

# Webservices: hype of silver bullet?

In de jaren negentig maakte component-based-development zijn intrede en is sindsdien een belangrijke manier geworden om herbruikbare softwaremodules te bouwen. Zo is binnen het Microsoft-kamp het Component Object Model (COM) een belangrijke techniek gebleken om component-based-development vanuit de techniek te ondersteunen. Ook op andere platforms zijn dergelijke componentmodellen ontstaan zoals: Common Object Request Broker Architecture (CORBA) en Enterprise Java Beans (EJB).

Alhoewel de implementatie en de technische details tussen de diverse componentmodellen verschillen, zijn er toch overeenkomsten aan te geven. Zo is de notie van een *interface*, een contract tussen de client en het object, in alle modellen prominent aanwezig. Ook het remoten van objecten laat overeenkomsten zien in de aard van de gebruikte protocollen. Zo maken de verschillende componentmodellen gebruik van complexe binaire protocollen waarvan de ondersteuning diep verankerd zit in het besturingssysteem. Dit is dan ook de reden dat de ondersteuning van DCOM (COM via remoting) op andere platforms nooit echt van de grond is gekomen; vele van de benodigde runtime-services voor DCOM zitten hard ingebouwd in het besturingssysteem en zijn daarmee een proprietary Microsoft-oplossing geworden.

Kijken we naar de evolutie van het internet dan zien we dat algemeen geaccepteerde protocollen en standaards zoals HTTP, XML en XMLSchema-technieken geleid hebben tot een andere benadering van het web. In de begintijd van het web werd het voornamelijk gebruikt om HTML-documenten uit te wisselen met als doel om binnen een webbrowser een userinterface aan te bieden. Met de komst van XML en

XML-gerelateerde technieken, kunnen we zeer nauwgezet de structuur van het te versturen document vastleggen en bovendien kan nu ook de typering van functie-argumenten worden beschreven.

Zo is het nu dus mogelijk om in een XML-document een beschrijving te geven van functionaliteit, naast de gebruikelijke data, waardoor het plotseling mogelijk wordt om een functie via het web aan te roepen. Deze technologie is beter bekend onder de naam *webservices*, waarbij het Simple Object Access Protocol (SOAP) het feitelijke wire-level protocol is. Je zou kunnen zeggen dat COM zich tot DCOM verhoudt als webservices tot SOAP.

Het grote voordeel van webservices ten opzichte van andere technieken is de brede acceptatie in de markt en de goede ondersteuning vanuit diverse platforms. Zo is het nu mogelijk om met webservices interoperabiliteit te verkrijgen over heterogene platforms, met één simpele techniek die gebruik maakt van algemeen geaccepteerde standaards.

De ondersteuning van webservices vanuit Visual Studio .NET is erg goed. Dit geldt zowel voor het bouwen van een webservice als het consumeren ervan. Om de vraag te beantwoorden of webservices een hype zijn of de nieuwe *silver bullet*,

moeten we meer aspecten in ogenschouw nemen dan louter het vermogen om een functie te kunnen aanroepen.

Vele gedistribueerde systemen maken gebruik van complexe enterprise-services zoals transacties en securitymanagement. Hoe laten deze enterprise-services zich combineren met webservices? Met webservices is er geen propagatie van transactie- en security context mogelijk, hetgeen wel het geval is bij DCOM.

Dit betekent dus dat webservices op dit moment minder functionaliteit bieden dan het oude COM-model. Gelukkig wordt er op dit moment hard gewerkt aan een nieuwe webservices standaard, de Global XML Architecture (GXA), waarin voorzieningen komen voor security, routing en transacties.

Op het moment dat de GXA brede ondersteuning in de markt vindt, zullen webservices bijzonder krachtig worden en een serieus alternatief zijn voor de gevestigde component-technieken. Tot die tijd hangt de toepasbaarheid van webservices af van de gewenste functionaliteit.

*Ing. Xander Buffart is werkzaam als IT-architect bij Info Support te Veenendaal. E-mail: xanderb@infosupport.com*