

Médiatartalom- kezelő rendszerek

Következtető-gép megoldások a szemantikus Web logikai rétegében

Készítették:

Fodor Bálint
Halányi Ferenc
Paróczy Zsombor
Porohnavec József

S02ZLO
RQNG93
WRVU7O
T715ZR

TARTALOMJEGYZÉK

Feladat ismertetése	3
Elméleti alapok.....	3
A szemantikus Web.....	3
Tudásmenedzsment és tudásreprezentáció	4
Ontológiák.....	7
Ágensek.....	8
Adatközpontúság	9
Logika	10
Következtető gépek vizsgálata.....	12
Wonderweb	12
EULER	12
FOWL.....	13
OWLP.....	13
Celebra	13
Racer.....	14
Pellet	14
Hoolet.....	14
Tesztelés.....	15
Távlati kitekintés	20
Az optimista	20
A pesszimista.....	22
Konklúzió	24

FELADAT ISMERTETÉSE

A félévi feladat során a szemantikus Web logikai rétegében működő következtető-gép megoldásokat vizsgáljuk meg és teszteljük őket.

ELMÉLETI ALAPOK

Gottdank Tibor: Szemantikus web (ComputerBooks Kiadó 2006) című könyve alapján mutatjuk be a szemantikus web koncepcióját és alapjait a következő részben.

A SZEMANTIKUS WEB

A mai Web adatokat közöl, ezek formája: természetes nyelvek, képek, multimédiás eszközök, oldalszerkezet. Az ember számára az értelmezés, a feldolgozás egyszerű, akár több forrásból származó információkból vagy részinformációkból is képes tényekre következtetni, összefüggéseket tud alkotni. Kérdés, hogy hogyan lehetne mindezt automatizálni.

A következő példák rámutatnak a szemantikus Webbel kapcsolatos motivációkra és egyben kijelölik a mai Web továbbfejlesztendő területeit:

- Automatikus kiszolgáló: Személyes (digitális) kiszolgálók segítenének nekünk.
 - Ismerik a preferenciáinkat, igényeinket, szokásainkat.
 - Képesek létrehozni egy tudásbázist, felhasználva korábbi történéseket, információkat.
 - Kombinálni tudják a helyi tudást a *távoli* szolgáltatásokkal (szállásfoglalás, repülőjegy-rendelés, étrend, orvosi tanácsok, napirend stb.)
- Adatbázisok integrálása: Az adatbázisok eltérnek egymástól szerkezetben és tartalomban. Sok alkalmazásnak szüksége van többféle adatbázis menedzselésére. Ezen adatok legtöbbje megtalálható a Weben, csak az adatok, adatbázisok szemantikáját, értelmét kellene tudni.
- Digitális könyvtárak: Hasonlóak a Webes keresőkhöz, nem mások, mint Webes katalógusok, melyekkel a következőket lehet megvalósítani:
 - Internetes könyvtárak létrehozása.

- Adatok kiterjesztése a multimédiás adatok területére.
- Szoftver ágensek lehetnének a könyvtárosok: segíthetnek a megfelelő dokumentum, publikáció megtalálásában.
- Webszolgáltatások szemantikája: Szükség van a szolgáltatások karakterizálására, mégpedig nem csupán input és output paraméterek szintjén, hanem azok szemantikai szintjén is. Például egy szolgáltatás *elegáns* levezetést keres a Heisenberg-féle határozatlansági összefüggésnek, majd keres egy másik *elegánsabbat*. Itt szükség van az *elegáns* és *elegánsabb* fogalmak definiálására is, nem csak arra, hogy az összefüggés levezetését találja meg.

Amire szükségünk van: forrás, amely saját magáról ad információt. Ezt hívjuk metaadatnak. A metaadatnak gép által értelmezhető formájúnak kell lennie. Az ágensnek pedig következtetéseket kell levonnia erről a metaadatról, ami tulajdonképpen számára adat. Ehhez metaadat-szókészletet kell definiálni.

Szükség van a következőkre:

- Egyértelmű forráselnevezés (URI)
- Közös adatmodell a metaadat kifejezéséhez (RDF)
- Módszer a metaadat Weben történő eléréséhez
- Ontológiák (szókészlet + köztük lévő kapcsolatok + szókészleten végezhető műveletek + axiómák, következtetési szabályok)

A szemantikus Web pontosított definíciója így: metaadat-alapú infrastruktúra Weben történő következtetéshez.

TUDÁSMENEDZSMENT ÉS TUDÁSREPREZENTÁCIÓ

A tudásmenedzsment az az eszköz, amellyel a strukturált, félig strukturált és strukturálatlan információt szerves egésszé szervezhetjük, s a tudásból pontosan azokat az elemeket jeleníthetjük meg – mégpedig abban az összeállításban és formában -, amelyekre egy adott döntés meghozatalához, illetve valamely tevékenység előkészítéséhez szükség van.

A jelenlegi tudásmenedzsment rendszereknek 4 jelentős gyengéje van:

- **Információkeresés:** A jelenlegi kulcsszó alapú keresések olyan lényegtelen információkat szülnék, melyek bizonyos kifejezéseket eltérő értelmű formában tartalmazzák. Az információkinyerés hagyományosan adott lekérdezés és az információkészlet közötti viszonyra figyel. A kiválasztott információrészek közti kölcsönös kapcsolat kiaknázásával (melyet az ontológiák használata segíthet elő) a másképpen izolált információt jelentéssel bíró kontextusba tehetjük. Az implicit struktúrák így sokkal hatékonyabban fognak segíteni a felhasználóknak az információ használatában és menedzselésében. A jegyzetben említett konferenciához kapcsolódó repülőjegy keresés tökéletes példája ennek.
- **Információkinyerés:** Jelenleg emberi böngészésre és olvasásra van szükség a forrásokból történő érdemi információ kinyeréséhez. Az automatikus ágensek nem képesek feldolgozni az olyan általános ismereteket, amelyek elengedhetetlenek az információk szövegalapú reprezentációjából való kinyeréshez, és nem képesek különböző forrásokon keresztül elosztott információkat integrálni sem.
- **Karbantartás:** A szerényen strukturált szövegforrások karbantartása nehéz és időigényes tevékenység. Ezen kollekciók konzisztensnek, pontosnak és mindig frissnek tartása szükséges a szemantikák gépiesített reprezentációjához. Ez utóbbival ugyanis kideríthetőek az anomáliák.
- **Automatikus dokumentumgenerálás:** Ezáltal a felhasználói profilhoz és egyéb fontos aspektusokhoz igazítva létrejöhetnek olyan weboldalak, melyek dinamikusan újrakonfigurálhatók. A múltkori prezentációnkban említett meseíró is például ennek megfelelően működik

Azokat a tárgyköri ismereteket, amelyeket egy feladat megoldása során felhasználunk, alapvetően két kategóriába lehet sorolni. Az elsőbe tartoznak a vizsgált világ összefüggéseit leíró elsődleges vagy lényegi ismeretek, míg a másodikba a következtetés hatékonyságát növelő heurisztikus ismeretek.

Tudásreprezentáció alatt a lényegi ismeretek ábrázolására gondolunk. Minden tudásreprezentációs módszer sajátos ábrázolási lehetőséget ad ezeknek az ismereteknek a kifejezésére. Minden ábrázolási módszerhez lényegileg hozzátartozik egy következtető eljárás, amely úgymond „életre kelti” a reprezentált ismereteket.

Fontos megjegyezni, hogy a tudásreprezentáló rendszerekkel gyakran limitálják azoknak a feltehető kérdéseknek a körét, amelyekre egyáltalán válaszolni lehet. A probléma visszavezethető Gödel tételéhez: bármely rendszerben, amely elég komplex, mindig találunk megválaszolhatatlan kérdéseket, a rendszeren belül megfogalmazható összes igazságot nem tudjuk levezetni. A paradoxonok és a megválaszolatlan kérdések jelentik annak az árát, amit a sokoldalúság elérése érdekében kell megfizetni.

A szemantikus Web kihívása egy olyan nyelvezet szolgáltatása, mely egyaránt tud adatokat és szabályokat kifejezni. Adatban gondolkodva lehetővé válhat adott szabály kinyerése bármely létező tudásreprezentáló rendszerből, és exportálható lesz az a Webhez.

A szemantikus Web képessé teszi a gépet arra, hogy megértsék a szemantikus dokumentumokat és adatokat, de az emberi beszéd és írás megértésére nem.

A szemantikus Web egyik alapechnológiája az RDF (Resource Description Framework), mely egy olyan adatmodell, aminek szemantikája gráfokkal írható le. Az RDF által történő jelentéskifejezés ún. hármasok csoportjába kódol. Minden egyes hármas egy elemi mondat alanyát, állítmányát, tárgyát szimbolizálhatja. Az RDF-et kifejezhetjük XML tagek alkalmazásával, de ez nem kötelező (bár a W3C eddig csak az RDF/XML-t dolgozta ki).

A mondatok alanyát és tárgyát az URI úgy azonosítja, mintha linkek lennének egy weboldalon. Az állítmányokat szintén URI-ken keresztül lehet identifikálni, így lehetővé válik új kifejezés, új állítmány definiálása. Az URI-k segítségével adott RDF dokumentum kifejezései nem korlátozódnak csak a dokumentum kifejezéseire, hanem olyan információkhoz is köthetők, amik a weben találhatóak meg.

ONTOLÓGIÁK

Egy programnak, mely információkat használ fel adatbázisokból, tudnia kell, hogy az alkalmazott fogalmak ugyanazt a dolgot jelentik. Ideális esetben a programnak rendelkeznie kell olyan módszerrel, mellyel fel lehet kutatni az adatbázisok belső jelentéseit.

Erre ad megoldást az ontológiák használata. Az ontológia egy formális, explicit leírása egy elosztott koncepciónak. Az „explicit” azt jelenti, hogy a fogalmak típusai használat alatt állnak, és felhasználásuk megszorításai egyértelműen, explicit módon definiáltak. Az „elosztott” azt fejezi ki, hogy egy ontológia olyan belső tudást foglal magába, melyet nem szűkítenek le adott egyének, viszont egy adott csoport elfogad.

A webbel összefüggő legtipikusabb ontológiatípusnak egy taxonómiája vagy osztályzása, valamint egy következtető szabálycsoportja van. A taxonómia objektumok osztályait és a köztük lévő relációkat határozza meg.

Képesek vagyunk óriási mennyiségű reláció kifejezésére azáltal, hogy osztályokhoz rendelünk tulajdonságokat, és lehetővé tesszük az alosztályoknak, hogy örököljék ezeket a tulajdonságokat.

Az ontológiák következtető szabályai további képességeket nyújtanak. Pl.: „Ha a *városkód* összefüggésben van a *megye kódjával*, és a *cím* használ *városkódot* akkor a *cím* tartalmazza a *megye kódját*.”

Az ontológiaalapú eszközkörnyezet három fő területet fog át:

- Ontológiák keresése és azok (nagy mennyiségű) adatokkal történő összekapcsolása. Ezt a folyamatot automatizálni kell az információkinyerő és a természetes nyelvfeldolgozó technológiák alapján. A minőség érdekében ehhez a folyamathoz szükség van az emberre, aki az ontológiaszerkesztőkön keresztül felépíti és kezeli az ontológiákat.
- Ontológiák és példányaik tárolása és karbantartása.
- Szemantikailag támogatott információforrások lekérdezése és böngészése.

ÁGENSEK

A szemantikus web igazi előnye akkor mutatkozik meg, amikor olyan programokat készítünk, melyek különféle forrásokból gyűjtik egybe a webtartalmat, feldolgozzák az információt és kicserélik az eredményeket más programokkal.

Az ágens jól meghatározható határokkal és interfészekkel rendelkező, egyértelműen azonosítható problémamegoldó egységek.

Jellemzői:

- Egy adott környezetbe ágyazva működik.
- A környezetük állapota szerint érzékelőiken keresztül bemeneti adatokat kapnak.
- Végrehajtóeszközeiken keresztül változtatják környezetüket.
- Adott célok elérése érdekében tevékenykednek.
- Autonóm módon irányítják belső állapotukat és tevékenységüket.
- A célok elérése érdekében rugalmas problémamegoldást alkalmaznak, azaz a környezet megváltozására időben reagálnak, és későbbi célok elérése előrelátóan felkészülnek.

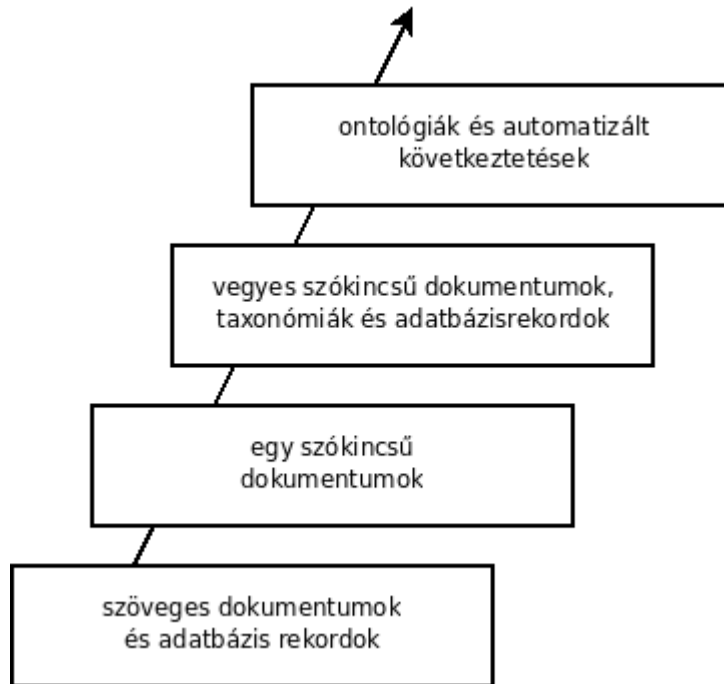
A szemantikus web elősegíti az összhangot az ágensek között. Az ágens működésének fontos része a „bizonyítékok” cseréje, melyet a szemantikus web belső nyelvével írhatunk le. Az ágenseknek mindaddig „szkeptikusnak kell lenniük” amíg az általuk olvasott állításokkal szemben, amíg saját maguk nem ellenőrizték az információforrást.

Itt kap lényeges szerepet a digitális aláírás. Ennek segítségével hitelesíthetik egymást az ágensek és megbizonyosodhatnak pl. olyan dolgokról, hogy „valóban tartozom pénzzel egy szolgáltatónak”.

ADATKÖZPONTÚSÁG

Ahhoz, hogy adataink sokaságát gépek hálózata képes legyen feldolgozni az adatokban való gondolkodásban változtatásra van szükség.

Ennek szemléltetését szolgálja az alábbi ábra:

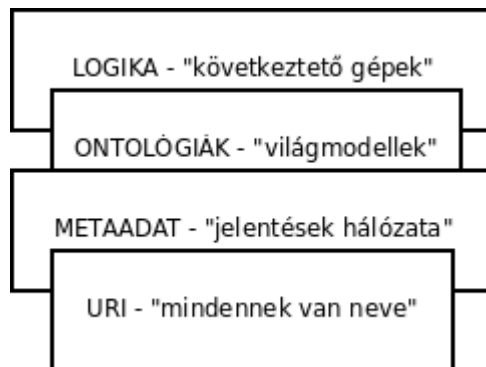


1. Szöveg és adatbázisok: Az adatok többsége kizárólag egy alkalmazáshoz tartozik. Az „intelligencia” nem az adatban („smart data”), hanem az alkalmazásban van.
2. Dokumentumok egyetlen tartományban, doménben: Az adatok függetlenek az alkalmazásoktól az adott értelmezési tartományon belül. Az adat elég „intelligens”, hogy alkalmazások közt mozgatni lehessen a domén keretein belül.
3. Taxonómiák és dokumentumok vegyes szókinccsel: Itt az adat több tartományból jön létre és meghatározott pontossággal van osztályozva az adott hierarchikus taxonómiában. Az osztályozást fel lehet használni az adatok felkutatásához. A taxonómián belüli kategóriák közti egyszerű kapcsolatokat fel lehet használni az adatok viszonyításához és kombinációjához. Elég „intelligens”, hogy felkutassák és más adattal vegyítsék.
4. Ontológiák és szabályok: Az új adatok a létezőkből alakulnak ki, logikai szabályok

követésével. Az adat leírható konkrét kapcsolatokkal, finom formalizmussal, így egyre atomibb szinten válik lehetővé az adatok kombinációja és rekombinációja.

A szemantikus webben tehát először olyan adatokat kell definiálnunk, melyek az alkalmazásuktól függetlenek, egymásba rakhatók, osztályozhatók és egy nagyobb információs rendszer (ontológia) részét képezik.

LOGIKA



Azok után, hogy leírtuk a webobjektumokat és kapcsolatba hoztuk őket egymással, szükség van a módszerre, mellyel újabb adatokra tudunk következtetni.

A területen a predikátum logika a következők miatt fontos:

- Magasszintű nyelvet ad, melyben a tudás világos módon fejezhető ki.
- Jól értelmezhető formális szemantikát nyújt, mely világos jelentést ad a logikai állításoknak.
- Pontos jelrendszerrel rendelkezik a logikai következtetésre, mellyel meghatározza, hogy egy állítás szemantikailag egy másik állításcsoportból (premisszákból) következik.
- Léteznek olyan hitelesítő rendszerek, melyeknél automatikusan származtatnak állításokat (szintaktikailag premisszákból).
- Léteznek olyan hitelesítő rendszerek, melyeknél a szemantikus logikai következtetés egybeesik a szintaktikus származtatással a hitelesítő rendszeren belül.
- A predikátum logika egyedinek mondható abban az értelemben, hogy alapos és

teljes hitelesítő rendszerek léteznek benne.

- Éppen a hitelesítő rendszer megléte miatt lehetséges a bizonyítás végigkövetkeztetése, mely elvezet a logikai következtetéshez.

A predikátum logika monoton, ami azt jelenti, hogy amennyiben megfogalmazódott egy következtetés, akkor ez érvényes marad akkor is, ha új tudás válik hozzáférhetővé. Monoton szabályrendszer megfogalmazása nem mindig lehetséges adott alkalmazáshoz, így a nem-monoton szabályok kezelésére is módot kell találni.

Amennyiben megengedjük a szabályok közti konfliktust, tehát olyan szabályokat amelyek ellentmondásra vezetnek, úgy a következtetés egyértelmű meghozatalához a szabályok közt prioritást kell állítani.

A prioritások a gyakorlatban kerülnek meghatározásra, és a következő alapelvekre támaszkodhatnak:

- Egy szabály forrása sokkal megbízhatóbb lehet, mind a másodiké, vagy magasabb hitelesítési szinten áll. Például az üzleti világban a felsővezetésnek több jogosítványa van mint a középvezetésnek.
- Egy szabály hivatkozható egy másik által, mivel ez sokkal újabb.
- Egy szabályra lehet hivatkozni egy másik által, mivel ez utóbbi sokkal specifikusabb. Egy tipikus példa: egy szabály több kivétellel; ilyen esetekben a kivételek erősebbek, mint az általános szabályok.

KÖVETKEZTETŐ GÉPEK VIZSGÁLATA

WONDERWEB

- **Webcím:** <http://wonderweb.semanticweb.org/> volt az eredeti, a FACT++ címe: <http://owl.cs.manchester.ac.uk/fact++/>
- **Operációs rendszer:** Linux, Mac OS X, Windows
- **Programnyelv:** C++
- **Ki tudtuk próbálni:** Nem sikerült összekapcsolni a Protegé rendszerrel sajnos, elvileg a Protegé egy régebbi verziójával kellene működnie.

A Victoria Egyetem (Manchester) által vezényelt project (2002-2004) ami a szemantikus web logikai rétegében működő következtető gép koncepció használhatóságának bizonyítását tűzte ki célul. A project bár lezárult, a FACT++ OWL alapú következtető gépbe „él tovább”, a FACT++ viszont összekapcsolható a protegével.

A FACT++ DIG interface-en keresztül webserviceként működik, OWL2 ontológiai nyelven leírt adatokon végezhetünk következtetéseket vele.

EULER

- **Webcím:** <http://www.agfa.com/w3c/euler/>
- **Operációs rendszer:** Független (Java nyelven is elérhető)
- **Programnyelv:** Java, C#, Python, Javascript, Prolog
- **Ki tudtuk próbálni:** Igen, beépített teszteseteket is tartalmaz, Protegével összekapcsolható

Az Euler project célja egy N3 („emberek számára olvasható RDF formátum”) alapú állítások és szabályok alapján egy adott kérdésre megkeresni a választ (amit ők „bizonyítéknak” hívnak). Egyik legérdekesebb példa egy útvonalkeresés volt, ahol útvonalak voltak megadva és hogy az útvonalak összekötése tranzitív tulajdonságú. A következtető mechanizmus 262 lépésben talált bizonyítékot arra, hogy két pont között létezik út.

(Állítások: <http://www.agfa.com/w3c/euler/graph.axiom.n3>

Kérdés: <http://www.agfa.com/w3c/euler/graph.lemma.n3>

Bizonyíték: <http://www.agfa.com/w3c/euler/graph.proof.n3>)

FOWL

- **Webcím:** <http://fowl.sourceforge.net/>
- **Operációs rendszer:** Független (Java nyelven is elérhető)
- **Programnyelv:** Java, Prolog
- **Ki tudtuk próbálni:** Igen, elég sok library-t kellett beszerezni hozzá, de tudtuk futtatni

A FOWL egy szabály alapú következtető gép, ami bemenetként Flora-2 nyelven megfogalmazott szabályokat kér. Tehetünk fel kérdéseket is, de ami a legérdekesebb, a rendszer képes magától új szabályokat létrehozni a meglévők alapján és képes rejtett összefüggések feltárására is. Az általunk futtatott mintában „elrejtettek” egy összefüggést, amit a következtető gép felfedezett. A példa látható online is a következő címen: <http://fowl.sourceforge.net/tutorial/t-case5.html>

OWLP

- **Webcím:** <http://projects.semwebcentral.org/projects/owlp/>
- **Operációs rendszer:** Független
- **Programnyelv:** Java
- **Ki tudtuk próbálni:** Nem, a BELL-LABS fejlesztette régen, azóta magára hagyták a projectet, az elérhető verziók működésképtelenek

Üzleti folyamatok OWL nyelvű modellezésére lenne használható, ha működne. Egyedül a Protégé plugin mutatott működési tüneteket, a csapat közös megegyezéssel lemondott a további kísérletezésről ezzel a rendszerrel kapcsolatban.

CELEBRA

- **Webcím:** <http://www.cerebra.com/>
- **Operációs rendszer:** nem tudtuk meg
- **Programnyelv:** nem tudtuk meg
- **Ki tudtuk próbálni:** Nem, a project eredményei/leírása/stb. jelenleg nem elérhető.

Sok publikáció hivatkozik a Celebra rendszerre, nekünk sajnos nem adatott meg a lehetőség, hogy megismerkedjünk vele, ugyanis a félév eleje óta próbálkozunk, de az

oldal „privát módban” működik, mi pedig nem rendelkezünk a megfelelő belépési információval.

RACER

- **Webcím:** <http://www.racer-systems.com/products/racerpro/index.phtml>
- **Operációs rendszer:** Mac OS X, Windows
- **Programnyelv:** nem tudtuk meg
- **Ki tudtuk próbálni:** Nem, a rendszer fizetős.

Szemantikus weben működő következtető és logikai leíró teljes rendszer, aminek üzleti alkalmazásait a portfólióoldal hosszasan részletez. A rendszer maga képes weboldalak által biztosított tudásbázist, vagy lokális állományokat használni üzleti intelligencia vagy döntéstámogatási folyamatokban.

PELLET

- **Webcím:** <http://clarkparsia.com/pellet/>
- **Operációs rendszer:** Független
- **Programnyelv:** Java
- **Ki tudtuk próbálni:** Igen, Protégé és Jana segítségével is tesztelni tudtuk.

A Pellet rendszer az egyik legjobban dokumentált nyílt forráskódú OWL 2 következtető gép megvalósítás. A projectet központossítottan fejlesztik, bőséges dokumentációval rendelkezik, a legtisztább RDF implementációt tartalmazza. A futtatott tesztek közül nekünk a „borválasztó” tetszett a legjobban, ahol egy bor adatbázist kapcsoltunk össze az „ételek és hozzá illő borok” szabályrendszerrel és a rendszer a megadott vacsora paraméterek alapján máris tudott borajánlatot tenni.

HOOLET

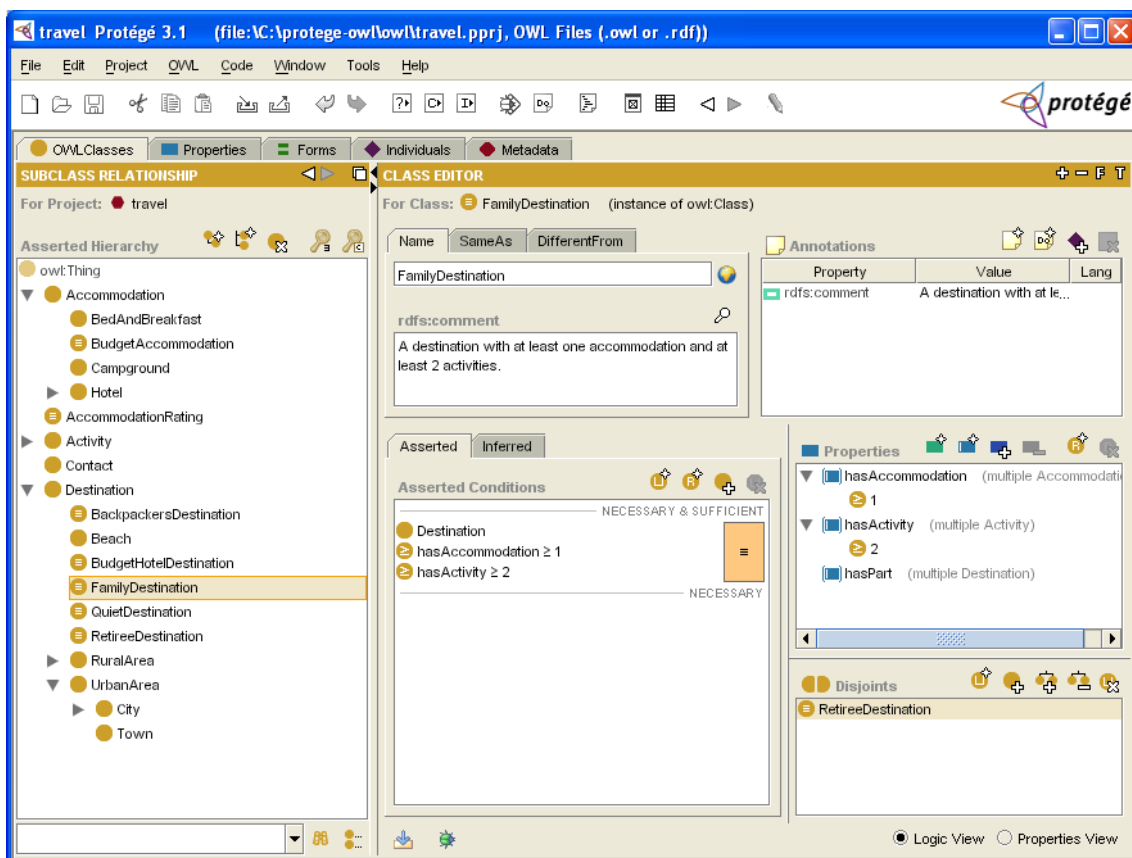
- **Webcím:** <http://owl.man.ac.uk/hoolet/>
- **Operációs rendszer:** Linux
- **Programnyelv:** Nem ismert
- **Ki tudtuk próbálni:** Nem, a project oldalán nem találtunk letölthető változatot.

A Hoolet egy egyszerű Linuxos program a WonderWeb (eredeti, tehát nem Fact++) API-jának és a Vampire következtetőgép összekapcsolásának kipróbálására. A

leírások szerint az összekapcsolás sikeres volt, a program működött. Mivel prototípus alkalmazás, a fejlesztéssel felhagytak.

TESZTELÉS

Csapatunk megpróbált saját példát létrehozni szöveges formában, de egy hét sikertelen próbálkozás után eltekintettünk ettől formától. Ezért a szemantikus web ontológiák készítésénél megszokott Protegé rendszerrel dolgoztunk, saját ontológia és feladat kialakítása helyett egy online tutorialt követtünk végig, ami segít megérteni mind a Protegé használatát, mind pedig az ontológiák készítését. A tutorial elérhető a http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_2.pdf címen.

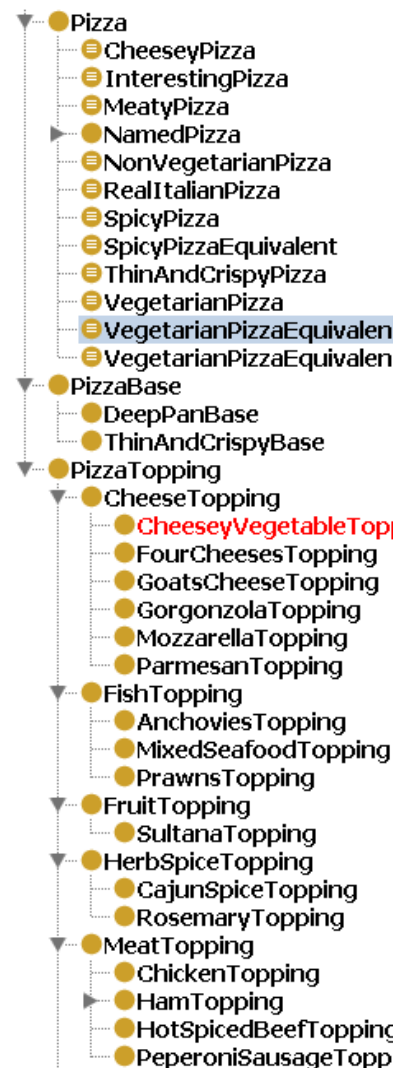


A Protegé a jelenleg elérhető legkiforrottabb owl, rdf, n3 editor, ami képes a következtető gépekkel összekapcsolódni, vagy plugin, vagy webservice segítségével. Az összes, általunk működésre bírt rendszerhez találtunk plugin-t, így mindegyik tesztelni tudtuk. A tesztelés nem hozott semmifajta különbséget a rendszerek között (bár a

Fact++ egy java warning-ot dobott), mivel a pluginek nem csak a kimeneti formát alakítják a megfelelő formára, de az eredményt is parsolják és közös formára hozzák.

A mintapéldában pizzákról, azok tésztáiról és feltétekről szól. A leírást követve létrehoztuk az alapvető osztályszerkezeteket, itt super-subclass viszonyokat definiáltunk, valamint az osztályok kapcsolatát, azok számosságát, típusát definiáltuk. Létrehoztunk két annotációt, a hasBase és a hasTopping, ezen keresztül teremtettünk kapcsolatot a Pizza mint osztály (és mint entitás) és a tészta/feltét között.

Egy pizza mindenképpen rendelkezik valamilyen tésztával (ez PizzaBase osztályú). A tutorialt követve egy, a feltétekhez kapcsolódó „csípősség” (~erősség) mértéket definiáltunk, amit az ontológia osztályhierarchiában tárol, majd ehhez is létrehoztunk egy annotációt, a hasSpiciness. Ezzel a feltéteket csípősség szerinti kategóriákba tudjuk sorolni. Az owl/rdf nyelv központjában az osztályok, entitások és kapcsolatok állnak, tényleges attribútum



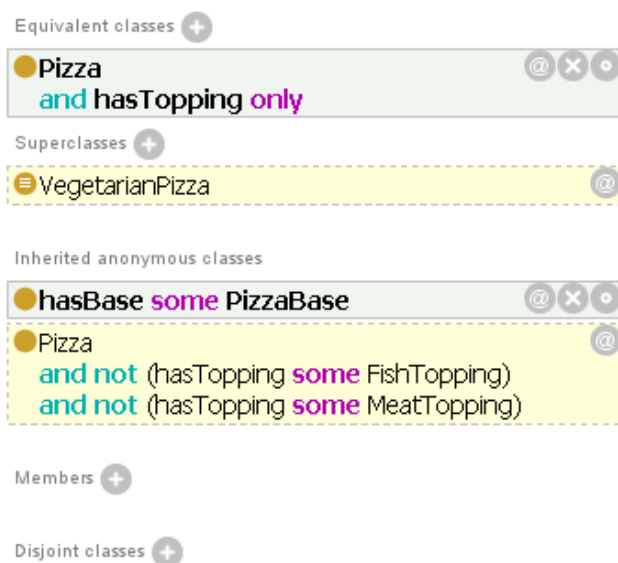
hozzárendelést csak végső esetben végzünk. Áthidalja az attribútum problémát, ha az attribútumokat is osztályoknak fogjuk fel, így hivatkozhatóak, könnyebben kezelhetőek és hierarchiába foghatóak. A csapat ezen a ponton értette meg, hogy a saját elképzelésünket miért nem sikerült egyszerűen ontológiává transzformálnunk. Ez az új személetbeli eltérés olyan, mint mikor valaki funkcionális programozás után deklaratív programozással kezd foglalkozni.

A mintamegoldást tovább követtük, nevesítettük az osztályokat, valamint nevesített ekvivalens osztály példányokat is létrehoztunk, mint például a vegetáriánus pizza.

A vegetáriánus pizza leírása a következő:

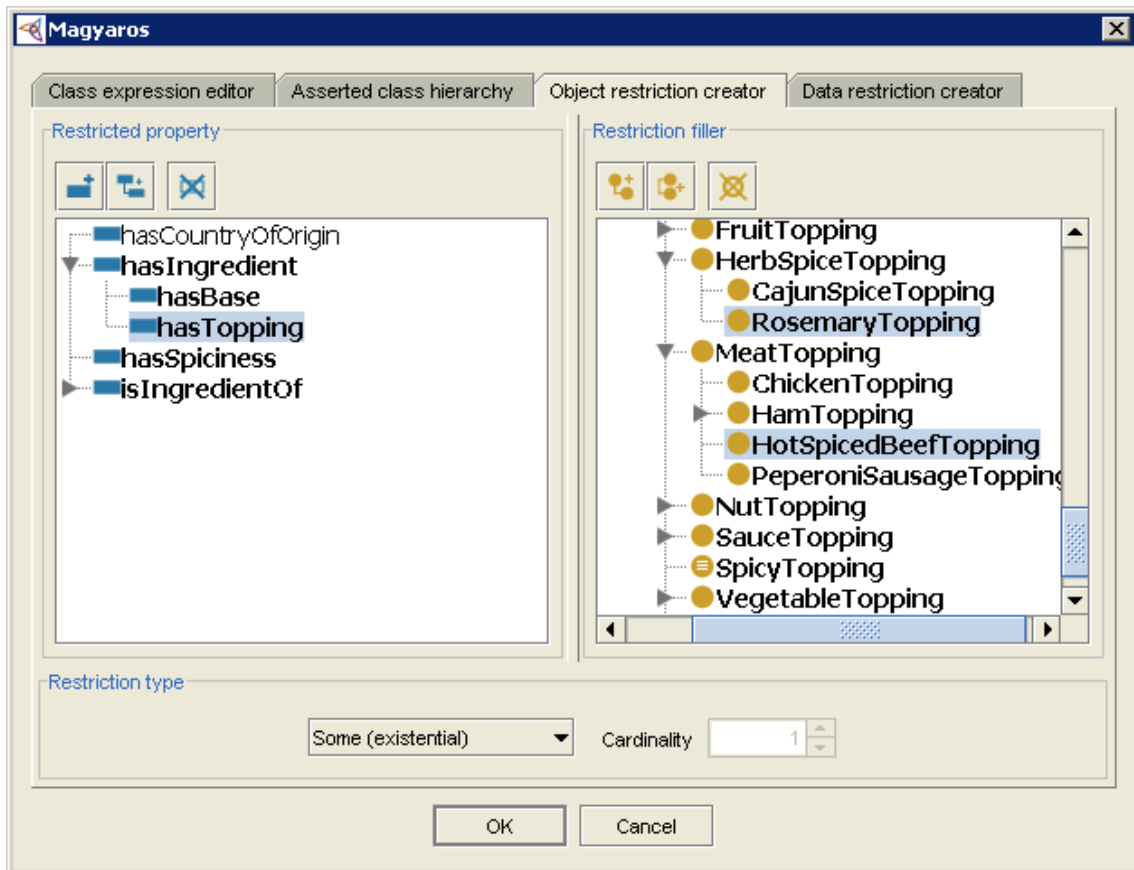
- egy pizza
- csak vegetáriánus feltéttel rendelkezik
- van tésztája
- sem hal, sem hús feltétet nem tartalmaz

Ennek a Protegé felületen látható megvalósulása a következő:



Mint látható, a Protegé egy igen könnyen kezelhető interfészt nyújt ontológiák létrehozására, kezelésére és karbantartására. Több tucat ilyen relációt hoztunk létre, igen sokfajta feltételt fogalmaztunk meg - tutorial lévén szinte az összes alapszintű rdf/owl nyelvi elemmel megismerkedtünk.

Az alábbi ábra például egy új, tutorialban nem szereplő, Magyaros, névvel ellátott Pizza típus létrehozását mutatja, ahol korlátozzuk a feltéteket:



Meglepő módon az ember több időt tölt az interface-en klikkeléssel, mint tényleges értékek/nevek beírásával. Az osztály struktúra is rettenetesen megkönnyíti a feladatot, a leszármazott, örökölt, annotált részek kitöltésével nem kell törődni, a rendszer ezt automatikusan elvégzi.

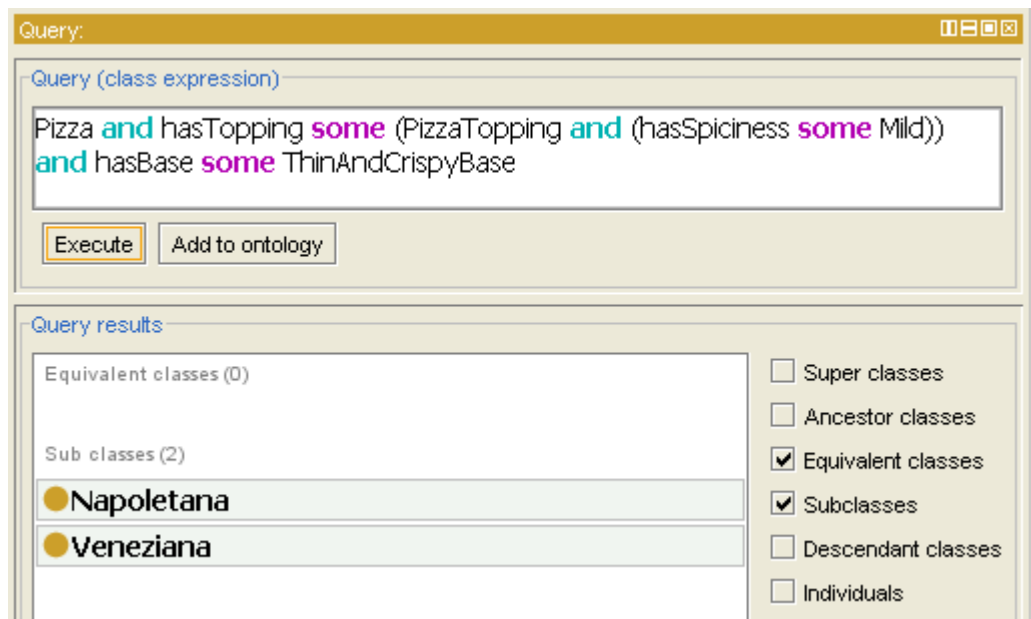
A rendszer lehetőséget nyújt ezen ontológiák következtető gépes lekérdezésére is, itt teszteltük a következtető gépeket – mint már említettük, nem találtunk különbséget a kimenetekben.

Legyen a következő a következtető gép irányába a lekérdezés:

Pizza and hasTopping some (PizzaTopping and (hasSpiciness some Mild)) and hasBase some ThinAndCrispyBase

Azaz, keresünk egy olyan pizzát, ami olyan feltéttel rendelkezik, ami PizzaTopping osztályú és „fűszerességi szintje” lágy, valamint rendelkezik tésztával és a tészta típusa vékony és ropogós.

A következtető gép futásából a felhasználó nem érez szinte semmit, hihetetlen gyorsasággal megadja a rendszer a megfelelő választ.



Azaz két ilyen nevesített Pizza létezik, a Napoletana és a Venziana. A feltét szerinti helyességet ellenőriztük, mindkét pizza 5 feltéttípust tartalmazhat, és csak azokat, viszont ezek opcionálisak, egyet viszont biztosan tartalmaz. (some, only és or relációk)

A tésztára vonatkozó definíciók pedig a következők:

hasBase only ThinAndCrispy

hasBase some PizzaBase

Tehát mindenképpen van tésztája, és az csak vékony lehet. Így a „megoldás” helyes, a következő gép jól működött.

A tutorial 109 oldalas, a csapat több részletben végezte a feladatokat - az ontológiák ugye menthetőek és visszatölthetőek - összegészében körülbelül 10-15 óra alatt készültünk el vele.

visszakeresési képesség részben már most is jelen van, az internet megadja a lehetőséget az esetek 99%-át kielégítő keresésre. A hétköznapi ember számára tehát nem ez lenne a „nagy újdonság”. A már említett példa is rávilágít arra, mennyivel egyszerűbb lenne az élet, hogyha a velünk kommunikálók, vagy interakciót végző tárgyak „ismernének” minket. Ahogy Kevin Kelly¹ kifejtette, ehhez nem kell mást tennünk, mint „transzparenssé” válni, megosztani magunkat a világgal. Minél többet osztunk meg magunkból, annál több hozzáadott értéket tud a világ számunkra biztosítani. Ez a folyamat, a szemantikus webtől függetlenül, már elkezdődött. Több olyan szolgáltatás is létezik, ami bizonyos szokásaink alapján ajánlásokat képes számunkra adni.

A last.fm² például a saját gépünkön történő zenehallgatási eseményeket gyűjti össze, rendszerezi és feldolgozza. Így nem csak kézzel fogható statisztikát kapunk zenei ízlésünkről, de számunkra eddig ismeretlen előadók számaiba hallgathatunk bele, amiket a rendszer „számunkra megfelelőnek” vél. A megfelelésség számítása egy igen egyszerű elven alapul (~klaszterhipotézis): aki olyan előadókat és számokat hallgat mint én, annak zenei ízlése hasonlít az enyémhez. Így az ő általa hallgatott többi, számomra ismeretlen, előadó is nagy valószínűséggel beleillik az én zenei profilomba. Meglepő módon igencsak működőképes a rendszer, természetesen ehhez több millió ember zenehallgatási szokásait kell megfigyelni. A zene világa elég gyorsan változik, de a felhasználók által rendszerbe vitt érték (zenehallgatás) mindig aktuálissá teszi a rendszert. Ez a szolgáltatás például remek példája annak, hogyan lehet (gyakorlatilag) felhasználók által létrehozott tudásbázis segítségével szolgáltatást készíteni.

A Web2-es világban a közösségi megosztás (ismerősi kapcsolatok megjelölése adott szolgáltatáson belül) általános dolog. Szinte minden weboldal lehetőséget ad valamilyenfajta felhasználók közötti kapcsolatrendszer felépítésére. A már említett last.fm szolgáltatás is kínál ilyen funkciót. Képzeljük csak el, hogy egy baráti társaság leül beszélgetni, és - csupán a zenei profiljuk alapján- a legtöbbszörnek megfelelő aláfestő zene szól a háttérben.

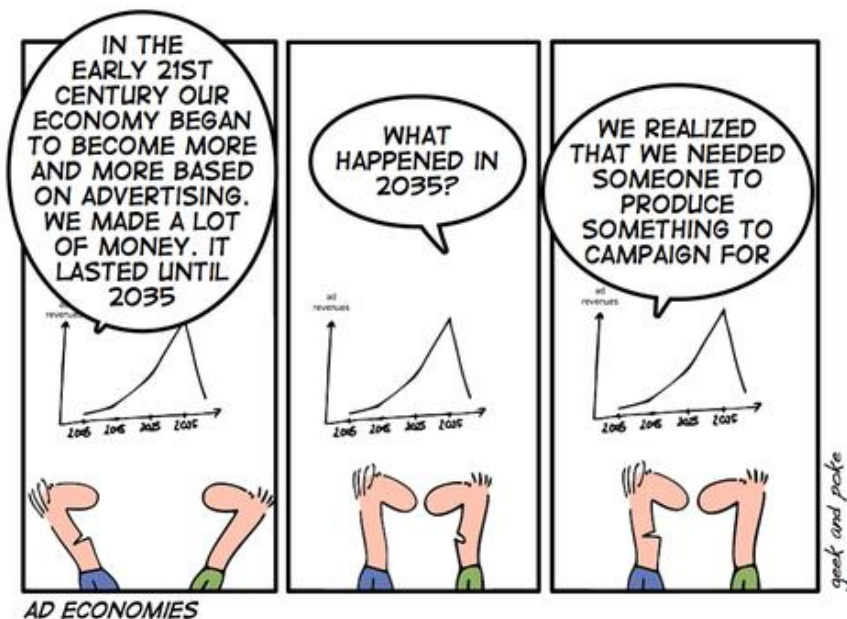
¹ Kevin Kelly: *Predicating the next 5000 days of the* (<http://tinyurl.com/6p83fy>)

² <http://www.last.fm>

Ez már létező technológia, az egyedüli problémája hogy nem univerzális koncepció, mint a szemantikus web. Ilyen és ehhez hasonló rész-szolgáltatások összekapcsolásával, mint látható megteremthető bizonyos szintű personalizáció.

Mint látható, a világ a szemantikus web által lefektetett elveket követi, viszont egy nagy rendszer helyett, több kisebb „szeletét” fedi csak le az életnek.

A PESSZIMISTA



Mi szab tehát gátat a fejlődésnek, a teljes personalizációnak, a világ teljes és kimerítő lefedésének informatikai rendszerekkel és azok szolgáltatásával? Mint minden szolgáltatással szemben, itt is a tradicionális – néha kifejezetten vaskalapos – nézet és a pénz szerepe az igazi korlátozó tényező. Ezen kívül a koncepcióból adódóan felfedezhetjük az embert mint korlátozó tényezőt is, hiszen egy ember nem képes az egész világ tudását – még vázlat szinten sem – rendszerezni, átlátni, globális szabályokat felállítani.

Egy világszintű, összefüggő rendszer létrehozása hatalmas feladat. Gondoljunk csak a rendszer alapjait alkotó fogalom szótárak eltérésére. Minden fogalom, gondolat, de még adateleíró egység alatt is más és mást értünk, és itt nem csak egyéni gondolkodások eltérőségéről van szó, hanem szakmacsoportok és nyelvek eltérőségéről is. Látható, hogy már az alapok szintjén milyen nehéz feladat a szemantikus web koncepcióját teljesíteni.

A szabályrendszerek kialakítása további száz és száz kérdést vet fel. A „modern ember” eddig egyedül a matematika „nyelvén” tudott egységes és zárt szabályrendszert alkotni. A tudomány bizonyos területei is jól kategorizálhatóak, de hogy az egész világot leíró szabályrendszert hozzunk létre, arra vajmi kevés esélyünk van.

A szemantikus web koncepciója jelenleg kisméretben működőképes. Kontrolált adatokon, kontrolált környezetben. Létrehozhatunk teszteseteket, összekapcsolhatunk különböző weboldalakat (mashups), létrehozhatunk hibrid szolgáltatásokat. A teljes webre kibővíteni az elméletet igencsak nehéz lesz.

Vegyük például a jelenlegi webes szolgáltatások, legfőképpen a weboldalak üzemeltetőit. Céljuk legtöbb esetben a profitszerzés, hiszen valamiből elő kell teremteni a működés és fenntartás költségeit, nem is beszélve a munkaköltségekről és a fejlesztésekről. A szemantikus web teljes átjárhatóságot biztosít az oldalak között, adat - információ - szabály hármásban láthatjuk a világot. A piaci szereplőket lehetetlen rávenni az általuk birtokolt tudásbázis megosztására, hogyha ebből ők nem profitálnak. A marketing alapú kapitalista szemlélet pedig nem része a szemantikus web ma ismert koncepciójának. Nem véletlen, hogy Tim Berners-Lee egyik előadásában³ a közönséget a „Linked data” szavak skandalálására bízta. Mert bár jelenleg az üzleti érdek más diktál, meg kell nyitnunk egymás és a világ számára az általunk tárolt/feldolgozott/mért adatokat. Ez az egyedüli módja a szemantikus web (akár részbeni) megvalósulásának.

Másik fontos kérdés minden adat szintű összekapcsolás esetén a megbízhatóság. Mi történik akkor, ha két ellentétes adatforrás két ellentétes információt tárol? Melyikben lehet megbízni, melyik a valós? Mi történik, ha egyszer egy olyan szabálycsoporttal találkozunk, ami teljesen ellentétes a jelenlegi rendszerrel? Melyiket tartjuk valósnak, relevánsnak számunkra. Hasonló eset történt már a fizika történelmében, mikor a svájci szabadalmi hivatal egyik dolgozója megírta „A mozgó testek elektrodinamikájáról” című művet, így írva át a fizikai törvényszerűségek leírásának jelentős hányadát.

³ Tim Berners-Lee: *The next Web of open, linked data* (<http://tinyurl.com/bxua4r>)

KONKLÚZIÓ

Kijelenthetjük tehát, hogy a szemantikus web egy utópia, egy álom, aminek megteremtésére rendelkezésünkre áll szinte minden tudás, minden technikai eszköz, mégsem valósítható meg. Az utóbbi években az egyes szolgáltatók nyitottabbá váltak az ilyen és ehhez hasonló adat és információ kapcsolatok megteremtésére bizonyos szolgáltatások területén. Itt említhető az OpenID vagy a SocialAPI is, de a végeláthatatlan mashup szolgáltatások is. A perszonalizációs törekvéseket sok piaci szereplő is támogatja – Microsoft (Live ID), Google (iGoogle, Youtube, Google Reader, AdSense), Facebook, Yahoo, Tumblr, Flickr, Twitter, Iwiw – megteremtve a lehetőséget a különböző szolgáltatások összekapcsolására.

Csapatunk bízik benne, hogy több ilyen kezdeményezés lesz a közeljövőben, és bár nem teljes egészében, de bizonyos részeiben megvalósulhat a szemantikus web koncepciója.