



**Budapesti Műszaki- és Gazdaságtudományi Egyetem**  
Villamosmérnöki és Informatikai Kar  
Elektronikai Technológia Tanszék

# **Séma-illesztési módszerek és futásidejű megoldások analízise és fejlesztése szolgáltatás alapú vállalati alkalmazás- integrációban**

---

Doktori értekezés téziszfüzete

Martinek Péter

Témavezető: Dr. Szikora Béla

Budapesti Műszaki és Gazdaságtudományi Egyetem  
Elektronikai Technológia Tanszék

2009.

## **A kutatás előzményei – Vállalati rendszerek integrációja, szolgáltatás alapú architektúra és módszertan**

A vállalati információs rendszerek elterjedése a múlt század '70-es éveiben kezdődött. A hardver eszközök és az informatika fejlődése maga után vonta a papír alapú adattárolás és továbbítás, valamint az emberi tevékenység informatikai rendszerekkel történő kiváltását. Továbbá egyre inkább bevett szokássá vált különféle számítógépes rendszerek használata a manuálisan nem átlátható, összetett vállalati folyamatok felügyeletére. Az ezredforduló után már nem létezett gyakorlatilag olyan nagy (vagy akár közép illetve kisméretű) vállalat és vállalkozás, mely ne használt volna valamilyen informatikai rendszert mindennapi folyamatainak informatikai támogatására.

Az elmúlt évtizedben újabb és újabb gyártók különféle rendszerei fokozatosan terjedtek el, lefedve más és más területeket a vállalat üzleti folyamataiból és napi tevékenységéből. A vállalati rendszerek elterjedésének fokozatosságából következik azonban azok heterogenitása is, mely heterogenitás megfigyelhető különböző vállalatok informatikai rendszerei között, de akár adott vállalaton belül is. Végeredményként a szervezetek adatai és folyamatai a különféle vállalati rendszerek mentén több, mesterséges darabra szakadtak. Ezen darabok újraegyesítésének, illetve a több rendszeren keresztül átvitelő szervezeten belüli és szervezetközi üzleti folyamatok megvalósíthatóságának igénye vezetett el a vállalati rendszerek integrációjához. A vállalati információs rendszerek integrációja, vagy más néven vállalati alkalmazás integráció néhány év alatt az informatika talán legnagyobb feladata és kihívása lett.

Az alig 4-5 éve megjelent szolgáltatás orientált architektúra (Service Oriented Architecture – SOA) és a kapcsolódó módszertan alapjaiban változtatta meg a vállalati informatikai rendszerek világát. A korábban csak elméletben létező folyamatelvűség és folyamatszemplélet a SOA módszertannal és gyakorlati eszközeivel kiegészülve teljes, éles vállalati környezetben is alkalmazható módszerré vált.

Mindezek után kézenfekvő, s nagy reményekkel kecsegtető törekvés a SOA koncepció és eszközpark alkalmazása vállalati rendszerek integrációjában. Standard vállalati információs rendszerek szolgáltatás alapú interfészeinek kialakítása után, az összetett üzleti folyamatok komponálására nem csak az egy rendszeren belüli adatcserét, de a különféle vállalati rendszerek integrációját is megvalósíthatja, szabványos üzleti folyamatok SOA alapú implementációjával.

## A kutatási terület egyes megoldandó problémái

Az elterjedt vállalati rendszerek sokféleségéből következik azok interfészeinek heterogenitása is. E probléma áthidalására a SOA megfelelő módszertani és eszköztámogatást nyújt az egyes standard interfészek és protokollok SOA kompatibilissá alakításával. Kiforrott szabványok és protokollok, mint például a WSDL (Web Service Description Language) a Web szolgáltatások interfészeinek specifikálásához vagy éppen a SOAP (Simple Object Access Protocol) az üzenetküldéshez, biztosítják a közös nyelvet és felületet a vállalati rendszerek együttműködéséhez. Mindazonáltal az integráció során komoly nehézségekbe ütközhetünk a különféle rendszerek eltérő adatrepresentációi következtében. Ezen eltérések az egyes rendszerek fejlesztése során kialakult különbségekből adódnak.

Nagyon gyakori, hogy például a különböző vállalatirányítási rendszereket fejlesztő szakértők a valós világ azonos entitásait más adatstruktúrával reprezentálták az egyes rendszerekben. Példaként vegyük azt az esetet mikor az *A* rendszerben a földrajzi címeknél az *utca* és *házzszám* külön mezőkben van eltárolva. Ezzel szemben elképzelhető, hogy egy másik, *B* rendszerben, mindez egy mezőben, a *cím* mezőben szerepel egy speciális karakterrel, a pontosvesszővel elválasztva. Tegyük fel, hogy egy összetett folyamatban egymás után következik *A* rendszer 1. szolgáltatása melynek kimenete egy cím és *B* rendszer 2. szolgáltatása mely bemenetként egy címet vár. Habár ezen egymással összekapcsolt szolgáltatások interfészei első ránézésre kompatibilisnek tűnhetnek, azaz sorban az 1. után következhetne a 2., *B* rendszer nem lesz képes önállóan feloldani az adatrepresentációs különbséget, és nem fogja tudni megfelelően értelmezni az *A* rendszer 1. szolgáltatásától kapott bemenetet. Ilyen és hasonló inkompatibilitási problémákkal gyakran találkozhatunk a vállalati alkalmazások integrációja során. Így ezek megoldása fontos, s valós feladatot jelent.

A szakirodalomban több megoldást találunk vállalati rendszerek folyamat szemléletű integrálására. Ezen munkák nagy része komoly figyelmet fordít az integráció megvalósítását képező folyamatok komponálására, azaz a megfelelő szolgáltatások megtalálására és az üzleti logika implementálására [2][4][5][7]. Ezen kutatások közül több is foglalkozik a fent említett példához hasonló adatrepresentációs különbségek áthidalásával [5][7].

Az adatrepresentációs különbségek megoldását megfelelő transzformációk definiálása és szolgáltatás interfészhez való kötése jelentheti. Ha például a fentebb bemutatott problémát akarjuk megoldani, akkor szükségünk van egy transzformációra, ami az *A* rendszerből érkező címek *utca* és *házzszám* mezőinek tartalmát összevonja egy értékbe, s közéjük pontosvesszőt rak mielőtt azt *B* rendszernek továbbítaná. A másik irányba pedig egy olyan

transzformációt kell használnunk, mely a  $B$  rendszer mezőjében szereplő értéket a pontosvessző mentén az utcát és a házsámot reprezentáló darabokra szedi szét az  $A$  rendszernek történő továbbítás előtt. Ezen transzformációkat alkalmazva a két rendszer szolgáltatásainak kommunikációjában a cím entitás adatrepresentációs különbsége áthidalható.

Transzformációk definiálására is számos példát találunk a szakirodalomban. Legelterjedtebb megoldás a szolgáltatások leírásának szemantikus bővítése mellett értelmezett transzformációk definiálása, melyek alkalmasak lehetnek adatrepresentációs különbségek kezelésére egy speciális, ontológiának nevezett tudásbázis felett. Ezen megközelítések legtöbbször komoly elméleti modellel és absztrakt háttérrel is rendelkeznek [3][15][14] s a kész, futásidejű megoldások bemutatása helyett sokszor az elméletre helyezik a hangsúlyt.

Másrészről, az irodalomban fellelhetőek futásidőre is optimalizált megoldások, melyek a nem funkcionális teljesítmény paraméterek (például rendelkezésre állás, hibaszázalék, átfutási idő) javítására törekcsenek [1][3][11].

A vállalati rendszerek integrációja során fontos mindemellett a már meglévő szabványos eszközök (vállalati rendszerek, folyamat-tervező és futtató eszközök stb.) megtartása. Ez kizárja a legtöbbször csak kísérleti implementációval rendelkező szemantikus szabványokat és technológiákat. Nem elképzelhető továbbá azon egyedi fejlesztésű keretrendszerek használata sem, melyek az optimalizálást a szolgáltatások folyamatos monitorozásával és dinamikus újracsatlakoztatásával érik el. Mindezek helyett, egy szabványos eszközökre és elterjedt szabványokra épülő szolgáltatás alapú integrációs megoldásra van szükség, mely képes transzformációk kezelésére, s csak elfogadható futtatási pluszköltséget eredményez.

A nemfunkcionális teljesítmény-paraméterek – köztük az áteresztőképesség és a válaszidő – becslése és számítása szintén sok helyen fellelhető a szakirodalomban, s az ismertett módszereknek egy része alkalmazható folyamatokra és szolgáltatás alapú architektúrákra is [6][16][17].

Vállalati rendszerek integrációja során a helyes elméleti modell megalkotása mellett legalább olyan fontos egy teljes, azaz futásidőben is alkalmazható megoldás kidolgozása. A helyes módszertan és összetett architektúra definiálása mellett tehát fontos lenne a gyakorlati alkalmazhatóságra esetleg egy működő keretrendszer építésére is koncentrálni. Egy adatrepresentációs különbségeket is áthidaló (azaz szükséges transzformációkat is tartalmazó) gyakorlatban is működő megoldás bemutatása új fejezetet nyithatna ezen megoldások vállalati alkalmazásában.

Továbbá tanulva a szakirodalom nem funkcionális futásidejű teljesítmény-paraméterek optimalizálására folytatott törekvéseiből, a transzformációkat is tartalmazó SOA alapú rendszerek hatékonyságának beható tanulmányozása is releváns feladat. Szolgáltatás orientált architektúrára történő transzformációkat is tartalmazó integrációs megoldások elemzésére és összehasonlítására érdemes volna tehát egységes és általános módszert is kidolgozni.

A tématerület egy másik összetett problémája a különböző rendszerek (illetve szolgáltatási interfészeik) adatstruktúráinak összehasonlítása, mely a transzformációk készítésének egy elengedhetetlen előfeltétele. Mivel a vállalati rendszerek világában nem ritka az 50-100 entitást tartalmazó bemenet/kimenet interfész struktúra sem, ezért nem lebecsülendő feladat a transzformációs függvények argumentumaiban szereplő – azaz egymáshoz képest tartalmilag releváns – entitások megtalálása és összepárosítása. A megfelelő párosítás megtalálását és egy, esetleges mindkét adatsémával kompatibilis séma létrehozását a szakirodalom séma-illesztésnek hívja [8][10][12].

Több rendszer integrációjakor először egy közös, ún. globális sémát szoktak definiálni, elkerülendő a pont-pont kapcsolatokat, majd ehhez igazítják valamennyi, az integrációban résztvevő vállalati rendszer összes szolgáltatásának az interfészét. A globális sémához történő illesztés tovább fokozza a séma-illesztési feladat komplexitását, hiszen míg a szolgáltatások esetén maximum 50-100 entitásról beszélünk, addig egy teljes alkalmazásterületet (domaint) lefedő globális séma akár több ezer entitást is tartalmazhat [18][19]. A vállalati rendszerek szolgáltatásainak globális sémához történő illesztése tehát már egyáltalán nem kezelhető azt támogató (vagy megfelelően előkészítő) automatikus eljárás nélkül.

A sémaillesztés legtöbbször egy az entitások közötti tartalmi kapcsolatok erősségét kifejező hasonlósági értékeket meghatározó eljárásra épül. Az ezt végrehajtó algoritmus bemenete tipikusan 2 séma, kimenete pedig a sémák egyes entitásai közötti hasonlósági értékek 0 és 1 közé súlyozva. Az 1-hez közel álló értékek valószínűleg tartalmilag kapcsolódó entitásokat jeleznek, így vizsgálódásunkat a későbbiekben elég csak ezekre korlátozni. Más szavakkal, a transzformációk készítésekor ezeket kell felhasználnunk közös transzformációs szabályokban. Az egymáshoz nem kapcsolódó (azaz alacsonyabb hasonlósági értékkel rendelkező) fogalmak azonos transzformációs függvénybe történő helyezésének szükségessége nem valószínű. A hasonlóság elemző algoritmusok alkalmasak továbbá a meglehetősen terjedelmes globális séma azon részhalmazának automatikus kiválasztására, amelyet használnunk kell egy adott szolgáltatás interfészének globális sémához történő illesztésekor.

A szakirodalom már több éve foglalkozik sémaillesztési és hasonlóság-elemző megoldásokkal. Az ismertett megközelítések sokfélesége és nagy száma egyértelműen mutatja, hogy fontos és élő kutatási területről van szó. A szerzők újabb és újabb algoritmusokat dolgoznak ki s elfogadott mértékek és módszerek segítségével bizonyítják annak hatékonyságát más a szakirodalomban található módszerekkel összehasonlítva [9][10]. A szerzők megközelítésük hatékonyságát sokszor több tesztésén illetve valós vállalati sémákon futtatott kísérletekkel a gyakorlatban is igazolják [8][13].

A hasonlóság-elemző módszerek elterjedésének egy komoly akadály, hogy sok megközelítés rendkívül komplex, s így nagy számítási igénnyel rendelkezik. Habár a szakirodalom főleg a publikált megoldások pontosságára koncentrálna, érdekes volna részletesen megvizsgálni az egyes megközelítések komplexitását is, mivel ez döntően befolyásolja a sémaillesztési algoritmusok gyakorlati alkalmazhatóságát. Egy nem túl magas számítási igénnyel rendelkező, de kellően pontos algoritmus pedig a napi gyakorlatban is alkalmazható megoldást jelenthetne SOA alapú integrációkban.

A szakirodalomban előforduló elméleti és kísérleti elemzések során a szerzők legtöbbször csak 1-1 tesztésén futtatva, adott konfigurációs paraméter beállításokkal hasonlítják össze a megoldások pontosságát. Figyelembe véve a létező hasonlóság elemző megoldások és az adatsémák sokféleségét, hasznos lehetne egy pontosságot meghatározó összetett módszertan kidolgozása is, mely számol a pontosság tesztésére és konfigurációs paraméter beállítás függésével is.

## **Célkitűzések, módszertan**

Megcélözva a fenti kutatási területek főbb problémáinak és nyitott kérdéseinek megoldását, kutatásaimat az alábbi célkitűzéseknek megfelelően végeztem:

Célul tűztem ki egy olyan módszertan megalkotását illetve keretrendszer tervezését és építését, mely

- támogatja vállalati rendszerek szolgáltatásainak globális sémához történő illesztését,
- lehetővé teszi a transzformációk adaptációját,
- képes a transzformációk valós idejű futtatására,
- előre jelezhető, s nem jelentős a járulékos számítási igénye, és ebből kifolyólag nem jelentős teljesítmény-visszaesést mutat a transzformáció nélküli összetett folyamatpéldányokhoz képest,
- kizárólag nyílt szabványokon és kapcsolódó szabványos eszközökön működik, továbbá

- nem teszi szükségessé a vállalati rendszerek standard szolgáltatásainak és azok interfészeinek a módosítását az integráció során.

Célom volt továbbá egy hatékony – akár a mindennapi gyakorlatban is használható séma – illesztési algoritmus tervezése és készítése is. Az algoritmus a vállalati rendszerek SOA alapú integrációjában résztvevő szolgáltatások bemenet/kimenet sémáira kell, hogy fusson. Így egy XML, illetve XML sémára működő megoldást szerettem volna létrehozni, mely adott szolgáltatások sémájához megtalálja a globális séma kapcsolódó fogalmait. Ez rendkívül hasznos segítséget nyújthat a transzformációk definiálásának előkészítéséhez.

Mivel a sémaillesztési feladatra sok megoldási javaslat létezik a szakirodalomban, ezért mindenképpen szükséges összevetni a saját magam által javasolt algoritmus pontosságát és futásidő szükségletét más létező javaslatokkal. Ezt egyrészt elméleti úton, majd implementációk segítségével konkrét futási idők mérésével kívántam elvégezni.

## Tudományos eredmények

### I. téziscsoport

#### Séma illesztési módszerek valamint azok hatékonyságának és bonyolultságának vizsgálata

##### I.1. Általánosan alkalmazható módszertant és algoritmust dolgoztam ki különböző adatsémák entitásainak összehasonlítására

Módszertant és algoritmust dolgoztam ki a SOA integrációs megoldásokban előforduló paraméter-transzformációk definiálásának előfeltételét képező entitások közötti kapcsolatok tartalmi erősségének automatikus meghatározására. A lokális sémák globális sémához történő illesztésének megoldását támogató összetett típusok (entitások) közötti hasonlósági értékeket egy három tagfüggvényből álló súlyozott átlag adja. A három tagfüggvény az entitások neveit, kapcsolt fogalmait és attribútumait tartalmazza, ezért NTA (Name, connected Terms, Attributes) algoritmusnak neveztem el. Az NTA algoritmus tetszőleges, vállalati alkalmazás-integrációban előforduló interfészekre alkalmazható, s minden esetben eredményre vezet.

Az NTA algoritmus hasonlósági mérték számításának alapját a következő összefüggés adja:

$$F(C_i, G_j) = w_1 * N(C_i, G_j) + w_2 * T(C_i, G_j) + w_3 * A(C_i, G_j),$$

ahol

az  $N(C_i, G_j)$  függvény két összetett entitást hasonlít össze nevük alapján,

a  $T(C_i, G_j)$  függvény összehasonlítja a  $C_i$  és  $G_j$  összetett entitáshoz

kapcsolódó fogalmakat.  $T(C_i, G_j)$  a következőképpen számítható:

$$T(C_i, G_j) = \frac{2 * \|Term(C_i) \cap Term(G_j)\| + \|Term(C_i) \nabla Term(G_j)\|}{\|Term(C_i)\| + \|Term(G_j)\|}$$

az  $A(C_i, G_j)$  függvény összehasonlítja a  $C_i$  és  $G_j$  összetett entitás

attribútumait.  $A(C_i, G_j)$  a következőképpen számítható:

$$A(C_i, G_j) = \frac{\sum_{i,j} C(a_i, a_j)}{\|Attr(C_i)\| + \|Attr(G_j)\|}$$

Részleteket lásd a disszertáció I.2 fejezetében.

Kapcsolódó közlemények: [L2], [K3]

**I.2. Analizáltam a leggyakrabban használt hasonlóság-elemző algoritmusokat. Kimutattam, hogy a hatékonyságuk a konfigurációs beállításaik és a tesztdataik együttes függvénye. Összehasonlításukhoz objektív módszert dolgoztam ki.**

Megmutattam, hogy az adatsémák hasonlóság-elemző algoritmusai hatékonyság szempontjából alapvetően érzékenyek mind a teszsémára mind a működésüket befolyásoló különféle konfigurációs beállításokra. Ezért az objektív összehasonlításukhoz új eljárásra van szükség.

A hatékonyság mérésére az irodalomban használt és általánosan elfogadott mérési módszerek és mértékek továbbra is használhatók. A mértékek: a pontosság, a felidézés és az F-mérték<sup>1</sup>. Az új eljárás lényege, hogy először a hasonlóság-elemző algoritmusok parciálisan optimális paramétereit kell meghatározni, és ezekkel a paraméterekkel kell kiszámolni a hatékonysági mértékeket. A minősítést végül az így nyert mértékek összehasonlításával kell elvégezni.

Az új eljárással megmutattam azt is, hogy az I.1-ben ismertetett NTA hasonlóság-elemző algoritmus közel olyan jó (vagy jobb) eredményeket szolgáltat, mint több más ismert algoritmus.

Az NTA algoritmus megtérülését (alkalmazhatóságát) egy általam definiált költségmodellben is igazoltam egy – 23 informatikában jártas testtalan részvételével elvégzett – kísérlet eredményeire építve.

A következő táblázat a pontosság egy mértékét (F-mérték) mutatja több megközelítésnél több teszsémára számítva különböző küszöbértékeknel:

Scenarios:			Company-customer			University-publication			Car-retailer		
Tre-NTA	Tre-Sim	Tre-WN	NTA	SF	WN	NTA	SF	WN	NTA	SF	WN
0,2	0,04	0,15	0,8	0,4	0,57	0,53	0,47	0,33	0,67	0,37	0,36
0,25	0,05	0,175	0,8	0,4	0,55	0,55	0,6	0,33	0,67	0,41	0,41
<b>0,3</b>	0,06	<b>0,2</b>	<b>0,89</b>	0,43	<b>0,86</b>	0,5	0,47	0,33	0,67	0,42	0,43
0,35	<b>0,07</b>	0,225	0,33	0,43	0,4	0,5	<b>0,58</b>	0,34	0,8	0,42	0,4
<b>0,4</b>	0,08	0,25	0,4	0,45	0,4	0,53	0,43	0,32	<b>0,86</b>	0,53	0,47
<b>0,45</b>	0,09	<b>0,275</b>	0,4	0,35	0	0,53	0,43	0,24	<b>0,86</b>	0,5	<b>0,5</b>
<b>0,5</b>	<b>0,1</b>	0,3	0,4	0,38	0	<b>0,55</b>	0,31	0,3	0,77	<b>0,55</b>	0,46
0,55	<b>0,11</b>	<b>0,325</b>	0,4	0,4	0	0,53	0,31	<b>0,4</b>	0,67	<b>0,55</b>	0,5
0,6	0,12	0,35	0	0,46	0	0,43	0,31	0,31	0,45	0,55	0,22
0,65	<b>0,13</b>	0,375	0	<b>0,55</b>	0	0,31	0,31	0,31	0,25	0,55	0,22
0,7	<b>0,14</b>	0,4	0	<b>0,55</b>	0	0,31	0,31	0,31	0,25	0,25	0,22
0,75	<b>0,15</b>	0,425	0	<b>0,55</b>	0	0,31	0,17	0,31	0,25	0,25	0,22
0,85	0,17	0,475	0	0,44	0	0,31	0	0,17	0	0,25	0,22

Részleteket lásd a disszertáció I.3. fejezetében.

Kapcsolódó közlemények: [K1], [K3]

<sup>1</sup> Összevont jósági mérték, definíció szerint a pontosság és a felidézés harmonikus közepe

### I.3. Analitikus eljárást dolgoztam ki hasonlóság-elemző algoritmusok számítási igényének összehasonlítására

Analitikus eljárást dolgoztam ki különböző hasonlóság-elemző algoritmusok komplexitásának – azaz számítási igényének – összehasonlítására. A komplexitás számszerű értékét a végrehajtás során szükséges lépések számával közelítettem, melyet konkrét mintapéldák esetén meghatározva a különféle megközelítések abszolút, és egymáshoz képesti bonyolultsága is meghatározható. A modell helyességét példa implementációval és futásidő-méréssel számos tesztdaton igazoltam.

A módszer segítségével bizonyítottam, hogy az NTA algoritmus számítási igénye a bemeneti sémák komplexitásától függően 1-2 nagyságrenddel kisebb, mint több más algoritmusé.

A számítási lépésszámban jelentkező nagyságrendi különbségeket konkrét, implementált sémák illesztésének végrehajtásával gyakorlati mérésekkel is igazoltam.

Az NTA algoritmikus számítási komplexitását a következő összefüggés adja:

$$\begin{aligned} \text{Steps}_{\text{NTA}}(C, G) &= \text{Steps}_{\text{NTA\_Perform}}(C, G) + \text{Steps}_{\text{NTA\_Prepar}}(C, G) = \\ &= \sum_{i,j} 1 + (\| \text{Term}(C_i) \| + 1) * (\| \text{Term}(G_j) \| + 1) + \| \text{Attr}(C_i) \| * \| \text{Attr}(G_j) \| + \\ &+ \sum_i \| \text{Attr}(C_i) \| + \sum_j \| \text{Attr}(G_j) \| + \| C \| + \| G \| + \| C \| * \| \text{Term}(C_i) \| + \| G \| * \| \text{Term}(G_j) \| \end{aligned}$$

Az elméleti bonyolultságot alátámasztó futásidőmérések eredményeit a következő táblázat mutatja:

Típus	NTA	Sim. Flood	WordNet
Sample vs Sample	0,03	0,06	0,08
Example1 vs Sample	0,03	0,06	0,08
Example1 vs Example1	0,03	0,08	0,11
Example2 vs Sample	0,08	0,63	1,00
Example1 vs Example2	0,08	1,11	1,95
Example2 vs Example2	0,92	25,94	98,23
Example3 vs Sample	3,56	41,91	101,88
Example1 vs Example3	2,52	406,48	159,06
Example2 vs Example3	40,52	N/A	N/A
Example3 vs Example3	2264,84	N/A	N/A

Részleteket lásd a disszertáció I.4 fejezetében.

Kapcsolódó közlemények: [K2]

## II. téziscsoport

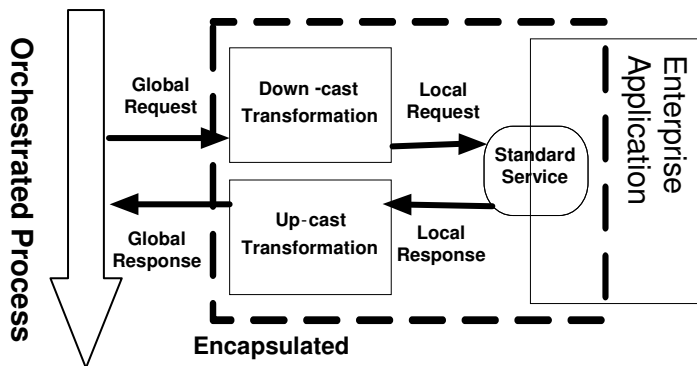
### Szolgáltatás alapú architektúrák és azok teljesítmény-analízise

#### II.1. Nyílt szabványokon alapuló megoldást dolgoztam ki vállalati alkalmazások SOA alapú integrációjában előforduló adatrepresentációs eltérések kezelésére

Nyílt szabványokon alapuló módszertant dolgoztam ki az adatrepresentációs különbségeket áthidaló transzformációk megvalósítására összetett folyamatokban. A megoldás lehetővé teszi meglévő vállalati szolgáltatások közvetlen használatát SOA alapú alkalmazás-integrációban. A javasolt megközelítés követi a modularitás, az újrafelhasználhatóság elveit, valamint nem igényli a vállalati standard szolgáltatások és azok interfészeinek módosítását. Kizárólag csak elterjedt nyílt szabványokat és azokhoz kapcsolódó eszközöket használ.

A módszer gyakorlati alkalmazhatóságát működőképes architektúra építésével igazoltam.

A megoldás lényegét jelentő becsomagolt szolgáltatást és annak meghívási folyamatát a következő ábra mutatja:



Részleteket lásd a disszertáció II.1 fejezetében.

Kapcsolódó közlemények: [L3], [K4], [K5], [K6], [K7], [K8], [K9]

## II.2. Új eljárást dolgoztam ki vállalati alkalmazások SOA alapú integrációs megoldásainak teljesítmény-analízisére

Elméleti és gyakorlati eljárást dolgoztam ki az adatrepresentációs különbségeket áthidaló transzformációkat is tartalmazó SOA alapú vállalati alkalmazás-integrációs megoldások teljesítmény-analízisére. A megközelítés újszerűségét a transzformáció végrehajtásához, a folyamat példányosításához, és a standard szolgáltatások futtatásához szükséges erőforrásigény elkülönített kezelése adja. Az új eljárás szerint a tetszőleges összetett folyamat erőforrás-szükségletét külön kell mérni a vállalati információs rendszereken és a folyamatpéldányokat futtató kiszolgálón. Ezen kívül meg kell határozni az adott konfigurációk áteresztő képességét és átfutási idejét megfelelő referencia-mérésekkel. Az összetett szolgáltatások várható válaszidejét és az időegység alatt végrehajtható folyamatpéldányok számát ezután már számítani lehet.

A módszer helyességét általánosítható tesztesetek mérési eredményeivel igazoltam a gyakorlatban.

Az általam javasolt megoldás válaszideje egy tipikus terhelési tartományban az alábbi összefüggéssel közelíthető:

$$\sum_{\text{path}} T_{\text{service}} + \sum_{\text{path}} T_{\text{add}} + T_{\text{constructs}} < T_{\text{response}} < \left( \sum_{\text{path}} T_{\text{service}} + \sum_{\text{path}} T_{\text{add}} \right) * \frac{\sum R_{\text{orchest}}}{R_{\text{avail\_enterprise}}} + T_{\text{bal\_constructs}} * \frac{\sum R_{\text{constructs}}}{R_{\text{avail\_process}}}$$

ha  $R_{\text{orchest}} \ll \sum_{n\_conc} R_{\text{orchest}} \leq R_{\text{avail\_enterprise}}$  és  $\sum_{n\_conc} R_{\text{constructs}} > R_{\text{avail\_process}}$

A folyamatszervert áteresztőképességét a következő összefüggések jellemzik:

$$Tp_{\text{process}} = Tp_{\text{process\_max}} * \frac{\sum R_{\text{process}}}{R_{\text{avail}}} \quad \text{ha} \quad \sum_{m\_conc} R_{\text{process}} \leq R_{\text{avail}}$$

$$Tp_{\text{process}} = Tp_{\text{process\_max}} \quad \text{ha} \quad \sum_{m\_conc} R_{\text{process}} > R_{\text{avail}}$$

$$Tp_{\text{process}} < Tp_{\text{process\_max}} \quad \text{ha} \quad \sum_{m\_conc} R_{\text{process}} \gg R_{\text{avail}}$$

Részleteket lásd a disszertáció II.2 és II.3 fejezeteiben.

Kapcsolódó közlemények: [L1], [K4], [K5]

## **Az eredmények alkalmazhatósága**

Az I.1 tézisben bemutatott séma-illesztési módszertan alkalmas adatsémák összehasonlítására és az egyes entítások közötti tartalmi összefüggések erősségének meghatározására produktív vállalati környezetben is. Az általam készített példa implementáció a módszertan és algoritmusok disszertációban található részletes leírása alapján kiváltható tetszőleges vállalati rendszer saját programozási környezetében elkészített implementációval is. Így alkalmas lehet tetszőleges produktív rendszerek entitásainak összehasonlítására vállalati rendszerek együttműködésének (pl. integrációjának) előkészítése során.

Az I.2 tézisben bemutatott megoldás használható tetszőleges séma-illesztési eljárás pontosságának megállapítására és különféle megoldások összehasonlítására. Segítségével megbecsülhetjük az algoritmusok várható pontosságát illetve testreszabhatjuk (optimalizálhatjuk) az egyes algoritmusokat éles alkalmazások adatsémáinak függvényében.

Az I.3 tézis eredményei alapján megbecsülhetjük különféle séma illesztési módszerek számításigényét a bemenet komplexitásának függvényében. Ez teszi lehetővé a séma illesztési megoldások hardver környezetének pontos méretezését, mely produktív rendszerek esetén a használatbavétel elengedhetetlen előfeltétele.

Az általam tervezett és implementált szolgáltatás alapú integrációs megoldás már jelen formájában is alkalmas kisebb vállalkozások rendszereinek integrálására. A II.1-es tézishez kapcsolódó keretrendszer eszközeinek hatékonyabb implementációja után nagyobb rendszerek közötti kollaborációs folyamatok futtatására is alkalmas lehet. Továbbá a disszertációban részletesen leírt módszertan és architektúra megvalósítható stabil ipari szoftver-eszközökkel is, mely lehetővé teszi az adatrepresentációs különbségek áthidalását illetve tetszőleges komplexitású SOA integrációs feladat megbízható megoldását.

A II.2 tézisben mutatott megoldás lehetővé teszi vállalati rendszerek integrációjának teljesítmény-analízisét. Az éles alkalmazásba vételt és a méretezést egyrészt az elméleti analitikus erőforrásigény számítása másrészt a nem-funkcionális teljesítményparaméterek referencia-méréseken alapuló becslése biztosítja. A bemutatott módszer alapján ezek mindegyike elvégezhető produktív, szolgáltatás alapú vállalati környezetben.

## **Továbbfejlesztési lehetőségek**

A tématerülethez több szorosabban és lazábban kapcsolódó további kutatási lehetőség is adott:

- adott sémák adott entitási közötti kapcsolat meghatározása után lehetőség van az illesztés implementációját jelentő adat-transzformációk készítésének támogatására (előkészítésére) is egyes speciális esetekben. Ez kiválthatja, illetve nagyban megkönnyítheti a jelenleg főként manuálisan elvégzendő transzformáció készítés feladatát.

- az általam ismertetett séma-illesztési módszer futásidejű teljesítménye és skálázhatósága is tovább javítható az egyes bemenetek lokális strukturális és tartalmi sajátosságainak figyelembevételével – például különböző absztrakciós szinten lévő entitások összehasonlításának kihagyásával a sémák fájnak bejárása során.

- a bemutatott teljesítményanalízisre szolgáló megoldás kiterjeszthető speciális klaszter-rendszerek esetére is, ahol az egyes homogén erőforrások felhasználásának monitorozása és átcsoportosítása további futásidejű optimalizálást eredményezne részletesebb teljesítményanalízis után.

## Tézispontokhoz kapcsolódó tudományos közlemények

### Magyarországon megjelent idegen nyelvű folyóiratcikk:

- [L1] **Martinek P.**, Szikora B.: SOA based Web service adaptation in Enterprise Application Integration, Periodica Polytechnica, 2009 - közlésre elfogadva
- [L2] **Martinek P.**, Szikora B.: Detecting semantically related concepts in a SOA integration scenario, Periodica Polytechnica, 2008 - közlésre elfogadva

### Külföldön megjelent idegen nyelvű folyóiratcikk:

- [L3] **Martinek P.**, Tóthfalussy B., Szikora B.: Implementation of Semantic Services in Enterprise Application Integration, WSEAS Transactions on computers, Vol. 7, Issue 10, 2008, pp. 1658 - 1668.

### Nemzetközi konferencia-kiadványban megjelent idegen nyelvű előadás:

- [K1] **Martinek P.**, Szikora B.: Computational complexity of schema matching approaches, 11th International Conference on Mathematical Methods, Computational Techniques and Intelligent Systems (Mamectis '09), Teneriffe, Spain, 2009, pp. 241-246.
- [K2] **Martinek P.**, Villányi B., Szikora B.: Optimization and comparison of schema matching solutions, 11th International Conference on Mathematical Methods, Computational Techniques and Intelligent Systems (Mamectis '09), Teneriffe, Spain, 2009, pp. 258-263.
- [K3] Villányi B., **Martinek P.**: Analyzing of Schema Matching Solutions, MicroCAD2009, Miskolc, Hungary, 2009, pp. 127-132.
- [K4] **Martinek P.**, Tóthfalussy B., Szikora B.: Execution of Semantic Services in Enterprise Application Integration, 12th WSEAS international conference on Computers, Heraklion, Görögország, 2008, pp. 128-134.
- [K5] **Martinek P.**, Tóthfalussy B., Szikora B.: Execution of semantically enriched business processes, ICE2008, Lisbon, Portugal, 2008, pp. 717-724.
- [K6] **Martinek P.**, Tóthfalussy B., Szikora B.: Semantically Described Services in the Enterprise Application Integration, ISSE 2007, Cluj-Napoca, Romania, 2007. május, pp. 335-338.
- [K7] **Martinek P.**, Szikora B.: Semantic Execution of BPEL processes, ISD 2006, Budapest, Hungary, 2006, pp. 361-367.
- [K8] **Martinek P.**, Kerekes J., Szikora B., Gál L.: Semantic Description and integration of Business processes, IMS 2006, Sakarya, Turkey, 2006. május, pp. 1002-1008.

- [K9] **Martinek P.**, Kerekes J., Szikora B.: Semantically-enriched Service-Oriented Business Applications, 29th International Spring Seminar on Electronics Technology (ISSE 2006), St. Marienthal, Germany, 2006. május, pp. 440-443.

## **Egyéb közlemények**

### Magyarországon megjelent magyar nyelvű folyóiratcikk:

- [E1] **Martinek P.**: Szoftverfejlesztés támogatása ARIS-szal, Elektronikai Technológia, Mikrotechnika, 2005/1, pp. 14-16.

### Nemzetközi konferencia-kiadványban megjelent idegen nyelvű előadás:

- [E2] Herczeg G., **Martinek P.**: Real-time scheduling of ambulances, MicroCAD2009, Miskolc, 2009, pp. 37-42.
- [E3] **Martinek P.**, Szikora B.: Automatic transformation generation in service oriented architecture, ISSE2008, Budapest, 2008, pp. 449-454
- [E4] Kiss Á., **Martinek P.**, Czilik I.: Semantic integration of HR offices, ISSE2008, Budapest, 2008. május 7-11, pp. 443-448.
- [E5] **Martinek P.**, Kiss G., Nagy Cs., Szikora B.: System Modelling and Simulation in Manufacturing Processes, IMS 2006, Sakarya, Turkey, 2006, pp. 994-1001
- [E6] Sviszt O., **Martinek P.**, Szikora B.: Typical features of PCB Production ERP Systems, 28th International Spring Seminar on Electronics Technology (ISSE 2005), Wiener Neustadt, Austria, 2005, pp.284-287
- [E7] **Martinek P.**, Szikora B.: Integrated enterprise resource planning systems, 28th International Spring Seminar on Electronics Technology (ISSE 2005), Wiener Neustadt, Austria, 2005, pp. 417-421
- [E8] **Martinek P.**: Supporting the printed circuit board (PCB) production at BUTE-ETT with a computer aided system, 10th International Symposium for Design and Technology of Electronic Modules, Bucarest, Romania, 2004, pp. 17-20
- [E9] **Martinek P.**: "Quality management and workflow control in printed circuit board production at the TU Budapest" - 27th International Spring Seminar on Electronics Technology (ISSE 2004), Sofia, Bulgaria, 2004. pp. 272-275

### Elektronikus formában közzétett egyetemi jegyzet:

- [E10] Szikora B., **Martinek P.**, Tertsch Á., Szalai-Dobos A., Szitás Z.: Vállalatirányítási Rendszerek Laboratórium, Laboratóriumi Segédlet, 2007.

## Hivatkozott irodalom

- [1] K. Votis, C. Alexakos, B. Vassiliadis, S. Likothanassis, An ontologically principled service-oriented architecture for managing distributed e-government nodes, *Journal of Network and Computer Applications*, Vol. 31, Issue 2, 2008, pp. 131-148.
- [2] S. Lamparter, A. Ankolekar, D. Oberle, R. Studer, C. Weinhardt, Semantic specification and evaluation of bids in web-based markets, *Electronic Commerce Research and Applications*, Vol. 7, Issue 3, 2008, pp. 313-329.
- [3] S. Izza, L. Vincent, P. Burlat, Exploiting semantic web services in achieving flexible application integration in the microelectronics field, *Computers in Industry*, Vol. 59, Issue 7, 2008, pp. 722-740.
- [4] G. Fenza, V. Loia, S. Senatore, A hybrid approach to semantic web services matchmaking, *International Journal of Approximate Reasoning*, Vol. 48, Issue 3, 2008, pp. 808-823.
- [5] G. Grossmann, Y. Ren, M. Schrefl, M. Stumptner, Behavior Based Integration of Composite Business Processes, *Lecture Notes in Computer Science*, Vol. 3649, 2005, pp. 186-204.
- [6] C. Pautasso, T. Heinis, G. Alonso, Autonomic resource provisioning for software business processes, *Information and Software Technology*, Vol. 49, Issue 1, 2007, pp. 65-80.
- [7] O. Moser, F. Rosenberg, S. Dustdar, Non-Intrusive Monitoring and Service Adaptation for WS-BPEL, *Proceeding of the 17th international conference on World Wide Web*, 2008, pp. 815-824.
- [8] H. Quyet Thang, V. Sy Nam, XML Schema Automatic Matching Solution, *International Journal of Computer Systems Science and Engineering*, Vol 4., Num. 1, pp. 68-74.
- [9] S. Melnik, H. Garcia-Molina, E. Rahm, Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching, *Proceedings of the 18th International Conference on Data Engineering*, 2002, pp. 117-128.
- [10] J. Madhavan, P. A. Bernstein, E. Rahm, Generic Schema Matching with Cupid, *Proceedings of the 27th International Conference on Very Large Data Bases*, 2001, pp. 49-58.
- [11] Aida Boukottaya, Christine Vanoirbeek, Schema Matching for Transforming Structured Documents, *Proceedings of the 2005 ACM symposium on Document engineering*, 2005, pp. 101-110.
- [12] Khalid Saleem, Zohra Bellahsene, Ela Hunt, Performance Oriented Schema Matching, *Lecture Notes in Computer Science*, Vol. 4653, 2007, pp.844-853.

- [13] Hong-Hai Do, Erhard Rahm, Matching large schemas: Approaches and evaluation, Information Systems, Vol. 32, Issue 6, 2007, pp. 857-885.
- [14] Nalaka Gooneratne, Zahir Tari, Matching Independent Global Constraints for Composite Web Services, Proceeding of the 17th international conference on World Wide Web, 2008, pp. 765-774.
- [15] Jun Shen, Yun Yang, Jun Yan, A p2p based service flow system with advanced ontology-based service profiles, Advanced Engineering Informatics, Vol. 21, Issue 2, 2007, pp. 221-229.
- [16] O. Moser, F. Rosenberg, S. Dustdar, Non-Intrusive Monitoring and Service Adaptation for WS-BPEL, Proceeding of the 17th international conference on World Wide Web, 2008, pp. 815-824.
- [17] N. Sato and K.S. Trivedi, Stochastic Modeling of CompositeWeb Services for Closed-Form Analysis of Their Performance and Reliability Bottlenecks, Lecture Notes In Computer Science, Volume 4749, 2007, pp. 107-118.
- [18] Open Applications Group (OAGIS), Oagis 9.4 Schemas, at <http://www.oagi.org/dnn/Default.aspx?alias=www.oagi.org/dnn/oagi>
- [19] XML Comman Business Library (XCBL) at <http://www.xcbl.org/>