeggen dat sowieso alleen maar de Javanen tot hier verder wilden lezen.

FILTERS In het Java kamp is een zelfde soort ondersteuning in de maak. Hoewel al vorig jaar op JavaOne aangekondigd wordt het toch nog even afwachten tot eind dit jaar voor de Java Server Faces specificatie het licht ziet. Tot dan heeft de andere partij nog de tijd om hun gat te dichten: Filters. Dat is voor het Java-kamp een sprong vooruit in een iets andere richting, maar in het licht van webservices minstens zo essentieel. Java Servlet Filters laten een zeer modulaire definiëring van een ketting aan pre en post processors op een webapplicatie toe. Op die manier kun je de verschillende aspecten in de afhandeling van de request, maar ook in het prepareren van de response toegedicht worden aan afzonderlijke componenten in het systeem. Die componenten hebben uiteraard het voordeel dat ze ook in andere pijplijnen kunnen herbruikt worden. Vooral de verwerking van XML laat zich op deze manier graag benaderen. In dit verband brengt de bijkomende recente finalisering van de JSTL (Java Standard Tag Library) in het Java kamp slechts het soort XML/XSL behandeling onder ieders bereik die ook al in ASP.Net aanwezig is.

NUANCEVERSCHILLEN Goed, voor velen is web services natuurlijk allang synoniem voor SOAP geworden en getuigt dit soort 'webservice' -achtige XML web applicaties van het soort zelfredzaamheid dat sinds hun padvindersjeugd allang geen doel meer mag zijn. Voor hen gaat het over SOAP, WSDL en UDDI ondersteuning waarvan eigenlijk maar een beperkt aantal nuanceverschillen op te merken zijn tussen de twee kampen. In het Java kamp heeft elk stukje van het verhaal een aparte specificatie met eigen onderscheidende naam gekregen (JAXP, JAXB, JAXRPC, JAXM, JAXR,...) zodat iedereen die dat wil heel correct aangeduid kan afficiëren dat

ze ook effectief deze of gene versie van bepaalde specificatie implementeert. Persoonlijk vind ik dat een subtiele vorm van hijacking natuurlijk, maar uiteindelijk zal het de .Net developer in het andere kamp wel worst wezen dat zijn toegang tot deze technologie eerder geijkte muisbewegingen (menukeuzes, point en drag) dan specificatiedocumenten zijn, die formeel erop na geslagen moeten worden. Al even subtiel heb ik dan

Het gebruiksrecht van elk gemaakt Word document leveren we af aan het Microsoft Word programma

weer een zwak voor de aanpak van iets als JAXM waar men van bij design voorziet dat verschillende Profiles kunnen ingeplugd worden die instaan voor 'verschillende manieren van SOAP-gebruik'. Aan Microsoft zijde moet je dan wel weer nageven dat ze met hun GXA roadmap al veel verder zijn in het definiëren van de verschillende manieren om die SOAP Headers effectief te gaan gebruiken. Het wordt afwachten of Visual Studio .NET dan ook wordt uitgebreid om ooit even makkelijk de niet GXA technieken in hun ontwikkelomgeving te benutten. Iets te vroeg om te zeggen, maar wat mij betreft alvast iets om na te kijken alvorens we onze score voor 'interoperabiliteit' weggeven.

WSDL Andere aanduiding in die richting vind ik dan weer de benadering van het WSDL gebeuren. De Visual Studio .Net omgeving gaat er spontaan van uit dat de WSDL best kan gegenereerd worden nadat je de echte business service hebt gebouwd. In realiteit wil je evenwel in staat zijn om de WSDL die je nu werd opgelegd door leverancier X door te geven aan een tweede, concurrerende leverancier. Zo is die in staat een contractueel gelijkwaardige dienst te leveren (voor minder geld hoop je dan maar) zodat jij niet de kosten hoeft te maken om je client-applicatie naar een nieuwe interface aan te passen. Het veronderstelt natuurlijk wel dat die tweede leverancier in staat is, uitgaande van de WSDL, vlot een nieuwe business service te ontwikkelen. De

4 Een drietal jaar geleden had SilverStream met hun 'page' idioom een eerste, boeiende Java-implementatie in deze richting. Iemand heeft hen sindsdien verteld dat ze een full blown J2EE server moesten implementeren. Gelukkig zijn ze recentelijk (vermoedelijk door een gelijkaardig soort economische randvoorwaarden) weer wat meer in hun pioniersrol gedwongen. Op gebied van webservice implementatie concepten zeker een speler in het Java kamp om weer geboeid te volgen. Ik kijk alvast uit naar wat ze zijdelings met xforms beloven te gaan doen.

omgekeerde beweging dus, die in het Java kamp, met zijn immer academisch georiënteerde inslag, het uitgangspunt was (JAXRPC compiler). Leuk om te zien is dat academici en pragmatici elkaar wat meer beginnen te vinden: De meeste Java Web Service frameworks zullen nu ook de WSDL en skeleton classes genereren als je

Leuk om te zien is dat academici en pragmatici elkaar wat meer beginnen te vinden

hen met je business interface (class) confronteert. En Microsoft van zijn kant ondersteunt ook het starten vanaf een WSDL file. Dat zulks niet ondersteund wordt via de menu's en wizards van de wij-maken-alles-makelijker Studio .Net, maar slechts door middel van het stilzwijgend meegeleverde wsdl.exe toont mij dan toch dat de beschreven business case van de webservice gebruiker die zich van zijn huidige leverancier wil onthechten alvast niet top-of-mind was.

COLA EN PEPSI Een aantal analisten heeft de strijd onlangs vergeleken met de legendarische cola oorlogen. Als de vergelijking opgaat, dan vind ik het alvast een geruststellende gedachte dat ik op een verloren moment even goed een Cola Light als een Pepsi Max in mijn koelkast kan vinden, en er even sterk van kan genieten. Zolang het maar verfrissend blijft voor de gebruiker.

Dat die gebruiker in deze hele beweging zelf een belangrijk aandeel heeft blijkt die alsnog niet te beseffen. Van Microsoft wordt altijd beweerd dat ze zo succesvol zijn omdat ze perfect in staat zijn te luisteren naar de klant (in hoofde van 'de markt') en die tegen een correcte prijs te bedienen op zijn wenken. Als dat allemaal klopt, dan hebben we er duidelijk zelf om gevraagd. De 'oorlog' gaat in deze weer niet tussen J2EE en .Net. Het gaat tussen de vechten-de partijen aan de ene kant, en de gebruiker aan de andere kant. In plaats van zich mee te laten polariseren in de ene of de andere richting zou die er beter aan doen in standaardisering of open source bewegingen naar een uniforme component en service infrastructuur te streven. De onzekerheid en angst van een dergelijk scenario ten opzichte van de zekerheid van de jaarlijks verzekeringspolis (license fee) komt uiteindelijk voort uit een misplaatst gebrek aan zelfvertrouwen. Het is uw probleem, pak het aan. Indien niet, laat u dan niet verleiden tot het aanschaffen van de oplossing.

Marc Portier (e-mail: mpo@outerthought.org)

Marc Portier is mede-oprichter van Outerthought, een klein en spits-technologisch Java & XML competentiecentrum. Als Java-advocaat van het eerste uur volgt hij van nabij de nieuwe API's en richtingen (zoals J2EE) die de taal en het platform sinds de beginnagen in '96 heeft genomen. Daarbij gaat zijn interesse vooral uit naar de Internet en XML gerelateerde onderwerpen. Outerthought helpt bij de beslissingen rond en de introductie van deze Java-XML technologieën als werkmiddelen voor applicatieontwikkeling bij zijn klanten.

PATCHES Patches PATCHES Patches PATCHES Patches PATCHES

Rational breidt uit met Java testcapaciteiten en nieuw ontwerpproduct

Rational Software introduceerde de Rational Suite v2002 Release 2, de achtste gesynchroniseerde release van Rational Suite in minder dan drie en een half jaar. Deze release breidt Rational's 'Extended Development Experience' uit met de introductie van Rational XDE Modeler, een nieuw lid van de Rational XDE familie en ScriptAssure, een toonaangevende mogelijkheid in de Rational Suite TestStudio. Deze produceert testscripts voor zakelijke Java- en webapplicaties.

Rational's 'Extended Development

Experience' verzorgt een ontwerp- en ontwikkelomgeving door Rational's producten te integreren in IBM WebSphere Studio Application Developer en Microsoft Visual Studio .Net. Hierdoor kunnen ontwikkelaars direct binnen hun geïntegreerde ontwikkelomgevingen (IDE's) werken. Rational integreert met deze release Rational XDE Modeler in deze IDE's. Tegelijkertijd heeft Rational Rational Suite TestStudio geïntegreerd in de op Eclipse gebaseerde WebSphere Studio IDE en Rational PurifyPlus in de Microsoft Visual Studio .Net IDE. Voor ontwikkelaars van embedded softwareproducten bevat Rational Suite v2002

Release 2 een nieuwe versie van Rational Rose RealTime.

Rational Suite v2002 Release 2 levert tools en 'best practices' voor architecten, ontwerpers en ontwikkelaars: gegevensmodellering, web publishing, rapportage- en 'assisted modeling' capaciteiten. Rational's XDE Modeler is een nieuw product waarmee men visuele modellen kan bouwen die UML combineert met 'free form' tekenen. Rational XDE Modeler draait binnen IBM WebSphere Studio Application Developer, IBM WebSphere Workbench (gebouwd op Eclipse technologie) en Microsoft Visual Studio.Net, waardoor architecten, ontwerpers en ontwikkelaars

dezelfde gebruikservaring kunnen delen. Ook testers hebben met ScriptAssure een krachtig ondersteunend tool in handen. Release 2 voegt bovendien Java toe als scripting taal. Met dit tool kunnen daarom nu geautomatiseerd testscripts geproduceerd worden voor zakelijke Java- en webapplicaties. ScriptAssure vereist ook minder programmeeronderhoud, zodat het testteam zich kan richten op belangrijkere activiteiten. De nieuwe versie van Rational Suite TestStudio bevat ook IBM WebSphere Studio Workbench, een Java IDE gebouwd op Eclipse technologie die integreert met IBM's WebSphere Studio producten.