

MIEO

MAGYAR EGYSÉGES ONTOLÓGIA

összefoglaló szakmai jelentés

NKFP-2/042/04.
3. munkaszakasz
2005. december 1.-
2006. május 31.
projektvezető:
Szakadát István
<http://ontologia.hu>

BME MOKK
BME TMIT
Morphologic Kft.
Scriptum Rt.
ALL Kft.
MTA NYTI
SZTE SZTCS

A 3. munkaszakasz részfeladatai:

- 1.4. távközlési közönségszolgálati tudásbázis konszolidálása II. szakasz
- 1.5. távközlési közönségszolgálati tevékenység gépi támogatása I. szakasz
- 2.5. következtetési rendszer implementálása II. szakasz
- 2.6. az OntoClean módszertan feldolgozása, implementálása
- 2.7. konzisztencia-menedzsment képesség kialakítása – II. szakasz
- 2.8. ontológia-formalizmus kialakítása, fejlesztői útmutató II. szakasz
- 3.2. a távközlési teaurusz felújítása, transzformálása
- 3.3. távközlési közönségszolgálati szakontológia felépítése
- 4.1. a hasznosítható csúcsontológiák előkészítése
- 4.2. az angol nyelvű terminusok fordítása
- 4.3. igei vonzatkeret
- 4.4. csúcsontológia felépítése – I. szakasz

Budapest, 2006. június 8.

Bevezető

A 3. számú projektjelentésben beszámolunk arról, hogy a Magyar Egységes Ontológia (MEO) projekt 3. munkaszakaszában milyen feladatokat végzett el a konzorcium. A jelentés tagolása igazodik a 3. fázis részfeladataihoz (melyeket a jelentés 1. oldalán soroltunk fel).

A közönségszolgálati tevékenység

A projekt rendszerfejlesztési feladatai között szerepel a távközlési közönségszolgálati tevékenység támogatására alkalmas informatikai rendszer megvalósítása. A projekt harmadik szakaszában az 1.4. és 1.5., illetve a 2.5 részfeladatok kapcsolódtak ide. Először is fel kellett építeni a rendszer működtetéséhez szükséges közönségszolgálati tudásbázist (1.4.). Ehhez el kellett kezdeni a közönségszolgálati tevékenység gépi támogatásának fejlesztését (1.5.), melynek során szükség volt következtetési képeségek igénybevételére is (2.5.).

A közönségszolgálati tevékenység központjában az ügyfélszolgálati munkatárs áll (akit a továbbiakban Call Center vagy CC-munkatársnak nevezünk). A CC-munkatárs munkájának gépi támogatásához az alábbi megfontolásokon alapuló rendszert terveztük meg és kezdük el fejleszteni.

Fogalmak, feltételezések

Alkalmazott struktúránk a következő:

- adott Problémáknak egy \mathbf{P} halmaza;
- adott Hibajeleknek egy \mathbf{H} halmaza (diszjunkt a Problémákétól);
- mindkét halmaz részbenrendezett a $<_P$ és a $<_H$ relációk szerint;
- mindkét halmaznak van legnagyobb eleme;
- adott egy $\mathbf{R}: \mathbf{P} \times \mathbf{H} \rightarrow (0,1]$ parciális leképezés;
- adott egy $\mathbf{V}: \mathbf{P} \rightarrow (0,1)$ leképezés;

A részbenrendezések szerepe: ha a $P_1 <_P P_2$, az azt jelenti, hogy a P_1 probléma a P_2 problémának egy speciális esete. Analóg jelentéssel bír a $H_1 <_H H_2$ jelzés is.

Az \mathbf{R} leképezés szerepe: ha $\mathbf{R}(P,H) = p$, annak az a szemléletes jelentése, hogy a P probléma fennállása p valószínűséggel vonja maga után a H hibajel fellépését.

A \mathbf{V} leképezés szerepe: ha $\mathbf{V}(P) = p$, annak az a szemléletes jelentése, hogy a P probléma p valószínűséggel áll fenn egy olyan rendszerben, amiről tudjuk, hogy előfordul benne valamilyen probléma (vagyis kb. annyit tesz, hogy a problémás esetek p részében szerepel P).

A rendszerről az alábbi feltételezéseket tesszük:

- *A problémák egymástól függetlenül állnak fenn egy rendszerben.* Ez matematikailag úgy fogalmazható meg, hogy ha P_1, P_2, \dots, P_k problémák, akkor annak esélye, hogy *mindegyik* fennáll egy rendszerben, $\mathbf{V}(P_1)\mathbf{V}(P_2)\dots\mathbf{V}(P_k)$. Ez ugyan nem teljesen valóság, azonban heurisztikánkhöz elegendő ekkora pontosság.
- *Az okozott hibák egymástól függetlenül lépnek fel.* Ez is egy olyan feltételezés, ami technikailag menedzselhetővé teszi az adatbázis szerkesztését.

Egy adott P problémahalmaz *lezártján* a legszűkebb olyan $<_P$ -re zárt $[P]$ problémahalmazt értjük, ami tartalmazza P -t. Ugyanez igaz analóg hibajelek H halmazának $[H]$ lezártjára is.

(Ábráinkban a hierarchiakat talpon álló háromszöggént szemléltetjük; a háromszög felső csúcán a legáltalánosabb fogalom (probléma vagy hibajel) szerepel, melyet lefelé haladva specializálunk. Nem fogjuk jelezni, de úgy tekintendő, hogy a háromszögek belsejében kijelölt pontok jelképezik az adott szintig specializált fogalmat, mely éllel kapcsolódik más pontokhoz. A felfelé menő éllel a közvetlen fölé-, míg a lefelé menő éllel a közvetlen alárendelttel kötik össze őket.)

A feladat

Feladatunk a következőképp fogalmazható meg: adott hibajeleknek egy H konfigurációja, vagyis egy VH és egy NH hibajel-halmazunk. Ezen halmazok diszjunktak. Jelentésük a következő: VH elemei azok a jelek, melyekről biztosan tudjuk, hogy a felhasználó rendszerében fennállnak; NH elemei pedig azok, melyek biztosan nem állnak fenn. Feladatunk problémák olyan halmazait kilistázni, melyek valamely mérce szerint „valószínű” okozói ezen hibajel-konfigurációnak.

A váz

Megoldásunk a következő lépésekből áll:

- Meghatározzuk a jelen konfigurációra nézve relevánsnak mondható problémák RP halmazát;
- Ezen RP halmaz bizonyos részhalmazaira (egy heurisztikus keresés során) kiszámítjuk egy F fitness-függvény értékeit (mely tehát problémák egy halmazához rendel egy pozitív számot). Ettől az F függvénytől azt várjuk el, hogy minél valószínűbb kiváltó ok egy adott problémahalmaz, annál nagyobb legyen az értéke.

A következőkben az alábbi kérdésekre adunk választ:

- Mely problémák mondhatók relevánsnak egy adott konfigurációban?
- Milyen heurisztikát alkalmazunk arra, hogy a releváns halmaz mely részhalmazait kell vizsgálnunk?
- Hogy definiáljuk az F függvényt, hogy elvárásainknak megfeleljen?

A releváns problémák

Intuíciónknak (is) megfelel, ha azokat a problémákat vesszük legfeljebb számba, melyek az ismert fennálló hibajelek közül legalább egyet okozhatnak. Vagyis a releváns problémák RP halmaza a $[VH]$ halmaznak az R súlyreláció melletti inverz képe lesz (azok a problémák, amikből egy nem nulla súlyú él $[VH]$ -ba), pontosabban ennek lezártja.

Keresési heurisztika

Induljunk ki abból, hogy van egy olyan F fitness-függvényünk, mely céljainknak megfelel és már meghatároztuk az RP halmazt. Kívánatos lenne, ha minél kevesebb problémának az *egyszerre* fennállását kellene megvizsgálnunk (egy valódi hibakeresés során is általában egy-két hibánál többet nem azonosítunk egyszerre). Vagyis, válasszunk egy olyan $mp > 0$ számot, ami azt jelzi, hogy legfeljebb hány probléma egyidejű fennállását lesz képes azonosítani a rendszer (ez a szám előzetes véleményünk szerint 3-4 lehet).

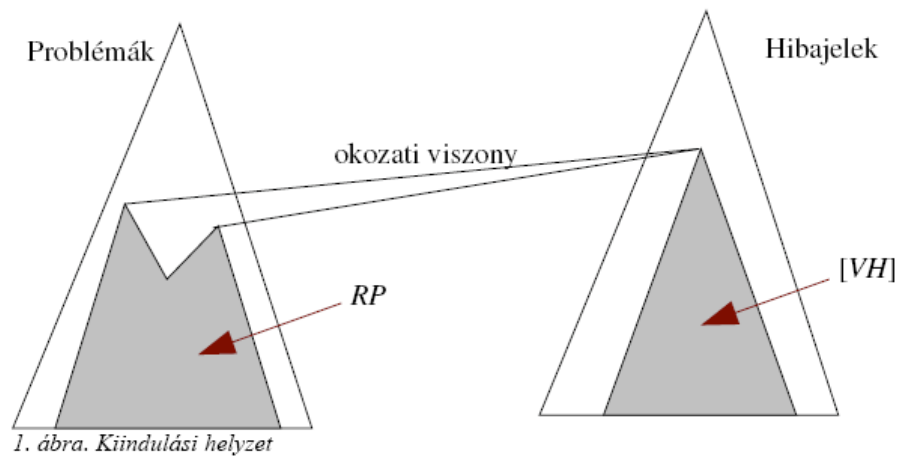
A heurisztika szempontjából szerencsés, ha az RP halmaznak van legnagyobb eleme. Amennyiben tehát nincs, felveszünk egy fiktív elemet, melynek RP összes maximális eleme közvetlen leszármazottja lesz (természetesen ez az elem nem kerül rögzítésre a hierarchiában, csak az aktuális keresés idején szerepel benne).

A keresés egy állapota az RP halmaz elemeinek egy olyan részhalmaza lesz, melynek elemszáma legfeljebb mp . Kezdetben kiindulunk abból az egyelemű részhalmazból, mely csupán RP legnagyobb eleméből áll. Amennyiben egy X részhalmaz a keresés aktuális lépése, annak a gyereke minden olyan X' részhalmaz, mely megkapható X -ből úgy, hogy X egy elemét eltávolítjuk, és helyére valahány (legalább egy) gyermekét beírjuk úgy, hogy $|X'|$ ne legyen nagyobb, mint mp .

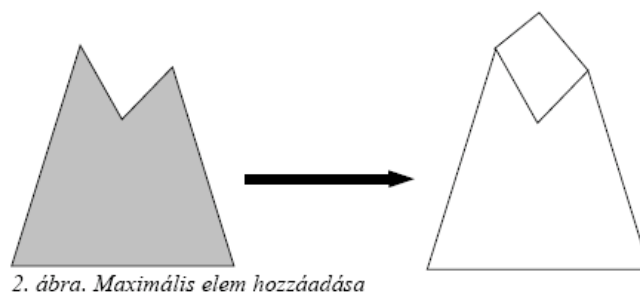
A keresés során minden érintett X részhalmazra kiszámítjuk az $F(X)$ értéket. Amennyiben egy részhalmazra ez az érték legalább olyan jó, mint bármelyik szülőjére, úgy ezt a részhalmazt is kifejtjük; ha pedig nem, úgy lezárjuk. A keresés lefutása után F szerint csökkenő sorrendben kilistázzuk az érintett részhalmazokat.

Példa a keresésre

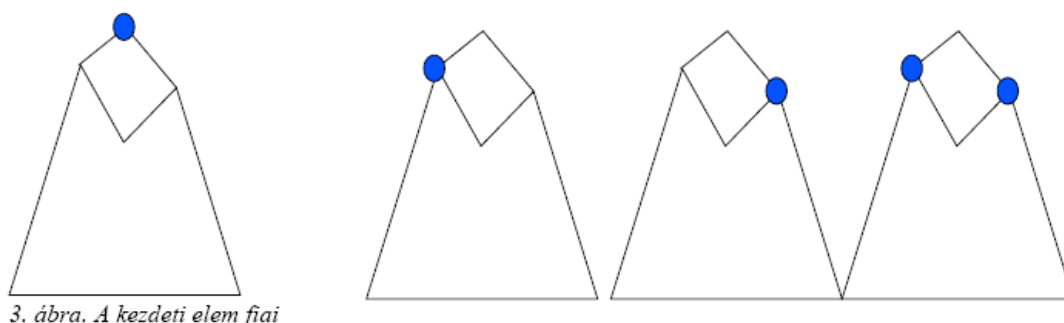
A példában legyen $mp = 2$. Egy részhalmazt úgy ábrázolunk, hogy az elemeinek megfelelő csúcsokat megjelöljük. Legyen a szituáció az alábbi:



Ebben az esetben az RP halmazzal kapcsolatos első tevékenységünk az, hogy hozzáadunk egy maximális elemet.



Ezek után elkezdve magát a keresést, az első lépésben csak a maximális elem szerepel a halmazban. Legyen pl. erre a halmazra az F értéke 1.

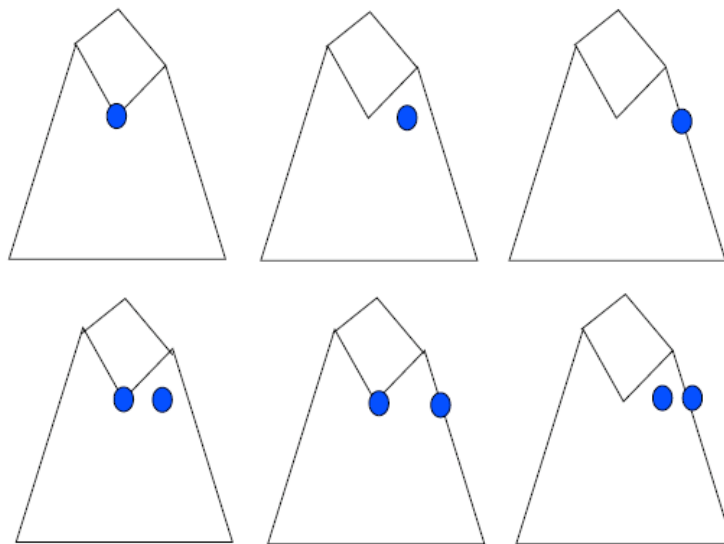


Az első lépés további három vizsgálandó csúcsot hoz be. Tegyük fel, hogy kiszámítva ezekre is az F értékét, az ábra szerinti sorrendben rendre a 0.5, 1.3, 0.8 értékeket kapjuk. Ez azt jelenti, hogy az első és harmadik variáns esetében rosszabb értéket kapunk, mint ahonnan érkezünk, így ezeknek nem kell tovább vizsgálnunk a fiait; a második csúcsot viszont ki kell fejtsük. Legyen a jelölt csúcsnak három gyermeke. Ekkor a kifejtés eredménye:

Vagyis, az egyetlen jelzett csúcsot ekkor hatféléleképp bonthatjuk tovább (mert $mp = 2$). Amennyiben $mp > 2$ állna fenn, úgy még egy esetet jelentene, amikor mindhárom gyermeket megjelöljük.

A kifejtést tovább folytatva, ha az F érték pl. a negyedik unoka esetén jobb, mint a kifejtett fiúnál, úgy mind a bal, mind a jobb jelölt csúcs szerint osztanunk kell tovább. Azonban ekkor az $mp = 2$ érték miatt bármelyiket is „csúsztatjuk lejjebb”, csak egyetlen gyermekét jelölhetjük meg.

Mindezt tehát addig folytatjuk, amíg csak az F érték javulását tapasztaljuk. Látható, hogy az egyszerre aktív csúcsok számára tett mp korlát azért fontos, mert különben nagyon sok csúcshalmaz vizsgálata lenne esetenként szükséges; továbbá egy hibakeresésnél, mint már említettük, természetes megközelítésnek tűnik, hogy egyszerre egy-két problémánál többet nem akarunk (esetleg nem is tudnánk) lokalizálni.



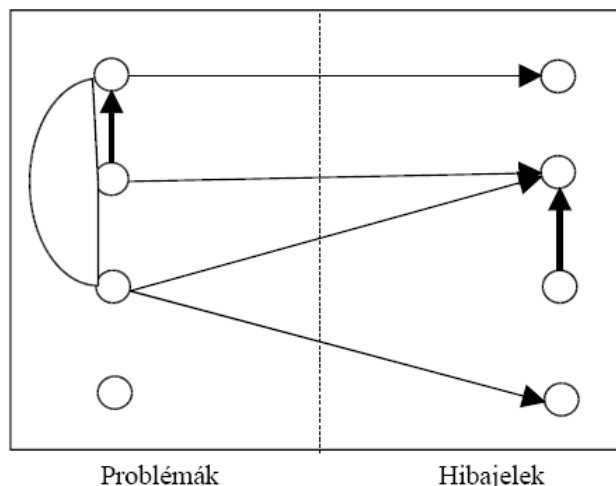
4. ábra. A második fiúból kapott unokák, ha $mp = 2$

Az F függvény

Célunk tehát egy olyan függvényt megadni, mely „jóság”-függvényként lehet használható. Mindenekelőtt lefektetjük a szükséges valószínűség-számítási alapokat, és megvizsgáljuk modellünket ebből a szemszögből.

A modell

Felrajzolhatunk egy következménygráfot, ami vázlatosan a következőképp fest:



5. ábra. Következmenygráf

Tehát az ábra bal oldalán a Problémákat reprezentáló csomópontok foglalnak helyet, jobb oldalon pedig a Hibajeleket reprezentálók.

Minden élnek van egy *súly*, ami egy 0 és 1 közé eső valószínűség, és azt fejezi ki, hogy mennyire vonja maga után az adott él forrás-csomópontjának aktiválása a cél-csomópont aktiválását. Konkrétan:

- Ha két Problémát köt össze egy $A \rightarrow B$ él, az azt jelenti, hogy az A Probléma fennállása valamennyire maga után vonja a B Probléma fennállását is. Abból indulunk ki, hogy a rendszerben lévő hibák függetlenek egymástól (mert egyébként nem lehetne elég számítási kapacitást rendelni a feladathoz). Ez tehát pontosan akkor következik be, ha az A Probléma speciális esete a B -nek. Ekkor viszont B is mindenképp bekövetkezik; így egy olyan élnek, mely két Problémát köt össze, **1** lesz a súly.
- Két Hibajeleket összekötő él esetében megismételhetjük az előző gondolatmenetet, ezen élnek súlya is **1**.
- Végül, a Problémákból a Hibajelek felé mutató élek jelzik azt, hogy a P problémának a H hibajel biztos ($súly = 1$) vagy lehetséges ($0 < súly < 1$) következménye.

A Bayes-tétel

Legyen A és B két esemény. Annak valószínűsége, hogy A bekövetkezik, $P(A)$; annak, hogy B bekövetkezik, $P(B)$. Annak valószínűségét, hogy A bekövetkezik, miközben tudjuk, hogy B bekövetkezik, $P(A | B)$ jelöli.

Ismert a következő összefüggés: $P(A|B) = \frac{P(A|B)}{P(B)}$. Mivel a metszet szimmetrikus, A és B szerepe megcserélhető. Némi átrendezéssel kapjuk a következő, *Bayes-tétel* néven ismert összefüggést: $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$.

A kérdés

Ha tekintünk egy X problémahalmazt és a H Hibajel-együttállást, akkor a fenti formalizmus szerint az X problémahalmaz mint kiváltó ok $P(X|H)$ valószínűséggel jön számításba. A Bayes-tétel szerint tehát a „legesélyesebb” problémahalmaz az az X halmaz, amire a $\frac{P(H|X)P(X)}{P(H)}$ kifejezés értéke maximális. Mivel minden X halmazra a $P(H)$ érték

ugyanannyi, így valójában elég, ha a megadott X -re kiszámítjuk a $P(H|X)P(X)$ szorzat értékét; eszerint rendezve a heurisztikánkkal megjelölt halmazokat egy jó sorrendet kapunk, melyben a leginkább valószínű magyarázat kerül előre.

Vagyis a keresett $F(X)$ függvényünk a $P(H|X)P(X)$ érték lesz.

A $P(X)$ tényező

Fontos, hogy a „minél nagyobb X , annál kisebb $P(X)$ ” ökölszabály érvényesüljön, hiszen ha – elrettentő példaként – minden X halmaznál 1 -re fixálnánk $P(X)$ értékét, akkor – a halmaz méretére járó büntetés elmaradása miatt – a lehető legnagyobb X halmazt adná vissza az algoritmus. Nekünk pedig végeredményben épp az kell, hogy a lehető legpontosabban írjuk le ezt az X halmazt (ami pedig kevés elem együttes fennállása kellene legyen).

Így a „független problémák modellje” mellett döntöttünk: explicite deklarálunk minden P problémához egy-egy valószínűséget (*szigorúan* kisebbet 1 -nél), ami nagyjából tükrözi a problémák előfordulásának gyakoriságát (mivel csak heurisztikáról van szó, itt nem kell pontosságra törekedni), majd a $P(X) = \prod_{i \in X} p_i$ képlettel, függetlenként kezelve a

problémákat, együttes előfordulásuk valószínűségét egyszerűen a valószínűsügek szorzataként kapjuk meg.

A $P(H|X)$ tényező

Kérdés tehát, hogy ha az X problémahalmaz elemei lennének a fennálló problémák az ügyfél rendszerében, akkor mennyi eséllyel kapnánk egy, a H Hibajel-együttállásnak megfelelő jelzéskonfigurációt.

Átfogalmazva a kérdést: arra van szükségünk, hogy a Probléma-Hibajel gráfunkat véletlengráfként kezelve, ha minden élt a rajta szereplő valószínűséggel húzunk be, akkor mekkora eséllyel kapunk olyan gráfot, amire

- minden h ismert fennálló hibajelbe vezet út X legalább egy eleméből (minden fennálló hibajel okoz valami) ÉS
- nincs olyan h ismert nem fennálló hibajel, amibe vezetne út X valamely eleméből (a nem fennálló problémákat nem okozza semmi).

Ennek kiszámításához nyilván elég azt a részgráfot tekinteni, aminek csúcsai és élei szerepelhetnek valamely, az X -ből ismert státuszú hibajelbe vezető irányított úton. Vagyis csúcsai:

- az összes, X -ből elérhető Probléma X' halmaza;

- az összes olyan Hibajel H' halmaza, amiből elérhető legalább egy ismert státuszú hibajel.

Élei pedig azok az élek (súlyaikkal), amik ezek közt futnak.

Mivel rögzítettük, hogy a két pontosztályon belül minden él **1** súlyú, így csak a Problémákból Hibajelekbe futó élek tekintendők valódi véletlen-élek.

Kezdjük először az ismertem nem fennálló hibajelek NH halmazával. Amennyiben egyetlen olyan (p, h) élt is beválasztunk, amire h -ból elérhető NH , nem teljesül egy szükséges feltétel, így ezeket az éleket nem szabad beválasztanunk. Ennek valószínűsége $\prod (1 - q_i)$, ahol a szorzat végigfut az összes ilyen élen (az élek súlyai q_1, q_2, \dots, q_n). Ezeket az éleket töröljük is munkagráfunkból.

A maradék éleket úgy kell beválogatni, hogy minden csúcs elérhető legyen a fennálló hibajelek FH halmazából.

Először kiszámítjuk minden egyes, a részgráfban szereplő Hibajelre, hogy mennyi eséllyel váltja ki őt „közvetlenül” egy Probléma, vagyis hogy milyen eséllyel alakul ki olyan út az élek bedobálása után, aminek az adott csúcs az utolsó csúcsa, és az utolsó előtti egy Probléma.

Ennek valószínűsége, ha az adott csúcsba az X' Problémák oldaláról bejövő éleken lévő valószínűségek p_1, p_2, \dots, p_n , mivel modellünk szerint ezek még mindig függetlenek, $1 - \prod (1 - p_i)$.

Ezek után annak valószínűsége, hogy egy H' -beli h Hibajel fellép, nem más, mint ezeknek a valószínűségeknek egy kombinációja, vagyis ha h elérhető a h_1, h_2, \dots, h_k Hibajelekből, amikre a számított valószínűségek p_1, p_2, \dots, p_k , akkor ismét alkalmazhatjuk az $1 - \prod (1 - p_i)$ képletet (a számítás időigénye javul, ha így csoportosítjuk a számítást).

A keresett $P(H | X)$ érték nem más, mint annak valószínűsége, hogy a jelzett Hibajelek egyszerre bekövetkeznek. Sajnos a hierarchia miatt ezek nem független értékek, így a szita-formulával kell számítanunk a végső eredményt. Azonban (várhatóan) Hibajelből keveset fog jelenteni a kliens.

Összességében a teljes időigénye a $P(H | X)$ mennyiség kiszámításának $O(X' + E + H'2^H)$, ahol a még nem használt E jelenti a P' és H' közti élek számát. Élünk azzal a feltételezéssel (egyelőre), hogy H kicsi – arra építünk, hogy egy kliens ritkán jelent sok hibát egyszerre be. További optimalizálás elképzelhető a biztos következmények külön kezelésével (sőt, ez kívánatos is lenne akkor, ha a korábbi Konfigurációt is Hibajelnek tekintenénk, ez ugyanis jelentősen megnöveli a H értéket).

Úgy döntöttünk, hogy mind a CC-munkatárs, mind az adatfeltöltő szemszögéből nézve megfelelő felületet biztosít a Protégé ontológia-szerkesztő program. Természetesen valóban *szerkeszteni* csak az adatfeltöltőknek lesz engedélye; a CC-munkatárs egy olyan „egyszerűsített” felületet kap meg, melyen csak megnézni lesz képes a tudásbázist (valamilyen nézet szerint).

A részfeladatot bemutató teljes tanulmányunkban ábrák segítségével mutatjuk be, hogy a CC-munkatárs hogyan kezelheti a Protégén keresztül a rendszert, melynek során az alábbi tevékenységekre képes:

- Felviheti az ügyfél által tapasztalt hibajeleket;
- Ezután kérhet egy javaslatot, aminek eredményeképp a rendszer valamennyi lehetséges problémát kilistáz (jelenleg a legvalószínűbb 20 probléma kerül be a listába, amennyiben van ennyi);
- További finomítások és ismételt javaslatkérések után eljut addig, hogy a problémát az ontológia részletességétől függő pontossággal be tudja határolni.

- Teszteket és konfigurációs feltételeket jeleníttethet meg a rendszerrel, melyek végrehajtását kérheti a felhasználótól;
- Ezek alapján problémákat zárhat ki, akár a javaslati listáról közvetlenül a kizárt problémák listájába való áthúzással, akár egy, a hierarchiában feljebb levő (általánosabb) probléma megkeresésével és kizárásával;
- Konfiguráció kizárásával is lehetőség van a szűkítésre, analóg módon;
- Lehetőség van annak deklarálására is, hogy egy adott hibajel nem lépett fel (ezzel a hibajel potenciálisan okozó problémák hátrébb kerülnek a rangsorban);
- Végül megadhatjuk, hogy egy adott konfigurációs komponens (software, hardware, protokoll vagy bármi egyéb) jelen van a rendszerben.

Az alkalmazás működésének szemléltetésére elkészítettünk egy animációt, amely lépésről lépésre bemutatja, hogyan használhatja a közönségszolgálati munkatárs a rendszert az ügyféllel folytatott kommunikáció során a hiba okainak feltárására.

Bővebben: [CC-kov], [CC-anim]

Ontológia-infrastruktúra építés

A projekt egyik kiemelt célkitűzése egy többretegű ontológia-építési módszertan megtervezése és kifejlesztése. A harmadik munkaszakaszban az OntoClean módszertan feldolgozása, implementálása (2.6.), konzisztencia-menedzsment képesség kialakítása – II. szakasz (2.7.), ontológia-formalizmus kialakítása, fejlesztői útmutató II. szakasz (2.8.)

Az ontológiaépítési módszertan kérdései

Az ontológiapítés módszertanával, a konzisztenciavizsgálattal kapcsolatos feladatokat az ontológiaszerkesztő alkalmazásra vonatkozó – többretegű – fogalmi modell, illetve a használati eset modell specifikálásán keresztül valósítottuk meg.

Az OntoClean módszertan

A 2.6. részfeladat az OntoClean módszertan feldolgozását írta elő.

Az OntoClean módszertan céljáról, összetevőjéről, elveiről, megoldásairól, szabályairól egy alapos összefoglaló tanulmány készült. A projekt második szakaszában sikerült mélyebben megérteni a módszertan filozófiáját, fogalomkészletét, a harmadik szakaszban a módszertani tanulmányt úgy alakítottuk át, hogy az útmutatóként, oktatási segédanyagként is használható legyen. Másfelől a MEO-modell kidolgozásakor a rendszertulajdonságok közé belevettük az OntoClean metatulajdonságok kezelésének lehetőségét, így az ontológiaszerkesztési munkák során ezeket figyelembe lehet venni, bár megjegyezzük, hogy a módszertannal kapcsolatban nem is annyira a gépi megvalósítás jelenti a problémát, mint inkább az az elvárás, hogy az egyes fogalmakhoz meg kell adni a metatulajdonságokat. Ez a feladat csak humán közreműködéssel valósulhat meg, és az eddigi vizsgálódásaink (melyek közül sok szerepel az Útmutatóban is) azt mutatják, hogy ez a feladat nem is olyan egyszerű.

Bővebben: [OntoCleanGuide]

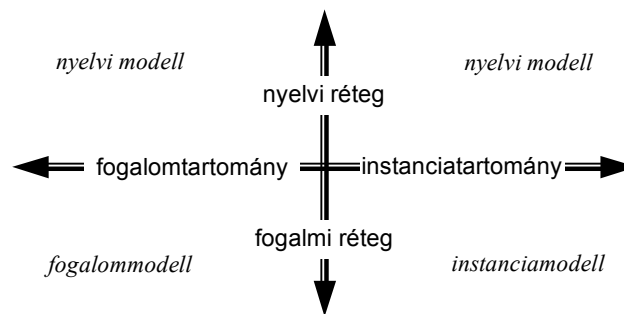
Az OntoClean módszertanból származó metatulajdonságok segítségével vizsgálhatóak a tudásszervezési rendszerek (pl. a teauruszok) elemei közti kapcsolatok erőssége és kompaktsága, és ez az elemzés újfajta szempontnak látszik az ilyen rendszerek bejárhatósága, kezelhetősége szempontjából. Az ilyen szempontú vizsgálat első lépéseként elkészült egy tanulmány, amely a problémákat vetette fel, illetve az elemzési szempontokat gyűjtötte össze.

Bővebben: [Ungváry 2006]

A MEO-ontológia modell rétegei

A 2.7. szakaszban az ontológiaszerkesztés és -használat során érvényesíthető, érvényesítendő konzisztencia-menedzsment képességgel kellett foglalkoznunk. A részfeladat megvalósítását két fázisra osztottuk, s a munkaszakasz mostani – második – részében felállítottunk egy olyan fogalmi sémát, melyen belül egyértelművé vált, hogy hol, milyenfajta további konzisztenciafeltételeket, konzisztenciakényszereket lehet megfogalmazni és előírni az ontológiaszerkesztők használata, tehát az ontológiaépítés folyamata, illetve – időben talán kicsit később – az ontológiákba épített tudás következtetési mechanizmusok révén történő, aktív hasznosítása során.

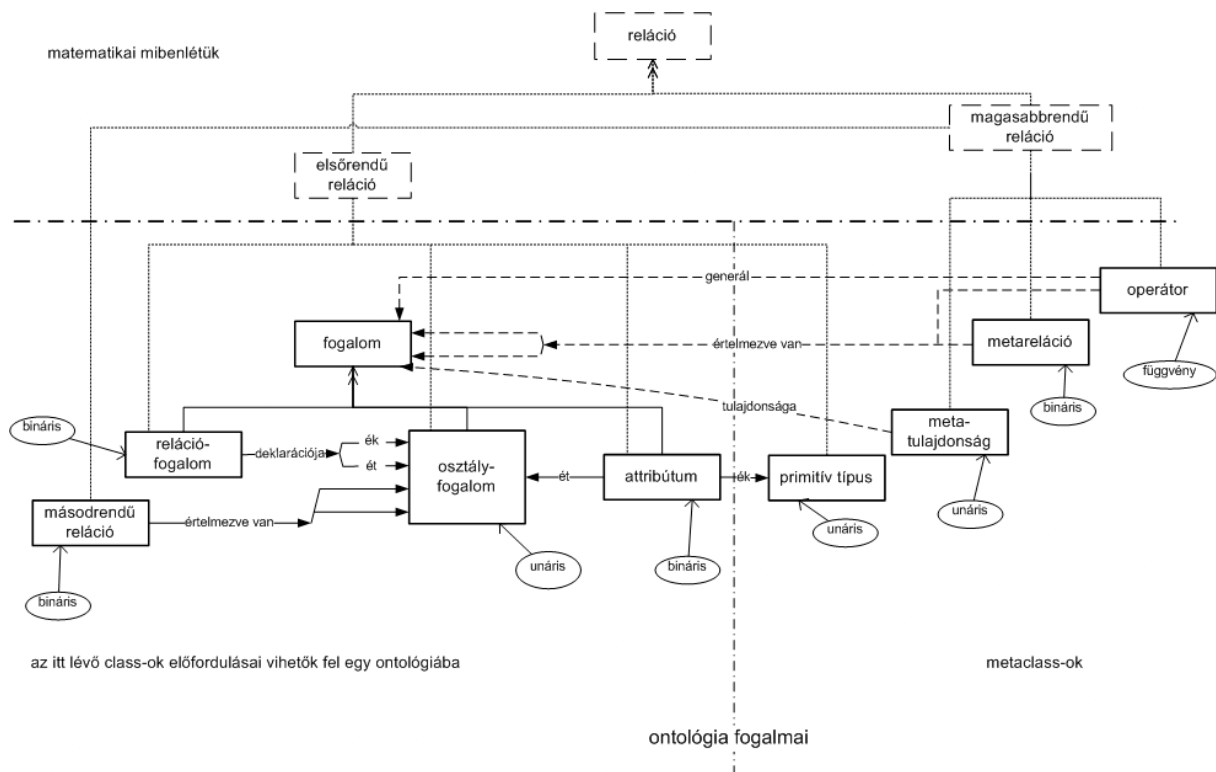
Elkészítettük a MEO-ontológia modelljét, melynek alapvető célja az volt, hogy pontosan kezelni (elválasztani, de ha kell, egymással összekötni) tudjuk az ontológia nyelvi és fogalmi rétegeit, illetve a típusok és a típusokba tartozó instanciák szintjét. A modell – éppen ezért – két dimenzióban két-két részre bontható, egyfelől a *nyelvi rétegre* és a *fogalmi rétegre*, másfelől a *fogalomtartományra* és az *instanciatartományra*. Az egyes részekre önálló modelleket építünk fel. A nyelvi rétegben egy modellt volt érdemes felépíteni (a *nyelvi modellt*), a fogalmi rétegben kettőt: a *fogalommodell* a FOGALMAK modellje, az *instanciamodell* a fogalmak terjedelmébe tartozó ELŐFORDULÁSOK modellje. A MEO-modell áttekintő képét az alábbi ábra mutatja:



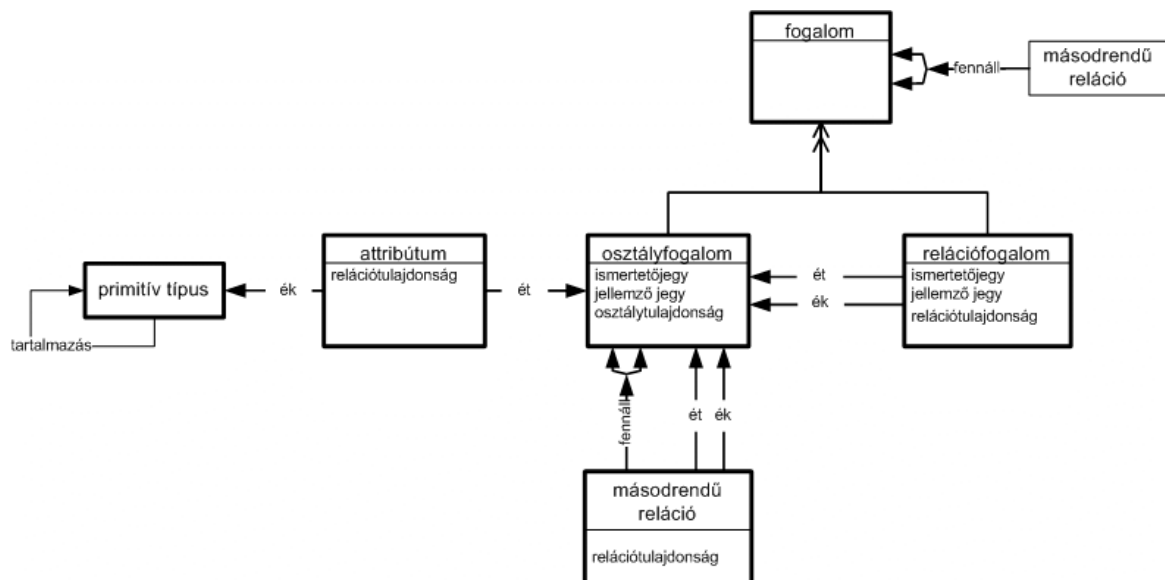
A fenti modellre támaszkodva belekezdünk egy saját ontológiaszerkesztő alkalmazás fejlesztésébe. A MEO-ontológia modell alapján felépülő informatikai rendszer a *MEO-ontológiaszerkesztő* vagy *MEO-rendszer*, de a munka befejezhetősége érdekében kikötöttük, hogy a MEO-rendszer első változata nem foglalkozik az instanciatartomány problémáival, illetve következtetési feladatokkal.

A fogalmi réteg modellje

A MEO-modell lelke az alapfogalmak egymáshoz való viszonyát rögzítő rendszer, melyet a következő ábrával szemléltethetünk. Az ábrában munkafogalomként használva a class metafogalmat elválasztjuk egymástól egyfelől a fogalmak *matematikai nyelven* megvalósuló leírását, illetve MEO-rendszer *tárgynyelvén* leírható fogalmak rendszerét, másfelől a *fogalmakat*, illetve *metafogalmakat* tartalmazó rétegeket egymástól. A MEO-tárgynyelv három alapfogalma az *osztály*-, a *reláció*- és az *attribútumfogalom*, és ebben a rétegben használjuk még a másodrendű relációkat is (melyekből azonban vélhetőleg nagyon kevés lesz). A metaréteg legfontosabb metafogalmai a *primitív típus*, a *metatulajdonság*, a *metareláció* (ide tartozik például a generikus reláció!) és az *operátor*.



A MEO-modellben kiemelt fontosságot tulajdonítunk a fogalmakat inherens módon jellemző jegyeknek (melyeknek fajtái lehetnek az ismertetőjegy, a jellemző jegy, az osztály- és relációtulajdonság). Ezeket a jegyeket azért is különösen figyelniük kell az ontológiaszerkesztés során, mert ezek egy része öröklődik, öröklődhet vagy nem öröklődhet, és mindezek még relációs kényszerekénél is szerepet játszhatnak. A jegyek és az alapfogalmak közti összefüggéseket mutatjuk be az alábbi ábrán:



- A MEO-modellhez az alapfogalmak meghatározásán túl tisztáztuk még
- a metarelációkat és a köztük levő összefüggéseket,
 - a relációtulajdonságokat – ezeken belül elkülönítve egymástól
 - az algebrai tulajdonságokat,
 - függvényszerű tulajdonságokat,
 - számosságkorlátozásokat,

- a relációtulajdonságok közti összefüggéseket, kényszereket,
- a rendszerben szükséges operációk meghatározását.

Bővebben: [MEO-usecase], [MEO-modell]

A nyelvi réteg modellje (nyelvi ontológiája)

Ha ontológiáról beszélünk, akkor nyelvfüggetlen fogalmak rendszerére gondolunk. Ha tehát a nyelvi fogalmak ontológiáját akarjuk felépíteni, akkor annak is nyelvfüggetlennek kell lennie. Bármire is akarnánk használni az ontológiákat, valószínű mindig szükségünk lehet arra, hogy az ontológiai ismeretet valamely nyelvhez tudjuk kapcsolni. Ezért a MEO-projekt elmúlt időszakában felértékelődött a fogalmi és nyelvi réteg kapcsolódásának kérdése, illetve a nyelvi réteg önmagában vett problémája.

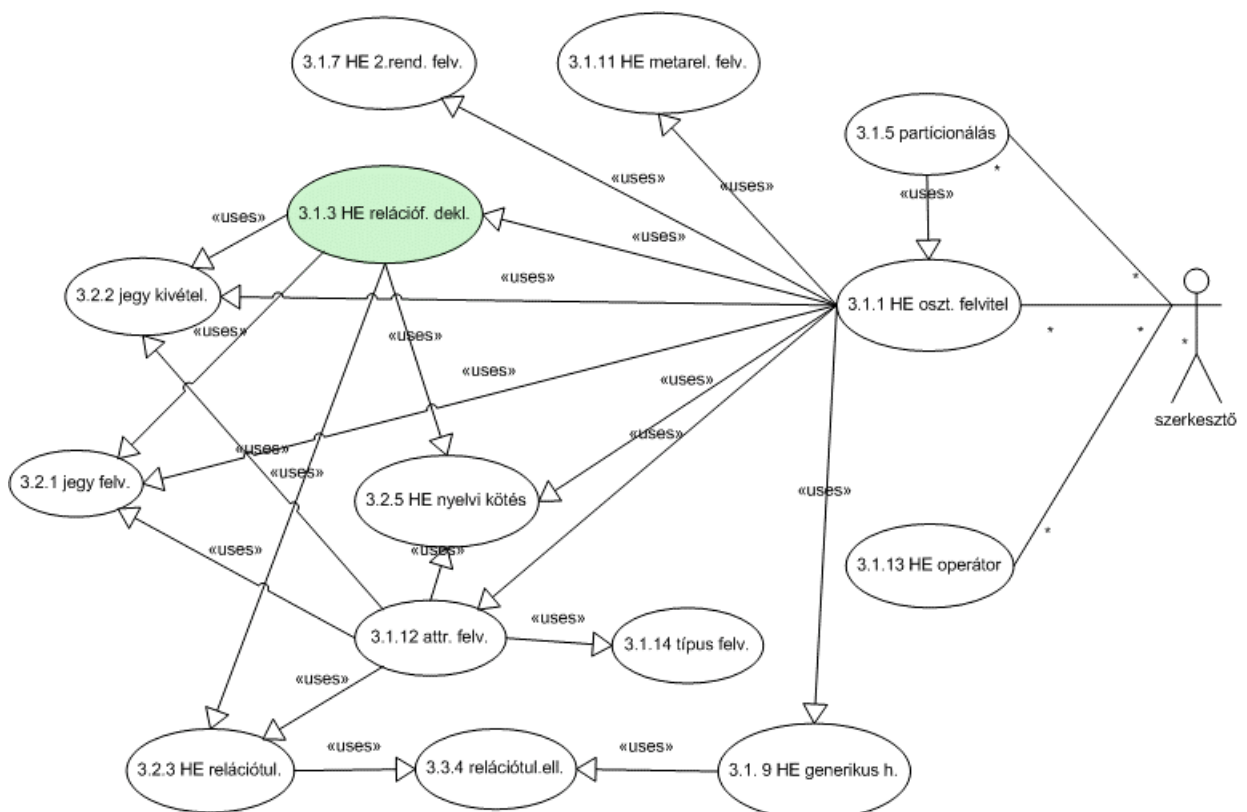
A nyelvi réteg legfontosabb entitásai a HANGALAK, az ÍRÁSALAK, a MORFOLÓGIAI EGYSÉG és a KONSTRUKCIÓ, melyek metaszintű – nyelvre, nyelvtani tulajdonságokra, nyelvi relációkra utaló – fogalmakkal jellemezhetőek. A hang- és írásalakok, a nyelvi jelek „hordozói“, melyek a morfológiai tulajdonságaikkal együtt már mondatokba, nyelvi megnyilatkozásokba illeszthetők. A nyelvi jelek a fogalmi rétegtől úgy kaphatják meg a jelentésüket, hogy a jelentéseket és a formai jegyeket konstrukciókba fogjuk össze.

A nyelvi réteg fontos továbbfejlesztéseként meg kell még csinálnunk a morfológiai jegyek rendszerezését, melyet annyiban készítettünk elő, hogy a projekt második munkaszakaszában adatbázisba rendezett Gold lingviztikai ontológia összes tételét lefordítottuk magyarra.

Bővebben: [MEO-modell], [hungold]

Használati eset modell

Az alapfogalmak tisztázása után elkészítettünk egy használati eset modellt, amelyben az ontológiaszerkesztő használatának lehetséges módozatait szedtük össze. A use case modell összetevőit az alábbi ábra jeleníti meg:



A használati eset modell a fogalmak szerkesztésével kapcsolatos folyamatokat rögzíti, a nyelvi réteghez köthető funkciókat a projekt következő munkaszakaszában tervezzük meg és implementáljuk.

Bővebben: [MEO-usecase]

Jelentésreprezentáció

A projekt a jelentésreprezentáció területén jelentős előrelépéseket tett meg a formális jelentésreprezentáció területén azzal, hogy a korábban lefektett – Jerry Hobbs elképzeléseire igazodó – elvek szerint több fontos tudástományban alapos leírásokat dolgozott ki.

Az általános ontológiához a természetes nyelv ontológiai elkötelezettségein keresztül lehet közelférközni, ezek az elkötelezettségek pedig a nyelv lexikai elemeinek részletes vizsgálatával tárhatók fel. Az alábbiakban alkalmazott ontológiaszemléletet ezért alapvetően két dolog jellemzi. Egyfelől az, hogy egy sajátos komponenciális (fogalmi összetevős) elemzés, másfelől pedig az, hogy eközben kitüntetett szerepet szán a skála (és általában a fogalmi tér) fogalmának. A komponenciális elemzés a Generatív Szemantika kudarca és Richard Montague hatása miatt gyakorlatilag eltűnt az elméletek színpadáról. Ez természetesen nem jelenti azt, hogy alapjaiban elhibázott út volt, legfeljebb azt, hogy a generatív szemantikusok atomi jegyekre építő strukturálatlan logikai rendszere elégtelen volt a hatékony elemzés céljaira. Ugyanakkor a komponenciális elemzés mindig is ott bujkált a lexikai szemantikában, és most megújult formában kezd visszatérni a Peter Gärdenfors és társai által a kognitív tudományban propagált Konceptuális Terek (Conceptual Spaces) megközelítésével.

A Konceptuális Terek hipotézise szerint a fogalmak szerveződése legalább annyira geometriai, mint szimbolikus: az egyes fogalmak régiók bizonyos sajátos topológiájú dimenziókra — tipikusan rendezett halmazokra, azaz skálákra — épülő terekben, és tulajdonságaikat az ezekben a terekben elfoglalt pozíciójuk határozza meg. Ennek alapján azt mondhatjuk, hogy egy fogalomból elérhetőnek tekinthetünk egy másik fogalmat, ha van olyan dimenzió, amelynek mentén összehasonlíthatók. Az egymásból kölcsönösen elérhető fogalmak tere egy-egy domaint alkot, amelyek viszonylag függetlenek egymástól. Ugyanakkor az ilyen terek között különböző rendezéstartó leképezések létezhetnek, aminek segítségével legalábbis részben megragadható az emberi gondolkodás metaforikus és metonimikus kiterjesztésekre való képessége, azaz a „kreatív nyelvhasználat” is.

Ennek a megközelítésnek az eredménye az N. Guarino vezette csoport által kidolgozott Dolce ontológia is, amelyet mi is sokban követtünk. Másik — Guarinoéktól független — elkötelezettségünk a Jerry R. Hobbs által kidolgozott rendkívül hajlékony, davidsoniánus alapokon nyugvó jelentésreprezentációs nyelv és a hozzákapcsolódó szemlélet követése volt.

A természetes nyelvekben (így a magyarban is) a lexikalizálódott fogalmak összefüggő rendszert (lexikográfusi kifejezéssel: szemantikai mezőt) alkotnak, és az ontológiaépítés alapfeladataként az e mezők mögött meghúzódó kognitív dimenziók feltárását tekintettük. Ezen egymástól többé-kevésbé független dimenziók összessége definiálja azt a teret, amelyben a (magyar) nyelv által lexikalizált fogalmak elhelyezkednek. A dimenzióknak — mint említettük — igen gyakran van skála-szerkezetük, és ez indokolja az általunk fejlesztett ontológiában a skálák kiemelt szerepét.

A fenti megközelítéshez rendkívül jól illeszkedik a mereológiai nézőpont, ezért a jelentésreprezentációról szóló tanulmány a mereológiai alapok tisztázásával kezdődik. A projekt során négy tartományt fogalmait dolgoztuk ki a hozzájuk rendelt dimenziókkal együtt. A pszichikai/mentális fogalmak tartományát ketté osztva a *kognitív* (propozicionális) *attitűdök*, illetve az *érzelmi fogalmak* tartományát, a *kommunikációval* kapcsolatos fogalmakat és a *fizikai mozgások* fogalmainak tartományát. E négy közül három (propozicionális/konatív attitűdök és érzelmek, valamint a kommunikáció fogalmi köre) igen

erősen összefügg. A részletes elemzéseken túl a résztanulmány Függelékben vázlatosan bemutatunk néhány kevésbé kidolgozott területet is, hogy a megközelítés hatékonyságát illusztráljuk, valamint röviden kitérünk Jerry Hobbs keretrendszerének elméleti vizsgálatára.

Bővebben: [jelrepr]

Az ontológiaszerkesztés gyakorlata

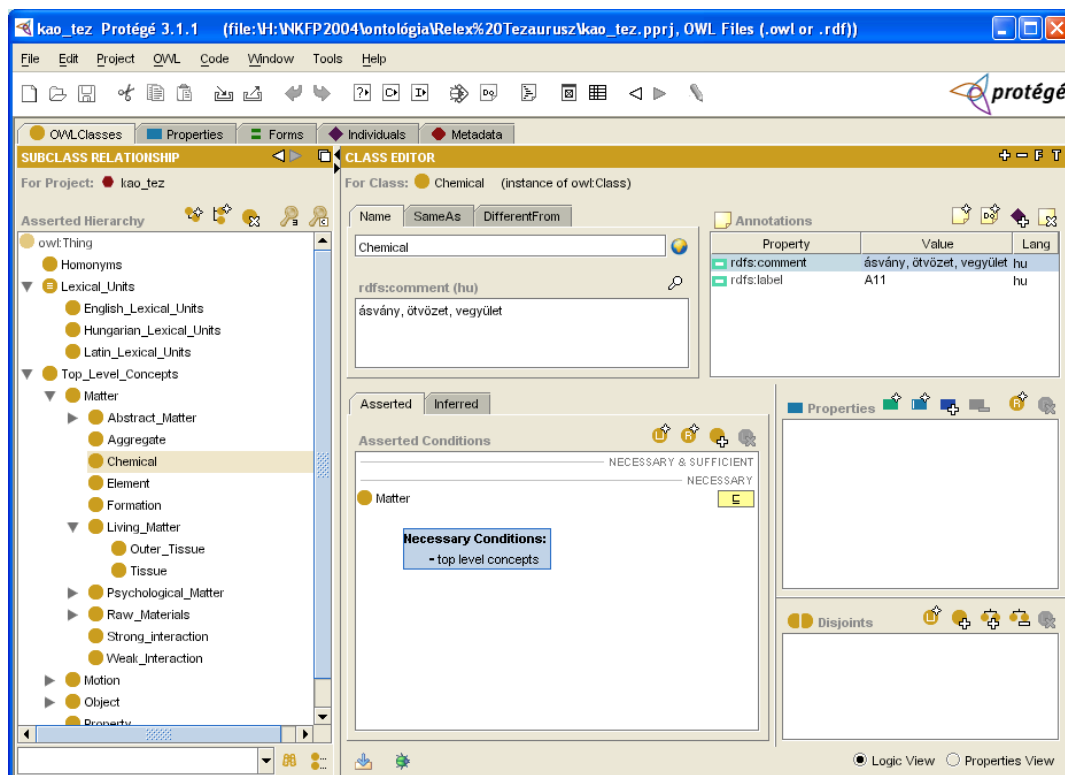
A MEO projektben kiemelt hangsúlyt helyezünk az ontológiainfrastruktúra kérdéskörére, s ezen belül is az ontológiaszerkesztést támogató eszközök biztosítására. A projekt korábbi fázisában a *Protégé* alkalmazást javasoltuk elsősorú ontológiaszerkesztőnek. A *Protégé* használatának megkönnyítésére két útmutatót készítettünk, egyet a *Protégé* szerverek telepítésével, menedzselésével, egy másikat a *Protégé* kliensek használatával kapcsolatos kérdések számbavételére.

A *Protégé* szerverrel kapcsolatos tudnivalókat az alábbi kérdéscsoportokba szerveztük:

- *Protégé* 3.1.1 szerver telepítése
- OWL adatbázis projektek létrehozása
- OWL adatbázis projektek beillesztése a *Protégé* szerverbe
- *Protégé* 3.1.1 kliensek telepítése
- A kapcsolat felépítése a kliensek és a szerver között
- A RacerPro 1.9. logikai következtető gép telepítése és használata

A *Protégé* felhasználói útmutató két nagyobb fejezetből áll:

- Projektek betöltése, mentése, importálása és exportálása
- Projektek szerkesztése



ibővebben: [ProtegeServerGuide], [ProtegeUserGuide]

MEOditor többnyelvű ontológiaszerkesztő

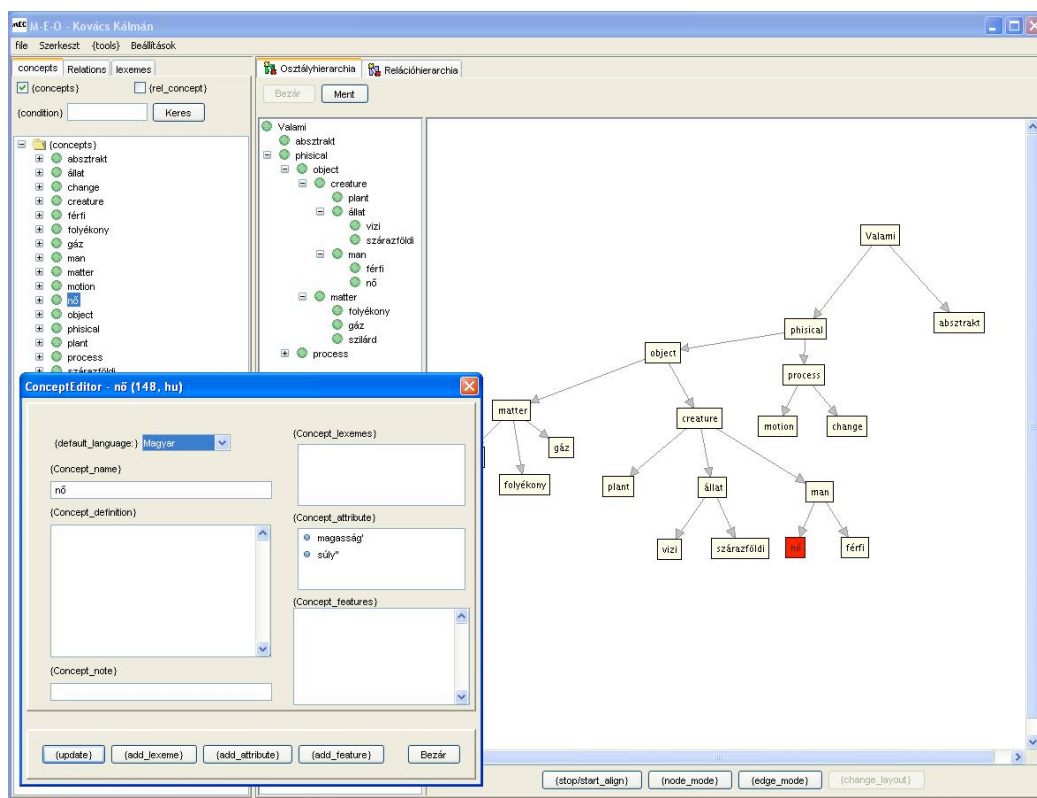
A *Protégé* sok szempontból a legmegfelelőbb választásnak tűnik ontológiaszerkesztési feladatokra. Bizonyos funkciók esetében azonban korábban is voltak fenntartásaink a *Protégé*

teljes körű használhatóságával kapcsolatban, és ezek a projekt harmadik szakaszában nem csak maradtak, hanem még meg is erősödtek. Végül is úgy döntöttünk, hogy a MEO ontológiainfrastruktúra-építési munkája saját ontológiaszerkesztőt is kifejlesztünk.

- beépíti magába más tudásszervezési rendszer-szerkesztők (például a Relex tezauruskezelő) jó tulajdonságait,
- hálózati üzemmódban képes működni,
- adatbázisból tud közvetlenül dolgozni,
- megfelel a MEO-projekt keretében kidolgozott fogalmi séma elveinek, definícióinak, a fogalmi séma kényszereit kezelni tudja,
- kompatibilis minden fontosabb nemzetközi eredménnyel (OWL, Protégé stb.)
- mindenféle nyelvi kötést könnyen és korlátok nélkül kezelni tud.

Bár az ontológiaszerkesztők fogalmi rendszerek szerkesztésére valók, s ezért a fogalmak nyelvi kötéseivel kapcsolatos menedzselési képességeket talán nem is kéne elvárni, ma már úgy látjuk, hogy az ontológiák nyelvi reprezentálásának kérdését komolyabban kell kezelni. A nyelvi és fogalmi rétegek pontos és egyértelmű elválasztása, a nyelvek közti különbségek kezelésének lehetősége, a felépített fogalmi rendszerek többnyelvű használhatósága mind megköveteli azt, hogy az addig szokásos gyakorlathoz képest nagyobb hangsúlyt fektessünk az ontológiaszerkesztők nyelvkezelési képességeire. Ezért a fenti elvárásokra és megfontolásokra támaszkodva a projekt harmadik szakaszában MEOditor néven egy többnyelvű, adatbázis-alapú, hálózati ontológiaszerkesztő alkalmazás fejlesztésébe kezdünk.

Az alkalmazás fogalmi réteget kezelő modul prototípusa elkészült (természetesen úgy, hogy a nyelvi réteggel való kapcsolat a MEO-modell szerint épült fel), a projekt következő szakaszára maradt a jogosultsági rendszer és a munkafolyamat-kezelő modul implementálása, elkezdődött a nyelvi réteget kezelő komponens fejlesztése, valamint megterveztük és fejleszteni kezdtük a fogalmakat és a köztük levő kapcsolatokat 3D-s környezetben reprezentálni képes megjelenítő modul rendszerbe illesztését.



Bővebben: [MEOditor]

Ontológiaépítés

A projektterv szerinti 3.2-3. és 4.1-4. részfeladatokról kell beszámolnunk ebben a fejezetben.

MEO-csúcsontológia

A projekt első szakaszában összeállítottuk a MEO csúcsontológiájába tartozó elemek gyűjteményét. A második projektszakaszban elkezdjük az elemek közti struktúra kialakítását, amikor elkészült az anyag, a tárgy és nagyrészt a tevékenység fogalmi kategóriák generikus hierarchiája. A harmadik fázisban előállítottuk az összes fogalmi egység tezaurszcikkét. Most tehát rendelkezésre áll a MEO-csúcselemek tezaurszrelációk mentén vett struktúrája. A harmadik munkaszakaszban felállítottuk az ontológia modelljét is, vagyis rögzítettük a legfontosabb alap- és metafogalmak definícióit és egymáshoz való viszonyát. Az kezdettől nyilvánvaló volt, hogy az ontológia más és több, mint a tezaursz, de hogy mit és milyen módon lehet az ontológiák számára többletként tulajdonítani, azt nem tudtuk. Ezért (is) kezdtük el keresni a jelentésrepresentációval foglalkozó munkacsoporton belül azokat a dimenziókat, fogalmi skálákat, melyek mentén a fogalmak egy adott csoportja összerendezhető, mert ezáltal feltárhatóvá és leírhatóvá válhat az a teljes fogalmi tér, melyben az ontológia fogalmait elhelyezhetjük. Ezt a dimenziófeltárási munkát a projekt utolsó szakaszában (sőt, nyilván a projekt lezárulta után is) folytatni kell.

Bővebben: [MEO-gener], [MEO-top], [jelrepr]

MEO-szakontológia

A 3.2. részfeladatban a távközlési tezaursz felújítása és transzformálása volt a feladat, míg a 3.3 részfeladat a távközlési közönségszolgálati szakontológia felépítését írta elő.

A távközlési tezaurszt sokféle – a MEO-projekten kívüli – célokra is fel lehet használni, ezért azt a projekt ideje után (s remélhetőleg majd azután is) folyamatosan karbantartjuk. Az ontológia portálon a projektjelentéssel egyidőben elérhetővé tettük a tezaursz legfrissebb változatát. Ezt a állományt még mindig a Relex tezaurszkezelővel szerkesztettük, és MS Access formátumban tettük elérhetővé. A projekt negyedik munkaszakaszában fogjuk ezt a tezaurszt is (a MEO-csúcsontológiával együtt) betölteni a projekt ontológiászerkesztőjének adatbázisába.

Bár a távközlési közönségszolgálati tevékenységet támogató alkalmazásnak is távközlési szakontológiát kell használnia, erre a speciális célra sokkal pontosabb, szűkebb terjedelmű fogalmakat kell alkalmaznunk, mint amit a „teljes“ távközlési tezaurszunk lehetővé tesz. Ezért az „általános célú“ távközlési tezaurszunk és a CC-munkatárs számára felkínált tudásbázis között csak részleges átfedés van. Természetesen a távközlési tezaursz jelentette a kiindulási alapot az alkalmazás tudásbázisa számára is, de az ügyfélszolgálati tevékenység konkrét tartalma megkívánta a további fogalmi mélyítést. A projekt utolsó szakaszában, az ontológiák életciklus-menedzsmentjével kapcsolatban feladatok elvégzése során figyelembe kell majd vennünk – többek között – az itt szerzett specializálási tapasztalatokat is.

Bővebben: [telcotez], [CC-kon]

SUMO-csúcsontológia

A 4.1. munkaszakaszban a szabadon felhasználható csúcsontológiák esetleges hasznosítását kellett előkészítenünk. Két nagy ontológiára (a WordNet és a SUMO-ra) hivatkoznak világszerte és ezeket egyre több célra kezdik el használni. Bár a projekt tervezési fázisában felmerült, hogy próbáljuk meg a WordNet valamilyen hasznosítását, de ezt elvetettük. A SUMO ontológiát viszont – legalább részeiben – alkalmazhatónak véljük. Éppen ezért az idő

problémakörét kiválasztva megvizsgáltuk, hogy ezen a területen milyen a SUMO és MEO elemeinek egymáshoz való viszonya – az illeszkedés szempontjából.

SUMO/MEO térontológia

A különböző ontológiák közti integrációs lehetőségek megvalósíthatóságát, az integrációhoz szükséges feladatokat a SUMO és a MEO térrel kapcsolatos részontológiái között „próbáltuk ki”. Bármennyire is voltak „műfaji” nehézségei a MEO és más ontológiák összehasonlításának, mivel a SUMO idővel kapcsolatos fogalmai formalizált axiómákban, logikai kijelentésekben álltak rendelkezésre, míg a MEO-ban az idő-fogalmaknak csak a külső kapcsolatai készültek el eddig (melyek főként a két hierarchikus reláció mentén épültek fel).

A térrel kapcsolatos osztályszintű fogalmak az alábbi főcsoportokba sorolhatók:

- a TERÜLETEK (REGION),
- az ÁLLANDÓ HELYZETŰ TÁRGYAK (STATIONARYARTIFACT),
- a MOZGÁS fajtái (MOTION),
- a TÉRRE vonatkozó MÉRTÉKEK (CONSTANTQUANTITY),
- a GEOMETRIAI ALAKZATOK (SHAPEATTRIBUTE) .

A térrel kapcsolatos relációs fogalmak típusai a következők:

- TÉRBELI relációk (SPATIAL)
- MEREOLÓGIAI relációk (MERELOGICAL)
- LYUKKAL kapcsolatos relációk (HOLE)
- ÚTVONAL és IRÁNY (PATH and DIRECTION)

A térrel kapcsolatos attribútumok a relációs attribútumok (RELATIONALATTRIBUTE) közé tartoznak, hiszen az entitások egymás közti viszonyát jellemzik. Fajtái:

- CÍM (ADDRESS)
- POZÍCIÓS ATTRIBÚTUMOK (POSITIONALATTRIBUTE) mint fent, lent, jobbra, balra, stb.

A fentiek alapján a SUMO és a MEO és térfogalmi közti összefüggéseket az alábbi táblázattal tudjuk megadni:

MEO-H fogalmak	SUMO-H fogalmak	Típ.	Megjegyzés
HELY	-	-	Nincs megfelelő
ÁLLAPOTHELY	SUBJECTIVEASSESSMENT-ATTRIBUTE	<	
HELYKEZDET	FRONTFN	~	
HELYVÉG	BEGINFN	~	
VISZONYHELY	SPATIALRELATION, POSITIONALATTRIBUTE	<, ~	
KÖZÖTT	BETWEEN	=	
ELŐTTE	NEAR, ADJACENT	~	
MÉRTANI HELY	GEOMETRICFIGURE	~	
PONT	GEOMETRICPOINT	=	
VONAL	ONEDIMENSIONALSHAPE	=	
SÍK, TÉR	SHAPEATTRIBUTE	<	
FIZIKAI HELY	REGION, STATIONARYARTIFACT	~, ~	
FÖLDRAJZI HELY	GEOGRAPHICAREA	=	
LÉTESÍTMÉNY	STATIONARYARTIFACT	>	
KÉPZŐDMÉNY	ASTRONOMICALBODY	>	
TÉRMÉRTÉKEGYSÉG	-	-	Nincs megfelelő

TERÜLETEGYSÉG	AREAMEASURE	~	
HOSSZEGYSÉG	LENGTHMEASURE	~	
HELYNÉVMÁS	-	-	Nincs megfelelő
AHOL	LOCATED, WHEREFN	~	
MEDDIG	SUPERFICIALPART	~	Eltérő megközelítés

A részfeladat teljesítése során kigyűjtöttük még a térrel kapcsolatos (általános, illetve speciálisan a létesítményekkel, helyszínekkel „foglalkozó”) nyelvi kifejezéseket (közneveket, névmásokat, határozószókat), melyek megteremthetik a projekt következő fázisában felépíteni kívánt földrajzi nevek tulajdonnév-rendszerének ontológiai kötési lehetőségeit.

Bővebben: [SUMO-ter], [SUMO-ido]

A MEO igei vonzatkeret-tárának felépítése

A projekt 4.3. részfeladatának megfelelően a most jelentett munkaszakaszban be kellett fejeznünk az igei vonzatkeret-tár felépítését. Ezt a munkát elvégeztük. Időközben azonban még inkább nyilvánvalóvá vált, hogy jóval kevesebb hasznát vehetjük a vonzatkeret jellegű információknak, mint azt a projekt tervezésekor reméltük. Azt természetesen a kezdetekben is tudtuk, hogy az igei vonzatkeret-tár nyelvi-lexikai, tehát szavak közti, nem pedig ontológiai, tehát fogalmak közti információt rögzít. A MEO-modell felépítésekor, a nyelvi és fogalmi réteg elválasztásakor tovább erősödött az a felismerés, hogy az igei vonzatkeretek „csak” útmutatást, ráutalást nyújthatnak az ontológiai építkezés számára.

A vonzatkeret információk a jelentésreprezentációs munkánkban sem jelentettek direkt segítséget, hiszen a fogalmak formalizálása során ezek inkább csak háttértámogatást, „formalizálási ötleteket” voltak képesek nyújtani.

Mindenesetre a MEO-csúcsontológiába tartozó igék vonzatkereteivel feltöltöttük a rendszerünket, bár ez mindaddig nem látható, amíg a MEODitor szerkesztő nem kezeli a nyelvi réteget információit is. Mivel a Protégé ontológiaszerkesztő nem képes nyelvi elemeket kezelni, azért vonzatkeret-információ betöltésével nem is érdemes próbálkozni.

Részben erről: [jelrepr]

Projektkommunikáció

A projekt kommunikációja a 3. munkaszakaszban is két csatornán keresztül zajlott. Egyrészt a munka során létrejött dokumentumokat, részeredményeket folyamatosan közzé tettük az ontológia portálon (<http://ontologia.hu>), másrészt a projekt résztvevői több belső levelezési listát használtak a szakmai és adminisztratív teendők megbeszélésére. Ebben a munkaszakaszban több munkacsoportban (pl. a modell-építő vagy a jelentésreprezentációs csoportokban) intenzív és folyamatos közös munka folyt.

A projekt eddig elért eredményeiről, az alkalmazott megközelítésmódokról és megoldásokról cikket adtunk be az ISD 2006 konferenciára (Fifteenth International Conference On Information Systems Development, Methods and Tools, Theory and Practice - New Methods and Practices for the Networked Society), és elkészült egy tanulmány „Ontológiai helyzetkép a teauruszokról az OntoClean szemlélet alapján” címmel, melynek a projekt következő fázisában megfelelő publikálási fórumot keresünk.

Hivatkozások, projektdokumentumok

Rész tanulmányok

[CC-kov]

Scriptum Rt., *Részletes terv adott hibakonfigurációból problémahalmazra való visszakövetkeztési mechanizmusra* (26 o.)

http://ontologia.hu/document/proj_doc/cc_szakontologia/cc-kov/download

[CC-anim]

Scriptum Rt., *A CC-munkatárs tevékenységét támogató alkalmazás működését bemutató animáció*

http://ontologia.hu/Members/Bakos_Gabor/bemutato/TelcomDemoFlash.zip

[jelrepr]

Gyarmathy Zsófia, Héja Enikő, Mittelholz Iván, Simonyi András, Szeredi Dániel, Varasdi Károly, *A magyar nyelv lexikai sajátosságaira épülő formális általános ontológia* (89 o.)

http://ontologia.hu/Members/varasdi/MEO_jelrep00601.pdf/download

[MEO-model]

Gyepesi György, Szakadát István, Szaszko Sándor, Szöts Miklós, Ungváry Rudolf, Kálmán László (lektor), *MEO-ontológia modell* (25 o.)

http://ontologia.hu/document/proj_doc/meo_model.pdf

[MEO-usecase]

Szöts Miklós, Gyepesi György, *MEOditor ontológiaszerkesztő használati eset modell* (37 o.)

http://ontologia.hu/Members/szots/MEO_usecase.pdf

[OntoCleanGuide]

Szöts Miklós, *Az OntoClean metodológia ismertetése* (43 o.)

<http://ontologia.hu/Members/szots/OntoCleanGuide.pdf>

[ProtegeServerGuide]

SZTE SZTTCS, *A Protégé 3.1.1 szerver használata* (22 o.)

http://ontologia.hu/document/proj_doc/ontologiai_szerkesztok/meo_protege_server_guide_hun.pdf

[ProtegeUserGuide]

SZTE SZTTCS, *Protégé 3.1.1 használati útmutató* (13 o.)

http://ontologia.hu/document/proj_doc/ontologiai_szerkesztok/meo_protege_user_guide_hun.pdf

[SUMO-ter]

Miháltz Márton, *Térrel kapcsolatos fogalmak a SUMO ontológiában és összehasonlításuk a MEO helyontológiájával* (17 o.)

http://ontologia.hu/Members/mihaltz/Tanulmanyok/mm_ter.pdf

[SUMO-ido]

Miháltz Márton, *Idővel kapcsolatos fogalmak a SUMO ontológiában és összehasonlításuk a MEO időontológiájával* (31 o.)

http://ontologia.hu/Members/mihaltz/Tanulmanyok/MM_idoter2.doc.zip

Részontológiák, alkalmazások

[hungold]

Gold – nyelvészeti ontológia

<http://ontologia.hu/document/KOS/Egyeb/gold/hungold>

[MEODitor]

MEODitor ontológiaszerkesztő

<http://ontologia.hu/meoditor/>

[MEO-top]

Ungváry Rudolf, *A MEO relációs struktúrája*

http://ontologia.hu/document/KOS/KOS_MEO_folder/KOS-MEO-02/meotop

[telcotez]

Ungváry Rudolf, *Távközlési teaurusz*

<http://ontologia.hu/document/KOS/thesaurus/tavkozles/telcotes060605>

A projekt keretében született publikációk

[Ungváry 2006]

Ungváry Rudolf, *Ontológiai helyzetkép a teauruszokról az OntoClean szemlélet alapján*, kézirat, 2006

http://ontologia.hu/document/paper/teaurusz_mint_kisvilag_0.pdf

[Szakadát 2005]

Szakadát István, *MEO Ontology Infrastructure*, ISD 2006, 2006

http://mokk.bme.hu/archive/meo_isd2006.pdf