

Markant begin van een revolutie of veel geschreeuw, weinig wol?

Van webservices naar serviceweb

Cor Baars

Voorzien ze in een sluimerende behoefte? Staan we werkelijk aan het begin van een nieuw tijdperk? Is het de marketingmachine van Microsoft? Hoe het ook zij, webservices staan in het middelpunt van de belangstelling, getuige de vele aandacht op Internet en in de pers. Wel wat veel commotie voor een gestandaardiseerde manier om via Internet 'remote procedure calls' (RPC's) te doen, nietwaar...?

Wat zijn webservices? Welke standaards spelen een rol? En wat zijn nu en in de nabije toekomst de toepassingsmogelijkheden?

NAAR HET 'SERVICEWEB'

Een webservice is een dienst die op afstand over Internet kan worden aangeroepen. Dat is van belang voor ontwikkelaars die overwegen een gedeelte van de functionaliteit van hun programma's uit te besteden aan anderen. Anderen op Internet, maar -in analogie met het Web- ook op het extranet (integratie van business-to-business) en op het intranet (enterprise application integration, EAI). Hiermee ontstaat op kleinere of grotere schaal een "serviceweb". Leveranciers kunnen daarmee hun diensten aanbieden, hetzij intern of extern.

Stel, we zijn ontwikkelaars van een marketingapplicatie en komen op het idee onze toepassing uit te rusten met een voorziening waarmee de klant doorlopend

kan nagaan in hoeverre men op het Web aan de bedrijfsnaam refereert. Wordt dat minder, of meer? Komen er veel nieuwe referenten bij? Wie refereren het meest?

Onderdeel van deze functionaliteit is die van een zoekmachine, maar die maken we uiteraard niet zelf. Sinds een paar weken kunnen we de complete functiona-

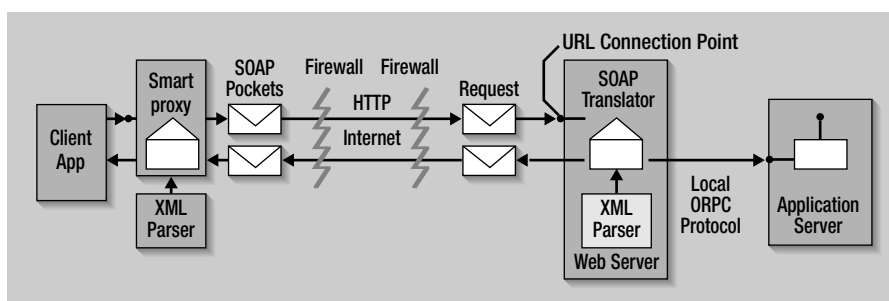
Standaards en technologieën zijn nog volop in ontwikkeling. Er blijft nog veel te wensen

liteit van Google in de vorm van een verzameling webservices via Internet op afstand aanroepen. Dankzij webservices kan dat aanroepen erg eenvoudig. Bij gebruik van een ontwikkelomgeving die webservices ondersteunt, tikken we de url in van Google. En daarmee verwijzen we naar een op deze site aangeboden gestandaardiseerde interfacespecificatie. De ontwikkelomgeving genereert direct een proxy waarmee we de functies die Google lokaal kunnen aanroepen.

Figuur 1 laat zien hoe dat even later werkt als we daadwerkelijk op deze wijze van Google gebruik maken. Onze applicatie doet een aanroep naar een procedure op afstand. Een XML-parser vertaalt de aanroep naar een XML-bericht, dat vervolgens in een SOAP-*envelope* als http-request naar de server wordt gestuurd. De server pakt het uit, vertaalt het in een procedure-aanroep in termen van de applicatieserver die de dienst levert. Deze produceert het antwoord en het geheel herhaalt zich in omgekeerde richting.

DE WEBSERVICE-STACK

Stapelning van standaards en protocollen maakt het voorgaande mogelijk. Wie de buitenwereld op de hoogte wil brengen van de technische interfacespecificatie van de diensten die hij levert, publiceert een zogenaamd Web Service Description Language-bestand (WSDL-file). Dit bestand is tamelijk onleesbaar en dan ook niet bedoeld voor menselijke consumptie. De ontwikkelomgeving van de dienstleverancier genereert WSDL en die van de



FIGUUR 1: HET AANROEPEN VAN EEN WEBSERVICE.

UDDI	Publicatie van webservices
WSDL	Documentatie webservices
SOAP	Berichten-envelop
XML	Communicatietaal
Http	Transportprotocol
TCP/IP	Communicatielaag

TABEL 1: DE WEBSERVICE-STACK.

dienstenconsument -zoals zojuist beschreven in het voorbeeld- interpreteert dat later om daaruit een client-proxy te genereren.

Het op XML gebaseerde berichtenprotocol SOAP definieert de envelop voor XML-berichten die aan webservices ten grondslag liggen. In de context van webservices hebben deze berichten dus betrekking op zowel vraag als antwoord van een procedure-aanroep. Consument en leverancier van een dienst kunnen de SOAP-berichten interpreteren, en dat geldt

Webservices en onderdelen van ebXML zullen beide een rol spelen in toekomstige B2B-toepassingen

ook voor eventuele knooppunten (nodes) die een rol spelen in de routing. In figuur 2 zijn de structuur en (een deel van) de inhoud van een SOAP-bericht geschetst.

De SOAP-envelop is de buitenste schil van het bericht en bevat algemene informatie, zoals een verwijzing naar de manier waarop de inhoud is gecodeerd: het *encoding style*-attribuut. Daarna volgen één of meer *headers* en *bodies*. De laatste bevatten de inhoud van het bericht; hier is dat een aanvraag voor gebruik van een creditcard voor een betaling. De headers bevatten meta-informatie over de inhoud van de body. Zo kan men in een header aangeven voor welk tussenknooppunt in de routing bepaalde body-inhoud is bedoeld (het *actor*-attribuut) of dat men eist dat de knoop die uiteindelijk het bericht verwerkt dit per se succesvol moet doen (het *must understand*-attribuut). Hiermee wordt een foutbericht afgedwongen in het geval dat toch niet lukt. Ook kan een identificatie in

de SOAP-header het bericht relateren aan een eerder of later bericht.

Inzet van SOAP als protocol voor een procedure-aanroep is slechts een voorbeeld van de vele mogelijke conversatiestijlen. Andere zijn notificaties, maar ook asynchrone verzoeken behoren tot de opties.

SOAP is onafhankelijk van het communicatieprotocol dat wordt toegepast om de berichten te versturen. Dat kan evengoed ftp als SMTP als een protocol voor message queueing zijn. Bij webservices hoort echter http, het synchrone protocol waarmee we normaliter webpagina's opvragen. SOAP-berichten gaan daarom via poort 80 soepel door de firewall. Dat gold niet voor allerlei eerdere berichtenprotocollen, die vaak hun eigen poortnummer hadden. De beveiligingsmogelijkheden, maar vooral selectieve openstelling die we met firewall bereiken, staan daarmee ook voor open voor webservices.

Rest de vraag hoe we de voor ons geschikte webservices vinden. Of vanuit het perspectief van de leverancier: hoe publiceert ik mijn diensten met hun eigenschappen? Daartoe dient Universal Description Discovery and Integration (UDDI). Dit is een standaard voor een machineleesbare catalogus voor webservices. In een UDDI-catalogus kunnen de functionele en technische kenmerken van een webservice worden vastgelegd, maar ook details over de leverancier die de diensten levert.

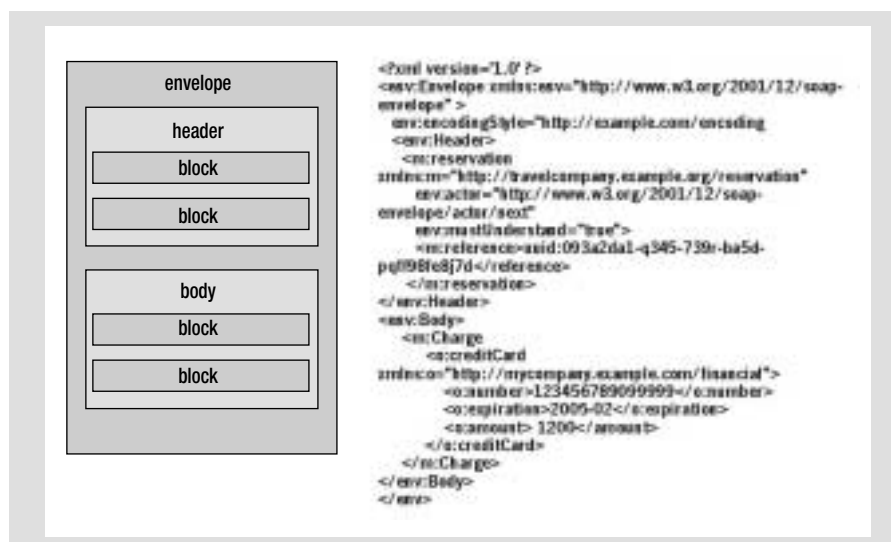
Achterliggend idee is dat een consument van diensten een programma kan schrijven dat dynamisch op zoek gaat in een UDDI-catalogus naar mogelijke leveranciers van een bepaalde dienst, op basis van bepaalde criteria. In ons voorbeeld zou dit programma zelf op zoek kunnen gaan naar een alternatieve zoekfaciliteit als mocht blijken dat Google uit de lucht is¹.

Inmiddels is wel duidelijk dat het idee van geautomatiseerd zoeken naar diensten althans op wereldwijde schaal niet erg bruikbaar is. Vanzelfsprekend is de samenwerking tussen partijen veel meer een kwestie van overleg en vertrouwen dan de technische mogelijkheid van opsporing van de diensten.

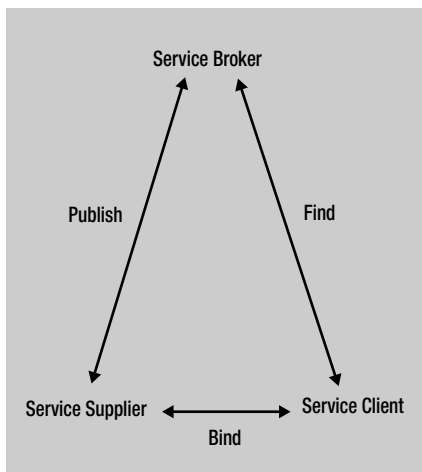
ONTWIKKELINGEN

De standaards en technologieën rond webservices zijn nog volop in ontwikkeling. Er blijft nog veel te wensen, zeker wanneer we kijken naar de verschillende toepassingsmogelijkheden.

Allereerst kan het zijn dat een webservice bediening met zich meebrengt. Dat is niet het geval bij het eerder genoemde Google-voorbeeld. Als we daarentegen denken aan het met webservices bepalen van een route tussen twee adressen, moet het ook mogelijk zijn de gevonden route te tonen. Daarom wordt gewerkt aan Web Services for Interactive Applications



FIGUUR 2: STRUCTUUR EN VOORBEELD VAN EEN SOAP-BERICHT.



FIGUUR 3: LEVERANCIER, KLANT EN MAKELAAR.

(WSIA), een standaard die specificatie van het interactieve aspect van een webservice mogelijk maakt.

Er zijn alternatieven in de maak voor UDDI. IBM ontwikkelt op dit moment de Web Service Inspection Language (WSIL) als taal om nauwkeuriger en gestandaardiseerder dan in UDDI, de functionaliteit en kenmerken van de te leveren diensten te beschrijven. Een WSIL-document, ter beschikking gesteld op de site van een bedrijf, openbaart de diensten van het bedrijf. Net als bij het Web zullen wellicht zoekmachines ontstaan die WSIL-documenten van bedrijven lezen en indexereren. Zo wordt de servicecatalogus door spidering gevuld en niet door rechtstreekse publicatie in een catalogus. Gerelateerd aan WSIL is de Web Service Endpoint Language in de maak, waarmee kwaliteitsaspecten als beschikbaarheid en continuïteit van webservices worden uitgedrukt.

Ook bij webservices is beveiliging essentieel. Weliswaar is SSL bruikbaar voor authenticatie en encryptie, omdat SOAP-messages als http-requests worden verstuurd. Maar het grote nadeel van SSL-inzet is dat elk knooppunt dat een header of een body wil lezen het gehele bericht moet ontcijferen. Daarom wordt gewerkt aan standaards waarbij alleen de body -of zelfs maar een deel van de body, het deel met de gevoelige informatie- encryptie ondergaat, terwijl de rest openstaat voor interpretatie door tussen- en eindknoten.

Bij toepassing van webservices voor

business-to-business-integratie gaat het niet langer om geïsoleerde procedure-aanroepen, zoals in ons Google-voorbeeld. Er is dan sprake van een langdurige complexe transactie, die doorgaans uit meerdere stappen bestaat. Zo komen er twee vereisten bij. We moeten zeker weten dat de onderdelen van deze complexe transactie adequaat verlopen. Hiervoor wordt gewerkt aan nieuwe mechanismen en standaards voor transactiecoördinatie, die ook op termijn onderdeel van de webservicestandaards moeten gaan uitmaken.

Verder moeten we beschikken over de mogelijkheid de procesgang ('flow') van de complexe transactie te definiëren. Microsoft gebruikt hiervoor Xlang, IBM ontwikkelt daartoe de Web Service Flow Language (WSFL). Er zijn overigens meer standaarden in de maak voor B2B-integratie via het Web, zoals ebXML². De verwachting is dat webservices en onderdelen

Bedrijven gebruiken op dit moment vooral combinaties van bestaande middleware en webservice-technologie

van ebXML beide een rol zullen spelen in toekomstige B2B-toepassingen.

Wat kunnen webservices ten slotte betekenen voor applicatie-integratie binnen een organisatie (EAI)? Het is een uitdagende gedachte een interne, al of niet op UDDI gebaseerde, catalogus te publiceren van herbruikbare diensten die door de organisatie zelf worden gerealiseerd. Is in dat geval de performance van webservices echter wel voldoende? Berichten in XML moeten worden geparst, ingepakt en uit-

gepakt. Daar komt bij dat XML als tekstueel protocol niet zo compact is als binaire protocollen.

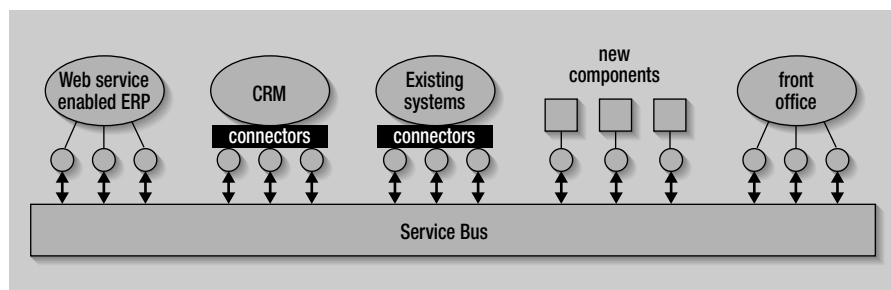
Zoals reeds opgemerkt, SOAP, het onderliggende berichtenprotocol, staat los van het toegepaste transportprotocol. Nu associëren we webservices met synchrone

Het concept past naadloos in de 'service-architectuur' die vele bedrijven voorstaan

RPC's met gebruikmaking van http. Nadeel is echter dat dit protocol ons geen garantie geeft dat een bericht ook daadwerkelijk aankomt. Dat is zeker voor EAI- en B2B-toepassingen onacceptabel. Daarom willen we webservices niet alleen baseren op http, maar ook op asynchrone messaging en zelfs reliable http (HTTPR), dat nog in ontwikkeling is. En ten slotte speelt ook hier de noodzaak van transactie-integriteit en al of niet gedistribueerde transactiecoördinatie een rol. Zaken waarin bestaande middleware wel maar webservices nog niet voorzien. Het is dan ook niet verwonderlijk dat bedrijven op dit moment voor hun applicatie-integratie vooral combinaties van bestaande middleware en webservice-technologie gebruiken.

WEBSERVICES EN CBD

Als u bij uzelf een déjà vu wilt uitlokken, kijk dan naar de potentiële voordelen van het gebruik van webservices binnen de bedrijfscontext, zoals hergebruik, onderhoudbaarheid en vervangbaarheid en de veel genoemde voordelen van component based development (CBD). Deze zijn grotendeels gelijk aan elkaar, maar worden op



FIGUUR 4: WEBSERVICES EN DE SERVICEBUS.

De markt

De markt voor webservices valt in een aantal delen uiteen. Allereerst zijn er de leveranciers van ontwikkeltools. In navolging van Microsoft, die webservices ziet als kern van gedistribueerde applicatiearchitecturen in haar .NET-framework, vallen zijn over elkaar heen in de haast mee te doen. IBM steekt veel energie in het ondersteunen van webservices in haar Java-ontwikkeltools, evenals bedrijven als Bea, Borland en Oracle. Het Java-kamp is echter op het moment wat verdeeld, en dat komt onder meer doordat het van nature met meer partners en daardoor met meer overleg te maken heeft dan Microsoft.

Een tweede deel van de markt bestaat uit bestaande en nieuwe leveranciers van middleware. Bedrijven als Tibco, Webmethods en Vitria lijken in webservices een nieuwe adapterstandaard voor applicatie-integratie te zien en zij doen dan ook enthousiast mee. Verder zijn er producenten van beheeromgevingen voor webservices. Er valt namelijk nog wel het een en ander te regelen. Neem Google. Wanneer je minder dan duizend procedure calls per dag doet is het gratis, daarna moet je wellicht betalen. Dat moet dan wel worden bijgehouden. Zaken als autorisatie, billing, routing van verzoeken naar de juiste servers - er zit nogal wat tussen de vraag en het antwoord. Hiervoor zijn webservice management-omgevingen te koop, zoals Cape Clear en Blue Titan. Maar er is ook een trend deze beheerdiensten extern aan te bieden. En u raadt het al, natuurlijk als webservice. Daarbij regelen we bijvoorbeeld niet zelf het afrekenen van dienstengebruik met de klant, maar een externe partij houdt dit gebruik bij en zorgt voor de facturering. En ten slotte moeten natuurlijk de aanbieders van de diensten zelf worden genoemd. Applicaties voor ERP en CRM, waaronder SAP en Siebel, stellen hun functionaliteit steeds meer ter beschikking in de vorm van webservices. Bestaande bedrijven, zoals Google, zullen hun functionaliteit in deze alternatieve vorm als halffabriek ter beschikking stellen, maar er zullen ook nieuwe bedrijven ontstaan met slimme diensten, zoals het net genoemde factureringsmodel.

eerde applicatie-architecturen. Dat doen zij door verbreding en verdieping van de standaards en technologie. Het gaat niet zonder slag of stoot. Zo verscheen onlangs het bericht in de pers dat binnen W3C verschil van mening bestaat over welke kant het op moet met verdere versies van SOAP (nu versie 1.2).

Hoe dan ook, de betekenis van het begrip webservices zal veranderen, de term zelf zal nog lang onderdeel van het IT-jargon uitmaken. ●

Noten:

1. Er is momenteel een centrale webservice-catalogus in bedrijf waar wereldwijd bedrijven hun diensten kunnen publiceren en dus ook kunnen zoeken naar diensten: www.uddi.org
2. Zie het thema-artikel over ebXML door Pim van der Eijk elders in dit nummer. Standaards als ebXML omvatten ook de semantische aspecten van B2B-integratie; het standaardiseren van de betekenis van de berichten tussen de betrokken partijen.

Drs. ing. C. Baars (cbaars@cibit.nl) is programme manager van de Master of Science-opleidingen bij CIBIT Adviseur en Opleiders.

een verschillende manier gerealiseerd.

Wie een bepaalde dienst nodig heeft, kan kiezen: of hij zoekt naar een reeds bestaande component in de bedrijfscomponentenbibliotheek en die onderdeel maken van de applicatie of hij roept vanuit de applicatie rechtstreeks de van binnen of buiten het bedrijf geleverde loggingdienst aan. Hieruit blijkt dat webservices en CBD dicht tegen elkaar liggen. Webservices zijn nu al een belangrijk medium voor de publicatie van de diensten van de componenten in een systeem.

SERVICE-ARCHITECTUUR

Misschien is een extra reden voor de grote interesse in webservices dat het concept naadloos past in de *service-architectuur* zoals die vele bedrijven die zien als basis voor hun enterprise architecture. In een service-architectuur dient een *servicebus*

als ontkoppelpunt tussen aanbieders en afnemers van diensten. Het is duidelijk dat webservices een grote rol kunnen spelen als implementatievorm van de servicebus, zeker met standaards als UDDI en WSDL, die de publicatie van de diensten ondersteunen.

In figuur 4 uit een publicatie van IBM wordt dit idee concreet gemaakt. Alle daar geschetste systemen worden door middel van webservices aan de servicebus gekoppeld en zo transparant aan elkaar gemaakt. Voordat dit werkelijkheid wordt, moet nog wel in nogal wat vereisten worden voorzien, zoals we hebben gezien.

NAAR VOLWASSENHEID

Er is nog veel werk te verzetten. Dat gebeurt dan ook. Veel partijen dragen hun steentje bij om webservices volwassen te maken als open standaard voor gedistribue-

Informatie op Internet

Webservice-portals

- www.webservices.org
- www.webservicesarchitect.com
- www.cbdiforum.com

Standaardisatie

- www.w3c.org
- www.oasis-open.org
- www.xml.org

Webservice-support

- www.microsoft.com/webservices
- www-106.ibm.com/developerworks/webservices/
- www.capeclear.com
- www.bluetitan.com
- www.openlink.co.uk
- www.sap.com