

EME FIATAL MŰSZAKIAK TUDOMÁNYOS ÜLÉSSZAKA

Kolozsvár, 2004. március 26-27.

WEB SERVICE RENDSZEREK

Gönczy László

Abstract

As use of Internet spreads dynamically in distributed applications there is a growing demand on loosely coupled systems. These integrate services running in heterogeneous environments of different providers without a knowledge of their implementation details. Web services architecture is the most current solution of this problem. However, there are a lot of open questions and uncovered aspects. Such aspects are optimization and automatic process composition. P-graphs (process graphs) developed for process synthesis provide an answer to these questions.

Összefoglaló

Az Internet széles körű elterjedésével párhuzamosan egyre nagyobb az igény olyan lazán csatolt alkalmazásokra, melyek több szervezet heterogén környezetben futó szolgáltatásait integrálja azok belső felépítésének és megvalósításának ismerete nélkül. Ennek egy megvalósítása a Web services architektúra, mely mindazonáltal jelenleg számos problémát nem fed le. Ilyen többek között az optimalizálás, illetve az összetett szolgáltatások automatikus kompozíciója. Erre egy lehetséges megoldás a folyamatszintézisre tervezett P-gráfok használata.

Bevezetés

Az Internet széles körű elterjedésével párhuzamosan egyre nagyobb az igény olyan lazán csatolt alkalmazásokra, melyek több szervezet heterogén környezetben futó szolgáltatásait integrálja azok belső felépítésének és megvalósításának ismerete nélkül. Ennek egy megvalósítása a Web services architektúra, mely mindazonáltal jelenleg számos problémát nem fed le.

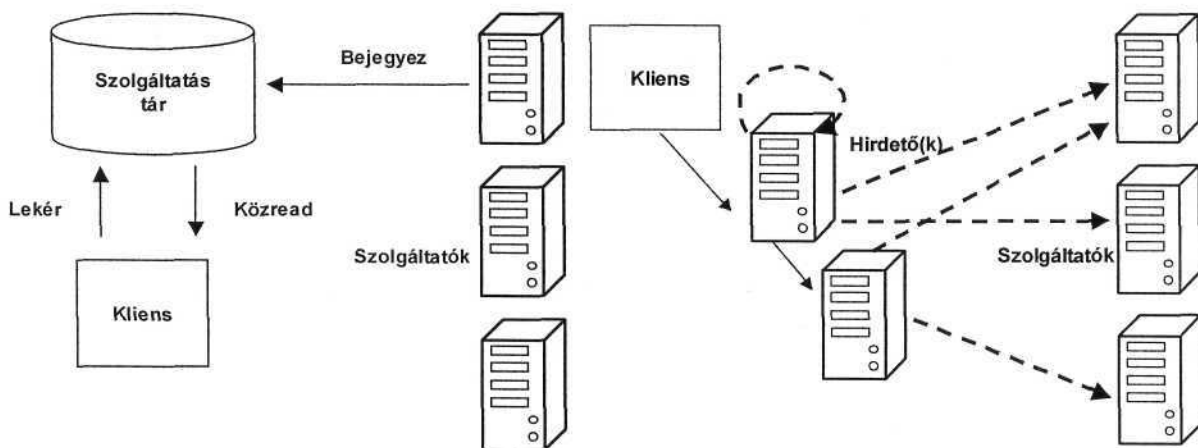
A Web services architektúra

A Szolgáltatásközpontú Architektúra (Service Oriented Architecture) paradigma lényegében a már említett lazán csatolt alkalmazások szemléletét tükrözi. Ellentétben a ma már széles körben használt objektum orientált modellezési és programozási nyelvekkel, a központban nem a rendszer objektumai és a külvilág számára elérhető metódusai, hanem a rendszer által nyújtott szolgáltatások és azok interfészei állnak, tehát magasabb szinten modellezzük az elosztott alkalmazásokat. A szolgáltatások belső működése ismeretlen, csak az interfész (mely lényegében input/output paraméterek összessége) és a szolgáltatás konkrét helyét jelölő mutató (pointer) ismert, melyek közül az utóbbi lehet akár egy webes környezetben megszokott URL is.

A Szolgáltatásközpontú Architektúra egy lehetséges megvalósítása a Web services architektúra. A *Web service* kifejezésre nincs általános definíció, egymással XML üzenetekben kommunikáló, elosztott, heterogén környezetben megvalósított, URI címükkel azonosított alkalmazásokat értünk alatta.

Szabványosnak tekinthető viszont a szolgáltatásokat leíró WSDL nyelv, ami leírja a szolgáltatás *portjait* (műveletek) és azok input/output paramétereit, tehát lényegében a szolgáltatás meghívásának *szintaktikáját*. Szintén széles körben elterjedt a SOAP (Simple Object Access Protocol), mint kommunikációs protokoll használata. A SOAP gyakorlatilag bármilyen szállító réteg (ahol ez nem feltétlenül jelent ISO szállítási réteget, lehet HTTP, JMS¹, FTP, stb.) felett működhet, elrejt az alkalmazás elől a hálózati megvalósítást, gondoskodik az *üzenetváltás* lebonyolításáról.

A szolgáltatás igénybevételéhez szükség van egyfajta katalógusra, melyben keresni tudunk kategóriák szerint. Ez a katalógus a UDDI Registry, amely egyfajta Arany Oldalak funkciót ellátva tárolja a szolgáltatások leírását (és a WSDL leírásokat). Egy másik felfogás szerint egy központi tár helyett különböző hirdetőik összesítik a szolgáltatásokat egy-egy WSIL (Web Service Inspection Language) leíróban, mely tartalmazhatja további WSIL leírók címeit is, így oldva meg a *keresés* problémáját.



1. ábra Keresés a szolgáltatások közt (UDDI, WSIL)

Mint a fenti ábrán is látható, a két módszer közt az a lényeges eltérés, hogy a UDDI katalógus használatakor egyetlen központi tárban kereshetünk, míg a WSIL többszintű keresést tesz lehetővé (a hirdetőik maguk is lehetnek szolgáltatók és hirdethetnek más hirdetőket is).

Nyitott kérdések

A fent ismertetett architektúra számos kérdést megválaszolatlanul hagy. Az egyik legfontosabb nyitott probléma a szolgáltatások *szemantikájának* megadása (pontosan mit csinál az adott szolgáltatás). A szemantikus kibővítés szükséges mind összetett szolgáltatások létrehozásához (mely szolgáltatások

¹ Java Message Service, <http://java.sun.com/products/jms>

kimenetét tudják más szolgáltatások bemenetként értelmezni), mind robusztus rendszerek tervezéséhez (mely szolgáltatások helyettesíthetők másokkal).

A szemantikus leírás egyik eszköze lehet az ontológiák használata. Az ontológia szó számítástechnikai környezetben a világ leírásának egy módját jelenti, itt „adott környezetben érvényes fogalmakat” ír le. Az ontologikus leírás ez esetben különböző profilok bevezetésével kategorizálja a szolgáltatásokat, a profilokon belül a szolgáltatások egymással felcserélhetők, mindezt a DAML-S nyelvvel írhatjuk le. A DAML-S hátránya [1], hogy gyakorlati tapasztalat nem áll rendelkezésre a módszer működőképességéről.

Az interfészükkel adott folyamatok összekapcsolása (kompozíciója) egy összetett folyamattá újabb kérdéseket vet fel. Egyrészt az üzleti folyamat modellezés egy lehetséges megvalósítása lehet a Web service alapú rendszerfejlesztés, másrészt nagyobb feladatok struktúráját meghatározva fel lehetne azokat bontani kisebb részfeladatokra, melyek optimális megoldásait keresve az egész rendszer egy optimális megoldását kaphatnánk.

Az első kérdésre több megoldási kísérlet is létezik, a legelterjedtebbnek a BPEL4WS (Business Process Modeling Language for Web Services) mondható [2]. Ez a specifikáció egy olyan nyelvet ír le, mellyel Web szolgáltatásokból építhetünk folyamat alapú rendszert, ahol az egyes lépéseket akár külső partnerek által nyújtott Web service-ek is végrehajthatják, míg az egész folyamat maga is meghívható egy Web service-ként. Erre példa egy online jegyrendelő rendszer, ami egyrészt több légitársaság kínálatát egyesíti - ezen légitársaságokat Web service interfészen keresztül lekérdezve —, másrészt más helyfoglaló rendszerek, utazási irodák számára meghívható szolgáltatást nyújt. A második kérdésre egy lehetséges megoldás a folyamatgráfok (P-gráf) használata.

Egy lehetséges megoldás: a P-gráfok használata

A P-gráf (Processz-gráf) konstrukciót eredetileg vegyipari folyamatok szintéziséhez találták ki, de alkalmas bármilyen folyamat leírására, mely valamilyen bemenő adatokból operátorok segítségével kimenetet állít elő [3]. Ezek a bemenő adatok vegyipari rendszereknél a folyamat nyersanyagai, a kimenet az előállítandó termék, az operátorok pedig a vegyipari folyamatok alapelemei, melyek lehetséges bemenet/kimenet párokkal és költséggel adottak. Ilyen lehet például egy desztilláló egység, amely A anyagot B és C anyagokra bontja. Ehhez hasonlóan a P-gráfok alkalmazhatók processzoros leírására is, ahol a processzorok hibaállapota a bemenet, a lehetséges döntések (valószínűségükkel együtt) az operátorok, és a teszteredmény a kimenet [4].

A fentiek analógiájára a P-gráfokat Web service rendszerek leírására is alkalmazhatjuk. A bemenet itt a rendszerhez intézett kérdés vagy szolgáltatáskérés, a kimenet a rendszer várt válasza, az operátorok

pedig a Web service komponensek, melyeknek adott valamilyen költsége, valamint a lehetséges bemeneteik és a rájuk adott válasz típusa. Mindennek az a mérnöki előnye van, hogy a P-gráfokon sokféle optimalizációs algoritmus értelmezhető, melyek egy része a hatékony ún. gyorsított branch-and-bound keresésre épül, mellyel akár 10,000-szeres gyorsulás érhető el. Ennél a módszernél korlátok adhatóak, melyek szűkítik a keresési teret (pl. bizonyos operátort letilthatunk, stb.) és ezzel jelentősen gyorsítják a keresést.

További feladat egy mintarendszer optimalizálása P-gráfok segítségével, valamint az S-gráfok lehetőségeinek vizsgálata. Az S-gráfokon (scheduling=ütemezés) értelmezett algoritmusok a folyamatszintézis időbeli viselkedést is figyelembe vevő kérdéseit oldják meg. Az időbeli viselkedésre vonatkozó megkötés lehet például egy részfolyamat lefutásának határideje vagy két lépés közt eltelt idő, ami a szolgáltatások esetében a kommunikációs késleltetésnek felel meg.

Összegzés

A cikkben leírtam egy dinamikusan fejlődő új technológiát, a Web service architektúrát, mely heterogén rendszerek laza összekapcsolását teszi lehetővé. Ezen technológia alacsony szintű megoldásai (szolgáltatás szintakszis, üzenettovábbítás, keresés) szabványosnak mondhatók, de számos nyitott kérdés marad megválaszolatlanul. Ezek közül a folyamatszintézis és optimalizáció problémájára javasoltam egy megoldást, a más területen már bevált a P-gráfok használatát, melynek részletes vizsgálata és tesztelése doktoranduszi munkám következő feladata.

Köszönetnyilvánítás

Köszönettel tartozom konzulensemnek, Dr. Bartha Tamásnak és Dr. Pataricza Andrásnak észrevételeikért, hasznos tanácsaikért, melyekkel munkámat segítették.

Hivatkozások

- [1] M. Sabou, D. Richards, S.v. Splunter, "An experience report on using DAML-S" in *Proc. of Twelfth International World Wide Web Conference Workshop on E-Services and the Semantic Web*, 2003.
- [2] Business Process Execution Language for Web Services, <http://www.bpmi.org/downloads/BPML-BPEL4WS.pdf>
- [3] Friedler, F., L. T. Fan, and B. Imreh, "Process Network Synthesis: Problem Definition", *Networks*, 28(2), 119-124
- [4] B. Polgár, „Multiprocessoros rendszerek maximum likelihood diagnosztikája”, *Fiatal Műszakiak Tudományos Ülésszaka*, Kolozsvár, 2003.

Gönczy László, doktorandusz

Méréstechnika és Információs Rendszerek Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem
H-1521, Budapest, Magyar tudósok körútja 2.
tel: +36 1 463 3579 fax: +36 1 463 2667
e-mail: gonczy@mit.bme.hu