



Az ERP rendszerek metamorfózisa
Doktori értekezés

Készítette: dr. Ternai Katalin

Budapest, 2008.

Budapesti Corvinus Egyetem
Gazdálkodástani Ph.D. program
Információrendszerek Tanszék

Az ERP rendszerek metamorfózisa
Doktori értekezés

Készítette: dr. Ternai Katalin

Témavezető: dr Gábor András

Budapest, 2008.

Tartalomjegyzék

I. A KUTATÁS ÁTTEKINTÉSE.....	4
I.1. Bevezetés	4
I.2. A kutatás célja, előzményei és főbb kérdései.....	6
I.3. A kutatási módszer	10
I.4. A kutatás jelentősége	12
I.5. A dolgozat felépítése	13
II. A KUTATÁS ELMÉLETI HÁTTERE.....	13
II.1. Az ERP rendszerek evolúciója.....	13
II.2 ERP rendszer a szervezetben	19
II.2.1. Vállalat – Irányítás.....	19
II.2.2. Integrált vállalatirányítási információrendszer	23
II.3. Az üzleti integráció	30
II.3.1. Az üzleti integráció szerkezete	30
II.3.2. Vállalati Integráció Technológiai Megoldásai.....	40
II.3.3. A Vállalati Integráció Folyamata.....	62
II.4 Új rendszerfejlesztési paradigmák	66
II. 4.1. Modell vezérelt fejlesztés	66
II. 4.2. SOA.....	74
II. 5. A szemantikus üzleti folyamatmenedzsment (Semantic Business Process Management – SBPM)	113
II.6 Az ARIS módszertan.....	132
II.6.1. Az ARIS koncepció háttere	133
II.6.2. Az ARIS koncepció alapelvei.....	134
II.6.3. Az ARIS nézetei és a tipikus modellek	135
II.6.4. Az ARIS leíró szintjei.....	144
II.6.5. Az ARIS architektúra	145
II.6.6. Az üzleti folyamatok modellezésétől az üzleti folyamatok menedzsmentjéig.....	146
II.6.7. ARIS HOBE – Az üzleti tervezés háza.....	148
II.6.8. Az ARIS for SOA.....	153
III. A KUTATÁS GYAKORLATI MEGVALÓSÍTÁSA	162
III.1 A Fő hipotézis.....	165
III.1.1. Az Első segédhipotézis és igazolása.....	165
III.1.2. A Második segédhipotézis és igazolása.....	180
III.1.3. A Harmadik segédhipotézis és igazolása.....	197
III.2. A kutatás folytatása.....	219
IV. ÖSSZEFOGLALÁS	221
V. IRODALOMJEGYZÉK.....	230

I. A KUTATÁS ÁTTEKINTÉSE

I.1. Bevezetés

A világgazdaság, a modern piacgazdaságok az ipusztindusztriális, majd a posztindusztriális társadalmakat követően új fejlődési korszakba léptek. Az információgazdaság dinamikus ágaiban a gazdasági folyamatok tér- és idő-koordinátái alapvetően megváltoznak „a tér és az idő összezsugorodik, a reakciók mind nagyobb része valós idejűvé válik, a gazdaság különböző ciklusai egymásba csúsznak, az idő pedig felpörög és folytonossá válik”. [Szabó–Hámori, 2006] A forradalmi nevezhető változások jelentős hatással vannak a modern piacgazdaságok alapegységeire – a vállalatokra, illetve a korszerű vállalatok működését támogató integrált informatikai rendszerekre.

Az új évezred gazdasági környezetét az egyre inkább kiélezetté váló verseny, és az ebben való sikeres helytállásra való törekvés jellemzi. A gyors technológiaváltási kényszer, az egyre inkább rövidülő termék életciklusok, a gyorsuló termékfejlesztések, az ár, minőség és a szolgáltatások tekintetében növekvő vásárlói igények egyre nagyobb kihívásokat jelentenek a kiéleződött versenyben.

A globalizáció hatására a piaci verseny egyre inkább világméretűvé válik, a nagyvállalatok versenystratégiái átlépik a nemzeti piac adta lehetőségeket, és kiterjesztik a piaci versenyt az egész fejlett világra.

Magyarországnak is meg kell küzdenie a rohamosan változó modern piacgazdaság kihívásaival, miközben valójában még tart a piacgazdaságba való átmenet és meg kell felelni az EU csatlakozáshoz szükséges követelményeknek is. A hazai szervezetekre nézve ez azt jelentheti, hogy fejlődésük során esetleg egy-két lépcsőfokot át kell ugraniuk.

A globális versenyben sikeres vállalatok tartós, több forrásból származó versenyelőnyökre törekuszenek, melynek alapja az innovációs képességben rejlik. A

tudásteremtés, a kutatás-fejlesztés, az innováció, és egyéb nem anyagi erőforrások ebből adódóan egyre dominánsabb szerephez jutnak a vállalati versenyképességet befolyásoló tényezők között. Az innovációk, az új technológiák gyors bevezetése versenyelőnyt jelent. Az innovációt egyre inkább a modern gazdaságok versenyelőnyeinek egyik legfontosabb forrásaként tarthatjuk számon. A tudás menedzselésének képessége meghatározza mind a vállalatok, mind pedig a térségek innovációs lehetőségeit, és ezen keresztül versenyképességét is.

Az információs rendszer szolgáltatja az inputot a tervezési és ellenőrzési rendszerhez, emiatt szorosan összekapcsolódik vele. Az irányíthatóság érdekében bizonyos információkat időben, a megfelelő mennyiségben és minőségben a lehető legköltséghatékonyabb formában a döntéshozók rendelkezésére kell bocsátania. Mind a vállalati célokat, mind a célok elérését egyre inkább meghatározza az, hogy a vállalat hogyan viszonyul a technológiai környezethez.

A számítógépek fejlődésével párhuzamosan a rájuk épülő technológiák, és ezek alkalmazásai rendkívül gyorsan változnak. Igaz ez a vállalati információs rendszerekre is, amelyek a lehetőségek tágulásával egyre összetettebb feladatok ellátására képesek. Ennek egyik oldala technológiai meghatározottságú, egyszerűen annak következménye, hogy olyan feladatokat is el tudnak látni, amelyeket korábban a rendelkezésre álló technológia nem tett lehetővé. Másrésztől legalább ilyen fontos, hogy az érintettek felismerjék az ezekben az új megoldásokban rejlő lehetőségeket, a vállalatok új üzleti modelleket alkalmazzanak. Egy ilyen típusú átalakulás teszi lehetővé, hogy a technológiai fejlődés által felkínált lehetőségeket a cégek valóban kiaknázzák.

Az egyik legszembetűnőbb jelenség, hogy a különböző információtechnológiai szolgáltatásokat moduláris logika szerint kezdik szemlélni; azaz a vállalati alkalmazásokat egymástól függetlenül is működő, mégis szorosan integrálható részekre bontják. Ennek a felfogásnak az úttörői az ERP(Enterprise Resource Planning)-rendszerek, amelyek a vállalaton belüli integrációt többé-kevésbé megvalósították. Az utóbbi években azonban az integrációs igény a vállalati határokon túl is jelentkezett, és az ehhez szükséges technológiai feltételek már vagy rendelkezésre állnak, vagy

fejlesztés alatt vannak. Ez a globális integráció pedig olyan új üzleti modellek előtt nyitja meg az utat.

Az integrált vállalatirányítási rendszereknek kellene az újabb és újabb kihívásokkal megbírkózni - pl. amikor a szűkösen rendelkezésre álló, újratermelendő erőforrások közé belép a mai gazdaságot generáló, nem mérhető tudás. Új működési feltételek között kell értékteremtő módon támogatni az irányításban az új típusú innovatív vállalatokat. [Nonaka, 1995]

Ahhoz, hogy a különböző rendszerek integrálhatóak legyenek, két alapvető problémát kell megoldani: az alkalmazások képesek legyenek kommunikálni egymással, és megértsék egymás nyelvét. Ezeket a feladatokat pedig a korszerű platformok és a szabványosítás együtt is csak részben tudják ellátni.

A dolgozat rámutat arra, hogy az új területek megfelelő absztrakciós szintű üzleti modelljei hogyan segítik az új terület menedzselésének integrációját a szervezet meglévő működésébe, és az azt támogató informatikai alkalmazásrendszerébe.

I.2. A kutatás célja, előzményei és főbb kérdései

Az új gazdaságban új üzleti modellek, szerepek, funkciók, folyamatok alakulnak ki, illetve a régiek eltűnnek vagy gyökeresen megváltoznak. [Turban, 2002] A modellek vizsgálata során a jövőre nézve jelentős eredmények születhetnek - jó lenne megjósolni hogy például mely modellek lesznek/maradnak életben , vagy képesek leginkább az értékteremtésre. Elméleti és gyakorlati szempontból is hasznos az új modellek megvalósítása során kialakuló technológiák elemzése abból a célból, hogy milyen hatásokat, folyamatokat indíthatnak be a gazdaságban, illetve milyen szervezeti változásokat okozhatnak az információs társadalomban.

A dolgozat egyik célja feltárni, hogy milyen fejlődési modell vázolható fel az együttműködő vállalatok, vállalat-hálózatok sokrétű tevékenységeit segítő integrált vállalatirányítási rendszerek esetében.

Kutatásom további célja feltárni, hogy ezek az említett technológiai lehetőségek az új üzleti modellek elterjedését mennyiben segítik elő, illetve milyen átalakulás előtt áll a számítógépes alkalmazások piaca az összekapcsolhatóság egyre fejlettebb szintre kerülése miatt. Az alkalmazások külső forrásokból, modulszerűen történő igénybevételének lehetősége nem csak a szoftverpiac teljes átalakulását hozza magával, hanem azt is jelenti, hogy a gazdaság szinte minden szereplőjének át kell értékelnie az információtechnológiával kapcsolatos elgondolásait.

A kutatás során közelebbről vizsgálom, hogy melyek azok az információtechnológiai megoldások, amelyek a leghatékonyabban tudják támogatni a szervezetek működtetését, együttműködését, és milyen típusú ez a támogatás.

Mivel ez a kérdéskör rendkívül összetett, a vizsgálat körét szűkítettem információtechnológiai területen az alkalmazások integrációs problémáira. Ezzel a kutatási témakörrel több éve foglalkozom, részt vettem tanszékünk különböző, a témával kapcsolatos kutatási projektjeiben, amelyek közül a jelen dolgozat szempontjából a „BKE-SAP pilot-”, és a HEFOP projektek, valamint a jelenleg zajló IBM-IDS kutatási projekt kapcsolódnak szorosabban a dolgozathoz. A projektmunkák során a problémák, majd a célravezető megoldás leírásában fontos eszköznek bizonyult a modell alkotás, a terület folyamatainak megfelelően magas absztrakciós szintű leírása.

Több éve vezetek graduális és posztgraduális kurzusokat ezen a területen, ahol a hallgatókkal való együttműködés során is érdekes kutatási problémák merültek fel. Több félévében tartottam a kutatási témámhoz szorosabban kapcsolódó szakszemináriumot, ahol a hallgatókkal az elméleten túl gyakorlati megoldásokat is teszteltünk. Több olyan kutatási feladattal találkoztam, amelyek modellezése és leképezése jelentős kihívás volt a résztvevők számára.

Számos megközelítés, megoldás és módszertan ismert a szakirodalomból, amelyek között a választás nem mindig egyértelmű és a felhasználókat leggyakrabban csak saját

tapasztalataik irányítják. Amíg nincs egységes fogalomrendszer, amelyhez a felhasználók igazodhatnak, nehéz együttműködésről beszélni. Ez az egyik oka a közösen értelmezhető modell fejlesztése iránti igénynek.

A rendszerintegrációs projektek nagy száma, a rendszerek együttműködése iránti megnövekedett igény következtében felértékelődtek a modellezési tevékenységek is. Nagy szerepük van a rendszerek kialakításában, elsősorban az integrációt érintő feladatokban. Egy másik jelentős tényező a költséghatékony szoftverfejlesztés, amelyet az újrafelhasználhatóságon keresztül támogat a modellezés. A szakirodalomban említett számos előny közül kiemelném a következőket:

- az információmegosztásban játszott szerep,
- az újrafelhasználhatóság,
- integráció támogatása,
- a rendszerek karbantartásának és továbbfejlesztésének megkönnyítése,
- a verifikáció támogatása.

A feladat komplexitásához mérten a kutatás is összetett, egymástól elkülönülő, önmagában is érvényesülő részekből áll. Célja a probléma minél teljesebb körüljárása, majd a konkrét modell tanulmányozása további finomítások érdekében.

A dolgozatban a következő fontosabb kutatási kérdésekkel foglalkozom:

- Hogyan „maradhatnak életben” az ERP rendszerek a mai agilis vállalati környezetben
- milyen fejlesztési módszertanok léteznek a szakirodalomban;
- melyek a leggyakrabban hivatkozott BPM eljárások;
- hogyan modellezzünk egy eddig informatikailag nem támogatott területet annak érdekében, hogy azt a szervezetek integrálni tudják alkalmazásaikba;
- hogyan kellene és lehetne ezeket a módszertanokat kiterjeszteni, módosítani, hogy a dinamikusan belépő új területekre alkalmazhatóak legyenek,

A dolgozatban tárgyalt kutatási kérdésekhez tartozó fontosabb részfeladatok:

1. Külföldi és hazai tapasztalatok, saját kutatási tapasztalataim, esettanulmányok, statisztikák, megfigyelések valamint a szakirodalom eredményeit elemzem a modellezés, valamint a támogató technológiák szerepére vonatkozóan a szervezet menedzsmentjében. Ebben a részfeladatban áttekintem a kapcsolódó információtechnológiai fogalmakat és elméleteket, az integrációt támogató módszertanokat, valamint azokat a legismertebb megoldásokat is, amelyek az általam választott területeken felhasználhatóak. Ebben a részfeladatban igazolom a megközelítés létjogosultságát.

2. Foglalkozom a modellezési eljárások szerepével a feltérképezésben, megragadásban és leképezésben. Ez a vizsgálat kiterjed arra is, hogy az egyes részterületeken kialakított modellek között milyen átjárhatóság állapítható meg. A modellezés az integrált rendszer komponensek területre vonatkozik. Alapvető kérdés, hogy melyik az az absztrakciós szint, amelynek segítségével az eltérő területek közös vonásai megragadhatóak. A modell alapú megközelítés adhat támogatást olyan magas szintű leírásra (metamodellek készítésére), amelyek lehetővé teszik az egyes területeken történő újrafelhasználást. A dolgozatban követett megközelítés, problémakezelés egyediségét az alkalmazott paradigmarendszer és eszközkörnyezet biztosítja.

3. Központi kutatási kérdésem volt, hogy mely módszertan lesz a legalkalmasabb a probléma megoldásához, milyen fontos jellemzőkkel rendelkezik, hogyan alkalmazható az adott területeken és hogyan fejleszthető tovább (többek között milyen integrációs lehetőségei vannak).

További kutatási kérdésem a gyakorlati megvalósításhoz kötődik. Melyek a leggyakrabban használatos implementációs megoldások ezen a területén? Milyen követelményeket lehet megfogalmazni ezekkel a megoldásokkal szemben? Melyik illeszkedik ezek közül az általam megfogalmazott kutatási feladathoz? A kutatási kérdés igazolásaként áttekintést adok a gyakorlati megvalósítást támogató megoldásokról és kiválasztom a fejlesztésben felhasználható alkalmazásokat.

4. Megvizsgálom a kialakított modell felhasználási lehetőségeit a rendszerintegrációs folyamat támogatásában. A kutatás során a témakörre vonatkozó elméleti megközelítésekől indulok ki, a modellezési eljárások vizsgálatán keresztül eljutok az módszertan értékeléséig és a saját változat kialakításáig, amelyet a gyakorlatban is alkalmazok. A kutatási kérdések zárásaként megvizsgálom a kialakított modell további felhasználási lehetőségeit.

I.3. A kutatási módszer

A kiválasztott kutatási feladat az informatika és a társadalomtudományok határterületéhez tartozik, ez befolyással bír a módszertan kiválasztására.

A szervezetelemzési módszereknél megszokott kvantitatív és kvalitatív megközelítéseket vehetjük alapul [Balaton, Dobák 1991]. A kvantitatív módszerek olyan kutatások esetén vezetnek eredményre, amelyek nagy mennyiségű adatokra vonatkoznak, ezért matematikai-statisztikai megoldásokat alkalmaznak az adatfeldolgozásban. A statisztikai eszközöket tovább csoportosíthatjuk leíró és tényezők közötti kapcsolatok kimutatására hivatott eszközökre.

A hetvenes években előtérbe kerültek a kvalitatív módszerek (a kvantitatív módszerekkel kapcsolatos kételyek miatt), amelyek az összefüggések mélyebb feltárására törekuszenek. A módszer alkalmazása nem kiforrott, egységes fogalomrendszerrel még nem rendelkező területek esetén javasolt. Ezen megközelítés lehetőséget ad egyéni gondolkodásmódok megismerésére.

A kvantitatív és kvalitatív módszerek alkalmazásának kombinációja a módszertani trianguláció [Klimkó, 2001].

A trianguláció fajtái:

- kvantitatív módszereken belül többféle eljárás használata,
- kvalitatív módszereken belül többféle eljárás használata, illetve

- kvantitatív és kvalitatív módszerek kombinációja [Balaton, Dobák 1991].

Kutatásom elvégzésére kvalitatív módszerek alkalmazhatók.

Az induktív és deduktív módszer a kutatási terület és az elméletek viszonya alapján különböztethető meg. Az induktív módszer kutatási problémák feltárására és új elméletek alkotására törekszik. Ezzel szemben a deduktív módszerek elmélettesztelésre épülnek. A szakirodalom elemzését követően meghatározandóak a kutatás kulcsfogalmai, azonosítói, majd a változók közötti összefüggések, tételek. A témára vonatkozó következtetések mindezek mentén logikus okfejtéssel állnak elő. [Babbie, 1996]

Kutatási témám deduktív jellegű, az elméletek új területen való alkalmazását helyezi középpontba.

A kutatás céljára vonatkozóan gyakran hivatkozott megközelítés a Yin-féle, amely szerint a következő kategóriákat különböztetjük meg: felfedező, leíró és magyarázó.

Kutatási stratégiái a következők:

- kísérleten alapuló;
- kérdőíves felmérésen alapuló;
- másodlagos elemzésen alapuló;
- történeti elemzésen alapuló és
- esettanulmány feldolgozáson alapuló stratégia [Yin, 1994].

Yin és Eisenhardt szerint az esettanulmányok nem csak felfedező módon használhatók. Eisenhardt szerint az esettanulmányozás módszerét legalább három cél elérésére lehet felhasználni: illusztrációs céllal (az elmélet megvilágítására), elmélet konstruálására, és a már kifejlesztett elmélet tesztelésére is [Eisenhardt 1989].

A kvalitatív kutatások három csoportját különbözteti meg Miles és Huberman: az interpretivizmust, a társadalmi antropológiát és az együttműködő társadalomtudományi kutatást. [Miles, Huberman, 1994]

A téma jellegéből adódóan a részletes, mélyszintű, esettanulmányokon, a szakirodalom eredményein és a kutatási projektek tapasztalatain alapuló megközelítés vezet eredményre.

A fenti megközelítések közül a vázolt kutatási feladathoz a Yin és az Eisenhardt-féle megoldás áll a legközelebb.

A dolgozatban a források hivatkozásaként a Harvard-rendszert alkalmazom [Hart, 1988].

I.4. A kutatás jelentősége

A kutatás során a legújabb szakirodalmi eredmények feldolgozása és gyakorlati alkalmazása is megtörténik. A modellezés témakörének feldolgozása során irányadónak tekintem az Scheer-féle ARIS módszertant és megközelítést. Ez utóbbi új alapokra helyezi a rendszerek fejlesztését a módszertanból kiindulva.

A kutatás legfontosabb eredményei a módszertan alkalmazása a kiválasztott területre, a modell kialakítása, integrációs lehetőségek feltárása, prototípus kialakítására tett javaslatok az implementációs környezetben. Újszerűnek tartom a választott területen a modellalapú megközelítés alkalmazását.

Kutatásom további jelentősége, hogy az oktatás területén is felhasználható, illeszkedik tanszékünk oktatási és kutatási portfóliójába is. Oktatási alkalmazását egyrészt a graduális képzés kurzusaiban, másrészt a szakszemináriumi tevékenységekben, esetlegesen posztgraduális képzésben képzelem el. Megítélésem szerint az eredmények a vállalati gyakorlat számára is hasznosíthatók lesznek.

A dolgozatban kialakított modell gyakorlati hasznosítási lehetőségei:

- modellként funkcionálhat a vállalati gyakorlatban,

- továbbfejleszhető a területet teljes mélységében leíró, „testre szabott” prototípussá,
- konvertálható egyéb rendszerekbe, így támogatva az újrahasznosíthatóságot,
- alapja lehet a területre vonatkozó rendszerfejlesztési projekteknek,
- támogathatja a rendszerspecifikáció elkészítését,
- elősegítheti különböző granularitású architektúrák összehasonlíthatóságát.

A fentiek alapján az eredmények széles körűen hasznosíthatók a gyakorlatban. Az elméleti területen kiemelem azt, hogy a modell lehetővé teszi az alkalmazások szemantikus együttműködését, azaz a rendszerintegrációban játszott szerepét.

I.5. A dolgozat felépítése

A dolgozat a következő fő részekből áll. Az első fejezet foglalkozik a kutatás céljával és jelentőségével. A 2. fejezet a kutatás elméleti hátterét mutatja be, így kitérek a több tudományterülethez is kapcsolódó alapfogalmakra és elméletekre, a technológia szerepére a olyan új kutatási területre, mint a szemantikus web és folyamatmenedzsment, SOA és modellvezérelt architektúra, valamint a leggyakrabban használatos módszertanokra. Hipotéziseimet, kutatási kérdéseimet és azok igazolását a 3. fejezetben tárgyalom.

II. A KUTATÁS ELMÉLETI HÁTTERE

II.1. Az ERP rendszerek evolúciója

Viszonylag nem is telt el hosszú idő azóta, hogy a kockás füzeteket leváltották az informatikai rendszerek a vállalati működés szinte valamennyi területén. Az első füzetek a pénzügyi (bérszámfejtés, könyvelés), a termelésirányítási, majd a tervezési szervezeti egységeknél váltak feleslegessé a '60-as évek előtt. Az ERP rendszerek elődjeinek az úgynevezett MRP rendszereket tekinthetjük, amelyek lényegében a raktározás-menedzsment technológiai támogatásából fejlődtek egyre komplexebb alkalmazásokká. Az MRP jó módszer a raktározás menedzselésre, de nem veszi figyelembe a szervezet más forrásait.

1970-ben ez az igény, valamint az IT fejlődéséből eredő lehetőség (a több alkalmazást is kiszolgálni képes szervezeti adatbázisok, az általános file-kezelők) készítette az MRP logika megváltoztatását és létrejött ismert nevén a Zárt Lánccú MRP (Closed Loop MRP). Ez a technológia integrálva az ún CRP (Capacity Requirements Planning) egységgel a szervezet egy bizonyos termékre vonatkozó kapacitását is figyelembe veszi.

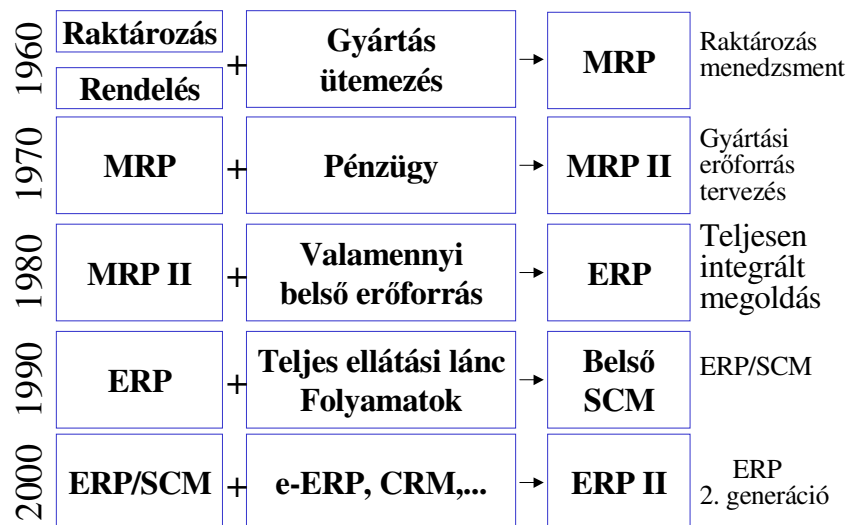
1980-ra érezhetővé vált a gyártási folyamat további erőforrásainak integrálása iránti szükséglet, amely életre hívta az összekapcsolt funkciók sokaságából álló ún. MRP II-t (Manufacturing Resources Planning). A Gyártási Erőforrás Tervezés ennek megfelelően a gyártó vállalat valamennyi erőforrásának hatékony tervezési módszere. Az MRP II rendszer a részrendszerekből származó outputokat már integrálja a pénzügyi riportokkal is.

Az MRP II hiányosságai (fix átfutási idő, végtelen kapacitás) és az újabb eszközök - CAD(Computer Aided Design), CAM (Computer Aided Manufacturing), CAL (Computer Aided Learning), stb. - iránt megjelenő igények vezettek a teljes integrált megoldás, az ERP kifejlesztéséhez a '90-es évekre.

Az elosztott rendszerek korszakában a személyi számítógépek használata egyre terjedt a szervezetekben [Holloway, 1990], egyre több új alkalmazási terület igényelte a számítógép támogatását az automatizálható munkafolyamatoknál, egyre újabb funkciók épültek be az ERP rendszerekbe. [Palaniswamy, 2000]

Az ezredfordulóra a piacvezető standard rendszerek a teljes ellátási lánc mentén integrálják a teendőket, miközben az üzleti eseményekre és ('best practice') folyamatokra fókuszálnak az üzleti funkciók és szervezeti egységek helyett.

A folyamatszempléttel együtt a folyamatok automatizált kezelése (workflow management) is bekerült a magukra valamit is adó ERP rendszerek alkalmazásai közé ugyanúgy, mint a többi intelligens irodai alkalmazás (dokumentum kezelés, csoportmunka és kommunikáció támogatás).



1. ábra Az integrált rendszerek evolúciója [Turban, McLean, Wetherbe 2002]

Egy ilyen komplexitású rendszer bevezetése [Davenport, 1998] (még ha fokozatosan, és nem az összes modult kiválasztva történik is) a szervezet számára több száz millió forintot, valamint a leggyorsabb bevezetési módszertanok alkalmazásával is több félét jelent [Koch, 1999]. Bármilyen bőséges funkcionalitással is rendelkezik egy tranzakciófeldolgozó (OLTP) [Claybrook, 1992] rendszer, használata ma már versenyelőnyt nemigen jelent [Lederer, 1998], a piacon maradás szükséges feltétele lett.

Így az ERP rendszerek tovább bővülnek az üzleti intelligencia [Peppers, 1999] értéknövelő eszközeivel (OLAP, adattárház, szakértő rendszerek, döntéstámogató rendszerek, adatbányászati eszközök, CRM [Gray, 2001], stb.), amelyek egyrészt a különböző szintű vezetői döntések hatásosságát növelik a komplex döntési szituációk megoldásában, illetve korábban ismeretlen, nem triviális ismereteket, összefüggéseket tárnak fel az üzleti előnyök kihasználása céljából. [Dutta, 1997] Az üzleti intelligencia ezen, valamint a vevőt (CRM) és a kiterjesztett értékláncot (SCM) középpontba helyező újjelvű informatikai eszközei önmagukban is elérhetik az ERP rendszerek méretét, bonyolultságát, adattartalmuk pedig az OLTP rendszerek adatbázisának akár többszöröse lehet.

Az új gazdaságban a tömegtermelést felváltja a “tömeges testreszabott termelés” (mass customization). [Pine, 1999] Az egyedi, minőségi igények színvonalas kielégítésére képtelenek a hagyományos, hiererchikusan felépülő szervezetek. A kisebb, rugalmas (stabil adatszerkezet, dinamikusan változó folyamatok) szervezetek életképesebbek lesznek. A rugalmasság nem csak szervezeten belül, de a szervezetek között is szükségszerű. Hálózatok, “amőba szervezetek” alakulnak. [Leary 1997] Hogyan élik túl (túlélnek?) ezt az alapvető változást a legtöbb nagyvállalat által mára már hatalmas összegekért megvásárolt és bevezetett ERP rendszerek? Hogyan tudják megvásárolni a kisebb szervezetek (KKV) az elektronikus kereskedelem gerincét jelentő standard vállalatirányítási rendszereket?

A vállalatokon átívelő üzleti folyamatok korszerű automatizálása rugalmas, együttműködő feldolgozási elemeket, hálózati üzleti objektumokat igényel, mivel az internetet már ‘meghódították’ a kód újrafelhasználását és a karbantartást jelentősen megkönnyítő objektumorientált alkalmazások [Booch, 1991] [Coad, 1991] [Chandra, 2000] [Johnson, 2000]. Az objektumokra épülő komponens-alapú alkalmazástervezés a modern szoftverfejlesztés egyik legfontosabb paradigmája. A dobozolt jellegű szoftverkomponensek (COTS – Commercial Off-The-Shelf) alkalmazása a szervezetek számára gazdasági és stratégiai jelentőséggel bír. A technológiai újdonságot a funkciók szerint könyvtárakba rendezett komponensek bináris újrafelhasználhatósága jelenti. A gazdasági jelentőségét pedig az ipari méretű szoftverfejlesztés lehetőségének megteremtése szolgálja.

Az új kihívásokra gondolva (egyesekek kicsit késve, a tőzsdei árfolyamcsökkenést észlelve) az ERP-termékek szállítói belekezdtek a szoftvercsomagok komponens technológiát (DCOM, CORBA) alkalmazó átdolgozásába. Természetesen a kliens-szerver alapú, struktúrált programnyelven megírt szoftver-kolosszusok átírása már csak a kompatibilitás miatt is megoldhatatlan, ezért a fejlesztők inkább objektum, illetve komponensként kezelhető kódba csomagolták a rendszer részeket.

A komponensek és a szabványos interfészek segítségével tetszőlegesen építhetők össze akár különböző forrásokból származó alkalmazások, valamint lépésenként egymásra épülve vezethetők be az újabb és újabb megoldások. Az ERP rendszerek szolgáltatásai az interfészekon keresztül 'kínált' objektumok által kívülről elérhetővé válnak más alkalmazások számára is.

Az üzleti komponensek az üzleti alapfogalmakat reprezentáló üzleti objektumokon alapulnak. Az üzleti kommunikáció legyen vállalaton belül, vagy vállalatok között tehát megoldható az üzleti folyamatban résztvevő akár különböző rendszerekből származó objektumok együttműködésével.

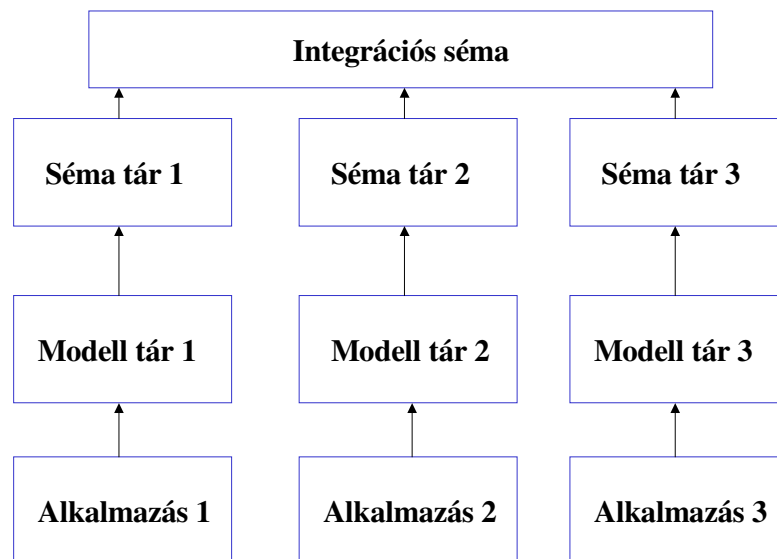
A jövő tehát a különböző fejlesztőktől származó, de egymással együttműködő elemekből felépülő vállalati rendszereké. A Service Oriented Architecture (SOA) technológia lehetővé teszi, hogy a vállalatok heterogén üzletiszoftver-környezetet működtethessenek házon belül és a partnerekkel való kapcsolataikban is.

A technológia adott, a komponensekhez kialakított szabványos interfészekkel a kommunikációs technológia anélkül fejlődhet, hogy az üzleti folyamatokon változtatni kellene. A „hogyan” mellett azonban legalább ilyen fontos a „mit” kérdése, azaz a különféle rendszerek közötti üzleti tartalom megfeleltetése.

A műszaki szabványosítás ellenére a felhasználói igények heterogenitása folyamatosan csökkenti a rendszerek szemantikai integrálhatóságát. A különböző (rész)rendszerek tartalmi összehangolása legtöbbször intuitív alapon zajlik, valamint az integrációs eljárás gyakran nem eléggé részletekbe menő és dokumentált. A szemantikai szintű integrálás a felhasználók mentális nézőpontjának megértését és ábrázolását igényli. A magas minőségű integráció létrehozását, az alkalmazások különböző szintű és jellegű modelljeiből előállított meta-adatok nagymértékben segíthetik. A szükséges ismeretek deklaratív specifikációja rendelkezésre állhat akár objektum orientált modellek, vagy például ontológia formájában. [Ushold, 1997] [McCarthy 1999, 2000] Az ontológia egy

szakterület közös értelmezésének a megjelenése, amely tartalmazza a szakkifejezéseket, terminológiákat és a szemantikát. [Uschold, 1996]A tartalmi integráció megvalósításához a technológia is létezik - a tudásmenedzsment területeken alkalmazott ismeretalapú rendszerek formájában. [Schreiber, 1998]

Az intelligens integrációs szolgáltatások megvalósításához a komponens könyvtárak mintájára ún. séma könyvtárak tartalmazhatnák a integrációs források sajátosságait. Az integráció segítésére alkalmas „sématár” egy nagyon alaposan kidolgozott területi modellen alapulhat, amelyet egy ismeretbázisban is lehetne rögzíteni a témaköröktől (amivel foglalkoznak) kezdve egészen a fizikai tulajdonságaikig. A rendszerintegrációs folyamat így séma integrációval indulhat, amelyben az input a rendszer források séma leírása, a kimenet pedig az egyeztetett, a célnak megfelelő leírása az összes bemeneti sémának (meta-séma). Az így létrehozott meta leírás annak a specifikációját is tartalmazza, hogy a bemeneti sémák hogyan felelnek meg a kimeneti séma bizonyos részeinek.



2. ábra Séma integráció

Az elképzelt sématarak architektúrájának fogalmi perspektívája az adott terület modellje (vállalati, ill. egyéb rendszerek, részrendszerek, komponensek modelljei), amely az integrációs források fogalmi leírásáról gondoskodik.

Az alkalmazásokat egyre inkább szabványos nyelveken (pl. UML, XML) [Haugen, 2001], modelleken keresztül fejlesztik, a modelltárak (logikai, szemantikai ábrázolás) építése ezek felhasználásával készülhet. A sématarak a konkrét integrációs feladatok szempontjából releváns üzleti modell-template-eket tartalmazzák.

Ez a megközelítés egyfajta technikai megoldást képvisel, amely eltér az európai, elsősorban akadémiai körökben elterjedt módszerektől. Az utóbbi esetben komponens alapú könyvtárak között kereső rendszereknek kell megtalálniuk az egymáshoz illeszkedő elemeket. A séma illetve ontológia alapú megközelítés menedzsment megoldásokkal kombinálva olyan módszert eredményez, amely ugyan nem univerzális megoldás, de az állandóan változó környezetben fellépő újabb és újabb szituációk kezelését támogatja. Alkalmazások kommunikációjakor többféle dinamizmussal is találkozhatunk: egyfelől a rendszerek maguk változnak, fejlődnek, másrészt pedig ad-hoc kapcsolatok alakulhatnak. Dinamikus egyensúlyban újra és újra meg kell oldani ezeket a problémákat.

A komplex integrált rendszerek által beépített funkcionalitás kritikus méretűvé válása, illetve a globális piac informatikával támogatni kívánt folyamatai alapján úgy tűnik, hogy integrációs probléma volt, van és lesz is.

II.2 ERP rendszer a szervezetben

II.2.1. Vállalat – Irányítás

A mai vállalatoknak igen eltérő piaci környezetben, változó feltételek között kell működniük, miközben a belső működésükhöz szükséges feladatokat is el kell látniuk.

A vállalat a fejlett üzleti vállalkozás egyik szervezeti formája. Jövedelmi célokat követő fejlődő rendszer, amely a ráfordítások megelőlegezésével erőforrásokat alakít át kibocsátásokká. [Koppányi Mihály, 2002]

Az üzleti vállalkozás olyan emberi tevékenység, amelynek alapvető célja, létének értelme a fogyasztói igények kielégítése nyereség elérésével. A vállalat az üzleti vállalkozás szervezeti kerete: a modern társadalmakban jogilag körülhatárolt olyan struktúra, amelyben az alapvető cél eléréséhez szükséges tevékenységek végbemennek. [Chikán Attila, 1997]

A gazdálkodó szervezetek rendszerek, erőforrások halmazai, amelyek úgy keletkeznek, hogy egyének ill. csoportok erőforrásaik egy részét egy rajtuk kívül álló rendező elv alá rendelik. [Kieser, 1995] [Fuchs,1979] [Grochla,1979]

A szervezetet tekinthetjük rendszernek, illetve, rendszerek összességének ahol a megfigyelés szintje szerint az alkotóelemek maguk is rendszernek tekinthetők. [Katz, Kahn, 1986]

A szervezet nem csak keretbe foglalja, hanem ciklikus tevékenység formájában át is alakítja az erőforrásokat. A vállalat működése során erőforrásokból transzformációs folyamatokon keresztül rögzített céloknak megfelelő teljesítményeket állít elő, miközben folyamatosan kapcsolatban áll környezetével. A vállalat teljesítményeinek előállítása tehát erőforrások felhasználása révén lehetséges.

Az erőforrások a vállalat értékteremtő folyamatainak a lehető legtágabban értelmezhető inputjai. [Antal-Mokos,Balaton,Drótos,Tari ,1997]

Anyagi erőforrások

Pénzügyi erőforrások - A vállalat hitelképessége

- A vállalat belső forrásképző képessége

Tárgyi erőforrások - A vállalat üzemének/berendezéseinek telepítése/elrendezése

- Alapanyagokhoz való hozzáférés

Emberi erőforrások - A vállalat egyes vezetőinek és beosztott dolgozóinak képzettsége, tapasztalata, ítélőképessége, intelligenciája, felismerő képessége, alkalmazkodóképessége, elkötelezettsége és hűsége

Szervezeti erőforrások - A vállalat szervezeti struktúrája, tervezési és controlling rendszere, koordinációs eszközei

Nem anyagi erőforrások

Technológiai erőforrások - Szabadalmak, védjegyek, szerzői jogok, kereskedelmi titkok és ezek sikeres alkalmazásának ismerete

Innovációs erőforrások - Műszaki és tudományos személyzet

- Kutatási infrastruktúra

Hírnév - a fogyasztók, a szállítók, a munkaerő és a pénzügyi

input-

szolgáltatók körében

[Antal-Mokos, Balaton, Drótos, Tari, 1997]

A vezetés alapvető feladata az erőforrások összehangolása és integrációja, valamint ehhez kapcsolódva a döntéshozatal. A szervezet vezetése a célok elérését támogató döntéseinek meghozatala, illetve az érték előállítási folyamat koordinációja érdekében koncentrálni az erőforrásokra. Amellett, hogy azonosítja és tudatosítja őket, igényt tart az erőforrás-felhasználási folyamat célorientált leképzésére - mérésére, értékelésére, tervezésére és elemzésére. A szervezet tehát olyan gazdálkodási modelleket használ, amely meghatározott célokkal, rögzített fogalmakkal és felépítéssel bír, az erőforrás-felhasználásról állít elő információkat és adott szabályok szerint támogatja a döntéseket. Az előállított információ, mint erőforrás egy újabb dimenzióban juttatja el a működtetéshez, koordináláshoz az erőforrásokra vonatkozó információkat a szervezet megfelelő szintjére.

A hatékony működés megköveteli, hogy ezen tevékenységeket egészében informatikai eszközökkel támogassák.

A sikernek alapfeltétele, hogy a vállalatok rugalmasan alkalmazkodjanak piaci környezetük változásaihoz, a felmerülő problémákat kreatív és korszerű módszerekkel oldják meg. Mindez gyors és hatékony intézkedéseket követel, amihez korszerű vállalatirányítási módszerek alkalmazása szükséges. A tulajdonosoknak és a vállalatvezetésnek egyaránt érdeke, hogy a vállalat hosszú- és rövidtávon is eredményes legyen. A multinacionális cégek terjeszkedésével egyre erősödik az igény, hogy a leányvállalatok működése is minél részletesebben megismerhető és nyomon követhető legyen - akár más országból is.

A vállalat vezetésének sikeressége nagymértékben függ attól, hogy rendelkezésre állnak-e megfelelő vezetési eszközök a döntések támogatásához és biztosítható-e a színvonalas információszolgáltatás. A vállalat működésének minden szintjén kulcsszerepe van az időben meghozott helyes döntéseknek – amelyekben a döntéshozó (a hierarchia bármely szintjén álljon is) saját intuícióján kívül leginkább az általa elérhető információkra támaszkodik.

Rendkívül fontos tehát megfelelő tervezési, ellenőrzési módszerekkel, számításokkal és különböző elemzésekkel alátámasztani a vállalati döntéseket és biztosítani a vezetéshez szükséges pontos, megbízható információkat. A cél a siker hosszú távú biztosítása, amely a múlt és jelen folyamatos elemző vizsgálatával, valamint a jövő stratégiai tervezésével érhető csak el.

Az üzleti folyamatok magasabb szintű szervezése, a várható folyamatok előrevetítése, valószínűségük és gazdasági hatásaik becslése megfelelő informatikai támogatás nélkül ma már elképzelhetetlen.

II.2.2. Integrált vállalatirányítási információrendszer

Az ERP rendszerek alatt itthon és külföldön egyaránt integrált vállalatirányítási információrendszert értünk. Az ERP elnevezés, ahogy láttuk, történeti okokra vezethető vissza.

Rendszerként felfogott szervezet esetén az ERP a szervezetben úgy tekinthető, mint rendszerben működő rendszer. [Fuchs,1979] [Grochla,1979] [Szintay, 1985]

Az ERP tehát, mint rendszer - egymással összefüggő entitások halmaza, ahol az egyes elemek hatást gyakorolnak egymásra. Ezek az entitások egy közös cél érdekében együttműködnek, egy teljes entitást hoznak létre, így a szinergiahatás érvényesülhet, azaz a hatások magasabb, mint a külön-külön vett egységek hatásokainak összege.

Mint információrendszer, egy szervezet olyan része, amely információt hoz létre, azt tárolja, szétválogatja, használja és elosztja, illetve működése során információt szolgáltat az igénylő egyéb részeknek.

Mint informatikai rendszer, a fentiek mellett szoftver és infrastrukturális elemei vannak.

Mint integrált vállalatirányítási rendszer, az integrálás következtében messze túlmutat egy-egy specializált rendszer funkcionalitásán - összefogja a szervezeti egységeket és tranzakciókat egyetlen közös adatbázist használó programrendszerbe. Alkalmazásával a különböző részlegek könnyebben oszthatják meg egymással az információt, és könnyebben kommunikálhatnak egymással.

„Egy olyan rendszerről van szó, amely külső és belső forrásokból képes adatokat konvertálni információkká, képes kommunikálni, azaz: közvetíteni funkcionális döntési pontokat, ahol időhöz kötött, hatásos, felelős döntéseket kell hozniuk – tervezési, irányítási és ellenőrzési célra.,,[Lucey, 1989]

A különböző méretű és integráltsági fokú szoftverek kifejlesztése kezdetben egy-egy vállalat egyedi megrendelése alapján készült. Egy ilyen rendszer elkészítése nagyon alapos körütekintést igényel, és megtervezése hosszú és bonyolult feladat, elkészítése több évbe telik. A vállalati méretek növekedésével, nő a tervezési-szervezési munka költsége, csökken a biztonság megszokott színvonala, egyre több az előre nem látható szervezési és költségvetési hiány, egyre igényesebbek a megrendelők, a felhasználók. Az informatikai szervezetek nem képesek megfelelő hatékonysággal, gazdaságosan olyan átfogó, integrált rendszereket kifejleszteni és folyamatosan korszerűsíteni, amilyen alkalmazásokra a kor követelményeihez, a megkövetelt fejlődéshez, hatékonyságnövekedéshez a vállalatoknak szükségük van. Ennek eredménye, hogy a rendszereket merevítették, kész modulokat hoztak létre a dolgok kézben tarthatósága érdekében, azaz létrejöttek a standard szoftverek.

A standard szoftverek a korábban szerzett tapasztalatok alapján egy elképzelt vállalati modellt ábrázolnak, amely azt vonja maga után, hogy egy ilyen standard szoftverben nagyon sokféle standard funkció található, amelyeket a bevezetés során a vásárló számára, a vállalat sajátosságainak, illetve igényeinek megfelelően lehet, illetve kell paraméterezni. Tehát a szoftver készen megvásárolható, az adott szervezet igényei szerint tesztre szabható. A standard szoftverek bevezetése mellett szól a sokéves tapasztalat, emellett kiküszöbölhetővé válik a bizonytalan hosszúságú fejlesztési idő.

A bevezetés leggyakoribb okai:

- közös IT platform megteremtése
- folyamatok fejlesztése iránti igény
- adatok elérhetősége
- működési költségek csökkentése
- fejlettebb vevőkiszolgálás
- stratégiai döntés hozatal javítása

További pozitívumok:

- az üzleti folyamatok támogatása

– szervezeti hatékonyság növelése

[Parr, Shanks, 2000]

Amikor egy szervezet ERP rendszert vásárol, akkor nem csak rendszert és eszközöket vásárol, üzemeltet és tart karban, hanem előnyöket, még hozzá stratégiai, üzleti előnyöket. Egyszerre kap valós idejű információt a vállalat eseményeiről, jobban integrált, automatizáltabb, rugalmasabb, áttekinthetőbb, menedzselhetőbb üzleti folyamatokat, nagyobb szervezeti hatékonyságot, jobb adat- és információ minőséget, a vállalati teljesítmények sokoldalú mérését, értékelését, az újabb üzleti lehetőségek jobb felismerésének lehetőségét, jobban előkészített vezetői döntéseket, integráltabb szervezeti tudást és még további előnyöket. ERP rendszerek nélkül a mai szervezetek nemcsak stratégiai céljaikat nem tudnák elérni, de a napi működésre sem lennének képesek.

A vállalatoknál jellemző tevékenységeket alapvetően reál- és pénz-folyamatokba sorolhatjuk. Ezekről elmondható, hogy párhuzamosak, de ellentétes irányúak. A vállalaton belüli tevékenységrendszer értékalkotó összekapcsolódását nevezzük értékláncnak.

A mai ERP rendszerektől elvárjuk, hogy funkcionális felépítésük ellenére hatékonyan támogassák az értéklánc mentén a vállalati folyamatokat. Az integrációból fakadóan az egymást követő folyamatok illetve a folyamatokon belüli lépések közötti információátadás eszköze nem változik. Ez a gyakorlatban annyit jelent, hogy nincs többszöri adatmentés, betöltés, nincs többszöri adatbevitel. Továbbá fontos jellemző, hogy egy ilyen rendszerben a funkciók, folyamatok nem keveredhetnek, és nem többszöröződhetnek meg. Az egyes alrendszerek között rendkívül szoros az együttműködés, amely a gyakorlatban úgy néz ki, hogy egy időben, egyszerre több felhasználó végezheti munkáját ugyanazon az állományon.

A cél a végső fogyasztó igényeinek minél jobban megfelelő, számára minél nagyobb értéket képviselő termék vagy szolgáltatás előállításának. Ennek az a feltétele, hogy a

folyamat elemei lehető legnagyobb mértékben járuljanak hozzá az értéknöveléshez. A vállalatok funkcionális egységei, tevékenységei, amelyek dominánsan minden üzleti vállalkozásban megjelennek függetlenül a cég tevékenységi (termelő, kereskedő, szolgáltató) körétől önállóan vagy összevontan részegységként, az alábbiak: marketing, kutatás és fejlesztés, emberi erőforrás menedzsment, logisztika, termelés és a pénzügyek.

A horizontális munkamegosztást követi a vállalatirányítási információs rendszerek funkcionális modulokra történő felbontása is, mely szerint ezen rendszereken belül pénzügyi-számviteli, kontrolling, tárgyi-eszköz gazdálkodási, termelésirányítási, készletgazdálkodási, kereskedelmi, humánerőforrás-gazdálkodási és egyéb modulokat különböztetünk meg.

Általános értelemben a vállalatirányítási információs rendszer a vállalat szinte minden tevékenységét lefedi. A vevőtől a szállítóig tartó logisztika, az előkalkulációtól az utókalkulációig tartó számvitel mind-mind az általános értelemben vett (érték) termelés, s így a vállalatirányítás hatókörébe tartozik. Ezen bonyolult folyamatok kezelését valamilyen rendező elv alapján lehet megoldani, vagyis tevékenységek részfolyamatonkénti – modulonkénti menedzseléseként. A modulokra bontás elvei között sok elméleti megközelítés létezik, amelyek között jelentősebbek az algoritmikus megközelítés, az adatmodell-alapú megközelítés, de a legelterjedtebb a funkcionális megközelítés

A vállalat értékteremtési folyamata nem ér véget a vállalat határainál, a sikerhez elengedhetetlen, hogy ki tudja elégíteni vevőkörének igényeit, és ez csak úgy lehetséges, ha szorosabb kapcsolatot tart fenn mind a vevőivel, mind a beszállítóival. Ahhoz, hogy ezt alaposan végezze, szüksége van információra és magas szintű ellátási lánc optimalizációra. A vállalatirányítás és az informatika hatalmas léptékű fejlődése eredményeképpen a mai szoftverek már biztosítják a felhasználó vállalatnál az üzleti folyamatok teljes integrációját mind vállalaton belül (tehát különböző funkcionális egységek között), mind vállalatok között.

A modularitásnak köszönhetően a vállalat növekedési pályájához, anyagi lehetőségeihez igazíthatja vállalatirányítási rendszerét, lényeg hogy nem kell egyszerre az összes modult bevezetni, hanem csak az éppen szükségeset vagy megfinanszírozhatóakat.

Egy standard vállalatirányítási rendszer amellet, hogy lefedi az összes funkcionális területet, bármilyen területen használható, nemzetközi, de egyben megfelel a helyi előírásoknak is. Nyílt rendszer, vagyis bármilyen más rendszerrel, illetve sziget megoldásokkal együtt is használható.

Az ERP rendszer alkalmazásainak integrálását három dimenzióban képzelhetjük el:

1. a hagyományos vállalati funkciók/tranzakciók (számvitel, logisztika, HR, ...)
2. az un. kereszt-funkcionalitás (dokumentum-, workflow- menedzsment, irodaautomatizálás, ...)
3. a különböző vállalati hierarchiaszinteknek megfelelő információk, döntéstámogatás

A bevezetés ellen szóló érvek:

- a gazdag funkcionalitás és rugalmasság ellenére sem mindig képesek a vállalatok egyedi, teljesen specifikus igényeit kielégíteni. Előfordulhat, hogy egy üzleti folyamatnak a vállalat és a szoftver által megvalósított módja között áthidalhatatlan eltérések vannak, esetleg a rendszer egyáltalán nem támogat bizonyos tranzakciókat.
- a szoftver és a vállalat között jelentősebb illeszkedési hiba forrása lehet, ha a rendszer nem támogatja a szervezet növekedési stratégiáját, struktúráját, kultúráját, döntéshozatali módját. A gyorsan növekvő, turbulens környezetben működő, ezért szervezeti stratégiát gyakran átalakító vállalat által általában nem alkalmazható.

- azok a problémák, amelyekre az integrált vállalatirányítási rendszerek megoldást kínálnak, más módon hatékonyabban oldhatóak meg.

[Markus, Tanis, Fenema, 2000]

Továbbá:

- Költséges - a bevezetés nem csupán a szoftver árából áll.
- Egy adott szoftver bevezetése előzetes vállalati elemzést igényel, hisz az adott rendszer a vállalat igényeire lesz szabva. Amennyiben a szoftver tervezőinek nem sikerül pontosan, élethűen felmérniük a vállalatban zajló igényeket, a szoftver nem tudja teljesíteni az elvárásokat.
- A bevezetéssel nem látható rögtön az eredmény, sőt az általa elért eredmények nehezen mérhetőek és számszerűsíthetőek, inkább a vállalat eszmei értékét növelik.
- A bevezetést követően nehéz átállni az új rendszer használatára. Ez eleinte a vállalati hatékonyságot csökkentheti. Bizonyos átállási időbe telik, amíg az alkalmazottak képesek és nyitottak lesznek az új rendszert használni.

A szervezetnek felkészültnek, érettnek kell lennie egy átfogó vállalatirányítási rendszer használatára. Ez azt feltételezi, hogy a szervezetben definiáltak a munkakörök, felelőségek és feladatok. Csak kialakult felelőségek és feladatkörök esetén lehet formalizálható irányításról beszélni. [Dobák, 1999]

Nagyon fontos feltétel, hogy a szervezetben működő folyamatok ismertek és definiáltak legyenek. Az olyan szervezeteknél, amelyeknél ezek nem léteznek, nem célszerű ilyen rendszer bevezetése, hiszen a szervezet működése olyan elvekre épül, amelyet nem lehet pontosan meghatározni, még kevésbé leképezni egy informatikai rendszer számára.

Fontos megemlíteni, hogy az ERP rendszer csak egy eszköz arra, hogy a szervezet működését kezelhessük. A bevezetés önmaga nem oldhat meg általában véve szervezeti problémákat.

Amikor egy nagyobb, vagy összetettebb vállalat elhatározza, hogy ERP rendszert vezet be, valószínű, hogy egymástól elszigetelt, esetleg valamilyen laza kapcsolatban álló rendszerek már működnek. A bevezetéssel az a cél, hogy egységes informatikai rendszert alakítsanak ki, kerülve a többszörös adattárolást, a többszörös adatbevitelt, automatizálják az adatok továbbítását. A központi rendszerhez aztán alrendszerek kapcsolódhatnak.

Az ERP rendszerek kínálata olyan széleskörű, hogy különböző igényű, tevékenységi körű, méretű szervezetek is megtalálhatják az elvárásaiknak és lehetőségeiknek leginkább megfelelő standard integrált irányítási szoftvert. A követelményeknek megfelelő szoftver alkalmazása tudja növelni a termelékenységet, szinkronba hozni a vállalati működés mozzanatait, segíteni a belső és külső szervezeti, valamint üzleti folyamatokat, hatékonyabbá, eredményesebbé tenni a vállalkozást.

Az ERP bevezetés a hosszú távú döntések közé sorolható, ezért mindenképp előtte alaposan átgondolt stratégiát kell kidolgozni.

Az ERP rendszer csak akkor tud hatékonyan működni, ha jól felépített, logikusan kialakított vállalat folyamataira épül rá. Az a tévhit, hogy az integrált rendszer bevezetésével minden probléma varázsütésszerűen megoldódik, több sikertelen bevezetéshez vezetett már. Rosszul szervezett folyamatokkal rendelkező vállalatnál nemcsak a működést, a hibák megjelenését is fel tudja gyorsítani. Ezért elkerülhetetlen, hogy az ERP rendszer bevezetése előtt a vállalat át ne tekintse az üzleti folyamatait. A bevezetésnek ez az egyik legfontosabb szakasza. A folyamatlemezési és értékelési áttekintést nem csak egy rendszer bevezetésekor éri meg végrehajtani, hanem rendszeres időközönként is, hiszen egy vállalat számára mindig fontos kell, hogy legyen, ezáltal a belső működés jelentősen javítható, egyszerűsíthető, optimalizálható.

A bevezetés során a standard rendszert az adott szervezetre adaptálni kell. Ez a testre szabás (customizing), vagy beállítás nagyon bonyolult feladat és sok időt és ráfordítást igényel. A beállítások elvégzéséhez szükséges, hogy mind a rendszer, mind pedig a szervezet jól ismert legyen. Elmondható, hogy fontos a megfelelő bevezetési stratégia melynek kiválasztásához szükséges a szervezeti jellemzők figyelembe vétele.

Amennyiben sikerül a bevezetés idejét csökkenteni, nagy valószínűséggel a költségek és a bizonytalanság is csökken. A bevezetés idejének lerövidítésére egy lehetséges módszer, a bevezetés folyamatának valamilyen standard módszerrel történő támogatása. Egy ilyen módszertan célja az összetett folyamatok támogatása és a bonyolult bevezetési eljárások egyszerűsítése. Mindezt úgy éri el, hogy többféle segédeszközzel támogatja a bevezetés teljes folyamatát.

Amikor egy ERP rendszert használatba vesznek úgy, hogy sikerül annak előnyeit kihasználni, általában minőségi változás következik be a szervezet működésében.

II.3. Az üzleti integráció

II.3.1. Az üzleti integráció szerkezete

A vállalatok növekedésével nő a vállalati integráció igénye, melynek megvalósítása többféle értelemben történhet. Az integráció igazából üzleti szükségszerűség, megfelelő szervezeti struktúra, célok, ösztönzés nélkül a technológiai szerkezet nem tud segíteni. Az eredmények elérése, a hatékonyság növelése érdekében a szoftver-/rendszer-integráció önmagában nem elegendő. A szervezet és a technológia összehangolása alapvető fontosságú. [Markus, 2000]

A szervezetnek elsősorban nyerő stratégiára van szüksége, melynek megvalósítása során az integrált architektúrának - technikai és szervezeti komponenseknek támogatni - kell a szervezet anyagi erőforrásainak, pénzeszközeinek és emberi erőforrásainak integrációját.

	Integrálandó elemek	Eszközök, mechanizmusok	Befogadó környezet/infrastruktúra	
Szervezet integráció	Szervezeti egységek	e-mail, csoportmunkát támogató szoftverek ----- Felsővezetői stratégia, büdzsék, teljesítménymutatók	Szervezeti politika/struktúra	
	Döntéshozók	e-mail, csoportmunkát támogató szoftverek, KM rendszerek ----- Személyes találkozók, kinevezések, teljesítménymutatók		
Rendszer integráció	Üzleti folyamatok	Workflow, csoportmunkát támogató rendszerek, SCM, CRM, Web szolgáltatások ----- Folyamat felelősök, munkacsoportok, teljesítménymutatók, szolgáltatás szintű megállapodások	Szabványok	Rendszer architektúra
	Alkalmazások	Folyamaton belüli kommunikáció, RPC, üzenetek, ERP, Web szolgáltatások	Hálózatok	
	Adatok	Adatszótárak, Adatbázisok, XML	Platformok	

3. ábra Üzleti integrációs modell [Edvard A. Stohr , Jeffrey V. Nickerson , 2003]

Az oszlopokban, horizontális hatáskörüknek megfelelően helyezkednek el a modell elemei. A technikai és szervezeti elemeket a cellákon belül szaggatott vonal választja el. Egyes eszközök több szinten is szerepet játszhatnak.

A hatékony integráció során minden szint minden elemére figyelmet kell fordítani – horizontálisan és vertikálisan is. A rétegek közötti integrációt is meg kell valósítani. Egy réteg integrálhatósága függ az alatta lévő rétegek integráltságától.

A vállalati integrációval foglalkozó irodalom elsősorban a technikai oldalt veszi górcső alá. A fő területek az adat, alkalmazási réteg, szabványok, architektúra, hálózati infrastruktúra lehetőségek, amelyek integrálni tudják ezeket a technikai elemeket. A szervezet tervezésével foglalkozó irodalom a döntéshozatal és részlegek integrációja réteget, valamint a támogató szervezeti struktúra, politika és stratégia részeket elemzi.

Az új irány, amire a szervezetelméletek is egyre inkább fókuszálnak a középső réteg, az üzleti folyamatok, felismervén, hogy alapvető, létfontosságú kapcsolatot jelentenek a technikai és szervezeti infrastruktúra között.

Adatintegráció

A vállalat értékes adatai, információi különböző alkalmazások adatbázisaiban helyezkednek el.

A cél a különböző forrásból származó adatok kombinálása, összesítése, és a jelentéskészítés. A dokumentálatlan, nem létező, nem szabványos interfészek, a különböző formátumú redundáns adatok csak bonyolult eljárásokkal, transzformációkkal integrálhatók. Az adatoknak jó minőségűeknek kell lenniük (aktuális, pontos, releváns) ahhoz, hogy az integrációs befektetés eredményes legyen.

Szintaktikai nézőpontból a programoknak kezelni kell a különböző meghajtókon, médiákon, különböző formátumban tárolt adatokat.

Szemantikai szempontból az adatelemeknek értelmezhetőeknek kell lenni, és ugyanannak az adatelemnek ugyanazt (definíció szerint) kell jelenteni az alkalmazások között cégen belül, és kívül is.

Alkalmazási területek:

- Időszerű és pontos információk szolgáltatása
- Elemzéshez és döntéshozatalhoz, vezetőknek - adattárház alkalmazás
- Vevők számára gyártmány/termékkatalógus
- Hiteles adatok teljesítmény-méréshez/értékeléshez, audithoz
- Alkalmazások közötti kommunikáció megkönnyítése – az alkalmazások és üzleti folyamatok integrációjához

A fő probléma a különböző típusú adatbázisok lekérdezése a sokféle nyelven írt programmal.

Mechanizmusok:

Az eltérő szintaxisú, inkonzisztens adatokat adattisztítással, párosítással egyesítik.

A legtöbb programozási nyelv lehetőséget ad osztott adatdefiníciók használatára.

Magasabb szinten adatszótár alkalmazása, ahol már a szemantikán van a hangsúly.

Az általános dokumentumoknak standard forma az EDI, amelynek a vállalatok közötti adatintegrációs szerepe jelentős.

Jelenleg a legelfogadottabb megoldás az XML, amely egyszerre tartalmazza az adatot és az adat leírását. [Ibbotson, 2001]

Adat- és alkalmazásszint integráció

Az alkalmazások csak saját feldolgozási és analitikus területükre fókuszálnak. Az adatbázisokhoz való közvetlen hozzáférés lehetővé teszi a különböző fájlokban és adatbázisokban tárolt adatok elérését és értelmezését. A különböző szállítóktól származó alkalmazások szabványos (pl. ODBC, JDBC) eszközökkel érhetik el az adatbázisokat.

Egymástól független alkalmazások azonos séma alapján használhatnak közös adatbázist. Egyszerű, költséghatékony, könnyen karbantartható megoldás, nincsenek szinkronizálási feladatok, a konzisztencia biztosított. Olyan esetben is alkalmazható, ahol az alkalmazás nincs felkészítve az integrációra. Az adatstruktúra módosítása minden alkalmazást egyszerre érint.

Alkalmazásintegráció

Alkalmazásintegrációra azért van szükség, mert az adat, ill. tartalom integráció nem minden esetben lehetséges (pl. az alkalmazás adatbázisa „zárt”, dokumentálatlan), vagy nem elégséges. Az integrált alkalmazások meghívhatják egymás funkcióit. Az alkalmazások az adatbázis fölötti szinten különböző „üzleti” logikát tartalmazhatnak. nem minden esetben lehetséges. A nagy teljesítményű batch tranzakciók alkalmazás integráció során sok overhead-et tartalmazhatnak.

A heterogén adatforrásokkal nehézkes az alkalmazások közötti integráció, vagy egységes adattárház kialakítása.

Célok:

- A komplex folyamatokat egymástól független programok által végrehajtott lépésekre bontani.
- A manuális lépések számát minimalizálni. Különös tekintettel az egyszeri adatfelvitelre!
- Az interaktív feldolgozás támogatása (pl. Online vásárlás - összekapcsolva a különböző alkalmazásokat egy felületre).
- Adat- és információintegráció megkönnyítése.

Mechanizmusok:

- Middleware
 - Szinkron RPC hívás
 - Tranzakció menedzser
 - AlkalmazászerverEgymásra épülnek. Nincsenek kapcsolatban emberekkel. Az üzleti folyamatokat indirekte támogatják az integrált alkalmazásokon keresztül.
- Aszinkron üzenetek
- Kijánlott metódusok, objektumok
- Web szolgáltatások, ESB
- ERP rendszer, mint magasabb szintű megoldás.

A funkcionális alkalmazások integrálása alapvető fontosságú a következő réteg integrációjához. Az ERP rendszerekkel szembeni legfőbb probléma, hogy nem tudják rugalmasan áthidalni az alkalmazás és folyamat réteget. Ez okozza azt a legtöbbször emlegetett problémát is, hogy könnyebb a szervezetet az ERP-hez igazítani, mint fordítva.[Esteves, Pastor, 2001]

Alkalmazás- és üzleti folyamat integráció

Az üzleti folyamat emberi vagy szoftver szereplők által az üzleti szabályoknak megfelelően végrehajtott tevékenységek csoportja. Az üzleti folyamatok alkalmazásokat hívhatnak meg. A versenyképesség fenntartása érdekében egyre fontosabbá válik, hogy a valós folyamatokat leképező információk minél frissebbek legyenek, a feldolgozás ne legyen fáziskésésben a valós folyamatokhoz képest, hiszen ez a kulcsa a folyamatba való gyors beavatkozásnak és a szabályozásnak. A folyamatok minél több elemét igyekeznek elektronikus formában megvalósítani, automatizálni, kihasználva ezzel az IT nyújtotta előnyöket és kiküszöbölve az emberi hiba szerepét.

Az első generációs workflow rendszerek a különálló alkalmazásokat „emberi interfészeken” keresztül illesztették a folyamatokba. Ilyen például a call center.

A második generációs workflow rendszerekben az infrastruktúra kialakítása az elsődleges, amelyben a broker vagy ügynök interfészeken keresztül zajlik a folyamatok támogatása vagy a vállalati alkalmazások integrációja.

Mechanizmusok:

- ERP. A vállalatirányítási rendszerek integrálják a különböző üzleti funkciókat, standard, „best practice” megoldásaikkal és ezzel párhuzamosan nyújtott testre szabhatóságukkal tovább javítják a vállalati működés és a vállalati folyamatok hatékonyságát.
- Workflow menedzsment rendszer
- BPM rendszer

Üzleti folyamatok integráció

Célok:

- Emberi és szoftver szereplők közötti koordináció.
- Szervezeti egységek közötti koordináció.

- A szervezet összekapcsolása a szállítóival és vevőivel.
- Az értékláncban szereplő partnerek együttes teljesítményének optimalizálása.
- Hatékonyságnövelés, gyorsan tudjon reagálni, költség és szolgáltatás kapacitásban.

Típusai:

- Within-process integration – a folyamat céljainak igazodnia kell a szervezet egészének céljaihoz
- Cross-functional integration – részlegeken átívelő folyamatok esetén
- Internal process-to-process integration - folyamatok egymáshoz hangolása
- External process integration - a vállalat és kereskedelmi partnerei között

Kettős cél:

- Teljes automatizálás – straight-through processing
- A tranzakciós idő csökkentése a kereskedelemben
- Value chain coordination – az értéklánc tagjai között

Mechanizmusok:

- Workflow menedzsment rendszerek
- BPM rendszerek

Minden szint integrációjában alkalmazhatóak. Kezdetben főleg folyamat, keresztfunkcionalitás és belső integráció megvalósításában használták. Ma a külső és kiterjesztett értékláncok integrációs támogatására is ezeket a megoldásokat alkalmazzák.

Üzleti folyamatok – döntéshozatal integráció

Az üzleti életben megfigyelhető munkafolyamatokról elmondható, hogy azok a közelmúltig a századok során alapjaiban keveset változtak. Vezetők osztják szét a feladatokat a beosztottak között azok képzettsége, tapasztalata alapján. A folyamat egyes lépéseit különböző erőforrások végzik, közöttük áramlik az információ dokumentumok

formájában. Az ember (vagy szoftver ágens) olyan döntést hoz, amely megváltoztat egy folyamatot, illetve a folyamat döntéshozók számára szolgáltat információt.

Az elmúlt 20 év során kifejlesztettek olyan eszközöket, amelyek már nem csak a munkát végzik el automatikusan, hanem a számítógépes program osztja ki a munkát, továbbítja a dokumentumokat, az információt, és követi nyomon a folyamatot az elejétől a végéig.

Mechanizmusok:

- Workflow rendszerek.
- Információ-szolgáltatás döntéshozók részére (EIS, KM).
- Felhasználói interface szint- legtöbbször a vállalat sikeressége múlik rajta.

Döntéshozók közötti integráció

Döntéshozatal felsőbb, stratégiai koordinációs szinten. Célja, hogy a döntéshozók könnyen kicserélhessék ötleteiket és tudásukat, összehangolják tevékenységüket a vállalat érdekében. Fontos, hogy a vezető ne legyen felső szinten izolálva.

A tevékenységek közötti interakción van a hangsúly, szemben a kezdeti szervezeti egységek, emberek közti interakció támogatásával. Ez az újabb fajta gondolkodási mód helyezi előtérbe a munkafolyamatokat, projekteket. [Malone, Crowston, 1994]

A vezetők közti interakció azonban nem merül ki a munkafeladatok körüli teendőkkal, fontos a kapcsolati tőke ápolása is.

A pszichológiai jelentőség sem lebecsülendő. Nagy, költséges projektek tudnak megbukni emberek közti súrlódások miatt. Ezen a szinten előtérbe kerül a kevésbé strukturált kommunikációs formák - találkozók, ebédek, e-mail, társasági kapcsolatok – támogatása.

A számítógépes rendszerek önmagukban nem szolgáltatnak kellően gazdag információt a döntéshozóknak.

Mechanizmusok:

- Telefon, személyes találkozás
- Hangposta, e-mail (aszinkron)
- Multimédiás lehetőségek: javítják az információ elérhetőségét és értékét
- Kollaborációs rendszerek – megosztott virtuális tér a döntéshozók számára, hogy összehangolják tevékenységüket, információkat és ötleteket osszanak meg, növelve a információ elérhetőségét és értékét (aszinkron kommunikációs lehetőség).

Döntéshozók–szervezeti integráció

A döntéshozók a hatáskörüknek megfelelő szervezeti egységükbe integráltak, tevékenységeiket összehangolják egymással a szervezeti céloknak megfelelően.

Mechanizmusok:

A klasszikus szervezetelmélet sokat foglalkozik ezzel a problémával. A javasolt ötletek – a felügyelet mértéke, a hatáskör átruházása, a célok kommunikálása alacsonyabb szintekre, és a költségtervezés - ma is érvényesek.

Jó, ha ösztönzőkkel támogatott a döntéshozók szervezeti egységekbe integrálása, a döntéshozók motiválása a hozzájuk tartozó csoportok céljaival való azonosulásra. Hatalmas irodalma van az ösztönzési sémáknak a közgazdaságtudományban, [Jensen, Meckling, 1973] a vezetéstudományban [Eisenhardt, 1989], és a teljesítménymenedzsment területeken (ld. BSC) [Kaplan, Norton 1992].

Fontos fejlesztés az összekapcsolt értékláncok kialakulása óta a különböző szervezetek döntési mechanizmusainak integrálása.

Szervezeti divízionális integráció

A szervezetek azért jönnek létre, hogy egy rendszerként valamely cél elérése érdekében tevékenykedjenek. Struktúrájuk, méretük a feladatok jellegétől függ, de abban

megegyeznek, hogy valamilyen bemenetre (input) van szükségük, és valamilyen kimenetet (output) bocsátanak ki a környezet számára, amely akár egy másik szervezet is lehet. A bonyolult szervezetek ugyanezen jellemzőkkel rendelkező jól elhatárolható alszervezetekre oszthatóak.

A szervezetek felépítése fontos abból a szempontból, hogy miként jutnak el az információk a megfelelő időben a megfelelő helyre. A hagyományos, piramisszerű, hierarchikus szervezeti felépítés egyáltalán nem hatékony, mégis túlsúlyban van a korszerűbb mátrix vagy folyamatorientált szervezeti felépítéshez képest.

A ranglétrán alapuló szervezeti felépítés nehezíti az információáramlást. Az információ az egyes hierarchia szinteken módosulhat, sőt el is akadhat. A csúcsvezetők sem szívesen adják fel monopolhelyzetüket.

A legjobban működő szervezetek erősen differenciáltak és erősen integráltak. A specializáció eredményeként más és más külső feltételekre kell koncentrálni. Eltérő vezetői szemléletek növelik az integrációs kihívást a célok, idő, személyes kapcsolatok, strukturális szerkezet dimenziókban. [Eisenhardt, 1989]

Az integráció feltételei:

- jól kiépített kommunikációs rendszer.
 - Szervezeti egységek közti folyamatok biztonságos végrehajtása.
 - Szervezeti egységek informálása egymás tevékenységeiről azon erőforrásokkal kapcsolatban, melyekért versenyeznek, ill. a megosztott folyamatokkal kapcsolatban.
- Szervezeti egységek céljainak összehangolása.

Az integráció az alábbi okok miatt célszerű:

- A szervezeti egységek (sokszor kölcsönösen is) függenek egymástól inputok tekintetében.
- A szervezeti egységeknek gyakran együtt kell működniük egy folyamat különböző részeinek végrehajtásában.
- Még hatékonyabb erőforrás-megosztás, szervezeti szabványok fejlesztése.

- A funkcionális integráció támogatja a folyamatintegrációt, mivel a funkcionális vezetők a folyamat menetét figyelembe véve jobban tudják koordinálni döntéseiket.

Mechanizmusok:

A klasszikus szervezetekkel foglalkozó elméletek alapján a szervezeti egységek integrálása vertikálisan irányított hatáskör-delegálással, hierarchikus parancs és ellenőrzési struktúra keretében valósítható meg. A funkcionális célokat és teljesítménymutatókat a felső vezetés határozza meg.

A számítógéppel támogatott együttműködés során eltérő helyen és időzónában élő dolgozók lehetnek tagjai ugyanannak a munkacsoportnak. Azaz feloldódnak az időbeli és a területi korlátok. A groupware termékek az eredményeket tárolják, kategorizálják, és meg tudják osztani a szervezeten belül, függetlenül annak keletkezési idejétől és helyétől. A korszerű portál alkalmazások hatékonyan támogatják a szervezeti és szervezetközi kollaborációt.

A valóságban a legtöbb szervezeti tevékenység különböző, ma már gyakran más szervezetekhez tartozó szervezeti egységeken áthaladó folyamatokba illeszkedik. A 90'-es évektől induló folyamatszemplélet és BPR elméletek alkalmazásával, a vertikális szervezetek mátrixszerűen átalakulnak a horizontális folyamatigények mentén. A szervezeti egységek, munkakörök, erőforrások és célok megosztása a vállalati folyamatokból és feladatokból származtathatók. Új feladatkör alakul ki, érteni kell az összes integrált szervezeti egység „nyelvét”.

II.3.2. Vállalati Integráció Technológiai Megoldásai

Jelenlegi Trendek

A vállalati integrációt több technikai és szervezeti szinten is meg kell valósítani. Az integrált vállalat összeköti az összes létező alkalmazását úgy, hogy azok könnyen tudják használni egymás adatait. Az üzleti folyamatok kiterjednek a különböző alkalmazásokra,

kombinálhatóak, az egyik kimenetét a másik bemenetként használhatja. Az integrációs mechanizmus növeli a szervezeti teljesítményt. Az az elképzelés, hogy az integrált vállalat legyen könnyen változtatható nem változott az idők folyamán. Nehéz megvalósítani, de a törekvés jól látható.

- Az integrációs problémát a lokális integrációs megoldások helyett a teljes vállalatot átfogó keretrendszer használatával oldják meg. A keretrendszer megvalósításához rendszerterv készül a fő alkalmazások figyelembevételével. Az integrációs platform üzembe helyezésével a felmerülő integrációs problémák gyorsabban orvosolhatóak, az új projektek gyorsabban véghezvihetőek.
- A laza csatolású környezet népszerűbb mint a szoros csatolású, mivel utóbbinál a rendszer változásakor minden hozzá kapcsolódó rendszert is változtatni kell, míg laza csatolás esetén a rendszer változtatása a hozzá kapcsolódó rendszerektől függetlenül is megvalósítható.
- Az XML az egységes fájlformátum, az adateleíró nyelv segítségével ténylegesen standardizálni lehet a vállalaton belüli és/vagy kívüli adatcserét. Nagyságrendekkel tudja csökkenteni a vállalaton belül előforduló fájlformátumok számát.
- Az egyre szorosabb vállalati együttműködés új integrációs kihívásokat teremt. A külső körülmények (mint például az értékláncban szereplő partnerek) befolyásolják a vállalat belső folyamatait. Az ipari szabványok befogadása is jelentős kihívást jelent.

Az Integráció Általános Megközelítései

Az alábbi három megközelítéssel a vállalati integrációban leggyakrabban alkalmazott technikai megoldások leírhatók.

Szekvenciális Integráció

Ez jelenti az integráció kezdeti formáját – az egyik rendszer outputja valamilyen fizikai eszközön jut el a következő rendszerig (pl. kártyaolvasó). Nem jelent megoldást

különböző rendszerek szinkronizációjára. Fejlettebb formája a főbb programok és az adatátvitel kezelésére közös programozási nyelv használata - batch programok, jobok (pl.: IBM Job Control Language). Modernebb megközelítése a közös megkötésekkel és interface-szel rendelkező UNIX keretrendszer.

Adatbázisok

Több különböző alkalmazás információ megosztása lehetséges úgy is, ha egy közös adatbázisba írják és onnan olvassák az adatokat. Az egyes alkalmazásoknak nem szükséges tudnia egymásról, csak a közös adattárolási és értelmezési szabályokat kell kialakítani.

Üzenetek

A modern rendszerek üzenet alapúak. A rendszer minden egyes része önálló entitás, amely alkalmas üzenetek létrehozására és fogadására. A megközelítés előnye, hogy alkalmazható chippek között küldött jelektől a nagy számítógépes rendszerek között cserélt file-okig. Az előző két megközelítés származtatható ebből, mint speciális esetek.

Az üzenetalapú technológia a távoli végrehajtási egységek közötti kommunikációt üzenetek küldésével szervezi meg.

Az üzenet orientált middleware alkalmas laza kapcsolású rendszer létrehozására, ahol a komponensek fekete dobozként kezelhetik a többi komponenst.

Szabványos Integrációs Mechanizmusok

Integráció Fájlok Segítségével

A rendszerek integrációja fájlok átvitelével, pont-pont kapcsolaton keresztül valósul meg. Kevés rendszer esetén hatékonyan, kis költségek mellett működhet. Könnyen megvalósítható. Csak olyan esetben használható, ahol nem szükségesek a valós időhöz közeli adatok. A rendszerek számának növelésével használhatósága egyre csökken.

A fájl-átvitel alapja a közös, vagy osztott fájlrendszer, illetve az FTP. Szinkron és aszinkron működés is megoldható. A kommunikáció jellege one-way ill. request-response lehet.

Távoli Eljárás Hívás (RPC - Remote Procedure Calls)

Az üzenetalapú technológia tipikus példája. A processeken átívelő eljárás hívások nem sokban különböznek attól, mintha lokális metódushívást végeznénk. Az interface - metódusok, bemeneti paraméterek, visszatérési értékek -leírása az IDL (interface definition language) nyelvvvel történik. Nincs szükség köztes rendszerre, elegendő a hálózati kapcsolat.

A technológia a kliens/szerver architektúrán alapul. A szerverek egyszerre több klienst is tudnak kezelni. Sokszor más szerverekhez is kell folyamodni, ilyenkor késik a válasz.

A megoldás előfeltétele a szoros csatolás, mivel a kliensnek tisztában kell lennie, hol a szerver, és a szerver által várt paraméterekkel. Sem a kliens, sem a szerver nem cserélhető a másiktól függetlenül.

A küldő alkalmazás választ vár a címzett alkalmazástól és amíg a válasz meg nem érkezik, addig nem folytatja a feldolgozást, vagyis a hívó fél blokkolódik. Közvetlen kapcsolatot, egyidejű rendelkezésre állást, stabil hálózati összeköttetés feltételez az üzenetváltás időtartamára.

Az OOM – Objektum Orientált Middleware -az RPC-hez hasonló, ám objektumorientált szemléletű. Egyszerű (mintha helyi metódusokat hívnánk), csökkenti a komplexitást.

Egyetlen komponens elérhetetlensége a teljes rendszer működését megakaszthatja. A leglassabb komponens teljesítménye dominál.

Több integrációs keretrendszer is használja (pl. CORBA, COM), melyek többsége platform, ill. nyelv – specifikus megoldás.

ERP

Az ERP elsősorban nem integrációs technológia, inkább egy megközelítés, de a szerepe ezen a területen is fontos. A vállalat funkcionális egységeit összekapcsolja. Az integráció a közös adatmodell, adatbázis és interface segítségével valósul meg. Közös beszámolóképzési lehetőségekkel szolgál funkciókon, szervezeti egységeken keresztül.

Az új generációs ERP rendszerek már beépítik az új technológiákat, mint például az alkalmazás szerverek. Ezzel rugalmasabbakká válnak, megvalósítható a laza csatolás, könnyebb a változtatás.

/Bővebben lásd: II.1.-2. fejezet/

Integráció Adatbázisok Segítségével

A legegyszerűbb esetben egy adatbázist használ osztottan több alkalmazás. Az adatbázis mint egy megosztott memória szolgálja ki a hozzá forduló alkalmazásokat.

A gyakorlatban azonban egy alkalmazás esetén optimalizálható jól az adatbázis, különben nehézkes a feladat közben tartása. Egy komplex tranzakció lokkolhat több adattáblát is, így más alkalmazás sokszor percekig várhat. A fő probléma a teljesítménykontroll elvesztése.

Jó megoldás, ha az alkalmazások egy helyi adatbázis alkalmaznak, majd a helyi adatbázis tartalom feltölthető egy közös adatbázisba aszinkron technológiával. Ezzel megoldható a lokális kontroll, a gyors tranzakciók "helyi" szinten hajthatók végre, ugyanakkor integrált, konzisztens, globális információ nyerhető.

Ehhez a megoldáshoz egységesített adatmodell kell, ami drága, időigényes és nehéz feladat. A karbantartása is nagy figyelmet kíván különösen a gyorsan változó üzleti feltételek között. A jelen trend iparági szinten közös modell készítése és alkalmazása.

Az egymástól független alkalmazások saját adatbázisokat, de (részben) azonos adatstruktúrákat használnak. Meghatározott időközönként az adatbázisok szinkronizációja és replikációja szükséges.

Az egymástól független alkalmazások külön adatbázisokat, eltérő adatstruktúrákat használnak, a transzformációt az ETL modulok végzik.

Az alkalmazások közötti adatcsere interfész-táblákon keresztül történik. Az interfész táblák kerülhetnek külön adatbázisba, de közvetlenül valamely alkalmazás adatbázisába is.

Az MDM (Master Data Management) a legújabb integrációs platformok alapvető eszköze, amely a törzsadatokat központi, alkalmazás-független információ-forrásként kezeli. Konzisztens, jó minőségű adatokat biztosít a tranzakcionális és analitikus rendszerek számára. Az adatok tisztítása már az adatforrásoknál történik, nem az adattárházba töltéskor. A törzsadatokat függetleníti az azt nyilvántartó alkalmazásoktól, ezzel egyszerűsíti az integrációt és az új alkalmazások fejlesztését.

A felépítése témacentrikus (pl. ügyfél, termék, partner, ...). A lehető legtöbb adatot tartalmazza egy adott entitásról. A témákat érintő összes adatforrással szinkronizált.

Amellett, hogy központi datbázisként használható, az új alkalmazások számára SOA szolgáltatásokat kínál, eseményfigyelést tartalmaz és szerep alapú jogosultságkezelést biztosít.

Az Újabb Integrációs Mechanizmusok

A világméretű elosztott szoftver-rendszerben az alkalmazások szoftver komponensként jelennek meg. Olyan komponensként, amelynek a végső felhasználási módja a tervezéskor nem ismert. A komponensnek van egy elképzelt funkciója, de tetszőleges módon lehet más rendszerekbe beépíteni. Ugyanígy ez a komponens is felhasználhatja más komponensek szolgáltatásait, azonban tervezéskor még nem ismert, hogy melyiket. Az ilyen architektúra lehetővé teszi, hogy a szoftverrendszerek is hasonlóan szerveződjenek, mint a gazdasági élet szervezetei. A szoftverkomponensek meghirdetik szolgáltatásaikat, a szolgáltatást felhasználó komponensek rákereshetnek a számukra szükséges szolgáltatásokra, a leírások és az egyéb forrásokból rendelkezésre álló

referenciák alapján kiválaszthatják a megfelelő komponenst, és annak felhasználásával érhetik el céljukat.

Az ilyen környezetben a szoftver rendszerek alkalmi társulásokkal alakulnak ki, a komponensek egymás szolgáltatásait dinamikusan kombinálják, az egyik alkalmi szoftver rendszerben szerzett tapasztalatokat a komponensek a következő rendszerben felhasználhatják. Minden a világhálón elérhető szoftver rendszer web-alkalmazássá válhat. Ilyen rendszerek kialakítására sokan törekednek, és már bizonyos módszerek és rendszerek rendelkezésre is állnak, de a kellő rugalmasságú megoldásokra még várni kell.

Az integrálható alkalmazások kialakítása új szoftver technológiát igényel. Az új technológiának lehetőséget kell biztosítania arra, hogy a rendszer teljes funkcionalitásának kialakítása kitolódjék a tervezés, megvalósítás és telepítés cikluson túl, az üzemeltetés idejére, amikor is a szoftverkomponensek saját maguk alakítják ki kapcsolataikat más szoftverkomponensekkel. Ez megköveteli azt, hogy a szoftver komponensek dinamikus és autonóm jellemzőkkel bírjanak.

Ugyancsak fontos elem, hogy a rendszer kialakításánál az egyes komponensek eddigi belső működéséhez képest sokkal nagyobb jelentőséget kap a komponenseket összeragasztó közeg. Ennek a közegnek biztosítania kell, hogy például a komponensek funkcionalitása, elérhetősége, minőségi jellemzőik, az általuk használt adatok, a munkafolyamatokba való beépíthetőségük gépi úton feldolgozható formában legyen leírva. Ugyancsak gépi úton kell kezelhetőnek lenniük a komponensek nyilvántartásának és kereshetőségének, ami egyben azt is jelenti, hogy a világháló infrastruktúrájában szabványoknak és konvencióknak kell kialakulniuk.

Ezeket a technológiai elemeket több megközelítés is célul tűzte ki, mint például a SOAP, WSDL, UDDI rövidítésekkel jellemezhető web-szolgáltatási technológia, a szemantikus web-elképzelés, vagy a GRID, a SOA, valamint az ágens alapú szoftverfejlesztés.

Üzenet-orientált köztes rendszer – MOM (Message Oriented Middleware)

Aszinkron kommunikáció során a küldő alkalmazás nem vár azonnali választ a címzett alkalmazástól és ahogy elküldte az üzenetet, folytatja a feldolgozást.

Az üzenet-orientált köztes réteg legfontosabb tulajdonságai:

- aszinkron, lazán csatolt, rendelkezésre-állástól független, de tranzakcióbiztos feldolgozás
- közvetített üzenetek biztos, egyszeri de csakis egyszeri továbbítása
- egyes alkalmazások eseményvezérelt üzemeltetése
- platform független, egyszerű programozási API felületet

Publish - Subscribe

A pont - pont modell esetében a üzeneteket készítő és küldő alkalmazás (kliens1) üzeneteket küld az üzenetsorba, a fogadó alkalmazás (kliens2) pedig kiolvassa azokat a sorból. A küldő ismeri az üzenet fogadóját és közvetlenül a fogadóhoz tartozó sorba pakolja az üzenetet.

Ebben az esetben egy üzenetet csak egy fogadó kap meg. A küldőnek nem kell futnia, amikor a fogadó megkapja az üzenetet és a fogadónak sem kell futnia, amikor a küldő elküldi az üzenetet. Minden sikeresen kézbesített üzenetet visszaigazol a fogadó.

A publish - subscribe modell adott témához (topic) tartozó üzenetek publikálását támogatja. Nulla vagy több „subscriber” regisztrálhat egy-egy témára. Ebben a modellben a kommunikáló felek nem ismerik egymást. Egy üzenetet több fogyasztó is megkaphat.

A küldő és a fogadó között időbeli függőség áll fenn. A küldőnek létre kell hoznia egy “topic”-ot, amelyre a kliensek feliratkozhatnak. A feliratkozott klienseknek folyamatosan aktívnak kell maradniuk ahhoz, hogy megkapják az üzeneteket vagy tartós feliratkozást kell használniuk (angolul durable subscription). Az utóbbi esetben újracsatlakozáskor minden üzenetet megkap a fogyasztó.

Standard megoldás például a Java Messaging Services.

Alkalmazás Integráció (EAI - Enterprise Application Integration)

A nem ritkán több tíz alkalmazást használó szervezet a különböző rendszerek közötti kommunikációt sokszor egyedi fejlesztések útján kényszerül megoldani. Ha minden egyes alkalmazást külön-külön összekapcsolnak a többivel a kapcsolatok száma $[N*(N-1)]/2$, ahol N az alkalmazások számát jelenti. Az egyedi, különleges interfészek rugalmatlanná teszik ezeket a kapcsolatokat, és a vállalat informatikai környezete képtelenné válik a változások követésére, mivel minden változtatás időigényes és költséges fejlesztést követel.

A vállalati alkalmazások integrálására kilencvenes években létre jött szoftverkategória - Enterprise Application Integration - a problémakört és a megoldást jelentő technológiát is magába foglalja. Az integráció során, a kapcsolatok nem esetlegesek az alkalmazások között, hanem átgondolt, megtervezett sémát követnek.

Az alkalmazások integrációját middleware-k - köztes szoftverek segítségével valósítják meg. Elosztott számítástechnikai rendszerekben azt a szoftverréteget nevezik köztesrétegnek, amely - elfedve az elosztottságot - egy egységes felületen keresztüli kommunikációt biztosít, beékelődve

az alkalmazások és az operációs rendszerek közé. A köztes réteg technológiával lehetővé válik osztozott feldolgozó környezetben valós időben integrálni az egymástól eltérő alkalmazásokat, adaterőforrásokat és folyamatokat, függetlenül azok helyétől és operációsrendszer, hálózati protokoll környezetétől. A kialakított hálózaton belüli együttműködés úgy valósítható meg, hogy a korábban önálló szoftverek, akár egy nagy program tudnak működni. A jól megvalósított összekapcsolásnak köszönhetően a rendszer felhasználói észre sem veszik, hogy az, amiben ők dolgoznak, és amit ők egy nagy vállalati rendszernek hisznek, az valójában több alrendszerből épül fel. Minden alkalmazás képes kommunikálni a többi alkalmazással, de nem egyesével kapcsolják őket össze. A kapcsolatok száma N, azaz annyi kapcsolati vonalat kell csak kiépíteni, ahány alkalmazás a központi szolgáltatáshoz kapcsolódik. Ez a megoldás jóval kevesebb

utólagos ráfordítást igényel, ha új alkalmazás bekapcsolása, vagy régi módosítása válik szükségessé az architektúrában.

Az EAI a köztesréteg technológiákra építve nem csak a vállalat különböző célszoftvereit integrálja, de lehetővé teszi a beszállítók és megrendelők egyszerű csatlakozását a vállalat rendszerébe szabványos interfészeken keresztül.

Integráció Web Szolgáltatókkal

A web egységes architektúrája nem csak mint technológia hanem működését tekintve is megközelíthető. A weben elosztott szoftvervilág és a felhasználók közötti kommunikáció formáját a webszolgáltatások jelentik. Információ, üzleti funkció, kapcsolat, egyéb szolgáltatás webre való leképezése szabványos webszolgáltatások kereteiben érhető el.

A webszolgáltatás modell alapvető célja, hogy az internet igen nagy számú alkalmazásából az egyedi igényeknek megfelelő rendszereket dinamikusan, futásidőben integrálja. Az ily módon dinamikusan integrált osztott alkalmazások szolgáltatásokat alkotnak. A model elemei a szolgáltatás igénylője (service requester), a szolgáltatást nyújtó (service provider) és a szolgáltatásbróker (service broker). Utóbbi az igényeknek megfelelő szolgáltatás azonosítását végzi.

Az együttműködést szabványos technológiai elemek segítik, mint az XML, a SOAP protokoll, az UDDI nyilvántartó rendszer vagy a WSDL (Web Service Definition Language) technológia.

Ezen új szabvány csomagok olyan programok, amelyek automatikusan megkeresik, majd használják a szolgáltatásokat az interneten.

A szabványok segítik a programokat, hogy elérjék a szolgáltatásokat anélkül, hogy hozzákötődjenek egy szerverhez. Ha egy program igényel egy adott szolgáltatást, futási időben lekérdez egy könyvtárat, hogy megtalálja a szolgáltatót. A program részletesen olvassa az üzenet által igényelt formátumot, és konvertálja az adatait a szerver által megkívántra. A szerveret meghívja és visszatér a válasszal.

A felhasznált szabványok, eszközök:

XML - Extensible Markup Language

A HTML-hez hasonló, de annál szigorúbb általános leírónyelv formátum. Meghatározhatjuk, hogy milyen struktúrában tároljuk az adatokat, azaz a tartalom mellett a sémát is mi adhatjuk meg. Az XML-ben tudunk elemeket definiálni ezért például létrehozható a HTML család másolata is.

1996-ban fejlesztette ki a World Wide Web Consortium. Tim Bray internetes szótárt szerkesztett és az állandó formázgatás helyett megalkotott egy struktúrát. Hasonlóan a HTML-hez a használata nem igényel speciális programot. Bármilyen programmal szerkeszthető, csak a kiterjesztés legyen .xml.

A fejlesztés célkitűzései az alábbiak voltak:

- legyen könnyen használható és programozható, hogy gyorsan elterjedjen,
- legyen széleskörűen felhasználható, hogy sok helyen terjedhessen el,
- legyen kompatibilis az SGML-el,
- legyen olvasható a programkód, nem a tömörség a lényeg,
- legyen a szabvány (nem a programkód) formális és tömör.

Felhasználási területei elsősorban az adattovábbítás és az alkalmazások közötti kommunikáció, valamint az adattárolás és archiválás. Mivel egyszerű szöveges állomány világos szerkezettel, ezért könnyedén feldolgozható, vagy olvasható (akár szabad szemmel is). Az XML dokumentumok faszervezetű struktúrája bármely (logikusan) csoportosított adat tárolását lehetővé teszi, ami stíluslap használatával könnyen megjeleníthetővé válik, ezért alkalmas bármilyen adat tárolására.

Az Extensible Stylesheet Language (XSL, kiterjeszhető stílusleíró nyelv) egy támogató technológia, ami leírja a böngészőknek, hogy hogyan kell formázni vagy átalakítani az XML dokumentumban található adatot.

Egyedi dokumentum-típus definíció létrehozásával, saját jelölőnyelv alakítható ki:

- WML - wap oldalakat leíró jelölőnyelv
- VML - vektorgrafika leírására létrehozott jelölőnyelv
- OFX - pénzügyi információk leírására létrehozott jelölőnyelv
- NML - internetes hírcsere

UDDI - Universal Description, Discovery, and Integration

Az **UDDI** (*univerzális leírás, felfedezés és integrálás*) egy platformfüggetlen, XML-alapú nyilvántartó rendszer. Nyílt ipari kezdeményezés (az OASIS szponzorálja), ami lehetővé teszi vállalatok számára, hogy megtalálják egymást és meghatározzák, hogy miként kommunikáljanak az interneten.

Az UDDI a WS-I (Web Services Interoperability) szabványok központi pillére. Úgy tervezték, hogy SOAP üzenetekkel legyen lekérdezhető, valamint hogy hozzáférést biztosítson a WSDL dokumentumokhoz, amelyek a könyvtárban felsorolt webszolgáltatások használatához szükséges protokollkötéseket és üzenetformátumokat írják le.

SOAP - Simple Object Access Protocol

A SOAP egy olyan protokoll, amely alkalmas különféle környezetbeli adatcsere lebonyolítására. Felfoghatjuk egyfajta borítékként, amibe a webes környezetben használt üzeneteket „csomagoljuk”.

Egy SOAP üzenet struktúrája:

```
<SOAP-ENV:Envelope ...>  
  <SOAP_ENV:Header>
```

```
...
...
</SOAP_ENV:Header>
<SOAP_ENV:Body>
...
...
</SOAP_ENV:Body>
</SOAP-ENV:Envelope ...>
```

WSDL - Web Services Description Language

A WSDL webszolgáltatás leírására szolgáló XML formátumú nyelv. A webszolgáltatás nyilvános felületét írja le. Fejlesztésével a World Wide Web Consortium foglalkozik, folyamatosan javítják a verziókat

Az XML-alapú szolgáltatás-leírás a web szolgáltatással történő kommunikációról, a protokoll kötésekről és az üzenet formátumokról tartalmaz információt, amelyek a könyvtárban szereplő web szolgáltatások használatához szükségesek. A támogatott műveleteket és üzeneteket absztrakt módon definiálják és aztán kötik a tényleges hálózati protokollhoz és üzenet formátumhoz.

A WSDL SOAP-pal és XML Schema-val egyetemben nyújt webszolgáltatást az interneten. Egy web szolgáltatáshoz kapcsolódó kliens elolvassa a WSDL-t, hogy megállapítsa a rendelkezésre álló funkciókat a szerveren. A felhasznált speciális adattípusok a WSDL fájlba ágyazva, XML Schema formátumban állnak rendelkezésre. A kliens ezután alkalmazhatja a SOAP-ot a WSDL-ben felsorolt funkciók használatához.

Legfőbb elemek a nyelvben:

- **<portType>** a webszolgáltatás által végrehajtható műveletek
Ez a legfontosabb WSDL elem. Meghatározza a web szolgáltatást végrehajtó műveleteket és a befoglalt üzeneteket.
A programnyelvek egy függvényéhez hasonlítható.

- **<message>** a webszolgáltatás által használt üzenetek
Meghatározza egy művelet adatait. Két fontos paramétere a név (name) és a típus (type).
Egy hagyományos programnyelv függvényhívási paraméteréhez hasonló.
- **<binding>** a webszolgáltatás által használt kommunikációs protokollok
Minden egyes portra meghatározza az üzenet formátumát, valamint a protokoll jellemzőit is tartalmazza. A „kötés” a <portType>-hoz kapcsolódóan a type tulajdonságon keresztül.
- **<service>** a webszolgáltatás által használt átjárókat adja meg
Felsorolja az átjárókat és meghatározza a címeket is. Az ehhez tartozó kötés megegyezik a <portType> névvel.

Ágens-Alapú Modell

A kilencvenes években kialakult ágens alapú programozás az objektum orientáltság megjelenése óta az egyik legfontosabb paradigma. A nyelvészetben az ágens szemantikai kategória, a cselekvő szereplő a mondatban. A szó latin eredetű - ago, agere - melynek elsődleges jelentése: mozgásba hozni, elintézni. A hétköznapi nyelvben a szó angol nyelvterületről érkezett, leszűkült jelentésű “ügynök” fordítást használják.

Az informatikában az ágens rendszer (hardver vagy szoftver) = architektúra + program.
Az ágens program az észleléseket cselekvésre képzi le, miközben belső állapotát frissíti.

Alapvető tulajdonságai:

- Beágyazottság
az ágensek mindig valamilyen környezetbe ágyazottak, abból kiemelve nem tudnak funkcionálni.
- Reaktivitás
az ágensek érzékelik környezetüket és valós időben reagálnak az abban bekövetkezett változásokra.
- Autonómia
az ágensek önállóan, emberek vagy mások direkt beavatkozása nélkül működnek és meghatározott mértékű kontroljuk van a saját akcióik és belső állapotuk fölött. Egy ágens annyira autonóm, amennyire cselekedetei csak saját tapasztalatán alapulnak, és nem a tervező által beépített, a környezetről szóló tudáson.
- Helyzetfüggőség
helyzethez és szerephez kötötten ágensek csupán

További (nem feltétlenül megkövetelt) tulajdonságok:

- Kezdeményezőkézség:
nem csak válaszolnak a környezet változásaira, hanem szükség esetén kezdeményeznek is.
- Célvezérelt viselkedés:
az ágensnek explicit vagy implicit céljai vannak, amelyeket igyekszik végrehajtani.
- Temporális kontinuitás:
az ágens identitása, állapota huzamosabb ideig létezik, illetve működik.

A mesterséges intelligencia a fenti tulajdonságokon felül elvárja az ágensektől a racionalitást is. Feltételezi, hogy az ágens nem fog tudatosan céljai ellen cselekedni, s igyekszik mindig a legjobb alternatívát választani. A tökéletes racionalitás azért az ágensek világában sem létezik. A maximális elvárás az ágensekkel kapcsolatban az,

hogy a rendelkezésre álló számítási kapacitás és egyéb erőforrások mellett a lehető legjobb alternatívát válaszzák.

Az információs ágens olyan program, amely hozzáfér legalább egy, de lehetőség szerint több információforráshoz és képes arra, hogy önállóan összegyűjtse és feldolgozza az azok által nyújtott adatokat. A feldolgozás során képesnek kell lennie az információk önálló szűkítésére a felhasználó igényeinek megfelelően.

A fejlesztők nemcsak ágens technológiával foglalkoznak, hanem hálózatokon és interfészekon átvélő kommunikációval is. Az információs ágensek speciális osztályát alkotják az Internet ágensek (www-ágens), amelyek a világháló különböző adatforrásait dolgozzák fel önállóan, a felhasználó igényeinek figyelembevételével. Egyszerű Internet ágenseknek számítanak a különböző keresőgépek indexelő szoftverei.

Az elektronikus kereskedelemben alkalmazott ágensek feladata a megfelelő árú kedvező ajánlatainak felderítésétől az alkudozáson keresztül egészen az önálló vásárlásig terjedhet. Erre a feladatra használt ágensek információs ágensek ill. legtöbbször Internet ágensek is.

Az interfész ágensek a könnyebb kezelhetőség érdekében sokszor szakítanak azzal a hagyományos megközelítéssel, amely szövegeken, ikonokon és ablakokon keresztül létesít kapcsolatot a felhasználó és a program között. A multi-modalitás jellemző rájuk, azaz a képekkel, hangokkal és egyéb módszerekkel történő kommunikáció. Új kutatási terület pl. a programok gesztusokkal történő vezérlése.

Az ágensrendszerek együttműködését elősegítendő 1996-ban létrehozták a Foundation for Intelligent Physical Agents (FIPA) szabványosítási testületet. A FIPA szabványokat dolgozott ki többek között ágens platform architektúrákra, ágensek kommunikációjára, és tartalom nyelvekre. A szabványosításnak köszönhetően számos vállalat - köztük sok európai is - készíti közös rendszerbe köthető ágens platformot.

Egy másik fontos szabványosítási testület az OMG (Object Management Group), amelyik mobil ágensek menedzselésére dolgozott ki irányelveket.

A mobil számítás koncepciójának lényege egy - akár világméretű - elosztott architektúra, amely számítási környezetekből és végrehajtási egységekből áll. Az alkalmazások végrehajtási egységekből – ágensekből - épülnek fel, melyek a számítási környezetekben futnak. Az újdonság az, hogy a végrehajtási egységek dinamikusan megváltoztathatják a környezetüket, azaz tetszésük

és algoritmusuk szerint vándorolhatnak az elosztott architektúra egyik gépéről a másikra. Ez azt jelenti, hogy a végrehajtási egység a futás tetszőleges pontján felfüggeszti működését, egy másik gépen a felfüggesztést követő utasítástól továbbfut. Ehhez a rendszernek be kell csomagolni a végrehajtási egység kódját és az aktuális program állapotát, valamint gondoskodni kell a csomag utaztatásáról is. A fogadó számítási környezetnek a megfelelő ellenőrzések után vissza kell állítani a program megszakításkori állapotát és a megfelelő utasításra kell adni a vezérlést.

A megoldás csökkenti a hálózati terhelést olyan alkalmazások esetén, ahol nagymennyiségű, elosztottan tárolt adat feldolgozására van szükség, mivel a kód hálózati továbbítása kevesebbe kerül, mint az összes érintett adat utaztatása.

Mobil ágensekkel hatékonyan megvalósítható komplex, szemantikai jellegű keresési feltételek támogatása, amikor az adatok olyan feldolgozása szükséges, mely meghaladja az adatbázis-szerver szolgáltatásainak képességeit. Ezeket az összetett feldolgozási műveleteket az ágens kódja tartalmazhatja, előzőleg szerzett speciális információkra alapozva.

A mobil technológiák az elektronikus kereskedelemben és a chipkártyás alkalmazások területein is megjelentek. Az elektronikus kereskedelemben például a virtuális piacok meglátogatására, és az üzleti tranzakciók előkészítésére használnak sokszor mobil ágenseket.

A webszolgáltatások terjedésével egyre nagyobb az ágensek létjogosultsága. Az ágens segítségével delegálhatunk egy nehéz tranzakciós feladatot. Az ágens fogad egy instrukciót egy programtól, megkeresi a szolgáltatásokat, futtatja őket és visszatér az eredményekkel.

Integrációs környezetben az ágens arra használható, hogy összefogja a különböző rendszereket a kitűzött feladat végrehajtása érdekében.

Az ágens-alapú modell egy alkalmas integrációs mechanizmus. Az automatizált és együttműködő ágensek problémaköre, valamint az ágens rendszerek ipari alkalmazása folyamatosan fejlődik.

Workflow

A workflow adott feladat végrehajtásához szükséges lépések leírása. Az ismertetett integrációs mechanizmusok közül egyedülként emberi interakciót is lehetővé tevő megoldás. A rendszer és a szervezeti szemlélet integrálására együttesen alkalmas.

A Workflow Management Coalition meghatározása szerint a workflow „Részben vagy teljes egészében automatizált vagy számítógéppel támogatott üzleti folyamat”[Hollingsworth, 1995]. Bővebb meghatározásuk alapján a workflow „folyamatok automatizálása, melyek során dokumentumok, információ és feladatok előre meghatározott szabályok alapján továbbítódnak a különböző résztvevők között az üzleti cél elérése illetve az ahhoz való hozzájárulás érdekében”[Hollingsworth, 1995].

A workflow modell a következő elemekből épül fel:

- Szervezeti felépítés:

A hierarchikusan felépülő szervezeti struktúrában a különböző szervezeti egységek között alá- és fölérendeltségi viszonyok vannak. A szervezeti felépítés ezeket a viszonyokat, a szervezeti egységek közötti kapcsolatokat írja le. A különböző szervezeti egységekhez különböző pozíciók tartoznak, melyeket az alkalmazottak töltenek be. Közöttük szintén alá- és fölérendeltségi viszonyok vannak.

- Hatáskörök:

Meghatározzák, hogy az egyes pozíciókban lévő alkalmazottak milyen feladatokat látnak el, meddig terjed a hatáskörük.

– Tevékenységek, feladatok:

A folyamat különböző tevékenységekre bontható. Pontosan meg kell határozni az egyes tevékenységeket, azok határait, az ott elvégzendő feladatokat. Meg kell határozni, hogy az egyes feladatok milyen feltételek esetén hajthatóak végre, végrehajtásukhoz milyen adatok szükségesek, és elvégzésük után milyen kimenő adatokat szolgáltatnak a folyamat további tevékenységei számára.

– Tevékenységvezérlés:

A teljes ügyviteli folyamatot leíró modell. Legjobb szemléltetése egy folyamatábra, amely leírja, hogyan követik egymást az egyes tevékenységek, mely feltételek esetén kell az egyes feladatokat végrehajtani.

– Adatáramlások:

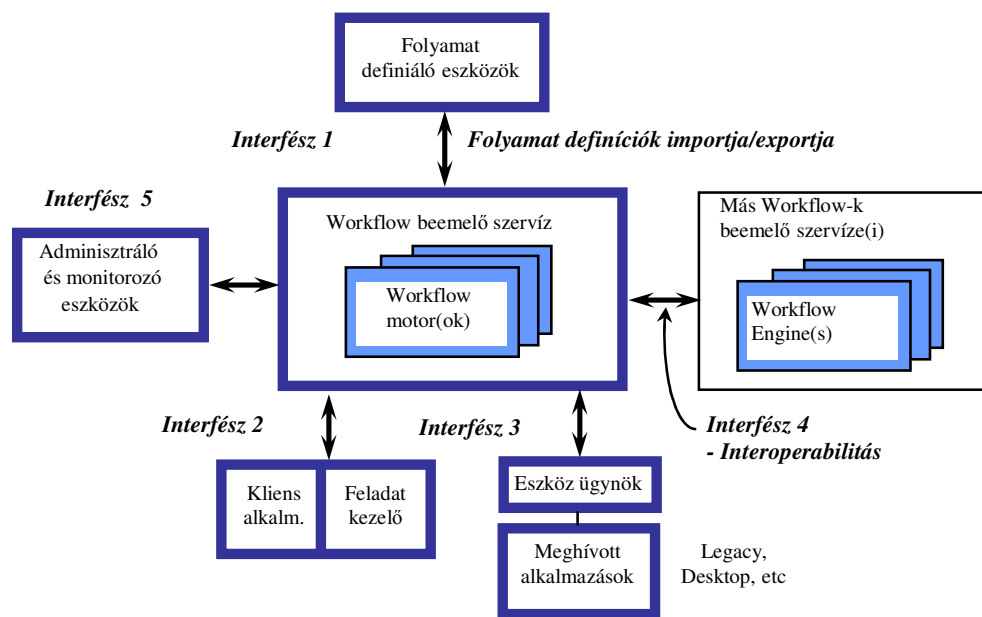
A folyamat modell meghatározza, hogy az egyes tevékenységek milyen adatokat igényelnek, valamint milyen adatokat szolgáltatnak. Ezen felül a folyamathoz is kapcsolódnak adatok, melyek a teljes folyamatot végigkísérik, annak során változhatnak. Az adatáramlások írják le, hogy a folyamat milyen kezdőadatokkal indul, azok honnan származnak, valamint a folyamat milyen adatokat szolgáltat az egyes tevékenységekhez, és milyen adatokat kap azoktól vissza. Egyes információk származhatnak a folyamattól független környezeti elemektől is.

– Eljárások:

Az egyes tevékenységeket, feladatokat különböző eljárások, algoritmusok alapján hajtják végre. Ezek dolgozzák fel a bemenő adatokat és szolgáltatják a kimenő információt.

- A workflow egyes lépéseinek kidolgozása során folyamatosan bővül a dokumentáció, amely szintén része a rendszernek. A pontos dokumentáció segítségével az egyes hibák azonosítása, a modell finomítása, megváltoztatása, átszervezése könnyebbé, gyorsabbá válik.

A Workflow Management Coalition referencia modellje:



4. ábra A WfMC referencia modell [WfMC, 1995]

A workflow rendszer részei és interfészei:

A modell középpontjában a workflow motorok állnak, amelyek az interfészeket keresztül hajtják végre a különböző feladatokat, mint pl. magának a munkafolyamatnak a levezénylése, adatok gyűjtése a lezajlott folyamatokról, együttműködés más workflow programokkal, stb.

Interfész 1

Az interfész a folyamat definiáló tervező eszközöket kapcsolja a rendszerhez. A szervezet folyamatait formális modellező eszközzel lehet megtervezni az üzleti modellek felhasználásával.

Interfész 2

Az interfész a folyamatban részt vevő emberi szereplők programjait kapcsolja a rendszerhez - ezen keresztül végzik az ügyintézők a feladataikat.

Interfész3

Az interfész az adott munkaegység elvégzéséhez szükséges, a külső alkalmazásokat hívja meg megfelelően paraméterezve.

Interfész4

Az interfész az interoperabilitásért felelős - a külső workflow rendszerekkel veszi fel a kapcsolatot, átad feladatot ill. fogadja azok válaszait ill. kéréseit.

Interfész 5

Az interfész hozzáférést enged a rendszer log-file-jaihoz, ezáltal ellenőrizni, elemezni lehet a folyamatok végrehajtását.

Szisztematikus Megközelítések

A bemutatott módszerek mellett akadémiai kutatások az integráció megvalósítására szisztematikus, technológiai szempontú megközelítéseket javasolnak. A széles körben elterjedt alkalmazás szintű együttműködést elősegítő technológiák irányába folyó kutatások legfőbb területei az ágens kutatások, a grid, a web szolgáltatások és a szemantikus web. A szoftver architektúra nézőpontjából az integráció nem az üzleti problémák megoldásáról szól, hanem arról, hogyan kapcsolódjanak egymáshoz különböző dolgok.

Elsőként Mary Shaw mutatott rá arra, hogy a rendszerek közötti interfész mindig másodlagos szempont, holott ezen kapcsolatoknak elsődleges szerepe van az integráció megvalósításában. [Shaw, 1996].

Chris Dellarocas ezzel szemben a komponensek egymástól való függőségét helyezi előtérbe. [Dellarocas, 1996]

Ezek az irányzatok nem igazán váltak be, túl bonyolult megjósolni az összes lehetséges kapcsolatot az üzleti tevékenységek között. Sokkal valóságosabb feltételezés, hogy a szoftver komponensek előre nem látható módokon fognak kombinálódni. Egy komponens legyen explicit, hogy a kompatibilis funkciók további befektetés és tesztelés nélkül is kombinálhatóak legyenek.

1994-ben MIT (MIT Center for Coordination Studies) kutatási program keretében kísérletet tettek arra, hogy az üzleti problémákat módszeresen leképezzék integrációs mechanizmusok halmazára. [Malone, Crowston, 1994]

Katalogizálták az üzleti folyamatokat, és az integrációs mechanizmusait, feltételezván, hogy a vállalati tevékenységeket folyamatosan lehet finomítani. 1999-re elkészült egy folyamatkézikönyv. [Malone et al., 1999]

Egy “top-down” törekvés keretében nemzetközi integrációs szabvány kialakítására is sor került: GERAM– Generalized Enterprise Reference Architecture and Methodology [Chalmeta, Campos, Grangel, 2001].

Kérdéses, hogy bármely generalizált architektúra működne. Szintén nem világos, hogy egy folyamat kézikönyv valaha is teljes lehetne. Sem az akadémiai elméletek, sem az ipari architektúrák nem hatnak olyan direkt módon a vállalatokra, mint a szállítói termékek és a belső IT csapat.

A gyakorlati tapasztalat az, hogy az architektúra elképzelések megtalálják az utat, hogy beépüljenek a szoftver termékekbe. Mindenképpen szükséges, hogy az integrációs keretrendszerek be tudjanak fogadni egyre több különböző technikai és szervezeti mechanizmust az újabb és újabb integrációs probléma természetének megfelelően.

Ehhez meg kell találni a megfelelő egyensúlyt a szabványosítás kötöttségei és az alulról jövő építkezés között. Ahogy a TCP/IP jobban elterjedt, mint az ISO-OSI szabványok, illetve ahogy a HTML jobban elterjedt, mint az SGML, ugyanúgy ezeknél a szabványoknál is fontos, hogy minél több alkalmazói próbálkozás legyen, és a felhasználók motiváltak legyenek, hogy másokkal megosszák a saját, adott területre vonatkozó tudásukat. Ha már kellően elterjedt a technológia, akkor a vállalkozások is profitálnak belőle.

II.3.3. A Vállalati Integráció Folyamata

Az a vállalat, amelyik nem kapcsolódik az internetre, komoly versenyhátrányban van. Az osztott, hálózatszervezetek olyan világban tevékenykednek, ami sokkal gyorsabban változik, mint eddig. Az egyedi igények kielégítéséhez a vállalatoknak az internetről érkező információt a belső informatikai rendszerükbe kell átvezetniük, ami kihat a belső rendszerükre. A vállalat csak akkor tud alkalmazkodni az igényekhez, ha ugyanezt megköveteli beszállítóitól is, ami kihat a vállalatok közötti kommunikációra is.

Ilyen változások mellett egyre kevésbé beszélhetünk különálló szoftver termékekről, a rendszerek egyre több szálon kapcsolódnak egymáshoz, és minden szoftver rendszernek valamilyen módon együttműködésre képesnek kell lennie más szoftver rendszerekkel. Ilyen környezetben a vállalati informatikai rendszerek is a világháló részévé válnak, és egyre kevésbé lehet önálló megoldásokat alkalmazni. A szoftvertechnológia egyre kevésbé jelenti az egyedi rendszerek megtervezését és megvalósítását, helyette egyre inkább egy összefüggő világméretű elosztott rendszer közös fejlesztéséről beszélhetünk. Ebben a környezetben az egyes részrendszerek fejlesztői nem alkalmazhatnak egyedi megoldásokat, figyelembe kell venni a kialakulóban lévő szabványokat. A tervezéskor nem látható át, és nem tervezhető a teljes világméretű rendszer, az adott rendszer tervezőjének olyan komponensre kell terveznie, amelyiknek a működése során a részleges információk birtokában a nem várható változásokra is felkészülve kell a világméretű rendszerbe integrálnia.

Akkor egyszerű az integráció, ha még semmilyen rendszer sem működik. Az esetek többségében azonban egy hatalmas és komplex infrastruktúráról van szó, az informatikai alkalmazásokat egy sor meglévő rendszerre, szolgáltatásra és üzleti folyamatra kell ráépíteni, illetve azokkal integrálni.

Szükséges egy olyan platform, amihez könnyen lehet kapcsolni az egyes rendszereket, alkalmazásokat, valamint figyelembe kell venni az üzleti integrációs modell minden rétegét!

A fő feladat a keretrendszer megalkotása. Az integrált rendszerek építésének folyamata ugyanaz, mint egyedi alkalmazás esetén és a megfelelő módszertanok használata itt is javasolt.

A tervezés az alkalmazás integrációnál is alapvető fontosságú. Az egyes rendszereket kreatívan lehet építeni a definiált szerkezeten belül, ha a szabványok léteznek, a folyamatok definiáltak, és kész a terv. Az üzleti alkalmazások csak hatékony üzleti konfigurációval válnak üzleti megoldássá.

A vállalati architektúra kialakítása során lehet támaszkodni szabványos architektúra keretrendszerekre, mint a Zachmann (az architektúra keretrendszerek őse) [Zachman, 1987], DODAF, TOGAF, stb.¹. Nyílt rendszeri architektúrára építkezve más területek piacvezető termékeivel is könnyen lehet kapcsolódni a teljes vállalati életciklus támogatása során. Ha több nagy szervezet kommunikál egymással, a szabványos megoldás nagyban megkönnyíti egymás megértését is. További előny, hogy a keretrendszerekkel járó egyéb funkciókat, szolgáltatásokat is igénybe lehet venni, mint pl. változáskezelési folyamat, avagy pl. a a TOGAF-nál elérhető certifikációs jellegű szolgáltatások.

¹ DODAF - „Department of Defense Architecture Framework” az Amerikai Védelmi Minisztérium keretrendszere.

TOGAF - „The Open Group’s Architecture Framework” nemzetközi, nyílt szabványú keretrendszer és módszertan egy szervezet felépítésének fejlesztésére. Tartalmaz egy jól meghatározott és bizonyított, lépésről lépésre haladó módszert a vállalati architektúra felépítéséhez, és fenntartásához. Ezt ADM-nek (Architecture Development Method - Architektúra Fejlesztési Módszer) hívják, és a négy fő architektúra területet fedi le: az üzletet, az információs rendszereket (alkalmazásokat és adatokat), valamint a technológiai környezetet. Az ADM erősen fókuszál az üzleti célokat és követelményeket támogató architektúra előállítására.

	WHAT	HOW	WHERE	WHO	WHEN	WHY	
	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION	
SCOPE (contextual)	List of Things Important to the Business Entity = Class of Business Thing	List of Processes the Business Performs Process = Class of Business Process	List of Locations in Which the Business Operates Node = Major Business Location	List of Organizations Important to the Business People = Major Organizational Unit	List of Events/Cycles Significant to the Business Time = Major Business Event/Cycle	List of Business Goals/Strategies End/Mean = Major Business Goal/Strategy	SCOPE (contextual)
Planner							Planner
BUSINESS MODEL (conceptual)	e.g., Semantic Model Entity = Business Entity Relationship = Business Relationship	e.g., Business Process Model Process = Business Process IO = Business Resource	e.g., Business Logistics System Node = Business Location Link = Business Linkage	e.g., Work Flow Model People = Organization Unit Work = Work Product	e.g., Master Schedule Time = Business Event Cycle = Business Cycle	e.g., Business Plan End = Business Objective Means = Business Strategy	BUSINESS MODEL (conceptual)
Owner							Owner
SYSTEM MODEL (logical)	e.g., Logical Data Model Entity = Data Entity Relationship = Data Relationship	e.g., Application Architecture Process = Application Function IO = User Types	e.g., Distributed System Architecture Node = IS Function (Processes, Storage, etc.) Link = Data Characteristics	e.g., Human Interface Architecture People = Role Work = Data/Work	e.g., Processing Structure Time = System Event Cycle = Processing Cycle	e.g., Business Rule Model End = Structural Assertion Means = Action Assertion	SYSTEM MODEL (logical)
Designer							Designer
TECHNOLOGY MODEL (physical)	e.g., Physical Data Model Entity = Segment/Table/etc. Relationship = Primary/Foreign/etc.	e.g., System Design Process = Computer Function IO = Data Element/Set	e.g., Technology Architecture Node = Sub-System Software Link = User Specifications	e.g., Presentative Architecture People = User Work = Screen Formats	e.g., Control Structure Time = Event Cycle = Component Cycle	e.g., Rule Design End = Condition Means = Action	TECHNOLOGY MODEL (physical)
Builder							Builder
DETAILED REPRESENTATIONS (out-of-context)	e.g., Data Definition Entity = Field Relationship = Attribute	e.g., Program Process = Language Statement IO = Control Block	e.g., Network Architecture Node = Address Link = Protocol	e.g., Security Architecture People = Identity Work = Job	e.g., Timing Definition Time = Interval Cycle = Machine Cycle	e.g., Rule Specification End = Sub-condition Means = Step	DETAILED REPRESENTATIONS (out-of-context)
Subcontractor							Subcontractor
FUNCTIONING ENTERPRISE	e.g., DATA	e.g., FUNCTION	e.g., NETWORK	e.g., ORGANIZATION	e.g., SCHEDULE	e.g., STRATEGY	FUNCTIONING ENTERPRISE

Zachmann architektúra keretrendszer [www.zifa.com]

A vállalati architektúra megoldás célja a vállalati működés megértésének támogatása, az elkülönült technológiák integrált és összetartó együttesének kialakítása és a vállalati célokhoz történő igazítása.

A vállalati architektúra - “leíró reprezentációk, modellek halmaza, amely egyrészt lényeges a vállalat működése szempontjából, másrészt a menedzsment elvárásai szerint állítható elő, és aktuális állapotban tartható a vállalat működése során.” [Zachman, 1992]

A vállalati architektúra (rendszer) a kulcsterületeket - üzlet, információ, rendszerek, és technológia- magába foglaló modellek és dokumentumok integrált gyűjteménye.

A vállalati architektúra eszközök legfontosabb jellemzői:

- Integrált formában, egyetlen repozitóriumban tárolja a vállalat üzleti tevékenységéről, alkalmazásairól, adatairól, és technológiájáról a releváns információkat.
- Metamodellel rendelkezik a tárolt információ strukturálásához.
- A repozitóriumban tárolt információ grafikus és szöveges megjelenítését többféle formában is lehetővé teszi.

Egy skálázható vállalati architektúra megoldás közös munkakörnyezetet biztosít a vezetők számára, hogy megértsék, hogyan fejleszthetik a szervezet felépítését, és javíthatják az üzleti folyamatokat úgy, hogy eközben

- Nőjön a szervezet reagálási képessége.
- Az üzleti folyamatok, és informatikai rendszerek az üzleti célokkal összhangba kerüljenek.
- Tervezzék, modellezzék, és végrehajtsák az üzleti folyamataikat.
- Gyorsan és hatékonyan válaszolhassanak az üzleti környezet változásaira.

A vállalati architektúra kifejlesztése során elkészített szervezeti terv javítja az üzleti minőséget, a hatékonyságot, és a számonkérhetőséget. Egy jól kifejlesztett vállalati architektúra lehetővé teszi a szervezet számára, hogy gyorsan és hatékonyan válaszoljon a kínálkozó lehetőségekre és kihívásokra, amelyeket a piaci változások, átrendeződések és a technológiai fejlődés támasztanak. Segíti az üzleti célok elérését, valamint az üzleti tevékenység és az alkalmazott IT összhangját.

A korszerű vállalati architektúra tartalmazza:

- Az üzleti folyamatok, információk, rendszerek, és technológiák modelljeit.
- A grafikus és szöveges elemi adatok leírását, amelyekből ezek a modellek felépülnek.
- A köztük levő kapcsolatok világos leírását.
- Teljes nyomon-követhetőséget a szervezet céljaihoz és célkitűzéseihez - ez a fő értelme az architektúra kifejlesztésének.
- Az architektúra tartalmi és prezentációs keretét adó szabványokat és elveket.

A vállalati architektúra, melyet egy átfogó és teljesen integrált repozitóriumban tárolunk és tartunk karban, a vállalat alapvető értékének számít.

II.4 Új rendszerfejlesztési paradigmák

II. 4.1. Modell vezérelt fejlesztés

A szoftverfejlesztési folyamat általában az alábbi lépésekből és termékekből áll:

1. Specifikáció
 - követelmények feltárása és elemzése
 - követelmény dokumentum előállítás – végfelhasználóknak, fejlesztőknek
 - megvalósíthatósági tanulmány
2. Tervezés
 - üzleti modell: a megvalósítandó rendszert a megvalósítási részletektől függetlenül, az alkalmazási terület fogalmaival magas szinten írja le, többnyire szöveges leírással kiegészítve.
 - technikai modell: a megvalósításra kiválasztott technológiáknak megfelelő fogalmakkal írja le a rendszert, esetenként szöveges leírással kiegészítve
3. Implementáció
 - a rendszer specifikációjának futtatható rendszerré alakítása
 - kódolás
4. Verifikáció, Validáció
 - tesztelés
 - annak igazolása, hogy megfelel az ügyfél kívánásainak
 - teszt tervek, tesztelési dokumentumok
 - felhasználói dokumentumok
5. Evolúció
 - üzemeltetés
 - új követelmények beépítése
 - változtatások végrehajtása

A klasszikus vízésés modell alkalmazásának az lehet az eredménye, hogy a megoldások nem felelnek meg az üzleti elvárásoknak. A modellben élesen elkülönülnek az elemzési, tervezési és megvalósítási fázisok, valamint nincs közöttük visszacsatolás. Az üzleti

elemző által leírt funkcionális specifikációról sokszor csak a megvalósítási fázisban derül ki, hogy technikailag kivitelezhetetlen. Az eredmény csak átvételkor válik láthatóvá az üzlet számára, és a javítás már csak súlyos határidőcsúszással lehetséges.

A mai szoftverek nagy része folyamatos fejlesztés és karbantartás alatt álló termék. A fejlesztés során újabb követelmények kerülnek színre, korábbi követelmények finomulnak, megváltoznak. Ilyenkor a követelményelemzéshez kell visszatérni, majd a lépéseken újra végigmenni. Olyan modellre van szükség, amely ezt a folyamatos iterációt magában foglalja. A spirálmodell már magában foglalja a folyamat ciklikus voltát azzal, hogy folyamatos visszacsatolást biztosít a fejlesztési folyamathoz és a rendszerhez egyaránt.

Az inkrementális modellt alkalmazva lehetőség nyílik a különböző részek egyenkénti kifejlesztésére és tesztelésére. A rendszer problematikusnak vélt részei külön prototípusként elkészíthetőek, így a fejlesztés korai szakaszában felderíthetőek és kiküszöbölhetőek az esetleges követelménybeli vagy technikai jellegű problémák. Az inkrementális modellben tehát nem folyamatosan halad előre a fejlesztési folyamat, mint pl. a vízésés modellben, hanem bizonyos részekkel előre szalad, hogy képet kapjunk a rendszer későbbi viselkedéséről.

Az állandóan változó igények okozta feszültségek megszüntetésére jó módszer lehet, ha felosztjuk a projektünket kisebb részprojektekre, amelyek befejezése a szoftver egy működő szeletét eredményezi. Ez a gyakori „szállítás” a megrendelőknek biztonságot nyújt, mivel állandó betekintést kapnak a fejlesztés menetébe. A rendszeres szállítás és az azt követő konzultáció maga után vonja a specifikáció finomodását, a hibák kijavítását és a szoftver egyre hatékonyabbá válását. Éppen ezért érdemes az agilis fejlesztés során rövid, néhány hetes részprojekteket definiálni, melyeket minden alkalommal a specifikáció újragondolása és a prioritások ártértékelése követ. Ez adja a projekt agilitásának az egyik felét [Cockburn, A., 2002].

Az agilitás másrészt a dokumentálás egy részének elhagyásából adódik - „a kód maga a dokumentáció”. Ez megköveteli a csapat tagjaitól egymás folyamatos tájékoztatását a

projekt menetéről, a különböző tervezési, követelménybeli, stb. változásokról. Minden projektben fontos szerepe van a csapat tagok közötti tacit tudásnak, de agilis fejlesztés esetén ez különösen hangsúlyos. Azzal, hogy a dokumentációkat nem szükséges állandóan frissíteni, jóval több változás eszközölhető rövid idő alatt.

Az extrém programozás (XP) és a dinamikus rendszerfejlesztési módszertan (DSDM) az agilis fejlesztési metodika képviselői, amelyek nem feltétlenül egymás alternatívái, inkább kiegészítői.

A fejlesztési folyamat első három lépése általában „kézzel”, emberi alkotó folyamat során történik. Ennek megfelelően azok termékei csak tervezési és dokumentációs célokat szolgálnak. Bár a feladat megértését segítik, a fejlesztés menetét inkább lassítják.

A technikai modellből viszonylag egyszerűen lehetne kódot generálni, mivel a megfeleltetés legtöbbször egyértelmű. Például több UML modellező eszköz képes valamilyen célnyelven (pl. C++, Java) kódot generálni. Ugyanakkor a technikai modell bonyolultsága szükségszerűen hasonló a kódéhoz, vagyis ennek létrehozása nem feltétlenül teszi jobban átláthatóvá a rendszert, sem rövidebbé a teljes fejlesztési időt.

Gyakori, hogy a változásokat egy idő után csak a forráskódban, vagy a technikai modellben hajtják végre, az első két-három lépést kihagyva. Emiatt az előkészítő lépések termékei nem lesznek összhangban az alkalmazással. Így a modell többé már nem hasznos eszköz, még dokumentációs célját sem tölti be, a közvetlen kódoláshoz képest tulajdonképpen nem sokat lehet nyerni vele. Ez magyarázza, hogy sokan az agilis, gyors módszereket választják inkább.

A vizuális modellezés lehetőségeit a szoftverfejlesztési folyamatokban jóval aktívabban is ki lehetne használni. Jóllehet a programkódot ma még nem tudjuk tisztán modellekből generálni, számos hasznos modell alapú segédeszköz létezik. Például a modellekből történő automatikus kódgenerálás, a kódvisszafejtéses modellek létrehozása, a modellkód szinkronizálása mind olyan, már rendelkezésre álló lehetőségek, amelyek

nagyban képesek felgyorsítani egy-egy konkrét, jól körülhatárolt terület szoftverfejlesztését.

A modell alapú fejlesztés a leginkább kutatott szoftverfejlesztési módszerek egyike, amelynek célja a fejlesztés hatékonyságának és az elkészült termékek minőségének növelése.

A vállalat-központú alkalmazásfejlesztési folyamat életciklusa alapvetően nem különbözik a hagyományostól, lényegében ugyanazokon a fázisokon kell végigmenni. A modellszemléletű, vállalat-központú szoftverfejlesztési munkánál első lépésként a valós rendszer működését kell leírni. A felhasználó igényeinek megfelelő rendszer egy véges modelljét kell megalkotni létrehozva a CIM-modellt (Computational Independent Model). Ennek a fázisnak a terméke egy olyan dokumentum, amely pontosan meghatározza a valós rendszer elemeit és funkcionalitását leíró doménmodellt, a fejlesztendő alkalmazással szemben támasztott elvárásokat, a felhasználói követelményeket.

Az OMG (Object Management Group) által támogatott MDA (Model Driven Architecture) kezdeményezés célja a formális módszerek bevezetése már a rendszerspecifikáció szintjén. Az MDA az „üzleti” és a „technikai” helyett a „platform-független” (PIM) (Platform-Independent Model) és a „platformspecifikus” (PSM) (Platform-Specific Model) megnevezéseket használja. A PIM és a PSM szinteket adott platformhoz viszonyítva relatívan kell értelmezni. A modellalkotás absztrakciós folyamata során egyrészt el kell hagyni a feladat szempontjából lényegtelen jellemzőket, másrészt pontosítani és formalizálni a lényegeseket, állandóan szem előtt tartva a feladatot majd megoldó eszköz tulajdonságait és képességeit.

A PIM kialakítása során platform független nyelvet kell használni. A PIM modell egy olyan leképezés eredménye, amely a támogatandó rendszert annak leghatékonyabb megoldása szempontjából specifikálja. A modell meta adatokat tartalmaz a valós rendszer elemeiről, ezek kapcsolatairól, struktúrájáról, az elemek viselkedéséről, feladatairól és interfészeiről, valamint az elemek között fennálló függőségi viszonyokról.

A fázis terméke a fejlesztési folyamat legmagasabb szintű absztrakciójának az eredménye, amely független mindenféle alkalmazott technológiától.

A PIM modell létrehozása iteratív folyamatban történik, az elemek sajátosságait fokozatosan kell finomítani - definiálás -> felülvizsgálat -> kiértékelés -> javítás.

A platform független modell tesztelhető, azaz ellenőrizhető a szemantikai helyessége, a rendszer funkcionalitásának való megfelelés, valamint a működőképesség.

Az informatikai modell a számítógépen tárolható adatszerkezetek, adatstruktúrák és az ezeket kezelő, átalakító, ezekből információkat lekérdező algoritmusok összessége, rendszere. A platform független leírás az alkalmazandó informatikai környezet figyelembevételével formális szabályok alapján lefordítható végrehajtható programmá. A PSM specifikációk elkészítéséhez az egyes PIM elemekhez hozzá kell rendelni a megvalósítás technológiáját. Egy PIM modelltől több PSM modell is kialakítható. A modellellenőrzés során a rendszer egy korábban megalkotott modelljéről kell bizonyítani, hogy teljesülnek rá a kívánt tulajdonságok. A gyakorlati tervezés általában egy iteratív, javító, módosító lépéseket is magában foglaló folyamat.

Az utolsó lépés a PSM modell megvalósítása, vagyis az implementációs és tesztelési folyamat végrehajtása. Két kérdéskört alaposan végig kell járni: a megrendelő igényeinek megfelelő, jó rendszer készült - validáció, illetve a fejlesztés során kapott különböző mélységű modellek ekvivalensek, azaz jól készült a rendszer - verifikáció.

A modell végessége elvileg garantálja, hogy az algoritmus futása előbb-utóbb véget ér. A modellek matematikai módszerekkel egymásnak megfeleltethetők. A matematikai módszerek adott modell érvényességén belül 100% valószínűséggel bizonyítják a struktúra helyességét.

Az épülő vagy kész rendszer jóságát tervezett teszteléssel is vizsgálhatjuk, azonban a legalaposabb tesztelés sem képes a teljes problématerát átvizsgálására, ezért sohasem adhat 100%-os bizonyosságot a rendszer helyes működéséről.

Általában a rendszert véges automaták formájában írjuk le. Az automata ugyanúgy matematikai modell, mint bármely más formalizmus, azonban rendelkezik azzal az előnnyel, hogy „ránézésre” sok minden leolvasható belőle.

Az automata mindig valamilyen meghatározott állapotban van – a lehetséges állapotok közül mindig egy, és csak egy érvényes.

A modellellenőrzési módszerek legfőbb veszélye az állapottér robbanása - a probléma bonyolultságának növelésével az állapotok és állapotátmenetek száma exponenciálisan nő. A mai minimalizáló és redukciós automatizált, gépi eljárások segítségével $\sim 10^{120}$ elérhető állapot is vizsgálható.

A szoftverfejlesztés az MDA keretrendszerben tehát nem más, mint transzformációk egymásutánja:

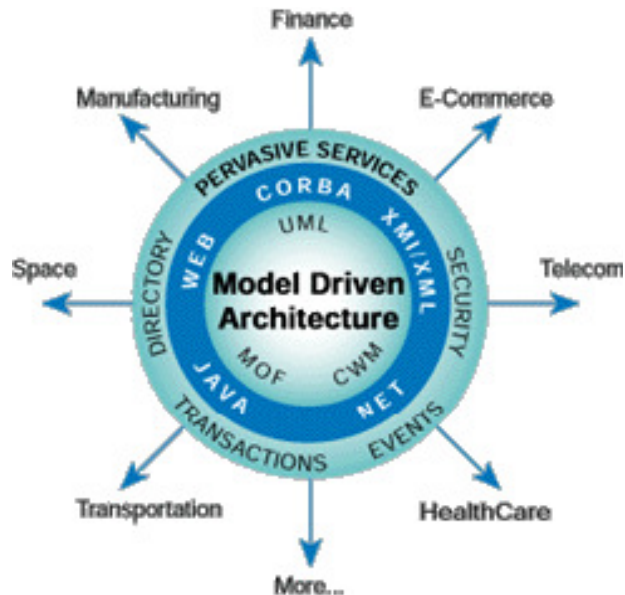
- CIM \rightarrow PIM
- PIM \rightarrow PSM
- PSM \rightarrow kód

Az elérendő cél az, hogy az emberi tevékenység az üzleti modell elkészítésével érjen véget, majd abból a technikai modell és a forráskód generálása automatikus legyen. Ezáltal az üzleti logika függetleníthető az implementációs technológiától és a fejlesztés magasabb absztrakciós szinten folyhat.

A módszer alkalmazása hagyományos fejlesztési modell említett problémáinak megoldása mellett az alábbi előnyöket jelenti:

- A modellezés célja nem csak a dokumentálás, hanem a fejlesztés alapvető eszköze.
- A fejlesztők a célplatform technikai részleteiben való elmerülés helyett az alkalmazási területre koncentrálhatnak.
- A célplatformra való transzformáció definíciójának változtatása az egész alkalmazásra kihat, így a hibajavítás és a továbbfejlesztés egyszerűbbé, az alkalmazás karbantarthatóbbá válik.

- Az alkalmazási platform - a rendelkezésre álló transzformációk keretein belül - tetszőlegesen megváltoztatható.



5. ábra Modellvezérelt architektúra [OMG, 2002]

Az MDA alapfogalmait az OMG szabványai és ajánlásai határozzák meg:

- UML: (Egységes modellező nyelv) az objektum-orientált-, valamint a komponens alapú elemzés és tervezés szabvány nyelve.
- Meta Object Facility (MOF): modellező nyelvek leírására szolgáló, önleíró modellező nyelv (pl. az UML-t is MOF-ban definiálták) A szabvány technológiai megoldást biztosít metaadatok definiálására olyan fejlesztésekben, amelyek a domén modellt objektumosztályokká és viselkedésmodellekké képezik le. A MOF további célja, hogy az együttműködésre kényszerülő meta modellek manipulálásához köztesréteg-interfészeket biztosítson, és a közös meta modellek specifikálásával lehetővé tegye az osztott környezet alkalmazásainak az együttműködését, a különböző platformon futó alkalmazások kommunikációját. Egy adott rendszerben számos meta adattípus lehetséges, amelyeket egységes meta modellként kell tudni kezelni. Ehhez megoldásként a szabvány modellező nyelvi definíciós mechanizmust - egy meta adat-keretrendszerben definiált absztrakt szintaxist kínál.

A MOF Metaadat architektúrája:

M3: meta-meta modell szint – absztrakt nyelv formájában kifejtett információkat tartalmaz a metamodellekre vonatkozóan.

M2: meta modell szint - a modellek sajátosságait leíró információkat tartalmazza. Definiálja a modellkomponenseket, ezek struktúráját és szemantikáját. Egy absztrakt nyelv a különböző modellek és modellnézetek leírására, amely modellkomponenseket, struktúrákat és szemantikát definiál.

M1: modell szint - információkat tartalmaz a valós folyamatokról és elemekről. Ezeket a leíró jellemzőket informális módon integráljuk metamodellekbe.

M0: információs szint - a vizsgált valós rendszer sajátosságait írja le, annak elemeit és viselkedését tükrözi (adat-, illetve objektum réteg).

[OMG, 2002]

- XML Metadata Interchange (XMI): modellek XML formátumú reprezentációja, eszközök közötti megosztás céljára. Heterogén, osztott környezetben lehetővé teszi a metaadatok, a modellező eszközök és a metaadat-repositoryk közötti forgalmat.
- Common Warehouse Metamodel (CWM): szabványosítja mindazokat a metamodelleket, amelyek a vállalatok különböző adatbázisainak az együttműködését, az adatbázisok közötti átjárhatóságot lehetővé teszik. Így ki lehet használni a vállalatoknál felhalmozódott adatvagyonban rejlő lehetőségeket.

Az integráció szempontjából az MDA „vízió” tehát az alábbi jelenti:

Támogatni az integrációt célzó specifikációk együttműködését a rendszerek teljes életciklusán keresztül, az üzleti modellezéstől a rendszertervezésen, a komponensek létrehozásán, összeintegrálásán, telepítésén, menedzselésén keresztül egészen azok evolúciójáig.

Az alkalmazott modellező nyelveket MOF-terminológia szerint kell megadni, lehetővé téve, hogy a meta adatokat szabványos módon lehessen értelmezni, ami az automatikus transzformációk előfeltétele.

Az MDA architektúra lapvető célja, hogy modellszinten biztosítsa a különböző elemek együttműködését, függetlenül az alkalmazott technológiáktól és az alkalmazási platformoktól.

II. 4.2. SOA

A szolgáltatás olyan ismételhető funkció, amely meghívásakor elvégez valamilyen meghatározott tevékenységet (pl. operációs rendszer funkció, saját fejlesztésű üzleti logika/művelet vagy „dobozos” alkalmazás egy modulja, stb.)

A szolgáltatásokat „fekete dobozoknak” tekinthetjük, mivel elrejtik implementációjuk részleteit. Az implementáció egy vagy több funkció megvalósítása - „maga a kód” - jelenthet egy, vagy több komponenset. Az illesztő felületeiket megfelelően kell definiálni. Az interfész protokoll-független, metódusokat, a be- és kimeneti paramétereket és adattípusaikat tartalmazza. Egy komponens több interfészt is implementálhat. A végpontok kötik az interfészt protokollhoz, fizikai címekhez.

A szolgáltatások meghívása nyílt szabványú mechanizmusokon keresztül történik. Egyaránt lehetnek elemi vagy összetett szolgáltatások. Megvalósításának egyik jó példája a web service.

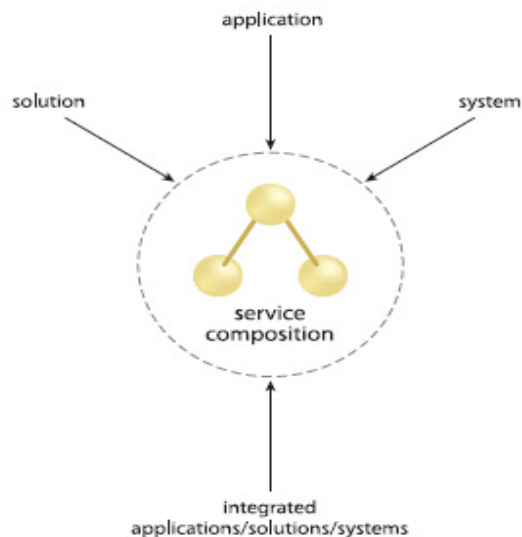
A SOA megszületését a gyakorlatban egy időre teszik a webszolgáltatások térhódításával, azonban jóval túlmutat az egyszerű web services alapú kapcsolatokon – az üzleti folyamatok hatékony támogatását célozza a vállalat jelenleg is meglévő alkalmazásainak segítségével.

A szolgáltatás orientált architektúra újrafelhasználható komponensekre, modulokra, szolgáltatásokra támaszkodik, amelyek az alkalmazásoktól és a futtató platformoktól függetlenül is meghívhatók.

A webszolgáltatások a hozzájuk kapcsolódó elérési protokoll (SOAP) és az interfészük leírására szolgáló nyelv (WSDL) segítségével biztosítják, hogy olyan építőköveket hozzunk létre és használjunk, melyek platformtól és programozási nyelvtől függetlenül elérhetőek egymás számára a Weben vagy a vállalati architektúrán keresztül.

A szolgáltatások tehát képesek egymás elérésére, de lényeges kérdés, hogy pontosan hogyan lehet biztosítani az építőkövek összekapcsolódását.

A SOA közvetlen előfutára az EAI, a nagyvállalati alkalmazás-integráció, amely köztesréteg technológiákra építve integrálja a vállalat különböző szoftvereit, egységes kezelést biztosítva a különböző rendszerek által közösen használt adatoknak, valamint lehetővé téve a partnerek egyszerű csatlakozását a vállalat rendszerébe szabványos interfészekon keresztül. A funkciók és erőforrások szolgáltatás orientált elvek szerinti távoli elérése viszonylag új megközelítést jelent az összetett alkalmazások építésénél.



6. ábra Szolgáltatás-orientált megoldás/alkalmazás/rendszer [Thomas Erl, 2005]

Egy szolgáltatás-orientált megoldás, alkalmazás, rendszer lényegében szolgáltatások kompozíciója. Egy SOA környezetben a hálózaton keresztül hozzáférhető független szolgáltatásokat anélkül használjuk, hogy ismernénk a működési platformjukat.

A szakirodalomban egy sor SOA definíciót találhatunk, stratégiai, üzleti és technikai irányvonalat képviselőt egyaránt:

A szolgáltatásorientált architektúra egy keretrendszer az üzleti folyamatok integrálására és az IT infrastruktúra támogatására - biztonságos, standardizált egységekből felépülő szolgáltatások, amelyek újrafelhasználhatóak illetve kombinálhatóak az üzleti prioritásoknak megfelelően. [Lawler, Howell-Barber, 2007]

A szolgáltatás-orientált architektúra egy olyan szoftver architektúris megközelítés, melyben lazán összekapcsolt szoftveres szolgáltatások szolgálják ki az üzleti folyamatok és a felhasználók szükségleteit. A laza csatolás következtében az új modulok beillesztése illetve a régi vagy nem használt egységek cseréje is könnyedén megoldható. [Bieberstein et al. ,2006]

Az OASIS hivatalos SOA definíciója:

„...különböző irányítás alatt álló szolgáltatók erőforrásainak használatára és rendszerezésére szolgáló minta. Egységes eszköztárat ad az erőforrások kereséséhez, szolgáltatásához és használatához, hogy elérjük az előre definiált és mérhető feltételekkel meghatározott célt.”

Az IBM definíciója szerint a SOA egy architektúris stílus olyan nagyvállalati IT rendszer megvalósítására, amely a szolgáltatás-orientáció (az üzlet integrálása egyedi szolgáltatások összekapcsolása segítségével) elveit követi annak érdekében, hogy szorosabb kapcsolatot építsen ki az üzlet és az üzletet kiszolgáló információs rendszerek között. [High, Kinder, Graham, 2005]

Bizonyos elvek minden SOA reprezentációban megjelennek, ezért a SOA alapelveinek tekinthetjük őket:

Egységbe zárttság:

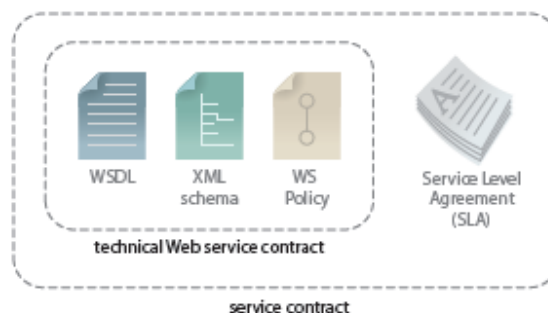
- Az implementációs részletek elrejtését jelenti. A szolgáltatásokat olyan burokkal kell körülvenni, hogy megfeleljenek a SOA alapelveket megvalósító tulajdonságoknak. Általános, implementáció-független interfészszel kapcsolódnak a szolgáltatások a környezetükhöz. Másrészt a szolgáltatás metódusai között erős kohézió valósul meg. Az erős kohézió következménye a tartósság és a jó újrafelhasználhatóság.

Laza csatolás:

- A különböző szolgáltatások nem foglalkoznak a velük kapcsolatban álló más szolgáltatások állapotával (elérhetőség, információtartalom, stb.), csak azok létezésével szükséges tisztában lenniük. A laza csatolás egyik legnagyobb előnye a modularitás, azaz hogy egy szolgáltatás implementációja lecserélhető anélkül, hogy a többi szolgáltatáshoz hozzá kéne nyúlni.

Szolgáltatási szerződések:

- A laza csatoláshoz szükségesek a szolgáltatási szerződések, melyek megszabják az egyes szolgáltatásokkal való kommunikáció szabályait és módját. Egy (vagy több) közös, minden szolgáltatás számára elérhető dokumentumból kérhetőek le.



7. ábra Szolgáltatási szerződés [Thomas Erl, 2005]

Absztrakció:

- A szolgáltatásokról csak annyi információ érhető el, amennyi a szolgáltatási szerződésekben le van írva, a szolgáltatások belső logikája így rejtve marad a többi szolgáltatás elől. Az absztrakciónak nyilvánvaló biztonsági és szerzői jogi előnyei mellett fontos szerepe van a laza csatolás megvalósításában.

Újrafelhasználhatóság:

- Az újrafelhasználhatóság érdekében az üzleti folyamatokat szolgáltatások valamely összekapcsolásaként, együttműködéseként definiálják. Ha változik az üzleti logika, nem kell az egész folyamatot újra implementálni, elég lehet mindössze néhány szolgáltatást lecserélni. Kiküszöbölhető a redundancia mind folyamat-, implementáció- és adat-szinten. Költség- és időmegtakarítást jelent fejlesztési időben és az üzemeltetés során egyaránt.

Autonómia:

- A szolgáltatások önállóan működnek, és tevékenységüket a többi szolgáltatás csak a címzett szabványos bementére küldött adatokkal képes befolyásolni, mivel az egyes szolgáltatások belső logikája rejtve marad a többi elől.

Optimalizáció:

- Ha minden más tényezőben megegyeznek, akkor két szolgáltatás közül a magasabb minőséget képviselő előnyt élvez az alacsonyabb minőséget nyújtó más szolgáltatásokkal szemben.

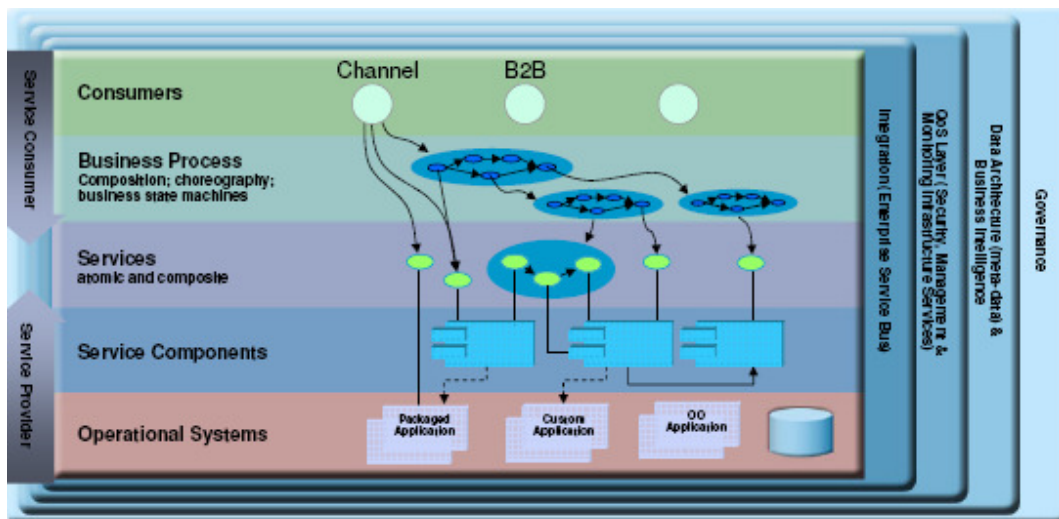
Megtalálhatóság:

- A szolgáltatásokat el kell látni megfelelő leírásokkal, annak érdekében, hogy az elérhető keresési mechanizmusokkal megtalálhatóak legyenek a kívánt szolgáltatást végző modulok. Ennek különösen akkor van

jelentősége, amikor felmerül egy szolgáltatás újrahasznosításának lehetősége, hiszen ha ilyen esetben nem áll rendelkezésre a megfelelően részletes leírása, akkor elképzelhető, hogy újra implementálásra kerül a szolgáltatás.

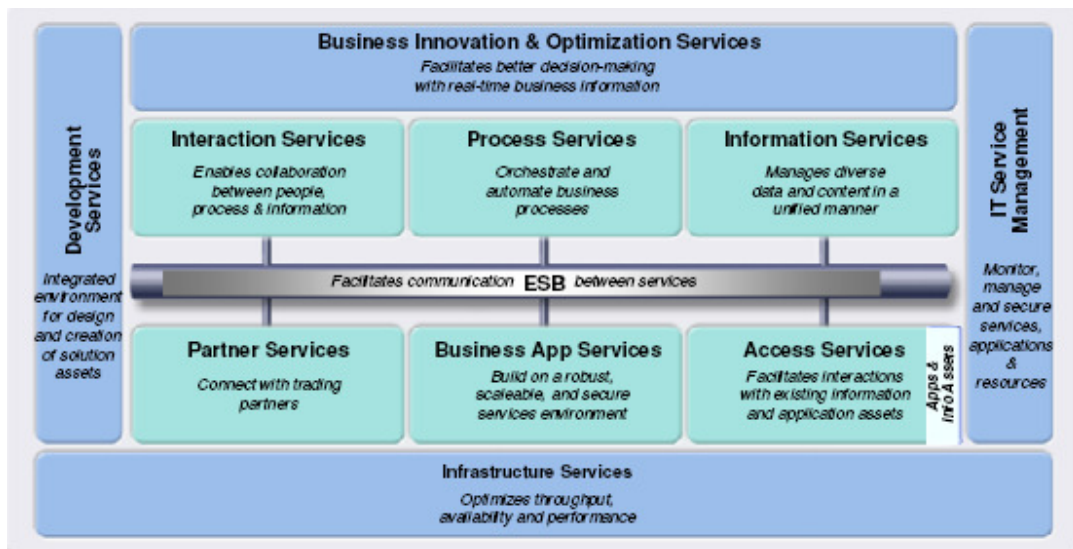
Granularitás:

- Az üzleti célokat megvalósító komplex folyamatok funkciói összetett - „coarse-grained” szolgáltatások, amelyek elemi komponensekből - „fine-grained” szolgáltatásokból épülnek föl.



8. ábra SOA rétegek [Bertrand, 2007]

A különböző szolgáltatások képesek összekapcsolódni és egy egységes rendszerként közösen működni. Vagyis megvalósíthatóak az üzleti folyamatok elemi szolgáltatások megfelelő módon történő összerendezésével.



9. ábra SOA referencia architektúra [Bertrand, 2007]

A SOA újszerű, szolgáltatás alapú architektúrális megközelítés a vállalat üzleti működésének feltárásához, elemzéséhez és javításához, valamint átlátható informatikai rendszer kialakításához és igény szerinti bővítéséhez, változtatásához.

Az architektúra középvonalában az ESB (Enterprise Service Bus) helyezkedik el. Korszerű SOA környezetben az ESB valósítja meg a middleware feladatait. Elsősorban architektúrális minta, melyet centralizáltan és elosztottan is implementálhatunk. Szabványokon alapuló integrációt tesz lehetővé lazán csatolt alkalmazások és szolgáltatások között szolgáltatás-orientált üzenet-vezérelt és esemény-vezérelt architektúrában

Az ESB-n belüli mediáció a szolgáltatáshívások / üzenetek / események intelligens feldolgozását teszi lehetővé az alkalmazások végpontjain, illetve a bus infrastruktúrájában. A hívó és szolgáltató között áramló üzenetek összetett mediációs logikák, amelyek újrahasznosítható mediációs primitívekből (Mediation Pattern) épülnek fel .

A szolgáltatás virtualizáció során a tartalom-alapú irányítás (routing) elfedi a szolgáltatót, a szolgáltatás-kiválasztás, conversion elfedi a hívás módját, a transzformáció pedig elfedi az interfészt.

Az ESB kommunikációs protokolljai támogatják a korszerű interakciós mintákat (pl.: request/reply, public/subscribe)

A hívó és szolgáltató között áramló adatstruktúrák leírása, kezelése meta-modellek felhasználásával történik. Az üzenetek leírására az XML Schema language szolgál. A konkrét üzenettípusokat tartalom-modellek (pl.: XML Schema) írják le - iparág-specifikus és vállalat-specifikus modellek, adattípusok. Az ESB-k legalább egy meta-modellt és több tartalom-modellt támogatnak.

Az ESB- k a következő szabványokkal dolgoznak:

HTTP/HTTPS, JMS, JAX-RPC, SOAP, WS-Security, WS-Policy, WS-Addressing.

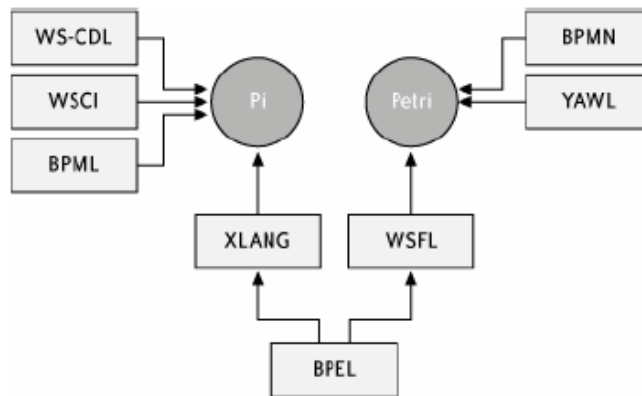
BPEL (Business Process Execution Language) - üzleti folyamatok végrehajtási nyelve

A nyelv XML alapú folyamatleíró nyílt OASIS szabvány. XML maga a program- és a folyamatmodellezési nyelv. A BPEL együttműködik az XPath-szal, az XSLT-vel és az XQueryvel az XML adatkezelés érdekében XML a kommunikáció nyelve is a folyamat és a szolgáltatások vagy más folyamatok között. A szabvány nem ad meg semmilyen konkrét grafikus megjelenítési formátumot, sem szabványos folyamat-tervezési módszertant.

A Web szolgáltatások standardjaira épít a folyamatok dekomponálásának és összeillesztésének modellezése során. Az üzleti működés leképezése eredményeként olyan végrehajtható program jön létre, amely a szolgáltatások összehangolását végzi.

A BPEL nyelv biztosítja egyrészt az absztrakt folyamatleírás lehetőségét, amely elsősorban az eljárás modellezésére szolgál, másrészt pedig a konkrét működés alapját képező végrehajtható folyamatleírás lehetőségét.

A BPEL az IBM WSFL és a Microsoft XLANG nyelveket egyesíti és kibővíti a korábbi alaptechnológiák funkcióit. A nyelv matematikai hátterét a Petri-hálók és a Pi-kalkulus jelenti.



Pi-kalkulus

A pi-kalkulust R. Milner a kommunikáció leírására szolgáló egyik legelső rendszerének a CCS (Calculus of Communicating Systems) kalkulus továbbfejlesztésével alakította ki a 90-es években. A CCS rendszer egyik alapvető jellemzője a megfigyelhetőség, azaz annak a leírása, amit egy külső megfigyelő tapasztalhat. A másik tulajdonság az egymástól függetlenül működő folyamatok közötti szinkron kommunikáció.

A pi-kalkulus összekapcsolt folyamatok hálózatában a kapcsolatok mozgásának leírására szolgál, egy process algebra, ami lehetővé teszi párhuzamos folyamatokból álló, dinamikusan változó hálózatok működésének leírását.

A pi-kalkulus alapfogalma a folyamat, a folyamatok csatornákon keresztül cserélhetnek információt. A csatornáknak azonosító nevük van, és a csatornákon továbbított információ nem csak adott szerkezetű adat, hanem egy csatorna neve is lehet. Lényegében ez utóbbi különbözteti meg a pi-kalkulust a korábbi CCS rendszertől.

A pi-kalkulus műveleti szemantikáját, azaz a kifejezések redukcióját relációkkal adhatjuk meg. Különböző redukciókkal különböző eredményeket kaphatunk. Párhuzamosan futó programoknál előfordulhat, hogy az eredmény attól függ, hogy a belső események milyen sorrendben hajódnak végre.

A rendszer működését redukciós szabályokkal adjuk meg. A redukciók nem adnak információt arról, hogy mi történik a rendszer környezetében. Két folyamat közötti címkézett átviteli reláció az akció. Az átviteli relációkra következtetési szabályokat lehet megadni, ezzel a matematikai logika módszereit használva egy redukciós sorozat helyességét lehet bizonyítani.

Többféle megközelítés létezik arra, hogyan tudjuk kifejezni azt, hogy két folyamat azonosan működik. Az operációs ekvivalencia szerint két folyamat ekvivalens, ha redukáltaik minden környezetben azonosan viselkednek. A problémát nehezíti, hogy a pi-kalkulusban a folyamatok futási ideje nem feltétlenül véges, tehát nem garantálható, hogy egy folyamat redukálása véges lépésben befejeződik.

Petri-háló

A Petri-hálókat az 1960-as években Carl Adam Petri határozta meg. A Petri-hálóok diszkrét elosztott rendszerek matematikai ábrázolása, lényegében gráf-modellek, amelyek párhuzamos folyamatok leírására, modellezésére, elemzésére alkalmasak. Mivel ez az ábrázolás az egy időben lejároló események megjelenítésére alkalmas, az automataelmélet általánosításának tekinthető.

A Petri-hálóok egyidejűleg nyújtanak grafikus és matematikai reprezentációt

- Konkurens egyidejűleg működő, önálló egységek kommunikálnak egymással úgy, hogy ezen egységek egymáshoz képest tetszőleges működési fázisban vannak
- Aszinkron eseményvezérelt rendszerek
- Elosztott az egyes rendszeremlék között funkcionális tagolódás van, azaz valamilyen megegyezés arról, ki milyen feladatot lásson el a teljes és hatékony működés érdekében
- Párhuzamos a rendszeremlék között szoros szinkronizáció áll fenn
- Nemdeterminisztikus adott állapotából nem egyértelmű, melyik állapot lesz a következő
- Sztochasztikus

rendszerek modellezésére.

A Petri-háló helyekből, átmenetekből és irányított élekből, mint elemekből áll. Az élek kötik össze a helyeket az átmenetekkel és fordítva. A helyek és az átmenetek saját csoportja között nincs közvetlen élel megvalósított kapcsolat, azaz a Petri-hálóok irányított páros gráfok. Az egyes helyeken tetszés szerinti számú „token” fordulhat elő, amely tokenek akkor kerülnek át a következő helyre - a helyhez kapcsolódó átmenet akkor „tüzel” - ha az átmenethez vezető élek mindegyikén a „tüzelési”

A gyakorlatban két folyamatot azonosnak tekinthetünk akkor is, ha az egyik több lépésben végzi el azt a műveletet, amit a másik kevesebb lépésben. Ahogy a λ -kalkulusban, a pi-kalkulusban is leírhatók a programozási nyelvekből jól ismert típusértékek és típusműveletek. A λ -kalkulus és a kombinátor logika kifejezései is leírhatók a pi-kalkulusban, és ebben az értelemben a pi-kalkulus a λ -kalkulus kiterjesztésének is tekinthető.

feltétel teljesül.

Petri-hálókkal különböző absztrakciós szinten írhatunk le modelleket. A modellezett rendszerek, például elosztott programok tulajdonságait úgy igazolhatjuk, hogy a nekik megfelelő Petri-hálót vizsgáljuk. Petri-háló segítségével elsősorban a program egyes folyamatai vagy folyamatrészei közötti sorrendi, függőségi és szinkronizációs kapcsolatokat, az információ áramlását, a kommunikációt, a konfliktus, illetve a konkurencia helyzeteket elemezhetjük, miközben az egyes értékadások, függvényhívások jelentésétől elvonatkoztatunk. Ha két különböző rendszert modellező Petri-háló lényegében azonos, akkor a két rendszer kommunikációs, szinkronizációs struktúrája is megegyezik az adott absztrakciós szinten. Ilyenkor a modellek dinamikus viselkedése is hasonló.

A Petri-háló a vezérlést és az adatszerkezetet egyaránt struktúrával ábrázolja. Ennek a megközelítésnek az eredménye, hogy minden más ábrázolásmód kiteríthető Petri-hálóra. Hátránya ugyanakkor, hogy már egyszerű feladatok leírása is hatalmas hálót eredményez. A kiforrott matematikai háttér miatt azonban ez a leírás mód rendkívül hatékony eszköz lehet rendszerek elemzésére, ha a rendszer modelljét valamely kompaktabb modellezésből automatikusan származtatjuk.

A BPEL a folyamatok magas szintű állapotátmeneteit modellezi. A nyelv az állapotátmeneteket absztrakt folyamatra vonatkoztatva értelmezi („programming in the large”). Az absztrakt folyamat azt írja le, hogy mikor van szükség üzenetek küldésére és fogadására, mi történjen, ha egy tranzakció nem sikerül, stb. A konkrét aktivitások kódolása más, imperatív programnyelven történik („programming in the small”) hagyományos fejlesztési folyamat keretében, amelyeket a BPEL folyamat meg tud hívni. A két felhasználási mód – a belső, futtatórendszerbeli ill. a külső, absztrakt nézet - viselkedése nem térhet el egymástól.

Az önálló, autonóm egységet alkotó BPEL folyamatok webszolgáltatásokra képezhetők le mind a futtatórendszerben, mind az absztrakt állapotváltozókra. A folyamat-modellek (de)kompozíciója webszolgáltatás-alapon történik, az elemi folyamatok ill. azok kompozíciói webszolgáltatásokat valósítanak meg. A szolgáltatások megvalósításánál törekedni kell a modularitásra és a kompozíció lehetőségének biztosítására - magasabb szintű folyamatok építése meglévő BPEL rész-folyamatokból.

A modellezett üzleti folyamatok webszolgáltatásokon keresztül kommunikálnak a külvilággal. A kommunikáció olyan értelemben absztrakt, hogy a hivatkozásokat nem konkrét port-okra, hanem portType-okra kell megadni.

A BPEL nyelv hierarchia és gráf alapú vezérlési struktúrákat is biztosít, illetve gondoskodik arról, hogy ezeket tetszőlegesen együtt lehessen használni.

A BPEL mint végrehajtási nyelv segítségével megadhatjuk milyen lépések sorozatát kell elvégeznie a tervezett szolgáltatásnak. Ugyanúgy, mint a hagyományos programozási nyelvekben, változókat lehet definiálni, azok tartalmát módosítani, és a szokásos nyelvi konstrukciók (ciklusok, feltételvizsgálat, többszörös elágazás) is rendelkezésre állnak.

Technikai szempontból a BPEL egy szabványos nyelvet biztosít, amellyel előírja, hogy miként kell XML-üzeneteket küldeni távoli szolgáltatásoknak, kezelni az XML-adatstruktúrákat, XML-üzeneteket aszinkron módon fogadni távoli szolgáltatásoktól, eseményeket és kivételeket kezelni, meghatározni a párhuzamosan végrehajtható műveleteket, illetve visszavonni az olyan részfolyamatok részeredményeit, amelyekben kivételesemények léptek fel. Ezek azok az építőelemek, amelyekkel a szolgáltatások egy adott csoportját együttműködő és tranzakció alapú üzleti folyamatokká lehet egyesíteni.

A BPEL elsődleges felhasználási területe az automatikusan lefutó, főleg rendszerszolgáltatások összehangolását végző folyamatok megvalósítása. Kiegészítő komponensekkel BPEL-ben is leírhatunk workflow jellegű üzleti folyamatokat, azonban ezek a megoldások már nem tekinthetők szabványosnak, mivel erősen függenek a szállítótól.

A BPEL nyelv elemei:

Process felépítés

<partnerLink>	- kapcsolat két szereplő között
<partner>	- partnerLink-ek halmaza
<variable>	- folyamat globális változói, ideértve a kimenő/bejövő üzeneteket
<Activity>	- folyamat lépései
továbbá:	<faultHandler>, <compensationHandler>, <correlationSet>, <eventHandler>

Alap tevékenységek

- <invoke> - Web service meghívása
- <receive-reply> - kérések fogadása-adatok küldése
- <throw> - belső hiba jelzése (kivétel)
- <wait> - várakozás
- <assign> - változók elérése/kezelése
- <compensate> - compensationHandler hívás sikeresen lefutott lépés visszavonása

Strukturált tevékenységek

- <sequence> - sorosan végrehajtandó lépések
- <switch> - feltételes elágazás
- <while> - ciklus
- <pick> - üzenetek, alarmok kezelése
- <flow> - párhuzamosan végrehajtható tevékenységek

Scope

tevékenységek egy sorozata melyre definiálható FaultHandler és Compensation handler.
Scope is tartalmazhat scope-ot.

A folyamatok példányosításának és a példányok egyedi azonosításának lehetősége biztosított. Az azonosítókat a futtató rendszer használja az üzenetcsere során. A külső társrendszerek is definiálhatják a rájuk vonatkozó azonosítókat, és meg kell teremteni ezek változtatásának lehetőségét.

A nyelv az alapvető életciklus műveleteket is biztosítja - folyamat létrehozása, terminálása, felfüggesztése, továbbbindítása.

Törekedni kell jól bevált megoldásokon alapuló (kompenzációs műveletek ill. az érvényességi kör behatárolása - scoping, hibaesemények kezelése) hosszú életciklusú tranzakciós modell megvalósítására. Az egyes folyamatok futásideje akár nagyon hosszú is lehet, tehát az állapot-változók csupán memóriában tartása nem elegendő.

A korábban javasolt folyamatszabványokkal szemben a BPEL élvezheti a vezető szoftvergyártók kritikus fontosságú támogatását is. Míg a korábbi szétaprózódott kezdeményezések nem voltak képesek arra, hogy az ügyfelek igényeit kielégítő,

egységes és mindenre kiterjedő előírásrendszert honosítsanak meg, a BPEL olyan átfogó szabványt kínál, amely megfelel a valós élet követelményeinek, ráadásul olyan nagy infrastruktúra- és alkalmazásfejlesztő vállalatok támogatását élvezi, mint az Oracle, a Microsoft, az IBM, az SAP és a Siebel.

A szervezeti és a határain átívelő üzleti folyamatok és webszolgáltatások összehangolására és végrehajtására szolgáló ipari szabvány bevezetése nemcsak felgyorsítja az integrációs projekteket, hanem csökkenti is a meglévő folyamatok felügyeletével, monitorozásával, módosításával, kibővítésével és korszerűsítésével kapcsolatos költségeket. A taktikai szinten jelentkező idő- és költségmegtakarítás mellett egy ilyen megoldás legfőbb stratégiai előnye az, hogy a vállalatok üzleti agilitása jelentős mértékben nő. Sokkal gyorsabban és rugalmasan képesek reagálni a változó piaci feltételekre, törvényi változásokra, új ipari szabványokra, konkurens fenyegetéseire. SOA környezetben hatékonyabbá válik az innováció támogatása, valamint versenyelőny szerezhető a gyors termékfejlesztési ciklus révén.

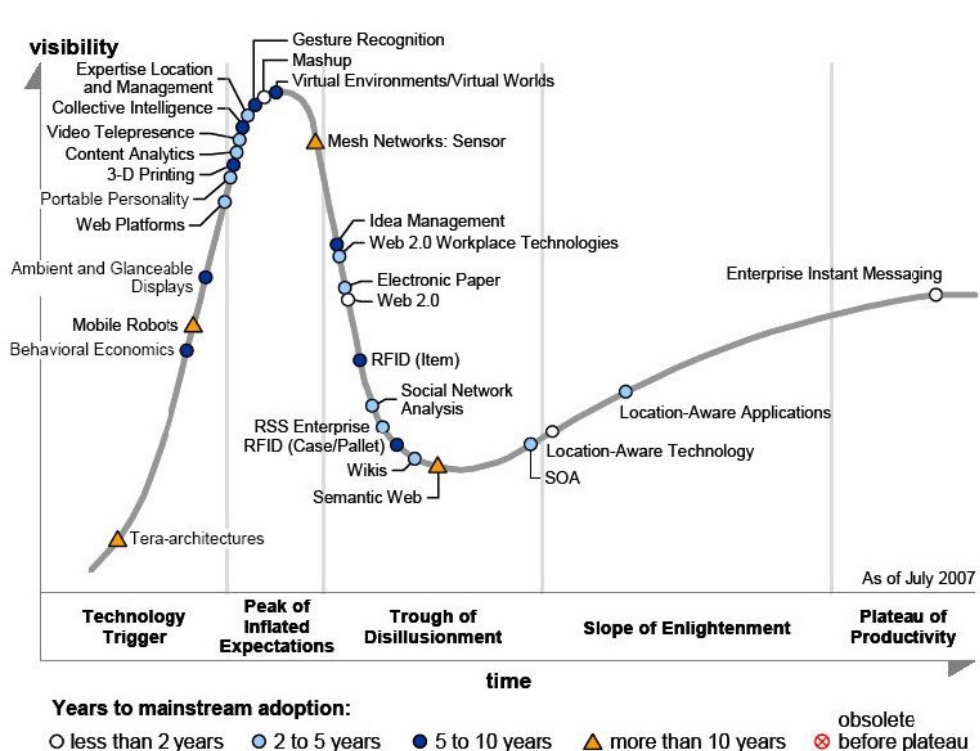
A SOA alkalmazása a szervezet számára abban az esetben ajánlható, ha különböző irányítás alá eső elosztott, heterogén rendszerek állnak rendelkezésére. [Josuttis, 2007]

A laza csatolásnak köszönhetően a SOA kiválóan alkalmas az állandóan változó környezetben működő elosztott rendszerek kiszolgálására. Az elosztott rendszerek fontos tulajdonsága, hogy annak különböző részei nem feltétlenül tartoznak azonos irányítás alá. Amellett, hogy eltérő egységek eltérő részlegek alá tartozhatnak a vállalaton belül, több vállalat rendszerei is kapcsolódhatnak egy elosztott rendszerbe. Közös irányítás hiányában magas szintű szervezettségre van szükség, amit a SOA képes biztosítani.

A különböző tulajdonú elosztott rendszerben az eltérő szervezeti egységek különböző hardver és szoftver infrastruktúrán futtathatnak más és más szoftvereket, amelyek szabványos felület nélkül nem képesek az együttműködésre. A SOA biztosítja a szolgáltatások szabványos kommunikációját.

A technológia érettsége, lehetőségei

Új technológiára való átállás tervezésekor a szervezet számára fontos tudni, hogy az adott technológia az érettség éppen melyik fokán áll. A Gartner kutatóintézet évente kiadja a feljövő technológiákról szóló riportját, melyben a hype-görbe szemlélteti a technológiák érettségét, illetve lehetőségeit.



10. ábra A feljövő technológiák hype-görbéje [Fenn, 2007]

Az ábráról leolvasható, hogy a SOA túl van a „technológia-indító” fázison, azaz a technológia széles körben ismertté vált a szakmai társadalomban. Ugyancsak túljutott a „magnövekedett várakozások csúcsán”, vagyis a technológiával kapcsolatos tévképzetek, hiedelmek és túlzott elvárások „lecsillapodtak” a részleges, vagy teljes kudarccal végződő projektek „eredményeképpen”. A szolgáltatás-orientált architektúra az értékelés szerint a „kiábrándulás völgyének” végéről emelkedik éppen a termelékenység fennsíkja felé. Az alkalmazások révén az első vállalatok már

kitapasztalták a technológia erősségeit és gyengeségeit. A technológia nemsokára eléri a teljes érettséget, ezáltal széles körben ismertté és elfogadottá válik. A második generációs termékek már meghozzák a kívánt stabilitást a technológia számára, a vállalatok nemsokára elkezdik érezni a SOA valódi hasznát.

A prioritás-mátrix azt mutatja meg, hogy a technológiák középpontba kerüléséig várhatóan még hány évig kell várni. Ezen felül a prioritás-mátrix kategóriák szintjén (alacsony, közepes, magas és átalakító) az adott technológia várható hasznát is megbecsüli.

benefit	years to mainstream adoption			
	less than 2 years	2 to 5 years	5 to 10 years	more than 10 years
transformational	Web 2.0	SOA Web 2.0 Workplace Technologies Web Platforms	3-D Printing Collective Intelligence RFID (Case/Pallet) RFID (Item) Virtual Environments/ Virtual Worlds	Mobile Robots
high	Enterprise Instant Messaging Location-Aware Technology Mashup	Electronic Paper Expertise Location and Management Location-Aware Applications Portable Personality Social Network Analysis	Behavioral Economics	Mesh Networks: Sensor Semantic Web Tera-architectures
moderate		Content Analysis RSS Enterprise Video Telepresence Wikis	Ambient and Glanceable Displays Idea Management	
low			Gesture Recognition	

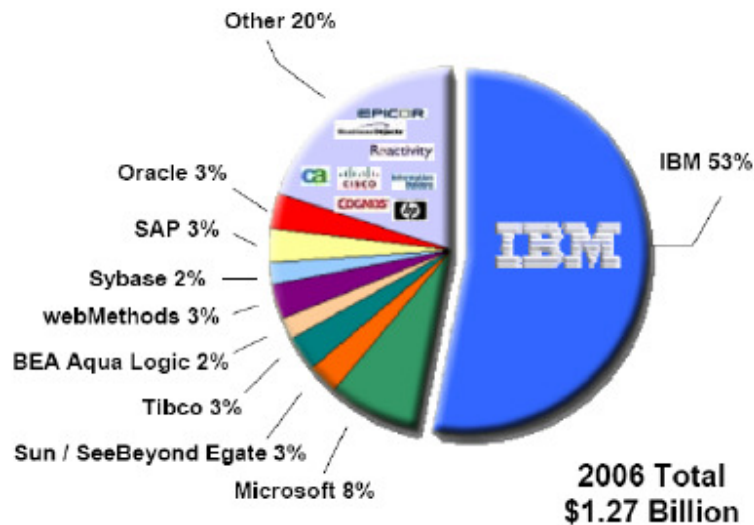
As of July 2007

11. ábra A feljövő technológiák prioritás-mátrix [Fenn, 2007]

A SOA az értékelés szerint 2007-ben az „átalakító” kategóriába tartozott és várhatóan 2-5 éven belül éri el az általános elterjedtséget.

A SOA Piac

Az IBM fölényesen uralja az eladott licenzek tekintetében a SOA folyamatmotor és együttműködési részegységek fiatal technológiának köszönhetően diverzifikált piacát. [Sassi, 2007].



12. ábra SOA folyamatmotor- és együttműködési részegységek piaca az eladott licenzek alapján

[Wintergreen Research 2006., Sassi, 2007]

A piaci részesedések tekintetében a SOA termékek piaca kompetitív, még akkor is, ha az IBM igen jelentős részt tudhat belőle magáénak.

Egyelőre a nagyvállalatok érdeklődnek a SOA iránt, így a piacot a kisszámú, de nagy értékű megrendelések uralják. Ez azonban feltehetőleg változni fog, amint a nagyvállalati piac telítődni kezd. Ekkor ugyanis előtérbe kerülhetnek a középvállalkozások számára fejlesztett olcsóbb, kompaktabb, egyszerűbben kezelhető és adminisztrálható megoldások. A legtöbb gyártó egységes, minden területre kiterjedő csomaggal igyekszik a piacon megjeleníteni. Ahogy a piac éretté válik, valószínűleg egyre több specializált termék jelenik majd meg, illetve az egyes szereplők felvásárlásokkal igyekeznek majd nagyobb részesedéshez jutni. Az IT világában nem ritka, hogy a „nagy hal bekebelezi a kis halat”, ha az valamit jobban, olcsóbban, eladhatóbban képes megvalósítani, mint azt ő teszi.

Steve Mills, az IBM Software Group alelnöke szerint az IBM felvásárlásainak legfőbb okai közé tartozik a piaci részesedés növelése, egy fontos, hiányzó technológia megszerzése, illetve a külső innováció bevonása a saját termékekbe és fejlesztésekbe. [Magid, 2006]

IBM

Az IBM magas piaci részesedése valószínűleg a korai piacra lépésnek, valamint a jól ütemezett felvásárlásainak köszönhető. Az IBM szoftvertechnológiai stratégiájának szerves részét képezi az akvizíció útján történő terjeszkedés. Az IBM régóta élenjáró a szabványosítási törekvésekben. A SOA-hoz köthető technológiák tekintetében többek között tagja a W3C és az OASIS szabványosító szervezeteknek.

Az IBM SOA területeken érintett részesedése az IBM 2007-es bevételeiből: a nagyvállalati szoftvereket gyártó IBM Software 20,4%, az üzleti és technológiai tanácsadással foglalkozó IBM Global Services 55,2%. A két terület együttesen több, mint a bevétel háromnegyedét termeli ebből is következik, hogy az IBM nagy hangsúlyt fektet SOA termékeire. [IBM, 2007a]

Az IBM szolgáltatás-orientált architektúrát támogató termékeit WebSphere márkánév alatt hozza forgalomba:

WebSphere Business Modeler - a rendszerelemzők számára készült üzleti folyamatok modellezésére és analitikus elemzésére alkalmas szoftver.

WebSphere Process Server - speciális alkalmazáserver, amelyen a Modelerben létrehozott üzleti folyamatok futtathatók. Az ESB és Message Broker termékek tartalmazzák.

WebSphere Integration Developer - segítségével a Modelerben (vagy más folyamatmodellező eszközben) létrehozott üzleti folyamatok lépéseit társítani lehet a megfelelő szolgáltatásokkal.

WebSphere Business Monitor - folyamatos adatgyűjtést végez az üzleti folyamatok futása során. Az aggregált adatok ezután grafikus felületen megjeleníthetők.

WebSphere ESB - az IBM ESB megoldása. Webszolgáltatások számára biztosít csatlakozási felületet.

WebSphere Message Broker - az ESB kiterjesztett változata, mely nem csupán XML-alapú kommunikációt tesz lehetővé - mint a WebSphere ESB -, ezáltal még jobban kiterjeszti az integrálható szolgáltatások körét.

ITCAM for WebSphere - az IBM Tivoli Composite Application Management for WebSphere az IBM Tivoli - egy komplex alkalmazás-felügyelő és –menedzselő rendszer - speciális változata, így csak érintőlegesen tartozik a WebSphere termékcsaládhoz. Segítségével monitorozható gyakorlatilag a teljes IT infrastruktúra. A WebSphere-hez készült változatban ez elsősorban az üzleti szolgáltatásokat, alkalmazásokat és az erőforrásokat foglalja magába.

Ezen IBM termékek elsősorban a nagyvállalatokat célozzák meg, ami megmutatkozik a komplexitásukban és konfigurálhatóságukban egyaránt. Ugyanakkor az IBM nyitott a kisebb költségvetésű és szakértőkkel nem rendelkező cégek felé is, amit az ITCAM for WebSphere egy „lebutított”, Basic elnevezésű változatának 2006-os piacra dobása is jelez [EMA, 2006].

Microsoft

A Microsoft (partnereivel együtt) is igen nagy hangsúlyt fektet a SOA tanácsadói oldalára, habár fontos számára termékeinek SOA-kompatibilissá tétele is. Ugyanakkor a legtöbb itt említett céggel ellentétben nem új szoftverek szolgáltatásként történő implementálását helyezi előtérbe, hanem a már meglévők elérhetővé tételét webszolgáltatásként.

A Microsoft alapvetően két fontos SOA termékkel rendelkezik (habár ennél jóval több terméke kapcsolódik valamilyen módon SOA technológiához):

Microsoft BizTalk Server - a Microsoft üzleti folyamatok menedzselésére szolgáló programcsomagja. A Microsofttól megszokott módon integráltan tartalmazza azokat az alkalmazásokat, amelyek segítségével egy vállalat megtervezheti, kifejlesztheti, futtathatja és menedzselheti üzleti folyamatait. (Az IBM mindezt négy különálló termékkel oldotta meg.) A BizTalk Server képes a vállalat alkalmazásaihoz webszolgáltatásként elérhető felületet biztosítani, valamint rendelkezik egy, az alkalmazások közötti üzenetváltást lehetővé tevő megoldással is, vagyis az ESB funkcióját hivatott ellátni.

Windows Communication Foundation - a WCF, mint az elnevezéséből is látszik, nagyobb hangsúlyt helyez a kommunikációra, mint azt a BizTalk Server teszi.

A korábban Indigo néven ismert keretrendszer a .NET Framework 3.0-ra épül és a forgalomban lévő Windows operációs rendszerek (XP, 2003 és Vista) részeként (az XP és a 2003 esetében kiegészítés) biztosítja a különböző, akár elosztott rendszerben lévő, alkalmazások közötti biztonságos és megbízható kommunikációt [Peiris et al., 2007].

A két termék együtt teszi lehetővé SOA elven működő rendszerek építését: biztosítják az üzenettovábbítást, ellenőrzést és transzformációt, az üzleti folyamatok integrálását a fejlesztéstől a menedzselésig illetve monitorozásig, valamint a szabálykezelést.

Oracle

Az Oracle mint a világ második legnagyobb szoftvergyártó cége a SOA területén is élenjáró. Már az EAI tekintetében is a korai piacra lépők közé tartozott, és ezt az előnyt sikerült integrálnia a SOA megoldásaiba is [EMA, 2006].

Az Oracle a Microsofthoz hasonlóan SOA termékcsomagot kínál különálló termékek helyett. Kezdetben az Oracle csak a csomag néhány elemével rendelkezett, a tudatos akvizícióinak köszönhetően kibővítette a tartalmát, így mára egy teljes, minden igényt kielégítő SOA szoftverrendszerrel rendelkezik, amely az Oracle Fusion Middleware nevet viseli.

Oracle Fusion Middleware - alapjában véve az Oracle Application Server (alkalmazáserver) és a BPEL Process Manager - amely üzleti folyamatokat megvalósító webszolgáltatások összekapcsolását teszi lehetővé BPEL alapon - fúziója. Ehhez aztán hozzáillesztettek még számos, Java alapú, SOA alkalmazást, hogy az így kapott szoftvercsomag lehetővé tegye üzleti folyamatok fejlesztést, integrálását és futtatását. A csomag tartalmaz nem közvetlenül SOA-hoz kapcsolható elemeket is (pl. Portal), amelyekre azonban szükség van egy nagyvállalati SOA megvalósítás során (pl. egy intranetes portálba integrálják a szolgáltatások monitorozását).

Oracle SOA Suite - az Oracle Fusion Middleware tartalmazza ezt a kisebb csomagot, ennek ellenére érdemes említést tenni róla, mivel itt koncentrálódnak a SOA-specifikus termékek, mint a már említett BPEL Process Manager, a monitorozást megvalósító Oracle Business Activity Monitoring vagy az Oracle ESB megoldása, az Oracle Enterprise Service Bus.

SAP

Természetesen a SAP sem maradhat le a SOA versenyben. A hagyományosan kevésbé rugalmas ERP, ahol a rugalmasságot elsősorban az előre gyártott modulok közötti választás szabadsága jelenti, SOA képességekkel versenyképesé tehető olyan környezetben is, ahol eddig a rugalmatlansága miatt nem volt erre lehetőség.

Enterprise SOA - egy keretrendszer, amely az SAP saját köztesréteg megoldására, a NetWeaver-re épül. Segítségével az SAP gyorsabban képes igazodni az ügyfelek újonnan felmerülő igényeihez, azáltal, hogy SOA alapú szolgáltatásokból építi fel egyes összetett alkalmazásait.

TIBCO

A TIBCO is nagy hagyományokkal rendelkezik a szoftverintegráció terén, mivel az Oracle-hez hasonlóan az EAI vezető gyártói közé tartozott, emellett a SOA piaci megjelenésekor is az elsők között volt [EMA, 2006].

A TIBCO szolgáltatás-orientált architektúrát támogató termékei egyenként is és különböző méretű és tartalmú csomagokban is elérhetőek. Az alaptermékek, amelyek ActiveMatrix márkanéven kerülnek a piacra a következők:

ActiveMatrix BusinessWorks - egy termékben megvalósítja a szolgáltatások létrehozását, összekapcsolását és integrációját, mindezt platform független módon (Java illetve .NET támogatással). Önmagában is használható, de szükség esetén könnyedén összekapcsolható a többi ActiveMatrix eszközzel vagy akár más gyártók termékeivel is, mivel nyílt szabványokra épül [TIBCO, 2008^a].

ActiveMatrix Policy Manager - segítségével központilag meghatározhatóak és kezelhetők az egyes szolgáltatásokhoz tartozó biztonsági, auditálási és naplózási beállítások, valamint az SLA-k. Ezáltal a szolgáltatásokba épített üzleti logika elkülöníthető a szerződési politikától, ami rugalmasabb kezelést tesz lehetővé [TIBCO, 2008^b].

ActiveMatrix Registry - szabványos módon felépülő (UDDI) szolgáltatás-adatbázis, mely támogatja kategóriák létrehozását és kezelését, szolgáltatások közzétételét és természetesen a különböző szempontok szerinti keresést. Ennél fogva elősegíti a szolgáltatások újrafelhasználását, hiszen ha valamilyen feladatot már implementáltak és a megvalósítást végző szolgáltatást közzétették, akkor egy újonnan felmerülő, hasonló igény esetén a meglévő szolgáltatás könnyedén megtalálható a Registry-ben [TIBCO, 2008^c].

ActiveMatrix Service Bus - a TIBCO ESB megvalósítása, amely tartalmaz automatikus mediációt, biztonságos csatorna lehetőséget, verziókövető rendszert és természetesen adat-, valamint protokoll- konverziót [TIBCO, 2008d].

ActiveMatrix Service Grid - keretrendszer, amelynek az a célja, hogy a segítségével megvalósított szolgáltatások minél szélesebb körben alkalmazhatóak legyenek, azáltal, hogy csak az üzleti logikára kell koncentrálniuk, mivel az adminisztratív logika a keretrendszerbe van építve és onnan egységes módon menedzselhető [TIBCO, 2008e].

webMethods/Software AG

A mostanra már a német Software AG tulajdonában lévő cég viszonylag fiatal, alig tíz éves múltra tekint vissza. A kilencvenes években alakult, kifejezetten webszolgáltatásokra és azok köré épülő termékek fejlesztése céljából. [EMA, 2006]

A webMethods SOA megoldásait egységbe foglaló termék a webMethods Fabric, amely számos díjat is nyert. Mindezt annak köszönheti, hogy sok versenytársával ellentétben már az alapoktól kezdve úgy fejlesztették, hogy integráltan tartalmazzon minden, egy SOA alapú rendszerben fontos megoldást.

A webMethods Glue - segítségével könnyedén létrehozható webszolgáltatás egy már meglévő Java objektumból vagy EJB-ből minimális (EJB esetében nincs rá szükség) extra forráskód hozzáadásával.

A webMethods Integration Server - a Fabric termékcsomag központi futtatókörnyezete, amelyen a különböző üzleti folyamatokat megvalósító webszolgáltatások futtathatók. Grafikus felületen keresztül megoldható benne webszolgáltatások létrehozása több, már meglévő szolgáltatásból, illetve WSDL generálása.

A webMethods Broker - nagysebességű, megbízható, aszinkron üzenettovábbítást megvalósító kommunikációs eszköz. Tartalmaz egy önálló JMS rendszert, amely

szabványos üzenetváltást tesz lehetővé, valamint egy beépített monitorozó eszközt, amellyel nyomon követhetők az üzenetek.

A webMethods Servicenet - közös felületet biztosít a webszolgáltatások menedzseléséhez: azaz információk tárolása egy egységes, kereshető adatbázisban, valamint a szolgáltatások folyamatos monitorozása oly módon, hogy az esetlegesen fellépő különleges eseményekhez akciók rendelhetők [webMethods, 2005b].

A webMethods Modeler - üzleti folyamatokat képes egy grafikus felületen keresztül modellezni, majd a kész modell elemeihez generálhatóak a futtatható egységek, amelyekkel aztán a folyamat lefutása tesztelhető.

A webMethods Optimize - az üzleti folyamatok futását megfigyelő BAM (Business Activity Monitoring), azaz üzleti tevékenységeket valós időben monitorozó rendszer. Különböző mutatószámok rendelhetők az aktivitásokhoz, amelyek aztán összesítve egy-egy KPI-t (Key Performance Indicator - kulcsteljesítmény-mutató) alkothatnak.

A webMethods Access - a grafikus megjelenítésért felelős alkalmazás, amely képes vállalaton belüli és kívüli hozzáférést is biztosítani az üzleti információkhoz, oly módon, hogy a különböző szerepkörökben lévő felhasználók más-más adatokat láthassanak [webMethods, 2005a].

A SOA irányítás piacát jól szemlélteti az alábbi Gartner felmérés alapján készült Magic Quadrant (mágikus kvadráns), amelyben a gyártókat két ismérv alapján négy szegmensbe osztották. [Kenny, Plummer, 2007]

A kivitelezés képessége (Ability to Execute) - a gyártók termékei milyen mértékben képesek megvalósítani az elképzeléseiket (víziójukat).

A vízió teljessége (Completeness of Vision) - a gyártók mennyire képesek megérteni illetve befolyásolni a piaci trendeket a saját érdekeik érvényesítéséhez.

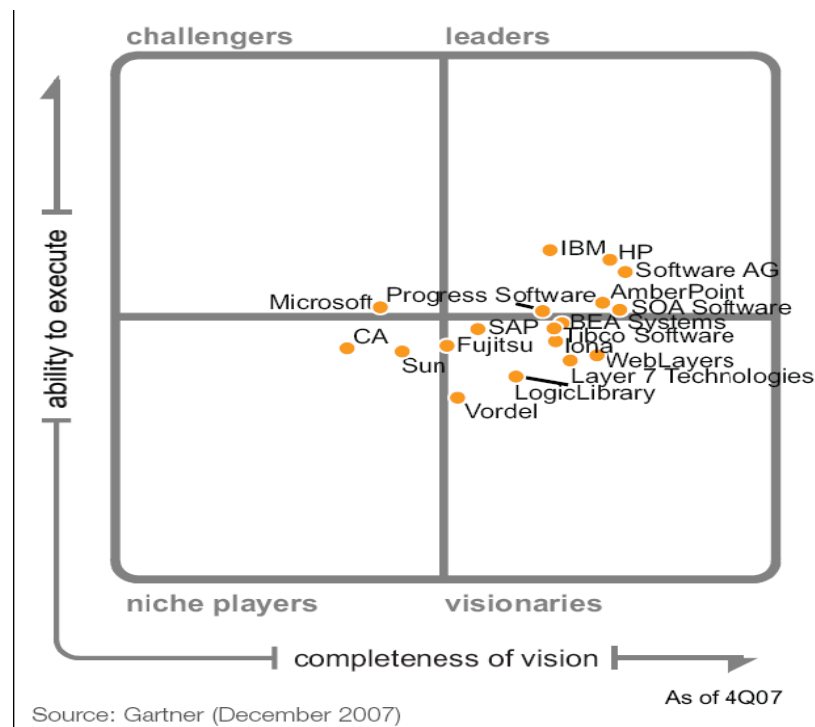
A fenti két ismerv alapján kialakult négy kategória (mágikus kvadráns) a következő:

Vezetők (Leaders) - Koherens elképzeléssel rendelkeznek a piac aktuális állapotáról, a trendekről és a lehetséges jövőbeli irányokról és képesek is alkalmazkodni hozzá, azaz a piaci igényeknek megfelelően alakítani termékportfóliójukat.

Kihívók (Challengers) - A kivitelezéssel nincsen gond, azonban a piaci igények felmérésén és az azoknak való megfelelésen még van mit javítaniuk.

Látnokok (Visionaries) - Jól látják a trendeket, sőt alakítják is azokat, azonban elképzeléseiket nem tudják következetesen megvalósítani.

Réspiacokat megcélzók (Niche Players) - Mivel nem rendelkeznek teljes képpel a piacról és a kivitelezésben is vannak hiányosságai, réspiacokat próbálnak megcélózni termékeikkel.



13. ábra Mágikus kvadránsok – SOA irányítás [Kenny, Plummer, 2007]

A vizsgált gyártók besorolását mutató ábra alapján az IBM itt is a vezetők között szerepel. Érdekes, hogy egy olyan nagy gyártó, mint például az Oracle, nem került be a felmérésbe. Ez azzal magyarázható, hogy nem rendelkezik önálló SOA irányítást megvalósító termékkel, az csak más termékükkel együtt érhető el. [Kenny, Plummer, 2007]

IBM WebSphere

Ebben a részben részletesebben bemutatom az IBM WebSphere termékcsalád azon elemeit, amelyeket a tézisek igazolásánál alkalmazni fogok. Az itt bemutatott eszközök az Eclipse technológiára épülnek. Az Eclipse eredetileg egy Java-alapú integrált programozási környezet, amelybe a beépülő modulok segítségével lényegében bármilyen funkcionalitás megvalósítható.

Business Modeler

A Modeler alapját a Holosofx üzleti folyamatmodellező és monitorozó termékeket gyártó cég által kifejlesztett modellező rendszer képezi. A modellező rendszer elméleti háttérét a BPMN szabványos jelölő nyelv adja. Az IBM relatíve régen, 2002-ben vásárolta fel a vállalatot. [IBM, 2002a]

A Business Modeler alapvetően üzleti folyamatok modellezését támogató eszköz. Jelenleg a 6.1-es verziónál tart és három változata elérhető:

- Basic - Az üzleti folyamatok modellezésén túl, azok dokumentálása és nyomtatása lehetséges benne, tehát alapfunkciókat tartalmaz. Ennek megfelelően az üzleti felhasználóknak ajánlható.
- Advanced - A felsorolt alapfunkciókon felül tartalmazza az elkészített üzleti folyamatmodellek szimulációjához illetve elemzéshez szükséges eszközöket is. Alapvetően IT szakembereknek ajánlja az IBM, hogy megbecsüljék az üzleti

folyamatok erőforrásigényét és ezáltal képet kaphassanak arról, hogy például milyen hardver támogatást igényel egy adott folyamat. A tézis igazolása során én is ezt a változatot használom.

- Publishing Edition - Az Advanced változaton felül tartalmaz egy Publishing Server nevű web-alapú eszközt, amelynek segítségével a modellek feltölthetők egy közös webtárházba. Ott azután egy egyszerű böngésző használatával a felhasználók megjegyzéseket fűzhetnek a modellhez vagy annak egyes részeihez, illetve egy kapcsolódó fórumban akár meg is vitathatják a problémás elemeket. A modell készítője az így összegyűlt információkat felhasználva javíthat illetve finomíthat a modelljén. [IBM, 2005b]

A Business Modelerben elkészült modellek átvihetők az Integration Developer-be, ahol a folyamat tevékenységeihez hozzárendelhetők webszolgáltatások, programkódok, stb.. Ha már a modellezés során ismert, hogy milyen platformon kell megvalósítani az integrációt, akkor ennek megfelelő nézetben lehet modellezni (például, ha a WebSphere Process Server-en fog futni a folyamat, akkor BPEL nézetben ajánlott dolgozni).

A modellezés lehetőségei

A modellezés grafikus felületen történik, a Modeler elemkészletének felhasználásával. A Modelert alap-, közép és fejlett-szintű üzemmódban használhatjuk, amelyek különböző mértékben teszik lehetővé a modellek testre szabását.

A modelleket két módon is szerkeszthetjük. A hagyományos „szabadkézi” elrendezésben az elemeket oda tehetjük a vásznon, ahová csak szeretnénk, míg elemsor (swimlane) elrendezés esetén erőforrások vagy szerepkörök szerinti sávokba kell elhelyeznünk a tevékenységeket (és alfolyamatokat), annak megfelelően, hogy az adott elem melyik erőforráshoz illetve szerepkörhöz van rendelve.

A modellelemek

Alapvetően globális és lokális elemeket használhatunk. A globális elemek katalógusokban helyezkednek el és bármely folyamatban újrafelhasználhatóak. A lokális elemek egy folyamaton belül használhatóak. A folyamat, tároló és a különböző típusú feladat elemek globálisan és lokálisan is megadhatók, a többi elem csak lokálisan. A szolgáltatás kizárólag globálisan definiálható.

A különböző modellelemek közötti kapcsolatot nyilakkal ábrázolhatjuk. Az így definiált kapcsolat mindig egy irányú. A kapcsolatok (mint élek) és elemek (mint csomópontok) által meghatározott irányított gráf a folyamat vezérlési szerkezetét adja meg. Ez alól mindössze egy kivétel van, a tárolókból jövő és tárolókba menő kapcsolatok minden esetben kizárólag adatfolyamot jelentenek, ezért külön jelölésük is van (kék, pontozott vonal).

Az adatáramlást az egyes elemek között a nyilakra írt munkaelem nevek jelzik. A munkaelemeknek két fajtája van, az előre definiált alaptípusok (például egész, szöveg, dátum, stb.) és a felhasználó által létrehozható összetett típusok vagy adatszerkezetek, melyek végeredményben alaptípusok valamilyen hierarchikus összekapcsolása révén hozhatók létre.

Feladatok

A feladatok az elemi tevékenységek a modellben. A feladat egy általános elemi tevékenység, amely nem bontható további résztevékenységekre. Általában egy szolgáltatás feleltethető meg neki a valóságban. Alapvetően három típust különböztetünk meg, aszerint, hogy ki végzi a feladatot:



14. ábra A feladattípusok jelölőelemei a Modelerben

Az üzletiszabály-feladat olyan speciális feladattípus, amely lehetővé teszi üzleti szabályok definiálását a folyamaton belül. Gyakorlatilag egy olyan elem, amelyben szabályok definiálása segítségével adható meg, hogy milyen bemenetre milyen kimenetet adjon. Ennek a megoldásnak két fontos előnye is van. Az egyik, hogy nincs szükség programozási ismeretekre ahhoz, hogy bonyolultabb logikát vigyünk egy folyamatba, így a modellt létrehozhatja üzleti szakember is. A másik előny a folyamat futtatásánál jelentkezik, mivel az ily módon megadott szabályokat - a lefordított programokkal ellentétben - a folyamat telepítése után is lehet (bizonyos mértékben) módosítani, ha a WebSphere Process Server-t használjuk.

Az emberi feladat olyan tevékenység, amely emberi beavatkozást, interakciót igényel. Például űrlapok kitöltése, jóváhagyás, stb.

Döntések

A döntések elágazást jelentenek egy folyamaton belül, amely futási időben, valamilyen belső logika alapján (általában a beérkező adatokat figyelembe véve) dönti el, hogy melyik vezérlési ágon haladjon tovább a folyamat.

A Modeler Advanced változata a folyamatok szimulációját is támogatja, ezért lehetőség van az egyes lefutási ágakhoz valószínűségi százalékértékeket is megadni. Ezeknek az értékeknek kizárólag a modell elemzése során van jelentőségük, mégpedig a szimuláció során az itt megadott valószínűség szerint halad tovább a folyamat a megfelelő irányba. A Modelerben kétféle döntéselemet használhatunk, a döntések utáni vezérlési utakat pedig egy összefésülő elem segítségével egyesíthetjük újra:

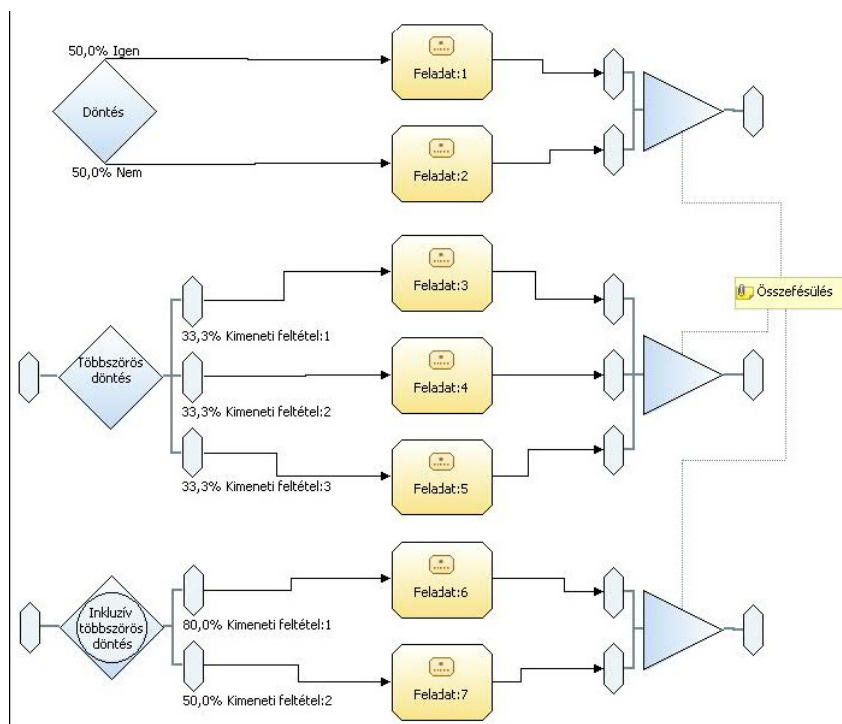
Egyszerű döntés esetén két irányba mehet a folyamat, aszerint, hogy a döntésben szereplő kérdésre a válasz igen vagy nem. A megfelelő logikát, amely a beérkező adatok alapján ezt eldönti, implementálni kell vagy még a Modelerben vagy a modell exportálása után az Integration Developer-ben. Értelemszerűen a válasznak mindig

egyértelműnek kell lennie, így ez a típusú döntésem nem engedi meg a folyamat egyszerre több ágon történő továbbfutását.

Kettőnél több kimenet esetén többszörös döntést érdemes alkalmazni, bár a feladat megoldható lenne egyszerű döntések összekapcsolásával is, de ez a módszer nagyon hamar átláthatatlanná tenné a modellt. Többszörös döntés esetén arra is lehetőség van, hogy egyszerre több irányban is továbbmehessen a folyamat, azaz inkluzív feltételrendszer megadására. Ilyen esetben a különböző ágak valószínűségeinek összege meghaladhatja a 100%-ot.

Az inkluzív döntés előnye a jobb áttekinthetőség és a kompaktabb modell, azonban az ilyen módon felosztott vezérlési útvonal újraegyesítése nem triviális feladat, ezért alkalmazása nem javasolt. Az ilyen típusú döntéseket inkább elágazások és exkluzív többszörös döntések kombinációjaként érdemes megoldani.

Az összefésülés a döntések utáni ágak újbóli összekapcsolására alkalmas segédelem. Minden esetben csak egy kimenettel rendelkezik és bármely bemenetén érkező vezérlési jel azonnal továbbhalad rajta.



15. ábra A döntések és az összefésülés jelölőelemei a Modelerben

Párhuzamosan futó feladatok

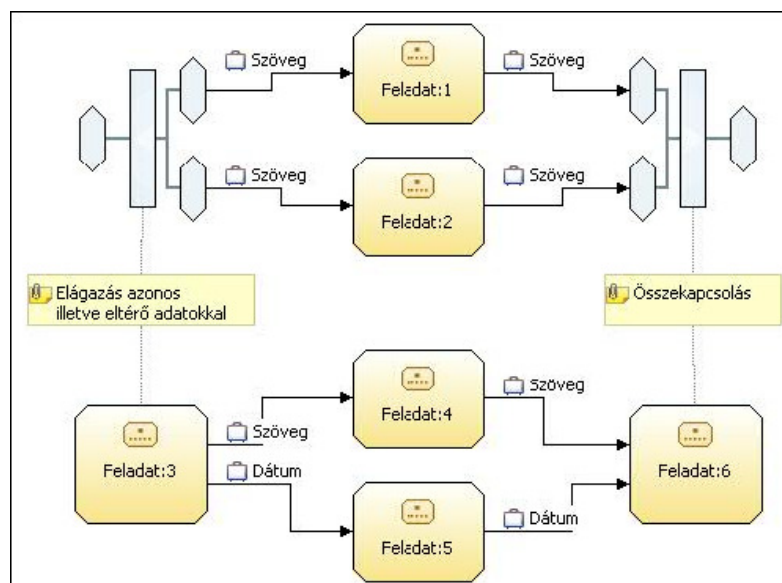
A párhuzamos végrehajtást a Modelerben az elágazás nevű elem teszi lehetővé, míg a különböző ágakat újraegyesítő szinkronizáció az összekapcsolás nevű elemmel történik. Az elágazásnak minden esetben egy bemenete van és minden kimenetén folytatódik a végrehajtás, méghozzá úgy, hogy a bemenetére érkezett minden adat minden kimenetére is továbbításra kerül.

Az összekapcsolásnak mindig csak egy kimenete van. A bemeneteire érkezett azonos típusú adatokat egy kifejezésben megadható módon (csak középszintű és fejlett módban) egy adatelemmé egyesíti. Így a kimenetén annyi adatelem jelenik meg, ahány különböző típusú a bemeneteire érkezett. Az Integration Developer-be történő export során az adategyesítő kifejezések Java kódra képződnek le.

A segédelemek nélkül is létrehozhatunk párhuzamos futást, azonban ilyen esetben nem azonos adatok vezérlik a különböző ágakat.

Megoldható volna, hogy az elágazást létrehozó feladat minden kimenetére ugyanazt az adatot tegyünk, de az ilyen megoldás nem ajánlott, mivel a folyamat logikájához igazítja a feladat logikáját, így a feladat implementációja kevésbé lesz újrahasznosítható.

Hasonlóan az elágazáshoz, az összekapcsolás is megoldható egy feladattal, amelynek megfelelő számú bemenete van. Érdemes elkerülni a feladat után szétágazó illetve a feladattal egyesített vezérlési utak használatát, mivel szimulációra csak a segédelemekkel létrehozott párhuzamos futás alkalmas.



16. ábra A párhuzamos végrehajtás segédelemei a Modelerben

A részfolyamat és a ciklusok

A főfolyamatokon belül lehetőség van részfolyamatokat definiálni, melyekhez hozzárendelhető egy logika is, ami megmondja, hányszor hajtódjanak végre. Az ilyen részfolyamatokat ciklusoknak nevezzük. A ciklusok csak lokálisak lehetnek, a részfolyamat lehet globális is.

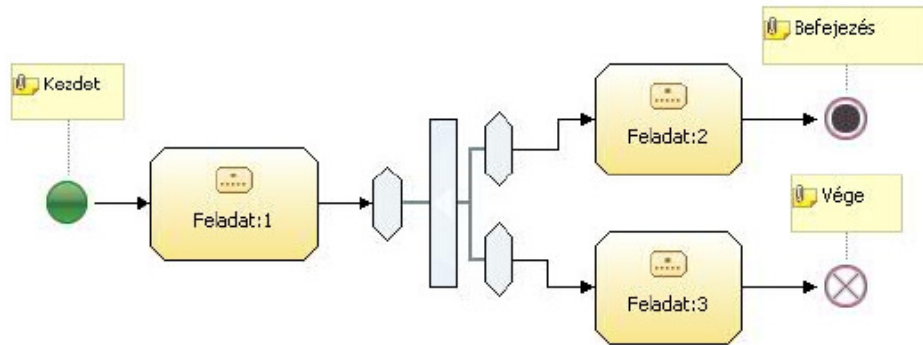
A ciklusok (while, do-while és for) azonban csak a modellben különbözőek, az Integration Developer-be történő export során mindegyik egy BPEL while ciklussá képeződik le (mivel az Integration Developer által támogatott BPEL szabvány ezt az egy fajta ciklust ismeri), a végrehajtást vezérlő logikából pedig Java kód lesz.

A részfolyamatok segítségével egy komplex folyamat dekomponálható, aminek eredményeképpen jobban áttekinthetővé válik.

Csomópontok

A csomópontok a folyamat (vagy részfolyamat) belépési és kilépési pontjai. Szerepük leginkább a vizuális megjelenítésben van. Új folyamat létrehozásakor a kezdő és

befejező csomópontok automatikusan generálódnak a folyamathoz. Összesen háromféle csomópontot használhatunk a folyamatokban:



17. ábra A csomópontok jelölőelemei a Modelerben

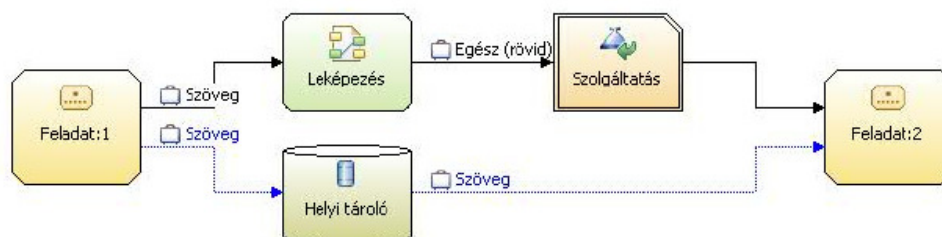
Kezdőpont - A folyamat belépési pontját jelöli, ezért pontosan egy kimenete van, bemenete pedig nincs. Egy folyamatnak lehet több kezdőpontja is (ez nem javasolt), de az is lehet, hogy egyáltalán nincs benne ilyen típusú csomópont. Utóbbi esetben a folyamat bal széléhez kell kapcsolni az első elemet. Ezt a megoldást akkor használjuk, ha bejövő adatai is vannak a folyamatnak, a kezdőpont ugyanis nem tudja átadni ezeket az adatokat. Egy kezdőpont használata olyan folyamatok esetén ajánlott, amikor nincsen bejövő adat, de ilyenkor sem kötelező.

Befejező csomópont - Az elnevezésének megfelelően a folyamat befejezését jelöli, így pontosan egy bemenete van, kimenete pedig nincs. Nagyon fontos, hogy minden folyamatban kell szerepelnie legalább egy befejező csomópontnak. A szimuláció nem is működik, ha ez a feltétel nem teljesül. A kezdőponthoz hasonlóan, befejező csomópontból is lehet több, ám itt is ajánlott egyet használni, mivel ha bármelyiket eléri egy vezérlési folyamat, a teljes folyamat leáll. A folyamat kimenő adatait - hasonlóan a kezdőpontnál látott bejövő adatok kezeléséhez - nem a befejező csomópontoz kell kapcsolni, hanem a folyamat jobb széléhez.

Végcsomópont - Kizárólag megjelenítési szerepe van. Olyan párhuzamos futási ágak lezárására használható, amelyek nem kapcsolódnak vissza a főágba.

Egyéb elemek

- Leképezés - A leképezés egy speciális feladat, amely adatkonverziót képes végrehajtani a bemenő munkaelemen úgy, hogy a kimenő munkaelem más típusú legyen. Természetesen ez általános esetben nem triviális feladat, így az implementáció gyakran nem a Modelerben, hanem a modell exportálása után az Integration Developer-ben történik.
- Szolgáltatás - A folyamat számára csak a szolgáltatás bemeneti és kimeneti paraméterei ismertek a funkcióján túl, a belső felépítéséről semmit nem lehet tudni. Éppen ezért szolgáltatásokból csak globálisat lehet definiálni.
- Helyi tároló - A tárolóban munkaelem példányokat lehet eltárolni a folyamat futása során. Egy későbbi feladat ezután hivatkozhat a tárolóban lévő valamely elemre. A tárolók egyik legfontosabb felhasználási területe a ciklusokba történő adatátadás. A ciklusok ugyanis nem képesek bemenő adatokat fogadni, így csak a szülőfolyamat helyi tárolóiban elhelyezve lehet egy cikluson belül a szülőfolyamat adatait felhasználni. Minden tárolóban csak egy típusú munkaelemet lehet tárolni. Megadható ezen kívül, hogy az adatok rendezetten legyenek-e tárolva, mekkora a tároló kapacitása (alapértelmezésben korlátlan) és hogy minden tárolt elem egyedi-e.
- A helyi tárolókba történő írás és a tárolóból olvasás a modell Integration Developer-be történő exportja során egy-egy szolgáltatáshívássá alakul át, amelynek automatikusan generálódik a Java implementációja is. Tárolóból is létre lehet hozni globálisat, ám ezek nem exportálhatóak az Integration Developer-be.



18. ábra A leképezés, szolgáltatás és helyi tároló jelölőelemei a Modelerben

Egyéb, nem exportálható elemek

Az alábbi elemeknek nem támogatott az Integration Developer-be történő exportja:

- Értesítésküldő - Speciális folyamat, amely előre definiált formájú értesítést generál, amelyet majd az értesítésvevő elem képes fogadni.
- Értesítésvevő - Az értesítésküldő üzenetét fogadja. Csak az előre meghatározott típusú értesítéseket képes fogadni.
- Időzítő - Szintén speciális folyamat, akár speciális értesítésvevőnek is lehetne nevezni. Az előre megadott időpontban (időszakonként, bizonyos idő elteltével) elindítja, vagy folytatja a folyamatot.
- Megfigyelő - A megfigyelő hasonlít az időzítőre, azzal a különbséggel, hogy nem az időt figyeli, hanem a folyamatot, és bizonyos szintén előre definiált feltételek teljesülése esetén elindítja, vagy folytatja a vezérlőfolyamatot.

Integration Developer

Az Integration Developer egy fejlesztő környezet eszköz, amellyel a Process Serveren futó alkalmazásokat lehet létrehozni. Az integrációs képességei miatt azonban több, mint egy hagyományos fejlesztő eszköz. A segítségével már meglévő szolgáltatások

komponensként felhasználhatóak üzleti folyamatot megvalósító alkalmazás részeként programozás nélkül. Ezáltal nem csak a fejlesztéssel eltöltött idő és a felmerült hibák száma csökkenthető, hanem a szolgáltatások összekapcsolásából felépített alkalmazások révén az üzleti logika elválasztható az implementáció részleteitől [Wahli, 2007].

Az Integration Developer alapja a szolgáltatás-komponens architektúra (Service Component Architecture - SCO), ami egy szolgáltatás-orientált komponens modell: az alkalmazás minden részegységét egy szolgáltatásként meghívható komponensnek tekinti, amely egy interfészen keresztül várja a hívásokat és referenciákkal rendelkezik azokra a komponensekre (illetve interfészeikre) amelyeket meghív. A komponensek implementációja már eltérő lehet:

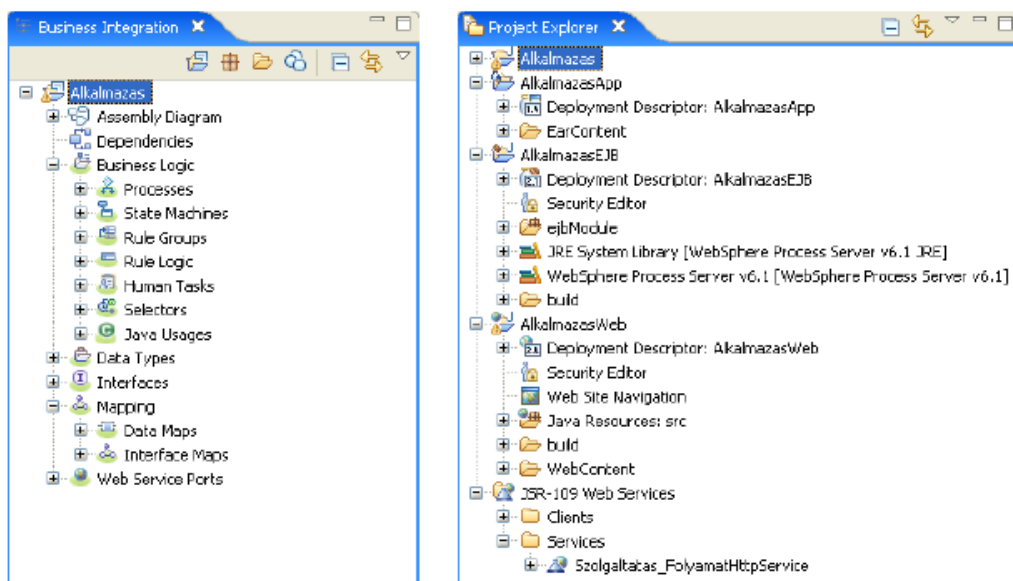
- Üzleti folyamat - A hagyományos üzleti folyamatokat BPEL folyamatokként implementáljuk.
- Üzleti állapotgép - Az UML 2.0 szabványra épülő állapotgép, mely az eseményvezérelt üzleti folyamatok reprezentálására alkalmas. A háttérben az üzleti állapotgépet is BPEL folyamatra képezi le az Integration Developer.
- Üzletiszabály - Feladata az üzleti logikát elkülöníteni az implementációtól. Segítségével nincs szükség az üzleti logikát az alkalmazásba kódolni, az futási időben is változtatható.
- Emberi feladat - Az emberek által végzett tevékenységeket is egy szolgáltatásba csomagolhatjuk.
- Kiválasztó - Képes választani több implementáció közül. Olyankor van értelme, ha bizonyos szolgáltatást több szolgáltató is nyújt, de más áron és minőségben vagy sebességgel. Ilyenkor a kiválasztó a megadott paraméterek alapján az eldönti, hogy az adott esetben a gyorsaság vagy az alacsony

költség a fontosabb és az annak megfelelő szolgáltatás felé irányítja a folyamatot.

- Leképezés - Különböző interfészek közötti, illetve üzleti objektumok közötti megfeleltetés, például eltérő adattípusoknál adatkonverzió.
- Java alkalmazás - Amit másképpen nem tudunk implementálni, azt megvalósíthatjuk hagyományos módon, programozással.

Az Integration Developer-ben is létre lehet hozni egy SOA alkalmazást, de érdekesebb egy már meglévő üzleti folyamatmodellből kiindulni, mivel alapvetően nem modellezésre tervezték. A legjobb a Business Modelerben készített modellből kiindulni, ebben az esetben szinte minden generálódik az alkalmazáshoz a modellből, amire a futtatáshoz szükség van. Mindössze az üzleti objektumok adatokhoz kapcsolását, az üzleti szabályok definiálását, a webszolgáltatások komponensekhez rendelését és a Java alkalmazások implementálását kell elvégezni és a folyamat futtathatóvá válik.

Az Integration Developer is Eclipse alapon működik, így lehetőség van különböző nézetekből megtekinteni a projekteket. Az integrációs feladatokhoz az üzleti integrációs perspektívát (Business Integration Perspective), a programok és szolgáltatások implementációjához a web perspektívát (Web Perspective) használhatjuk.



Üzleti integrációs

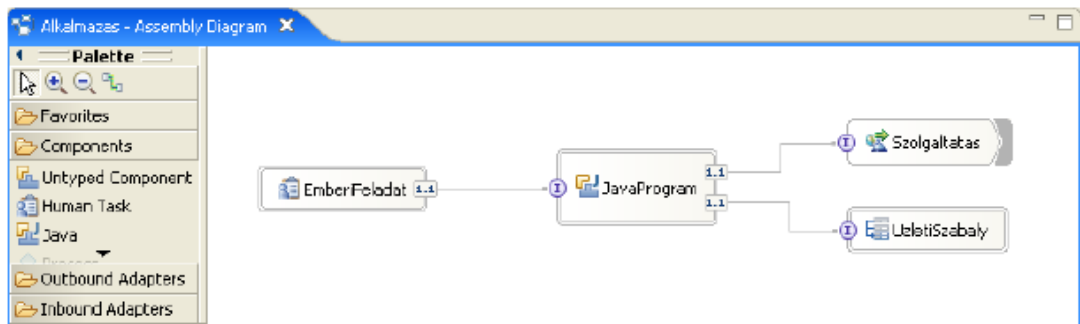
Web

19. ábra Különböző perspektívák az Integration Developer-ben

Az integrációs perspektíva elemei

Az összeszerelési diagram (Assembly Diagram)

Az integrációs perspektívában a komponenseket és kapcsolataikat az összeszerelési diagramon szerkeszthetjük. A különböző komponensek ikonjai annak megfelelően alakulnak, hogy milyen az implementációjuk, de alapvetően ugyanúgy néznek ki. A meghívott komponensek interfésszel (I) rendelkeznek, a hívásokat a számosságukkal együtt a referenciák jelképezik (az ábrán 1..1 számosságú kapcsolatok szerepelnek).



20. ábra Az összeszerelési diagram az Integration Developer-ben

Az üzleti logika (Business Logic)

Ebbe a gyűjtőbe tartoznak a különböző implementációk a leképezés kivételével, azaz a folyamatok, állapotgépek, szabályok, emberi feladatok, kiválasztók és Java alkalmazások. Minden implementációs típushoz saját szerkesztőfelület tartozik, így például a folyamatokhoz grafikus BPEL szerkesztő, a Java alkalmazásokhoz pedig alkalmazás-fejlesztő.

Az adattípusok (Data Types)

Ebben a gyűjtőelemen kizárólag összetett adattípusok találhatóak. Ezek olyan üzleti objektumokat reprezentálnak (például megrendelés), amelyeknek attribútumai vannak, amik már lehetnek egyszerű és összetett típusúak is.

A különböző adattípusok között öröklődés definiálható, így nem kell a hasonló üzleti objektumok esetében külön adminisztrálni az azonos attribútumokat.

Interfészek (Interfaces)

Minden komponensnek, amelyre hivatkozás történik rendelkezni kell pontosan egy interfésszel. Az interfész egy adott komponens kommunikációs felülete. Tartalmazza a komponens által nyújtott funkciók listáját, azok esetlegesen szükséges bemenő paramétereit és típusukat, valamint a választ (vagy válaszokat) és típusát. Típusként összetett adattípusok is használhatóak.

Az egyes funkciókhoz a bemenő paramétereken és a válaszokon felül megadhatók hibakimenetek is, amelyek a különleges esetek kezelésére szolgálnak.

A funkcióknak két fajtáját különböztetjük meg: a hívás-válasz funkciók (például kérelem elbírálása), amelyek választ és esetlegesen hibajelzést is képesek küldeni; illetve az egyirányú funkciók (például naplózás), amelyeknek csak bemenetük lehet.

Leképezések (Mapping)

A leképezés lehet üzleti objektumok közötti, illetve interfészek közötti megfeleltetés. Az üzleti objektumok közötti megfeleltetést úgy kell megadni, hogy a kimenő üzleti objektum minden attribútuma kapjon valamilyen értéket. Ez az érték lehet a bemenő üzleti objektumtól függő, például egy attribútum, attribútum attribútuma, stb., vagy lehet fixen megadott érték, külső forrásból származó, stb.

Lehetőség van automatikus leképezésre is, amikor a program az attribútumok nevei alapján megpróbálja megtalálni az összetartozókat és azokat megfelelteti egymásnak.

Az interfészek közötti megfeleltetést először funkció szinten, majd azon belül a paraméterek szintjén is meg kell adni. Itt már használhatóak az üzleti objektumok között korábban definiált leképezések is. Az interfészek közötti leképezéseknél nincs lehetőség automatikus megfeleltetést alkalmazni.

Webszolgáltatás címek (Web Service Ports)

Ebben a gyűjtőben található a folyamatokban használt (külső) webszolgáltatások elérhetőségei és kötése (binding). A kötés a webszolgáltatáshoz is egy interfészt rendel, így biztosítva, hogy komponensként tudjon működni. A webszolgáltatások - mivel üzleti logikájukba nem lehet belelátni - nem hívhatnak meg más komponenseket, így ezek komponensei nem rendelkezhetnek referenciával. Ez látható az összeszerelési diagramnál szereplő ábrán is. A Szolgáltatás nevű komponens egy webszolgáltatást hív meg, ezért a komponens jobb szélé szürke, jelezvén, hogy nem lehetnek referenciái.

Az integrációs rendszerek legfontosabb tulajdonságai tehát a mai információs kihívások függvényében:

- szabványokon alapulnak
- flexibilitás
- laza csatolás
- új rendszerek gyors integrációja

Ahhoz, hogy a számítógépes alkalmazások közötti kapcsolatok minél magasabb szinten jöhessenek létre, illetve ezeknek a fogalmaknak a gépi feldolgozása lehetővé váljon, szükséges, hogy a gépi ábrázolás mind inkább emberi fogalmakhoz közelebb álló szemantikai jelentéssel bírjon. A kellően magas szintű szemantikai együttműködés az, ami a web-alkalmazásokban a komponensek közötti öszszeragasztó közeget adja.

Az új irány, amire a szervezetelméletek egyre inkább fókuszálnak a középső réteg, az üzleti folyamatok, felismerve, hogy alapvető, létfontosságú kapcsolatot jelentenek a technikai és szervezeti infrastruktúra között.

II. 5. A szemantikus üzleti folyamatmenedzsment (Semantic Business Process Management – SBPM)

Napjainkra az üzleti folyamatmenedzsment (Business Process Management – BPM) olyan integrált és összefüggéseket kezelő megközelítési móddá, sokoldalú irányítási feladattá vált, amely egyidejűleg foglalkozik a szervezeti és a technológiai kérdésekkel. A hatékonyság fokozására számos vállalat alkalmaz már üzleti folyamatmenedzsment rendszereket, amelyek az átfutási idő csökkentésével valóban segíti a működési és szervezeti változások folyamatokba történő adaptálását.

Az üzleti folyamatok a vállalati erőforrások mozgatásával, felhasználásával, átalakításával szolgálják az üzleti célok elérését. Minden folyamat időhöz és térhez kötött, meghatározott be- és kimenettel rendelkezik. Az üzleti folyamat értéket ad a bemenethez, amely bemenet lehet a megelőző folyamatok végterméke is. Az üzleti

folyamat hatáskörén azokat a részfolyamatokat és tevékenységeket értjük, amelyek a folyamatot alkotják. A folyamat hatásköre és értékteremtő képessége között szoros összefüggés van: általában igaz az, hogy minél nagyobb a folyamat kiterjedése, annál nagyobb értéknövelés várható el tőle, valamint minél szerteágazóbb egy folyamat, annál nehezebben áttekinthető és menedzselhető! [Dobay, 1997]

Az üzleti folyamat az egy bizonyos objektum feldolgozásával összefüggő funkciókat, közreműködő szervezeti egységeket, az ahhoz szükséges adatokat és a kivitelezés menetét írja le. Ezen operatív folyamatoknak a vállalati célokhoz – költségcsökkentés, az átfutási idők csökkentése, ügyfél-orientáció, a versenyképesség javítása, stb. – való igazításához, megfelelő koordinációs- és információs támogatásra van szükség.

A BPR középpontjában a vállalat meghatározó üzleti folyamatainak racionalizálása, optimális kialakítása áll. Ez azt jelenti, hogy először felméri a jelenlegi vállalati folyamatokat, majd az eredmények alapján, ha szükséges, átszervezik őket. A felesleges folyamatokat megszüntetik, vagy összevonják más hasonló célt szolgáló folyamatokkal; a nem hatékonyan működő folyamatoknál megnézik, hogy hol lehet beavatkozni, és ott elvégzik a szükséges átalakításokat.

A BPR tevékenység azonban nem csupán a vállalat folyamatainak és folyamatszerkezetének egyedüli megváltoztatásáról szól. Legalább ennyire fontos az emberi tényező magatartásváltozásának elérése is. A BPR kivitelezése csak akkor lehet sikeres, ha maga után vonja az összes résztvevő megváltozott hozzáállását és készségét, hogy a folyamatosan változó kihívásoknak megfeleljenek, és az új követelményekhez dinamikusan alkalmazkodjanak.

Az üzleti folyamatmenedzsment (BPM) már túlmutat a BPR keretein is. A kifejezés arra utal, hogy már nem elég a folyamatok csupán egyszeri átszervezése, hanem folyamatos monitoringra és fejlesztésre van szükség a hatékonyság és versenyképesség megőrzéséhez. Ennek a koncepciónak a bemutatására a későbbiek során még részletesen kitérek.

A folyamatmenedzsment elméletek kimutatták, hogy matematikai értelmében mindaz, amit korábban számításnak (computation) és amit korábban kommunikációnak értettünk, azonos dologként - folyamatokként - értelmezhető és modellezhető. Az üzleti folyamatokat kezelő rendszerek (BPMS) a szoftver új kategóriáját jelentik, csakúgy, mint korábban a RDBMS volt. [R. Milner, 1999]

Az üzleti folyamatok menedzsmentjének tartományát az alábbi fő területekre oszthatjuk föl:

- Üzleti Folyamat Motorok: képesek egy adott vagy több alkalmazáson belül számos tevékenység végrehajtására vagy irányítására, mégpedig emberi feladatok / munkafolyamat (workflow) lépések bevonásával vagy ezek nélkül. Az ilyen motorok manapság lehetnek teljes mértékben centralizáltak vagy osztottak, egészen a folyamatlépések fizikai csomópontokra történő dinamikus kiosztásáig. Az Üzleti Folyamat motorok két fő típusát különböztetjük meg: az előíró (prescriptive) folyamat motorok és a szabályalapú motorok. Utóbbiak a Pi Matematikában gyökerező matematikai alapokon nyugvó esemény-alapú (/szabályokon alapuló) és a Petri Hálókból gyökerező matematikai alapokkal rendelkező klasszikus munkafolyamat motorok, amelyek a kontroll folyamatra helyezik a hangsúlyt. [H. Smith, P. Fingar, 2003]

- Formális specifikációk és az absztrakt üzleti/ügyintézési folyamatokat vagy interakciókat támogató számítógépesített reprezentációk. Vagy - más szóval – az üzleti folyamat motor által importálható, exportálható vagy végrehajtható formátumok meghatározása.

-Folyamatok és Interakciók modellezése amely két fő csoportra osztható:

- hangszerelések (orchestration) : a folyamat motor által végrehajtott folyamatokat írja le

- koreográfiák (coreography) : a folyamat leírása helyett az egyes szolgáltatások közötti interakciót írja le (vagy azt, hogy miképpen kell azokat egy meghatározott eredmény elérése érdekében használni)

- A több folyamat motor közötti interakciók elősegítésének eszközei, pl. valahol egy üzleti folyamat megindítása és a folyamat máshol történő folytatása; vagy egyes lépések távoli folyamat motorhoz delegálása.

- A komplex üzleti/ügyintézési folyamatok menedzselésének és monitorozásának eszközei, függetlenül attól, hogy azokat folyamat motorok hajtják végre, vagy informálisan valósulnak meg. (Üzleti Tevékenység Monitoring - Business Activity Monitoring, BAM).

A legnagyobb kihívás az üzleti folyamatmenedzsmentben az állandó, kétirányú megfeleltetés a szervezeti folyamattér két nézete

1. üzleti követelmények
2. IT rendszerek + erőforrások + emberi munka

között.

A folyamatok modellezése az információs rendszer területén végzett kutatásoknál tradicionális és jól megalapozott eredményeket mutat. A üzleti folyamatok grafikus modellezéséhez (UML, EPC, eEPC, BPEL,...) modellezési technikák és eszközök széles palettájából választhatunk. Az azonban elméletileg nem igazolható, hogy mennyire pontos, és teljes egy modell az adott üzleti folyamatról, vagy információs rendszerről. Különböző céllal, és különböző erőforrások felhasználásával készülnek. Gyakori az egyszerű, workflow-szerű ábrázolása az üzleti folyamatoknak, ami lényegében egy tevékenységssorra redukálódik. [Aalst, Pesic, 2006]

A BPM nem biztosít szemantikai szinten egységes reprezentációs hátteret a szervezeti folyamattér számára. Az egyes információs rendszer diszciplinák mögött sincs közös elméleti háttér. [Weber, 1997]

Az információs rendszerek alapkonstruációinak elméletileg megalapozott reprezentálására több kutatás is ontológiákat alkalmazott (pl. BWW: Bunge-Wand-Weber). Ezek a kutatások azonban elsősorban tradicionális, adatmodellező technikákra fókuszáltak. Wand és Weber még a filozófiai ontológia eszközeit javasolta információs rendszerek építéséhez. Bunge munkáit alapul véve kidolgozták a Bunge-Wand-Weber modelleket, amelyeket információs rendszerek modelljeinek elemzéséhez alkalmaztak. [Weber, 1997], [Recker et al., 2006]

Green és Rossman alkalmazott elsőként ontológiai elemzést a folyamatmodellező technikák terén, az ARIS-t vizsgálták. [Rosemann, Green, 2004]

A szemantikus üzleti folyamatmenedzsmentben (Semantic Business Process Management – SBPM) egy olyan új paradigma, amely növeli a két nézet közötti fordítás automatizálásának szintjét. [Hepp, Martin et al., 2005] A megközelítés lényege a két szint reprezentálása ontológiai nyelvek segítségével, majd automatikus fordítás gépi következtető mechanizmus alkalmazásával. Ezen új irányzat maga mögött tudhatja az ERP, BPM és web szolgáltatások vezető piaci képviselőinek támogatását. Egyelőre nincs közös megegyezés abban a tekintetben, hogy melyek az SBPM pontos reprezentációs követelményei, mely ontológiákra (OWL, WSML) és formalizmusokra lesz szükség, mire lesznek használhatóak, hogyan fognak a már elfogadott standardok (BPEL, EPC-k, ...) beilleszkedni egy ilyen keretrendszerbe. [W3C, 2005], [Bruijn, Jos et al, 2005]

Egy terület ontológiája a jellemző kategóriákat (fogalmakat, objektumokat, kifejezéseket), illetve a köztük fennálló kapcsolatokat írja le a jelentésükkel együtt. Az ontológiák alkalmazása ezen területeken azért jelentős, mert olyan kommunikációs környezetet biztosítanak, amelyben az adott terület fogalmai vitathatóak, egyértelműen elemezhetőek. Azonos területen működő közösség az ontológia közvetítésével azonos módon képes értelmezni és használni a közösen használt fogalmakat, objektumokat, tulajdonságaikat és relációikat.

A filozófia területén az ontológia építés során a fő kérdést, hogy mi az, ami létezik a cél pedig a valóságról megszerezni az összes igazságot.

Az információrendszerek esetében egy ontológia egy olyan formális nyelven leírt alkotás, amelyet valamely számítógépes környezet specifikus használatára terveztek. [Smith, 2002]

Korán felismerték annak szükségességét, hogy valamiféle szisztematikus módszer kellene találni a terminológiai és a fogalmi inkompatibilitás feloldására. Először konkrét esetenként próbálták ezt megoldani, majd rájöttek, hogy egy közös hivatkozási taxonómia ebben nagy segítséget nyújtana. Később ezt ontológiának kezdték nevezni. Ebben az esetben ez egy olyan szótár, amelyben a fogalmak kanonikus, hitelesnek elismert szintaxissal vannak leírva, és közösen elfogadott definícióval rendelkeznek. Az ismeretek reprezentálása lexikális és taxonómiai keretet jelent a különböző információrendszer-közösségek számára. Megfelelő axiómákkal megtámogatva (ahol az axiómák implicit definíciókkal vagy értelmezésre vonatkozó korlátozásokkal adottak) ez már formális elmélet.

Guarino (a FOIS (Formal Ontology and Information Systems) konferencia-sorozat elindítója) szerint az információrendszerek ontológiája mérnöki alkotás, amely egy bizonyos realitás leírását adó specifikus szójegyzék, továbbá a szójegyzékben szereplő szavakra vonatkozó explicit feltételezések halmaza. A legegyszerűbb esetben egy ontológia a tartalmazási relációt feltüntető fogalom hierarchia. Bonyolultabb esetben ehhez megfelelő axiómák is tartoznak, amelyek a fogalmak közötti relációkat, valamint a lehetséges interpretációs megszorításokat fejezik ki.

Az általa javasolt ontológia-építési módszer egyrészt az adatbázis-kezelő rendszereknél alkalmazott módszerekből, másrészt a logikában és az analitikus filozófiában használt (pl. axiomatikus) módszerekből származik. [Guarino, 1998]

Olyan általános, (tárgyterület-független) kategóriákat keres, mint idő, tér, beletartozás, példányosítás, azonosság, anyag, ok, mérték, mennyiség, funkcionális függőség, folyamat, esemény, attribútum, határ stb., ami tükrözi, hogy Arisztotelésztől kezdve több filozófus is hatással volt rá.

Az információrendszer ontológiájának készítése során nehéz megbirkózni az adatbázis problémáival. Az adat- és ismeretbázisú rendszerek építői egyéni jellemzők szerint dolgozzák ki rendszerük információ-reprezentációját, azaz ugyanazt a fogalmat másképpen nevezik, ill. ugyanazt a szót más fogalom megnevezésére használják. Az ilyen jellegű információ mennyisége egyre nő, megosztása és közös használata átfordítás nélkül lehetetlen. Az adatbázisok integrálásának problémája óriási méretű. Az ontológia-kiterjesztés jelentős akadályokba ütközik, hasonlóan, mint a világtörténelem közös ontológiájának kidolgozásakor. Utóbbi esetben egy semleges és közös keretben kellett (volna) a már megtörtént összes történelmi tényt, jogi és politikai rendszert, törvényt, hiedelmet, erőforrást leírni – ráadásul eltérő eredetű források felhasználásával. Egyetlen legmagasabb szintű ontológiát készíteni, majd széles körben felhasználni (Cyc projekt, amelyet végül feladtak) túl nagy kihívás. Egyrészt az ontológia építése sokkal bonyolultabb, mint kezdetben gondolják, másrészt az információrendszerek területe eléggé szubjektív és sokszor csak rövid távú horizonttal rendelkezik. [Lenat, Guha, 1990].

További problémát jelent az elfogadás szintje. Bármennyire is semleges és hitelesnek elismert egy ily módon kidolgozott, igen nagyszámú kifejezés definícióját tartalmazó ontológia, bármennyire is megegyeztek a különböző adatkezelő közösségek, a gyakorlatban nagyarányú ellenérdekeltség, ütközés van a követelmények között.

Az ontológiaépítés során két szintet különböztetünk meg:

1. Általános szintű (több szakterületen alkalmazható)
2. Szakterület-specifikus

A különböző forrásokból származó adatok összerendelése során szükséges a meta adatok szabványainak formalizálása, ahol a meta adatok célja rendszerezett módon információt szolgáltatni a felhasznált adatokról, minőségéről, eredetéről, természetéről és felhasználási módjáról.

Ahogy ontológiai módszerek szükségesek az adatbázisoknál az intelligens információ-integrálás során az adatszintű integrációhoz, ugyanúgy ontológiai módszerek

szükségesek az együttműködő információrendszerek kidolgozásánál az alkalmazásszintű integrációhoz is. Az alkalmazások integrálása szemantikai szempontból a különböző ontológiák automatikus integrálását jelenti.

Két rendszer egyesítésekor pontosan meg kell adni azok kapcsolatát, valamint hogy ugyanazt a dolgot a két rendszer hogyan dolgozza fel. Ebben az esetben egy közös fejlesztés azzal jó, ha kezdődik, hogy kidolgoznak egy közös ontológiát. Átfogó információrendszerek fejlesztését közös ontológia kidolgozásával, a meglévő ontológiák egyesítésével célszerű kezdeni!

A szoftverfejlesztés területén az ontológiával kapcsolatban problémát jelent a jól ismert zárt világ feltételezés. Adatbázisoknál, programoknál például ez azt jelenti, hogy feltételezzük, hogy az adatbázis/program a tárgyterület objektumairól szóló összes pozitív információt tartalmazza, azaz amit nem tartalmaz, az hamis információ. A zárt világ feltételezés nem csak azt jelenti, hogy csak azok az entitások léteznek, amelyeket reprezentáltunk, hanem hogy ezek az objektumok csak olyan tulajdonságokkal rendelkeznek, amelyeket reprezentáltunk a rendszerben.

A zárt világ feltételezés alapján megfogalmazott modellek mind egyszerűbbek a valóságnál. A gyorsan változó világ ontológiájához el kell vetni a zárt világ feltételezést, ami ez esetben azt jelenti, hogy a szoftverfejlesztés sokkal nehezebb.

Valós, létező tárgyterület reprezentálásánál szisztematikusan meg lehet feleltetni az ontológiai kategóriákat valós entitásoknak.

Egy üzleti alkalmazás esetében azonban, ahol semmi más realitás nincs, csak amit a rendszerbe építettek, az az üzlet létezik, amelyet a rendszer kezelni tud. Az üzleti világ a rendszeren belül létezik, az a rendszeren belüli eseménysorozattal reprezentálható. Ha egy információrendszer-ontológia környezete fejlődik (pl. e-kereskedelem), a programnak ezt követni kellene ahhoz, hogy a rendszer kezelni tudja az új entitásokat. Ezen a területen robusztus és konzisztens ontológiai hierarchiát nehéz alkotni, pragmatikus lesz.

A vállalati ontológiák (Enterprise Ontology), a vállalat koncepcionális modellje és folyamatai széleskörű és gazdag ábrázolása területén lényeges eredmények születtek. A hatékony üzleti tervezés és integráció elérésének egyik módja, hogy valamennyi résztvevő (az üzleti menedzserektől a szoftverfejlesztőig) a vállalat megfelelő aspektusát átlátja, kialakul a vállalat “megosztott áttekintése, megértése”. Különösen, amikor bizonyos kifejezéseket egy adott szöveggörnyezetben használnak fontos, hogy valamennyi résztvevő tisztán lássa melyik fogalomról van szó.

A szervezeti ontológia e cél elérése érdekében került kifejlesztésre; nagyszámú, a vállalatok leírására használatos kifejezés definícióját tartalmazza. Olyan egységes kifejezés és definíció halmaz létrehozása volt a cél, amely pontosan lefedi a vállalati modellezés során használatos fogalmakat, így a kifejezések használata során a félreértések elkerülhetők. A szervezeti/vállalati modellezés elsődleges célja, hogy a szervezetről egy olyan vállalati szintű áttekintést adjon, amely a döntéshozatal alapjául szolgál. A szervezetet nem tradicionális módon szemléli, hanem azokat a tartományokat alapul véve, amelyeken a szervezet működik.

Az ontológia elkészítésének elsődleges célja az üzleti tervezés javítása, a rugalmasság növelése, a hatékonyabb kommunikáció és integráció, alkalmazkodás a gyorsan változó üzleti környezethez. A fejlesztés során módszerek, eszközök integrációjára alkalmas, vállalati modellezésre létrehozott keretrendszer kialakítása történt meg.

[Fox, Gruninger, 1998], [Fox et al, 1998], [Gruninger et al., 2000], [Dietz, 2006].
Uschold, King, Moralee és Zorgios olyan szervezeti ontológiát fejlesztettek ki, amely a szervezeti modellezés egy keretrendszere lehet [Uschold et al., 1997].

A szervezet számára fontos, hogy üzleti folyamatai találkozzanak a változó környezet elvárásaival, céljaival.

A folyamat modellek azonban nem tartoztak ezen kutatások és közösségek érdeklődésének a középpontjába, mint ahogy a vállalati ontológiák modelljei sem használhatóak a jelenlegi BPM infrastruktúrákkal és eszközökkel.

A BPM jelenlegi fejlődése erősen kapcsolódik a Világhálós Szolgáltatások és a Szolgáltatásorientált Architektúrák (SOA) fejlődéséhez, a technológiák párhuzamos fejlődése következtében, de azért is, mert a szolgáltatásorientált megközelítések jól illeszkednek az osztott folyamatok architektúráihoz. Egyes folyamat menedzsment nyelvek - mint pl. a BPEL - ráadásul csakis az Internetes szolgáltatásokon alapulnak - amelyeket sokan tekintenek túlzottan korlátozónak és korlátozottnak. A termékek viszont a feldolgozó erőforrásokkal folytatott interakciók számos formája előtt nyitottak.

A második generációs BPM eszközök, a „Business Process Management Suite”-ok legfontosabb jellemzője, hogy egységes, a SOA-infrastruktúrába illeszthető eszközkészletet adnak, támogatva az üzleti folyamatok implementációját a modellezéstől a szimuláción át az üzleti aktivitás monitorozásig. Az eszközök elsődleges feladata a rendszerek, valamint a humán erőforrások és szervezeti egységek között átívelő üzleti folyamatok megvalósítása, futtatása és monitorozásának támogatása.

Az üzleti folyamatok életciklusa a modellezéssel, tervezéssel kezdődik. Ebben a fázisban különösen fontos az informatikai és üzleti területek szoros együttműködése. A BPMS-nek olyan tervező/modellező eszközt kell adnia, amely az UML és BPMN modellezési szabványokat grafikus felülettel is támogatja. Az eszköznek emellett biztosítani kell, hogy az üzleti elemzők kódolási, illetve az informatikai rendszerek mély technikai ismerete nélkül legyenek képesek megtervezni és szimulálni az üzleti folyamatokat. Amikor a folyamat üzleti szempontból teljessé válik, átkerül az IT-hez, ahol a fejlesztők elvégzik a folyamat implementációját. Fontos, hogy az átjárás az üzleti elemzők által készített modell és az implementáció ne váljon el egymástól, vagyis a BPMS modellező és implementációs eszköze ugyanazon a fizikai struktúrán és formátumon tudjon dolgozni. Az üzleti aktivitás monitorozása által szolgáltatott adatok alapján az üzleti folyamatok módosíthatók, és ha a modell és az implementáció ugyanarra a fizikai struktúrára van leképezve, a folyamatfejlesztők azonnal reagálhatnak a változásra, vagyis az IT és az üzlet szorosabb kapcsolatba kerül.

Jelenleg nem létezik valódi hivatalos BPM standard, de számos javaslatot és tervezetet dolgoztak ki. A pi matematika (esemény alapú folyamat modellek) és a (helyekből, átmenetekből, és a helyeket az átmeneteken keresztül összekötő ívekből álló) petri hálók támogatói közötti küzdelem gátolja a BPM standardok fejlődését. Az egyik oldalon ott a BPMI és az OASIS konzorciumok által támogatott BPEL, a másik oldalon ott a WfMC konzorcium által támogatott XDPL. Mindkettő XML alapú szabvány. A BPEL egy process orchestration nyelv, vagyis az üzleti folyamatokat tipikusan egy üzleti szereplő szemszögéből írja le. Ezzel szemben az XPDL process choreography nyelv, amelyben több üzleti szereplő közötti üzleti folyamat modellezhető. Mára bizonyítottnak tűnik, hogy a pi matematika hatásosabb (különös tekintettel a hibakezelésre, a kompenzációra és a dinamikus kompozícióra), jóllehet a formalizmus a kevésbé intuitív koncepciók megismerését is megköveteli. A grafikai interfészek a pi matematika legfurcsább vonásait javarészt elrejtik, olyannyira, hogy kevés felhasználó veszi észre, hogy ilyen elméleti háttérrel dolgozik, miközben a jelenleg a piacon elérhető termékeket használja.

[H. Smith, P. Fingar, 2003]

A BPEL4WS vagy egyszerűbben BPEL a BEA, az IBM, a Microsoft által javasolt standard. A támogatók köréhez később csatlakozott a SAP AG, a Siebel Systems, az Oracle és mások is. A javaslat az OASIS-hez került és az eredeti szerzők számos más javaslatának helyébe állt (különös tekintettel az Xlang (Microsoft) a WSFL (IBM) rendszerekre). A javasolt standardból számos jellemző hiányzik például a változók, az állapot, az iniciálás, a tranzakciók, a tömbök és dinamikus listák, amelyeket a gyakorlatban előforduló esetekben kellene használni.

A WSCI a Web Service megfigyelhető viselkedésének leírásával egészíti ki a BPEL-t. A BPML számos vonatkozásának magába építésével a WSCI az Internetes szolgáltatások koreográfiájára, vagyis a kicserélt üzenetek közötti időbeli és logikai függőségekre összpontosít, a sorba állítási (sequencing) szabályok, korrelációk, a kivételek kezelése és a tranzakciók kiemelésével. A WSCI ennél együttműködőbb megközelítést igényel, megkövetelve, hogy az üzenetváltásban érintett mindkét fél meghatározzon egy WSCI interfészt míg a BPEL4WS fordított perspektívával él, és a végrehajtandó folyamatot

valamelyik résztvevő szemszögéből írja le. A WSCI széles körben hivatkozik a WDSL-re.

A BPMI (Business Process Management Initiative) konzorcium által létrehozott a BPML (Modellező Nyelv), a BPMN (Modellező Fogalom), a BPQL (lekérdezési nyelv) a BPELhez viszonyítva érettebb specifikációk körét adja, és számos közös tag támogatja. A BPEL az Internetes szolgáltatásokra fókuszál, míg a BPML általánosabb és elvontabb rendszer.

Az XPDL a BPEL ellentéte! A rendszert a WfMC (Workflow Management Coalition), egy a workflow motorokkal összekapcsoló interfészek - interoperációjának lehetővé tételére irányuló - egységesítésén fáradozó szervezet támogatja. Az XPDL alapja a kontroll folyamat modell, szemben az eseményalapú BPEL rendszerrel. Az XPDL rendszerben a tevékenységeket annak érdekében hozzák egymással kapcsolatba, hogy azok egy a tranzakció által védett átmeneteken keresztül egy kontroll folyamatot alkossanak.

A Wf-XML kiegészíti az XPDL rendszert és meghatározza, hogy milyen módon jeleníthet meg a folyamat más folyamatokat (más szervereken vagy munkafolyamat végrehajtó motorokon), lekérdezheti az ilyen folyamatokat az állapotuk tekintetében és más módokon is szabályozhatja (kontrollálhatja) azokat. A lekérdezések üzenet-alapúak, de eltérnek a WSDL BPML és BPEL általi használatától.

Az ebBPSS az ebXML standardsorozat Üzleti Folyamat Specifikációs Sémája. A standardok az üzleti/ügyintézési tranzakciók (Business Transactions) üzleti/ügyintézési együttműködéseké (Business Collaborations) alakításának koreográfiáját szabályozzák.

A BPEL mögött jelenleg álló nagynevű cégek bizonyosan elérik, hogy e tervezet jövőbeni verzióiból váljon standard annak ellenére, hogy a jelenlegi formájában erősen korlátozott képességekkel bír. A BPMN is komoly vállalatok által támogatott grafikai rendszer.

Függetlenül attól, hogy a BPM kialakítás melyik iskoláján alapulnak, a jelenlegi BPM motorok meglehetősen kinyilatkoztató jellegűek és erősen rögzített folyamatsémákat javasolnak, amelyek nem képesek alkalmazkodni a nagy és folyamatosan mozgó

környezetekhez. A szolgáltatásorientált architektúra és különösen az Internetes szolgáltatások jelentős mértékben hozzájárultak a jelenlegi BPM technológiák korlátainak felfedéséhez, amelyekkel a vállalati kereteken kívüli alkalmazásukkor kerülhetünk szembe. Különösen fontos e tekintetben, hogy a „külső” környezet dinamikus jellege nehezen követhető, a dinamikus felfedező képességek (amelyek ráadásul a szolgáltatás helyszínének és címének megoldására korlátozódnak) meglehetősen korlátozottak, sőt, a rendszerek többségéből teljességgel hiányoznak. A központi folyamatszabályozó az osztott környezetek többségében a „szerződési folyamatokra” helyezi a hangsúlyt a „végrehajtó folyamatok” helyett, amelyre pedig a rendszerek többségét hangolták. A hosszan tartó, lazán összekapcsolt tranzakciók támogatása gyenge és mintegy manuálisan kell a folyamat logikájába huzalozni. Ha több üzleti partnerrel van dolgunk, még akkor is, ha ugyanazt a tevékenységet végezzük, a folyamatmodellek gyorsan túlzottan kuszává és bonyolulttá válnak a fenntartáshoz. Az alapfolyamattal szembeni kivételeknek és az alapfolyamat változatainak kifejezése nem lehetséges.

Az előrelépést a szolgáltatások meghatározásának szemantikai kezelése és a célorientált folyamatok felépítése (szemben a deklaratív folyamatokkal) jelentené.

A legtöbb jelenlegi BPM rendszer nagymértékben az Internetes szolgáltatások kezelésére irányul. Ez a trend egyre erősebbé válik, és úgy tűnteti föl az Internetes szolgáltatásokat, mint a termékbe épült komponensek (pl. adatok átalakítása) és minden, a korábbi alkalmazásokból megmaradt, újra felhasználható logika vagy erőforrások univerzális interfészét.

A jelenleg a kereskedelmi forgalomban elérhető BPM technológiák egyike se mutat olyan képességeket, amelyek segítségével ontológiákat használhatnának a web szolgáltatások futtatási idő során történő dinamikus összeillesztésére. Egyelőre egyik se képes megfelelő érveket felhozni a követendő útvonal mellett és egyik sem célorientált ahelyett, hogy deklaratív lenne. Az egyetlen, bizonyos mértékű rugalmasság a web szolgáltatás helyszínének futtatási idő során való dinamikus megoldásával kapcsolatos. Ez

már jelentős előrelépés, de messze van még azoktól a képességektől, amelyekre egy nagyléptékű és teljesen osztott infrastruktúra igényelhet. [Aalst 2006]

Egy alternatíváktól és kivételektől zsúfolt környezetben a folyamatok dinamikus felfedezésére, a velük való interakcióra és a folyamatok végrehajtására képes intelligensebb osztott rendszerek előnyei kihasználásának titka az ontológiák és az azokat manipulálni képes rendszerek fejlesztésében rejlik.

„A metaadatok megértése, kezelése, kontrollja és újrahasználata a SOA lehetővé tételének kulcsfontosságú része. A metaadatok kezelése emellett kulcsfontosságú számos alkalmazás üzleti és technikai nézetei közös megközelítésének biztosításában.” [Gartner, 2006]

A „szemantikus web” következtetéseket lehetővé tevő metaadat struktúra a weben, kiterjeszti a web jelenlegi lehetőségeit (de nem helyettesíti azt). A szemantikus web fogalma Tim Berners-Lee-től származik:

„A szemantikus web a hagyományos web egy olyan kiterjesztése, amelyben az információnak jól definiált jelentése van, és így hatékonyabban támogatja a számítógépek és a felhasználók közötti együttműködést.” [Berners-Lee, et al. 2001].

A szemantikus web kutatásokat a World Wide Web Consortium (W3C) kezdeményezi és támogatja.

A szemantikus web a World Wide Web-en használatos adatrepresentáció. A W3C által vezetett szemantikus web kutatási és fejlesztési együttműködés, akadémiai és ipari közreműködőket egyaránt foglalkoztat. Alapja a Forrásleíró Keretrendszer az RDF (Resource Description Framework), magában foglalja az XML szintaxist és az URI-t (Uniform Resource Identifier) is, az elnevezésekre.

A szemantikus web projekt céljai közé tartozik, hogy a web-es dokumentumokat a számítógép által is értelmezhető jelentéstartalommal töltsse fel. Ez lehet az alapja többek között a szó-ontológia keresőrendszerek kialakításának, amelyek segítségével teljes mondatok alapján is lehet keresni az Interneten. A web-es forrásokban információk

megtalálását segíthetik az ontológiák, amelyek tartalomtól és kontextustól függő osztályozáson keresztül segítik a keresést.

A W3C által preferált OWL (Web Ontology Language) nyelv válik várhatóan a szemantikus web legfontosabb leíró nyelvévé. Az OWL nyelv az RDF, a Darpa Agent Markup Language (DAML), és az OIL (Ontology Inference Layer) nyelvekre épül.

Az eXtensible Markup Language (XML) egységes adatcsere formátumot biztosít, hatékonyan támogatja a web tartalom kialakítását, de nem megfelelő az objektumok közötti kapcsolatok leírására. A szemantika területére irányuló valódi ugrást az RDF (Resource Description Framework, forrásleíró keretrendszer) jelenti. Az RDF számos metaadat közösség - köztük a Dubli Core és a PICS - eredménye. Az RDF támogatja a szemantikai reprezentációt, de nem ad származtatott kvantitatív és kvalitatív információkat. Az RDF olyan modellt kínál a metaadatok bemutatására, amely jelenleg számos egyéb specializált standard alapja. Az RDF nyelv egy továbbfejlesztése az RDFSchema, amely adatrepresentációs forma is egyben. Lehetővé teszi a fejlesztő számára, hogy egyedi szókészletet definiáljon az RDF adat számára és meghatározza azoknak az objektumoknak a fajtáját, amelyekre ezek a tulajdonságok érvényesek.

A következő szint tartalmazza az ontológia szótárakat és modellező nyelveket (pl. OIL). Ezek a megoldások támogatják a W3C szabványok többségét, így a különbözően strukturált dokumentumok közötti együttműködést is.

OWL-S (OWL for Semantic Web Services) a dinamikus web szolgáltatás felfedezés szolgáltatási profilok koncepcióján alapuló pontos ontológiája. [Martin et al., 2005] Ez az ontológiáknak a web szolgáltatások felfedezések pontos tartományára történő gyakorlati alkalmazása. Annak ellenére, hogy már most létezik javaslat a tervezett standardra, egyelőre csak kutatók laboratóriumaiban léteznek.

Az SWRL eszközöket biztosít az OWL tudásbázis további szabályokkal való kiegészítéséhez, és az intelligens ágens absztrakt tartalmának valamilyen módon történő befogásához. Az SWRL értelmezésére szintén csak laboratóriumi rendszerek állnak rendelkezésre.

A WSMF (Web Service Modeling Framework) [Fensel, Bussler, 2002] egy még átfogóbb keretet ad, aminek a továbbfejlesztése a WSMO (Web Service Modeling Ontology). [Roman et al., 2005] A WSML (Web Service Modeling Language) pedig a WSMO –t támogató nyelvcsalád . [Bruijn, Jos et al, 2005]

Ahhoz, hogy a szemantikus web és az erre épülő alkalmazások sikeresek legyenek, meg kell találni a megfelelő egyensúlyt a szabványosítás kötöttségei és az „alulról jövő” építkezés között. A szemantikai háló területén meglévő standardok nem értelmezhetőek a vonatkozó technológiák érettségének jeleként. Ahogy a TCP/IP jobban elterjedt, mint az ISO-OSI szabványok, illetve ahogy a HTML jobban elterjedt, mint az SGML, ugyanúgy a szemantikus web szabványoknál is fontos, hogy minél több alkalmazói próbálkozás legyen, és a felhasználók motiváltak legyenek, hogy másokkal megosszák a saját, adott területre vonatkozó ontológiájukat. A metaadat tartalmak távolról sem közösen használható tartalmak. Paradox módon a metaadatok révén bizonyos mértékű integrációt hatékonyan megvalósító szervezetek éppen a saját tulajdonú készletet használó szervezetek, mert a standardok a képességek tekintetében elmaradnak. Ha már kellően elterjedt a technológia, akkor a vállalkozások is profitálnak belőle.

A szemantikus üzleti folyamatmenedzsment egyesíti a szemantikus webszolgáltatások keretrendszerét, az ontológiai infrastruktúrákat és az üzleti folyamatmenedzsment módszertanokat és eszközöket egyesített technológia kifejlesztéséhez. Az alapvető megközelítés a vállalat üzleti és rendszer nézetének együttes reprezentálása ontológiák alkalmazásával, és a két szint közti megfeleltetés támogatása gépi következtető motorokkal.

A Gartner kutatóintézet feljövő technológiákról szóló riportja szerint a technológia éppen a Hype görbe „kiábrándulás völgyében” látható.(10. ábra) A prioritás-mátrix a technológia várható hasznát magasra becsüli. (11. ábra) Azonban mind a Hype görbéről, mind pedig a prioritás-mátrixról azt olvashatjuk le, hogy a technológia középpontba kerüléséig várhatóan még több, mint 10 évig kell várni. A rendszerek tehát várhatóan 10 év múlva válnak képessé arra, hogy web szolgáltatásokat fedezzenek fel maguknak és

azokhoz alkalmazkodjanak. Ezen „funktionalitás” magasabb szintű képzettség nélküli elérhetősége még kérdéses.

A jelenlegi szemantikus üzleti folyamatmenedzsment kutatások a meglévő eszközök és sztenderdek kompatibilis együttműködtethetősége irányába történnek. Ez azt jelenti, hogy egy BPEL, vagy egy EPC modell felhasználható legyen egy SBPM környezetben, ugyanakkor egy SBPM rendszerben létrehozott folyamat legyen exportálható BPEL-be, hogy a standard BPEL motorok végre tudják hajtani.

Egy egészen új terület a szemantikus middleware menedzsment, amelyik a szemantikus web szervizek és a vállalati IT területek menedzselésének kombinálása. [Oberle, 2006]

A jelenleg piacon lévő alkalmazási platform programcsomagok APS (Application Platform Suites) prezentációs (Portál), integrációs (EAI, B2B) és alkalmazásfejlesztési környezeteket kínálnak BPM és workflow képességekkel. A különböző információtechnológiai szolgáltatásokat moduláris logika szerint szemléli és a vállalati alkalmazásokat egymástól függetlenül is működő, mégis szorosan integrálható részekre bontva kezeli. Ahhoz, hogy a különböző rendszerek integrálhatóak legyenek, két alapvető problémát kell megoldani: az alkalmazások képesek legyenek kommunikálni egymással, és megértsék egymás nyelvét. Ezeket a feladatokat pedig a korszerű platformok és a szabványosítás együtt tudják ellátni. Az ARIS ezen middleware platform programcsomagok létfontosságú összetevője. Az ARIS eszköztár már rendelkezik standardizált terminológiával és újra felhasználható koncepcionális modellekkel.

A BPMN (Business Process Modeling Notation) – jelölésrendszer

A BPMN az üzleti folyamatok és Web szolgáltatások modellezésének szabványos jelölésrendszere.(2005 június óta OMG specifikáció) Tervezésekor szempont volt a tipikus üzleti folyamatok egyszerű modellezhetősége, és az összetett üzleti folyamatok

ábrázolása is. A BPMN ezen túl használható a közvetlen végrehajtásra alkalmas BPEL4WS generálásra is.

A BPMN a BPD (Business Process Diagram) által biztosítja a külső, és belső folyamatok definiálását. Alkalmazásával szabványos módon, írhatók le az eljárások és kommunikálhatók más szervezetek számára is. A BPD és az UML 2.0 tevékenység diagrammja közt nagyon sok hasonlóság van ugyanakkor a részletek, például az összetettebb elágazások ábrázolása, és értelmezése jelentős különbségeket mutat.

A modellező nyelv alapelemei:

- Esemény – valamilyen választevékenységet igényel a bekövetkezése. Hatással van a folyamat lezajlására, elindíthat, megszakíthat, vagy megállíthat folyamatokat. Az eseményeket egy kör jelöli.
- Folyamat – tevékenységek határozott kezdettel, és véggel rendelkező sora, amelyeket a cél elérése érdekében hajtunk végre. A modell alapelemei maguk a folyamatok. A folyamatok végrehajtási sorrendjét nyíl jelöli.
- Csomópont – a munkafolyamat során bekövetkező vezérlési elágazásokat jelöli. Döntések, elágazások, vagy egyesülések jelölésére a rombusz szimbólumot használja. Ezeknek a csomópontoknak meghatározható a működés logikája, és ezt a szimbólum is megjeleníti. Az elágazást meghatározhatja valamilyen feltétel, vagy valamely esemény bekövetkezése is. Alkalmas a párhuzamosan futó folyamatok modellezésére is.
- Sorrend – a tevékenységek végrehajtási sorrendjét mutatja egy folyamaton belül (használható feltételes változat is) A feladatok végrehajtási sorrendjének ábrázolása egy adott szervezeten belül nyíllal történik. Ezek csak folyamatok, események, és csomópontok között fordulhatnak elő.

- Üzenet – két szereplő közti üzenetfolyamokat jelöli.
- Asszociáció – a folyamat objektumaihoz kapcsolható vele további információ

Az ún. Pool-ok (medencék) és lane-ek (sávok) segítségével részletesen meghatározható, hogy melyik folyamat pontosan hol történik, és ki hajtja végre. A pool többnyire egy szervezet jelöl, a lane pedig egy részleget az adott szervezeten belül. Az egyes folyamatokat a megfelelő pool-ba, vagy lane-be helyezve egyúttal meghatározzuk a folyamat végrehajtóját, vagy a végrehajtás helyét. Az események tekintetében pedig, hogy hol vagy kik hozzák a döntéseket. Egy ilyen pool-t, mint valamilyen erőforrással rendelkező szervezetet lehet elképzelni. Amennyiben más erőforrásokra van szükség, úgy másik pool-ban folytatódik a folyamat.

Egy pool sok mindent jelenthet egy adott környezetben (szervezeti egység, szerep, alkalmazás, helyszín, szoftver komponens, entitás). Amennyiben a folyamat más szervezetet is érint, tehát másik pool is részt vesz benne, úgy azok közt üzenetek használunk. Élesen elkülönül az egyazon pool-hoz tartozó folyamatok, események, és döntések közti átmenetek, a pool-ok közti üzenetektől. Különösen B2B folyamatok leírására alkalmas.

Olyan esetekben, amikor nem fontos, vagy nem ismert a másik szervezet belső működése, csak az számít, hogy milyen üzenetváltások történnek, a másik szereplőt „fekete dobozként” kezeljük. Ebben az esetben a megfelelő üzenetek sem a másik pool belsejébe, csak annak határára mutatnak, illetve onnan érkeznek.

A szervezetben zajló folyamatok során kapcsolódó dokumentumok, adatok keletkeznek, és változnak. A folyamatban megjelenő dokumentumok és adatok modellezésére az adat objektum szimbólum szolgál a BPMN-ben. A folyamatban hivatkozott adatelemek állapota szögletes zárójelben jelölhető az adatelem neve alatt.

A modellezés során bármelyik diagram elemhez fűzhető szöveges kiegészítés. Ez nem helyettesíti a modellező eszközben a folyamatokhoz, csomópontokhoz tartozó egyéb, akár a szimulációt, erőforrásigényt meghatározó jellemzőt, inkább a diagrammon ábrázolt folyamatot igyekszik érthetőbbé tenni.

A csoport szimbólum folyamat elemek vizuális csoportosítására szolgál. Elsősorban a folyamat bizonyos részeinek kiemelésére szolgál, például mikor nincs szükség azt alfolyamatként másik diagrammon ábrázolni. Valójában csak dokumentálási, vagy elemzési célja van, magát a folyamatot nem befolyásolja. Használható például elosztott, több pool-ban lezajló tranzakció műveleteinek jelölésére.

A BPMN nyelv eszközeivel a szervezeti működés több szinten modellezhető és ezen szintek kapcsolódásai is megvalósíthatóak:

- Magas szintű belső folyamatok ábrázolása részletezés nélkül
- Részletes belső munkafolyamatok modellezése
 - A jelenlegi folyamat
 - A tervezett új folyamat
- Részletes belső folyamat, melynek kapcsolatai vannak egy, vagy több külső szereplővel („fekete doboz”)
- Két, vagy több részletes belső folyamat közti interakció
- Részletes belső folyamat kapcsolata egy absztrakt folyamattal
- Részletes belső folyamatok együttműködnek egy absztrakt folyamaton keresztül
- ...

II.6 Az ARIS módszertan

Az ARIS mozaikszó, az „Architektur integrierter Informationssysteme” német szavak kezdőbetűiből áll össze, ami magyarul azt jelenti, hogy az „Integrált Információrendszerek Architektúrája”. Az ARIS tehát leírja a vállalati információkat

integráltan összefogó és kezelő, ezáltal a hatékony működést lehetővé tevő rendszerek felépítését, architektúráját.

Az ARIS módszertan és eszközcsalád segítségével lehetségessé válik az üzleti folyamatok támogatását szolgáló informatikai rendszerek, illetve teljes vállalatok átfogó, folyamat-centrikus leírása.

A tudományos alapokon kifejlesztett ARIS termékcsaládot és módszertant az IDS Scheer vállalat alkotta meg, és fejlesztette a folyamatszervezési világpiac vezető eszközévé. A céget 1984-ben Prof. Dr. August-Wilhelm Scheer hozta létre, a Saar vidéki Egyetem munkatársaival együtt.

II.6.1. Az ARIS koncepció háttere

Az ARIS koncepció megalkotásakor a vállalat víziója az volt, hogy egy olyan közös nyelvet hozzanak létre az informatikusok, a tanácsadók és a menedzserek között, amely könnyen érthető, egyszerűen elsajátítható, és ezáltal segíti a hatékony munkavégzést. Ennek szellemében az ARIS egy olyan egységesített grafikus leíró nyelv, amely szemléletesen dokumentálja a működést, a funkciókat, adatokat, szervezeteket és folyamatláncokat. A modellezési tevékenységet strukturált módszertannal, eszköztámogatott eljárásokkal és speciális referenciamodellekkel támogatja.

Az ARIS Toolset programcsomag az ARIS koncepció gyakorlati megvalósításához szükséges eszköztár. A vállalat alaptermékének tekinthető szoftvert 1993-ban hozták forgalomba. A programcsomag több komponensből épül fel (Modellezés, Analízis/Szimuláció, Tevékenység-alapú költség számítás, Balanced Scorecard, Jelentéskészítés, stb.), amelyek alkalmazása segíti a felhasználókat az összetett üzleti folyamatok modellezésében, az elkészített modellek sokoldalú kiértékelésében és elemzésében. Hozzájárul az üzleti folyamatok és a vállalati működés jobb megértéséhez. [Scheer – Abolhassan – Jost – Kirchmer, 2002]

II.6.2. Az ARIS koncepció alapelvei

Az ARIS a vállalati működést, az üzleti folyamatokat modellező és elemző, folyamatelvű módszertan, és az ezt támogató komplett informatikai eszköztár. Ahhoz, hogy ez a modellezési koncepció az összes felmerülő vállalati információs igénynek eleget tudjon tenni, egységes szabályok szerint kell működnie. Ezt a szabályrendszert két fő rendező elv mentén alakították ki: a szétválasztás elve és a leíró szintek elve szerint. Az ARIS architektúra kialakításának kiindulópontját, mindkét esetben, egy, az üzleti folyamatok leírását szolgáló vállalatmodell képezi. [Scheer, 1993]

A szétválasztás elve a vállalati működés komplexitásának csökkentése érdekében különböző statikus leíró nézetekben vizsgálja meg a vállalatot (adat-, szervezeti és funkcionézet). Az egyes nézetekben különféle modell típusokat használ a vállalati működés ábrázolására, amelyeket végül egy dinamikus (irányítási-) nézetben kapcsol össze egy teljes modellé.

A folyamatok tervezése során általában több lépésen keresztül juthatunk el a ténylegesen kiválasztott folyamatokig. Az elemzések megkönnyítésére különböző mélységig részletezett modellekre van szükségünk, ezáltal ugyanis jobban áttekinthetőek a folyamatok. Az IDS Scheer vállalat által kidolgozott leíró szintek elve szerint háromféle leírási szinten vizsgálhatóak az információrendszerek (szakkonceptió, adatfeldolgozási koncepció, műszaki megvalósítás). A leíró szintek az információ-technológiához való közelségük szerint rendeződtek. Az általános IT szempontú szervezetleírástól haladnak a felhasználók által igénybe vehető konkrét hardver- és szoftver eszközök részletezéséig. [Scheer – Abolhassan – Jost – Kirchmer, 2002]

Az ARIS architektúra a leíró nézetek és a leíró szintek összességéként keletkezik. Az egyes komponensek mindegyikéhez különféle leíró módszereket lehet alkalmazni modellek formájában.

Az ARIS módszertan további fontos alapelve, hogy a vállalati folyamatstruktúra kialakítását top-down módszerrel kell végrehajtani. Ennek megfelelően a vállalati folyamatok legfelsőbb szintje az adott vállalat főbb folyamatszoportjait ábrázolja. Ebből

a fő áttekintő modellből kiindulva a főtevékenységeket, kisebb logikai egységekre kell lebontani, és további 1-2 áttekintő modellezési szinten kell őket részletesebben ábrázolni. A cél, hogy a „legalsó” áttekintő szint funkciói olyan logikai blokkokat képezzenek, amelyeket egyértelműen ki lehet fejteni részletező modellek formájában.

II.6.3. Az ARIS nézetei és a tipikus modellek

A vállalatok leírására nézetenként különböző modell típusok állnak a rendelkezésünkre. Ezek a modell típusok valójában egyfajta sablonok, amelyek alkalmazásával elkészíthetjük a saját vállalatunkra szabott leíró modelljeinket.

A modellek úgynevezett objektumokból épülnek fel, amelyeket a modellezés során kell meghatározni. Ilyen objektumok például az események, folyamatok, felhasználók, szervezeti egységek, IT erőforrások, alkalmazási rendszerek, stb.. Az objektumok megjelenítésére előre definiált szimbólumokat, grafikus jeleket használhatunk. Ezek biztosítják, hogy az ARIS nyelv használata világszerte egységes és könnyen elsajátítható legyen.

Az objektumok meghatározása után definiálnunk kell az objektumok között fennálló kapcsolatokat is, amelyeket nyilak illetve vonalak segítségével ábrázolhatunk grafikusán. Az objektumok közötti kapcsolatok az egyes nézeteken belül igen erősek, a nézetek között relatív egyszerűek és lazák. Így az egyes nézetekben a modellek független elemzésére nyílik lehetőség. Mind a modellekhez, mind az objektumokhoz és kapcsolatokhoz meghatározhatunk attribútumokat, amelyek az elemek tulajdonságainak tárolására alkalmasak.

Az adat nézet

Az üzleti tranzakciókkal, tevékenységekkel, folyamatokkal és a közöttük fennálló összefüggésekkel kapcsolatos adatokat, amelyeket a kialakítandó rendszernek kell majd kezelnie, az ARIS adat nézetében ábrázolhatjuk. Az adatok általában az absztrakció különböző szintjeinek használatával vannak megjelenítve.

A modellezés során az adatstruktúrák megtervezése az egyik legmeghatározóbb és legmesszebbre ható tevékenység. Csak a logikusan felépített adatstruktúra esetén képzelhető el a folyamatok hatékony számítógépes támogatása.

Kibővített egyed kapcsolat modell – eERM (extended Entity Relationship Model)

Az adat nézetén belüli részletes modellezésnél a leggyakrabban használt modell típus az eERM. A modell legfontosabb objektumai az egyedek – olyan valós dolgok, ill. Elvont fogalmak, amelyek a vizsgált vállalati folyamatok és azok megvalósításához szolgáló tevékenységek szempontjából jelentőséggel bírnak, a kapcsolatok – amik jelzik az entitások közötti logikai összefüggéseket, és az attribútumok – amik képviselik az egyedek és kapcsolataik konkrét tulajdonságait.

Adatmodellezéskor a hasonló tulajdonságokkal rendelkező információs objektumokat egy csoport alá rendeljük, melyet egyed típusnak nevezünk. Az egyed típusokat olyan táblázatként is felfoghatjuk, melynek soraiban az egyedek, oszlopaiban pedig az egyedeket leíró tulajdonságok találhatók. Az egyed típusok közötti logikai összefüggéseket kapcsolat típusnak nevezzük. Az adatkör (cluster) egy komplex objektum leírásához szükséges adatmodell entitás- és kapcsolattípusainak logikai úton történő összefoglalása.

Szakkifejezés diagram

A vállalatok többségénél az adat-objektumok definiálására nagy számú kifejezés áll rendelkezésre. Gyakran előfordul azonban, hogy a vállalaton belüli terminológia nem egységes, többféle kifejezést használnak ugyanarra a fogalomra, vagy egy kifejezéssel különböző dolgokat jelölnek. Ez teszi szükségessé egy szakkifejezés modell létrehozását, amelynek célja egyedülálló és konzisztens szakkifejezések definiálása a vállalaton belül, elsősorban a modellezési folyamatot megelőzően, illetve a folyamat közben.

Egy szakkifejezés az adott vállalat kommunikációs környezetének egy kifejezése, ami lehetővé teszi a különböző vállalati csoportok közötti hatékony kommunikációt. A vállalat különböző részeiben létező szinonimák egyesítése az üzleti objektumokkal kapcsolatos kommunikációt sokkal egyszerűbbé teszi.

Tudás struktúra diagram

Ezt a diagrammot az adott vállalati terület működtetéséhez szükséges tudás összegyűjtésére és megjelenítésére használhatjuk. A tudás definiálása nem könnyű feladat, számos magyarázat létezik rá. Esetünkben olyan tanult és használt információs háttérként értelmezhetjük, amely segít értelmet adni az új információknak, és lehetővé teszi a kitűzött feladatok végrehajtását.

Ezekon a modell típusokon kívül az ARIS Toolset-be épített szűrők segítségével más modellezési nyelven írt adatmodellek is integrálhatóak a rendszerbe, mint például az egységes modellezési nyelven (UML-ben) íródott adatmodellek is.

A szervezeti nézet

A szervezeti nézet azoknak a különböző szervezeti egységeknek a statikus kapcsolatait írja le, amelyek a vállalaton belül a tevékenységek végrehajtásáért felelősek. Lényegében a szervezeti struktúra kerül dokumentálásra, amelynek alkalmas eszköze lehet a szervezeti ábra. Ezeknek a modelleknek a segítségével megállapítható, hogy a vállalaton belül ki miért felelős, kiolvashatóak a jelentési kötelezettségek is. A folyamatokkal kapcsolatos esetleges problémák gyorsabban megoldhatóak általuk, hiszen a felmerült probléma helye könnyen azonosítható, a megfelelő szervezeti egységekhez, felhasználói tevékenységekhez köthető.

Szervezeti ábra

A szervezeti ábra a vállalat szervezeti struktúráját rajzolja le, összhangban a szervezeti elemekkel, azok kapcsolataival, ill. Strukturális kritériumaival.

A vállalati célok eléréséhez szükséges feladatok végrehajtói a szervezeti egységek, amelyek a megadott kritériumok alapján vannak kialakítva (például egy szervezeti egységet alkotnak a hasonló vagy kapcsolódó feladatok végrehajtói). A hasonló tulajdonsággal rendelkező szervezeti egységek egy csoportba, a szervezeti egység típusba sorolhatóak. Hasonló tulajdonság lehet például az azonos hatáskör és felelősség. A beosztás a legkisebb szervezeti elem a vállalatban belül.

A személyek a vállalat valós alkalmazottai, akik beosztásokhoz vagy közvetlenül szervezeti egységekhez lehetnek rendelve. A személy típust az azonos tulajdonságokkal (például felhatalmazott, felelős, stb.) leírható személyek csoportja képezi. A külső személy olyan munkatársat jelöl, aki nem tagja a vállalat szervezeti hierarchiájának, de meghatározott időtartamig részt vesz a vállalat működésében. Egy csoport olyan személyek együttesét jelöli, akik együtt dolgoznak egy specifikus feladaton, egy meghatározott időtartam alatt, meghatározott erőforrás kerettel gazdálkodva (például egy projekt csapat).

A funkció nézet

A funkció nézetben a vállalati működés funkciók szerinti tagolása és bemutatása a cél. A funkció az üzleti folyamaton belül egy információs objektumon elvégzett szakmai feladatot illetve műveletet jelent, amely a vállalati célok elérése érdekében történik. Ez a tevékenység jól körülhatárolható, egyértelműen meghatározza, hogy a folyamat adott szakaszában mi a teendő (például a beérkezett számla rögzítése, hitelképesség elbírálása).

A folyamat sok egymás után végrehajtandó tevékenységből épül fel, amely a funkcionézetben statikusan van megjelenítve. A funkció és a folyamat közötti

különbség, hogy egy funkció leírja, mit kell tenni (statikus nézet); egy folyamat leírja, hogyan kell valamit tenni (dinamikus nézet).

Az egyes funkciók az alapján helyezkednek el a modell-hierarchiában, hogy az absztrakció milyen fokát képviselik. Az értékesítési funkció például felbomolhat az ügyfélkapcsolat-kezelés, az ügyfél érdeklődésének feldolgozása, az ügyfél ajánlatadásának feldolgozása, a raktári anyag értékesítése alfunkciókra.

A folyamatleírásban a funkciók mindig egy eseményből indulnak ki (például az útlevegélyigénylés beérkezése), és egy esemény jelzi a végüket (például az útlevegélyigénylés feldolgozva). Ez a szoros kapcsolat teszi lehetővé a funkciók dinamikus nézetbe való átültetését, hiszen az események a funkciók időbeli sorrendjét is meghatározzák.

Funkciófa

A funkciók ábrázolására legtöbbször a funkciófát alkalmazzák, ahol az egyes funkciók hierarchikus tagolásban követik egymást. Az ábrázolás célja a fő tevékenységek közötti statikus relációk áttekintése. A több alfunkcióra felbomló komplex funkciók így grafikusán is megjeleníthetők. A funkcionézet szerinti modellezés ezen a szinten akkor fejeződhet be, ha már az összes alapvető funkciót sikerült feltérképezni.

Céldiagram

A vállalatunk számára definiált és egyben hierarchizált célokat a céldiagrammal jeleníthetjük meg. A célok elérésének mérésére mutatókat illetve paramétereket állapíthatunk meg. Kijelölhetjük a célokhoz tartozó kritikus sikertényezőket. Továbbá minden egyes célnál ábrázolhatjuk, hogy a vállalat mely tevékenységei vagy folyamatai segítik a cél elérését.

Az alkalmazási rendszer (típus) diagram

A modell a vállalatnál használt alkalmazási rendszer típusok (például az ARIS Toolset), modul típusok (például az SAP modulok) és IT funkció típusok hierarchizált struktúráját írja le, tekintettel a különböző operációs rendszerekre és interfészekre. Az adathordozók kiválasztásánál többek között fájl, dokumentáció, fax, telefon és dosszié objektumok közül választhatunk.

A diagram segítségével megállapíthatjuk, hogyan támogatják a vállalati tevékenységeket az informatikai alkalmazások; milyen moduljait használja a vállalat az adott alkalmazási rendszernek; milyen tranzakciók, program modulok szükségesek az adott feladat elvégzéséhez.

Az IT erőforrás nézet

Az IT erőforrás nézet nem egy különálló nézet, mivel csak a többi nézet szempontjából lényeges informatikai eszközöket tartalmazza. Az üzemgazdasági irányultságú komponensek leírásához csupán az információtechnológiai keretfeltételeket szabja meg. Ha összerakjuk az adat, a szervezeti és a funkció nézetben, mindhárom leírási szinten számba vett IT eszközöket, akkor kapjuk meg ezt a kiegészítő nézetet. A szakkoncepció szintjén még csak az alkalmazási rendszer típusát határozzuk meg, míg a technikai megvalósítás szintjén már a konkrét hardver/ szoftver eszközöket. [Scheer, 1993]

Az irányítási nézet

Az irányítási nézet egy dinamikus nézet, amely megteremti a kapcsolatot az egyes statikus nézetek között. Lehetőségünk van a statikus nézetek páronkénti összekapcsolására, illetve megjeleníthetjük mind a három nézetet egy közös modellben.

A funkció- és adat nézet összekapcsolása révén például megállapíthatjuk, mik a tevékenységeink input/output adatai, milyen adatok cserélődnek ki a tevékenységek között, melyik tevékenységnek van szüksége ugyanarra az adatra.

A szervezeti nézet és az adat nézet összekapcsolása abból áll, hogy a szervezeti egységekhez adatokat rendelünk hozzá. Ezáltal könnyen áttekinthetjük, hogy például kinek a felelőssége az adatbiztosítás; melyik információ melyik hálózaton érhető el; ki, milyen felelősséggel melyik adatot érheti el; melyik szervezeti elemnek milyen adatra van szüksége.

A funkció- és szervezeti nézet összekapcsolását például a funkciófában ábrázolt funkciók és a szervezeti ábrában szereplő szervezeti egységek összerendezésével érhetjük el. Ennek eredményeként megtudhatjuk, hogy például mely tevékenység, mely szervezeti elemre hat; mely szervezeti elemek mely tevékenységek végrehajtásáért felelősek; mely szervezeti elemeket kell informálni a tevékenység elvégzésének eredményéről.

Mindhárom nézet összekapcsolásához például a kibővített eseményvezérelt folyamatlanc modellt használhatjuk. Ebben a diagramban az összes többi modellben használt információs objektum, funkció, esemény, szervezeti egység megjelenik, a köztük lévő üzleti központú kapcsolatokkal együtt. A nézet attól lesz dinamikus, hogy a folyamatmodellben szereplő objektumok már egy időbeli és logikus rendet követnek.

Az áttekinthető és teljes adatbázis létrehozása érdekében a statikus objektumokat csak egyszer lehet létrehozni/ letárolni. A dinamikus objektumokra (funkció, esemény, logikai kapcsoló) ez a szabály nem vonatkozik. Az ARIS hivatkozási másolása segítségével újra felhasználhatóak a már létező objektumok, ugyanabban vagy más modellekben is.

Egy hivatkozási másolás a már létező objektum definícióhoz új szimbólumot hoz létre. Ha egy objektum attribútumai módosulnak, az érinteni fogja az objektum összes

hivatkozását, függetlenül attól, hogy a módosítás melyik objektum hivatkozásában, és melyik modellben történt. Ez biztosítja azt, hogy egy objektum csak egyszer legyen definiálva, de annyiszor lehessen újrafelhasználni a modellekben, ahányszor csak szükséges. Végül soron ez teszi konzisztenssé az adatbázist.

Tudástérkép

Az emberi erőforrás, a szervezet és a tudás összekapcsolására használt modell, amivel lehetővé válik, annak ábrázolása, hogy ki, mit tud a vállalatban belül. A szakértők könnyen felismerhetők a konkrét feladatokhoz és kérdésekhez. Elemezhető a tudás- és kompetencia-portfólió, feltárhatóak a hiányok. Így meghatározható a vállalati tudásmenedzsment fejlesztése.

Értékteremtő lánc diagram (value added chain diagram)

Egy vállalat komplex üzleti folyamatait lehetetlen úgy megjeleníteni egyetlen diagramban, hogy azok könnyen átláthatóak legyenek. Ez a komplexitás egy folyamat hierarchia létrehozásával csökkenthető (*top down elv*), amely az absztrakció különböző szintjeit eltérő részletességű modellekkel rajzolja le.

Ennek a hierarchiának a tetején lévő áttekintő folyamatábrából kiderül, hogy mik a vállalat fő tevékenységei. A részletezés egy alacsonyabb szintjén a kulcsfolyamat minden egyes funkciója kifejthető egy másik értékteremtő lánc diagrammal. A funkciók tehát alfunkciókra bomolhatnak. Ez segíti a vállalat átláthatóságát és üzleti folyamatainak könnyebb megértését.

Folyamat kiválasztási mátrix (process selection matrix)

Ahogy az értékteremtő lánc diagram, úgy a folyamat kiválasztási mátrix is áttekintő diagramként használható. Kétdimenziós struktúrájának köszönhetően alkalmas folyamat

variánsok, úgynevezett folyamat szcenáriók (forgatókönyvek) ábrázolására úgy, hogy a folyamat fő lépéseit hozzárendeljük az egyes szcenáriókhoz.

Bizonyos vállalati folyamatok azonos lefutásúak, de különböző döntési helyzeteket vizsgálva felfedezhetjük, hogy a folyamat lefutásokban eltérések jönnek létre. Például egy vállalat életében a beszerzés folyamata általában azonos: igény keletkezik, megvizsgálják jogos-e az igény, megvizsgálják belső forrásból kielégíthető-e, ha igen kielégítik, ha nem külső forrást keresnek, kiválasztják a megfelelőt, a megrendelést engedélyeztetik az illetékesekkel, majd megrendelik az igény kielégítésére alkalmas terméket-szolgáltatást.

Ebben a szokásos folyamatmenetben azonban számos eltérés következhet be az egyes döntési pontoknál (például az engedélyeztetésnél, kiválasztásnál). Ezeket a különbségeket megjeleníthetjük egy modellen belül is elágazásokkal, vagy akár több rész-folyamatmodellben is.

Kibővített eseményvezérelt folyamatlánc diagram- eEPC (extended Event driven Process Chain)

Ez az egyik leggyakrabban használt modell típus a folyamatok ábrázolására, mert az összes többi nézet objektumait képes egy modellben megjeleníteni. Egyszerre ábrázolhatóak az események, tevékenységek, szervezeti objektumok, adat objektumok, alkalmazási rendszereket leíró objektumok, valamint a folyamat időbeli és tartalmi lefutását meghatározó logikai elágazások leírására használt logikai kapcsolók (és/ vagy/ kizárólagos vagy operátorok).

A „karcsú” EPC-ben a folyamatok funkciók szerint dinamikusan vannak megjelenítve, tehát a folyamatot leíró események és funkciók időrendi és logikai sorrendbe vannak rendezve. Az események előidézhetnek funkciókat és lehetnek funkciók eredményei is. A folyamatlánc az események és a funkciók egymás utáni váltakozásával áll elő.

A kibővített eseményvezérelt folyamatlánc tovább bővül a folyamatban érintett szervezeti egységekkel és a kapcsolódó információ- és adatfeldolgozó rendszerekkel. Így nemcsak arra kapunk választ a folyamatláncot tanulmányozva, hogy mit kell tenni a kitűzött célok elérése érdekében, hanem arra is, hogy mikor és hogyan, milyen adatok és IT-eszközök segítségével tegyük azt.

II.6.4. Az ARIS leíró szintjei

[Scheer, 1993]

Az integrált információrendszerek leíró szintjeinek kialakítása a jelenlegi rendszerek vizsgálatával kezdődik. Elemezni kell a meglévő rendszerek erősségeit, gyengeségeit, lehetőségeit és veszélyeit. Ezután a vállalati vezetés definiálja a jövőbeli fejlesztési irányokat és elérendő célokat. Ezek a célkitűzések az informatika számára közvetlenül még nem értelmezhetőek, ezért ezt a szintet nevezhetjük a problémafelvetés szintjének, amely még nem egy valódi leírási szint.

A problémafelvetés során megfogalmazott célokat és fejlesztési irányokat, a létrehozandó rendszerrel kapcsolatos követelményeket, le kell fordítani az informatika nyelvére. Az elvárások szakmai nyelvre történő átültetése révén alakítható ki az információrendszerek legmagasabb leíró szintje, a szakmai koncepció szint. Itt történik a vállalati információrendszer architektúrájában lévő főbb alkotóelemek meghatározása

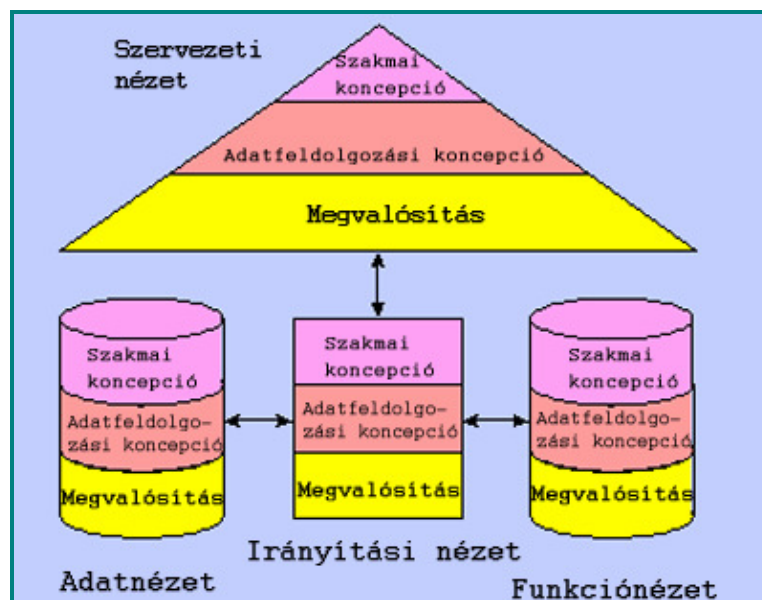
A következő leíró szinten a megfogalmazott célokhoz és tevékenységekhez már hozzá kell rendelni a megvalósításhoz szükséges informatikai háttérrel is. Ki kell jelölni a funkciók végrehajtásához szükséges tranzakciókat, alkalmazási rendszer típusokat és szoftvermodulokat. Ez a szint az adatfeldolgozási koncepció szintje vagy más néven az IT specifikáció szintje.

A harmadik és egyben legalacsonyabb leíró szint, a technikai megvalósítás szintje, ahol a felhasználók tevékenységeihez részletesen hozzá kell rendelnünk a végrehajtás eszközeit. Meg kell határozni a konkrét hardver és szoftver eszközöket, amelyeket az új rendszer bevezetése során a hatékony munkavégzés érdekében elérhetővé kell tenni.

II.6.5. Az ARIS architektúra

Az ARIS architektúra koncepciója szerint a vállalati működést minden nézetben és minden leíró szinten meg kell vizsgálni, mert csak így lehet elérni az üzleti folyamatok teljes körű elemzését. Az ARIS koncepció szerint az egyes nézetekben egyre részletesebb modelleket alkotva juthatunk el a teljes vállalati működést leíró modellekig.

Az áttekinthetőség megőrzése végett az absztrakció legmagasabb fokán álló modellek objektumai mögé újabb modellek helyezhetőek, amelyek objektumai mögé ismét újabbak és így tovább egészen addig, amíg „leásunk” a technikai megvalósítás szintjéig. Az egyes objektumok mellett megjelenő apró szimbólumok jelzik számunkra, hogy az adott objektum mögött található egy kapcsolódó, részletesebb modell (lásd 2.3 ábra!) Így az egyes objektumoknak az adott folyamatban betöltött szerepét könnyebben átláthatjuk és megérthetjük.



21. ábra Az ARIS architektúra – nézetek és leíró szintek

[Szendró, 2003]

A modellekben az egyes leíró szintek eltérő időtávra szólnak, és a tervezés különböző szakaszaiban válnak szükségessé. A legmagasabb szintet képviselő szakmai koncepció által definiált irányoknak van a legnagyobb hatása a vállalat jövőjére, mert ezek a célok szólnak a leghosszabb távra. Ezzel szemben a technikai megvalósítás szintjén hozott döntések viszonylag rövid távúak: a különféle hardver és szoftver elemek gyorsan elavulttá válhatnak, így szükséges lehet új eszközök beszerzése a tevékenységek támogatásához. [Szendrő, 2003]

II.6.6. Az üzleti folyamatok modellezésétől az üzleti folyamatok menedzsmentjéig

[Scheer –Abolhassan – Jost – Kirchmer, 2002]

Minden szervezet, függetlenül annak méretétől és tevékenységi területétől, leírható az üzleti folyamataival. Azok a vállalatok, amelyek nem képesek felismerni és körülhatárolni az alapvető üzleti folyamataikat, nem életképesek hosszú távon. Az üzleti folyamatok definiálása nélkül ugyanis elveszítik annak lehetőségét, hogy aktívan támogassák hatékonyságukat és hatásosságukat.

A modellezés legfőbb feladata, hogy az üzleti folyamatoknak egy jól átlátható, határozott struktúrát alakítson ki. Definiálja a stratégiai célokat, tevékenységeket, a kapcsolódó felelősségeket és a végrehajtás eszközeit. Meghatározza, hogy *ki, hogyan és mit* csinál. Ez a rögzített struktúra jelentheti később az alapot az üzleti folyamatok költség, idő és minőség szempontú kiértékeléséhez. A modellezés azonban a folyamatok azonos idejű irányítását és folyamatos monitoringját nem teszi lehetővé.

A teljes körű üzleti folyamatmenedzsment nemcsak az üzleti folyamatok tervezését és szemléletes leírását jelenti, hanem magában foglalja a folyamatok irányítását, ellenőrzését és állandó fejlesztését (Continuous Process Improvement-CPI) is. A vállalatoknak folyamatosan össze kell hasonlítaniuk kitűzött céljaikat a folyamatok végrehajtása során elért eredményekkel. Az eredmények értékelésére visszajelző

mechanizmust kell működtetniük, amelyet aztán felhasználhatnak a tevékenységeik optimalizálásához.

Az üzleti folyamatmenedzsment során az úgynevezett életciklus szemléletet alkalmazzák. A ciklus a folyamatok tervezésével kezdődik, majd a folyamatok implementálása után a teljesítmények mérése következik, végül az eredményeket kiértékelve az optimalizálással zárul. Napjaink gyorsan változó világában azonban nem elég csupán egyszer megtervezni a folyamatokat, hanem állandó folyamatfejlesztésre és elemzésre van szükség!

A vállalatok menedzsmentje az üzleti helyzetekről általában múltbeli adatok elemzése révén tájékozódik, mint például az eladási volumen adatok, a cashflow kimutatások, az elért profit nagysága, stb. Alapján. Sok eseményt, amik azonnali észrevételt és beavatkozást igényelnének, túl későn észlelnek, ha egyáltalán észlelnek a havi vagy heti jelentések alapján. Így sokszor már kifutnak a cselekvési időből.

Az új mérési módszereknek megbízható alapot kell nyújtaniuk a folyamatok hatékonyságának közel azonos idejű kiértékeléséhez. A hagyományos Üzleti Intelligencia (BI) eszközök ebből a szempontból értékes információforrást jelentenek, ám mégsem tudják tökéletesen ellátni a feladatot, mert elemzéseik főként működési adatokon alapulnak, amelyek nem korrelálnak teljesen az üzleti folyamatokkal.

A fogyasztói magatartás elemzése a jelenlegi területi eladási adatokból, vagy a keresztértékesítési lehetőségek kiszámítása nem igényel semmilyen kapcsolódást az üzleti folyamatokkal, de vannak olyan kérdések is, amelyeket nem lehet megválaszolni csak dokumentumalapú vagy tranzakció-orientált alkalmazások révén. Ilyen kérdések például, hogy mennyire költség-és idő-hatékonyak a beszerzési és értékesítési csatornák; hol vannak a gyenge pontok és a szűk keresztmetszetek a folyamatokban; vannak-e késve kiegyenlített vevőrendelések; mennyire voltak sikeresek a véghezvitt fejlesztési akciók. A folyamatok teljesítményméréséhez Kulcs Teljesítmény Jelzőket (Key Process Indicator –KPI) kell meghatároznunk az érintett alkalmazásokból vett adatokra (dokumentumok, log fájlok, vevő adatok, stb.) építve. Ezeket a folyamat-orientációjú

teljesítménymérési eszközöket Üzleti Folyamat Intelligencia (BPI) eszközöknek nevezzük.

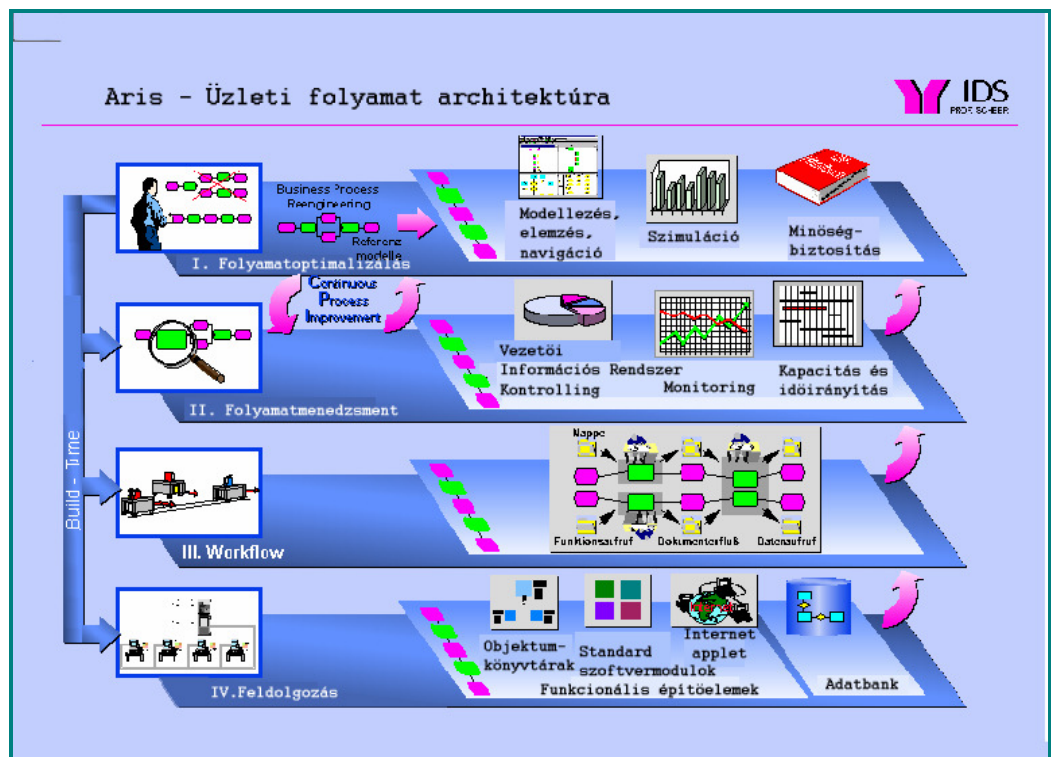
A BI és a BPI eszközök használatával persze még nem lesznek automatikusan optimalizálva a folyamataink. Ez a feladat még a menedzsmentre vár, de a lehetőséget megteremtettük számára az üzleti folyamatok életciklus szemléletének alkalmazására. Folyamatosan ellenőrizhetik stratégiai intézkedéseik következményeit, és azonnal megtehetik a szükséges módosításokat.

Az üzleti folyamatok fejlesztése csak akkor lehet sikeres, ha a folyamat résztvevői tényleg „részt vesznek”, tehát képesek alkalmazkodni a megváltozott körülményekhez. A folyamatok változása mindig együtt kell, hogy járjon az emberek magatartás-változásával is.

II.6.7. ARIS HOBE – Az üzleti tervezés háza

[Scheer –Abolhassan – Jost – Kirchmer, 2002]

A teljes körű üzleti folyamatmenedzsment támogatására az IDS Scheer vállalat 1995-ben megalkotta az üzleti tervezés házát (House of Business Engineering). Ez a négylépcsős modell a komplett életciklus szemlélet jegyében lett kialakítva, amely alkalmazásával megvalósulhat a folyamatok átfogó támogatása, elemzése és fejlesztése. Ehhez sokoldalú segítséget nyújt az ARIS koncepció és a hozzá kapcsolódó eszköztár (ARIS Toolset, ARIS Workflow, ARIS Szimuláció stb.).



22. ábra ARIS Üzleti tervezés háza [Szendrő, 2003]

Folyamattervezés

A ház legfelső szintjén történik a vállalat üzleti folyamatainak tervezése és leírása. A modelleket a vállalat jelenlegi adat-, funkció-, szervezeti- és IT struktúrájával szinkronban kell kialakítani, szem előtt tartva a jövőbeli célokat és fejlesztési irányokat is. Az üzleti folyamatok pontos definiálása és minden igényt kielégítő modellezése igen komplex feladat, amely komoly szakértelmet igényel. Hosszú távon az itt kialakított vállalati modellekre építkeznek a vállalatok.

A folyamatmodellek kialakítása előtt szükséges lehet egy BPR projekt kivitelezése, amely során feltárják, elemzik és újratervezik a meglévő folyamatokat. A projektcsoport tagjai hatékonysági szempontok szem előtt tartásával racionalizálják a folyamatokat. Összevonnak, kifejtenek, megszüntetnek, vagy átalakítanak bizonyos tevékenységi területeket.

A modellezés során kiindulhatunk kész referenciamodellekből is, amelyek bizonyos iparágak vállalatainak legjobb gyakorlata (best practice) szerint lettek kialakítva tanácsadók által. A referenciamodellekben tükröződő iparági szaktudás mellé kell tenni a vállalati tudást is, és a kettő együttes felhasználásával kell létrehozni a testre szabott, egyedi vállalati modelleket. Ezzel a módszerrel rengeteg időt és költséget takaríthatunk meg, hiszen egy jól működő, kész megoldást kell csak adaptálnunk a vállalati specialitásainkhoz.

Az ARIS Szimuláció komponens segítségével lehetővé válik a folyamatok dinamikus lefutásának vizsgálata. A folyamatmodellek (pl. eEPC) logikai elágazásainál bekövetkezési valószínűségeket rendelhetünk az egyes ágakhoz, továbbá különböző paramétereket adhatunk meg az egyes tevékenységeknek is (pl. Idő-, költség-, kapacitás-kihasználási szint). Ezáltal kiértékelhetjük az egyes folyamat-alternatívák teljesítményeit, és kiválaszthatjuk a számunkra optimális megoldást. Feltárhatjuk a folyamatok kritikus pontjait, szűk keresztmetszeteit, kijelölhetjük a fejlesztési irányokat látván, hogy hol lehet párhuzamosítani, összevonni illetve automatizálni a folyamatokat.

A helyes vállalati folyamatok kialakítására minőségbiztosítási projektbe is kezdhetünk. Az ISO 9000-es minősítés megszerzése garantálja a jól működő, standardizált folyamatokat, amelyek a későbbi fejlesztések alapjául szolgálhatnak.

Folyamatmenedzsment

Az üzleti tervezés házának második szintjén történik a már működő folyamatok adatainak folyamatos figyelése és elemzése. A korábban kialakított modellek alapján az egyes tevékenységekért felelős szervezeti egységek pontosan meghatározottak. Számukra biztosítani kell az őket érintő adatokhoz való teljes körű hozzáférést, hogy nyomon tudják követni a folyamatok menetét. A folyamatok állandó monitoringja révén a felelősök mindig friss információval rendelkeznek az aktuális állapotokról. Így ha szükséges, időben be tudnak avatkozni a folyamatokba, a megfelelő helyeken.

Az üzleti folyamatok menedzsmentje számára különféle kapacitás- és időirányítási módszerek állnak rendelkezésre. Ezek segítségével tervezhetik és ellenőrizhetik a folyamatok hatékonyságát. Feltárhatják a nem jól működő területeket és változtathatnak rajtuk az optimális erőforrás kihasználás igényének szem előtt tartásával.

A költségekről, átfutási időkről, bevételekről, stb. Aggregált információkat kaphat a menedzsment a Vezetői Információs Rendszerek használatával. Ezeknek az adatoknak az elemzése révén elsősorban a stratégiai jelentőségű kérdések megválaszolása, és a kulcsfontosságú döntések meghozatala válik lehetővé.

A kitűzött célokat folyamatosan össze kell hasonlítani a folyamatok működése során elért eredményekkel. Az ARIS háznak ez a szintje teszi lehetővé a folyamatok azonos idejű irányítását, tehát azt, hogy ha eltérés tapasztalható a kívánt menettől, rögtön be lehessen avatkozni a szükséges helyen. A ház első szintjével szorosan együttműködve nyílik lehetőség a folyamatok állandó fejlesztésére (CPI), amely a gyorsan változó piaci környezetben minden vállalat számára alapvető feladat.

Workflow

Az ARIS ház harmadik szintjén a folyamatok végrehajtásának támogatása zajlik egy workflow rendszer működtetésével. Ez a rendszer a folyamatok irányítását oldja meg úgy, hogy az egyes funkciók végrehajtásához szükséges alkalmazási rendszereket képes egy rendszerben összefogni. A folyamatok feldolgozása során a workflow rendszer hívja elő megfelelő sorrendben a feladatok ellátásához szükséges alkalmazásokat.

A feldolgozandó objektumok mozgatásáért is a workflow rendszer felel. A felhasználók között virtuális irattárolókban szállítja a feldolgozandó ügyeket és a kapcsolódó dokumentumokat. Ha egy felhasználó elvégezte az adott ügyiraton a szükséges módosításokat, akkor kimenő kosarából átkerül az ügyirat a soron következő felhasználó bemenő kosarába. Így a dokumentumok útja és feldolgozottsági szintjének alakulása pontosan követhetővé válik. Ebből kifolyólag könnyebb azonosítani az esetleges

problémák helyét, és gyorsan be lehet avatkozni ott, ahol elakadt a folyamat végrehajtása.[Szendró, 2003]

Ezen a szinten, mint láthatjuk, a korábban meghatározott folyamatmodellek még részletesebb kifejtése válik szükségessé: az általános szervezeti egységek helyett itt már konkrét felhasználókat kell az egyes tevékenységekhez rendelni; általános folyamatleírások helyett a tényleges folyamatokat kell ábrázolni (pl. egy konkrét megrendelés beérkezése). Ezáltal válik lehetővé a workflow rendszer számára a munka irányítása és felügyelete.

A workflow rendszer elsősorban a jól strukturált, gyakran ismétlődő, operatív feladatok támogatására alkalmas. Komplex, egyedi feladatok végrehajtása esetében hasznos lehet groupware rendszerek alkalmazása, amelyek ugyan a folyamatok irányítását nem segítik, de különféle eszközökkel támogatják a hatékony csoportmunkát (pl. e-mail, naptár), és ezáltal a folyamatok eredményes feldolgozását is.

Folyamat-feldolgozás

Az ARIS ház negyedik szintjén történik a folyamatok konkrét végrehajtása, amelynek menete a workflow rendszer által irányított. Az egyes tevékenységek elvégzése, a szükséges dokumentumok feldolgozása különféle alkalmazási rendszerek segítségével történik. Ezek lehetnek például irodai szoftverek (szövegszerkesztők, táblázatkezelők), a vállalatirányítási rendszer tranzakciói, komplett szoftvermodulok vagy akár internetes alkalmazások is. Az ARIS modellekhez majdnem mindegyik alkalmazási rendszer gyártónak a termékét hozzá lehet illeszteni. Ez a szint ezeket az információtechnológiai megoldásokat helyezi a középpontba.

Az ARIS HOBE koncepciójában tehát a workflow és az alkalmazási rendszerek külön szinten szerepelnek, ami azt jelenti, hogy a folyamatok irányítása és konkrét végrehajtása elválnak egymástól. Ez azért nagyon kedvező megoldás, mert így a

végrehajtás folyamata nem az alkalmazási rendszer programkódjában van rögzítve, hanem egy külön modellben. Ezáltal a folyamatok átalakítása egyszerűen és rugalmasan kivitelezhetővé válik, mivel nem igényel komolyabb programozói beavatkozásokat. [Szendró, 2003]

Az üzleti tervezés házának négylépcsős koncepcióját sok integrált vállalatirányítási rendszer gyártó majdnem változatlanul átvette. A rendszerek architektúráját leíró térképként használják termékeik kialakításakor. Ez is bizonyítja a koncepció megalapozottságát és életképességét.

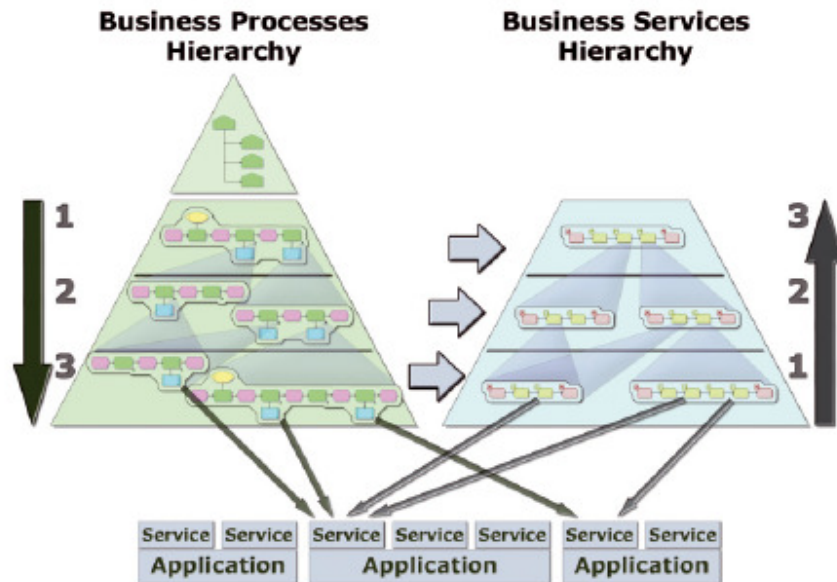
II.6.8. Az ARIS for SOA

A szolgáltatásorientált megközelítés új – jelentősen rugalmasabb – lehetőségeket kínál az üzleti folyamatok IT rendszerekbe történő adaptálására, gördülékeny beépítésére és működtetésére. Ezen a téren az üzleti folyamatmenedzsment (BPM) és a SOA kiegészítik egymást: az átláthatóan kialakított vállalati architektúra lehetővé teszi az üzleti folyamatoknak a vállalat informatikai rendszerébe történő rugalmas és pontos adaptálását. Összehasonlítva a „dobozos” szoftver megoldások által nyújtott funkcionalitásra épülő standard folyamatokkal, a folyamatok nagyobb rugalmassággal tervezhetők, és jobban illeszkednek a vállalati stratégiához.

Az ARIS SOA Designer segíti a vállalatokat abban, hogy üzleti folyamataikat működő technikai folyamatokká alakíthassák. Az üzleti folyamatstruktúrákból nyert technikai folyamatok pedig gyors és rugalmas alapot kínálnak az üzleti stratégia és az ezt támogató folyamatok informatikai rendszerekbe történő leképezéséhez, működőképes alkalmazás kialakításához.

Az üzleti folyamatoktól a SOA-ig vezető út első lépése az üzleti folyamatok és a támogató IT környezet rögzítése. Az ARIS SOA Designer alkalmazásával ellenőrizhető, hogy a vállalaton belül az üzleti folyamatok támogatásához rendelkezésre áll-e a megfelelő technikai szolgáltatási háttér. Az üzleti folyamat lépéseinek és a szolgáltatások leírásainak automatikus leképezése leegyszerűsíti a szükséges

szolgáltatások meghatározását. Az üzleti folyamatban található tevékenységeknek és a hozzájuk tartozó szolgáltatásoknak az összekapcsolása eredményként egy továbbfejlesztett, technikai folyamatstruktúrát hoz létre.



23. ábra Üzleti folyamatok specifikálása – szolgáltatások meghatározása [Jörg Klückmann, 2006]

Az üzleti folyamatok specifikálása felülről lefelé irányuló folyamatfinomítással történik. Már ezzel egy időben elindulhat az alulról felfelé történő szolgáltatás-meghatározás vállalati szolgáltatástárak és a háttérrendszerek képességeinek figyelembevételével. [Jörg Klückmann, 2006]

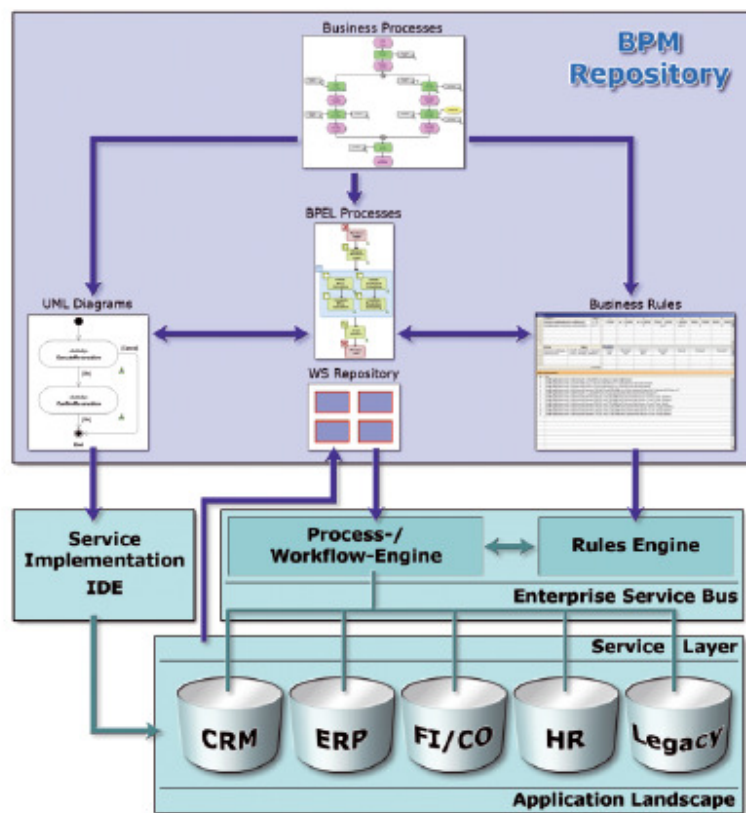
Az ARIS SOA Designer ennek a kibővített üzleti folyamatstruktúrának a platformfüggetlen BPEL-folyamatokká történő automatikus átalakítását valósítja meg. Nemcsak a folyamatok lefutása, hanem a megjelenő szolgáltatási és adatinformációk is átvitelre kerülnek.

A szolgáltatások gyorsabb modellezéséhez a SOA Designer grafikus BPEL modellezési felületet biztosít.

Ezt követően a BPEL-folyamatot XML és WSDL állományokká alakítjuk, amelyeket aztán a további implementáláshoz, működtetéshez különböző informatikai rendszereknek adhatunk át.

A SOA üzleti és technikai rétegeinek egy központi ARIS elemtárban történő összevezetése nyomán láthatóvá és irányíthatóvá válnak a rétegek közötti kölcsönös összefüggések.

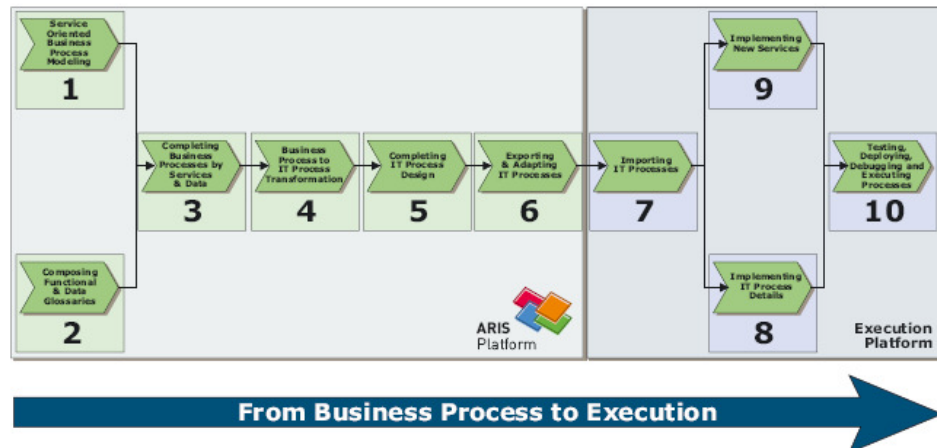
Egy kattintással megtekinthető, hogy adott szolgáltatás melyik folyamatban használatos. Ennek segítségével a szolgáltatás kiesése esetén könnyen megtalálható, hogy melyek az érintett folyamatok, és kiket kell értesíteni a vállalaton belül. Az eredmény egy teljes körű szemantikai szolgáltatás-elemtár az ARIS-ban.



24. ábra Integrált repository az ARIS-ban [Jörg Klückmann, 2006]

Az üzleti folyamatok BPEL állományokká történő automatikus átalakítása pedig nagyban leegyszerűsíti és felgyorsítja az üzleti elvárások és a technikai végrehajtás összehangolását.

Jörg Klückmann az ARIS Expert Paper-ben ismerteti azt a 10 lépéses eljárást, amely módszertani útmutatóul szolgálhat a folyamatfelméréstől a SOA implementációig való eljutásra. [Jörg Klückmann, 2007]



25. ábra A 10 lépéses eljárás [Jörg Klückmann, 2007]

1. Service orientált üzleti folyamatok modellezése

Az első lépést üzleti elemző szakember végzi.

A fázis fő tevékenységei:

- az üzleti folyamatok intuitív modellezése az ARIS szoftverrel
- az üzleti folyamatok szolgáltatásorientált részletezése,
- a folyamatmodellek külső forrásból importált, vagy saját speciális szolgáltatás-leírásokkal történő kibővítése.

2. Szolgáltatások és adatjegyzékek összeállítása

A második lépést IT architektúra szakember és folyamatfejlesztő szakember közösen végzi.

A fázis fő tevékenységei:

- Tevékenység jegyzék összeállítása
- Adatjegyzék összeállítása
- Service-k és a végrehajtó platform adatobjektumainak összehangolása

3. A service-k összekapcsolása folyamatokkal és adatokkal

A harmadik lépést folyamatfejlesztő szakember végzi.

A fázis fő tevékenységei:

- A modellezett folyamatok végrehajtható service-eket tartalmaznak
- Az üzleti folyamatok és az IT eljárások közötti tényleges kapcsolat meghatározása

4. Üzleti folyamatok IT folyamatokká konvertálása

A negyedik lépést folyamatfejlesztő szakember végzi.

A fázis fő tevékenységei:

- Futtatható BPEL modell generálása a folyamat struktúra megtartásával
- A modellezett service-k helyes meghívása
- Az adatfolyam adatainak átvitele
- Interface description létrehozása a business service-hez (WSDL)
- Az ismétlődő transzformációk egybeolvasztása
- Modellek összekapcsolása

5. IT folyamatok készítése

Az ötödik lépést üzleti elemző és folyamatfejlesztő szakember együttesen végzi.

A fázis fő tevékenységei:

- Az automatikusan létrehozott folyamatok végrehajtó platform specifikus átalakítása.

6. IT folyamatok exportálása és beillesztése

A hatodik lépést folyamatfejlesztő szakember végzi.

A fázis fő tevékenységei:

- Végrehajtható kód generálása (BPEL, WSDL)
Model - >programkód transzformáció

7. IT folyamatok importálása

A hetedik lépést szoftverfejlesztő szakember végzi.

A fázis fő tevékenységei:

- Transzfer fájlok generálása ARIS platformon a végrehajtó platform számára

8. IT folyamatok részletes implementálása

A nyolcadik lépést szoftverfejlesztő és integrációfejlesztő szakember közösen végzi.

A fázis fő tevékenységei:

- Az olyan platform részeket, amiket nem lehet ARIS-ban létrehozni, hozzá kell adni a végrehajtó platformhoz

9. Új service-k implementálása

A kilencedik lépést szoftverfejlesztő szakember végzi.

A fázis fő tevékenységei:

- Az új szolgáltatások létrehozása és implementálása a végrehajtó platformon

10. Tesztelés, telepítés, hibajavítás, végrehajtás

A tizedik lépést integrációfejlesztő szakember végzi.

A fázis fő tevékenységei:

- Tesztelés, telepítés, hibajavítás, végrehajtás a végrehajtó platformon

Az üzleti elemző szakember fő tevékenységei:

- A követelmények meghatározása
- A mostani és a célfolyamatok modellezése
- A folyamatváltoztatások engedélyezése

Az IT architektúra szakember fő tevékenységei:

- A rendszer, a service-k és az IT infrastruktúra menedzselése
- Architectura standardok és IT fejlesztési tervek meghatározása

A folyamatfejlesztő szakember fő tevékenységei:

- Üzleti folyamatok technikai folyamatokká történő átalakítása
- folyamatos szinkronizálás

A szoftverfejlesztő szakember fő tevékenységei:

- Új service-k fejlesztése
- Létező rendszerek elkülönítése

Az integrációfejlesztő szakember fő tevékenységei:

- Az IT vagyon összegyűjtése
- A létező és új alkalmazások logikus integrációja

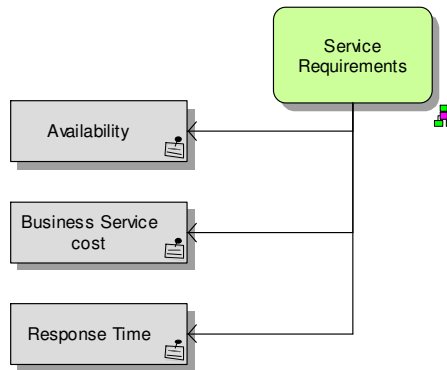
[Jörg Klückmann, 2007]

Az ARIS for SOA kiegészítésben létrehozhatjuk a szolgáltatások magas absztrakciós szintű platform független modelleket. Megadhatjuk a funkcionális és nem funkcionális képességeket, követelményeket és makroadatokat, amik a működéshez szükségesek

Az absztrakt szolgáltatások implementálhatóak szervezeti egységekkel, szoftver service-vel, vagy folyamatokkal, majd újrafelhasználhatóak, azaz a tartományokon belül kombinálhatóak.

Koncepcionális szint

Ezen a szinten történik a követelmények specifikálása, azaz a szolgáltatások üzleti követelményeinek leírása (SLA). A feladatot üzleti elemző végzi.

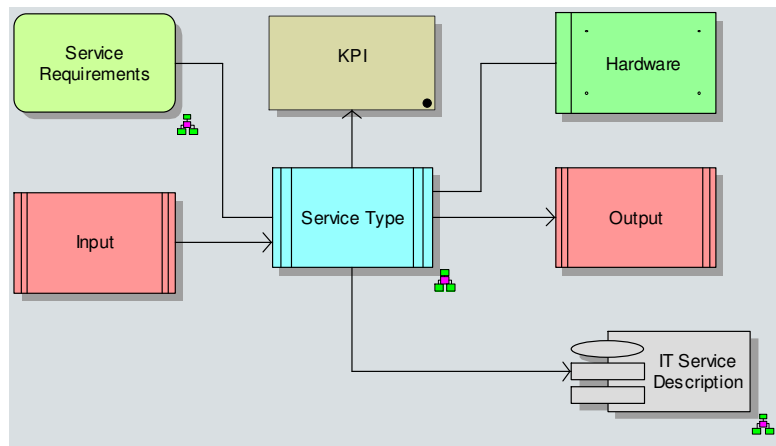


26. ábra IS contextus modell

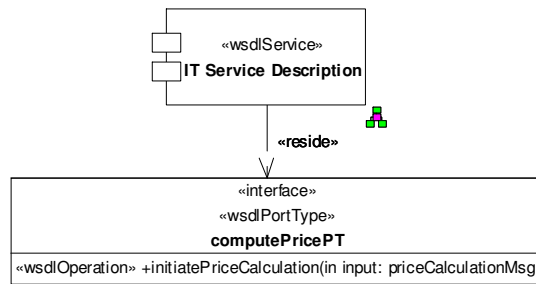
Logikai szint

Szoftverszolgáltatások logikailag leírhatóak az általuk nyújtott szolgáltatásokkal, logikai adatokkal, felhasználókkal, és a képességeikkel az IT tervezés szintjén. A feladatot IT szakember végzi.

A szolgáltatás- funkciókat, metaadatokat, és interfészeket platform független technikai szinten (UML implementation and WSDL, IDL, REST, XSD, etc.) adjuk meg.



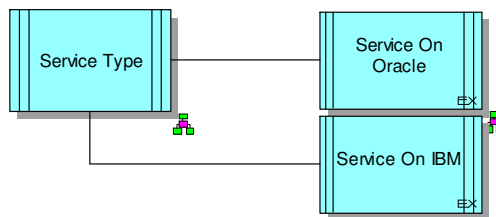
27. ábra Access diagram



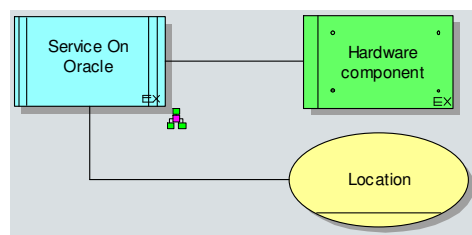
28. ábra UML komponens diagram

Végrehajtás szint

Ezen a szinten írjuk le a szolgáltatás IT implementációjának részleteit. A feladatot IT architektúra szakember végzi.



29. ábra Alkalmazás rendszer diagram



30. ábra Access diagram (IT)

Összefoglalva tehát az ARIS for SOA megoldás közvetlen összeköttetést teremt az üzleti folyamat leírások és az azokat folyamatmotorral végrehajtható (Process Execution Engine) alkalmazások (SAP, IBM, Microsoft, Oracle, BEA...) között. Az ARIS for SOA egy közös ARIS adatbázisban rögzíti a vállalat üzleti folyamatait (EPC-k), az IT

specifikus folyamatleírásokat (BPEL modellek), illetve a szoftverek adat kapcsolatait (UML modellek).

A bemutatott módszertan és eszközök alkalmazásával lehetővé válik:

- a folyamatok intuitív dokumentálása és módosítása,
- a platformfüggetlen, végrehajtható BPEL-folyamatok üzleti folyamatokból történő automatikus előállítás,
- az ARIS Process Performance Manager opcionális integrációján keresztül történő állandó folyamatmonitoring,
- a SOA összes üzleti és technikai elemére vonatkozóan egy minden részletre kiterjedő SOA-elemtár létrehozása,
- a változó piaci feltételekre történő válaszadás felgyorsítása.

A SOA módszertannal elkészített adatbázist (folyamat és adat modelleket) nem csak egyszer, a szolgáltatás orientált alkalmazásokba történő közvetlen betöltésekor lehet felhasználni, hanem a modellek karbantartásával folyamatos felsővezetői kontrollt (folyamatok) és azonnali fejlesztési lehetőséget (BPEL és UML modellek) tud biztosítani.

III. A KUTATÁS GYAKORLATI MEGVALÓSÍTÁSA

A II.3 fejezetben részletesen bemutatott üzleti integrációs modell alapján a hatékony integráció során minden szint minden elemére figyelmet kell fordítani – horizontálisan és vertikálisan is. A rétegek közötti integrációt is meg kell valósítani. A model alapján az is látható, hogy az üzleti folyamatok alapvető, létfontosságú kapcsolatot jelentenek a technikai és szervezeti infrastruktúra között is.

A 90'-es években a vállalatok integrációs problémáinak teljeskörű megoldásának tartott ERP rendszerek a gazdag funkcionalitás és rugalmasság ellenére sem mindig képesek a

vállalatok egyedi, teljesen specifikus igényeit kielégíteni. Előfordulhat, hogy egy üzleti folyamatnak a vállalat és a szoftver által megvalósított módja között áthidalhatatlan eltérések vannak, esetleg a rendszer egyáltalán nem támogat bizonyos tranzakciókat.

A szoftver és a vállalat között jelentősebb illeszkedési hiba forrása lehet, ha a rendszer nem támogatja a szervezet növekedési stratégiáját, struktúráját, kultúráját, döntéshozatali módját. A gyorsan növekvő, turbulens környezetben működő, ezért szervezeti stratégiát gyakran átalakító agilitásra törő vállalat által általában nem alkalmazható.

Azok a problémák, amelyekre az integrált vállalatirányítási rendszerek megoldást kínálnak, más módon ma már hatékonyabban oldhatóak meg.

[Markus, Tanis, Fenema, 2000]

A kulcsfolyamatok kialakításának és az erre alapozott működés megteremtésének képessége ma már minden szervezetben alapkövetelmény. A folyamatelvű szervezet működése során gyors és hatékony üzleti folyamatokba építi specializált funkcionális szakértelmét. Az információs technológia korszerű alkalmazásával a vevői/szállítói kapcsolatokra figyelve integrálja a beszerzést, az előállítás és az elosztást, s mindezt úgy, hogy a vállalat működését alapvetően a vevői megrendelések mozgatják. Megtanul egyedi, a fogyasztók egyéni igényeinek megfelelő termékeket és szolgáltatásokat nyújtani változatos fogyasztói szegmenseik számára a szokásos széles választékkal és az alacsony termelési mennyiséggel járó felár elkerülésével. A globalizáció vállalatainak az új termékek és szolgáltatások kifejlesztéséhez szükséges nagy beruházások következtében gyakran világméretű piacra van szükségük a megfelelő megtérülés eléréséhez így egyesíteniük kell a globális működés nyújtotta hatékonysági és versenyelőnyöket a helyi fogyasztói igényekre való érzékenységgel. A termék-életciklusok rövidülésével a termék-életgörbe egy szakaszában elért versenyelőny nem garantálja ennek megtartását a technológia következő fejlettségi szintjén is. A fogyasztók jövőbeli igényeinek előrejelzése, az új gyártási technológiák gyors alkalmazásának képessége, a folyamatok, termékek és szolgáltatások állandó fejlesztése – az innováció kritikus a vállalati siker szempontjából.

Központi kutatási kérdésem volt, hogy mely módszertan lesz a legalkalmasabb az ERP-t alkalmazó szervezetek integrációs problémáinak a megoldásához, milyen fontos jellemzőkkel rendelkezik, hogyan alkalmazható az adott területeken és hogyan fejleszhető tovább.

A kiválasztott modellezési eljárásokat a gyakorlatban alkalmazom. Alapvető kérdés, hogy melyik az az absztrakciós szint, amelynek segítségével az eltérő területek közös vonásai megragadhatóak. A modell alapú megközelítés adhat támogatást olyan magas szintű leírásra (metamodellek készítésére), amelyek lehetővé teszik az egyes területek szemantikus integrálását. Így az általam választott folyamat és modellvezérelt megközelítés gyakorlati megvalósítása prototípusként egy gyakorlati igazolását jelenti ezen irányú alkalmazhatóságának.

Az ARIS hatékony koncepcionális keretrendszert biztosít, bizonyos hiányosságai azonban korlátozzák az ARIS alapú BPM megfelelő szintű automatizálhatóságát. Ezért javaslatot teszek az ARIS modellek SBPM trendeknek megfelelő felhasználására. Ez az irány megfelel a jelenlegi SBPM kutatási tendenciáknak, amelyek a meglévő eszközök és sztenderdek kompatibilis együttműködtethetősége irányába történnek. Ez azt jelenti, hogy egy BPEL, vagy egy EPC modell felhasználható legyen egy SBPM környezetben, ugyanakkor egy SBPM rendszerben létrehozott folyamat legyen exportálható BPEL-be, hogy a standard BPEL motorok végre tudják hajtani.

Az ARIS modellek alapul szolgálhatnak több részletesebb modellnek, amelyek jelentős hatásúak az automatizálás és interoperabilitás fokára, mivel közös halmazokat definiálnak a különböző forrásból származó adatok számára. Továbbá az elkészült üzleti folyamatmodellek segítik a vállalat dolgozóit a folyamatok jobb megértésében és ezzel a pontosabb és gyorsabb végrehajtásban. Ezen felül a folyamatok elemzése során felfedezhetőek a gyenge pontok (például a leginkább erőforrásigényes tevékenységek), amelyek azután megszüntethetők vagy ha ez nem lehetséges, kezelhetővé tehetők. A teljes folyamathoz tartozó mérőszámrendszer kialakításával és egy mérési környezet létrehozásával a folyamatok monitorozásával folyamatosan optimalizálható a vállalat működése.

ERP alapú környezetekben a folyamatmodellek által leírt tények nagy része használható lesz az ERP és más alkalmazások integrációját megvalósító rendszer koncepcionális modelljéhez. Az erőforrások és a szervezeti struktúrák tárolhatók már az ERP rendszer relációs adatbázisában, és fordítva, a tranzakciós és customizing adatok, referencia modellek felhasználhatóak onnan. Ehhez megfelelő leképezés szükséges az említett magas absztrakciós szintű modellek és az ERP rendszer adatbázis struktúrák között.

III.1 A Fő hipotézis

Az ERP rendszereket alkalmazó szervezeteket körülvevő konglomerátumban az integrációs problémák megoldásának kulcsa már középtávon is a korszerű alkalmazásfejlesztési paradigmák és módszerek – BPMS, SOA, MDA – kollaboratív használatával elérhető **szinergiák** kiaknázása.

III.1.1. Az Első segédhipotézis és igazolása

Az ARIS módszertan és eszköztár felhasználásával megvalósítható az ERP rendszert alkalmazó szervezet üzleti integrációs modellje.

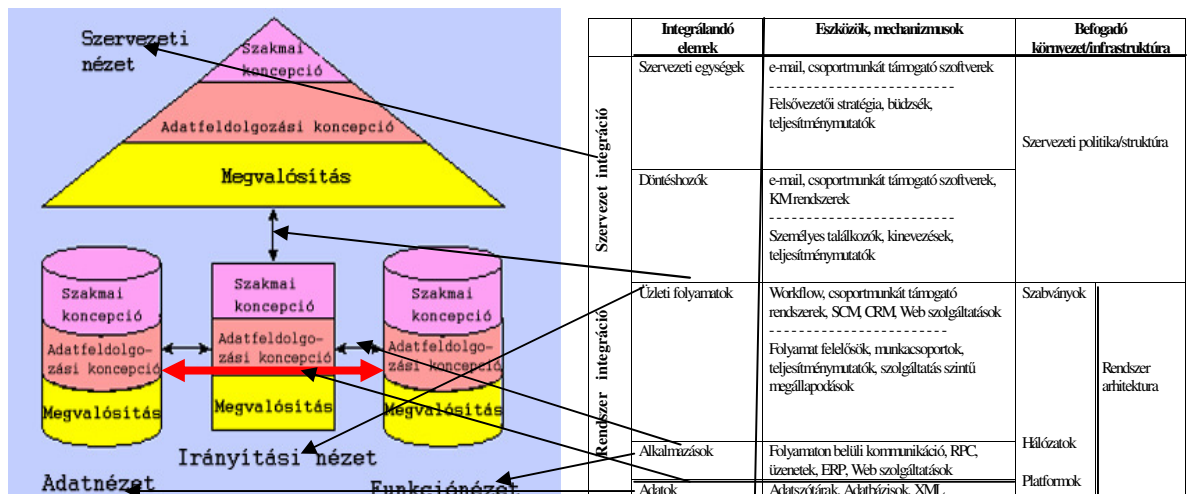
A dolgozat II.3.1. fejezetében adtam áttekintést a szervezetek üzleti integrációs szerkezetéről, a II.6. fejezetében pedig az ARIS módszertanról.

Az üzleti integrációs modell megvalósításához a következő okok miatt választottam az ARIS módszertant:

1. Az ARIS – „Integrált Információrendszerek Architektúrája” a vállalati információkat integráltan összefogó és kezelő üzleti folyamatok támogatását szolgáló informatikai rendszerek felépítését írja le.
2. A folyamatszervezési világpiac vezető modellező eszköze.
3. BPR céllal hazánkban is sok szervezet alkalmazza.

4. Piacvezető ERP rendszerek (pl. SAP) referenciamodelljei rendelkezésre állnak.
5. Az ARIS koncepció alapelvei (szétválasztás-, leíró szintek elve) lehetővé teszik az üzleti integrációs modell horizontális és vertikális rétegei közötti integráció megvalósítását.
6. A vállalati folyamatstruktúra kialakításához „fönről-le” módszert használ.

Az üzleti integrációs modell megvalósításához nyújtott támogatást az ARIS különböző nézeteiben és szintjein alkalmazható diagramtípusoknak az integrációs modell rétegeihez rendelésével szemléltetem.



31. ábra ARIS architektúra – üzleti integrációs modell

Adatintegráció

Az adatok különböző absztrakciós szinten is megjeleníthetők az adat nézetben. A szervezetek többségében az adat objektumok definiálására nagy számú kifejezés áll rendelkezésre, azonban a szervezeten belüli terminológia nem egységes – többféle kifejezést használnak ugyanarra a fogalomra vagy egy kifejezéssel különböző dolgokat jelölnek. A modellezési folyamatot megelőzően, illetve a folyamat közben célszerű az egyedülálló és konzisztens szakkifejezések definiálása a szervezeten belül.

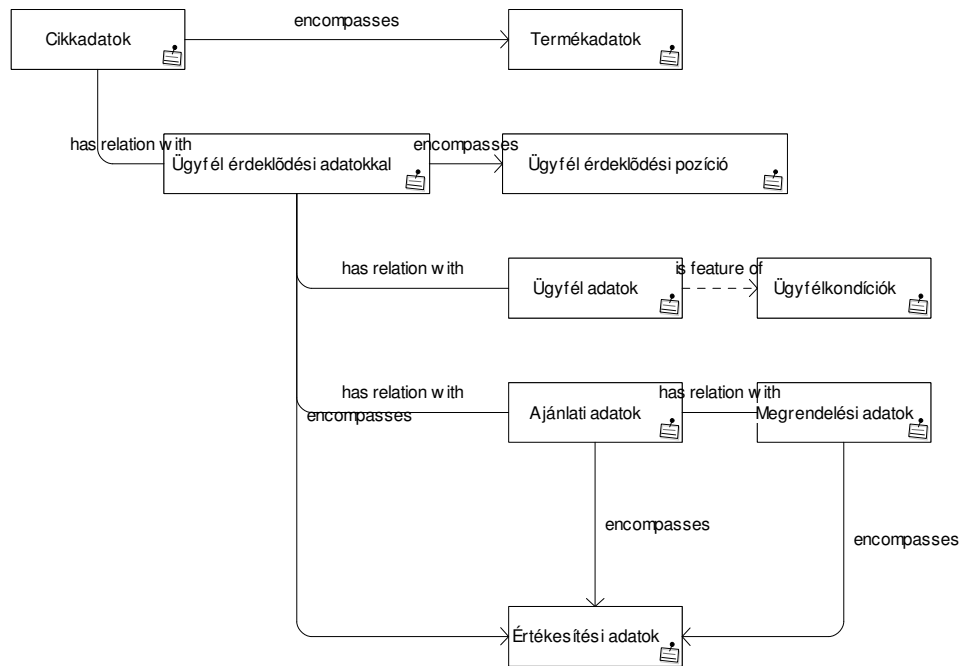
A különböző vállalati csoportok közötti hatékony kommunikációt a különböző fogalmak egységes kezelését lehetővé tevő szakkifejezés modell támogatja. A szakkifejezéseket a vállalaton belüli különböző absztrakciós szinteken lévő adat objektumok üzlethez kapcsolódó leírására használhatjuk.

Egy szakkifejezés ábrázolhat bizonyos számú szakkifejezést vagy egyetlen egy szakkifejezés példát. Az adat objektumok közt fennálló kapcsolatok megadásának a szervezeten belüli homogén (technikai) terminológia létrehozásában van jelentősége. A vállalat különböző részeiben létező szinonim kifejezések egyesítése sokkal egyszerűbbé teszi az üzleti objektumokkal kapcsolatos kommunikációt. A szakkifejezés modellben a szakkifejezések és az objektumok közti kapcsolatokat egyed típusként, kapcsolat típusként és attribútumként is leírhatjuk. A szakkifejezések hierarchiába rendezhetők.

A szakkifejezések kapcsolattípusai:

- ez egy – a szakkifejezések általánosításait jelöli
- magába foglal – a leíró szakkifejezések elrendezését írja le
- jellegzetesség – leíró jellegzetességek szakkifejezésekhez való hozzárendelését jelöli
- fajtája – absztrakt kifejezéseket rendel konkrét hivatkozásokhoz
- lehet – két szakkifejezés közti lehetséges hozzárendeléseket hoz létre
- osztályoz – szakkifejezéseket jellemez
- szinonima – alternatív elnevezést rendel az azonos hivatkozású objektumokhoz.

Értékesítési szakkifejezések

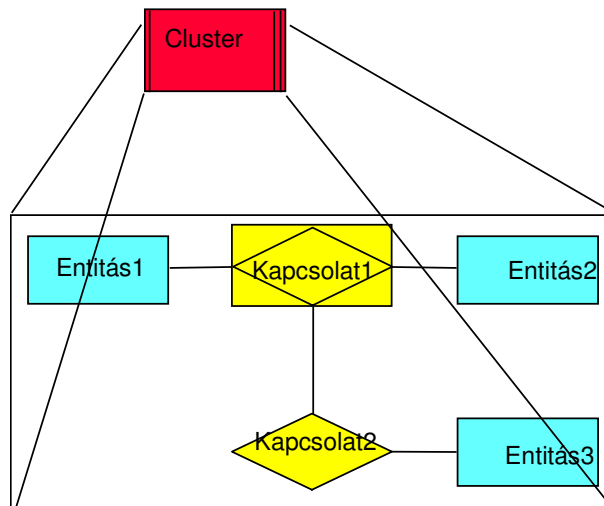


32. ábra Értékesítési adatok szakkifejezés modellje

Az ARIS-ban kialakított szakkifejezés modell alapul szolgálhat ontológiák kialakításához, amelyekkel magasabb szintű elemzések, ill. További integrációs problémák megoldása valósíthatók meg.

Az elsősorban üzlet-orientált szakkifejezés modellek mellett szükséges az IT- orientált adat modellezés is. Az ARIS-ban lehetőség van a klasszikus eERM (*extended entity relationship model*) kibővített egyed kapcsolat modellek alkalmazására. A két modellezési megközelítés kombinálható. Az adatok konzisztens, redundanciamentes integrálását az adatbázis elméletek igazolt osztályozás, generalizálás, specializáció és aggregálás mechanizmusai segítik.

Az aggregálás-összegyűjtés új objektumtípusok létrehozását jelenti már meglévő objektumok összefoglalásával. Az új objektumtípus új tulajdonságok hordozója lehet, így az aggregálás a kapcsolattípusok képzésének is megfelel.



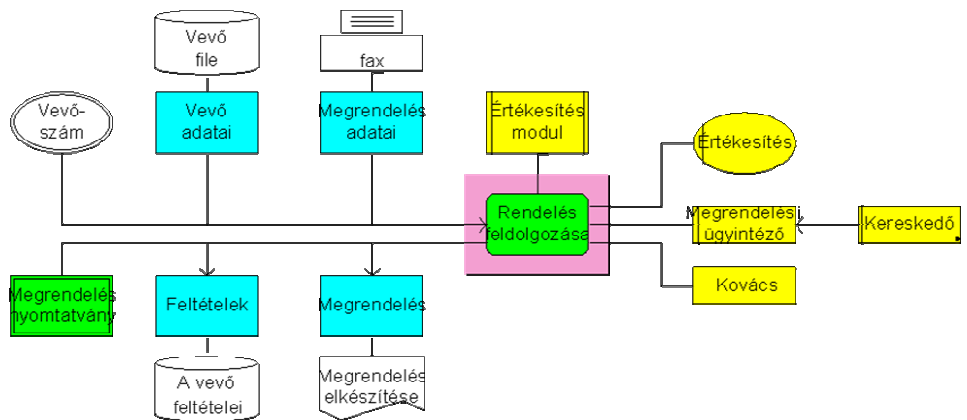
33. ábra Klaszter - kapcsolattípus képzés

A cluster komplex objektumok leírásához szükséges adatmodell entitás- és kapcsolat-típusainak logikai összefogása. A cluster-ek részletes kibontásában mind az entitás- és kapcsolat- típusok, mind pedig további clusterek is szerepelnek.

Az ARIS további modell típusokat kínál fel UML nyelven alapuló objektum-orientált adat modellezéshez. Ezek a modellek a folyamat nézethez lettek hozzárendelve az adatok és funkciók integrációja miatt.

Adat- és alkalmazásszint integráció

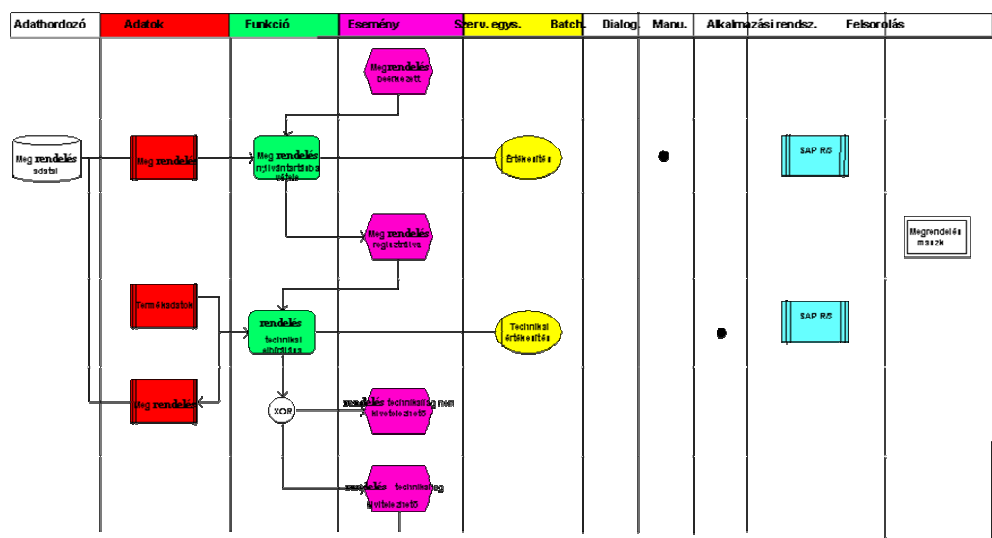
Az ARIS nézetek elemei és funkciói közti statikus kapcsolatokat funkció allokációs diagram segítségével ábrázolhatjuk és hierarchizálhatjuk. A hozzárendelési diagram lehetőséget teremt a tevékenységekhez rendelt input, output adatok, alkalmazások és információs objektumok integrációjának modellezésére.



34. ábra Funkció allokációs diagram

Az adat – és alkalmazásszint integrációjának részletes modellezésére további lehetőségek is adtak az ARIS-ban. A már említett UML modellezési lehetőség, valamint a kibővített eseményvezérelt folyamatlánc diagram szintén integrálja a funkciók, alkalmazások és adatok közti kapcsolatokat.

A folyamatláncokat un. „swim-lane” módon is megjeleníthetjük, így a modellezés az adatokra, a szervezeti és alkalmazási rendszerekre vonatkozó szempontokra fog koncentrálni.



35. ábra ARIS „swimlane” folyamatlánc

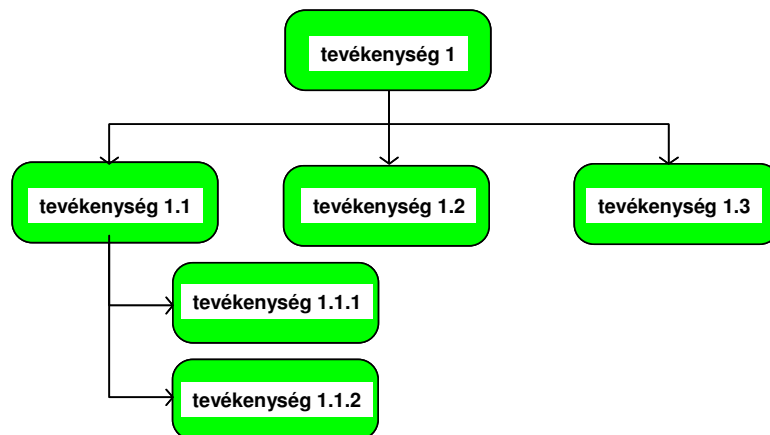
A funkciók oszlopokba való elhelyezésével a modell elemei között implicit kapcsolatok jönnek létre. Az UML Activity Diagramban szintén van oszlop-orientált modell típus.

Alkalmazásintegráció:

A vállalat által elvégzendő tevékenységek strukturális hierarchiáját funkciófával ábrázolhatjuk a funkcionézetben. A funkciófa folyamatorientált, objektumorientált, valamint tevékenységorientált lehet.

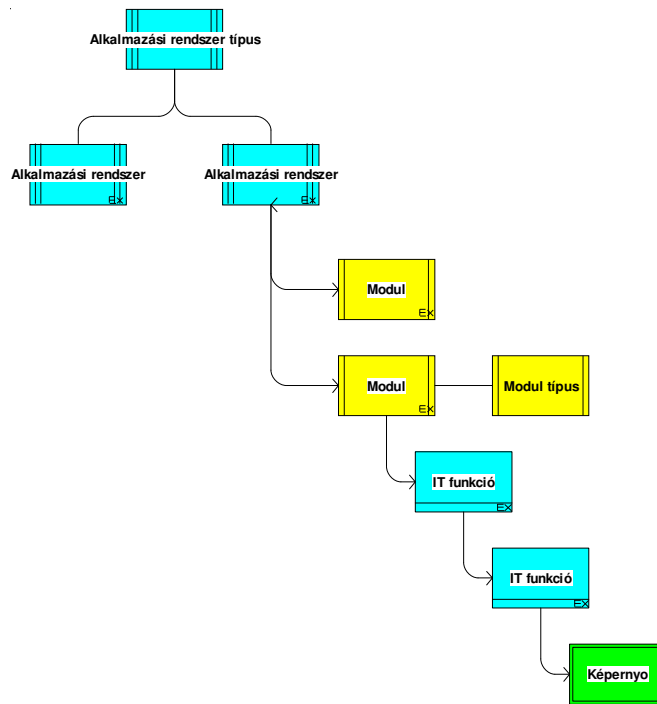
A hierarchiadiagramban statikus relációkat ábrázolhatunk a tevékenységek között. A tevékenységek a vállalati célok elérése érdekében az (információs) objektumokon elvégzett szakmai feladatokat ábrázolják.

A komplex funkciókat alfunkciókra bonthatjuk.



36. ábra Grafikus funkciófa (hierarchiadiagram)

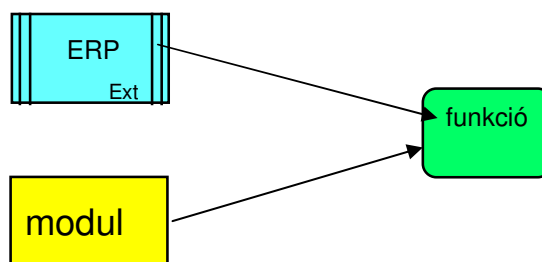
Ugyancsak a funkcionézetben, de a megvalósítás szinten írhatjuk le az alkalmazási rendszer típusok, modul típusok és IT funkció típusok hierarchizált struktúráját az alkalmazási rendszer típus diagram segítségével.



37. ábra Alkalmazási rendszer típus diagram

Alkalmazási rendszer diagrammal pedig az alkalmazási rendszerek, modulok és IT funkciók hierarchizált struktúráját ábrázolhatjuk a különböző operációs rendszerek és interfészek figyelembe vételével. A modell előnye, hogy az alkalmazás belső összefüggéseit egyszerre láthatjuk és nem kell minden elemet a folyamatban feltüntetni, elég ha a feldolgozási szint legalsó részét, pl. a tranzakciót jelenítjük meg.

Az alkalmazási rendszereken, modulokon átívelő üzleti folyamatok modellezése során az egyes funkciókat támogató informatikai alkalmazások ill. modulok hozzákapcsolhatók a funkcióhoz.



38. ábra Alkalmazás, modul funkcióhoz rendelése

Alkalmazás- és üzleti folyamat integráció

Az alkalmazások modellezése a funkcionézet megvalósítás szintjén történik ld. előző rész. Az alkalmazás- és üzleti folyamat integráció modellezési lehetőségei sok ponton egyeznek az „Adat- és alkalmazásszint integráció” részben leírt lehetőségekkel. Az alkalmazások az általuk támogatott funkciókon keresztül kapcsolódnak a kibővített eseményvezérelt folyamatláncokhoz.

Az ARIS kibővített folyamatmodelljei a „top-down” tervezés során felhasználhatóak az alkalmazási rendszerek követelményspecifikációinak kialakításához, valamint a különböző informatikai rendszerek illesztésével szemben támasztott követelmények meghatározásához.

Üzleti folyamatok integráció

Az ARIS-ban a folyamatok modellezése és elemzése a dinamikus irányítási nézetben lehetséges. A vállalat több szervezeti egységet is átívelő valós folyamatainak funkcionális és dinamikus jellemzése lehetetlen egyszerű diagramban. Az ARIS-ban a komplexitás folyamat hierarchia létrehozásával csökkenthető. Az absztrakció különböző szintjein magas szintű és részletes modellekkel dolgozhatunk.

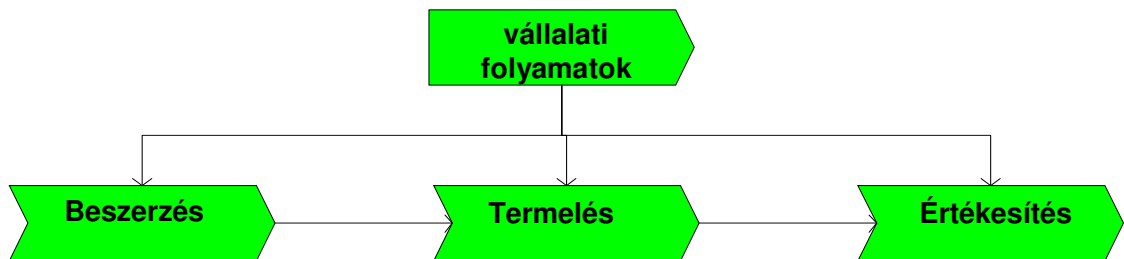
Az értékteremtő lánc diagrammal azon tevékenységek kapcsolatait reprezentáljuk, amelyek közvetlenül részt vesznek a vállalat értékteremtésében. Átfogó folyamatot

írhatunk le vele magas absztrakciós szinten, amit azután induló és áttekintő modellként használhatunk.

Az értékteremtő lánc diagramban a tevékenységek szekvenciális sorrendben követik egymást, ahol az időbeliség szabja meg sorrendet (nem úgy, mint a funkciófában).

A tevékenységek feloszthatóak altevékenységekre, így hierarchia hozható létre.

A funkciók között kapcsolatok ábrázolhatók. A hierarchiában folyamat-orientált felsőbbrendű / alsóbbrendű (mint a funkciófában). Az időben egymást követő azonos szinten levő funkciók, pedig „előzménye” kapcsolattal ábrázolhatók.



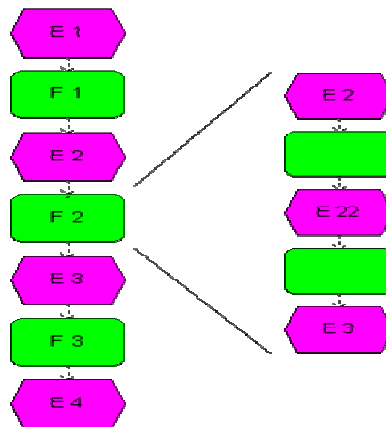
39. ábra Vállalati értékteremtő lánc diagram

Az értékteremtő lánc diagramot használhatjuk a kulcsfolyamatok azonosítására és sorba rendezésére. Egy alacsonyabb részletező szinten minden kulcsfolyamat komponenseit megjeleníthetjük egy újabb értékteremtő lánc diagrammal.

Az eseményvezérelt folyamatlánc diagram (EPC - Even driven Process Chain) segítségével lehetséges a vállalat által elvégzendő összes feladat konzisztens megjelenítése és leírása tartalmi és időbeni függőségükkel együtt. A feladatok összekapcsolása az őket kiváltó, valamint a feladat elvégzése által létrejött eseményeken keresztül történik. A folyamatmodelleket különböző részletezési szinteken és elvonatkoztatási síkokon lehet definiálni.

A “karcsú” EPC csak az időbeli és a logikai folyamat aspektusokat ábrázolja. A kibővített eseményvezérelt folyamatlánc diagram integrálja a funkciók és adatok, illetve a termék / szolgáltatás és szervezeti nézet közti statikus kapcsolatokat is az eEPC-be (extended Even driven Process Chain).

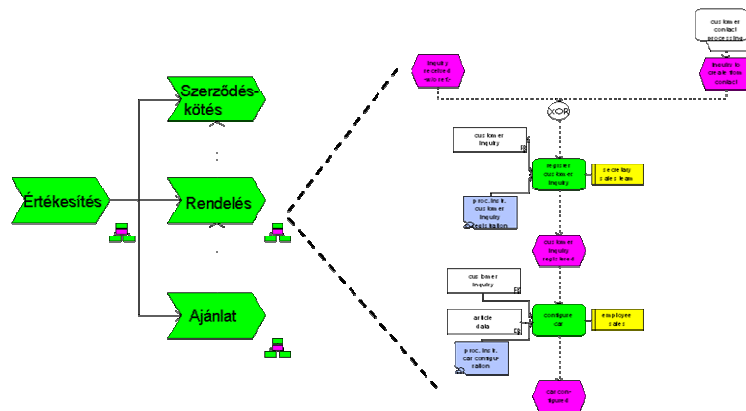
A folyamatláncok összetettségének és hosszának csökkentésére a folyamat egyes részei, azaz más szóval egy tevékenység részletezhető egy folyamatláncsal. A tevékenységet kiváltó, ill. az eredményét jelentő események megjelennek a részletező folyamatlánc kezdeti, ill. befejező eseményeként is. A folyamatok összekapcsolása azonos szinteken is eseményeken keresztül valósítható meg. Az egyik folyamat befejező eseménye megegyezik a másik folyamat kiváltó eseményével.



40. ábra Részfolyamat integrálása EPC-be

A rész-folyamatlánc a tevékenység mögé helyezhető és a következő hierarchiaszinten tárolható.

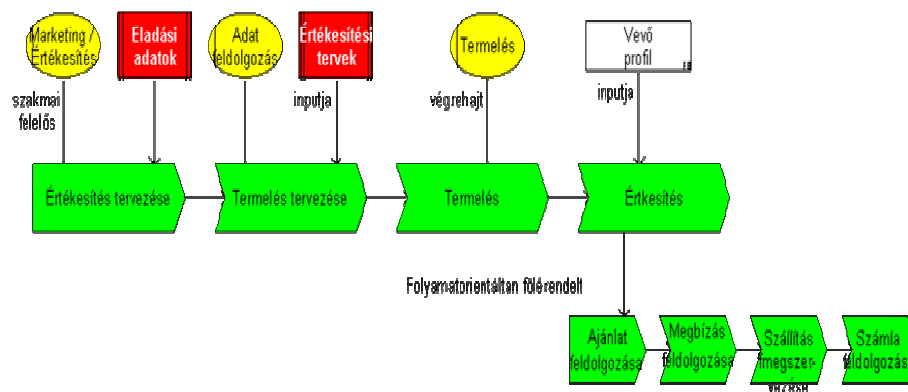
Értékteremtő lánc diagram tevékenységei kifejthetők eseményvezérelt folyamatláncsal, vagyis EPC folyamatok integrálhatóak értékteremtő lánc diagramba.



41. ábra eEPC integrálása értékteremtő lánc diagramba

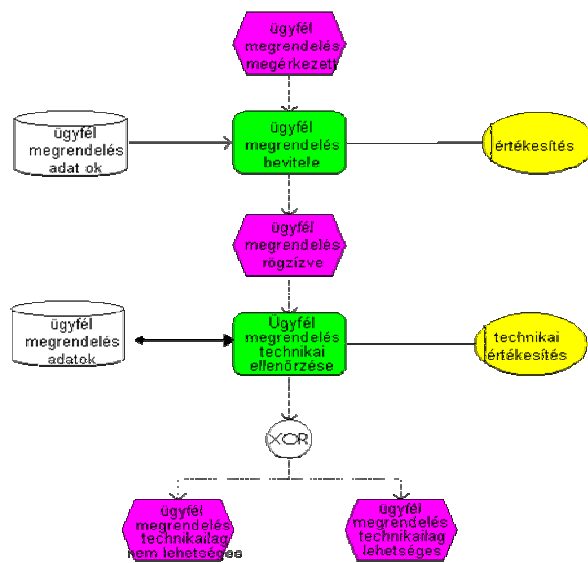
Üzleti folyamatok – szervezeti struktúra integráció

Az értékteremtő lánc diagramban folyamat struktúra mellett leírhatjuk a szervezeti elemek felelősségeit és az adatáramlást is a diagramban. A folyamatok aggregált megjelenítéséhez szervezeti egységek és absztrakt adatok leírását használhatjuk.



42. ábra Szervezeti elemek – magas szintű folyamatok integráció

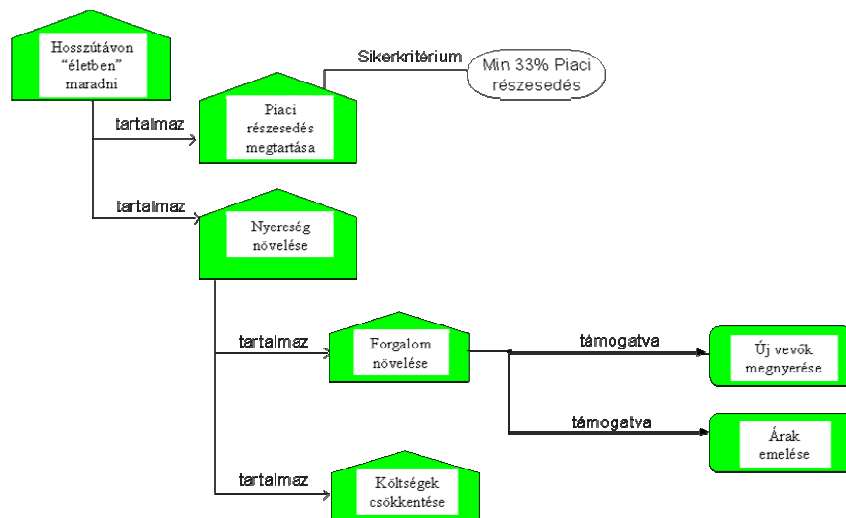
Az eEPC modellekben integrálhatjuk a funkciók, adatok és szervezeti nézet közti különböző típusú kapcsolatokat.



43. ábra Szervezeti elemek – eEPC integráció

Üzleti folyamatok – döntéshozatal integráció

A döntéshozók szempontjából a vállalat üzleti folyamatainak a vállalat stratégiai céljainak megvalósításának megfelelően kell működni. A hierarchizált vállalati célokat az ARIS-ban a céldiagramban definiálhatjuk. A céldiagram az ARIS funkció nézetében hozható létre. A célok elérésének mérésére mutatókat és paramétereket adhatunk meg. A sikerfaktorok hozzárendelésével lehet minősíteni a célokat. Minden egyes célnál ábrázolni lehet, hogy a vállalat mely tevékenységei vagy folyamatai segítik a cél elérését.



44. ábra Stratégiai célok – folyamat/tevékenység integráció

Szervezeti integráció

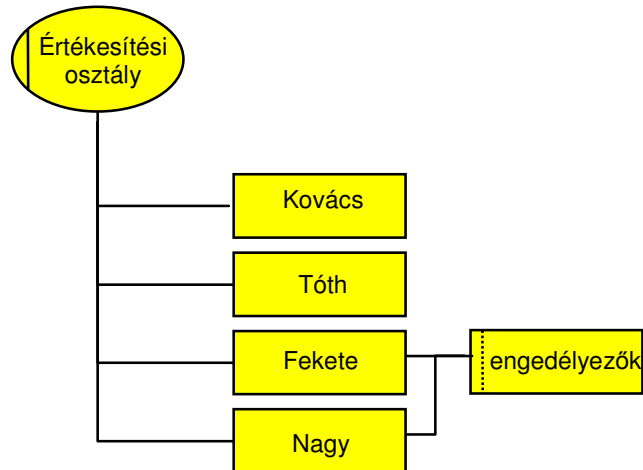
Az ARIS szervezeti nézetében azoknak a különböző egységeknek a statikus kapcsolatait ábrázolhatjuk, melyek a vállalatban belül a funkciók végrehajtásáért felelősek. A vállalati célok eléréséhez szükséges feladatok végrehajtói a szervezeti egységek. A hasonló tulajdonsággal rendelkező szervezeti egységek egy csoportba, az ún. szervezeti egység típusba sorolhatóak. Hasonló tulajdonság pl. az azonos hatáskör és felelősség.

A szervezeti ábra rajzolja le a vállalat szervezeti struktúráját a szervezeti elemekkel, azok kapcsolataival, ill. strukturális kritériumaival összhangban. A szervezeti egységek közötti kapcsolat típusok a következők lehetnek: funkcionális felelős, fegyelmi felelős, felelőse, felettes.

A beosztás a legkisebb szervezeti elem a vállalatban belül. A felelősségeket és a vezetői hatáskört a megfelelő munkaköri leírásban kell definiálni. A kapcsolat típusok a következők lehetnek: alkotja, technikai felettes, szakmai felettes.

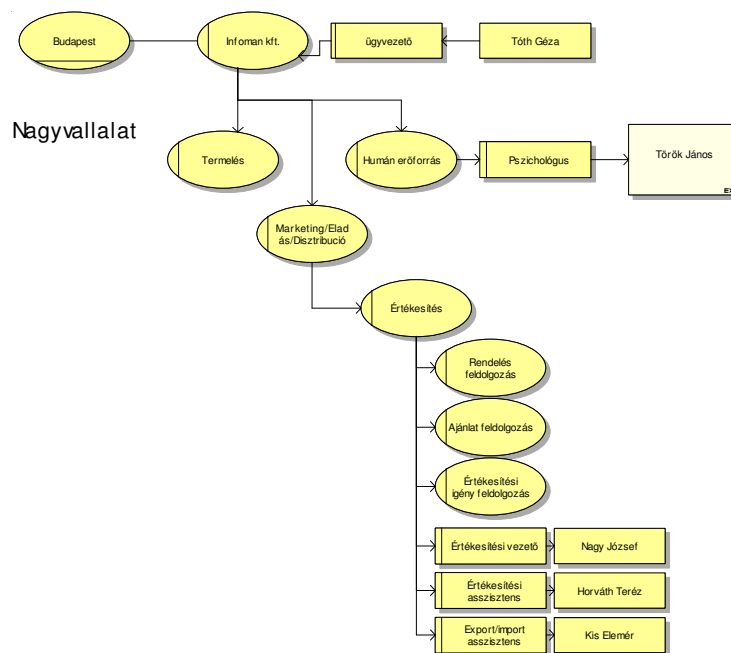
A személyek a vállalat valós alkalmazottai beosztásokhoz vagy közvetlenül szervezeti egységekhez lehetnek rendelve. A kapcsolat típusok a következők: dolgozója, betölti, helyettesíti.

Azonos tulajdonságokkal leírható személyek csoportját képezi a személy típus (pl. felhatalmazott, felelős).



Külső személyként olyan munkatársakat is ábrázolhatunk, akik nem tagjai a vállalat szervezeti hierarchiájának, de meghatározott időtartamig részt vesznek a vállalat működésében.

Csoportként jeleníthetünk meg a szervezetben specifikus feladaton (projekten) meghatározott ideig, adott erőforrás kerettel együtt dolgozó személyeket. A csoport szervezeti elemhez rendelhető, külső és belső munkatársak lehetnek a tagjai és vezetőit is kijelölhetünk hozzá.



III.1.2. A Második segédhipotézis és igazolása

Egy szervezet integrációs problémái a BPM, a SOA és a modell-vezéreltség elveinek kollaboratív alkalmazásával üzletvezérelt módon megoldhatóak.

Egy alkalmazás SOA alapú kifejlesztésének első lépése az üzleti folyamat modelljének megvalósítása. A SOA megfelelő kialakításának szempontjából azért fontos a BPM, hogy a működő folyamatokat optimalizálja, ahol szükséges felülvizsgálja, annak érdekében, hogy az újonnan kialakított IT rendszer lehető legjobb teljesítményét garantálni lehessen. Legtöbb esetben a meglévő folyamatok modellezéséből indulnak ki, majd ezt követi az optimalizálás. Az üzlet és az informatika közötti határt technológiai oldalról az üzleti folyamatok kezelését támogató BPM eszközök, a megvalósítás során pedig a különböző fejlesztési és tervezési módszertanok hidalják át. A SOA + BPM projekteket támogató kollaboratív módszertan lényege, hogy a különböző fázisok szereplői szorosan együttműködnek egymással és folyamatos a visszacsatolás. Az üzleti szakértők feladata a folyamatok modellezése, optimalizálása, monitorozása, az IT

alkalmazottak feladata pedig az optimalizált folyamatokat a valós IT működésbe átültetni.

Egy konkrét üzleti folyamat megvalósításán keresztül fogom bemutatni, hogy hogyan lehet szolgáltatás-orientált architektúrát kialakítani és egy komplex integrált üzleti folyamatot magas absztrakciós szintről a modell vezérelt módszertan szabályai szerint végrehajtható folyamattá transzformálni. Az alkalmazás szolgáltatás-orientált módon való fejlesztéséhez az IBM WebSphere termékcsalád eszközeit és modelltranszformációt használtam.

A következő okok miatt alkalmazom az IBM WebSphere eszközeit:

Az IBM WebSphere termékcsalád, mint azt a II.4.2 fejezetben részletesen kifejtettem piacvezető, kiforrott, megbízható, testre szabható megoldásokat kínál a folyamatmodellezés, a SOA, az integráció megvalósítására.

Beépített, validált modell transzformációs eszközt tartalmaz a BPMN -> BPEL leképezés megvalósítására.

Egy egyszerűen átlátható, gyakran előforduló és adott vállalat esetében gyakran végrehajtható üzleti folyamatot választottam az illusztrálásra, a megrendelést. Az, hogy sűrűn alkalmazzák az adott eljárást, indokolja a folyamat automatizálásának igényét. A folyamat tartalmaz emberi szereplőket, automatikus feladatokat és külső szolgáltatásokat egyaránt, így lehetőséget nyújt a különböző szintű integrációs szereplők folyamatba kapcsolásának bemutatására.

Modellezés

A modellezéshez az IBM WebSphere Business Modeler Advanced Trial Edition 6.1.1 magyar változatát használtam.

A feladatok közötti adatcserét, azaz a feladatok be- és kimenetét a munkaelemek teszik lehetővé.

A munkaelemek meghatározása

A munkaelemeknek célszerű valamilyen egyedi azonosítót adni (ID attribútum), amely a különböző példányokat megkülönbözteti egymástól. Mivel ez az attribútum minden munkaelemnek része, érdemes egy sablon munkaelemet definiálni, amelytől az összes többi származik, és amelynek egyetlen attribútuma egy egyedi azonosító.

Név	Típus	Minimum	Maximum	Egyedi	Alapértelmezés
ID	Egész (rövid)	1	1	Igaz	

45. ábra Egyedi attribútum megadása a munkaelem sablonnál

A megrendelési folyamatban a következő munkaelemek vesznek részt:

Áru

Az áru, ami a megrendelési igényben szerepel, azaz megrendelést lehet rá feladni.

Attribútumai:

Név - Az áru rövid megnevezése, mely szöveg típusú és meg kell adni.

Leírás - Általános leírás az árurol, mely szintén szöveg típusú, de nem szükséges megadni.

Ár - Az áru aktuális ára, melyet tizedes tört alakban kell megadni.

Megrendelési tétel

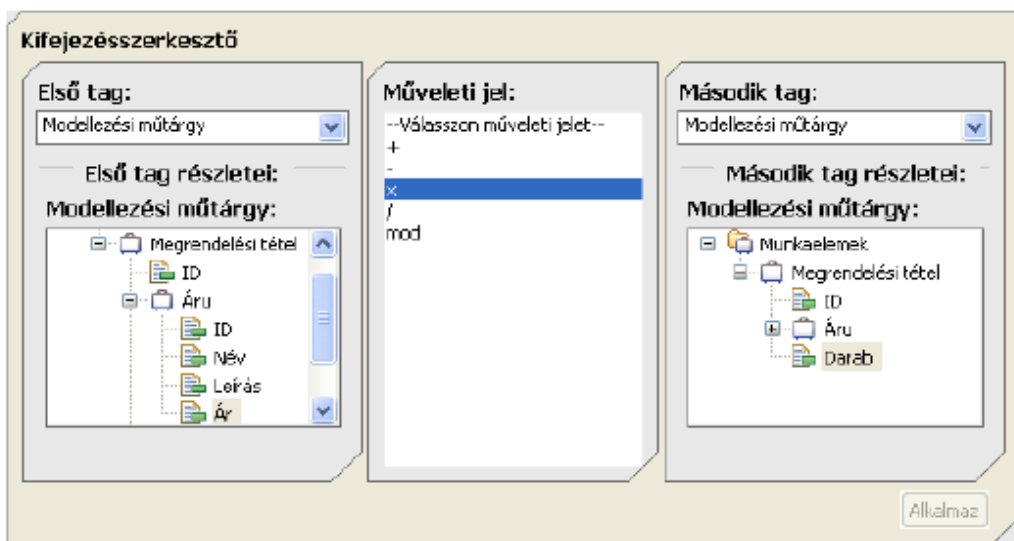
Egy adott áruból képzett csomag, amely tartalmazza azt is, hogy hány darabot rendelünk abból az áruból.

Attribútumai:

Áru - Az áru egy példánya, mely a tétel típusát adja meg. Ennek megfelelően Áru típusú és pontosan egyet kell megadni.

Darab - Egy egész típusú attribútum, amely azt mutatja meg, hogy hány darabot tartalmaz a tétel az adott áruból, ezért szükséges megadni.

Összeg - A megadott áru tételenkénti ára szorozva a darabszámmal. Az attribútum alapértelmezett érték mezőjében megadható egy kifejezés, ami ki is számítja az összes árat.



46. ábra A megrendelési tétel összértékének kiszámítására megadott kifejezés

Megrendelési igény

Egy konkrét megrendeléshez köthető megrendelési igény, amely tartalmazza a megrendelt tételek listáját, a feladás idejét és azt, hogy ki adta fel a megrendelést.

Attribútumai:

Megrendelési tételek - Megrendelési tétel típusú lista, amely tartalmazza a megrendelésben szereplő tételeket. Mivel itt elméletileg akármilyen sok tétel szerepelhet, ezért a maximum értéket n-re kell állítani.

Feladás ideje - Egy dátumidő típusú attribútum, mely az igény feladásának idejét rögzíti.

Feladó - Az igény feladójának neve, szöveges attribútum.

Név	Típus	Minimum	Maximum
ID	Egész (rövid)	1	1
Megrendelési tételek	Megrendelési tétel	1	n
Feladás ideje	Dátumidő	1	1
Feladó	Szöveg	1	1

47. ábra Számosság megadása egy munkaelem attribútumához

Számla

Egy megrendelési igényhez tartozó számla, amely tartalmazza az egyes tételek összegéből az esetleges engedmények után számított végösszeget.

Attribútumai:

Megrendelési igény - A megrendelési igény, amelyhez a számlát kiállították.

Kiállítás dátuma - Egy dátum típusú attribútum, mely a számla kiállításának dátumát jelöli.

Végösszeg - Az esetleges engedmények után kialakult végösszeg, mely tizedestört típusú attribútum.

Kedvezmény - A megrendelési folyamat során megállapított kedvezmény összege, amely szintén tizedestört típusú attribútum.

Teljesítési ütemterv

Szöveges dokumentum, amely tételesen tartalmazza a teljesítés vállalt ütemezését.

Attribútumai:

Megrendelési igény - A megrendelési igény, amelyhez az ütemterv készült.

Ütemterv leírás - Az ütemterv tételes leírása, amely szöveges attribútum. A megrendelési igény feladó attribútumához hasonlóan, az ütemterv részletes kidolgozása sem növelné a modell prezentációs értékét, ezért ebben az esetben is egy egyszerű szöveges attribútum az ütemterv.

A szükséges erőforrások meghatározása

Az erőforrások emberek vagy gépek lehetnek, ők végzik el a különböző feladatokat. A folyamatban csak emberi erőforrásokat használók, mivel a gépek csak impliciten szerepelnek a folyamatban, költségei marginálisak, ezért a modellezésük elhagyható.

Időtáblák

Az időtáblák segítségével lehet megadni, hogy a különböző erőforrások mikor érhetőek el. Az emberi erőforrások számára triviális időtábla a nappali műszak, mely minden nap reggel nyolc órától délután négyig tart. Természetesen lehetőség van szofisztikáltabb időtáblák megadására is, azonban az időtáblák szerepének bemutatására ennyi is elegendő.

Emberi erőforrás

Az emberi erőforrásokat szerepekként célszerű definiálni, mivel a feladatokat végző egyének kiléte nem feltétlenül ugyanaz a folyamat különböző lefutásai során, ennek az absztrakciónak a segítségével azonban közösen modellezhetőek.

Kétféle szerepet hoztam létre a modellben:

Termelésvezető - A teljesítési ütemtervét felelős személy. Ellenőrzi, hogy az elkészült ütemterv megvalósítható-e vagy sem. A termelésvezető fizetése, azaz az erőforrás költsége 5 000 Ft/óra. A szerephez rendelt szín a narancssárga.

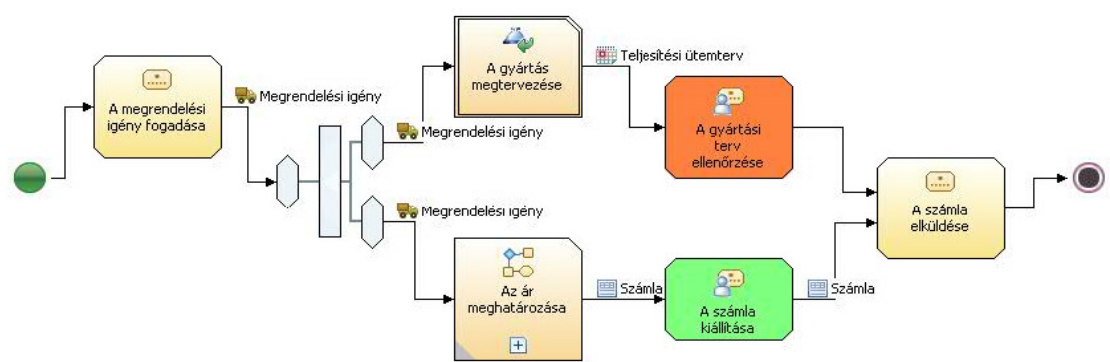
Értékesítési alkalmazott - A számla kiállításáért felelős személy. Egy utolsó ellenőrzést végez a kész számlán, majd továbbítja a címzett felé. Az értékesítési alkalmazott fizetése, azaz az erőforrás költsége 1 000 Ft/óra. A szerephez rendelt szín a zöld.

A folyamatok felvázolása és finomítása

A megrendelés modelljében egy globális folyamat van (Megrendelés), amely tartalmaz egy alfolyamatot (ár meghatározás) és egy globális szolgáltatást (gyártás megtervezés).

Megrendelés

Minta alkalmazásként a megrendelés üzleti folyamatát modellezem, így ez a neve az egyetlen globális folyamatnak is. Ebben a folyamatban összesen hat feladat és egy elágazás szerepel.



48. ábra A Megrendelés globális folyamat

A megrendelési igény fogadása - Ez a feladat fogadja a megrendelési igényeket a megrendelőktől és továbbítja őket feldolgozásra.

Elágazás - A folyamat ezen a ponton két párhuzamos szálon fut tovább ugyanazzal a megrendelési igény munkaelemmel. Ez logikailag azt jelenti, hogy a gyártási és a számla kiállítási részfolyamatok egymástól függetlenül hajthatók végre.

A gyártás megtervezése - Egy globális szolgáltatás, mely feldolgozza a megrendelést és létrehozza a gyártáshoz szükséges teljesítési ütemtervet.

Az ár meghatározása - Egy részfolyamat, amely a megrendelési igény számlázási értékét számolja ki és előkészíti a számlát.

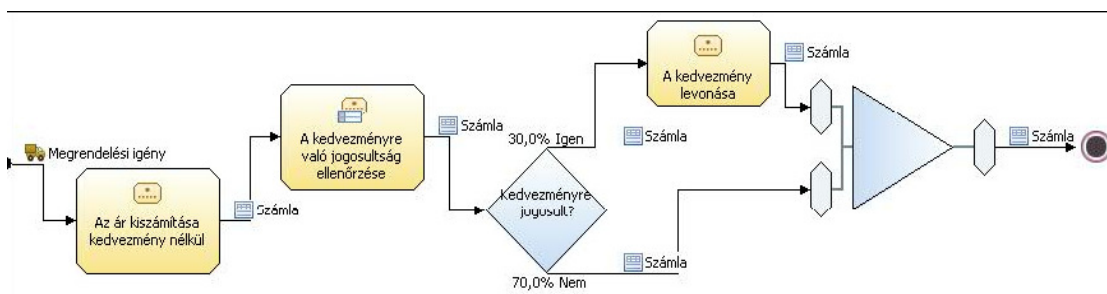
A gyártási terv ellenőrzése - Egy emberi feladat, amit a termelésvezető végez (ezt jelzi a narancssárga szín). Bemenetként megkapja a teljesítési ütemtervet, majd ellenőrzi, hogy minden rendben van-e, végül pedig továbbbenedi a folyamatot.

A számla kiállítása - Szintén egy emberi feladat, amelyet pedig az értékesítési alkalmazott végez (zöld szín jelzi). A bemenetként kapott számlát kiegészíti a szükséges információkkal és továbbítja elküldésre.

A számla elküldése - Egy automatikus feladat, amely a bemenetére érkező számlát továbbítja a megrendelő, azaz a megrendelési igény feladója felé.

Az ár meghatározása

Az ár meghatározása egy részfolyamat a megrendelési folyamaton belül, amely kiszámítja a bemenetére adott megrendelési igény összértékét. Ez után megnézi, hogy lehetséges-e kedvezményt adni a vevőnek. Ha igen, akkor kiszámítja a kedvezményt, majd az értékével csökkenti a megrendelés összegét. A kimenete pedig egy számla munkaelem, amely már tartalmazza a végösszeget.



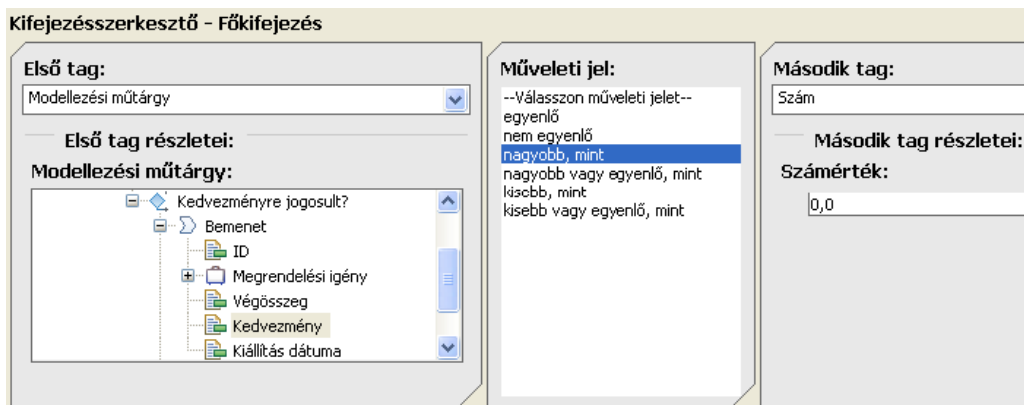
49. ábra Az ár meghatározása részfolyamat

Az ár kiszámítása kedvezmény nélkül – A feladat mindössze annyit csinál, hogy az összes a megrendelési igényben szereplő tétel összegét összeadja és az eredménnyel létrehoz egy számla munkaelemet, amelyet továbbít a következő feladatnak.

A kedvezményre való jogosultság ellenőrzése - Ez egy üzleti szabály feladat, amelyben üzleti szabályok segítségével adható meg, hogy milyen esetekben lehet kedvezményt adni a megrendelés árából. A megállapított kedvezmény értéke a számla munkaelem kedvezmény attribútumában tárolódik.

Kedvezményre jogosult? - Ezen a ponton kettéágazik a folyamat, aszerint, hogy volt-e kedvezmény megállapítva vagy sem, azaz hogy a számla munkaelem kedvezmény attribútuma nagyobb-e 0-nál. Amennyiben volt, a kedvezmény levonásra kerül a végösszezből, míg ha nem volt, akkor a végösszeg helyes, így az aktuális számla munkaelem lesz a kimenet.

A kedvezmény levonása - Ez a feladat a számla munkaelem kedvezmény attribútumának értékét levonja a végösszeg attribútumának értékéből.



50. ábra A Kedvezményre jogosult? - döntést kiértékelő kifejezés megadása

A gyártás megtervezése

A külső szolgáltatás által megvalósított feladatnak csak a bemenete és a kimenete ismert a folyamat számára. Ez a szolgáltatás lehet akár egy gyártásütemező szoftver, amelynek a bemenete egy megrendelési igény, a kimenete pedig a kész ütemterv.

A Modeler szimulációs lehetőséget biztosít a modellezett folyamatok dinamikus elemzésére. Ezzel már modell szinten optimalizálható a folyamat lefutási idő, erőforrásfelhasználás és költség szempontjából is.

Integráció és végrehajtás

Exportálás

A kész modellt exportálni kell az Integration Developer számára az integráció és a végrehajthatóság megvalósítása érdekében. A Modeler és az Integration Developer közötti export – import tranzakció „mögött” modell transzformáció, BPMN ->BPEL leképezés áll. A folyamatokat és feladatokat alapértelmezésben kérés-válasz típusú BPEL folyamatként exportálja a Modeler. Exportálás előtt meg kell győződni arról, hogy a folyamat valóban kompatibilis a Process Server-rel (amit az Integration Developer a folyamatok futtatására használ). Ehhez át kell váltani a modellezés

módját WebSphere Process Server-re és megnézni, hogy nem jelentek-e meg figyelmeztetések vagy hibák a projektben. Ez egyben validációt is jelent.

Amennyiben nincsen se hiba, se figyelmeztetés, a modell exportálható. Az egyes elemek exportálásához beállíthatjuk a műszaki attribútumokat, amelyek befolyásolják, hogy milyen objektum generálódik az elemből.

A megrendelés globális folyamata nem rendelkezik visszatérési értékkel, így egyirányú folyamattá kell leképezni. Ennek érdekében a műszaki attribútum nézet kérés fülén be kell állítani az egyirányú művelettípust.

Lényeges momentum a nem speciális feladatok implementációjának megadása. (Speciális feladatok az emberi feladat, üzletiszabály feladat és a globális szolgáltatás.) Jelen esetben ezek mind Java alkalmazások lesznek, amit a műszaki attribútum nézet implementációs fülén kell beállítani.

Az exportálásnál a WebSphere Integration Developer típust kell választani, ami egy összecsomagolt fájlt eredményez, benne a projekt modelljével és a generált fájlokkal.

Integráció

Az integrációt az IBM WebSphere Integration Developer 6.1 angol változatával valósítottam meg, amely integrálva tartalmaz egy teljes értékű IBM WebSphere Process Server tesztkörnyezetet is.

A modell importálása előtt szükséges néhány beállítást megtenni a Developerben. Ezek közül a legfontosabb, hogy az integrált tesztkörnyezetként szolgáló Process Server-t konfigurálni kell és el kell indítani.

ERP-t alkalmazó vállalatnál a feladatok adatbázisból dolgoznak. A mintaalkalmazásban nincs szükség elosztott képességekkel rendelkező, robosztus adatbázisok használatára, így az egyszerűség kedvéért az adatok bele vannak kódolva a Java programokba.

Az AmegrendelésiiigényfogadásaImpl.java fájl Bemenetifeltétel metódusának forráskódja:

```
public DataObject Bemenetifeltétel () {
    String namespace = " http :// SOA/ Munkaelemek ";
    System .out. println ("A megrendelési igény fogadása ");
    BOFactory boFactory =
    ( BOFactory ) ServiceManager . INSTANCE . locateService ("com/ibm/
    webspHERE /bo/
    BOFactory ");
    // alma áru
    DataObject aru1 = boFactory . create ( namespace , "Áru");
    aru1 . setString ("Név", " alma ");
    aru1 . setString (" Leírás ", " gyümölcs ");
    aru1 . setDouble ("Ár", 50);
    // körte áru
    DataObject aru2 = boFactory . create ( namespace , "Áru");
    aru2 . setString ("Név", " körte ");
    aru2 . setString (" Leírás ", " szintén gyümölcs ");
    aru2 . setDouble ("Ár", 70);
    // első tétel
    DataObject mt1 = boFactory . create ( namespace , " Megrendelésítétel ");
    mt1.set("Áru", aru1 );
    mt1. setInt (" Darab ", 1000) ;
    mt1.set(" Összeg ", aru1 . getDouble ("Ár") * mt1. getInt (" Darab "));
    // második tétel
    DataObject mt2 = boFactory . create ( namespace , " Megrendelésítétel ");
    mt2.set("Áru", aru2 );
    mt2. setInt (" Darab ", 2000) ;
    mt2.set(" Összeg ", aru2 . getDouble ("Ár") * mt2. getInt (" Darab "));
    // tételek listája
```

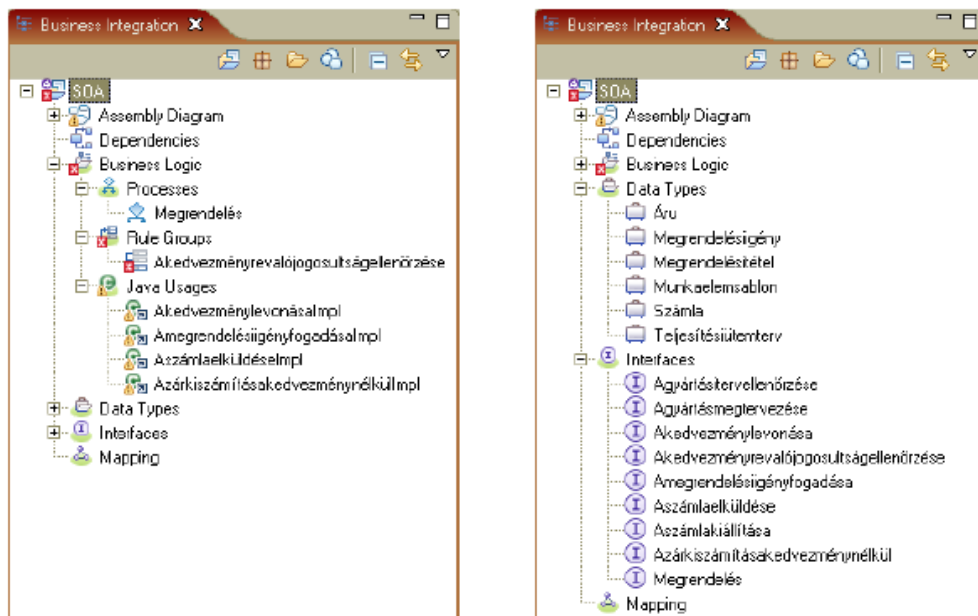
```

LinkedList < DataObject > mtlist = new LinkedList < DataObject >();
mtlist .add(mt1);
mtlist .add(mt2);
// megrendelési igény
DataObject mi = boFactory . create ( namespace , " Megrendelésiigény ");
mi.set(" Megrendelésitételek ", mtlist );
mi. setString (" Feladó ", " Nagymama ");
mi. setDate (" Feladásideje ", new Date ( System . currentTimeMillis ());
return mi;
}

```

Importálás

Az importálás során a folyamat modelljét importáljuk, a megvalósítást nem, ezért láthatók a projektben hibák és figyelmeztetések.



(a) Üzleti logika

(b) Adattípusok és interfészek

51. ábra A projekt az importálás után az Integration Developer-ben

A Modelerben nem lehet üzleti szabályt megadni, így az nem is generálódhatott az importálás során. A Modeler műszaki attribútum nézetében csak szöveges instrukciók adhatók az üzleti szabályok implementációjának módjára.

Üzleti szabályból két fajtát tudunk megadni: szabálycsoportot és döntési táblát. Az alkalmazásban a döntési táblát fogom használni, mivel a felépítése jobban megfelel a kedvezményre való jogosultság összetett feltételrendszerének definiálásához.

A Kedvezményre jogosultság üzleti szabály, amely eldönti, hogy mely megrendelés esetén adható kedvezmény és az milyen mértékű a következő szabályokból áll:

- Ha a megrendelés összege meghaladja a 10 000-et, akkor 3% kedvezmény adható.
- Ha a különböző tételek száma meghaladja a 10-et, akkor 1% kedvezmény adható.
- Ha mindkét feltétel teljesül, akkor 5% kedvezmény adható.

Conditions		
Bemenet.Végösszeg	<= 10000	> 10000
Bemenet.Megrendelésigény.Megrendelésitételek.size()	Kimenet.Kedvezmény	Kimenet.Kedvezmény
<= 10	0	Bemenet.Végösszeg * 0.03
> 10	Bemenet.Végösszeg * 0.01	Bemenet.Végösszeg * 0.05
Actions		

52. ábra Az üzletiszabály feladathoz megadott döntési tábla

Az emberi feladatoknál a feladat elvégzésének módját, azaz az interakciót lebonyolító szoftverklienst kell megadnunk. Jelen esetben ez a kliens a Business Process Choreographer Explorer, amely az Integration Developer-ben található beépítve.

A Java feladatok implementálása programozói feladat, és megfelelő programozási környezetben történik. SOA környezetben a kódok nagy része „újrafelhasználható” módon rendelkezésre áll, illetve lehetőség van az üzleti modellekből szintén modell

transzformációval UML modelleket képezni, majd ezeket felhasználni az adott programozási környezetben.

A folyamat futtatásához szükség volt kódolásra a Java feladatokba, hogy lehessen értékelni az eredményeket a futás során. A kódrészletekre nem térek ki teljes részletességgel, mivel a téma szempontjából nem relevánsak. A programok alapvető funkciókat látnak el, például kiírják, hogy éppen melyik feladat fut, vagy már láttuk a bemeneti adatokat kódolva.

Az Azárkiszámításakedvezmény nélkül Impl.java fájl Bemenetifeltétel metódusának forráskódja:

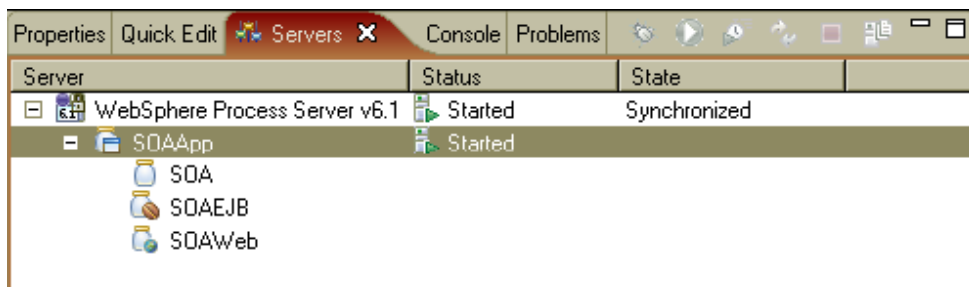
```
public DataObject Bemenetifeltétel ( DataObject Bemenet ) {
    System .out. println ("Az ár kiszámítása kedvezmény nélkül ");
    // Számla objektum létrehozása
    String namespace = " http :// SOA/ Munkaelemek ";
    BOFactory boFactory =
    ( BOFactory ) ServiceManager . INSTANCE . locateService ("com/ibm/
    websphere /bo/
    BOFactory ");
    DataObject szamla = boFactory . create ( namespace , " Számla ");
    org. eclipse .emf. ecore . util . EObjectEList mtlist =
    (org. eclipse .emf. ecore . util . EObjectEList ) Bemenet .get("
    Megrendelésítételek ");
    int size = mtlist . size ();
    double vegosszeg = 0;
    int i = 0;
    while (i < size ) {
    vegosszeg += mtlist .get(i). getDouble (" Osszeg ");
    i++;
    }
}
```

```

// Számla feltöltése adatokkal
szamla . setDouble (" Végösszeg ", vegosszeg );
szamla . setDouble (" Kedvezmény ", 0);
szamla .set(" Megrendelésiigény ", Bemenet );
return szamla ;
}

```

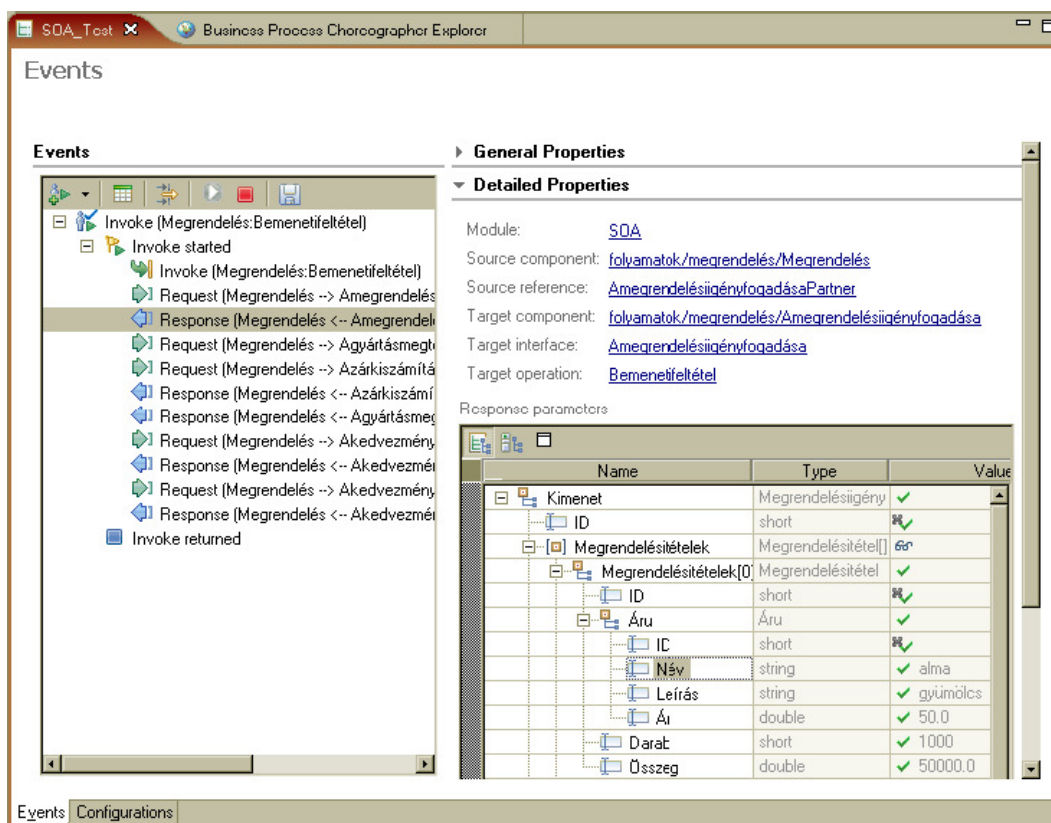
Az alkalmazás futtatásához a projektet publikálni kell a már futó tesztszerveren. Ha a közzététel hiba nélkül lezajlik, az alábbi ábrán látható eredményt kapjuk a „Servers” fülön.



53. ábra A futtatható SOA alkalmazás

Tesztelés

A folyamatot a beépített tesztkliens segítségével teszteltem. Tesztelés során figyelemmel kísérhetjük, hogy az egyes feladatok milyen bemenettel és kimenettel rendelkeznek.

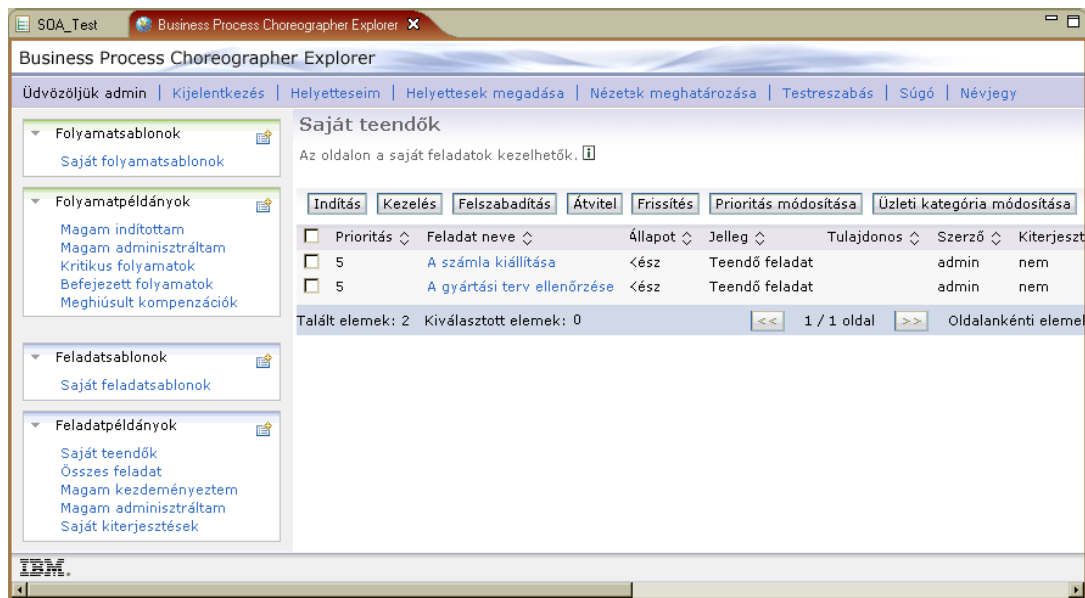


54. ábra A Megrendelés folyamat a tesztkörnyezetben

Az ábrán látható, hogy az aktuális folyamat során a megrendelések között szerepel egy alma tétel, aminek az ára 50 egység és 1000 darabot rendeltek belőle, így a tételösszeg 50 000 egység.

Emberi feladatok

Az emberi feladatok emberi interakciót kívánnak meg, ezért a korábban beállított Business Process Choreographer Explorer-t alkalmazom az emberi tevékenységek elvégzésére. Amint a folyamat eljut egy emberi feladathoz, megáll és vár a beavatkozásra. A BPC Explorer-t elindítva azonnal látható, hogy mely feladatok várnak.



55. ábra A várakozó emberi feladatok a BPC Explorer-ben

Az Explorer-ben nem csak továbbindíthatjuk a folyamatot, hanem láthatjuk a bemenő és szerkeszthetjük a kimenő változókat. Éppen ezért az emberi feladatok kiválóan alkalmasak az ellenőrző funkció betöltésére a folyamatban.

Természetesen nem csak a BPC Explorer használható az emberi feladatok elvégzésére, lehetőség van generált form-okat alkalmazni vagy akár saját webalkalmazást írni erre a célra, azonban ez nem célja a dolgozatnak.

Az alkalmazás megvalósításával tehát – melyben egy megrendelési folyamatot modelleztem a WebSphere Modeler-ben, majd implementáltam a WebSphere Integration Developer-ben - bemutattam, hogy hogyan hozható létre üzletvezérelt módon egy integrált alkalmazás szolgáltatás-orientált környezetben.

III.1.3. A Harmadik segédhipotézis és igazolása

Az ERP rendszert alkalmazó szervezet a bevezetés során kialakított, majd működése során optimalizált BPMS folyamatmodelljeit újrafelhasználva gyorsan és költséghatékonyan képes a környezet változásait integrált informatikai rendszereivel és alkalmazásaival is követni.

Tegyük fel, hogy a vizsgált vállalat ARIS módszertant alkalmazott a különböző alkalmazásai és az ERP rendszerének a bevezetése, üzleti architektúrájának kialakítása során, és a folyamat modelljeit a bevezetés óta karbantartja a közös adatbázisban. Az ARIS piaci helyzete miatt ezzel igen sok vállalatot bevettünk a „gondolatkísérletbe” még hazai vonatkozásban is.

Tegyük fel, hogy a vállalat IBM Websphere platformon működteti különböző alkalmazásait. Ezen „jövőbe mutató” feltételezéssel jelentősen leszűkítettük ugyan a vizsgált vállalatokat, de az

IBM Websphere piaci eredményei alapján még mindig jelentős méretű a kísérleti halmazunk.

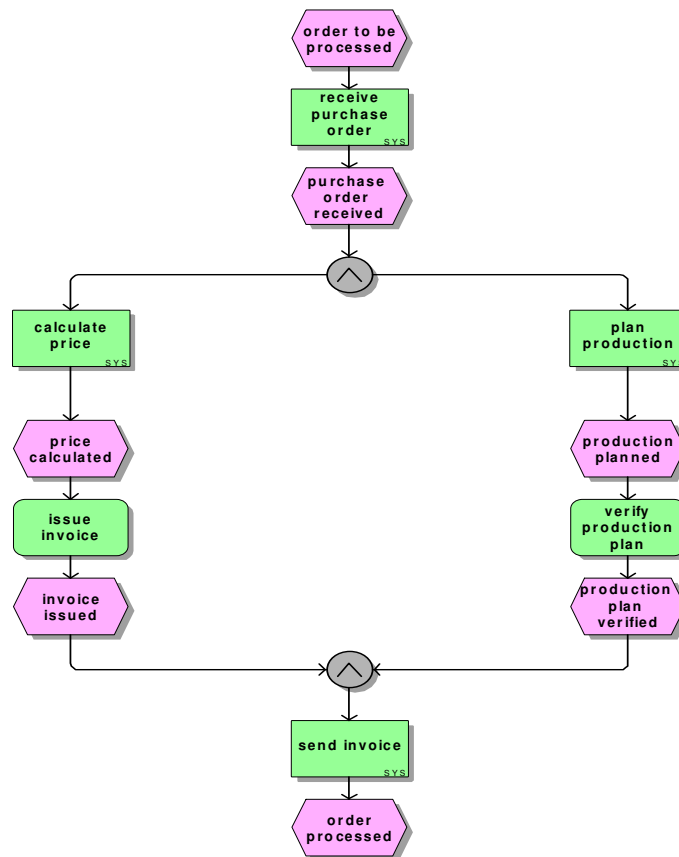
Tekintsünk egy olyan változást a vállalat életében, ami stratégiai szempontból gyors reagálást igényel. Egy olyan esetet vizsgáljunk, ami manapság igen gyakori, tehát a kísérleti halmazunk biztosan nem az üres halmaz.

A vállalat termékeit online árusítja egy katalógus segítségével. Mióta az online rendelések száma naponta nő és a hosszú feldolgozási idő már nem elfogadható, a cél az lett, hogy az alkatrészek rendelését a lehető legmagasabb szinten automatizálják service-k segítségével.

A tézis igazolásához az ARIS for SOA eszközt, és a 10 lépéses módszertani eljárást fogom alkalmazni.

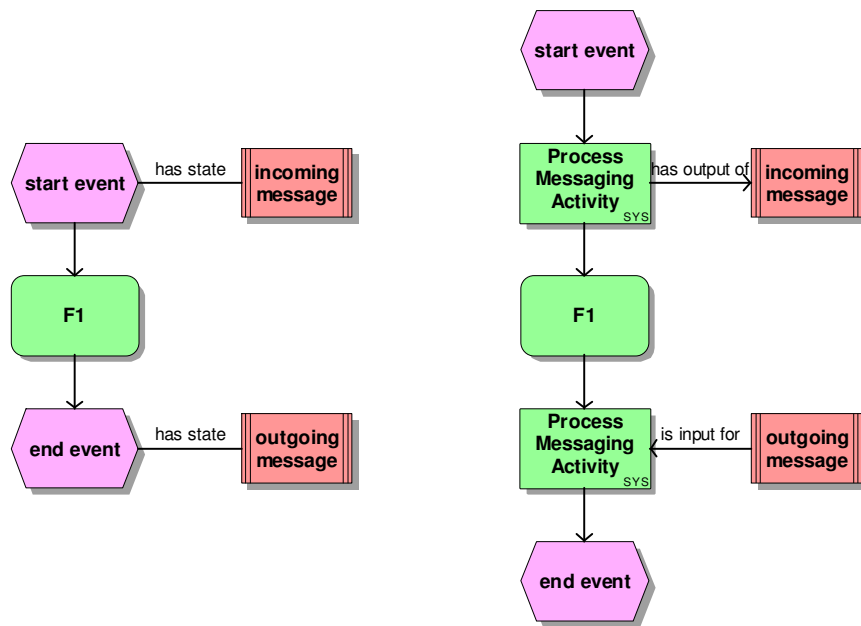
1. Service orientált üzleti folyamatok modellezése

Az induló, jelenlegi EPC folyamatot az ARIS adatbázisból egyszerűen betöltjük az ARIS SOA Architect-be, és továbbfejlesztjük.

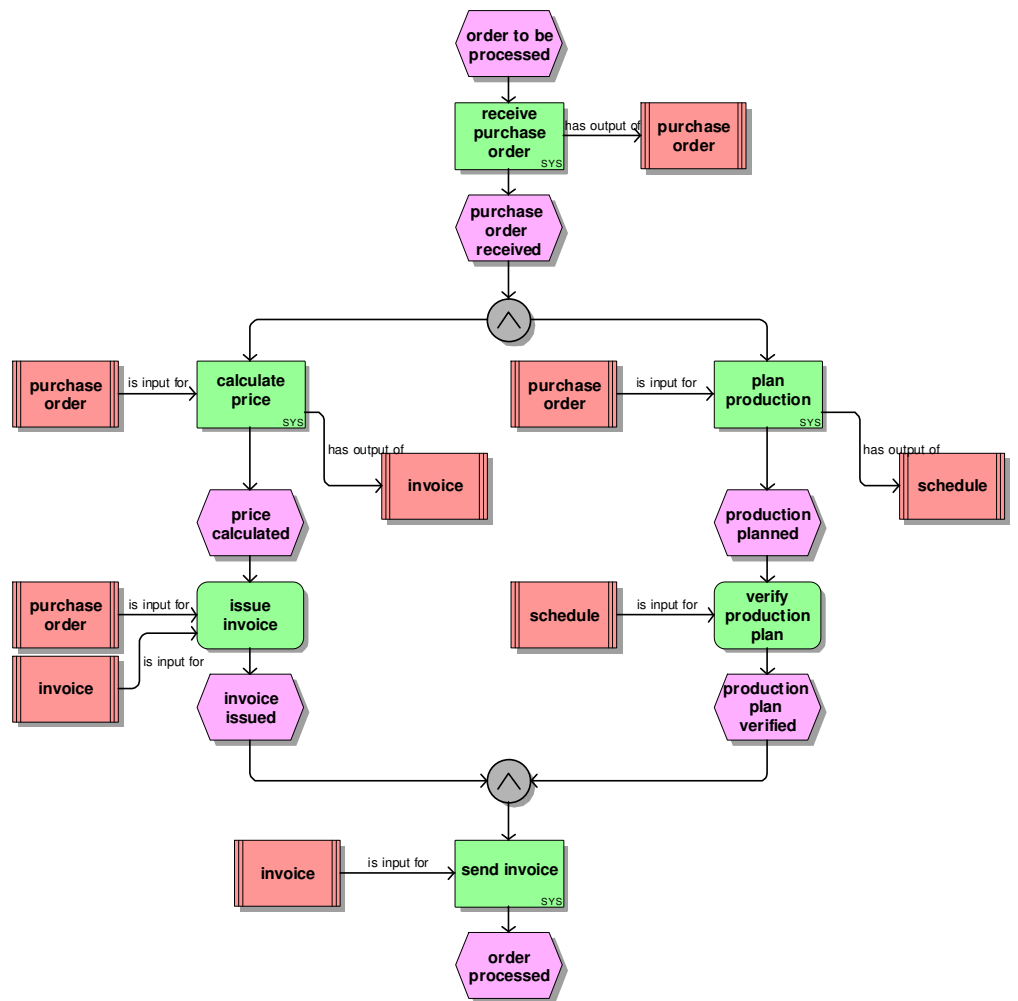


56. ábra A megrendelés ARIS modellje

Ahhoz, hogy a folyamatot majd szolgáltatásként használhassuk, interface-t (WSDL) kell tudni generálni hozzá. Az EPC-ben az események reprezentálják az interface-ket. (Az esemény írja le egy információs objektum üzleti folyamatban betöltött státuszát. Ez a státusz irányíthatja, vagy befolyásolhatja az üzleti menetet. Az eseményekben szereplő információs objektumok hivatkozásai az adat nézetben találhatóak.) A szolgáltatásorientált üzleti folyamat Input/output adatait össze kell kapcsolni a start és end eseményekbe, vagy üzenetközvetítő tevékenységekbe (adott feladat elvégzésére rendelt rendszerfunkció).

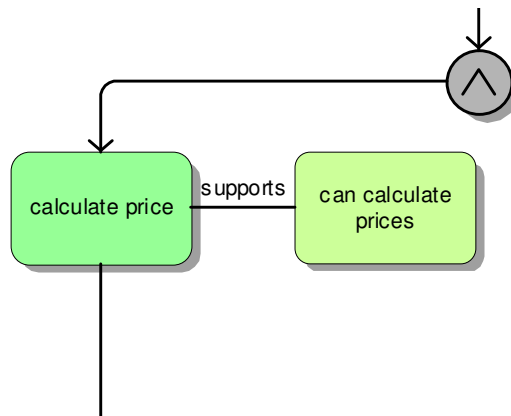


57. ábra A szolgáltatás interface modellezése EPC-ben



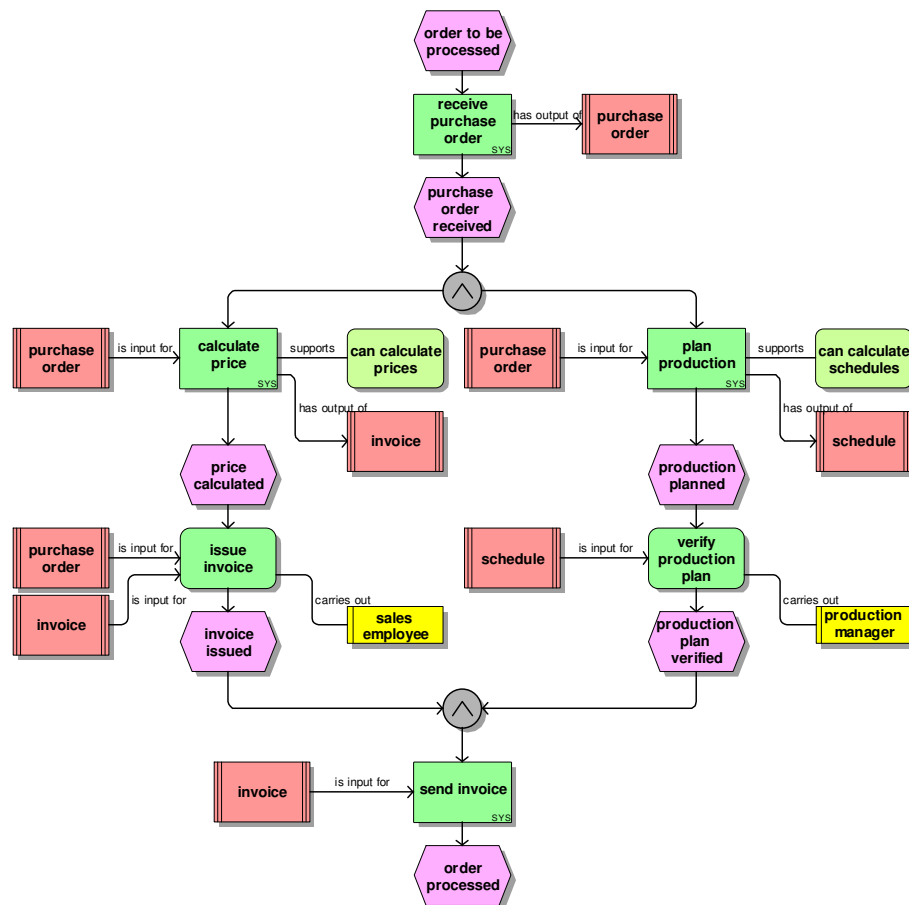
58. ábra Adatokkal, rendszerfunkciókkal és interfészekkel kiegészített EPC

Azokhoz a funkciókhoz, amelyet szolgáltatások végeznek, hozzákapcsoljuk a követelményeket. A Service követelmények “IS function” objektum típusok az ARIS-ban.



59. ábra Service – követelmény hozzárendelés

A manuális funkciókat a “carries out” kapcsolaton keresztül kapcsolhatjuk a szervezeti elemekhez.



60. ábra A megrendelés szolgáltatás orientált üzleti folyamat modellje

2. Szolgáltatás- és adat- jegyzékek összeállítása

Az üzleti folyamat alapján tudja az ARIS SOA Architect azonosítani és összehangolni a service-eket és a végrehajtó platform adatobjektumait. Az üzleti objektumok mellé importálni kell a végrehajtó platformról a service-eket és adatobjektumokat az ARIS SOA Architect-be!

Az adatjegyzék összeállítása

Az adatjegyzék egy helyre gyűjti a szolgáltatások által használt input és output adatobjektumokat. Az üzleti és technikai adatobjektumok a folyamaton keresztül képezhetők le egymásra.

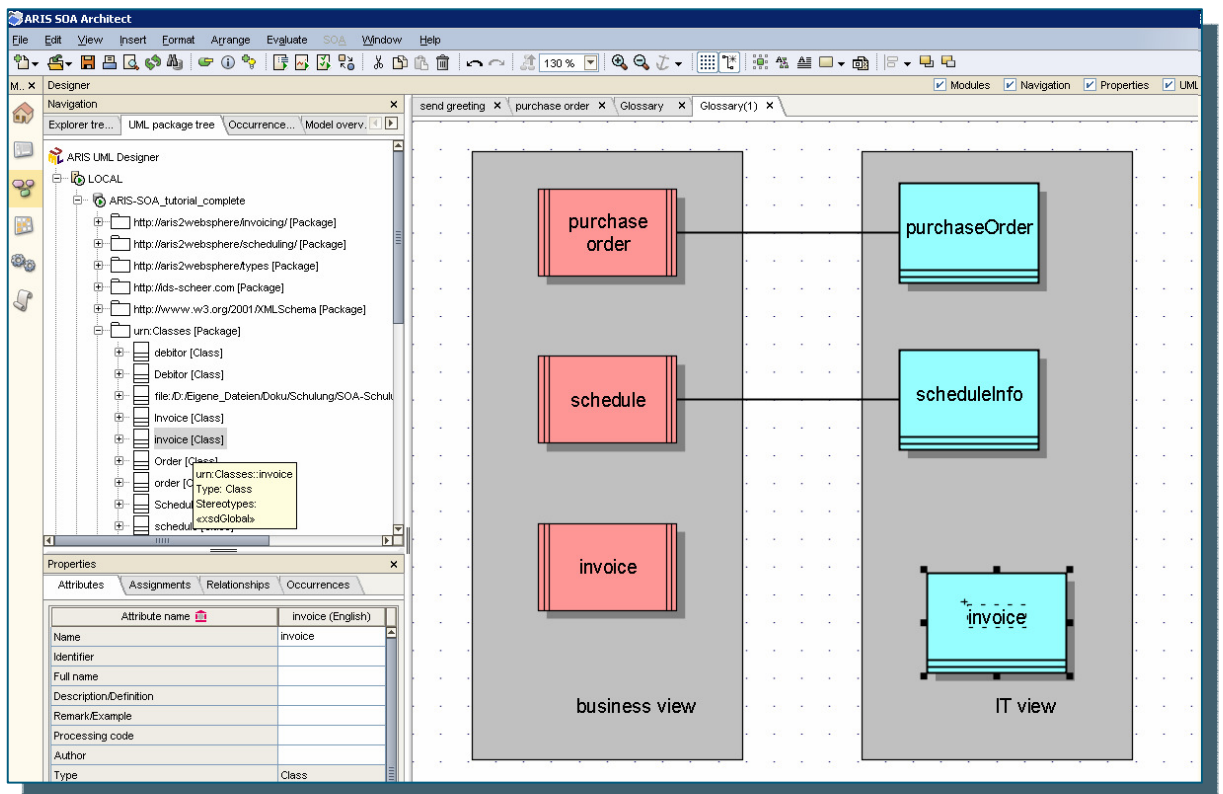
A SOA adat struktúrák importálása a végrehajtó platformról XSD file formájában történik.

Az ARIS “Designer” minden importált adatfájlhoz egy UML package-t hoz létre, ami tartalmazza az adat objektumokat. Minden importált adat objektumhoz két osztályt hoz létre: egy “xsdComplexType” stereotype és egy “xsdGlobal” stereotype osztályt.

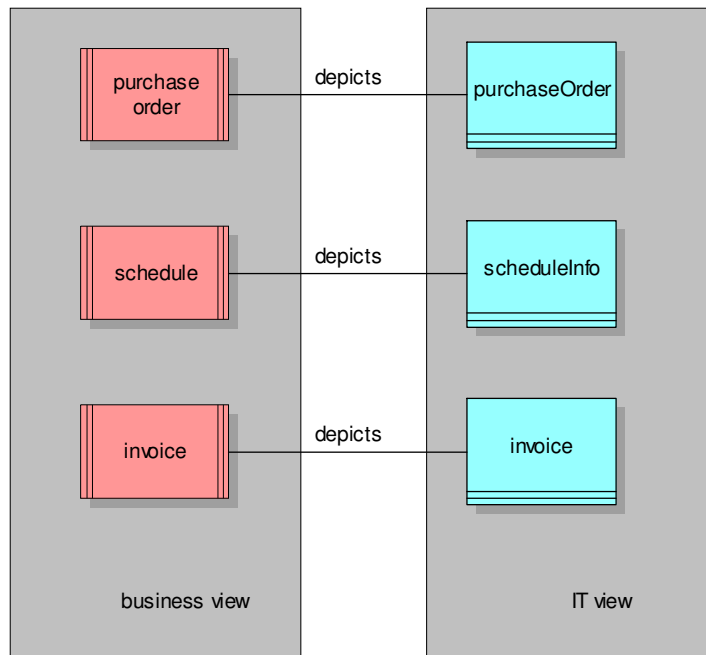
A stereotype-ok használatával kiterjeszthetjük a metamodelt újonnan hozzáadott új (meta)modell konstrukciókkal, amelyek használhatóak és előfordulhatnak UML diagramokban. A stereotype mindig az UML meta modell alapobjektumára vonatkozik. Megjelölt értékek és konstrukciók által határozható meg, speciális tulajdonságokat adnak át és így megkülönböztethetőek lesznek az alap objektumoktól

A stereotype-k alkalmazásával az újrafelhasználhatóság és beépíthetőség foka megnő, ugyanakkor a karbantartásra fordított idő lecsökken

Az “xsdGlobal” stereotype osztálynak létre kell hozni egy class diagram-ot, azaz a class diagram modell típus új ARIS modelljét kell létrehozni. Az EPC minden input és output adatát le kell írni (business data objektumok), majd ki kell másolni és beilleszteni az EPC minden csoportját a class diagramba.



61. ábra Class diagram létrehozása (Drag & Drop technikát alkalmazhatunk)



62. ábra Üzleti és technikai entitások közötti kapcsolat

Az adatjegyzékben kapcsolhatjuk össze az EPC-ben használt adatobjektumokat és a service platformról importált osztályokat.

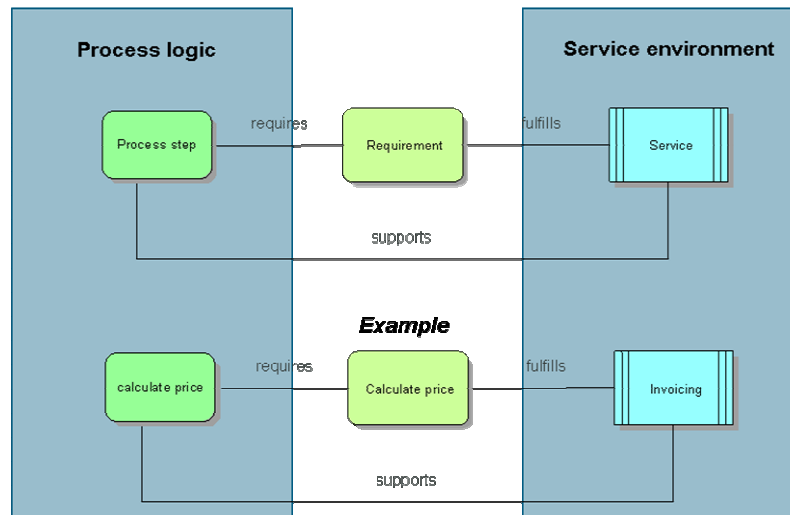
- Az üzleti nézet leírja az EPC minden input és output adatát (business data objects)
- Az IT nézet leírja az importált adattípusokat (class) (technical equivalents)
- Minden input/output adatot a megfelelő műszaki adattípusba kapcsol a depicts kapcsolat típus által

Tevékenység jegyzék összeállítása

A tevékenység jegyzék egy helyre gyűjti a szolgáltatások egy csoportját és azok tevékenységeit.

A service leírás importálása WSDL formában történik. Az ARIS SOA Architect automatikusan képes generálni a szükséges új modelleket: access diagram az üzleti nézethez, UML component diagram az IT nézethez.

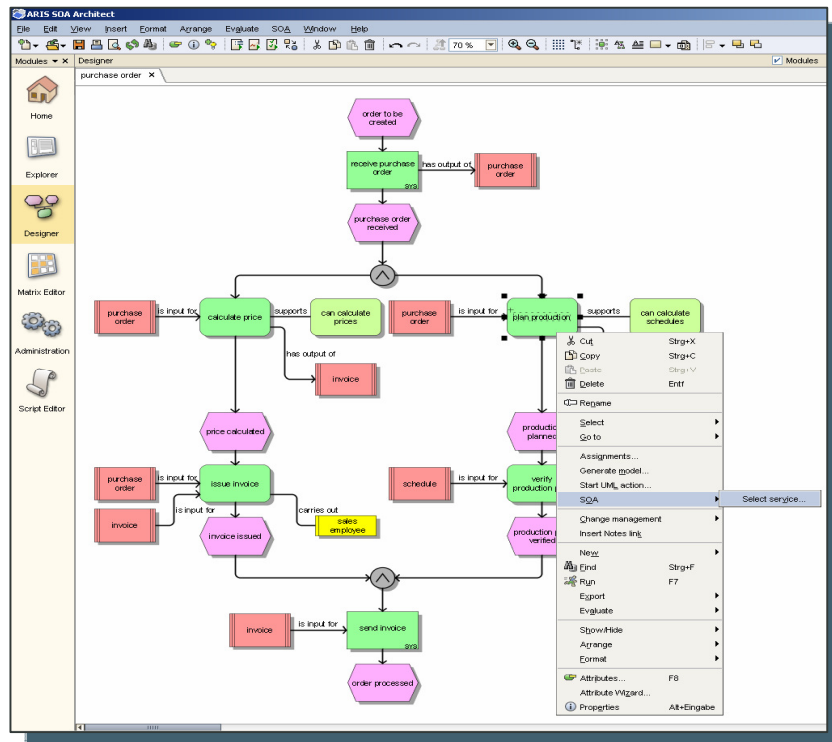
A szolgáltatás szemantikai leírásához a folyamat PIM szintű modelljében megadott szolgáltatás követelmény objektumot használhatjuk fel.



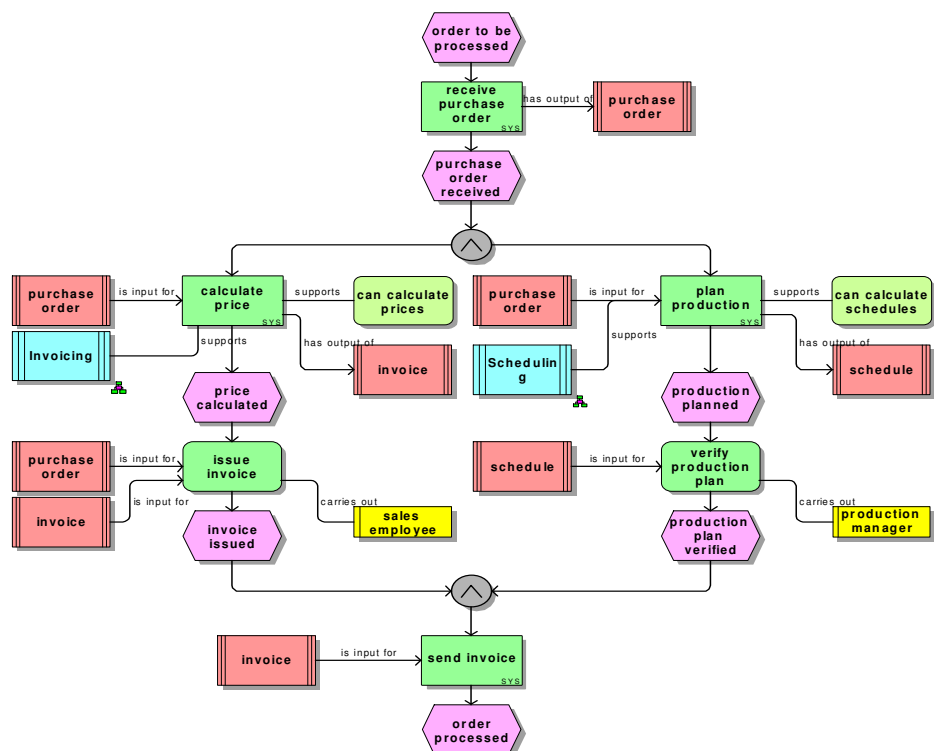
63. ábra A folyamat és az IT környezet közötti kapcsolat

3. A service-k összekapcsolása folyamatokkal és adatokkal

Az üzleti folyamatok és az IT eljárások közötti tényleges kapcsolatot kell meghatározni. A modellezett folyamatok végrehajtható service-eket tartalmaznak, amelyeket szemantikai leírásuk alapján lehet kiválasztani. A kiválasztás lehetséges az adatelemek, illetve a szolgáltatás követelményeket leíró objektumok által.



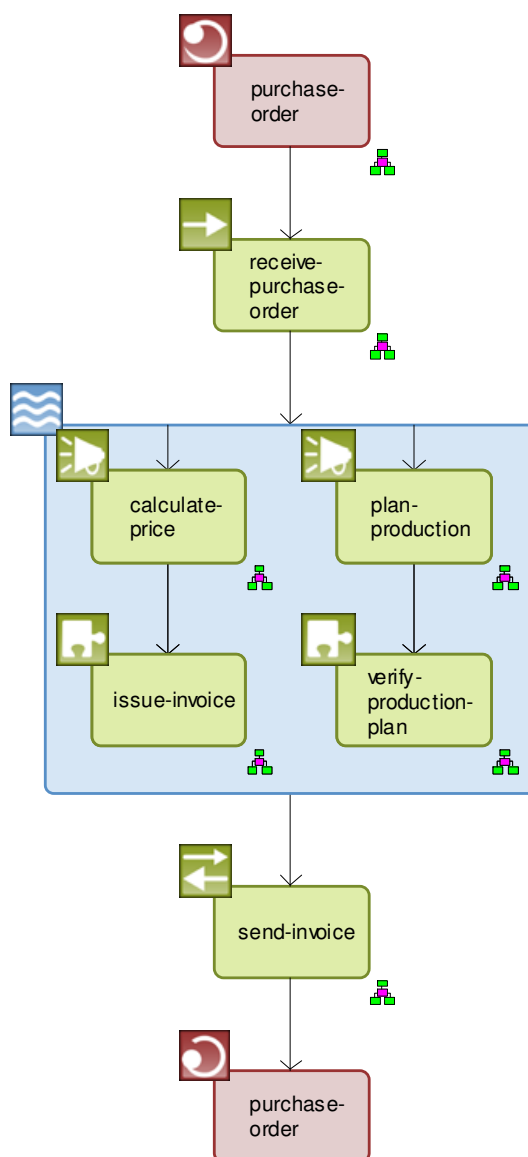
64. ábra A szolgáltatások kiválasztása



65. ábra A megrendelés szolgáltatás-orientált folyamata adatokkal és szolgáltatásokkal

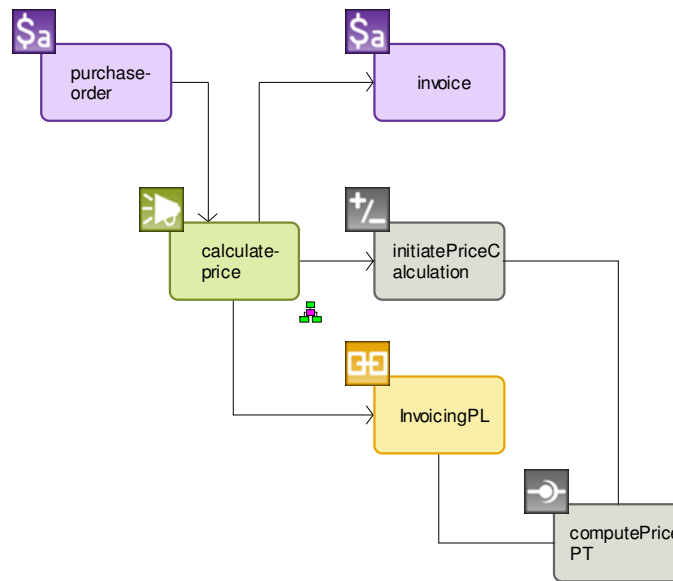
4. Üzleti folyamatok IT folyamatokká konvertálása

Ebben a lépésben történik a futtatható, standardokhoz alkalmazkodó BPEL modell generálása a service orientált alapon készült folyamat struktúrájának megtartásával. Az EPC -> BPEL modell transzformáció előtt szemantikai ellenőrzést, validálást kell végezni. A transzformáció teste szabható, azaz beállítható például, hogy szinkron, vagy aszinkron a folyamat típusa, illetve, hogy service-ként szeretnénk-e majd használni.



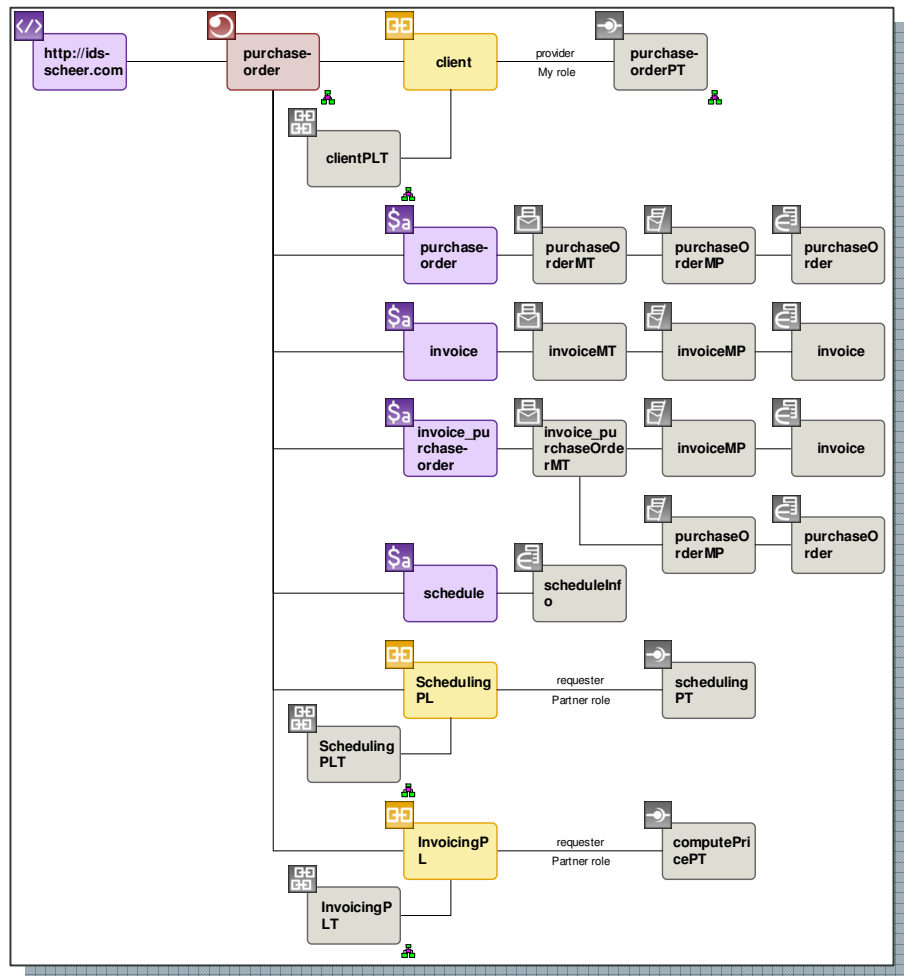
66. ábra A megrendelés szolgáltatás-orientált folyamatból generált BPEL folyamat

A BPEL folyamatba események nem kerültek leképezésre. A start és end eseményekből receive, illetve reply generálódik. A folyamat kezdetéhez és végéhez új objektum készült. A BPEL allokációs diagramokba EPC objektumok, például a klaszterek és a service-k kerülnek.



67. ábra A calculate price allokációs diagram

Az adat klaszterekből változók lettek a leképezés során (purchase order, invoice).



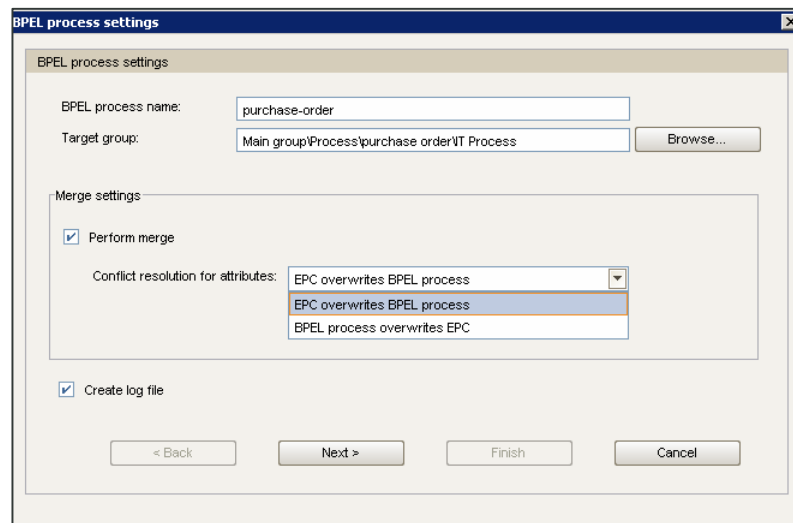
68. ábra A megrendelés BPEL allokációs diagramja

Az allocation diagram részletes információkat tartalmaz a folyamat BPEL elemeiről. A változók közvetítik az üzeneteket a szolgáltatások és az aktivitások között.

5. IT folyamatok készítése

Ebben a lépésben történik az automatikusan létrehozott folyamatok végrehajtó platform specifikus átalakítása. Ki kell dolgozni azon BPEL tevékenységek részleteit, amelyek nem határozhatóak meg az EPC -> BPEL transzformáció által. A platform specifikus átalakítások manuális módon végezhetők.

Figyelembe kell venni, hogy az üzleti folyamatok változhatnak. Lehetőség van az EPC -> BPEL transzformáció ismételt elvégzésére (Perform merge), ami magába foglalja a megváltozott elemeket és hagyja azokat, amikben nem történt változás, anélkül, hogy törölné őket. Ezzel a lehetőséggel biztosítható, hogy az üzleti folyamat az idővel ne szakadjon el az IT folyamattól, ami lényeges szempont a modellvezérelt alkalmazásfejlesztés során.



69. ábra A folyamat változások követése

6. IT folyamatok exportálása és beillesztése

A végrehajtható kód generálása során a folyamat strukturális aspektusait BPEL kódként, a service aspektusait WSDL kódként exportáljuk.

A megrendelés BPEL kódja:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Generated by the ARIS SOA Designer, IDS Scheer AG. All rights reserved.
www.ids-scheer.com-->
<process xmlns="http://schemas.xmlsoap.org/ws/2007/09/business-process/"
xmlns:tns="http://ids-scheer.com"
```

```

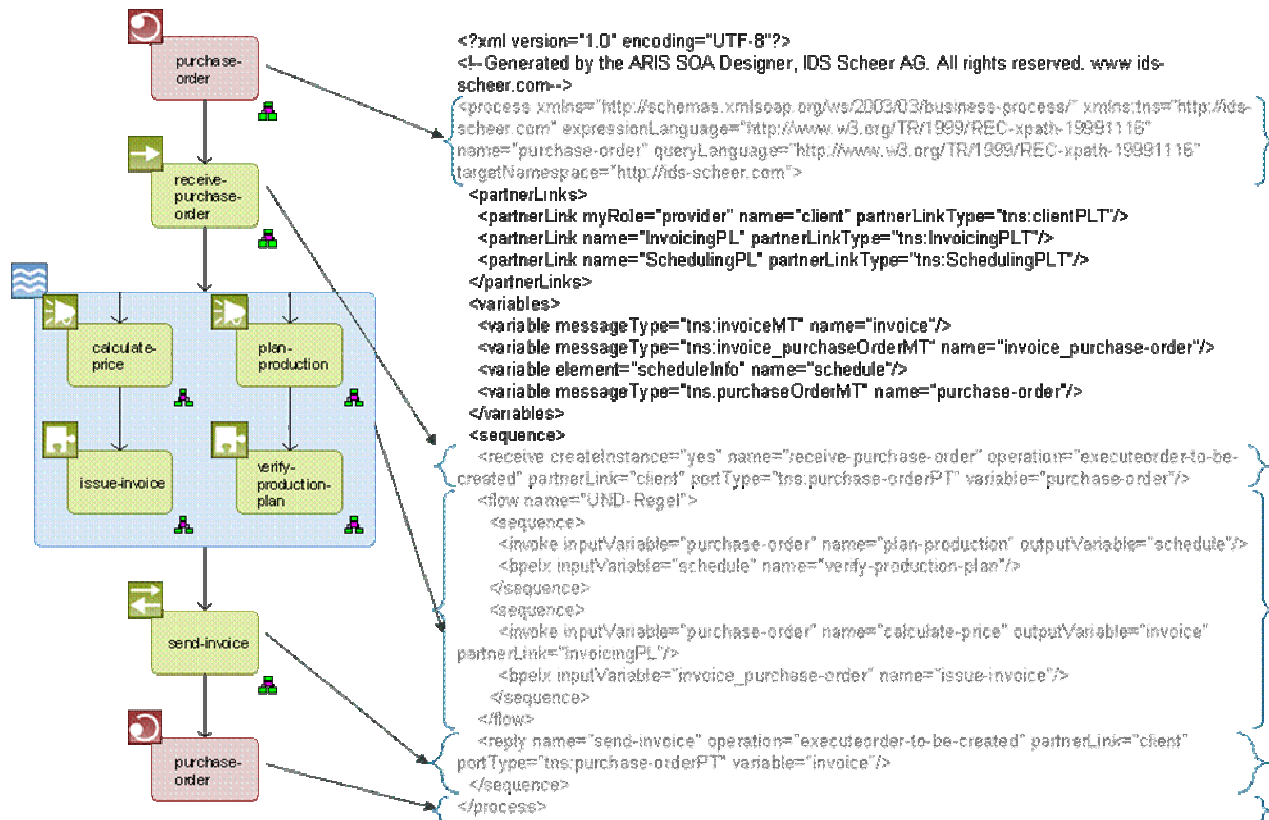
expressionLanguage="http://www.w3.org/TR/1999/REC-xpath-19991116"
name="purchase-order" queryLanguage="http://www.w3.org/TR/1999/REC-xpath-
19991116" targetNamespace="http://ids-scheer.com">
  <partnerLinks>
    <partnerLink myRole="provider" name="client" partnerLinkType="tns:clientPLT"/>
    <partnerLink name="InvoicingPL" partnerLinkType="tns:InvoicingPLT"/>
    <partnerLink name="SchedulingPL" partnerLinkType="tns:SchedulingPLT"/>
  </partnerLinks>
  <variables>
    <variable messageType="tns:invoiceMT" name="invoice"/>
    <variable messageType="tns:invoice_purchaseOrderMT" name="invoice_purchase-
order"/>
    <variable element="scheduleInfo" name="schedule"/>
    <variable messageType="tns:purchaseOrderMT" name="purchase-order"/>
  </variables>
  <sequence>
    <receive createInstance="yes" name="receive-purchase-order"
operation="executeorder-to-be-created" partnerLink="client" portType="tns:purchase-
orderPT" variable="purchase-order"/>
    <flow name="UND-Regel">
      <sequence>
        <invoke inputVariable="purchase-order" name="plan-production"
outputVariable="schedule"/>
        <bpelx inputVariable="schedule" name="verify-production-plan"/>
      </sequence>
      <sequence>
        <invoke inputVariable="purchase-order" name="calculate-price"
outputVariable="invoice" partnerLink="InvoicingPL"/>
        <bpelx inputVariable="invoice_purchase-order" name="issue-invoice"/>
      </sequence>
    </flow>

```

```

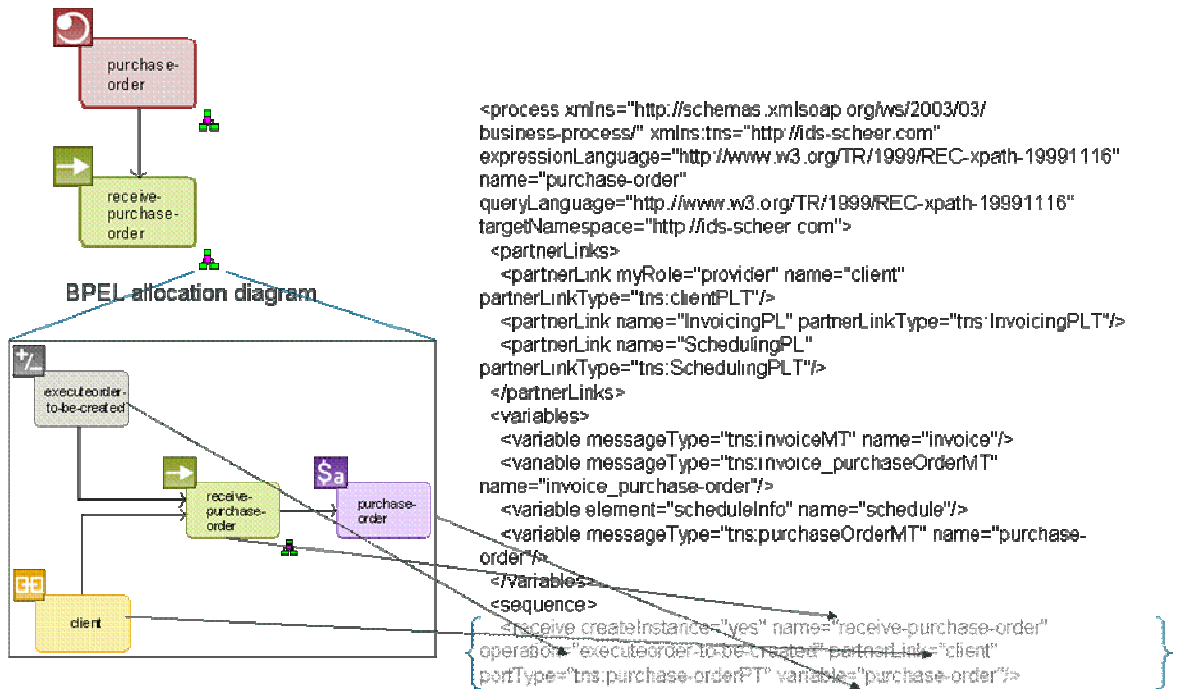
    <reply name="send-invoice" operation="executeorder-to-be-created"
    partnerLink="client" portType="tns:purchase-orderPT" variable="invoice"/>
  </sequence>
</process>

```

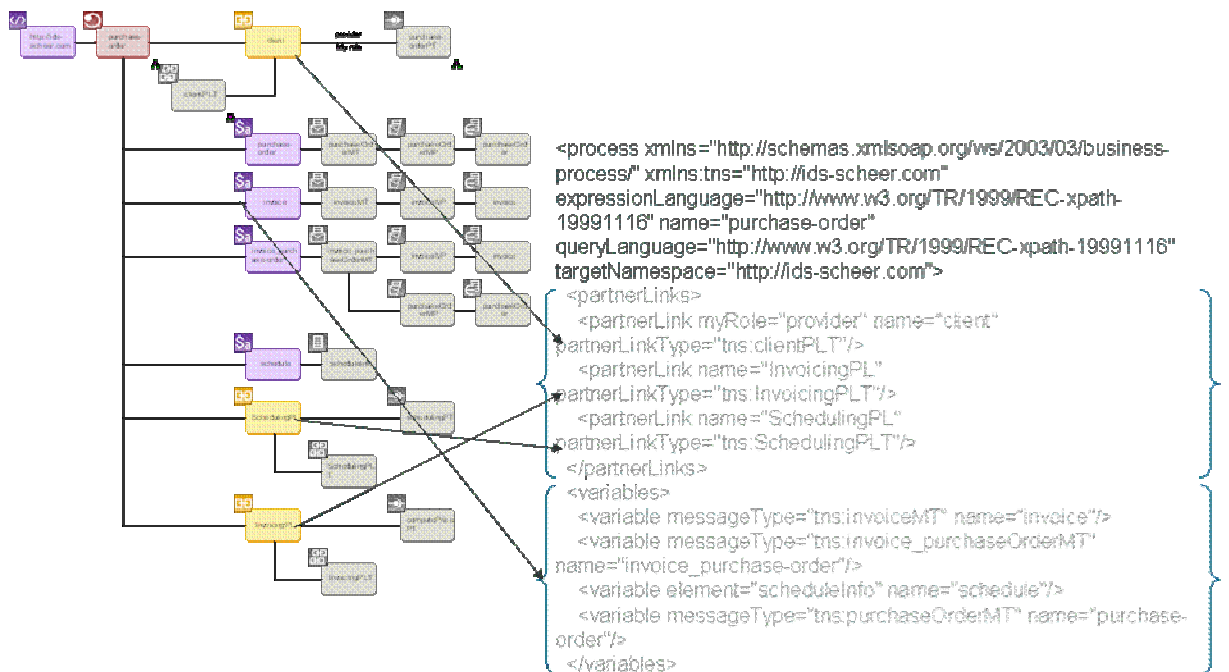


70. ábra A modell és a BPEL kód megfeleltetése

A BPEL kód tartalmazza az allokkációs diagramok információit is:



71. ábra A receive purchase order allokációs diagramjának részletei a megrendelés folyamat BPEL kódjában



72. ábra A megrendelés BPEL alokációs diagram részletei a BPEL kódban

A változók típusa vagy XML séma alapú elem vagy típus, vagy egy WSDL üzenet típus.

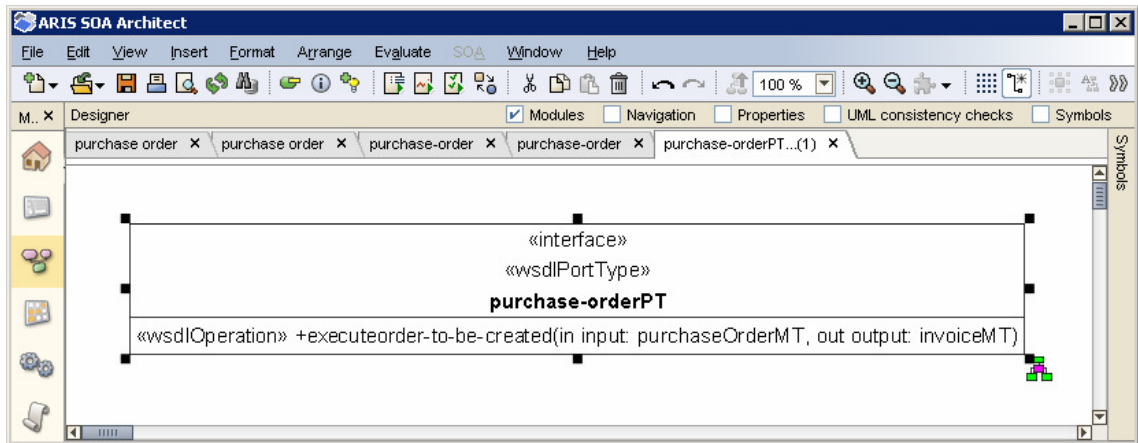
A megrendelés folyamat, mint service, WSDL file:

```

<portType name="purchase-orderPT">
  <operation name="executeorder-to-be-created">
    <input message="tns:purchaseOrderMT" name="executeorder-to-be-created_Request"/>
    <output message="tns:invoiceMT" name="executeorder-to-be-created_Response"/>
  </operation>
</portType>
<plnk:partnerLinkType name="clientPLT">
  <plnk:role name="provider">
    <plnk:portType name="tns:purchase-orderPT"/>
  </plnk:role>
</plnk:partnerLinkType>

```

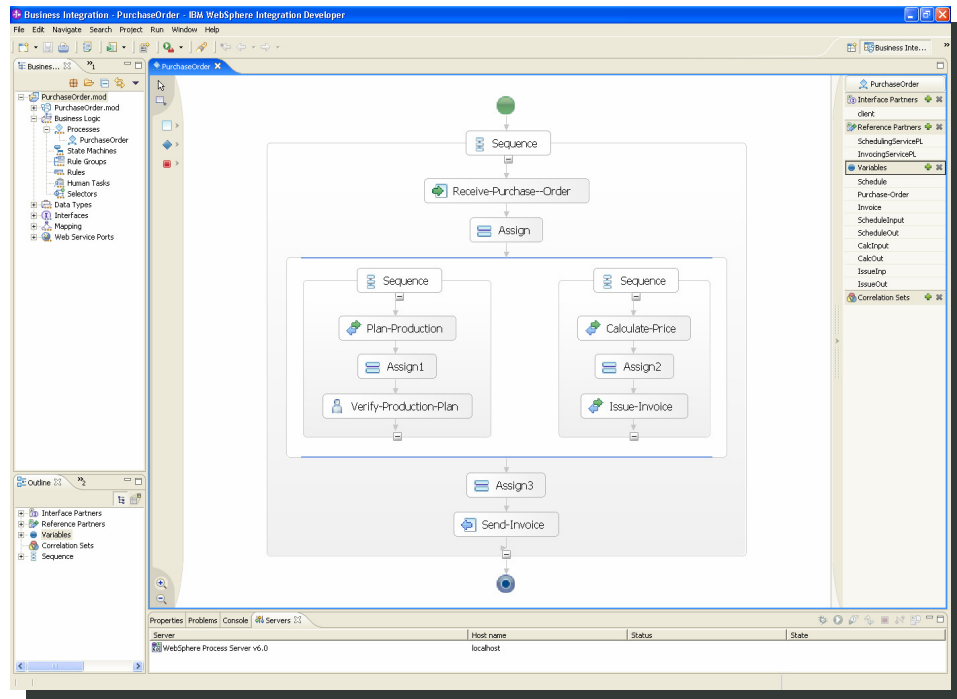
Azt, hogy honnan vesszük igénybe a szolgáltatást, nincs a folyamatba kódolva. Ha változik a szolgáltatás az a folyamatot nem érinti. A WSDL ben a partner link hozzárendeléssel hozzuk létre a laza csatolást.



73. ábra A megrendelés szolgáltatás UML class diagramja

7. IT folyamatok importálása

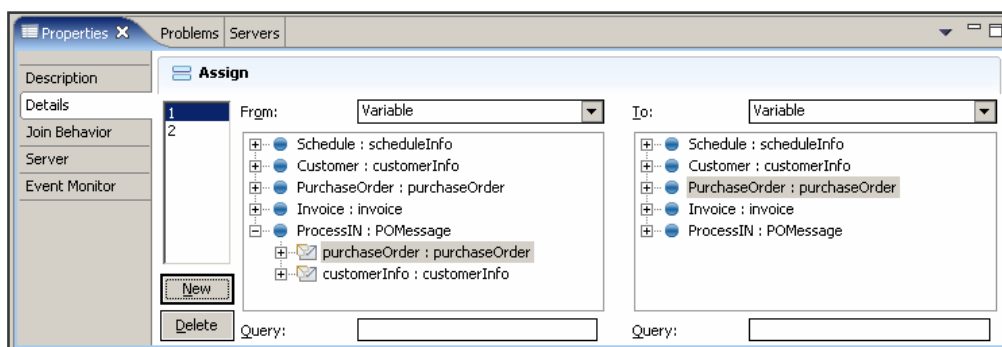
Ebben a lépésben történik a transzfer fájlok generálása ARIS platformon a végrehajtó platform számára. A validált BPEL kódokat fogadni tudják a futtató platformok.



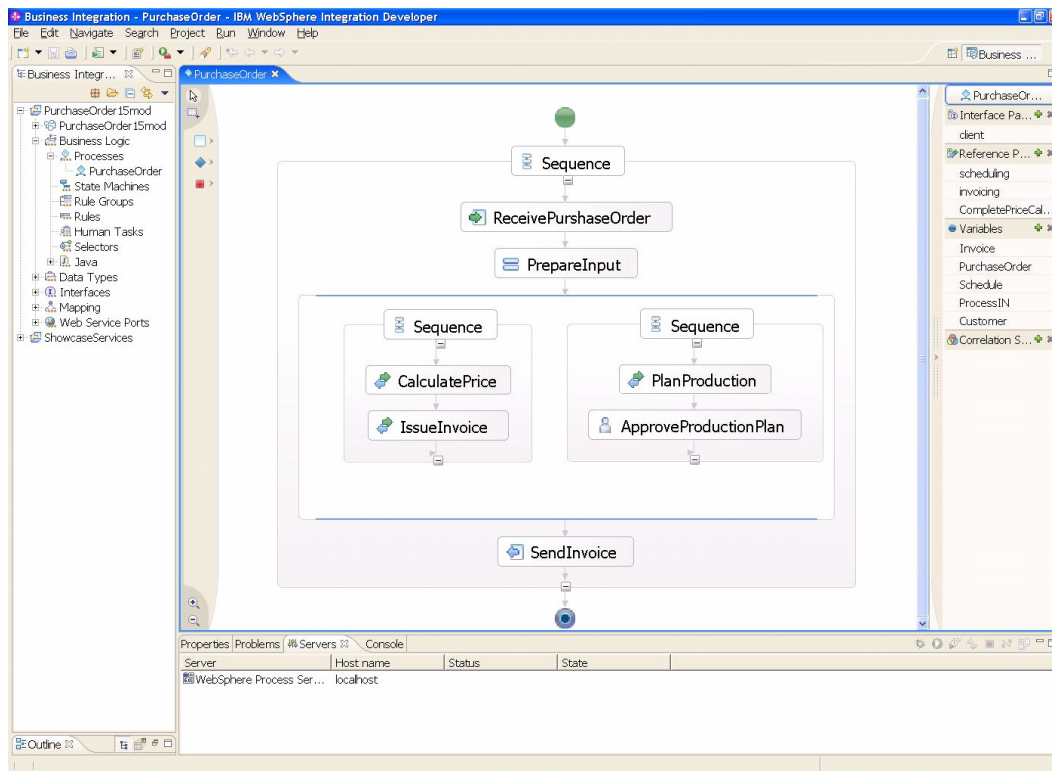
74. ábra A megrendelés folyamat BPEL modellje az IBM Websphere Integration Developer-ben

8. IT folyamatok részletes implementálása

Ebben a lépésben kell az olyan platform részeket hozzá adni a végrehajtó platformhoz, amiket nem lehet ARIS-ban létrehozni. Például az aktivitások platformfüggő részletes implementálása, tulajdonságaik beállítása, új adatváltozók létrehozása.



75. ábra Az input adatokat kezelő aktivitás implementálása



76. ábra A megrendelés folyamat BPEL modellje a változtatás után

9. Új service-k implementálása

Ebben a lépésben történik az új szolgáltatások létrehozása és implementálása a végrehajtó platformon.

10. Tesztelés, telepítés, hibajavítás, végrehajtás

Az utolsó fázis a tesztelés, telepítés, hibajavítás, végrehajtás a végrehajtó platformon. Ezeket a lépéseket a megrendelés folyamattal az IBM Webspher környezetben már bemutattam a 2. tézis igazolásánál.

A fenti levezetéssel igazoltam, hogy egy új üzleti kihívásra az általam javasolt folyamatmenedzsment módszertannal és eszköztárral támogatott, az általam javasolt módszertannal és eszköztárral kialakított ERP és egyéb alkalmazásokat implementáló SOA környezetben, modelltranszformációk felhasználásával gyors és költséghatékony megoldás készíthető, amely mind stratégiai, mind magas absztrakciós szintű üzleti

modellek, mind pedig az implementáció szintjén illeszkedik a szervezet üzleti integrációs architektúrájába.

III.2. A kutatás folytatása

A kutatási eredmények felhasználása és továbbfejlesztése céljából az IBM Corporation Magyarországi Kft. és az IDS Magyarország Kft. együttműködésével 2008. szeptemberétől kutatási projektet indítottunk az üzletvezérelt SOA környezet kialakítása, tanulmányozása és továbbfejlesztése témában.

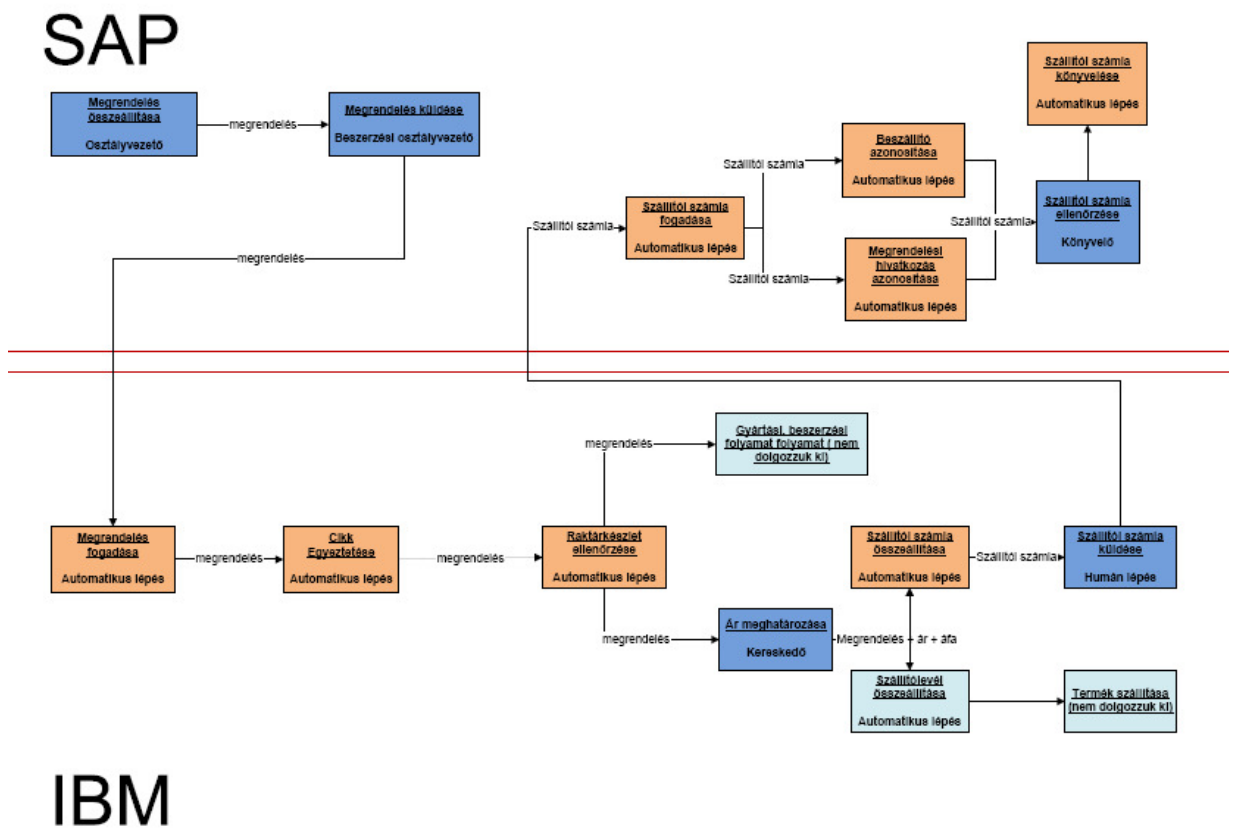
A kutatás célja üzleti modellekből végrehajtható folyamatok leképezési lehetőségeinek vizsgálata, levezetése, tesztelése IBM Websphere környezetben. Folyamatmenedzsment eszközök és a modellezés kapcsolatának vizsgálatában fókuszálunk a következőkre:

- (1) Innovatív üzleti folyamatok ARIS modelljéből végrehajtható BPEL folyamatok levezetése, a leképezés lehetséges automatizálása, tesztelése és monitorozása az IBM Websphere környezetben.
- (2) Innovatív üzleti folyamatok ARIS modelljének leképezése az IBM modellező eszközeibe. A modell leképezés elemzése, tesztelése.

A projekt megvalósítása során a tézisek igazolásánál ismertetett módszertan alapján a kialakított üzletvezérelt környezetben, a kifejlesztett prototípus folyamataiból kiindulva az elektronikus számlázás megoldását tervezzük.

Kialakítjuk azokat a service-eket, amelyek az üzleti szolgáltatások funkcióit lefedik. A prototípusos fejlesztési módszertan alapján, a projektben skálázható megoldást kívánunk kialakítani, amely a számlázás core funkcióiból kiindulva bővül, egyrészt a kapcsolódó üzleti folyamatokhoz (például pénzügyi teljesítés, stb.), másrészt a vállalatközi és hatósági információ- kapcsolatba épül be.

A projekt következő fázisában az elektronikus számlázás üzleti folyamat megvalósítását kiterjesztjük back office, legacy rendszerrel való integráció irányába. A B2B integráció az IBM Websphere és a SAP Netweaver platformokat köti össze.



77. ábra A tervezett pilot rendszer VISIO modellje

A projekt szakmai irányítójaként lehetőségem nyílik arra, hogy szakszeminarista diákjaim közül néhányat bevonjak a projekt egyes részfeladatainak megvalósításába. A kutatás eredményeit pedig az MSC képzésünk tantárgyaiba szeretném majd beépíteni.

IV. ÖSSZEFOGLALÁS

Központi kutatási kérdésem volt, hogy mely módszertan lesz a legalkalmasabb az ERP-t alkalmazó szervezetek integrációs problémáinak a megoldásához, milyen fontos jellemzőkkel rendelkezik, hogyan alkalmazható az adott területeken és hogyan fejleszhető tovább.

A sokáig a vállalatok integrációs problémáinak teljes körű megoldásának tartott ERP rendszerek a gazdag funkcionalitás és rugalmasság ellenére sem minden esetben képesek a vállalatok egyedi, teljesen specifikus igényeit kielégíteni. Előfordulhat, hogy egy üzleti folyamatnak a vállalat és a szoftver által megvalósított módja között áthidalhatatlan eltérések vannak, esetleg a rendszer egyáltalán nem támogat bizonyos tranzakciókat.

Azokban az esetekben is, amikor a bevezetéskor a szoftver teljes körűen támogatja a szervezet funkcionalitását, struktúráját, kultúráját, döntéshozatali módját, az ERP nem tudja dinamikusan követni a szervezet növekedési stratégiáját és az újabb és újabb szervezeten átnyúló kapcsolatokat. A gyorsan növekvő, turbulens környezetben működő, ezért szervezeti stratégiáját gyakran átalakító agilitásra törő vállalat által általában nem alkalmazható.

A dolgozatban rámutattam arra, hogy azok a problémák, amelyekre az integrált vállalatirányítási rendszerek megoldást kínálnak, más módon ma már hatékonyabban oldhatóak meg.

A dolgozat egyik célja volt feltárni, hogy milyen fejlődési modell vázolható fel az együttműködő vállalatok, vállalat-hálózatok sokrétű tevékenységeit segítő integrált vállalatirányítási rendszerek esetében.

A kutatás során közelebbről vizsgáltam, hogy melyek azok az információtechnológiai megoldások, amelyek a leghatékonyabban tudják támogatni a szervezetek működtetését, együttműködését, és a támogatás típusait.

Kutatásom további célja volt feltárni, hogy az említett technológiai lehetőségek az új üzleti modellek elterjedését mennyiben segítik elő, illetve milyen átalakulás előtt áll a számítógépes alkalmazások piaca az összekapcsolhatóság egyre fejlettebb szintre kerülése miatt. Az alkalmazások külső forrásokból, modulszerűen történő igénybevételének lehetősége nem csak a szoftverpiac teljes átalakulását hozza magával, hanem azt is jelenti, hogy a gazdaság szinte minden szereplőjének át kell értékelnie az információtechnológiával kapcsolatos elgondolásait.

A kutatás elméleti háttérének bemutatása során kitértem több tudományterülethez is kapcsolódó alapfogalmakra és elméletekre, a technológia szerepére, olyan új kutatási területekre, mint a szemantikus web és folyamatmenedzsment, SOA és modellvezérelt architektúra, valamint a leggyakrabban használatos módszertanokra.

A dolgozat rámutat arra is, hogy az új területek megfelelő absztrakciós szintű üzleti modelljei hogyan segítik az új terület menedzselésének integrációját a szervezet meglévő működésébe, és az azt támogató informatikai alkalmazásrendszerébe.

A dolgozatban a következő fontosabb kutatási kérdésekkel foglalkoztam:

- hogyan modellezzünk egy eddig informatikailag nem támogatott területet annak érdekében, hogy azt a szervezetek integrálni tudják alkalmazásaikba;
- melyek a leggyakrabban hivatkozott modellezési eljárások;
- milyen fejlesztési módszertanok léteznek a szakirodalomban;
- hogyan kellene és lehetne ezeket a módszertanokat kiterjeszteni, módosítani, hogy a dinamikusan belépő új területekre alkalmazhatóak legyenek,
- milyen implementációs lehetőségek vannak az fejlesztésre vonatkozóan;
- milyen támogatást nyújt a modell egy rendszerintegrációs folyamatban?

A modellezési eljárások vizsgálata kiterjedt arra is, hogy az egyes részterületeken kialakított modellek között milyen átjárhatóság állapítható meg. A modellezés az integrált rendszer komponensek közös vonásainak megragadását az erre alkalmas absztrakciós szinten valósította meg. A modell alapú megközelítés ad támogatást olyan magas szintű leírásra, metamodellek készítésére, amelyek lehetővé teszik az egyes területeken történő újrafelhasználást. A dolgozatban követett megközelítés, problémakezelés egyediségét az alkalmazott paradigmarendszer és eszközökörnyezet biztosította.

Megvizsgáltam, hogy mely módszertan a legalkalmasabb a probléma megoldásához, milyen fontos jellemzőkkel rendelkezik, hogyan alkalmazható az adott területeken és hogyan fejleszthető tovább, milyen integrációs lehetőségei vannak.

Az általam megfogalmazott kutatási feladat gyakorlati megvalósításához áttekintést adtam a támogató megoldásokról és kiválasztottam a fejlesztésben felhasználható alkalmazásokat.

A kutatás során a témakörre vonatkozó elméleti megközelítésekől indultam ki, a modellezési eljárások vizsgálatán keresztül eljutottam az módszertanok értékeléséig és a saját változat kialakításáig, amelyet a gyakorlatban is alkalmaztam. A kutatási kérdések zárásaként megvizsgáltam a kialakított modell további felhasználási lehetőségeit.

A kutatás legfontosabb eredményei a módszertan alkalmazása a kiválasztott területre, a modell kialakítása, integrációs lehetőségek feltárása, prototípus kialakítása az implementációs környezetben. Újszerű megoldás a választott területen a modellalapú megközelítés alkalmazása is.

A dolgozatban kialakított megvalósítási modell gyakorlati hasznosítási lehetőségei:

- modellként funkcionálhat a vállalati gyakorlatban,
- továbbfejleszthető a területet teljes mélységében leíró, „testre szabott” prototípussá,
- konvertálható egyéb rendszerekbe, így támogatva az újrahasznosíthatóságot,
- alapja lehet a területre vonatkozó rendszerfejlesztési projekteknek,

- támogathatja a rendszerspecifikációk elkészítését,
- elősegítheti különböző granularitású architektúrák összehasonlíthatóságát.

A fentiek alapján az eredmények széles körűen hasznosíthatók a gyakorlatban. Az elméleti területen kiemelem azt, hogy a modell lehetővé teszi az alkalmazások szemantikus együttműködését, azaz a rendszerintegrációban játszott szerepét.

A részletesen bemutatott üzleti integrációs modell alapján a hatékony integráció megvalósításához minden szint minden elemére figyelmet fordítottam – horizontálisan és vertikálisan is. A rétegek közötti integrációt is megvalósítottam. A kutatás során a témakörre vonatkozó elméleti megközelítések és az üzleti integrációs modell konvergenciájaként megállapítható, hogy az üzleti folyamatok alapvető, létfontosságú kapcsolatot jelentenek a technikai és szervezeti infrastruktúra között. Dolgozatomban az üzleti folyamatoknak kulcsszerepe van az üzleti integráció üzletvezérelt módon való megvalósításában.

A kulcsfolyamatok kialakításának és az erre alapozott működés megteremtésének képessége ma már minden szervezetben alapkövetelmény. A folyamatelvű szervezet működése során gyors és hatékony üzleti folyamatokba építi specializált funkcionális szakértelmét. Az információs technológia korszerű alkalmazásával a vevői/szállítói kapcsolatokra figyelve integrálja a beszerzést, az előállítást és az elosztást, s mindezt úgy, hogy a vállalat működését alapvetően a vevői megrendelések mozgatják. Megtanul egyedi, a fogyasztók egyéni igényeinek megfelelő termékeket és szolgáltatásokat nyújtani változatos fogyasztói szegmenseik számára a szokásos széles választékkal és az alacsony termelési mennyiséggel járó felár elkerülésével. A globalizáció vállalatainak az új termékek és szolgáltatások kifejlesztéséhez szükséges nagy beruházások következtében gyakran világméretű piacra van szükségük a megfelelő megtérülés eléréséhez így egyesíteniük kell a globális működés nyújtotta hatékonysági és versenyelőnyöket a helyi fogyasztói igényekre való érzékenységgel. A termék-életciklusok rövidülésével a termék-életgörbe egy szakaszában elért versenyelőny nem garantálja ennek megtartását a technológia következő fejlettségi szintjén is. A

fogyasztók jövőbeli igényeinek előrejelzése, az új gyártási technológiák gyors alkalmazásának képessége, a folyamatok, termékek és szolgáltatások állandó fejlesztése – az innováció kritikus a vállalati siker szempontjából.

A kiválasztott modellezési eljárásokat a gyakorlatban alkalmaztam. Alapvető kérdés volt, hogy melyik az az absztrakciós szint, amelynek segítségével az eltérő területek közös vonásai megragadhatóak. A modell alapú megközelítés adott támogatást olyan magas szintű leírásra (meta-modellek készítésére), amelyek lehetővé tették az adott területek szemantikus integrálását. Így az általam választott folyamat és modellvezérelt megközelítés gyakorlati megvalósítása prototípusként egy gyakorlati igazolását jelenti ezen irányú alkalmazhatóságának.

Az ARIS hatékony koncepcionális keretrendszert biztosított a megvalósítás során. Az ARIS modellek alapul szolgáltak több részletesebb modellnek, amelyek jelentős hatásúak az automatizálás és interoperabilitás fokára, mivel közös halmazokat definiáltak a különböző forrásból származó adatok számára.

ERP alapú környezetekben a folyamatmodellek által leírt tények nagy része használható az ERP és más alkalmazások integrációját megvalósító rendszer koncepcionális modelljéhez. Az erőforrások és a szervezeti struktúrák tárolhatók már az ERP rendszer relációs adatbázisában, és fordítva, a tranzakciós és customizing adatok, referencia modellek felhasználhatóak onnan. Ehhez megfelelő leképezés szükséges az említett magas absztrakciós szintű modellek és az ERP rendszer adatbázis struktúrák között.

Dolgozatomban egy fő hipotézist, igazolásához pedig három segéd hipotézist fogalmaztam meg.

A Fő hipotézis:

Az ERP rendszereket alkalmazó szervezeteket körülvevő konglomerátumban az integrációs problémák megoldásának kulcsa már középtávon is a korszerű

*alkalmazásfejlesztési paradigmák és módszerek – BPMS, SOA, MDA – kollaboratív használatával elérhető **szinergiák** kiaknázása.*

Az Első segédhipotézis:

Az ARIS módszertan és eszköztár felhasználásával megvalósítható az ERP rendszert alkalmazó szervezet üzleti integrációs modellje.

A Második segédhipotézis:

Egy szervezet integrációs problémái a BPM, a SOA és a modell-vezéreltség elveinek kollaboratív alkalmazásával üzletvezérelt módon megoldhatóak.

A Harmadik segédhipotézis:

Az ERP rendszert alkalmazó szervezet a bevezetés során kialakított, majd működése során optimalizált BPMS folyamatmodelljeit újrafelhasználva gyorsan és költséghatékonyan képes a környezet változásait integrált informatikai rendszereivel és alkalmazásaival is követni.

Az Első segédhipotézis igazolása

A hipotézist a szervezetek üzleti integrációs modellje és az ARIS módszertan és eszköztár felhasználásával bizonyítottam. Az üzleti integrációs modell megvalósításához nyújtott támogatást az ARIS különböző nézeteiben és szintjein alkalmazható diagramtípusoknak az integrációs modell rétegeihez rendelésével adtam meg.

Az üzleti integrációs modell megvalósításához a következő okok miatt választottam az ARIS módszertant:

- Az ARIS – „Integrált Információrendszerek Architektúrája” a vállalati információkat integráltan összefogó és kezelő üzleti folyamatok támogatását szolgáló informatikai rendszerek felépítését írja le.

- A folyamatszervezési világpiac vezető modellező eszköze.
- BPR céllal hazánkban is sok szervezet alkalmazza.
- Piacvezető ERP rendszerek (pl. SAP) referenciamodelljei rendelkezésre állnak.
- Az ARIS koncepció alapelvei (szétválasztás-, leíró szintek elve) lehetővé teszik az üzleti integrációs modell horizontális és vertikális rétegei közötti integráció megvalósítását.
- A vállalati folyamatstruktúra kialakításához „fönről-le” módszert használ.

A Második segédhipotézis igazolása:

Egy alkalmazás SOA alapú kifejlesztésének első lépése az üzleti folyamat modelljének megvalósítása. A SOA megfelelő kialakításának szempontjából azért fontos a BPM, hogy a működő folyamatokat optimalizálja, ahol szükséges felülvizsgálja, annak érdekében, hogy az újonnan kialakított IT rendszer lehető legjobb teljesítményét garantálni lehessen. Legtöbb esetben a meglévő folyamatok modellezéséből indulnak ki, majd ezt követi az optimalizálás. Az üzlet és az informatika közötti határt technológiai oldalon az üzleti folyamatok kezelését támogató BPM eszközök, a megvalósítás során pedig a különböző fejlesztési és tervezési módszertanok hidalják át. A SOA + BPM projekteket támogató kollaboratív módszertan lényege, hogy a különböző fázisok szereplői szorosan együttműködnek egymással és folyamatos a visszacsatolás. Az üzleti szakértők feladata a folyamatok modellezése, optimalizálása, monitorozása, az IT alkalmazottak feladata pedig az optimalizált folyamatokat a valós IT működésbe átültetni.

Egy konkrét üzleti folyamat megvalósításán keresztül mutattam be, hogy hogyan lehet szolgáltatás-orientált architektúrát kialakítani és egy komplex integrált üzleti folyamatot magas absztrakciós szintről a modell vezérelt módszertan szabályai szerint végrehajtható folyamattá transzformálni. Az alkalmazás szolgáltatás-orientált módon való kifejlesztéséhez az IBM WebSphere termékcsalád eszközeit és modell-transzformációt használtam.

A következő okok miatt alkalmaztam az IBM WebSphere eszközeit:

Az IBM WebSphere termékcsalád piacvezető, kiforrott, megbízható, testre szabható megoldásokat kínál a folyamatmodellezés, a SOA, az integráció megvalósítására. Beépített, validált modell transzformációs eszközt tartalmaz a BPMN -> BPEL leképezés megvalósítására.

Egy egyszerűen átlátható, gyakran előforduló és adott vállalat esetében gyakran végrehajtandó üzleti folyamatot választottam az illusztrálásra, a megrendelést. Az, hogy sűrűn alkalmazzák az adott eljárást, indokolja a folyamat automatizálásának igényét. A folyamat tartalmaz emberi szereplőket, automatikus feladatokat és külső szolgáltatásokat egyaránt, így lehetőséget nyújt a különböző szintű integrációs szereplők folyamatba kapcsolásának bemutatására.

Az alkalmazás megvalósításával tehát – melyben egy megrendelési folyamatot modelleztem a WebSphere Modeler-ben, majd implementáltam a WebSphere Integration Developer-ben - bemutattam, hogy hogyan hozható létre üzletvezérelt módon egy integrált alkalmazás szolgáltatás-orientált környezetben.

A Harmadik segédhipotézis igazolása:

A harmadik hipotézis bizonyításánál olyan vállalatot vizsgáltam, amely ARIS módszertant alkalmazott a különböző alkalmazásai és az ERP rendszerének a bevezetése, üzleti architektúrájának kialakítása során, és a folyamat modelljeit a bevezetés óta karbantartja a közös adatbázisban. Az ARIS piaci helyzete miatt ezzel igen sok vállalatot bevettem a „gondolatkísérletbe” még hazai vonatkozásban is.

Feltettem továbbá, hogy a vállalat IBM Websphere platformon működteti különböző alkalmazásait. Ezen „jövőbe mutató” feltételezéssel jelentősen leszűkítettem ugyan a

vizsgált vállalatokat, de az IBM Websphere piaci eredményei alapján még mindig jelentős méretű a kísérleti halmaz.

Olyan változást tekintettem a vállalat életében, ami stratégiai szempontból gyors reagálást igényel. Egy olyan esetet vizsgáltam, ami manapság igen gyakori, tehát a „kísérleti halmaz” biztosan nem üres.

A vállalat termékeit online árusítja egy katalógus segítségével. Mióta az online rendelések száma naponta nő és a hosszú feldolgozási idő már nem elfogadható, a cél az lett, hogy az alkatrészek rendelését a lehető legmagasabb szinten automatizálják service-k segítségével.

A tézis igazolásához az ARIS for SOA eszközt, és a 10 lépéses módszertani eljárást alkalmaztam.

A levezetéssel igazoltam, hogy egy új üzleti kihívásra az általam javasolt folyamatmenedzsment módszertannal és eszköztárral támogatott, az általam javasolt módszertannal és eszköztárral kialakított ERP és egyéb alkalmazásokat implementáló SOA környezetben, modell-transzformációk felhasználásával gyors és költséghatékony megoldás készíthető, amely mind stratégiai, mind magas absztrakciós szintű üzleti modellek, mind pedig az implementáció szintjén illeszkedik a szervezet üzleti integrációs architektúrájába.

V. IRODALOMJEGYZÉK

[Aalst, Pesic, 2006] van der Aalst, W.M.P.; Pesic, M.: Specifying, Discovering, and Monitoring Service Flows: Making Web Services Process-Aware. BPM Center Technical Report, No. BPM-06-09, 2006.

[Antal-Mokos, Balaton, Drótos, Tari ,1997] Antal-Mokos, Z.,Balaton, K.,Drótos, Gy.,Tari, E.: Stratégia és szervezet Közgazdasági és Jogi Könyvkiadó, Budapest, 1997.

[Babbie, 1996] Babbie, E. (1995): A társadalomtudományi kutatás gyakorlata Balassi Kiadó, Budapest

[Balaton, Dobák, 1991] Balaton, K., Dobák, M: Mennyiségi és minőségi módszerek az empirikus szervezetkutatásban, 1991.

[Bertrand, 2007]Bertrand Portier: Service, architecture, governance, and business terms, May 2007, <http://www.ibm.com/developerworks/webservices/library/ws-soa-term1/>

[Bieberstein et al. ,2006] Bieberstein, N. Bose, S. Fiammante, M. Jones, K. Shah, R., Service-Oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap. Pearson Education, 2006.

[Booch, 1991] Booch, G.: Object-Oriented Design, Benjamin/Cummings, Redwood City, Calif., 1991.

[Bruijn, Jos et al, 2005] de Bruijn, Jos et al.: D16.1v0.21 The Web Service Modeling Language WSML. WSML Final Draft. <http://www.wsmo.org/TR/d16/d16.1/v0.21/>, retrieved November 7, 2005.

[Chandra, 2000] Chandra., J., et al.: Information Systems Frontiers, Communications of the ACM, January 2000.

[McCarthy, 1999] McCarthy, William és Geerts, Guido. : An Accounting Object Infrastructure for Enterprise Models, IEE Intelligent Systems, 1999, www.msu.edu/user/mccarth4.

[McCarthy, 2000] McCarthy, William és Geerts, Guido. 2000. The Ontological Foundation of REA Enterprise Information Systems, www.msu.edu/user/mccarth4/alabama.doc.

[Chalmeta, Campos, Grangel, 2001] Chalmeta, R., C. Campos, R. Grangel. (2001) References Architectures for Enterprise Integration, Journal of System and Software, 57(3), 175-91.

[Chikán Attila, 1997] Chikán Attila: Vállalatgazdaságtan. Aula Kiadó. Budapest. 1997.

[Claybrook, 1992] B. Claybrook: OLTP Online Transaction Processing, John Wiley & Sons, 1992.

[Coad, 1991] Coad, P., Yourdon, E.: Object-Oriented Analysis, Second Edition, Yourdon Press, Prentice Hall, Englewood Cliffs, New Jersey, 1991. ISBN 0-13-629981-4

[Davenport, 1998] Davenport, T. H., : Putting the Enterprise into the Enterprise System, Harvard Business Review, 1998.

[Dellarocas, 1996] Dellarocas , C. (1996) A Coordination Perspective on Software Architecture: Towards a Design Handbook for Integrating Software Components. MIT, Cambridge.

[Dietz, 2006] Dietz, Jan L. G.: Enterprise Ontology. Springer, Berlin / Heidelberg 2006.
Scheer, A.-W.: ARIS - Vom Geschäftsprozess zum Anwendungssystem. 3rd. Aufl., Springer, Berlin etc. 1998.

[Dobák, 1999] Dobák Miklós: Szervezeti formák és vezetés, KJK 1999.

[Dobay, 1997] Dobay Péter (1997): Vállalati információmenedzsment. Nemzeti Tankönyvkiadó, Budapest.

[Dutta, 1997] Dutta, S., et al.: Designing Management Support Systems, Communications of the ACM, 1997.

[Edvard A. Stohr , Jeffrey V. Nickerson , 2003] Edvard A. Stohr , Jeffrey V. Nickerson (2003) Intra Enterprise Integration. Competing in the information Age. Second edition. Oxford University Press

[Eisenhardt, 1989] Eisenhardt, K. M.: Building Theories from Case Study Research. Academy of Management Review, 1989.

[EMA, 2006]EMA, SOA Market and Products 2006: Current State, Future Directions, Enterprise Management Associates, Ápr. 2006. Letöltve: 2008. április 8.
http://www.optier.com/downloads/analysts/EMA%20-%20SOA%202%20-%20Market%20and%20Products%202006_Apr-06.pdf

[Esteves, Pastor, 2001] Esteves, J., & J. Pastor. (2001) Enterprise Resource Planning Systems Research: An Annotated Bibliography, Communications of AIS, 7.

[Fenn, 2007] Fenn, J., Gartner's Hype Cycle Special Report for 2007, Gartner, Aug. 2007.

Letöltve: 2008. április 8.

<http://www.gartner.com/DisplayDocument?id=511189>

[Fensel, Bussler, 2002] Fensel, Dieter; Bussler, Chris: The Web Service Modeling Framework WSMF. In: Electronic Commerce Research and Applications 1 (2002) 2, pp. 113-137.

[Fox, Gruninger, 1998] Fox, Marc S.; Gruninger, Michael: Enterprise Modeling. In: AI Magazine (1998) Fall 1998, pp. 109-121.

[Fox et al., 1998] Fox, Marc S. et al.: An Organisation Ontology for Enterprise Modeling. In: M. Prietula; K. Carley; L. Gasser (Hrsg.): Simulating Organizations: Computational Models of Institutions and Groups. AAAI/MIT Press, Menlo Park CA 1998, pp. 131-152.

[Fuchs, 1979] Fuchs, H.: Rendszerszemlélet Szerk: Bleicher, K: A szervezet mint rendszer Közgazdasági és Jogi Könyvkiadó, Budapest, 1979.

[Gray, 2001] P. Gray , J. Byun : Customer Relationship Management, University of California, Irvine, www.crmassist.com/documents, 2001.

[Grochla,1979] Grochla, E: Rendszerszemlélet és szervezettelmélet Szerk: Bleicher, Knut: A szervezet mint rendszer Közgazdasági és Jogi Könyvkiadó, Budapest, 1979.

[Gruninger et al., 2000] Gruninger, Michael et al.: Ontologies to Support Process Integration in Enterprise Engineering. In: Computational & Mathematical Organization Theory 6 (2000) 4, pp. 381-394.

[Guarino, 1998] Guarino, N. (ed.), „Formal Ontology in Information Systems”, Amstardam, Berlin, Oxford: IOS Press. 1998.

[Hart, 1988] Hart, C. : Doing a Literature Review – Releasing the Social Science Research Imagination, Sage Publications, London, 1998.

[Haugen, 2001] Haugen, Robert, REA Ontology as foundation for ebXML metamodel, 2001, <http://www.supplychainlinks.com>.

[Hepp, Martin et al., 2005] Hepp, Martin et al.: Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management. IEEE International Conference on e-Business Engineering (ICEBE 2005).

[High, Kinder, Graham, 2005] High, R. Kinder, S. Graham, S., An Architectural Introduction and Overview, IBM's SOA Foundation, Nov. 2005.
<http://download.boulder.ibm.com/ibmdl/pub/software/dw/webservices/ws-soa-whitepaper.pdf>

[Hollingsworth, 1995]Hollingsworth, D. – The Workflow Management Coalition (1995): *The Workflow Reference Model*.
<http://www.wfmc.org/standards/docs/tc003v11.pdf>,

[Holloway, 1990] S. Holloway: The Distributed Development Environment, Chapman and Hall, 1990.

[Ibbotson, 2001] Ibbotson, J. (Ed.) (2001) XML Protocol Usage Scenarios. W3C.
<http://www.w3.org/TR/2001/WD-xmlp-scenarios-20010217/>

[IBM, 2002a] IBM, IBM Acquires Holosofx to Extend WebSphere Business Integration Software Portfolio, IBM.com, Szept. 2002. Letöltve: 2008. április 13.
<http://www-03.ibm.com/press/us/en/pressrelease/534.wss>

[IBM, 2005a] IBM, IBM Acquires Bowstreet, Inc. to Help Customers Quickly Create Portals that Integrate Existing Business Applications, IBM Press Room, Dec. 2005.
Letöltve: 2008. április 15.
<http://www-03.ibm.com/press/us/en/pressrelease/19064.wss>

[IBM, 2005b] IBM, IBM Acquires DataPower, IBM Press Room, Okt. 2005.
Letöltve: 2008. április 15.
<http://www-03.ibm.com/press/us/en/pressrelease/7934.wss>

[Jensen, Meckling, 1973] Jensen, M. C., W. H.Meckling, (1973) Theory of the Firm: Managerial Behavior, Agency Costs, and Ownership Structure, Journal of Financial Economics, 3(3), 305-60.

[Johnson, 2000] Johnson, R. A.: The Ups and Downs of Object-Oriented Systems Development, Communications of the ACM, 2000.

[Kaplan, Norton 1992] Kaplan, R. S. , D. P. Norton. (1992) The Balance Scorecard: Measures That Drive Performance, Harward Business Review, 70.

[Katz, Kahn, 1986] Katz, D., Kahn, R.L.: A szervezetek és a nyílt-rendszer elmélet Szerk: Kovács S.: Szöveggyűjtemény a szervezetelmélet történetének tanulmányozásához Tankönyvkiadó, Budapest, 1986.

[Kenny, Plummer, 2007] Kenny, L. F. _ Plummer, D. C., Magic Quadrant for Integrated SOA Governance Technology Sets, Gartner RAS Core Research, Dec. 2007. Letöltve: 2008. április 13. http://www.softwareag.com/Corporate/Images/REPRINT-Magic%20SOA%20Governance%20Technology%20Sets,%202007_tcm16-37108.pdf

[Kieser, 1995] Kieser, A. (1995): Szervezetelméletek Aula, Budapest

[Klimkó, 2001] Klimkó Gábor: A SZERVEZETI TUDÁS FELTÉRKÉPEZÉSE, Ph.D. értekezés, 2001.

[Koch, 1999] Koch., C., et al.: The ABCs of ERP, CIO Magazine (cio.com), December 22, 1999.

[Koppányi Mihály, 2002] Koppányi Mihály (szerk.): Mikroökonómia, Budapest, KJK-KERSZÖV Kiadó, 2002.

[Lawler, Howell-Barber, 2007] Lawler, J. P., Howell-Barber, H., Service-Oriented Architecture: SOA Strategy and Methodology and Technology. Auerbach Publications, 2007.

[Leary 1997] O’Leary, D., et al.: Artificial Intelligence and Virtual Organizations, Communications of the ACM, 1997.

[Lederer, 1998] Lederer. A. L., et al.: Using Web-based Information Systems to Enhance Competitiveness, Communications of the ACM, July 1998.

[Lenat, Guha, 1990] Lenat, D. B., and Guha, R. V., „Building large knowledge-based systems. Representation and inference in the Cyc project”, Addison-Wesley, Reading, Massachusetts, 1990.

[Lucey, 1989] Lucey, Terry.: Management Information Systems (8th edition), D.P. Publications Ltd., 1989.

[Magid, 2006] Magid, D., Driving innovation through acquisitions, among other ways (an Interview with Steve Mills), The Venture Capital Group Report, Júl. 2006. Letöltve:2008.április15.http://www304.ibm.com/jct03004c/businesscenter/venturedevelopment/us/en/thoughttemp/!/gcl_xmlid=54809

[Malone et al., 1999] Malone, T.W., K. Crowston, et al. (1999) Tools for inventing Organizations: Toward a Handbook of Organizational Processes. Management Science, 45(39, 425-43.

[Malone, Crowston, 1994] Malone, T.W., K. Crowston, et al. (1999) Tools for inventing Organizations: Toward a Handbook of Organizational Processes. Management Science, 45(39), 425-43.

[Markus, 2000] Markus, M.L. (2000) Paradigm Shifts: E-Business and Business/Systems Integration, Communications of AIS 4(10).

[Markus, Tanis, Fenema, 2000] M. Lynne Markus, Cornelis Tanis, Paul C. van Fenema: Enterprise resource planning: multisite ERP implementations, Communications of the ACM, Volume 43 , Issue 4 (April 2000) , Pages: 42 – 46, ACM Press New York, NY, USA

[Martin et al., 2005] Martin, David et al.: OWL-S 1.0 Release. <http://www.daml.org/services/owl-s/1.0/>, retrieved Nov 30, 2005.

[Miles, Huberman, 1994] Miles, B. M., Huberman, A. M.: Qualitative Data Analysis (2nd ed.). London: Sage Publications, 1994.

[R. Milner, 1999] Robin Milner: Communicating and mobile systems: the pi calculus, ISBN 052164320, Cambridge University Press 1999.

[Nonaka, 1995] Nonaka, I., Takeuchi, H.: The Knowledge-Creating Company, New York: Oxford University Press, 1995.

[Oberle, 2006] Oberle, Daniel: Semantic Management of Middleware. Springer, New York 2006.

[OMG, 2002] MetaObjectFacility(MOF) Specification, Version 1.4, <http://www.omg.org/docs/formal/02-04-03.pdf>

[Palaniswamy, 2000] Palaniswamy, R., and T. Frank: Enhancing Manufacturing Performance with ERP Systems, Information Management Journal, 2000.

[Parr, Shanks, 2000] Parr AN and Shanks G. 2000. A taxonomy of ERP implementation approaches. 33rd Annual Hawaii International Conference on System Sciences, Sprague RHJ (ed) Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Los Alamitos: IEEE Computer Society. ISBN 0-7695-0493-0.

[Peiris et al., 2007] Peiris, C. _ Mulder, D. _ Cicoria, S. _ Bahree, A. _ Pathak, N., Pro WCF: Practical Microsoft SOA Implementation. Apress, 2007.

[Peppers, 1999] Peppers, D., and M. Rogers: Enterprise One to One: Tools for Competing in the Interactive Age. New York: Doubleday, 1999.

[Pine, 1999] Pine, B. J., S. Davis: Mass Customization: The New Frontier in Business Competition.. Boston: Harvard Business Review, 1999.

[Recker et al., 2006] Recker, Jan et al.: How Good Is BPMN Really? Insights From Theory And Practice. 14th European Conference on Information System (ECIS 2006)

[Roman et al., 2005] Roman, Dumitru et al.: D2v1.2 Web Service Modeling Ontology (WSMO). WSMO Final Draft April 13, 2005. <http://www.wsmo.org/TR/d2/v1.2/>, retrieved Nov 30, 2005.

[Rosemann, Green, 2004] Davies, Islay G and Rosemann, Michael and Green, Peter (2004) Exploring proposed ontological issues of ARIS with different categories of modellers. In: Proceedings of the Australasian Conference on Information Systems, Hobart, Australia.

[Sassi, 2007] Sassi, G., SOA e Business Flexibility, IBM, 2007. Letöltve: 2008. április 8.

[Schreiber, 1998] Schreiber, A. Th., Akkermans, J. M., Anjewierden, A. A., de Hoog, R., Shadbolt, N. R., Van de Velde, W., Wieliga, B. J.: Knowledge Engineering and Management - The CommonKADS methodology, University of Amsterdam, 1998.

[Shaw, 1996] Shaw, M. (1996) Procedure Calls are the Assembly Language of Software Interconnection: Connectors Deserve First-Class Status, in D. A. Lamb (Ed), Studies of Software Design, Lecture Notes in Computer Science No. 1078, 17-32. Springer.

[Scheer, 1993] Scheer, A.-W. (1993): Handbuch Informationsmanagement. ARIS-Architektur integrierter Informationssysteme. Gabler, Wiesbaden.

[Scheer -Abolhassan – Jost – Kirchmer, 2002] Scheer, A.-W. - Abolhassan, F. - Jost, W. - Kirchmer, M. (szerk.) (2002): Business Process Excellence - ARIS in practice. Springer, Berlin.

[Smith, 2002] Smith, B, „Ontology and Information Systems”, 2001.13 <http://ontology.buffalo.edu/ontology.doc>(http://www.vtt.fi/tte.staff.publications/ontologies_ho.pdf)

[H. Smith, P. Fingar, 2003] Howard Smith & Peter Fingar Workflow is just a Pi process, V2.1, November 2003.

[Szabó-Hámori, 2006] Szabó Katalin-Hámori Balázs (2006): Információgazdaság. Digitális kapitalizmus vagy új gazdasági rendszer? Akadémiai Kiadó, Budapest

[Szintay, 1985] Szintay, I.: Rendszerelmélet, rendszerszervezés I. Tankönyvkiadó, Budapest, 1985.

[Szendrő, 2003] Szendrő Tamás (2003): A workflow és az ARIS kapcsolata szakdolgozat. BKÁE Információrendszerek Tanszék, Budapest.

[Thomas Erl, 2005] Thomas Erl : "Service-Oriented Architecture: Concepts, Technology, and Design", PRENTICE HALL, ISBN 0-13-185858-0, 2005.

Thomas Erl : SOA Principles of Service Design, PRENTICE HALL, ISBN 0132344823, 2005

[TIBCO, 2004] TIBCO, TIBCO Software Inc. to Acquire General Interface, TIBCO Press Release, Okt. 2004. Letöltve: 2008. április 15.
<http://www.tibco.com/company/news/releases/2004/press641.jsp>

[TIBCO, 2007]TIBCO, TIBCO to Acquire Spot_re, TIBCO Press Release, Máj. 2007. Letöltve:2008.április15.<http://www.tibco.com/company/news/releases/2007/press810.jsp>

[TIBCO, 2008a]TIBCO, TIBCO ActiveMatrix BusinessWorks, TIBCO Software Inc., 2008.Letöltve:2008.április10.<http://www.tibco.com/resources/software/applicationintegration/businessworks.pdf>

[TIBCO, 2008b] TIBCO, TIBCO ActiveMatrix Policy Manager, TIBCO Software Inc., 2008.Letöltve:2008.április10.http://www.tibco.com/resources/software/soa/ds_activematrix_policy_manager.pdf

[TIBCO, 2008c] TIBCO, TIBCO ActiveMatrix Registry, TIBCO Software Inc., 2008. Letöltve: 2008. április 10.
http://www.tibco.com/resources/software/soa/ds_activematrix_registry.pdf

[TIBCO, 2008d] TIBCO, TIBCO ActiveMatrix Service Bus, TIBCO Software Inc., 2008. Letöltve: 2008. április 10.
http://www.tibco.com/resources/software/soa/ds_activematrix_service_bus.pdf

[TIBCO, 2008e] TIBCO, TIBCO ActiveMatrix Service Grid, TIBCO Software Inc., 2008e.Letöltve: 2008. április 10.
http://www.tibco.com/resources/software/soa/ds_activematrix_service_grid.pdf

[Turban, McLean, Wetherbe, 2002] E.Turban, E. McLean, J. Wetherbe: Information Technology for Management 3rd edition, John Wiley & Sons, 2002.

[Turban, 2002] E.Turban: Electronic Commerce 2nd edition, Upper Saddle River, NJ: Prentice – Hall, 2002.

[Uschold, 1996] Mike Uschold & Michael Gruninger: Ontologies: Principles, Methods and Applications, The University of Edinburgh, 1996.

[Uschold, 1997] Mike Uschold, Martin King, Stuart Moralee and Yannis Zorgios: The Enterprise Ontology, The University of Edinburgh, 1997.

[Yin, 1994] Yin, R.K: Case Study Research (2nd ed.). London:Sage Publications, 1994.

[Wahli, 2007] Wahli, U., Business Process Management: Modeling through Monitoring Using WebSphere V6.0.2 Products, IBM International Technical Support Organization, Aug. 2007. Letöltve: 2008. április 24.

<http://www.redbooks.ibm.com/redbooks/pdfs/sg247148.pdf>

[Weber, 1997] Weber, R. (1997) Ontological foundations of Information Systems. Coopers & Lybrand Accounting Research Methodology. Monograph No. 4. Melbourne

Wand, Y. and Weber, R. (1993). On the ontological expressiveness of information systems analysis and design grammars. *Journal of Information Systems*. 3(4), 217-237.

Wand, Y. and Weber, R. (1995) On the deep structure of Information Systems. *Information Systems Journal*. 5, 203-223.

[webMethods, 2005a]webMethods, Service-Oriented Architecture: Revolutionizing IT Systems Development, www.webMethods.com, Dec. 2005. Letöltve: 2008. április 13.

http://www1.webmethods.com/PDF/whitepapers/SOA_Revolutionizing_IT_Systems_Development.pdf

[webMethods, 2005b]webMethods, ServiceNet, www.webMethods.com, 2005. Letöltve: 2008. április 13.

<http://www1.webmethods.com/PDF/ServicenetTechnicalDatasheet.pdf>

webMethods, webMethods Completes Acquisition of Infravio, webMethods Press Releases, Szept. 2006. Letöltve: 2008. április 15.

http://www.webmethods.com/About/News/PressReleases/Details?pressReleaseDetails_param0=6973

webMethods, Software AG Completes Acquisition of webMethods for \$546 Million, webMethods Press Releases, Jún. 2007. Letöltve: 2008. április 15.

http://www.webmethods.com/News/PressReleases/Details?pressReleaseDetails_param0=7336

[WFMC, 1995] WFMC. (1995) The Workflow Reference Model, Version 1.1., <http://www.wfmc.org/standards/docs/tc003v11.pdf>

[W3C, 2005] W3C: OWL Web Ontology Language Guide. W3C Recommendation 10 February 2004. <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>, retrieved Nov 30, 2005.

[Zachman, 1987] Zachman, J.A., "A Framework for Information Systems Architecture", *IBM Systems Journal* Vol 26, No 3, 1987.

[Zachman, 1992] Zachman, J.A., and Sowa, J.F., "Extending and Formalizing the Framework for Information Systems Architecture", IBM Systems Journal, Vol 31, No 3, 1992

ÁBRAJEGYZÉK

1. ábra Az integrált rendszerek evolúciója [Turban, McLean, Wetherbe 2002].....	15
2. ábra Séma integráció	18
3. ábra Üzleti integrációs modell [Edvard A. Stohr , Jeffrey V. Nickerson , 2003]	31
4. ábra A WfMC referencia modell [WFMC, 1995].....	59
5. ábra Modellvezérelt architektúra [OMG, 2002].....	72
6. ábra Szolgáltatás-orientált megoldás/alkalmazás/rendszer [Thomas Erl, 2005].....	75
7. ábra Szolgáltatási szerződés [Thomas Erl, 2005]	77
8. ábra SOA rétegek [Bertrand, 2007].....	79
9. ábra SOA referencia architektúra [Bertrand, 2007]	80
10. ábra A feljövő technológiák hype-görbéje [Fenn, 2007].....	87
11. ábra A feljövő technológiák prioritás-mátrix [Fenn, 2007].....	88
12. ábra SOA folyamatmotor- és együttműködési részegységek piaca az eladott licenzek alapján .	89
13. ábra Mágikus kvadránsok – SOA irányítás [Kenny, Plummer, 2007]	97
14. ábra A feladattípusok jelölőelemei a Modelerben	100
15. ábra A döntések és az összefésülés jelölőelemei a Modelerben	102
16. ábra A párhuzamos végrehajtás segédelemei a Modelerben.....	104
17. ábra A csomópontok jelölőelemei a Modelerben.....	105
18. ábra A leképezés, szolgáltatás és helyi tároló jelölőelemei a Modelerben	107
19. ábra Különböző perspektívák az Integration Developer-ben	110
20. ábra Az összeszerelési diagram az Integration Developer-ben	111
21. ábra Az ARIS architektúra – nézetek és leíró szintek.....	145
22. ábra ARIS Üzleti tervezés háza [Szendrő, 2003]	149
23. ábra Üzleti folyamatok specifikálása – szolgáltatások meghatározása [Jörg Klückmann, 2006].....	154
24. ábra Integrált repository az ARIS-ban [Jörg Klückmann, 2006].....	155
25. ábra A 10 lépéses eljárás [Jörg Klückmann, 2007]	156
26. ábra IS contextus modell	160
27. ábra Access diagram	160
28. ábra UML komponens diagram.....	161
29. ábra Alkalmazás rendszer diagram.....	161
30. ábra Access diagram (IT)	161
31. ábra ARIS architektúra – üzleti integrációs modell.....	166
32. ábra Értékesítési adatok szakkifejezés modellje	168
34. ábra Funkció allokációs diagram.....	170
35. ábra ARIS „swimlane” folyamatlánc	170
37. ábra Alkalmazási rendszer típus diagram	172
38. ábra Alkalmazás, modul funkcióhoz rendelése	173
39. ábra Vállalati értékteremtő lánc diagram.....	174
40. ábra Részfolyamat integrálása EPC-be.....	175
41. ábra eEPC integrálása értékteremtő lánc diagramba.....	176
42. ábra Szervezeti elemek – magas szintű folyamatok integráció.....	176
43. ábra Szervezeti elemek – eEPC integráció.....	177
44. ábra Stratégiai célok – folyamat/tevékenység integráció.....	178
45. ábra Egyedi attribútum megadása a munkaelem sablonnál	182
46. ábra A megrendelési tétel összértékének kiszámítására megadott kifejezés.....	183
47. ábra Számosság megadása egy munkaelem attribútumához	184
48. ábra A Megrendelés globális folyamat	186
49. ábra Az ár meghatározása részfolyamat	188
50. ábra A Kedvezményre jogosult? - döntést kiértékelő kifejezés megadása	189
51. ábra A projekt az importálás után az Integration Developer-ben	192
52. ábra Az üzletiszabály feladathoz megadott döntési tábla	193
53. ábra A futtatható SOA alkalmazás.....	195
54. ábra A Megrendelés folyamat a tesztkörnyezetben.....	196
55. ábra A várakozó emberi feladatok a BPC Explorer-ben.....	197
56. ábra A megrendelés ARIS modellje.....	199

57. ábra A szolgáltatás interface modellezése EPC-ben.....	200
58. ábra Adatokkal, rendszerfunkciókkal és interfészekkel kiegészített EPC	201
59. ábra Service – követelmény hozzárendelés	202
60. ábra A megrendelés szolgáltatás orientált üzleti folyamat modellje.....	202
61. ábra Class diagram létrehozása (Drag & Drop technikát alkalmazhatunk)	204
62. ábra Üzleti és technikai entitások közötti kapcsolat.....	205
63. ábra A folyamat és az IT környezet közötti kapcsolat	206
64. ábra A szolgáltatások kiválasztása	207
65. ábra A megrendelés szolgáltatás-orientált folyamata adatokkal és szolgáltatásokkal	207
66. ábra A megrendelés szolgáltatás-orientált folyamatból generált BPEL folyamat.....	208
67. ábra A calculate price allokációs diagram	209
68. ábra A megrendelés BPEL allokációs diagramja	210
69. ábra A folyamat változások követése.....	211
70. ábra A modell és a BPEL kód megfeleltetése.....	213
71. ábra A receive purchase order allokációs diagramjának részletei a megrendelés folyamat BPEL kódjában	214
72. ábra A megrendelés BPEL allokációs diagram részletei a BPEL kódban	215
73. ábra A megrendelés szolgáltatás UML class diagramja	216
74. ábra A megrendelés folyamat BPEL modellje az IBM Websphere Integration Developer-ben	217
75. ábra Az input adatokat kezelő aktivitás implementálása	217
76. ábra A megrendelés folyamat BPEL modellje a változtatás után.....	218
77. ábra A tervezett pilot rendszer VISIO modellje.....	220