

Web Services - o arhitectura orientata pe servicii

Lect.dr. Carmen TIMFOTE

Catedra de Informatica Economica, A.S.E. Bucuresti

Pe piata aplicatiilor distribuite, putem vorbi de trei mari tehnologii implementate cu succes la ora actuala: DCOM, CORBA si Java RMI. De la aparitia lor si pâna în prezent aceste tehnologii au evoluat datorita necesitatilor impuse de piata aplicatiilor distribuite. Cu toate acestea au ramas niste bariere tehnologice, comune acestor tehnologii. Web Services si-a facut aparitia pe piata IT într-un moment de rascruce al aplicatiilor distribuite. O implementare a acestei tehnologii a fost facuta de Microsoft pentru platforma .NET Framework.

Cuvinte cheie: XML, WSDL, SOAP, UDDI, .NET.

Arhitectura Web Services

Nu cu mult timp în urma, la conferintele internationale, marile companii de software calificau aceste Web Services ca fiind "Revolutia Internet". Este o tehnologie care de fapt nu a inventat nimic nou, ci a dat o noua valoare a ceea ce aveam deja, a reusit sa schimbe modul de a concepe si realiza aplicatiile software. Pentru a construi o aplicatie distribuita, robusta si flexibila, trebuie îndeplinite anumite cerinte:

- *integrarea resurselor software* trebuie sa fie slab cuplate, distincte si separate de restul aplicatiei;
- *intercomunicarea între componente* trebuie sa se realizeze cu ajutorul unor protocoale standard Internet;
- *interfata catre componentele software* trebuie sa fie publicata pentru a fi accesibila (interfata si documentatia serviciului).

Implementarea cerintelor anterioare este posibila doar într-o *arhitectura orientata pe servicii*, care contine trei componente de baza:

- *Furnizorul de Servicii*: nod în Internet care printr-o interfata asigura accesul la un serviciu software care executa un anumit set de operatii;
- *Consumatorul de Servicii*: nod în retea care se leaga la Furnizorul de Servicii, si foloseste functionalitatea oferita de acesta, construind solutia de afacere ("business solution");
- *Broker-ul*: nod în Internet unde sunt stocate informatii despre furnizorii de servicii. Clientii se conecteaza pentru a obtine infor-

matiile despre interfetele pe care le expun Furnizorii de Servicii.

Comunicatia între cele trei componente se realizeaza cu ajutorul lui HTTP, între ei transferându-se mesaje SOAP (care sunt de fapt mesaje XML formate conform standardului SOAP). Broker-ul foloseste protocolul UDDI (Universal Description, Discovery and Integration) pentru cautarea si actualizarea informatiilor din baza sa de date cu privire la localizarea serviciilor Web înregistrate. Dupa localizarea serviciului Web, acesta este interogata pentru a obtine informatii despre metodele sale, respectiv interfata pe care o expune, informatii organizate sub forma unui document WSDL (Web Services Description Language).

Cum functioneaza XML Web Service?

Clientul ia legatura cu brokerul sau chiar direct cu serviciul web pentru a obtine informatii legate de interfata Web Service. Pentru aceasta foloseste "Discovery document" care poate fi realizat în doua moduri:

- *Static Discovery*: localizarea unui Web Service este deja cunoscuta, adica URL-ul catre acesta.
- *Dynamic Discovery*: singura informatie despre un Web Services este punctul din Internet unde se afla furnizorul de Web Services, URL-ul site-ului. Se genereaza automat o lista cu informatii despre Web Services expuse de catre acel furnizor, servicii accesate prin URL-uri diferite.

Dupa ce a fost localizat Web Service, clientul cere o descriere a acestuia, care trebuie sa

cuprinda metodele expuse, parametrii lor, precum si ce tip au valorile care se returneaza în urma executiei. Toate aceste informatii sunt încapsulate într-un WSDL.

WSDL este o gramatica XML folosita pentru a descrie un Web Service din punct de vedere al mesajelor pe care le primeste si le emite. Acest fisier activeaza ca un contract între Consumer si Provider. Este echivalent cu rolul unui TLB (Type Library) în DCOM.

Cu informatiile continute în WSDL, clientul stie cum sa încapsuleze cererile catre furnizorul de Web Services în pachete SOAP si cum anume sa interpreteze pachetele SOAP venite sub forma de raspuns al cererii efectuate de client.

Urmatorul pas depinde de platforma utilizata pentru implementarea clientului de Web Service. Pe platforma .Net clientul implementeaza conexiunea cu un Web Services sub forma unui proxy.

Rolul unui proxy este acela de a împacheta cererile catre Web Services în pachete SOAP si de a despacheta raspunsurile venite de la Web Services sub forma de pachete SOAP, degrevând astfel aplicatia de logica împachetarii si despachetarii. (asemanator conceptului "proxy-stub" din DCOM).

Modul de functionare a serviciilor Web este prezentat în figura 1.

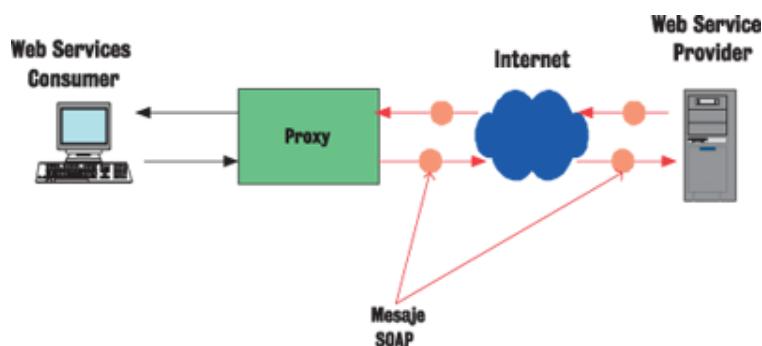


Fig. 1. Functionarea Web Services

Servicii .NET

.NET My Services reprezinta o schimbare radicala în modul în care aplicatiile folosesc si acceseaza informatii utilizator. Cunoscut si sub numele de cod Hailstorm este oferit de Microsoft la sfârșitul anului 2002. Aceste servicii vor oferi controlul datelor si informatiilor personale si ofera un mecanism universal de acces la informatii de catre aplicatii Web, dispozitive si companii.

Diferite dispozitive pot accesa informatiile utilizator prin mesaje SOAP standard prin Internet, datorita stocarii datelor într-o locatie centrala, fiind protejate prin serviciul de autentificare Microsoft's Passport. Astfel, un telefon mobil va avea acces catre aceleasi informatii ca si un PDA, iar daca unul face o modificare asupra datelor, toate celelalte dispozitive vor accesa datele noi.

Serviciile .NET My Services sunt o tehnologie care face posibila ca dezvoltatorii sa se concentreze asupra problemelor de business specifice, în loc de infrastructura. Ei voi ges-

tiona stocarea de informatii utilizator prin transmiterea de notificari catre dispozitive capabile sa comunice cu .NET My Service. Aplicatiilor vor putea accesa aceste informatii indiferent de tipul dispozitivului sau al aplicatiei. Un telefon mobil va putea sa acceseze calendarul utilizatorului la fel de usor ca si un laptop, deoarece ele vor accesa un serviciu Web central oferit de .NET My Services.

Urmatoarea lista de servicii va fi oferita prin .NET My Services :

- .NET Address - informatia de adresa a utilizatorului.
- .NET Profile - informatii utilizator cum ar fi: numele, porecla, date speciale, fotografie etc.
- .NET Contacts - agenda si informatii de contact.
- .NET Location - informatii electronice si geografice de locatii.

- .NET Inbox - accesul catre date cum sunt e-mail, voicemail, incluzând accesul catre sistemele de e-mail actuale.
 - .NET Calendar - acces catre calendarul utilizatorului.
 - .NET Documents - documentele utilizatorului pentru acces ulterior de pe orice alt dispozitiv.
 - .NET Application Settings - diferite setari ale aplicatiilor.
 - .NET FavoriteWebSites - lista adreselor de pagini Web preferate.
 - .NET Wallet - gestioneaza carti de credit, chitante, cupoane si alte întregirari de tranzactii.
 - .NET Devices - setarile pentru diferitele dispozitive pe care le are utilizatorul.
 - .NET Services - informatii despre serviciile pe care le are utilizatorul.
 - .NET Usage - informatii de acces a serviciilor .NET My Services.
- .NET My Services va fi o mare oportunitate de afaceri pentru Microsoft, majoritatea serviciilor fiind contra cost. Utilizatorii vor plati pentru a stoca informatiile personale, iar dezvoltatorii si companiile vor plati pentru a accesa aceste informatii. Nu se poate prevedea daca comunitatea de dezvoltatori si de utilizatori va dori sa plateasca pentru stocarea si accesul centralizat al informatiilor personale. Orice se va întâmpla, .NET My Services reprezinta primul pas în directia de software ca serviciu.

Concluzii

Suntem la începutul unei noi ere a dezvoltarii solutiilor enterprise. Serviciile Web vor schimba modalitatea în care aplicatiile sunt dezvoltate si vor schimba modul de conducere al afacerilor pe Internet. Suportul pentru Web Services este în continua crestere. Marii jucatori de pe piata au inclus în platformele lor atât posibilitatea utilizarii, cât si a crearii acestui tip de servicii.

Care va fi impactul lor în viitor ramâne de vazut. Ramân înca multe semne de întrebare privind fiabilitatea si siguranta acestora. Totodata si standardele evolueaza spre definitivare si, de aceea, acest tip de aplicatii nu poate fi privit ca o platforma matura.

Bibliografie

1. Banerjee, Corera, Greenvoss – C# Web services, 2001 Wrox Press
2. Scott Short , - Building XML Web Services for the Microsoft .NET Platform, 2001 Microsoft Press
3. www.microsoft.com
4. www.netreport.ro