

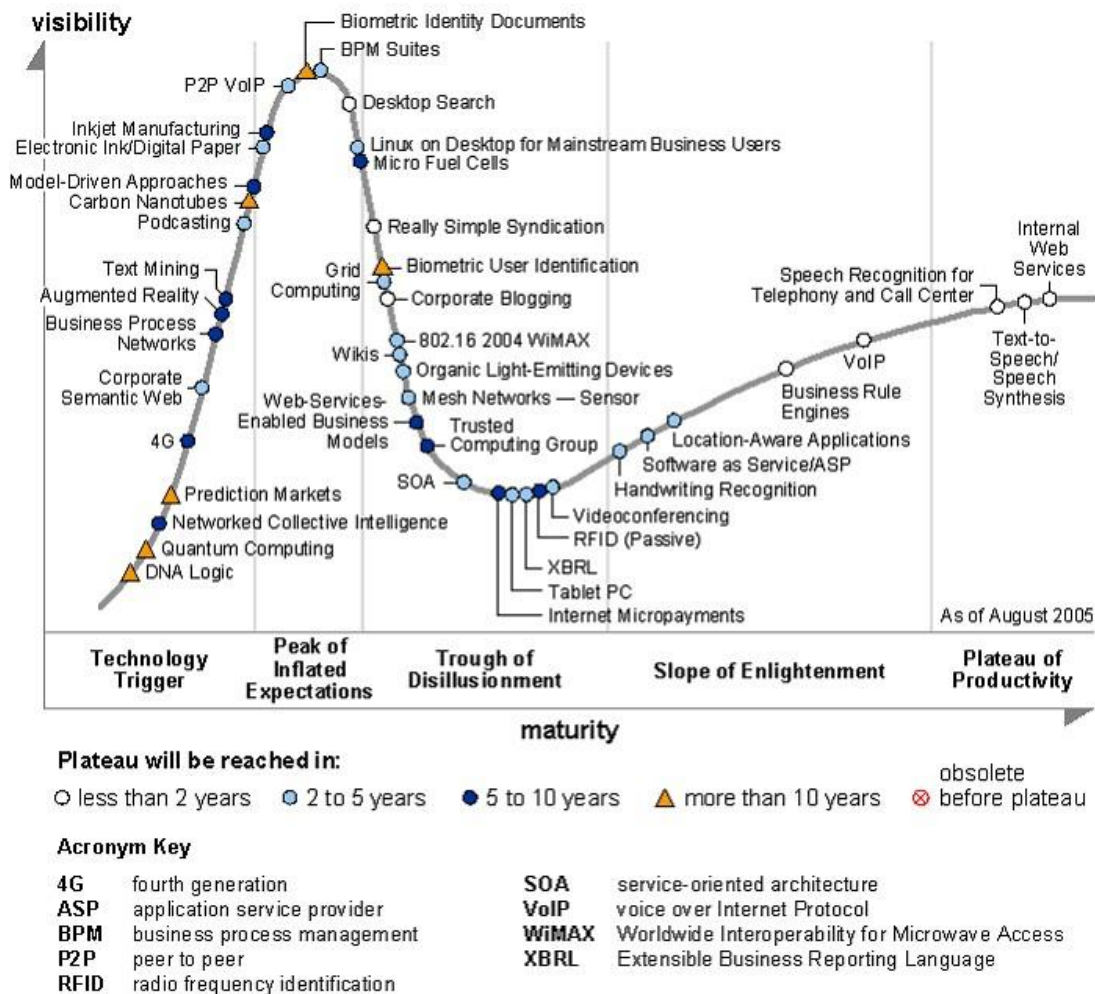
Szolgáltatás Orientált Architektúra Koncepció és alapelvek

Telbisz Ferenc, telbisz@sunserv.kfki.hu
KFKI RMKI SzHK és Magyar Telekom PKI-FI

Gódor Balázs, godor.balazs@t-com.hu
Magyar Telekom PKI-FI

Bevezetés

Az utóbbi időben meglehetősen sokat emlegetik a Szolgáltatás Orientált Architektúrát: (SOA: Service Oriented Architecture). A szolgáltatás orientáltság azt jelenti, hogy egy nagy feladat jobban megoldható, kivitelezhető és menedzselhető, ha azt egymáshoz kapcsolódó elemekre bontjuk. Mindegyik elem a nagy feladat egy-egy specifikus részét oldja meg. A SOA-nál az egyes elemek (részek) autonóm módon léteznek, de egymástól mégsem elszigetelten. Az egyes logikai egységeknek meg kell felelniük bizonyos általános elveknek, amelyek lehetővé teszik azt, hogy egymástól függetlenül lehessen őket kezelni, de emellett egységes rendszerben kell őket összefogni, és be kell tartani bizonyos szabványokat. A SOA-ban ezeket a logikai egységeket szolgáltatásoknak nevezzük [1].



1. ábra. A Gartner Hype Görbe 2005-ben

Bár a SOA-t úgy reklámozzák, mint vadonatúj architektúrát, valójában nem más, mint újra feltalálása és alkalmazása azon elveknek és gyakorlatnak – mint a strukturált programozás, top-down tervezés, az üzenetek nyugtázása, az abnormális események megfelelő kezelése, a láthatóság kérdése az egyes program moduloknál, stb. – amelyeket az egyetemeken számítástudományi kurzusain már 30 évvel ezelőtt tanítottak., de időközben elfelejtettek és nem vagy ritkán alkalmaztak a gyakorlatban. Ugyanakkor a SOA gyakran úgy jelenik meg, mint reklám jelszavak gyűjteménye és a legújabb felkapott technológiai szakkifejezések egyike. A SOA megjelenése elismerése annak a ténynek, hogy az Információs Technológia több évtizedes használata alatt az alkalmazások nagy részét (talán a többségét) silány minőséggel implementálták. Ezért a SOA nagyon szükséges új gondolkodási és munkastílus.

Ez a megközelítés nagyon sok olyan eszközt használ fel, amit a WWW megjelenése óta fejlesztettek ki, mint az XML, SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), REST (Representational State Transfer), UDDI (Universal Description, Discovery and Integration), stb.

A SOA egy új, még fejlődésben levő „technológia”. Bár a Gartner Csoport u.n. hype görbéjén 2005-ben [2] a SOA az úgynevezett kiábrándulási völgy mélyén volt (1. ábra), a Gartner csoport akkor azt várta, hogy növekedni fog a SOA mint technológia támogatottsága és tíz éven belül érett technológia lesz, bár addig még sok változásnak kell bekövetkeznie mind a felhasználói, mind a gyártó szervezeteknél és technológiáknál ahhoz, hogy a SOA elérje a potenciális lehetőségeinek a teljességét.

A Szolgáltatás Orientáltság Alapelvei

A Szolgáltatás Orientált Architektúra olyan szoftver architektúrát jelent, amelyben szolgáltatásokat használnak a felhasználók igényeinek a kielégítésére [1]. SOA környezetben az egyes hálózati csomópontok az erőforrásaikat független szolgáltatásként hozzáférhetővé teszik mások számára, amelyet azok szabványosított módon vehetnek igénybe. A szolgáltatások független, újra felhasználható modulok jól definiált csatlakozó felülettel, és függetlenek az alkalmazások implementációjától és attól a számítástechnikai (hardver-szoftver) platformtól, amelyeken azok futnak.

A SOA legtöbb definíciója a web szolgáltatásokat használja fel az implementálásnál. (L. alább.). A SOA azonban implementálható bármilyen szolgáltatás orientált technológián. Az OASIS SOA Reference Model Technical Committee [3] azon dolgozik, hogy a SOA-t technológia független módon definiálja. A Bizottság egy Referencia Modellt készít a Szolgáltatás Orientált Architektúrához [4]. Ezt elsősorban azért teszik, mert a SOA kifejezést egyre gyakrabban használják valamely specifikus technológia kontextusában és implementációjában. Ezért a Referencia Modellt azért fejlesztik ki, hogy bátorítsák a különböző differenciált SOA implementációk fejlesztését, és ugyanakkor meg lehessen őrizni egy közös értelmezést.

A SOA segíti olyan összetett alkalmazások létrehozását, amelyek több forrásból származó funkcionalitást használnak fel, akár még a vállalaton kívülről is, ilyen módon lehetővé téve horizontális üzleti folyamatokat is. Ez a vállalati IT infrastruktúra speciális stílusa, amely lehetővé teszi olyan alkalmazások létrehozását, amelyek lazán csatolt és együttműködésre képes szolgáltatásokra épülnek. Ezen szolgáltatások együttműködése egy formális definíción (szerződés) alapul, amely független az alattuk levő platformoktól és a programozási nyelvektől. Az interfész definíció elrejtja a nyelv specifikus szolgáltatás implementációt. A SOA független a fejlesztéshez használt technológiától (mint a JAVA, NET, stb.) és ezért szállító független is.

Bár a szolgáltatás orientált architektúra alapelvei nincsenek hivatalosan vagy de-facto rögzítve, létezik azonban az alapelveknek egy olyan halmaza, amit általában hozzákapcsolunk a szolgáltatás orientáltsághoz. A különböző szervezetek publikálták azt, hogy ők mit tekintenek a szolgáltatás orientáltság alapelveinek. Ennek következtében sok változat létezik. A leggyakrabban említettek a laza csatolás, autonómia, felfedezhetőség, összeépíthetőség, ismételt felhasználhatóság,

a szolgáltatásra vonatkozó szerződés, absztrakt és állapot-független. A továbbiakban ezeket vizsgáljuk meg kissé részletesebben.

A szolgáltatások ismételten felhasználhatók. Az algoritmust szolgáltatásokra kell felbontani, amelyek elősegítik az ismételt felhasználhatóságot. A szolgáltatás orientáltság megkívánja, hogy minden szolgáltatás ismételten felhasználható legyen, függetlenül attól, hogy pillanatnyilag van-e erre szükség. Az, hogy olyan tervezési szabályokat használunk, amelyek lehetővé teszik, hogy minden szolgáltatás potenciálisan újra felhasználható legyen, elősegíti annak a lehetőségét, hogy kevesebb fejlesztési erőfeszítéssel tudjuk a jövőbeli követelményekhez igazítani. Az alapvetően újra felhasználható szolgáltatások csökkentik annak szükségét, hogy később „burkoló” szolgáltatásokat kelljen készíteni, amelyek generikus interfészben rejtik el a kevésbé újra felhasználható szolgáltatást/funkciót.

Ez az elv megkönnyíti az újra felhasználás minden formáját, beleértve az alkalmazások közötti együttműködést, azok egybeépítését, és általánosan használható szolgáltatások (utility) létrehozását is. Mivel egy szolgáltatás összetartozó műveletek együttese, ezért az egyes műveletekben megvalósított logikának kell újra felhasználónak lennie ahhoz, hogy egy szolgáltatás újra felhasználható legyen.

A szolgáltatások között formális "szerződések" vannak: betartanak egy közös megállapodást, amint az egy vagy több szolgáltatást leíró dokumentumban le van írva. Ezek az alábbiak formális definícióját adják:

- a szolgáltatás határfelületét,
- minden egyes szolgáltatás leírását,
- minden egyes művelethez az összes támogatott (megengedett) bemenő és kimenő üzenetet,
- a szolgáltatásnak és működésének szabályait és jellemzőit.

A szolgáltatási szerződések ezért egy SOA csaknem minden primer tulajdonságát meghatározzák. A jó szolgáltatási szerződések szemantikus információt is tartalmazhatnak, amelyek kifejtik, hogy egy szolgáltatás hogyan működhet, amikor egy meghatározott feladatot hajt végre. Ez az információ minkét irányban létrehozza a megállapodást a szolgáltatás nyújtója és felhasználója között.

Mivel ez a szerződés közös valamennyi szolgáltatás számára, ezért rendkívül fontos. A szolgáltatás igénybe vevői, amelyek elfogadták ezt a szerződést, ettől kezdve függnék tőle és ezért azt gondosan karban kell tartani és a változatokat regisztrálni a kezdeti kibocsátás után.

A szolgáltatások között laza csatolás van, olyan viszony van közöttük, amelyik minimalizálja a kölcsönös függéseket és csupán azt feltételezi, hogy tudnak egymásról. A laza csatolás azt feltételezi, hogy a szolgáltatásnak tudomása van egy másik szolgáltatásról, de független marad attól. A laza csatolást a szolgáltatási szerződések által lehet elérni (a szolgáltatás leírása), amelyek csak azt engedik meg, hogy a szolgáltatások csak meghatározott paraméterekkel lépjenek egymással kölcsönhatásba.

A szolgáltatások az őket megvalósító algoritmusnak az absztrakcióját jelenítik meg. Azon túlmenően, ami a szolgáltatási szerződésben le van írva, a szolgáltatások elrejtik a belső algoritmusukat a külső világ elől. Ezt úgy is említik, mint a szolgáltatási interfész absztraktságát. Ez az alapelv teszi lehetővé, hogy a szolgáltatások fekete dobozként működjenek, elrejtve a külső világ elől a részleteket. A szolgáltatás belső algoritmusuk ugyanakkor jelentősen befolyásolja a működésének a definícióját és a szerepét egy folyamatban.

Nincs semmi korlátozás arra vonatkozóan, hogy egy szolgáltatás belső algoritmusuk mennyi végezzen el. Egy szolgáltatást tervezhetnek úgy, hogy egy nagyon egyszerű feladatot lát el, és lehet a kapuja egy teljes automatizált megoldásnak is. Az is lehetséges, hogy egy szolgáltatás algoritmusuk

két különböző rendszer összevont szolgáltatását nyújtja a külső felhasználónak. A működés granularitása elsődleges tervezési szempont.

A szolgáltatások autonóm szolgáltatások: teljes ellenőrzésük van a belső logikájuk fölött. A autonómia megköveteli, hogy egy szolgáltatás által megvalósított belső algoritmus behatárolt és zárt legyen. Ez teszi lehetővé, hogy a szolgáltatás saját magát kormányozza. Ugyanakkor kiküszöböli azt, hogy más szolgáltatásoktól függjön és ez megszabadítja a szolgáltatást azoktól a megkötöttségektől, amelyek gátolhatnák a telepítését és továbbfejlesztését. A végrehajtás során a szolgáltatás kizárólagosan rendelkezik az általa végrehajtott algoritmus fölött. Ez szorosan kapcsolódik a interfész szintű absztrakcióhoz – gyakorlatilag azonos vele. A szolgáltatás autonómiája az elsődleges szempont, amikor azt tervezik meg, hogy az algoritmust hogyan kell felosztani szolgáltatásokra, és hogy milyen műveleteket kell csoportosítani, összevonni egy szolgáltatásba.

Az autonómia nem jelenti azt, hogy egy szolgáltatás kizárólagosan rendelkezik valamilyen algoritmus, vagy erőforrás fölött., csak azt garantálja, hogy *a végrehajtás alatt van kizárólagos rendelkezési joga*. Ezért meg lehet különböztetni az autonómia két típusát:

- *Szolgáltatás szintű autonómia:* A szolgáltatások egymástól szigorúan elkülönülnek, de a szolgáltatás más szolgáltatásokkal megosztottan használhat bizonyos erőforrásokat.
- *Tiszta autonómia:* A szolgáltatás alatti erőforrások a szolgáltatás kizárólagos ellenőrzése alatt és tulajdonában vannak.

A szolgáltatásokat más szolgáltatásokból fel lehet építeni, azaz a szolgáltatások valamely gyűjteménye koordinálható és összerakható oly módon, hogy egy összetett szolgáltatást alkot. Egy szolgáltatás megjeleníthet különböző forrásokból származó algoritmusokat, beleértve más szolgáltatásokat is. Ezen alapelv garantálja a szolgáltatások olyan megtervezését, hogy más szolgáltatások alkatelemei lehessenek, ha arra szükség van. *Az összerakhatóság egyszerűen egy más megfogalmazása az újra felhasználhatóságnak,* ezért az egyes műveleteket szabványosított módon kell megtervezni, és megfelelő granularitással annak érdekében, hogy maximalizálni lehessen az összerakhatóságot.

A szolgáltatások állapotnélküliek, azaz minimalizálni kell azt az állapotinformációt, amit kezelnek, és az időtartamot is, amennyi ideig azt nyilvántartják. Az állapot információ specifikus az éppen folyó tevékenység adataiban, a szolgáltatás állapotörző addig, amíg feldolgoz egy üzenetet, De ha egy szolgáltatásnak hosszabb ideig meg kell őriznie egy állapotot, az komolyan akadályozza azt, hogy az alatt más tranzakciók számára hozzáférhető legyen.

Az állapotnélküliség preferált a szolgáltatásoknál, elősegíti az újra felhasználhatóságot és a skálázhatóságot. A SOA állapotnélküliségét elsősorban a dokumentum stílusú üzenetek használata teszi lehetővé. Minél több információ van egy üzenetben, annál függetlenebb és önállóbb lesz.

A szolgáltatások felfedezhetősége: a szolgáltatásokat úgy kell tervezni, hogy külső leírásuk világos legyen és meg lehessen őket találni a "felfedezési" (discovery) eljárások segítségével. A szolgáltatások olyan megtervezése, hogy természetes módon felfedezhetők legyenek, lehetővé teszi egy olyan környezet kialakítását, ahol a szolgáltatásokat új potenciális felhasználók is igénybe tudják venni. Ezért a szolgáltatás felfedezhetősége szorosan kapcsolódik az alábbi szolgáltatás orientáltsági elvekhez:

- A szolgáltatási szerződések azok, amiket a szolgáltatás felhasználói – vagy azok, akik azokat implementálják – felfedeznek és megállapítják, hogy arra van szükségük. Ezért a szolgáltatás felfedezhetősége nagyon függ a szolgáltatási szerződés világos megfogalmazásától és a kellőképpen informatív leírástól.
- A szolgáltatás újra felhasználhatósága az, amit a felhasználói néznek, amikor szolgáltatót keresnek és ez az, ami egy szolgáltatást potenciálisan felhasználhatóvá tesz, amikor

felfedezték. Semmi értelme sincs egy nem újra felhasználható szolgáltatás felfedezésének, mivel valószínűleg egy speciális felhasználó számára készült.

Az alapelvek összefüggése

Az itt ismertetett alapelvek nem függetlenek egymástól hanem nagyon szorosan összefüggnek, mindegyik következménye és feltétele más alapelveknek. Ennek részletesebb kifejtése megtalálható az irodalomban [5].

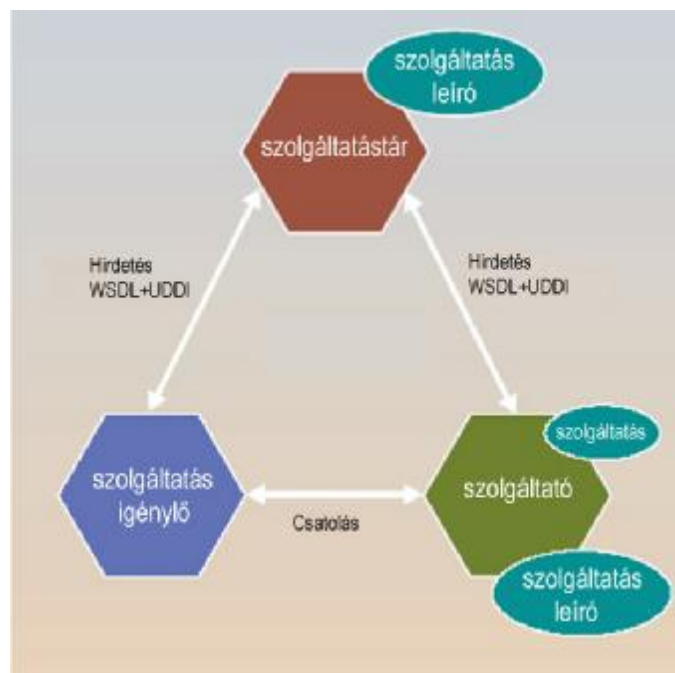
A Web Szolgáltatások Technológiai háttere

A szolgáltatás orientált rendszerhez szükség van egy implementációs platformra is. Bár sok különböző platform elképzelhető és lehetséges, a web szolgáltatások alkalmas platformként kínálóznak ehhez és a gyakorlatban a legutóbbi időkig ez volt az egyetlen, amit ilyen célra felhasználtak.

A Web szolgáltatás egy interfész, mely olyan műveletsoportot ír le, ami szabványos XML üzenetváltásokkal elérhető a hálózatról. Egy Web szolgáltatást egy szabványos és formális XML struktúra határoz meg, amit szolgáltatás leírónak nevezünk. Ez magába foglal minden részletet, ami a szolgáltatással való kapcsolatfelvételhez szükséges (művelet leíró üzenet formátumok, átviteli protokollok, stb.) Az interfész elrejtja a szolgáltatás implementációjának részleteit, lehetővé téve ezáltal, hogy hardver és szoftver független módon el lehessen azt érni. Így a Web szolgáltatás alapú alkalmazások lazán csatoltak, komponens orientáltak és platform függetlenek lesznek. Egy-egy Web szolgáltatás önmagában, vagy más Web szolgáltatásokkal együttműködve használható (komplexebb üzleti tranzakciók megvalósítása céljából).

A Web szolgáltatások modellje

A Web szolgáltatások architektúrája három funkció közötti kölcsönhatáson alapul, melyek a szolgáltató (service provider), szolgáltatástár (service registry) és a szolgáltatás igénylő (service requestor). A műveletek a hirdetés (publish), felderítés (find) és csatolás (bind) műveletek.



2. ábra. A Web szolgáltatások modellje [6]

Általában a szolgáltató birtokol egy hálózatról elérhető szoftver modult (egy Web szolgáltatás implementációt). A szolgáltató meghatározza a Web szolgáltatás leírását és hirdeti ezt a szolgáltatás igénylő, vagy szolgáltatástár felé. A szolgáltatás igénylő a felderítés műveletet használja a szolgáltatás leírás szolgáltatástárból való (vagy helyi) eléréséhez, majd ezzel tud csatolni egy szolgáltatót és kommunikálni az adott Web szolgáltatással. A szolgáltató és szolgáltatás igénylő funkciók logikai konstrukciók, egy szolgáltatás mindkét szerepkörben feltűnhet. A 2. ábra a felsorolt funkciók és műveletek egymáshoz való viszonyát szemlélteti.

Funkciók és műveletek a Web szolgáltatások architektúrájában

A web szolgáltatások architektúrájában a következő szereplők vannak:

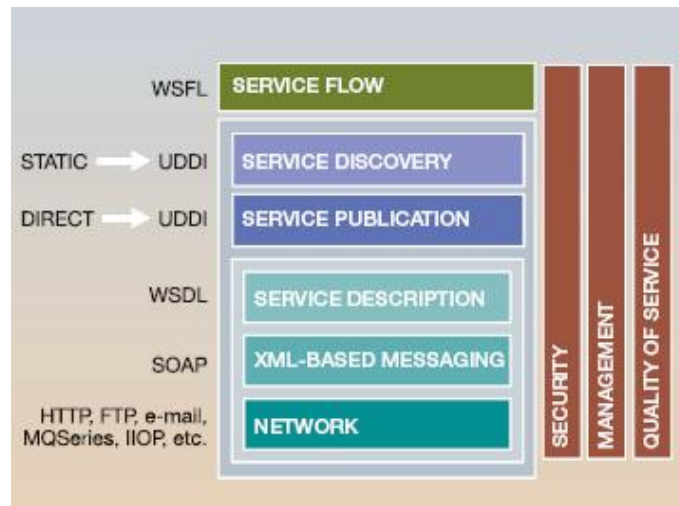
- **Szolgáltató:** Az architektúra szempontjából ez egy platform, mely a szolgáltatás elérését biztosítja.
- **Szolgáltatás igénylő:** Az architektúra szempontjából, ez az alkalmazás, mely felderít szolgáltatásokat, vagy csatolja azokat. Ezt a funkciót pl. egy böngésző is betöltheti.
- **Szolgáltatástár:** Ez a szolgáltatás leíróknak egy kereshető tára, ide töltik fel a szolgáltatók a szolgáltatás leírókat. A szolgáltatás igénylők itt deríthetik fel a szolgáltatásokat és csatolási információhoz juthatnak a szolgáltatás leírókból. Statikusan csatolt szolgáltatás igénylőknek a szolgáltatástár egy opcionális funkció, ui. statikus esetben a szolgáltató a szolgáltatás leírókat közvetlenül a szolgáltatás igénylőnek is küldheti. Hasonlóképpen, a szolgáltatásigénylők a szolgáltatástáron kívül más elemektől is igényelhetnek szolgáltatás leírókat, mint például: helyi fájl, FTP szerver, weboldal, ADS (Advertisement and Discovery of Services) vagy DISCO (Discovery of Web Services).

Ha egy alkalmazás a Web Szolgáltatásokat szeretné használni, három műveletet kell megvalósítania:

- **Hirdetés:** a szolgáltatás leírók hirdetése. A szolgáltatás leírásokat hirdetni kell, hogy a szolgáltatás igénylő el tudja azokat érni.
- **Felderítés:** a szolgáltatás leírók felderítése. A felderítés műveletével a szolgáltatás igénylő közvetlenül elérheti a szolgáltatás leírót vagy lekérdezheti a szolgáltatástárat, adott típusú szolgáltatás elérésének céljából. A felderítés művelet a szolgáltatás igénylő két különböző életciklusában is szerephez jut: 1) tervezés alatt el kell érni a szolgáltatás interfész leíróját a szoftverfejlesztéshez 2) futási időben el kell érni a szolgáltatás csatolás és hely leíróit
- **Csatolás:** csatolása vagy meghívása a leírókon alapuló szolgáltatásoknak. A csatolás művelettel a szolgáltatás igénylő kapcsolatot kezdeményez a szolgáltatással futási időben, felhasználva a csatolás információkat, melyeket a szolgáltatás leírókban tárolunk. Ezekkel tudja meghatározni a szolgáltatás helyét és azt, hogy miként lehet azt csatolni. [7]

A Web szolgáltatások architektúrája és szabványos technológiák

A hirdetés, felderítés és csatolás műveletek általános használatához szükséges egy szabványos protokoll készlet. A 3. ábra a Web szolgáltatások koncepcionális felépítését mutatja. A felsőbb rétegek az alsóbb rétegek által megvalósított eljárásokra épülnek. A függőleges oszlopok az összes síkot érintő elvárásokat reprezentálnak. A 3. ábra bal oldalán pedig konkrét protokollok, eljárások is találhatóak, megfeleltetve az egyes rétegeknek. A protokollok és eljárások részletei az irodalomban megtalálhatóak.



3. ábra. A Web szolgáltatások rétegzett architektúrája

Az Egyes Rétegek Szerepe

Hálózati réteg (Network)

A Web szolgáltatások architektúrájának alapja a hálózat. A Web szolgáltatásoknak elérhetőnek kell lenniük a hálózatból a szolgáltatás igénylők számára. Egyes Web szolgáltatások az Interneten keresztül is elérhetők; ezek eléréséhez általában HTTP protokollt használnak, de más protokollok is használhatók (SMTP, FTP). Intranetes tartományokban megbízható üzenet küldési és hívási infrastruktúra használható (pl.: MQSeries, CORBA)

XML alapú üzenetkezelő réteg (XML-based Messaging)

Az XML alapú üzenetkezelő rétegben az üzenetkezelés alapjaként a SOAP protokollt a használatos. Ennek legfőbb okai az egyszerűség (HTTP POST alapú), szabványosság és a Web szolgáltatások műveleteinek általános támogatása.

Szolgáltatás leíró réteg (Service Description)

Ez a réteg leíró dokumentumokat tartalmaz. A gyakorlatban WSDL-t használnak a szolgáltatások leírására. Ez a minimálisan szükséges szolgáltatásleírás, ami együttműködő szolgáltatások kialakításához kell. További, részletesebb specifikáció is szükséges, ami meghatározza az üzleti kontextust, a szolgáltatás minőségét, valamint a szolgáltatások egymáshoz való viszonyát.

Szolgáltatás hirdető réteg (Service Publication)

Minden művelet, ami egy WSDL dokumentumot elérhetővé tesz egy szolgáltatás igénylő számára (az igénylő életciklusának bármely szakaszában) szolgáltatás hirdetésnek minősül. A legegyszerűbb példa, hogy a szolgáltató WSDL dokumentumot küld közvetlenül az igénylőnek. Ezt közvetlen hirdetésnek hívjuk, de emellett a szolgáltató a WSDL dokumentumot egy szolgáltatástárba is feltöltheti.

Szolgáltatás felderítés rétege (Service Discovery Layer)

A szolgáltatás felderítés a hirdetésre épül, mivel egy szolgáltatást addig nem lehet felderíteni, amíg nem volt hirdetve. A hirdetési és felderítési mechanizmusoknak sokféle módja létezik. Minden olyan módszer, mely a szolgáltatás igénylő számára hozzáférést biztosít a szolgáltatás leírókhoz, és az alkalmazások számára is elérhetővé teszi azokat (futási időben), szolgáltatás felderítésnek nevezhető. Ennek legegyszerűbb példája, amikor a szolgáltatás igénylő helyi fájlból jut hozzá egy WSDL dokumentumhoz.

Szolgáltatás folyam rétege (Service Flow)

Mivel egy Web szolgáltatás egy szoftver modul, nem meglepő ha új Web szolgáltatásokat más Web szolgáltatások kombinációjával állítunk elő. Cégen belüli Web szolgáltatások együttműködhetnek pl. úgy, hogy egy egységes Web szolgáltatások interfészt alakítanak ki a külvilág számára. A szolgáltatás folyam a szolgáltatások egymásközi kommunikációját írja le.

Összefoglalás

A SOA új és fejlődő technológia. Biztosan sok előnyt nyújt a legtöbb IT szervezetnek. Az általános kiábrándulás ellenére, ami 2005-ben érte el a mélypontot, a Gartner csoport úgy gondolta, hogy hosszabb távon át fogja alakítani az üzletmenetet a vállalatoknál. Kérdéses azonban, hogy ez be fog-e következni, mivel a 2006-ban az új technológiákra vonatkozóan bemutatott véleményünkben egyáltalán nem szerepel [8].

Nem szabad azonban elfelejteni, hogy a SOA mindenekelőtt nem technológia, hanem egy gondolkodási mód és munkamódszer. amely elveket mindig is kellett volna alkalmazni, mióta csak IT rendszereket implementálunk és használunk. Ha az informatikusok egy új korszakot hozó elvként üdvözölhették a SOA-t, – márpedig ezt tették – akkor nem szabad csodálkoznunk azon, hogy a világ tele van rosszul megírt, silány programokkal. A fenti alapelveknek a nem-ismerése, vagy figyelmen kívül hagyása csak kontár munkát eredményezhet. Ettől függetlenül persze lehet és kell is örülni annak, hogy a szoftver gyártók olyan eszközökkel jelennek meg, amelyek megkönnyítik a SOA alapelveinek megfelelő rendszerek létrehozását.

Vannak SOA képes technológiák, és vannak, amelyek erre nem képesek. A Web szolgáltatások technológiája, amelyek a szolgáltatási platformot jelenthetik a SOA számára, mindenképpen fontos architektúrális elemek maradnak a jövőbeli rendszereknél is, nevezzék bár azokat SOA-nak, WEB 2.0-nak, EDA-nak, vagy bármi másnak.. Az a lehetőség, hogy az alkalmazások „építő-kockaként” használhatók, amelyeket végtelenféleképpen át lehet rendezni – és rendszerint igen gyorsan – új lehetőséget ad nemcsak a szolgáltatások és az „üzletmenet” átrendezésére, hanem a beszállítók, együttműködő partnerek és felhasználók, ügyfelek kapcsolódására is.

Irodalom

- [1] Service-oriented architecture, From Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Service-oriented_architecture
- [2] Jackie Fenn, Alexander Linden, David Cearley: Emerging Technologies Hype Cycle 2005, © 2005 Gartner, Inc. and/or its Affiliates. http://www.gartner.com/teleconferences/attributes/attr_129930_115.pdf
- [3] OASIS SOA Reference Model Technical Committee, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm
- [4] RM for SOA Public Review Document (PDF), June 2006. <http://www.oasis-open.org/committees/download.php/18486/pr-2changes.pdf>
- [5] Thomas Erl: Service-Oriented Architecture, Concepts, Technology and Design. Prentice Hall, 2005, ISBN: 0131858580
- [6] K. Gottschalk, S.Graham, H.Kreger, J. Snell: Introduction to Web Services Architecture, IBM Systems Journal, Vol41, No2, 2002
- [7] H. Kreger: Web Services Conceptual Architecture (WSCA 1.0), IBM Software Group, May 2001
- [8] Hype Cycle for Emerging Technologies, 2006, Gartner, 18 July 2006, http://www.gartner.com/DisplayDocument?doc_cd=141901