

9. SZOLGÁLTATÁSALAPÚ ALKALMAZÁSKÉSZÍTÉS

Krauth Péter

Az alkalmazási feladatok megoldásánál az egyedi programfejlesztés és a monolitikus rendszerek helyébe kész elemek szabványos szoftverarchitektúra keretében való összeépítése lép.

1. Megnevezés és rövid leírás

Az informatika egyre közvetlenebb kapcsolata az üzleti tevékenységekkel magában hordja annak lehetőségét, hogy innovatív, újszerű üzleti megoldások gyorsan, hatékonyan és eredményesen valósulhassanak meg. Ez állt eddig is – és fog állni ezután is – az informatika egyedi (értsd: innovatív) alkalmazásai mögött. Az egyedi üzleti igényeket kielégítő alkalmazások létrehozásának mikéntje azonban radikálisan átalakul. Az egyedi szoftverfejlesztés visszaszorul már meglévő komponensek, *rendszerek összeépítésének, integrációjának* rovására.

Azonban nemcsak az alkalmazáskészítés jellege változik, hanem azok az elemek is, amiből az új alkalmazások felépülnek. A nagyon szorosan kapcsolódó, és ezzel saját használhatóságukat is korlátozó, komponensobjektumok helyett, egymáshoz jóval *lazábban kapcsolódó, azonban kifelé is nyitott szolgáltatások* (pl. ún. web-szolgáltatások) elterjedése várható. Ezek felhasználásának képessége egyre döntőbbé válik a vállalatok⁸ informatikai vonatkozású versenyképességében, és ezáltal átalakítja az informatikai architektúrákat: olyan busz-architektúrára van szükség, amely biztosítja a dinamikus és hatékony építkezést a web-szolgáltatásokból.

2. Jelenlegi helyzet

Az alkalmazáskészítés területén ma új megközelítés térhódítását lehet megfigyelni, amely az alkalmazásokat lazán integrált (és jellemzően már kész) szoftverkomponensekből (ún. szolgáltatásokból⁹) építi fel: az „először összeépíteni”

⁸ A „vállalat” kifejezés a továbbiakban nemcsak üzleti (nyereségorientált) vállalkozásra utal, hanem minden olyan szervezetre, amelynek határozott elképzelése van az eredményességét illetően, azaz arról a kérdésről, hogy milyen hasznot (akár közhasznot) kell hoznia.

⁹ A „szolgáltatás”-t itt nem a hagyományos értelemben használjuk, azaz „emberek valamilyen csoportjának szervezett és meghatározott igényeket kielégítő tevékenysége”-ként, hanem olyan „szabványos szoftverkomponens”-ként, amelyet szervezeten kívüli használatra is terveztek, és ezt szem előtt tartva fejlesztettek ki.

elvet követi ellentétben a hagyományos szoftverfejlesztés „először programozni” (és csak azután összeépíteni) megközelítésével. Ez a fajta integráció egyre inkább kiterjed külső – tehát nem csak egy adott szervezetben történő, hanem szélesebb körű használatra kifejlesztett – elemek bevonásának képességére is. A kódkészítő programozó helyét az üzletet értő közvetítő (tulajdonképpen bróker) szerepe veszi át.

Az alkalmazáskészítés fő megközelítései

A korszerű alkalmazáskészítés manapság négy megközelítés együttélésével jellemezhető:

a) Szolgáltatás-orientáltság

A „szolgáltatások” válnak a szoftverek komponensekre való felosztásának elsődleges egységeivé. Az egymáshoz nyílt és szabványos felületen kapcsolódó szolgáltatásokból a korábbinál jóval rugalmasabb architektúrával rendelkező szoftvereket lehet létrehozni.

b) Architektúraközpontúság

Az alkalmazások készítésénél az architekturális szempontok meghatározó jelentőségűekké válnak. Az egyes alkalmazásokat nem elszigetelten hozzák létre, hanem olyan egységes architekturális keretben, amely következetesen teszi lehetővé az alkalmazások jövőbeni együttműködését.

c) Gyorsfejlesztés

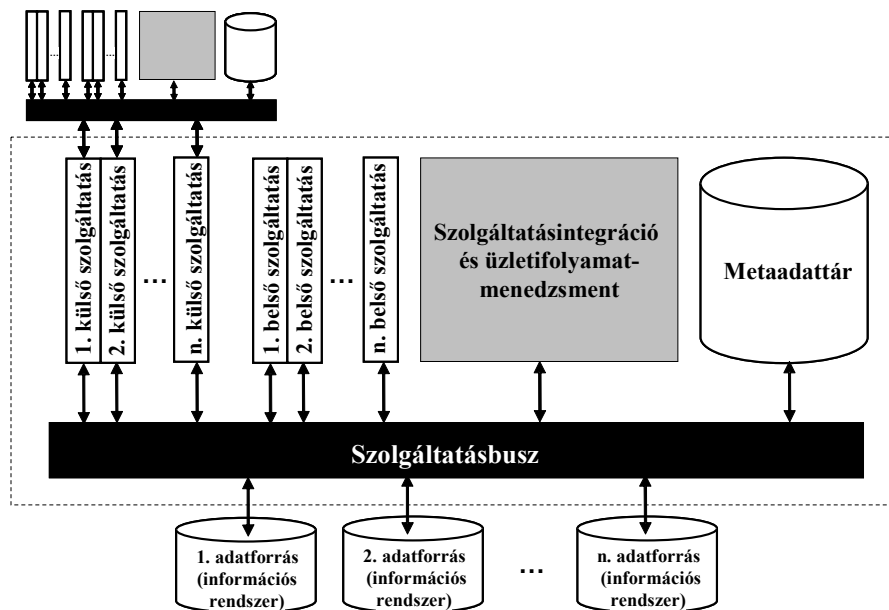
E megközelítés mögött az az egyre fontosabbá váló igény áll, hogy az alkalmazási szoftverek rövidebb idő alatt, a felhasználók nagyobb megelégedése mellett és kevesebb ráfordítással készüljenek el. Az egyre gyorsabban változó üzleti követelmények ugyanis gyakran kizárják a hagyományos megközelítések használatát: radikálisabb hozzáállást igényelnek, amit a gyorsfejlesztés a maga sajátos technikáin keresztül biztosít (a felhasználó nagyobb mértékű bevonása, a gyakoribb tesztelés, gyors visszajelzések, többfunkciós fejlesztők stb.).

d) Modell-vezéreltség

A modellvezérelt alkalmazásfejlesztés a fejlesztés tárgyát képező terület további gépi feldolgozást lehetővé tevő, formális leírásaiból, azaz modelljeiből indul ki. A modellekben az alkalmazások magas szinten és technológia- ill. platformfüggetlen módon leírt struktúrája és részletei lehetőséget adnak az alkalmazás programkódjának bizonyos szintű meggenerálására is.

Az első (a+b) ill. a második (c+d) kettő megközelítés szorosabban összefügg egymással. A modell-vezérelt fejlesztésre ugyanis éppen amiatt, mert működő programkód generálására is lehetőséget ad, a gyorsfejlesztési megközelítések gyakran építenek (célirányos, a felhasználót minden lehetséges módon bevonó projektmenedzsmenttel párosulva).

Más szempontból a szolgáltatás-orientáltság az architektúra-központúsággal együtt alkotja az ún. *szolgáltatás-orientált architektúra* (SOA, ld. 1. ábra) által jellemezhető megközelítést. Az újrahasznosítható komponensek, a jól elhatárolt funkcionális MODULok és a szabványos felületek ugyanis *egységes integrációs platformot* igényelnek a projektekhez, hogy az alacsonyabb fejlesztési költségeket, a rövidebb projekthatáridőket és az üzemeltetési feladatok egyszerűsödését ténylegesen el lehessen érni.



1. ábra: A szolgáltatás-orientált architektúra fő alkotó elemei

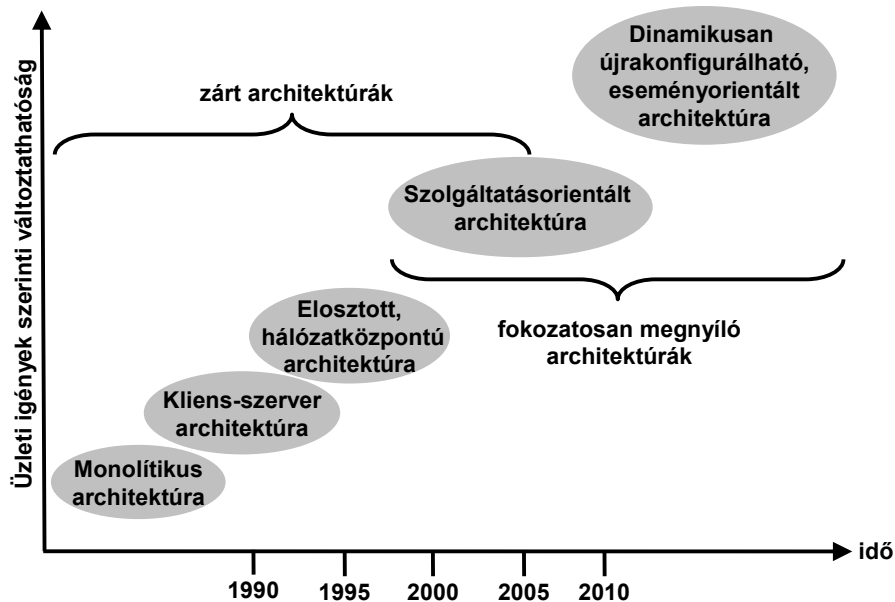
Az ilyen típusú architektúrák nagyobb mértékű elmozdulást jelentenek a ma uralkodó, zárt szoftverarchitektúrákhoz képest, mint amilyen a kliens-szerver vagy az elosztott, hálózatközpontú (és böngésző alapú) architektúra-modellekre való áttérés volt korábban a nagyszámítógép-terminál alapú monolitikus architektúrákról (ld. 2. ábra).

A SOA-elveket használva ma a szoftvercégek szolgáltatás-orientált alkalmazásokat (service-oriented application) fejlesztenek ki ún. szolgáltatás-orientált alkalmazásfejlesztési (service-oriented application development) megközelítést használva.¹⁰ Az ilyen fajta alkalmazásintegráció kulcsszerepet játszik mind a szervezeten belüli (hagyományos rendszerek újrafelhasználása), mind a szervezetek közötti (elektronikus kereskedelem, B2B) információkezelés összehangolásában.

E két fő megközelítés, azaz az architektúraközpontú, szolgáltatásorientált alkalmazásintegráció (SOA) és a modellvezérelt gyorsfejlesztés (rapid application development - RAD), különböző indítatásból kifejlődött, de azonos cél irányába mutató konvergenciája ma az egyik meghatározó tendencia.

¹⁰ Az egyszerűség kedvéért a továbbiakban a „SOA” rövidítést egyaránt használjuk a Service-Oriented Architecture, a Service-Oriented Application és a Service-Oriented Application Development fogalmak jelölésére, ha a szövegkörnyezetből meghatározható, hogy melyikről is van szó egy adott esetben.

Az eddigi tapasztalatok és vizsgálatok arra utalnak, hogy az említett négy fő megközelítést *következetesen együtt használva* a szolgáltatásorientált alkalmazásfejlesztés során, jelentős termelékenységjavulás érhető el. Az ezt biztosító, pl. ún. modellvezérelt szolgáltatásfejlesztési keretrendszerek (model-driven service frameworks) akár 80%-át is meg tudják generálni az alkalmazás programkódjának – drasztikusan csökkentve ezzel a programozáshoz szükséges időt.



2. ábra: Az IT-architektúrák evolúciója

Alkalmazásintegráció és a KÖZTES SZOFTVEREK új generációi

Az alkalmazásintegráció és az ezt támogató SOA az (alkalmazások és az operációs rendszerek közötti) KÖZTES SZOFTVERinfrastruktúrán alapul. Sok piaci megfigyelőnek volt komoly fenntartása, amikor az IBM a 90-es vége felé bejelentette, hogy felhagy az alkalmazási csomagok készítésével és forgalmazásával, és ehelyett az informatika üzleti alkalmazását jóval hatékonyabbá tevő *KÖZTES SZOFTVEREKRE* (MIDDLEWARE) helyezi stratégiájában a hangsúlyt. Mára az alkalmazásgyártók többsége felismerte, hogy ez a köztes réteg¹¹ kritikus lesz a jövő alkalmazási rendszereiben.

Nem lehet meglepő tehát, hogy az alkalmazásintegráció volt a szoftveripar területén az innováció fő hajtóereje az elmúlt időszakban. A monolitikus, elszigetelt alkalmazások folyamatosan szorultak ki a piacról, míg az új rendszerek MODULARIZáltak, elosztottak, és már nemcsak integráltak, de úgy is tervezték őket, hogy integráltak legyenek.

Vannak olyan újonnan kialakult technológiák és megközelítések az alkalmazásintegráció területén, amelyek már *bizonyítottak*, és amelyek egyre inkább

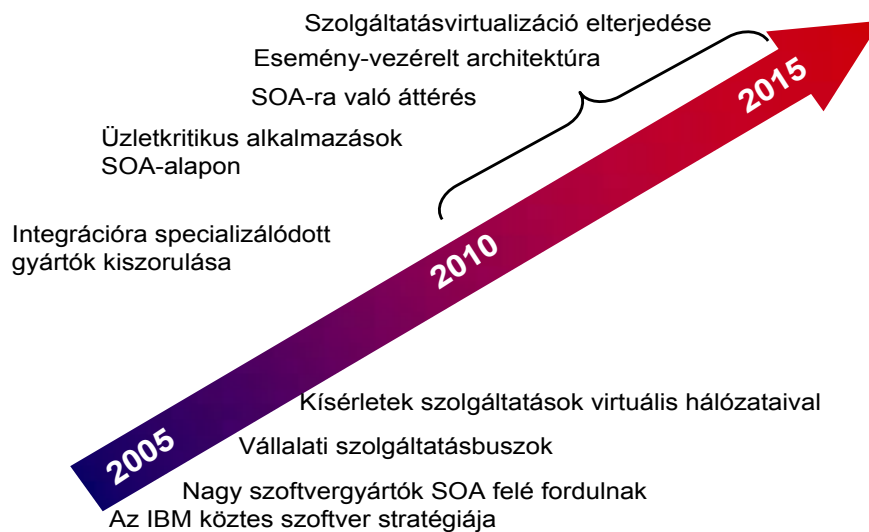
¹¹ Előfutárai közé sorolható a Common Object-Request Based Architecture (CORBA) az OMG-től, a Distributed Common Object Management (DCOM) a Microsoft-tól és az Enterprise Java Beans (EJB) a Sun-tól.

beérnek (például: web-szolgáltatások, portálermékek, üzletifolyamat-elemzés, üzletiszabály-végrehajtó motorok és összetett alkalmazások).

Jellemzője a mai alkalmazásintegrációs termékeknek, hogy éppen csak elkezdték támogatni a *központi, integrált metaadattár* koncepcióját, ami az összes integrációval kapcsolatos METAADAT egységes kezelését biztosítaná. Pedig a kereskedelmi partnerek közötti pont-pont kapcsolatok egyre fokozódó elburjánzása már ma is jól érezhető „B2B-integrációs spagetti”-t eredményez. E kapcsolatokra vonatkozó (és természetesen sok egyéb más) METAADAT hatékony és minél automatikusabb kezelése ez irányban is megoldást jelentene.

3. A várható fejlődés eredményének jellemzése

A várakozások szerint a szolgáltatásalapú architektúrák és az ezekre épülő szolgáltatásorientált alkalmazáskészítési (alkalmazásintegrációs) módszerek az évtized végére már jelentős hatás váltanak ki, míg 2015 végére meghatározóvá válnak.



3. ábra: A fejlődés mérföldkövei a SOA területén

A SOA elterjedése a szoftvercsomagok licencvásárlása helyett a szolgáltatások *előfizetése* felé fogja eltolni a bevételeket, valamint a monolitikus szoftvercsaládoktól az *összetett alkalmazások* – azaz több, különböző szolgáltatásból összeépített alkalmazások – irányába.

Már 2010 előtt az új alkalmazási szoftverekből származó bevételek nagy része SOA-t használó szoftvertermékekből fog realizálódni akár hagyományos licenc-, akár előfizetési díjak formájában. Emellett a szoftverintegrátorok és a szoftvergyártók közötti megkülönböztetés egyre inkább elmosódik, ahogy az alkalmazási csomagokat részekre bontják, és azokat SOA-ként szállítják.

A SOA-ra való áttérés 2010-re már jelentős hatást vált ki, míg 2015-re meghatározóvá válik.

Az *alkalmazásintegráció* továbbra is meg fogja határozni az innováció menetrendjét, és vezérli az alkalmazási infrastruktúrák és a köztesszoftver-termékek fejlődését. A vállalati szolgáltatásbuszok megjelenése, a B2B-átjárók (B2B-gateway) növekvő száma és a külső integrációszoftverek piaci jelentkezése csak a legismertebb példái az alkalmazásintegráció területén történő jelentős változásoknak.

Az *üzenetkezelés* a szoftverprojektek köztesszoftver-megoldásainak stratégiai modelljévé válik, és hosszabb távon is megteremti az összhangot a szolgáltatások és az események között.

A hagyományos – az üzenetek sorba állítására épülő – KÖZTES SZOFTVEREKET ki fogja váltani a *szervezeti szolgáltatásbusz* (ENTERPRISE SERVICE BUS – ESB) technológiája. Az ESB az üzenetkezelést magasabb szintre emeli. Az új ESB-gerinc, amely az integrációs termékek új generációjának is utat nyit, radikális javulást hoz a legtöbb szervezet szoftverinfrastruktúrájába.

Az üzletkritikus alkalmazásokat *összetett, multimodális architektúrára*¹² építik.

Öt év múlva a valós idejű, esemény- és szolgáltatásalapú vállalati infrastruktúra nemcsak lehetőség lesz, hanem elemi versenykövetelmény is a legtöbb iparágban és vállalatnál. 2010-re azon élenjáró, üzletkritikus alkalmazások száma, amelyek összetett, szolgáltatás-orientált architektúrára építenek, több mint háromszorosára növekszik az évtized első feléhez képest.

A *METAADAT-kezelés* fejlettsége fogja megkülönböztetni a piaci szereplőket az alkalmazásintegráció tekintetében. A METAADATTÁRAK és az ezek körüli eszközök fontos, megkülönböztető tulajdonságai lesznek a vezető platform- és integrációs termékeknek.

A *szolgáltatásvirtualizáció* lesz a következő logikus lépés a szolgáltatás-orientált és az esemény-vezérelt architektúrák továbbfejlesztésében.

A jövő szoftverarchitektúrái meg fogják követelni, hogy a szolgáltatáskérések megválaszolása a mindenkor legkényelmesebben és leghatékonyabban használható hardvererőforrásokon legyen végrehajtva.

Ezt biztosítja a szolgáltatások virtualizációja (service virtualisation), amely egy újabb absztrakciós szint megjelenését jelenti, és amely elválasztja a szolgáltatási funkció kérését egy adott szolgáltatás meghívásától. Ez a virtuális szolgáltatás-hozzáférés lehetővé teszi – mint a virtualizált, GRID-alapú hardvermegoldásoknál –, hogy az erőforrásokat akkor és úgy lehessen igénybe venni, ahogy arra a szükség van: egy mögöttes virtualizációs mechanizmus határozza meg, hogy mely szolgáltatások – vagy bizonyos típusú szolgáltatásból hány – fog válaszolni egy adott kérésre¹³.

A szolgáltatások ilyen virtuális hálózatai (virtual networks of services) 2010-re valósággyá válnak, jóllehet még további öt évet fog igényelni ennek az architektúráis

¹² Különböző technikai platformokra, gyártókra épülő és akár több szervezetet is átfogó architektúrák.

¹³ Hasonlítható ez a tárolóhálózatok (storage area network – SAN) működési elvéhez is, ahol a tárolóberendezéseket hálózatra kötik, és a SAN határozza meg azt a berendezést, ami válaszolni fog egy adott tárolási kérésre. Tárolóberendezéseket el lehet távolítani, vagy ki lehet cserélni anélkül, hogy az operációs rendszerek vagy az alkalmazási rendszerek ezt észre vennék.

megközelítésnek az olyan mértékű kiérlelődése a gyakorlatban is, amikor már a virtualizáció a szolgáltatások „GRID”-jéhez való hozzáférést megszokottabbá teszi, mint az egyedi szolgáltatásokhoz való hozzáférést.

4. Szükséges technológiai előfeltételek

A SOA kisebb mértékű fejlesztési ráfordítást ill. az ilyen ráfordítások hatékonyabb felhasználását ígéri. A SOA ugyanakkor a mainál sokkal *nagyobb kapacitásokat igényel és nagyobb követelményekkel lép fel az infrastruktúra terén.*

A SOA hallgatólagosan feltételezi, hogy a távközlés területén (gyakorlatilag) *korlátlan sáv szélesség* lesz elérhető, és megvalósul a (fizikai) *hálózatok konvergenciája*. Ilyen módon egy-egy szervezet fizikai és jogi határain kívül lévő szoftverkomponensek valós idejű felhasználása, ill. beintegrálása nem ütközik műszaki korlátokba.

Végül fontos technológiai feltétel *a felmerülő biztonsági kérdések folyamatos és megfelelő szintű kezelése*. A SOA és az igényelt sáv szélesség növeli a kockázatát az informatikai jellegű támadásoknak, mivel további nagyságrendekkel növelheti az automatikusan (közvetlen emberi kezdeményezés nélkül) létesülő hálózati kapcsolatokat, a hálózati forgalmat. Ilyen módon e kapcsolatoknál az információ hozzáférhetőségének, hitelességének, sértetlenségének, letagadhatatlanságának, számonkérhetőségének stb. biztosítása a mainál fejlettebb információvédelmi megoldásokat igényel.

5. Folyamatban lévő kutatások és fejlesztések

A nagy vállalatiszoftver-gyártók mindegyike szinte egyszerre döntött úgy, hogy a SOA-t fogja a saját fejlesztési és partnerépítési megközelítésének alapjává tenni. Az IBM, a Microsoft, az Oracle, az SAP, a Fujitsu (a Software AG-val, mint partnerrel), a BEA, a Tibco, és sok más gyártó egyaránt abban reménykednek, hogy ilyen módon tudják egyszerre mind az ügyfeleik, mind a partnereik bizalmát megtartani.

Ugyanezen szoftvergyártók a W3C keretében állapodtak meg a web-szolgáltatások használatának legalapvetőbb szabványaiiban¹⁴. A BEA (Liquid Computing) és az IBM (WebSphere Business Integration) legújabb termékei minden eddiginél jobban koncentrálnak a SOA megvalósítására és e nyílt szabványok támogatására.

A SOA-ra való áttérés folyamán az egyik legnagyobb kihívást az *üzleti folyamatok* és adatok *szemantikai egységesítése* jelenti, amelynek révén ezen az új szinten kell megvalósulnia a folyamatok közötti gyakorlati együttműködésnek.

¹⁴ Simple Object Access Protocol – SOAP, WEB SERVICE Definition Language – WSDL, Business Process Execution Language – BPEL

Az EU jelenleg több mint 70 millió €-val 16 új projektet támogat a „szoftverek és szolgáltatások” (software and services) stratégiai témakörben, amelyben közel 200 szervezet vesz részt. E projektek fő témakörei 1) a NYÍLT FORRÁSKÓDÚ SZOFTVEREK fejlesztése és használata 2) különböző élenjáró fejlesztési módszerek és keretrendszerek 3) szolgáltatás-orientált megközelítések. Ez utóbbiak célja

egyrészt az, hogy a SOA-ra épülő alkalmazások fejlesztéséhez és működtetéséhez szükséges eszközök NYÍLT FORRÁSKÓDban is rendelkezésre álljanak az EU-ban, pl.:

- *módszerek, eszközök és technikák* létrehozása megbízható szolgáltatások és *szolgáltatás-központú alkalmazások* költséghatékony fejlesztése és használata céljából (SECSE – Service Centric Systems Engineering);
- *intelligens keretrendszer* kifejlesztése, amely lehetővé teszi, hogy nyílt és bővíthető fejlesztési platformokat hozhassanak létre *WEBSZOLGÁLTATÁS-alapú alkalmazások* céljából (INFRAWEB);

másrészt, hogy egy-két különösen innovatív megoldás (pl. szemantikus web-szolgáltatás, adaptív szolgáltatásbevonás, heterogén szolgáltatásintegráció) kikísérletezésre lehetőséget biztosítsanak az EU versenyképessége érdekében, pl.:

- *nyílt fejlesztési platform* kifejlesztése adaptív szolgáltatásmegtalálás, -létrehozás, -integráció és -bevezetés céljából (ASG – Adaptive Services GRID)
- szabványalapú, átfogó megoldás biztosítása a *heterogén szolgáltatások integrációjára* modellek, nyelvek és NYÍLT FORRÁS kódú KÖZTES SZOFTVEREK kifejlesztésével (SODIUM – Service Oriented Development in a Unified Framework).

6. Az IKT más területeire való hatások bemutatása

A SOA-ra épülő alkalmazások készítésénél alapvető fontosságú az érzékenység az üzleti-szervezeti igények és a szervezeti-működési folyamatok gyors változásaira, ezért a szóban forgó témakör szoros kapcsolatban áll az alkalmazáskészítés egyéb tendenciáival – elsősorban a *modell-vezérelt alkalmazásfejlesztéssel* és a *gyorsfejlesztéssel*.

Egy másik, figyelemre méltó kapcsolat a *jelentésalapú technológiákkal* áll fenn. A szemantikus integráció egyes komponensek vagy információforrások magas szintű, akár az emberi szemlélethez is – bizonyos értelemben – közel álló összekapcsolását ígéri. Ebben kiemelt szerepe van e komponensekről ill. információforrásokról igen szerteágazó, leíró információkat tárolni és kezelni képes METAADATTáraknak. Ez jelentős hatással lesz az alkalmazásintegrációra annak későbbi (várhatóan csak 2010 utáni) szakaszában, amikor az alapvető architektúráis változások már lezajlottak, azaz a szolgáltatás-orientált ill. esemény-vezérelt architektúrák elterjedtek.

A tendencia, hogy komplex alkalmazási szoftverek a *közműszerű IT-szolgáltatás* területén is megjelennek, szintén jelentős hatást fog kifejteni. Ez erősíteni fogja azt, hogy az alkalmazásokat külön-külön is – és távolról is – jól használható komponensekre (szolgáltatásokra) bontsák fel, és így ezek SOA-kompatibilisek legyenek. Másrészt arra is pozitív hatással lesz, hogy az ilyen módon egyre növekvő

számú szolgáltatások GRIDbe – tulajdonképpeni virtuális hálózatba – szervezhetők legyenek.

A web-szolgáltatásokra épülő alkalmazáskészítés (SOA) bizonyos tekintetben a *NYÍLT FORRÁSKÓDÚ SZOFTVEREK* használatának „versenytársa”-ként is megjelenhet, mivel kisebb, menedzselhetőbb, szabványosabb, és ezért konkurens termékre könnyen lecserélhető szolgáltatások esetén kevésbé élesen merül fel a forráskódhoz való szabad hozzáférés kérdése. Ráadásul gyakran nem igazán megalapozottan vélik, hogy jelentős költségeket lehet megtakarítani a NYÍLT FORRÁSKÓDÚ SZOFTVEREK alkalmazásával, mivel ez nem feltétlenül és nem automatikusan teljesül – különösen, ha teljes körű költségszámítást (total cost of ownership – TCO) alkalmazunk. Emellett az üzletileg kritikus alkalmazásokat szabványos, jól méretezhető, megbízható infrastruktúrán érdemes csak megvalósítani.

A SOA terjedése még tovább erősíti a *biztonságtudatos fejlesztés* szükségességét, mert növeli a szervezetek közötti egymásra utaltságot, a függőséget távoli rendszerek működésétől. Az IT-szolgáltatók (web-szolgáltatásokat nyújtók) elemi érdeke lesz a „megbízható partner” státusának elnyerése és fenntartása, amiért minden tőlük telhetőt meg fognak tenni az információvédelem és a biztonságos szolgáltatás területén. Sajnos, ez fordítva is igaz: ha nem következik be radikális fejlődés a biztonságtudatosság terén a szolgáltatók körében, akkor az erősen visszafoghatja a SOA terjedését.

Végül a SOA, mint a hálózatban összekötött, komplex alkalmazások fejlesztésének nagyüzemi „módszere” számos más technológiai területen is várhatólag meg fog jelenni, mint pl. *SZENZORrendszerek, ÁGENSEK, P2P-megoldások*.

7. Társadalmi-gazdasági hatások elemzése

A kiterjedtebb nagyszámítógépes (mainframe) bázissal rendelkező országokkal ellentétben a közép-kelet-európai régió országai modernebb technikát használnak, aminek előnyeit a SOA bevezetésénél is élvezhetik. Az arányokat tekintve a térség vállalatai integrációs és portálalkalmazások bevezetésében ma megelőzik nyugat-európai versenytársaikat.

Korszerű vállalatirányítási eszközöket azonban ma még jobbra csak nagyvállalatok tudnak megfizetni. A SOA-eszközök gyártói is csak nemrégiben kezdtek el foglalkozni a közép-vállalatok felső harmadával (> 10 MFt forgalom).

Az alkalmazáskészítés fentiekben körvonalazott szolgáltatás-orientált átalakulása és ezzel egyidejűleg a monolitikus alkalmazási csomagok felbontása kezelhetőbb szolgáltatási komponensekre ugyanakkor nagyban elősegítheti, hogy a *kis- és közép-vállalkozások* is az üzleti igényekre gyorsan reagálni tudó alkalmazásfejlesztési ill. -integrálási képességeket szerezhessenek – jelentősen növelve ezzel versenyképességüket.

A bekövetkező változások kihasználása azonban csak *következetes IT-beruházási stratégiával* érhető el – az ehhez szükséges finanszírozási és szolgáltatói háttér biztosításával együtt. A közép-vállalkozások számára így az is fontos, hogy a legkorszerűbb technológiákat *helyi IT-szolgáltatók* is képesek legyenek megbízhatóan

és költséghatékonyan szállítani. A SOA-val kapcsolatos technológiai változások más oldalról azt jelentik, hogy a globalizáció a testre szabható alkalmazások területére is kiterjed – gyakran a helyi IT-szolgáltatók rovására.

Az integráció előtérbe kerülése az alkalmazáskészítésben felveti a *IT-képzési programok alkalmasságának* kérdését is. Várhatólag egyre kevesebb programozóra lesz szükség, viszont növekszik azok iránt a kereslet, akik a *megvalósítások előkészítésében, felügyeletében és átvételében* tudnak jól közreműködni (pl. üzleti igények megértése, rendszertervezés, folyamatos közvetítés az IT és a szervezet között, projektmenedzsment). A megvalósítás terén is elsősorban a korszerű alkalmazáskészítés négy – említett – megközelítésében való kellő jártasság megszerzésére kell összpontosítani, és nem a hagyományosnak mondható programozásra. Ezek lesznek ugyanis azok a kompetenciák, amelyek hosszabb távon is megmaradnak Európában.

8. Magyar vonatkozások

Magyarországon az egyedi alkalmazásfejlesztés „hagyományosan” aránytalanul nagy piaci részesedéssel rendelkezik összehasonlítva a vezető nyugati országokkal. Ugyanakkor a hazai szoftverfejlesztő szervezetek versenyképességben nem tudnak lépést tartani olyan országokkal, mint India, Fülöp-szigetek, Oroszország, Ukrajna és Románia – *költségekben már nem, minőségben még nem* tudjuk felvenni a versenyt. Mindez pedig párosul a magyar piac túlságosan kis méretével, ami nem teszi lehetővé *megfelelő méretgazdaságosság* elérését a hazai szoftvertermékek piacán.

A *vezető magyar alkalmazásfejlesztő cégek* (pl. IQSYS, Alerant, Fornax, FreeSoft) évekkel ezelőtt megismerkedtek az alkalmazásintegráció alapeszközeivel (pl. BEA WebLogic, IBM WebSphere), és azóta több sikeres projektben is használták azokat – elsősorban a távközlés és a pénzügy területén. A vonzó üzleti modell (létező alkalmazások újrafelhasználása), az integráció eszközkészletének minőségi javulása és a biztató tapasztalatok együttesen az *alkalmazásintegrációt* egyre gyakrabban használt megközelítéssé teszik minden alkalmazásfejlesztési cégnél – a jelenleginél jóval szélesebb alkalmazási területen (pl. ipar, közlekedés). Keresleti oldalról ezt segíti elő, hogy számos nagyvállalatnál 30-50, de gyakran 100-nál is több alkalmazást működtetnek.

Érdekességképpen megemlítendő, hogy a legkorszerűbb külföldi alkalmazásintegrációs eszközök birtokba vétele mellett az IQSYS (ill. korábban az IQSOFT) hazai ill. nemzetközi K+F projektek segítségével törekszik a VÁLLALATI INFORMÁCIÓINTEGRÁCIÓ és METAADATTár-kezelés szűkebb területein olyan élenjáró technológia kialakítására és ehhez kapcsolódóan technológiai ismeretek megszerzésére, amelyek felhasználásával középvállalati piacon versenyképes integrációs megoldások szállítójává válhat.

Egyes nagy cégek, mint a Vodafone, a T-Mobile, a MOL, az OTP, a BKV és más nagyvállalatok figyelemre méltó előrelépést tettek a SOA irányában. A *szolgáltatás-orientált architektúrára való teljes áttérés* azonban a vállalati IT-architektúra jelentős átalakítását igényelné. Jelenleg elsősorban csak kísérleti jelleggel próbálkoznak magyar szervezetek (pl. OSZK) kívülről származó web-szolgáltatások

felhasználásával. Jóval többen tervezik a web-en megjelenő szolgáltatásaik web-szolgáltatássá való átalakítását (pl. elektronikus áruházak, könyvesboltok), azonban ez még messze van a SOA és az ESB léptékű váltástól.

9. Következtetések

Az *alkalmazásfejlesztés és -integrálás* egyre szorosabban összefonódik a szoftverarchitektúrákkal és az ezeken működő szoftverekhez való hozzáférés szolgáltatásszerű és szolgáltatásként történő biztosításával.

A „semmitől egy új” alkalmazást készítő programozó helyét fokozatosan a létező alkalmazásokat mindenféle más célokra is felhasználni tudó *integrációs bróker* szerepköre veszi át, aki jól meg tudja érteni egyrészt az üzleti igényeket, másrészt hatékonyan képes az adott igényekhez alkalmazható, már létező (külső vagy belső) komponenseket megtalálni és ezeket összeépíteni.

Amennyire „védi” azonban a létező alkalmazásokat az alkalmazásintegráció, legalább annyira fogja felforgatni a szervezetek informatikai architektúráját és szoftverinfrastruktúráját a szabványos web-szolgáltatásokra építő *szolgáltatás-orientált architektúra* és az ezt támogató vállalati szolgáltatásbusz mint korszerű köztesszoftver-megoldás.

A kívülről – ma még csak – paraméterezhető alkalmazási csomagokat külön-külön is jól integrálható, *szabványos szolgáltatáskomponensekre* bontják, és így is forgalmazzák őket. Ugyanakkor ez nem jelenti automatikusan, hogy NYÍLT FORRÁSKÓDÚAK is lesznek. Sőt, az ilyen komponensek egyfajta kihívást is jelentenek a NYÍLT FORRÁSKÓD mozgalma felé: a forráskód nyíltsága által keletkező bizalomnövekedés összevethető a komponens szabványossága miatt bekövetkező kockázatsökkenéssel.

A korszerű alkalmazáskészítés mindegyik megközelítése (szolgáltatás-orientáltság, architektúraközpontúság, gyorsfejlesztés, modell-vezéreltség) kézzel fogható *gazdasági hasznot* képes hozni, de együttes használatuk megsokszorozza ennek lehetőségét.