

## IQ\*SILK: Modell-vezérelt adatintegrálás logikai alapokon

Benkő Tamás — Krauth Péter\*

### Absztrakt

A jelenleg használatos technológiák nem tudják megfelelő módon kezelni azt a komplexitást, ami pedig elengedhetetlen az elosztott rendszerekben tárolt adatok aggregációjához. Amire szükség van, az egy olyan *új absztrakciós szint*, amely tartalmazza egyrészt az adatforrásokra és az adatkapcsolatokra vonatkozó *metaadatokat*, másrészt mindazokat az *adatnézeteket*, amelyeket az adatokat „fogyasztó” végfelhasználó, ill. alkalmazási rendszer megkövetel.

Ez a cikk az IQSoft Intelligens Software Rt. (a KFKI Csoport tagja) *IQ\*SILK* elnevezésű, olyan *új adatintegrációs technológiáját* ismerteti, amely mindezeket a funkciókat megvalósítja.

IQ\*SILK lehetővé teszi az informatikai részlegek számára, hogy a több – szervezeten belül vagy akár kívül elhelyezkedő – forrásból származó adatokat könnyen integrálhassák *valós időben*, azaz késlekedés nélkül, és azokat az üzleti felhasználók igénye szerint szolgáltatthassák. A SILK valós időben nyújt *átláthatóságot* és *egységes rálátást* akár a beszállítói láncokra, akár az ügyfelekre vonatkozó és más egyéb fontos információk terén, és így elősegítheti, hogy információkkal jobban megalapozott döntéseket hozzanak.

A meglévő informatikai infrastruktúrákhoz való minél könnyebb és teljesebb illeszkedés érdekében olyan szabványos ipari megközelítésekre épít, mint a *Java*, *XML* (eXtended Markup Language), *UML* (Universal Modelling Language), *OCL* (Object Constraint Language), *OQL* (Object Query Language) és *XMI* (XML Metadata Interchange). A SILK azonban több egy egyszerű adatintegrációs környezetnél. Építve az IQSoft évtizedes tapasztalataira intelligens, logikai alapokon működő szoftverek készítésében, a SILK élenjáró *Prolog-megoldásokat* alkalmaz az adatintegrálás hagyományosan nem, vagy csak nagy nehézségekkel kezelhető területein (lekérdezésoptimalizálás, metaadat-ellenőrzés, hasonlóságelemzés, modellegyesítés stb.)

---

\* Benkő Tamás az IQSoft Rt. (KFKI Csoport) szoftvertervezője és -fejlesztője, a SILK-projekt szakmai vezetője.

Krauth Péter a KFKI Számítástechnikai Rt. munkatársa, az IQSoft SILK-projekt vezetője.

## 1. Bevezetés az adatintegrációba

Az adatok iránti igényeknek és az adatok elérhetőségének hallatlan mértékű megnövekedése következtében ma egy átlagos vállalat alkalmazásfejlesztési költségvetésének nagy részét olyan programok írására fordítja, amelyek összekapcsolják a háttérben működő (back-office), a közvetlen kiszolgálást támogató (front-office) és az elektronikus üzletviteli alkalmazásokat, hogy megoszthassák közöttük azokat a kritikus információkat, amelyek az üzleti folyamatok teljes körét vezérlik az eladásoktól kezdve, a vevőszolgáltatón át, a beszállítók kezeléséig. Az adatok „körbe mozgatására” fordított összegek azonban elveszik az erőforrásokat ugyanezen adatok versenyelőny elérésére történő kihasználása elől, ezzel is csökkentve annak lehetőségét, hogy az elektronikus üzletvitel technológiába történő befektetések kifizetődőek legyenek. Valójában a szervezetek az általuk kezelt információ kevesebb mint 20%-át használják fel újra valamilyen más célra, és nem használják ki az információk sokoldalú felhasználásból származó előnyöket sem a szervezeten belül, sem azon kívül.

E probléma megoldását szem előtt tartva az integrációs szoftverek egy új osztálya, az ún. *vállalati adatintegrációs* (Enterprise Information Integration, EII) *rendszerek* azt ígéri, hogy képesek megfelelni annak az elvárásnak, amit az elosztott adatoknak üzleti információkká való átalakítása jelent. Ez a cikk részletezi az adatintegrációs követelményeket, leírja, hogyan lehet az EII rendszereket ezek kielégítésére felhasználni, és bemutat egy nemzetközi összefogással készülő, konkrét megoldást adó terméket (*IQ\*SILK*).

### 1.1. Az elosztott adatok újrafelhasználása költséges

Egyes alkalmazások esetén a különböző, ill. elkülönült adatforrások integrálásának komplexitása még mindig túl nagy sok mai, piacvezető, elektronikus üzletviteli alkalmazások építésére és integrálására szolgáló környezet számára. Az adatintegrációs megoldások iránti igény gyorsan növekszik, amit elsősorban az elektronikus üzletvitel és a portálok követelményei határoznak meg<sup>1</sup>.

A vállalatok jelentős befektetéseket eszközölnek integrációs megoldásokba a teljes üzleti integráció érdekében, hogy az alkalmazottak, a partnerek és az ügyfelek hatékonyabban tudjanak vásárolni, dolgozni és eladni. A befektetések volumene a következő 4 évben 3,4 milliárd \$-ról várhatóan 6,1 milliárd \$-ra fog emelkedni<sup>2</sup>.

---

<sup>1</sup> Giga Information Group, 2000

<sup>2</sup> IDC, 2001

A hatékonyság érdekében ezeknek a megoldásoknak a megfelelő információt a megfelelő emberhez, ill. rendszerhez kell eljuttatniuk. De az érdemi adatok megtalálása, amelyek többnyire több rendszerben vannak elszórva, és ezek olyan módon történő kombinálása, hogy az valóban odaillő és időszerű legyen, igen bonyolult probléma. Az egyik külföldi, a Fortune 100-as listáján szereplő nagyvállalatnál például az informatikai vezetők úgy becsülik, hogy a külső elektronikus üzletvitelt szolgáltató, valamint a belső intranet-rendszerük több mint 130 belső alkalmazástól és külső alkalmazások százaitól igényel rendszeresen adatokat, és ezek egyike sem szolgáltatja azokat rendszeres és szisztematikus módon egyik rendszerüknek sem.

E vállalat dilemmája tipikusnak mondható, mivel – legalábbis mindmáig – az egyedi programfejlesztés volt az egyetlen módja az összes ilyen különböző forrásból (akár a tűzfalon belülről, akár azon kívülről) származó adatok aggregációjának. A végfelhasználói adatkérések számának és változatosságának hirtelen és eddig nem látott mértékű megnövekedésével az ilyen rendkívül munkaigényes technológia alkalmazása jelentős elmaradásokat idézett elő a vállalatok informatikai területein. Ennek „eredményeképpen” 35-40%-ra tehető az informatikai szervezetek költségvetésének az a része, amit integrációs projektekre fordítanak<sup>3</sup>.

Sokakat ez a helyzet arra az adatelérési és -integrációs problémára emlékeztet, amellyel az informatikai szervezeteknek akkor kellett szembenéznük, amikor egyedi COBOL-kódokat kellett fabrikálniuk az információ belső szigeteinek „meghódítására”. Az SQL elnevezésű magas szintű adatlekérdező nyelv megjelenésével az egyedi programkódok tömegeit lehetett néhány soros olyan lekérdezésekkel helyettesíteni, amelyek bármilyen relációs adatforráshoz hozzáfértek, függetlenül azok adatszerkezetétől és -szervezésétől. Mint majd látni fogjuk, az *IQ\*SILK* – ebben a vonatkozásban – hasonló képességgel rendelkezik heterogén forrásokból származó adatok aggregációjánál.

## 1.2 Elosztott adatok információvá alakítása

Az üzleti felhasználók számára az, hogy a közvetlen üzleti jelentősséggel bíró adatokat könnyen és gyorsan el lehessen érni, teljesen egyértelmű motivációt jelent. Az értékesítési szakemberek lerövidíthetik az értékesítési ciklust a különböző értékesítés-automatizálási (Sales Force Automation, SFA) és ügyfélkapcsolat-kezelő (Customer Relationship Management, CRM) rendszerek között szétszortott információkhoz való időben történő

---

<sup>3</sup> Gartner Group, 2002

hozzáféréssel. A partnerek hatékonyabban tervezhetik a termelést a jobban átlátható beszállítói lánc mentén. Az ügyfelek jobb kiszolgálást kaphatnak a vevőszolgálatától, ha az a számlázási, a szállítási és a CRM-rendszerekből kinyerhető információkhoz egyaránt könnyen hozzáfér.

Az informatikai szakértők számára nyilvánvaló, hogy az adatelérés útjának leegyszerűsítése jelentősen csökkentheti a lemaradást a munkában, és lerövidítheti az üzleti adatkérések válaszidejét. Megtalálni az utat az adatintegrálásra fordított pénzüsszegek csökkentéséhez jelentős haszonnal járhat. A jelenlegi technológiák azonban nem tudják megfelelő módon kezelni azt a komplexitást, ami pedig alapvetően jellemzi az elosztott rendszerekben tárolt adatok aggregációját. Amire szükség van az egy olyan új absztrakciós szint, amely tartalmazza egyrészt az adatforrásokra és az adatkapcsolatokra vonatkozó metaadatokat, másrészt mindazokat az adatnézeteket, amelyeket az adatokat „fogyasztó” végfelhasználó, ill. alkalmazási rendszer megkövetel.

### **1.3 Technológiai körkép**

A vállalati adatok kihasználásának szükségessége nem új dolog. Az elmúlt időszakokban az informatikai szervezetek számos technológiát vetettek be a vállalati és a vállalatközi adatokhoz való hozzáférés biztosítása érdekében. Az alábbiakban teljes körképet adunk ezekről a technológiákról, beleértve az EII-t is.

#### **1.3.1 Vállalati alkalmazásintegrálási rendszerek (Enterprise Application Integration, EAI)**

Ezek a technológiák alkalmazásokat integrálnak az üzleti folyamatok szintjén. Alapvető fontosságúak egymással együttműködni képes elektronikus üzletviteli alkalmazások létrehozásában, mivel biztosítják az adatok integritását, és hogy az üzletvitel szabályai megőrződjenek. Azonban mivel csak az üzleti folyamatok viszonylag absztrakt szintjén alkalmazhatók, ezek a technológiák az adatintegrációt általában „bedrótozva” hagyják az adatbázisok összekapcsolási interfészében vagy az egyedi programkódokban.

#### **1.3.2 Onlájni analitikus feldolgozás (Online Analytical Processing, OLAP), adatbányászat és adattárház-alapú rendszerek**

Ezek a rendszerek adattárházakba vagy „adatpiacokba” duplikálják és konvertálják a meglévő adatokat, amelyek speciális szerkezetük segítségével igen kifinomult elemzési és jelentéskészítési lehetőségeket nyújtanak az üzleti szakértők számára. Alapvető fontosságúak a bonyolult üzleti elemzésekhez, de mivel új, komplex adatstruktúrákat

igényelnek, amelyeket a meglévő rendszerekkel állandóan szinkronban kell tartani, nem ideálisak valós idejű lekérdezésre az átlagos üzleti felhasználók esetén.

### **1.3.3 Üzletközi integrálási (Business-to-Business Integration, B2BI) rendszerek**

Ezek a technológiák az üzleti tranzakciók továbbítását és az üzenetközvetítést helyezik a középpontba, az XML használatával hozva létre pont-pont kapcsolatokat az üzleti alkalmazások között. E rendszerek magas szintű funkcionális lehetőségeket adnak a Web üzletviteli célú kihasználásához – különösen azon vállalatoknál, ahol korábban már megvalósítottak EDI (Electronic Data Interchange) technológiákat, és kiépítettek virtuális magánhálózatokat (Virtual Private Network, VPN). Általában nem rendelkeznek adatintegrálási képességekkel.

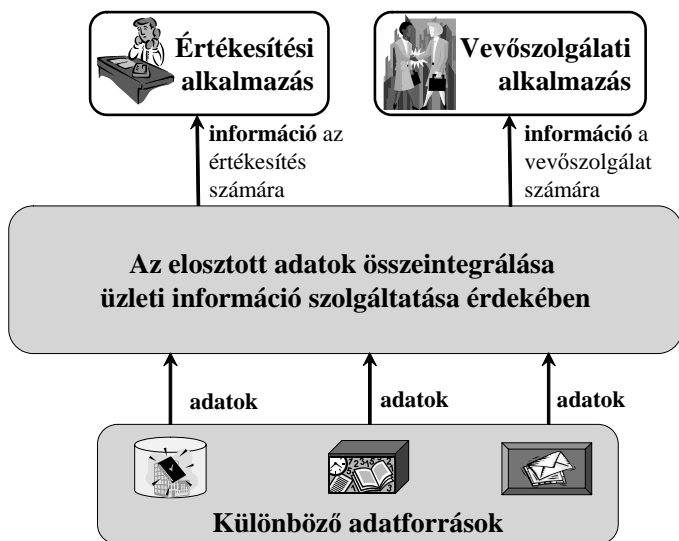
### **1.3.4 Tartalomaggregálási (Content Aggregation) technológiák**

Ezek a rendszerek többnyire strukturálatlan tartalomhoz, mint pl. HTML-lapokhoz, dokumentumokhoz és szövegállományokhoz nyújtanak hozzáférést. Habár hasznosak egyes műveleteknél, mint pl. egy termékkatalógus feltöltése, nem nyújtanak olyan típusú adatbázis-hozzáférést és adatfeldolgozási műveleteket, amelyek ahhoz szükségesek, hogy az integrált adatokat a vállalati és az együttműködő elektronikus üzletviteli alkalmazások teljes köréhez lehessen szolgáltatni.

### **1.3.5 Vállalati adatintegrálási (Enterprise Information Integration, EII) technológia**

A vállalati adatintegrálás valós idejű integrációt biztosít különböző adatforrások között, bárhol, bármilyen alkalmazás esetén, valamilyen szabványos adatlekérdezési nyelv, pl. Object Query Language (OQL) vagy XML Query (XQuery) segítségével, ahol az adatfeldolgozást és az adatszolgáltatást a rugalmas adathozzáférés és a hatékony adatintegrálás érdekében optimalizálják.

Az egyes vállalatok sajátos helyzetétől függően az EII-t gyors, magas szintű adatintegráció megvalósítására lehet használni akár önálló megoldásként, akár egy nagyobb integrációs projekt részeként. Az EII-technológiák szabványos Web-kapcsolatokon keresztül férnek hozzá az adatforrások adott köréhez, és olyan eredményeket adnak vissza, amelyek automatikusan címkékkel (ú.n. tag-ekkel) vannak struktúrálva, Web-alkalmazásokban történő megjeleníthetőség céljából. Ezért Web-szolgáltatásként is meghívhatók, vagy gyorsan és könnyen be lehet ágyazni őket valamilyen integrált alkalmazásba.



1. ábra: A vállalati adatintegráció (EII) leegyszerűsíti az információszolgáltatást

#### 1.4 Példa: „önkiszolgáló” alkalmazás ügyfeleknek

Az EII egyik legvonzóbb felhasználási lehetősége, hogy teljes körű, naprakész információt biztosítson az ügyfeleknek olyan helyeken, ahol a lényegi adatokhoz való hozzáférés döntheti el, hogy megtartanak vagy elvesztenek-e egy értékes ügyfélkapcsolatot.

Egy ismert nagyvállalat komplex, rövid életciklusú termékcsaládot értékesít. A termékek között az összefüggések kiterjedt hálózata áll fenn. Az ügyfeleknek tudniuk kell, hogy melyik termék melyik másikkal kompatibilis, melyiket bővítették, milyen szoftverjavítások állnak rendelkezésre az általuk megvásárolt termékeknél, és sok hasonló dolgot. Mielőtt az EII-technológiát felhasználták volna, ezeket a kérdéseket csak úgy lehetett megválaszolni, ha meglátogattak hét különböző Web-címet a vállalat Web-helyén, hosszú és nehezen érthető táblázatokban megtalálták a kívánt információt, feljegyzéseket készítettek, és manuálisan összeállították a kívánt információhalmazt.

A vállalat – felismerve a problémát és a benne rejlő veszélyeket – egy olyan rendszer készítését vette tervbe, amelyik egyetlen, integrált és személyre szabható Web-lapot kínál ügyfeleinek, valamint lehetővé teszi számukra, hogy gyorsan és könnyen böngészhessenek

a szükséges információk között. Az új integrált Web-alkalmazás mindig a megfelelő információt fogja megjeleníteni az ügyfél múltbeli vásárlásaira vonatkozó bejegyzésekből (XML Web-szolgáltatás), a termékinformációs adatbázisból (Oracle adatbázis), a hardver-szoftver kompatibilitási nyilvántartásból (HTML-lap), a bejelentéskövető adatbázisból (Oracle), valamint a terméknevek, ill. -konvenciók metaadat-nyilvántartásából (házi készítésű XML-alkalmazás). Az integrált alkalmazás lehetővé teszi majd az egyes ügyfeleknek, hogy néhány perc alatt mindent megtekinthessenek, amire csak szükségük lehet, hogy a számukra érdekes hardver- és szoftvertermékeket meghatározhassák.

A megoldáshoz várhatóan mindössze két hétre lesz szükség. Az új megoldás üzleti előnyei közé tartozik a rövidebb értékesítési ciklus, a nagyobb ügyfélelégedettség, az alacsonyabb vevőszolgálati költségek és az a lehetőség, hogy a műszaki támogató személyzetet a bonyolultabb ügyfélproblémák megoldására irányíthassák át.

## 2. Egy konkrét adatintegrációs rendszer: az IQ\*SILK

Az előző fejezetben ismertetett okok miatt (hatékonyság mind a működésben, mind a fejlesztésben, befektetések megtérülése, költségkímélés) a vállalatok egyre gyakrabban nem a kockázatos, teljes átírásokat igénylő, a meglévő rendszerek „kidobásával” együtt járó, ilyen módon „felforgató” hatású és revolúciós jellegű megközelítéseket keresik, hanem az újrafelhasználást előtérbe állító, evolúciós jellegű megoldásokat. Azok a kérdések, amelyek a vállalatoknál található üzleti alkalmazások teljes rendszerének evolúciójára és karbantartására irányulnak (pl. milyen módon lehet gyorsan és megbízhatóan együttműködővé tenni az alkalmazásokat, vagy hogyan lehet minél többet kihozni az alkalmazásokban „elrejtett” adatokból), egyre inkább az informatika hatékony alkalmazásának központi témáivá válnak.

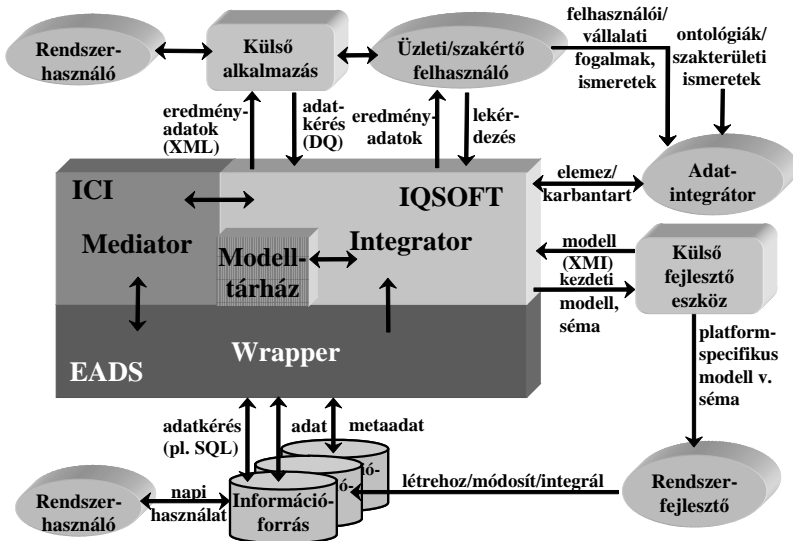
Ebben a fejezetben egy Európai Unió támogatásban részesülő projekt (az ún. SILK-projekt<sup>4</sup>) keretében fejlesztés alatt álló *adatintegrációs rendszert* (ld. 2. ábra) ismertetünk. A SILK-fejlesztés fő hajtóereje a vállalatok – korszerű vagy elavult, de – működő rendszereiben található adatok minél teljesebb újrafelhasználására vonatkozó igény, az, hogy összetett, együttműködő alkalmazási rendszereket lehessen létrehozni új, dinamikusnak keletkező üzleti problémák hatékony megoldására.

---

<sup>4</sup> A SILK-projekt (System Integration via Logic and Knowledge) az IST (Information Society Technologies) 5. keretprogram része. A projekt résztvevői: IQSOFT Ltd. (Magyarország, koordinátor), EADS Systems & Defence Electronics (korábban Matra Systems & Information, Franciaország), a National Institute for Research and Development in Informatics (Románia) és az Industrial Development and Education Centre (Görögország).

A SILK-projekt elsődleges célja ennek megfelelően olyan *platform-, gyártó- és eszköz-független adatintegrációs eszközkészlet* kidolgozása, amely egységes és hatékony hozzáférést tesz lehetővé a – mind technikai, mind szemantikai értelemben – heterogén adatforrásokhoz. A SILK az információforrások meglehetősen széles körét képes bevonni az adatintegrálás hatókörébe, támogatja a relációs és az objektum-orientált adatbázisokat, a félig strukturált (pl. XML-állományok), valamint a Web-szolgáltatások formájában elérhető forrásokat.

A SILK-rendszer legfontosabb felhasználója az ún. *adatintegrátor*, aki az adatintegráció teljes folyamatának megfelelő végrehajtásáért felelős. Habár a SILK – intelligens eszközöket nyújtva az integrációs folyamathoz – sok fáradságos részfeladat megoldását átvállalja, az adatintegrátor interaktív beavatkozási lehetőségei és ezen keresztül az ellenőrző szerepe mindvégig biztosítottak. Bár a végfelhasználók és a rendszer közötti *közvetítő* szerepe nélkülözhetetlen az adatintegrációhoz, elsősorban nem informatikai szakképzettséggel, hanem általános modellezési, megértő és áttekintő képességekkel kell rendelkeznie. Többek között meg kell értenie a vállalat egyes felhasználói csoportjainak adatokkal kapcsolatos sajátos látásmódját, és célirányosan kell használnia egy-egy adott szakterület már létező fogalmi rendszereit, ismeretanyagát.



2. ábra : A SILK-rendszer használata



A SILK-rendszer legigényesebb felhasználói ezzel szemben az üzleti tevékenységek *vezetői és szakértői* közül kerülnek ki. Ők támasztják a legnagyobb elvárásokat az adatintegrációval szemben. A napi munkájukban *ad hoc* felmerülő, *azonnali* választ igénylő, *üzleti indíttatású* kérdéseik megválaszolása gyakran jelent kihívást az adatintegrációs rendszerek számára.

„A SILK-rendszer leggyakoribb felhasználói” címet azonban valószínűleg azok az alkalmazások érdemlik ki, amelyek a SILK szolgáltatásait használva *változatos*, de *tervezett* módon dolgozzák fel és szállítják a végfelhasználók részére az információt, ami máskülönben az adatforrásokban elrejtve, csak nehezen hozzáférhetően van jelen. Az operatív szinten elhelyezkedő végfelhasználók rendszerint csak ilyen alkalmazásokat közbeiktatva részesülnek az adatintegráció szolgáltatásaiból.

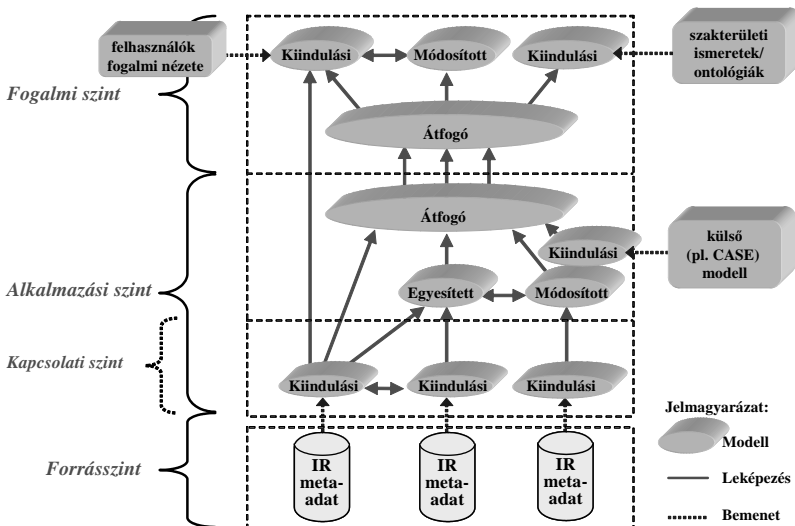
A SILK-rendszer azonban nemcsak a heterogén adatforrások dinamikus elérését (az ún. *mediációt*) támogatja, hanem ugyanezen adatforrások egységesebb formába való átalakítását, pl. egységes adatbázisba való integrálását is. A metaadatok kezelésére szolgáló funkciói (ld. a 235. oldalon: összehasonlítás, ellenőrzés, egyesítés stb.) segítségével ugyanis az adatforrások átfedései (redundanciák) feltérképezhetők, majd ezen keresztül olyan integráltabb modelljeiket lehet előállítani, amelyek az adatforrások későbbi –evolúciós vagy revolúciós – továbbfejlesztésének alapjai lehetnek.

## 2.1 Metaadatok kezelése

A metaadatoknak az adatintegrációban betöltött kiemelt központi szerepe miatt a következőkben részletesebben ismertetjük a kezelésükkel kapcsolatos legfontosabb fogalmakat és funkciókat.

Az egységes adatlekérdezés érdekében az integrálandó adatforrásokra vonatkozó információkat össze kell gyűjteni, rendszerezni kell, kapcsolatba kell hozni őket egymással, és mindezen információkat folyamatosan karban kell tartani. A metainformációk ezen összességét az adatintegráció *ismeretbázisának* is nevezhetjük.

A SILK-megközelítés sajátossága, hogy a tényleges adatintegrációt megelőzően az adatforrások metaadatait (tkp. modelljeit) először egy speciális felépítésű, ún. *modelltárházban* (ld. 3. ábra) integrálja. A SILK modelltárháza *modelleket*, *szabályokat* és *leképezéseket* tartalmaz. Ezek az adatintegráció folyamatában keletkező fő információcsoportok, és számos funkció áll rendelkezésre ezek kezelésére (pl. összehasonlítás, ellenőrzés és egyesítés).



3. ábra: A modelltárház vázlatos felépítése

Ilyen jellegű „metaadattár” tulajdonképpen minden adatintegrációs rendszernek használnia kell valamilyen formában. Ami ebben a vonatkozásban megkülönbözteti a SILK-et más hasonló eszközöktől, az a modelltárház komplexitása, és az ehhez biztosított, egészen magas szintű támogatás az adatintegrátorok számára. A modellek maguk ugyanis olyan UML-osztálymodellek (azaz öröklődési képességgel rendelkező objektummodellek), amelyeket OCL-szabályokkal lehet tovább finomítani, és ilyen módon rendkívül gazdag szemantikai tartalommal bővíteni. A *szemantikai gazdagság* azért különösen fontos, mert fordítottan arányos az adatlekérdezés *tömörségével*, és – ezen keresztül – az adatkérések kidolgozásához szükséges *ráfordításokkal*.

### 2.1.1 Modellek

A modellek a rendszerek szerkezeti tulajdonságaira vonatkozó ismereteket tartalmazzák, mint pl. egyedek vagy objektumok és a köztük fennálló kapcsolatok. A SILK-ben a modell fogalma az UML-en alapul [3], bizonyos ún. leíró logikai (Description Logics [4]) lehetőségekkel kiegészítve. Ez utóbbiakra elsősorban a fogalmi szintű modellekben van szükség a minél kényelmesebb fogalommegadás és -tömörség érdekében. A SILK-rendszer alapvetően kétféle modellt kezel:

- *Alkalmazási modellek* ábrázolják valamilyen működő (vagy megvalósítandó) rendszer szerkezetét. Ennek megfelelően meglehetősen részletesek és pontosak.
- *Fogalmi modellek* ábrázolják a felhasználók fogalmait és fejezik ki adatfelfogásukat. Ezek többnyire jóval kevésbé kidolgozottak és részletezettek az alkalmazási szintüeknél. Az átfogóbb vállalati vagy szakterületi modellek is ide tartoznak, és egységes, széles körben (vállalat, vállalatcsoport vagy szakterület) elfogadott fogalomrendszert biztosítanak az adatkérések minél hatékonyabb megfogalmazásához.

Más oldalról lehet *egyesített*, ill. *átfogó* modellekről beszélni, amelyeket a modellintegrációs folyamat során más modellekből, ill. más modellekkel összefüggésben hoztak létre; és lehet *kiindulási* modellekről szólni, azaz olyan modellekről, amelyek pl. egy felhasználó vagy felhasználói csoport közvetlen adatkészletét, ill. az információforrások közvetlen adatmodelljét (*kapcsolati* modellek) ábrázolják. Ez utóbbiak többnyire kiindulásul szolgálnak további, származtatott modellekhez.

Annak érdekében, hogy a fenti típusú modellek mindegyikét ábrázolni tudja, a SILK három fontos megközelítés legjobb elemeit egyesíti:

1. *objektum-orientált modellezést* az adatforrások szerkezeti tulajdonságainak leírására;
2. *szabálylogikát* a nem szerkezeti jellegű tulajdonságok leírására;
3. *leíró logikát* a fogalmi viszonyok és tulajdonságok leírására.

### 2.1.2 Szabályok

Gyakran előfordul, hogy olyan információkat kell kezelni, amelyek nem szerkezeti jellegűek (például objektumok között olyan összefüggések kifejezéséhez, hogy a „keresztnév” és a „családnév” összefűzése adja ki a „teljes nevet”). Erre a célra *szabályokat* használ a SILK. A szabályok kiemelt szerepet kapnak a SILK-ben (ld. még alább a 2.1.3 Leképezések c. szakaszban), és alapvető jelentőségükre a SILK-kel érintkező területeken (pl. adatbázisok tervezése, rendszerintegráció) már többen felhívták a figyelmet:

„A rendszerek integrációjánál az első igazi forradalom a relációs adatbázistermékek megjelenése volt. A második nagy forradalmat a *szabályok* jelentik, mint az üzleti logika konzisztens, explicit megfogalmazásai. A szabályokat úgy lehet tekinteni, mint amik végre megvalósítják az első forradalom régóta beváltatlan ígéretét: azt a képességet, hogy az adatintegritás biztosítása *elkülönüljön* az alkalmazási programoktól, és hogy azok egy olyan

helyre<sup>5</sup> kerüljenek, ahol minden alkalmazásra egységesen lehet őket alkalmazni.” [1]

„A szabálymegfogalmazás *maga* az adatbázistervezés.” [2]

A SILK-ben a szabályok hasonlóak az OCL-lel, az UML kiegészítő szabályleíró nyelvével megfogalmazott szabályokhoz, és kezelésükre, logikai elemzésükre CLP (Constraint Logic Programming) technikákat [5] használunk az adatintegrációhoz elengedhetetlen metaadat-integrálásnál.

### 2.1.3 Leképezések

Amennyiben több adatforrást érintő lekérdezést akarunk végrehajtani, akkor a megfelelő modellek között először *leképezéseket* kell kialakítani, azaz a modellek megfelelő objektumait össze kell kapcsolni, és a kapcsolatokat alkalmas *szabályokkal* kell leírni. Csak így lesz lehetséges, hogy egy magas szintű felhasználói fogalomra hivatkozva valamilyen adatkérésnél automatikusan létrejőjenek azok a konkrét adathozzáférések, amelyeket az egyes adatforrások már közvetlenül végre tudnak hajtani. A kinyert adatokat szintén e leképezések alapján lehet összeállítani, az eredeti kérdés szintjén értelmes, használható információvá változtatni (ld. még 2.2.2 A Mediator alrendszer c. szakaszt).

## 2.2 A SILK-rendszer fő komponensei

A SILK-rendszer három fő alrendszerből épül fel, a 4. ábrán látható módon:

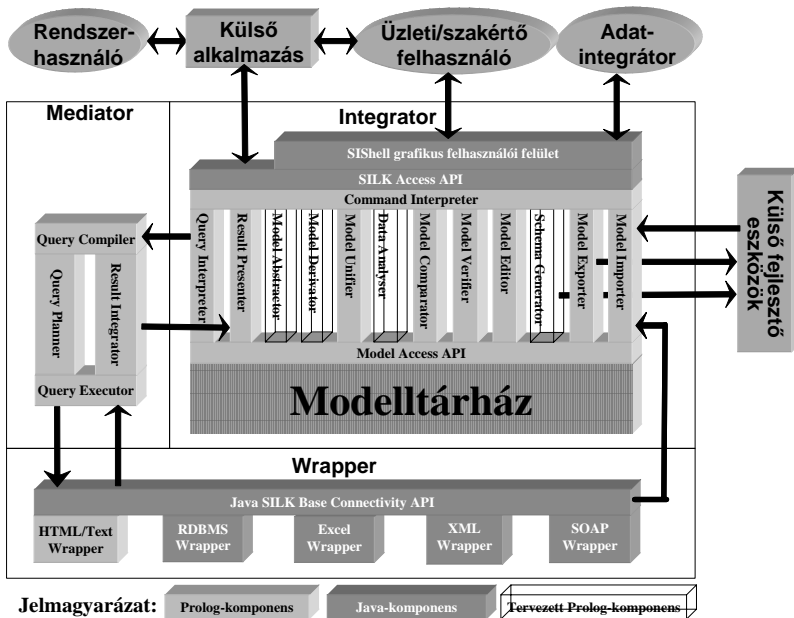
- Az *Integrator* támogatást ad az adatintegrátornak a SILK modelltárházának felépítésében és használatában. Eszközöket biztosít a kiindulási modellek létrehozására és módosítására, a modellek összekapcsolására, összehasonlítására és új, egyesített vagy származtatott modellek létrehozására. Az adatkérések összeállítása, előzetes optimalizálása, valamint az eredményül kapott adatok különböző formátumokba történő transzformálása szintén itt valósul meg.
- A *Mediator* fordítja le a felhasználói/vállalati szinten (fogalmi modelleken keresztül) megadott lekérdezéseket az adatforrások által értelmezhető, a saját adatkezelő mechanizmusuk (pl. SQL-kiértékelő) által végrehajtható adatkérésekre [6].

---

<sup>5</sup> A fent említett modelltárház egy ilyen *helyet* valósít meg.

- A *Wrapper* hozzáférést biztosít az adatforrásokhoz egy egységes objektum-orientált megközelítésen keresztül. Támogatja a relációs és az objektum-orientált adatbázisokat, az XML félig strukturált formátumát és a Web-szolgáltatások formájában megjelenő adatforrásokat. Az adatforrások értelmezéséhez szükséges leíró metaadatok kinyerése szintén itt történik.

Az *Integrator* és a *Mediator* alrendszer lényegében Prolog-ban (SICStus Prolog [7]) valósult meg, míg a *Wrapper* alrendszer Java-ban, hogy a különböző adatforrásfajták kezeléséhez rendelkezésre álló könyvtári támogatásokat ki lehessen használni. A Prolog- és a Java-komponensek a *Jasper*-en, a SICStus Prolog Java-kapcsolatot biztosító könyvtárának felhasználásával képesek kommunikálni egymással.



4. ábra: : A SILK-rendszer architektúrája

### 2.2.1 Az Integrator alrendszer

A SILK központi eleme a *modelltárház* (tkp. adatintegrációs ismeretbázis). Ez az a hely, ahol az összes modellt és az ezeket összekapcsoló *leképezéseket* tárolja a rendszer. Az itt tárolt modellek és kapcsolataik meghatározó szerepet töltenek be az adatintegráció megvalósulásában. Az *Integrator* alrendszer ezek kezelését, valamint az adatkérések előkészítését (*Query Interpreter*) és az eredmények megjelenítését (*Result Presenter*) végzi (ld. részletesebben a 2.3 Lekérdezés DataQuery-vel c. szakaszban).

A *Model Editor* (modellkezelő komponens) a modellek létrehozásáért, módosításáért és a modelltárházban való keresésért felelős. A *Model Comparator* (modellösszehasonlító komponens) feladata a hasonló elemek megtalálása a modellekben, és a köztük feltárt hasonlósági kapcsolatok leképezések formájában való felvétele. A *Model Verifier* (modellellenőrző komponens) a modellek konzisztenciájának ellenőrzésére használatos, de az adatforrások utólagos, és a tervezett adtmódosítások előzetes integritásellenőrzésére is alkalmazható. A *Model Unifier* (modellegyesítő komponens) segítségével lehet új, egyesített modelleket létrehozni, amelyek kiindulási pontként szolgálhatnak az érintett adatforrások újrafelkészítéséhez. A *Model Importer* (modellimportáló komponens) külső forrásokból (pl. fejlesztő eszközök, modellező eszközök) képes átvenni már elkészült modelleket XMI formátumban. A *Model Exporter* valamint a *Schema Generator* (modellexportáló és sémageneráló komponensek) a modelltárházban létrejött/módosult modelleket tudják átadni külső fejlesztő (pl. adatbáziskezelő vagy modellkészítő) eszközöknek, további feldolgozás céljából. A *Data Analyser* (adatelemző komponens) az adatforrásokban redundáns adatok lokalizálását végzi, segítségével a redundáns adatforrások szinkronizálásának tervezését lehet könnyebben elvégezni. A *Model Derivator/ Abstractor* (modellszármaztató és -általánosító komponensek) végső soron a rendszerfejlesztőknek adnak támogatást új információs rendszerek, adatbázisok tervezésénél. A modelltárházbeli információk alapján platformközeli (származtatás) vagy logikalizált (általánosított) – tehát pl. egy régi rendszer esetlegességeitől elvonatkoztatott – modellt hoz létre, ill. segíti az adatintegrátort annak létrehozásában.

### 2.2.2 A Mediator alrendszer

A *Mediator* saját formalizmust használ az adatkérések feldolgozása, optimalizálása során (Mediator Logical Language, MLL). A *Query Planner* komponens az erre a formalizmusra a *Query Interpreter* által áttett kérés, és az abban nyíltan vagy burkoltan hivatkozott modellelemek (pl. osztályok, attributumok, kapcsolatok, szabályok, leképezések, képességeleírások) alapján előállítja az adatforrásokra specifikus, optimalizált kéréseket. A *Query Executor* ezeket a kéréseket hajtja végre a megfelelő adatforrásokon.

A *Query Planner* az ún. *Constraint Handling Rules (CHR)* technika használatával — visszafelé és előre haladó következtetési módokat dinamikusan váltogatva — állítja elő a forrás-specifikus kéréseket. Az ugyanarra az adatforrásra vonatkozó kéréseket aggregálja, így minimalizálja a *Mediator* és az adatforrások közötti tranzakciókat, továbbá maximálisan kihasználja az adatforrások saját adatkérés-optimalizálási lehetőségeit.

Az adatkérés tervezése során arra törekszik, hogy amennyire lehet, csökkentse az adatkérések esetleges meghiúsulását a források szintjén, még azelőtt, hogy az adatforrásokhoz tényleges hozzáférés történne. Teljesen biztosan végrehajtható adatkérési tervet azonban nem lehet mindig előállítani, ezért – szükség esetén – a *Query Executor* ún. *automatikus újratervezési* mechanizmust indíthat el a végrehajtás során. Ilyen módon a tervezés és a végrehajtás mindig optimálisan átlapolva, egymással összefésülve történik.

Alkalmas esetben a *Query Planner* egyéb optimalizálást is végezhet: az alternatív tervek közül a legalacsonyabb „költségű” kiválasztása olyan metaadatok alapján történik, mint pl. az adatforrások elérési költsége, a becsült végrehajtási idő, a terv hosszúsága.

Az egyes adatforrásoktól megkapott eredményadatokat a *Result Integrator* komponens állítja össze olyan eredménnyé, ami már konzisztensen az eredeti kérés logikáját és szemantikáját tükrözi. A modelltárház megfelelő (kérésben hivatkozott) részét újrafelhasználva ez a megoldás az adatforrások szemantikai sokféleségének egységes és automatikus kezelését teszi lehetővé.

Bizonyos, adatkérésben előírt szabályok fennállásának ellenőrzése csak az egyes adatforrásokból származó részeredmények ismeretében lehetséges. Ezek elvégzése különböző logikai műveletek segítségével (visszalépés, szabálykiterjesztés stb.) szintén a *Result Integrator* feladatkörébe tartozik.

### 2.2.3 A Wrapper alrendszer

A SILK jelenleg a következő adatforrásokhoz való hozzáférést támogatja: 1) relációs adatbázisok (RDBMS), 2) Excel-adatlapok, 3) XML-állományok, és 4) SOAP-alapú Web-szolgáltatások. Minden ilyen adatforrás-típusra rendelkezésre áll egy lekérdező komponens a *Wrapper* alrendszerben, amely az adott forrástípusra specifikus adatkéréseket végrehajtja, az adathozzáféréseket kezeli.

Annak érdekében, hogy újabb adatforrás-típusokat természetes és könnyen megvalósítható módon lehessen a SILK-rendszerhez illeszteni, egy közbenső, kellően általános hozzáférési felületreget iktattunk közbe *Wrapper*-ben – ez a *Java SILK Base Connectivity (JSBC) API*. Ez a megoldás teszi azt is lehetővé, hogy az adatforrások metaadataihoz közvetlenül

hozzá lehessen férni, és a *Model Importer*-en keresztül be lehessen integrálni a modelltárházba ú.n. *kapcsolati modellek* formájában (ld. 2.1.1.1 Modellek c. szakasz).

A JSBC-nek a SILK-ben hasonló szerepe van, mint az ipari szabvány JDBC-nek (Java Data Base Connectivity) a közvetlen adatbázis-hozzáférések kezelésében, de azzal ellentétben a JSBC magasabb szintű funkcionalitást nyújt, továbbá általános UML-fogalmakat és objektumokat valósít meg és kezel.

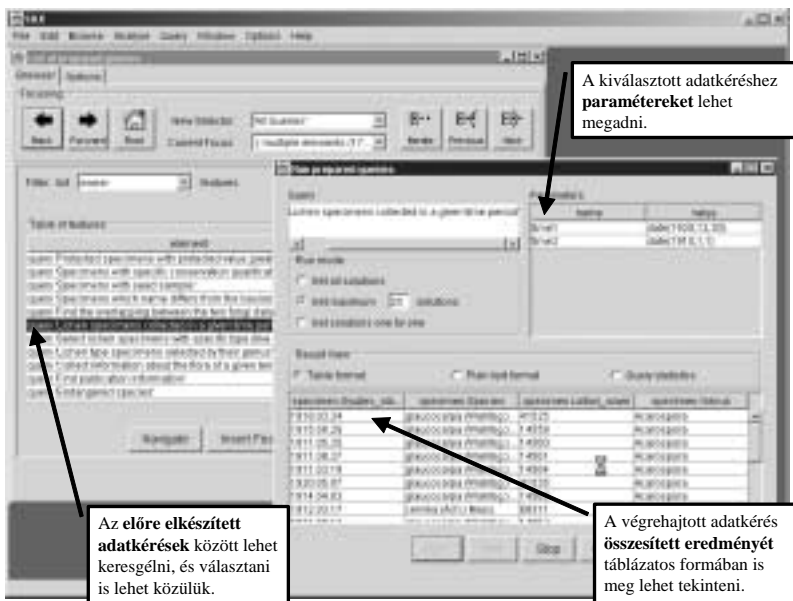
## 2.3 Lekérdezés DataQuery-vel

A SILK-eszközkészlet szabványos OQL (Object Query Language) és OCL (Object Constraint Language) nyelvekre épülő DataQuery (DQ) lekérdezésekkel használható, hogy magas szinten optimalizált, adatspecifikus, valós idejű adathozzáférést és adatfeldolgozást tudjon biztosítani, tűzfalon innen és túl, a forrás helyétől, formátumától és típusától függetlenül. Az SQL-hez hasonlóan a DQ oldalsó programkódokat és több hónapnyi, pont-pont kapcsolatokra épülő adatelérések egyedi fejlesztői munkáját képes helyettesíteni magas szintű adathozzáférésekkel, szűréssel, transzformációval és megjelenítéssel – mindezt néhány perc alatt, a szokásos adatbázis-lekérdezések stílusában. A SILK XML (vagy más pl. Excel, szöveges) formátumban adja vissza az eredményeket, hogy pl. Web-alkalmazások is könnyen felhasználhassák azokat.

A DataQuery hatékony nyelv, ahol a lekérdezések tömörek és könnyen érthetőek. Elég rugalmas ahhoz, hogy az XML (és egyéb) információforrások széles körét le lehessen kérdezni vele, beleértve mind az adatbázisokat, mind a dokumentumokat. A DataQuery-t lehet használni egy dokumentum megsűrésére a tartalomjegyzék előállítás céljából; több, különböző adatforrás összekapcsolására adatértékek alapján (join); csoportosításra és aggregálásra; valamint szekvenciális kapcsolatok lekérdezésére dokumentumokban. Vállalatoknál különösen az összekapcsolási lehetőség a fontos. A DataQuery nyelve több forrásból származik (SQL, OQL, OCL). Az SQL-t ismerő fejlesztők a DataQuery-t könnyen megtanulhatnak fogják találni.

Az adatkéréseket előre el lehet készíteni DataQuery-ben, és külső alkalmazásokba beágyazva végrehajtatni. De a felhasználó maga közvetlenül, interaktívan is megadhatja őket, modelltárházban tárolhatja, azonnal végrehajthatja, vagy a végrehajtásukat a későbbiekre halaszthatja.



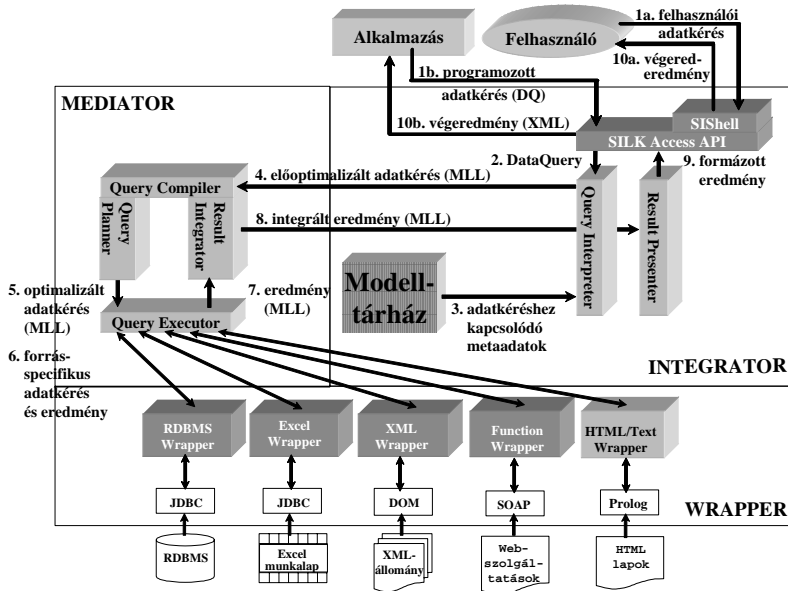


5. ábra: Az interaktív lekérdezés felhasználói szempontból

Az adatkérések végrehajtásának lépéseit a SILK-ben a következőképpen lehet összegezni (ld. 6. ábra):

- lépés:** Adatkérés megadása közvetlenül a *felhasználó* által a *SIShell* grafikus funkciói segítségével vagy programozottan *alkalmazásból*.
- lépés:** A végrehajtandó adatkérés összeállítás DataQuery formátumban a *SILK Access API*-n keresztül.
- lépés:** Az adatkérés ellenőrzése, a kapcsolódó metaadatok meghatározása és összegyűjtése a modelltárházból a *Query Interpreter* segítségével
  - az adatkérésbeli hivatkozások ellenőrzése a kiindulási modellekből,
  - a kiindulási és a kapcsolati modellek összekötése a leképezéseken keresztül.
- lépés:** Az adatkérés előzetes optimalizálása a *Query Interpreter* segítségével
  - alternatív DataQuery-változatok előállítás és értékelése,
  - felhasználói preferenciák/útmutatások figyelembe vétele,

- a DataQuery lefordítása a *Mediator* belső ábrázolási módjára (Mediator Logical Language, MLL)
5. **lépés:** Az adatkérés optimális végrehajtási tervének előkészítése és kialakítása a *Query Compiler*-ben, ill. a *Query Planner*-ben
  6. **lépés:** Az optimalizált adatkérés végrehajtása a *Query Executor* által
    - az érintett adatforrásokra specifikus lekérdezések összeállításával,
    - a megfelelő Wrapper-ek meghívásával,
    - szükség esetén az adatkérési terv újratervezése a *Query Planner*-rel.
  7. **lépés:** Az egyes, adatforrásokra vonatkozó specifikus lekérdezésekre kapott válaszok összegyűjtése a *Query Executor*-ban
  8. **lépés:** Az eredmények átalakítása és összesítése a *Result Integrator*-ban a kiindulási adatkérés eredeti szintjére, az adatkérésben hivatkozott kiindulási modellek fogalmaiban
  9. **lépés:** Az eredményadatok megjelenítésre történő előkészítése a *Result Presenter*-ben
    - sorba rendezéssel, egyediségre vonatkozó szűréssel,
    - szükség szerinti aggregálással (pl. szumma, min, max, átlag, számosság) valamint
    - a kívánt formátumba való (XML, Excel, szöveg stb.) átalakítással.
  10. **lépés:** A végeredmény
    - átadása az adatkérést kibocsátó alkalmazásnak a *SILK Access API*-n keresztül, vagy
    - megjelenítése a *SIShell* grafikus funkcióival (egyszerű táblázatos formában, adatspecifikus „nézőké”-vel, vagy formázott szöveggént).



6. ábra: A lekérdezés folyamata a SILK-ben

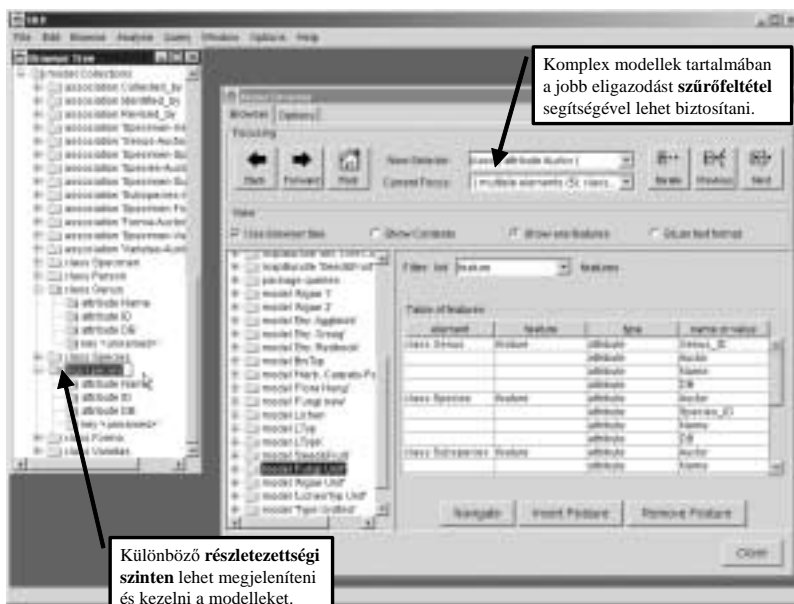
### 3. A SILK értékelése EII-szempontról

#### 3.1 Jellemzők

*Univerzális adathozzáférés* — Különböző fordítók és optimalizált lekérdezések segítségével a SILK relációs adatbázisokból, Excel-fájlokból, XML-fájlokból, szöveges állományokból, HTML-lapokból (Web-helyek) és egyéb forrásokból képes adatokat kinyerni mind a vállalat tűzfalán belülről, mind azon túlról. Minden olyan (nagygépes, kisgépes, kliens-szerver vagy elektronikus üzletviteli) alkalmazás, amely képes — vagy képessé tehető — XML-formátumban közzé tenni az adatait, bekapcsolható adatforrásként.

*Grafikus fejlesztési eszköz* — A SILK grafikus tervező eszközt biztosít a fejlesztőknek a források sémáinak gyors vizsgálatára, átvételére, a szükséges adatelemek azonosítására és az adatnézetek különböző szintű modellek formájában való megadására (ld. 7. ábra). Ezen

túlmenően képes más grafikus modellező eszközöktől (pl. Rational Rose) is átvenni adatmodelleket szabványos XML Metadata Interchange formátumon keresztül.



## 7. ábra: Grafikus fejlesztési lehetőségek a SILK-ben

*Újrafelhasználható és rugalmas adatnézetek* — Az ügyfelekre, megrendelésekre és készletekre vonatkozó információkra újra és újra szükségük van a vállalat különböző csoportjainak. A SILK-kel az adatok különböző nézeteit — fogalmi szintű modellek formájában — igény szerint lehet kialakítani, és lehet újra felhasználni. Emellett a SILK-kel az egyes nézetek könnyen módosíthatók, amikor új adatforrásokat kell hozzáadni (a *Model Comparator* használatával) a meglévőkhöz.

**Lekérdezés optimalizálása** — A felhasználók lekérdezés-specifikus útmutatásainak elfogadásával és követésével, az egyes adatforrások lekérdezési lehetőségeinek maximális kihasználásával és alternatív lekérdezési utak feltérképezésével a SILK komplex teljesítményoptimalizálási lehetőségekkel rendelkezik.

### 3.2 Előnyök

*Az informatikai projektek hasznosulásának felgyorsítása* — A fejlesztők részére biztosított magas szintű lekérdezési nyelv és a kapcsolódó vizuális eszközök segítségével több hónapi egyedi kódolási munka helyettesíthető egyszerű, ismerős lekérésekkel.

*Az integrációs költségek csökkentése* — Nemcsak a kódolási igény csökken, hanem az adatnézetek újrafelhasználhatók és könnyen aktualizálhatók lesznek. Ez lehetővé teszi, hogy a változtatásokat gyorsan és olcsón lehessen elvégezni.

*Hozzáférés biztosítása a valós idejű információkhoz* — Az üzleti vezetők számára közvetlen és késedelem nélküli hozzáférést biztosít az aktuális adatokhoz, hogy azok megfelelő információk birtokában hozzák meg döntéseiket.

*Biztonságos hozzáférés* — A vállalatoknak gyakran van szükségük adatokra partnereiktől, beszállítóiktól és ügyfeleiktől külső hálózaton keresztül. A nyílt hozzáférés lehetőségét azonban egyensúlyban kell tartani a biztonság és az ellenőrizhetőség szempontjaival, hogy az üzleti hálózatok szereplői megvédhessék kritikus információforrásaikat. A SILK biztosítja, hogy a különböző nézetekhez (azaz modellekhez) a hozzáférést felhasználók és alkalmazások szerint egyedileg meg lehessen adni, és így szabályozni lehessen azt.

## 4. Összefoglalás

Az elektronikus üzletviteli architektúrák és a valós idejű műveletek térhódítása a nagy, a közepes és a kis méretű vállalatok körében egyaránt eddig nem tapasztalt új kihívásokat jelent arra nézve, hogy információforrásaikat gyorsan, pontosan és intelligens módon kapcsolják össze. Habár a mai információs architektúrák sok eleme rendelkezik fontos, alkalmazási szintű integrációs képességgel, ezek a megoldások mindeddig egy alapvető láncszemet nem tartalmaztak: az *adatintegrálást*.

Lehetővé téve a Web-alkalmazásoknak és a felhasználóknak, hogy közvetlenül, biztonságosan és valós időben férjenek hozzá a vállalati és a vállalatközi adatokhoz, a SILK-rendszer lecsökkentheti a fejlesztési időt és költségeket, felgyorsíthatja az alkalmazások fejlesztését, és ezzel biztosíthatja a vállalatok számára, hogy a szoftverinfrastruktúrába és az alkalmazásokba történő befektetéseik után kellő megtérülést érjenek el.

Az IQSoft Intelligens Software Rt. (a KFKI Csoport tagja) *IQ\*SILK* elnevezésű, új *adatintegrációs technológiája* a SILK-projekt eredményeinek terméksztintű változata. Az eszközkészlet rugalmassága és sokirányú hasznosíthatósága lehetővé teszi, hogy egyrészt

költséghatékony megoldást adjon a kis- és közepes vállalatok alapszintű integrációs problémáira, másrészt fontos, kiegészítő szerepet kapjon a nagyvállalatok komplex alkalmazásintegrációs stratégiáiban.

### **Irodalom- és hivatkozásjegyzék**

- [1] Morgenthaler, G.: Data Base Newsletter, 1997. május-június
- [2] Date, C.: Business Rules and the Relational Model, Business Rules Journal, 2. kötet, 10. szám, (2001. október),  
URL: <http://www.BRCommunity.com/a2001/b087.html>.
- [3] OMG: Unified Modeling Language Specification. 1999.
- [4] Fensel, D. et al.: OIL in a Nutshell. Knowledge Acquisition., Modeling and Management, 1-16 old., 2000.3
- [5] Jafiar, J. – Michaylov, S.: Methodology and Implementation of a CLP system. Logic Programming - Proceedings of the 4<sup>th</sup> International Conference. Editor: J.L. Lassez, 1. kötet, MIT Press, Cambridge, MA, 1987.5
- [6] Badea, L. – Tilivea, D.: Query Planning for Intelligent Information Integration Using Constraint Handling Rules. IJCAI-2001 Workshop on Modeling and Solving Problems with Constraints, 2001.
- [7] SICS: SICStus Prolog Manual, 2001. április