Lengoaia eta Sistema Informatikoak Saila

Informatika Fakultatea

IZAERA HETEROGENEOKO BALIABIDE LEXIKALEN INTEGRAZIORAKO ARKITEKTURA BATEN PROPOSAMENA

Datu-integrazioaren ikuspegitik egindako ekarpena

Aitor Soroa Etxabek

Informatikan Doktore titulua eskuratzeko aurkezturiko

TESI-TXOSTENA

Donostia, 2004ko iraila.

Lengoaia eta Sistema Informatikoak Saila

Informatika fakultatea

IZAERA HETEROGENEOKO BALIABIDE LEXIKALEN INTEGRAZIORAKO ARKITEKTURA BATEN PROPOSAMENA

Datu-integrazioaren ikuspegitik egindako ekarpena

Aitor Soroa Etxabek Xabier Artolaren zuzendaritzapean egindako tesiaren txostena, Euskal Herriko Unibertsitatean Informatikan Doktore titulua eskuratzeko aurkeztua

Donostia, 2004ko iraila

Gaien aurkibidea

I Sarrera eta motibazioa. 1

I.1 Aurkezpen orokorra. 1
I.2 IXA taldea eta lexikografia konputazionala. 4
I.3 Ingeniaritza linguistikoa. 5
I.4 Datu-integrazioa. Sarrera gisa. 8
I.5 ELHISA. 13
I.6 Txostenaren eskema. 15

II Baliabide lexikalak. 17

II.1 Baliabide lexikalak eta LNPa. 18
II.1.1 Baliabide lexikalen estandarizaziorantz. 24
II.1.2 Informazio lexikalaren integrazioa. 32
II.2 Informazio lexikalaren errepresentazioa. 38
II.2.1 Testu-ereduak. Markaketa. 42
II.2.1.1 SGML/XML markaketa-lengoaiak eta TEI ekimena. 46
II.2.2 Datu-baseak. 50
II.2.2.1 Datu-base sasi-egituratuak. 54
II.2.3 Ezagutza-baseak. 59
II.2.3.1 Ezagutzaren errepresentazioa. 61
II.2.3.2 Ezagutza lexikalaren errepresentazioa. 64
II.3 Baliabide lexikalen sailkapen modukoa eta zenbait adibide. . 67
II.3.1 Hiztegiak. 68
II.3.1.1 EH hiztegia. 69
II.3.2 Datu-base lexikalak. 70
II.3.2.1 EDBL. 71
II.3.3 Ezagutza-base lexikalak. 72
II.3.3.1 EDR. 73

II.3.3.2 WordNet eta EuskalWordNet. 73
II.3.3.3 Hiztsua. 74

III Datu-integrazioa. 75
III.1 Informazioaren integrazioaren nondik norakoak. 76
III.1.1 Federazioak eta datu-base anizkoitzak. 78
III.1.2 Datu-biltegiak. 81
III.2 Datu-integrazioa eta adimen artifiziala. 83
III.2.1 Ontologian oinarritutako datu-integrazioa. 83
III.2.2 Plangintza. 87
III.3 Datu-integrazioa. 88
III.3.1 Aurrekariak. Definizioak. 90
III.3.1.1 Galderak. 90
III.3.1.2 Galderen barne-hartzea. 94
III.3.1.3 Datu-integrazioaren gure erreferentzia-eredua. 95
III.3.2 "Globala bistatzat" (GAV) delako hurbilpena. 95
III.3.3 "Lokala bistatzat" (LAV) delako hurbilpena. 99
III.3.3.1 Galderen erantzutea, bistetan oinarrituz. . . . 99
III.3.3.2 Algoritmoak. 103
III.3.4 Galdeketa-gaitasunak. 115
III.3.5 Datu-arazketa. 116
III.4 Wrapper-en teknologia. 121
III.5 Deskribapen-logikak. 123
III.5.1 Deskribapen-logikak eta datu-integrazioa. 126
III.5.2 CLASSIC deskribapen-logika. 131
III.6 Datu-integraziorako zenbait sistema. 136

IV ELHISA: informazio lexikalaren integraziorako arkitektura bat. 151
IV.1 Informazioaren integraziorako arkitektura ELHISAn. 152
IV.2 Maila kontzeptuala. 160
IV.2.1 Eredu Kontzeptual Orokorra. 162
IV.2.1.1 EKOaren goi-mailako sailkapena. 165
IV.2.1.2 EKOko objektu lexikalak: hitzak. 166
IV.2.1.3 EKOko objektu lexikalak: kontzeptuak eta adierak. 166
IV.2.1.4 EKOko objektu lexikalak: ezaugarriak. 168
IV.2.1.5 EKOko erlazioak. 168

IV.2.2 Baliabidearen Eredu Kontzeptuala (BEK). 170
IV.2.2.1 TEIren arabera kodeturiko hiztegi elebakar konplexu bat: Euskal Hiztegia. 173
IV.2.2.2 Datu-base lexikalak: EDBL. 174
IV.2.2.3 Datu-base lexikalak: EDR erraldoia. 175
IV.2.2.4 Ezagutza-baseak: *Hiztsua*. 179
IV.2.2.5 Ezagutza-baseak: Sinonimo multzoetan oinarritutako *EuroWordNet*. 180
IV.2.3 Baliabidearen Eduki-Deskribapena (BED). 185
IV.3 Bitartekoa eta galderen itzulpena. 190
IV.3.1 Galderen Itzultzailea. 190
IV.3.2 Optimizatzailea. 195
IV.4 Planifikatzailea. 196
IV.5 Wrapper-ak. 198
IV.5.1 OEM lengoaia. 200
IV.6 Datu-arazketa ELHISAn. 202
IV.6.1 Datu-garbiketa. 203
IV.6.2 Objektuen identifikazioa. 208
IV.7 ELHISAk gauzatutako integrazioari buruzko zenbait gogoeta. 212
IV.8 Integratu diren baliabideak. 216
IV.9 Datu-integrazioko beste sistema batzuekiko konparazioa. . . 219

V Galderen itzulpena eta erabilera-adibideak. 225
V.1 Interfazea. 225
V.2 Galderen itzulpena. 230
V.2.1 *MiniCon* algoritmoari eginiko aldakuntzak 232
V.2.2 Galderen itzultzailearen zenbait datu. 238
V.3 Erabilera-adibideak. 240

VI Baliabide berriak integratzen. 249
VI.1 Kasu praktikoa. Lematizazio datu-base baten integrazioa. . . 250

VII Ondorioak eta aurrera begirakoak. 263
VII.1 Ondorioak. 263
VII.2 Aurrera begirakoak eta zabaldutako ikerlerroak. 266
VII.2.1 Sistemaren hobekuntzak. 266
VII.2.2 Internetera begira. 269
VII.2.3 LNPko aplikazioak integratzen. 270

Irudien zerrenda

I.1 Datu-baseen arteko mapaketak. 12

II.1 *acknowledge* hitzaren 3 adierazpen desberdin. 21
II.2 *laguntza* sarrera EH hiztegian. 44
II.3 *laguntza* sarreraren markaketa prozedurala. 45
II.4 *laguntza* sarreraren markaketa deskriptiboa. 46

III.1 Datu-base federatuen bost mailako arkitektura 80
III.2 Datu-biltegi sistema baten oinarrizko arkitektura 81
III.3 Ontologien erabilpenaren hiru arkitektura 85
III.4 Bitarteko-arkitektura 97
III.5 Eskema orokorra eta S1 iturriarena 108
III.6 $\mathcal{AL}$ deskribapen-logiken familiako eraikitzaileak eta beren interpretazio semantikoa 127
III.7 CLASSIC sistemak onartutako eraikitzaileen sintaxi eta semantika. 132
III.8 Kategoria gramatikalen hierarkia, CLASSICen bitartez adierazia 134

IV.1 Arkitektura orokorra 158
IV.2 EKO. Goi-mailako sailkapena. 166
IV.3 EKO. Hitzak eta beren sailkapena. 167
IV.4 EKO. Kontzeptuak eta adierak. 168
IV.5 EKO. Ezaugarriak eta beren sailkapena. 169
IV.6 EKOaren zati bat, hitz-formei buruzkoa, NeoClassic-ez 172
IV.7 EH hiztegiaren BEKa. 174
IV.8 *EDBL* datu-basearen BEKa: hierarkia orokorra. 176
IV.9 *EDBL* datu-basearen BEKa: hiztegi sarrerak eta bestelako sarrerak (laburtua). 176
IV.10 *EDBL* datu-basearen BEKa: unitate estandar eta ez-estandarrak. 177
IV.11 *EDBL* datu-basearen BEKa: Hitz Anitzeko Unitate Lexikalak (HAUL) eta Zuriunerik Gabeko Sarrerak (ZGS). 177

IV.12 EDR datu-base lexikalaren BEKa. 179
IV.13 Hiztsua ezagutza-base lexikalaren BEKa 182
IV.14 EWN ezagutza-base lexikalaren BEKa 183
IV.15 EDBL baliabidearen BEDaren zati bat 188
IV.16 Galderen Itzultzailea 193
IV.17 ELHISAren planifikatzailearen osagaiak. 197

V.1 ELHISAren interfazea 229

VI.1 LM iturriaren modelizazioa (BEK). 251
VI.2 LM iturriaren BEKaren zati bat, NeoClassic-ez 253
VI.3 LM baliabidearen BEDa 255

Taulen zerrenda

IV.1 Klaseen arteko erlazio batzuk, EKOan zehazturik 171
IV.2 Objektu lexikalen arteko erlazioak, EHren BEKean. 175
IV.3 Objektu lexikalen arteko erlazio batzuk, *EDBL*ren BEKean. . . . 177
IV.4 Objektu lexikalen arteko erlazio batzuk, *EDR*ren BEKean. . . . 180
IV.5 Objektu lexikalen arteko erlazio batzuk, *Hiztsua*ren BEKean. . . 181
IV.6 Objektu lexikalen arteko erlazioak, *EuroWordNet*en BEKean. . . 184
IV.7 EH hiztegitik jasotako datu "zikinak" 204
IV.8 EH hiztegitik jasotako erantzun "garbia" 205
IV.9 EHrako kategoriaren normalizazio-hash-a (`EHKatHash`) 207
IV.10 "arrazoi" hitzaren aldaerak, EH eta EDBLtik jasoak 209
IV.11 Erantzunak, objektuak identikatu baino lehen. 210
IV.12 Erantzunak, objektuak identikatu ondoren. 211
IV.13 "otu" aditzaren azpikategorizaio-balioak 213
IV.14 ELHISAn integratutako baliabide lexikalak eta bertan aurki daitekeen informazioa. 218

V.1 Galdera-itzulpenaren hainbat neurri. 239
V.2 EDBL eta EH iturrietan egingo diren galderak 242
V.3 "arrazoi" hitzaren forma ez-estandarrak. 243
V.4 Determinatzaile singularrak. 244
V.5 "hori"hitzaren definizioak. 245
V.6 "hori"hitzaren itzulpenak. 246

VI.1 LM iturriko erlazio lokalak 252
VI.2 LM eta ELHISAren arteko hash taula bat. 259

I. KAPITULUA

Sarrera eta motibazioa.

I.1 Aurkezpen orokorra.

Baliabide lexikalek oso paper garrantzitsua betetzen dute Lengoaia Naturalaren Prozesamenduaren (LNP) arloan. Linguistika teorikoaren zein konputazionalaren egungo joera hizkuntz ezagutza gramatikalaren arlotik lexikoarenera lerratu da, neurri handi batean. Garai bateko lengoaia naturaleko ikerketetan sintaxi-formalismoak ziren informazio-iturri nagusiak, hizkuntzen alderdi erregularrak deskribatzen zituztelakoan edo. Horrela, lexikoia hizkuntza batek dituen berezitasunez osatutako biltegia baino ez zen; hitzen adierei buruzko hainbat arazo, halaber, sintaxi mailara lerrarazten ziren. Hala ere, ikuspuntu hori erabat aldatu zen laurogeiko hamarkadan, eta aldaketa horren arrazoiak teorikoak zein praktikoak izan ziren: beharbada, 1970.eko hamarkadako teoria linguistiko sortzaileek bultzatua, gramatikaren inguruko ikuspuntua aldatu egin zen, eta gramatika-erregelek maila lexikaleko auziak ezkutatzen zituztela nabaritu zen.

Gramatika-erregelak informazio lexikalari zuzenduta zeudela nabaritu zenean, lexikoietan gordetako informazioaren egituran ipini zen arreta, lexikoiak informazio linguistiko orokorraren gordelekua izatera pasa baitziren. Komunitate zientifikoa, hizkuntzari buruzko orokortasunaren muina dimentsio lexikalean sartzen zela konturatu zenean, lexikoiaren garrantziaz jabetu zen.

Horrela, lexikoiaren eraikuntza LNPko funtsezko ataza dugu gaur egun.

Izan ere, LNPrako sistemek, neurri errealeko testuekin lan egin behar badute, milaka sarrera dituzten baliabide lexikal aberatsak behar dituzte, ezinbestean. Baliabide lexikalen eraikuntza, baina, lan neketsua eta astuna da, eta, askotan, LNPko aplikazioak garatzeko topa daitekeen arazo larriena dela esan daiteke. (Briscoe, 1991) lanean aitortzen den bezala, lexikoiak LNPrako sistemen itogune nagusia izatera pasatu dira. Hori horrela, saio ugari egin dira, baliabide lexikal berri bat eraikitzeko garaian, aurretik dagoenaz baliatzeko, hots, informazio lexikala berrerabiltzeko. Konparazio batera, lexikoi konputazional bat huts-hutsetik eraikitzeko, hiztegi arruntetara jo daiteke lexikoi horrek beharko dituen hitz-sarreren zerrendaren bila.

LNPko aplikazioez gain, linguista konputazionalek edo lexikografoek ere behar handia dute, beren eguneroko lanetan, iturri lexikal anitzetara jotzeko. Datu-base lexikalak eraikitzeko edo aberasteko, hiztegigintzan, edo itzulpenlanak burutzeko, behar-beharrezkoa dute jadanik existitzen diren baliabide lexikal anitz kontsultatzea, direla hiztegiak, direla LNPrako garatutako datubase zein ezagutza-base lexikalak.

Hiztegi elebidun eta eleanitzak erabiltzearen premia areagotzen da gurea bezalako gizarte eleanitzetan. Hizkuntzaren industriak, horrela, tresna eleanitzen —*on-line* hiztegiak, itzulpengintzarako laguntza-sistemak, etab.— beharra du, eta, jakina, tresna hauen garapenak iturri eleanitzen atzipena eskatzen du.

Egun dauden iturri lexikalen izaera, baina, heterogeneoa da, ikuspuntu, xede eta teoria linguistiko desberdinetatik garatuak baitira: zerrenda sinpleak gordetzen dituztenetatik, maila anitzeko informazio linguistiko konplexua errepresentatzen dutenetaraino, baliabide lexikaletan zehar gordetako informazioa oso desberdina eta askotarikoa da.

Baliabide bakoitzak informazio lexikala bere erara gordetzen duelarik, bertara informazio bila jo nahi duenak —dela giza-erabiltzailea, dela LNPrako aplikazioa— ezagutu behar ditu iturri horrek eskaintzen dituen atzibide eta kontsulta-lengoaia bereziak. Cunningham *et al.*-en lanean (2000) aipatzen den bezala, informazio lexikala iturri anitzetatik jasotzeko bi erronka nagusiri egin behar zaie aurre:

1. each resource has its own representation syntax and corresponding programmatic access mode
2. resources must generally be installed locally to be usable, and how this is done depends on what operating systems are supported etc., which varies from site to site.

Horrela, baliabide lexikalen artean zenbait egitura komun dagoen arren —adibidez, lexikoi orok hitz-formak gordeko ditu—, oso zaila da, egileen aburuz, informazio komunaz baliatzea, nork bere erara kodetzen baitu informazio hori.

Informazio lexikala adierazteko formalismo estandar bat garatzeko ekimen ugari egin da, eta informazio lexikala aplikazio zehatzekiko zein teoria linguistikoekiko independentea den eredu bakar batean biltzeko proiektuek gaurdaino irauten dute (Normier eta Nossim, 1990; MacNaugth, 1990; Uszkoreit *et al.*, 1996; Calzolari *et al.*, 2002; Ruimy *et al.*, 1998; Bel *et al.*, 2000). Informazio lexikala adierazteko eredu estandar bat eratzea, baina, auzi benetan konplexua da, eta helburu horrekin sortutako ekimenak arazo larriekin topatu dira bidean. Estandarizazio-ekimenek, horrela, ez dute nahi bezainbesteko arrakasta bereganatu, ezta komunitate zientifikoaren baitan erabateko adostasunik lortu (Zajac, 1999).

Txosten honetako tesi-proiektua testuinguru honetan koka daiteke. Horrela, hemen aurkeztutakoa iturri lexikal anitzetan informazio bila dabilenaren beharrak asetzera dator. Hala ere, guk ez dugu baliabide lexikaletan zehar gordetako informazioa formalismo estandar batera bihurtuko, ezta iturri bakoitzaren formalismoak aldatuko ere. Horren ordez, iturri heterogeneo anitz "baliabide heterogeneoen federazio" moduko batean integratzea da guk landutako proposamena. Integrazio-proposamen honetan, alderdi hauek izan ditugu kontuan, oro har:

- *Informazio lexikalaren atzibidea.* Finean, hainbat iturritan adierazitako informazio lexikala eskuratzeko atzibide bateratua eskaini nahi diegu erabiltzaileei. Erabiltzaileak galdera bat jarri, eta guk zuzenean joko dugu iturri(eta)ra, erabiltzaileak jarritako galdera erantzungo duen datu sortaren bila. Emaitza bezala, iturri anitzetatik datozen datuak era bateratuan eskainiko zaizkio erabiltzaileari. Azken honek, bestalde, ez du arduratu behar, bere galdera adierazterakoan, informazioa zein iturritatik jaso nahi duen, baizik eta zein informazio nahi duen.

- *Iturrien independentzia.* Iturri bat ez da ezertan aldatu beharrik gure federazioan integratzeko. Esan bezala, guk ez dugu iturrietan dagoen informazioa formalismo batera bihurtuko, ezta bertoko datuen kopiarik egingo ere. Horrela, iturrien autonomia erabatekoa da, alegia, datu lokalak gehitzeko, aldatzeko edo ezabatzeko askatasun osoa izango dute. Gerta daiteke, beraz, iturriek "ez jakitea" ere federazioko parte-hartzaileak direnik.

Beraz, erabiltzaileari sisteman integratutako ezagutzaren eskema orokor bat eskainiko zaio, eta eskema horren gainean egin ahal izango ditu galderak, informazio lexikala kontsultatzeko lengoaia bakar batez baliatuz.

I.2 IXA taldea eta lexikografia konputazionala.

Hemen aurkeztutako tesi-lana Donostiako Informatika Fakultateko IXA taldearen barnean garatu da. IXA duela hamabost urte inguru jaio zen ikerkuntza-taldea da, konputagailuen bidez euskararako trataera automatikoa bideratzea xede nagusi, Lengoaia Naturalaren Prozesamenduan aritzen dena.

Taldearen barruan lexikografia konputazionaleko azpitaldeak baliabide lexikalak eta lexiko-semantika lantzen ditu. Azpitaldearen interesa, besteak beste, baliabide lexikalen formalizazioan datza, eta, baita ere, euskarazko baliabide lexikal egituratuen eraikuntza sendotzeko teknikak lantzean. Bere jardueran, taldeak badu eskarmenturik baliabide lexikalak sortzen, aberasten eta ustiatzen.

IXA taldearen iturri lexikal nagusia Euskararen Datu-Base Lexikala (EDBL) delakoa da (Aldezabal *et al.*, 2001). EDBL datu-base lexikal zabala da —egun 80.000 sarrera inguru ditu—, eta xede askotarako erabilgarria izateko bokazioarekin diseinatu da, alegia, LNParen arloko lanetan beharrezkoak diren lexikoien iturri nagusia izateko.

Lexikografia konputazionaleko azpitaldeak lan ugari garatu ditu hiztegien gainean, bertatik erauzitako informazioa lexikoi berriak garatzeko, edota dauden lexikoiak aberasteko asmoarekin. (Artola, 1993) lanean, egileak HIZTSUA ezagutza-basea eratu zuen *Le Plus Petit Larouse* (LPPL) frantsesezko hiztegi elebakarraren definizioetatik abiatuz. Bertan oinarrituz, hiztegi-sistema adimentsu baten prototipoa diseinatu da (Agirre *et al.*, 2000). HIZTSUA elebakarra bada, ANHITZ proiektuak eleaniztasunaren dimentsioa eransten du, frantses eta euskarazko bi ezagutza-base lexikal elebakarren arteko zubia eraikiaz, hiztegi elebidunetan oinarrituz (Arregi, 1995; Agirre *et al.*, 2001). Bi lan horiek izan ziren azpitaldean egindako lehen doktorego-tesiak.

(Arriola eta Soroa, 1996) lanean EH hiztegiko paperezko bertsiotik euskarri elektronikoan gordetako datu-base lexikala lortzearren eginiko urratsak ikus daitezke. EHko bertsio elektronikoaren informazioa ustiatuz, bestalde, hainbat lan burutu dira: (Arriola, 2000) tesi-lana euskarazko aditzen azpikategorizazioen gainean jorratu da, informazio-iturri gisa EHko aditzen erabile-

ra-adibideek osatutako corpusa hartu delarik. Ildo bera jarraitu da (Aldezabal, 2004) lanean, aditzen azpikategorizazioaren informazio aberatsa lortuz, Levin-en lanean oinarrituz. Horretaz gain, EHko definizioak aztertuz bertako kontzeptuen taxonomia garatu da (Agirre *et al.*, 2003), edota eratorrien informazioa landu da. Lexikografia konputazionaleko azpitaldeak informazio lexikalaren kontsulta-sistemak ere ikertu ditu, esaterako, Arregi *et al.*-en lanean (2003), zeinean EH hiztegiaren informazioa eskuratzeko atzibide aurreratuak azaltzen baitira.

EuskalWordNet ezagutza-basearen proiektua euskarazko wordnet baten eraikuntzan datza, EuroWordNet-eko gaitasun eleanitzak erabiliaz. Euskarazko ezagutza-base bat garatzeaz gain, EuskalWordNet beste hainbat hizkuntzatako wordnet-ekin erlazionatu ahal izango da, baliabide eleanitz estimagaitza lortuz (Agirre *et al.*, 2002).

Taldeak iturri anitz atzitzearen beharra izan du maiz. Konparazio batera LPPLtik eratorritako HIZTSUAko taxonomiaren ahuleziak aztertzerakoan —hala nola, kontzeptu gehiegi taxonomiako goi-mailatan sailkatzea, edota begizten agerpena— , zera aipatzen da Agirre-ren lanean (1999):

> Ezagutza-baseen artean zubiak ezartzea erabilgarri suertatzen da oso. LNP orokorrerako sistema batek behar duen ezagutza ez da normalean iturri bakarrean egoten, horretarako espreski eskuz sortutako ezagutza-base bat egin ez bada behintzat. Halakoetan ere beti da aberasgarria beste ezagutza-iturrietarako zubiak sortzea.

Izan ere, ezagutza-base lexikalak garatzean edota aberastean komenigarria da zinez iturri bat baino gehiagotara jotzea (Ide eta Véronis, 1994).

Horrela, bada, taldea iturri lexikal anitz eta heterogeneoz baliatzen delarik, iturri horien atzibide bateratua aztertzea izan da tesi-proiektu honen estreinako motibazio nagusietako bat.

I.3 Ingeniaritza linguistikoa.

Hemen aurkeztutako tesi-lana ingeniaritza linguistikoan (IL) kokatu behar da. Lengoaia Naturalaren Prozesamenduko proiektuetan ingeniaritza-tekni-

kak[1] betidanik erabili badira ere, 90eko hamarkadan jaio zen Ingeniaritza Linguistikoa[2] izenarekin ezagutzen den arloa.

LNParen helburua lengoaia naturalaren, hots, hizkuntzaren azterketa dela esan daiteke, linguistika tradizionalaren antzera; aitzitik, LNPak konputagailuak erabiltzen ditu teoria linguistikoen zatiak modelatzeko —edo egiaztatzeko edo baliogabetzeko—. Testuinguru honetan, ILak LNPko arazo berak aztertzen ditu, baina ikuspuntu praktiko bat hartuz.

IL arloa sortu izanaren arrazoiak ugari dira. LNPko estreinako sistemen "jostailu sindromea" erabat gainditzeko asmoarekin, industriak mundu errealeko testuen gainean lan egiteko gai ziren tresnak behar zituen. Hartara, epe motzean edo ertainean zeregin zehatzak ebazteko tresna eraginkorrak garatzearen beharra bultzatu zuen, eta LNPko sistemen gaineko ebaluazio enpirikoa gauzatu behar zela agerian utzi.

Hein handi batean, ILa 90eko hamarkadan LNPan jazotako paradigma-elkarketatik jaio zen. Hamarkadaren hasierako LNPko sistemek bi paradigma desberdini jarraitu ohi zieten: bata ezagutza linguistiko sinbolikoan oinarritua (teorikoa, nolabait esateko), eta bestea estatistikan oinarritua (enpirikoa). Paradigma bien arteko elkarketak, hots, estatistikaz eta ezagutza linguistikoaz baliatzen diren sistema hibridoen garapenak, software-sistema egonkorrak, doiak eta eraginkorrak sortzeko aukera eman zuen.

Horretaz gain, 90eko hamarkadan egindako baliabide lexikalen berrerabilgarritasunaren azterketak —euskarri elektronikoan gordetako hiztegien informazioaren eskurapena, neurri handi batean— testu errealak prozesa ditzaketen sistemak garatzeko aukera eman zuen, eta, ondoren, mundu errealeko zenbait arazori aurre egiteko beste ziren sistemen garapena ekarri zuen. Honek LNPrako sistemak osatzeko metodologia berriak ezarri zituen.

Cunningham-en aburuz (1999) honako faktore hauek izan ziren ILaren arloa sustatzearen zioak, besteak beste:

[1]Ingeniaritza terminoarentzat definizio zorrotza ematea lan honetatik at dago, eta luzeegi joko liguke. Hala ere, termino horretaz ulertzen duguna azaltzearren, zientzia terminoarekin alderatuko dugu, bien arteko aldeak argigarriak izango direlakoan. Horrela, bada, esan daiteke zientziaren kezka munduaren funtzionamenduaren aurkikuntzan datzala, eta, ingeniariarena, berriz, ezagutza zientifikoaren erabileran, irizpide batzuen arabera taxuz jokatuko duten tresnak garatzearren (Cunningham, 1999). Horrela, ingeniaritzan kokatutako proiektuek egokitasun-irizpideak —seguruenik kanpotik etorriko direnak— jarraitu behar dituzte beti.

[2]Giza-lengoaiaren teknologiak ("Human Language Technologies", HLT) ere deiturikoa.

- Konputagailuen hardwarearen aurreratzeak azkarragoak diren prozesadoreak eta memoria handiagoa kudeatzeko aukera ematen du. Horrela, konputazio-zama handia behar duten prozesuak garatzea bideragarria da egun, eta sistema hauen inplementazioa kostu ekonomiko txikiez egin daiteke.

- Neurri erreala duten eta Internet saretik *on-line* eskuragarriak diren iturri linguistikoen kopuruaren handitzea, hala nola, hiztegiak, *thesaurus*ak, eta corpusak.

- Testu elektronikoen kopurua etengabe handitzen doan mundu batek, komunikazio elektronikoek zein mugikortasunak komunikazio eleanitzen garrantzia areagotzen duelarik, aplikazioen gainean eskari berriak ezartzen ditu.

- LNPko teknologiaren heldutasunak aukera ematen du, zenbait lanetarako, zehaztasun handiko aplikazioak sortzeko.

- Horri gehitu behar zaizkio softwaregintza arloan sortutako softwareingeniaritzaren emaitza agerikoak, hala nola, osagaietan oinarritutako softwarea edo diseinu-patroien erabilpenaren ekarpen garrantzitsuak.

Lan berean aurki dezakegu ILaren definizio bat (Cunningham, 1999):

> Language Engineering is the discipline or act of engineering software systems that perform tasks involving processing human language. Both the construction process and its outputs are measurable and predictable. The literature of the field relates to both application of relevant scientific results and body of practice.

Egilearentzat ILa komunitate zientifikora ekarpen garrantzitsuak dakartzan diziplina bat da, bere publikazio eta kongresu bereziekin. ILa softwarea garatzeko ingeniaritzaren arloan —eta, bereziki, software-ingeniaritzan— kokatu behar da, eta bere emaitza, azken finean, software-sistemak dira. Arloak kezka eta lehentasun propioak ditu, eta, oro har, mundu errealeko arazo praktikoei aurre egitea da bere helburu nagusia. Auzi horretan, edozein teoria linguistiko onartzeko prest dago, beti ere emaitza hobeagoetara badarama. Historikoki LNPko sistemen erabiltzaileak maiz linguistika arlokoak izan badira ere, ILko sistemek linguistak ez direnentzat ere baliagarriak izan behar dute, eta, ondorioz, merkatuak zein industriak ezarriko ditu, maiz, IL proiektuen egokitasun-irizpideak (Gazdar, 1996).

Esan bezala, ILko kezkak geureganatu ditugu tesi-proiektu honetan. Informazio lexikala integratzearen arazoari aurre egiteko, datu-baseen arloan garatutako datu-integrazioko teknikez baliatu gara. Auzi horretan, honako alderdi hauek izan ditugu irizpide nagusi:

- *Praktikotasuna.* Sistemaren garapenean oso kontuan izan dugu bere erabilera praktikoa, eta erabiltzaileentzat lagungarria den tresna sortzera bideratu ditugu gure indarrak. Erabiltzaile sorta handi batentzat baliagarria izango den sistema bat garatu nahi izan dugu: hizkuntzalaritza zerikusirik ez duten erabiltzaileak, linguistak zein lexikografoak, edo LNPko beste aplikazio automatikoak.

- *Sendotasuna.* Gure asmoa ez da jostailuzko sistema bat garatzea, aitzitik, tamaina errealeko eta egun erabilera zabala duten baliabideak atzitu nahi ditugu gure sistemaren bidez, bertan gordetako informazioaz baliatzearren.

- *Eraginkortasuna.* Sistemak denbora onargarrian bete behar ditu bere eginkizunak. Horrela, erabiliko ditugun teknikek dakartzaten zamaren azterketa egin dugu.

I.4 Datu-integrazioa. Sarrera gisa.

Egin dezagun kontu hainbat informazio-iturri desberdin ditugula eskura, eta iturri horietan gordeta dagoen informazio guztia eskuratzeko atzibide bateratua nahi dugula. Oro har, informazioaren integrazioaren arloan egindako lanen arabera, bi hurbilpen desberdin jarrai ditzakegu gure helburua betetzeko.

Hurbilpen bat —*eskemaren integrazioa* delakoaren hurbilpena (Batini *et al.*, 1986; Seth *et al.*, 1993)—, iturri bakoitza birdiseinatu eta berrinplementatzean datza, iturrien informazio guztiaren adierazpen homogeneoa duen informazio-iturri bat garatzearren. Hartara, erabiltzaileak adierazpen homogenoaren bitartez egingo ditu galderak, eta erantzunak iturri guztietatik jasoko ditu.

Hurbilpen honek, baina, iturriak berrantolatzera behartzen gaitu, eta, askotan, berrantolaketa hori ezin izango da egin. Guk planteatzen dugun integrazio-prozesuan, iturrien berezitasunak mantendu nahi ditugularik, ez dirudi *eskemaren integrazioaren* hurbilpena lagungarria izango dugunik.

Badago, hala ere, iturri anitz eta heterogeneoen informazioa elkarren artean trukatzeko beste biderik. *Datu-integrazioa* delako hurbilpenean (Jarke *et al.*, 2000), iturriak beren horretan eta ukitzeke lagatzen dira, eta, integrazioa burutzeko, erabiltzaileak jarritako galdera edo iturrietako datuak eraldatzen dira. Eraldaketa gauzatzeko baliabideen artean mapaketa-erregelak adierazten dira, direla erregela prozeduralak, direla eskemen arteko murriztapen deklaratiboak. Datu-integrazioa galdera eta eskema-unitate mailan soilik egiten bada, *integrazio birtuala* ("virtual integration") egiten dela esaten da. Integrazioa datu-mailan egiten bada, hots, iturrietako datu heterogeneoak sailkatu, eraldatu eta fusionatzen badira, *integrazio-eskema* baten arabera datu homogeneoak izango balira bezala agertzearren, *gauzatutako integrazioa* ("materialized integration") dela esaten da.

Integrazio-arkitekturak bi mota nagusitan sailkatu ohi dira: eskema orokor bat behar dutenak —zeinaren bitartez iturriak integratzen diren—, eta eskema orokorrik behar ez dutenak. Horrela, zenbait arkitekturak —esaterako, datu-base anizkoitzek ("Multidatabases", Heimbigner eta McLeod, 1985), datu-base federatuek ("Federated Databases", Seth eta Larson, 1990), edo informazio-sistema kooperatiboek ("Cooperative Information Systems", adibidez, Mena *et al.*, 1996)— ez dute horrelako integrazio-sistema bakar baten beharrik. Informazio-sistema globalak ("Global information systems", Ives eta Jarvenpaa, 1991) edota datu-biltegiak ("Data Warehouses", Jarke *et al.*, 2000), aldiz, integrazio-eskema orokor batean oinarritzen dira integrazioa gauzatzeko.

Bestela, integrazio-sistemek, oro har, hiru osagai nagusi izan ohi dituzte (Calvanese *et al.*, 2001): iturri lokalen eskemak, eskema orokor bat —edo ez— eta eskemen arteko mapaketak. Iturri lokalek gordetako datuen antolamendua eskema lokaletan adierazten da. Eskema orokorra, bestalde, iturrien bista bateratua bezala ikus daiteke, eta erabiltzaileari, sistemari galderak egiteko garaian, abiapuntu bateratua eskainiko dio. Azkenik, eskema lokalen eta eskema orokorraren artean baliokidetza gauzatzen duten mapaketak behar dira.

Integrazio-sistemei helarazitako galderen gainean dedukzio-prozesuak burutu behar dira maiz. Hartara, lengoaia aberatsez errepresentatu ohi dituzte sisteman parte hartzen duten iturrien eskemak, eta baita eskema orokorra ere. Izan ere, informazioaren integrazioak adimen artifizialaren arloarekin erlazio estua baitu (Levy, 1998). *Deskribapen-logikak* (DL) objektu egituratuak errepresentatzeko formalismo logikoak dira, objektuen gainean arrazoiketak bideratzen dituztenak. Datuen modelizaziorako maiz erabiliak dira, eta inte-

grazio-sistema anitz baliatzen da DLez sistemaren ezagutza errepresentatzeko.

Integrazio-sistema baten arazo larriena, jakina, iturrien heterogeneotasunak ezarria da. Iturrien arteko heterogeneotasuna, halaber, hainbat mailatan sailkatu ohi da: integrazio semantikoa, integrazio estrukturala eta datu-mailako integrazioa.

- *Integrazio semantikoa.* Integrazio semantikoa gauzatu behar da baldin eta iturriek kontzeptualizazio desberdina erabiltzen badute informazio komuna adierazteko. Hor barne koka daitezke, besteak beste, honako arazo hauek:
  - *Izen-gatazkak.* Eskemek izenen bidez erreferentziatzen dute modelatu nahi duten domeinua, kontzeptuen, erlazioen edo atributuen bidez normalean. Domeinuko informazio bera erreferentziatzeko, baina, nork berea egin ohi du, izen eta terminologia desberdinak erabiliaz. Horrela, izenen arteko gatazkak sortzen dira.
  - *Domeinu-gatazkak.* Iturrien artean, izen bereko kontzeptuek domeinu desberdinak eta, are, inkonsistenteak izan ditzakete. Esate baterako, aditzak soilik gordetzen dituen iturri bateko `Hitzak` entitateak aditz-formak gordeko ditu; beste iturri batek, ordea, izen bera erabil dezake kategoria orotako hitz-formak gordetzeko. Nahiz eta bi iturrietako kontzeptuek izen bera izan, bertan gordetako datuen domeinua desberdina da.
  - *Moten arteko desberdintasuna.* Kontzeptu bera bi era desberdinetan dago adierazita baliabideetan. Adibidez, batek entitate bezala adierazten duen bitartean, besteak, berriz, erlazioen bidez.
- *Integrazio estrukturala.* Integrazio estrukturala dago iturriek informazioa errepresentatzeko jarraitutako erak desberdinak direnean. Ondorioz, iturriak datu-eredu, programazio-interfaze eta kontsulta-lengoaia desberdinez baliatzen dira informazioa biltegiratzeko.
- *Datu-mailako integrazioa.* Datu-mailako integrazioa maila estentsionalean gauzatzen da. Horrela, bere helburu nagusia datuak konparatzean datza, esaterako, munduko entitate erreal bera adierazten duten bi instantzia izanik —bi baliabide desberdinetatik jasoak—, hauek berdinak edo, behintzat, baliokideak direla erabakitzeko ahalmena izatean. Horri

esker, baliabide lokal bakoitzak entitate horri buruz duen informazioa erlaziona eta aldera daiteke.

Bestalde, datu-integrazioaren arloan eginiko lanak bi norabide orokorretan sailka daitezke, integrazioa gauzatzeko eskemen arteko mapaketak adierazteko metodoaren arabera: *globala bistatzat* ("Global as view", GAV) eta *lokala bistatzat* ("Local as view", LAV) paradigmak (Ullman, 1997; Florescu *et al.*, 1998b).

GAV da paradigma ohikoena. Paradigma honetako mapaketa-erregelek —askotan bitartekoak ere deiturikoak— entitate birtualak inplementatzen dituzte, eta beren interfazeek iturri heterogeneoen gaineko bistak esportatzen dituzte, heterogeneotasuna ebazteko iturrietako datuak konbinatzeko egon daitezkeen bideak zehaztuz. GAV hurbilpeneko bitartekoak datu-baseko bista arrunten hedapena izan daitezke —adibidez, SQL-ko CREATE VIEW eraikitzaileen bidez—, edota kode prozedural batez inplementa daitezke.

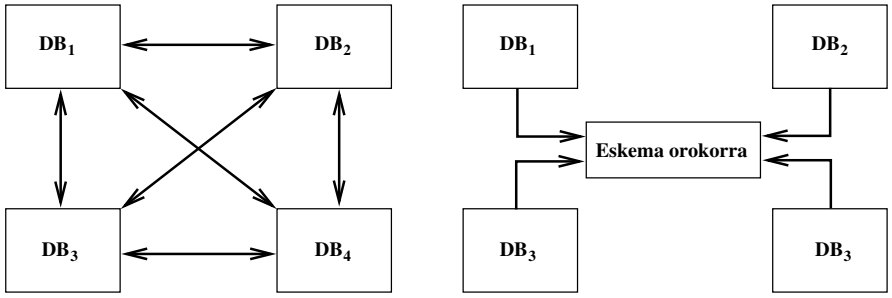
LAV hurbilpenean, aldiz, mapaketa-erregelen noranzkoa GAVekoaren alderantzizkoa da. *Bitarteko eskema* baten gainean eginiko galderak erantzuteko, iturriak eskemaren bidez gauzatutako bistak izango balira bezala hartzen dira, non galdera erantzuteko bista hauek soilik erabil daitezkeen. Horrela, LAV hurbilpenaren pean, galderen berridazketak datu-baseen komunitatean sakon ikertutako *galderen erantzutea, bistetan oinarrituz* ("Answering queries using views": Levy *et al.*, 1995; Rajaraman *et al.*, 1995; Duschka eta Genesereth, 1997a; Abiteboul eta Duschka, 1998; Calvanese *et al.*, 1999a; Levy, 2000; Calvanese *et al.*, 2001) delako arazoarekin zerikusi handia du, eta, azken honek, *galderen barne-hartzearen* ("Query containment", Chan, 1992; Florescu *et al.*, 1998a) arazoarekin.

LAV hurbilpenaren abantaila nagusia informazio-iturriak adierazteko malgutasunean datza, batez ere, iturri berriak gehitu, ezabatu edo aldatu nahi direnean. Hala ere, galderaren itzulpena, LAV hurbilpenean, prozesu konplexua da, informazioaren eta domeinu-ereduaren artean mapaketa semantikoa deskribatzeko erabili den lengoaiaren espresibotasunaren araberakoa; izatez, adierazpen-ahalmen handiko lengoaietarako erabakiezina den prozesua izan daiteke (Abiteboul eta Duschka, 1998).

GAV hurbilpenean, berriz, galderen itzulpena prozesu sinpleagoa da (Ullman, 1997). Hala ere, iturriak gehitzea edo aldatzea sistema osoan eragin zuzena duen prozesua izango da, eta, horrela, iturri batean aldaketak egiteak sistema osoa birplanteatzea ekar dezake. Edonola ere, itzulpena gauzatu ahal izateko behar-beharrezkoak diren bitartekoen definizioaren lanari ekite-

ko, baliabideen arteko erlazioen ulermen sendoa behar da eta, hortaz, lan konplexua bilakatzen da. Horrela, bada, esango da LAV hurbilpenean galdera-itzulpenaren prozesua exekuzio-denboran gauzatzen dela, eta GAV hurbilpenean, berriz, itzulpena diseinu-mailan gauzatuko dela. Ondorioz, GAV hurbilpenaren pean kokatutako integrazio-sistemen problema larriena bitartekoen definizioan datza.

Edonola ere den, integrazio-sistema baten garapena lan konplexua da oso, eta datu-integrazioa helburu duen edozein sistemaren garapenak hainbat buruhauste aurkituko ditu bidean.



I.1 Irudia: Datu-baseen arteko mapaketak. a) Datu-base federatuetan b) Informazio-sistema globaletan

Arestian esan bezala, datu-integrazioko teknikez baliatu gara lan honetan, informazio lexikalaren integrazioa helburu. Horrela, iturri heterogeneo anitz "baliabide heterogeneoen federazio" moduko batean integratzea da guk landutako proposamena. "Federazio" hitza aipatu badugu ere, termino hori bere zabaltasunean erabiltzen ari gara, eta, bereziki, ez da datu-baseen arloko federazio terminoarekin nahastu behar. Horrela, gure proposamenak, integrazioa gauzatzeko, iturri lokal guztiak eredu orokor komun bakar batekin parekatzen ditu. Datu-baseen federazioetan, aitzitik, mapaketak iturri guztietatik iturri guztietara dira. Horrela, n datu-base izanik, n^2 mapaketa beharko lirateke federazioa osatzeko, eta n soilik informazio-sistema globala osatzeko (ikus I.1 irudia).

I.5 ELHISA.

Atal honetan ELHISAren aurkezpen orokor bat egingo dugu, eta bere ezaugarri esanguratsuenak aurkeztuko ditugu. ELHISA, Ezagutza Lexikal Heterogeneoa Integratzeko SistemA, informazio lexikala eskuratzeko integrazio-sistema da. ELHISAren helburua, berriz, hainbat iturri lexikaletara informazio bila jo behar duenari atzibide bateratua eskaintzea litzateke.

Zinez komenigarria litzatekeen arren, informazio lexikal guztia errepresentatzeko aukera ematen duen formalismo estandar baten zehaztapena ez da etorkizun ertainean ikusten, komunitate zientifikoan onarpen zabala izan nahi badu, behinik behin.

ELHISAk ez du informazio lexikalaren adierazpide edo errepresentazio estandar bat izateko bokaziorik. Hori baino, iturriak ukitu gabe beren baitan gordeta dagoen informazioaz baliatzea da guk proposatutakoa. Helburu horrekin, bada, iturri lexikalak "baliabide heterogeneoen federazio" batean integratzea da guk proposatuko duguna.

Sisteman integra daitezkeen baliabide lexikalen mota oso zabala da: atributu-balio bikoteak gordetzen dituen fitxategi sortetatik, ezagutzan oinarritutako sistemetan errepresentatutako ezagutza lexikaleraino, oso informazio desberdin eta heterogeneoen biltokia izango da guk garatutako sistema. Halaber, iturri bakoitzak bere antolamendu propioa izango du, hots, datu-eredu, kontsulta-lengoaia eta datuak fisikoki gordetzeko egitura bereak.

ELHISA integrazio-sistema bat da, eta sistema garatzerakoan oso kontuan izan dugu, esan dugun bezala, iturrien berezitasuna eta independentzia mantentzea. Horrela, ELHISAn integratutako baliabideek "bizitza propioa" izaten jarraituko dute, alegia, bakoitzak hainbat bezeroren hornitzaile lexikala izaten jarraituko du. Iturrien gainean, halaber, informazioa gehitu, ezabatu edo aldatzeko aukera dago, eta ELHISA aldaketa horietaz automatikoki jabetuko da.

ELHISA *informazio-sistema globala* da: integrazioa gauzatzeko, eskema orokor batean oinarritzen da, Eredu Kontzeptual Orokorra (EKO) deiturikoa. Iturri lokal guztien arteko harremana, horrela, EKOaren bitartez gauzatuko da. Izan ere, iturri bakoitzaren edukiak EKOarekin parekatuko dira, mapatze-erregelen bitartez.

Erabiltzaileak, ELHISAri informazio eskatzeko, EKOa izango du informazio lexikalaren atzibide bateratua: bere galderak egiteko, EKOtik zein informazio eskuratu nahi duen zehaztu behar du. Hala ere, erabiltzaileak ez

du zehaztu behar eskatutako informazioa zein baliabidetatik jaso nahi duen. Izan ere, erabiltzaileak ez baitu, berez, ELHISAn integratutako iturrien berezitasunak —datu-eredua, kontsulta-lengoaiaren sintaxia, antolamendu fisikoa eta abar— zertan ezagutu. Sistema arduratuko da eskatutako informazioa erantzutearren egin beharreko urrats guztiez, eta iturrien arteko heterogeneotasunak dakartzan arazoak ebazten. Horrela, galdera bat jaso ondoren, galdera erantzungo duten iturri lokalak identifikatu, iturri horiei galderak igorri, eta iturrietatik datozen emaitzak jasoko ditu. Ataza horiek guztiak automatikoki exekutatuko dira, erabiltzailearen parte hartzerik gabe.

EKOa *birtuala* da, bertan deskribatzen diren kontzeptu eta erlazioak ez baitira "errealak", alegia, erlazioek ez baitute gauzatze fisikorik. Galdera erantzuteko, horrela, iturrietara jo behar da: erabiltzaileak jarritako galdera —EKOaren arabera adierazita dagoena— iturri bakoitzak ulertzen duen eredura itzuli behar da. Itzulpenean ziurtatu behar da, jakina, iturrietara heltzen diren galderen esanahia jatorrizko galderaren baliokidea dela. ELHISAk, horrela, *integrazio birtualeko* paradigma jarraitzen du integrazioa gauzatzeko, integrazioa galdera prozesatzeko garaian burutzen baitu.

Galderak itzultzeko, EKOa eta iturriak harremanetan jartzen dituen mapaketa semantikoa zehaztu behar da. Iturri bakoitza modelizatu egiten da, bertan gordetako informazioaren gainean kontzeptualizazio bat gauzatuz, klaseen, atributuen eta klaseen arteko erlazioen bidez. Horrela, mapaketa semantikoak iturriaren modelizazioa —Baliabidearen Eredu Kontzeptuala (BEK) izena duena— EKOarekin erlazionatzen du, eta horretarako mapaketa-erregelez baliatzen da.

Mapaketa-erregelak LAV ereduari jarraiki osatzen dira. Horrela, bada, iturriko kontzeptu, atributu zein erlazio oro EKOaren arabera deskribatu egiten da, iturria EKOaren bista bat izango balitz bezala. LAV paradigma jarraitu izanaren arrazoia aztertuko dugu lan honetan zehar, eta izan dituen ondorioak azpimarratuko ditugu. Edonola ere, esan dezagun LAV paradigmak sistemaren malgutasuna areagotzen duela, iturri bakoitza isolamenduan definitzen baita. Esaterako, sisteman iturri bat gehitzen bada, iturria definitzeko behar diren mapatze-erregelek ez dute aurretik integratutakoen gainean inongo aldaketarik zertan ezarri.

Esan dugun bezala, ELHISAn integratutako iturrien barne-antolamendua oso desberdina izan daiteke, testu-fitxategi lauak —kontsultak egiteko aplikazio bereziak behar izan ditzaketenak—, datu-base tradizionalak, ezagutza-baseak eta abar. Iturri orok software-modulu bat izango du atxikita, *wrapper* izenekoa, zeinek iturriaren berezitasunak estaliko dituen, iturriaren eta siste-

maren arteko bitartekari-lanak burutuz. Iturri bakoitzeko *wrapper*-ak, bada, iturriko ereduarekin bat datorren galdera jasoko du, eta galdera hori iturriaren kontsulta-lengoaiaren arabera berridatziko du. Galderaren emaitzak jasoko ditu, eta sistemara berriro bidaliko. *Wrapper*-en teknologiak, hortaz, integrazio estrukturala deritzonari irtenbide bat emateko aukera eman digu.

Iturrietatik galderen erantzunak jaso ondoren, ELHISAk *datu-arazketako* prozesu bat burutzen du datuen gainean, iturri desberdinetatik datozen datuak elkarrekin erlazionatuz. Prozesu horretan bi eginkizun nagusi burutuko ditu. Batak, *datu-garbiketa* deiturikoa, iturriek informazioa gordetzeko izan ditzaketen zenbait errore eta akats zuzenduko ditu, eta, baita ere, zenbait eremuren balioak —kategoria lexikalak, azpikategoriak eta abar— normalizatu egingo ditu[3]. Bigarrenak, *objektuen identifikazioa* delakoak, hainbat iturritik datozen objektuek mundu errealeko entitate bera erreferentziatzen dutenetz egiaztatuko du, eta, hala izanez gero, bikoizketak ezabatuko ditu. Horrela, erabiltzaileak iturri desberdinetatik datozen datuak erkatu ahal izango ditu, iturrien arteko erlazioak antzemanez.

I.6 Txostenaren eskema.

Bigarren kapituluan baliabide lexikalez arituko gara. Informazio lexikalak LNPan izan duen eragina aztertuko dugu, eta informazioa era estandar batean errepresentatzeko izan diren saiakera eta proiektuak azalduko ditugu labur. Halaber, lexikoiak integratzeko egin diren zenbait lan aztertuko ditugu. Gero, informazio lexikala euskarri elektronikoan gordetzeko dauden eredu desberdinen azterketa bat ere egingo dugu, eta, azkenik, baliabide lexikalen sailkapen moduko bat burutuko.

Hirugarren kapituluan informazio-integrazioan ipiniko dugu arreta. Arloa aurkeztuko dugu, eta bere ezaugarri nagusiak aztertuko. GAV eta LAV hurbilpenen arteko bereizketan sakonduko dugu, eta bigarren hurbilpenerako dauden galderen itzulpenerako zenbait algoritmo ikusiko ditugu. Deskribapen-logikei ere tarte bat egingo diegu, informazio-integraziorako erabilpen handikoak baitira, eta ELHISAn horiez baliatu baikara hainbat eginkizun burutzeko. Azkenik, informazio-integrazioko hainbat sistema ikusiko ditugu.

Aurreko bi kapitulu bibliografikoen ondoren, laugarren kapituluan ELHI-

[3]Horretarako, zenbait estandarizazio-ekimenen —esaterako, PAROLE (Ruimy *et al.*, 1998) edo EAGLES (Heid, 1996)— proposamenak geureganatu ditugu.

SA sistema aurkeztuko dugu. Sistemaren ezaugarriak, arkitektura eta moduluak azaldu ondoren, bere osagaiak aztertuko ditugu.

Bosgarren kapituluan, berriz, ELHISAren funtzionalitateaz arituko gara, eta, halaber, sistemari egindako galderen itzulpenaren prozesuan sakonduko dugu. Itzulpen-prozesuaren ebaluazio bat aurkeztuko dugu, eta erabilera-adibide batzuk ikusiko ditugu.

Seigarren kapituluan ELHISAn iturri berri bat integratzeko burutu behar diren urratsak deskribatuko ditugu, horretarako kasu praktiko batean oinarrituz.

Azkenik, zazpigarren atalean konklusioak eta aurrera begirako lanak aztertuko ditugu. Bertan azalduko ditugu, ELHISA beste ingurune batzuetan aplikatzeko posibilitateekin batera, lan honen ondoren ikergai gertatutakoak, eta, gure ustez, aurrera egiteko bide interesgarrienak izango liratekeenak.

II. KAPITULUA

Baliabide lexikalak.

Kapitulu honetan baliabide lexikalei buruz arituko gara. Baliabide lexikala hizkuntza bateko —edo hainbat hizkuntzatako— hitzen edo/eta hitz-adieren informazio morfologiko/morfosintaktiko, sintaktiko eta semantikoaren biltegia da. Lengoaia Naturalaren Prozesamenduan oinarrizko baliabidea da, eta behar-beharrezkoa da hizkuntzaren tratamendu sendoa egin behar bada. Baliabide lexikala edo *lexikoi* terminoa erabiltzen dugunean, lengoaia naturalaren prozesamenduko edozein prozesutako informazio lexikalaren biltegiak adierazi nahi dugu[1].

Lehenengo atalean baliabide lexikalek Lengoaia Naturaleko Prozesamenduaren (LNP) arloan duten papera aztertuko dugu, eta lexikoiaren garapenean egungo joerak ikusiko ditugu. Hurrengoan, sarrera lexikalen estandarizazioa helburu izan zuten —eta duten— hainbat proiektu aztertuko ditugu, eta proiektu hauek aurkitutako arazo nagusiez mintzatuko gara. Informazio lexikalaren integrazioari begira egin diren zenbait proiekturi ere erreparatuko diegu. Gero, informazio lexikala gordetzeko usuen erabilitako hainbat errepresentazio ikusiko ditugu. Azkenik, baliabide lexikalen sailkapen sinple bat aurkeztuko dugu.

[1]Lexikoiari buruz hitz egiten dugunean, Wilks *et al.*-en artikuluan aipatutako esanahia geureganatzen dugu (1998):

> [A lexicon is] a set of formalized entries to be used in conjunction with computer programs and by dictionary the physical printed text giving lexical information, including meaning descriptions.

II.1 Baliabide lexikalak eta LNPa.

Atal honetan lexikoi konputazionalek LNPan izan duten —eta izaten jarraitzen duten— papera aztertzeari ekingo diogu. Jadanik aipatu dugu —I.1 atalean— baliabide lexikalen garrantzi handia LNPan: nekez jarriko du inork zalantzan, gaur egun, LNPko edozein sistemaren muina biltegi lexikalean datzala, neurri handi batean. LNParen helburua prozesu linguistikoen automatizazioa da, hala nola, lengoaiaren eskuratze, ulertze eta sorkuntza, eta prozesu horiek hizkuntza baten hiztegiaren ezagutza sakona behar dute, domeinu zabalekoak eta sendoak izatea nahi bada.

Hala ere, lexikoien garrantzia ez da beti hain handia izan LNPan. Aitzitik, urte askotan LNPko biltegi lexikalak linguistika konputazionalaren *anaia pobreak* izan dira (Boguraev 1991:3). Garai horietan, LNPrako sistemak aplikazio zehatzetara mugatzen ziren batez ere, eta garatutako sistemak aplikazio espezifikoetarako baino ez ziren baliagarriak izaten.

Bestalde, garai horietako sistemei "jostailu tresnak" esan ohi zitzaien, laborategietan sortutako arazoei aurre egiten saiatzen baitziren maiz. Sistema horiek garrantzi txikia eman ohi zioten informazio lexikalari, eta, horrela, lexikoi oso murritzak eta txikiak behar izaten zituzten beren eginkizunak betetzeko. Hala ere, arazo larriekin —saihestezinak gehienak— topatzen ziren laborategitik atera eta mundu errealeko domeinu zabaleko testuekin lan egin behar zutenean.

Whitelock *et al.*-en lanean (1987) hainbat sistema hartzen dira aztergai eta, beste neurri batzuen artean, sistema hauetan baliabide lexikalen batezbesteko tamaina aztertzen da. Emaitzak nahikoa adierazgarriak dira: batez beste, sistema zahar huen lexikoien tamaina 1500 hitzekoa zen (kopuru hau 25 hitzetara jaisten zen itzulpen automatikoko sistema konplexu bi kontuan hartzen ez baziren). Horrela, bada, sistema hauen ahultasuna azpimarratzen da, laborategitik atera eta erabilpen errealerako badira.

Lengoaia Naturalaren Prozesamendua industriako arlora lerratu zen neurrian, argi ikusi zen konputagailuen bidezko lengoaia-sistemek laborategitik atera eta "mundu errealera" egokitu behar zirela. Hizkuntzaren industriako produktuek, bai zuzentzaile ortografikoek, informazio erauzketarako sistemek edo itzulpen automatikoko sistemek, baliabide lexikal sendoak behar zituzten testu errealekin lan egitekoak baziren. Gauzak horrela, baliabide lexikalen garapena hizkuntz industriaren oinarriko ataza bihurtu zen.

Honen pean dagoen ideia nagusia zera da: emaitza onargarriak emango

dizkigun LNPko edozein atazak ondo egituratutako eta neurri erreala duen lexikoi aberats batean egon behar duela oinarrituta, ezinbestean. Lexikoiaren informazioak aberatsa izan behar du, eta informazio horren atzipen azkarra egiteko aukera eman behar zaie, bai aplikazio lexikografikoen erabiltzaileei, eta baita aplikazio automatikoei ere.

Ez da harritzekoa, beraz, Calzolari-ren lanean (1994) irakur daitekeen honako pasarte hau:

> It is almost a tautology to affirm that a good computational lexicon is an essential component of any linguistic application whithin the so-called "language industry".

Beraz, lexikoaren eraikuntza LNPko funtsezko ataza dugu, are gehiago lengoaiaren industria sortzen eta garatzen ari den heinean. Mundu errealeko sistemek milaka sarreratik gora dituzten baliabide lexikalekin lan egin behar dute, ezinbestean. Nabarmentzekoa da gaur egungo joera, hastapenetakoarekin alderatuz gero, erabat aldatu dela. Neurri handi batean, linguistika teorikoaren zein konputazionalaren egungo joera hizkuntz ezagutza gramatikalaren arlotik lexikoarenera lerratu da.

Informazio lexikalaren inguruko joera-aldaketa honetan garrantzi handia izan zuen *Marina di Grosseto* hiri italiarrean gertatutako *Automating the Lexicon: Research and Practice in a Multilingual Environment* izeneko mintegiak, baliabide lexikalekiko kontzientziak esnatzea eta hauen garrantziaz jabetzea lortu baitzuen. Bere antolatzaileen esanetan, laurogeiko hamarkadan suertatutako baliabide lexikalei buruzko interesaren gorapen izugarriari aurre egiteko burutu zen konferentzia hau (Walker *et al.*, 1994). Lexikoarengan suertatutako interes honen zergatia honako faktore hauetan oinarritzen zela adierazi zuten:

- Linguistika teorikoaren arloko ikerketek informazio sintaktiko eta semantikoaren funtsezko biltegia lexikoia dela erakusten dute.

- LNPrako sistemen bideragarritasuna maila internazionaleko milaka sarrera lexikaletik gora hornitutako sistema banatu handien mende dago.

- Baliabide lexikal hauen garapenak ahalegin erraldoia eskatzen duela azpimarratu zen, eta lexikoien eraikuntzak behar duen ahalegina LNPrako sistemen atazarik neketsuenetakoa zela onartu. Lexikoiak garatzeko orduan, halaber, hauek aplikazio zehatzen nahietara soilik egokitzearen

joera salatu zen. Izan ere, sistema jakin batentzat garatutako baliabide lexikal bat nekez erabil baitzitekeen helburu desberdinekin diseinatutako beste sistema batean. Horrela, bada, ikuspuntu-aldaketaren beharra ikusi zen baliabide lexikalen eraikuntza-lanetan: baliabide lexikalek *berrerabilgarriak* izan beharko lukete, sistema anitzen iturria izan daitezen. Hala ere, garai horretan lexikoietan gordetako informazioa, eta informazioa kodetzeko erabilitako egiturak bateraezinak ziren maiz.

- Euskarri elektronikoan metatutako giza-erabilerari zuzenduta zeuden hiztegien kopurua handitzen ari zen garai hartan, eta gehikuntza hori gaurdaino nabari daiteke. Komunitate linguistikoa konturatu zen hiztegi horietan gordetako informazioa probetxagarria izan zitekeela lexikoiak garatzeko garaian, eta informazio lexikala hiztegietatik erauzten hasi zen.

- Lexikografoen, lexikologoen, linguisten, linguista konputazionalen, editoreen eta LNPrako tresnak ekoizten aritzen ziren enpresen arteko komunikazio-kanalak sendotzen joan ziren, eta kanal hauek beren arteko helburu komunak azalarazi zituzten.

Beharbada, *Grosseto* mintegian aztertutako faktore horien artean informazio lexikalaren *berrerabilgarritasunak* izan zuen eragin handiena mintegiaren ondoren eginiko ikerketetan. Izan ere, laurogeita hamarreko hamarkadan eginiko baliabide lexikalen inguruko ikerketa-proiektu gehienek berrerabilgarritasunaren kontzeptua izan baitzuten erreferente nagusietako bat, aurrerago ikusiko dugun bezala.

Hamar urte lehenago, laurogeiko hamarkadan zehar, lexikoaren inguruan hainbat ikerketa eta proiektu garatu baziren ere, proiektu horien barruan lexikoa ikergai zuten hainbat arloko ikertzaileek hamaika era desberdin asmatu eta erabiltzen zituzten. Nork berea —eta bere modura— egiten zuelarik, ordea, azkenean batek egindakoaz beste batek baliatu nahi zuenean aurretik egindako lan guztia ez zen nahi litzatekeen bezain lagungarri suertatzen eta, maiz, erabili ezina ere bai.

Boguraev eta Briscoe-k (1989) adibide baten bidez azaltzen dute aurrean aipatutako egoera, hiru sistema desberdinek ingelesezko *acknowledge* hitzerako duten adierazpena azaltzen digutenean (ikus II.1 irudia).

II.1 irudiko hiru sarrerek *acknowledge* hitzari buruzko antzeko informazioa gordetzen dute: kategoria sintaktikoa, hitzaren azpikategorizazioa, eta abar. Hala ere, informazio hori hain modu desberdinean dago adierazita, ia

```
[ACKNOWLEDGE
    Category:   V
    Base:       acknowledge
    Features:   (TRANSITIVE (REALNP) (PASSIVIZES))
                (CLAUSE (REALNP) (THATCOMP)
                (INDICATIVE: TENSE) (WH-))
                (NP-VP :AGR :AGRX (REALNP) :AGRX
                (PASSIVIZES) (INF) (WH-))]
```

```
[ACKNOWLEDGE
    FEATURES   (TRANS
                PASSIVE
                THATCOMP
                THATREQUIRED
                NPTOCOMP)
V S-D]
```

```
(acknowledge
    ((v +) (n -) (subcat npl)) acknowledge nil)
(acknowledge
    ((v +) (n -) (subcat sfin)) acknowledge nil)
    ;acknowledge that they were defeated
(acknowledge
    ((v +) (n -) (subcat se3)) acknowledge nil)
    ;acknowledge having been defeated
(acknowledge
    ((v +) (n -) (subcat or)) acknowledge nil)
    ;acknowledge him to do the best
```

II.1 Irudia: *acknowledge* hitzaren 3 adierazpen desberdin.

ezinezkoa bihurtzen dela hiru formalismo hauen arteko informazioa bateratzea.

Egoera tamalgarria da, zinez, aipatu berri duguna, batez ere lexikoi konputazional baten eraikuntzak eskatzen duen lan neketsua kontuan hartzen badugu. Esate baterako, Neff *et al.*-en lanean (1993) itzulpen automatikoko sistema baterako unitate lexikal baten eskuzko garapenak, batez beste, 30 minutu behar dituela aurreikusten da. LNPrako sistema baten lexikoiak milatik gora sarrera behar dituela kontuan harturik (ikusiko dugun bezala, ez da harritzekoa 50.000, 60.000 edo sarrera gehiagoz hornituriko datu-base

lexikala aurkitzea), erraz asma daiteke horrelako baliabide lexikalak eraikitzeak suposa dezakeen zama, bai denbora aldetik baina baita diru aldetik ere.

Calzolari-ren lanean (1994), egileak berrerabilgarritasunaren alde egiten du, nabarmen. Bere ustean, aplikazio bakoitzerako baliabide lexikal bereziak sortzea ekidin behar da, ahal den neurrian. Hori baino, komunitate linguistikoak ahalegindu beharko luke dagoeneko existitzen diren lexikoien informazioa berrerabiltzen eta estaldura zabala duten baliabide lexikalak eraikitzen.

Edonola ere, esan beharra dago berrerabilgarritasunaren terminoak bi adiera desberdin biltzen dituela bere baitan, bi adiera horiek elkar-erlazionatuta egon badaude ere. Batetik, baliabide lexikala berrerabiltzea bere informazio lexikalaz baliatzea da, lexikoi berri bat garatu nahi denean edota dagoen lexikoiren bat aberastu nahi denean. Bestetik, baliabide lexikala berrerabilgarria izango da baldin eta estaldura zabala duen, hots, domeinu zehatz bati lotuta ez badago, eta bere baitan biltzen duen informazioaren adierazpidea formalismo berezi bati lotuegia ez badago.

Gure tesi-lanean, iturri lexikal anitz eta heterogeneoak era bateratu batean atzitzeko diseinatu dugun sistema garatu dugularik, domeinu zabal eta anitzeko baliabidea eskainiko diogu, azken finean, erabiltzaileari. Lexikoi zabalak eraikitzerakoan eginiko ikerketak izan ditugu irizpide nagusi, beraz, gureak ere baliabide lexikal berrerabilgarria —berrerabilgarritasunaren bigarren adierari buruz ari gara— izan beharko baitu.

Jarraian, baina, terminoaren lehenengo adieran jarriko dugu arreta. Horretarako, informazio lexikalaren erauzketa automatikoaren inguruko proiektuak gainbegiratuko ditugu, eta, batez ere, lan horietan sortutako zenbait arazo aztertuko, gure proposamenak, neurri batean, arazo horiek arindu ditzakeelakoan sinetsiak baikaude.

Jadanik eskura dauden baliabide lexikalen informazioaz aprobetxatzea, baliabide berri bat eraikitzerakoan, luze ikertutako lerroa dugu, batez ere 90.eko hamarkadan garatutako hainbat proiektutan. Izan ere, oso interesgarria baitirudi aldez aurretik existitzen den informazio lexikalaz baliatzea —altxor lexikaltzat hartuz, azken finean— baliabide berriak garatu edota dauden baliabideak aberasteko. Altxor lexikal horien artean giza erabiltzaileari zuzendutako hiztegiak izan dira, behar bada, arreta handiena jaso dutenak.

Euskarri elektronikoan gordetako hiztegietatik ("Machine Readable Dictionary", MRD) abiatuta informazio lexikala erauztea era askotan egin da. Lexikoi konputazional bat huts-hutsetik eraiki nahi bada, hiztegi batek lexikoi horrek beharko dituen sarreren zerrenda eskainiko digu. Horretaz gain,

sarrera lexikal horien informazio morfologiko, kategoria sintaktiko, azpikategoriak etab. azaltzen dira hiztegi gehienetan, eta ez dirudi zaila informazio horretaz era automatiko batean baliatzea.

Bestalde, hiztegietan gorderiko informazio *inplizitua* ere eskura daiteke. Hiztegi-sarreretako definizioen edukiak analizatuz, hitzen (edo adieren) arteko taxonomiak eraiki daitezke. Amsler-en lan aitzindari eta famatuan (1981), MRDetako definizioetan gordetako informazio lexiko-semantikoa erabiltzen da, estreinakoz, *Merrian-Webster Pocket Dictionary* hiztegiko izen zein aditzen azpimultzo adierazgarri baten taxonomia erauzteko[2], era erdi-automatiko batean.

Lan horrek ereindako haziaren ondoren, eta ingeles hizkuntza ikasten ari denari zuzendutako hiztegi espezializatuen ekoizpenak gora egin zuen heinean, linguista konputazionalak hiztegi hauen egiturak LNPrako oso egokia zirudiela jabetu ziren. Izan ere, hiztegi hauek informazio zehatzagoa gordetzen dute maiz, eta beren barne-antolamendua askoz ere formalizatuagoa izaten da. Bestalde, esplizituagoak izaten dira sarrera lexikalen ezaugarri sintaktiko, morfologiko eta semantikoei buruz ari direnean. Horrela, *Oxford Advanced Learner's Dictionary* (OALD), *The Collins English Language Dictionary* (COBUILD), *Longman Dictionary of Contemporary English* (LDOCE), *Webster's Seventh Collegiate Dictionary* (W7) edo jadanik aipatutako *Merrian-Webster Pocket Dictionary* (MWPD) hiztegi garrantzitsuetan oinarritutako lanak etorri ziren [3].

Edonola ere, MRDetatik ezagutza eskuratzeak hainbat arazori aurre egin behar dio. Ide eta Véronis-en lanean (1994), hiztegi elektronikoen ustiapenean aritutako hainbat proiektu aztertzen dira. Urte horretarako, proiektu eta ikerketa txosten franko garatu dira; egileen arabera, ordea, linguistika konputazionalaren inguruko kongresuetan arlo honekin lotuta dauden artikuluen kopurua jaitsi egin da nabarmenki. Beren ustez, bi izan dira jaitsiera horren arrazoiak: alde batetik, ez da erauzpen-metodo egonkorrik garatu MRDen eskuratze orokorra egin ahal izateko. Hasierako metodoen emaitza itxaropentsuek optimismoa ekarri bazuten ere, ez zen argi ikusi metodo horien egokitasuna neurri errealeko hiztegiekin lan egiterakoan. Bestetik, komunitate zientifikoak corpusetan gorderiko informaziora lerrarazi du arreta.

Eskurapen-prozesuak erroreak izan ditzake, hiztegietako informazio lexi-

[2]Hitz-adieren definizioak analizatuz kontzeptuen hiperonimoak eskuratu zituen, beren *genus* delakoa aztertuz. Ondoren, hiztegian inplizituki adierazita dagoen kontzeptu-hierarkia eraiki zuen.

[3](Wilks *et al.*, 1998) lanaren 6. kapituluan ikus daiteke hurbilpen horien azalpen bat.

kala akastuna izaten baita maiz. Horrela, definizioetan oinarrituz erauzitako adiera-taxonomietan termino asko goregi kokatzen dira, eta inkonsistentzia hori dela-eta hiztegi desberdinetatik erauzitako hierarkien itxura desberdina eta bateragaitza izaten da. Ide eta Véronis-en lanean (1993), 5 hiztegitatik automatikoki erauzitako hierarkiak konparatzen dituzte. Egileen arabera, batez beste terminoen %21-34 goregi kokatzen da hierarkian.

Aurrean aipatutako (Ide eta Véronis, 1994) lanak arazo hauek aztertzen ditu, besteak beste. Haien aburuz, arazo hauei aurre egiteko informazio lexikala hainbat hiztegitatik erauzi behar da. Horrela, nahiz eta hiztegi batetik erauzitako informazioa inkonsistentea izan, ez da probablea beste hiztegi batean inkonsistentzia berarekin topatzea. Horrela diote egileek:

> We foresee that the creation for knowledge bases in the future will be accomplished by giving the human knowledge-base creator access to multiple resources, including MRDs and corpora, together with tools to extract differend kinds of information and combine it more or less by hand.

ELHISA bezalako sistema batek, informazioa hiztegi zein datu-base anitzetik eskura dezakeen neurrian, paregabeko tresna eskainiko dio baliabide lexikaletik informazioa erauzi behar duen orori. Izan ere, ELHISAren bidez hiztegi soil batek munduari buruz duen ezagutza murritza —ikusi dugun bezala, bere baitan inkonsistentziak ere izan ditzakeena— gainditzeko aukera emango du, hiztegi horretako informazioaren osagarritzat har daitekeen hainbat eta hainbat informazio eskainiko baitu, bere baitan izango dituen beste iturri lexikaletan oinarrituta.

II.1.1 Baliabide lexikalen estandarizaziorantz.

ELHISAk iturri lexikal anitz eta heterogeneoak hartuko ditu bere baitan, eta baliabide horietan gordetako informazioa erabiltzaileari eskaini, era bateratu batean. Informazioa ez da domeinu berezietara murriztuko, eta sistemak, horrela, estaldura zabala izango du, hainbat behar eta jakin-min desberdin betetzeko aukera emanez.

Ikusi dugu baliabide lexikalak estaldura zabalekoak izatearen beharra LN-Pan, eta, orain, ildo hori jorratu zuten laurogeita hamarreko hamarkadatik gaur arteko ikerlanei erreparatuko diegu, gure lanerako beharrezkoa baita lan horietatik ateratako konklusioak ulertu eta geureganatzea.

Arestian aipatu bezala, baliabide lexikal berezituen aurka agertu zen komunitate linguistikoa, LNPan lexikoiak duen garrantziaz jabetu bezain laster. Horrela, informazio lexikalaren berrerabilgarritasuna helburu, informazio lexikal aberatsaz hornitutako baliabide zabalak eratzea proposatzen zen.

Informazio lexikal sendoak, zabalak, aberatsak, eleanitzak eta erabileranitzak eraikitzeko ekimen ugari egin dira denboran zehar, hala nola, GENELEX (Normier eta Nossim, 1990), MULTILEX (MacNaugth, 1990), PAROLE/SIMPLE (Ruimy *et al.*, 1998; Bel *et al.*, 2000) edo EAGLES/ISLE (Uszkoreit *et al.*, 1996; Calzolari *et al.*, 2002) proiektuak. Informazioa metatzeko hiru baldintza orokorrez mintzo ohi dira proiektu hauetan murgildutakoak:

- Baliabide lexikaletan gordetako informazioak teoria linguistiko finkoetara ez luke lotuta egon behar.
- Informazioaren errepresentazioak, bestalde, estandarra behar luke izan, *hardware* edo aplikazio espezifikoekiko independentea.
- Azkenik, baliabide zabal hauek atzipen-metodo bateratu, eraginkor eta orokorrak eskaini behar dituzte.

Estaldura zabaleko baliabideek, beraz, era askotako aplikazioen hornitzaile lexikalak izan behar dute. Jakina, LNPrako aplikazioek, behar desberdinei erantzun behar dieten neurrian, maila desberdineko informazioa eskatuko diote lexikoiari: zuzentzaile ortografiko arrunt batek behar duen informazio lexikala ez da domeinu espezifikoko itzulpen-sistema konplexu batek behar duenaren berdina izango; lehenengoak tamaina handiko baina egitura aldetik laua den lexikoa behar duen bitartean, bigarrenak informazio egituratuagoa duen domeinu jakin bati buruzko hiztegi konplexua beharko du. Baliabide lexikalak, funtzionalitate anitzekoa izan behar badu, aipatutako bi baldintzak bete beharko lituzke, hots, hitzen zerrenda luzea gorde, eta hitz orori informazio lexikal aberatsa eta dimentsioanitzekoa esleitu.

Ikuspuntu funtzional batetik, hainbat urrats bete behar dira LNPrako sistemek estaldura zabaleko baliabideetatik informazioa ustia dezaten, proiektu hauetan jardun dutenen aburuz. Lehenik eta behin, aplikazioaren domeinua aztertu behar da, behar duen informazio-maila, hizkuntza, eta abar zehaztuz. Erabilera-esparrua definitu ondoren, baliabide zentralizatutik aplikazioaren eskakizun zehatzak aseko dituen azpilexikoa sortu behar da. Azpilexikoak independentea izan beharko luke iturri nagusitik, era honetan sistema jakinaren

beharretara egokia izan dadin optimiza baitaiteke. Batetik, azpilexikoak sistemak behar duen informazioa baino ez du gorde behar. Bestetik informazio lexikalaren adierazpena aldatzeko aukera dago, sistemaren notazioari egokituz. Horrela, azpilexikoak konpilatu beharko lirateke, sistemek informazio lexikala era eraginkor eta trinkoan atzi dezaten.

Baliabide zabal hauetan gordetako informazio lexikalaren mailak anitza behar du izan, hitzen informazio ortografiko, morfologiko, morfosintaktiko edo semantikoa gordetzeko aukera emanez. Baliabide lexikal zabalek abstrakzio-maila desberdinak izan beharko lituzkete, gordetzen duten informazio lexikalaren dimentsioaniztasuna bermatzeko, hots, informazio bera hainbat ikuspuntu desberdinetatik ikusteko aukera emateko. Baliabide sendoen garapena lan oso konplexua izanik, bertan gordetako osagai lexikalen egituraren malgutasuna bermatzea ezinbestekoa da. Horrela, bada, informazio minimo eta komuna gordetzeko egitura minimo bati hainbat geruza pilatu ohi zaizkio, abstrakzio-maila desberdinak osatuz.

Ez da nolanahiko lana, beraz, horrelako baliabide lexikalak garatzea. Arloa ikertu zuten proiektu gehienek, baliabide lexikal zabalak garatzearen zailtasuna ulertu zutenean, irizpide nagusi bat izan zuten beren jardueran: informazio lexikalaren estandarizazioa aztertzea.

Estandar batek, zer esanik ez, datu lexikalak konputagailuan kodetzerakoan izugarri lagunduko luke, eta, hortaz, baliabide lexikal aberatsak eraikitzeko aukera eman. Hari esker, LNPan ari direnek egun dagoen informazio lexikala berrerabiltzeko aukera izango lukete, baliabide lexikalak eratzeko emandako denbora eta zama ekonomikoak errentagarri bihurtuz.

Zajac-en lanean (1999) aipatzen den bezala, bi erronka nagusiri egin behar zaie aurre baliabide lexikalak estandarizatu nahi badira:

- Lehenengo arazoa formatu estandarrik ez izatetik dator. Formatu estandar batek, gainera, bi ezaugarri bete behar ditu: batetik, malgua izan behar du, hizkuntza guztietako informazioa kodetzeko aukera eman behar baitu, aplikazio desberdinek behar duten zehaztasun-mailarekin; bestetik, baina, formatuak zorrotza ere izan behar du, aplikazio horiek atzibide bateratua izan dezaten, informazio lexikala atzitzeko.

- Bigarren arazoa lehenengoarekin hertsiki lotua dago: oso zaila da — ezinezkoa ez bada— arkitektura lexikal oso bat garatzea, NLPrako aplikazioek eska ditzaketen ezaugarri linguistiko guztiak zerrendatzea, azpiegituren arteko erlazio posible guztiak jakitea, edota hiztegiak atzitu behar dituzten NLPrako aplikazioen beharrak aurreikustea.

EAGLES ekimena ("Expert Advisory Group on Language Engineering Standars") 90.eko hamarkadan burutu zen, eta, bere helburuen artean, informazio lexikalaren errepresentazio estandarra proposatzea zegoen (Heid, 1996). Haien esanetan, estandarra ez da, ezinbestean, zerbait egiteko modurik hoberena; hori baino, zer hori egiteko era hitzartua da, komunitateak adostutakoa. Estandarrak ezartzeko bi bide orokorrez mintzo dira: *De facto* estandar delakoak, industriak edo aplikazio jakin baten erabilpen zabalak ezarriak, eta estandar formal eta ofizialak, erakunde internazionalen onespenak dituztenak. Nolanahi ere, garai horretan informazio lexikala adierazteko estandarraren eza azpimarratzen dute, eta ez dute uste, gainera, estandar bat era "naturalean" sortuko denik. Haien aburuz, ezin da espero ingeniaritza linguistikoa bezalako arlo berri batean estandar bat sortuko denik, baldin eta estandarra taxuz definitzeko baliabideak —ekonomikoak zein giza-baliabideak— jartzen ez badira.

Horrela, adostasun-estandarra definitzen saiatuko da EAGLES proiektua. Adostasun-estandarrak, jakina, arloan aritutakoen bat etortze batetik etorri beharko du, arrakastatsua izatea nahi bada behinik behin. Horrela, estandarrak arloko adituen masa kritikoaren oniritzia behar duela azpimarratzen dute. Halaber, arloko ikuspegi orokor batean oinarriturik behar du izan, egindako lanak bere baitan integratu ahal izateko, eta garaiko formalismo garrantzitsuenak ez baztertzeko. Bere egokitasuna ere erakutsi behar du, aplikazio praktikoetan zein teknologia berrien garapenean. Azkenik, estandarrak erraz izan behar du eskuratzen eta aztertzen, erabiltzaileek bere onurak baieztatu eta estandarri etekina atera ahal diezaioten.

Estandar baten betebeharren artean —akaso estandar baten ekarpen nagusia— estandarizatu nahi den erreferentzia-arkitektura osatzea da. Arkitektura lexikal batek oinarrizko objektuak definituko ditu, eta domeinu osoaren oinarrizko terminologia ezarri. Bestalde, objektuen funtzionalitatea definituko du (software moduluen API-en[4] antzera), eta ataza zehatz bat burutu ahal izateko bete behar diren urratsak zehaztu. Azkenik, arkitektura komun batek sistema lexikalak ebaluatzeko bidea irekitzen du.

EAGLESen aburuz, honako hauek lirateke, informazio lexikalaren estandarizazioari begira, arkitektura lexikalak finkatu beharko lituzkeen terminoak (Uszkoreit *et al.*, 1996):

- Sistemaren oinarrizko objektuak zeintzuk diren ezarri behar du, eta baita ere, objektuen artean egon daitezkeen erlazio motak. Horrela,

[4] Application Programming Interface

estandar osorako balioko duen oinarrizko terminologia zehaztuko da, zeinaren bitartez osagai lexikalei eta beren arteko erlazioei buruz mintzatu ahal izango den.

- Estandarrak hainbat hizkuntzatarako baliagarria izan behar duenez, eta kodetutako informazioa maila linguistiko anitzekoa izango bada, arkitekturak hizkuntzen arteko erlazioak eta maila linguistikoen arteko definitu behar ditu.

- Arkitekturak informazio lexikala adierazteko oinarrizko egitura sendo bat definitu behar du. Oinarrizko egiturak osagai lexikal generikoak izan dezakeen informazio minimoaren gordelekua izan beharko luke; bestalde, hedagarria eta modularra izan behar du, ezinbestean, estandarrak aurreikusitako informazio motetatik at dagoena ere kodetu ahal izateko.

- Arkitektura malgua izan dadin, bere baitan gordeko den informazioa sailkatu egin behar du. Datu-baseen arloan meta-eskema, eskema eta eskemaren instantzia bereizten diren legez, horrelaxe egin beharko du arkitektura lexikal batek:

  - **Meta-eskema**: eskemek bete behar dituzten baldintza eta murriztapen minimoak ezartzen ditu. Meta-eskema, azken finean, eskura dauden informazio lexikalaren motei buruzko informazioa da.
  - **Eskema**: hizkuntza jakin bateko deskribapen linguistikoen formatu logikoa ezarriko du.
  - **Instantziak**: datu lexikalak; datu hauek eskemarekin bat etorri behar dute.

Ildo bera jarraitu zuten estandarizazio lexikala helburu zuten proiektu garrantzitsuenek. Ondoren, ekimen horietatik gure lanerako interes handiena izan dutenak azalduko ditugu. Izan ere, luzeegi joko bailiguke estandarizazioaren inguruan egin diren proiektu guztien zerrenda aurkezteak.

GENELEX

GENELEX (Normier eta Nossim, 1990) proiektua estreinetarikoa dugu, eta bere helburua hainbat hizkuntza europarrentzat hiztegi elektronikoen egitura lexikal generikoa eraikitzea zen, oso egitura deskribatzaile —eta konplexu—

baten bidez. Hiztegi generikoa garatzeaz gain, hiztegiak berak generikoa izaten jarraituko duela bermatzeko estrategiak eta metodologiak ere landu ziren.

Unitateen deskribapena zehaztasun handiz errepresenta daiteke GENELEXeko ereduan. Informazioaren granularitatea aldakorra da, gainera, eta ereduak aukera ematen du, esaterako, oinarrizko informazio lexikala estreinako urrats batean kodetzeko, azaleko errepresentazio baten bidez, eta, hurrengo batean, informazio hori aberastea edo fintzea.

Eredua deskribatzailea da, eta osagai deskribatzaileen arteko elkarrekintza bermatzen du. Hainbat osagai konplexu osagai sinpleen bidez definitzen dira, eta azken hauek, berriz, sinpleagoak diren beste batzuen bidez. Informazioa, horrela, hainbat geruzatan antolatzen da, konplexutasun-maila anitz errepresentatzeko aukera emanaz.

GENELEXek hiru maila definitzen ditu: morfologikoa, sintaktikoa eta semantikoa. Hirurak daude elkarrekin erlazionatuak, baina, aldi berean, maila bakoitza independentea da. Unitate lexikala hitz-adiera da, unitate morfologiko (*UM*, "Morfological Unit"), sintaktiko (*SyntU*, "Syntactic Unit") eta semantikoen (*SemU*, "Semantic Unit") arteko erlazioek definitua. Hiztegia entitate-erlazio tankerako formatu baten bidez definitzen da, unitate lexikala grafo bezala irudikatzen delarik.

MULTILEX

MULTILEX (MacNaugth, 1990) proiektuan egitura lexikal estandarra proposatzen zen —GENELEXek proposaturiko egitura lexikal estandarrean oinarrituz—, eta 15 lexikoi eleanitz[5] egitura horren arabera kodetu ziren. Itzulpen automatikorako pentsatua, MULTILEXen arkitektura hizkuntzarekiko independentea izateko asmoarekin eraiki zen. Proposatutako eredua funtzioanitzekoa da, alegia, teoria desberdinetarako erabilgarria eta aplikazio desberdinek behar duten informazio lexikala integratzeko aukera ematen duena. MULTILEXen arkitektura linguistikoak hitzen informazio morfosintaktikoaren eta semantikoaren artean bereizketa garbia egiten du.

Esan bezala, proiektuak izaera eleanitza du, itzulpen automatikorako baliagarria izan behar duen heinean. Haatik, eta aurrean aipatutako funtzioaniztasunaren ondorioz, MULTILEXen integratutako hizkuntza bakoitza lexikoi elebakarra bezala ikusi ahal da. Hau horrela izanik, bere arkitektura bikoitza da: alde batetik arkitektura elebakarra eta bestetik, arkitektura

[5]MULTILEX proiektuan ingelesa, frantsesa, gaztelera, alemanera, daniera, nederlandera, italiera eta grekoa landu ziren.

eleanitza.[6] Horrela, bada, hiztegi elebakar bakoitzaren beharrak espezifikatuko dira eta, bestalde, hiztegi elebakar hauen arteko erlazioak zehaztuko dira.

Informazio lexikala bi nodotan sailkatzen da: 1) *GPMU* ("Graphic-Phonologic-Morphological Unit") delako nodoetan sarreraren informazio ortografiko, fonologikoa zein portaera morfologikoa adieraziko da. 2) *LU* ("Lexical Unit") delako nodoetan, berriz, sarrera horrek hizkuntza jakin batean dituen adierak jasoko dira.

PAROLE/SIMPLE

PAROLE (Ruimy *et al.*, 1998) proiektuak hiztegi elektronikoen auzi morfologikoak deskribatzeko eredu estandar bat eraikitzeari ekin zion[7]. Horretarako, objektu lexikalen maila morfologikoaren kodekera-estandar bat proposatu zuen. PAROLE ereduak forma aldeko zein hitzen propietate gramatikalen aldaketak adierazteko aukera ematen du. Hitz bakoitzaren erroa —lema, alegia— *UM* deituriko unitate morfologiko batean kodetuko da, eta, unitate horri lotuta, hitzaren flexio posibleak adierazten dituzten ezaugarri morfologikoak. Flexioen informazioa generikoa da, ahal den neurrian. Horretaz gain, PAROLEk banaketa garbia egiten du hitzen flexio konbinatorio guztien eta hitzaren ebakeraren edota ortografiaren artean, eta, horrela, flexioek hitz-erroaren ortografian —edo ebakeran— izan ditzaketen eraginak beste unitate-mota batzuetan kodetzen ditu. EAGLES proiektuan bezala, PAROLEk ere saiakera sendoa egin zuen datu lexikalen notazioa normalizatzeko, eta kategoria zein azpikategoria lexikalen balio posibleen zerrenda estandarra proposatu zuen[8]. Ikusiko dugu (IV kapituluko IV.7 atalean[9]), gure ELHISA sistemak ere informazio lexikalaren normalizazioa beharko duela, eta, horretarako, PAROLEn oinarritu garela zenbait eremu normalizatu ahal izateko. Emaitza bezala tamaina ertaineko 12 lexikoi garatu zituzten.

PAROLE proiektuaren ildo berari jarraituz, SIMPLE proiektuak (Bel *et al.*, 2000) alde lexiko-semantikoa lantzen du, bere helburu nagusia hainbat hizkuntzatarako estaldura zabaleko baliabide lexikal semantikoak gara-

[6] MULTILEXen hurbilpena itzulpen automatikoan *transfer* delako ereduan kokatzen da, EUROTRAren parametroak jarraituz.

[7] Neurri handi batean, PAROLEren lana, geruza morfologiko honetan, GENELEX proiektuan dago oinarrituta.

[8] Balio hauek, hein handi batean, EAGLESek proposatukoetan oinarritzen dira.

[9] 212. orrian.

tzea delarik. PAROLEren arreta auzi morfologiko eta sintaktikoetan jartzen den bitartean, SIMPLEk hitzen arteko erlazio lexiko-semantikoetara lerratzen du, eta, batik bat, PAROLEk estalitako 12 hizkuntzetarako baliagarriak diren ezaugarri semantikoen proposamen bateratua eskaini nahi du. Proiektuan 10.000 adierako baliabide eleanitza eraiki zutelarik, baliabide honen eraikuntzan oso kontuan hartu dira informazioaren osotasuna eta bateratasuna, hizkuntzatik independenteak diren generalizazioak atzemanez. Azterketa horretatik abiatuta, mota semantikoen oinarrizko multzoa definitu zuten —SIMPLEren ontologia—, eta, baita ere, unitate semantikoetan kodetu beharko litzatekeen nozioen oinarrizko multzoa: domeinua, definizioa, argumentu-egitura, "qualia structure"[10], hautapen-murriztapenak eta abar.

EAGLES/ISLE

EAGLES proiektuak, esan dugun bezala, egitura lexikal estandarren proposamen zehatzak egin zituen, eta izan zuen berak eginiko lanaren hedapenik, ISLE ekimenaren barruan (Calzolari *et al.*, 2002). Ekimenaren helburuen artean, informazio lexikal eleanitza kodetzeko eskema orokor bat eskaintzea dago, MILE ("Multilingual ISLE Lexical Entry") izenekoa. MILE oso modularra eta geruza anitzekoa da. Horrela, modulu independenteek —baina, aldi berean, erlazionatuta daudenak— sarrera lexikaleko dimentsio desberdinak[11] eskura ditzakete. Bestalde, geruzetan oinarritutako arkitekturak hainbat granularitate mailatako informazioa eskuratzeko aukera ematen du, azaleko informazioa zein informazio sakona eskuratzeko bidea emanaz. Goi-mailako bi modulu nagusi ditu ISLEk, mono-MILE eta multi-MILE izenekoak, errepresentazio elebakarrak eta eleanitzak ematen dituztenak, hurrenez hurren. Mono-MILEk, bere aldetik, hainbat modulu ditu, sarrera lexikalen deskribapen morfologikoa, sintaktikoa eta semantikoak adierazteko. Beste estandarizazio-ekimen askok ez bezala, MILEk hitz anitzeko unitate lexikalak eta kolokazioak ere hartzen ditu bere baitan. Multi-MILEk, bestalde, sarrera lexikal eleanitzen arteko elkarrekikotasunak adierazteko eredu formala ematen du. Hizkuntza desberdinetako sarrera lexikalen arteko erlazioak mono-MILEren gainean eratzen dira, eta erlazioak adierazteko sarreren deskribapen elebakarra ustiatzen da, eta, zenbaitetan, baita aberastu ere. MILE arkitek-

[10] Pustejovky-ren "Generative Lexicon" delako teorian, autoreak lau adierazpide-maila bereizten ditu ezagutza lexikala errepresentatzeko, horien artean "qualia structure" delakoak item lexikala ezaugarritzen duten atributu nagusiak finkatzen dituelarik.

[11] II.2 atalean informazioaren dimentsio-aniztasuna aztertzen da.

turak, horrela, sarreren deskribapen elebakarraren independentzia bermatzen du, eta, halaber, erlazio eleanitzak osatzeko eredu aberats bat eskaini. ISLE proiektuak informazio lexikalaren meta-datuei —hots, eskura dagoen informazio lexikalaren motei buruzko informazioa— buruzko proposamen bat ere garatu du, (Gibbon *et al.*, 2001) lanean. Bertan bi talde nagusitan sailkatzen dira meta-datu lexikalak: kanpokoak (baliabide lexikalaren informazio orokorra, hots, bere izena, sorkuntza-data, hizkuntzak eta abar), eta barrukoak (sarrera lexikalaren informazioa).

TEI

Azkenik, aipatzeke utzi ezin dezakegu dokumentuak era estandarrean kodetzeko helburua duen TEI ("Text Encoding Initiative") izeneko proiektua (Sperberg-McQueen eta Burnard, 1995). Proiektuan edozein dokumentu kodetzeko irizpideak proposatzen dira; horien artean, hiztegiak edo, are, informazio linguistikoa kodetzeko hain famatuak diren ezaugarri-egiturak. TEIk garrantzi berezia izan du geure proiektuan, integratutako hainbat baliabide —hiztegiak, batez ere— formatu horretan kodetuta baitaude. Izan ere, TEIk hiztegiei eskainitako kapituluak hiztegietako informazioa luze eta zabal jorratzen du, eta, horrela, baliabide lexikal horietan aurki daitekeenaren berri aberatsa ematen digu, ikusi berri ditugun estandarizazio-ekimen gehientsuen antzera. II.2.1.1 atalean azalduko ditugu TEIren gorabeherak.

II.1.2 Informazio lexikalaren integrazioa.

Aurreko atalean baliabide lexikal estandarrak osatzea helburu zuten hainbat proiektu eta ekimen ikusi ditugu. Ekimen horietan estaldura handiko baliabide lexikalak definitu nahi ziren, domeinu zehatzetara ez murriztuak, eta edozein aplikazio linguistiko datu lexikalez hornitzeko aukera ematen zutenak. Horrelako baliabideak diseinatzearren sortutako proiektuek, baina, arazo franko aurkituko dute bidean.

Estandarizazio-ekimenek informazio lexikalaren berrerabilgarritasuna izan dute erreferente nagusia, eta, hortaz, baliabideak ahal den neurrian teoria linguistikoekiko independenteak izan beharko liratekeela azpimarratzen da askotan. Izan ere, informazio lexikala teoria linguistiko jakin bati hertsiki loturik badago, zaila edo ezinezkoa izango baita lexikoi bera hainbat aplikazio desberdinen hornitzaile izatea. Horrela, baliabide hauek "neutralak" izan behar dute, ahal den neurrian.

Teoria linguistikoetatik erabat independente diren estandar lexikalak eratzeak, baina, hainbat oztopori egin behar dio aurre. Izan ere, historikoki LNPrako sistemek teoria desberdinetatik edan ohi baitute, eta, sistemak berak desberdinak diren bezala, beren errepresentazio lexikalak ere desberdinak dira; hain desberdinak izaten dira, zaila egiten dela auzi lexikalak sistema hauek euskarri dituzten teoria linguistikoetatik banantzea. Halaber, kontuan hartu behar da teoria linguistikoak ez direla, maiz, maila berekoak izaten, motibazio zein ikuspuntu desberdinetatik —linguistikoa, psikologikoa, filosofikoa— sortuak baitira; askotan, teoria linguistikoek interes kontrajarriak ere eduki ditzakete.

Luze ikertu da, hala ere, lexikoi neutralen ideia, hiztegi zein lexikoi desberdinek gertaera antzekoak adierazten dituzten intuizio sendoan oinarrituz. Hala ere, baliabide hauen baliagarritasuna kolokan dago. Esate baterako, teoria neutraleko lexikoi baten eraikuntza ezinezkoa dela dioen Ramsay-ren lanean (1995), egileak LNPrako teoria linguistiko guztietarako baliagarriak liratekeen baliabide lexikaletan ipintzen du arreta. Horrela, bada, "neutral" terminoaren adiera teoria guztiek komun duten "izendatzaile komunetan txikiena" izango da zeren, Ramsay-en arabera, soilik minimo hori izango baita egokia teoria linguistiko ororentzat. Bere konklusioa garbia da: horrelako lexikoi batek ez du erabilera praktikorik edukiko.

(Cahill *et al.*, 1999) laneko egileen aburuz, estandar izateko asmoarekin eratzen den arkitektura lexikal bat, teoria linguistiko jakin bati hertsiki loturik badago, nekez onartuko da komunitate zientifikoan. Hala ere, arkitektura teoriarekiko erabat neutrala izatea kolokan jartzen dute. Horrela, aitortzen dute teoria linguistikoen arteko adostasun-maila minimoa behar-beharrezkoa dela. Izan ere, teoria linguistikoen arteko adostasun-mailarik gabe ezin baitira LNPrako sistemak ebaluatu, edo sistemak harremanetan jarri.

Beraz, teoria linguistiko neutralaren ildoa ikertu dutenak bi ondorio kontraesangarritara iritsi ohi dira. Batetik, nahikoa garbi dirudi sistema lexikal estandarrak ezin duela teoria linguistiko jakin bati estuki lotuta egon. Bestalde, aitzitik, teoria linguistiko neutrala zer den ez dago nahi bezain argi. Hortaz, erabat politeorikoa den baliabideak eratzeak lan zaila edo ezinezkoa dela dirudi, eta, gainera, ez dago oso garbi horrelako sistema batek erabilera praktikoa izango duenetz (Kanngießer, 1996).

Bestalde, estreinako estandarizazio-ekimenen emaitza praktikoak ez ziren, antza, komunitate zientifikoan hedatu. Horrela dio, behintzat, Zajac egileak (1999):

> Some projects have concentrated on developing lexical resources directly in a format suitable for further use in NLP (e.g. Genelex, Multilex). [...] The lexical knowledge encoded in these systems can truly be called reusable since neither the format nor the content is application-dependent. The results of these projects is however not available to the research community.

Informazio lexikalaren estandarizazioa helburu zuten proiektuak puripurian zeudenean, ikertzaile askok ez zuten proiektu hauekin bat egin —beharbada proiektuen emaitzarik ikusi ez zutelako—, eta, nolabait, bizkarra ere eman zieten. Baliabide lexikal estandarrek komunitatean hedatzeko izan zuten arazoen seinale, LNPrako ikertzaileak estandarra izateko inolako asmorik ez zuen baliabide jakin bat erabiltzen hasi ziren beren jardueran: WordNet ezagutza-base lexikala[12].

WordNet baliabidea ez dator estandarizazio-ekimenen helburuekin bat. Batetik, ez du adierazpide estandarrik erabiltzen datuak gordetzeko. Horren ordez, datuen antolamendu berezi bat jarraitzen du, eta, WordNet-en gainean kontsultak egiterakoan, tresna bereziak ere behar dira. Bestetik, WordNet oso lotuta dago zenbait teoria psikolinguistikorekin, eta, ikusi dugu, estandarizazio-ekimenen eskakizunen artean teoriarekiko neutraltasuna zen bat.

Hala eta guztiz ere, WordNet oso zabala da, eta bere baitan hainbat eta hainbat kontzeptu gordetzen dira[13]. Gainera, baliabidea eskuz garatua izan zenez, datuen *kalitatea* oso handia da.

(Cunningham *et al.*, 2000) lanean horixe bera aipatzen da:

> The issues of standards is a vexed one: experience with repositories of lexical materials [...] suggested that if resources had to have standarised formats, they would not be deposited or used. The success of WordNet worldwide is a demonstration of how researcher choice can defy any committee's standards.

Horrela, bada, informazio lexikala berrerabiltzeko beste ikuspuntu desberdin bat jorratu zen, hots, informazio lexikalaren *integrazioa*. Hitzak eta beren arteko erlazio aberatsak adierazteko formalismo estandarrik garatzeak eziñezkoa zirudienez —eta horrelako estandar neutral baten erabilera ere kolokan izanik—, proiektu hauen helburua informazio lexikala eredu komun batean

[12] II.3.3.2 atalean azalduko dira WordNet-en gorabeherak.

[13] WordNet-eko asken bertsioak 115000 kontzeptu inguru ditu.

biltzea da. Eredu komuna ez da estandar izateko asmoarekin eraikitzen; hori baino, integratzen diren baliabide lexikaletan gorderiko informazioaren bildura bezala ikusten da. Helburu horretarako, informazio lexikalaren *abstrakzioa* hartzen da erreferente nagusitzat.

Integrazioa helburu duen proiektu baten lekuko bat gorago aipaturiko (Cunningham *et al.*, 2000) lanean aurki daiteke. Bertan, informazio lexikalaren gainean abstrakzio-geruzak pilatzea proposatzen da; hainbat baliabide lexikal oinarritzat hartuz[14], beren baitan gordetako informazio guztia kapsulatzen duen objektuei zuzendutako eredu komuna eratzea proposatzen dute. Horretarako, baliabide lexikal bakoitza modelatzen dute —bertan gordetako informazioaren kontzeptualizazioa, alegia—, objektuei zuzendutako eredu batez, hots, UMLz. Modelizazio honetan baliabide bakoitzak duen terminologia eta konbentzioak errespetatzen dira.

Baliabide bakoitzaren modelizazioaren gainean *eredu bateratua* izeneko eredu berri bat eraikitzen dute, baliabideen gaineko generalizazioak adierazten dituenak. Eredu bateratuak baliabideak integratzeko bidea ematen du.

Eredu horiek abiapuntu, objektuen taxonomia bat osatzen da, zeinaren goi-mailako objektuek abstrakzio orokorrak adieraziko dituzten, eta hostoetan kokatutako objektuek, berriz, baliabide jakinen elementuak. Taxonomia independentea da baliabide lexikalekiko, modu paraleloan eratua baita. Berari esker, informazio lexikala era kontzeptualean adierazten ahal da, erabiltzaileak hitz bati eta bere ezaugarriei buruz galde baitezake, datuak gordetzeko egitura zuzenean atzitu behar izan gabe. Hala ere, erabiltzaileek datu lexikalak beren jatorrizko formatuan atzitu ahal izatea ezinbestekotzat jotzen da. Horrela, aipaturiko taxonomiaren elementuak estekak dira, *funtzio* bera duten jatorrizko elementu lexikaletara seinalatzen dutenak. Taxonomia-maila bakoitzean atzipen programatikoa egiteko aukera ematen zaio erabiltzaileari, azpian dituen datuen gaineko APIaren bitartez.

Hala ere, egile hauek integrazio-prozesuan sortutako zailtasunak aztertzen dituzte, eta azterketa hori oso baliagarria izango zaigulakoan gaude, gure ELHISAk ere informazio lexikalaren integrazioa gauzatzea baitu helburu. Bi arazo nagusi aipatzen dituzte baliabideak integratzerakoan:

- Baliabideek ez dute informazio lexikala maila berean gordetzen, ezta

[14]Lanean honako baliabide hauekin egiten dute lan: WordNet, Comlex, Celex, EuroWordNet, CRL-LDB eta Mikrokosmos. Bertan gordetako informazioa, egileen esanetan, oso zabala eta aberatsa da: informazio ortografikoa, morfosintaktikoa (Comlex), sintaktikoa eta fonologikoa (Celex), eta semantikoa (WordNet, EuroWorNet, Mikrokosmos).

granularitate berarekin ere, eta, gainera, ez dituzte konbentzio berdinak erabiltzen ezaugarri berdinak gordetzeko. Horrela, zenbait fenomeno lexikalentzat baliabide batzuek informazio oso aberatsa gordetzen duten bitartean, beste batzuek oso informazio orokorra eskaintzen dute.

- Integrazio-prozesuan bi helburu kontrajarri bilatzen dira: batetik, baliabide bakoitzaren berezitasunak gorde nahi dira, baina, bestetik, baliabideak komun duten informazioa elkarrekin erkatu eta bil daitekeela bermatu behar da. Zenbait kasutan, konpromisoa erraza da; konparazio batera, WordNet-eko `word` atributua eta Celex-eko `lemma` atributuak hitz-formak adierazteko erabiltzen dira. Horrela, bi atributuak `lemma` klase baten instantziaren pean sailka daitezke. Arazoak sortuko dira, baina, baliabideek egitura propio eta bereziak erabiltzen dituztenean zenbait fenomeno linguistiko adierazteko. Adibide gisa azpikategoria-ereduak azpimarratzen dituzte: baliabide lexikalek era oso berezian kodetu ohi dituzte hitzen azpikategoriak. Beste batzuek maila linguistiko desberdinak konbinatzen dituzte zenbait eremutan (adibidez, morfosintaxia eta azpikategorizazioa). Informazio hori guztia elkarrekin alderatu nahi bada, errepresentazio-eredu komun bat aukeratu behar dela azpimarratzen dute, ezinbestean (adibidez, EAGLES proiektuak proposatutakoa). Horrela, baliabideek izan ditzaketen egitura bereziak eredu komunera bihurtu behar dira integrazio-prozesuan, eta, horretarako, baliabideetan inplizituki gordeta dagoen hainbat informazio azalarazi behar da. Hala ere, errepresentazio-eredu komunak ez ditu arazo guztiak konponduko, hizkuntzen berezitasunak, kasu, ez baititu maiz kontuan hartzen.

Oso arkitektura malgua proposatzen da, aztertutako (Cunningham *et al.*, 2000) lanean. Oso interesgarria iruditzen zaigu baliabide lexikalak ez direla aldatu behar sisteman integratzeko, eta, baita ere, erabiltzaileak une oro erabaki dezakeela datuak bere jatorrizko formatuan eskuratzea. Bestalde, bertan proposatutako eredu bateratua behetik gorako estrategia bati jarraituz osatzen da, baliabideen kontzeptualizazioak oinarritzat hartuz, bertoko objektuen generalizazioak eginez osatzen baita.

Integrazioaren inguruan sortutako beste lan garrantzitsu bat (Zajac, 1999) lanean aurki dezakegu, *Habanera* delako ezagutza-base lexikalaren eraikuntzaren gorabeherak azaltzen direnean. *Habanera* ezagutza-basea CLR[15] era-

[15]CLR ("Computing Research Laboratory") Mexiko Berria estatuko uniber-

kundeak kudeatutako hiztegi konputazionalen informazioarekin sortua da —hiztegi horien informazio lexikalaren integrazioari esker—, hiztegi konputazional berriak sortzeko ingurunea izatearen helburuarekin.

Habanera-k bat egiten du EAGLES proiektuak arkitektura lexikaletarako egin zuen gomendioekin[16], eskema, meta-eskema eta datuen arteko bereizketa garbiak egiten baititu. Bere arkitektura geruza anitzekoa da, hots, sarrera lexikalen definizioak hainbat geruza ditu, zeintzuek sarreraren gainean murriztapen berriak ezartzen dituzten. Geruzek informazio lexikalaren gainean hainbat abstrakzio egiteko aukera ematen dute. Horrela, hiztegi jakin batek geruzen azpimultzo bat bakarrik erabil dezake, eta baita geruza baten murriztapenak hedatu. Izan ere, geruzen bidez datuen gaineko eskema desberdinak defini baitaitezke. Inork ezin du, hala ere, meta-eskemaren gainean aldaketarik egin.

Habanera-ren arkitekturako meta-eskema definitzerakoan, egileek TEIko gidalerroak[17] izan zituzten erreferente nagusi, bertatik aterata baitaude, hein handi batean, sarrera lexikal generikoek izan ditzaketen eremuen zerrenda. Hala ere, meta-eskema inplementatzeko *Ezaugarri-Egitura Motatuak* (EEM) (Zajac, 1992) erabili dira, formalismo honek herentziarako eta baterakuntzarako dituen mekanismo aberatsak direla eta.

CRLn garatutako EEM lengoaiari esker, sarrera lexikalak deskribatzen dituzten ezaugarrien motak moduluetan taldeka daitezke. Horrela, modulu multzo batek hiztegi jakin baten eskema modelatuko du, hiztegi horren instantziek —hots, sarrera lexikalek— bete behar dituzten egituren ezaugarriak finkatuz. Moduluak era inkrementalean sailkatzen dira. Modulu generiko bat dago, sarrera lexikal generikoen ezaugarriak zehazten dituena. Hizkuntza berezien moduluak, berriz, modulu generikoaren gainean eraikiak daude, modulu generikoa aberastuz, hizkuntzari lotuta dagoen informazio berezia erantsiz (adibidez, hizkuntza bateko ezaugarri morfosintaktikoen zerrenda) edota modulu generikoak ezarritako baldintzak murriztuz.

Esan bezala, hiztegi-sarrerak, eta beren arteko erlazioak (sinonimoak, antonimoak, hizkuntza desberdinetako hiztegien arteko erlazioak, etab.), ezaugarri-egitura motatuen bidez adierazten dira, alegia, ezaugarriek mota batekoak izan behar dute. Moten definizioak, oro har, onargarriak diren sarrera

tsitateko ikerkuntza-saila da, testu elebidunen prozesaketan espezializatua. Ikus `http://crl.nmsu.edu/` orria.

[16]Ikus II.1.1 atala.

[17]Ikus, aurrerago, II.2.1.1 atala.

lexikalen modelizazio bezala ikus daitezke[18].

Motek, bestalde, hierarkia bat osatzen dute, eta, honekin, bi funtzionalitate nagusi lortzen dira. Batetik, ezaugarrien balioak hereda daitezke, informazio lexikalaren izaera hierarkikoa islatuz. Bestetik, *Habanera*-n parte hartzen duten hiztegi guztien modelizazioa lor daiteke, hizkuntzen arteko mota-hierarkia bat eratuz: mota-hierarkia eleanitz honek hizkuntzarekiko dependenteak diren elementuak zehazten ditu (adibidez, kategoria morfosintaktikoen balio posibleak), hizkuntza batek baino gehiagok komun dituzten goi-motak definituz. Horrela, mota lexikalen herentzia-hierarkia eleanitza sortzeko aukera ematen da. Izan ere, egilearen esanetan, ingurune eleanitzetan oso litekeena baita mota eleanitzak erabili behar izatea (adibidez, kategoria lexikalen zerrenda estandarra), nahiz eta estandarizaziorako proiektuek gai honi garrantzi txikia eman diotela nabarmendu.

Habanera-ren eraikuntzan, hortaz, sarrera lexikalen integrazio-teknikak erabili dira. Jada existitzen diren CRLko baliabide lexikalak abiapuntu, beren informazio guztia era egituratu batean antolatzeko ingurune bat eskaintzen dute. Ingurune honetan, gainera, informazio lexikala hainbat ikuspuntutatik kudea daiteke, sarrera lexikalak modelatzeko erabilitako geruza-sistema dela eta. Horrela, eta informazioaren gainean abstrakzio-maila desberdinen bidez, informazio lexikalaren integrazioa gauzatzen da.

Informazio lexikalaren integrazioa helburu garbia ez badute ere, saio ugari egin da existitzen diren baliabideak elkartzen, linguistikako hainbat arlotan, dela lengoaia naturalaren sorkuntza (Jing eta McKeown, 1998), ahotsaren tratamendua (Ribeiro *et al.*, 2003), edo hizkuntza minoritarioetan arreta jarria duen landa-linguistika (Wittenburg *et al.*, 2002). Proiektu hauen helburua ez da informazio lexikalaren integrazioko arazoaren azterketa sakona egitea, ezta prozesuak dituen arazoak aztertzea ere. Hori baino, existitzen diren baliabideen informazioa biltzeko saioak dira, ikuspuntu praktiko batetik, baliabide berriak lortzearren baliabideen informazioaz profitatzeko.

II.2 Informazio lexikalaren errepresentazioa.

Informazio lexikala errepresentatzeko modua garrantzitsua da. Baliabide lexikalek estaldura zabalekoak izan behar badute, informazio kopuru handia eta heterogeneoa errepresentatzeko aukera eman behar dute. Zaila da, ordea,

[18] XML edo SGML lengoaien DTDen antzera.

adierazpen-eredu bakarra aurkitzea datu lexikalak errepresentatzeko. Izan ere, datu lexikalak datu-baseen arloan erabili ohi den datu-motak baino aski konplexuagoak baititugu, edozein hiztegi-sarreraren informazioan arreta jartzen badugu nabari daitekeen legez. Horrela, bada, datu-eredu klasikoak —erlazionala, kasu— ez dira ondo egokitzen datu-mota hauetara.

Gauzak horrela, errepresentazio-eredu desberdinak erabili izan dira gorde nahi den informazioaren izaeraren arabera. Esate baterako, atributu-balio motako datuak gordetzeko (adib. kategoriak etab.), datu-base arruntak egokiak izango dira. Baterakuntza-gramatiketan behar diren datu lexikal konplexuagoek, aldiz, eredu ahaltsuagoen mende egon beharko dute. Gure lanaren muina izaera heterogeneoa duten baliabide lexikalen integrazioa denez, baliabide hauen adierazpen-ereduak aztertu egin behar ditugu, eta, atal honetan, azterketa horri ekingo diogu.

Informazioa lexikala gordetzeko erabili izan diren eta erabiltzen diren errepresentazio desberdinak aztertu baino lehen, adierazpen-ereduak eskaini beharko lituzkeen ezaugarriak ikusiko ditugu eta, horretarako, (Simmons, 1998) lanean oinarrituko gara. Egilearen aburuz, LNPrako tresnen garapen sendo eta azkarra egin nahi bada, datuen izaera modela dezaketen metodoak erabili behar dira ezinbestean. Horrela, bada, LNPrako datu-base ingurune baten garapenerako, datu lexikalen sei ezaugarri bereizten ditu:[19]

1. **Datuen eleaniztasuna.** Konputagailuetan metatutako hitz oro hizkuntza jakin batekoa izango da. Hiztegi elebidun edo eleanitzetan, halaber, bi edo hizkuntza gehiagoko hitz-sortak nahasirik gordeko dira. Eleaniztasunaren ondorio garbi bat karaktere berezien arazoa delakoa da. Izan ere, hizkuntzatako alfabeto-hizkien grafiak aldakorrak dira, eta hizkuntza guztietako hizki bereziak kodetzeko ez dago ASCII bezalako estandarrik[20]. Pentsa dezagun hizkuntzen artean aurki daitezkeen hizki berezietan: gaztelaniazko ñ karakterea edo karaktere azentudunak ez dira ingeles hizkuntzan aurkituko; zer esanik ez hizkuntza arabiar edo ekialdekoak (txinera, japoniera), zeintzuen grafia erabat desberdina den mendebaldeko hizkuntzekiko. Konputagailuak haien arteko desberdintasun guztiak adierazteko aukera eman behar du, erabilpen zabala izan

[19]Egilea, bere lanean, datu linguistiko orokorrei buruz mintzatzen bada ere, bere konklusioak datu-base lexikalei bete-betean egokitzen zaizkiela uste dugu.

[20]Azken aldian, ordea, karaktere berezien arazoa lasaitu da, neurri handi batean, hizkuntzen eleaniztasunari buruzko gorabeherak sistema eragilearen arlora lerrarazi baitira, *Unicode* edo ISO-Latin karaktere sortak arazo honen irtenbide desberdinen lekukoak izanik.

nahi duen datu-base lexikala egin nahi bada. Gauzak horrela, datuen eleaniztasunari aurre egitea bi eginkizunen menpe egongo da. Batetik, datu oro zein hizkuntzatakoa den gordetzea. Bestetik, karaktereen kodekera funtzionala erabiltzea, hau da, karaktere batek dokumentu osoan betetzen duen funtzioa adieraztea; karaktereak irudikatzeko bete beharreko urratsak —bere testuinguru eta funtzioaren arabera— konputagailuari lagako zaizkio, era bateratu batean. Ikus Becker-en lana (1984), karaktere-kodekeran forma eta funtzioa aldentzearen buruzko azterketa ikusteko.

2. **Datuen sekuentzialtasuna.** Datu lexikalen elkarrekiko ordenak garrantzia du maiz. Esate baterako, definizio-testu baten osagaien arteko ordena aldatzean bere esanahia ere alda daiteke. Aitzitik, egun usuen erabilitako datu-baseak kudeatzeko sistemek (eredu erlazionalean kokatutakoek) ez dute datuen arteko ordena erlatiboak islatzeko aukerarik ematen: erlazioak multzoen bidez adierazten dira, hau da, eredu erlazionalaren oinarria multzoa da eta, beraz, segidaren kontzepturik gabekoak.

3. **Datuen izaera hierarkikoa.** Datu lexikalak maiz antolatu izan dira egitura hierarkikoetan, lexikoia sare hierarkikoa bezala antolatua dagoenean datuak era labur eta zehatzean adieraz baitaitezke, ezaugarri linguistikoen taldekatze egokia lortuz. Klase/azpiklasean oinarritutako diseinu garbi bat arras garrantzitsua bilakatuko da mantentze-lanak edo datu lexikalen gaineko aldaketak egiteko garaian. Gauzak horrela, herentzia funtsezko ezaugarria bilakatzen da datu-base lexikalen garapenean. Nahi bezain beste adibide aurki ditzakegu: hiztegi-sarreren egitura hierarkikoa da[21]. Maila sintaktiko batean ere, datu lexikalen kategoriak hainbat mailatan modelatu ohi dira, maila desberdinen artean atributuak heredatuko direlarik. Hitzak morfemaz osaturik daude eta soilik osagaien informazio puskak konbinatuz, hots, atributuak heredatuz, lortuko da hitzaren informazio morfologiko edo morfosintaktikoa asmatzea. Ikus daitekeenez, zerrenda luzea da.

4. **Dimentsioaniztasuna.** Ikuspuntu linguistiko batetik, dimentsio bakuneko karaktere-segida bezala adierazitako datu zatiren batek hamaika esanahi eta interpretazio desberdin izango ditu. Jadanik ikusi dugun

[21] II.2 irudiak, 44. orrian, EH hiztegiko sarrera bat erakusten du.

bezala, *Grosseto*ko mintegian jaio zen estreinako aldiz tamaina errealeko lexikoi zabal eta berrerabilgarriaren ideia. Lexikoi hauek hainbat domeinu zein ikuspuntu linguistiko desberdin eduki ditzaketen aplikazioetarako iturri lexikal izan behar duten heinean, datu hauen interpretazioaren anizkoiztasuna bermatu beharko dute. Ikuspuntu hauen zenbatekoa teoria linguistikoen kopuruaren adinakoa da: morfologikoa, morfofonologikoa, morfotaktikoa, morfosintaktikoa, sintaktikoa, lexiko-semantikoa eta abar luze bat.

5. **Datuen izaera integratua.** Datu lexikalen arteko erlazioa berezia da, datu hauen izaera integratua baita. Sarrera lexikal batean kode daitekeen kategoria gramatikala, kasu, ez dagokio berez sarrerari berari, gramatika baita kategoria sintaktiko posible guztiak definitzen dituena. Horrela, bada, sarrera lexikalen kategoria, aldez aurretik jakina den kategoria normalizaturen batera eramaten gaituen estekaren bitartez adierazi beharko da. Hitzen kasua ere esanguratsua da: hitzak morfemaz osatzen dira, eta morfema hauek hitzaren azaleko forma gauzatuko dute, sorkuntza morfologikoaren bidez, hots, morfofonologia edo morfotaktikaren bidez. Datu-baseen arloan diseinu okerraren seinale den datu errepikatuen arazoa saihesteko, hitzaren forma ez litzateke karaktere-segida bezala adierazi behar; horren ordez, forma osatzen duten morfemen gehi morfema hauek konbinatzen dituzten erregela morfofonologiko/morfotaktikoen bidez errepresentatu beharko da. Hitz anitzeko unitate lexikaletan ere terminoak hainbat hitzez daude osaturik. Datu hauek errepresentatzeko, beraz, osagaien erakusleak gehi termino konposatuak jasotzen dituen atributuak adierazi beharko dira. Hauek denak datu lexikalen izaera integratuaren ondorioak dira. Izaera integratua islatzeko datu hauek asoziazio-amaraunak osatzen dituztela esaten da. Weber-en lanean (1986) datu linguistikoen amaraun bezalako adierazpidearen beharra azaltzen da.

6. **Informazioaren eta formatuaren arteko bereizketa.** Datu lexikalei buruzko informazioa eta forma bereiztea ezinbesteko ezaugarria dugu datu-base lexikala zabala izatea nahi bada. Horrela, bada, datuen informazioa bitan banatuko da: alde batetik, bere atributu lexikalak, eta bestetik, datua azaltzeko bete beharreko prozedurak. Hurbilpen honi jarraituz abantailak ugari lortuko ditugu: formatuari buruzko gorabeherak ez dira datu berriak sartzerakoan erabaki behar, eta gerorako

> utz daitezke. Elementuen egonkortasuna bermatuko da, mota bereko elementuek forma bera edukiko baitute. Formatuen aldaketa ere globalki egingo da, hots, mota bereko datuak era uniformean aldatuko dira. Bestalde, datu-baseen plataformen arteko portabilitatea erraztuko dugu. Azkenik, bereizketa honek datuen trataera konputazionala ere erraztuko du, behar den informazio lexikala esplizituki baitago gordeta.

Hauek lirateke, beraz, aplikazio linguistikoak datu lexikalez hornitzeko sistemek kontuan hartu beharreko ezaugarri nagusiak, datu linguistikoen biltegiak izango badira. Datu hauen izaerak ezartzen ditu, azken finean, gordelekuen ezaugarriak. Hala ere, nekez aurkituko dugu, egun, baldintza guztiak betetzen dituen sistemarik, ezta informazio lexikalaren arloan ere. Izan ere, hitzen —eta adieren— arteko erlazio lexikal oro errepresentatzea, errepresentazio horren gainean interpretazio anitzen ikuspuntutik informazioa berreskuratu ahal izatea, datuen atzipen eraginkorra eskaintzea, eta, gainera, datu kopuru handiak gordetzeko aukera izatea eskakizun oso gogorrak baitira biltegiratze-sistema batentzat.

Kontuak kontu, datu-base lexikalak hainbat formatu eta ereduren pean paratu izan dira. Jarraian, datu hauek metatzeko erabili ohi diren hainbat eredu ikusiko ditugu. Azalpen honetarako, hiru bereizketa nagusi egingo ditugu: testu ereduak, datu-baseak eta ezagutza-baseak. Bereizketa hau egiteko erabilpen praktikoan oinarritu gara, informazio lexikala gordetzeko eredu usuenak direla sinetsita baikaude. Eredu bakoitza aztertuko dugu, eta bere alde onak eta txarrak azaldu zeren, ikusiko dugun legez, ez baitago eskakizun guztiak beteko dituen eredu bat, informazio lexikala adierazteko eredu bakarra ez dagoen bezala.

II.2.1 Testu-ereduak. Markaketa.

Testu-eredua delakoa erabiltzen dela esaten da baldin datuak testu-fitxategietan gordetzen badira. Horrela, datu-basea testu-segida linealean adieraziko da, datu-baseko unitate desberdinak nolabait bereiziz. Datuekin batera datuen egitura ere adierazi behar da, jakina, eta datuak zenbat eta konplexuago izan, orduan eta egitura aberatsagoen mende egongo dira.

Egitura aldetik konplexutasun txikiena eskaintzen duena formatu tabulatua delakoa da. Formatu horretan, lerro bakoitzean item bakarra definituko da, item bakoitzak hainbat eremu dituelarik. Eremu hauen kopuruak finkoa izan behar du datu-base osoan, hau da, atributuen zenbatekoa aldez aurretik

jakina da eta item guztietarako konstantea. Eredu erlazionalaren terminologiaren ikuspuntutik, esan daiteke testu tabulatuek erlazio edo taula bakarra isla dezaketela.

Formatu tabulatuak errepresentazio-eredu sinplea eskaintzen badu ere, ez da, normalean, datu lexikalak gordetzeko euskarri egokia. Konparazio batera, datuak modu hierarkikoan gordetzekoak badira —eta ikusi dugu informazio lexikala, izaeraz, informazio hierarkikoa dela—, formatu tabulatuak hainbat eta hainbat informazio errepikatzera behartuko gaitu. Horrela, bada, formatu tabulatua informazio laua soilik gordetzeko erabiltzen da. Bere onura handiena datuak atzitzeko garaian azalduko da. Izan ere, testu-fitxategi egituratuak izanik, sistema-mailako edozein tresna erabil baitaiteke galderak egiteko (konparazio batera, *awk* edo *Perl* script-ak). Beraz, oso malgutasun handia izango dugu formatu tabulatutik datuak eskuratzeko, ia edozein murriztapenen arabera egin baitaitezke galderak. Hori bai, galderak egiteko, normalean, *script* berezi bat idatzi beharko da.

Formatu tabulatua aipatu dugu gure lan honetan, egunero erabiltzen den formatua delako, baina, egia esan, ez du interes handirik informazio lexikalaren adierazpen-formalismoak aztertzen dituen lan baterako. Badago, hala ere, informazio lexikal aberatsa testu-fitxategietan gorde ahal izateko beste formatu hedatuago eta indartsuago bat: testuak markatzea edo etiketatzea.

Oro har, testu etiketatuetan marka berezi batzuk txertatzen dira testuekin batera —fitxategi berean, alegia—, zeinei esker jakin daitezkeen datuaren antolakuntzari buruzko gorabeherak. Horrela, testu etiketatuetan informazioa gordetzen da, eta, informazio horrek datu-base tradizionaleko datu-ereduekin bat egiten ez badu ere, bertan gordetako datuak egituratuta daude. Markaketaren gorabeherak aztertuko ditugu jarraian, eta azterketa formatu honen pean maiz gorde diren baliabide lexikalen laguntzaz egingo dugu, hots, hiztegi elektronikoenez.

Historikoki, hiztegi elektroniko gehienak testu-ereduan kokatzen dira, hots, hiztegia testu gisa gorde ohi da. Hiztegiko sarrera lexikalak informazio osagarriez aberasturik gorde ohi dira, marka bereziak erabiliz. Marka hauen bidez sarreren ezaugarri inplizituak azalean agertuko zaizkio konputagailuari. Izan ere, ezaugarri franko azalaraz daiteke markaketa erabiliz. Txertatzen diren marka hauek esanahi jakina izango dute—hau da, semantika bat—, eta gordetzen diren elementuen ezaugarri inplizituak azalarazteko erabiliko dira.

Markaketa egiturazko informazioa edo informazio analitikoa adierazteko erabil daiteke. Hiztegien kasuan, hiztegi-sarreren informazio tipologikoa eskain dezake (letra lodiak, etzanak, adieren ikurrak eta abar), eta baita hiz-

laguntza *iz.* (1571) **1.** Norbaiten alde egitea, bere ahaleginak haren ahaleginei batuz edo haren premia edo beharrei erantzunez; laguntzeko egiten edo ematen den zera. Ik. **laguntasun.** *Zure laguntzaren beharra dut. Laguntza eske dator. Laguntza eman. Eskuzabalki eskaini dute laguntza. Zeruko laguntza ugariak. Inoren eta inongo laguntzarik gabe. Elkarren artean eta elkarren laguntzarekin. Izpiritu Santuaren laguntzaz. Laguntza bereziak. Aitaren diru laguntza zuelako. Ze laguntza mota izan duzue erakundeetatik?* **2.** (1802). Bizk. Lagunartea. *Aingeru guztien laguntzan.*

II.2 Irudia: *laguntza* sarrera EH hiztegian.

tegiaren egiturari buruzko informazioa ere (sarrera baten elementuak noiz hasi eta noiz bukatzen diren). Horren arabera, markaketa *tipologikoa* edo *deskriptiboa* dela esango da.

Markaketa tipologikoak —edo *prozeduralak*— euskarri elektronikoan gorderiko sarreren azaleko itxura adieraziko digute. II.2 irudian EH hiztegiko *laguntza* hitzari dagokion sarrera ikus dezakegu. Nabaria denez, sarrera ikur tipografiko zein lexikografikoez hornituta dago (letra lodia sarrera-buruetan, zenbakiak adierak bereizteko eta abar). Estreinako hiztegigintza digitalean konputagailua hiztegiaren testu-edizio lanetan soilik erabiltzen zenez, markaketa tipologikoak testu-edizioko programekin erlazio estua zuen[22]. Horrela, bada, hiztegia inprimatzeko beharrezkoak diren azaleko ezaugarriak gordeko dira. Markaketa mota hau *prozedurala* dela esaten da, prozesu lineal batek kode hauek aurkitzerakoan bete behar duen prozedura adierazten baita (adib. letra-etzanetara pasa).

Markaketa prozedurala ez da batere egokia datu lexikalak adierazteko. Batetik, atal honen hasieran aipaturiko informazioaren eta formatuaren arteko bereizketarik ez da egiten. Bestetik, markaketa mota hau aplikazio zehatzetara loturik dago ia beti. Azkenik, informazio lexikala ez da esplizituki adierazten. Adibidez, II.3 irudian hiztegia era prozeduralean markaturik agertzen zaigu, testuari itxuraketa-markak erantsi baitzaizkio. Hala ere, konputagailuak ezin du asmatu, esaterako, sarreraren definizioa zein den (noiz hasi/bukatzen den) edo sarrerak zenbat adiera dituen.

Markaketa prozeduralaren kontrako aldean markaketa deskriptiboa deritzoguna dugu. II.2 irudira itzuliz nabari daiteke, sarrera hainbat eremutan dagoela antolatua, hala nola, sarrera-burua, kategoria, adiera-ikurrak, defi-

[22]Konparazio batera, Microsoft enpresaren RTF formatua Word fitxategiak gordetzeko.

```
{\b\f57\fs16 laguntza. }{\i\fs14 iz. }{\fs14 (1571). }
{\b\fs14 1}{\fs14 . Norbaiten alde egitea, bere ahaleginak
haren ahaleginei batuz edo haren premia edo beharrei
erantzunez; laguntzeko egiten edo ematen den zera. Ik. }
{\b\fs14 laguntasun}{ \fs14 . }{\i\fs14 Zure laguntzaren
beharra dut. Laguntza eske dator. Laguntza eman.
Eskuzabalki eskaini dute laguntza. Zeruko laguntza ugariak.
Inoren eta inongo laguntzarik gabe. Elkarren artean eta el
karren laguntzarekin. Izpiritu Santuaren laguntzaz.
Laguntza bereziak. Aitaren diru laguntza zuelako. Ze
laguntza mota izan duzue erakundeetatik? }{\b\fs14 2}
{\fs14 . (1802). }{\i\fs14 Bizk. }{\fs14 Lagunartea. }
{\i\fs14 Aingeru guztien laguntzan.}
```

II.3 Irudia: *laguntza* sarreraren markaketa prozedurala (RTF).

nizioak, adibideak, erreferentzia gurutzatuak etab. Gorago aipatu dugun bezala, eremu horiek ikur tipografiko zein lexikografikoez hornituta daude, eta, kode horiek aztertuz, sarreraren eremuak markaketa deskriptiboaren bidez azalaraz ditzakegu[23]. II.4 irudian *laguntza* sarreraren markaketa deskriptiboa ikus daiteke[24]. Oraingo honetan, sarreraren elementu logiko oro marka baten barruan agertzen da, zeinek adierazten duen informazio zatiaren funtzioa.

Informazio lexikal anitz aurki daiteke etiketatze ereduaren mende. Konparazio batera, LDOCE ospetsua, *Cambridge International Dictionary of English* hiztegia, eta abar. II.1.1 atalean ikusi ditugun zenbait proiektuk ere, estandarizazioa helburu zutenez, markaketa deskriptiboa bereganatu zuen informazio lexikala adierazteko formatutzat (adibidez, MULTILEX edo GENELEX proiektuek).

Testu-ereduaren gainean eginiko azterketarekin amaitzeko, informazio linguistikoa kodetzeko formatu estandar aski sonatuaz arituko gara: TEI ekimenak proposatutako gidalerroak aztertuko ditugu, batik bat hiztegietarako

[23]Hiztegi-sarreren kode tipologiko zein tipografikoetatik beren egitura sintaktikoa automatikoki ezagutzeko saiakera ugari egin dira. Trataera automatiko hauek emaitza onak ematen badituzte ere, ezin izaten da % 100eko arrakasta lortu. Salbuespenak zein akats tipografikoak izan ohi dira errore-iturri nagusiak. Ikus Arriola eta Soroa (1996)

[24]Sarrera TEI gidalerroak jarraituz markatua dago. Ikus II.2.1.1 atala.

```
<entry>
 <form><orth>laguntza</orth></form>
 <GramGrp><pos>iz.</pos></GramGrp>
 <usg type="time">1571</usg>
 <sense n="1">
  <def>Norbaiten alde egitea, bere ahaleginak
   haren ahaleginei batuz edo haren premia
   edo beharrei erantzunez; laguntzeko egiten
   edo ematen den zera.</def>
  <xr type="syn"><lbl>Ik.</lbl><ref>laguntasun</ref></xr>
  <eg><q>Zure laguntzaren beharra dut.</q><q>Laguntza eske
   dator.</q><q>Laguntza eman.</q><q>Eskuzabalki eskaini
   dute laguntza.</q><q>Zeruko laguntza ugariak.</q>
   <q>Inoren eta inongo laguntzarik gabe.</q><q>Elkarren
    artean eta elkarren laguntzarekin.</q><q>Izpiritu
   Santuaren laguntzaz.</q><q>Laguntza bereziak.</q>
   <q>Aitaren diru laguntza zuelako.</q><q>Ze laguntza
   mota izan duzue erakundeetatik?</q></eg>
 </sense>
 <sense n="2">
    <usg type="time">1802</usg>
    <usg type="geo">Bizk.</usg>
    <def>Lagunartea.</def>
    <eg><q>Aingeru guztien laguntzan.</q></eg>
 </sense>
</entry>
```

II.4 Irudia: *laguntza* sarreraren markaketa deskriptiboa, XMLz, TEI gidalerroak jarraituz.

asmatutako etiketatzea nabarmenduz.

II.2.1.1 SGML/XML markaketa-lengoaiak eta TEI ekimena.

1978. urtean, ANSI ("American Standard National Institute") erakundeak testu-prozesamenduan ari ziren hainbat talde jarri zituen harremanetan, edozein motatako testuak kodetzeko, egituratzeko eta elkarren artean trukatzeko balioko lukeen lengoaia estandar eta orokorra definitzeko helburuarekin;

1980 urterako lengoaia horren lehenengo txostenak argitaratu baziren ere, 1985 urtean elkarlanaren emaitzaren azken bertsioa argitaratua izan zen, ISO ("International Standard for Organization") erakundeak estandartzat onartu zuena: ISO 8879 edo SGML (Standard Generalized Markup Language) lengoaia. Testuak osagaien bidez (paragrafoak, listak, izenak, atalak, lerroak, etab.) zatitu, eta zati horiek habiatu daitezkeela da SGMLren oinarrizko hipotesia. SGML lengoaiak printzipio deskriptiboa bereganatu zuen hasieratik. Horrela, testua kodetze-lanetan ari denak testu-objektua *zer* den markatuko du, testu-objektu horrekin konputagailuak zer egin behar duen markatu ordez. Honen ondorioz, testu bera hamaika kodekera desberdinen arabera marka daiteke, aplikazio desberdinen beharrei aurre eginez.

Hala ere, SGML lengoaia aberatsegia eta zabalegia suertatu zen konputagailuez tratatu behar zenean, eta, horrela, aplikazio gutxik inplementatu zituzten SGMLk eskaintzen zituen aukera guztiak. Gauzak horrela, SGML lengoaiatik azpilengoaia bat sortu zen, XML ("eXtensive Markup Language") deiturikoa, nolabait SGMLk eskaintzen zituen aukera zabalak murriztu nahian. SGML eta XML lengoaiak helburu desberdinak betetzeko jaio ziren: lehenengoak edozein dokumentu markatzeko euskarria eskaintzen zuen bitartean, bigarrenaren xedea Interneteko orrietara egokitu zen nagusiki. Horrela, bada, XML lengoaia *web*-eko estandarra zen —eta oraindik den— HTMLren[25] gabeziak gainditzeko asmoarekin jaio zen.

SGML dokumentuaren sortzaileak, testu batean objektu bezala markatu beharreko ezaugarriak definituko ditu eta *elementuen* izenak emango dizkio. Testu batean ager daitezkeen elementu oro Dokumentu Motaren Definizio ("Document Type Definition", DTD) batean gordeko dira eta, hortaz, esaten da SGMLk dokumentu motaren nozioa ezartzen duela: dokumentu bakoitza mota batekoa da, eta dokumentu mota batek dokumentu multzo bat definituko du. DTDan elementuen arteko erlazioak definituko dira —testuingururik gabeko gramatika baten arabera—, eta dokumentuaren mota formalki definituko da, bere osagaiak eta egitura esplizituki adieraziz. SGML dokumentu baten zuzentasuna baiezta daiteke, SGMLko *parser* baten laguntzaz, kodetutako testua DTDarekin bat datorrenetz egiaztatuz.

XMLrekin batera, DTDa baino ahalmentsuagoak diren definizio-lengoaiak agertu dira *XML Schema*, *Relax NG Schema* eta abar. Dokumentuen egitura

[25]XML, SGML bezala, metalengoaia da, eta lengoaia desberdinak definitzeko aukera ematen du. HTML, aldiz, SGML bidez definitutako lengoaia soil bat da, esan bezala aplikazio konkretu bat.

sintaktikoa ezartzen dute hauek ere, baina dokumentuetan egon daitezkeen elementuak zein atributuak zehazteko askatasun eta adierazkortasun handiagoa ematen dute DTDen aldean.

SGML/XML lengoaiak, beraz, testuaren gaineko markaketa burutzeko plataforma ezin hobea ditugu. Edonola ere, testu linguistikoak kodetzeko garaian, kodekera-lengoaia edukitzea arazoaren zati bat soilik ebaztea da: testuen errepresentazioa. Haatik, beste arazo garrantzitsu bati ere aurre egin beharko diogu: testuak trukatzeko aukera eskaini. Izan ere, SGML/XML lengoaiek ez baitute DTDko elementuak zeintzuk izan behar diren, ezta ere elementuen arteko egiturarik proposatzen. Testu linguistikoei dagokienez, TEI ("Text Encoding Initiative") delako ekimena jaio zen hutsune hau betetzeko. SGML/XMLek testuak markatzeko oinarrizko tresneria baino ez dute eskaintzen, gauzak egiteko hamaika bide zabalik utziz. Izan ere, SGML/XML marka multzo bat baino, marka multzoak espezifikatzeko metalengoaia baitira, eta metalengoaia horretaz baliatuz diseinatu dira TEI gidalerroak.

Gauzak horrela, 1987an, ACH ("Association for Computers and Humanities") elkarteak bilera batera deitu zituen 30 aditu (New York-eko *Vassar* eskolara), testuen kodeketaren estandarizazioa aztergai. Aditu horiek ados etorri ziren esatean ezen, aurrera jotzeko, jardunbide komun bat ez ezik, literatura- eta hizkuntza-datuak kodetzeko eta trukatzeko gidalerro batzuk ere zehaztea behar-beharrezkoa zela. Vassar-ekoaren ondoren, ACHri ALLC ("Association for Literary and Linguistic Computing") eta ACL ("Association for Computational Linguistics") batu zitzaizkion, eta hortik sortu zen TEI izeneko ekimena. Eta oraingoan, arrazoiak arrazoi, gaiak aurrera egin zuen, eta horren ondorio dira argitara emandako bere gidalerroak, hasiera batean TEI P3 deiturikoak, SGMLn oinarrituak (Sperberg-McQueen eta Burnard, 1995), eta, beranduago, TEI P4 delakoak, XMLn oinarrituak (Sperber-McQeen eta Burnard, 2002). TEIren helburuak erdiestea, giza zientzien arloan ez ezik, hizkuntzaren industriaren munduan, oro har, ere beharrezkoa zela ikusi zen berehala, ikerkuntzan zein industrian ezinbesteko bihurtu baita, edozein testu hainbat aplikaziotan erabili edota, hobe, berrerabili ahal izatea, testua kodetu zen garaian imajinatu ere egin ez ziren aplikazioetan barne.

Hona hemen Vassar-eko eskolako bilkura hartan ezarritako printzipio batzuk (Ide eta Véronis, 1995) :

- Gidalerroen helburua giza zientzietako ikerkuntzan datu-trukerako eta testuen kodeketarako formatu estandarra eskaintzea da.

- Gidalerroek gomendatu behar lukete sintaxi jakin bat formatu horretarako, testu-kodetze eskemak deskribatzeko metalengoaia bat definitu, eta formatu berria deskribatu metalengoaia horretan eta hizkuntza arruntean.

- Aldez aurretik kodetutako testu konputagailuz irakurgarriak formatu berrira itzultzeak haien kodeketa-sintaxia formatu berrikora aldatzea esan nahi du, baina ez da eskatuko testu haietan ez zegoen informaziorik eransterik.

Gidalerroen garapenean, TEIk identifikatu zituen askotariko ikertzaileek zer-nolako kodetze-premiak zituzten informazioaren elkartrukeari zegokionean, horretan oinarritu zituen orokor izan nahi zuen eskema batek betebeharreko kodeketa-printzipioak, eta identifikatu zituen zein ziren kodetze-arauak behar zituzten testu-klase eta ezaugarriak. TEIk eskaintzen dituenetan arakatzen hasita, hona hemen batzuk:

- SGML/XML markatze-lengoaiak egokitzat jotzea gidalerroen garapenerako oinarri gisa.

- SGML/XML erabiltzeko gomendioak — zenbait murriztapen —, beren orokortasunari eta malgutasunari eutsiz aldi berean.

- Testu-datuak kodetzerakoan beharrezko diren kategoria eta ezaugarrien identifikazioa eta analisia, maila askotan.

- Testu-egitura definizio orokorren multzo malgu eta hedagarria.

- Testu elektronikoak dokumentatzeko metodo bat, biblioteketan erabiltzen den katalogatze-arauekin bateragarria dena.

- Kodetze-arauak testu mota eta ezaugarri desberdinetarako: karaktere multzoak, hizkuntz corpusak, linguistika orokorra, hiztegiak, datu terminologikoak, ahozko testuak, hipermedia, literatur prosa, olerkia, antzerkia, iturburu historikoak eta testu-kritikarako aparatua.

Hasiera-hasieratik diseinatu zen TEI eskema hardware, software eta aplikazioetatik independente izateko helburuarekin. Aplikazioetatik independente izan nahi horrek izugarrizko garrantzia du, gure ustez, eta testu baten

ikuspegi desberdinak kodetu ahal izateko aukera ematen digu. Izan ere, testu bat har baitaiteke objektu fisikoen bilduma bezala (liburukiak edo paperorri solteak), edo objektu tipografikoen segida bezala (karaktere-sekuentziak, letra-molde eta marjina-eskema desberdinen arabera antolatuak), edo objektu linguistikoen sekuentzia bezala (grafema edo fonemak, morfemak, unitate lexikalak, sintagmak, ...), edo objektu formalez osatutako egitura bezala (estrofak, lerroak, kapituluak, atalak, ...), eta abar eta abar. TEI gidalerroek helburu orokorreko kodetze-eskema bat definitzen dute, ikuspegi desberdin horiek guztiak era desberdinetan kodetzeko aukera ematen duena, eta, nahi izanez gero, aldi berean gainera.

Informazio lexikala kodetzeko gomendioak argitaratu ditu TEIk, batik bat, hiztegiak kodetzeko era estandar eta orokorra proposatuz. Hiztegietako informazioa luze eta zabal jorratzen duelarik, baliabide lexikal horietan aurki daitekeenaren berri aberatsa ere ematen digu TEIk.

SGML lengoaiak XMLra bide eman zion heinean, TEI ekimenak ere ildo beretik egin zuen, eta, horrela, XML lengoaiarekin bat datozen TEI P4 gidalerroak proposatu zituen. TEI gidalerroak SGML/XMLren gainean "eraikiak"dira, beraz. Hortaz, TEI gidalerroek, SGML/XML beren egitean, beste kodetze-eskema askorekiko bateragarritasuna lortu dute *de facto*.

II.2.2 Datu-baseak.

Datu-baseak 1950eko hamarkadaren erdi aldera jaio ziren, konputagailuekin batera ia, hauek eskainitako tresna nagusiak bilakatu zirelarik. Beren helburu nagusia datuen informazio kopuru handiak kudeatzea da, datu hauek atzitzeko, eskuratzeko eta aldatzeko aukera ematen duen eskema edo "datuen eredu" jakin baten arabera. Datu-baseei informazioaren biltegi pasiboak deitu ohi zaie, beraiek gordetako datuen gaineko operazioak datu-basetik kanpoko aplikazioek egiten baitituzte, era esplizitu batean. Aurrerago ikusiko dugun bezala, ezaugarri hauxe izango da datu-base / ezagutza-base ereduen arteko desberdintasun nagusietako bat.

Nolanahi ere, datu-baseek 30 urteko eboluzioa izan dute, eta egungo datubaseen kudeaketaren teknologia oso heldua dago. Datu-baseak kudeatzeko sistema (DBKS) baten ezaugarri nagusiak honako hauek dira:

- **Independentzia fisikoa / logikoa**, hots, arkitektura orokorraren maila jakin bateko eskemaren definizioan aldaketak egiterakoan, aldaketa horiek zuzenean goian dagoen mailan eraginik ez izatea.

- **Erredundantzia minimoa**, zeinaren bitartez datu-basea aplikazio desberdinetarako datu-biltegia izango den.

- **Atzipen konkurrentea.** Hainbat erabiltzailek datu-basea paraleloan atzi dezake. Horrela, bada, datuen egonkortasuna bermatzeko —erabiltzaile bat aldatzen ari den datu bat beste erabiltzaile batek eskatzen badu, kasu— atzitutako datuen blokeo-teknikak erabiliko dira.

- **Datuen banaketa espaziala.** Independentzia fisikoak/logikoak datubase sistema banatuak ahalbideratzen ditu. Horrela, bada, datu-basea hainbat tokitan egon daiteke banatua: datuak beste gela batean, beste eraikuntza batean edo beste herri batean egon daitezke baina erabiltzaileari datu-basearen ikuspegi orokorra eskainiko zaio.

- **Datuen egonkortasuna**, hots, datu akastunak sartzea ekidingo duten segurtasun-neurriak. Datuak akastunak izan daitezke arrazoi fisiko (hardware arazoak) zein logikoengatik (zentzurik gabeko datuak sartzen badira).

- **Kontsulta konplexuen optimizazioa.** Optimizazioaren bitartez kontsulta konplexuak azkarrago erantzungo dira.

- **Atzipen-segurtasuna eta auditoria**, hau da, pertsona zein erakunde desberdinek eskubide desberdinak edukiko dituzte datu-baseko datuak atzitzeko. Auditoriak, bestalde, datuen gainean atzipen-kontrola mantenduko du, datu-basean aldaketa bat nork eta noiz egin duen jakitearren.

- **Lehengoratzea.** Datu-galera baten aurretik zegoen egoerara itzultzeko ahalmena.

- **Programazio-lengoaia estandarren bidezko atzipena**, hots, lengoaia arrotzak erabili ahal izatea datu-baseko datuak atzitzeko.

Oro har, datu-baseen xedea bikoitza dela esan dezakegu:

- Dituen datuen galderei erantzun.

- Transakzioak burutu.

Galdera bat eskema kontzeptualean definitutako objektu zein erlazioen gaineko espresio logiko baten bidez adieraziko da, eta galderaren emaitza datu-basearen azpimultzo logiko baten identifikazioa izango da. Transakzio bat, berriz, azpieskema baten gaineko kontsulta-/aldaketa-eragiketen multzo bat da. Transakzioak *atomikoak* dira, hau da, transakzioko urrats bakoitza bete eta baieztatu egin behar da, ezinbestean, transakzioa bera burutzeko. Transakzioaren urrats bat ezin bada bete, transakzio osoa ezeztatuko da.

Datuen eredua delakoa matematikoki definitutako kontzeptu multzo bat da, zeinek eskainiko dien oinarri formal bat informazio-sistemen garapen eta erabilpenerako teknikei eta tresnei. Halaber, datuen erabilpen trinkoa egiten duten aplikazioei ere oinarri kontzeptuala eskainiko die, eta aplikazio hauen ezaugarri estatiko zein dinamikoak adierazten lagunduko. Kontzeptualki, aplikazio batek honako ezaugarri hauek izango ditu:

1. Ezaugarri estatikoak: entitate (edo objektu) eta entitate horien propietateak (edo atributuak), entitateen arteko erlazioak islatzen dituztenak.

2. Ezaugarri dinamikoak: entitate edo propietateen arteko eragiketak, eta eragiketa hauen arteko erlazioak.

3. Entitate eta eragiketen gaineko egonkortasun-erregelak.

Horrela, bada, datu-ereduak ezaugarri hauei eskaintzen dien trataeraren arabera bereiziko dira. Datu-eredu baten bidezko datu-modelizazioaren emaitza bi osagai dituen errepresentazioa da: ezaugarri estatikoak eskemaren bidez adieraziko dira, eta ezaugarri dinamikoak transakzio edo kontsulten espezifikazioen bidez adieraziko dira. *Eskema* bat aplikazio baten objektu mota guztien definizioa da, beren atributu, erlazio eta murriztapen estatikoekin batera. Datu-basea bera, hortaz, eskemaren instantzia bat izango da.

Oro har, aplikazioek eskema batean definitutako entitateen artean azpimultzo bat soilik atzitu behar izaten dute maiz. Beraz, aplikazio hauek soilik ezaugarri estatikoen azpimultzoa behar izaten dute. Ezaugarri estatikoen azpimultzo honi *azpieskema* esaten zaio. Transakzio bat, eskema edo azpieskemaren entitateen gaineko eragiketa sorta izango da. Kontsulta bat, bestalde, eskema batean definituriko entitate eta erlazioen gaineko espresio logikoa bezala adieraz daiteke; kontsulta batek datu-basearen azpimultzo bat identifikatuko du.

Datu-baseen teknologian garatu izan diren datu-eredu guztiak azaltzea urrundu egiten da lan honen helburutik. Edonola ere, interesgarria deritzogu

gutxienez datu-ereduen artetik bi eredu aipatzea, datu lexikalak gordetzerakoan usuen erabilitakoak baitira: eredu erlazionala eta objektuei zuzendutako eredua.

Datu-baseetan egun usuen erabilitzen den eredu erlazionalak informazio lexikala kudeatzeko aukera hobeak ematen ditu testu-ereduak eskainitakoarekin erkatuz, zalantzarik gabe. Horrela, datu-baseak maiz erabili dira informazio lexikala adierazteko. Batez ere, trataera konputazionalera begira dauden baliabide lexikal gorde dira era honetan, esate baterako, GENELEX (Normier eta Nossim, 1990), COMLEX (Grisham *et al.*, 1994) edo EDBL (Aldezabal *et al.*, 2001).

Bestalde, giza-erabiltzaileei zuzendutako hiztegiak ere datu-baseetan errepresentatzeko saio ugari egin dira. Nakamura eta Nagao-ren lanean (1988) LDOCE hiztegia datu-base erlazionalean gordetzeko estreinako saiakeretako bat ikus daiteke. Egileek sortutako eskeman hiztegia erlazio multzo bat bezala errepresentatzen da, erlazio bakoitzak zenbait atributu dituelarik. Atributu hauek kode gramatikalak, definizioak, adibideak, etab. errepresentatzen dituzte. Ide *et al.*-en lanean (1994), berriz, hiztegietako sarrera lexikalen egitura errepresentatzeko hainbat ereduren egokitasuna aztertzen da. Lanean garbi esaten da eredu erlazionala ez dela egokia hiztegiko informazioa errepresentatzeko. Besteak beste, arazo hauek azpimarratzen dituzte:

- *Datuen fragmentazioa.* Fragmentazioa sortzen da, batez ere, hiztegietako sarreren atributuek har dezaketen balio kopurua oso aldakorra delako. Esaterako, sarrerek ebakera, kategoria, azpikategoria edo definizio anitz gorde ditzakete, eta, aldi berean, beste eremuei buruzko —adibideak, sinonimoak, informazio geografikoa eta abar— inongo informaziorik ez. Datu kopuruaren bikoizketa izugarria nahi ez bada, informazio hori guztia hamaika taulatan antolatu beharko da. Ondorioz, datuak oso sakabanatuak, hots, fragmentatuak, gorde behar dira. Fragmentazioa dela eta, galdera konplexuak eta nahasiak idatzi behar dira datuak atzitzeko.

- *Egitura hierarkikoa.* Aipatu dugu jadanik informazio lexikalaren izaera hierarkikoa, eta hiztegietako sarrera lexikalak ez dira salbuespena. Adibidez, II.4 irudian[26] ikus daitekeenez, EH hiztegiko *laguntza* sarrerak bi zati ditu, bat adiera bakoitzeko. Kategoria gramatikala, baina, sarreraburuan definitzen da, eta pentsa daiteke bi adierek kategoria gramati-

[26] 46. orrian.

kala bera jasoko dutela. Informazio hori, horrela, faktorizatua agertzen da: sarrerako adiera guztiek (besterik ezean) kategoria bera heredatuko dute. Datu-base erlazionalak ez dira egokiak hiztegietako sarrera lexikalen izaera hierarkikoa errepresentatzeko: ereduaren egonkortasuna bermatzeko, faktorizatutako atributu bakoitza adiera zein azpiadiera guztietan errepikatu behar da eta, beraz, informazioa biderkatu. Adieren arteko hierarkia maila handia eta konplexua izan daitekeenez[27], hierarkiarena da hiztegiak datu-base erlazionaletan gordetzeko traba larrienetako bat.

Lan berean objektuei zuzendutako ereduaren alde egiten da nabarmen, eta eredu horren egokitasuna azpimarratzen da hiztegietako informazioa errepresentatzeko. Hala ere, eta salbuespenak salbuespen[28], objektuei zuzendutako eredua jarraitzen duten datu-base lexikalen kopurua ez da oso handia, eta, oraindik ere, gehienak eredu erlazionalean kokatzen dira. Horrela, nahiz eta baliabide lexikalaren osakeran objektuei zuzendutako diseinua erabili, inplementazioa eredu erlazionalean gauzatzen da askotan.

II.2.2.1 Datu-base sasi-egituratuak.

Gaur egun informazioa izugarri hedatu da, batez ere, Internet sarea dela medio. *Informazioaren gizartean* murgilduta gauden garai hauetan, informazioaren eskurapenaren arazoak ikuspuntu berriak hartu ditu. Horrela, datubaseen arloan erronka berriak agertu dira, datuak gordetzeko bide berriak sortu baitira.

Datu-base tradizionalaren ikuspuntuan informazioa bilgune bakar eta zentralizatu batean metatuta dagoela aurreikusten da[29], eta, halaber, datu horiek datu-eredu zehatz baten arabera gorderik daudela. Egoera, baina, bizkor aldatzen ari da: gaur egun, hainbat informazio banatuta agertzen zaigu, *web* orrietan zehar sakabanaturik. Bestalde, Interneten aurki daitekeen informazioa, datu-base tradizionaletan gorderikoarekin alderatuz, askoz ere *lausoagoa* da, hots, datuak ez daude hain eskema finkoez murriztuta. Gauzak

[27] Adibidez, EH hiztegian 4 adiera-maila aurki daitezke: adiera multzoak, adiera arruntak, adiera xeheak eta ñabardurak (Sarasola, 1996).

[28] Adibidez, CELLAR sistema objektuei zuzendutako eredua jarraitzen duen datu-base batean errepresentatzen da (Simons, 1997).

[29] Horrela izan zen, behintzat, datu-baseen hastapeneko sistemetan. Alabaina, arloan aritutakoek berehala ikusi zuten informazioa sakabanatua izatearen onura —eta arazoa—, eta datu-base banatuek sortutako erronka berriei heldu zieten (Özsu eta Valduriez, 1999)

horrela, datu-base sasi-egituratuen arloak garrantzi handia hartu du azken boladan, laurogeita hamarreko azken aldetik gaurdaino (Abiteboul, 1997; Florescu *et al.*, 1998b; Abiteboul *et al.*, 1997; Florescu *et al.*, 2000).

Sasi-egituratutako testuen ezaugarriak, datu-base tradizionalen aldean, hauek lirateke (Florescu *et al.*, 2000):

- Eskema ez dago aldez aurretik finkatua, eta, zenbaitetan, datuekin batera inplizituki egon ohi da. XMLz kodetutako dokumentuek, adibidez, bi osagai nagusi dituzte: testua bera eta elementuen egitura deskribatzen duen gramatika[30]. Dokumentuaren egitura analizatzerakoan (*parser* baten bidez), informazio-zatiak identifika daitezke, eta, baita ere, zati horien guztien arteko erlazioak. Hala ere, erlazio horien interpretazioa XML lengoaiatik at geratzen den zerbait da, aplikazio zehatzek ezarri beharrekoa. Beraz, XML dokumentuen eskema inplizitua dela esan ohi da, zeren, alde batetik, prozesu bat egikaritu behar baita ezagutzeko —analisia, alegia—, eta, bestetik, analisi-zuhaitzetik ezin baita adierazpen logikoa zein den zuzenean ezagutu.

- Eskemaren tamaina nahiko handia da, eta, gainera, aldakorra. Datu-base tradizionalaren eskemaren tamaina txikia da, datuen kopuruarekin alderatzen bada. Halaber, datu-baseen eskemak nahiko finkoak dira, hots, ez da aurreikusten, datu-basearen jarduera arruntean, eskemaren gainean aldaketa handiak egingo direnik. Egoera desberdina da datu-base sasi-egituratuetan: testuinguru horretan, datuen eskema malguagoa da, eta eskema alda daitekeela aurreikusi behar da, datuak ere aldatzen diren legez. Ondorioz, erabiltzaileek ez dituzte, orokorrean, eskemaren zehaztasun guztiak ezagutuko. Galderak egiteko, baina, datu-basearen eskema nolakoa den ezagutzeak ezinbestekoa dirudi.

- Eskema deskribatzailea da, arauemaile baino. Horrela, datu-base sasi-egituratuen eskema datuen egoera deskribatzeko erabiltzen da, baina eskemarekin bat etortzen ez den daturik ere ager daiteke.

- Datuen motak ez dira zorrotzak, hau da, objektu desberdinetarako, atributu baten balioak mota desberdinetakoak izan daitezke.

Funtsezko arazo bat datu-eredu formal baten aukeraketan datza —eta, ondoren, kontsulta-lengoaiarena—. Datu-base sasi-egituratuen datu-ereduak

[30]Gramatika deskribatzeko *Document Type Definition* (DTD) delako dokumentua dago, edo ahaltsuagoa den eskema-lengoaia batez adierazitakoa (*XML Schema*, *Relax NG* ...).

aberatsa, zorrotza eta konplexua izan behar du? Edo, aitzitik, sinplea eta zama gutxikoa? (Abiteboul, 1997) lanean, autoreak dio datu-ereduak bi baldintza batera bete beharko lituzkeela. Batetik, eredua ezin da oso zorrotza izan, datuen egitura ere zurruna ez den neurrian. Horrela, datu-ereduak datuen elkartrukatzeari zuzendua izan beharko lukeela dio. Izan ere, datu-eredu malgu eta sinple batek informazio sorta handia kode lezake, nahiz eta datuen gainean murriztapen zorrotzik ezarri ez.

Bestalde, baina, datu-eredu aberats batek datuen gaineko analisiak egiteko aukera emango luke, hots, datuen gaineko semantika aztertzeko. Datubase sasi-egituratuek ere datu oso egituratuak gorde ditzakete beren baitan, eta, horrela, sistemak egiturari buruzko informazio zehatzaren laguntza izan dezake hainbat eginkizunetan: datu-eredu aberats batek horixe bera egiteko aukera emango du.

Datu-base sasi-egituratuen arloan XMLz etiketatutako testuek arreta berezia jaso dute, Interneten zehar informazio franko modu horretan baitago adierazia. Horrela, bada, XML lengoaia testu egituratuen formalismo estandarra bihurtu da, eta informazio sasi-egituratuaren inguruko ikerkuntzak XML lengoaiaren gainean kontsulta-lengoaia sendoen garapenari ekin dio, datuak adierazteko formalismo berriak asmatzeari baino (Baeza-Yates eta Navarro, 2002).

XML lengoaia, formalismo estandarra bihurtu den neurrian, XML dokumentuen gaineko XPath (WWW Consortium, 2000) datu-eredua onartu da komunitate zientifikoan. XPath ereduak —XSLT transformazio-lengoaiaren osagaia ere badenak— XML dokumentuen zatiak identifikatzeko ematen du aukera, eta, horretarako, XML dokumentuak zuhaitz-tankerako egituren bidez adierazten ditu. Zuhaitzaren nodoak hainbat motatakoak izan daitezke (elementuak, atributuak, testu-osagaiak etab.) eta nodoen arteko erlazioak arkuen bidez adierazten dira, hauek ere hainbat motatakoak izan daitezkeelarik (*parent-of, child-of, sibling*). XPath lengoaiaren ezaugarri garrantzitsu bat nodo baten umeen ordena ere kontuan hartzearena da.

XPath lengoaiaz gain, XML dokumentuen gainean kontsultak egiteko lengoaia sorta handi bat sortu da azken boladan, hala nola, Lorel (Abiteboul *et al.*, 1997)[31], XQuery (Chamberlin, 2002) edo XML-QL (Lapp *et al.*,

[31]Lorel ez da, berez, XML dokumentuak adierazteko, bazik eta edozein datu sasi-egituratu adierazteko lengoaia. OEM lengoaia hartzen du datu-eredu bezala (ikus IV.5.1 atala). OEM eta XPath lengoaiak antzekoak badira ere, ez dira erabat baliokideak. Alde garrantzitsuena, beharbada, XPath lengoaia zerrendetan oinarritzen den bitartean, OEM multzoetan oinarritzen dela. Horrela, bada, OEM lengoaian nodo baten umeen ordenak

1998)[32].

Lengoaia horiek guztiek datu sasi-egituratuen gainean kontsultak egiteko aukera ematen dute. Gorago ikusi dugun bezala, datu sasi-egituratuek ez dute, maiz, eskema finkorik jarraitzen, datu-base tradizionalean ez bezala. Gainera, erabiltzaileak ez du zertan jakin datuak zein eskemaren arabera gorde diren. Horrela, bada, kontsulta-lengoaiek erreferentziak gauzatzeko adierazpen erregularren tankerako espresioak onartzea ezinbesteko baldintza da. "Regular Path Queries" (Calvanese *et al.*, 2002) delako espresioen bidez, dokumentu bateko zenbait datu eskura daitezke, datu horiek XML zuhaitzean duten posizio zehatza jakin gabe. Konparazio batera, II.4 irudiko (46. orrian) definizio guztiak eskura daitezke, definizio horiek edozein adieratakoak izan daitezkeelarik[33]. Are gehiago, irudian agertzen den sarrerako agerpen-data guztiak eskura daitezke, data horien habiaketa-maila edozein izanda ere[34].

Kontsulta-lengoaia aberatsez daude, bada, datu sasi-egituratuak hornituta. Hala ere, aberastasun honek galderen erantzute-prozesuaren eraginkortasunean kalte handia eragiten du. Datu sasi-egituratuen gainean —XML dokumentuen zatiak, kasu— datuak eskuratzea ataza konplexua da, datuen kopuruarekiko denbora esponentziala behar izaten duena, kasu orokorrenean. Konplexutasun honetan zerikusi handia du, noski, datuek eskema finkorik jarraitu ez izateak. Goldman eta Widom-en lanean (1997) aipatzen den bezala, datu-base tradizionalen eskemak bi xede betetzen ditu:

---

ez du garrantzirik, XPath-en ez bezala (Deutsch *et al.*, 1999).

[32]Edonola ere, badirudi XQuery izango dela XML dokumentuen gainean galdeketa-lengoaia estandarra, nahiz eta oraingoz horrelakorik erabaki ez den. Ikus `http://www.w3.org/TR/xquery/` orria.

[33]XPath-eko honako espresio honen bidez:

Espresioa:
sense/def
Erantzuna:
`<def>`Norbaiten alde egitea, bere ahaleginak ...`</def>`
`<def>`Lagunartea.`</def>`

[34]Izan ere, bi agerpen-data azaltzen baitira *laguntza* sarreran: bata goi-mailan kokatua, sarrera guztiari dagokiona, eta bestea sarreraren bigarren adierari soilik dagokiona. XPath-eko espresio honekin, baina, sarrerako agerpen-data guztiak eskura daitezke:

Espresioa:
//usg[@type="time"]
Erantzuna:
`<usg type="time`»1571`</usg>`
`<usg type="time`»1802`</usg>`

- Eskemak erabiltzaileari laguntzen dio datu-basearen egitura ulertzen, eta, hortaz, zentzuzko galderak egiten.
- Galdera-prozesatzaileek eskemaren informazioa erabiltzen dute galderak erantzuteko plan eraginkorrak lortzeko.

Gauzak horrela, XML datu-base sasi-egituratuak eta datu-base tradizionalak —normalean, eredu erlazionalekoak— uztartzeko saio ugari egin dira. Lan horietan, XML dokumentuak datu-base tradizionaletan gordetzen dira, hots, DBKSak erabiltzen dituzte XML dokumentuak gordetzeko eta haien gainean kontsultak era eraginkorrean burutu ahal izateko. Horrela, XML dokumentuak datu-baseko taula (erlazio) bihurtzen dituzte. Bihurketa hori, jakina, ez da nolanahikoa, eta hurbilpen desberdinak jarraitzen dira XML dokumentuak eskema erlazionaletara bihurtzeko garaian. Konparazio batera, (Florescu eta Kossmann, 1999) lanean XML dokumentuak gordetzeko eskema generikoa erabiltzen da, *erlazio hirutarrak* ("ternary relations") deiturikoa, grafoak adierazteko balio duena. (Shanmugasundaram *et al.*, 1999) lanean XML dokumentuen DTDak ezartzen ditu datu-baseko erlazioak. (Bohannon *et al.*, 2002) lanean, DTDa baino informazio semantiko aberatsagoa eskaintzen duen *XML Schema*-z adierazitako eskema-informaziotik abiatuz, datu-base moldakorrak sortzen dira.

Nabarmentzekoa da, gure ustean, datu-base sasi-egituratuek informazioaren integrazioarekin —hurrengo kapituluan aztertuko duguna— duten erlazio estua. Ez da harritzekoa datu sasi-egituratuak adierazteko estreinako datu-eredua, OEM delakoa, informazio-integrazioko proiektu baten barruan eraiki izana. Ikusiko dugun bezala, informazio-integrazioaren arloaren xede nagusia informazio oso heterogeneoa integratzea da, eta, hortaz, Interneten zehar gorderiko datuez baliatzea arloaren helburuen artean egon da hasiera-hasieratik.

Hurrengo kapituluan ikusiko dugun bezala, informazio-integrazioaren funtsezko kontzeptu bat galderaren barne-hartzearena da. Galderaren barne-hartzearen arazoa horrela azal daiteke: bi galdera izanik, jakin al daiteke galdera baten erantzuna bestearen azpimultzoa den? XML eta datu-baseak uztartzen dituzten proiektuetan ere arazo honekin topatuko dira. Izan ere, XML dokumentuak datu-base erlazionaletan gordeko badira, dokumentuen gainean eginiko galderak, nahiz eta jatorriz XMLko kontsulta-lengoaia baten arabera egon, datu-baseak ulertzen duen kontsulta-lengoaia batera itzuli beharko dira automatikoki. Itzulpen-prozesuak, baina, itzulpen anitz sortuko ditu, jeneralean, jatorrizko galdera baterako. Itzulpen multzo horretatik galdera

eraginkorrena aukeratu behar bada, galderen barne-hartzea kalkulatu behar da (Suciu, 2001).

II.2.3 Ezagutza-baseak.

Ezagutza-baseak adimen artifizialaren arloan jaio ziren, arloko aplikazioek eskatzen zituzten beharrei aurre egiteko. Informazio lexikala adierazteko oso erabiliak izan diren neurrian, atal honetan ezagutza-baseen azterketari ekingo diogu, azterketa hori oso sakona izango ez bada ere. Izan ere, ezagutza-baseen arloa adimen artifizialeko ikerlerro oso garrantzitsua baitugu, akaso garrantzitsuena, eta komunitate zientifikoan hainbat buruhauste eta liskar sortu ditu denboran zehar.

Adimen artifizialaren arloan adimenaren gaur egungo paradigmak, aurrekoaren aldean —heuristikoetan oinarritutako bilaketa-teknikek ezarria, batik bat— ezagutza jakitunaren beharra azpimarratzen du ataza adimentsuak burutu behar badira. LNPko arloak, jakina, paradigma-aldaketa berbera jaso du, eta, kapitulu honen hasieran aipatu den bezala, ezagutza linguistiko aberatsean oinarritzea ezinbesteko baldintzatzat hartzen da hizkuntza ulertuko bada.

Ezagutzan oinarritutako sistemak ezagutzaren errepresentazioarekin oso loturik daude. Izan ere, ezagutza, biltegiratzekoa bada, egituratu eta modelatu behar da, notazio egokiak erabiliz. Biltegi hauei *ezagutza-baseak* deitu ohi zaie, eta, ezagutza-baseen definizio formalik ez badago ere —batez ere datu-baseekin erkatzen baditugu—, ezagutza-baseek bi baldintza minimo bete beharko lituzkete:

- Arrazoibide-mekanismoez horniturik behar dute izan, hots, memento batean duen informazioa abiapuntutzat hartuz, ezagutza-baseek dedukzio logikoak egiteko aukera eman behar dute.
- Ezagutza-baseek gordetzen duten informazioari buruzko azalpenak emateko aukera izan behar dute.

Interesgarria deritzogu “ezagutzaz” ari garenean zer adierazten dugun azaltzeari, eta, horretarako, datu-base eta ezagutza-baseen arteko desberdintasunetan jarriko dugu arreta. Datu-base tradizionalen helburua datu kopuru handiak biltzea da, datu horiek eskema jakin baten arabera antolatuak izanik, eta datu horien eskurapen, aldaketa eta ezabapenak era eraginkorrean egiteko aukera ematea. Ezagutza-baseak datu-base sistema tradizionalen eboluzioa dira, datu kopuru handia adierazteko baino, *ezagutza-osagaiak*

adierazten baitituzte —normalean, egitate eta erregelen bidez adieraziak—, osagai horien erabilera nolakoa izan daitekeen azaltzeaz batera. Bestalde, ezagutza-baseek beren baitan gordetzen dutenari buruzko ezagutza ere behar dute, "dakitenari buruzko ezagutza", alegia.

Beren izenak salatzen duen legez, datu-baseek "datuak" metatuko lituzkete, eta ezagutza-baseek, aldiz, "ezagutza". Horrela, bada, datu-baseen ikuspuntutik ezagutza-osagaiak *Diskurtsoaren Unibertsoko* ("Universe of Discourse", UoD) osagaiak dira. Datuak, hortaz, UoDko egoera jakin baten gaineko asertzioak lirateke: "`ardo`aren hiperonimoa `edaria` da" UoD baten asertzio bat da, eta datu-egitura jakin batez adieraz daiteke. Ezagutza bezala ulertzen dugunera heltzeko, baina, abstrakzio-maila igo behar dugu, eta, horrela, datuei buruzko ezaugarriez gain, ezaugarri horien arteko erlazioak adierazteko ere aukera izan behar dugu. Horrela, bada, "`x,y` eta `z` guztietarako, `x`-ren hiperonimoa `y` bada, eta `y`-rena `z` bada, orduan `z` `x`-ren arbasoa da" ezagutza izango litzateke. Ezagutza mota hau[35] ez da, normalean, datu-baseen domeinukoa; bai, ordea, ezagutza-baseena.

Horrela dio Wiederhold-ek, datu-base eta ezagutza-baseen arteko desberdintasunak aztertzen dituen lanean (Wiederhold, 1984):

> A database is a collection of data representing facts (...) A knowledge base, as opposed to a database, contains information at a higher level of abstraction.

Esan daiteke, oro har, ezagutzak kontzeptu orokorren gaineko informazioa errepresentatzen duela; datuek, aldiz, entitate jakinari buruzko informazioa eskainiko digute. Ikuspuntu horretatik, datuek UoDaren une jakin bateko egoera adieraziko dute, eta, hortaz, oso aldakorrak izan daitezke. Datu horiek nola interpretatu eta erabili argitzen digun ezagutza, aitzitik, askoz ere egonkorragoa izango da. Ezagutza gordetzen duten sistemek notazio konplexu eta aberatsak beharko dituzte, eta, oro har, ez daude informazio kopuru handiak kudeatzeko prestatuak. Datu-baseak, berriz, oso egokiak dira datu kopuru handiak kudeatzeko. Wiederhold-en hitzak berriro geureganatuz (Wiederhold, 1986):

> Since data reflects the current state of the world at the level of instances, it will include much detail, will be voluminous, and will appear in reports wich are used at lower levels of enterprise

[35] *Metadatuak* ere deiturikoak.

> verification. Where instances change rapidly much data must be collected over time as well, if a complete historical picture is desired. Knowledge will not change as frequently. Knowledge may be complex but will deal with generalizations and hence refer to entity types rather than to entity instances.

Ikuspuntu kognitibo batetik, ezagutza-baseek, informazioa gordetzerakoan, gizakiok gordetzen dugunaren antzera egingo lukete. "Ezagutza" terminoak, horrela, informazioa erabiltzeko era posibleak ezartzen dituzten erregelak erreferentziatuko ditu, eta, baita ere, informazioaren eguneraketarako behar diren prozesu kognitiboak.

Sistema batek datuei buruzko orokortasunak adierazten duen informazioa badu, informazio hori era adimentsuan kudea badezake, sistemaren beharren arabera informazioa egunera badezake, eta dakienari buruzko azalpenak emateko aukera ere baldin badu, sistema hori ezagutza-basea izango da. Bestalde, sistemak datu kopuru handiak kudeatzen baditu, datu horiek Diskurtsoaren Unibertso batekoak izanik, eta datuen atzipena eta aldaketa egiteko era eraginkorrak eskaintzen baditu, sistema datu-basea izango da.

Datu-base eta ezagutza-baseen artean dauden aldeekin bukatzeko, esan behar da lehenengoak informazioaren biltegi pasibotzat har ditzakegula. Izan ere, datu-baseetan gorderiko datuen gaineko operazioak datu-basetik at geratzen baitira, aplikazio bezeroen mende, eta operazio horiek esplizituki adierazi ohi dira maiz. Ezagutza-baseak, aitzitik, sistema aktiboak dira izaeraz (Albano eta Attardi, 1989), disparadoreez ("trigger") hornituak baitaude. Hauen bidez, sistemak automatikoki egikarituko ditu zenbait erregela informazioa gehitzerakoan, ezabatzerakoan, edo dagoen informazioa eguneratzerakoan.

II.2.3.1 Ezagutzaren errepresentazioa.

Ezagutza errepresentatzeko formalismoen artean —batik bat hizkuntzaren errepresentazioari dagokionean— bi familia nagusi aurki daitezke: formalismo logikoak edo "semantika formalaren" aldekoak, eta "sareen" aldekoak.

Logika hasiera-hasieratik erabili da ezagutzaren egiturak errepresentatzeko. Logika notazio zehatza eta malgua da, eta informazioaren gainean prozesu deduktiboak gauzatu ahal izateko mekanismoez hornitua dago. Logikan oinarritutako errepresentazio-formalismoek orokortasun mota anitz adierazteko aukera ematen dute, nahiz egoera jakin bati buruzko informazio osoa ez

izan. Dedukzioaren bidez, bestalde, galdera logikoei erantzun dakieke, galdera zuzenean ebaluatu ezin denean ere.

Sistema logikoek giza-ezagutza modelatzekoak badira, baina, bi arazo nagusiri egin behar diete aurre. Alde batetik, logikan oinarritutako sistemek ez dute arrazoibide ez-monotonikoa taxuz kudeatzen, logika klasikoa monotonikoa baita[36]. Hala ere, giza-ezagutza —eta, bereziki, hizkuntzari buruzko ezagutza— ez da monotonikoa, eta, maiz, egitate bat aurretik dakiguna ezeztatzera dator. Adibidez, arrazoibidea monotonikoa bada, ezin da besterik ezeko herentziarik ("default inheritance") adierazi, eta, beraz, kontzeptuen arteko taxonomia aberatsak osatu. Izan ere, besterik ezeko herentzia ez-monotonikoa baita, izaeraz: goi-kontzeptu baten definizioa bere ume guztiei hedatuko zaie, baldin eta umeren batek kontrakorik adierazten ez badu —eta aurretik onartutako postulatu bat ezeztatu.

Beste alde batetik, logika klasikoen bidez —adibidez, predikatu-logika— bidera daitezkeen arrazoibide-mekanismoak orokorregiak dira, eta, horrela, inferentzia-mekanismoek prozesu amaiezinetara eramaten dute maiz. Gainera, logika klasikoaren bidez burututako arrazoibideak nolakoa izan behar duen ez dago nahi bezain finkatua —esaterako, inferentzia egiterakoan aplikatu behar diren erregelen ordena finkatzea—, eta, maiz, aplikazio zehatzek ezarria da. Dena dela, ezagutza adierazteko adierazpide oro logikaz baliatzen da bere baitan gordetako informazioaren koherentzia bermatzeko.

Sareetan oinarritutako sistemek, bestalde, kontzeptuak beren erlazioen arabera deskribatzen dituzte. Gizakion senak mundu errealeko objektuak kontzeptuetan sailkatzen dituela dirudi, eta, gertakari berri baten aurrean, kontzeptu berriak —edota ezagunak ditugun kontzeptuen arteko erlazio berriak— sortzen ditugula, jadanik ezagutzen ditugun zenbait kontzepturekin erlazionatuz. Horrela, ezagutzaren adierazpidea grafoetan oinarriturik dagoela pentsa dezakegu, zeinaren nodoak kontzeptuak diren, eta arkuak, berriz, kontzeptuen arteko erlazioak. Era berean, sareetan oinarritutako ezagutza adierazteko formalismoek sare antzeko egituretan antolatzen dituzte kontzeptuak. Eredu hauetan, kontzeptu batek sarean duen posizioak eta beste kontzeptuekiko erlazioek definituko dute kontzeptuaren "esanahia".

Izen desberdinekin —sare semantikoak, menpekotasun kontzeptualeko grafoak, herentzia-egiturazko sareak— bada ere, "grafoak" adimen artifizialeko hainbat eta hainbat sistematan erabili izan dira.

[36] Arrazoibide monotonikoa delakoan postulatu batek ezin du aurretik onartutako beste postulaturik ezeztatu.

Herentzia-egiturazko grafoek —ISA sareak ere deiturikoak— kontzeptu-hierarkiak eratzen dituzte, eta, bereziki, hierarkiaren arteko herentzia-erlazioa lantzen dute. Hierarkia hauetako nodoek kontzeptuak adierazten dituzte, eta arkuek, berriz, kontzeptuen arteko erlazioak. Arkuak, hortaz, etiketatuak agertzen dira, kontzeptuen arteko erlazioaren izenarekin. ISA sareetako nodoek bi interpretazio izan ohi dituzte —interpretazio *generikoa* edo *espezifikoa*—, nodoek indibiduo bakar bat edo indibiduo multzo bat adierazten duten (Brachman, 1983). Horrela, bada, hierarkiako maila baxueneko kontzeptuek —*token* deiturikoak— indibiduoak adierazten dituzte, eta goragoko mailetakoek —*type* deiturikoak—, berriz, indibiduo multzoak.

Grafo kontzeptualak (Sowa, 1984) ezagutzaren errepresentaziorako notazio bat dira, konputazionalki erabilgarriak. Grafo kontzeptualek sare semantikoen formalismoari zenbait gehigarri ekarri diote, lengoaia naturalaren aberastasuna eta konplexutasuna adierazteko era errazago eta malguagoak eskainiz. Konparazio batera, grafo kontzeptualek esanahi gabeko adierazpenak murritz ditzakete[37], *grafo kanonikoak* direlakoen bitartez, hautapen-murriztapen semantikoak adierazteko bidea eskainiz. Grafo kontzeptualak hizkuntz errepresentaziorako erabiltzen direnean, esaldi edo hitz bat kontzeptuz eta erlazio kontzeptualez (bi nodo mota) osatutako grafo orientatu batez errepresentatuko dira.

Sare semantikoen bilakaera kontatzen denean aipatzeke utzi ezin den formalismo bat KL-ONE (Brachman eta Schmoke, 1985) da. KL-ONE ezagutzaren errepresentaziorako lengoaia bezala definitzen da, edo, hobeto esanda, sistema bezala, inplementazioak gehigarrizko aukerak ematen baititu, hala nola, sareak gordetzeko aukerak, errepresentatu gertakariei buruzko galderak egitekoa, eta abar. Sare semantikoen eta, batez ere, *frame*-en tradiziotik etorritako lengoaia da funtsean, eta hierarkikoki egituratutako *kontzeptu generikoen* bitartez osatutako sareak dira ezagutza errepresentatzeko ematen duen bidea. Egitateak deskribatzeko bi egitura primitibo mota eskaintzen ditu: kontzeptuak eta *rolak*. Alderdi inferentzialetik begiratuta, berriz, funtsezkoa den kontzeptua ekarri zuen lengoaia honek: *sailkapen-mekanismoa* alegia, zeini esker deskribapen bat —kontzeptu berri bat— hierarkian dagokion lekuan kokatzea posiblea baita, berak subsumitzen duen ororen gainean, eta bera subsumitzen duten guztien azpian.

KL-ONE sistemak ereindako hazia jarraituz, ezagutza errepresentatzeko

[37] Adibidez, Chomsky-ren "Colorless green ideas sleep furiously" esanahi gabeko esaldi sonatua baztertzeko.

eta logiketan oinarritutako sistemen familia bat garatu da, deskribapen-logiken (DL) izenaren pean. Hurrengo kapituluaren III.5 atalean luze arituko gara DLez, eta beren gorabehera nagusiak aztertuko ditugu. Hala ere, esan dezakegu, orain, DL sistemen helburu nagusia bikoitza dela: alde batetik, hainbat eremutako ezagutza errepresentatzeko alderdi formalak eskaintzea eta, bestetik, ezagutza horren gainean burututako arrazoibideen erabakigarritasuna bermatzea. Horrela, DLetan lehen mailako logika baino espresibotasun eskasagokoak diren lengoaiak erabili ohi dira —semantika formala dute beren oinarri—, beti ere DLen bitartez adierazitakoaren gainean inferentziak egiteko arrazoibide-prozesuen konputagarritasuna oso kontuan hartzen delarik.

DLak ezagutza orokorra errepresentatzeko erabili dira, eta, bereziki, datubaseen eskemen ezagutzan oinarritutako adierazpideak lortzeko, edo informazioaren integrazioa gauzatzeko. Izan ere, DLek informazio partzialarekin lan egin baitezakete, eta hori informazioaren integrazioan behar-beharrezkoa den ezaugarri bat da. KL-ONE sistema informazio linguistikoa errepresentatzeko erabili den bezala, DLak ere helburu berarekin erabili dira. Franconi-ren lanean (1994) interpretazio semantikoa adierazteko DLak erabiltzen dituzten hainbat proiektu ikus daitezke. Autorearen esanetan, baina, hauen erabilpena nolabait murriztu egin da azken urteotan, batez ere, sistema estatistikoen arrakasta dela eta, zeintzuek azaleko semantikoa besterik errepresentatzen ez duten.

II.2.3.2 Ezagutza lexikalaren errepresentazioa.

Atal honetan ezagutza lexikalean jarriko dugu arreta, eta ezagutza mota hau errepresentatzeko erabili ohi diren formalismoak gainbegiratuko ditugu.

Azken hamarkadan *baterakuntza* kontzeptua oinarri duten formalismo gramatikalen erabiltzeko joera egon da. Linguistika konputazionalean baterakuntzaren erabilpenaz (Kay, 1979) mintzo da estreinakoz, eta ondoren hedapen linguistiko eta konputazionalak oso nabariak izan dira. Baterakuntza kontzeptuan oinarritutako formalismo gramatikal ugari garatu da, hala nola, LFG ("Lexical Functional Grammar"), HPSG ("Head-Driven Phrase Structure Grammar"), CUG ("Categorial Unification Grammar") edo FUG ("Functional Unification Grammar").

Kapituluaren hasieran aipatu dugu lexikoaren paperaren garrantzia arras handitu dela LNPan azken urte luzeetan, eta, ondorioz, gramatiketan islatu ohi zen informazio zati handi bat lexikora lerratu dela. Horrela, lexikoaren

errepresentazioa oso lotuta agertu ohi da gramatiketara, gramatikek ezagutza lexikalarekin burutuko dituzten prozesuak oso kontuan hartu baitira ezagutza lexikala errepresentatzerakoan.

Formalismo hauek *deklaratiboak* eta *murriztapenetan oinarrituak* izan beharko luketela azpimarratu da askotan: horrela, hizkuntzaren deskribapenaren eta hizkuntzaren konputagailuen bidezko tratamenduaren inguruko hainbat arazo tekniko ebatz daitezke (Shieber, 1992).

Formalismoa deklaratiboa bada, gramatikaren azterketa ezagutzaren errepresentazioaren adar bat bezala ikus daiteke: gramatika, horrela, hizkuntza bati buruz ezagun diren zenbait gertaera adierazteko formalismoa da, makina batek ulertzeko bezain esplizitua (Gazdar eta Mellish, 1989). Testuinguru honetan, hizkuntza bat erregelen bitartez zehazki espezifika daitekeen multzo bat besterik ez da. Horrela, bada, gramatika sistema matematikoa da, zenbait multzo infinitu hizkuntza bateko osagaiak direnetz zehazten duena.

Mota honetako gramatikek esanahiaren ikuspuntu konposizionala ematen dute, beren bidez onartutako espresioek esanahi zehatza baitute, osatzen dituzten azpi-espresioen esanahien arabera sorturikoa. Hitz baten esapidearen "esanahia" jakiteko, bada, bere informazio sintaktikotik abiatuko da gramatika.

Sistema hauen beste propietate bat ezaugarrietan oinarrituak izatearena da. Propietate honek arras desberdintzen ditu formalismo hauek eta LNPko hastapeneko gramatikak, hots, testuingururik gabeko gramatikak. Izan ere, testuingururik gabeko gramatikek bi arazo nagusi baitzituzten lengoaia naturalarekin lan egiterakoan: gainsorkuntza (analisi-zuhaitz gehiegi sortzea, alegia), eta arbitrariotasuna (erregelen aplikazioa ezin baita murriztu). Ezaugarrien bidez lortuko da, besteak beste, garai batean gramatika-erregeletan islatutako informazio kopuru handia maila lexikora lerratzea, hots, objektu lexikalekin batera gordetzea. Halaber, gramatiketan informazio semantikoa kudeatzeko bidea ematen dute, informazio hori objektu lexikalei eransten ahal baitzaie, ezaugarri berrien bidez.

Informazio linguistiko orokorra kodetzeko usuen erabilitako formalismoen artean *Ezaugarri-egiturak* (EE, "feature structures") ditugu zalantzarik gabe. Ezaugarri-egiturak deklaratiboak dira, eta objektu linguistikoak adierazteko metodologia eraginkor eta zehatza eskaintzen dute. Gainera, interpretazio semantiko zehatza eta formalizatua dute, eta beren bidez hiztegietan edo LNPrako datu-base lexikaletako informazioa era naturalean kode daiteke (Ide *et al.*, 1994).

Informalki, ezaugarri-egitura bat atributu-balio bikoteez osaturiko mul-

tzo bat bezala ikus daiteke, non balioak atomikoak izan daitezkeen —hitz-kateak, zenbakiak eta abar—, edota beste ezaugarri-egitura habiatuak. Horretaz gain, bi ezaugarri edo gehiagok balio bera erreferentzia dezakete. EE-ek, hortaz, grafo azikliko eta zuzenduen (DAG) tankerako adierazpena dute.

EE-ek bi eragile onartzen dituzte, hots, *subsuntzioa* eta *baterakuntza*. Lehenengoak, ohi bezala, EE bat bestea baino orokorragoa denetz erabakitzen du. Bigarrenak, ostera, sarrera bezala bi EE onartu, eta emaitza gisa EE berri bat sortuko du, alegia, jatorrizko EE-ek biek sumsumitzen duten EE-en arteko orokorrena. Gerta daiteke, jakina, bi EE-en arteko baterakuntzarik ez osatzea, jatorrizko EE-ek informazio kontrajarria adierazten dutelako.

Ezaugarri-egituren formalismoa ez dago teoria linguistiko zehatzekin lotuta, eta, hortaz, hurbilpen mota anitz adierazteko bezain malgua da. Haatik, formalismoak ez du kasik murriztapenik ezartzen sarrerak idazterakoan: unitate lexikal batek ia edozein informazio bil dezake bere baitan. Gauzak horrela, ezaugarri-egituratu motatuen (EEM) formalismoa garatu zen (Carpenter, 1991; Zajac, 1992), zeinaren bitartez mota-sistema bat erazagut daitekeen: mota-sistema lagun adieraz daiteke, esaterako, unitate lexikalak zehazki hiru osagaiz —fonologia, sintaxia eta semantika — osaturik daudela, eta, nahi izanez gero, osagaien balio posibleak murriztu. Mota-sistemak moten hierarkia ezartzen du, informazio lexikalaren izaera hierarkikoa islatzeko aukera emanez; adibidez, hitz sorta batek komun duen informazioa faktorizatzeko aukera emanez, informazio hori toki bakar batean adieraz baitaiteke.

EEak oinarrizko datu-egitura dituzten hainbat formalismo garatu da. PATR (Schieber *et al.*, 1983; Shieber, 1986) izeneko formalismo ezagunak, esaterako, EEak erabiltzen ditu, eta testuingururik gabeko gramatika batean integratzen, gramatika hauek duten arazo larrienetako bat —gainsorkuntza, alegia— saihesteko. Ez ditu, hala ere, EEMak onartzen; horren ordez *kategoriak* izeneko nozioa ezartzen du: PATRko elementu orok `cat` izeneko ezaugarri bat du, ezinbestean, elementuaren kategoriaren adierazle gisa. Hierarkia tankerako egitura batean atributuen balioak faktorizatzeko —kontzeptu baten umeen artean heda daitezen—, PATRk makro antzeko egiturak erabiltzen ditu. Horrela, PATR-ren herentzia monotonikoa da, hots, ez du besterik ezeko herentziarik onartzen.

DATR formalismoa, bestetik, ezagutza lexikala adierazteko lengoaia dugu, murriztapenetan oinarritua (Evans eta Gazdar, 1996). Ezagutza lexikalaren errepresentazioaren arloan zenbait aitzindarik osatua, DATR ezagutza lexikala adierazteko formalismo neutrala da. Alegia, baterakuntza-gramatika orok erabil dezake DATR formalismoa lexikoa errepresentatzeko, gramatikak

jarraitzen duen teoria linguistikoa edozein dela ere. PATR formalismoarekin alderatuz —zeinek lotura handia ezartzen duen errepresentatzen denaren eta inferentziak gauzatzeko jarraitutako prozesuaren artean—, DATR deklaratiboa da erabat, eta inferentzia-eragiketa bakarra du, hau da, herentzia ezmonotonikoaren bidezko inferentzia. Horrela, DATRk besterik ezeko herentzia onartzen du objektuen arteko hierarkiak eratzerakoan, unitate lexikalen ezaugarriak definitzeko era trinko bezain aberatsa eskainiz. Hala ere, ez ditu EE-ek onartutako oinarrizko eragileak —baterakuntza eta subsuntzioa— inplementatzen, eta, hori horrela, ezin dira objektu egituratu konplexuak taxuz adierazi.

ALE sistemak (Carpenter eta Penn, 1997), berriz, EEMak inplementatzen ditu. Baterakuntza- eta subsuntzio-eragileez gain, besterik gabeko herentziaren eragilea ere inplementatzen du, eta, horrela, sistemak aukera ematen du ezaugarrien mota-hierarkiak eratzeko. ALEren mota-sistema zorrotza da: EE orok gutxienez mota batekoa izan behar du, eta mota bakoitzaren definizioan ezarri behar da: i) zein ezaugarrirentzat defini daitekeen, eta ii) ezaugarri horien balioek zein motatakoak izan behar duten. ALE ez da PATR edo DATR bezain "neutrala", formalismo bati hertsiki loturik baitago, hots, HPSGri.

II.3 Baliabide lexikalen sailkapen modukoa eta zenbait adibide.

Azpiatal honetan, LNPan erabiltzen diren baliabide lexikaletan jarriko dugu arreta. Baliabide lexikal ugari dago, jakina, hizkuntza desberdinetan LNPan edota hiztegigintzan aritutako hamaika taldek eta erakundek eratuak. Guztiak aztertzeak oso luze joko liguke, hortaz, eta horien guztien deskribapen sakona egitea tesi-txosten honen helburuetatik at dago.

Gure azterketa, beraz, ez da exhaustiboa. Hori baino, LNPrako erabilgarri diren baliabideen tipologia antzeko bat aurkeztu nahi dugu, adibide batzuen bidez. Hori horrela, hiru multzo handitan sailkatu ditugu aztergai izango diren sistemak:

- Hiztegiak.
- Datu-base lexikalak.
- Ezagutza-base lexikalak.

Egindako sailkapena ez da, jakina, baliabide lexikalena bezalako arlo zabal batean egin daitekeen bakarra. Izan ere, irizpide anitz jarrai baitaitezke horrelako sailkapenak egiterakoan, motibazio desberdinek eraginda. Konparazio batera, zenbait lanek guk ezagutza-base gisa multzokatu ditugun baliabideak bi azpi-multzotan sailkatu ohi dituzte: ezagutza-baseak eta ontologiak. Ontologiak ezagutza-baseen goi aldeko kontzeptuak lirateke, zeren, espero daitekeen bezala, taxonomia semantikoen goi-mailako kontzeptuak oso orokorrak izango baitira, eta, horrela, mundu errealaren kontzeptualizazio nagusia eskainiko baitigute, hizkuntzatik independentea.

Edonola ere den, orain ekingo diogun azterketa hau egungo baliabide lexikal garrantzitsuenen lagin adierazgarria delakoan gaude. Azterketari ekiteko, lehenik eta behin sail bakoitza ilustratuko duen adibide bat edo beste aurkeztuko dugu, esan bezala, exaustiboak izateko inongo asmorik gabe. Bestalde, azpimarratuko ditugu klase bakoitzako baliabideak gure sisteman sartzeko motibazioak, ELHISA diseinatzean klase honetako guztietako iturriak hartu baitira aintzat.

II.3.1 Hiztegiak.

Hiztegiak lehen mailako baliabide lexikalak dira, bertan jasotzen baitira hizkuntza bateko —edo gehiagotako, hiztegi eleanitzak kasu— hitzen definizioak. Tradizio lexikografiko zabal eta aberatsean oinarrituta, hiztegietan gordetako informazioa lan handiaren emaitza da. Sarrerak adieretan banatuak daude, adiera bakoitzak sarreraren esanahi berezi bat adierazten duelarik. Horretaz gain, hiztegiek azpisarrerak ere gorde ohi dituzte: hitz anitzeko unitateak, esanahi berezia dutenak, gehienetan.

Kapitulu honen hasieran —II.1 atalean— aztertu ditugu, jada, hiztegiak LNPrako iturri lexikalak izateko garatu diren hainbat proiektu desberdin. Lan horietan, hiztegietako informazioaz baliatzea proposatzen da LNPrako baliabide lexikalak eraikitzeko. Ikusi dugu, hala ere, prozesu honek hainbat zailtasun eta buruhausteri aurre egin behar diola emaitza onargarriak eman behar baditu.

(Levin, 1991)-en onartzen den bezala, hiztegiak gizakiak sortuak eta gizakientzat zuzendutako baliabide lexikalak dira. Giza-erabiltzaileak hizkuntza baten lexikoaren egitura inplizitua ezagutzen du, eta lexikografoek, hiztegia sortzeko garaian, erabiltzaileen ezagutza hori kontuan hartzen dute. Horrela, bada, hiztegietako sarreretan aurki daitekeen hizkuntzari buruzko informazioa zuzenduta dago. Izan ere, aldez aurretik ezagupen linguistiko orokorra

duen giza-erabiltzaileak erabiliko du, eta hiztegi batean aurkituko duen informazioa soilik hizkuntza jakin bat ulertzeko behar duena izango da. Horrela, bada, giza-erabiltzailea bere ezagumendu linguistikoaz eta inteligentziaz baliatzen da hiztegi arrunt gehienetako informazioaz jabetzeko.

ELHISAk bere baitan hiztegien informazioa izan behar duela uste dugu, bertan gordetako informazioa oso baliagarria baita. Izan ere, ELHISAren xede nagusia ez baita soilik LNPrako hornitzaile lexikala izatea. Aitzitik, erabiltzaileei informazio lexikal zabala eta aberatsa eskainiko die ELHISAk, baliabide anitzetik jasoa. Hiztegiak gure sisteman integratutakoan, erabiltzaileak hitz bati buruz galdetu, eta hainbat hiztegitako informazioa jasoko du, era bateratu batean.

Internet sarean aurki daitezkeen *on-line* hiztegiak atzitzea ere oso baliagarria da. Gaur egun mota honetako informazio-iturrien kopuruaren gehikuntza kontuan harturik, baliabide horiek gure sisteman integratzeari oso interesgarria deritzogu.

II.3.1.1 EH hiztegia.

Euskal Hiztegia (EH, Sarasola 1996) euskarazko hiztegia da. 33.111 sarrera eta 41.699 adierez hornitua, erabilera zabaleko hiztegi elebakarra dugu EH, batik bat hitzen erabilpen tradizionala —sarreren estreinako agerpen-data, literaturan sarrera bati buruz agertutako adibide esanguratsuak, eta abar— jasotzen duena.

Sarreretan informazio nahikoa aberatsa gordetzen da EHn: forma kanonikoa (lema), kategoria, azpikategoria, hiztegiratze-data, erabilera-eremuak (geografikoa, erabilera zaharkituak edo arkaikoak, eta abar), definizioa, adibideak, sinonimoak eta antonimoak, eta abar.

EH hiztegia, jatorrian, testu-prozesadore baten laguntzaz eraiki zen. Formatu elektronikoan egon da, hortaz, EH hiztegia hasiera-hasieratik. Testu-prozesadoreko fitxategiak, ordea, ez dira batere egokiak hiztegiko informazioa ustiatu nahi bada[38]. Horrela, bada, jatorrizko formatutik konputagailuak uler dezakeen beste formatu batera pasa da EH hiztegia, XML formatura, alegia, TEI ekimenak hiztegiak kodetzeko proposatzen dituen hiztegi-elementuak erabiliz (Arregi *et al.*, 2003).

EH hiztegian, tradizio handiko hitzak jasotzen badira ere, gabeziak erakusten ditu hitz *modernoak* —azken urteetan sortutakoak, terminologia bere-

[38]Izan ere, testu-prozesadorearen formatua markaketa prozeduralaren pean baitago. Ikus II.2.1 atala.

zia batik bat— islatzerakoan. Bestalde, EH hiztegia ez dator guztiz bat Euskaltzaindia erakundeak emandako hitz estandarren zehaztapenarekin, berak argituratutako *Hiztegi Batua* (HB) arauemailearekin, alegia: EHk zenbait sarrerarentzat forma jakin bat hobesten duen bitartean, HBk beste forma desberdin bat hobesten du askotan[39].

II.3.2 Datu-base lexikalak.

Datu-base lexikalak, hiztegiak ez bezala, LNPko aplikazioen iturri lexikal nagusia izateko helburuarekin sortu ohi dira. Hainbat aplikazioren *hornitzaile lexikalak* izateko asmoarekin eraikiak dira, eta, horrela, bertan gordetako informazioa hizkuntzaren tratamenduari begira zuzenduta dago.

Honako ezaugarri hauek aurkituko dira datu-base hauetan, besteak beste:

- **Tamaina handia**. Normalean, datu-base lexikalak hizkuntza bateko sarrera lexikal asko gordeko ditu. Esate baterako, datu-base lexikala aditzen informazioa gordetzeko eraikia izan bada, hizkuntza horretako aditz *guztien* informazioa gordeko du seguru aski. Eremu zabalekoa bada, berriz, ez da harritzekoa hizkuntzaren hitz-forma guztiak bere baitan gordetzea.

- **Informazio zehatza**. Arestian aipatu dugun bezala, hiztegietan gordetako informazioa ez da nahi bezain zehatza, LNPan erabiltzeko behintzat. Askotan, hiztegia eraikitzean erabilitako metodologia ez-sistematikoa dela medio, datu akastunak edo ez bateratuak aurki daitezke bertan. Konparazio batera, zenbait eremu ez daude normalizatuak[40], eta zenbait eremutan informazio habiatua aurki daiteke. Datu-base lexikaletan, aldiz, informazioa nekez izaten da akastuna edo normalizatu gabea: LNPrako aplikazioen iturri lexikal nagusia izango badira, oso informazio zehatza eta zuzena gordeko dute oro har.

- **Informazio berezitua**. Askotan, datu-base lexikaletan metatutako informazioa oso lotua dago datu-base horrek hornitu behar dituen aplikazioei. Adibidez, datu-basea analizatzaile morfologiko baten iturri le-

[39]Hala ere, denbora kontua dira EH eta HB hiztegien arteko bat ez etortzeak. Izan ere, EH argitaratua izan zenetik HB egiten joan da, eta oraindik ez da bukatu.

[40]EH hiztegian, adibidez, forma desberdinak erabiltzen dira kategoria lexikal bera adierazteko. Horrela, “iz.”, “ize” edo “iz” balio desberdinak aurki ditzakegu “izena” kategoria adierazteko.

xikala bada, oso ohikoa da hitz-formei *diakritikoak* eranstea, erregela morfotaktiko eta morfofonetikoen bidez hitz horretatik sor daitezkeen forma posible guztiak sortu edo ezagutu ahal izateko. Informazio hori, gainera, oso lotuta egon ohi da erabili den teoria linguistiko bereziarekin[41]. Beste hainbeste gertatuko da gainerako maila linguistikoetan ere —sintaktikoa, semantikoa eta abar.

- **Biltegiratze eraginkorra**. Beren izenak salatzen duen legez, datu-base lexikalak datu-baseak dira izaeraz, eta, hortaz, beren baitan gordetako informazioa era eraginkorrean eskuratzeko aukera eman ohi dute. Horrela, baliabide hauek biltegiratze-sistema finko eta estandarra erabili ohi dute datuak gordetzeko (adibidez, objektuei zuzendutako datu-baseak edo datu-base erlazionalak).

ELHISAk gai izan behar du datu-base lexikalen informazioa bere baitan hartzeko, LNPrako iturri lexikala izateko sortu ez bada ere, datu-base lexikaletan jasotzen den informazio zabal eta zehatzaz baliatzea oso garrantzitsua delakoan baikaude.

II.3.2.1 EDBL.

Euskarazko Datu-Base Lexikala (EDBL) euskararen tratamendu automatikorako —hala nola, ortografiaren egiaztapena eta zuzenketa, analisi morfologikoa, lematizazioa edo sintaxiaren tratamendua— iturri lexikal nagusia da. IXA taldearen barruan eta batik bat eskuz garatua, honako ezaugarri hauek bideratu dituzte EDBLren diseinu eta osaera:

- **Helburu anitzekoa**, analisi morfologiko zein beste mailetako egiteko guztietarako oinarri lexikal egokia.

- **Neutrala**, bertan egindako deskribapen linguistikoak etorkizuneko aplikazioak ez baititu baldintzatzen.

- **Malgua eta irekia**, edozein unetan helburu berrietarako egokitzen erraza, alegia.

[41] Adibidez, EDBL datu-baseak hainbat informazio gordetzen du morfotaktika kudeatzeko, eta informazio hori erabat lotua dago bi mailatako morfologia delako formalismoarekin (Aldezabal *et al.*, 2001)

- **Erabilerraza**, erabiltzaileei behar bezalako laguntzak eta erraztasunak eskaintzen dizkiona.

Horretaz gain, EDBL Euskaltzaindiak lexiko kontuetan ematen duen arautegiaren —hiztegi batua, batez ere— gordailu eguneratua da, unitate lexikalen forma estandar onartuaren berri ematen baitu.

EDBL datu-base erlazional batean dago gordeta, nahiz eta bere eskema adierazteko datu-eredu ahaltsuagoak erabili izan diren. Horrela, bada, datubaseko eskemak objektuei zuzendutako baten tankera osoa du, ezaugarri-egitura motatuen bidez eginiko mota-sistema bati esker.

II.3.3 Ezagutza-base lexikalak.

Amsler eta Walker egileek aipatzen dute ezagutza-base lexikalaren kontzeptua estreinako aldiz 1981-1982 tartean, hiztegietako definizioen erlazio lexiko-semantikoak ustiatzen hasi zirenean. Izan ere, lengoaia naturalen prozesamendu sintaktiko eta semantikoa egin ahal izateko, lexikoiak hitz-zerrenda izatetik ezagutza-base lexikala izatera pasatu dira, hitz eta adierei buruzko informazioa duten ezagutza-base konplexuetara, alegia. Ezagutza-base lexikalen ezaugarri garrantzitsuena herentzia izaten da, adierak klase/azpiklase hierarkien inguruan antolatzen dira eta (Copestake, 1990).

Ezagutza-base lexikalak eratzeko bi hurbilpen jarraitu ohi dira: eskuz edo erdi-automatikoki, hiztegietan oinarrituz. Ezagutza-base lexikalak erdi-automatikoki eratzeko, hiztegietatik erauzi izan den informazio semantikoa definizioen azterketatik etorri ohi da batez ere, adieren hierarkia osatuz, eta hitzen (edo adieren) arteko bestelako erlazio lexiko-semantikoak finkatuz[42].

Bestalde, ezagutza-base lexikalek, beren baitan duten informazioa adierazteko, ezagutzan oinarritutako teknikak erabiltzen dituzte, dela ezaugarri-egitura motatuak, PATR sarrerak edo *framee*-tan oinarrituak. Gainera, informazioa eskuratzeko kontsulta-lengoaia ahaltsuak onartzen dituzte, informazio partziala kudea dezaketenak, eta dedukzioak egiteko aukera ematen dutenak.

ELHISAk aukera eman behar du ezagutza-baseetan gordetako informazioa eskuratzeko. Horrela, hitzen arteko erlazio lexiko-semantikoak adierazteko bidea emango du, eta halaber, hitzen edo adieren arteko taxonomiak

[42]Hala-nola, sinonimia/antonimia erlazioak, erlazio meronimikoak, holonimikoak eta abar.

islatzeko gauza izango da. Ezagutza-baseak ELHISAren baitan onartzean, sistema osoa informazio semantiko aberatsez hornituko dugu.

II.3.3.1 EDR.

Japoniako ikerkuntza-agentziak, itzulpen automatikorako lexikoaren garapenak zeukan garrantzia ikusita, *Japan Electronic Dictionary Research Institute* sortu zuen 1986 urtean, japoniera eta ingelesaren tratamendu automatikorako lexikoa eraiki zezaten (Yokoi, 1995). Proiektu handi honek 9 urtetan bere emaitzak eman zituen, ia 300.000 hitz dituen lexikoi elebiduna sortuz. Horretaz gain, lexikoi elebidunak eta 4.000.000 kontzeptu biltzen dituen ezagutza-base lexikala ere sortu zuten. Kontzeptuak biltzen dituen ezagutza-baseak kontzeptuen deskribapenak eta kontzeptuen arteko erlazio lexikal asko biltzen ditu, hierarkia osatzen duen azpiklase erlazioa garrantzitsuena izanik.

EDR hainbat hiztegiz edo datu-basez dago osatua: hitz-biltegiak, kontzeptu-biltegiak, agerkidetza-hiztegiak eta hiztegi elebidunak (ingelesa eta japoniera).

II.3.3.2 WordNet eta EuskalWordNet.

WordNet (Miller, 1990) sinonimiaren inguruan antolatutako ingelesezko ezagutza-basea da, hitz-adierez sortutako sarea. Sinonimo multzo bakoitza, *synset* deiturikoa, hitz-adierez osatua dago, eta kontzeptu bat errepresentatzen du. WordNet-eko synset-en artean erlazio lexikal ugari daude, haien artean hiperonimia eta hiponimia landuenak izanik. Synset-ak hierarkietan antolatzen dira, baina multzo semantiko nagusietan ere multzokatuak daude[43]. Kontzeptu kopuruari dagokionez, WordNet 2.0 bertsioan orotara 115.000 kontzeptu daude 152.000 hitzentzat.

WordNet edonork eskura dezake Interneten bidez, eta oso erabilia da LNP inguruko ikerkuntzan azken orteotan.

EuroWordNet (Vossen, 1997) proiektua 1996an hasi eta 1999raino luzatu zen proiektu europarra da. WordNet-en diseinuaren antzekoa erabiltzen du, baina Europako zortzi hizkuntzatara zabaltzen da. WordNet-en baino hizkuntza barneko erlazio mota gehiago daude, baina, hemen ere, hiperonimia eta hiponimia erlazioak daude landuenak, sinonimiaz gain, noski. Hizkuntzaren barne-erlazioez gain, kontzeptuak WordNet-eko synset-etara lotuta daude, *Inter-Lingual Index* delakoaren bidez, hizkuntza arteko ordainak adieraziz.

[43] Adibidez, izenen kasuan 15 eremu semantiko bereizten dira.

Horretaz gain, hizkuntzatik aparteko moduluan Goi-ontologia bat (*top ontology*) eta Domeinu-ontologiak (*domain ontology*) ere badaude. Lehenbizikoak wordnet ezberdinen goi aldeko synset-ak ezaugarri semantikoen arabera sailkatzeko aukera ematen du, eta WordNet-eko eremu semantikoaren papera jokatzen du, nahiz eta motibazio linguistiko sakonagoak hartu diren kontuan.

Donostiako Informatika Fakultateko Lengoaia Naturalaren Prozesamendurako IXA Taldea EuroWordNet proiektura lotua zegoen, kanpoko eraikitzaile bezala. Horren inguruan, EuskalWordNet, euskarazko wordnet-a, eraikitzen ari gara (Agirre *et al.*, 2002), EuroWordNet-en diseinua jarraituz.

II.3.3.3 Hiztsua.

Hiztsua (Artola, 1993) *Le Plus Petit Larouse* (LPPL) hiztegitik erauzitako Hiztegi Sistema Urgazle Adimentsua da. Bere funtzionalitatearen oinarrian automatikoki sortutako ezagutza-base lexikal aberats bat dago. Hiztegiko definizioen errepresentaziorako eredu konplexu bat proposatzen du, zeinen gainean inferentziak egin ahal diren. Hiperonimia/hiponimia erlazioen inguruan antolatua dago batez ere, baina beste hainbat erlazio ere erauzi ziren hiztegitik, izen, aditz eta adjektiboentzat. LPPLko adiera guztietatik 6.130 adiera sartu ziren ezagutza-basean, nahiz eta zenbait definizio osorik ez analizatu.

III. KAPITULUA

Datu-integrazioa.

Informazio-sistemetan gorderiko datuen integrazioa ikerlerro mamitsua bilakatu da azkenengo urteetan. Laurogeiko hamarkadan ikerlerro honen helburua datuak datu-base anitzetan zehar banatzea bazen ere –datu-base hauek bereziak eta inkonpatibleak izanik–, 90eko hamarkadako xedea DBKSen arteko datuen elkar-trukaketa izan da. Heimbigner eta McLeod-en lana (1985) datu-base anizkoitzen sistemen inguruko estreinetarikoa dugu. Gerora, datu-base federatuaren terminoa azaldu zen, datu-base autonomo, heterogeneo eta banatuen artean atzipen integratua eskaintzearen arazoari eusteko nahian (Litwin *et al.*, 1990; Seth eta Larson, 1990).

Edonola ere, datu-integrazioaren beharrak asko aldatu dira hasieratik hona. Izan ere, hastapeneko datu-integrazioak zituen zenbait arazo tekniko ia desagertu egin dira egun. Konparazio batera, CORBA edo Java bezalako tresna estandarrek konponbide dotorea eskaintzen diote duela ez hainbeste urte arazo larria zen banaketa fisikoari. Bestalde, gaur egun askoz errazagoa da makinak elkarrekin konektatzea, Internet sarearen arrakasta dela eta. Hastapenetik hona —non, askotan, makinen artean sarerik ez zegoen, edo, egonda ere, sareen artean protokolo desberdinak erabiltzen ziren— egoera franko aldatu da, beraz.

Halaber, egun atzi daitezkeen informazio-iturrien kopurua izugarri handitu da, *World Wide Web* (WWW) deituriko Interneteko amarauneko orriei esker. Kopuru handi honek, amaraunak datu-atzipenerako eskaintzen duen ahalmen eskasarekin batera, ikuspuntu-aldaketak sortu ditu informazio-integrazioaren ikerlerroan. Konparazio batera, eskemaren integrazioak ez du

zentzurik izango, datuak eskemen bitartez definitzen ez badira[1]; halaber, amarauneko datu-iturriak sistema integratu batean biltzerakoan, iturri honek ez du jakingo "integratua" izan denik ere. Horrela, iturrien autonomiaren arazoa zeharo aldatzen da, Internet aurreko garaiekin alderatuz.

Kapitulu honetan informazioaren integraziorako hainbat teknika ikusiko ditugu. Lehenik eta behin, arloaren nondik norakoak azalduko ditugu. Gero, eta arloa bere testuinguruan kokatzeko asmoz, datu-federazioez eta datu-biltegiez arituko gara. Informazioaren integrazioak adimen artifizialaren arloarekin dituen erlazio estuei erreparatuko diegu segidan. Gero, datu-integrazioaz arituko gara: hainbat kontzeptu garrantzitsu definituko ditugu lehen, eta galderen itzulpenaren prozesuan jarriko dugu arreta jarraian, auzi horretarako zenbait algoritmo azalduz. Galdeketa-gaitasunei, *wrapper*-ei eta datu arazketari ere tartea egingo diegu. Jarraian, deskribapen-logikez arituko gara, eta gure sistemaren inplementazioan erabili dugun *NeoClassic* sistema aurkeztuko dugu. Bukatzeko, datu-integrazioa helburu duten hainbat proiektu ikusiko ditugu.

III.1 Informazioaren integrazioaren nondik norakoak.

Atal honetan, informazioaren integrazioaren arloa aztertuko dugu, eta, horretarako, arloak berak dituen azpi-arlo desberdinak azalduko ditugu laburki. Jo beza irakurleak (Jarke *et al.*, 2000) lanera, sailkapen honi buruzko informazio osagarriaren bila.

Iturburu-integrazioa ("Source integration"; Jarke *et al.*, 2000) delako arloaren helburua hainbat iturri (datu-base) entitate komun batean integratzean datza. Termino hau prozesu orokorrago baten barruan kokatu ohi da, non, iturburuaren integrazioaren prozesuari, agregazioa eta *prozesu analitiko online* delakoa ("online analitic process", OALP) bezalako atazak gehitzen zaizkion. Iturburu-integrazioaren bi mota dago: *eskemaren integrazioa* ("schema integration") eta *datu-integrazioa* ("data integration").

Eskemaren integrazioa (Batini *et al.*, 1986; Seth *et al.*, 1993) softwarearen edo ezagutzaren ingeniaritzaren arloan kokatzen da; bere helburu nagusia

[1] *Web*-ean aurkitutako datuak HTML edo XML lengoaiez kodeturik agertzen dira, eta ez dira, orokorrean, datu-eskema baten pean definitzen, datu-base tradizionaletan ez bezala. Horrela, datu hauek sasi-egituratuak direla esaten da, nahiz eta, XML lengoaian, datu horien *sintaxia* definitzen den, DTD edo eskema-lengoaiaren baten bitartez. Ikus II.2.2.1 atala.

hainbat informazio-sistemaren artean eskema "integratu" bakar bat lortzea da, informazio-sistemen gainean alderantzizko ingeniaritza-teknikak zein eskemaren gaineko berringeniaritza-teknikak erabiliz. Eskemaren integrazioa beharrezkoa da datu-baseen diseinua garatzerakoan, eta, oro har, informazio-iturburuaren intentsio-mailako deskribapenak konparatu eta integratu nahi direnean.

Datu-integrazioaren helburua, bestalde, informazio-sistemak elkarreragingarritasunaz hornitzean datza, sistemen arteko heterogeneotasuna ebatziz, beti ere datu-mailan. Beraz, iturburuen eskemak konparatu eta integratzeaz gain, iturri hauek fisikoki gordetzen dituzten datuen integrazioaren arazoa ere aztertuko da, hala nola, *objektuen identifikazioa* ("object matching") delako arazoa, hots, bi iturburu edo gehiagotatik jasotako objektu desberdinek munduko elementu bera erreferentziatzen dutenetz ebaztearen arazoa. Gure tesi-lan honekin zerikusi handiena duen integrazioa da "datu-integrazioa" delako hau.[2]

Datu-integrazioaren arazoa informazio-sistema kooperatiboen ("cooperative information systems") arlotik bereizi ohi da. Izan ere, azken arlo honetan, datuen integrazioaz gain, lan-fluxua eta enpresaren prozesuak edo hornidurakateak bezalako kontzeptu aurreratuagoak ere kontuan hartzen dira, eta azpisistemen arteko koordinazio eta elkarrekintzak ikertzen dira. Arazo hauek, ordea, datu soilak integratzeko behar diren tekniketatik urruntzen dira.

Bi azpiarlo nagusitan deskonposa daiteke datu-integrazioaren arazoa: *integrazio estrukturala*[3], egiturari buruzko heterogeneotasunean datzana, hala nola, datu-eredu, kontsulta, datu-atzipenerako lengoaia edo protokoloen heterogeneotasuna. Arazo honek garrantzi handia du "legacy systems" direlakoetan. Oro har, sistema hauek zaharkituak geratu dira, eta sistema zein azpisistema baten kodea eskakizun edo teknologia berrietara egokiezina da, normalean, administratzaileak kodea ulertzen ez duelako, edo iturburuak galdu egin direlako. Beraz, "legacy system" hauek itxura jakin bat duten sistemak dira, mundu ideal batean alda zitezkeenak baina praktikan aldaezinak direnak (Alderson eta Shah, 1999).

Beste azpiarloa *integrazio semantikoa* delakoa da, eta bere helburu nagusia eskemen arteko alde semantikoaren ebazpenean datza. Eskeman agertuta-

[2] *Datu* eta *informazio* terminoak nahastea komenigarria ez bada ere, *informazioaren integrazioa* eta *datu-integrazioa* arloen mugak oso lausoak dira; horrela, bada, bi termino hauek sinonimotzat hartu ohi dira literaturan: adibidez, Wiederhold-en lanak (1996; 1992).

[3] *Wrapping* ere deiturikoa, adibidez (Genesereth eta Ketchpel, 1994; Roth eta Schwartz, 1997)

ko kontzeptuen arteko desberdintasuna hainbat arrazoirengatik gerta daiteke (adib., ikus Garcia-Molina *et al.* (1997); Kashyap eta Seth (1996)), ezagutzaren ingeniari desberdinek egindako kontzeptualizazioen desberdintasunak direla medio, seguruenik. Aldeak ez dira soilik eskema-entitateen artean izango (eredu erlazionaleko datu-baseko erlazioak, objektuei zuzendutako datu-baseko klaseak eta instantziak, eta abar), eta datu mailan ere agertuko zaizkigu. *Datuen arazketa* ("Data cleansing") delako teknikek (Jarke *et al.*, 2000) objektuen identifikazioa azpimarratzen dute (hots, datu-iturri desberdinetan adierazitako objektuen arteko baliokidetzak ebaztearen arazoa), datuak eskuratzerakoan gertatutako akatsen kudeaketarekin batera.

Nolanahi dela, informazioaren integrazioaren arloan, azken urteotan, bultzakada eta perspektiba-aldaketa sakonak burutu diren arren, bere hasierako helburuak, teknika multzoak eta arazo nagusien identifikazioa datu-base federatuetan eta datu-base anizkoitzetan aurkituko ditugu. Gauzak horrela, bi arlo hauen sarrera egingo dugu hurrengo ataletan, labur-labur bada ere, irakurlea arlo garrantzitsu honen testuinguru aproposean kokatzeko.

III.1.1 Federazioak eta datu-base anizkoitzak.

Datu-base anizkoitzen ("multidatabase") arloan aritu zirenak ohartu ziren datuen integrazioaren arazoaz estreinako aldiz. Datu-base anizkoitzak hainbat datu-baseren bildumak dira; datu-base hauek heterogeneoak izan daitezke eta, horrela, arazo franko sortuko da datu-baseen artean datuak elkartrukatu nahi badira. Seth eta Larson lanean (1990) agertzen den sailkapenaren arabera, datu-base federatuek (Heimbigner eta McLeod, 1985) datu-base anizkoitzen azpimultzoa osatzen dute. Datu-base federatuak elkarlanean jardun behar duten baina berez autonomoak diren hainbat datu-baseren bilduma diren bitartean, datu-base ez-federatuetako datu-baseak ez dira autonomoak izango, nahiz eta hainbat eskema heterogeneorekin lan egin ahal izan. Datu-base anizkoitz ez-federatuetan administratze-maila bakarra egongo da, eta datu-kudeaketako operazioak datu-base guztietan uniformeki gauzatuko dira.

Datu-base federatuen barruko datu-baseak autonomoak dira, eta autonomia hori hainbat mailatakoa izan daiteke (ikus Seth eta Larson (1990); Heimbigner eta McLeod (1985)):

- *Diseinu mailako autonomia*: datu-base bakoitzaren administratzaileak askatasuna du datu-basearen diseinuari buruzko aukeraketan. Horrela,

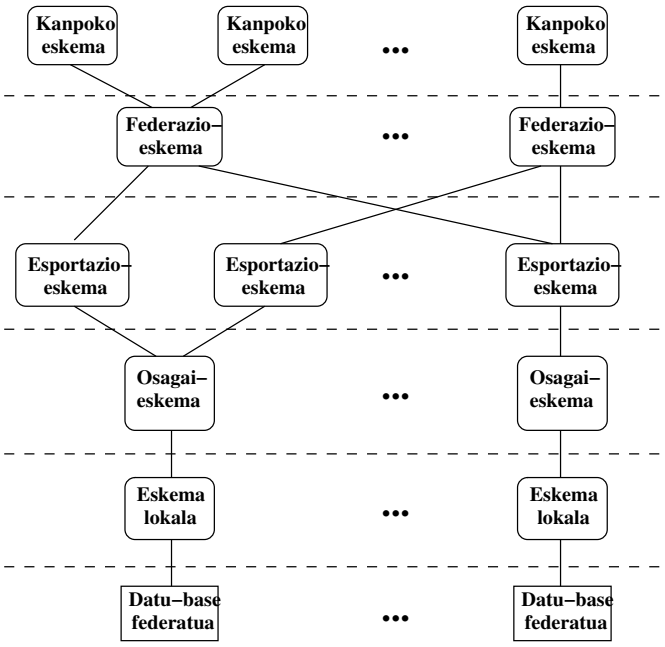
bada, datu-basearen datu-eredua, kontsulta-lengoaia, eskemaren eraketa eta datuen gaineko kontzeptualizazio zein interpretazio semantikoak ezarri ahal izango ditu.

- *Komunikazio mailako autonomia*: datu-base lokalek erabakiko dute kanpotik datorren galdera bat *noiz* eta *nola* erantzun.
- *Exekuzio-autonomia*: datu-base lokalek erabakiko dute eskatutako transakzioen ordena zein izango den, eta ordena horri buruzko informaziorik ez dute eskainiko.
- *Asoziazio-autonomia*: datu-base bakoitzak erabakiko du zein informazio banatuko duen federaziora.

Oro har, autonomia mailaren eta informazioaren elkarrekikotasunaren beharraren artean konpromisoa bilatu behar da: datu-baseen artean zenbat eta autonomia handiagoa izan, orduan eta konplexuagoa bilakatuko da datu-base hauen informazioa konpartitzea. Horrela, bada, autonomia mailaren bat (edo gehiago) lasaitu egin ohi da praktikan, elkarreraginkortasuna eskaintzearren.

Datu-base federatuen barnean kokatutako sistemak bi multzotan sailkatu ohi dira: uztardura hertsia duten sistemak eta uztardura lausoa duten sistemak (Seth eta Larson, 1990). Uztardura hertsia duten sistemak entitate komun bakarra izango balira bezala administratu ohi dira: eskema orokor bateratua edukiko dute, eta erabiltzaileak eskema horren gainean igorriko ditu galderak federazio osora. Eskema hau prozesu (erdi-)automatiko baten emaitza izan daiteke edo *ad hoc* sorturiko eskema izan daiteke. Bestalde, federazioaren osagai diren iturrietako informazioaren estaldura semantikoa osoa edo partziala izan daiteke. Edonola ere den, eskema komuna federazioan osagai diren eskemen arteko batura bezala ikusi ohi da. Uztardura lausoa duten sistemek, ordea, ez dute eskema komunik edukiko; horrela, bada, sistema hauek eskaintzen dutena galdeketa-lengoaia bateratua da, iturrietan gorderiko datuen atzipena bideratuko duena.

Datu-base sistema modernoek hiru mailako arkitektura erabili ohi dute (Tsichritzis eta Klug, 1978), non adierazpen fisikoa logikotik banatua dagoen, eta eskema logikoa erabiltzaile edo aplikazioen ikuspuntutik aldentzen den, bisten laguntzaz. Hala ere, datu-base federatuetan hiru maila hauek ez dira nahikoak, eta bost mailako arkitektura behar dela esan ohi da (Seth eta Larson, 1990). Gauzak horrela, datu-base federatuko osagai den datu-base batean honako maila hauek aurkituko dira:



III.1 Irudia: Datu-base federatuen bost mailako arkitektura

1. **Eskema lokala.** Eskema lokala bat dator, bete-betean, arkitektura klasikoko eskema lokalarekin.

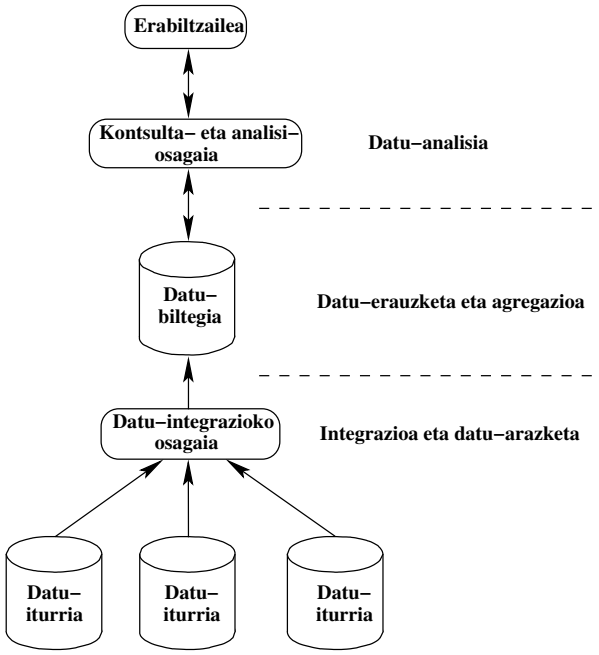
2. **Osagai-eskema.** Datu-base baten osagai eskema bere eskema lokalaren bertsio bat da, federazioan zehar banatutako datu-eredu eta adierazpen-formalismora itzulia.

3. **Esportazio-eskema.** Esportazio-eskemak federazio-eskema bakar bati dagokion eskemaren zatia soilik adierazten du.

4. **Federazio-eskema**[4]**.** Eskema honek federazioaren bista integratu eta homogeneoa adierazten du, zeinari esportazio-eskema anitz mapatzen zaizkion (datu-integrazioko teknologia lagun). Federazio batean ez du zertan federazio-eskema bakarra egon, atzigarri dauden datuen bista integral desberdinak federazio-eskema desberdinen bitartez adieraz baitaitezke.

5. **Kanpoko eskema.** Aplikazioek edo erabiltzaileek behar dituzten bista

[4]Inportazio-eskema edo eskema globala ere deiturikoa (Seth eta Larson, 1990)



III.2 Irudia: Datu-biltegi sistema baten oinarrizko arkitektura

bereziak adierazten ditu kanpoko eskemak, hiru mailako arkitektura klasikoan bezala.

Bost mailako arkitektura honek datu-base autonomo eta heterogeneoen integrazioa ahalbideratzeko aukera handiagoak emango ditu hiru mailako eskema klasikoak baino.

III.1.2 Datu-biltegiak.

Datu-biltegien ("Data Warehousing") arloa datu-integrazioa bera baina harago doan diziplinarteko ikerlerroa dugu. Enpresaren ingurunean kokatu ohi da, eta bere xede nagusia hainbat leku desberdin eta banatutatik datuak jaso, *garbitu* eta integratzea litzateke; azkenik, datu guztiak biltegi nagusi bakar batean —enpresako ikuspegi orokorraren adierazpidea den *Corporate Data Warehouse* delakoan— gordeko dira. Arlo honetan ere arreta berezia jartzen da datu esanguratsuen agregazioan. Ondoren, datuak erauzi eta transforma

daitezke, erabiltzaile jakin edo aplikazio berezietarako[5] moldatutako eskema baten arabera (Jarke *et al.*, 2000).

Erabilitako datuak askotan enpresako datu kritikoak direnez, eta datuen kopurua oso handia izan daitekeenez, teknologia bereziak garatu izan dira datuen agregazioa burutzearren, hala nola, dimentsioanitzeko datu-baseak (*Multi Dimensional Databases, MDDBMS*) edo datu-kuboak direlakoak (*data-cubes*) (ikusi, berriro ere, (Jarke *et al.*, 2000).

Datu-integrazioa funtsezkoa da datu-biltegietan. Horrela, biltegiak datu-integrazioa gauzatuko duten *wrapper*-ez, bitartekoez eta kargatzaileez baliatzen dira. *Wrapper*-ek eta kargatzaileek informazio-iturrietatik datuak biltegian kargatu, transformatu, garbitu eta eguneratuko dituzte. Bitartekoek, berriz, datuak biltegian integratuko dituzte, informazio-iturrietako inkonsistentziei eta gatazkei irtenbidea emanez.

Datu-integrazioko paradigma baten adierazpide garbienetarikoa da datu-biltegiek usuen erabilitakoa, hots, gauzatutako integrazioa ("materialized integration") delakoa. Mota honetako integrazioan, iturburutik datuak eskuratu, eredu komun batera bihurtu —lehenik aipaturiko enpresaren eredu orokorrera— eta datuak biltegi lokal batean *bikoiztu* egiten dira. Hortaz, eredu orokorreko datu-biltegian enpresako informazio-iturri guztien kopia gordeko da, beti ere eredu komun baten arabera adierazirik. Horretaz gain, datuen historikoa gordetzen da, datuek denbora joan ahala izan dituzten aldakuntzak gordez. Izan ere, zenbat eta informazio gehiago eta aberatsagoa gorde, orduan eta zehatzagoak izango baitira datu horien gainean egin daitezkeen analisiak.

Datu-biltegien, eta, oro har, gauzatutako integrazioko hurbilpena erabiltzen duen sistema guztien arazo larri bat datuen eguneraketan datza. Datuen kopiak egiten direnez, jatorrizko datuak aldatzen direla aurreikusi behar da, eta, horrela, kopien eguneraketaren politika egokia jarraitu. Bestela, enpresak datu zaharkituekin lan egiteko arriskua izango du, eta, beraz, datu horien gainean eginiko analisi eskasek erabaki okerrak hartzea eragingo dute.

Datu-biltegiek interes handia sortu dute industria arloan, eta hainbat inplementazio komertzial egin da, hala nola, Informix edo MicroStrategy sistemak. Horretaz gain, ikerketa-proiektu franko garatu da arlo honen ildotik, adibidez, WHIPS (Garcia-Molina *et al.*, 1998), SQUIRREL (Zhou *et al.*, 1995) edo DWQ (Jarke eta Vassiliou, 1997; Calvanese *et al.*, 1999b).

[5] *Online Analitic Processing, OALP* izendatutako analisi-tresnak, kasu.

III.2 Datu-integrazioa eta adimen artifiziala.

Datu-integrazioko sistemen aurreneko garatzaileek datu-integrazioa eta adimen artifizialaren artean dauden erlazio estuak nabaritu zituzten. Horrela diote, esaterako, Information Manifold sistemaren garatzaileek (Levy, 1998):

> The approach we took in designing the Information Manifold was based on the observation that data integration is a problem at the intersection of the fields of Database Systems and Artificial Intelligence.

Hortaz, datu-integrazioa datu-base tradizionalen eta adimen artifizialaren arteko mugan dagoen ikerlerroa dugu. Egileen esanetan, datu-integrazioaren helburu nagusia informazio-sistema autonomoen elkarreragingarritasuna bermatzea izanik, eta iturrien autonomia maila handia izango bada, informazio-iturrien gainean bat-bateko aldaketak egongo direla aurreikusi behar da. Horrela, bada, informazio-iturrietan gorderiko datuak adierazteko mekanismo aberatsak behar direla azpimarratzen dute, hots, ezagutza-adierazpeneko teknikak. Bestalde, erabiltzaileak jarritako galdera erantzuteko, ingurune banatu eta heterogeneo batera jo beharko da, eta, horretarako, planifikatzaile baten beharra suma daiteke.

Beraz, ontologiek, ahalmen-deskribapenak, plangintzak edo ezagutzan oinarritutako sistemen arloak erlazio zuzena dute datu-integrazioarekin. Atal honetan adimen artifizialeko teknika horietan jarriko dugu arreta, datu-integrazioan izan duten eragina bereziki aztertuz.

III.2.1 Ontologian oinarritutako datu-integrazioa.

Filosofia arlotik etorri zaigun *ontologia* hitza definitzeko hainbat saio egin dira. Erabateko akordioa ez dagoenez, (Studer *et al.*, 1998) lanean azaldutako definizio bat geureganatuko dugu eta, horrela, ontologia *kontzeptualizazio konpartitu baten espezifikazio formal eta esplizitua* dela esango dugu. Zera dio egileak definizio hau azaltzerakoan:

> Conceptualization refers to an abstract model of some phenomenon. Explicit means that the type of concepts used, and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine-tractable. Shared

> reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group.

Ontologiak, beraz, domeinu zehatz bati buruzko ezagupena modelatuko du, ezagupen hori hainbat entitatetan zehar banatu ahal izateko. Informazio *guztia* adieraztea ezinezkoa dela onartuz, ontologia batek domeinuaren alde zehatz baten ikuspuntua emango digu. Horretarako, ontologiek, oro har, domeinuan parte hartuko duten kontzeptu multzoa eta kontzeptuen arteko erlazioak zehaztuko dituzte, hots, domeinuaren gainean kontzeptualizazioa deritzoguna zehaztuko dute. Horrela, bada, ontologiak domeinu bateko kontzeptualizazioaren gaineko teoria formalak izango lirateke[6].

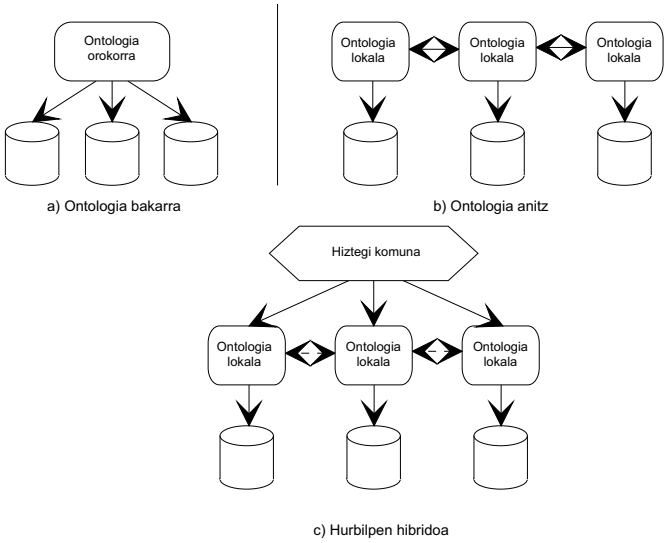
Ontologiak informazio-sistemetan erabili izan dira, besteak beste, iturriko informazioaren semantika deskribatzeko, hots, edukiaren esanahia azalarazteko, edo erabiltzailearekin elkar harremana bermatzeko. Datuak integratzerakoan, semantikoki bat datozen informazio-kontzeptuak identifikatzeko ere erabili ohi dira. Edonola ere, sistema hauek hainbat modutan baliatu ohi dira ontologiez. Oro har, hiru hurbilpen nagusi aurki ditzakegu (Wache *et al.*, 2001):

- **Ontologia bakarra.** Sistema hauek semantikaren espezifikaziorako hiztegi konpartitua adierazten duen ontologia bakarra erabiliko dute. Informazio-iturri oro ontologia orokor honen arabera deskribatuko da (ikus Arens *et al.*, 1996).

- **Ontologia anitz.** Hurbilpen honetan, informazio-iturri bakoitza bere ontologiaren arabera deskribatuko da. Sistema hauetan, iturri guztiek komun duten konzeptualizaziorik ez da egin behar izango eta, halaber, iturriak gehitzea/kentzea/aldatzea lan independentea bilakatuko da. Iturrien arteko informazioa elkartrukatu nahi bada, ordea, iturri desberdinen ontologien kontzeptuen arteko baliokidetasunak — *inter-ontology mappings* direlakoak[7]— ezarri beharko dira. OBSERVER sistema (Mena *et al.*, 1996, 2000) hurbilpen honen adibide garbia dugu, iturri bakoitzaren semantika ontologia batean deskribatzen baita. Ontologia guztien artean komuna den hiztegirik ez dagoenez, baliokidetasun

[6] Jakina, kontzeptualizazioa ezagutza-ingeniari baten garunetan egongo da, ziurrenik.

[7] *Inter-ontology mapping* delakoa ontologia desberdinetan semantikoki baliokideak diren kontzeptuen identifikazioan datza.



III.3 Irudia: Ontologien erabilpenaren hiru arkitektura

hauek gauzatu ahal izateak adierazpen-formalismo gehigarrien menpe egon beharko du (Mena *et al.*, 1996; Preece *et al.*, 1999; Calvanese *et al.*, argitaratzeke; Wache, 1999).

- **Hurbilpen hibridoa.** Ontologia anitzeko kasuan bezala, iturri bakoitzaren semantika ontologia propioaren arabera deskribatuko da. Haatik, ontologia lokalen arteko baliokidetasuna gauzatu ahal izateko, ontologiek hiztegi komun eta konpartitua dute (Goh, 1997; Wache *et al.*, 1999). Hiztegi konpartituak sistema osoan erabilitako oinarrizko terminoak (primitibak) adieraziko ditu, eta termino hauen domeinua ontologia lokaletan zehar banatua dago. Hiztegi komun konpartituaren erabilpena dela eta, ontologia lokalen arteko baliokidetasunak egiten ahal dira; beraz, hurbilpen hibridoa ontologia anitzekoaren arazoak baretzera dator, nahiz eta berak ere puntu ahulak erakutsi. Izan ere, zaila baita aldez aurretik existitzen diren ontologiak berrerabiltzea, hauek erabilitako hiztegia sistemaren hiztegi komunera egokitu beharko baita.

III.3 irudian ontologi(ar)en erabilpenaren hiru hurbilpen hauek ikus ditzakegu.

Ontologiak, edukiaren esanahia deskribatzeko ez ezik, galdeketa-eredu orokorreko lanetarako ere erabili izan dira. SIMS sisteman (Arens *et al.*,

1996), ontologia sistema banatuaren galdeketa-eredua da: erabiltzaileak galderak ontologia orokorraren gainean adieraziko ditu, SIMSek galdera hori baliabide lokalaren azpiereduetara itzuliko duelarik. IV kapituluan ikusiko dugun legez, ELHISAk hurbilpen horixe bereganatzen du iturri lexikalen integrazioa bideratzerakoan.

Ontologiaren edukia deskribatzeko bost osagai mota erabili ohi dira: klaseak, erlazioak, funtzioak, axiomak eta instantziak (Gruber, 1993). Horrela, bada, ontologiak ez dira soilik kontzeptuen arteko taxonomiak izango eta, printzipioz, edozein motatako ezagutza adieraz dezakete. Gure lanerako interes gehien duen datuen integrazioaren arloan, deskribapen-logikak (ikus III.5) edo logika hauen aldaeraren bat —*datu-base terminologiko* delakoaren paradigmaren pean— maiz erabili izan dira informazio-sistemen ontologiak adierazteko (Mena *et al.*, 1996; Kashyap eta Seth, 1996).

Ontologien ingeniaritza (Guarino, 1997; Gruber, 1992, 1993) tamaina zabaleko ontologien garapen eta mantentze-lanetan datza. Arlo honetan hainbat ikerketa-proiektu eta tresna garatu izan da. Ikus (Jones *et al.*, 1998; Duineveld *et al.*, 1999), non ontologia ingeniaritzako diseinu-metodologia zein tresnerien zerrenda zabala azaltzen baita. *Ontolingua* zerbitzaria (Farquhar *et al.*, 1996), esate baterako, ontologien ingeniaritzako tresna lagungarria dugu. Aurretik existitzen diren ontologiak batzearen arazoak[8] ere ontologien ingeniaritzako —edo ontologien gaineko berringeniaritzako— arloarekin erlazio estuak ditu. Edonola ere, eskala handiko ontologien diseinuak zein mantentze-lanak hainbat arazori egin behar dio aurre, eta arlo honen ikerkuntzan *diseinu-patroiak* edo *mikro ontologien* liburutegiak, eskaripean nahi den domeinuko ontologiak garatzeko eraikitze-bloke minimoez osaturiko liburutegiak, alegia, erabili izan dira.

Adimen artifizialaren bitarteko datu-integrazioa helburu duten sistemek, domeinua deskribatzeko (adierazten den munduaren teoria) ongi formalizatutako ontologia globala[9] duen arkitektura erakutsi ohi dute, non datu-iturriak ontologiaren arabera *wrapper* direlakoen bidez integratzen diren. Horrelako sistemak, *informazio-sistema globala* delako kategorian kokatu ohi dira.

[8]Eskema-integrazioaren arazoaren antzekoa dena (Batini *et al.*, 1986).

[9]Nahiz eta, III.2.1 atalean ikusitakoaren arabera, munduaren ezagutza ontologia *bakar* baten bidez adierazi behar ez izan. Haatik, ontologia anitzaren arkitekturan, domeinuaren formalizazioa ontologien arteko baliokidetzaren bidez gauzatuko da.

III.2.2 Plangintza.

Problemen ebaztearen ("problem solving") arloko aplikazio garrantzitsua den *plangintza* adimen artifizialeko oinarrizko aztergaien artean dago, ikerlerro berezia bilakatu delarik, bere emaitza teoriko eta algoritmo bereziekin (Weld, 1999).

Plangintza-arazoak formalki deskribatzeko munduaren hasiera-egoera, xede-egoera eta plangintza-eragileak —aurre/ondoko baldintzen bitartez adierazitakoak— behar dira. Plangintzaren emaitza, berriz, mundua hasierako egoeratik xede-egoerara igaroaraziko duen eragile-segida bat izango da, partzialki ordenatua egon daitekeena.

Plangintza behar duten sistemek eragileen deskribapenarekin hertsiki lotua dagoen *edukiaren deskribapena* izan behar dute, eta eduki-deskribapen hauek definitzeko hainbat lengoaia garatu izan da, hala nola, LARKS lengoaia (Wickler eta Tate, 1998), Retsina sistemak (Sycara *et al.*, 1998) duen edukiaren deskribapenerako formalismoa, deskribapen-logiketan oinarritutakoak etab.

Datu-integrazioaren arazoa ere plangintza-arazoa izango balitz bezala formula daiteke; edukiaren deskribapenek eskura dauden datu-iturrietatik lor daitekeen informazioa azalaraziko dute, eta, horrela, edukiaren deskribapenak planifikatzailearen arrazoibidea ahalbideratuko du. Arazoa ikuspuntu honen arabera ikusita, datu-iturrien eta ontologia orokorraren arteko baliokidetasunen noranzkoa metodo klasikoen (datu-base federatuak, datu-base anizkoitzak etab.) alderantzizkoa da. Izan ere, metodo klasikoetan xede-kontzeptuak datu-iturrien gaineko bisten bidez adierazten baitira, integrazio-eskema orokorraren partea izango balira bezala[10]. Datu-integrazioko metodo klasiko honi *globala bistatzat* ("global as view (GAV)") integrazioa esaten zaio (ikus III.3.2 atala).

Plangintzaren bidezko datu-integrazioan, aldiz, datu-iturrien edukiak munduko eredu orokorraren terminoetan osatutako eduki-deskribapenen bidez adieraziko dira. Galderak erantzuteko plan bat eraiki eta exekutatuko da. Plana eraikitzeko, bestalde, datu-iturrietako datuak erauzi eta konbinatu egin beharko dira, eta ataza horretan datu-iturrietako edukiaren deskribapenak funtsezkoak dira. Datu-integrazio mota honetan erabilitako baliokidetzak —mapaketa— iturri lokalen deskribapenen bidez adierazten dira, eta deskribapen hauek, bestalde, ontologia orokorraren terminoen araberakoak

[10]Kontzeptu-mailako deskribapena da hau. Baliokidetasun hauek metodo prozeduralak erabiliz deskribatzen dira maiz.

izango dira. Horrela, *lokala bistatzat* ("local as view, (LAV)") hurbilpena jarraitzen duen integrazioa dela esaten da (ikus III.3.3 atala).

III.3 Datu-integrazioa.

Oraindaino, informazio-integrazioaren ikuspegi orokorra ikusi dugu, arloaren sailkapen bat aurkeztu, eta informazio-sistema heterogeneo eta autonomoak harremanetan jartzeko egin diren saioak azaldu ditugu. Halaber, datu-integrazioaren eta adimen artifizialaren artean dauden erlazioak azpimarratu ditugu. Atal honetan datu-integrazioan jarriko dugu arreta, eta arazoa formalki aztertuko dugu.

Datu-integrazioa prozesu konplexua da oso, bere baitan arazo desberdinak hartzen dituena. Horrela, bada, integrazio-sistemak hainbat eratan sailka daitezke, ezaugarri desberdinetan oinarrituz: integrazio birtuala vs. gauzatutako integrazioa, datu sasi-egituratuak —hots, *web* orrietan aurkitutakoak— vs. datu egituratuak —datu-base tradizionalak—, ontologia bakarra vs. ontologia anitz, eta abar.

Guk, atal hau aurkezteko, *globala bistatzat* (GAV), eta *lokala bistatzat* (LAV) hurbilpenak bereiziko ditugu nagusiki, eta galderen berritzulpenaren prozesuan arreta berezia jarriko dugu. Atal honetan zehar luze arituko gara GAV vs. LAV auzia azaltzen, eta bakoitzaren ezaugarri nagusiak eta abantailak/desabantailak aipatuko ditugu. Orain, hala ere, hurbilpenen deskribapenari ekingo diogu, labur-labur bada ere.

Ikusiko dugun bezala, integrazio-sistema bat formalki deskribatzeko hiru osagai nagusi behar dira: eskema orokor bat, iturri lokal bakoitzaren eskema eta mapaketa semantikoak adierazten dituzten erregelak, zeintzuen bitartez eskema orokorra iturrietakoekin harremanetan jar daitekeen. LAV eta GAV hurbilpenak mapaketa semantikoaren *noranzkoaren* arabera bereizten dira. Horrela, bada, GAV hurbilpena jarraitzen dela esango da baldin eskema orokorreko osagaiak iturrietako eskemetako osagaien arabera definitzen badira. LAV hurbilpenean, berriz, iturrietako osagaiak eskema orokorraren arabera definituko dira.

Ez dirudi hainbesterako aldea dagoenik LAV eta GAV hurbilpenen artean. Aitzitik, ondorio sakonak edukiko ditu, integrazio-sistema bat garatzerakoan, hurbilpen bat edo bestea aukeratzeak. LAV hurbilpenaren abantaila nagusia informazio-iturriak adierazteko malgutasunean datza, batez ere, iturri berriak gehitu/ezabatu/aldatu nahi direnean. Izan ere, iturri bat sisteman

integratzeko behar den informazioa independentea baita, LAVen, gainontzeko iturriekiko. Hala ere, galderaren itzulpena, LAV hurbilpenean, prozesu konplexua da, mapaketa semantikoa deskribatzeko erabili den lengoaiaren espresibotasunaren araberakoa; adierazpen-ahalmen handiko lengoaietarako, berez, erabakiezina den prozesua izan daiteke (Abiteboul eta Duschka, 1998). GAV hurbilpenean, berriz, galderen itzulpena prozesu sinpleagoa da (Ullman, 1997). Hala ere, iturriak gehitzea edo aldatzea sistema osoan eragina zuzena duen prozesua izango da, eta, horrela, iturri batean aldaketak egiteak sistema osoa birplanteatzea ekar dezake.

Datu-integrazioaren beste arazo inportantea iturrietan galderak egiterakoan egon daitezkeen galdeketa-gaitasunetan datza. Gaitasun hauei garrantzia eman zaie, batez ere, kontsulta-lengoaia propioak dituzten iturrien edo Interneteko orrien informazioa integratu nahi izan denean. Izan ere, *web* orrietatik ezin baita informazioa edonola erauzi. Askotan, galderak formularioen bitartez jarri behar zaizkio orriei, eta gerta daiteke formularioetan soilik zenbait atributuri buruz galdetu ahal izatea. Konparazio batera, Interneten *on-line* dauden hiztegi-orriek informazio aberatsa eskain dezakete, hala nola, hitzen definizioak, kategoriak edo erabilpen-adibideak. Hala ere, galderak egiteko behar-beharrezkoa da hitzaren forma aldez aurretik jakitea. Ezinezkoa da, horrela, "*Izen kategoriako A letraz hasten diren sarrerak eman*" bezalako galderak egin. Bai, ordea, "lagun *hitzaren definizioa eman*" bezalako galderak.

Azkenik, *datuen arazketa* delako arazoa azalduko dugu. Integrazio-sistemek semantikoki desberdinak diren entitateekin lan egin behar dute, eta entitate horiek nolabait integratu. Semantikoki bat ez etortzeak, baina, bi maila nagusitan sailka daitezke, hots, intentsio-maila eta maila estentsionala. Intentsio-mailako parekatzeak, neurri handi batean, eskemaren integrazioarekin zerikusi handia du (Batini *et al.*, 1986; Seth *et al.*, 1993), nahiz eta, esan bezala, datu-integrazioan iturrien autonomia-maila sistema klasikoetan baino handiagoa izan. Bestalde, maila estentsionaleko integrazioa —datuen arazketa—, datuen gainean egiten da (Galhardas *et al.*, 2000; Hernandez eta Stolfo, 1995; Monge, 1997; Jarke *et al.*, 2000). Oro har, bi prozesu burutu behar dira. Batetik, zenbait iturritatik datu *zikinak* etorriko direla aurreikusi behar da (errore tipografikoak dituztelako, atributu bakar batean balio anitz txertaturik daudelako eta abar). Bestetik, bi iturri edo gehiagotik datozen datuak mundu errealeko entitate bera erreferentziatzen dutenetz erabaki behar da. Ikusiko dugun bezala, datuen arazketaren prozesua behar beharrezkoa da kalitatezko integrazioa burutu nahi bada, eta, besteak beste, informazio-itu-

rri guztietatik jasotako informazioa elkar erlazionatzeko ezinbestean burutu beharreko prozesua da.

III.3.1 Aurrekariak. Definizioak.

Datu-integrazioaren arazoa aztertu aurretik, datu-baseko teoriaren arloko hainbat definizio azalduko ditugu, definizio hauek behar-beharrezkoak baititugu atala ulertu nahi bada. Nolanahi ere, erreparorik gabe aitortuko dugu ez dela lan honetan konputagarritasunaren teoria edo konplexutasunaren teoriaren azterketarik egingo. Izan ere, luzeegi joko bailiguke orain azaldutako kontzeptuen deskribapen teoriko sendo eta zabala eskaintzeak. Horrela bada, irakurleari logika matematikoa, datu-baseen printzipioak eta, bereziki, datu-baseen eskema eta galdeketa-lengoaien ulermena aurresuposatuko zaizkio. Nahi duenak, jo dezala, adibidez, (Abiteboul *et al.*, 1995) lanera, kontzeptu hauen azalpen zabala eta zehatza aurkitzeko.

III.3.1.1 Galderak.

Jo dezagun **dom** dela balio atomikoen domeinu infinitu zenbakigarria, eta definizio hauek ditugula:

- R erlazio-eskema bat, honako osagaiak dituena: erlazio-izen bat, atributu-izenen multzo bat eta aritatea.
- **R** eskema erlazionala, erlazio-eskemez osaturiko multzoa dena.
- I erlazio bat, n-koteen multzoa, eta $I \subseteq \mathbf{dom}^n$[11].
- **I** datu-basearen instantzia, erlazioen multzoa dena.

Formalki, Q galdera erlazionala izango da baldin **R** eskemaren eta **dom** domeinuaren gaineko **I** instantzia bakoitza $\mathbf{R}'$ eskema desberdin baten gaineko beste **J** instantzia batekin lotzen duen aplikazioa bada.

Galdera erlazionalak gutxienez bi ikuspuntutatik ikus daitezke, hots, aljebraikoa (ALG) edo kalkuluen bidezkoa ($CALC$). Aljebra erlazionalak (ALG) oinarrizko eragiketa aljebraiko hauek ditu:

[11] $\mathbf{dom}^n$ espresioak **dom** domeinuaren gainean n aldiz eginiko biderketa kartesiarra adierazten du.

- Multzoetan oinarritutako aritate bera duten erlazioen gaineko eragiketak: bildura $\cup$, ebakidura $\cap$ eta diferentzia $\setminus$.

- N-koteetan oinarritutako eragiketak; erlazioen errenkadaren bat ezabatu edo izena aldatzen duen π proiekzioa, erlazio baten n-koteak predikatu baten arabera iragazten dituen σ hautaketa.

- Biderketa cartesiarra (x) non, R_1 eta R_2 erlazioak emanda, n eta m aritatekoak hurrenez hurren, $(n+m)$ aritateko erlazio berri bat sortzen duen. Erlazio berri honek $< t_1, t_2 >$ n-kotea izango du t_1 eta t_2 bikote bakoitzeko, non $t_1 \in R_1$ eta $t_2 \in R_2$.

Lehen mailako domeinu-kalkulu erlazionalean (CALC), aldiz, galderek ondoko forma dute:

$$\left\{\left\langle \bar{X} \right\rangle \mid \Phi(\bar{X})\right\}$$

non $\bar{X}$ aldagaien n-kotea den eta Φ lehen mailako formula, predikatu erlazionalen gainean $\forall, \exists, \vee, \wedge$ eta $\neg$ eragileak erabiltzen dituena.

$\forall$ eta $\neg$ eragileak ez dituzten galderei kalkulu erlazional positiboko galderak direla esango zaie. Soilik $\exists$ eta $\wedge$ dituzten galderei, berriz, galdera konjuntiboak ("Conjunctive Query (CQ)") deituko zaie. Galdera konjuntiboetan konstanteak ager badaitezke ere, ezin da, printzipioz, eragile aritmetikorik egon.

Galdera konjuntiboak erregela logikoen bitartez adierazi ohi dira. Horrela, $\left\{\left\langle \bar{X} \right\rangle \mid \exists \bar{Y} : p_1(\bar{X}_1) \wedge \ldots \wedge p_n(\bar{X}_n)\right\}$ galdera konjuntiboa, Q erregela honen bitartez adieraz daiteke:

$$h(\bar{X}) \text{ :- } p_1(\bar{X}_1), \ldots, p_n(\bar{X}_n).$$

$head(Q) = h(\bar{X})$ predikatuari galderaren *burua* ("head") esaten zaio, eta galdera erantzun ondoren lortutako instantzia multzoa adierazten du. Galderaren *gorputza* ("body"), bestalde, $body(Q) = p_1(\bar{X}_1), \ldots, p_n(\bar{X}_n)$ azpihelburuen multzoa da. $p_i(\bar{X}_i)$ azpihelburu bakoitzak bi osagai ditu: p_i erlazio-izen bat eta erlazio-eskema baten araberakoak izango diren $\bar{X}_i$ n-koteen argumentuak. Argumentuak aldagaiak edo konstanteak izan daitezke. Buruan agertutako $\bar{X}$ aldagaiei *aldagai nabarmenduak* ("distinguished variables") esango zaie. Galdera konjuntiboek *seguruak* izan behar dute, hots, buruan agertutako aldagai orok gorputzean ere azaldu behar du; matematikoki adierazita, $X \subseteq X_1 \cup \ldots \cup X_n$. Gorputzeko predikatuetan agertutako $\bar{X}_i = x_{i1}, \ldots, x_{im}$ aldagaiak hiru motatakoak izan daitezke: aldagai nabarmenduak ($x_{ij} \in \bar{X}$),

existentzialki kuantifikatutako aldagaiak ($x_{ij} \notin \bar{X}$) edo konstanteak. Q galdera konjuntiboan agertutako aldagaien multzoari $Vars(Q)$ esango zaio, hots, $Vars(Q) = X \cup X_1 \cup \ldots \cup X_n$.

Galdera konjuntiboek baliokide zuzena dute SQL lengoaiako select-from-where klausulekin, non where klausuletan murriztapenetan (=) berdintasun eragilea eta ("and") juntagailua soilik erabiltzen diren.

III.3.1 Adibidea Eman ditzagun $r(A_1, A_2)$ eta $s(B_1, B_2, B_3)$ erlazioak, eta honako galdera konjuntibo hau:

$$q(X, Z) \text{ :- } r(X, Y), s(Y, Z, 1).$$

galdera konjuntibo segurua da hau, hots, $((X, Z) \subseteq (X, Y) \cup (Y, Z))$, eta honako SQL galdera honen baliokidea da:

SELECT	r.A_1, s.B_2
FROM	r, s
WHERE	r.A_2 = s.B_1
and	s.B_3 = 1

bestalde, galdera (ALG) aljebra erlazionalean ere adieraz daiteke, honako eran:

$$\pi_{A_1,B_2}(\sigma_{A_2 \equiv B_1, B_3 \equiv 1}(r(A_1, A_2) \bowtie s(B_1, B_2, B_3)))$$

□

Demagun Q galdera konjuntiboa dela, eta D datu-base bat. Q-ren erantzuna, $ANS(Q, D)$ deiturikoa, honako eragiketa hauek burutu ondoren lortzen diren Q galderaren buruko predikatuekin sortutako multzoak izango dira:

- Q-ko gorputzaren aldagaiak konstanteekin era posible guztietan ordezkatu
- Azpihelburu guztiak "egia" bihurrarazi

III.3.2 Adibidea Demagun $D = r(2, 3), s(3, 4, 1), s(3, 5, 1)$ datu-basea. Aurreko galderaren erantzunaren multzoan $ans(2, 4)$ eta $ans(2, 5)$ n-koteak egongo dira, honako ordezkapen hauek burutu ondoren:

1. $X \to 2, Y \to 3, Z \to 4$;

2. $X \to 2, Y \to 3, Z \to 5$;

□

Desberdintasun-murriztapenak (adib., $\neq, \leq, \geq$) dituzten galderak ALG edo CALCetik at geratzen dira. Nolanahi ere, hauen gaineko hedapenak defini daitezke. Desberdintasun-murriztapenak dituen galdera konjuntibo batek honako forma du:

$$p(\bar{X}) \text{ :- } p_1(\bar{X}_1), \ldots, p_n(\bar{X}_n), x_{i_{1,1}}\theta_1 x_{i_{1,2}}, \ldots, x_{i_{m,1}}\theta_m x_{i_{m,2}}.$$

non $x_{i_{j,k}}$ aldagaiak $\bar{X}_1, \ldots, \bar{X}_n$ barnean dauden, eta $\theta_j \in \{\neq, \leq, \geq\}$.

Galdera konjuntiboak funtziorik gabeko Horn klausulen bitartez –*datalog notazioa* deiturikoa– adieraz daitezke. Horrela, bada, D datu-basearen gaineko galdera konjuntibo bat *datalog erregela* bezala honela adieraz daiteke:

$$p(\bar{X}) \text{ :- } p_1(\bar{X}_1), \ldots, p_n(\bar{X}_n).$$

Gorputzeko p_i bakoitza predikatua edo erlazioa izan daiteke. Erlazioa D datu-basean gorderik badago, datu-base estentsionaleko ("extensional database (EDB)") erlazioa dela esango da. Gainerako predikatuei —erregelen ezker aldean agertu ohi diren predikatuak, kasu—, bestalde, datu-base intentsionaleko ("intensional database IDB") erlazioak deituko zaie.

Datalog programa bat $\mathcal{P}$ *datalog* erregelen multzoa da. $\mathcal{P}$ *datalog* programaren *mendekotasun-grafoa* $G = \langle V, E\rangle$ grafo zuzena da, non $V = \{p_1, \ldots, p_m\}$ multzo bat den, $\mathcal{P}$-ren predikatu-izenekin sorturikoa, eta E-k p_i-tik p_j-ra doan arkua edukiko duen, baldin eta soilik baldin $\mathcal{P}$ programaren barnean P erregelarik badago, eta:

- P-ren burua p_i predikatua da.
- P-ren gorputzean p_j predikatua azaltzen da.

Datalog programa bat errekurtsiboa izango da baldin eta soilik baldin G grafoa ziklikoa bada.

Galdera positiboak (select-from-where-union galderak SQL lengoaian) adierazteko errekurtsiboak ez diren *datalog* programak erabil daitezke. Bestalde, galdera positibo oro galdera konjuntibo multzoaren bidez adieraz daiteke, multzo horretako erregela guztien burua berdina izango delarik. Multzo horien tamaina, ordea, *datalog* programa ez-errekurtsiboarenak baino aski handiagoa izan daiteke. *Datalog* programa ez-errekurtsibo batetik galdera konjuntiboen multzoa lortzen duen prozesuari *galderaren hedapena* esaten zaio.

III.3.1.2 Galderen barne-hartzea.

Q_1 galdera bat beste Q_2 galdera baten *barnean* dagoenetz erabakitzeari *galderen barne-hartzearen* ("query containment") arazoa esango zaio.

III.3.1 Definizioa Q_1 galdera bat Q_2 galderaren barruan egongo da, $Q_1 \sqsubseteq Q_2$, baldin eta soilik baldin edozein D datu-basetarako, Q_1 galderaren erantzun multzoa Q_2 galderaren erantzun multzoaren azpimultzoa bada, hots, $ANS(Q_1, D) \subseteq ANS(Q_2, D)$. Bi galdera baliokideak izango dira, $Q_1 \equiv Q_2$, baldin eta soilik baldin $Q_1 \sqsubseteq Q_2$ eta $Q_1 \sqsupseteq Q_2$. □

Aurrerago ikusiko dugun legez, erabat beharrezkoa izango dugu, informazioaren integrazioko lanetan dihardugunean, galdera baten erantzun multzoa beste galdera baten azpimultzoa denetz jakitea. Horretarako, zalantza hori argituko diguten zenbait prozeduraren bideragarritasuna aztertu beharko dugu. Estreinako azterketa batek, baina, berehala erakutsiko digu galderen barne-hartzearen arazoa ebaztearen bideragarritasunak galderak adierazteko erabili den lengoaiaren espresio-ahalmenarekin zerikusi zuzena duela. Izan ere, galdera-lengoaiaren espresibotasuna ahaltsua bada, galderen barne-hartzearen arazoa erabakiezina bihurtzen baita. Aljebra eta kalkulu erlazionalerako, konparazio batera, arazoa erabakiezina da (Sagiv eta Yannakakis, 1980), eta baita ere galderak adierazteko *datalog* programa errekurtsiboak erabiltzen badira (Shmueli, 1987). Galdera konjuntiboentzat, ordea, arazoa erabakigarria da, $\mathcal{NP}$-osoa konplexutasunarekin (Chandra eta Merlin, 1977). Izan ere, bi galdera konjuntibo izanik, bat beste batean egoteko, bien artean *barne-hartzearen mapatze* ("containment mapping") bat lortzea besterik ez da behar:

III.3.2 Definizioa Izan bitez Q_1 eta Q_2 bi galdera konjuntibo. τ *barne-hartzearen mapatze* bat Q_1-en aldagai zein konstanteetatik Q_2-ren aldagai zein konstanteetara doan funtzio bat da, zein(ek)

- identitatea den, Q_1-en konstanteetarako
- Q_1-en azpihelburu bakoitza Q_2-ren azpihelburu batekin lotzen duen
- Q_1-en burua Q_2-ren buruarekin lotzen duen

□

Q_1 galdera Q_2 galderaren barruan egongo da, baldin eta soilik baldin Q_1-etik Q_2-ra doan *barne-hartzearen mapatze*rik baldin badago (Chandra eta Merlin, 1977).

III.3.1.3 Datu-integrazioaren gure erreferentzia-eredua.

Azkenik, azpiatal honi amaiera emateko, datu-integrazioari buruz ari garenean zeri buruz ari garen zehaztuko dugu, oso labur bada ere, ondorengo atalak ulergarriagoak izan daitezen.

$\mathcal{I}$ datu-integrazioko sistema bat $< \mathcal{G}, \mathcal{S}, \mathcal{M} >$ hirukote bat da, non:

- $\mathcal{G}$ eskema globala den.
- $\mathcal{S}$ iturrien eskema den, integratu nahi diren informazio-iturri guztien eskemez osatua. Orokortasunik galdu gabe, bi aurrebaldintza suposatuko ditugu: 1) eskemak eredu erlazionalaren arabera adierazita egongo dira, eta 2) iturri bakoitzeko eskemen elementuen izenak (entitate zein erlazioenak) disjuntuak dira.
- $\mathcal{M}$ iturrien eta eskema orokorraren artean *mapaketa semantikoa* den. Mapaketak iturrien eta sistema osoaren eskema orokorraren arteko baliokidetasunak adieraziko ditu.

III.3.2 "Globala bistatzat" (GAV) delako hurbilpena.

Datu-integrazioaren arloan globala bistatzat eredua erabiltzen dela esango da, baldin eskema orokorra datu-iturrien eskemen arabera deskribatzen bada. Datu-base federatuek, datu-base anizkoitzek, datu-biltegiek edo agenteen bidezko integrazio-sistemek globala bistatzat hurbilpena jarraitu ohi dute; beraz, datu-integrazioan erabilitako estreinako arkitektura dugu.

Ikus dezagun, adibide baten bidez, globala bistatzat hurbilpenaren araberako bitarteko batek gauzatutako datu-integrazioa. Horretarako, zenbait datu gordetzen dituzten datu-iturriak edukiko ditugu[12]. Integrazioa gauzatzeko, sistemari eginiko galdera xede-iturrietarako itzuli beharko dugu. Suposatuko dugu xede-iturriak datu-baseen eredu erlazionalaren arabera daudela adieraziak. Arestian aipatu dugun legez, GAV hurbilpenean erlazio orokorrak datu-iturrien erlazioen arabera deskribatuko dira. Eman dezagun $p(\bar{X})$

[12]Datu-iturri hauek "benetako" datu-baseak, edo beste bitarteko baten emaitzak izan daitezke.

erlazio orokorra (eskema orokorrari dagokiona) eta $p_1, \ldots, p_n$ datu-iturrietako erlazioak direla. p erlazioa honako forma hau duten galdera konjuntiboen multzo finituen bidez adieraz daiteke:

$$p(\bar{X}) \leftarrow p_1(\bar{X}_1), \ldots, p_n(\bar{X}_n).$$

Erlazio orokorren arabera eginiko galdera bat itzultzeko, *galdera-hedapena* delako teknika erabiliko da: azpihelburu bakoitza, atzekari bezala azpihelburu hori daukan erregelaren aurrekariarekin ordezkatzen da, erregelako aldagaien izenak aldatuz (Ullman, 1997).

III.3.3 Adibidea Demagun hitzak gordetzen dituzten bi datu-base ditugula (DB_1 eta DB_2), hitz bakoitzeko lau atributu gordetzen dituztenak: identifikatzaile uniboko bat, hitzaren forma, zein hiztegitik jasoa den eta bere kategoria gramatikala. Suposa dezagun sistema orokorrean iturri horietatik, hitz-formak eta kategoriak adierazi nahi ditugula, *HitzKat* erlazio birtualaz baliatuz:

$$HitzKat(forma, kat) \leftarrow DB_1(id, forma, hiztegia, kat). \quad (1)$$
$$HitzKat(forma, kat) \leftarrow DB_2(id, forma, hiztegia, kat). \quad (2)$$

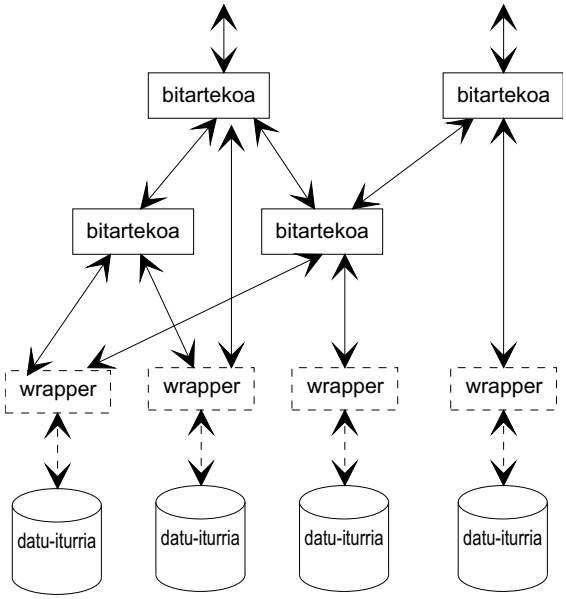
Demagun orain DB_3 hirugarren datu-basea dugula, DB_1 datu-baseko identifikatzaileekin bat datorrena, eta hitz-formen lemak gordetzen dituena. Ondorengo erregelak, *HitzLema* erlazioko n-koteak ateratzeko era adieraziko digu:

$$HitzLema(forma, lema) \leftarrow DB_1(id, forma, hiztegia, kat), \; DB_3(id, lema). \quad (3)$$

Suposa dezagun, azkenik, izen kategoriako hitz-forma baten lema eskatzen duen galdera dugula, erlazio orokorren arabera adierazia:

$$g(forma, lema) \text{ :- } HitzKat(forma, \text{"}Izena\text{"}), HitzLema(forma, lema).$$

Iturrietako erlazioen araberako itzulpenak lortzeko, jatorrizko galdera hedatu egiten da, galderako azpihelburuekin bat datozen erregelak aukeratuz. Horrela, (1) eta (2) erregelen atzekaria bat dator galderako lehenengo $HitzKat$ azpihelburuarekin, aldagaien arteko $\phi_1 = \{forma \rightarrow forma, kat \rightarrow \text{"}Izena\text{"}\}$ mapaketa burutuz. Bigarren azpihelburua, hots $HitzLema$, (3)



III.4 Irudia: Bitarteko-arkitektura

erregelarekin dator bat, aldagaien arteko mapaketa identitatea izanik. Hortaz, itzulpen hauek ditugu:

$$\begin{array}{lll} g(forma, lema) & :- & DB_2(id, forma, hiztegia, \text{“}Izena''), \\ & & DB_1(id, forma, hiztegia, \text{“}Izena''), \\ & & DB_3(id, lema). \\ g(forma, lema) & :- & DB_1(id, forma, hiztegia, \text{“}Izena''), DB_3(id, lema) \end{array}$$

□

Ikusi dugun bezala, GAV hurbilpenean eskema orokorra iturrien arabera definitzen da. Bien arteko mapaketa-erregelak zehazturik daudenean, berriz, kontzeptu (edo erlazio) orokorrak iturri lokaletik eskuratzeko modua erabat finkatua geratuko da. Hortaz, sistemaren diseinatzaileak iturri lokaleko datuak bildu eta informazio-sistemara bidaliko dituzten *bitartekoak* garatuko ditu, era prozedural batean beharbada. Izan ere, GAV hurbilpeneko integrazioak *bitarteko-arkitektura* (“Mediator architecture”) delakoarekin erlazio estua du. Bitartekoek datuen zenbait azpimultzotan kodetutako ezagutza ustiatzen dute, aplikazioko goi-mailetara informazioa hedatuz. Horrela dio egileak, bitartekoak definitzerakoan (Wiederhold, 1992):

A **mediator** is a software module that exploits encoded know-

> ledge about some sets or subsets of data to create information to a higher layer of applications

Horrela, bada, bitartekoak informazio-sistemari heterogeneotasun zehatz baten ebazpen pragmatikoa eskaintzen dioten "kaxa beltzak" dira. Bitartekoek hainbat iturriren —hala datu-baseak nola beste bitartekoak— esportazio-eskema ezagutu, eta sistemaren goi-mailetara eskemaren bat esportatuko dute.

Arkitektura honetan, integrazio-arazoa domeinu zehatzetako adituengana bideratzen da, haiek baitira heterogeneotasun berezia ikertu eta bitartekoa eraikiko dutenak. Bitarteko bakoitzak, beraz, heterogeneotasun bereziren bat ebatziko du. III.4 irudian ohiko bitarteko-arkitektura azaltzen da. Ikus daitekeenez, bitartekoek datu-iturrietatik —zeini, eskema esportatuak sistema orokorraren lengoaia zehatzera itzuliko dituzten *wrapper*-ak erantsi zaizkien— nahiz beste bitartekoetatik jasoko dute informazioa, eta goiko mailetara era bateratuan igorriko dute.

Arkitektura osoan bitartekoek betetzen duten lana interfaze dinamikoarena dela esan ohi da. Interfazea da, sistemaren bi mailaren arteko komunikazioa gauzatzen baitu. Dinamikoa dela esaten da, mailen arteko protokolo edo formatuak batu ez ezik, datu zein ezagutzaren abstrakzio- eta adierazpide-arazoei ere aurre egin behar baitie. Horrela, bitarteko batean interfazea jardunarazteko behar den prozesamendu-komandoak, datuen gaineko transformazioak gauzatzeko ezagutza-egiturak eta tarteko biltegiratzea aurkituko ditugu. Edonola ere, bitartekoek aurre egin behar dieten arazoen izaera aldakorra den neurrian, berek bete beharreko eginkizunak ere aldakorrak dira (Wiederhold, 1992).

Galdera-hedapenaren prozesua ez da konplexua, batez ere LAV hurbilpenekoarekin alderatzen badugu (ikus III.3.3 atala). Edonola ere, hedapena gauzatu ahal izateko behar-beharrezkoa den bisten definizioaren lanari ekiteko, baliabideen arteko erlazioen ulermen sendoa behar da eta, hortaz, ataza konplexua bilakatzen da. Horrela, bada, esango da GAV hurbilpenean galdera-itzulpenaren prozesua diseinu-mailan gauzatuko dela, eta LAV hurbilpenean, berriz, itzulpena exekuzio-denboran gauzatzen dela. Ondorioz, GAV hurbilpenaren pean kokatutako integrazio-sistemen problema larriena *bitartekoen* definizioan datza.

III.3.3 "Lokala bistatzat" (LAV) delako hurbilpena.

Lokala bistatzat delako eredua *galderen erantzutea, bistetan oinarrituz* ("Answering Queries using Views") problema teorikoarekin hertsiki lotua dago (Yang eta Larson, 1987; Levy *et al.*, 1995; Abiteboul eta Duschka, 1998; Beery *et al.*, 1997; Calvanese *et al.*, 1999a).

Datu-integrazioaren arloan, LAV hurbilpena informazio-sistema globalen arkitektura jarraitzen duten sistemetan erabili ohi da, hala nola, Information Manifold, InfoMaster edo SIMS[13]. Datu-integrazioaren arlotik at, bistetan oinarritutako galderen erantzutearen arazoa beste eginkizunetarako ere erabili ohi da, hala nola, galdera-optimizazioan edo datuen independentzia fisikoa mantentzea helburu duen arloan.

III.3.3.1 Galderen erantzutea, bistetan oinarrituz.

Demagun $p_1, \ldots, p_n$ eskema orokorreko erlazioak adierazten dituzten predikatuak direla, eta v datu-iturriren baten eskemakoa den erlazio bat dela. Lokala bistatzat hurbilpenean, v datu-iturriko erlazioa $p_1, \ldots, p_n$ predikatuak erabiliz osatutako galdera konjuntibo baten bidez adieraziko da, hots:

$$v(\bar{X}) \leftarrow p_1(\bar{X}_1), \ldots, p_n(\bar{X}_n).$$

Demagun $p_1, \ldots, p_m$ predikatu orokorren arabera adierazitako g galdera. Kontua da, orduan, g galdera berridaztea, datu-iturrien predikatuen arabera soilik adierazita egon dadin. Jakina, g galderaren gainean burututako berridazketaren emaitzak jatorrizko galderarena izan beharko luke, hots, berridazketa *baliokidea* izan beharko luke jatorrizko galderarekiko. Ikusiko dugu, ordea, baldintza hori zorrotzegia dela datu-integrazioaren arloan, eta arazo horri aurre egiteko jorratu izan diren asmabideak aztertuko ditugu.

III.3.4 Adibidea Eman ditzagun bi iturri-erlazio: hitzen adierak gordetzen dituen *hitzaAdiera(Hitza, AdieraId)* eta adieren definizioak gordetzen dituen *adieraDef(AdieraId, Def)*. Suposa dezagun honako bista hau dugula:

$$V : v(H, A, D) \leftarrow hitzaAdiera(H, A), adieraDef(A, D).$$

hots, bi erlazioen elkarketa (*join*). Ondorengo galderak

$$q(A, D) \text{ :- } hitzaAdiera(\text{"}lagun\text{"}, A), adieraDef(A, D).$$

[13] III.6 atalean sistema hauen azalpen laburra ikus daiteke.

lagun hitzaren definizio guztiei buruz galdetuko du. q galdera bistak erabiliz berridatz dezakegu, ondoko eran:

$$q'(A, D) \text{ :- } v(\text{"}lagun\text{"}, A, D).$$

□

Alde batetik, datu-integrazioaren ikuspuntutik, *berridazketa osoak* ("complete rewritings") behar ditugu, hots, jatorrizko galderako predikatu oro bistekin ordezkatu behar da. Bestalde, *minimoak* diren berridazketak nahi ditugu: Q galdera konjuntiboa *minimoa* dela esango dugu baldin ez badago Q' beste galdera konjuntiborik, non $Q' \equiv Q$ eta Q' galderak Q-k baino predikatu gutxiago dituen. Galdera positiboetan (galdera konjuntiboen multzoak), minimoaren propietatea erabili ahal izateko, galdera konjuntiboak, binaka hartuta, ezin dira erredundanteak izan: demagun $\{Q_1, \ldots, Q_n\}$ galdera positiboa dela, non Q_i bakoitza galdera konjuntiboa den, $1 \leq i \leq n$. Horrela, bada, $Q_i \not\subseteq Q_j$ eta $Q_j \not\subseteq Q_i$ edozein i, j bikotetarako, non $0 \leq i, j \leq n$ eta $i \neq j$.

Arestian aipatu bezala, berridazketaren erantzunak jatorrizko erantzunaren berdina izan beharko luke. Berridazketa horri *berridazketa baliokidea* esango zaio.

III.3.3 Definizioa Izan bitez $\mathcal{V}$= $\{v_1, \ldots, v_m\}$ bista konjuntiboen multzoa eta Q' galdera konjuntiboa, non Q'-ren gorputzako predikatu oro $\mathcal{V}$ multzoko erregela-bururen batekin bat datorren. Q'-ren *hedapena*, $(Q')^{hed}$, Q'-ren gorputzako predikatu bakoitza dagokion bista-erregelarekin ordezkatuz sorturiko galdera da. Bisten erregeletan nabarmenduak ez diren aldagaiak izen berria eta bakarra duten aldagaiekin ordezkatzen dira.

III.3.4 Definizioa Izan bitez Q galdera konjuntiboa eta $\mathcal{V}$ bista konjuntiboen multzoa. Q' galdera konjuntiboa Q-ren *berridazketa baliokidea*[14] dela esango da baldin:

- Q' berridazketa osoa bada, $\mathcal{V}$ multzokoarekiko.
- Q'-ren gorputzean agertutako predikatu bakoitza $\mathcal{V}$ multzoan duen definizioarekin ordezkatuz lortutako galdera (galderen hedapena) Q galderaren baliokidea bada, hau da, $(Q')^{hed} \equiv Q$.

[14]Existitzen bada.

III.3.4 adibidean, q' berridazketa q-ren *berridazketa baliokidea* da zeren, q'-ren hedapena osatzen badugu:

$$(q')^{hed}(A, D) = hitzaAdiera(\text{“}lagun\text{”}, A), adieraDef(A, D).$$

galderak baliokideak baitira, hau da, $(q')^{hed} \equiv q$. □

Nolanahi ere, datu-integrazioaren esparruan bistak datu-iturrien edukiak deskribatzeko erabili ohi dira eta, testuinguru honetan, ezin da galdera baten berridazketa baliokidea aurkituko denik ziurtatu. Izan ere, ez baitakigu jatorrizko galderaren bidez eskatu den informazio guztia datu-iturrietan aurki daitekeenetz. Pentsa dezagun izenak soilik gordetzen dituzten datu-baseak ditugula atzigarri. Nahiz eta erabiltzaileak edozein kategoriatako hitzei buruzko galderak egin, sistemak izen kategoriako hitzen informazioa bakarrik itzuliko dio.

Gauzak horrela, berridazketa baliokideak lortzeak baino interes handiagoa edukiko du, espresio-lengoaia zehatz batean eginiko galdera izanik, bistetatik lor daitekeen erantzun maximoak eskaintzen duen espresioa aurkitzeak. Espresio honi bisten araberako *maximoki barne-harturiko berridazketa* (“maximally contained rewriting”) esango zaio:

III.3.5 Definizioa Izan bitez Q galdera positiboa eta $\mathcal{V}$ bista konjuntiboen multzoa. Q' galdera konjuntiboa Q-ren (galderen positiboen araberako[15]) *maximoki barne-harturiko berridazketa* dela esango da, baldin eta soilik baldin:

- Q' $\mathcal{V}$-ko predikatuak soilik erabiltzen dituzten galdera konjuntiboen batura bada.
- Edozein datu-basetarako, Q' berridazketaren erantzun multzoa Q-ren erantzun multzoaren azpimultzoa bada ($ANS(Q', D) \subseteq ANS(Q, D)\ \forall D$)
- Aurreko bi baldintzak bete eta, halaber, Q' galdera barnean duen Q'' galdera positiborik ez badago, $Q'' \equiv Q'$ ez bada.

□

[15]Maximoki barne-harturiko berridazketak galdeketa-lengoaia baten arabera definitu behar dira.

Definizio hauek galdera baten berridazketaren bat benetan jatorrizko galderaren baliokideak direla jakiten lagunduko digute eta, bide batez, itzulpenak gauzatzen dituzten algoritmoen ontasuna neurtzeko eredu teorikoa eskainiko digute. Hurrengo atalean, LAV ereduan galderak itzultzeko zenbait algoritmo ikusiko ditugu baina, aurretik, zenbait gogoeta aterako ditugu plazara.

Eredu orokor baten arabera deskribatutako bisten definizio multzoak ezagututa galdera baten itzulpen *guztiak* aurkitzearen arazoa oinarrizkoa da, bistetan oinarritutako galderen erantzutearen arloan. Izan ere, galdera baten maximoki barne-harturikoak diren berridazketak ez dira beti itzulpen posible guztiak izango (Levy, 2000). Honen arrazoi intuitiboa maximoki barne-harturiko berridazketek lengoaia batekiko duten mendekotasunean aurki daiteke: gerta daiteke adierazpen-ahalmen handiagoa duen lengoaiaren batean adierazitako galderaren batek erantzun kopuru handiagoa eskaintzea.

Bisten multzoak emanda, galdera baten berridazketa guztiak itzultzearen arazoa *erantzun ziurrak* ("certain answers") delakoan formalizaturik dago (Abiteboul eta Duschka, 1998). Lan horretan emandako definizioan, *mundu itxiaren asuntzioa* "Closed World Assumption (CWA)" eta *mundu irekiaren asuntzioa* "Open World Assumption (OWA)" direlako terminoak bereizten dira: mundu itxiaren asuntzioan, bistaren hedapenak osoak direla suposatzen da, hau da, datu-basean bista aplikatu ondoren lortutako n-kote guztien adierazpenak direla. Mundu irekiaren asuntzioan, aitzitik, bisten hedapenak ez dituzte n-kote guztiak adieraziko (baina ez dute n-kote okerrik adieraziko). Mundu irekiaren asuntzioa egokia da datu-integrazioaren arloan, arestian esan dugun bezala, ezin baita ziurtatu datu-iturriek mota jakin bateko informazio guztiaz hornitzeko aukera emango digutenik. Mundu itxiaren asuntzioa, berriz, galdera-optimizazioan edo datuen independentzia fisikoa mantentzea helburu duen arloan erabiltzen da.

III.3.6 Definizioa Izan bitez Q galdera konjuntiboa eta datu-basearen $R_1, \ldots, R_n$ eskemaren gaineko $\mathcal{V} = V_1, \ldots, V_m$ bista konjuntiboen multzoa. Izan bedi $v_1, \ldots, v_m$ n-kote multzoa $V_1, \ldots, V_m$ bisten hedapena, hurrenez hurren.

$v_1, \ldots, v_m$ aldez aurretik jakinda, $\bar{a}$ n-kotea Q galderaren erantzun ziurra izango da mundu itxiaren asuntzioan, baldin $\bar{a} \in Q(D)$ bada edozein D datu-basetarako, non $V_i(D) = v_i$, edozein i-rako, $i \leq i \leq m$.

$v_1, \ldots, v_m$ aldez aurretik jakinda, $\bar{a}$ n-kotea Q galderaren erantzun ziurra

izango da mundu irekiaren asuntzioan, baldin $\bar{a} \in Q(D)$ bada edozein D datu-basetarako, non $V_i(D) \supseteq v_i$, edozein i-rako, $i \leq i \leq m$. □

Intuitiboki, definizio honen esanahia honakoa da: bista multzo baten hedapenak ez du datu-baseko erlazioen hedapen bakarra definitzen. Hortaz, bista baten hedapena ezagutzen badugu, datu-basearen egoera errealari buruzko informazio partziala dugu soilik. N-kote batek, Q galderaren erantzun ziurra izango bada, bisten definizioen hedapenekin bat datozen datu-baseko egoera posible guztien erantzuna izan behar du. Ikus dezagun, adibide baten bidez, galdera baten emaitza desberdina izan daitekeela, baldin mundu irekiaren asuntzioa edo itxiarena onartzen den (Abiteboul eta Duschka-ren (1998)-ko artikuluan aurki daiteke adibide hau).

III.3.5 Adibidea Izan bitez $q(x, y) \leftarrow p(x, y)$ galdera konjuntiboa, eta bi bista, hots, $v_1(x) \leftarrow p(x, y)$ eta $v_2(y) \leftarrow p(x, y)$. Demagun bisten hedapena honako hau dela: $\{v_1(a), v_2(b)\}$. Mundu irekiaren asuntzioa hartzen bada, bisten definizioek adierazten dutena zera da: badago bere lehenengo osagai bezala a balioa duen p n-kote bat, eta beste p n-kote bat (beharbada desberdina), zeinek bere bigarren osagai bezala b balioa duen. Hortaz, q galdera ezin dugu erantzun. Mundu itxiaren asuntzioa hartuz, baina, bisten informaziotik jakin dezakegu p n-kote guztiek a balioa dutela lehenengo osagai bezala, eta b balioa bigarren osagai bezala. Beraz, q galderaren erantzun ziurra jakin daiteke: p n-kote bakar bat, hots $< a, b >$.

Bistez baliatutako galderen erantzunaren arazoa $\mathcal{NP}$-osoa da galdera zein bisten azpihelburuen kopuruarekiko, nahiz eta bistak adierazteko galdera konjuntiboak erabiltzen badira (Chandra eta Merlin, 1977; Levy *et al.*, 1995). Arazoa zaila dugu, beraz, adierazpen-ahalmen eskaseko lengoaietarako ere. Adierazpen-ahalmen aberatsagoa duten lengoaietarako —*datalog* erregela errekurtsiboak, esate baterako—, bestalde, itzulpenak lortzearen arazoa zailagoa edo erabakiezina bilakatuko da. Abiteboul eta Duschka-ren lanak (1998) erantzun ziurrak aurkitzearen arazoak duen konplexutasuna du aztergai, bisten definizioak eta galderak adierazteko lengoaia desberdinetarako.

III.3.3.2 Algoritmoak.

Levy *et al.*-en lanak (1995) frogatzen du, Q galdera eta $\mathcal{V}$ bistak adierazteko desberdintasun-murriztapenik gabeko galdera konjuntiboak erabiltzen

badira, maximoki barne-harturikoa den edozein berridazketak jatorrizko galderaren azpihelburu kopuru bera edo gutxiago beharko dituela. Frogapen honek algoritmo *sinple* bat asmatzeko aukera ematen digu: jatorrizko galderaren azpihelburu kopurua baino gehiago izan gabe, $\mathcal{V}$-ko predikatuekin soilik osatutako Q' galderak asmatu, eta Q' itzulpena Q galderaren baliokidea den egiaztatu. Algoritmo hau batere eraginkorra ez bada ere, azpimarratzekoa da itzulpenak denbora finituan asmatuko lituzkeela, azken finean, bilaketa-espazioa finitua baita, Q galderaren azpihelburuen kopuruak murriztua. Dena dela, espazioa esponentzialki handituko da galderaren azpihelburuen kopuruarekiko.

Algoritmo *sinple* horren emaitzak hobetzen dituzten algoritmoak garatu izan dira, horietatik *Bucket* algoritmoa, Information Manifold sisteman erabilia (Levy *et al.*, 1996), *Inverse Rules* delakoa, InfoMaster sistemakoa (Duschka eta Genesereth, 1997a), *Shared-Variable Bucket* (Mitra, 1999) edo *MiniCon* algoritmoa (Pottinger eta Levy, 2000) direlarik usuen erabilitakoak.

Bucket algoritmoa.

Bucket algoritmoaren xedea erabiltzaileak bitarteko-eskema (birtual) baten gainean eginiko galderen birformulaketa lortzean datza, galdera bera eskura dauden datu-iturrien eskemaren arabera egon dadin (Levy *et al.*, 1996). Datu-iturriak zein galderak adierazteko galdera positiboak (galdera konjuntiboen bildurak) erabiltzen ditu, galdera konjuntibo hauek desberdintasun-murriztapenak eduki ditzaketelarik. AT&T enpresako *Information Manifold* (IM) integrazio-sistemaren inguruan garatua izan zen *bucket* algoritmoak jatorrizko galderaren barruan dauden —ez dute ezinbestean baliokideak izan behar— berridazketa guztiak kalkulatzen ditu. Bere ekarpen nagusia berridazketa posible guztien espazioa[16] "kimatzea" da: berridazketa posibleen kopurua jaistearren, algoritmoak galderaren azpihelburu bakoitza bere horretan hartzen du eta, azpihelburu bakoitzeko, iturburu-erlazio aproposak aukeratu eta pertz (*bucket*) batean sartuko ditu.

Zehatzago: izan bedi $Q = g_1, \ldots, g_n$ galdera konjuntibo bat. *Bucket* algoritmoak bi urratsetan betetzen ditu eginbeharrekoak. Lehenengo urratsean (ikus algoritmo 1), g_i azpihelburu bakoitzeko aproposak diren bistak aukeratuko ditu eta pertz batean sartuko ditu. $V = v_1, \ldots, v_m$ bista bat aproposa izango da g_i azpihelburu baterako, baldin V bistan v_j azpihelburu bat bada-

[16] Aurreko atalean azaldutako algoritmo sinplearen bilaketa-espazioa.

Algoritmoa 1 *Bucket* algoritmoa. Lehenengo urratsa.

{$\mathcal{V} \rightarrow$ eduki-deskribapenen multzoa }
{$Q \rightarrow$ Galdera konjuntiboa, honako forma honekin
$Q : Q(\bar{X}) \leftarrow g_1(\bar{X}_1), \ldots, g_m(\bar{X}_m).$}
egin $1 \leq i \leq m$ guztietarako
$Bucket_i \leftarrow 0$
amaiera egin
egin $1 \leq i \leq m$ guztietarako
egin $v \in \mathcal{V}$ bakoitzeko
Izan bezan v-k honako forma hau:
$v(\bar{Y}) \leftarrow v_1(\bar{Y}_1), \ldots, v_n(\bar{Y}_n)$
egin $1 \leq j \leq n$ guztietarako
baldin $g_i = v_j$ edo $g_i \cap v_j \neq \emptyset$ **orduan**
Izan bedi ψ, v aldagaien gainean definitutako mapaketa, ondoko eran:
baldin y aldagaia, $\bar{Y}_j$-ren j-garren aldagaia bada, eta $y \in \bar{Y}$ **orduan**
$\psi(y) = x_j$, non x_j aldagaia $\bar{X}_i$-ren j-garren aldagaia den
bestela
$\psi(y)$, Q-n edo v-n agertzen ez den aldagai berria da
amaiera baldin
amaiera baldin
amaiera egin(guztietarako)
gehitu $v(\psi(\bar{Y}))$ elementua $Bucket_i$-ari
amaiera egin(bakoitzeko)
amaiera egin(guztietarako)
Idatzi-Berridazketak (*Bucket*)

go, non g_i eta v_j predikatuen arteko θ bateratzailerik dagoen (ikus Abiteboul *et al.* (1995)). Baldintza betetzen bada, $\theta(head(V))$ predikatua g_i-ri dagokion pertzean sartuko da, non $\theta(head(V))$ V erregelaren buruko aldagaien gainean θ bateratzailea aplikatu ondoren sorturiko predikatua den, aldagai *askeak* eduki ditzakeena. g_i azpihelburua V bista batekin behin baino gehiagotan batera badaiteke, V hainbat alditan azalduko da azpihelburuari dagokion pertzean.

Algoritmoaren bigarren urratsean (ikus algoritmo 2), pertzaren biderketa kartesiarraren gainean erabateko bilaketa burutuko da, eta konbinazio posible bakoitzaren bateragarritasuna egiaztatuko da. Azkenik, bateragarriak diren itzulpen guztietarako, itzulpen horiek jatorrizko galderaren barruan dauden egiaztatu behar da: demagun Q galdera eta Q' *bucket* algoritmoaren bitartez

Algoritmoa 2 *Bucket* algoritmoaren sasi-kodea. Bigarren urratsa

Prozedura Idatzi-Berridazketak
{$Bucket \leftarrow B_1, \ldots, B_m$ lehen urratsean eraturiko Bucket-a}
egin $Z \in B_1 \times B_2 \times \ldots \times B_m$ bakoitzeko
  Izan beza Z-k honako forma hau: $Z(\bar{X}) \leftarrow R_1, \ldots, R_m$.
  Izan bedi $H \leftarrow (R_1)^{hed}, \ldots, (R_m)^{hed}$
  baldin $Betegarria(H)$ **orduan**
    Gehitu H *Berridazketak*-era
  bestela
    $H' = SaiatuBetegarriaEgiten(H)$
    baldin $H' \neq \emptyset$ **orduan**
      Gehitu H' *Berridazketak*-era
    amaiera baldin
  amaiera baldin
amaiera egin
Itzuli *Berridazketak*

lorturiko itzulpena. Itzulpena egokia izango da baldin $(Q')^{hed} \sqsubseteq Q$. Berridazketa egokia ez bada, *bucket* algoritmoak azken saiakera egingo du eta, horrela, berridazketari zenbait desberdintasun-murriztapen gehitzen saiatuko da, berridazketa bera jatorrizko galderaren barruan egon dadin. Azken prozesu hau arrakastatsua ez bada, berridazketa bertan behera utziko da.

Ikus dezagun adibide bat:

III.3.6 Adibidea III.5 irudian informazio-integrazioko egoera tipikoa aurkezten da, non hitzei buruzko informazioa gordetzen duen S1 datu-basea eskema orokor batean integratu nahi den. Eskema orokorrak hitz polisemikoei buruzko informazioa adieraz dezakeen bitartean (hitz-forma bat hainbat adierarekin erlaziona daiteke, eta adiera bakoitzak bere definizioa izango du), S1 datu-baseak hitz monosemikoak soilik gordeko ditu (hitz-forma bakoitza definizio bakar batekin erlazionaturik dago). LAV hurbilpenari jarraituz, erregela hauek idatziko ditugu:

$$
\begin{array}{rcl}
S1\_Unit(u) & \leftarrow & Adierak(u). \\
S1\_forma(u, hf) & \leftarrow & Adierak(u), forma(u, f), \\
 & & Hitzak(f), hForma(f, hf). \\
S1\_def(u, def) & \leftarrow & Adierak(u), def(u, def), \\
 & & Definizioak(def). \\
S1\_Def(def) & \leftarrow & Definizioak(def). \\
S1\_gloss(def, g) & \leftarrow & Definizioak(def), testua(def, g).
\end{array}
$$

Demagun "lagun" hitzaren definizioari buruzko galdera egiten dela, hots:

$$
\begin{array}{rcl}
Q(defT) & :- & Adierak(a), forma(a, f), Hitzak(f), \\
 & & hForma(f, "lagun"), def(a, d), testua(d, defT).
\end{array}
$$

Lehenengo urratsean, algoritmoak pertz bat eraikiko du Q-ren azpihelburu bakoitzeko:

$Adierak(a)$	$forma(a, f)$	$Hitzak(f)$
$S1\_Unit(a)$	$S1\_forma(a, \_)$	$S1\_forma(\_, \_)$
$S1\_forma(a, \_)$		
$S1\_def(a, \_)$		

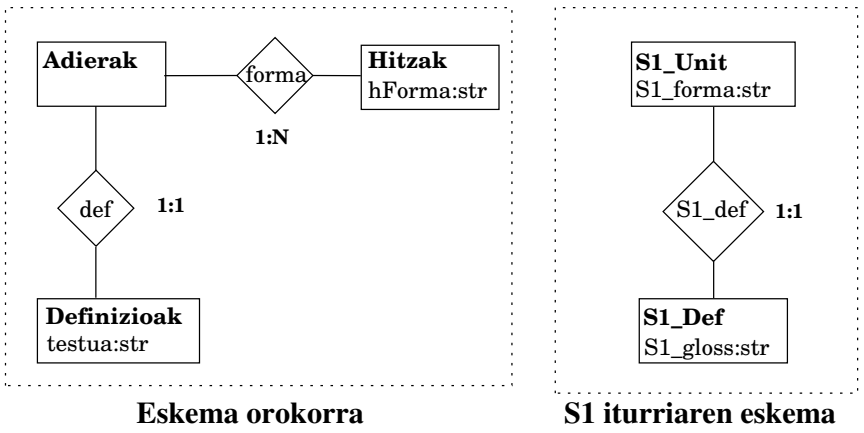
$def(a, d)$	$testua(d, defT)$	$hForma(f, '' lagun'')$
$S1\_def(a, d)$	$S1\_gloss(d, defT)$	$S1\_forma(\_, '' lagun'')$

Bigarren urratsean, pertz bakoitzeko elementu bat aukeratuz lortutako berridazketak bateratzen saiatuko da (kasu honetan, 3 konbinazio posible daude). Hala ere, konbinazio horietatik soilik berridazketa hauek dira zilegi:

$$
\begin{array}{rcl}
Q'(defT) & :- & S1\_Unit(a), S1\_forma(a, '' lagun''), \\
 & & S1\_def(a, d), S1\_gloss(d, defT). \\
Q'(defT) & :- & S1\_forma(a, '' lagun''), S1\_def(a, d), \\
 & & S1\_gloss(d, defT).
\end{array}
$$

eta, predikatu erredundanteak ezabatuz, jatorrizko galderaren berridazketa dugu:

$$
\begin{array}{rcl}
Q'(defT) & :- & S1\_forma(a, '' lagun''), S1\_def(a, d), \\
 & & S1\_gloss(d, defT).
\end{array}
$$



III.5 Irudia: Eskema orokorra eta S1 iturriarena

Erraz egiazta daiteke Q' itzulpena Q galderaren maximoki barne-harturiko berridazketa dela. Izan ere, galderaren hedapena honako hau baita:

$$\begin{array}{rcl}(Q')^{hed}(defT) & :- & Adierak(a), forma(a,f), hForma(a,''lagun'') \\ & & Adierak(a), def(a,d), Definizioak(d), \\ & & testua(d,defT).\end{array}$$

eta, hortaz, jatorrizko Q galderaren baliokidea. □

Nabarmentzekoa da, hala ere, III.3.6 adibidean berridazketaren erantzuna ez dela izango, halabeharrez, jatorrizkoarena. Izan ere, S1 datu-baseak hitz monosemikoei buruzko informazioa gordetzen du soilik (eta, apika, hitz polisemikoentzat soilik adiera usuena gorde dezake). Hortaz, erabiltzaileak, hitz polisemiko bati buruzko definizioak eskatzerakoan, ez ditu akaso espero zituen erantzun guztiak lortuko. Hala ere, ziurtatzen zaio jasoko dituen erantzunak, nahiz eta erantzun posible *guztiak* ez izan, ez direla ere okerrak izango.

Alderantzizko erregelen algoritmoa (Inverse rules algorithm).

Bucket algoritmoa bezala, alderantzizko erregelen algoritmoa datu-integraziorako diseinatu zen, Infomaster proiektuaren barruan (Duschka eta Genesereth, 1997a; Duschka *et al.*, 2000). Alderantzizko erregelen algoritmoak *datalog* ez-errekurtsiboan adierazitako galderak zein bisten definizioak onar-

tzen ditu sarrera gisa, eta, hortaz, *bucket* algoritmoak baino espresibotasun aberatsagoa onartzen du.

Algoritmo honen ideia nagusia zera da: itzuli behar den galdera bat izanik, galdera horren baliokidea den programa logikoa eraikitzea, eta programa hori exekutatzea, galderaren itzulpena lortzeko. Programa logikoa eraiki ahal izateko, bisten definizioak *inbertitu* behar dira, hots, bisten n-koteetatik datu-baseko erlazioen n-koteak lortzeko adierazten dituzten erregelak idatzi. Adibidez, hitz-formak eta aldaerak lotzen dituen bista hau badugu:

$$formaAldaerak(f, ald) \leftarrow Forma(kontz, f), Aldaerak(kontz, ald).$$

Honako programa logiko hau osatuko da :

$$\begin{array}{lll} Forma(Z = f_1(f, ald), forma) & \Leftarrow & formaAldaerak(f, ald). \\ Aldaerak(Z = f_1(f, ald), ald) & \Leftarrow & formaAldaerak(f, ald). \end{array}$$

Alderantzizko erregelen esangura honako hau da: $formaAldaerak$ bistako hedapenekoa den (f, ald) n-kote bat $Forma$ eta $Aldaerak$ erlazioen hedapeneko zenbait n-koteren *lekukoa* da, bi gauza adierazten baitizkigu:

1. $Forma$ erlazioak (Z, f) n-kote bat edukiko du, Zren balio baterako.
2. $Aldaerak$ erlazioak (Z, ald) n-kote bat edukiko du Z balio *bererako*.

Alderantzizko erregela bietan Z-ren balioa bera dela adierazteko $f_1(f, ald)$ *Skolem funtzioa* erabiltzen da, Z balioa f eta ald aldagaien funtzioa baita.

Jatorrizko galdera erantzuteko galdera bera eta bistak adierazten dituzten alderantzizko erregelen konbinaketa programa logikoa bezala exekutatuko da, behetik gorako estrategia jarraituz. Exekuzioa bukatuko dela bermatua dago, eta erabiltzaileari Skolem funtziorik ez duten berridazketak soilik igorriko zaizkio.

Bucket algoritmoak eta alderantzizko erregelen algoritmoak badute antzik. Izan ere, pertzak lortzeko urratsak zerikusi handia du alderantzizko erregelen osaeraren prozesuarekin: biak datu-baseko erlazioekin zerikusia duten bistak lortzen dituzte. Alde handiena da *bucket* algoritmoak azpihelburu baterako bista aproposak lortzeko azpihelburu horrek galderan duen testuingurua kontuan hartzen duela, alderantzizko erregelen algoritmoak ez bezala. Haatik, alderantzizko erregelen algoritmoak behin bakarrik lortu behar ditu bisten definizioen alderantzizko erregelak eta, ondoren, erregela horiek edozein galderatarako baliagarriak izango dira. Horretaz gain, alderantzizko

erregelen algoritmoa hedagarriagoa da *bucket* algoritmoa baino eta, arestian aipatu bezala, galdera konjuntiboez gain, *datalog* erregelak ere erabil daitezke galdera zein bistak adierazteko. Nolanahi ere, *datalog* erregelak erabiltzeak algoritmoaren konputazio-zama nabarmenki areagotuko du. Bestela, algoritmoaren konplexutasuna polinomiala da galdera zein bisten azpihelburuen kopuruekiko.

MiniCon algoritmoa.

MiniCon algoritmoak (Pottinger eta Levy, 2000) ere bi urratsetan egingo du lan. Lehenengo urratsean, *bucket* algoritmoak bezala, galderaren azpihelburu bakoitzerako aproposak izango diren bistak aukeratzeari ekiten dio. Hala ere, jatorrizko galderan predikatu bat baino gehiagoren artean banatuta dauden aldagaietan —*shared variables* direlakoak— jarriko du arreta.

Demagun, berriro ere, $\mathcal{V}$ bisten definizioak, $V \in \mathcal{V}$ bista bat, eta $Q = g_1, \ldots, g_n$ erabiltzaileak jarritako galdera bat ditugula, denak galdera konjuntiboen bidez adierazita. g_i azpihelbururen bat V bista batekin ordezkatu ahal izateko, honako baldintza hauek bete behar dira:

- Badaude g_i eta $v_j \in V$ predikatuak, non bien artean θ bateratzailerik defini daitekeen;
- θ mapaketak g_i predikatuko x aldagairen bat bistako definizioan nabarmendua ez den[17] aldagai batekin erlazionatzen badu, x aldagaia duen g_j galderako predikatu guztiek ere bista berarekin bateragarriak izan behar dute, θ bateratzaile beraren bidez[18].

Aurreko azalpena hobeto ulertuko da adibide baten bidez[19]:

III.3.7 Adibidea Eman ditzagun eskema orokorraren araberako erlazioen bitartez adierazitako bista hauek ditugula:

$$\begin{array}{rcl} DefZirkular(a) & \leftarrow & def(a,b), def(b,a). \\ sin(c,d) & \leftarrow & sinonimoak(c,d). \\ defZirkSin(f,h) & \leftarrow & def(f,g), def(g,h), sinonimoak(f,g). \end{array}$$

[17] Ikus 92. orria.

[18] Murriztapen hau *Shared-Variable Bucket algorithm* delakoan ikus dezakegu estreinako aldiz. Ikus (Mitra, 1999).

[19] (Pottinger eta Levy, 2000) lanean azaldutako adibidean oinarritua.

Lehenengoak adierazten ditu definizio zirkularrak dituzten hitzak, hots, a hitzen multzoa zeintzuen definizioetan b hitza agertzen den eta b-ren definizioan, berriz, a hitza agertzen den. Bigarren bistak sinonimo-bikoteak adierazten ditu. Azkeneko bistak, berriz, (f, h) hitz-bikoteak, non f-ren definizioan g hitz bat azaltzen den, f eta g sinonimoak izanik, eta g-ren definizioan, berriz, h hitza agertzen den.

Demagun sinonimoak izanda definizio zirkularra duten hitzei buruzko galdera egiten dela, hots:

$$Q(x) \quad :\text{-} \quad def(x,y), def(y,x), sinonimoak(x,y).$$

□

Bucket algoritmoak, bere lehenengo urratsean, pertz bat eraikiko luke Q-ren azpihelburu bakoitzeko:

$def(x,y)$	$def(y,x)$	$sinonimoak(x,y)$
$DefZirkular(x)$	$DefZirkular(x)$	$sin(x,y)$
$defZirkSin(x,\_)$	$defZirkSin(\_,x)$	$defZirkSin(x,\_)$

MiniCon algoritmoak, aitzitik, ez du $DefZirkular$ bista sartuko $def(x,y)$ eta $def(y,x)$ azpihelburuei dagozkien pertzetan, eta ez du egingo arrazoi honengatik: demagun galderako $def(x,y)$ azpihelburua ordezkatzeko $DefZirkular$ bista erabili nahi dela. Kasu horretan, $\{x \to a, y \to b\}$ baliokidetasuna edukiko genuke, y eta bistan nabarmendua ez den b aldagaien arteko baliokidetza dagoelarik:

$$\begin{array}{rlllll} Q(x) & :\text{-} & def(x, & y), def(& y,x), sinonimoak(x, & y). \\ & & & \downarrow & \downarrow & ? \\ DefZirkular(a) & :\text{-} & def(a, & b), def(& b,a). & \end{array}$$

Horrela, bada, $def(x,y)$ azpihelburua $DefZirkular(a)$ bistarekin ordezkatzen bada, $sinonimoak(x,y)$ azpihelburua umezurtz geratuko da, hau da, ezin izango da jatorrizko galderan adierazitako $def(x,y)$-ren eta $sinonimoak(x,y)$ arteko elkarketarik (*joinik*) gauzatu:

$$Q'(x) \text{ :- } DefZirkular(x), sinonimoak(x,y) \qquad ???$$

Froga daiteke, beraz, $DefZirkular$ bista ezin dela inoiz erabili def jatorrizko galderako azpihelburua ordezkatzeko. *Bucket* algoritmoa, ordea, ez

litzateke honetaz jabetuko algoritmoaren bigarren urratsa arte. Bestalde, algoritmo biek itzuliko dute galderaren berridazketa zilegi bakarra, hots:

$$Q'(x) \text{ :- } defZirkSin(x, x)$$

MiniCon algoritmoa *MiniCon deskribapena* (MCD) delakoan oinarritzen da. MCD bakoitza galderako aldagaien azpimultzo batetik bista baten aldagaietara doan mapaketa da, jatorrizko galderatik berridazketara doan *barnehartzearen mapatze* baten zatia adierazten duelarik:

III.3.7 Definizioa Demagun Q galdera bat eta V bista bat ditugula. C MCD bat $\left\langle V(\bar{Y})_C, h_C, \varphi_C, G_C \right\rangle$ n-kote bat da, non:

- h_C, V bistaren gaineko buru-homomorfismoa[20] den
- $V(\bar{Y})_C, V$ bistaren gainean h_C buru-homomorfismoa aplikatu ondoren sortutako predikatua den
- $\varphi_C : Vars(Q) \rightarrow h_C(Vars(V))$ mapaketa partziala den, Q galderaren zenbait aldagai $h_C(Vars(V))$ bistaren gainean h_C buru-homomorfismoa aplikatu ondoren sorturiko aldagaiekin lotzen dituena
- G_C, Q galderaren zenbait azpihelburu

□

Algoritmo 3an MCDak osatzen duen algoritmoaren lehenengo urratsaren sasi-kodea dugu. Hor aipatutako 1. propietatea honako hau da:

- $g \in G_C$ azpihelburu bakoitzeko, badago $v \in body(V)$ predikatua, non $\varphi_C(g) \in h_C(v)$ (hau da, G_C multzoko azpihelburu bakoitza V bistako predikaturen batekin bateratuko da, bistaren gainean h_C buru-homomorfismoa aplikatu ondoren).

[20] V bista baten gaineko h buru-homomorfismoa $h : Vars(V) \rightarrow Vars(V)$ aldagaien gainean eginiko mapaketa da, existentzialki kuantifikatutako aldagaientzat identitatea dena baina buruko aldagaien (aldagai nabarmenduak) arteko baliokideak defini ditzakeena; zehatzago, $x \in Vars(Head(V))$ aldagai nabarmendua bere buruarekin ordezkatuko da $h(x) = x$ edo beste y aldagai nabarmendu batekin, $h(x) = y$, non y aldagairako h mapaketa identitatea den; hau da,

$$h(x) = y, \quad y \in Vars(Head(V)), \quad h(y) = y$$

Algoritmoa 3 *MiniCon* Algoritmoa. MCDak osatzen

$\{\mathcal{V} \rightarrow$ eduki-deskribapenen multzoa $\}$
$\{Q \rightarrow$ Galdera konjuntiboa $\}$
$\mathcal{C} = \emptyset$
egin $g \in Q$ azpihelburu bakoitzeko
  egin $V \in \mathcal{V}$ erregela bakoitzeko, eta $v \in V$ predikatu bakoitzeko
    Izan bedi h, V-ko buru-homomorfismoen artean gutxien murrizten duena, eta φ mapaketa bat, non $\varphi(g) = h(v)$ betetzen den.
    baldin h eta φ existitzen badira **orduan**
      gehitu $\mathcal{C}$-ri ondoko baldintza hauek betetzen dituzten MCD berri guztiak:
      (a) φ_C (edo h_C), φ (edo h) mapaketen hedapena da,
      (b) G_C, Q galderaren azpihelburuen multzo txikiena da, non $G_C - k, \varphi_C$-k eta h_C-k 1. propietatea betetzen duten, eta
      (c) ezin da φ eta h mapaketak φ'_C eta h'_Cra hedatu, non (b) baldintza betetzen den, eta (b)-n definitutako G'_C, G_C-ren azpimultzoa den.
    amaiera baldin
  amaiera egin(bakoitzeko)
amaiera egin(bakoitzeko)
Itzuli $\mathcal{C}$

- φ_C mapaketak galderako aldagai nabarmenduak bistako aldagai nabarmenduekin ordezkatuko ditu.

- φ_C mapaketak $x \in Vars(Q)$ galderako aldagai ez-nabarmenduren bat bistako aldagai ez-nabarmendu batekin lotzen badu, x aldagaia duen Q galderako p predikatu orok G_C multzoan egon behar du.

- G_C aurreko baldintzak betetzen dituen Q-ren azpihelburuen multzo *minimoa* da, hau da, aurreko bi baldintzak betetzen dituen eta G_C baino azpihelburu gutxiago dituen G'_m-rik ez dago.

- h_C galdera eta bista bateratzen dituzten buru-homomorfismoen artean gutxien murrizten duena da.

Behin MCD guztiak lortu eta gero, horien konbinaketatik etorriko dira berridazketa guztiak. Ikusi dugunez, MCD bakoitzak jatorrizko galderako hainbat azpihelburu estaliko ditu, G_C multzoaren bitartez. Horrela, bada, itzulpenak lortzeko jatorrizko galderako azpihelburu guztiak estaltzen dituzten MCDen konbinaketak behar dira. Haatik, MCDak osatzerakoan erabili den arreta

Algoritmoa 4 *MiniCon* Algoritmoa. Itzulpenak idazten.

{$\mathcal{C} \rightarrow$ algoritmoaren lehenengo urratsean sorturiko MCDak }
{MCD bakoitzaren forma $(h_C, V(\bar{(}Y)), \varphi_C, G_C, EC_C)$ da }
$C_1, \ldots, C_n$ MCDen multzoa emanda, $Vars(Q)$ gainean EC funtzioa definituko dugu, ondoko eran:
baldin $i \neq j$rako, $EC_{\varphi_i}(x) \neq EC_{\varphi_j}(x)$ bada **orduan**
  Izan bedi $EC_C(x)$ bietako bat, baina $EC_{\varphi_i}(y) = EC_{\varphi_i}(x)$ betetzen duten y-ekin kontu berezia izanik
amaiera baldin
$Erantzuna \leftarrow \emptyset$
egin $C_1, \ldots, C_n$, $\mathcal{C}$-ko azpimultzo bakoitzeko, non $G_{C_1} \cup G_{C_2} \cup \ldots \cup G_{C_2} = azpigalderak(Q)$ eta $i \neq j$ bakoitzeko, $G_{C_i} \cap G_{C_j} = \emptyset$
  $\bar{Y_i}$ aldagaien gainean, Ψ_i mapaketa definitu, ondoko eran:
  baldin badago $x \in Q$ aldagairik, non $\varphi_i(x) = y$ **orduan**
    $\Psi_i(y) = x$
  bestela
    $\Psi_i(y)$ y-ren kopia berria da
  amaiera baldin
  Sortu honako berridazketa konjuntiboa:
  $Q'(EC(\bar{X})) \leftarrow V_{C_1}(EC(\Psi_1(\bar{Y_1}))), \ldots, V_{C_n}(EC(\Psi_n(\bar{Y_n}))).$
  $Erantzuna \leftarrow Erantzuna \cup Q'$
amaiera egin
Itzuli $Erantzuna$

berezia dela eta, MCDen arteko konbinaketak berezia izan behar du: Q galdera baterako eta $\mathcal{V}$ bisten definizioetarako $\mathcal{M}$ *MiniCon deskribapenen* multzoa emanda, maximoki barne-harturikoak diren berridazketak $\mathcal{M}$-ko $m_1, \ldots, m_k$ elementuak hartuz lortuko dira, non beren $G_{m_1}, \ldots, G_{m_k}$ multzoa jatorrizko galderaren partiketa disjuntua den, hau da, $G_{m_1} \cup \ldots \cup G_{m_k} = body(Q)$ eta $G_{m_i} \cap G_{m_j} = \emptyset$ edozein $i, j = 1, \ldots, k$ bikotetarako, non $i \neq j$. Ezaugarri honek berridazketa kopurua nabarmenki murriztuko du galderaren itzulpenak lortzeko garaian. Algoritmo 4k bigarren urratsari ekiten dio.

Adibidez, III.3.6 adibideko galderaren berridazketak lortzerakoan, MCD hauek osatuko lirateke:

$\mathcal{M}$	$V_C(\bar{Y})$	h_C	φ_C	G_C
m_1	$sin(c,d)$	$c \rightarrow c, d \rightarrow d$	$x \rightarrow c, y \rightarrow d$	g_3
m_2	$defZirkSin(f,f)$	$f \rightarrow f, g \rightarrow f$	$x \rightarrow f, y \rightarrow g$	g_1, g_2, g_3

non g_i jatorrizko galderaren i. predikatua den.

MiniCon algoritmoak *bucket* algoritmoaren bigarren urratsean eginiko lanaren zati bat MCDen osatze-urratsera lerratzen du eta, hortaz, bigarren urratsean aintzakotzat hartu beharreko berridazketa kopurua nabarmenki jaisten da. Ikusitako adibidean *bucket* algoritmoak 3 konbinazio kontuan hartzen zituen bitartean —eta, konbinazio bakoitzeko, konputazionalki garestia den galderen barne-hartzea ebatzi beharra—, *MiniCon* algoritmoak soilik partiketa bakarra behar du eta, MCDak osatu diren modu berezia dela medio, sortutako berridazketa maximoki barne-hartua dela ziurta daiteke.

III.3.4 Galdeketa-gaitasunak.

Informazio-iturriek galdeketarako interfaze murritzak izan ditzakete. Horrela, zenbait *web* orritatik edo sistema propioetatik informazioa eskuratzeko, atributu jakin batzuetatik soilik galdetu ahal izango da.

Murriztapen hauek aldaketa sakonak eragiten dituzte bistetan oinarritutako galderak erantzutearen arazoan. Murriztapenak formalizatzeko *bete beharreko patroiak* ("binding patterns") erabili ohi dira (Rajaraman *et al.*, 1995). Patroi hauen bidez, iturriek dituzten galdeketa-gaitasunak adieraz daitezke.

Konparazio batera, on-line hiztegiek hitzen definizioak eskaintzen dituzte maiz, baina datuak eskuratzeko hitzaren forma eman egin behar zaie. Horrela, ezin izango da, definizio bat emanda, definizio hori zein hitz-formari dagokion galdetu, adibidez. Galdeketa-gaitasunak galdera konjuntiboen bidez adieraz daiteke, LAV hurbilpenean, bete beharreko patroien bidez. Adibide honetan, honako hau izango dugu:

$$hitzaDefinizio^{bf}(hf, def) \leftarrow Hitza(hf), definizio(hf, def).$$

Ikusten denez, bistako definizioaren aldagai nabarmenduei b edo f gehitu zaizkio, lotutako parametroa (b, "bind parameter") edo parametro askea (f, "free parameter") izan daitekeela adieraziz, hurrenez hurren. Lotutako parametroek bistaren sarrera-parametroak izan behar dute ezinbestean.

Bete beharreko patroien bidez definitutako bisten gaineko galderaren itzulpen-prozesua konplexuagoa da, zenbait berridazketak ez baitituzte parametroen gainean ezarritako baldintzak beteko. Izan ere, galdeketa-gaitasunen aurrean, galdera konjuntibo baten maximoki barne-harturiko berridazketa ezin da, oro har, galdera konjuntiboen multzo finitua izan, baina *datalog*

programa errekurtsiboen bitartez adieraz daitezke (Duschka eta Genesereth, 1997a; Duschka *et al.*, 2000). Gorago aipatu dugun bezala, baina, barne-hartzearen arazoa, galdera konjuntiboak ez diren galderetan, erabakiezina den arazoa da. Zenbait lanek ((Li eta Chang, 2000)-ek, adibidez), *datalog* programa errekurtsibo bereziak —*datalog* programa errekurtsibo *monodikoak* deiturikoak— sortzen dituzte galdera baten berridazketa optimoa lortzeko, eta lengoaia horrentzat barne-hartze arazoaren konputagarritasuna erabaki daitekeen arazoa dela frogatzen dute.

Galdera konjuntiboek duten espresibotasuna baino handiago duten lengoaia ahalmentsuetarako, azkenik, galdeketa-gaitasunen aurrean galderen barne-hartzea irekia dagoen ikerlerroa dugu oraindik (Vassalos eta Papakonstantinou, 2000).

III.3.5 Datu-arazketa.

Integrazio-sistema bateko iturri desberdinetan gorderiko informazioa elkartrukatu nahi bada, datuen arteko parekatzea burutu behar da, ezinbestean. Orain arte aztertutako integrazio-arazoak baliabideen eskema kontzeptualen gainekoak izan dira, hots, maila intentsionalean kokatuak.

Hala ere, atal honetan zehar ikusiko dugun legez, iturri lokaletik jasotako erantzunen gainean integrazio-arazoak sortu ohi dira, eta arazo hauei ere egin beharko zaio aurre. Konparazio batera, munduko entitate erreal bera adierazten duten bi instantzia baditugu —bi baliabide desberdinetatik jasoak—, hauek berberak edo, behintzat, baliokideak direla erabakitzeko ahalmena izan behar dugu, baliabide lokal bakoitzak entitate horri buruz duen informazioa erlazionatu ahal izateko.

Bestalde, gerta daiteke, baliabide lokaletan zehar, datu akastunak edo inkonsistentziak agertzea. Datu akastunen edo inkonsistentzien kopurua baliabide bakoitzaren izaeraren araberakoa izango da, ikusiko dugun bezala: zenbait baliabide era zorrotzean daude diseinatua, eta horien artean akatsak aurkitzea zaila da; beste batzuetan, ordea, diseinu askeagoa dutelako edo, datu akastun edo inkonsistenteak aurkitzeko aukera egongo da.

Estentsio mailan datuak parekatzeari —eta, beraz, integratzeari—, datu-arazketa ("Data Cleansing") esan ei zaio(Rahm eta Do, 2000; Galhardas *et al.*, 2000; Hernandez eta Stolfo, 1995; Monge, 1997; Tejada *et al.*, 2001), eta, behar-beharrezko prozesua da kalitatezko integrazioa burutu nahi bada: datuak arazteko teknikek integrazio-sistemaren kalitatearen neurria emango dute, hein handi batean, hainbat baliabidetako informazioa trukatzeko bi-

dea eskainiz. Horrela, bada, datuen arazketari ekin behar zaio kalitatezko integrazioa burutu nahi bada.

Informazio-biltegi bakar baten kalitatea aztertu nahi badugu ere, datu-mailako zein eskema-mailako arazoekin topo egin dezakegu. Eskema-mailako arazoak azalduko dira baliabideak integritate-murriztapenik ez duenean —dela ereduak onartzen dituen murriztapenak, dela aplikazio bereziek ezarritakoak—, edota dauden integritate-murriztapenak egokiak ez direnean. Arazo hauek, jakina, maizago agertuko dira eskema zehatza ez duten baliabideetan, hots, forma libreko testu-fitxategietan, *web* orrietatik erauzitako datuetan, etab. Nolanahi ere den, eskema-mailako arazoak integritate-murriztapenak urratzen dituzten datuetan azalduko dira: atributuen arteko dependentziak urratzen badira, gakoen balioak errepikaturik azaltzen badira, erreferentziazko integritate-murriztapena apurtzen bada, etab.

Horietaz aparte, eta betiere baliabide bakar baten informazioaz ari garela, datu-mailako arazoekin ere aurki gaitezke. Instantzia-mailakoak diren arazo horiek, oro har, eskeman aurreikusi edo eragotzi ezin diren inkonsistentziak daudenean azalduko dira: atributu baten balioa bete ez denean, edo balioa sartzerakoan sakatze-erroreak egon direnean, atributu batean informazio anitz txertatuta dagoenean (normalean testu-kateetan), etab. Horrelakoetan, baliabideak *datu zikinak* dituela esango da; horregatik, datu-baseetan egon daitezkeen datu akastunak zuzentzeari datu-garbiketa ("data cleaning") deitu ohi zaio.

Edonola ere, datu-base soil batekin aurki daitezkeen instantzia-mailako arazoak biderkatu egingo dira hainbat datu-base elkartu nahi direnean, berdin diolarik elkartu nahi diren datu-baseak sistema federatu, datu-base anizkoitz edota integrazio-sistema batekoak diren. Izan ere, datu-base bakoitzak eduki ditzakeen datu zikinez gain, adierazpen desberdinak erabil ditzakete mundu errealeko entitate bera erreferentziatzen duten objektuentzat. Hortaz, entitate bera ager daiteke datu-baseetan zehar bikoiztuta, bat ez datozen errepresentazioekin. Horretaz gain, gerta daiteke bi baliabide desberdinek —edo gehiagok— entitate beraren informazio teilakatua ematea, edo, baita ere, mundu errealeko entitate berari buruzko informazio kontraesankorra eskaintzea. Baliabide anitzetatiko datuen garbiketari objektuen identifikazio-arazoa ("object identity problem"), edo elkartu/purgatu arazoa ("merge/purge problem") esan ohi zaio (Galhardas *et al.*, 2000; Hernandez eta Stolfo, 1995; Monge, 1997; Tejada *et al.*, 2001).

Datu-baseetan zehar objektu bikoiztuak ezagutzea —eta, ondoren, bikoizketak ezabatzea— ez da gaur egungo arazoa. Monge-ren lanean (1997),

egileak gai honi buruzko 100 artikulu baino gehiago aurkitu ditu azken 50 urteetan zehar. Hala ere, algoritmo gehienak domeinu zehatzetarako diseinatuak direla azpimarratzen du. Azken urte hauetan, baina, domeinu askeko objektuen identifikazioan jarri da arreta, besteak beste, informazioaren integrazio-sistemak edozein domeinutakoak izan baitaitezke.

Besteak beste, arazo hauek aurki ditzakegu baliabide heterogeneoetatik lortutako datuetan (Galhardas *et al.*, 2001):

- Iturri desberdinetatik datozen datuak ohitura desberdinak dituen hainbat jendek eginda daude. Testuinguru honetan, entitate bera adierazten duten datuak ezagutzea ezinbestekoa da.

- Datuek formatu desberdinak eduki ditzakete. Estandarizazio-eza dela medio, hainbat eremu formatu desberdinetan gordeko dira baliabideetan, eta, horrela, izaera desberdineko informazioa txertatua ager daiteke (normalean, testu-segidetan).

- Erroreak egon daitezke datuetan. Arestian aipatu dugun bezala, sakatze-erroreak direla eta, zenbait datu akastunak izan daitezke.

- Inkonsistentziak ager daitezke. Nahiz eta erregistro desberdinek entitate beraren informazioa gordetzen dutela jakin, erregistro hauek informazio kontraesankorra eskain dezakete elkarrekiko.

- Eremu bereko datuek formatu desberdinak dituzte. Kategoria lexikalaren adierazpena da arazo honen adibide garbia. Nahiz eta kategoria lexikalak multzo itxia izan, iturri bakoitzak kodekera berezi bat erabiliko du metatutako hitzen kategoria adierazteko (hizkuntza desberdinetan adieraziak daudelako, laburdura desberdinak erabili direlako, eta abar).

Datu-arazketaren arazoei aurre egin behar zaie, taxuzko integrazio-sistema osatu nahi bada. Datu-mailako adosterik egin ezean, integrazio-sistemaren lana iturrietatik datuak jaso eta erabiltzaileari eskaintzea izango da, baina datu hauek lotu gabekoak izango dira: iturri bakoitzetik lorturiko informazioa uharte bat bezalakoa izango da, gainontzeko iturrietatik jasotako informazioarekin erkaezina. Informazioaren integrazioko sistemek informazio-uharteak elkarrekin erlazionatu egin beharko lituzkete, eta, horrela, erabiltzaileari informazio uniforme eta aberatsa eskaini, baliabideetatik lorturiko informazio osagarri guztiaz hornituz.

Rahm eta Do-ren artikuluan (2000), datuen arazketarako egun erabiltzen diren hurbilpenak aztertzen dira. Lan horretan, datu-arazketa bi eginkizunetan sailkatzen da: datuen garbiketa eta datuen transformazioa. Datuen garbiketa baliabide bakoitzaren gainean egiten da, eta bere helburu nagusia datu akastunak konpontzen saiatzea da batez ere. Datuen transformazioa, bestalde, iturri anitzetatik datozen datuen arteko loturak ezartzen saiatzen da, hau da, gorago aipatutako objektuen identifikazio-arazoa. Oro har, datu-arazketa lan konplexua dela onartu ondoren, arazoari aurre egiteko hainbat urrats behar direla azpimarratzen dute:

- **Datuen analisia:** egon daitezkeen errore eta inkonsistentziak aurreikusteko, datuen analisia burutu behar da. Eskuartean erabiliko diren datuen gaineko eskuzko azterketaz gain, analisi-tresnak erabili beharko lirateke datuen arteko erlazio ezkutuak azalarazteko.

- **Transformazioaren lan-fluxua:** Integrazio-sistema batek onartuko dituen iturrien kopurua, heterogeneotasun-maila eta hauek gorderiko datuen "zikintasun" mailaren arabera, datu-garbiketa eta transformazio-urratsen kopurua handia izan daiteke. Datu-garbiketa ahalik eta lasterren egitea komeni da, datuak integraziorako prestatuz. Transformazio-urratsa, bestalde, datu-integrazioaren sistemaren oinarrizko ataza da, eta, hortaz, sistemak duen datu-fluxuko eginkizunetariko bat. Halaber, bai datu-garbiketa eta baita transformazio-urratsak ere modu deklaratiboan zehaztu behar direla azpimarratzen du. Horrela, bada, datuen kalitatea bermatzearren integrazio-sistemak egikaritzen dituen urratsak datuei buruzko estentsio-mailako ezagutza beharko du.

- **Egiaztapena:** datu-arazketaren prozesua arrakastatsua izan dadin, ezagutza estentsionalaren egokitasuna egiaztatu beharra dago. Hortaz, transformazio-erregelak eta datuen mapaketak (bir)findu egin beharko lirateke, prozesu iteratibo eta elkarreragile baten bidez, non urrats bakoitzean sistemaren diseinatzaileak erregelen doitasun-maila neurtu eta egokituko duen, atalase-balio batera iritsi arte.

Arazketa-prozesuak adierazkorra eta hedagarria izan behar du. Parekoak diren datu-baseko objektuak ezagutzeko, multzokatzeko eta elkartzeko irizpide-arau anitz eta aberatsak adierazteko aukera eman behar du, eta, baita ere, domeinuari zuzenean loturiko informazioa gehitzeko. Beraz, datu-arazketaren arazoak metaezagutzaren beharra du, era deklaratibo batean zehaztuta,

eta metaezagutza honen bidez gauzatu ahal izango da datuen garbiketa zein objektuen identifikazioa. Datuen kalitatea bermatzeko, halaber, metadatu hauek informazio aberatsa eskaini beharko liokete sistemaren diseinatzaileari, era elkarreragile batean, metadatuen gainean finketak egin ahal izan ditzan.

Horretaz gain, datu-arazketak bete beharko lukeen beste ezaugarri arras garrantzitsua azaltzen da: integrazio-sistemak *erabiltzaileak definitutako funtzioak* onartu beharko lituzke, seguruenik, integrazio-sistemaren diseinatzaileak definituak egongo direnak. Izan ere, datu zikinekin egon daitekeen kasuistika izugarria izanik, integrazio-sistemaren domeinuarekiko dependenteak diren funtzio bereziak beharko dira maiz datuak elkartu ahal izateko.

Hernandez eta Stolfo-ren lana (1995) datu heterogeneoetatik jasotako datu-arazketa ikertzen duen estreinetarikoa dugu. Egileen esanetan, datu zikinik egongo ez balitz, bi objektu edo gehiagok munduko entitate bera erreferentziatzen dutenetz jakitea algoritmo sinple baten mende egongo litzateke: bi objektu edo gehiago berdintzat jotzeko, algoritmo tribial honek datu-baseko erregistro guztiak atributu jakin batzuen arabera ordenatu, eta ondoren elkarren aldamenean agertutako erregistro berdinak ezabatuko lituzke.

Algoritmo honek, ordea, ez du emaitza onargarririk emango datu zikinak daudela aurreikusten badugu. Testuinguru honetan, gerta daiteke datu akastunak agertzea, eta, horrela, entitate bera adierazten duten bi objektuen gakoak desberdinak izatea. Gauzak horrela, datuen arteko konparazio sinplea ez da aski izango datu desberdinek munduko entitate bera adierazten duten jakiteko. Horren ordez, "equational theory" delakoa azaltzen dute: baliokidetasuna definituko duten ekuazioek izan beharko dute konparazio-eragile, ekuazio hauek zenbait erregelaz osatuak daudelarik. Edonola ere den, konparazio-eragile diren ekuazioek lan zaila dute beren gain, bi datuk objektu bera adierazten duten jakitea konplexua baita oso. Hala diote, behintzat, Hernandez eta Stolfo-k, 1995eko lanaren egileak:

> Determining that two records from two databases provide information about the same entity can be highly complex.

Transformazio-erregelak aplikazioaren domeinuarekiko dependenteak eta eskuz eraikiak dira. Jakina, erregela hauek eskuz kodetzea lan neketsua eta konplexua da integrazio-sistemaren diseinatzailearentzat, eta nekez erabil daiteke domeinu bateko ezagutza beste domeinu baterako. Arazoa larriagoa da Internetetik datuak erauzi eta integratu nahi direnean, datu hauen egitura oso desberdina izan baitaiteke. Gauzak horrela, zenbait lanetan (adib. Knoblock *et al.* (2001)) objektuak identifikatzeko erregelak automatikoki ikasten

saiatzen dira. (Knoblock *et al.*, 2001) lanaren arabera, bi ezagutza mota behar dira objektu-identifikazioa taxuz egin ahal izateko: (1) erregistroak elkarrekin erkatzeko erabili behar diren atributuak zein diren zehaztu, eta (2) aplikazioaren domeinuan testu-mailako desadostasun edo transformazioak zehaztu.

Azkenik, objektu-identifikazioaren arazoa ebazteko algoritmoen eraginkortasunari begiratu azkarra botako diogu. Historikoki, objektu-identifikazioaren arazoa lan konplexua eta motela izan da, objektuen artean hainbat erregela aplikatu behar direlako, eta erregelek zama handia izaten dutelako. Beraz, gorago aipaturiko algoritmo tribiala ezin da zuzenean erabili, berdintasun-eragile bezala transformazio-erregelak erabiliz. Hau horrela izanik, betidanik egon da, ikertzaileen artean, objektu-identifikazioa arintzea xedetzat duten tekniken garapenaren beharra. Datu-base osoan objektuak identifikatzen dituen algoritmoaren zama arintzeko, bi urratseko irtenbidea aurkezten dute: lehenengo batean, datu-baseko erregistro guztiak ordenatzen dituzte, heuristiko baten arabera asmatutako gako-erauzle baten bidez, antzeko erregistroak gertu azalduko direlakoan; bigarren urratsean, berriz, w kopuruko leihoa korritzen dute datu-base ordenatuan zehar. Leihoan sartzen den erregistro bakoitza aurreko $w-1$ erregistroekin alderatzen da, "equational theory"-ren bidez sortutako baliokidetasun-erregelak aplikatuz. Ondoren, eta baliokidetasun-erlazioa trantsitiboa dela kontuan hartuz, itxitura trantsitiboa kalkulatzen dute datu-baseko erregistroetan zehar, eta itxitura bakoitzeko elementu esanguratsu bakarra aukeratzen dute. Azkenik, datu-baseko erregistro guztiak beren elementu esanguratsuarekin ordezkatzen dituzte.

III.4 Wrapper-en teknologia.

Wrapper-ak software-moduluak dira, iturri lokalak eta integrazio-sistemaren arteko komunikazioa ahalbideratzeko sortuak (Roth eta Schwartz, 1997). *Integrazio estrukturala* deritzogun arazoa saihesten lagunduko dute, alegia, iturrien artean egon daitezkeen datu-ereduen, programazio-interfaze eta galdeketa-gaitasunen arteko desberdintasunek sortutako arazoak, besteak beste. Izan ere, iturriak oso egitura desberdinez balia baitaitezke datuak gordetzeko: datu-base erlazionalez, objektuei zuzendutako datu-baseez, modu sasi-egituratuan antolatutako datu-baseez eta abar. Datu-ereduekin hertsiki lotuta, iturri bakoitzari bere kontsulta-lengoaian adierazi beharko zaizkio galderak, gordeta dituen datuak eskuratu ahal izateko.

Wrapper-en garapena lan neketsua da, beren pean duten iturri lokalaren

sakoneko egiturak eta ezaugarriek erabat baldintzatua. Horrela, datu-base erlazional baten *wrapper*-aren garapenak oso gutxi lagunduko du, konparazio batera, datuak eskuratu ahal izateko liburutegi bereziak behar dituzten iturrien gaineko *wrapper*-ak eraikitzeko garaian.

Badago *wrapper*-ak automatikoki edo erdi-automatikoki osatzera zuzendu diren lanak. Estreinetako bat TSIMMIS proiektuaren barruan kokatzen da[21], proiektuaren helburuetako bat, eta ez apalena, *wrapper*-en osaera automatikoki edo erdi-automatikoki egiteko aukera ematen duen formalismo eta tresneriak eskaintzea izanik. Papakonstantinou *et al.*-en lanean (1995b) datu-integraziorako *wrapper*-ak automatikoki garatu ahal izateko ingurunea azaltzen da. Bertan, OEM lengoaia erabiltzen da[22] *wrapper*-aren eta integrazio-sistemaren arteko komunikazioa adierazteko, eta OEMren gainean galderak egiteko logiketan oinarritutako MSL kontsulta-lengoaia ere definitzen da; integrazio-sistema MSL lengoaiaz baliatuko da informazio-eskaria *wrapper*-ari iritsarazterakoan, eta honek, berriz, OEMz bueltatuko dizkio galdera betetzen duten datuak. *Wrapper*-ak, testuinguru horretan, QDTL (*Query Description and Translation Language*) lengoaia baten bidez erabat defini daitezke: datu-base lokalek onartutako galdera-mota bakoitzeko QDTL erregela bat idazten da, zeinek zehazten duen, era deklaratibo batean, zein eragiketa burutu behar den iturri lokalean eskatutako datuak eskuratzeko.

Beste alde batetik, Interneteko orrietatik informazioa erauzi eta informazio-sistemei eskaintzen dieten *wrapper*-en garapena ere sakon ikertu da azken bolada luzean (Kushmerick *et al.*, 1997; Ashish eta Knoblock, 1997; Knoblock *et al.*, 2001). Tesi-lan honetan sakondu ez dugun arloa bada ere, *web* orrietan gordetako informazioaren egitura automatikoki ezagutzeak interes handia du informazio-sistementzat. Interneteko orrietan zehar informazio franko ezkuturik agertzen zaigu, beren baitan metatutako informazioaren egitura azalarazi gabe, alegia, eta *wrapper*-en esku dago informazioa orrietatik erauzi eta sistema osora banatzea. Orriek eskainitako informazio ezkutua azalarazten bada, datu kopuru galanta integratu ahal izango da informazio-sistemetan. Horrela, bada, orrien egitura erdi-automatikoki ikasten saiatzen diren *wrapper*-en garapenerako teknika ugari sortu da.

Web orrietan paratutako informazioa sasi-egituratua dela esan ohi da[23], egitura finkoa jarraitzen ez badu ere —datu-base erlazionalak ez bezala—,

[21] III.6 atalean TSIMMIS proiektua aztertuko dugu.
[22] Ikus IV.5.1 atala, 200. orrian.
[23] Ikus II.2.2.1 atala, 54. orrian.

beren baitan duten informazioa gramatika formal bat erabiliaz koka baitaiteke. Horrelako gramatika izanik, orrietako informazioa eskura daiteke, beraz, hizkuntzaren tratamenduan oinarritutako ulermen-teknika berezirik erabili behar izan gabe. Orrien gainean *wrapper*-ak erantsi behar badira, baina, orrien egitura deskribatzen duten informazio sintaktikoa zein semantikoa behar dira.

Gauzak horrela, saiakera ugari egin dira *web* orrien egitura sintaktikoa automatikoki ikasteko, eta, informazio sintaktikoa abiapuntu, orriaren eduki semantikoa azalarazten duten teknikak garatzeko. Teknika hauek, askotan, giza-erabiltzailearen laguntza behar izaten dute hasieran: interfaze lagungarri batez baliatuz, erabiltzaileak orri batzuk eskuz kodetzen ditu, trebatze-corpus gisa; informazio horretan oinarrituz, mota bereko orrietatik informazioa erauzten duten *wrapper*-ak automatikoki sortzen dira.

III.5 Deskribapen-logikak.

Ezagutzaren adierazpenaren arlotik datozen deskribapen-logiken[24] (DL) formalismo-familiaren helburu nagusia bikoitza da: alde batetik hainbat eremutako ezagutza adierazteko alderdi formalak eskaintzea eta, bestetik, ezagutza horren gainean egindako arrazoibideen erabakigarritasuna bermatzea. Ezagumenaren egiturari buruzko adierazpenean oinarritutako ezagutzaren adierazpen-sistemen estreinako saioa sare semantikoetan eta *frame* delakoetan dugu[25]. Hala ere, formalismo hauek semantikaren definizio-eza erakusten zuten eta, maiz, sistema hauen inplementazio zehatzek ezartzen zituzten frogabidearen inguruko gorabeherak. Sare semantiko zein frame-etarako oinarri semantiko sendoa emateko jaio zen KL-ONE errepresentazio-sistema ospetsua (Brachman eta Schmoke, 1985), eta lan hark ereindako haziaren ildotik datoz, zuzen-zuzenean, deskribapen-logikako lehenengo sistemak, hala nola, BACK (Quantz eta Royer, 1990), LOOM (McGregor eta Bates, 1991) edo CLASSIC (Bordiga *et al.*, 1989). Estreinako sistema hauek *datu-base terminologikoaren* izenarekin ezagutzen ziren; izan ere, erabiltzen duten adie-

[24]Datu-base terminologiko edo kontzeptu-lengoaiak ere deiturikoak.

[25]Frame eta sare semantikoen artean desberdintasun ugari egon arren, beren oinarriek zerikusi handia dute. Izan ere, sarearen egitura izango baita indibiduoen multzoak eta beren arteko erlazioak adierazteko bidea bi formalismoetan. Hori horrela, bi formalismo hauek sarean oinarritutako egituren motako familian kokatu ohi dira.

razpen-lengoaiak modelatu nahi den domeinuaren oinarrizko terminologia zehazki ezartzen baitu.

Deskribapen-logikak objektuei (indibiduoak), objektu-multzoei (kontzeptuak) eta hauen arteko erlazioei (rolak) buruzko informazioa adierazteko aukera ematen duen lengoaiaz horniturik daude. Esate baterako, $\mathsf{Pertsonak} \sqcap \mathsf{Gizonezkoak}$ espresioa, gizonezko gizabanakoen multzoa adieraziko luke. $\mathsf{Pertsonak} \sqcap \forall \mathsf{ume.Emakumezkoak}$ espresioak, berriz, alabak dituzten pertsonen multzoa soilik adieraziko luke. *Kontzeptu-lengoaiak* deituriko lengoaia hauek kontzeptu-espresio edo role-espresio konplexuak osatzeko aukera ematen dute, zenbait eraikitzaileren bitartez: kontzeptuen arteko ebakidura ($\mathsf{C_1} \sqcap \mathsf{C_2}$), kontzeptuen arteko bildura ($\mathsf{C_1} \sqcup \mathsf{C_2}$), kontzeptu baten ukapena ($\neg\mathsf{C}$), kuantifikazio unibertsala ($\forall R.C$, R rolean definitzen ari garen kontzeptuarekin agertutako indibiduo guztiak C kontzeptukoak direla adierazten duena), kuantifikazio existentzial mugatua ($\exists R.C$, R rolean definitzen ari garen kontzeptuarekin agertutako indibiduoren bat C kontzeptukoa dela adierazten duena) edo mugagabea ($\exists R$, R rola hutsik ez dagoela adierazteko), rolen gainean eginiko kopuru-murriztapenak ($\leq n\ R$ eta $\geq n\ R$) eta abar. Eraikitzaile hauen aukeraketa arras garrantzitsua da, kontzeptu-lengoaia bat erabat identifika baitaiteke onartzen dituen eraikitzaileen multzoaren bitartez; beraz, eraikitzaileen aukeraketa izango da, azken finean, deskribapen-logika zehatz baten adierazpide-ahalmena —eta, bide batez, arrazoibidearen konplexutasuna— ezarriko duena.

Kontzeptu-espresioen oinarrizko inferentzia *subsuntzioa* da. C kontzeptuak D kontzeptua subsumitzen duen ($C \sqsupseteq D$) determinatzearen arazoa, C kontzeptua D baino orokorragoa den zehaztean datza. Bestela esanda, subsuntzioak bigarren kontzeptuak (D) adierazitako indibiduo-multzoa lehenengo kontzeptuak (C) adierazitakoaren azpimultzoa denetz zehaztuko du. Horrela, bada, deskribapen-logiketan oinarritutako sistemek subsuntzioa erabiliko dute kontzeptuen arteko hierarkiak (taxonomiak) eraikitzeko.

Deskribapen-logikako sistemetan oinarritutako ezagutza-baseek (DLEB) bi osagai dituzte: alderdi terminologikoa (ezagutza *intentsionala* ere deiturikoa) eta instantziak (ezagutza *estentsionala*). Ezagutza intentsionalean gorderikoa ezagutza-base batean aldatuko ez den informazioa izango den bitartean, ezagutza estentsionalak eskainitakoa inguruak ezarritako zenbait baldintzaren araberako informazioa litzateke eta, beraz, aldakorra.

Ezagutza terminologikoak errepresentatu nahi diren kontzeptuen (indibiduo multzoen) ezagutza gordeko du, hala nola, kontzeptuen ezaugarri orokorrak eta kontzeptuen arteko erlazioak. Subsuntzio-eragilea dela medio, kon-

tzeptuen arteko taxonomiak ere osatu ahal dira. Osagai terminologiko honetan, TBox ere deiturikoa, kontzeptuen definizioa ere gauzatzen da. Adibidez, erazagupen honek:

$$\mathsf{Gizonak} \equiv \mathsf{Pertsonak} \sqcap \mathsf{Gizonezkoak}$$

pertsonen eta gizonezkoen arteko ebakidurako indibiduo multzoari Gizonak kontzeptu-izena emango dio. Mota honetako erazagupena inplikazio logikoa izango balitz bezala interpretatzen da, hau da, indibiduo batek Gizonak kontzeptuak adierazitako multzokoa izateko bete behar duen baldintza nahikoa eta beharrezkoa zehazten dizkigu. Hala ere, definizio mota hau zorrotzegia izan daiteke datu-baseen arloa bezalako beste hainbat esparrutan non, indibiduoren bat klase baten barruan sailkatu ahal izateko, indibiduo horrek betetzen dituen ezaugarriak nahikoak ez diren. Hau horrela izanik, DL sistemetan bi kontzeptu mota definitu ohi dira: aurrean ikusitako *definizio-kontzeptuak* ($C \doteq D$), baldintza beharrezko eta nahikoak eskatzen dituztenak eta, bestalde, *kontzeptu primitiboak* ($C \sqsubseteq D$), kontzeptu horretako indibiduo guztiek bete behar dituzten baldintzak ezartzen dituztenak (baldintza beharrezkoak).

Terminologia bat eraikitzearen oinarrizko helburua *sailkapena* gauzatzean datza: sistemak definitutako kontzeptuak hierarkia batean gordetzen ditu eta, kontzeptu berri bat definitu bezain laster, automatikoki kokatuko du taxonomia-zuhaitzean "dagokion" lekuan. Sailkapena, arrazoitze-zerbitzu oinarrizkoa den subsuntzioaren bitartez gauzatuko da.

Ezagutza estensionalak, berriz, indibiduo zehatzen informazioa gordeko du eta, horrela, indibiduo bat dagokion klasearekin lotzen duten asertzioez edo indibiduo desberdinen arteko erlazioak zehazten dituzten asertzioez osaturik dago. Instantzia-alderdia, ABox deiturikoa, terminologiak definitutako eskemaren instantziazio (partziala) litzateke. Adibidez, asertzio honek:

$$\mathsf{Pertsonak} \sqcap \mathsf{Gizonezkoak}(\mathsf{Jokin})$$

Jokin indibiduoa gizonezko pertsona dela adierazten digu. Arestian ikusitako gizonen definizioa izanik, Jokin indibiduoa Gizonak kontzeptuaren instantzia dela erator dezakegu. Bestalde, beste asertzio honek:

$$\mathsf{umeaDu}(\mathsf{Joseba}, \mathsf{Jokin})$$

Joseba indibiduoa Jokin indibiduoaren gurasoa dela adieraziko luke.

ABox instantzia-alderdiaren oinarrizko ataza *instantzia-test-a* da, hau da, indibiduo bat izanik kontzeptu batekoa den erabakitzea.

Deskribapen-logikaren arloan garrantzi handia eman zaio espresio-ahalmen eta eraginkortasunaren arteko konpromisoaren bilaketari, hasiera-hasieratik. Oro har, adierazpen-lengoaia zenbat eta ahaltsuagoa izan, orduan eta konplexuago bilakatuko dira arrazoibide-prozesuak. Brachman eta Levesque-ren lanean (1985) arazo honen lehenengo adibidea ikus daiteke, $\mathcal{FL}^-$ kontzeptu-lengoaia[26] aztertzerakoan. Lengoaia honekin subsuntzioa denbora polinomikoan ebatz daitekeela erakusten dute baina, rol-murriztapeneko eraikitzailea gehituz[27] gero, subsuntzioa co$\mathcal{NP}$-osoko konplexutasuna duen arazoa bilakatzen da. Deskribapen-logiken arloko ekarpen garrantzitsuenetariko bat lengoaia batek onartutako eragile logikoen eta lengoaia horren gainean egin daitekeen arrazoibidearen konplexutasunaren arteko mendekotasunaren ikerketa formala da.

Gauzak horrela, eraikitzaile multzo desberdinak dituzten kontzeptu-lengoaia ugari garatu da. $\mathcal{FL}^-$ lengoaiari $\top$ kontzeptu orokorrena (beste edozein kontzeptu subsumitzen duena), $\bot$ indibiduorik gabeko kontzeptu hutsa eta $\neg$ ukapen-eraikitzaileak gehituz, ospe gehien duen $\mathcal{AL}$ lengoaia dugu. Lengoaia honi zenbait eraikitzaile gehitu ondoren, $\mathcal{AL}$ familiako lengoaiak ditugu. Lengoaia hauen izenak, bestalde, onartzen dituzten eraikitzaileen araberakoa izango dira:

$$\mathcal{AL}[\mathcal{C}][\mathcal{E}][\mathcal{N}][\mathcal{O}][\mathcal{R}][\mathcal{U}]$$

III.6 irudian $\mathcal{AL}$ lengoaia-familiek onartzen dituzten eraikitzaileak eta dagozkien interpretazio semantikoa ikus daitezke. Lengoaia hauen interpretazio semantikoa egiterakoan, kontzeptuak domeinu orokor baten azpimultzoak bezala interpretatzen dira, eta rolak erlazio bitarren bitartez. Formalago, $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ *interpretazioa* $\Delta^{\mathcal{I}}$ multzoa eta $\cdot^{\mathcal{I}}$ interpretazio-funtzioaren bitartez definituko da, zeinaren bitartez C kontzeptu bat $\Delta^{\mathcal{I}}$ multzoko $C^{\mathcal{I}}$ azpimultzoarekin lotzen duen, eta P rola $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$-ko $P^{\mathcal{I}}$ erlazio bitarrarekin.

III.5.1 Deskribapen-logikak eta datu-integrazioa.

Ezagutzaren errepresentazioko arloan sakon aztertu da DLen formalismoa. Ez da batere harritzekoa, beraz, mundu errealeko egoeren semantika adierazteko

[26] *Frame Language* delakoa oinarrizko lengoaiatzat hartu izan da, eta kuantifikazio unibertsal ($\forall R.C$) zein mugagabeko kuantifikazio existentzialeko ($\exists R$) eraikitzaileak ditu.

[27] ($\leq n\ R$) edo ($\geq n\ R$)

Eraikitzailea	Sintaxia	Interpretazioa
Kontz. orokorra	$\top$	$\Delta^{\mathcal{I}}$
Kontz. hutsa	$\bot$	$\emptyset$
Kontzeptu-izena	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
Konjuntzioa	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
Disjuntzioa ($\mathcal{U}$)	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
Ukapena ($\mathcal{C}$)	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
Kuantif. unibertsala	$\forall R.C$	$\{d \in \Delta^{\mathcal{I}} \mid \forall e : (d,e) \in R^{\mathcal{I}} \rightarrow e \in C^{\mathcal{I}}\}$
Kuantif. existentziala ($\mathcal{E}$)	$\exists R.C$	$\{d \in \Delta^{\mathcal{I}} \mid \exists e : (d,e) \in R^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}$
Kopuru-murriztapena ($\mathcal{N}$)	$(\geq n\ R)$ $(\leq n\ R)$	$\{d \in O^{\mathcal{I}} \mid \#\{e \mid (d,e) \in R^{\mathcal{I}}\} \geq n\}$ $\{d \in O^{\mathcal{I}} \mid \#\{e \mid (d,e) \in R^{\mathcal{I}}\} \leq n\}$
Objektu-bilduma ($\mathcal{O}$)	$\{a_1, \ldots, a_n\}$	$\{a_1^{\mathcal{I}}, \ldots, a_n^{\mathcal{I}}\}$

III.6 Irudia: $\mathcal{AL}$ deskribapen-logiken familiako eraikitzaileak eta beren interpretazio semantikoa

ere aukera ematea, egoera hauen artean datuen semantika dagoelarik. Hortaz, DLak egokiak dira datu-eredu semantikoak edo objektuei zuzendutako ereduak adierazteko. Demagun klase baten honako definizio hau dugula:

class Ikasleak **is-a** Pertsonak **with**
ikasZbk : INTEGER
maila : {1,2,3,4}

DLen terminoetan, Ikasleak eta Pertsonak kontzeptu primitiboak dira —pertsona bat ezin da soilik bere kanpoko ezaugarrien bitartez ezagutu—. Horrela, bada, erazagupen honek murriztapen bat ezarriko digu, hots, Ikasleak kontzeptuko instantzia guztiek bete behar dituzten beharrezko baldintzak zehaztuko ditu. Ezaugarri hauek DLen bitartez adierazi ahal izateko, honako Ikasleak kontzeptua adierazi behar dugu:

$$\begin{aligned} \text{Ikasleak} \sqsubseteq\ & \text{Pertsonak} \sqcap \\ & \forall \text{ikasZbk.INTEGER} \sqcap (\geq 1\ \text{ikasZbk}) \sqcap (\leq 1\ \text{ikasZbk}) \sqcap \\ & \forall \text{maila.}(\{1,2,3,4\}) \sqcap (\geq 1\ \text{maila}) \sqcap (\leq 1\ \text{maila}) \end{aligned}$$

Datu-baseen eskema DLen bitartez adierazteak ondoko abantailak ditu, besteak beste (Borgida, 1995):

- Datuen semantikaren informazio gehigarria eskain dezakete.
- Datu-basearen eskemaren egonkortasuna egiaztatzen lagundu dezakete. Esate baterako, datu-base eskema batek C kontzepturen bat inkontsistentea bihurtzen duenetz egiazta daiteke.
- Eskemaren errepresentazioko informazio erredundantea urritu dezake, klase bakoitzak gurasoengandik heredatutako informazioaren gainean ezaugarri minimoak soilik gehituz.

DLen aplikazioa datu-integrazioaren arloan sakon ikertu da formalismo hauen hasieratik (Catarci eta Lenzerini, 1993; Blanco *et al.*, 1994; Arens *et al.*, 1996; Seth *et al.*, 1993). Ikuspuntu tradizionalean, datu-base batek munduaren eredu osoa eta doia eskaintzen du, non indibiduo guztiak balio primitiboak diren (karaktere-segidak, zenbakiak, etab.), eta haien arteko erlazioak espreski adierazita dauden. Datu-integrazioaren arloan, baina, munduari buruzko osatu gabeko informazioa adierazi behar da maiz, kapitulu honetan zehar ikusi dugun bezala. Deskribapen-logikek osatu gabeko informazioa adierazteko aukera ematen dute, indibiduoak kontzeptuetan sailkatu ahal izateko ez baita euren informazio *guztia* jakin behar. Halaber, datu-base tradizionalean galderek erantzun estentsionala eskatzen dute, hots, galderaren erlazioarekin bat datozen indibiduo atomikoen (edo n-koteen) lista. Osatu gabeko informazioa onartzen bada, ordea, erantzun deskriptiboak itzul daitezke: galderarekin bat datozen indibiduoek bete behar dituzten ezaugarriak adierazi, indibiduoak zehatzak zeintzuk diren jakin gabe ere.

Catarci eta Lenzerini-ren lanean (1993) esaten den legez, iturrien deskribapena eta eskema globalak errepresentatzeko deskribapen-logika erabiltzen bada, logikaren arrazoibide-prozesuak erabili ahal dira galdera baten iturri egokiak zein diren ezagutzeko. Ikus dezagun, adibide baten bidez, DLen subsuntzio- eta sailkapen-eragileen erabilera datu-integrazioaren arazoan.

III.5.1 Adibidea Demagun honako kontzeptu hauek ditugula:

$$\begin{aligned}
&\mathsf{AitonAmonaEdoUmeGabekoak} \doteq \mathsf{Pertsonak} \sqcap \forall \mathsf{ume}.(\exists \mathsf{ume}) \\
&\mathsf{IgeldotarrarenGurasoak} \doteq \mathsf{Pertsonak} \sqcap \exists \mathsf{ume}.(\exists \mathsf{biziDa}.\mathsf{Igeldo}) \\
&\mathsf{Igeldo} \sqsubseteq \mathsf{GipuzkoakoHerriak}
\end{aligned}$$

non AitonAmonaEdoUmeGabekoak eta IgeldotarrarenGurasoak iturri datu-base

baten erlazioak diren, eta ume zein biziDa rolak diren. Lehenengo erazagupenak AitonAmonaEdoUmeGabekoak klaseko indibiduo multzoa deskribatzen du: klase horretako indibiduoa izateko, edo umerik ez du, edo, baldin baditu, ume horiek beren aldetik gurasoak dira. IgeldotarrarenGurasoak, bestalde, umeak igeldotarrak dituzten indibiduo multzoa identifikatzen du. Hirugarren erazagupenak Igeldo GipuzkoakoHerriak kontzeptuaren azpiklasea dela adierazten digu. Galde dezagun, orain, umeak dituzten eta Gipuzkoan bizi diren pertsonen gurasoei buruz:

$$\mathsf{Pertsonak} \sqcap \exists \mathsf{ume}.((\exists \mathsf{ume}) \sqcap (\exists \mathsf{biziDa}.\mathsf{GipuzkoakoHerriak}))$$

Galdera erantzuteko sistemak ezagututako indibiduo guztiak sailka ditzake. Erantzuna AitonAmonaEdoUmeGabekoak eta IgeldotarrarenGurasoak kontzeptukoak diren indibiduo guztiak izango dira, izan ere, eginiko definizioen arabera ondorengo subsuntzioa beteko baita:

$$\mathsf{Pertsonak} \sqcap \exists \mathsf{ume}.((\exists \mathsf{ume}) \sqcap (\exists \mathsf{biziDa}.\mathsf{GipuzkoakoHerriak})) \sqsupseteq \mathsf{AitonAmonaEdoUmeGabekoak} \sqcap \mathsf{IgeldotarrarenGurasoak}$$

□

Aurreko adibidean LAV hurbilpena jarraitu da iturburu-erlazioak —AitonAmonaEdoUmeGabekoak eta IgeldotarrarenGurasoak— erlazio orokorren bitartez adieraziak baitaude —Pertsonak, ume eta biziDa—. Galdera bera ere eskema orokorraren gainean eginikoa da. Hala ere, DLen bitartez adierazitako murriztapenek LAV edo GAV hurbilpena jarrai dezakete.

Zenbait egileren ustetan, bi integrazio mota gauza daiteke DLen bitartez (Calvanese *et al.*, 1999a), eta horrela, *Galderen berridazketa* ("Query Rewriting") eta *Galderen erantzutea* ("Query Answering") bereiziko dituzte. Lehenengo motako integrazioa, oro har, orain arte ikusi dugunarekin bat dator. D datu-base bat eta erabiltzaileak ezarritako Q galdera bat izanik, Q galderaren erantzun baliokidea emango digun Q' berridazketa behar da. Bigarren urrats batean, berridazketa datu-basearen gainean egikarituko da, datuak lortzearren. Berridazketak erantzun baliokidea eskuratzeko duen gaitasuna, berridazketa eta galdera adierazteko erabili den lengoaiaren espresibotasunaren araberakoa izango da.

Galderen erantzutea delako integrazio motan, berriz, bi prozesuak —hots, berridazketa eta erantzunaren eskuratzea— batera burutzen dira. Hortaz, galdera bat itzultzerakoan, ez da bisten definizioen informazioa soilik erabiliko; horretaz gain, bisten *estentsioak* ere kontuan hartuko dira. Horrela,

galdera baten erantzuna bilatzerakoan, galderaren prozesaketaren gainean ez da inongo murriztapenik ezartzen; teknika honen helburu bakarra jatorrizko galderaren erantzun baliokidea lortzea da, eskura dagoen informazio guztia ustiatuz.

Hala ere, esan beharra dago galderen erantzutea, DLetan oinarritzen bada, ataza konplexua izango dela, halabeharrez: galderak DLen bitartez erantzuteko, datu-base guztien n-koteak DLen bidez adierazi beharko lirateke, eta DLetako ezagutza-basean asertzioen bidez txertatu, datu bakoitza ABO-Xean dagokion tokian kokatuz. Hala ere, DLetako ezagutza-baseak ez daude, oro har, hainbeste informazio kudeatzeko prestatuak. Horrela, pentsa daiteke galderen erantzutea, DLen paradigma baten pean, ez dela errealista.

DLak hedatu izan dira datu-informazioaren arazoari egokitzeko. Hor ditugu, esate baterako, $\mathcal{DLR}$ lengoaia (Calvanese *et al.*, 1998a), zeinaren bitartez adieraz daitezkeen bi osagai baino gehiagoko erlazioak; $\mathcal{CIQ}$ lengoaia, alderantzizko erlazioak adierazteko gai izateaz gain, espresioetan adierazpen erregularrak onartzen dituena[28] edo $\mathcal{SI}$ familiako deskribapen-logikak (Horrocks *et al.*, 2000), erlazioen itxitura trantsitiboak ere beren baitan hartzen dituztenak[29].

Edonola ere den, deskribapen-logiketan oinarritutako datu-integrazioa irekia dagoen ikerlerroa da egun, (Levy, 2000) lanean aitortzen den bezala. Besteak beste, huts-hutseko DL datu-integrazioan, galderak erantzutea, bistetan oinarrituz, arazoa oso zaila baitugu (ikus Calvanese *et al.* (1999a)), eta, askotan, berridazketak *datalog* programa errekurtsiboak bezala adierazi behar izango dira (Levy, 2000). Gauzak horrela, DLek bi eginkizun nagusi bete dituzte datu-integrazioaren arloan: datuen sailkapena eginez galderen erantzutea (Calvanese *et al.*, 1999a) eta galderen barne-hartzearen arazoaren ebazpena, subsuntzioa erabiliz.

[28] Adierazpen erregularrak onartzea ezinbesteko baldintza dugu sasi-egituratutako datu-baseen atzipenean

[29] Erlazio meronimikoak (zatia-da eta abar) adierazteko beharrezkoa da itxitura transitiboa.

III.5.2 CLASSIC deskribapen-logika.

Deskribapen-logikan oinarritutako CLASSIC[30] (Brachman *et al.*, 1992; Bordiga *et al.*, 1989; Resnick *et al.*, 1996) sistema lengoaia hauen estreinako inplementazioetarikoa da. DLen arloko "eraginkortasunaren vs. adierazpen-ahalmenaren" arteko konpromisoan *limited but complete* delako hurbilpena jarraitzen du CLASSICek (Baader *et al.*, 2001): sistemak duen kontzeptu-lengoaiaren eraikitzaile kopurua —eta, beraz, espresio-ahalmena— mugatua da baina, horrela, subsuntzioa konputatzea eraginkorra da (denbora polinomiala behar du).

III.7 irudian CLASSIC sistemak erabiltzen dituen eragile logikoak ikus daitezke, eragile bakoitzaren interpretazio semantikoarekin batera.

Deskribapen-logiketan ohikoa den bezala, CLASSICek kontzeptuez osatutako alderdi terminologikoa (TBox), indibiduoez osaturiko instantzia-alderdia (ABox) eta indibuoen arteko erlazioak adierazteko aukera ematen du (*rol*en bidez). Osagai oro *deskribapenen* bidez definitzen da, non deskribapenek indibiduo multzo batek dituen propietate komunak adierazten dituzten[31]. Deskribapenak indibiduoei aplikatzen zaizkie: indibiduo bat deskribapen baten instantzia dela —edo indibiduoak deskribapena betetzen duela— esango da, baldin indibiduoari buruz ezaguna den informazioa deskribapenekoarekin bat badator.

TBOXa definizio-kontzeptu zein kontzeptu primitiboz osaturik dago. Kontzeptuak deskribapenak konbinatuz osatzen dira, eta deskribapenak konposatzeko multzoen arteko ebakidura-eragilea (`and`) erabiltzen da. Kontzeptu orok, bestetik, guraso bat edo gehiago izan behar du, ezinbestean. Hortaz, bi kontzeptu berezi daude CLASSICen: `ClassicThing` kontzeptu berezia, kontzeptu guztien arbasoa dena, eta umerik izan ez dezakeen `NoThing` kontzeptua. Datu-ereduen modelizaziorako beharrezkoa den kontzeptu primitiboen arteko disjuntzioa ere onartzen du CLASSIC sistemak[32]. Horrela, kontzeptu primitiboak disjuntzio multzoetan bana daitezke. Bi kontzeptu disjuntuak badira, ezin da bi kontzeptu horiek guraso bezala dituen indibiduorik egon.

[30]Tesi-lanean ez gara CLASSIC sistemaz baliatu, bere ordez *NeoClassic* sistema erabili baitugu. NeoClassic CLASSIC sistemaren C++ lengoaiarako dagoen bertsioa da (Resnick *et al.*, 1996). Edonola ere den, eta bi sistemak baliokideak direnez, CLASSIC zein *NeoClassic* terminoak erabiliko ditugu hemen, bereizketarik egin gabe.

[31]Nolabait, esan daiteke deskribapenen *egiturek* zehazten dutela, CLASSICen, kontzeptuen esanahia (Bordiga *et al.*, 1989).

[32]Definizio-kontzeptuak, bestalde, beti dira disjuntuak.

Eraikitzailea	*Interpretazioa*
$THING$	$\Delta^{\mathcal{I}}$
$NOTHING$	$\emptyset$
$(and\ C_1 \ldots C_n)$	$C_1^{\mathcal{I}} \cap \ldots \cap C_n^{\mathcal{I}}$
$(all\ p\ C)$	$\{d \in \Delta^{\mathcal{I}} \mid p^{\mathcal{I}}(d) \subseteq C^{\mathcal{I}}\}$
$(oneOf\ b_1, \ldots, b_n)$	$\{b_1^{\mathcal{I}}, \ldots, b_n^{\mathcal{I}}\}$
$(atLeast\ n\ p)$	$\{d \in \Delta^{\mathcal{I}} \mid \#\{p^{\mathcal{I}}(d)\} \geq n\}$
$(atMost\ n\ p)$	$\{d \in \Delta^{\mathcal{I}} \mid \#\{p^{\mathcal{I}}(d)\} \leq n\}$
$(fills\ p\ b_1, \ldots, b_n)$	$\{d \in \Delta^{\mathcal{I}} \mid b_j^{\mathcal{I}} \in p^{\mathcal{I}}(d)\}$

III.7 Irudia: CLASSIC sistemak onartutako eraikitzaileen sintaxi eta semantika. C sinboloak kontzeptuak adierazten ditu, p-k rolak eta b-k indibiduoak

Horretaz gain, CLASSICek baditu zenbait eragile berezi. Esaterako, kontzeptuak enumerazioaren bitartez defini daitezke, `oneOf` eragilea medio. Horrela, `AdjektiboMotak` kontzeptua honako deskribapenaren bitartez adieraz daiteke:

```
(oneOf (izenLagun izenOndo))
```

non izenLagun eta izenOndo indibiduoak diren.

CLASSICek *test-kontzeptuak* deituriko kontzeptu mota berezia du. Test-kontzeptuek erabiltzaileak propio kodetutako test-funtzio bat dute esleitua. Test-funtzioak, bestalde, edozein programazio-lengoaiatan defini daitezke, eta beren emaitzaren motak boolearra izan behar du, hots, test funtzioak predikatuak dira. Kontzeptu arruntek bezala, test-kontzeptuek indibiduo multzo bat adierazten dute: indibiduo sorta bat, zeinentzat test-funtzioak egiazko balioa itzultzen duen. Horrela, bada, `ZenbakiBikoitiak` test kontzeptua horrela adieraz daiteke:

```
(test bikoiti)
```

non ***bikoiti*** argumentu bakarreko test-funtzioa den, eta egiazko balioa itzultzen duen baldin eta soilik baldin argumentua zenbaki bikoitia bada. `Integer` kontzeptua aldez aurretik definiturik balego[33], kontzeptua horrela adieraz zitekeen:

[33]CLASSICek `Integer` kontzeptua definitzen du, beste oinarrizko motekin batera, hots, `Real`, `String` etab.

```
(and Integer (test bikoiti))
```

Test-kontzeptuak CLASSICeko berezitasunak dira, eta tentu handiaz erabili beharrekoak. Izan ere, kontzeptu primitiboez ez bezala, test-kontzeptuen bidez oinarrizko baldintza nahikoak adierazten dira, hots, indibiduoak test-kontzeptu baten instantzia bezala aintzat hartzeko nahikoa diren baldintzak. Baldintza nahikoak, bestalde, era prozeduralean zehazten dira, test-funtzioen bitartez. Hala ere, test-funtzioen konplexutasuna funtsezkoa da sistema osoaren arrazoibidearena murrizteko. Atalaren hasieran aipatu dugu CLASSICek duen estrategia "eraginkortasuna vs. adierazpen-ahalmena" auzian. Test-funtzio desegokiak (konplexuak) erabiltzen badira, ordea, sistema osoaren eraginkortasuna apur daiteke —eta, adibidez, subsuntzioa prozesatzea erabakiezina den arazoa bihur daiteke.

CLASSICen beste osagai nagusia *rol*ak dira. *Rol*ak TBoxeko kontzeptuen indibiduoen arteko erlazioak adierazteko erabiltzen dira, non erlazioen aritateak bikoa izan behar duen. *Rol*ak murritz daitezke, kuantifikazio unibertsalaren eta kopuru-murriztapeneko eragileen bitartez. Lehenengoarekin (`all` eragilea), *rol*eko hein-kontzeptua zein den zehatz daiteke; kopuru-murriztapenen bidez (`atMost` eta `atLeast` eragileak), berriz, indibiduo batek *rol* jakin baten bidez eduki ditzakeen elementu maximo eta minimoak zehazten dira, hurrenez hurren. Bi *rol* mota daude, bata indibiduoen arteko erlazioak gauzatzeko, eta bestea indibiduoen atributuak definitzeko. *Rol*a indibiduo baten atributua izango da baldin indibiduoa, *rol* horren bitartez, elementu bakar batekin erlazionatu ahal bada.

Adibidez, `Hitzak` eta `HitzEstandarrak` kontzeptuak aldez aurretik definituta badaude, `HitzEzEstandarrak` kontzeptua honako eran defini dezakegu:

```
(and Hitzak
     (all estandarraDagokio HitzEstandarrak)
     (atLeast estandarraDagokio 1))
```

hau da, `HitzEzEstandarrak` kontzeptua `Hitzak` kontzeptuaren umea da. Kontzeptu honen instantziek, bestalde, `HitzEstandarrak` kontzeptukuoa den indibiduo batekin edo gehiagorekin erlazionaturik egon daitezke —gutxienez, baina, elementu batekin erlazionaturik egon behar du—, `estandarraDagokio` *rol*aren bitartez.

*Rol*en balioan indibiduo jakin bat izatearen murriztapena ezar daiteke. Horrela, honako espresio honek:

```
(createRole kat true) ;; kat atributua
(createConcept KategoriaGramatikalak
               (and THING
                    (all kat (oneOf "IZE" "ADI" "ADJ")))
                true) ;; kontzeptu primitiboa
(createConcept Izenak
               (and KategoriaGramatikalak
                    (fills kat "IZE"))
               false) ;; definizio-kontzeptua
(createConcept Aditzak
               (and KategoriaGramatikalak
                    (fills kat "ADI"))
               false) ;; definizio-kontzeptua
```

III.8 Irudia: Kategoria gramatikalen hierarkia, CLASSICen bitartez adierazia

```
(fills urtea 1980)
```

`urtea` *rol*ean `1980` indibiduoa duten indibiduo multzoa adieraziko du.

III.8 irudian kategoria lexikalak adierazten dituzten kontzeptuen hierarkia ikus daiteke. Bertan, `KategoriaGramatikalak` kontzeptuaren definizioa dago, eta baita ere kontzeptu horren bi ume, hots, `Izenak` eta `Aditzak`. Bi ume hauek definizio-kontzeptuak dira, hau da, indibiduo bat kontzeptu horietako baten umea izateko baldintza beharrezkoa eta nahikoa zehazten dute.

CLASSICen bitartez indibiduoak ere sor daitezke. ABoxeko indibiduoak sortzerakoan, halaber, ez dugu zertan indibiduo horren informazio guztia adierazi behar. Horren ordez, une jakin batean indibiduoari buruz dakigun informazioa zehatz daiteke, eta, aurrerago, informazio gehigarriarekin osatu[34]. CLASSIC automatikoki saiatuko da indibiduoa dagokion kontzeptuaren pean sailkatzen. Sailkapena dinamikoa da, indibiduo horri buruzko informazio gehiago esleitzerakoan, indibiduo bera berriro sailkatuko baita alderdi terminologikoaren kontzeptu-hierarkian.

Demagun, III.8 irudiko kontzeptu-hierarkia izanik, `ind` indibiduoa sortzen dugula:

```
(createIndividual ind KategoriaGramatikala)
```

[34]Betiere informazio gehigarria ez bada aurrekoarekin kontraesankorra, jakina.

hau da, `KategoriaGramatikala` kontzeptuaren instantzia. Demagun, orain, indibiduo horri buruzko informazio gehiago dakigula, eta horrela jakinarazten diogula CLASSIC sistemari:

```
(addToldInformation ind (fills kat "ADI"))
```

ondoren, CLASSICek `ind` indibiduoa kontzeptu-hierarkian birkokatuko du, `Aditzak` kontzeptuaren pean ipiniz.

Azkenik, CLASSICen bitartez erregelak ere defini daitezke. Erregelek honako egitura jarraitzen dute:

```
(createRule erregelarenIzena
            aurrekaria
            atzekaria)
```

Adibidez, honako erregela honek:

```
(createRule aditzakAzpikatErregela
       (and Aditzak
            (fills azpikategoria "da"))
            (fills aditz-mota "iragangaitza"))
```

zera adierazten du: `azpikategoria` rolean "da" balioa duten `Aditzak` kontzeptuko instantzia orok `aditz-mota` *rol*ean "iragangaitza" balioa duela.

Hala ere, erregelen bitartez gehitzen den informazioa ez dagokio indibiduoaren definizioari, eta, hortaz, indibiduoa sailkatzeko orduan ez da kontuan hartuko. Halaber, erregeletan adierazitako informazioa ez da inplikazio logikoa bezala ikusi behar. Horren ordez, erregelak CLASSICek aurreranzko kateamendu baten bidez abiatuko dituen disparadoreak dira. Izan ere, sisteman indibiduo berri bat gehitzerakoan, erregela baten aurrekariarekin bat datorrenetz egiaztatuko du, eta, hala izanez gero, erregelaren atzekariko informazioa gehituko dio.

CLASSIC tresna oso egokia da informazio-integrazioko sistemetan, eta iturri lokalak zein eskema orokorra modelizatzeko erabili ohi da[35].

[35] *Information Manifold* proiektua (Kirk *et al.*, 1995; Levy *et al.*, 1996) edo OBSERVER sistema (Mena *et al.*, 1996) CLASSICez baliatzen dira eskuartean dituzten iturriak modelizatzeko.

- Osatu gabeko informazioa adierazteko aukera ematen du, eta, horrela, indibiduoei buruz dakigun informazioa gehitzen joan gaitezke, baldin eta, jakina, informazio berria aurretik adierazitakoarekin inkonpatiblea ez bada. Informazio gehigarria gehitzerakoan, indibiduoa dagokion kontzeptuaren pean kokatuko da, automatikoki.

- Aurreko puntuaren ondorioz, CLASSICek *mundu irekiaren asuntzioa* hartzen du, informazio-integrazioko sistemek hartu ohi duten bezala. Informazio partziala kudeatzen duenez, ezin da ziurtatu indibiduoei buruzko informazioa osoa denik, hots, mundu errealarekin bat datorrenik[36]. Horretarako, CLASSICek `close` eragile berezia du, zeinaren bitartez adierazten den indibiduo bati buruz dakiguna osoa dela, hots, ezin dela indibiduo horri informazioa gehitu, eta, hortaz, indibiduoari buruz dakigun informazioa bat datorrela mundu errealarekin.

- CLASSIC sistema irekia da, bere helburu nagusia konputagailuz lagunduriko software-ingeniaritzarako tresna izatea baita. Horrela, bada, CLASSIC programazio-lengoaia[37] arrunten liburutegi bezala erabil daiteke. Halaber, lengoaia ostalariko oinarrizko datu-motak (zenbaki osoak, zenbaki errealak, karaktere-segidak, eta abar) CLASSICen integratu ahal dira. Bestalde, CLASSICek, erregelen bitartez, produkzio-sistema baten erara funtziona dezake.

Ezaugarri hauek guztiak direla medio, CLASSIC sistema oso irekia eta malgua da, eta tresna paregabea dugu ezagutzaren gainean kudeaketa- eta arrazoibide-prozesuak gauzatzeko.

III.6 Datu-integraziorako zenbait sistema.

Hainbat sistema garatu da datu-integrazioa gauzatzeko, eta, kapitulua amaitzeko, sistema horietatik gure lanerako interesgarrienak suertatu direnak aztertuko ditugu. Sistema hauek guztiek badute hainbat ezaugarri komun, horietatik nagusiena integraziorako arkitektura delarik. Izan ere, sistema oro baliatuko baita bitartekoetan oinarritutako arkitektura delakoaz, hots, bitartekoez eta *wrapper*-ez.

[36] Nahiz eta jakin mundu errealeko indibiduoak dakigun informazioa beteko duela.

[37] CLASSIC sistema, hasiera batean, LISP eta C programazio-lengoaietarako liburutegia zen. NeoClassic sistema, berriz, C++ lengoaiako liburutegia da.

Geroago, gure integrazio-proposamena azaldu ondoren, alegia, gurearen eta sistema hauen arteko aldeak eta desberdintasunak aztertuko ditugu, hurrengo kapituluko IV.9 atalean[38].

TSIMMIS.

TSIMMIS proiektua integrazio-sistemen arteko estreinetarikoa dugu (Chawathe *et al.*, 1994). Bere helburua hainbat informazio-iturriren atzipena ahalbideratuko duten tresnen garapena da, eta, aldi berean, baliabide edo iturri horietatik jasotako informazioaren egonkortasuna bermatzea. Integra daitezkeen baliabideak askotarikoak eta dinamikoak izango dira; informazioa egituratua edo sasi-egituratua izan daiteke, eta, maiz, datu lokalak deskribatzen dituen eskemarik ez da egongo. Datu-iturrien edukiak, bestalde, aldakorrak izan daitezke.

Sisteman integrazio-lana giza-arduradunen menpe egongo da. Izan ere, TSIMMIS sisteman informazio-integrazioa ez baita erabat automatikoa izango. Horren ordez, bere helburua gizakiari datu-iturriak integratzeko lagunduko dion tresneria eskaintzea da.

Bere arkitektura nagusia bitartekoetan oinarritzen da: aplikazioak bitartekariez osaturiko sare batera zuzenduko ditu galderak, sare honen bukaeran datu-iturriak daudelarik; datu-iturri bakoitzaren gainean, datu-objektu lokalak sistemak ulertzen duen datu-eredu orokorrera bihurtuko dituzten *wrapper*-ak daude. TSIMMIS sistemak OEM (*Object Exchange Model*) deituriko datu-eredu orokorra erabiltzen du bitartekoen arteko komunikazioa bermatzeko. OEM eredua, nahikoa sinplea izanik, integraziorako aproposa da, besteak beste, objektu sasi-egituratuak adierazteko aukera ematen baitu. OEM ereduaren gainean LOREL galdeketa-lengoaia ere garatu da (Abiteboul *et al.*, 1997). LOREL lengoaia OQL lengoaiaren hedapena da, OEM objektuekin lan egiteko prestatua, eta, hortaz, datu sasi-egituratuen gaineko kontsultak egiteko aukera emango du.

Sistemaren beste helburu bat bitartekoen zein *wrapper*-en garapen sasi-automatikoa ahalbideratzea da. Bitartekoak zein *wrapper*-ak automatikoki sortu ahal izateko, beren funtzionalitatea zehaztuko duten maila altuko deskribapenak erabiliko dira. Horrela, bada, bitartekoak definitzeko balio duen MSL lengoaia, edota *wrapper*-ak definitzeko balio duen WSL lengoaia ere garatu dira. Bitartekoen definizioak era deklaratiboan zehaztuko dira, konpila-

[38] 219. orrian.

dore batek, deklarazio horietaz abiatuz, bitartekoaren kodea sortuko duelarik. Jakina, bitartekoren bat aldatu nahi izanez gero, aldaketa hauek deklarazioetan egin beharko dira, eta gero bitartekoa birkonpilatu.

TSIMMIS sistemak ez du domeinuaren eredu globalik. Bitartekoak garatzeko, berak eskura dituen iturrien deskribapenekin aski da. Hori horrela izanik, erabiltzailearen esku geratuko da bitarteko bakoitzak eskaintzen duen informazio inplizituaren ulermena.

OBSERVER.

OBSERVER (*Ontology Based System Enhaced with Relationships for Vocabulary hEterogeneity Resolution*) informazio-sistema globalak (Mena *et al.*, 2000) banatuta dauden datu-iturriei galderak zuzentzeko aukera ematen du. Haatik, erabiltzaileak sistemari galdera bat egiterakoan, ez du datu-iturrien kokapen fisikoez edo datuen egituraz arduratu behar izango. Datu-iturri bakoitzaren antolamendua erabat heterogeneoa izan daiteke: fitxategi-sortak, datu-baseak etab. OBSERVER sistemak iturri berriak une oro integratzeko aukera ematen du, eta, horretarako, ontologiak oinarrizko osagaiak dira. Izan ere, sistema hau ontologia anitzen hurbilpenaren adibide garbia dugu.

Ontologia bakoitzak zenbait datu-iturriren semantika islatuko du eta, horretarako, ontologia bera definitzeko deskribapen-logiketan oinarritutako CLASSIC lengoaia (ikus gorago III.5.2 atala) erabiltzen da. Bestalde, ontologiak domeinu zehatzeko adituek sortuak izango dira, eta, beraz, domeinu horretako informazio doia adieraziko dute.

Datu-iturri bakoitza sistema osoan integratuko duen giza-arduradunaren eskuetan egongo da iturri bateko zein informazio esportatuko den erabakitzea, edota iturri horrekin bat datorren ontologia zehaztea. Azkenik, ontologia honen terminoak iturri lokalaren eskemako osagai diren objektuekin lotuko ditu, erregelen bitartez.

Sistemari galderak igorri nahi dizkion erabiltzaileari ontologia sorta bat eskaintzen zaio, eta bere esku du zein ontologia erabili galdera formulatzeko (erabiltzaile-ontologia). Behin galdera OBSERVERi igorriz gero, honek galdera analizatu, osagaietan banatu, osagai horiek erabiltzaile-ontologiarekin erlazionaturik dauden datu-iturriei igorriko dizkion plana garatu, datu-iturriak aurkitu, azpigalderak banatu, emaitzak jaso eta konbinatu, eta, azkenik, erabiltzaileari emaitza bateratua aurkeztuko dio.

Hasierako erantzun honetan, sistemak erabiltzaile-ontologiarekin zerikusi zuzena duten datu-iturriak erabiltzen ditu. Erabiltzaileak jatorrizko galde-

rari buruzko informazio gehigarria nahiko balu, sistema galdera "itzultzen" saiatuko da: ontologia jakin batean oinarritua zegoen jatorrizko galdera ontologia horrekin erlazionaturik dauden beste ontologien terminoen arabera jarriko du. Itzulpena gauzatu ahal izateko, OBSERVER sistema ontologien arteko baliokidetzez baliatzen da eta, horrela, ontologiak binaka erlazionatzen dituzten loturak gordeko ditu. Lotura hauek ontologien arteko terminoak erlazionatuko dituzte, honako era hauetan:

- **Sinonimia.** Terminoak baliokideak direnean.
- **Hiponimia.** Terminoren bat bestea baino murritzagoa denean.
- **Hiperonimia.** Terminoren bat bestea baino orokorragoa denean.
- **Ebakidura.** Bi terminok informazioa konpartitzen dutenean.
- **Disjuntzioa.** Bi terminoren esanahia erabat desberdina denean.
- **Estaldura.** Termino baten esanahia beste hainbat terminoren esanahien bildura denean.

OBSERVER sistemaren arkitektura banatuan, seguruenik kokapen fisiko desberdina[39] duten hainbat *nodo* komunikatzen dira eta, horietatik, *erabiltzaile-nodoa* da erabiltzailearekin elkarrizketatuko den bakarra. Bestalde, *nodo* bakoitzak datu-iturri multzo bat kudeatuko du. Hauek dira nodo baten osagaiak:

- **Galdera-prozesatzailea.** Prozesatzaileak erabiltzaileak aukeratu duen erabiltzaile-ontologiaren terminoetan osatutako DLko galdera jasoko du. Ondoren, galderako terminoekin zerikusirik duten ontologiak aukeratuko ditu, galdera ontologia hauetara itzultzearren. Itzulpenak partzialak izan daitezke, beharbada, jatorrizko murriztapenen bat ezin delako ontologia zehatz batean adierazi. Horrela, bada, prozesatzaileak itzulpen partzial guztiak konbinatuko ditu, beti ere, jatorrizko galderaren semantika zainduz. Gero, itzulitako galderarekin bat datozen datu-iturriak bilatuko ditu. Azkenik, baliabide/ontologia desberdinetatik jaso diren datuak konbinatu eta erabiltzaileari aurkeztuko dizkio.

[39]Izan ere, Interneteko *web* orriek osatutako amaraunaren gainean informazio-sistema globala eskaintzea baita OBSERVERen helburuetariko bat.

- **Ontologia-zerbitzaria.** Nodo bakoitzean dauden ontologiak kudeatuko ditu. Horretaz gain, ontologia bakoitzak adierazten duen datu-iturriak ere kudeatuko ditu, eta, horrela, ontologiaren baten araberako galderak datu-iturrietara bideratuko ditu.

- **Ontologien arteko baliokidetzen kudeatzailea.** Modulu honetan, ontologia desberdinen arteko terminoak lotzen dituzten erlazioak gordeko ditu. Terminoak elkarrekin lotzeko, arestian aipatutako erlazio motak erabiliko ditu. Modulu honen bitartez, hiztegi banatuaren arazoari aurre egingo zaio.

OBSERVER sistemak *globala bistatzat* paradigma jarraitzen du, datu-iturriak sistema banatuaren parte-hartzea bermatzen duten erregelak definitzerakoan. Erregela hauek, bestalde, datu-iturri zehatz bakoitzeko adituren batek bideratuko ditu. Paradigma bera jarraituko da, halaber, sistema honen berezitasuna diren ontologien arteko loturak zehazterakoan.

OBSERVER sistemaren ezagumendua adierazteko deskribapen-logiketan oinarritutako ontologiak erabiltzen direnez, ontologia hauen gaineko ezagutza, meta-ezagutza, modelatzeko aukera dago. Ezagutza osagarri honek, mapaketa eta eskemaren gainean eragiketa indartsuak egiteko aukera emango digu, hala nola, datu-integrazioan hain garrantzitsuak diren itzulpen-planak osatzeko arras baliagarriak diren galdera-patroiak erabiltzeko (Bermúdez, 2001).

SIMS.

SIMS integrazio-sistemaren xede nagusia banatuta dauden informazio-sistema heterogeneoen atzipen adimentsua eskaintzea da (Arens eta Knoblock, 1992; Arens *et al.*, 1996). Horrela, SIMSen erabiltzaileak hainbat informazio-iturri atzitu ahal izango du, era uniforme batean, iturriek duten berezko antolamenduaz edo kontsulta-lengoaiaz arduratu behar izan gabe.

SIMS sistema aplikazio-domeinuarekiko independentea da, alegia, berak proposatutako integrazio-arkitekturak edozein domeinutako informazio-iturriak integratzeko aukera ematen du. Bere helburua aurrera atera ahal izateko, SIMSek aplikazioaren *domeinu-eredua* osatzen du, LOOM ezagutzaren errepresentaziorako lengoaiaz. Domeinu-ereduak domeinuko objektuak, beren atributuak eta objektuen arteko erlazioak deskribatzeko oinarrizko hiztegia ezartzen du, eta, beraz, integrazio-sistema osoko ontologia banatuaren

funtzioa betetzen du. Sistemari igorritako galderek ere domeinu-ereduaren arabera adierazita egon behar dute.

SIMS KL-ONE familiakoa den LOOM lengoaia aberatsaz baliatzen da domeinu-eredua, iturrien ereduak eta galderak adierazteko. LOOMen bidez zehazten dira klaseen definizioak, klaseen arteko erlazioak (hierarkikoak eta abar), eta *rol*en definizioak. Horretarako, LOOMek eskainitako oinarrizko objektuez baliatzen da, hots, klaseez (edo kontzeptuak) eta *rol*ez. Ohi bezala, lehenengoek instantzia multzoak adierazten dituzte, eta bigarrenek, berriz, klaseen atributuak.

Informazio-iturri bat SIMSen integratzeko, bere edukia LOOMez modelatu behar da lehen; gero, iturriko kontzeptu eta erlazioak domeinu-eredukoekin lotu behar dira, `IS-Link` ("Information Source Link") delakoen bidez. `IS-Link`en noranzkoa informazio-iturrietatik domeinu-eredura da[40], eta bi kontzeptu lotzen dituenean, iturriko kontzeptuak eta domeinu-ereduko kontzeptuak indibiduo-sorta bera adierazten du, nahiz eta iturriko kontzeptuen atributuek ez dituzten domeinu-ereduak kontzeptu horretarako aurreikusita dituen guztiak adierazi behar. Horrela, bada, SIMSek, `IS-Link`en bidez, informazio-iturrien informazioaren semantika esplizituki adierazten du.

Iturrien ereduek, beren erlazioekin zuzenean lotuta dauden domeinu-erlazio zein kontzeptuekin batera, *eredu minimoa* delakoa osatzen dute SIMSen. Eredu minimoaren domeinuko erlazio zein kontzeptu orok `IS-Link`ez horniturik izan behar du, iturrietako kontzeptu eta erlazioekin lotu ahal izateko.

Horretaz gain, sistemaren malgutasuna areagotzeko, SIMSek beste abstrakzio-geruza bat gehitzen du. *Eredu zabaldua* delako geruza horretan, domeinu-ereduko kontzeptu eta erlazio berriak definitzen dira, LOOMez, eredu minimokoak diren domeinu-ereduko kontzeptu zein erlazioen gainean. Horrela, bada, domeinuko kontzeptu edo erlazio aberatsagoak definitu ahal izango dira, erabiltzaileari —dela giza-erabiltzailea, dela konputagailu bidezko softwarea— aukera zabalagoak eskainiz, galderak sistemari igortzerakoan. Eredu zabalduaren kontzeptu eta erlazio berriek ez dute erlazio zuzena iturriko erlazioekin; bai, ordea, eredu zabalduaren kontzeptuak definitzeko erabili diren eredu minimoko kontzeptu eta erlazioek.

Domeinu-ereduaren arabera dauden galderak erantzuteko, SIMSek hainbat ataza burutzen ditu:

- Galderen birformulazioa. Galdera erantzun ahal izateko informazio-iturriak identifikatu eta, erabiltzaileak eskatutako informazioa lortzearren,

[40] LAV hurbilpena jarraitzen du, beraz, SIMSek.

iturriko datuak zein eratan konbinatu behar diren erabakitzen du. Horretarako, SIMSek jatorrizko galdera —domeinu-ereduko kontzeptu eta erlazioez adierazita— informazio-iturri zehatzek ulertzen duten galderetara itzultzen du.

- Plangintza. Galderaren itzulpenak lortu ondoren, berridazketek adierazitako informazioa eskuratuko duten plan logikoak eraikitzen dira. Planaren barruan, besteak beste, eginkizun hauek burutzen dira: informazio-iturri zehatz batera galdera jakin bat bidali, datuak iturri batetik beste batera mugitu, iturri desberdinetatik datozen datuak elkartu eta emaitza partzialak behin-behinean gorde.

- Galdera-planen optimizazio semantikoa. SIMSek iturrietan metatutako edukien gainean ikasketa automatikoa burutzen du, planak semantikoki optimizatzearren. Ikasketari esker, iturri bakoitzeko informazioa taxuz eskuratzeko behar diren optimizazio-erregelak automatikoki sortuko ditu.

- Exekuzioa. Azkenik, plan optimizatua exekutatuko du, hots, jatorrizko galdera erantzuteko egokiak diren iturrietara galderak paraleloan igorri, datuak transferitu, eta erabiltzaileari eskaini behar zaion erantzuna osatuko du. Exekuzioa arrakastatsua ez bada, hots, emaitzarik lortu ez badu, SIMSek galderaren zati bat —edo galdera osoa— birplanifikatuko du. Exekuzioa gauzatzeko, SIMS ere *wrapper* teknologiaz baliatzen da.

Erabiltzaileak ipinitako jatorrizko galdera birformulatzerakoan, SIMS hainbat birformulatze-eragilez baliatzen da. Eragile bakoitzaren aplikazioak galderaren zenbait klausula ordezkatzen ditu, semantikoki baliokideak diren beste klausula batzuekin. Bi eragile mota daude: lehenengoak *eredu zabalduaren* kontzeptu eta erlazioen gainean aplikatuko dira, kontzeptu zein erlazio horiek eredu minimokoak diren beste terminoekin ordezkatuz; bigarrenek, berriz, eredu minimoko terminoak iturrikoekin ordezkatuko dituzte, `IS-Link`eko informazioa ustiatuz. Eragileak behin eta berriro aplikatzen zaizkio galderari, prozesu iteratibo batean, eskuratu beharreko informazioa gordetzen duten iturrien arabera soilik adierazita egon arte. Halaber, berridazketek esplizituki adieraz dezakete, jatorrizko galderaren erantzuna lortzearren, iturrietako informazioa zein eratan konbinatu behar den.

Birformulatze-eragileek, jatorrizko galderaren gainean aplikatu ostean, iturrietako erlazioetara zuzenduko diren hainbat berridazketa sortuko dituzte, baina, jakina, jatorrizko galdera bererako hainbat berridazketa desberdin osa dezakete. Izan ere, galdera bakar baterako egon daitezkeen berridazketa posibleen espazioa handia da oso. Hala ere, zenbait berridazketa besteak baino hobeak izango dira, eraginkorragoak diren planak egiteko aukera emango baitute, hots, atzipen-kostua txikiago duten iturrietara jotzen duten planak, edo beren baitan dituzten datu-mailako eragileak (datuak mugitu, elkartu, hautatu etab.) zama txikiagoa dutenak. Beraz, berridazketa posibleen espazioan zehar galdera baten berridazketa optimoa topatuko duen *branch-and-bound* motako algoritmoa du SIMSek: algoritmo honen sarrera jatorrizko galdera eta birformulatze-eragileak dira, eta, emaitza gisa, galderaren berridazketa optimoa lortuko du, berridazketa exekutatzeko behar den planarekin batera.

SIMS sistema LAV hurbilpena jarraitzen dutenen artean lehenengoetarikoa da. Halaber, aplikazioaren domeinu-eredua, ezagutzaren errepresentazioko lengoaia aberatsean adierazita egotea —eta, hortaz, sistemaren arkitekturarekiko independentea izatea— oso garrantzitsua da, datuen gaineko errepresentazio abstraktua egiteko aukera ematen baitu. Iturri eta domeinuereduko erlazioak adierazteko, `IS-Link`ez baliatzen da SIMS. Esteka hauek nahikoa sinpleak dira, soilik kontzeptu-kontzeptu edo erlazio-erlazio loturak egiteko aukera ematen baitute. Ezin dira, hortaz, iturriak eta domeinu-eredua lotzen dituzten espresio konplexuak erabili. Integrazio birtuala gauzatzen du SIMSek; izan ere, erabiltzaileak ipinitako galderak ezarriko baitu bere erantzuna lortuko duen plan logikoa, plana exekuzio-denboran kalkulatuko baita. Plana aurkitzeko, SIMS bilaketa-algoritmo orokor batez baliatzen da: jatorrizko galderaren gainean birformulatze-eragileen aplikazioaren ordena asmatuko duen algoritmoak galderaren berridazketa optimoa —plan eraginkorrenak egiteko aukera ematen duena— bilatuko du. Plana optimoa noiz den estimatzeko, informazio-iturrien gainean ikasitako optimizazio-erregelak erabiltzen ditu SIMSek.

Information Manifold.

AT&T enpresan garatutako Information Manifold (IM) proiektuak (Kirk *et al.*, 1995; Levy *et al.*, 1996) datu-iturri anitzetan informazioa eskuratzeko interfaze komuna eskaintzen du. Bere helburu nagusia, datu-integrazioa gauzatzerakoan, bikoitza da. Batetik, datu-iturri anitz integratu, era malguan,

iturriek gordetzen duten informazioa oso antzekoa izan daitekeela kontuan izanik. Bestetik, IMren erabiltzaileek jarritako galderak modu eraginkorrean erantzun, galdera erantzuteko aproposak diren iturriak soilik atzituz.

IMren arkitektura nagusia —datu-integrazioaren arloan aurkitu ohi den legez— informazio-iturriak adierazteko aukera ematen duen domeinu-eredu aberats eta hierarkiko batean oinarriturik dago. Domeinu-eredua, aplikazio-eremu jakin baterako beharrezkoak diren kontzeptu zein erlazioez horniturik dagoena, ezagutza-base batean gordetzen da.

Informazio-iturriak IMn integratzeko, ohi den bezala, informazio-iturri bakoitzaren eduki-adierazpena adierazi behar da. Iturrien eduki-adierazpenak bi osagai nagusi ditu: iturriko kontzeptu zein erlazioen modelizazioa, eta iturriaren ereduaren eta domeinu-ereduaren arteko mapaketa semantikoa gauzatuko duen eduki-deskribapena. Bertan, iturriko kontzeptu zein erlazio bakoitzeko, galdera konjuntibo bat idatziko da, domeinu-eredukoak diren kontzeptu zein erlazioetan oinarrituz. Horretaz gain informazio-iturrietako edukiaren gainean murriztapen konplexuak adierazteko aukera ematen du IMk.

IMko domeinuak —domeinu-eredua zein iturrietakoa—, eta iturrien eduki-deskribapenak CARIN lengoaiaz (Levy eta Rousset, 1998) adierazten dira. CARIN lengoaia hibridoa da, Horn erregelak eta $\mathcal{ALCNR}$ deskribapen-logika[41] uztartzen baititu: $\mathcal{ALCNR}$ osagaiaz iturrien zein domeinu-ereduko kontzeptuen arteko erlazio hierarkiko konplexuak modeliza daitezke, eta mapaketa semantikoa, berriz, Horn klausulen bitartez adierazten ahal da. CARIN adierazpen-lengoaia oso aberatsa da, eta informazio-integraziorako egokia da. Izan ere, CARINez adierazitako informazioren gainean inferentziak egitea erabakigarria da [42]. Esan bezala, iturrietako edukiaren gaitasunak adieraz daitezke IMn. III.3.4 atalean ikusi dugun legez, zenbait iturrik murriztapenak ezartzen dituzte galderak egiterakoan, eta, horrela, gerta daiteke zenbait atribuzuri buruz soilik galdetu ahal izatea. Informazio hori guztia, bada, edukiaren gaitasunetan gordeko du IMk eta, horrela, sistemak jakingo du, informazio-iturrietara datuen bila jotzerakoan, iturri horren datuak eskuratzeko dauden galdeketa-murriztapenak. Edukiaren gaitasunei esker, hortaz, mota anitzeko informazio-iturriak integra daitezke IMn.

[41] Deskribapen-logikak kapitulu honen III.5 atalean azaltzen dira.

[42] Hala ere, nahiz eta inferentziaren prozesua erabakigarria izan, bere konplexutasuna nahiko handia da. Hortaz, kontzeptu-lengoaia xumeagoak erabili izan dira ere CARINen, esaterako, co-CLASSIC lengoaia. Lengoaiaren espresibotasuna urrituz, inferentzia-prozesua franko eraginkorragoa da.

Domeinu-ereduan adierazitako galdera jasoko du IMk, eta berak galdera hori informazio-iturrietara igorriko du. Igorri baino lehen, baina, galdera iturri bakoitzak ulertuko duen eredura itzuli behar du. Galderaren itzulpena gauzatzeko, IM sistemak *Bucket Algorithm* delakoa erabiltzen du. IMk, algoritmo horren bidez[43], galdera erantzuteko egokiak diren iturrietara soilik joko duten planak osatuko ditu. IMk trataera berezia egiten du zenbait kasutan. Konparazio batera, iturri batek informazio bati buruzko datu *osoak* dituela adierazi ahal da, eta IM informazio horretaz baliatuko da galdera-itzulpenaren garaian. Esate baterako, iturri batek galdetutako informazio bati buruzko datu guztiak dituela adierazten bazaio, ez du beste iturrietara joko informazio horren bila.

Bestalde, eta informazio jakin bati buruz egokiak diren iturri anitz egon daitezkeela kontuan izanik, hots, galdera erantzuteko plan ugari egongo dela aurreikusiz, IM datu probabilistikoez baliatzen da itzulpen-algoritmoak osatutako plan horiek guztiak ordenatzeko. Datu probabilistikoek iturrien egokitasuna sailkatzeko balioko dute, beraz.

IM sistemak oso garrantzi handia izan du datu-integrazioaren ikerlerroan. Sistemaren garapena ikerlerroa hastapenetan zegoela egin zelako edo, sistema honek ezarri du datu-integrazioko arazoa formalki aztertu ahal izateko hainbat termino. Horrela, LAV hurbilpeneko datu-integrazioaren nondik norakoak aztertu zituen lehen ikerlana dela esan daiteke. Horretaz gain, galderen berridazketak itzultzen dituen lehenengo algoritmo berezia garatu zuen, (Levy *et al.*, 1995) lanean oinarriturik. Algoritmoak, SIMSeko planifikatzaile orokorrak ez bezala, azpian dagoen adierazpen-sisteman oinarrituta dauden arrazoibide-mekanismoak erabiltzen ditu, galdera bat itzultzean egokiak diren iturrietara soilik joko dela bermatuz.

Infomaster.

Infomaster (Genesereth *et al.*, 1997; Duschka eta Genesereth, 1997b) jadanik existitzen diren informazio-iturriak integratzeko sistema orokorra dugu. Iturriak mota anitzekoak izan daitezke: datu-base erlazionaletik *web* orrietako informazio sasi-egiturauraino. Horretarako, sistemak ohikoa den integrazio-arkitektura du, hau da, *wrapper* eta bitartekoetan oinarritutako arkitektura.

Infomaster sistemak malgutasun handia eskaintzen du informazio-iturriak sisteman gehitu edo aldatzeko. Horretarako, hiru mailako eredua jarraitzen

[43] Ikus III.3.3.2 atala.

du sisteman zehar behar diren kontzeptu zein erlazioak modelatzeko. Batetik, interfaze-eredua dago, sistemaren erabiltzaileekin elkarrizketa bermatuko duena; erabiltzaileek, sistemari galderak jartzerakoan, interfaze-ereduko kontzeptu zein erlazioez baliatu behar dute. Bestetik, oinarri-eredua dago, hots, sistemaren muineko kontzeptu zein erlazioak. Azkenik, iturrien ereduak daude, hau da, iturri bakoitzaren informazioa adierazten duten kontzeptu eta erlazioak.

Interfaze-eredua eta iturrien ereduak oinarrizko ereduarekin lotzen dira, mapaketa semantikoen bidez. Lehenengo motako mapaketak definizioen bidez gauzatzen dira, hots, interfaze-ereduko kontzeptu zein erlazio ororen deskribapena zehazten da, oinarri-ereduko osagaiak erabiliz.

Bigarrenek, ostera, desberdintasun-murriztapenak eduki ditzaketen *datalog* erregelen bitartez gauzatzen dira, iturrien integritate-murriztapenekin batera. *Datalog* erregelek oinarri-ereduaren gaineko bistak definitzen dituzte —LAV hurbilpena jarraitzen dute, hortaz—. IM sistemaren antzera, bistak definitzeko bi erregela mota daude, hots, inklusio-erregelak eta identitate-erregelak. Lehenengo motakoen esanahia datu-integrazioan izan ohi duena da, eta mundu irekiaren asuntzioarekin bat dator. Bigarrenek, ordea, erregelako ezker eta eskuin aldeak multzo *bera* deskribatzen dutela adierazten dute, hots, informazio-iturri batek informazio *guztia* gordetzen duela. Sistemaren planifikatzailea informazio horretaz baliatuko da galderen gainean optimizazioak egiterakoan.

Hiru prozesu betetzen ditu Infomaster-ek, erabiltzaileak jarritako galdera erantzuteko:

1. Galderako predikatuak oinarri-eredura itzuli. Prozesu sinplea da lehenengo hau, galdera-hedapena egitea besterik ez baita behar.

2. Galdera prozesatu. Behin galdera oinarri-ereduarekin bat datorrenean, galdera erantzuteko egokiak diren iturrien ereduetara itzuli behar da. Prozesu honetarako algoritmo propioa garatu dute Infomaster proiektuan, alderantzizko erregelen algoritmoa izenekoa[44].

3. Galdera optimizatu. Prozesatzailearen irteera iturriei zuzendutako galderak egingo dituzten plan logikoak dira. Azken prozesu honek sistemak iturriei buruz dakien informazioa ustiatzen du, planen exekuzioaren eraginkortasuna minimizatzeko. Azkenik, plan optimoa exekutatuko du, galderako erantzun-datuak eskuratzearren. Galderak iturriei

[44] III.3.3.2 atalean azaldu dira algoritmo horren xehetasunak.

zuzentzerakoan eta erantzunak jasotzerakoan KIF izeneko lengoaiaz baliatzen da.

Ariadne.

Ariadne sistema SIMS sistemaren hedapena dugu, eta bere helburu nagusia Interneteko *web* orrietan gordetako informazioa integratzen duten informazio-agenteak era eraginkor eta azkar batean garatzeko eredua eskaintzea da (Knoblock *et al.*, 2001; Tejada *et al.*, 2001). Hortaz, Ariadne sistemak datubase sasi-egituretan jartzen du arreta.

Hiru osagai ditu Ariadne-k proposatutako integrazio-ereduak: aplikazio-domeinuaren eredua, galdera-planifikatzailea eta *wrapper*-ak. Bere arkitektura agenteetan oinarriturik dago, eta, horrela, informazio-agenteek egingo dituzte bitarteko eta *wrapper* lanak.

Bere asaba den SIMS sistemarekin hainbat ezaugarri komun ditu. Esaterako, domeinu-eredua eta informazio-iturrien ereduak LOOM lengoaiaz adierazten dira, eta haien arteko mapaketa semantikoak LAV hurbilpenari jarraikiz adierazten dira.

Domeinu-ereduan adierazita dagoen jatorrizko galderan eskatutako informazioa lortuko duten plan logikoak osatzeko, berriz, planifikatzaile orokor batean oinarritzen da Ariadne. Planifikatzailearen konplexutasuna zein exekuzio-denbora urritzeko, LAV eredua jarraitzen duten iturrien deskribapenak konpilatzen dira, GAV hurbilpeneko integrazio-axiometara bihurtuz. Konpilazioa lan konplexua da oso, baina, bestalde, behin bakarrik egin beharreko prozesua da.

Ariadne sistemaren beste helburu bat *web* orrien gaineko *wrapper*-en garapenerako laguntza-tresnak eskaintzean datza. *Wrapper* batek, *web* orrietan adierazitako informazioa integrazio-sistema osora eskainiko badu, orriaren eduki semantikoa zein egitura sintaktiko sakona ulertu behar du, hots, *web* orriaren gramatika ezagutu behar du. Lan neketsua izaten da gramatikak zein eduki semantikoak osatzea, eta, maiz, *web* orri jakin baterako eginikoak ez du beste orrietarako balio. Hortaz, Ariadne *web* orri baten egitura semantiko zein sintaktikoa sasi-automatikoki ikasten saiatzen da.

Sistemak integrazio birtuala eta gauzatutako integrazioa elkartzen ditu. Interneteko zenbait orri atzitzeko denbora handia behar duela aurreikusiz, orri horien gainean dauden informazio-agenteek datuak aldez aurretik eskuratu eta datu-base lokalean gorde ditzakete. Hortaz, sistemak agenteak gorderik duen informazioa behar badu, ez da iturrietara jo behar izango:

gordetako datu lokalak —iturriaren eredu semantikoaren arabera adierazita egongo direnak— izango dira eskaeraren emaitza. Jakin beharko da, jakina, zein informazio *gauzatu*, hots, zein informazio merezi duen datu-base lokalean gorderik egotea erabaki beharko da, sistemaren eraginkortasunari begira. Horretarako, Ariadnek zenbait irizpide erabiltzen ditu, hala nola, usuen galdetzen diren galdera-patroiak edo galderak iturrietara igortzeak duen exekuzio-zama. Halaber, erabaki beharko da agenteek gordetako datuak zein frekuentziarekin eguneratu behar diren, iturriko datuen eta modu lokalean gordetako datuen arteko egonkortasuna bermatzeko.

Azkenik, Ariadnek eredu bat eskaintzen du maila estentsionaleko parekatzea gauzatzeko, hots, iturri anitzetatik datozen datu guztiak mundu errealeko entitate beraren adierazpenak direnetz erabakitzeko[45]. Izan ere, *web* orrietan aurki daitezkeen datuak akastunak izaten baitira maiz, eta errore-iturriak askotarikoak izan daitezke. Hortaz, Ariadne sistemak informazio-iturrietako datuak elkarrekin erlazionatzeko ikasketa automatikorako algoritmo bereziak ditu.

PICSEL.

PICSEL integrazio-sistema (Lattes eta Rousset, 2000) deskribapen-logiketan dago oinarriturik. Izan ere, CARIN lengoaian osaturiko ezagutza-base batek egingo baititu, PICSELen, bitartekari-lanak. Ezagutza-baseak aplikazio jakin bateko domeinu-ereduaren informazioa zein integratuko diren iturrien informazioa adieraziko ditu.

PICSELen, lehenik, aplikazio-domeinuaren oinarrizko hiztegia zehaztu behar da. Hiztegi horren gainean, domeinuko kontzeptu zein erlazioak zehazten dira, deskribapen-logiketako espresioak erabiliz.

LAV vs. GAV auzian, erdibideko irtenbidea hartzen du PICSELek. Nolabait, LAV hurbilpeneko malgutasuna eskaintzen du informazio-iturri berriak integratzerakoan, hurbilpen honen konplexutasun-iturri nagusia —galderaren itzulpenaren zailtasuna— ekidinez. Horrela, bi osagai behar dira informazio-iturriak sisteman integratzeko. Lehenengo osagaia, iturrietako eredua domeinu-ereduarekin lotzen duen mapaketa semantikoa, GAV hurbilpena jarraituz osatzen da. Mapaketa semantikoa iturrietako edukiek dituzten murriztapenekin osatzen da, eta murriztapen horiek, deskribapen-logiketan oinarrituta daudenak, LAV ereduaren arabera kodetzen dira.

[45]Datu-arazketa burutzeko, azken finean. Ikus III.3.5 atala.

Galderen itzulpena gauzatzeko, CARIN lengoaiak dituen inferentzia-mekanismo orokorrak erabiliko dira. Horretarako, iturriak eta domeinu-eredua lotzen dituen mapaketa semantikoan dauden erregelak ustiatuko ditu, aurreranzko kateamendu baten bidez, jatorrizko galderaren hedapena lortzeko. Hala ere, sortutako berridazketek iturrien murriztapenak betetzen dituztela egiaztatu behar da, berriro ere, CARIN lengoaiaren inferentzia-ahalmenean oinarrituz.

IV. KAPITULUA

ELHISA: informazio lexikalaren integraziorako arkitektura bat.

Kapitulu honetan, ELHISA sistemaren nondik norakoak aztertuko ditugu, eta, baita ere, sistemaren arkitektura orokorra deskribatu. Arkitekturaren eraketa, jakina, funtsezkoa da, arkitekturaren bitartez hainbat baliabide lexikal eta eleanitz integratuko dituen sistema garatu ahal izango baitugu. Bere osaeran erabili diren irizpideak ere azalduko ditugu, eta, horretarako, ELHISA integrazio-sistemak dituen ezaugarri nagusiak aurkeztuko dira. Ikusiko dugun bezala, ELHISAk integratuko dituen baliabide lexikalen izaerak finkatuko du, neurri handi batean, arkitektura guztiaren nolakotasuna.

Sistema osoaren zati garrantzitsuenetariko bat lan honetan proposatuko dugun informazio lexikalerako Eredu Kontzeptual Orokorrean (EKO) datza, zeinek informazio lexikal orokorra eta heterogeneoa adierazteko aukera emango duen. Kapitulu honetan informazio lexikalaren eredu kontzeptual orokor bat deskribatuko dugu, hortaz, eta, eredu honen bideragarritasuna frogatzearren, egun dauden hainbat baliabide desberdinen ereduak gure eredu orokor honekin parekatuko ditugu.

Ideia hau tesi-lan honen estreinako motibazioa izan bada ere, iturri lexikaletan gorderik dauden datu lexikalen benetako integrazioa eta elkar-trukaketa lortzea hainbat eginkizun osagarriren mende egongo da. Izan ere, integrazio-sistema baten garapena lan konplexua da oso, eta datuak integratzea helburu duen edozein sistemak hainbat buruhauste aurkituko ditu bidean.

Kapitulua ELHISAren arkitekturaren azalpenarekin hasiko da, hortaz.

Gero, aipatutako eredu kontzeptualetan jarriko dugu arreta —integratuko diren iturrienetan zein Eredu Kontzeptual Orokorra deitu dugun horretan—, eta, azkenik, ELHISAk bere lanak ganoraz bete ahal izateko beharrezko diren moduluak azalduko ditugu: eredu kontzeptualaren deskribapen soiletik eredua exekutagarri bihurtzera pasatu nahi dugu, eta jauzi horretan sortzen diren inplementazioari eta konputagarritasunari buruzko galderei erantzuten saiatuko gara.

IV.1 Informazioaren integraziorako arkitektura ELHISAn.

Jadanik azaldu bezala, ELHISA hainbat baliabide lexikal integratzeko informazio-sistema dugu. III kapituluan ikusi dugun legez, informazioaren integrazioa xedetzat duten sistemak konplexuak dira oso, eta integrazioa bera gauzatzeko hainbat hurbilpen dago (integrazio birtuala vs. gauzatutako integrazioa, LAV vs. GAV, etab.). Hurbilpen bakoitzak bere abantailak eta desabantailak ditu, jakina, eta, azken finean, integratu nahi den informazioaren ezaugarriek ezarriko dituzte integrazio-sistema osoaren propietate nagusiak.

ELHISA sistema osatu izana informazio lexikala bateratzeko arazoetatik dator. Lengoaia Naturaleko Prozesamenduaren alorrean informazio lexikalak garrantzi handia du, ikusi dugu. Lexikoiek, LNPrako tresna sendoen hornitzaile lexikalak izan behar badute, estaldura zabaleko informazio lexikal aberatsaren gordeleku izan behar dute. ELHISAren azken xedea, hain zuzen ere, LNPan diharduen orori hain baliagarria den informazio lexikal aberats eta estaldura zabalekoa eskaintzea da, eta, horretarako, informazioa hainbat iturritatik bilduko du. Horrela, bada, ELHISAren bidez eskuratutako informazioa ez da iturri batek ezarritako domeinuetara mugatuko, informazioa hainbat baliabide ezagun eta aberatsetatik jasoko baita, era bateratu batean.

Oso komenigarria izango balitz ere, ez da errealista, egun, informazio lexikal guztia bere baitan onartuko duen adierazpen estandarra osatuko dela suposatzea. Hau horrela izanik, gure proposamena “baliabide lexikalen federazioa” osatzea litzateke, hots, baliabideen gainean inongo bihurketarik egin gabe, euren informazioaz aprobetxatzen den sistema bat eraikitzea.

Informazio-sistema globalen antzera, ELHISAk iturri lokal guztien adierazpena eskema orokor bakar baten arabera adierazten du. Hortaz, iturri guztien artean “lingua franca” antzera jokatuko duen eskema orokor berezi bat sortu dugu, eta eskema horri Eredu Kontzeptual Orokorra (EKO) deitu

diogu. EKOak sistema osoaren terminologia ezarriko du, domeinuko objektuak, euren atributuak eta objektuen arteko erlazioak adieraziz.

Atal honetan zehar azalduko ditugu EKOaren ezaugarri nagusiak, eta baita sistema osoan betetzen duen papera ere. Edonola ere, esan dezagun jada, EKOak bi eginkizun nagusi betetzen dituela integrazio-sistema osoan. Alde batetik, EKOak sistemaren eta erabiltzailearen arteko komunikazioa bermatuko du, erabiltzaileak EKOaren arabera igorriko baitizkio galderak ELHISAri. Sistemari igorritako galderetan ez da adierazi behar, galdera horri erantzuna emateko, zein iturritara joan behar den. Galdera erantzutearren beharrezko datuak jasotzea eta datu horiek konbinatzea ELHISAk bere gain hartu beharreko lana da.

Sistemaren arkitektura orokorra aurkeztu baino lehen, hortaz, ELHISAk dituen ezaugarri nagusiak aurkeztuko ditugu, ezaugarri hauek baitira, neurri handi batean, erabaki estrategikoak hartzerakoan eskuartean erabili ditugun irizpideak. ELHISA sistemaren ezaugarri nagusiak hauek dira:

- **Datuen heterogeneotasuna**.

  Baliabide lexikaletan metaturiko informazioa heterogeneoa izango da, nahiz eta baliabide orok hitz eta hitz-formei buruzko informazio lexikala eskaini.

  II atalean azaldu den bezala, informazio lexikala aplikazio zehatzetara egokiturik egon ohi da. Informazio lexikalaren bezeroak diren aplikazio horiek ezarriko dute, maiz, baliabide lexikalen antolaketa. Aplikazio zein teoria linguistikoekiko independenteak diren baliabide lexikal estandarrak eraikitzea helburutzat izan duen hainbat ekimen izan bada ere (Normier eta Nossim, 1990; MacNaugth, 1990; Uszkoreit *et al.*, 1996; Calzolari *et al.*, 2002; Ruimy *et al.*, 1998; Bel *et al.*, 2000), hitzen arteko erlazio lexikalen izaera guztiz adierazten duen formalismorik ez da garatu. Informazio lexikal hori guztia era uniformean eskuratu ahal izatea oso baliagarria litzateke mota horretako informazioa behar dutenentzat, eta hausnarketa hauxe izan zen, hasiera batean, proiektu osoaren motibazio nagusia. Hasieratik bagenekien, beraz, ELHISAk eskuartean erabiliko duen informazioa, baliabide lexikaletan agertutakoa, alegia, berez heterogeneoa izango dela.

  Heterogeneotasuna, bestalde, maila desberdinetakoa da: baliabide bakoitzak datuak euskarri elektronikoan gordetzeko errepresentazio-sistematik datu hauen antolaketa semantikoraino.

- *Adierazpide-maila*. Datuak euskarri elektronikoan gordetzean, baliabide bakoitzak errepresentazio-sistema desberdinak erabil ditzake. Testu-fitxategi lauetatik, informazioa dinamikoki inferi dezaketen ezagutza-baseetaraino, datu-lexikalak formatu oso desberdinetan paratzen dira.
- *Galdera-maila*. Galdera-maila adierazpen-mailako heterogeneotasunarekin hertsiki loturik dago, adierazpen bakoitzak bere kontsulta-lengoaia baitu. Lengoaia hauek, bestalde, deskribapen-ahalmen desberdinak eduki ditzakete. Edonola ere den, ELHISAk lengoaia hauek guztiak erabili behar ditu, baliabideetatik datuak eskuratu nahi baditu.
- *Maila semantikoa*. Baliabide bakoitzak gorderiko datuak eskema[1] baten arabera metaturik daude, baina eskema horiek, jakina, ez datoz bat bata bestearekiko. Horrela, bada, maila semantikoan kokatzen den heterogeneotasuna ere ebatzi beharko dugu, galdera bati erantzuterakoan, galdera horren esanahia sistema zehatz bakoitzera itzuli egin beharko baita.
- *Datu-maila*. Datu-mailako heterogeneotasuna —heterogeneotasun estentsionala ere deiturikoa— azalduko da, iturri desberdinetatik lortutako objektu desberdin bi edo gehiagok munduko entitate bakar bat erreferentziatzen dutenean. Ikusi dugun bezala, datu-mailako heterogeneotasuna egotearen arrazoiak ugari dira. Hala ere, bere ebazpena funtsezkoa da, iturri anitzetatik jasotako erantzunak elkarrekin erlazionatu nahi badira.

Heterogeneotasun-maila horietatik, lehenengo biek —adierazpen- eta galdera-mailako heterogeneotasunak, alegia— III kapituluan azaldu den datu-integrazioaren *integrazio estrukturala*rekin dute zerikusia, eta, jadanik aipatu dugun bezala, *wrapper* eta bitartekoetan oinarritutako arkitekturak, hein handi batean, auzi horiek lasaitzera datoz. ELHISAren arkitektura *wrapper*-en teknologiaz (Roth eta Schwartz, 1997) baliatzen da, informazio-iturri bakoitzaren gainean *wrapper* bat erantsiz. Teknologia honek abstrakzio bat egiteko aukera emango digu, eta, horrela, informazio-iturriak zenbait erlazioz[2] osaturik daudela suposa-

[1] Hemen "eskema" hitza bere esanahi zabalenean erabiltzen ari gara. Baliabide baten eskemak, horrela, metaturiko datuen semantika deskribatuko du.

[2] Erlazio horiei *erlazio lokalak* deituko diegu.

tuko dugu, euren barne-antolamendu sakonaz eta galdeketa-lengoaiaz arduratu behar izan gabe.

Datu-mailako zein maila semantikoko heterogeneotasunaren ebazpenak, datu-integrazioari dagozkion arazoak izanik, eskema ororen errepresentazioa—iturrienak zein eskema orokorrarena— zein eskemen arteko baliokidetza semantikoa definituko duen ezagutza beharko du, atal honetan zehar ikusiko dugun bezala.

- **Datuen eleaniztasuna**.

  Sistemak bere baitan hartuko duen informazio lexikala eleanitza izango da, hots, hainbat hizkuntzatakoa. Informazio lexikoaren bila dabilenarentzat zuzendutako kontsulta-sistema den neurrian, ELHISAk ezin du hizkuntza jakin batera murriztu. Hizkuntza desberdinetako hitzei buruz kontsultak egitea, edo hitzen itzulpenak lortu ahal izatea —hiztegi elebidunen bidez— lexikografo edo linguisten eguneroko lanak diren neurrian, ELHISAren bidez lan horiek berak egin ahal izatea ezinbesteko baldintzatzat hartu dugu, hasiera-hasieratik.

- **Sistema banatua**.

  ELHISAren portaerak banatua izan behar du, nahitaez, atzitu behar dituen iturri lexikalak ez baitaude fisikoki leku berean. Urruntasun fisikoa dela eta, datu-fluxuak sistema orokorra eta baliabide lokal bakoitza kokatuta dauden tokien artean "bidaiatu" beharko du, sare telematikoetatik zehar. Gaur egun, Internet sarea dela eta, sistema banatuetako hainbat arazo konpondu edo erraztu egin dira. Hor daude, besteak beste, sistema banatuen arteko komunikazioa gauzatu eta errazteko aukera emango dizkiguten CORBA edo SOAP protokoloak. Hala ere, badago sistema banatuek dituzten berezko arazo franko. Konparazio batera, baliabide batek, bat-batean, gure galderei erantzuteari uzten badio, sistemak gai izan behar du baliabidearekin berriro kontaktuan jarri eta komunikazioa jarraitzeko.

- **Informazio-iturrien autonomia**.

  Integrazio-sistema bateko iturrien autonomia-mailak ezarriko du sistema osoaren arkitektura. Hasiera bateko datu-base federatuek, datuak hainbat iturritatik jaso eta era uniformean kudeatzen bazituzten ere, osagai zituzten datu-baseen autonomia ez zen nahikoa handia: datu-base federatuek bazekiten federazio bateko osagaiak zirela. Ikuspegia

aldatu zen, hala ere, datu-integraziorako sistemak garatzerakoan. Horrela, datu-integraziorako sistemek hainbat iturri jartzen dituzte harremanetan, iturri hauetan inongo aldaketarik egin behar izan gabe: iturriek ez dute "jakingo", maiz, integrazio-sistema baten osagaiak direnik ere.

ELHISAn, integratu nahi den informazio lexikala autonomoa da: baliabide bakoitzaren eraikitzaileak erabaki du iturri lokal horren diseinua, eta beren eskuetan dago, besteak beste, datu berriak sortu, ezabatu eta aldatzeko ahalmena. Izan ere, baliabide bakoitzak bere "berezko bizitza" baitu.

Datu-integrazioaren arloan autonomia-maila desberdinak identifikatu ohi dira (ikus, adibidez, Seth eta Larson (1990); Heimbigner eta McLeod (1985)), eta, horietatik, gure baliabideak ia guztuz autonomoak direla ondoriozta dezakegu.

Alde batetik, iturri bakoitzak "erabakiko" du zein informazio *esportatu* ELHISAra: iturria integratuko duen administratzaileak ezarriko du sistematik kanpora joango den informazioa beren *esportazio-eskema* zehazterakoan. Halaber, iturri lokalen esku geratuko da, galdera bat iristean, galdera horri erantzuteko egin beharreko eragiketak zeintzuk diren, eta zein ordenatan exekutatu behar diren. Hauen esku geratuko da, beraz, galdera lokalak noiz eta nola erantzun behar diren erabakitzea.

Azkenik, iturrien diseinua ere bat-batean alda daiteke. Iturri lokal baten diseinu-aldaketarik egiten bada, jakina, lan osagarria burutu beharko da, iturria berriro ELHISAn integratzeko. Horrela, bada, iturriaren datu-eredua edo kontsulta-lengoaia aldatzen bada, iturri horri erantsita dagoen *wrapper*-a aldatzearekin aski da. Iturriko aldaketak sakonagoak badira, hots, iturriaren eskema bera aldatzen bada, diseinatzaileak iturria berriro integratzeko eginbehar guztiak burutu beharko ditu.

- **Dinamikoa**.

  Sistemak dinamikoa izan behar du, hau da, kanpotik sortutako arazoen aurrean moldatu egin behar da. Sistema dinamikoen ezaugarri nagusiak bi lirateke (Seth eta Larson, 1990): iturri lokalek euren eskema bat-batean alda dezakete, eta ez dago iturri baten sorkuntza kontrolatzeko aurreikusitako prozesurik. Ikuspuntu horretatik, ELHISAren arkitek-

turak aurreikusi behar du iturri lokalen dinamikotasuna, eta, horrela, moldatu egin behar da iturrien bat-bateko aldaketen aurrean.

- **Hedagarria.**

  ELHISAk aukera eman behar du, une oro, informazio-iturri berri bat integratzeko. Iturri berria integratzeak, gainera, ez du aurretik zeuden iturrien berrantolaketa ekarri behar, eta, ahal bada, ezta sistemak erabiltzen duen eskema orokorrarena ere. Hala ere, iturri berrien integrazioak eskema orokorra aldatzera behartzen badu, aurreikusirik ez zeuden kontzeptu edo erlazio berriekin osatzeko, adibidez, aldaketek ez dute gainerako iturrien informazioa ezeztatu behar.

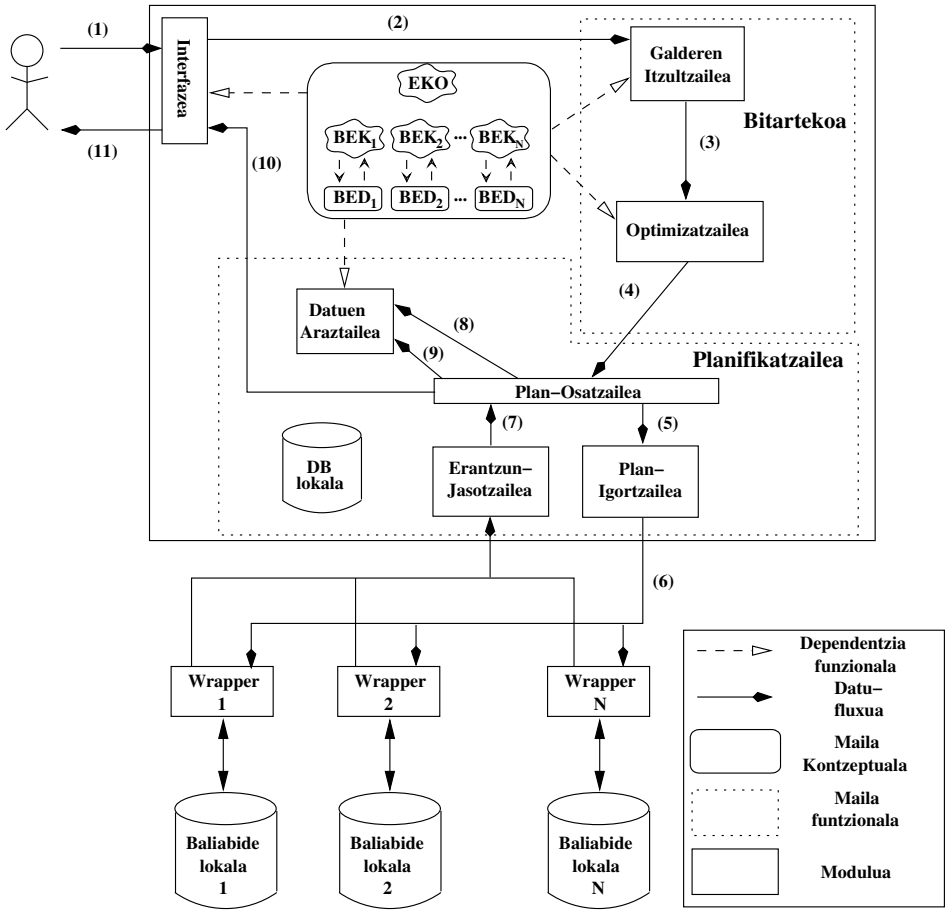
- **Irakurtzeko soilik**.

  ELHISAren bitartez kontsulta daitezkeen datuak bakarrik irakurtzeko dira, hots, ezin da iturri lokaleko datuak aldatu, edo informazioa gehitu. Sistema osoa linguistentzako kontsulta-tresnaren bokazioarekin eraiki denez, datu lokalen aldaketak interes xumea du. Ezaugarri honek, bestalde, arras errazten digu sistema osoaren diseinua. Izan ere, datuak aldatzeko behar diren kontrol-mekanismoak konplexuak baitira oso, batez ere datu-base bakar batean hainbat lagunek modu konkurrentean datuak aldatzeko aukera eskaini behar denean. Konplexutasuna areagotu egiten da, nabarmenki, datu-baseekin izan beharrean informazio-sistema batekin ari garenean.

IV.1 irudian sistemaren arkitektura orokorra ikus daiteke. Nabaria denez, guk proposatutako arkitektura ez da datu-integrazioa helburu duten sistemenetatik asko aldentzen, eta, batik bat, datu-integraziorako beharrezkoak diren eginkizun nagusiak islatzen dira bertan.

Hona hemen, arkitektura horretan oinarriturik, erabiltzailearen eta sistemaren arteko elkarrekintza nola gauzatuko den:

1. Erabiltzaileak, *Interfazea* lagun, galdera bat igorriko dio ELHISAri. Informazio-eskaria egiteko, aurrerago ikusiko dugun eredu kontzeptual orokorrean agertutako kontzeptuak zein erlazioak erabiliko ditu. Interfazea maila kontzeptualeko ezagutzaz baliatzen da eta erabiltzaileari aurkezten dio, bere galderak adierazi ahal ditzan.



IV.1 Irudia: Arkitektura orokorra

2. Interfazeak galdera *Galderen Itzultzaileari* igortzen dio, eta azken honek eredu orokorrean adierazitako galderaren itzulpenari ekiten dio. Berridazketa multzo bat sortzen du emaitza gisa, iturri lokalen arabera soilik adierazita egongo dena. Itzulpena gauzatu ahal izateko, moduluak iturriei zein eredu orokorrari buruzko ezagutza behar du: Eredu Kontzeptual Orokorra (EKO), baliabide bakoitzaren Baliabidearen Eredu Kontzeptuala (BEK) eta Baliabidearen Eduki-Deskribapena (BED).

3. Galderen Itzultzaileak sortutako berridazketa multzotik *Optimizatzaileak* hainbat berridazketa kentzen saiatzen da, baldin bertan eskatutako informazioa gainerako berridazketetatik eskura badaiteke. Bestalde,

erabiltzaileak eginiko informazio-eskaria erantzuteko behar adina informazio jadanik sisteman badagoen erabakitzen du. Bere zereginetan, iturrien BEKari buruzko ezagutza behar du.

4. Optimizatzaileak berridazketa multzoa *Plan-Osatzaileari* pasatzen dio. Honek berridazketa bakoitzean adierazitako informazioa eskuratuko duten planak osatzen ditu, eta beren exekuzioaz arduratzen da.

5. Plan-Osatzaileak *Plan-Igortzaileari* pasatzen dizkio planak, eta honek planak dagozkien iturrietara heltzen direla ziurtatzen du.

6. Iturri orok erantsita dituzten *wrapper*-ek galdera lokal bakoitza exekutatzen dute, eta erantzunak sistemari helarazten dizkote bueltan.

7. Iturrietako erantzunak era asinkrono batean heltzen zaizkio *Erantzun-Jasotzaileari*. Honek, erantzun bat jaso ondoren, Plan-Osatzailearen esku uzten du.

8. Plan-Osatzaileak erantzun-partzialak *Datuen Araztaileari* igortzen dizkio, datuen garbiketari ekin diezaion.

9. Emaitza guztiak jasotzen direnean, Plan-Osatzaileak berriro jotzen du Datuen Araztailera, objektuen identifikazioari ekin diezaion.

10. Plan-Osatzaileak erantzun bateratua Interfazeari igortzen dio.

11. Azkenik, interfazeak erantzun bateratua erabiltzaileari erakusten dio.

IV.1 irudian ikusten diren moduluetatik, Galderen Itzultzailea, maila kontzeptuala deritzona eta zenbait *wrapper* inplementatu ditugu tesi-lan honetan. Horretaz gain, Datuen Araztaile modulua eratzeko diseinua finkatu dugu, eta, baita ere, erabiltzaileak sistemarekin komunikatzeko beharrezkoa den interfazea.

Hurrengo ataletan zehar arkitektura orokorraren osagaietan jarriko dugu arreta. Maila kontzeptualarekin hasiko gara, bertan bilduko baitugu integrazioa gauzatzeko behar den ezagutza guztia. Galderaren prozesatzaileari buruz ere hitz egingo dugu, bereziki Galderen Itzultzailean sakonduz. Gero planifikatzailea azalduko dugu, zeinaren xede nagusia den berridazketak adierazitako informazioa jasoko duen plan bat osatzea, galderak iturri lokal zehatzetara igortzea, eta baliabideek erantzundako datuak jasotzea. Erantzunak

jasotzerakoan, maila estentsionaleko parekatzea burutuko duen erantzunen prozesatzailea ikusiko dugu. Ondoren, ELHISAk gauzatutako integrazioari buruzko zenbait gogoeta plazaratuko ditugu. Kapitulua bukatzeko, integratu ditugun baliabideak aurkeztuko ditugu, eta, azkenik, ELHISA beste integrazio-sistema batzuekin alderatuko dugu.

IV.2 Maila kontzeptuala.

Informazioaren integrazioa nahi bada, baliabide bakoitzak gordetzen duen informazioari buruzko semantika ezagutu behar da. Halaber, eskemen arteko baliokidetasunak ezarri nahi baditugu, eskema horien adierazpen aberatsa erabiltzea komeni da, baliabide bakoitzak bere informazioaren semantika kodetzeko erabilitako berezko formalismoetatik aparte. Informazio hori guztia maila kontzeptualean dago adierazia. Maila kontzeptual deritzogun honetan, informazio-sistema osoan erabiliko den eredu orokorraren adierazpen kontzeptuala zehaztuko da, iturri lokal bakoitzaren datuen adierazpen kontzeptualarekin batera. Maila honetan, halaber, iturri bakoitzaren eredu kontzeptualaren eta eredu kontzeptual orokorraren arteko erlazioak islatzeko beharrezkoa den informazioa gordeko da.

Eskuartean ditugun iturri lexikalak datu-basetzat[3] har ditzakegu, eta, beraz, gorderiko datuek egitura zehatza jarraituko dutela ondorioztatu. Datuen antolamendua adierazteko formalismo baten beharrean gaude, eta guk arras ezaguna den Entitate-Erlazio Hedatuaren (EEH) formalismoa geureganatu dugu, iturri lokalak zein eredu orokorra deskribatzeko.

Bere izenak salatzen duen legez, EEH formalismoa Entitate-Erlazio formalismoaren hedapena dugu, batez ere entitateen arteko ISA hierarkiak adierazteko aukera ematen baitu, hots, entitate bat beste hainbat azpientitate baino orokorragoa dela adieraz baitezakegu. Hierarkiak osatzerakoan, halaber, entitate/azpientitate arteko zenbait erlazio adieraz daitezke: entitate baten umeak diren hainbat azpientitate disjuntuak direla adieraz dezakegu[4], entitate baten umeek gurasoa *estaltzen* dutela zehatz daiteke[5], eta abar.

Hala ere, nahiz eta EEH datu-base erlazionalak modelatzeko formalismo

[3]Hemen, datu-base terminoa bere zentzu zabalean ulertu behar da.

[4]Bi entitate disjuntuak izango dira baldin bi entitateen instantzia den objekturik ezin bada egon.

[5]Entitate baten umeek gurasoa estaliko dute baldin entitatearen instantzia den objektu bat, gutxienez, entitatearen ume baten instantzia ere bada.

egokia izan, semantikoki aberatsagoa den datu-eredua beharko dugu, ezagutzaren errepresentazioko lengoaia baten bidez adierazita. Datu-base federatuen arloan —informazioaren integraziokoaren aitzindaria— aipatzen dira, integratu nahi diren datu-baseen eskemak zein eskema federatua adierazterakoan, semantikoki aberatsak diren formalismoen onurak. Horrela aipatzen da Seth eta Larson-en lanean (1990):

> Database design and integration is a complex process involving not only the structure of the data stored in the database but also the semantics of the data. Thus it is desirable to use a high-level, semantic data model for the Common Data Model.

Izan ere, datu-eredu semantiko *on* batek datu-baseen arteko erlazio eta propietate konplexuak errepresentatzeko aukera eman behar du, eta errepresentazio horrek *zuzena* eta aberatsa izan behar du. Are gehiago datu-integrazioko sistema eraiki nahi bada, non iturrien autonomia-maila oso handia izan ohi den. Horrela esaten da (Levy *et al.*, 1995) lanean:

> Unlike multidatabase systems a data integration system must deal with a large and constantly changing set of data sources. These characteristics raise the need for richer mechanisms for describing our data, and hence the opportunity to apply techniques from Knowledge Representation.

Baliabideak modelizatzeko formalismo semantiko aberatsa erabiliz datuen gaineko adierazpen abstraktua egin ahal izango da, hots, datuak hautemateko dugun era esplizituki adierazteko aukera emanez (Catarci eta Lenzerini, 1993). Hortaz, EEH formalismoak datuen antolamenduari buruzko informazioa eskaintzen badigu ere, informazio horren gainean arrazoiketa bideratzeko aukera emango digun formalismo baten beharrean gaude. Izan ere, ezagutza intentsionalaren gainean arrazonamendu-prozesuak bideratu ahal izatea funtsezko ezaugarria izango da sistemako hainbat eginkizunetan.

Eskemetan islatutako ezagutza —intentsio-mailako ezagutza— adierazteko, ezagutzaren gainean arrazoitzeko aukera ematen duen formalismoa hartu dugu, logikan oinarriturikoa. Horrela, bada, ELHISA deskribapen-logikez[6] baliatzen da baliabide lokalaren zein eskema orokorraren modelizazioa egite-

[6] ikus III.5 atala

ko. Zehatzago, *NeoClassic*[7] deskribapen-logiketan oinarritutako formalismoa erabiliko da datu-eredu bezala.

Ezaguna da deskribapen-logikaren bitartez EEH formalismoa adierazten ahal dela (Calvanese *et al.*, 1998b)[8] eta, horrela, deskribapen-logiketan oinarritutako ezagutza-base bat eraiki dugu baliabide bakoitzaren eskema adierazteko.

Maila kontzeptualean, hiru osagai nagusi aurkituko dira ELHISAn:

- Eredu Kontzeptual Orokorra (EKO)
- Baliabidearen Eredu-Kontzeptual bat iturri bakoitzeko (BEK)
- Baliabidearen Eduki-Deskribapen bat baliabide bakoitzeko (BED)

Ikus ditzagun, banan-banan, osagai horiek.

IV.2.1 Eredu Kontzeptual Orokorra.

Maila kontzeptualeko zati garrantzitsuenetariko bat Eredu Kontzeptual Orokorra (EKO) deritzoguna da. EKOak ezagutza lexikalari buruzko kontzeptualizazioa formalizatzen du, eta bere asmo nagusia ezagutza lexikalaren eredu orokorra eskaintzea da. Tesi-lan honen ideia nagusietariko bat ezagutza lexikal orokorra deskribatzen duen eredu bat eraiki daitekeela izanik, EKOan adierazitako kontzeptu zein erlazioak eredu horren adibide ditugu.

[7] *NeoClassic* sistema *CLASSIC* sistema ospetsuaren inplementazioa da, C++ lengoaian idatzia. Ikus III.5.2 atala

[8] Lan horretan, deskribapen-logikaren familiakoa den $\mathcal{ALCQI}$ lengoaia erabiltzen da EEHtik oso hurbil dagoen formalismoa adierazteko. $\mathcal{ALCQI}$ lengoaiaren adierazpen-ahalmena *NeoClassic*-ena baino aberatsagoa da, eta azken honek onartzen ez dituen zenbait eragile ditu, adibidez, ukapen-eragilea. Artikuluan, baina, ukapen-eragilea hierarkia batean kokatutako kontzeptuen arteko disjuntzioa islatzeko erabiltzen dute. Horrela, A kontzeptuak bi ume disjuntu baditu, B eta C, asertzio hauek beharko lirateke:

$$\begin{aligned} A &\dot{\succeq} B \cup C \\ B &\dot{\succeq} A \cap \neg C \\ C &\dot{\succeq} A \end{aligned}$$

Zorionez, *NeoClassic*ek kontzeptuen arteko disjuntzioa onartzen du, hauen arteko ukapenik erabili gabe. $\mathcal{ALCQI}$-en eta *NeoClassic*-en artean alde gehiago dauden arren, aldeak nahikoa xumeak dira, eta, hortaz, (Calvanese *et al.*, 1998b) lanaren konklusioak geureganatu ditugu.

Integrazioaren ikuspuntutik, EKOa aplikaziorako interesgarriak diren kontzeptu eta erlazio orokorren deskribapen kontzeptuala dugu. Horrela, bada, EKOa eskema-integrazio tradizionaleko eskema kontzeptual integratuarekin bat etorriko litzateke, integrazio-sistemaren erabiltzailearentzat interesgarriak diren kontzeptu zein erlazioen bista bateratua eskainiko baitu.

Bestalde, EKOa bat dator datu-integrazioko *bitarteko eskema*ren kontzeptuarekin ("mediated schema")[9]. Bitarteko eskema, erabiltzaileak galderak egiteko helburuarekin eraikitako eskema logikoa da; hots, erabiltzaileak, integrazio-sistemari galdera bat igortzerakoan, bitarteko eskemaren kontzeptuak eta erlazioak soilik erabiltzen ahal ditu. Horrela, bada, EKOak sistemaren *terminologia* deskribatuko du, berak ezarriko baitu zeri buruz galde daitekeen, eta zeri buruz ez.

EKOak, bertan deskribatutako entitate eta erlazioen bidez, ezaugarri lexikal nagusiei buruzko galderak egiteko aukera emango digu. Nolanahi ere den, galdera horiek erantzuteko beste alde batera jo beharko da. Izan ere, EKOa adierazpen birtuala dugu, deskribatzen diren kontzeptu eta erlazioak ez baitira "errealak". EKOan erlazioek ez dute hedapenik: erlazioen n-koteak ez daude fisikoki inon gorderik. Are gehiago, seguruenik, EKOaren erlazioekin zuzenean bat datorren erlaziorik ez dugu aurkituko iturri lokalen eskemen artean. Beraz, EKOa ELHISAk integratuko duen informazioaren adierazpen gisa ikusi behar da, benetako eskema tradizional bezala baino.

EKOaren osaera, hein handi batean, tesi-lan honen funtsezko ekarpena da. EKOaren helburua handinahikotzat jo daiteke, eta nork edo nork esango du, arrazoi osoz, baliabide lexikal heterogeneoetan aurki daitekeen informazio lexikal guztia —bakoitza bere adierazpidearen arabera— eredu komun batera bihurtzea ezinezkoa dela. Hala ere, esan behar dugu gure helburuak xumeagoak direla. Izan ere, guk ez baitugu nahi informazio lexikal *guztia* ELHISAn integratzea. Hori baino, gure interesa eta ikuspuntua informazio lexikal aberatsa behar duen balizko erabiltzailean dude; erabiltzaile horri baliabide lexikal anitzetatik jasotako informazio lexikala eskaintzea baita gure helburu nagusia, informazio hori guztia elkar-erlazionatu eta erkatzeko aukera emanez. Erabiltzailearen interesa, apika, ez dago informazio lexikal guztia eskuratzean, baizik eta interfaze komun batetik ahalik eta informazio gehien eskuratu ahal izatean. Iturri lexikal anitz kontsultatzea linguista zein

[9]Bitarteko eskemari izen desberdinak eman ohi zaizkio. Horrela, erreferentzia-eskema (*InfoMaster* sisteman, Duschka eta Genesereth (1997b); Genesereth *et al.* (1997)), munduaren eredua (*Occam* sisteman, Kwok eta Weld (1996)) edota domeinu-eredua (*SIMS* sisteman, Arens *et al.* (1996)) izenekin agertu ohi da literaturan.

lexikografoen eguneroko lana izanik, gure sistemaren erabiltzaileak informazio aberatsaz hornitutako baliabide lexikal ezagunetatik jaso ahal izango du informazioa, era uniforme batean, baliabide bakoitzaren sakoneko barne-antolamendu eta kontsulta-lengoaia zehatzak ezagutu behar izan gabe.

EKOak, hortaz, bi ezaugarri nagusi izan behar ditu. Batetik, baliabide anitzen informazio lexikala errepresentatzeko aukera eman behar du. Bestetik, sistemaren erabiltzaileari adierazpide intuitibo eta aberatsa eskaini behar dio, erabiltzaileak bere galderak modu errazean egin ahal ditzan. Ezaugarri hauek kontuan hartuz, bi hurbilpen nagusi jarraitu dira, EKOan dauden kontzeptu zein erlazioak zehazterakoan:

- **Behetik gora:** IXA taldeak hainbat baliabide lexikalekin lan egin ohi du. Tesi-lan honen hasierako hausnarketa baliabide horiek guztiak integratuko lituzkeen sistema baten eraikuntzaren bideragarritasuna aztertzea izanik, baliabide lokal bakoitzaren entitate zein erlazioen adierazpideak —eskemak, alegia— abiapuntu paregabea eskaini digu hastapeneko eredu kontzeptual orokorraren nondik norakoak zehazterakoan.

- **Goitik behera:** ELHISAren erabilgarritasuna bermatzearren, IXA taldeak baliabide lexikaletan duen eskarmentuaz baliatu gara, EKOan agertu behar duten kontzeptuak zein erlazioak zehazterakoan, eta horien gainean egin daitezkeen galdera motak zeintzuk diren finkatzerakoan. Horrela, bada, baliabide lexikalak atzitzeko bideak aztertu ditugu, baliabide hauek euskarri elektronikoan gordeta daudenean. Izan ere, datu lexikalak konputagailuz metatzeak aurrerapen franko ekartzen baitu datuak eskuratzeko garaian. Kasu paradigmatiko bat hiztegiena da. Paperezko hiztegien arazoa da hitzaren forma ezagutu behar dela beren informazioa eskuratu nahi bada. Izan ere, hitzak ordena alfabetikoan agertzen baitzaizkigu hiztegi klasikoetan. Hitz-forma ezaguna ez bada —ahaztu egin zaigulako, kasu— ezinezkoa izango zaigu hitzarekin topatzea, nahiz eta hitz antzekoak edo, are, sinonimoak jakin. Hiztegia konputagailuz gordeta badago, aitzitik, sarreren ordena alfabetikoa ez ezik, beste hainbat atzibide ere irekiko zaizkio erabiltzaileari informazio lexikala eskuratzerakoan, bilatu nahi den hitzaren forma zehatza jakin behar izan gabe: hitz-forma baten sinonimoak eskuratzea, definizioari buruzko galderak egitea, galdera egiterakoan hitzen arteko erlazio lexiko-semantikoez baliatzea, eta abar. Horrela, bada, IXA taldeko baliabide lexikalen erabiltzaileen beharrak kontuan hartu izan dira, eta,

batik bat, baliabideei igorritako galdera mota usuenak aurreikusi eta aztertu, galdera horiek ELHISAn egiteko aukera eman ahal izateko. Galdera usuen horietatik, hona hemen, adibide gisa, azpimultzo esanguratsu bat:

- Hitz-forma jakin baten kategoria/azpikategoria lexikala.
- Hitz-forma jakin baten adiera guztien definizioa.
- Kategoria/azpikategoria lexikal jakin bat duten hitz-formak.
- Hitz-forma jakin baten bidez adierazitako kontzeptuen erlazio lexiko-semantikoak (hiperonimia, hiponimia, sinonimia, meronimia, etab.).
- Hitz-forma jakin baten aldaera / forma ez-estandarrak / errore tipikoak.
- Hitz baten erabilera-adibideak
- Definizioak edo/eta adibideak testu-corpus gisa hartuz egin daitezkeen galderak (hiztegietan batik bat)
- Aurreko guztien konbinaketak.
- ...

Ondoren, EKOaren osaerari buruz arituko gara, eta ereduari berari gainbegiratu azkarra botako diogu. EKOko kontzeptuen zein erlazioen zerrenda osoa eranskinetan aurki daiteke[10].

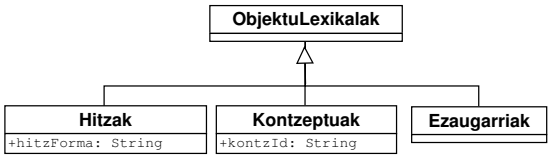
IV.2.1.1 EKOaren goi-mailako sailkapena.

EKOa orokorra izatea bada helburua, eredu orokor sinplea izan behar du nahitaez. Baina, orobat, bildu behar ditu bere baitan ezagutza lexikaleko baliabide edo iturri batek zehatz ditzakeen ezaugarri eta erlazio oro. Edo, hobeto esan, gai izan behar du ezagutza lexikalaren adierazpide diren ezaugarri eta erlazio horiek deskribatzeko.

Oro har, ezagutza lexikala errepresentatzeko bi elementu mota erabiltzen ditugu EKOan:

- Klase-hierarkia (ikus IV.2 irudia), non era guztietako "objektu lexikalak" baitaude `ObjektuLexikalak` klase nagusian bildurik, eta hiru azpiklase nagusitan banaturik: `Hitzak, Kontzeptuak` eta `Ezaugarriak`.

[10] A eranskina.



IV.2 Irudia: EKO. Goi-mailako sailkapena.

- Erlazioak: hierarkiako klaseen artean ezarritako erlazio bitarren multzoa.

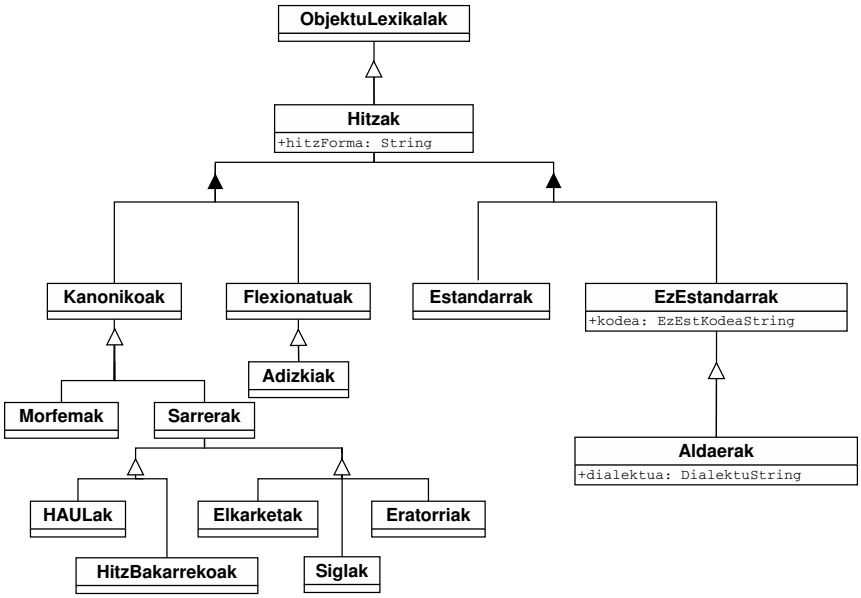
Arestian esan dugun bezala, iturri lexikalak errepresentatzeko —eta baita EKOa ere—, EEH formalizazioa erabiliko dugu. EEHko klase-hierarkiak adierazteko, berriz, UML modelatze-lengoaia orokorraz baliatuko gara. Horrela, klaseak eta atributuak azalduko ditugu, grafikoki, eta klaseen arteko erlazio bitarrak taulen bitartez zehaztuko. Klaseen atributuen heinak oinarrizko mota izan behar du, hala nola, `String` edo `Bool`, bestela klase arteko erlaziotzat hartuko baita. Horretaz gain, guk propio definitutako domeinu abstraktuak ere agertuko dira, esaterako, `HizkString` mota karaktere-kate bat da, baina soilik balio jakin batzuk har ditzake (kasu honetan, hizkuntza desberdinak adierazteko ISO kodeak). Bestalde, eredu kontzeptual oro *NeoClassic* DLen bidez definitu izan dira, A eranskinean ikus daitekeen bezala.

IV.2.1.2 EKOko objektu lexikalak: hitzak.

Lexikoaz diharduganean, hitzei buruz ari gara ezinbestean. Hitzak dira eredu lexikal baten muina, eta, gehienetan, baita atzibide nagusia ere. Hitz kontzeptuaren definizio bat ematea ez da erraza, eta aitortu behar dugu gure EKOan hiztzat hartzen duguna oso kontzeptu zabala dela (ikus IV.3 irudia): "hitz-hitzetatik" hasi (hiztegietako "sarrerak" izan ohi direnak) eta morfema ez-independenteetaraino hedatzen da gure `Hitzak` klase honen azpian sailkatuko duguna, eratorriak, elkarketak, forma flexionatuak eta hitz anitzeko unitate lexikaltzat har daitezkeenak barne, estandarrak nahiz ez-estandarrak.

IV.2.1.3 EKOko objektu lexikalak: kontzeptuak eta adierak.

Hitzen esanahia nolabait deskribatzea helburu duen baliabide lexikal orok adiera, esanahi, hitz-zentzu, edo antzeko termino bat erabili beharko du, ezinbestean, hitzen esangura, ñabardura, etab. desberdinak bereziko baditu.



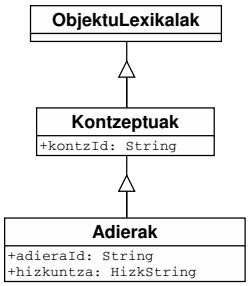
IV.3 Irudia: EKO. Hitzak eta beren sailkapena.

EKOan, hitzen esangurak bi klasetan bildu ditugu, hots, `Kontzeptuak` eta `Adierak`.

`Kontzeptuak` klasearen bitartez adieraziko da hitzaren zentzua zein den, hau da, mundu errealeko zeren adierazpena den. `Kontzeptuak` klaseko kideak lexikoietan aurki daitezkeen unitate semantikoekin bat datoz[11]. Horrela, bada, `Kontzeptuak` klasea informazio semantikoaren adierazpenaren gakoa da, eta kontzeptuei lotuko zaie ezaugarri semantiko oro. `Kontzeptuak` klasea da, halaber, erlazio lexiko-semantikoen zein hizkuntzen arteko baliokidetasunen domeinu eta heina.

Bestalde, `Adierak` klasea dugu, zeinaren bidez errepresentatuko diren hiztegietako adierak zein homografoak, edo zentzu desberdinak bereizteko erabilitako edozein entitate. Hiztegietako adierak edo homografoak hizkuntza jakin batekoak direnez, klase honek hizkuntza-identifikadorea izango du. Horretaz gain, adiera-identifikadore bat izango du beti, hitz horrek dituzkeen beste adieretatik bereizi ahal izateko. `Adierak` kontzeptua `Kontzeptuak` kontzeptuaren espezializaziotzat hartzen dugu, kontzeptu baten hizkuntza eta

[11] Adibidez, GENELEX edo SIMPLE proiektuek proposatzen duten `USem` unitate semantikoarekin.



IV.4 Irudia: EKO. Kontzeptuak eta adierak.

sistema lexikal jakin batean egindako espezializazioa, alegia.[12]. Baliabide lexikal batzuek, hitzak eta kontzeptuak ez ezik, adierok ere aintzat hartzen dituzte, eta elementu lexikalekiko beren erlazioak zehazten dituzte beren eskemetan. Horregatik hartu ditugu aintzat guk ere, ezagutza lexikala deskribatzeko eredu kontzeptual orokorra izan nahi duen honetan (ikus IV.4 irudia).

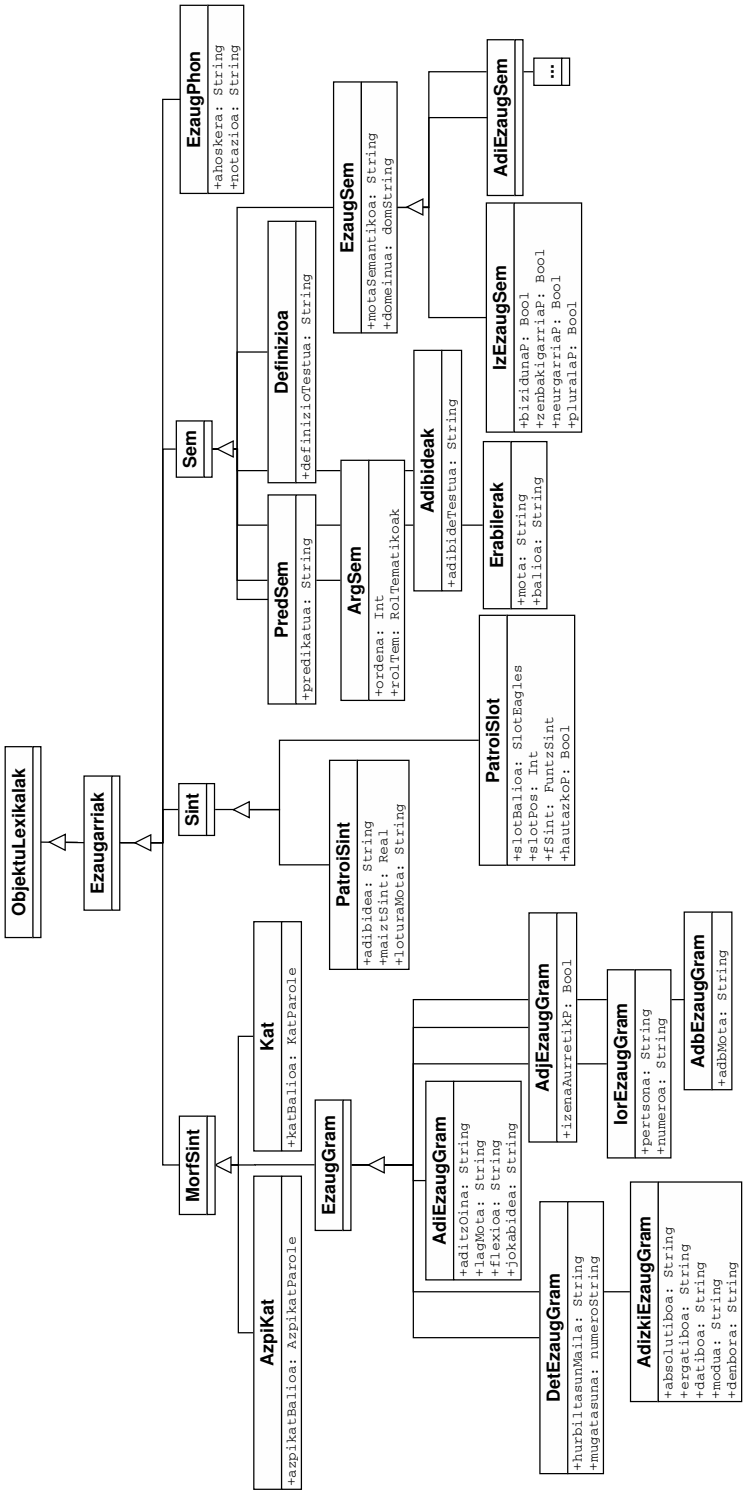
IV.2.1.4 EKOko objektu lexikalak: ezaugarriak.

Azkenik, ezaugarriak ditugu. Mundu lexikalaren deskribapen bat egin nahi bada, era guztietako ezaugarri eta atributuak agertuko zaizkigu: fonetikoak, gramatikalak, erabilerari dagozkionak, semantikoak, etab. Ezaugarriok entitatetzat hartzen ditugu EKOan, eta, ondoren ikusiko diren erlazioei esker, harremanetan jartzen ditugu klase-hierarkiako beste klaseetako elementuekin, lexikoaren deskribapena horrela osatuz (ikus IV.5 irudia).

IV.2.1.5 EKOko erlazioak.

Erlazioei dagokienez, hierarkiako klaseak lotzen dituzten hainbat eta hainbat erlazio bitar zehazten ditu EKOak. Erlazio horiek definitzeko, hiru elementu adierazi behar dira:

[12]Adieren eta kontzeptuen arteko bereizkuntza ez da, batzuetan, oso argia, baliabide askok ez baitituzte desberdintzat hartzen. Esan dezagun, azaltze aldera, kontzeptu bat errepresentatzen ahal dela, hizkuntza baten baitan, hitz-adiera baten edo gehiagoren bitartez: hainbat hitz-adiera sinonimok kontzeptu bakar bat errepresentatuko dute. Beste hizkuntza batean erreparatzen badugu, kontzeptu hori bera beste adiera multzo batek errepresentatzen du, lexikalizaturik badago, noski. Kontzeptuak esplizituki adierazten dituzten baliabide lexikaletan ohikoa da, kontzeptuok errepresentatzeko, zenbakizko gako batzuk erabiltzea, hizkuntzako elementuetatik (hitzetatik) bereiztearren.



IV.5 Irudia: EKO. Ezaugarriak eta beren sailkapena.

- Domeinua: erlazioa zein klasetako elementuei aplikatzen zaien, alegia.

- Heina: erlazioaren helburua, hau da, domeinuko elementuak zein klasetakoekin lotzen diren.

- Kardinalitate maximoa: EKOan, eredu abstraktua den neurrian, erlazioak —klaseak bezalaxe, bestalde— birtualak dira, eta ez dute, berez, inolako informaziorik "gordetzen" beren baitan. Horregatik, erlazioon kardinalitateaz ari garelarik, euren kardinalitate maximoa baino ez daukagu zehazterik.

IV.1 taulan daude definitu ditugun erlazio esanguratsuenak, sailka.

Esan bezala, EKOa *NeoClassic*-ez inplementatu dugu. IV.6 irudian EKOaren zati bat ikus daiteke, eta irudia hona ekarri dugu irakurleak bere itxuraren antza har dezan. Bertan, hitz-formei buruzko sailkapena agertzen da. Jo beza irakurleak A eranskinara EKOa osorik ikusteko.

IV.2.2 Baliabidearen Eredu Kontzeptuala (BEK).

Baliabide lokal bakoitza berezko eskemaren arabera antolatua egongo denez, eskema horien guztien deskribapen-ahalmena gainditzen duen formalismoa behar da. Formalismo horrek ahaltsua behar du izan, hots, datuek adierazten dituzten entitateen zein datuen arteko erlazio posibleen deskribapen aberatsa egiteko aukera eman behar du.

Atal honetan ELHISAn integratu diren hainbat baliabideren eredu kontzeptualak (BEKak) espezifikatuko ditugu, lehen EKOa zehazteko egin dugun molde berbera erabiliz, objektu lexikalen klase-hierarkia eta euren arteko erlazioak zehaztuz, alegia[13]. Berriro ere, EEH eredu pean modelizatutako iturriak adierazteko UML diagramaz baliatuko gara, eta, baita ere, domeinu abstraktuak erabiliko ditugu zenbait klaseren atributuen heina zehazteko. Horrela, bada, `EHKatString` motak, konparazio batera, kategoria lexikalak gordeko ditu, *Euskal Hiztegian* azaltzen diren bezala. IV.6 atalean ikusiko dugunez, paper arras garrantzitsua beteko dute iturrietako domeinu abstraktuek, iturri horietatik jasotako datuak "garbitu" eta sisteman integratu behar

[13]Litekeena da, zenbait baliabide modelizatzerakoan, jatorrizko entitate, ezaugarri edo/eta erlazio batzuk aintzat ez hartzea, ez zaizkigulako interesgarriak iruditu, edota horiek hartu ez izanak ez duelako ezertan aldatzen ereduaren baliagarritasuna. Aurrerago, IV.7 atalean, kontu hauei buruz arituko gara sakonago.

Erlazioa	Domeinua	Heina	Kard. max.
Adiera eta hitzen artekoak			
adierak	Hitzak	Adierak	n
adierazia	Siglak	HAULak	n
atzizkiak	Eratorriak	Morfemak	n
formaFlexionatuak	Adierak	Flexionatuak	n
estandarra	EzEstandarrak	Adierak	1
hobestenDa	Estandarrak	Adierak	n
lemaDagozkio	Flexionatuak	Adierak	n
lemaDa	Kanonikoak	Adierak	n
oinarria	Eratorriak	Adierak	1
...			
Kontzeptuen artekoak: erlazio lexiko-semantikoak			
hiperonimoak	Kontzeptuak	Kontzeptuak	n
hiponimoak	Kontzeptuak	Kontzeptuak	n
sinonimoak	Kontzeptuak	Kontzeptuak	n
zatiaDa	Kontzeptuak	Kontzeptuak	n
ekintza	Kontzeptuak	Kontzeptuak	n
...			
Maila semantikoa			
argSem	PredSem	ArgSem	n
...			
Maila sintaktikoa			
patroiEzaug	PatroiSint	PatroiSlot	1
argSint	PatroiSint	PatroiSlot	n
...			
Maila sintaktiko-semantikoa			
sintDagokio	PredSem	PatroiSint	n
Adiera eta kontzeptuen ezaugarriak			
definizioa	Kontzeptuak	Definizioa	1
adibideak	Kontzeptuak	Adibideak	n
kat	Adierak	Kat	1
azpikat	Adierak	AzpiKat	n
ezaugPhon	Adierak	EzaugPhon	n
ezaugGram	Kontzeptuak	EzaugGram	n
gauzatzeSint	Kontzeptuak	PatroiSint	n
ezaugSem	Kontzeptuak	EzaugSem	n
gauzatzeSem	Kontzeptuak	PredSem	n
...			
Hizkuntzen artekoak			
baliokideak	Kontzeptuak	Kontzeptuak	n
...			

IV.1 Taula: Klaseen arteko erlazio batzuk, EKOan zehazturik

```
;; Hitzak eta hitz-formak

(createDisjointGroup kanonikotasuna)
(createDisjointGroup estandartasuna)
(createConcept HitzakRef
               ObjektuLexikalak
               topDg)

(createRole adierak false)
(createRole hitzForma true)
(createRole hitzErlazionatuak false)
(createConcept Hitzak
         (and HitzakRef
              (all adierak Kontzeptuak)
              (all hitzForma String)
              (all hitzErlazionatuak HitzakRef)
              )
         true)

(createRole hobestenDa false)
(createConcept Estandarrak
         (and Hitzak
              (all hobestenDa Adierak)
              )
         estandartasuna)

(createRole ezestandarKodea true)
(createRole estandarra false)
(createConcept EzEstandarrak
         (and Hitzak
              (all ezestandarKodea String)
              (all estandarra Adierak)
              )
         estandartasuna)

(createRole dialektua true)
(createConcept Aldaerak
         (and EzEstandarrak
              (all dialektua String)
              )
         true)
...
```

IV.6 Irudia: EKOaren zati bat, hitz-formei buruzkoa, NeoClassic-ez

direnean. Izan ere, domeinu abstraktuek zenbait metodo erantsita izango baitute, domeinu horietako balioak ELHISAn integratzeko behar-beharrezkoak izango direnak. `EHKatString` domeinua, esate baterako, *Euskal Hiztegiko* kategoria lexikalak ELHISAk ulertzen duen formatura bihurtuko dituen metodo batez egongo da horniturik.

Baliabideon eredu kontzeptualak oso desberdinak izango dira, heterogeneoak baitira deskribatu nahi diren baliabideak berak. Iturri lexikalen sakoneko egitura eta antolamendu fisikoa, beraz, askotarikoa izan daiteke, baina *wrapper*-etan oinarritutako teknologiaz (Roth eta Schwartz, 1997) baliatuko garenez, abstrakzio bat egin, eta informazio-iturria objektu lexikalen bildumatzat hartuko dugu, arestian esan dugun bezala, klase-hierarkia baten eta erlazio bitarren multzo baten bidez zehaztuz. Kasu honetan, erlazioen kardinalitate maximoa baino, entitateek —domeinukoak zein heinekoak— erlazioan zer parte-hartze duten zehaztu ahal izango dugu, oraingoan erlazio *errealak* (eta ez *birtualak*) direlako, baliabidean fisikoki gauzatutakoak, alegia.

Ondoren, beraz, zenbait baliabideren eredu kontzeptualak deskribatzeari ekingo diogu (deskribapen osoak aztertu nahi izanez gero, jo B eranskinera).

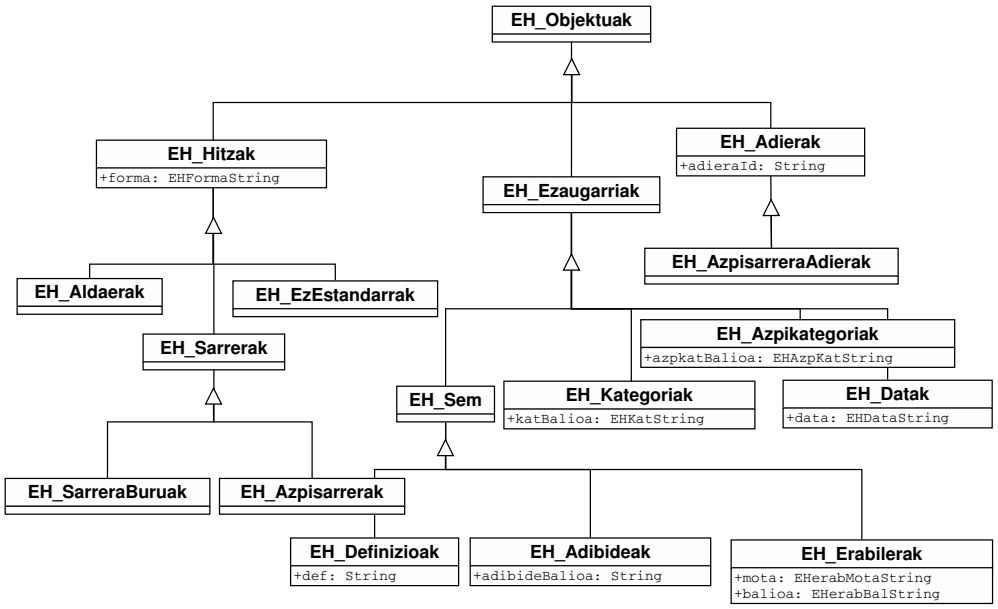
IV.2.2.1 TEIren arabera kodeturiko hiztegi elebakar konplexu bat: Euskal Hiztegia.

Euskal Hiztegia (EH) (Sarasola, 1996) hiztegi elebakar konplexu bat da, eta TEIren arabera (Sperberg-McQueen eta Burnard, 1995) SGMLz kodetua dugu. Giza-erabiltzaileentzako hiztegien adibide bezala integratu dugu ELHISAn, EKOa, era horretako baliabide lexikalak integratzeko orduan, egokia denetz frogatzearen.

Honako objektu-klase hauek aurkitzen ditugu EHren klase-hierarkian (ikus IV.7 irudia):

- Hitzak: hiztegiko sarrerak —sarrera-burukoak zein azpisarrerak—, aldaerak eta forma ez-estandarrak.
- Adierak eta azpisarrera-adierak[14].
- Ezaugarriak: kategoriak eta azpikategoriak, definizioak, adibideak, datak eta erabilera-oharrak.

[14]Azpisarrera-adierak adiera arruntak dira, baina adierek ez duten erlazio batez horniturik daude: azpisarrera sarrera nagusiko adierarekin lotzen duen *sarreraGurasoa* erlazioa.



IV.7 Irudia: EH hiztegiaren BEKa.

IV.2 taulan ikus daitezke EHrako definitu ditugun erlazio esanguratsuenak, sailka.

IV.2.2.2 Datu-base lexikalak: EDBL.

Euskararen Datu-Base Lexikala (EDBL) (Aldezabal *et al.*, 2001) hizkuntzaren tratamendurako IXA taldean darabilgun datu-base lexikala da, eta DBMS erlazional batean dago gorderik. LNPrako datu-base lexikal handi baten adibide gisa integratu dugu ELHISAn, EKOa, era horretako baliabide lexikalak integratzeko orduan, egokia denetz frogatzearren.

EDBLn, honako objektu-klase hauek aurkitzen ditugu, hiru espezializazio nagusiren arabera sailkatuak. Klase horiek, orobat, azpiklaseetan sailkatzen dira, eskema kontzeptual konplexu samarra osatuz. Hori dela eta, EDBLren eskema hainbat iruditan aurkeztuko dugu:

- EDBLren goi-mailako hierarkia (IV.8 irudia).
- Hiztegi-sarrerak eta bestelakoak (IV.9 irudia).
- Unitate estandarrak eta ez-estandarrak (IV.10 irudia).

Erlazioa	Domeinua	Heina	Parte-hartzea (dom : heina)
Sarrera-hitz, aldaera eta adieren artekoak			
adierak	EH_Sarrerak	EH_Adierak	(1,n) : (1,1)
formaIdatzia	EH_Adierak	EH_Sarrerak	(1,1) : (1,n)
lemaKanonikoa	EH_Aldaerak	EH_Sarrerak	(1,1) : (0,n)
sinonimoak	EH_Adierak	EH_Adierak	(0,n) : (0,n)
antonimoak	EH_Adierak	EH_Adierak	(0,n) : (0,n)
...			
Adieren ezaugarriak			
kategoria	EH_Adierak	EH_Kategoriak	(1,n) : (1,n)
azpiKategoria	EH_Adierak	EH_Azpikategoriak	(0,n) : (1,n)
definizioa	EH_Adierak	EH_Definizioak	(1,1) : (1,1)
adibideak	EH_Adierak	EH_Adibideak	(0,n) : (1,1)
erabilerak	EH_Adierak	EH_Erabilerak	(0,n) : (1,n)
noizAurkitua	EH_Adierak	EH_Datak	(0,n) : (1,n)
...			

IV.2 Taula: Objektu lexikalen arteko erlazioak, EHren BEKean.

- Hitz bakarreko unitateak (Zuriunerik Gabeko Sarrerak, ZGS) eta Hitz Anitzeko Unitate Lexikalak (HAUL) (IV.11 irudia).

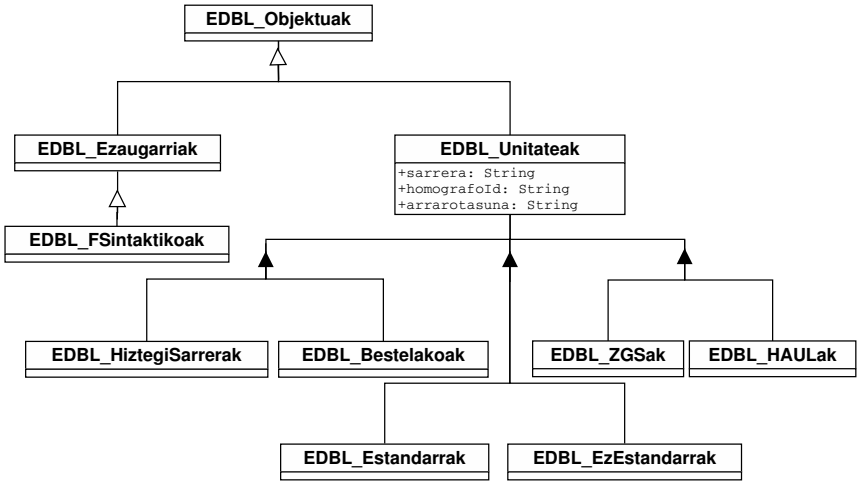
IV.3 taulan ikus daitezke EDBLrako definitu ditugun erlazio esanguratsuetako batzuk, sailka.

IV.2.2.3 Datu-base lexikalak: EDR erraldoia.

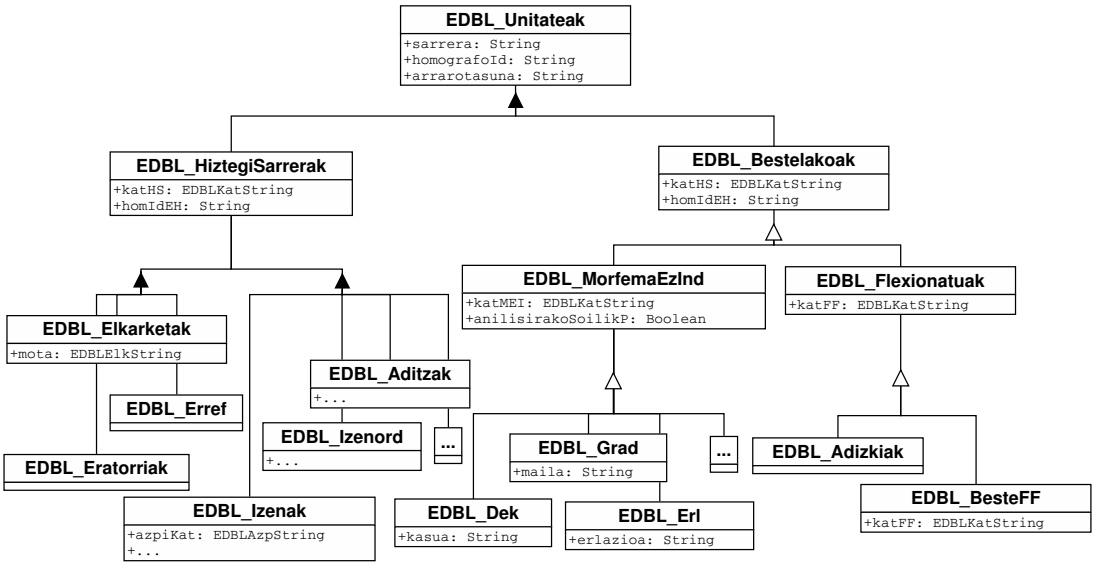
Japonian garaturiko *Electronic Dictionary Research* (EDR) delako datu-base lexikal handia kontzeptua oinarri duen datu-base eleanitza da, ordenagailu bidez —eta, batik bat, itzulpen automatikoan— erabiliko den informazioa biltzen duena (Yokoi, 1995). Hainbat hiztegi edo datu-base hartzen ditu bere baitan: hitz-biltegiak, kontzeptu-biltegiak, agerkidetza-hiztegiak eta hiztegi elebidunak (ingelesa eta japoniera)[15]. Esan behar da, bestalde, datu-baseetan bildutako unitate lexikalen kopurua izugarri handia dela, eta hortik atal honen izenburuan EDRri eman diogun kalifikatzailea.

Esan bezala, EDRren arkitektura kontzeptuen inguruan antolatua da: hizkuntzatik independenteak diren kontzeptuak jaso eta deskribatzen dira

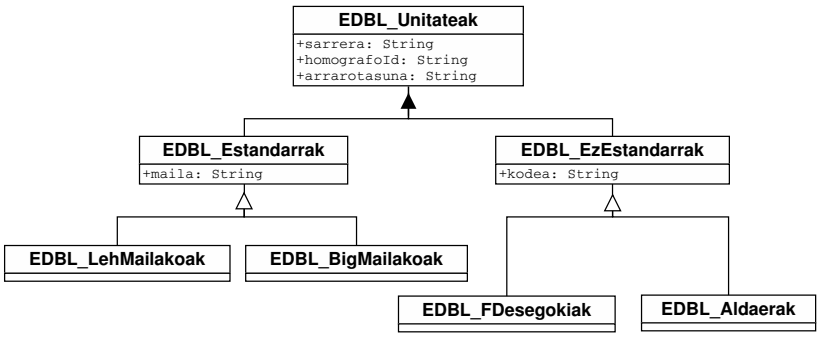
[15]EDRren eskema kontzeptuala modelizatzerakoan, kontzeptu-biltegiak eta hiztegi elebidunak baino ez ditugu kontuan hartu.



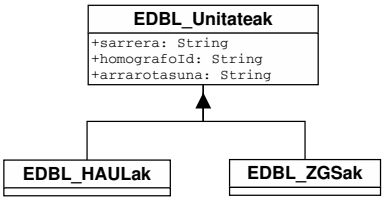
IV.8 Irudia: *EDBL* datu-basearen BEKa: hierarkia orokorra.



IV.9 Irudia: *EDBL* datu-basearen BEKa: hiztegi sarrerak eta bestelako sarrerak (laburtua).



IV.10 Irudia: *EDBL* datu-basearen BEKa: unitate estandar eta ez-estandarrak.



IV.11 Irudia: *EDBL* datu-basearen BEKa: Hitz Anitzeko Unitate Lexikalak (HAUL) eta Zuriunerik Gabeko Sarrerak (ZGS).

Erlazioa	Domeinua	Heina	Parte-hartzea (dom : heina)
Estandartasunaren ingurukoak			
estandarDagokio	EDBL_EzEstandarrak	EDBL_Estandarrak	(1,n) : (0,n)
hobestenDa	EDBL_BigMailakoak	EDBL_LehMailakoak	(1,n) : (0,n)
Lema kanonikoa			
lemaNon	EDBL_Unitateak	EDBL_HiztegiSarrerak	(0,1) : (0,n)
Erreferentziazkoen adierazia			
adieraziaDu	EDBL_Erref	EDBL_HiztegiSarrerak	(1,1) : (0,n)
Hitz-elkarketak eta eratorriak			
lehenOsagaia	EDBL_Elkarketak	EDBL_ZGSak	(1,1) : (0,n)
bigarrenOsagaia	EDBL_Elkarketak	EDBL_ZGSak	(1,1) : (0,n)
oinarriDu	EDBL_Eratorriak	EDBL_ZGSak	(1,1) : (0,n)
atzizkiDu	EDBL_Eratorriak	EDBL_AtzizkiLex	(0,2) : (0,n)
aurrizkiDu	EDBL_Eratorriak	EDBL_AurrizkiLex	(0,1) : (0,n)
...			

IV.3 Taula: Objektu lexikalen arteko erlazio batzuk, *EDBL*ren BEKean.

kontzeptu-biltegian, eta hizkuntza desberdinetako sarrera lexikalekin lotzen dira. Bestalde, ingelesa-japoniera eta japoniera-ingelesa hiztegi elebidunak ere sistemaren parte dira.

Unitate lexikalei buruzko informazio orokorraz gain (sarrera-burua, flexionatuetan aldaezina den zatia, ahoskera, silaba-egitura, erabilera, maiztasuna, etab.), informazio gramatikala ematen da atributu-balio zerrenda baten bidez (kategoria, zuhaitz sintaktikoa HAULen kasuan, aditz-jokoa, aspektua, funtzio sintaktikoa, etab.), eta, informazio semantikotzat, kontzeptuekiko estekak zehazten dira batik bat, kontzeptu-identifikadoreen bitartez.

Kontzeptuei buruzko informazioa nahiko aberatsa da: definiziorik ematen ez den arren, deskribapen edo azalpen laburrak ematen dira, ingelesez zein japonieraz. Kontzeptuak beren arteko erlazioek deskribatzen dituzte: batetik, sailkapen-erlazioak (hiperonimo eta hiponimoak), zeintzuek kontzeptu arteko hierarkia eratzen baitute; eta, bestetik, kontzeptu-deskribapena izeneko egitura batean zehaztutako erlazio lexiko-semantikoak[16].

Laburbilduz, honako entitate hauek aurkitzen ditugu EDRren arkitekturan (ikus IV.12 irudia):

- Hizkuntza bakoitzeko sarrera lexikalak (hitz-formak): japonierazkoak zein ingelesezkoak.

- Kontzeptuak.

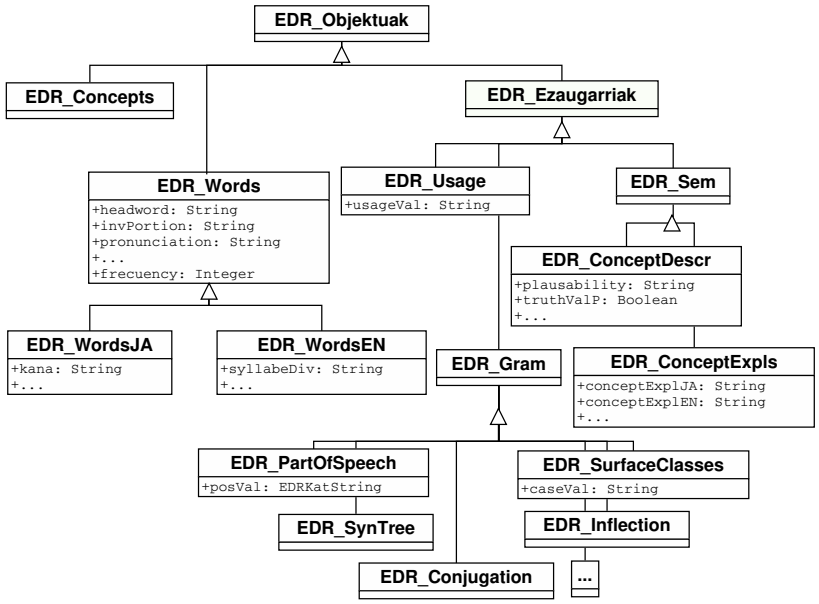
- Ezaugarriak: gramatikalak eta semantikoak, batik bat.

BEKeko elementuak lotuz, berriz, honako erlazio mota hauek bereiz ditzakegu (ikus IV.4 taula):

- Hitz eta kontzeptuen arteko erlazioak, sarrera lexikalen semantika deskribatzen dutenak. Atal horretan sartu ditugu, halaber, kontzeptuak beren ordezkari diren hitzekin (*headconcepts*) lotzen dituztenak, eta hitz baten esanahia azaltzeko emandako testuzko azalpenak (*concept explications*).

- Hiztegi elebidunetako hitz arteko baliokidetzak adierazten dituztenak.

[16]Sérasset-en txostenak (1993) dioenez, 32 erlazio eta 5 atributu —erlazio bakunak— erabiltzen dira EDRn; (Miike *et al.*, 1990) txostenaren arabera, bestalde, 18 erlazio nagusi erabiltzen omen dira kontzeptuak deskribatzeko.



IV.12 Irudia: EDR datu-base lexikalaren BEKa.

- Kontzeptu-kontzeptu erlazioak, non, hitzen kasuan bezalako kontzeptu-azalpenez gain, kontzeptu-sailkapena zehazten dutenak (hiperonimo eta hiponimoak) eta kontzeptu-sarea eratzen dutenak sartu baititugu (*concept description* izeneko egitura batzuen bidez adieraziak).

IV.2.2.4 Ezagutza-baseak: *Hiztsua*.

Hiztsua giza-erabiltzailearentzako hiztegi-laguntzako sistema adimentsu baten prototipo bat da (Artola, 1993), hiztegi elebakar erreal batetik sortua. Lan horretan, prototipoa bada ere, hiztegiko definizioen adierazpide sakona proposatzen da, eta, hiztegiko entitateen arteko erlazio lexiko-semantikoez baliatuz, sare semantiko aberatsa eratzeko bideak eskaintzen dira (Artola, 1993; Agirre *et al.*, 1994). Horregatik integratu nahi izan dugu *Hiztsua* ELHISAn, EKOa, era horretako baliabide lexikalak integratzeko orduan, egokia denetz frogatzeko.

Hiztsua-ren errepresentazio-ereduan hiru ezagutza-base ageri dira, elkarrekin erlazionatuak: *Structures, Dictionnaire* eta *Thesaurus*. Entitate lexikalak azken bietan daude, lehenengoan adierazitako meta-ezagutzaren arabera antolatuak. Hori dela eta, hemen *Dictionnaire* eta *Thesaurus* ezagutza-basee-

Erlazioa	Domeinua	Heina	Parte-hartzea (dom : heina)
Hitz eta kontzeptuen artekoak			
conceptId	EDR_Words	EDR_Concepts	(1,1) : (1,n)
headConceptEN	EDR_Words	EDR_WordsEN	(1,1) : (1,n)
headConceptJA	EDR_Words	EDR_WordsJA	(1,1) : (1,n)
wordForm	EDR_Concept	EDR_Words	(1,n) : (1,1)
conceptExpl	EDR_Words	EDR_ConceptExpls	(1,1) : (1,1)
Hiztegi elebidunetakoak			
correspWordJAEN	EDR_WordsJA	EDR_WordsEN	(0,n) : (0,n)
correspWordENJA	EDR_WordsEN	EDR_WordsJA	(0,n) : (0,n)
Kontzeptuen artekoak			
conceptExpl	EDR_Concepts	EDR_ConceptExpls	(1,1) : (1,n)
superConcept	EDR_Concepts	EDR_Concepts	(0,n) : (0,n)
subConcept	EDR_Concepts	EDR_Concepts	(0,n) : (0,n)
conceptDescInfo	EDR_Concepts	EDR_ConceptDescr	(0,n) : (1,1)
...			

IV.4 Taula: Objektu lexikalen arteko erlazio batzuk, *EDR*ren BEKean.

tako elementuak hartuko ditugu kontuan, baliabidearen eredu kontzeptuala modelizatzeko orduan.

Honako objektu-klase hauek aurkitzen ditugu sisteman, informazio lexikalari dagokionez (ikus IV.13 irudia):

- Sarrerak (hiztegikoak).
- Kontzeptuak: mota-kontzeptuak (*concepts type*), kontzeptu sintagmatikoak (*configurations*) eta kontzeptu anbiguoak (ezagutza-basearen eraikuntza-prozesuan erabat desanbiguatu gabe gertatutakoak).

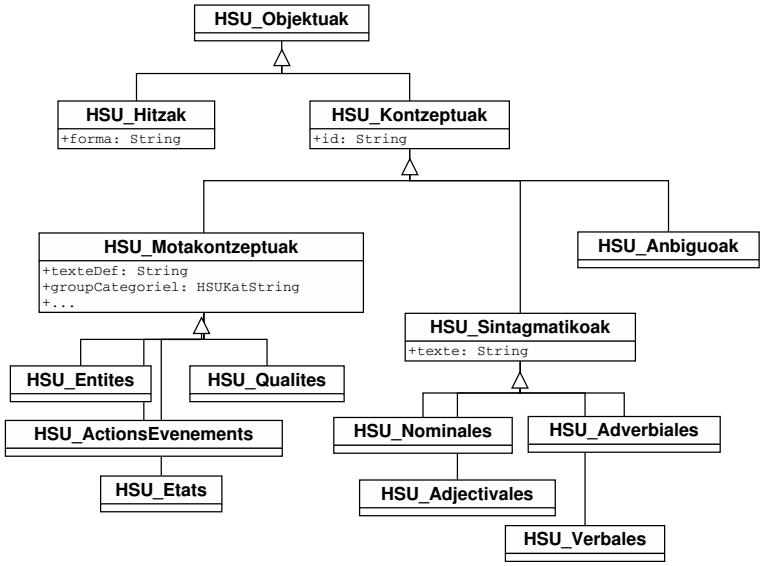
Gorago esan bezala, *Hiztsua*ren *Structures* ezagutza-basean sistemaren meta-ezagutza errepresentatzen da klase-hierarkia baten bidez, eta klase horietan hainbat atributu definitzen dira. Baliabidearen eredu kontzeptual honetan ez ditugu hor azaltzen diren erlazio-atributu guztiak ipiniko, ezpada erlazio lexikalak adierazten dituztenak batik bat. IV.5 taulan *Hiztsua*ko erlazio esanguratsuenak ikus daitezke.

IV.2.2.5 Ezagutza-baseak: Sinonimo multzoetan oinarritutako *EuroWordNet*.

WordNet (Miller, 1990) da munduan gehien erabili eta erabiltzen den ezagutza-base lexikala. Sinonimo multzoetan oinarritua eta goi-mailako ontologia

Erlazioa	Domeinua	Heina	Parte-hartzea (dom : heina)
Hitz eta kontzeptuen artekoak			
sens	HSU_Hitzak	HSU_Motakontzeptuak	(1,n) : (1,1)
motEntree	HSU_Motakontzeptuak	HSU_Hitzak	(1,1) : (1,n)
Definizio-eskema tipikoei dagozkienak			
defClassique	HSU_Motakontzeptuak	HSU_Sintagmatikoak	(0,1) : (0,n)
defSynonyme	HSU_Motakontzeptuak	HSU_Motakontzeptuak	(0,1) : (0,n)
defEnsembleDe	HSU_Motakontzeptuak	HSU_Entites	(0,1) : (0,n)
defCeQui	HSU_Motakontzeptuak	HSU_ActionsEvenements	(0,1) : (0,n)
...			
Sintagmatikoak			
caracteristique	HSU_Nominales	HSU_Qualites	(0,n) : (0,n)
avecNom	HSU_Nominales	HSU_Entites	(0,n) : (0,n)
avecVerb	HSU_Verbales	HSU_Entites	(0,n) : (0,n)
...			
Erlazionalak			
definitionDe	HSU_Sintagmatikoak	HSU_Motakontzeptuak	(0,1) : (0,1)
definiPar	HSU_Motakontzeptuak	HSU_Sintagmatikoak	(0,1) : (0,1)
hyperonyme	HSU_Motakontzeptuak	HSU_Motakontzeptuak	(0,n) : (0,n)
hyponyme	HSU_Motakontzeptuak	HSU_Motakontzeptuak	(0,n) : (0,n)
synonyme	HSU_Motakontzeptuak	HSU_Motakontzeptuak	(0,n) : (0,n)
...			

IV.5 Taula: Objektu lexikalen arteko erlazio batzuk, *Hiztsua*ren BEKean.



IV.13 Irudia: Hiztsua ezagutza-base lexikalaren BEKa

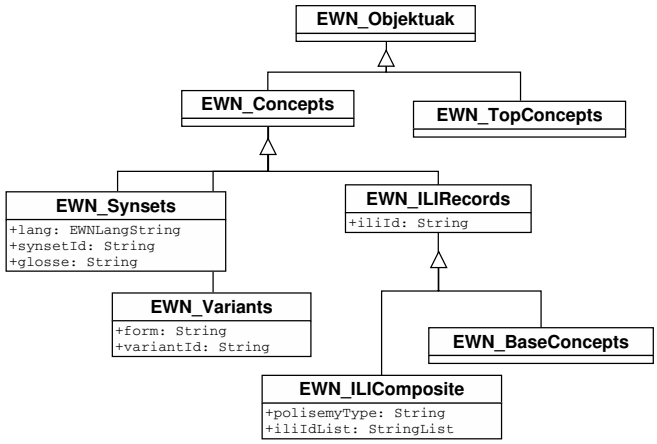
baten pean antolatua, kontzeptu andana handia biltzen du ingeleserako.

EuroWordNet (Vossen, 1997) proiektuaren ondoren, berriz, beste zenbait hizkuntzatako *wordnet*-ak ere sortu eta eratu dira, euskarazkoa barne (Agirre *et al.*, 2002), eta sinonimiaz gain beste erlazio lexiko-semantiko batzuk gehitu zaizkio ezagutza-baseari. Arlo-ontologia bat[17] (*domain ontology*) eta ontologia goren bat (*top ontology*) ere badaude, eta bertako nodoak hizkuntza-arteko indize bateko (ILI, *inter-lingual index*, hasierako *WordNet*-en oinarritua) kontzeptuekin lotuta daude, hizkuntzatik independente diren esteken bitartez. Bestalde, *EuroWordNet*-en integraturiko hizkuntzetako kontzeptuak lotuta daude, erlazio eleanitzen bitartez, ingelesezko kontzeptuekin. Horregatik integratu nahi izan dugu *EuroWordNet* ELHISAn, EKOa, era horretako baliabide lexikalak integratzeko orduan, egokia denetz frogatzeko.

Honako objektu-klase hauek aurkitzen ditugu ezagutza-basean (ikus IV.14 irudia):

- Hizkuntza bakoitzeko item lexikalak: hitzak eta adierak (*variants*).
- Sinonimo multzo edo *synset*-ak, adierez osatuak, eta gurean EKOko kontzeptutzat hartu ditugunak.

[17] Arlo-ontologia ez dugu aintzat hartu, baina, baliabidea modelizatzerakoan.



IV.14 Irudia: EWN ezagutza-base lexikalaren BEKa

- Ontologia goreneko kontzeptuak: *top concepts*.

- Hizkuntza arteko lotura bideratzeko erabiltzen den ILIko erregistroak, sinpleak zein konposatuak. ILIko kontzeptuen artean badira oinarrizko kontzeptutzat hartzen diren batzuk, hizkuntza desberdinetako *wordnet*-ak garatu dituztenen arteko nolabaiteko adostasun baten ondorioz finkatuak: *base concepts* esaten zaie horiei.

EuroWordNet-en BEKean kontzeptutzat hartu ditugu sinonimo multzoak, hizkuntzatik independente diren kontzeptutzat har baitaitezke, nolabait, nahiz eta hizkuntzaren baitakoak izan. Halaber, kontzeptutzat hartu ditugu ILIko erregistroak eta ontologia gorenekoak (horiek denak bai, hizkuntzatik erabat independente direnak). Adieratzat, berriz, hizkuntza bakoitzeko *variant*-ak hartu dira.

Hierarkiako objektu-klaseen arteko erlazioak hiru mailatan sailkatu ditugu (ikus IV.6 taula):

- Sinonimo multzoa erlazionatuta dago bera osatzen duten hitz-adierekin. Hitz-adiera horiek, elkarrekiko, sinonimoak dira.

- Kontzeptuen arteko erlazioak, berriz, bi multzotan bereizi ditugu: batetik, hizkuntza baten barrukoak direnak, sinonimo multzoak erlazio lexiko-semantikoen bitartez (hiperonimia/hiponimia, meronimia etab.) lotzen dituztenak; eta, bestetik, hizkuntza arteko erlazioak gauzatzeko helburuarekin, hizkuntzetako *synset*-en eta ILIko erregistroen artean

Erlazioa	Domeinua	Heina	Parte-hartzea (dom : heina)
Sinonimo multzoen (synset) osaketa			
osagaiDitu	EWN_Synsets	EWN_Variants	(1,n) : (1,1)
synsetekoKide	EWN_Variants	EWN_Variants	(0,n) : (0,n)
Hizkuntza baten baitakoak (*synset*-en artekoak)			
hasNearSynonym	EWN_Synsets	EWN_Synsets	(0,n) : (0,n)
hasHyperonym	EWN_Synsets	EWN_Synsets	(0,n) : (0,n)
hasHyponym	EWN_Synsets	EWN_Synsets	(0,n) : (0,n)
hasAntonym	EWN_Synsets	EWN_Synsets	(0,n) : (0,n)
hasHolonym	EWN_Synsets	EWN_Synsets	(0,n) : (0,n)
hasMeronym	EWN_Synsets	EWN_Synsets	(0,n) : (0,n)
...			
Hizkuntza artekoak (*synset* eta ILIko erregistroen artean)			
toILI	EWN_Synsets	EWN_ILIRecords	(0,n) : (0,n)
eqSynonym	EWN_Synsets	EWN_ILIRecords	(0,n) : (0,n)
hasEqHyperonym	EWN_Synsets	EWN_ILIRecords	(0,n) : (0,n)
hasEqHyponym	EWN_Synsets	EWN_ILIRecords	(0,n) : (0,n)
causes	EWN_Synsets	EWN_ILIRecords	(0,n) : (0,n)
...			
ILI erregistroen artekoak			
ILIOsagaiDitu	EWN_ILIComposite	EWN_ILIRecords	(2,n) : (0,n)
Ontologia gorenekin zerikusia dutenak.			
superTypeTop	EWN_TopConcepts	EWN_TopConcepts	(0,1) : (0,n)
superTypeBase	EWN_BaseConcepts	EWN_TopConcepts	(0,n) : (0,n)
...			

IV.6 Taula: Objektu lexikalen arteko erlazioak, *EuroWordNet*en BEKean.

ezartzen diren estekak (sinonimo baliokideak, hiperonimo eta hiponimo baliokideak, etab.). Bestalde, hor ditugu orobat ontologia goreneko kontzeptuekin zer ikusia duten erlazioak.

IV.2.3 Baliabidearen Eduki-Deskribapena (BED).

Esan bezala, ELHISAk sistemaren EKOa zein baliabideen BEKen deskribapenak behar ditu, eskuartean kudeatuko dituen datuak ulertu ahal izateko. EKOak, halaber, informazio lexikal orokorraren kontzeptualizazioaren funtzioa beteko du; erabiltzaileak EKOaren kontzeptu zein erlazioez baliatu beharko du sistemari igorritako galderetan. Galdera horiek, baina, baliabide bakoitzari igorri behar zaizkio, baliabide bakoitzaren BEKeko kontzeptu zein erlazioak erabiliz. Horrela, bada, jatorrizko galdera baliabide bakoitzak ulertzen duen adierazpidera itzuli beharko da.

EKOaren arabera adierazitako galderak BEK bakoitzera itzuli ahal izateko, BEK bakoitzak EKOarekiko duen erlazioa zehaztu behar da, ezinbestean. ELHISAn erlazio hori mapaketa-erregelen bitartez gauzatuko da, Baliabidearen Eduki-Deskribapena (BED) izeneko moduluan. BEDean, hortaz, iturri lokalen edukiak deskribatuko dira, iturri lokal bakoitzeko kontzeptu zein erlazio ororentzat EKOko kontzeptu eta erlazioekiko duen erlazioa esplizituki adieraziz.

BEDa mapaketa semantikotzat ikusi ohi da, kontzeptuen arteko mapaketa den neurrian. Mapaketa hauek kontzeptu zein erlazioen arteko lotura semantikoak adieraziko dituzte, eta galderen itzulpen-prozesua bideratuko dute. Mapaketak, hala ere, ez du kontzeptu (edo erlazio) soil bat beste kontzeptu (edo erlazio) batekin parekatuko. Izan ere, kontzeptu-kontzeptu (edo erlazio-erlazio) mapaketek ez dute baliabide lokalen izaera heterogeneoa islatuko. Esaterako, gerta daiteke eredu kontzeptualeko zenbait kontzeptu (edo erlazio) bat ez etortzea baliabide lokaleko inongo kontzepturekin (edo erlaziorekin). Konparazio batera, EH hiztegiko `Data` kontzeptuak hiztegiko sarreren estreinako agerraldiaren urtea gordetzen du, eta ez dago EKO osoan horrelako kontzeptuaren baliokide zuzenik[18]. BEDak, hortaz, mapaketa sinplea baino espresibotasun aberatsagoa duen formalismoan egon behar du adierazia.

[18]Horrela, EHko `Datak` kontzeptua EKOko `Erabilerak` kontzeptuko azpimultzo batekin erlazionatuko da: `Erabilerak` kontzeptuko instantziak, zeintzuek "data" balioa izango duten `mota` atributuan.

Levy *et al.* (1996) lanean aurkezten da, estreinakoz, eduki-deskribapenaren kontzeptua. Era berean, eduki-deskribapenei eska dakizkiekeen zenbait ezaugarri azaltzen ditu:

- Informazio-iturrien kopurua handia eta aldakorra izan daitekeenez, sistemak aukera eman behar du iturri berriak integratzeko. Iturri berriak gehitzeak, baina, ez du bitarteko eskemaren gainean inongo aldaketarik ekarri behar.

- Iturri batzuek informazio berdintsua gordetzen dutenez, deskribapenek edukiei buruzko informazio zehatza adierazteko aukera eman behar dute. Iturriei buruzko informazio doia gorde ahal izatea lagungarri suertatuko da galdera-itzulpenean, non erabaki beharko den zein informazio-iturri den egokia galdera bat erantzuteko.

- Eduki-deskribapena gauzatzerakoan, galdera-itzulpenaren prozesuaren konputagarritasuna oso kontuan hartu behar izateko auzia da. Izan ere, eduki-deskribapenak egiteko formalismoaren espresibotasunak eragin zuzena edukiko du itzulpena gauzatzeko erabilitako algoritmoetan.

Eduki-deskribapenei buruzko kezka horiek gureganatuz, hau izan da, iturri bakoitzeko BEDa eraikitzerakoan, hartu dugun irtenbidea: iturri lokaleko kontzeptu eta erlazio bakoitzerako, EKOaren gainean egindako erregela bat zehaztuko da, zeinek ezarriko duen kontzeptu (edo erlazio) lokal horren objektuek bete beharreko baldintza.

Horrela, bada, iturri lexikal bakoitzaren kontzeptu zein erlazio bakoitzeko erregela bat idatzi da. Erregelaren sintaxia zertxobait aldatzen da, definitzen dena kontzeptua edo erlazioa den arabera.

Kontzeptu bakoitzeko honako erregela bat definitu behar da:

$$V(\bar{X}) \leftarrow p_1(\bar{X}_1) \wedge \ldots \wedge p_n(\bar{X}_n)$$

non:

- V iturri lokaleko kontzeptu bat den, hots, iturri horren BEKeko elementu bat.

- p_i bakoitza eredu orokorreko kontzeptua edo erlazioa den, hots, EKOko elementu bat, eta $\bar{X}_i$-ak, kontzeptu edo erlazio horien argumentuak.

Erlazio bakoitzeko, berriz, honako erregela zehaztu behar da:

$$V(\bar{X}) \leftarrow p_1(\bar{X}_1) \wedge \ldots \wedge p_n(\bar{X}_n) \mid \alpha.$$

non:

- V iturri lokaleko erlazio bat den, berriro ere iturriaren BEKeko elementua izango dena.
- p_i-en esanahia bat dator gorago ikusi berri dugun erregelarekin.
- α erregelaren *apaingarria* den[19]. Apaingarria iturri lokaleko kontzeptu bat da, eta mapatzen ari garen erlazioaren domeinua azaltzen du, iturri lokalaren BEKaren arabera.

IV.15 irudian EDBL baliabide lexikaleko BEDaren zati bat ikus daiteke[20]. Lehenengo erregelak `EDBL_Unitateak` kontzeptua zehazten du. EDBL iturri lokalaren `EDBL_Unitateak` kontzeptua EKOko `Adierak` kontzeptuarekin egokituko da, beti ere `Adierak` kontzeptuko `hizkuntza` atributuak "`EU`" balioa badu[21]. Bestela esanda, `EDBL_Unitateak` kontzeptua, EKOaren arabera, euskarazko hitz baten adiera da[22].

Bigarren erregelak, berriz, `EDBL_sarrerak` atributua definitzen du. Atributua EKOaren `Hitzak` klaseko instantzia baten `hitzForma` atributuari egokituko zaio, baldin eta instantzia hori `Adierak` kontzeptuko instantzia batekin `adierak` erlazioaren bitartez lotuta badago.

Hala ere, kontuan izan behar da EDBL iturriak ez dituela euskaraz dauden hitzen adiera *guztiak* gordeko. Berez, iturri lokalek gordetzen duten informazioak nekez adieraziko du egon daitekeen ezagutza lexikal osoa, baizik eta ezagutza horren zati bat. Adibidez, gorago aipatutako IV.15 irudiko lehenengo erregelak esanahi konkretua du: `EDBL_Unitateak` kontzeptuaren

[19]Erlazioen erregeletan apaingarriaren beharra hurrengo kapituluan ikusiko dugu, V.2 atalean (230. orrian).

[20]Jo beza irakurleak B eranskinera ELHISAn integratutako iturrien BEDak ikusteko.

[21]Zehatzago, erregelaren esanahia honako hau da: baldin EKOan `Adierak` kontzeptuko u instantzia bat badago, zeinaren `hizkuntza` atributuak $''EU''$ balio duen, orduan u objektua, EDBLko BEKean, `EDBL_Unitateak` kontzeptuko instantzia izango da.

[22]EDBLko unitateak, sarrera-homografo bikoteak direnez, hitza baino esanahi-unitate bat errepresentatzen dute (berez adieratzat hartu ohi dena ez badira ere).

$$
\begin{array}{lcl}
EDBL\_Unitateak(u) & \leftarrow & Adierak(u), hizkuntza(u,''EU''). \\
EDBL\_sarrera(u,hf) & \leftarrow & Adierak(u), adierak(h,u), \\
 & & Hitzak(h), hitzForma(h,hf) \mid \\
 & & EDBL\_Unitateak(u). \\
\ldots & &
\end{array}
$$

IV.15 Irudia: EDBL baliabidearen BEDaren zati bat

pean EDBLk gordetzen duen instantzia multzoa EKOaren "euskarazko hitzen adierak" kontzeptuaren azpimultzoa dela.

Kalkulu erlazionalean horrela adieraziko dugu eduki-deskribapeneko V erregelaren esanahia :

$$\{\left\langle \bar{X} \right\rangle \mid V(\bar{X})\} \supseteq \{\left\langle \bar{X} \right\rangle \mid \exists \bar{Y} : p_1(\bar{X}_1) \wedge \ldots \wedge p_n(\bar{X}_n)\}$$

Eduki-deskribapeneko erregelei ematen diegun esanahi zehatz hori mundu irekiaren asuntzioarekin[23] bat dator, bete-betean, datu-integrazioaren arloan gertatu ohi den bezala. Beraz, iturri lokaleko erlazioen hedapenek erlazio horretan egon daitezkeen n-kote guztien azpimultzo bat baino ez dute adieraziko.

Bestalde, gure BEDak *Lokala bistatzat* (LAV) eredua jarraitzen du, iturri lokaleko erlazio zein kontzeptuak EKOaren arabera zehazten baitira: erlazio lokal bakoitza EKOaren gainean definitutako bista bezala ikus daiteke, gorago aipatu dugun mundu irekiaren asuntzioarekin batera. LAV eredua hartu izanak ondorio sakonak edukiko ditu galderak itzultzeko garaian, eta baita ere galderak erantzuteko espero daitekeen erantzun multzoaren egokitasuna zehazterakoan.

LAV eredua jarraitu izana, *globala bistatzat* (GAV) ereduaren aldean, arrazoi sendo batetik dator. GAV ereduan, mapaketa-erregelen noranzkoa LAVarekiko alderantzizkoa da, hots, EKOaren klase zein erlazioak baliabide lokaleko erlazio zein klaseen arabera —BEKekoen arabera, alegia— adierazten dira. III.3.2 atalean ikusi den bezala, GAV ereduko integrazioan galderaren prozesaketa errazagoa da LAV ereduan baino. Integrazio-sisteman informazio-iturri berri bat integratzea, ordea, lan korapilotsuagoa da, zeren iturri berri bat izanik, EKOko erlazio eta kontzeptu orotako datuak bertatik eskuratu ahal izateko egon daitezkeen bide guztiak —iturri berriaren eta

[23] Ikus III.3.3.1 atala, 102. orrian, mundu irekiaren zein mundu itxiaren asuntzioak zertan dautzan jakiteko.

jadanik integratutakoen artean erlazionatzeko era posible guztiak— aztertu behar baitira. Horrek, berriz, EKOa berridaztea ezartzen du maiz.

GAV hurbilpenak ezartzen dituen murriztapenak zorrotzegia dira guk eraiki nahi dugun integrazio-sistema lexikalerako. EKOa birdiseinatu behar izatea —behin eta berriro— baliabide lexikal berri bat integratu nahi denean, ez dator bat EKOaren izaerarekin. Izan ere, nekez kontsidera dezakegu EKOa informazio lexikalaren errepresentazio orokor eta unibertsala, hain aldakorra bada.

LAV hurbilpenaren pean, berriz, iturri lokalak banaturik definitzen dira, EKOaren arabera adierazita baitaude, beste iturriekin inongo erlaziorik ez dutelarik. Horrela, bada, iturri berri bat integratzean, bere BED erregelek ez dute zertan gainerako iturrien erregelen gainean aldaketarik ezarri behar.

Hala ere, pentsatzekoa da, zenbaitetan, EKOaren gainean aldaketarik egitera behartzen duen egoerarik aurki daitekeela. ELHISAk proposatutako integrazio-eskema zorrotzegia ere izan ez dadin, aurreikusi behar da, baliabide berri bat onartzerakoan, EKOan erlazioak edo kontzeptuak gehitu behar izatea. Izan ere, integratu nahi diren iturri berriek eskema orokorrean aurreikusita ez dauden erlazioak eduki baititzakete, eta erlazio horiek ezin izango dira sisteman integratu, baldin eta eskema orokorrari erlazio berriak gehitzen ez bazaizkio.

Horrelakoetan, LAV hurbilpena jarraitzeak sistema osoaren malgutasuna bermatzen du. Izan ere, GAV hurbilpenaren pean eskema orokorrari erlazio berriak gehitzeak eragin zuzena du integratutako gainerako informazio-iturrietan: adierazi egin beharko da erlazio orokor berri horien definizioa, iturri bakoitzeko ereduaren arabera. Horrela, EKOaren aldaketa bakoitzak iturri guztietan eragin zuzena izango du, EKOko kontzeptu edo erlazio berri guztiak iturri lokalen arabera zehaztu beharko baitira. LAV hurbilpenean, aldiz, eredu orokorraren gainean eginiko aldaketek ez dute gainontzeko iturriekiko erlazioa zertan aldatu, eta beren erlazio zein kontzeptu lokal guztiek baliozkoak izaten jarrai dezakete. Hala ere, ezin izango dira baliatu gehitu berri diren erlazioez, beren BEDean erlazio horiek islatu arte.

Azkenik, GAV hurbilpenari jarraitzean, EKOa integratuak dauden baliabideen "bildura" bezala ikusi beharko da, hau da, iturri lokalak oinarritzat hartuz, behetik gorako diseinu batetik erauzitako eskema orokorra. LAV hurbilpenak, aitzitik, askatasun osoa eskaintzen du integrazio-sistemaren eskema orokorra diseinatzerakoan, iturri lokalak ez baitira halabeharrez oinarritzat hartu behar. Hortaz, EKOa garatzeko jarraitu dugun bi norabideen hurbilpena —behetik gora eta goitik behera— ezinezkoa izango litzateke GAV

ereduaren pean.

IV.3 Bitartekoa eta galderen itzulpena.

Aurreko ataletan behin eta berriro aipatu dugun legez, erabiltzailea EKOko kontzeptu zein erlazioez baliatuko da ELHISAri galderak egiterakoan. EKOan agertutako kontzeptu eta erlazio horiek, ordea, ez dira berez existitzen, hots, erlazio eta kontzeptu *birtualak* dira. Horrek esan nahi du, besteak beste, erlazio horien hedapenik ez dagoela inon gordeta, ez baitago kontzeptu edo erlazio horien pean dagoen n-koterik. Hala ere, galderaren erantzun multzoa osatuko duten n-kote errealak lortu behar dira, erabiltzaileari erantzun behar bazaio. N-kote horiek, jakina, iturri lokal bakoitzean daude metaturik, eta, beraz, bertara jo beharko du ELHISAk galderaren erantzunen bila.

Iturrietara jotzeko, baina, bi prozesu burutu beharko dira jatorrizko galderaren gainean, ezinbestean. Batetik, galdera horri erantzuteko zein iturri den egoki erabaki behar da. Izan ere, zenbait baliabidek ez dute galdera zehatz bat erantzuteko informaziorik. Adibidez, EDBL datu-base lexikalak ez du hitzen definiziorik eskaintzen; EH-k edo EDR-k, ordea, bai.

Galdera batentzat zein baliabide diren egokiak ebatzi ondoren, EKOaren gainean eginiko jatorrizko galdera hori iturri bakoitzak ulertzen dituen galderetara itzuli behar da. Galderen Itzultzaileak EKOaren arabera dagoen jatorrizko galdera itzuliko du, baliabide lokal bakoitzaren BEKaren arabera adierazita egon dadin. Itzulpena ezin da nolanahikoa izan, jakina. Erabiltzaileak ez du ELHISAren lana sumatu behar, eta iturri lokaletan zehar banaturik dauden datu guztiak datu-base bakar batean egongo balira bezala ikusi behar ditu. Hortaz, berak ipinitako galderaren erantzun multzoak datu-base handi horretan espero zezakeenaren baliokidea izan behar du.

ELHISAn, bi eginkizun horiek bitartekoak burutuko ditu. Funtsean, erabiltzaileak galdetutakoari erantzuna eskainiko dio, iturri bakoitzean erantzuna eskuratu ahal izateko ipini behar den galdera logikoa osatuz. Hurrengo azpiataletan azalduko da prozesu hori xehetasunez.

IV.3.1 Galderen Itzultzailea.

Bitartekoari galdera konjuntibo bat iritsiko zaio, erabiltzaileak sistemari egindako galderaren adierazgarri. Berak Galderen Itzultzailea erabiliko du, informazio-iturri lokal bakoitzaren arabera adierazita dauden berridazketak lor

ditzan. Itzultzailearen emaitza Optimizatzaileak jasoko du, berridazketa guztien artean zenbait galdera erredundante baztertzeko.

Arestian aipatu den legez, informazio-iturriak LAV hurbilpenaren bitartez modelizatu izanak eragin zuzena du galdera-itzulpenaren prozesuan. Eragin horren zergatia argitzeko, ikus dezagun, lehenik eta behin, zertan datzan alderantzizko hurbilpenaren pean —GAV hurbilpena, alegia— eginiko galderen itzulpena.

GAV hurbilpenean, gorago esan bezala, eskema orokorraren erlazio zein kontzeptuak iturri lokalen erlazioen (edo kontzeptuen) arabera deskribatzen dira; adibidez, galdera konjuntiboen bidez. Mapaketa gauzatzen duten galdera konjuntiboek eskema orokorrarekin bat datozen datu lokalak zeintzuk diren esplizituki adierazten dute. Bestela esanda, mapaketek erlazio zein kontzeptu orokorrei dagozkien datuak eskuratzeko bidea adierazten dute. Halaber, esan daiteke mapaketek iturri lokaletan gordetako datuen adierazpen esplizitua eskaintzen dutela, eta adierazpenaren oinarria eskema orokorreko kontzeptuak direla.

Hortaz, eskema orokorraren gainean eginiko galdera bat izanik, galderaren erantzuna osatuko duten erlazio eta kontzeptu lokalak atzitzeak ez dirudi hain konplexua GAV hurbilpenean: besterik gabe, galderaren azpihelburu bakoitza dagokion mapaketa-erregelarekin ordezkatu behar da, aldagaien izenekin arreta berezia ipiniz. Prozesu honi *galdera-hedapena* esaten zaio[24] (Ullman, 1997).

LAV hurbilpenean, aldiz, mapaketek ez dute adierazten, zuzenean, zein datu datozen bat eskema orokorreko erlazio edo kontzeptuekin. Aitzitik, hurbilpen honetan informazio partziala dugu soilik. Hortaz, erabiltzaileak eskema orokorraren gainean eginiko galderak erantzun ahal izateko, informazio partzialaren gainean *inferentzia-prozesuak* bideratu beharko dira. Galderak itzultzearen prozesua, LAV paradigman, misteriozko eleberriekin konparatu izan da. Eleberri horietan, detektibe argi batek —guk, alegia— misteriozko auzi bat argitu behar izaten du, horretarako hainbat lekutatik informazioa bilduz. Galdera bat itzultzea misterio bat argitzearen parekoa izan daiteke, besteak beste, arrazoi hauengatik:

- Iturri lokalak lekukoak dira, azken finean, haiek baitakite galdera erantzuteko jakin behar den guztia.

- Hala ere, lekukoa den iturri lokal bakoitzak istorioaren zati bat soilik

[24] Ikus III.3.2 atala.

ezagutzen du, eta bere baitan gordetzen dituen datuek adierazten dute iturriak dakiena.

- Bestalde, iturriek dakitenari buruzko adierazpen esplizitua dugu, iturri bakoitzaren BEKari esker.

- Guk misteriozko auzia ebatzi behar dugu —hau da, galdera erantzutea— lekukoek eskaintzen diguten informazioaz baliatuz.

Hortaz, Galderen Itzultzailearen lana informazio hori guztia ustiatzea da, eta, eskema orokorraren arabera —hots, EKOaren arabera— jarritako galdera baterako, itzuli nahi dugun iturri lexikalaren BEKaren araberako galdera sorta bat sortzea.

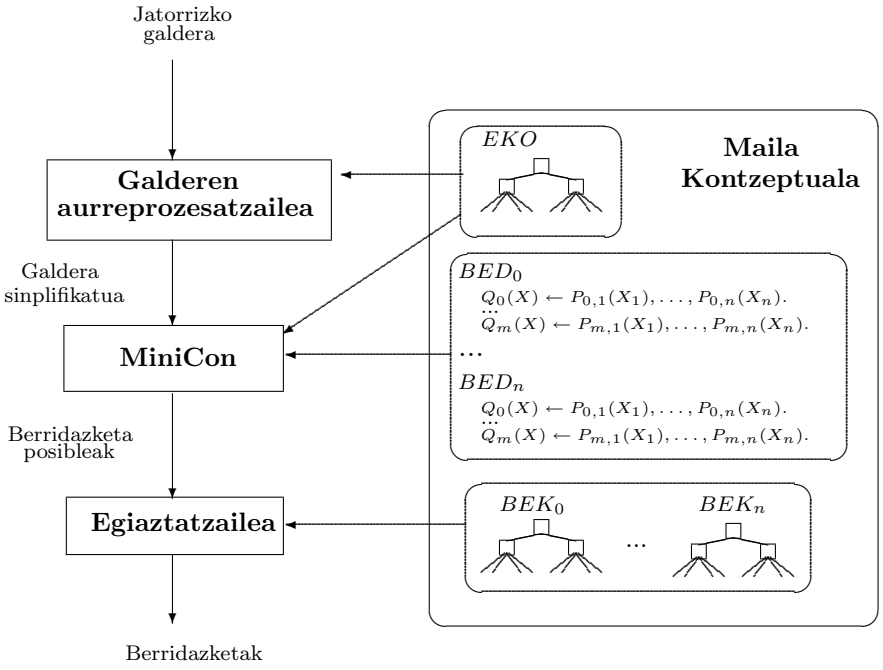
ELHISAk, auzi horretarako, *MiniCon* algoritmoaz (Pottinger eta Levy, 2000) erabiliko du, jatorrizko galderarako interesa duten iturri lokalak identifikatu, eta iturri lokal bakoitzaren araberako berridazketak lortzeko. III.3.3.2 atalean azaldu dugu[25] *MiniCon* algoritmoa zertan datzan, eta algoritmo horri esker lortutako berridazketen *onurak*, hots, sortutako berridazketak zein neurritaraino diren baliokideak jatorrizko galderarekiko. Izan ere, *MiniCon* algoritmoaren bidez sortutako berridazketak *maximoki barne-harturiko berridazketak* baitira galdera konjuntiboekiko, hots, jatorrizko galderak lortuko lituzkeen datu berberak itzultzen dituztenak.

IV.16 irudian Galderen Itzultzailea ikus daiteke eskematikoki. Funtsean, erabiltzaileak igorritako galdera —EKOaren arabera adierazita— *Galderen aurreprozesatzaileari* iritsiko zaio. Honek galdera analizatu eta berridatziko du, *MiniCon* algoritmoari pasatu aurretik. *MiniCon* algoritmoak, iturrietako BEDetan adierazitakoaren arabera, galdera itzuliko du, iturrietako kontzeptu eta erlazioak soilik erabil ditzan. Emaitza gisa, hainbat berridazketa sortuko ditu, *berridazketa posibleak* deiturikoak. Algoritmoaren gainean zenbait aldakuntza egin ditugu, guk erabiltzan dugun datu-eredura egokitu dadin[26]. Irudian ikus daitekeen bezala, berridazketa posibleen gainean egiaztatze-prozesu bat burutu behar da. Izan ere, gerta baitaiteke berridazketa posibleen artean iturriko BEKekiko zilegia ez denik ere agertzea[27] —eta, beraz, baztertu beharreko berridazketa izatea.

[25] 103. orrian.

[26] Aldakuntzak hurrengo kapituluko V.2.1 atalean ikus daitezke, 232 orrian.

[27] *Minicon*-en emaitza iturriekiko zilegi ez izatearen zergatia hurrengo kapituluan azalduko dugu —V.2 atalean, 230. orrian—, ELHISAren portaeraren zenbait adibide ikusterakoan.



IV.16 Irudia: Galderen Itzultzailea

Algoritmoak, jatorrizko galdera itzultzerakoan, iturri desberdinen BEDak banan-banan hartzen ditu, eta era berean sortzen ditu berridazketak. Alegia, *Minicon* algoritmoaren emaitzako berridazketa oro itur bakar baten BEKaren arabera soilik egongo da adierazita. Horrela, galderen itzultzaileak, erabiltzaileak jarritako galdera itzultzeko, ez ditu iturri guztiak batera kontuan hartzen, eta, aitzitik, iturri bakoitzaren informazioa isolatua ustiatzen du.

Aipatu berri dugun ezaugarri horrek hasiera batean dirudiena baino garrantzia handiagoa izan dezake: iturri batek galderako azpigaldera guztiak ez baditu estaltzen —hots, iturriko BEDeko erregeletan zehar galderako predikatu guztiak agertzen ez badira— ez baita iturri horretarako berridazketarik sortuko, nahiz eta iturria galderaren zati bat erantzuteko gai izan. Pentsa daiteke, horrela, ELHISAk iturrien informazioa batera erabili beharko lukeela, eta, ondoren, *Minicon*-ek sortutako berridazketa bakar batean iturri bateko baino gehiagotako kontzeptuak agertu beharko liratekeela.

Adibidez, erabiltzaileak izen kategoriako hitzen forma eta definizioak nahiko balitu, honako galdera hau idatziko luke:

$$\begin{aligned} Q(forma, def) \quad :- \quad & Hitzak(h), hitzForma(h, \mathbf{forma}), \\ & adierak(h, ad), Adierak(ad), kategoria(ad, kat) \\ & katBalioa(kat, "n"), definizioa(ad, d), \\ & definizioTestua(d, \mathbf{def}). \end{aligned}$$

Galdera hori sarrera gisa emanda, *MiniCon*-ek ez du, kasu, EDBL iturrirako berridazketarik lortuko, EDBLk ez baititu hitzen definizioak gordetzen. Hala ere, EDBLk kategoria jakin baten hitzak eskuratzeko balio dezake. Horrela, bada, pentsatzekoa da gure galdera-itzultzaileak iturri bat baino gehiagotara joko duten berridazketak sortzeko aukera eman beharko lukeela. Adibidez, algoritmoak, aurreko galderarako, EDBL eta EH iturriak konbinatzen dituen honako berridazketa hau itzul zezakeen:

$$\begin{aligned} Q'(forma, def) \quad :- \quad & EDBL\_sarrera(u, \mathbf{forma}), EDBL\_Izenak(u), \\ & EH\_Adierak(u), EH\_adierak(h, u), \\ & EH\_hitzForma(h, \mathbf{forma}), \\ & EH\_definizioak(u, d), EH\_def(d, \mathbf{def}). \end{aligned}$$

Berridazketa honek, lehendabizi, EDBLra jotzen du izen kategoriako hitz-formak lortzeko eta, ondoren, EH iturrira, forma horien definizioa eskuratzearren. Hala ere, erraz ikus daiteke berridazketak ez duela zentzurik. Adibidez, bertoko `u` objektua bi kontzepturen instantzia da, hots, `EDBL_Izenak` eta `EH_Adierak` kontzeptuenak. EDBL iturriak `EDBL_Izenak` adierazteko duen gako-kodea, baina, ez dator bat EH iturriak `EH_Adierak` adierazteko duenarekin, eta, hortaz, kontzeptuek bi indibiduo multzo disjuntu adierazten dituzte. Hortaz, `u` objektua ez da zilegi, bi kontzeptu disjuntuen instantzia den heinean.

Iturri guztiak batera kontuan hartzeko, iturriek informazio ez-disjuntua adierazten duten kontzeptuak izan behar dute. ELHISAk integratutako iturrien artean disjuntuak ez diren kontzeptuen kopurua, ordea, oso txikia da. Oro har, kontzeptu horiek domeinu mugatuetakoak dira —hala nola, kategoria edo azpikategoria, zeintzuen instantziak balio bera gordetzen duten— edo, bereziki, hitz-formak gordetzen dituztenak. Hala ere, jatorrizko galderan kontzeptu horietatik at dagoen predikatu bakar bat azaltzea aski da, galdera hori itzultzeko iturriak batera kontuan hartzeak zentzurik ez izateko.

Iturri guztiak batera hartzeak *MiniCon* algoritmoak sortutako berridazketa posibleen kopurua izugarri handitzen du. Horien artean, bestalde, ez dugu aurkitu iturri bat baino gehiagotara jotzen duen eta zilegia den berridazketarik[28].

IV.3.2 Optimizatzailea.

Galdera-itzultzailearen emaitza —galdera positiboa dena— Optimizatzaileak jasoko du, zuzenean, berridazketen artean informazio erreduntantea, baldin badago, ezaba dezan. Informazio erreduntantea sortuko da, besteak beste, *MiniCon* algoritmoak bide anitz jarraitzen baititu jatorrizko galderaren berridazketak lortzeko. Berriro ere, iturrien BEKaren eta EKOaren artean kontzeptu-kontzeptu eta erlazio-erlazio baino mapaketa semantiko aberatsagoak erabiltzeagatik sortuko dira berridazketa erredundanteak.

Optimizatzailea ez dugu oraindik inplementatu, luze baino lehen prototipo bat izatea nahi badugu ere. Optimizatzailearen lana oso konplexua da, eta datu-basearen arloko ikerlerro berezia da. Hala ere, uste dugu gureak honako arazoei egin beharko liekeela aurre:

- Optimizazio semantikoa. Gure optimizatzaileak, berridazketa guztiak hartuz, optimizazio semantikoa gauzatzen saiatuko beharko da. Izan ere, gerta daiteke zenbait berridazketa beste batzuen barnean egotea, eta, horrelakoetan, berridazketa orokorrenekin geratu beharko da optimizatzailea. Galderaren prozesatzailearen irteera —galdera positiboa, hots, buru bera duten galdera konjuntiboen multzo bat— jatorrizko galderaren maximoki barne-harturiko berridazketak da, galdera konjuntiboekiko. Hala ere, gerta daiteke galdera positiboko galdera konjuntibo batzuk erredundanteak izatea, bertatik eskuratutako informazioa gainontzeko galdera konjuntiboetatik jadanik eskura baitaideke. Optimizatzaileak baztertu beharko lituzke galdera konjuntibo erredundante horiek, eta, horretarako, DLetan oinarritutako barne-hartze algoritmoak lagungarri suertatuko zaizkio zalantzarik gabe (Calvanese *et al.*, 1999a, 2001).

- Kontsulten historikoa. Optimizatzaileak jadanik erabilitako kontsulten historikoa kudeatu beharko luke, eta, galdera bat izanik, galdera horixe bera aurretik egina dagoenetz egiaztatu. Horrela izango balitz, ez

[28]Normalean galderetan iturrien artean erlazionatu ezin daitezkeen entitateak azaltzen baitira —*kontzeptuak* eta *adierak* bezalakoak.

litzateke, seguru aski, iturrietara jo behar izango, galderaren emaitza datu-base lokalean gorderik egongo bailitzateke. *Cache* memoria bezala jokatuko luke, hortaz, datu-base lokalak (Goñi *et al.*, 1997). Erantzunen historikoa biltoki lokalean gordetzeak beste arazo garrantzitsu bati aurre egin beharko dio, datuen egonkortasunaren bermatzearen arazoari, alegia[29]: sistemak ziurtatu beharko du datu-base lokalean gorderiko erantzunak ez direla zaharkiturik geratu, iturri lokalak aldatu direlako.

IV.4 Planifikatzailea.

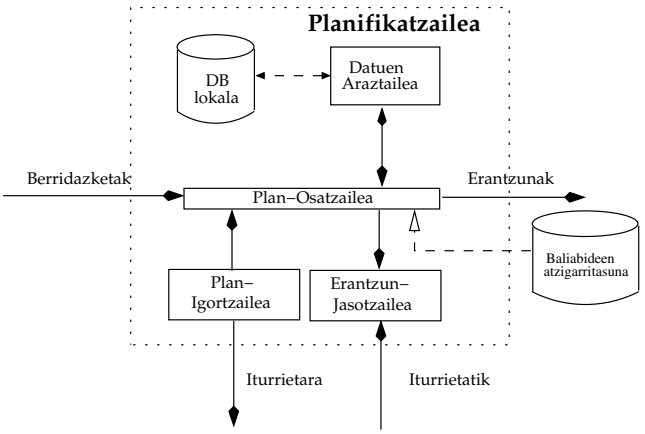
Bitarteko osoaren emaitza —iturri lokalen arabera adierazita dagoen berridazketa multzo optimizatua, erabiltzailearen jatorrizko galderaren baliokidea— planifikatzaileak jasoko du. Bere betebeharra berridazketan adierazitako informazioa eskuratzeko plan bat osatzea da, eta, emaitzen bila iturri anitzetara jo ondoren, erantzun bateratua osatu eta interfazeari igortzea.

Planifikatzaileak jatorrizko berridazketak deskonposatuko ditu, zuzenean exekutagarriak diren planak —beharbada banatuak izango direnak— lortzearren. Planifikatzailearen esku geratuko da iturrietara doazen planak antolatzea, eta plan horien exekuzio-ordena zein den erabakitzea. Tarteko urratsak ere bete beharko ditu, plan bakoitzak sortutako emaitza-datuak besteek sortutakoekin elkartu beharko baititu, azken emaitza lortzeko.

Jakina, berridazketa batetik hainbat plan desberdin era daitezke eta plan bakoitzak zama desberdina izango du, bai exekuzio-denbora aldetik, baita ere plana exekutatzeko behar den datu-jarioaren tamaina aldetik. Hortaz, Plan-Osatzailearen esku geratuko da *plan optimoa* aurkitzea, hots, exekuzio-denbora minimoa duena, edo datuen arteko elkartze minimoak baino egikarituko ez dituena. Auzi horretarako, datu-base banatuen arloan ezagunak diren kostuetan oinarritutako metrikak edo ebaluaketarako metodo bereziak aplika daitezke. Izan ere, plan logiko bat izanik, bere exekuzioa era optimizatu batean lortzearen arazoa sakon ikertutako arloa baita (ikus Meng eta Yu, 1995; Ullman, 1989; Özsu eta Valduriez, 1999).

Datu-integrazioaren arloan kokatutako planifikatzaileek badute, hala ere, zenbait ezaugarri berezi, planifikatzaile orokorretan bete ohi ez direnak. Esaterako, datuei buruzko estatistika eskasekin egin behar dute lan, integrazio-sistemako iturrien autonomia-maila handia dela eta. Estatistika horiek, bai-

[29] Arazo hauxe bera dute *gauzatutako integrazioa* delako hurbilpena jarraitzen duten integrazio-sistemek, hala nola, datu-biltegietan erabili ohi direnak. Ikus III.1.2 atala.



IV.17 Irudia: ELHISAren planifikatzailearen osagaiak.

na, arras garrantzitsuak dira planek duten kostua aldez aurretik ebaluatzeko. Bestetik, iturri lokalek, heterogeneoak izateagatik, informazio erredundante edota teilakatua eduki dezakete. Planifikatzaileek jakin beharko dute, planen kostuak minimizatzeko, iturri anitzetako datuak era eraginkorrean biltzen edo ahal den informazio erredundante gutxien eskuratzen (Kwok eta Weld, 1996; Friedman eta Weld, 1997; Ives *et al.*, 1999; Arens *et al.*, 1996).

Guk ez dugu, ELHISA garatzerakoan, planifikatzailea inplementatu, nahiz eta etorkizunean landu beharreko lerroa dela sinetsita egon, zalantza izpirik gabe. Gure erabakia justifikatzeko-edo, esan behar dugu bitartekoak, planifikatzaileak informazio-eskaria jasotzerako, planifikatzaileari egokitu ohi zaion lan nagusia egina duela: bitartekoak erabakiko du galdera baten informazioa lortzeko zein iturritara jo behar den.

Nolanahi ere den, ELHISAren planifikatzaileak beharko lituzkeen eskakizunak aztertu ondoren, bere diseinua aurreikusi dugu, eta, horrela, lau azpimodulutan banatu ditugu planifikatzailearen betebeharrak (ikus IV.17 irudia):

- Plan-Osatzailea planifikatzaile osoaren muina da, berak jasoko baititu bitartekoak sortutako berridazketak, eta, planifikatzailearen betebeharrak egikaritu ondoren, galderaren emaitza-datu bateratuak eskainiko baitizkio interfazeari. Plan-Osatzailearen esku geratuko da, baita ere, planaren exekuzio-ordena ezartzea, eta tarteko datuen gainean eragiketa lokalak burutzea[30]. Planaren exekuzioa optimizatzeko teknika edo-

[30]Datu-arazketa barne, ahal den neurrian. Plan-Osatzaileak, datu lokalak jaso bezain

nolakoa izanik ere, Plan-Osatzaileak atzigarri dituen iturri lokalei buruzko informazioa beharko du. Informazio hori guztia datu-base lokal batean metatuko da —*Baliabideen atzigarritasuna* deiturikoa—, planifikatzaileak iturri bakoitzari galdera bat bidaltzearen kostu zehatza ezagut dezan. Bestalde, ELHISA sistemaren dinamikotasuna kontuan hartuz, denboran zehar iturri baten atzigarritasuna aldakorra dela ere aurreikusi behar da: planifikatzaileak, iturri lokalei plana igorri ondoren, erantzunak jaso arte joandako denbora neurtuko du, iturrien atzigarritasunaren datu-basea eguneratuz.

- Plan-Igortzailea azpiplanen exekuzioaz arduratuko da. Funtsean, azpiplanak dagozkien iturrien gainean dauden *wrapper*-etara bidali behar ditu, iturriak atzigarri daudela egiaztatu ondoren. Izan ere, eta informazioak sistema banatu batetik bidaiatu behar duenez, iturrien dinamikotasuna oso kontuan hartu beharreko auzia baita.

- Erantzun-Jasotzaileak iturrietatik emaitzak jasoko ditu, era sinkrono batean, eta datu-base lokalean —*DB lokala* deiturikoa— gorde. Jasotzaileak harreman estuak behar ditu izan Plan-Osatzailearekin, iturri bakoitzetik bildutako informazioa iritsi bezain laster, datuak Plan-Osatzaileari igorri behar baitizkio, datu horiek gainontzeko baliabideetatik datorren informazioarekin elkar ditzan.

- Datuen Araztailea iturrietatik datozen datuak arazteaz arduratuko da. Ikus, beherago, IV.6 atalean, datu-arazketaren arazoari aurre egiteko hartutako bidea.

IV.5 Wrapper-ak.

Erabiltzailearen jakin-mina asetu behar bada, berak jarritako galderari erantzungo dioten datuak eskuratu behar dira, eta datu horiek, jakina, iturri lokaletan daude. Bitartekoari esker, erabiltzaileak jarritako jatorrizko galdera iturri lokalek ulertzen duten eredu kontzeptualen arabera egongo da adierazia; Plan-Osatzaileak, berriz, berridazketa horiek deskonposatuko ditu, eta

pronto, datu garbiketari —ikus IV.6.1 atala— ekingo dio. Iturri guztietatik etorritako datu guztiak jaso eta garbitu ondoren, berriz, objektu-identifikazioa —ikus IV.6.2 atala— burutuko du.

iturri batera baino gehiagotara joko ez duten azpiplanak osatuko ditu. Galderen Igortzaileak azpiplanak jasoko ditu Plan-Osatzailetik, eta bere lana azpiplanak iturri lokal bakoitzera zuzenean bideratzea izango da.

Hala ere, azpiplanak iturrietara igorri behar badira, eta iturrietan gorderiko informazioa berriro ELHISAri heldu behar bazaio, bi arazori egin behar zaio aurre, nagusiki: batetik, iturriek "ulertu" egin behar dute azpiplanetan adierazita dagoen informazio-eskaria, eta, bestetik, iturrietan gordetako informazioa datu-eredu lokaletik ELHISAk ulertzen duen eredu komunera itzuli beharko da.

Arazo hauei aurre egiteko, informazio-sistemek hartu ohi duten estrategia aski ezaguna jarraitu dugu: informazio-iturri lokal orok *wrapper* (Roth eta Schwartz, 1997) bat du erantsita, zeinek iturri lokalaren eta integrazio-sistema osoaren bitartekari-lanak egiten dituen. III.4 atalean[31] ikusi dugun bezala, *wrapper*-ak software-moduluak dira, iturri lokalak eta integrazio-sistemaren arteko komunikazioa ahalbideratzeko sortuak. Iturriek datuak metatzeko erabilitako egitura desberdinak edonolakoak izanda ere, haiei atxikitako *wrapper*-ek ELHISAren eta iturrien arteko komunikazioa bermatuko dute, *integrazio estrukturala* deritzogun arazoa ebazteko bidea emanaz.

Jadanik baliagarria izan zaigu iturri lokal ororen gainean *wrapper* bat erantsi izana. Izan ere, iturri lokalak erlazio multzo balitz bezala adierazteko aukera eman baitigute, eta, horri esker, iturrien eredu kontzeptualak zehaztu ahal izan dira ELHISAn, iturrietako erlazioak eredu kontzeptualarekin lotzen dituzten mapaketa semantikoekin batera. Orain dugu tenorea, baina, iturrien gainean eginiko abstrakzioaz sakondu eta zenbait iturriren gainean garatutako *wrapper*-en lana aztertzeko.

ELHISAren funtzionalitatea osatzeko, bi *wrapper* desberdinen garapenari ekin diogu —bata, datu-base erlazionalen gainean dago, eta, bestea, berriz, XMLz kodeturiko datu-base baten gainean— Informatika Fakultateko karrera-bukaerako proiektu baten pean (Valverde, 2003). *Wrapper*-ek, sarrera gisa duen galdera positiboak jaso, eta iturri lokal zehatzen kontsulta-lengoaiaz adierazitako galdera bihurtzen dituzte. Inplementatu ditugunei dagokionez, batak SQL lengoaiara itzuliko ditu galderak, eta besteak, berriz, *XQuery* lengoaiara.

Galderak iturri lokalean exekutatu ondoren, sortutako emaitza-datuak ELHISAri igortzen dizkiote bueltan. ELHISAren eta iturrien arteko komunikazioa bermatzeko, datu-eredu aski ezaguna erabili dugu: OEM (*Object*

[31] 121. orrian.

Exchange Model; (Papakonstantinou *et al.*, 1995a); ikus IV.5.1 atala beherago) delako eredu sasi egituratua, XMLz kodetua[32].

III.4 atalean[33] ikusi dugun bezala, *wrapper*-en garapena lan neketsua da, baliabideen izaerak zeharo baldintzatua. Gure esperientzia erabat bat dator aurreko adierazpenarekin: eredu erlazionaleko datu-basearen *wrapper*-arentzat eginiko lana ez baita oso lagungarria izan bigarrenarentzat —XMLz kodeturiko datu-base batentzat diseinatua.

Atal berean ikusi dugu, baita ere, *wrapper*-ak era automatiko edo erdi-automatikoan garatzeko zuzendutako ikerlanak. Hala ere, esan dezagun guk ez dugula lan honetan *wrapper*-en osaera automatikoari zuzendutako teknikетan sakondu, eta ELHISArako *wrapper*-ak garatzeak eskulangintzarekin izan duela zerikusi handiagoa, *wrapper*-ak automatikoki garatzen saiatzen diren teknikekin baino.

IV.5.1 OEM lengoaia.

Standford unibertsitateko TSIMMIS proiektuaren[34] barruan garatua, OEM eredu erraza eta malgua da, informazio-sistemen arteko komunikazioa bermatzeko balio duena. OEM eredua autodeskribatzailea da: objektuen egiturak ez du aldez aurretik definitua egon behar, eta ez dago eskema finkoaren edo objektu-klaseen beharrik. Horren ordez, OEMko objektu orok etiketa deskribatzaile bat du erantsita, zeinek adieraziko duen objektuaren egitura zein den. Horrela, klaseak, metodoak edota herentzia, zuzenean onartzen ez baditu ere, OEM ereduak emula ditzake (Papakonstantinou *et al.*, 1995a).

OEMko objektuek honako egitura jarraitzen dute:

Etiketa	Mota	Id.	Balioa

non

- **Etiketa**: gorago aipaturiko etiketa deskribatzailea den, objektuaren egitura zein den jakiteko balio duena;
- **Mota**: objektuaren balioaren mota. Motak *atomikoak* (`string`, `integer` eta abar) edo multzoa (`set`) izan daitezke;

[32] OEM objektuak XMLz kodetzeko jarraitu dugun DTDa eranskinetan aurki daiteke.
[33] 121. orrian.
[34] Ikus III.6 atala.

- **Id**: objektuaren identifikatzaile unibokoa;
- **Balioa**: objektuaren balioa.

Egitura sinple bezain malgu honen bidez, iturri heterogeneoko datu sorta handia adieraz daiteke, egitura konplexutakoak zein lauetakoak. Adibide gisa, demagun EH hiztegian "lehen" hitzaren definizioa eskatu dela. Honako erantzuna jasoko dugu, iturriak erantsia duen *wrapper*-etik:

```
<oem>
  <match n="1">
    <obj id="Definizioak_EH_def_699" label="Definizioak"
         type="String">
      Aurretik besterik ez duena, beste guztien
      aurretik gertatzen dena.</>
    </>
  </>
</>
```

XML lengoaia lagun, OEMko objektu bakoitza `obj` etiketa batez dago kodeturik, eta OEMko objektuek duten lau osagaietatik hiru —identifikatzailea, etiketa eta mota— etiketaren atributuak izango dira. Objektuaren balioa, berriz, `obj` elementuaren barruko testua izango da.

OEM lengoaiaz adieraz daitezke, baita ere, balio konposatuak, eta, horietan, osagai guztiak elkartuko dituen erro objektu bat agertuko da erantzunean. Adibidez, demagun "lahar" hitzaren definizioa zein adiera identifikatzailea nahi ditugula —hots, iturri lokaleko bi atributuren elkarketa—. Honako emaitza hau jasoko dugu:

```
<oem>
  <match n="1">
    <obj id="EH_konp_01" label="Objektu konplexua" type="set">
      Adierak_EH_AdieraId_138, Definizioak_EH_def_92
    </>
    <obj id="Adierak_EH_AdieraId_138" label="String" type="String">
      1
    </>
    <obj id="Definizioak_EH_def_92" label="Definizioak"
         type="String">
```

```
      Sasia.
    </>
  </>
</>
```

Ikusten denez, `EH_konp` identifikatzailea duen objektua emaitzaren adierazpena da, gainerako objektuak eskuratu eta antolatzeko abiapuntutzat hartu behar dena: berak adierazten du emaitza bi objektuz osaturiko multzoa dela, eta baita ere objektu horien identifikatzaileak zeintzuk diren.

OEM lengoaiak, XMLz kodetua, eredu paregabea eskaintzen digu *wrapper*-en eta sistemaren arteko komunikazioa bermatzeko. Lengoaia oso malgua denez, informazio sorta handia kodetzeko aukera ematen du. Bestalde, XML teknologia Interneten hain erabilia izanik, sistemaren alde banatuari lotutako hainbat arazo ekiditeko aukera eman digu.

IV.6 Datu-arazketa ELHISAn.

Aurreko ataletan ikusi dugun legez, baliabideetan adierazita dauden entitateak eredu komun batekin parekatu behar dira, informazioa integratzen duen sistema bat eraiki nahi bada. Intentsio mailakoa den parekatze horrek sistema osoan erabiliko den terminologia eskainiko du, eta, ondorioz, baliabide lokalak komunikatu ahal izango dira sistemarekin.

Baliabide lokal bakoitzari —hobe esanda, baliabide bakoitzak atxikita duen *wrapper*-ari— galdera bat igorri ondoren, berak galdera exekutatu eta exekuzio horren emaitza (erlazio baten n-kote multzoa) itzuliko du sistemara. Hala ere, hainbat baliabide heterogeneotatik jasotako emaitza multzoaren benetako integrazioa burutzeko, sistemak datu mailako parekatzea burutu behar du, hots, maila estentsionaleko parekatzea.

Datuen arazketa[35] lan konplexua izan da historikoki, zama handia eskatzen duena (Rahm eta Do, 2000). Prozesua bereziki garrantzitsua da datu-biltegietan eta *web* orrien informazioa integratzen duten sistemetan. Datu-biltegiak industria munduan erabilia dira, eta, erabilgarria izan behar badute, integratutako informazioaren gainean datu-analisia eta erabakiak hartzeko aukera eman behar dute. Horrela, bada, datu integratu horiek munduko entitateei buruzko informazio zehatza eta doia eskaintzea funtsezkoa da (Jarke *et al.*, 2000; Calvanese *et al.*, 1999b).

[35] ikus III.3.5 atala, 116. orrian.

Web orrietatik jasotako informazioaren integrazioan, bestalde, informazioaren heterogeneotasunaren maila handia da oso, eta, maiz, datuen antolaketa ezartzen duen eskema zehatzik ez dago. Beraz, datuen *zikintasun-maila* ez da nolanahikoa izango (Knoblock *et al.*, 2001).

Arazo hauek, baina, baliabideetan inkonsistentziak aurkitzen direnean azalduko dira, hau da, integrazio-sistema bateko baliabide batzuen informazioa fidagarria ez denean, edo baliabide lokalek metatutako informazioa erabat zehatza ez denean. Hala ere, eta salbuespenak salbuespen, informazio oso zehatza gordeko da baliabide lexikaletan: baliabide batzuek, lengoaia naturaleko tresnen jatorrizko iturriak diren neurrian, ahal den doien adieraziko dute hitzei buruzko informazio lexikala. Dena dela, hain doia ez den informazio lexikala errepresentatzen duten baliabideak egon ere badaude, eta ELHISAk aukera eman behar du informazio-iturri hauek ere integratzeko.

Oro har, ELHISAn integratu nahi diren baliabideak bi motatakoak izango dira:

- Tresna linguistikoen hornitzaile izango diren datu-base / ezagutza-base lexikalak. Esan bezala, baliabide hauetan metaturiko informazioa ahalik eta modu zehatzenean egongo da adierazia.

- Giza-erabiltzailearentzako baliabideak, hala nola, hiztegiak. Baliabide hauetan, gizakientzat zuzenduak izanik, zenbait inkonsistentzia eta datu akastun aurki ditzakegu. Hiztegietan errepresentatutako informazioaren *zikintasun*-maila datu-base lexikalena baino handiagoa bada ere, ez da, esaterako, *web* orrietan gordetzen denaren mailera iristen.

Hori horrela izanik, ELHISAren lana, datuen garbiketa dela eta, nahikoa xumea izango da, batez ere beste integrazio-sistema batzuekin alderatuz (adibidez, Knoblock *et al.* (2001)).

ELHISAk bi mailako datu-arazketa burutuko du emaitza-datuekin. Arazketa aurrera ateratzeko, sistemak bi eginkizun burutzen ditu, bata bestearen ondoren: datu-garbiketa eta objektuen identifikazioa. Ikus ditzagun bi prozesu hauek, banan-banan.

IV.6.1 Datu-garbiketa.

Lehenengo urratsean, batez ere, datu-garbiketari zuzendutako araztea dugu. Datu-garbiketako prozesua baliabide bakoitzetik jasotako emaitzen gainean egingo da, hau da, erantzunak jaso bezain laster.

hitzForma	kategoria	erabilera	definizioa
anixko	zenbatz.	Iparr eta Naf.	askotxo
ankila	ize	bizk.	aingura
zabaldu, zabal	ad.	—	Zabalera handiagoa eman
zabalgo	iz.	Iparrald.	Zabaltasuna; zabalera

IV.7 Taula: EH hiztegitik jasotako datu "zikinak"

(Rahm eta Do, 2000) lanaren oharrei jarraiki, ELHISAren garbiketa-prozesuak hiru eginkizun beteko ditu, ordena zehatz honetan:

- Forma libreko atributuetatik datuak erauzi. Arestian aipatu dugun bezala, zenbait baliabidek datu anitz meta ditzakete atributu bakar batean, batez ere atributu honen domeinua forma libreko testua denean. Datuak erauztea helburu duen prozesu bat bideratu behar da atributu horien gainean, txertatutako informazioa lortzearren.

- Egiaztapena eta zuzenketa. Zenbait prozesu aplika daiteke datu akastunen gainean, datu horiek zuzentzearren. Prozesu tipikoa zuzenketa ortografikoa litzateke, honen bidez sakatze-erroreak konpon baitaitezke. Beste prozesu tipiko bat akronimoen hedapena litzateke, etab.

- Estandarizazioa. Integrazioa errazteko, atributuak forma tinko eta uniforme batera itzuli beharko dira. Adibidez, data eta denborari buruzko informazioa formatu espezifiko batera bihurtu beharko da. Informazio lexikalari dagokionez, zenbait datu ere (kategoriak, azpikategoriak, etab.) modu uniforme batera mapatu behar dira. ELHISAk PAROLE proiektuak (Ruimy *et al.*, 1998) proposatutako balio-zerrendarekin bat egiten du kategoria zein azpikategoria lexikalak adierazteko.

ELHISAk baliabide bakoitzetik jasoko duen erantzuna n-kote multzo bezala adierazia dago, azken finean, baliabideari igorritako galderaren buruko aldagaien balio zehatzak baitira. IV.7 taulan Euskal Hiztegia (EH) baliabide lokaletik jasotako erantzun multzoa azaltzen da. Ikus daitekeenez, zenbait datu "zikin" agertzen dira: *kategoria* eremuan, kategoria lexikalen forma aldakorra da. Hala ere, pentsa daiteke "iz." eta "ize" kategoria beraren erreferentzia direla, hots, "izena" kategoriarenak[36]. Bestalde, *hitzForma* eta baita *erabilera* eremuetan ere zenbait datu bikoiztu azaltzen direla ohar gaitezke.

[36] "noun", PAROLE estandarra jarraitzen badugu.

hitzForma	kategoria	erabilera	definizioa	Mota
anixko	numeral	Ipar.	askotxo	(1,3)
anixko	numeral	Naf.	askotxo	(1,3)
ankila	noun	Bizk.	aingura	(1)
zabaldu	verb	—	Zabalera handiagoa eman	(1,3)
zabal	verb	—	Zabalera handiagoa eman	(1,3)
zabalgo	noun	Ipar.	Zabaltasuna; zabalera	(1,3)

IV.8 Taula: EH hiztegitik jasotako erantzun "garbia"

Eremu horietan balio bat baino gehiago txertatua denez ("Iparr eta Naf." edo "zabaldu, zabal"), erregistro horiek guztiak bikoiztu egin behar dira, erregistro bakoitzak eremu honen balio bakarra gorde dezan. IV.8 taulan datu-garbiketaren ondoren geratutako erregistro multzoa agertzen da. Ikus daitekeenez, balioak estandarizatu egin dira (*kategoria, erabilera*), eta datu txertatuak bikoiztu (*hitzForma, erabilera*).

Garbiketa aurrera eraman ahal izateko, bestalde, ELHISAk atributuen domeinu abstraktuei (Calvanese *et al.*, 1999b) lotuak dauden garbiketa-erregelak egikarituko ditu. Domeinu abstraktu bakoitzak erregela multzo propioa du, non domeinuko instantziek metaturiko informazioa garbitzeko behar diren urratsak zehazten diren, gainontzeko iturrietatik jaso denarekin erlazionatu ahal izateko.

Erregelek, beraz, baliabide bakoitzean datuak garbitzeko burutu behar diren eginbeharrak era deklaratibo batean zehazteko aukera emango digute. Halaber, baliabideko atributuen domeinu abstraktuei loturik behar dute izan, hots, baliabideko atributu-mota bakoitzak erregela multzoa eduki behar du atxikita. Izan ere, garbiketa-prozesua desberdin egin behar da domeinu abstraktuaren arabera. Konparazio batera, zenbait domeinutako instantziak estandarizatu egin behar dira. Beste domeinu abstraktu batzuetako instantziek, bestalde, informazio txertatua eduki dezakete beren baitan, edota informazio akastuna gordetzeko arriskua. Azkenik, zenbait domeinu abstraktutako balioak beti dira zuzenak, hau da, ez dute inongo garbiketarik behar.

Erregelak inplementatzeko, *NeoClassic* DLaren motorrak erregelak adierazteko duen aukeraz baliatu gara. Domeinu abstraktu baten instantzia agertu bezain laster, sistemak erregelak askatuko ditu, aurreranzko kateamenduaren bitartez, instantziaren balio zehatza "garbituz". Halaber, *NeoClassic*ek erregeletan *erabiltzaileak definitutako funtzioak* (EDF) zehazteko aukera ematen du. Hona hemen erregela hauen sintaxia:

```
(createRule erregelarenIzena
            aurrekaria
            atzekaria)
```

non:

- `erregelarenIzena` erregelaren izena den.
- `aurrekaria` erregela exekutatu ahal izateko bete behar den betekizuna den. Aurrekaria bi motatakoa izan daiteke: *Neoclassic* sistemako deskribapen arrunta, edo erabiltzaileak definituriko test-funtzio (predikatu) berezia.
- `atzekaria` erregela askatu ondoren bermatuko den baldintza. Berriro ere, bi motatakoa izan daiteke: *Neoclassic* sistemaren ezarpen arrunta, edo erabiltzaileak definituriko funtzioa.

Hiru erregela mota definitu ditugu: txertaketarako erregelak, zuzenketa-erregelak eta normalizazio-erregelak, bat gorago aipatutako garbiketa-prozesu bakoitzeko:

- *Txertaketarako erregelak:* atributuetan balioak txertatuak egon daitezkeenean, txertaketa egiaztatu, eta, txertaketarik balego, balio sinpleak erauziko lituzketen zenbait txertaketa-funtzio (EDF) erabiliko dira.
- *Zuzenketa-erregelak:* atributuren batean balio akastunak agertu ahal badira, balio hauek zuzenak direnetz egiaztatu, eta, akats tipografikorik egongo balitz, horiek zuzenduko lituzketen zenbait zuzenketa-funtzio (EDF) erabiliko dira.
- *Normalizazio-erregelak:* atributuen normlizazioari begira, domeinu abstraktu bakoitzetik sistemak erabiltzen duen domeinura bihurtzeko *hash* bat behar da. Konparazio batera, EHko `kategoria` atributuaren domeinua `EHKategoria` da, eta, hortaz, `EHKategoria` domeinutik `Kategoria` domeinura bihurtzeko *hash*a adierazi beharra dago. IV.9 taulan EH hiztegiko kategorien *hash*a ikus daiteke. *Hash* honek EH iturriko kategoria-balio posible oro PAROLEk proposatutako batekin mapatzen dituelarik. Mapaketa hauek gauzatu ahal izateko, `mapaketaEDF` EDF berezia dago (ikusi beheko adibidean).

Ikus ditzagun datu-garbiketa gauzatuko duten erregelen adibide pare bat:

ad	⇒	Verb
determ	⇒	Determiner
erak	⇒	Determiner
interj	⇒	Interjection
iz	⇒	Noun
izlag	⇒	Adjective
izond	⇒	Adjective
izord	⇒	Pronoun
junt	⇒	Conjunction
lok	⇒	Adverb
zenbatz	⇒	Numeral

IV.9 Taula: EHrako kategoriaren normalizazio-hash-a (`EHKatHash`)

IV.6.1 Adibidea EH baliabidean kategoria lexikalei dagozkien balioak estandarizatu:

```
(createRule EHKatNorm
            EHKategoria
            (computedFillers mapaketaEDF balNorm katBalioa
                     EHKatHash)
```

`EHKategoria` motako instantzia bat sartu ondoren, erregela hau bere balioa normalizatzen saiatuko da. Horretarako, `mapaketaEDF` funtzioa erabiliko du, non *hash* baten arabera instantziaren balioa estandarizatu eta `balNorm` atributuan gordeko duen. □

IV.6.2 Adibidea EH baliabidean erabilera-eremuan balio txertatuak azalarazi:

```
(createRule EHErabileraTxertaketa
            (and EHErabilera
                 EHErabileraBikoiztuaP)
            (computedFillers EHErabileraBikoiztuEDF
                 erabileraBalioGarbiak balioa))
```

Erregela horrek EH baliabideko erabilera eremuan balio txertatuak azalaraziko ditu. `EHErabilera` motako instantzia bat sartu ondoren, baldin instantzia horrek `EHErabilpenaBikoiztuaP` predikatua betetzen badu, erregela honek bere balioa bikoiztuko du. Horretarako, `EHErabileraBikoiztuEDF`

EDFa erabiliko du, non balio bikoiztuak `erabileraBalioGarbiak` atributuan gordeko dituen [37]

□

Informazio hori guztia baliabide bakoitzean desberdina izan daiteke, eta, hortaz, mapaketak eta EDFak zehaztu beharko dira baliabide berri bat ELHISAn integratzerakoan.

IV.6.2 Objektuen identifikazioa.

Datu-garbiketaren prozesua baliabide bakoitzaren emaitzaren gainean egingo da, modu independentean. Prozesuaren ondoren emaitza garbituak egongo dira, eta une horretan egongo gara prest emaitza guztiak elkarrekin erlazionatzeko. Informazio hau guztia lotzeko, baina, azken urrats bat geratzen da, hots, entitate berari erreferentzia egiten dioten objektuak identifikatzea, eta datu bikoiztuak ezabatzea. Azken urrats horri objektu-identifikazioa deitu diogu.

Ikus dezagun, adibide baten laguntzaz, objektu-identifikazioa gauzatzearren burutu beharreko urratsak.

IV.6.3 Adibidea Demagun erabiltzaileak "arrazoi" hitz-formak dituen aldaerak nahi dituela, eta, horiekin batera, aldaerak hitzaren zein adierari dauden lotuta, eta kategoria gramatikalak. Jo dezagun, galdera horri erantzunez, EH eta EDBL baliabideek honako emaitza-erlazio hauek itzuli dituztela, hurrenez hurren:

EH iturriak emaitzatzat bost aldaera desberdin itzuli ditu, eta EDBLk, berriz, hiru. Informazio teilakatua ere agertzen da, "arrazoin" aldaera bi iturrietatik jaso baitugu. Aldaera guztiak, gainera, hitzaren adiera bakar bati daude lotuta (EHko `EHad1` adiera eta EDBLko `EDBLad1` adiera). Bestalde, iturriek kategoria bakarra gordetzen dute "arrazoin" hitzarentzat, hots, izen kategoria (*iz.* EHn, *IZE* EDBLn).

[37] Emaitza atributuaren motak ez du zertan sarrerakoarena izan behar. Alde batetik, datuak bikoiztu egin behar badira, sarreraren mota balio bakuna den bitartean, irteera balio-zerrenda izango da. Bestalde, gerta daiteke sarreraren mota testu librea izatea, eta irteera, berriz, mota erabat desberdina; pentsa dezagun data eremuan, baliabide lokalean karaktere-segida bezala gorderik dagoena. Erregela batek data formatu normalizatu batera bihur dezake.

EH-kat	EH-adId	EH-aldForma
iz.	EHad1	arrazoin
iz.	EHad1	razoin
iz.	EHad1	arrazio
iz.	EHad1	arrazoa
iz.	EHad1	razoe

EDBL-kat	EDBL-adId	EDBL-aldForma
IZE	EDBLad1	errazoe
IZE	EDBLad1	arrazoin
IZE	EDBLad1	errazoi

IV.10 Taula: "arrazoi" hitzaren aldaerak, EH eta EDBLtik jasoak

Emaitza-taula hauek ikusita, nekez erabaki daiteke munduko zein entitate egon daitekeen errepikaturik. *q* galderaren emaitza-taula elkarketa baten emaitza da, eta, beraz, eredu orokorrean aurreikusirik ez dagoen erlazioa. Horrela, taula hauetan islatutako informazioa ELHISAk ulertzen duen eredura egokitu behar da, hots, EKOra. Informazioa berrosatu ondoren, emaitza partziala biltegi batean gorde behar da. *DB Lokala* izeneko datu-baseak —emaitza partzialen behin behineko biltokia— EKOko kontzeptu eta erlazioak gordeko ditu, eredu erlazionalari jarraiki sortutako tauletan zehar. Arestian aipatu dugun bezala, iturrietatik jasotako datuak *garbitu* egin behar dira DB Lokalean gorde aurretik, objektuen arteko konparazioa egin ahal izateko. Darabilgun adibidearekin jarraiki, IV.11 taulak DB Lokalak izango lukeen itxura ikus daiteke, datu garbiketari ekin ondoren.

Informazioaren berrosatzea burutu ondoren —EKOarekin bat etor dadin— errazago dugu entitate bera erreferentziatzen duen objektu errepikaturik dagoenetz jakitea. Horrela, garbi ikus daiteke *Kategoriak* taulako objektu guztiek entitate bakarra erreferentziatzen dutela ("Izena" kategoria), edo *Aldaerak* taulako *ald1* eta *ald7* kodeko elementuek hitz bera errepresentatzen dutela ("arrazoin" aldaera, alegia).

Sistemak, baliabideetatik jasotako erantzunak EKOaren arabera berrantolatu ondoren, objektu-identifikazioari ekingo dio. Horretarako, taula bakoitzeko elementuak ordenatu ondoren, errepikapenak bilatu eta ezabatzen saiatuko da.

Inplementatutako modulua ez bada ere, objektuak identifikatzeko *Datuen Araztailea* moduluak (Monge, 1997) lanean agertzen denaren antza izan

Estandarrak	Kodea	hitzForma
	s1	arrazoi
	s2	arrazoi
	s3	arrazoi
	s4	arrazoi
	s5	arrazoi
	s6	arrazoi
	s7	arrazoi
	s8	arrazoi

Kategoriak	kodea	katBalioa
	kat1	noun
	kat2	noun
	kat3	noun
	kat4	noun
	kat5	noun
	kat6	noun
	kat7	noun
	kat8	noun

Aldaerak	kodea	aldForma
	ald1	arrazoin
	ald2	razoin
	ald3	arrazio
	ald4	arrazoa
	ald5	razoe
	ald6	errazoe
	ald7	arrazoin
	ald8	errazoi

estandarrak	EEkod	ADkod
	ald1	K1
	ald2	K2
	ald3	K3
	ald4	K4
	ald5	K5
	ald6	K6
	ald7	K7
	ald8	K8

Adierak	kodea	Id	sarreraKodea	katKodea
	K1	EHad1	s1	kat1
	K2	EHad1	s2	kat2
	K3	EHad1	s3	kat3
	K4	EHad1	s4	kat4
	K5	EHad1	s5	kat5
	K6	EDBLad1	s6	kat6
	K7	EDBLad1	s7	kat7
	K8	EDBLad1	s8	kat8

IV.11 Taula: Erantzunak, objektuak identikatu baino lehen.

Estandarrak	Kodea	hitzForma
	s1	arrazoi

Kategoriak	kodea	katBalioa
	kat1	noun

Aldaerak	kodea	aldForma
	ald1	arrazoin
	ald2	razoin
	ald3	arrazio
	ald4	arrazoa
	ald5	razoe
	ald6	errazoe
	ald8	errazoi

estandarrak Dagozkio	EEkod	ADkod
	ald1	K1
	ald2	K1
	ald3	K1
	ald4	K1
	ald5	K1
	ald6	K6
	ald1	K6
	ald8	K6

Adierak	kodea	Id	sarreraKodea	katKodea
	K1	EHad1	s1	kat1
	K6	EDBLad1	s1	kat1

IV.12 Taula: Erantzunak, objektuak identikatu ondoren.

beharko lukeelakoan gaude. Gure sistemaren berezitasunak ezarritako zenbait finketa egin beharko dira, baina. Konparazio batera, elementuak berdinak diren jakiteko, atributu bakoitzaren domeinuaren araberako konparazio-eragilea erabiltzea behar-beharrezkoa dugu[38]. Izan ere EKOko kontzeptu baten instantziak konparatzeko —eta objektu berbera direnetz jakiteko— taula horretan konparaziorako beharrezkoak diren atributuak zeintzuk diren jakin behar da, nahitaez (Knoblock *et al.*, 2001). Har dezagun, adibide gisa, *Adierak* kontzeptuaren instantzia multzoa. Ikus daitekeenez, *kodea* atributua sistemak berak n-kote bakoitzari jarritako identifikatzaile unibokoa da, hots, instantzia orok kode desberdina izango du. Erraz kontura gaitezke, beraz, *kodea* atributua ez dela erabili behar instantzien arteko konparazioak egiterakoan. Horrela, ikusitako adibide xume honetan, K2 kodea duen *Adierak* taulako elementua ezabatu dugu, K3 adieraren berdina baita, nahiz eta bakoitzak bere kode propioa izan. Gauzak horrela, EKOko kontzeptu bakoitzak `atributuEsanguratsuak` izeneko erregela berezia du, zeinaren bitartez jakingo den klase horretako instantziak konparatzeko erabili behar den atributu multzoa.

IV.12 taulan, IV.6.3 adibideko erantzunaren adierazpena ikus daiteke,

[38]Monge-ren lanean (1997) konparazio-eragile gisa domeinuekiko independentea den karaktere-kateen distantzia neurtzen duen algoritmo bat erabiltzen dute.

ELHISAk gordetzen duen erara. N-koteen kopurua franko gutxitu da, espero zitekeen legez. Horretaz gain, objektuen arteko erlazioak adieraziak daude, objektu-identifikazioaren ondoren.

IV.7 ELHISAk gauzatutako integrazioari buruzko zenbait gogoeta.

ELHISAk informazio lexikalaren integrazioa du helburu. Ez da, baina, informazio lexikalaren estandarizazioari begira arituko, nahiz eta estandarizazio-ekimenek eragin handia izan duten gure sistema eraikitzeko garaian. Hala ere, guk oso kontuan izan dugu baliabide bakoitzaren independentzia, informazio lexikalaren integrazioa helburu, eta iturrien autonomia oso handia da ELHISAn. Horrela, gure sistemak ez ditu inondik inora jatorrizko baliabideak ordezkatuko. Hori baino, guk atzibide bateratua eskaini nahi diogu baliabide anitzetara jo nahi duen orori —dela giza-erabiltzailea, dela LNPrako aplikazioa.

Atal honetan, gure ELHISA sistema informazio lexikala integratzeko jarraitu dugun hurbilketa azalduko dugu, alderdi linguistiko batetik, eta sistemen onurak eta ahuleziak aztertu ditugu.

Baliabideen modelizazioen arteko heterogeneotasuna ebazteko, gure integrazio-hurbilpena EKOan oinarritzen da. EKOa osatzerakoan, oso kontuan izan dugu ereduaren irekitasuna eta orokortasuna —eta, ahal den neurrian, neutraltasuna ere—, eta abstrakzioa izan dugu erreferente nagusienetako bat. Horrela, EKOan gordetako informazioa hainbat geruzatan pilatzen da, zeintzuek informazio mota desberdinak kodetzen baitituzte. EKOaren muina kontzeptuek eta hitz-adierek errepresentatzen dituzten egituretan datza, eta egitura horiei hainbat mailatako informazioa —hitz-formaren maila, maila fonologikoa, morfosintaktikoa, sintaktikoa eta semantikoa— lotzen zaie, erlazioen bitartez. Horrela, maila desberdinetako informazioa hainbat geruzatan kodetzen da, eta erabiltzaileak erabaki dezake unitate lexikal baten zein informazio eskuratu.

Baliabide estandarren hainbat ekimenetan ikusi dugu datu lexikalen normalizazio-beharra, eta ELHISAk ere behar horixe du. Horrela, sistemak notazio estandar bat erabili beharko du, baliabide desberdinek informazio bera kodetzeko erabilitako konbentzioen arteko aldeak saihestu ahal izateko. Baliabide batetik informazioa sistemara heldu bezain pronto, eta baliabidea-

EH	EDBL	EuskalWordNet
zaio	P15	*> Something is —ing PP

IV.13 Taula: "otu" aditzaren azpikategorizaio-balioak

ren modelizazioan definituriko normalizazio-erregelei esker, ELHISAk datuen gainean normalizazio-prozesu bat egikarituko du, zeinaren helburua den zenbait eremu komunetan —kategoriak, azpikategoriak eta abar— gordetako informazioa notazio estandar batera bihurtzea. Notazio horretarako, PAROLE proiektuak proposatutakoak hartu ditugu (Ruimy *et al.*, 1998).

Bestalde, baina, baliabideek granularitate-maila desberdinak izan ohi dituzte informazioa errepresentatzeko: batena oso aberatsa den bitartean, fenomeno horren errepresentazio sakona egiteko aukera ematen duena, beste batena, berriz, laua da, fenomenoa azaletik soilik errepresentatzen baitu. Horrelako egoera batean, zaila da bi baliabideek duten informazio komuna erlazionatu ahal izatea.

Adibidez, IV.13 taulan "otu" aditzarentzat hiru baliabidek kodetutako informazioa ikus daiteke. Iturri bakoitzak notazio desberdinak erabiltzeaz gain, granularitate desberdinekin errepresentatzen dute informazioa. EHk aditzari atxiki dakiokeen laguntzaile mota baino ez digu adierazten. EDBLk, berriz, aditzaren erregimena adierazten duen patroi sintaktikoaren kodea eskaintzen digu, *P15* izena duena[39], eta adierazten duena "otu" aditzak bi argumentu behar dituela, baten kasua datibo delarik, eta bigarrena, berriz, konpletiboa. Azkenik, EuskalWorNet-ek informazio semantikoa ematen digu, alegia, "zerbait norbaiti —edo sintagma proposizional baten bidezko eraiketari— otzen zaiola" adieraziz. Nahiz eta hiru iturri horiek aditzaren azpikategorizazioari buruzko informazioa gorde, zailtasun ugari izango ditugu informazioa elkarrekin erkatzen, esaterako, EDBLren *P15* patroi-sintaktikoa EuroWordNetekoarekin bat datorrenetz jakiteko.

EKOaren osaeran zailtasun nagusi batekin egin dugu topo, hain zuzen ere. Izan ere, EKOak fenomeno linguistikoen modelizazio zorrotza, sakona eta konplexua adierazteko aukera ematen badu[40], gauza izango da errepresen-

[39]Egun, EDBLk ez du aditzen erregimenari buruzko informaziorik gordetzen. Hala ere, lan horri etorkizun hurbilean helduko zaio, eta auzi horretan (Aldezabal, 2004) lanaren ekarpenak izango dira oinarri nagusia. Bertan, motibazio semantiko batetik eratorritako patroi sintaktikoen zerrenda proposatzen da, patroiek aditzek adieraz ditzaketen predikatu motak sintaktikoki nola gauzatzen dituzten zehazten dutelarik.

[40]Esate baterako, EKOak GENELEX edo PAROLE estandarizazio-ekimenek proposa-

tazio aberatsak bere baitan gordetzeko, esaterako, gorago aipatutako patroi sintaktikoen errepresentazio zehatza gordetzeko.

Hurbilpen honi jarraitzeak, baina, baditu zenbait alde txar. Batetik, EKO oso konplexua diseinatzen bada, erabiltzaileak zailtasun handiak izan ditzake informazioa eskuratzeko garaian, eta ez du bide intuitiborik aurkituko bere informazio-eskariak —galderak— adierazteko. Izan ere, erabiltzailea EKOko kontzeptu zein erlazioetan oinarrituko baita bere galderak sistemari egiteko. Bestetik, baliabide lokal bakoitzeko informazioa EKOan mapatzerakoan, gerta daiteke informazioa oso era desberdinean kodetzea EKOan eta jatorrizko iturrian, nahiz eta informazio bera izan. Ikusi berri dugun adibidean, egitura konplexuena jarritzea erabakitzen badugu, EKOa patroi sintaktiko konplexuak adierazteko gai izango litzateke. Gerta daiteke, baina, baliabideren batek adibide-esaldien bidez errepresentatzea aditzen azpikategorizazio posibleak. Horrelakoetan, jatorrizko esaldiak patroi bihurtu beharko lirateke, aditzaren argumentuak eta kasuak azalaraziz. Hala ere, eztabaidagarria izan daiteke EKOan eta jatorrizko baliabidearen artean era arras desberdinak jarraitzea informazio bera kodetzeko (Cunningham *et al.*, 2000).

Beste muturreko hurbilpena jarraitzen badugu, hots, EKOaren egitura oso laua bada, baliabideko egitura konplexuak EKOko atributuekin elkartu beharko lirateke. Erlazio franko ezkutu egongo da horrela, eta ezin izango da, oro har, erlazio horiez jabetzea. EKOak fenomeno lexikalen behe mailatako errepresentazioa onartzen ez badu, sistemaren erabilera murriztu egingo da zeharo. Fenomeno lexikalen azaleko errepresentazioa egokia izan badaiteke giza-erabiltzailearentzat, laguntza eskasa eskainiko dio LNPrako aplikazioei, azken hauek informazio zehatza eta egituratua behar baitute.

EKOaren osaeran, auzi honetan geundelarik, erdibideko hurbilpena jarraitzea erabaki genuen. Horrela, EKOan islatuko diren fenomeno lexikalen xehetasun-maila ertainekoa da, hots, baliabideetan gordetako informazio usuena errepresentatzeko aukera ematen duena, baina, bestalde, informazio horren eskurapenerako galdera konplexuak eskatzen ez dituena. Azpikategorizazio-ereduak errepresentatzeko, adibidez, EKOak bi adierazpen mota — alegia, sakona eta azalerakoa— onartzen ditu. Horrela, EKOko `PatroiSint` kontzeptuak hainbat `Slot` izan baditzake ere, `adibidea` atributua ere badu, azpikategorizazioaren azaleko errepresentazioa onartzeko.

Datu-base federazioetan datu-baseen eskema lokala eta esportazio-eskemak bereizten diren bezalaxe, ELHISAn integratuko diren baliabideek ez du-

tzen duten ereduaren antzekoa.

te zertan bere informazio guztia —bere konplexutasun-maila guztiarekin— ELHISAn integratu. Hori baino, baliabideen modelizazioa eratzerakoan, oso kontuan hartu dugu zein informazio mota esportatu nahi den, eta, horrela, baliabideen formatu propio eta bereziak maiz utzi ditugu integrazio-prozesutik at. Adibidez, EDBL euskarazko datu-basea sisteman integratzerakoan, sorkuntza morfologikorako beharrezkoa den hainbat informazio —bi mailatako morfologiako teoria linguistikoari hertsiki lotuta dagoena— ez dugu ELHISAn islatu, informazio hori, hein handi batean, oso berezia baita.

Hartu dugun erdibideko hurbilpen hau bat dator integrazioko beste proiektu batzuekin, adibidez, (Cunningham *et al.*, 2000) lanean aurkezten denarekin. Auzi honetan murgilduta izanik, horrelaxe diote egileek:

> We propose to stop decomposing the object structure of resources at a fairly high level, leaving the developer to handle the original data structures [...]. Even at this stage we still expect substantial benefit from uniform access to higher level structures.

ELHISA sistema osoa informazio lexikal banatuen interfaze bateratua den heinean, baliabideetan gordetako zenbait informazio xehegia dela, edota teoria linguistiko edo aplikazio zehatzei lotuegia dagoela usteak badu zentzurik, gure uste apalean.

Hausnarketa hauek, berriro ere, EKOa garatzean bi hurbilpen —behetik gora eta goitik behera— jarraitzera eraman gaitu. Batetik, azterketa egin dugu integratu nahi diren baliabideen gainean —bertan gordetako informazioa islatu ahal izateko—, baina, bestalde, erabiltzailearen ikuspuntua oso kontuan hartu dugu, eta, nolabait, usuen eskatuko duen informazioaren errepresentazio egokia egiteko aukera eman nahi izan diogu. Laburbiltzeko, honako prozesu hauek burutuko ditu ELHISAk iturrietako erlazioen arteko datuak erkatu ahal izateko:

- Balioak normalizatu, hots, iturri ororen notazioak estandar batera ekarri. Horretarako, esan bezala, ELHISAk PAROLE proiektuan proposatutako notazioak erabiliko ditu kategoria/azpikategoria etab. gordetzeko. Iturrietako datuak notazio estandarrarekin bat etor daitezen, sistemak *datuen garbiketa* delako teknikak burutzen ditu iturrien gainean erantzunak ematerakoan.

- Iturri baten informazioa EKOak onartzen duena baino xeheagoa bada, iturri horretako kontzeptu edo erlazio franko EKOko kontzeptu edo

erlazio batekin soilik etorrarazi behar dira. EKOaren eta iturri desberdinen modelizazioen arteko mapaketa semantikoek aukera emango digute urrats hau betetzeko, mapaketak adierazteko egiteko erabili dugun espresibotasun aberatsa dela eta. Bestalde, gerta daiteke iturriak gordetzen duen informazio aberatsa ez egokitzea ELHISAri, konparazio batera, teoria linguistiko jakin bati estuegi lotuta dagoelako. Horietan, informazio horren modelizaziorik ez da egin, hots, iturriko BEKak ez du informazio hori islatzen.

- Iturri baten informazioa EKOkoa baino orokorragoa bada, ordea, iturrietako datuak EKOko hainbat kontzeptu eta erlaziotan banatu edo "despaketatu" behar dira. Horietan, iturriko BEKak informazioa EKOaren "antzera" —granularitate-maila berarekin, alegia— islatu behar du, nahiz eta iturriarekin bat ez etorri: *wrapper*-ak arduratuko dira iturrietako balioak kontzeptu eta erlazio berri horietan despaketatzen.

Azkenik, ELHISAren arkitekturak aukera eman behar du EKOa bera fintzeko, baliabide batek aurreikusi ez den kontzeptu edo erlaziorik gehitu nahi badu, existitzen den kontzeptu-hierarkia bat aberastu nahi bada, edo eginiko modelizazio baten gainean finketak egin nahi badira. LAV hurbilpena jarraitu denez, EKOaren beraren gainean aldaketak egitea posible da, eta, horrelako aldaketek dagoeneko adierazita dauden mapaketa semantikoak errebisatzea maiz eskatuko duten arren, ez gaituzte sistema osoa birdiseinatzera behartuko, ereduaren irekitasuna bermatuz.

IV.8 Integratu diren baliabideak.

Atal honetan, ELHISA sisteman integratu ditugun baliabideak azalduko ditugu. II.3 atalean[41] iturri lexikalak sailkatu ditugu, eta, exhaustiboak izateko inongo asmorik gabe, klase bakoitzeko adibide esanguratsu batzuk ikusi ditugu labur-labur. Edonola ere, ELHISAn mota guztietako iturri lexikalak integratzeko intentzioa azaldu dugu. Izan ere, ELHISAk bere baitan askotariko iturriak —heterogeneoak— integra ditzakeela baitugu tesi-lan honen hipotesi nagusiena; horrela, saiatu gara bai hiztegiak, bai datu-base zein ezagutza-base lexikalak gure sisteman integratzen.

IV.2.2 atalean ELHISAn integratu ditugun baliabideen BEKak ikusiko ditugu. Esan dugun bezala, hiztegi, datu-base lexikal eta ezagutza-base lexikal

[41] 67. orrian.

motako iturriak integratu nahi izan ditugu. Egin diezaiegun tarte txiki bat, iturri horiek —eta ez beste batzuk— zergatik integratu ditugun ELHISAn justifikatzearren, baliabide hauen ezaugarri esanguratsuenei.

- *Euskal Hiztegia* (EH) integratu nahi izan dugu euskarazko hiztegi konplexua eta aberatsa delako. Bestalde, hiztegia TEI gidalerroek hiztegiak kodetzeko proposamenaren arabera errepresentatuda dagoenez, gure ustea da TEIren arabera kodetutako beste hiztegiak ere sisteman integratzea ez dela lan zaila izango.

- *Euskarazko Datu-Base Lexikala* (EDBL) euskarazko datu-base lexikal aberatsa da, eta bertan gordetako sarrera kopurua oso handia da. LNPko aplikazioen hornitzaile lexikala izateko bokazioarekin eraikia delarik, oso interesgarria izan zaigu bertan gordetako informazio sistematiko eta formalizatua profitatu ahal izatea ELHISAren bidez.

- EDR datu-base zein ezagutza-basea erraldoia da, ingelesa eta japonieraren tratamendu automatikorako pentsatua —hortaz, EDR baliabide elebiduna da.

- *EuskalWordNet* (EWN) ezagutza-basea WordNet-en oinarrituta dago. WordNet-ek komunitate zientifikoan izan duen arrakasta itzela dela eta, oso komenigarria ikusi dugu gure ELHISAn integratzea. Halaber, EDRk bezala, ELHISAn alderdi eleanitza aztertzeko bidea eman digu baliabide honen integrazioak ere.

- *Hiztsua* (HSU) frantsesezko hiztegi baten definizioen analisi automatiko batetik sortutako ezagutza-basea da. Kontzeptuak eta beren arteko erlazioak sare semantiko baten antzera errepresentatzen ditu, hiztegi-definizioen egitura berezian oinarritutako eredu konplexu baten bidez.

ELHISAren bidez datu lexikalak eskuratu nahi dituenak, horrela, informazio aberatsa lortuko du. Ez bakarrik kuantitatiboa —iturri soil batera kontsultatuz lortuko lukeena baino handiagoa— baizik eta baita kualitatiboa ere. Batetik, izaera anitzetako iturri lexikalak integratzen dituenez —hiztegiak, datu-base lexikalak eta ezagutza-base lexikalak—, informazio mota desberdinak baina osagarriak eskuratu ahal izango ditu. Hitz bati buruz galdetzerakoan, hiztegietatik giza-erabiltzaileak ulertuko dituen definizio zein

	EH	EDBL	EDR	EWN	HSU
Hizkuntza	EU	EU	JP EN	EU EN ES	FR
Ortografia					
Hitz-forma	X	X	X	X	X
Aldaerak	X	X	X	X	–
Morfosintaxia					
Kategoria lexikala	X	X	X	X	X
Flexioa	–	X	X	–	–
Aditz-jokoa	–	X	X	X	–
Zenbakigarritasuna	–	X	X	–	–
Generoa	–	–	–	–	–
Morfologia					
Konposaketa/Eratorriak	–	X	–	–	–
Segmentazioa	–	–	X	–	–
Morfotaktika	–	X	–	–	–
Sintaxia					
Alternantzia	–	–	–	–	–
Osaketa (Complementation)	–	–	–	–	–
HAULak	–	X	X	–	–
Kolokazioak	X	X	X	X	–
Semantika					
Adierak	X	–	X	X	X
Sailkapen ontologikoa	–	–	X	X	X
Erlazio semantikoak	–	–	X	X	X
Definizioa	X	–	X	X	X
Domeinua	X	–	–	X	–
Esamoldeak	X	–	X	X	–
Murriztapen-hautapenak	–	–	X	X	–
Adibideak	X	–	X	X	X
Bestelakoak					
Erabilpen-oharrak	X	–	–	X	–
Maiztasunak	X	–	X	X	–

IV.14 Taula: ELHISAn integratutako baliabide lexikalak eta bertan aurki daitekeen informazioa.

erabilpen-adibideak jasoko ditu; datu-base lexikaletik, bestalde, hitzaren informazio xehatua jasoko du, seguruenik espezifikoa izango dena, maila linguistiko desberdinetan (morfologikoa, morfosintaktikoa, eta abar). Azkenik, ezagutza-base lexikalek hitz horren adierek beste kontzeptuekin dituzten erlazioak erakutsiko dituzte. Beste alde batetik, ELHISAn integratu ditugun baliabideek egun erabilpen zabala eta fidagarritasun handia dute, eta, beraz, eskuratutako informazioaren kalitatea ona izango da.

ELHISAn integratu ditugun baliabideak aurkezteko zenbait ezaugarri definituko ditugu, maila linguistiko desberdinetan, eta integratutako baliabideak informazio mota hori jasotzen duenetz nabarmenduko dugu. Era honetara, sistemak oro har estaliko duen informazio mota ikusi ahal izango da, ELHISAk informazio horren bildura aurkeztuko baitio erabiltzaileari. IV.14 taulan ikusten den bezala hainbat maila jasotzen dira ELHISAn: ortografikoa, morfologikoa, morfosintaktikoa, sintaktikoa edo semantikoa. Jakina, baliabide bakoitza ELHISAn erabat integratu arte —adibidez, bakoitzarako *wrapper* bat garatu arte— ezingo da ELHISAren bidez informazio hau guztia eskuratu. Beraz, taula hau ELHISAren erabiltzaileari eskainiko litzaizkiokeen informazio motak aztertzeko ekarri dugu hona.

IV.9 Datu-integrazioko beste sistema batzuekiko konparazioa.

Kapitulua bukatzeko, ELHISAk gauzatzen duen integrazioa beste zenbait integrazio-proiekturekin alderatuko dugu. Horretarako, III kapituluko III.6 atalean[42] ikusi ditugun integrazio-proiektuetan ipiniko dugu berriro ere arreta, baina proiektuen gorabeherei ez gara orain arituko, lan hori jadanik egina baitugu. Horren ordez, sistema hauen eta ELHISAren arteko aldeak eta desberdintasun nagusiak azpimarratu nahiko genituzke oraingoan.

ELHISAren diseinua proiektu hauetan oinarritu dela esan beharrik ez dago. Izan ere, orain ikusiko ditugun integrazio-proiektuak oso garrantzitsuak izan baitira datu-integrazioaren arloan, arazo nagusiak zehaztu eta identifikatu baitzituzten, terminologia berezia sortuz, eta beren ebazpena bideratzeko hainbat teknika garatu, hala nola, bitarteko zein *wrapper*-en erabilpena, galderen itzulpena edo datu-arazketaren prozesua. Horrela, proiektu hauen

[42] 136. orrian.

ekarpenei esker garatu ahal izan dugu gure integrazio-sistemaren proposamena.

Bestalde, gurea domeinu jakin bati buruzko integrazioa —informazio lexikala, alegia— burutzeko sortu da, eta, kapitulu honetan ikusi dugun bezala, informazioaren ezaugarriek finkatu dituzte sistemaren gorabeherak. Orain ikusiko ditugun integrazio-proiektuen helburuak, berriz, gureak baino harago doaz, edozein domeinutako informazioa integratzea baitute xede nagusia.

TSIMMIS

TSIMMIS proiektua integrazio-sistemen arteko estreinetarikoa dugu (Chawathe *et al.*, 1994). Proiektuaren helburua hainbat informazio-iturriren arteko heterogeneotasuna aztertzea zen, eta hauek ebazteko irtenbide bat proposatzea. Integrazio-lana giza-arduradunen menpe dago, alegia, sisteman informazioa integratzea ez da erabat automatikoa.

TSIMMIS sistemak ez du domeinuaren eredu orokorrik, eta integrazioa bitartekoek burutzen dute. Hori horrela izanik, erabiltzailearen esku geratzen da bitarteko bakoitzak eskaintzen duen informazio inplizituaren ulermena. Bitartekoak erregela deklaratiboen bidez adierazten dira, eta erregelak definitzeko GAV hurbilpena jarraitzen da.

Guk beste estrategia bat erabili dugu, gure integrazio-sistema EKOan oinarritzen baita. Horrela, ELHISAren erabiltzaileak EKOari buruz jakin behar du soilik, eta integrazioa lortzeko egin beharrekoak sistemak burutzen ditu. Bestalde, ELHISAk LAV hurbilpenari jarraitzen dio iturriak deskribatzeko, TSIMMISek ez bezala. Hala ere, TSIMMIS proiektuko hainbat ekarpen geureganatu ditugu, adibidez, sistemaren eta iturrietako *wrapper*-en arteko komunikazioa bermatzen duen OEM lengoaiaren erabilpena.

OBSERVER

OBSERVER sistemaren helburua hainbat informazio-iturri integratzean datza (Mena *et al.*, 2000). Sistema informazio-sistema orokorra da, eta ez dago integratu nahi den informazioa adierazten duen eredu orokorrik: sistemak ontologia anitz jarriko ditu harremanetan. Ontologia bakoitza domeinu jakin bateko espezifikazio formala da, zeinaren bidez adierazita egongo den domeinuari buruzko hainbat informazio-iturri. Integrazio-prozesua bideratzeko, OBSERVERek bi mapaketa-mota erabiltzen ditu, bata ontologiaren eta itu-

rrien artekoa, eta bestea ontologien artekoa. Mapaketak GAV hurbilpenari jarraiki adierazten dira.

Oso bestela, ELHISAk ontologia bakarraren hurbilpenari jarraitzen dio, hau da, integrazio guztia eredu komun eta orokor batean oinarritzen da. Izan ere, uste dugu informazio lexikalaren esparru zabaleko hainbat informazio komun eredu bakar baten arabera deskriba daitekeela, eta deskribapena lagungarria izango zaiola bertatik informazioa eskuratu nahi duenari.

ELHISA, OBSERVER bezala, CLASSIC deskribapen-logikaz baliatzen da sistemaren alderdi kontzeptualak errepresentatzeko. Hala ere, OBSERVERi galderak CLASSICez egiten zaizkio, eta gureari, aldiz, galdera konjuntiboez. Hau horrela izanik, OBSERVER arrazoibide-prozesuez balia daiteke galderen gainean hainbat dedukzio burutzeko. Horri esker, galdera findu daiteke: ontologia baten arabera adierazitako galderak erantzunik jasoko ez balu, galdera bera beste ontologia baten arabera berridazten da —eta berridazketa horretan informazioaren galerarik egon daitekeela aurreikusten da—, adibidez, galdetutako zenbait kontzeptu sinonimoak diren beste batzuekin ordezkatuz.

SIMS

SIMS sistema (Arens eta Knoblock, 1992; Arens *et al.*, 1996) LAV hurbilpena jarraitzen dutenen artean lehenengoetarikoa da. Halaber, aplikazioaren domeinu-eredua ezagutzaren errepresentazio-lengoaia aberats batez adierazita dago, eta, hortaz, sistemaren arkitekturarekiko independentea da. SIMS sistema aplikazio-domeinuarekiko independentea da, alegia, berak proposatutako integrazio-arkitekturak edozein domeinutako informazio-iturriak integratzeko aukera ematen du. Bere helburua aurrera atera ahal izateko, SIMSek aplikazioaren *domeinu-eredua* osatzen du, ezagutzaren errepresentazioko lengoaia baten laguntzaz.

ELHISAk hainbat ekarpen jaso du SIMSek emandakoetatik, hala nola, LAV eredua jarraitu izana, edo domeinua adierazteko eredu orokorra izatea. Hala ere, badago bi sistemen artean alde franko. Esaterako, iturriko eta domeinu-ereduko erlazioak adierazteko, nahikoa sinpleak diren estekez baliatzen da SIMS, soilik kontzeptu-kontzeptu edo erlazio-erlazio loturak egiteko aukera ematen dutenak. Ezin dira, hortaz, ELHISAk iturriak eta domeinu-eredua lotzeko dituen espresio konplexuak erabili.

Bestalde, integrazio birtuala gauzatzen du SIMSek, ELHISAk bezala; izan ere, erabiltzaileak ipinitako galderak ezarriko du bere erantzuna lortuko duen

plan logikoa, eta plana exekuzio-denboran kalkulatzen da. Galdera erantzuteko, horrela, SIMS planifikatzaile orokor batez baliatzen da. Plan optimoa zein den jakiteko, berriz, jatorrizko galderaren gainean birformulatze-eragileen aplikazioaren ordena asmatuko duen bilaketa-algoritmo bat erabiltzen du. ELHISAk, berriz, domeinuaren araberakoa den *MiniCon* algoritmoaz baliatzen da galderen itzulpena burutzeko.

Information Manifold

AT&T enpresan garatutako Information Manifold (IM) proiektuak (Kirk *et al.*, 1995; Levy *et al.*, 1996) datu-iturri anitzetan informazioa eskuratzeko interfaze komuna eskaintzen du. Bere arkitektura informazio-iturriak adierazteko aukera ematen duen domeinu-eredu aberats eta hierarkiko batean oinarriturik dago.

Informazio-iturriak IMn integratzeko, ohi den bezala, informazio-iturri bakoitzaren edukia adierazi behar da. Iturrien eduki-adierazpenak bi osagai nagusi ditu: iturriko kontzeptu zein erlazioen modelizazioa, eta iturriaren ereduaren eta domeinu-ereduaren arteko mapaketa semantikoa gauzatuko duen eduki-deskribapena. IMk informazio-iturrietako edukiaren gainean murriztapen konplexuak adierazteko aukera ematen du. Mapaketa semantikoez gain, IMk aukera ematen du iturrien galdeketa-gaitasunak adierazteko, hots, iturritik informazioa eskuratzeko erabili beharreko atributuak zeintzuk diren zehazteko. Bestalde, iturri batek informazio bati buruzko datu *osoak* dituela adieraztzen ahal da, eta IMk informazio horretaz baliatuko da galdera-itzulpenaren garaian. Esate baterako, iturri batek galdetutako informazio bati buruzko datu guztiak dituela adierazten bazaio, ez du beste iturrietara joko informazio horren bila.

IMren arkitekturatik ezaugarri franko erabili dugu ELHISA diseinatzeko. IM bezala, gu ere ezagutzaren errepresentazioko lengoaia batez baliatu gara domeinu-ereduaren pareko eredua garatzeko. ELHISAn erabilitako mapaketa semantikoek ere IMkoen tankera dute. ELHISAk, baina, ezin du iturrien gainean galdeketa-gaitasunik adierazi. Galdeketa-gaitasunak oso garrantzitsuak dira Interneteko orrietan dagoen informazioa integratzeko, eta uste dugu, helburu horretara jo behar badugu, oso kontuan hartu beharko ditugula etorkizunean.

Galderen itzulpena burutzeko *bucket* algoritmoaren hedapena den *MiniCon* algoritmoaz baliatzen da ELHISA, eta, IMn bezala, birformulaketa ez da planifikatzaile orokor batez gauzatzen. Bestalde, ELHISAn ezin da adiera-

zi iturri batek informazio mota bati buruzko datu guztiak dituela. Hala ere, sinetsiak gaude nekez aurkituko dugula horrelako baldintzarik betetzen duen iturri lexikalik. Hortaz, ez dugu uste, gurea bezalako domeinuan, iturrien *osotasuna* adieraztea hain beharrezkoa denik.

Infomaster

Infomaster (Genesereth *et al.*, 1997; Duschka eta Genesereth, 1997b) jadanik existitzen diren informazio-iturriak integratzeko sistema orokorra da. Datu-integrazioa burutzen du, eta mota anitzetako iturrietako informazioa integratzen saiatzen da.

Infomasterrek hiru mailako eredua jarraitzen du sisteman zehar behar diren kontzeptu zein erlazioak modelatzeko: interfaze-eredua, sistemaren erabiltzaileekin elkarrizketa bermatuko duena, oinarri-eredua, sistemaren muineko kontzeptu zein erlazioek osatua, eta iturrien ereduak, iturri bakoitzaren informazioa adierazten duten kontzeptu eta erlazioak. Mapaketa semantikoak iturrien ereduaren eta oinarri-ereduaren artean gauzatzen dira, eta desberdintasun-murriztapenak izan ditzaketen *datalog* erregela ez-errekurtsiboez baliatzen da, LAV hurbilpenari jarraiki. IM bezala, iturrietako informazioaren osotasunari buruzko informazioa gorde dezake, alegia, iturriek kontzeptu bati buruzko informazio guztia gordetzen dutela adieraz daiteke.

ELHISAk bezala, Infomaster sistemak malgutasun handia eskaintzen du iturriak gehitu, aldatu edo ezabatzeko. Hala ere, erabiltzaileari lagungarri suertatuko zaizkion erlazio edo kontzeptu berriak sortzeko malgutasun handiagoa eskaintzen du Infomasterrek ELHISAk baino. Izan ere, oinarri-ereduaren geruzak bien arteko zubilanak egingo ditu. Bestalde, iturrien edukiak deskribatzeko erabiltzen den formalismoak, *datalog* erregelek, alegia, espresibotasun aberatsagoa dute gure sistemak erabiltzen dituen galdera konjuntiboek baino. Galderen itzulpenaren prozesurako garatutako "Inverse Rules" algoritmoaren eraginkortasuna, baina, guk erabilitako *MiniCon* algoritmoarena baino okerragoa bide da, batez ere erregela askorekin lan egin behar badu (Pottinger eta Levy, 2000).

V. KAPITULUA

Galderen itzulpena eta erabilera-adibideak.

Atal honetan sistemaren funtzionalitatea aztertzeari ekingo diogu, eta ELHISAren jardueraren zenbait adibide aurkeztuko ditugu. Lehenik eta behin, sistemaren interfazean ipiniko dugu arreta, izan behar dituen ezaugarri nagusiak azpimarratuz. Ondoren, ELHISAk iturriak integratzeko beharrezkoa duen galderen itzulpena aztertuko dugu sakonki. Azkenik, zenbait erabilera-adibide aurkeztuko ditugu, sistema osoaren portaera ikustearren.

V.1 Interfazea.

ELHISAk bi erabiltzaile mota nagusi izango ditu, hots, LNPko aplikazioak eta giza-erabiltzaileak. Lehenengo motakoek jakin behar dute, ELHISArekin elkarrizketarik izan nahi badute, galderak zein eratan adierazi, eta, baita ere, sistemak itzulitako erantzunak nola ulertu. Horretarako, ELHISAren moduluek komunikatzeko formalismo bera erabiliko dute: aplikazioek galdera konjuntiboak igorriko dizkiote ELHISAri, eta honek, ostera, OEM lengoaian itzuliko dizkie erantzunak. ELHISA atzitu nahi duen aplikazio orok jakin beharko du, hartara, EKOaren osaera zein den, EKOak berak beteko baititu sistemaren API[1] delakoaren funtzioak.

Bigarren erabiltzaile motakoek, giza-erabiltzaileek, alegia, sistema atzitzeko bide intuitiboagoak behar dituzte galderak igorri eta erantzunak ikusteko.

[1] Application Programming Interface

Izan ere, ezin baitezakegu jo ELHISAren erabiltzaileak —aplikazio automatikoak ez bezala— sistemaren barne-antolaketa, datu-eredua edo kontsulta-lengoaia ezagutuko dituela. Horixe izango da, hain zuzen ere, interfazearen eginbeharra: sistemaren eta erabiltzailearen arteko elkarrekintza bermatzea, erabiltzaileari sistemaren xehetasunak estaliz.

ELHISA integrazio-sistema izanik, bere erabilpena ez da datu-base tradizionaletatik aldentzen. Izan ere, ELHISAri informazioa eskatzerakoan ez baitzaio zehaztu behar informazioa nondik eskuratu, baizik eta zein informazio nahi den. Hori horrela, ELHISA erabiliko duenak datu-base handi baten aurrean dagoela usteko du, datu-base bera birtuala bada ere. ELHISAren interfazeak, hortaz, ez luke DBKS batenaren arras desberdina zertan izan. Hala ere, ELHISAren balizko erabiltzailea ez da, seguru aski, konputagailuetan edo datu-baseetan aditua izango, eta, horrela, bere kontsultak era atseginean eta intuitiboan egin ahal izateak garrantzi berezia du.

Datu-baseak atzitzeko interfazeena, jakina, sakon ikertutako ikerlerroa dugu, datu-baseen hastapenetik ia. Gai konplexu eta sakona izanik, atal honetara zenbait zertzelada baizik ez ditugu ekarri nahi, besterik ez bada ere, ELHISAren interfazearen diseinu-ezaugarriak finkatzeko. Izan ere, interfaze-modulua ez baitugu inplementatu gure sisteman.

Galdeketarako sistema bisualak.

Datu-baseetatik informazioa erraz eskuratzearen arazoa sakon ikertu den heinean, SQL bezalako kontsulta-lengoaia tradizionalen alternatiba diren hainbat galdeketa-mota proposatu dira. Horietatik asko, galdeketarako lengoaia bisualak ("Visual Query Languages", VQL) izenekin ezagututakoak, bistaratze eta eragiketa grafikoen[2] erabilpenean oinarritzen dira erabiltzailearekiko elkarrekintza bideratzeko. VQL lengoaiak erabiltzen dituzten sistemei galdeketarako sistema bisualak ("Visual Query System", VQS) esaten zaie[3]. VQSak giza-ikusmenaren kanalak duen uhin-luzera zabalaz baliatzen dira, zeinaren bidez uler eta kudea dezakegun gizakiok informazio kopuru itze-

[2] Adibidez, datu-baseko entitateak adierazten dituzten objektu grafikoetan arrastatu, edota klikatu. Eragiketa grafikoak objektu grafikoen gaineko *manipulazio zuzena* ("direct manipulation") delakoan oinarritzen dira. Hainbat autorek abantaila ugari aurkitzen diote manipulazio grafiko zuzenari, hala nola, erabiltzailearen mundu-eredua eta konputagailuak proposatzen duenaren arteko distantzia murriztea, galdeketarako lengoaia zehatzak ikasi behar ez izatea, ikasteko erraztasuna eta abar.

[3] (Catarci *et al.*, 1997a) lanean VQSei buruzko azterketa ikus daiteke.

la, eta atzeraelikadura bisuala erabiltzeko aukera ematen dute, gizakien eta ordenagailuen arteko elkarrekintza areagotuz. Horrela aipatzen da (Catarci *et al.*, 1997b) lanean:

> VQSs may be defined as query systems essentially based on the use of visual representations to depict the domain of interest and express the related requests. VQSs provide user-friendly query interfaces for accessing a database.

VQSek, horrela, datu-basean gordetako informazioa eskuratzeko bide atsegin eta intuitiboak eskaintzen dituzte. Beren xede nagusia informazioa era garbian aurkeztea da, eta, hortaz, datu-basearen oinarrizko ezaugarrietan ipintzen dute arreta, beharrezkoak ez diren xehetasunetan sartu gabe. Datu-baseko ereduaren nolabaiteko bistaratze grafikoa eskainiz —diagrama edo grafoen bitartez, kasu—, erabiltzailea eragiketa grafikoez baliatuko da datubasea kontsultatzeko.

Oro har, honako ezaugarriak eskaintzen dituzte VQSek:

- Informazioa atzitzeko eskuzko nabigazioa.
- Eskemako kontzeptuen sinplifikazioa, adibidez, erabiltzailearentzat garrantzi gabekoak diren osagaiak baztertuz.
- Galderen formulatzea, osagai grafikoen gainean manipulazio zuzen sinpleak erabiliz.
- Erabiltzailearen atzeraelikadura.
- Zenbait informazio osagarri teklatutik sartzeko aukera ere eman ohi dute (adibidez, entitateko atributu baten balioa finkatzeko).

VQSak, hortaz, SQL bezalako lengoaien alternatiba oso egokia dira, batez ere, datu-basea atzituko duena aditua ez denean.

ELHISAren interfazea.

ELHISAren interfazearen azken xedea, esan dugu jadanik, sistemaren eta giza-erabiltzailearen arteko elkarrizketaz arduratzea da. Interfazearen bidez hiru eragiketa nagusi burutuko ditu erabiltzaileak, hots, galderak adierazi, galderen erantzunak aztertu, eta, nahi izanez gero, erantzunetako datuei buruzko xehetasunak eskatu. Interfazeaz arduratzen den moduluak, horrela,

erabiltzaileak lan horiek era atsegin eta intuitiboan egin ahal izatea bermatu behar du.

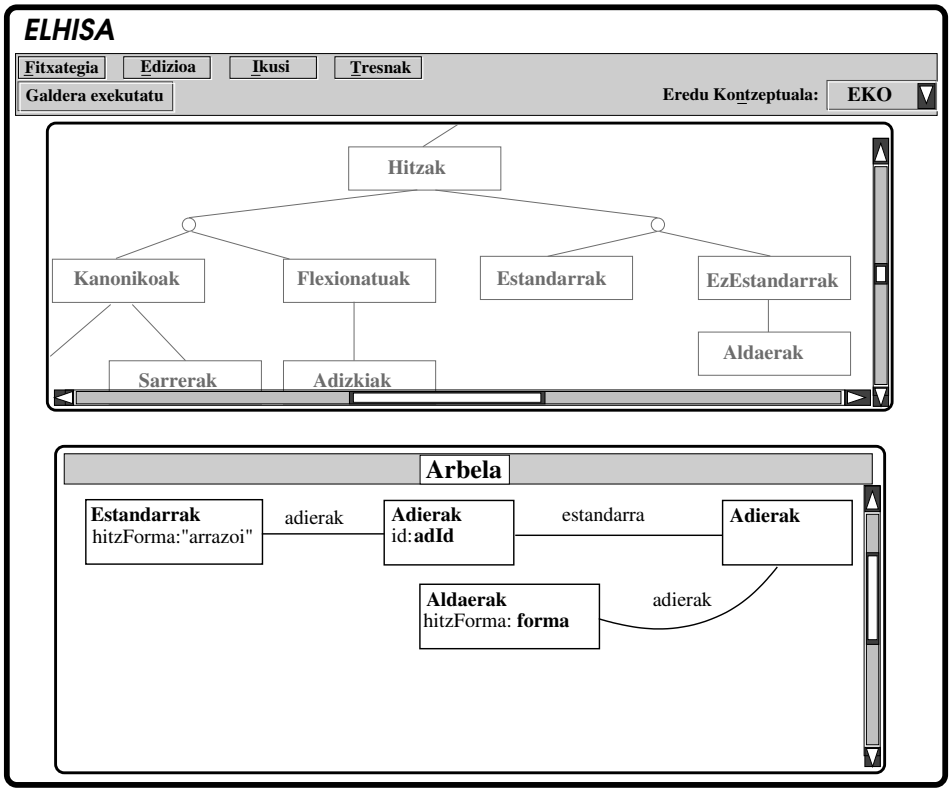
Erabiltzailearen interesa eskuratu gura duen informazio lexikalean datzalarik —eta ez sistemaren barne-antolaketan edo kontsulta-lengoaian— oso komenigarria da bere eskakizunak interfaze grafikoen bidez sistemara helaraztea. Aurreko azpiatalean ikusi dugun legez, galdeketarako sistema bisualek ez dute behartzen kontsulta-lengoaia zehatzen sintaxia eta semantika ulertzera datu-baseetako informazioa eskuratzeko garaian. Horrela, ELHISAren erabiltzaileari oso egokia suertatuko zaio VQS bezalako sistema bat, konputagailuekin edo datu-baseekin trebezia eskasa duten erabiltzaile ez-adituei suertatzen zaien bezala.

Jo dezagun ELHISAren erabiltzaileak "arrazoi" hitzaren definizioa nahi duela. Bere nahia sistemari helarazteko, jakina, EKOa izan behar du irizpide nagusi, bertako kontzeptu eta erlazioez baliatu beharko baita bere galdera adierazteko. Jarri berri dugun adibidean, zera adierazi beharko luke erabiltzaileak: `Definizioa` kontzeptuko instantzia baten `definizioTestua` atributua nahi duela, `Definizioa` kontzeptuaren instantzia `Adiera` kontzeptuko bati `def` erlazioaren bidez loturik dagoela, eta `Adiera` kontzeptuaren instantzia, bere aldetik, `Sarrerak` kontzeptuko batekin `adierak` erlazioaren bidez erlazionaturik dagoela, non `Sarrerak` kontzeptuko instantziaren `forma` atributuak, hain zuzen ere, "arrazoi" balioa duen.

EKOko kontzeptuei edo kontzeptuen atributuei buruz galdetuko du erabiltzaileak, halabeharrez, eta, beraz, interfazeak EKOari hertsiki loturik agertu behar du, bere kontzeptu eta erlazioei, alegia. Interfazeak EKOa diagramen bitartez adierazi behar du, eta, erabiltzaileak, eragiketa grafikoak lagun, diagrama horien osagaiak —kontzeptuak eta erlazioak islatuko dituztenak— maneiatuko ditu, sistemara doan galdera osatuz.

EKOaren izaera hierarkikoa ere kontuan izan behar da, kontzeptu baten erlazioei buruz galdetzerakoan, erlazioa kontzeptuarena edo kontzeptuaren arbaso batena izan baitaiteke. Kontuan hartu behar da, halaber, EKOa bera aldakorra izan daitekeela, eta, beraz, interfazeak aldaketa hauei aurre egiteko bezain moldagarria izan behar duela. Interfazearen lanaren emaitzak, berriz, EKOarekin bat datorren galdera konjuntiboa izan behar du, erabiltzailearen beharrak asetuko dituen galderaren adierazle.

Galderaren erantzunak ere interfazearen laguntzarekin ikusiko ditu erabiltzaileak. Interfazeak galderak egiteko eskaintzen dituen aukera grafikoekin bezala, erantzunaren datuak ere modu intuitibo batean eskaini behar zaizkio erabiltzaileari. Erantzuna EKOko objektu bilduma erlazionatua izanik, era-



V.1 Irudia: ELHISAren interfazea

biltzaileak objektu horietan gordetako informazioa nahiz objektuen arteko erlazioak erraz hauteman behar ditu. Horretaz gain, edozein daturen iturria zein den jakiteko aukera izan behar du. Horrela, ELHISAri azalpenak eska dakizkioke, eta informazioren bat nondik etorri den jakin.

Bestalde, oso garrantzitsua deritzogu interfazeak emaitzaren gainean nabigazio-aukerak eskaintzeari, hala nola, dagoeneko eskuratutako datu baten gainean —edo, datu hori abiapuntutzat hartuz—, galdera gehiago egitearen aukera. Era honetan, erabiltzaileak era inkrementalean egin diezazkioke galderak sistemari: hasiera batean, eskuratu nahi duen informazioari buruzko xehetasun guztiak ez dituelako edo, galdera sinple bat egiten du (adibidez, azpikategorizazio-eredu jakin bat jarraitzen duten formak eskuratzea). Galdera horren emaitzak ikusita, informazio osagarria eska dezake (aurreko adibidearekin jarraituz, jasotako hitz-forma baten definizioa edota ezaugarri sintaktiko gehiago).

V.1 irudian ELHISAren interfazeak izan beharko lukeen itxura ikus daiteke[4]. Leiho nagusian bi azpileiho txertatzen dira. Goiko aldekoan EKOko kontzeptuak agertzen dira, grafikoki, eta beren arteko erlazio taxonomikoak islatzen dira. *Arbela* izenarekin agertzen den beheko leihoa, berriz, erabiltzaileak galderak osatzeko erabiltzen da. Horrela, bada, galdera bat egiteko goiko leihoko laukiak —EKOko kontzeptuak— beheko leihora —arbelera— arrastatu behar ditu erabiltzaileak, eta, ondoren, arbelean dauden objektuen arteko erlazioak bete. Horretarako, arbelean ipini berri dituen kontzeptuen artean arkuak marraz ditzake —bi kontzeptuak erlazionatzea zilegi bada, jakina—, eta arkuari balio bat eman diezaioke, erlazioaren izenarekin. Kontzeptuen atributuak finkatzeko antzeko eran joka dezake: kontzeptuaren gaineko menu kontestuala lagun, kontzeptu horrek izan ditzakeen atributuen zerrenda —kontzeptuarenak zein kontzeptuaren arbasoenak— agertuko zaio. Erabiltzaileak zerrendatik atributu bat aukeratu eta bere balioa finkatuko du —galderaren murriztapen bat ezarriz—, edota atributuaren balioei buruz galdetu nahi duela adieraziko du.

Horretaz gain, interfazeak erantzunak jasotzeko beste leiho bat beharko du. Erabiltzaileak erantzuna osatzen duten datuak nondik datozen jakiteko aukera izan behar du, eta, jadanik aipatu dugun legez, datuen gainean galderak berrosatzeko aukerak eman.

V.2 Galderen itzulpena.

Aurreko kapituluan, ELHISAren arkitektura azaltzen ari ginelarik, bitartekoari buruzko zenbait gogoeta egin baditugu ere, atal honetan galderen itzulpen-prozesuan jarriko dugu berriro arreta. Galderen itzulpena gauzatzen duen *Galderen Itzultzailea* modulua aztertuko dugu, eta bereziki bere oinarria den *MiniCon* algoritmoa. Gure sistemaren berezitasuna dela eta, itzulpen-prozesua ganoraz bideratzeko algoritmoaren gainean harturiko zenbait erabaki propio azaltzea eta justifikatzea da, finean, atal honen xedea.

III.3.3.2 atalean[5] aurkeztu dugu *MiniCon* algoritmoa, bere formulazio orokorrean. Hala ere, atal honetan zehar ikusiko dugun legez, algoritmoak ez du bere lana taxuz burutuko, baldin eta bere gainean zenbait finketa egiten

[4] Jadanik aipatu dugun bezala, interfazea ez dugu oraindik inplementatu. Hori dela eta, irudian agertzen dena interfazeak izan beharko lukeen itxura erakusteko baino ez dugu hona ekarri.

[5] 110. orrian.

ez badira. Ikusiko dugu, horrela, zenbaitetan algoritmoak emaitza okerrak sor ditzakeela, iturriak modelatzeko erabili dugun datu-eredua dela medio.

Lehenik eta behin, galdera-itzulpena prozesuaren jarduera zuzena ikusiko dugu, eta horretarako adibide bati helduko diogu.

V.2.1 Adibidea Jo dezagun erabiltzaileak "arrazoi" hitz-forma estandarraren forma ez-estandar guztiak jaso nahi dituela, eta, horiekin batera, forma ez-estandarrek hitzaren zein adierari dauden lotuta[6]. Erabiltzaileak, interfazea lagun, V.1 irudian ikus daitekeen galdera sortuko du, honako galdera konjuntiboaren baliokidea dena:

$$
\begin{array}{lll}
q(\mathbf{forma}, \mathbf{adId}) & \text{:-} & Estandarrak(std), hitzForma(std,'' arrazoi''), \\
 & & adierak(std, stdAd), id(std, \mathbf{adId}), \\
 & & estandarra(ezStdAd, stdAd), \\
 & & adierak(ezStd, ezStdAd), \\
 & & ezEstandarrak(ezStd), hitzForma(ezStd, \mathbf{forma}).
\end{array}
$$

Erabiltzaileak galdera adierazi bezain pronto, EKOaren arabera adierazita dagoen galdera konjuntiboa galdera-itzultzaileari iristen zaio, eta honek, ELHISAn integratu diren baliabideetako BEKei eta BEDei esker, baliabide lokalen modelizazioekin bat datozen hainbat galdera konjuntibo sortzen ditu. Alegia, EKOaren arabera idatzitako galdera konjuntiboa iturrietako BEKekin bat datozen galderetara —berridazketak deiturikoak— itzuliko du. Horretarako, *MiniCon* algoritmoaz baliatzen da.

Eskuartean darabilgun adibidean, Galderen Itzultzaileak bi baliabide egoki aurkituko ditu jatorrizko galderarako (EDBL eta EH), eta baliabide bakoitzeko zenbait berridazketa osatzen ditu. Horien artean, berridazketa hauek aurki ditzakegu:

[6] IV.6.3 adibidearen antzera.

EDBL baliabidea:

$q(\mathbf{forma}, \mathbf{adId})$:- $EDBL\_sarrera(stdAd,''arrazoi'')$,
$EDBL\_homografoId(stdAd, \mathbf{adId})$,
$EDBL\_UnitateEstandarrak(stdAd)$,
$EDBL\_estandarraDagokio(ezStdAd, stdAd)$,
$EDBL\_UnitateEzEstandarrak(ezStdAd)$,
$EDBL\_sarrera(ezStdAd, \mathbf{forma})$.

EH baliabidea:

$q(\mathbf{forma}, \mathbf{adId})$:- $EH\_forma(std,''arrazoi'')$, $EH\_Estandarrak(std)$,
$EH\_adierak(std, stdAd)$, $EH\_adieraId(stdAd, \mathbf{adId})$,
$EH\_hobetsi(ezStd, stdAd)$, $EH\_forma(ezStd, \mathbf{forma})$
$EH\_EzEstandarrak(ezStd)$.

Nabarmendu behar da berridazketa hauek jatorrizko galderarekiko baliokideak direla. Iturrietako BEKekin alderatuz, egiazta daiteke biek bilatuko dituztela, dagozkien baliabideetan, "arrazoi" hitzaren forma ez-estandarrak.

□

V.2.1 *MiniCon* algoritmoari eginiko aldakuntzak

Galderen itzulpenak du, beharbada, garrantzi handiena galdera baten prozesamendu osoan, beraren bidez gauzatzen baita integraziorik konplexuena, hots, integrazio semantikoa deritzoguna. Eredu kontzeptual baten arabera adierazitako galdera beste ereduetara itzultzean, bi ereduen arteko baliokidetasuna —integrazioa— lortzeko bide handia egin da.

IV kapituluko IV.3.1 irudian[7] ikusi dugu galderen prozesatzailearen moduluak dituen osagaiak zein diren, hots, Galderen Aurreprozesatzailea, *MiniCon* algoritmoa eta Galderen Egiaztatzailea. Lehenengoak galdera berridatzi eta sinplifikatu egiten du, bigarrenak galdera sinplifikatua itzultzen du, *baliokideak* diren berridazketak sortuz, eta hirugarrenak, berriz, berridazketen artean zilegi ez diren galderak baztertzen ditu.

Azpiatal honen hasieran aipatu dugun bezala, *MiniCon* algoritmoari zenbait aldaketa egin dizkiogu, batik bat, ELHISAren berezitasunetara egokitu dadin. Izan ere, *MiniCon* algoritmoak, bere jatorrizko formulazioan, galdera konjuntiboekin eta entitate-erlazio (EE) ereduaren pean lan egiten du, eta, besteak beste, ez dago herentzia onartzen duen eredu batekin lan egiteko

[7] 190. orrian.

diseinatua. Hori horrela izanik, eta BEK desberdinak zein EKOa adierazteko erabiltzen dugun lengoaiaren espresibotasuna dela medio, entitate-erlazio hedatua (EEH) formalismoarena, alegia, ELHISA ezin izango da algoritmoaren irteeraz fio. Azter ditzagun, bada, *MiniCon* algoritmoaren gainean eginiko aldaketak, datu-ereduan egitura hierarkikoa onar dezan.

Lehenik eta behin, galderen aurreprozesatzaileak erabiltzaileak jarritako galdera analizatu eta berridazten du, *MiniCon* algoritmoaren sarrera egokia izan dadin. Prozesuak ziurtatzen du jatorrizko galderan agertutako klase oro izan daitekeen espezifikoena dela EKOaren arabera. Adibidez, galderak `Hitzak` klaseko elementu bati buruz galdetzen badu, zeina erlazionaturik dagoen `Estandarrak` klaseko beste hitz-forma bati `estandarra` erlazioaren bidez, orduan aurreprozesatzaileak galderan agertutako `Hitzak` klasea `EzEstandarrak` klasearekin ordezkatuko du.

Galdera berridatzi ondoren, *MiniCon* algoritmoak jasotzen du, eta galderaren itzulpenari ekiten dio. Batetik, galdera horrekin zerikusia duten BEDeko erregelak bilatuko ditu. Horretarako, eta jatorrizko galderaren azpihelburu bakoitzerako, BEDeko erregelen gorputzetan zehar begiratzen du, azpihelburu horrekin parekatzen den predikaturik ote dagoen jakiteko[8]. Parekatzen den predikaturik badago, predikatu hori bere baitan duen erregelako buruak badu zeresanik galderaren itzulpenean[9]. Edonola ere, parekatzea burutu ahal izateko, algoritmoak bi predikatu horiek disjuntuak ez izatea ezartzen du baldintza gisa. Guk baldintza hori murriztu egin dugu: galderako eta BEDeko predikatuak parekatzeko, galderako predikatuak BEDekoa subsumitu egin behar du, alegia, galderako predikatuak BEDekoa baino orokorragoa izan behar du. Izan ere, BEDeko predikatua galderakoa baino orokorragoa izango balitz, eta bien arteko parekatzerik burutuko balitz, sortutako berridazketak jatorrizko galderaren erantzunaren goi-multzoa lortuko bailuke.

V.2.2 Adibidea Izan bedi `EH_Hitzak` klasea, EH hiztegiko hitz guztiak bere baitan hartzen dituena (ikus IV.7 irudia[10]). Klase hori, bestalde, hiru azpiklasetan sailkatzen da, `EH_Sarrerak`, `EH_Aldaerak` eta `EH_EzEstandarrak`, hiztegiko sarrera diren hitz-formak, aldaerak eta forma ez-estandarrak errepresentatzen dituztenak, hurrenez hurren. EKOan ere hitz-formak adieraz-

[8]Galderako azpihelburuak zein BEDeko erregelen gorputzekoak EKOaren araberako predikatuak dira.

[9]Beti ere jatorrizko galdera eta erregelen artean *MCD*rik osa badaiteke. Berriro ere, jo beza irakurleak III.3.3.2 atalera *MiniCon* algoritmoaren xehetasunen bila.

[10]174. orrian.

teko klase anitz dago, hierarkia konplexua osatzen dutelarik, orokorrena den `Hitzak` klasetik `Siglak` edo `HAULak` klaseetaraino (ikus IV.3 irudia[11]). Edonola ere den, eman dezagun honako erregela hau idazten dugula `EH_Hitzak` klasea EKOko `Hitzak` klasearekin erlazionatzeko:

$$EH\_Hitzak(h) \leftarrow Hitzak(h).$$

Eta izan bedi honako galdera hau:

$$q(h) \text{ :- } Aldaerak(h).$$

Galdera-itzulpen prozesuan, *MiniCon* algoritmoak `EH_Hitzak` predikatua kontsideratuko du, `Aldaerak` eta `Hitzak` klaseak ez baitira disjuntuak (bata bestearen umea da, berez). Hortaz, honako berritzulpena sortuko litzateke:

$$q'(h) \text{ :- } EH\_Hitzak(h).$$

Hala ere, nabari daiteke berritzulpenak EH hiztegiko hitz-forma guztiak eskuratuko dituela, eta, horien artean aldaerak ere agertuko badira ere, erantzun multzoa jatorrian eskatutakoa baino orokorragoa da: erabiltzaileak, aldaerak ez diren, eta, beraz, berak eskatu ez dituen hainbat hitz-forma jasoko ditu. Itzulpenaren prozesuan predikatuak disjuntuak izatea baino, galderako predikatuak BEDeko bistakoa subsumitzea ezartzen bada, berridazketek ez dute jatorrizko galderak eskatutakoa baino informazio gehiago itzuliko.

□

Murriztapen honek —galderaren predikatuak bistakoa subsumitzearena, alegia—, baina, arazo orokorrago bat azalarazten du: BEKen eta EKOaren izaera hierarkikoa ez da BEDean islatzen. Egitura hierarkikoa jarraitzeagatik, BEKeko edo EKOko erlazioak zein atributuak hierarkia bateko kontzeptuen artean heredatzen dira. Herentzia, ordea, ez da BEDeko erregeletan islatzen, eta, hau horrela izanik, zenbait berridazketa ez dira taxuz osatuko.

V.2.3 Adibidea BEKeko egitura hierarkikoa BEDean ez islatzeagatik sortzen diren arazoak ulertzeko, EH hiztegian jarriko dugu, berriro ere, arreta. V.2.2 adibidean ikusi dugun bezala, hiztegiko hitz-formek hierarkia bat osatzen dute, eta hor ditugu, adibide gisa, `EH_Hitzak` eta `EH_Aldaerak` klaseak.

[11] 167. orrian.

Izan bitez, EHren BEDean, honako erregela hauek:

$$\begin{aligned} EH\_Hitzak(h) &\leftarrow Hitzak(h). && \text{(V.1)} \\ EH\_forma(h,f) &\leftarrow Hitzak(h), hitzForma(h,f) \mid \\ &\qquad EH\_Hitzak(h). && \text{(V.2)} \\ EH\_Aldaerak(a) &\leftarrow Aldaerak(a). && \text{(V.3)} \end{aligned}$$

eta demagun honako hau dela erabiltzaileak jarritako galdera:

$$q(f) \;\; :\!\!- \;\; Aldaerak(a), hitzForma(a,f).$$

Galdera honetan, erabiltzaileak `Aldaerak` klaseko elementuen `hitzForma` atributua eskuratu nahi du, hots, aldaera guztiak eskuratu nahi ditu. Hala ere, ez du EHrako berridazketarik lortuko, eta, hortaz, erantzunik ere ez. Kontua da, `hitzForma` erlazioa BEDeko V.2 erregelarekin soilik pareka daitekeela, baina, berridazketa guztiz eratzeko, algoritmoak galderako `Aldaerak` kontzeptua V.2 erregelako `Hitzak` kontzeptuarekin parekatu beharko lukeela, ezinbestean. Hala ere, parekatze hori ez da egingo, erregelako kontzeptua galderakoa baino orokorragoa delako. Egoera hau saihesteko, honako erregela hau eduki beharko genuke EH hiztegiko BEDean:

$$\begin{aligned} EH\_forma(h,f) &\leftarrow Aldaerak(h), hitzForma(h,f) \mid \\ &\qquad EH\_Aldaerak(h). \end{aligned}$$

Hemen arazoa hauxe da: BEDeko V.2 erregelak `EH_Hitzak` kontzeptuko `EH_forma` atributua definitzen du, EKOko kontzeptu eta erlazioen arabera. Bestalde, EHko BEKak dio `EH_Aldaerak` kontzeptua `EH_Hitzak` kontzeptuaren umea dela, eta, hortaz, gurasoaren atributu guztiak heredatuko dituela, horien artean `EH_forma` ere bai. Hala ere, ez dago, esplizituki behintzat, `EH_Aldaerak` kontzeptuko `EH_forma` atributua definitzen duen BEDeko erregelarik. Horrela, algoritmoak galdera guztiak erantzun ditzan, `EH_forma` atributua `EH_Hitzak` klaseko umeen artean *desfaktorizatu* behar da EHko BEDean, hots, herentziaren arabera hedatu behar da.

□

ELHISAn, BEDaren gainean *konpilazio-prozesu* bat burutu behar da, zeinaren bitartez BEDeko erregelak hedatu egiten diren, eta, BEKeko informazioaz baliatuz —*NeoClassic* sistemaren laguntzaz—, atributu zein erlazio

guztiak egon daitezkeen hierarkietan zehar barreiatzen diren, BEDak iturriaren izaera hierarkikoa isla dezan. Konpilazio-prozesua erdi-automatikoa eta iteratiboa da: konpiladoreak hasierako BEDeko erregelak hedatzen ditu, erlazioak desfaktorizatuz, eta emaitza sistemaren diseinatzaileari erakutsiko dio. Diseinatzaileak nahi dituen aldaketak egin ditzake —erregela berriak fintzeko, edo, nahi izanez gero, iturriaren BEKean erlazio berriak gehitzeko—, eta konpilazio-prozesua berriro abiatu. Prozesua errepikatu egingo da, diseinatzaileari iturriaren gainean ez direla aldaketa gehiago egin behar iruditzen zaion arte. BEDeko erregelak konpilatzeak *MiniCon* algoritmoan subsuntzio-eragilea aplikatzeko aukera ematen digu, predikatuen arteko parekatzeak burutzerakoan, eta, hortaz, berma dezakegu sortutako berridazketek ez dutela eskatutako informazioa baino gehiago berreskuratuko.

Behin jatorrizko galderaren gainean berridazketak lortu eta gero, berridazketen egokitasuna neurtu behar da, eta, batez ere, zilegiak direla ziurtatu. Berridazketa bakoitza iturri lokal zehatz baten BEKaren arabera egongo da adierazita, eta galderen itzulpen-prozesuaren azken urratsean berridazketek islatutako adierazpideak iturriko BEKarekin bat datozenetz egiaztatuko da. Izan ere, gerta baitaiteke *MiniCon*-ek emandako zenbait berridazketa zilegi ez izatea.

Horretarako, algoritmoaren berridazketa bakoitza *apaindu* egiten da, bere baitan dituen predikatu lokalen apaingarriak berridazketan txertatuz. Gero, berridazketa berriak BEK lokalarekin bat datozenetz egiaztatzen da, bat ez datozenak baztertuz. Urrats honetan, berriro ere, *NeoClassic* motorraz baliatzen da itzulpen-prozesua.

V.2.4 Adibidea Berriro ere, EH hiztegia hartuko dugu adibide honetarako. Hiztegi honetako sarreretan informazio semantikotzat hartzen dugun informazioa aurki daiteke, IV.7 irudian[12] ikus daitekeen legez: sarrera baten definizioa (`EH_Definizioak`), adibideak (`EH_Adibideak`), erabilerak (`EH_Erabilerak`); bestalde, sarrera baten estreinako agerpen-data ere adierazi dugu, `EH_Datak` kontzeptuan. Hona hemen EHren erabilerari zein agerpen-datari buruzko klaseei dagozkien BEDeko erregelak:

$$\begin{aligned} EH\_Erabilerak(e) &\leftarrow Erabilerak(e). \\ EH\_erabMota(e,m) &\leftarrow Erabilerak(e), erabMota(e,m) \mid \\ &\quad\quad EH\_Erabilerak(e). \end{aligned}$$

[12] 174. orrian.

$$EH\_Datak(d) \leftarrow Erabilera(d), erabMota(d,''data'').$$

Ikus daiteke `EH_Erabilerak` klasea bat datorrela EKOko `Erabilerak` klasearekin. Hiztegiko sarrerek duten estreinako agerpen-data gordetzeko, ordea, ez da, EKOan, klase berezirik sortu: EKOan hiztegiko sarreren agerpen-data erabilera mota bat da, zeinek `erabMota` atributu bezala `"data"` duen.

Gauzak horrela, eman dezagun erabiltzaileak honako galdera hau egiten duela:

$$q(a) \text{ :- } Erabilerak(a), erabMota(a,''data''), erabBalioa(a,''1800'').$$

MiniCon algoritmoak, besteak beste, honako berridazketak lortuko ditu EH hiztegirako:

$$\begin{aligned} q(a) &\text{ :- } EH\_data(d,''1800''), EH\_erabilerak(a,d).\\ q(a) &\text{ :- } EH\_data(d,''1800''), EH\_noizAurkitua(a,d).\\ &\ldots \end{aligned}$$

Lehenengo berridazketa, baina, ez dator bat EHko BEKarekin[13]. Berridazketari predikatuen apaingarriak txertatuz, horrela geratuko da:

$$\begin{aligned} q(a) \text{ :- } & EH\_Datak(d), EH\_data(d,''1800''),\\ & EH\_Adierak(a), EH\_erabilerak(a,d). \end{aligned}$$

`a` objektua `EH_Adierak` klaseko instantzia da, eta `d` objektuarekin erlazionatuta dago, `EH_erabilerak` erlazioaren bidez, `d` objektua `EH_Datak` klaseko instantzia izanik. Hala ere, EH-ko BEKaren arabera, `EH_erabilerak` erlazioaren heina `EH_Erabilerak` da, baina aurreko berridazketaren arabera, `d` objektua `EH_Erabilerak` eta `EH_Datak` klaseen instantzia da. Klase horiek, baina, disjuntuak dira EHko BEKean. Hortaz, berridazketa baztertu egin beharko da.

□

Berridazketak baztertzea, normalean, modelizazio zorrotza ez jarraitzea da, hots, kontzeptu-kontzeptu edo erlazio-erlazio mapaketa semantikoak baino aberatsagoak erabili izanagatik. Horrela, iturrietako kontzeptu batzuk ez datoz bat EKOko inongo kontzepturekin zuzenean, eta, horietan, EKOko klase baten azpimultzo bezala modelatu ohi dira. Horrelakoetan, *MiniCon*

[13]Ikus IV.7 taula, 174. orrian.

algoritmoak, EKOko galdera itzultzen saiatzen denean, berridazketa gehiegi itzuliko ditu, eta horien artean BEK lokalarekin inkonpatibleak direnak ere agertuko zaizkigu.

ELHISA sisteman *MiniCon* algoritmoa erabili ahal izateko, azken zailtasun bati egin beharko diogu aurre. Izan ere, algoritmoaren jatorrizko bertsioan iturri lokal bakoitza predikatu bakar baten bidez adieraz baitaiteke. Gure sisteman integratutako baliabideetan aurki daitezkeen klase eta erlazioak, baina, ezin dira, jakina, galdera konjuntibo bakar baten bidez adierazi. Iturri bat adierazteko galdera anitz behar direnean, iturri logiko anitz daudela suposatzen da, normalean, eta iturri logiko bakoitza galdera bakar baten bidez adierazten dela.

V.2.2 Galderen itzultzailearen zenbait datu.

MiniCon algoritmoaren gainean aipatu berri ditugun aldaketak burutu ondoren, galderen prozesatzaileak zuzen itzuliko ditu —ahal duenean— EKOaren arabera jarritako erabiltzailearen galderak. Itzulpen-lanaren ebaluazio samur bat izatearren, V.1 taulan hainbat galderaren gainean eginiko itzulpenei buruzko zenbait datu agertzen dira. Bertan, eta hainbat galderatan ebaluatuak[14], iturri lokal bakoitzeko sortutako berridazketaren tamaina[15] eta berridazketak sortzeko igarotako denborak ikus daitezke.

[14]Hauek dira galderak:

Q_1 Forma estandar baten forma ez-estandarrak eskuratu.

Q_2 Forma baten kategoria eskuratu.

Q_3 Forma baten definizioa eskuratu.

Q_4 Kategoria jakin bateko formak eskuratu.

Q_5 Forma baten sinonimoak eskuratu.

Q_6 Forma bati dagozkion adieren zein kontzeptuen hiperonimoak eskuratu.

Q_7 Forma bati dagozkion adieren zein kontzeptuen hiperonimoak eta beren definizioak eskuratu.

Q_8 Forma baten itzulpen posibleak.

Emaitzak C eranskinean ikus daitezke.

[15]Galderen prozesatzaileak galdera bakoitzeko lortutako berridazketa, galdera positiboa dena, galdera konjuntibo anitzez osatua dago. Berridazketaren tamaina, horrela, bera osatzen duten galdera konjuntiboen kopurua da.

	EDBL		**EWN**		**EDR**		**EH**		**HSU**	
Q_1	19	1.94	0	0.08	0	0.04	2	0.44	0	0.04
Q_2	1	0.37	0	0.1	1	0.21	1	1.1	1	0.07
Q_3	0	0.26	1	0.07	2	0.18	1	0.25	1	0.06
Q_4	2	0.07	0	0.01	1	0.02	1	0.04	2	0.02
Q_5	0	0.29	1	0.08	0	0.04	1	0.2	3	0.1
Q_6	0	0.58	2	8.94	1	0.19	0	0.16	1	2.35
Q_7	0	0.65	2	10.54	2	0.42	0	0.18	1	3.21
Q_8	0	0.99	1	0.35	0	0.13	0	0.25	0	0.07

V.1 Taula: Hainbat galdera: berridazketa kopurua eta igarotako denborak segundutan.

Algoritmoak itzulitako berridazketaren tamaina, —eta, ondoren igarotako denborak— zerikusi handia du iturrien mapaketa semantikoko erregelekin. Batetik, erregela-kopuru oso handia bada, itzulpen-prozesuan kontuan hartu beharreko informazioa areagotzen da nabarmen. Hauxe da, esaterako, *bucket* algoritmoaren akats larriena, alegia, erregela gutxiekin denbora gutxi behar duen arren, denbora hori areagotzen doala, esponentzialki, erregela kopurua handitzen den heinean. Konklusio berara eraman gaitu ELHISAren itzultzailearekin izandako eskarmentuak. Izan ere, hasiera batean itzulpen-lanak burutzeko *bucket* algoritmoa inplementatu bagenuen ere, itzulpenean igarotako denbora onartezina bihurtu baitzen neurri errealeko iturriak integratzen joan ginen ahala, iturri hauetako BEDen erregela kopurua handia zen eta. *MiniCon* algoritmoa, ordea, oso optimizatua dago, eta erregela kopuru handiekin ere arin burutzen ditu bere eginbeharrak.

Bestalde, modelizazioa zenbat eta zorrotzagoa izan, berridazketen tamaina orduan eta txikiagoa suertatuko da, hots, berridazketa finagoa izango da. Hau da, mapaketa semantikoak ez badira zehazki adierazten, eta iturriko kontzeptu zein erlazio bakoitza EKOarekin taxuz lotzen, algoritmoaren irteeraren tamaina nabarmen has daiteke, galdera erredundante ugari sortuz.

Edonola ere, oso emaitza txukunak dira V.1 taulan agertutakoak, gure ustez. Nabarmentzekoa da galdera guztietarako berritzulpen zuzenak lortzen dituela, eta nahiko arin egiten dituela berridazketak[16].

Galderaren arabera, zenbait baliabidetarako ez da inongo itzulpenik lor-

[16]Frogak 450Mhz AMD prozesatzailea duen PC makina batean eginak dira. Makinak 256Mb ditu memoria nagusi, eta Linux sistema eragilearen pean egiten du lan.

tzen. Esate baterako, Q_3 galdera, forma jakin baten definizioak eskatzen dituenak, ez du EDBLrako berritzulpenik lortzen, EDBLek ez baitu hitzen definiziorik gordetzen. Bai, ordea, gainerako iturriak, hots, EHek, EWNek, HSUek eta EDRek.

Bestalde, berridazketaren tamaina handia da zenbaitetan. Kasu okerrena Q_1 galderak ematen du, forma jakin baten forma ez-estandarrak eskatzerakoan, EDR baliabiderako sortutako berridazketak 19 galdera konjuntibo sortzen dituelarik. Optimizatzailearen lana da galdera konjuntibo horietatik erredundanteak baztertzea, eta galdera positibo txikiena igortzea planifikatzaileari, berridazteka dagokion baliabideari igor diezaion.

Denbora aldetik ere, emaitzak nahikoa onargarriak dira. Salbuespen bakarra, apika, EWN iturriarekin gertatzen da, forma bateko adierek duten hiperonimoen definizioak eskatzerakoan (q_1), 10 segundo inguru behar baititu galdera itzultzeko. Gainerakoetan, baina, itzulpenaren denbora txikia da, informazio-sistema baten erabiltzailearentzat onargarriak, alegia.

V.3 Erabilera-adibideak.

ELHISAren ezaugarri nagusiak eta funtzionalitatea aztertu ditugu neurri handi batean, eta, bereziki galderen itzulpenean sakondu badugu ere, ELHISA osatzen duten gainerako moduluen zeregina ere aurkeztu dugu. Zenbait erabilera-adibide ekarriko ditugu orain plazara, ELHISAren bitartez lor daitezkeen emaitzei gainbegiratu bat ematearren. Horretarako, erabiltzaileak jarritako galdera baten prozesua ikusiko dugu, alegia, jatorrizko galdera izatetik azkenengo erantzuna lortu arte burututako urratsak ikusiko ditugu.

Esan dezagun, berriro ere, integrazio-sistemen garapenak suposatzen duen lan itzela dela eta, ELHISAren zenbait modulu baino ez direla inplementatu, Galderen Itzultzailea batik bat, nahiz eta gainerakoak —optimizatzailea, planifikatzailea, interfazea, etab.— diseinatu diren, eta alderdi kontzeptualeko auziak landu. Horrela, bada, orain ikusiko ditugun adibideak sistemaren irudi bat jasotzeko baino ez dira. Modulu guztiak inplementatu eta integratu arte, nekez izan dezakegu sistema osoaren erabileraren lekukorik, ezta sistema taxuz ebaluatzeko bide zuzenik ere. Sinetsiak gaude, luze baino lehen horretara jo behar dugula, zalantzarik gabe.

1. Erabilera arrunta.

Sistemaren erabilera ikusteko, V.2.1 adibidearekin jarraituko dugu. Erabiltzaileak "arrazoi" hitzaren forma ez-estandarrei buruz galdetzen du, eta Galderen Itzultzaileak EH eta EDBL iturrietarako soilik sortzen ditu berridazketak:

EDBL baliabidea:

$$
\begin{array}{lll}
q(\textbf{forma}, \textbf{adId}) & :- & EDBL\_sarrera(stdAd, ''arrazoi''), \\
 & & EDBL\_homografoIf(stdAd, \textbf{adId}), \\
 & & EDBL\_UnitateEstandarrak(stdAd), \\
 & & EDBL\_estandarraDagokio(ezStdAd, stdAd), \\
 & & EDBL\_UnitateEzEstandarrak(ezStdAd), \\
 & & EDBL\_sarrera(ezStdAd, \textbf{forma}).
\end{array}
$$

EH baliabidea:

$$
\begin{array}{lll}
q(\textbf{forma}, \textbf{adId}) & :- & EH\_forma(std, ''arrazoi''), EH\_Estandarrak(std), \\
 & & EH\_adieraId(std, \textbf{adId}), EH\_hobetsi(ezStd, std), \\
 & & EH\_forma(ezStd, \textbf{forma}), EH\_EzEstandarrak(ezStd).
\end{array}
$$

Berridazketa bakoitza dagokion baliabidera iritsiko da, eta bertako *wrapper*-ak galdera logikoa itzuliko du, baliabideak ulertuko duen kontsulta-lengoaia berezira. EDBL iturria datu-base erlazionalaren pean dago gordeta, eta kontsultak SQL lengoaiaz egin behar dira. EH iturria, berriz, XML dokumentu sorta bat da, eta XQuery lengoaiaren bidez helarazi behar zaizkio galderak. V.2 taulan, bada, EDBL zein EH baliabideen gaineko *wrapper*-ek exekutatu beharko luketena ikus daiteke.

Wrapper-ek galderak exekutatuko dituzte, eta sortutako erantzunak ELHISAri igorriko dizkiote berriro. Hona hemen EH eta EDBL iturrietatik jasotako erantzunak[17]:

[17]ELHISAri erantzun-datuak OEM lengoaiaz iritsiko zaizkion arren, nahiago izan dugu formatu tabulatua erabiltzea gure azalpen hauetan, iturri bakoitzaren erantzunak argiago ikusten direlakoan.

EDBL baliabidea

```
SELECT   EDBL_UnitateEzEstandarrak.sarrera
         EDBL_UnitateEzEstandarrak.homografoId
FROM     EDBL_UnitateEstandarrak,
         EDBL_UnitateEzEstandarrak
WHERE    EDBL_UnitateEstandarrak.sarrera = "arrazoi"
and      EDBL_UnitateEzEstandarrak.estandarraDagokio =
         EDBL_UnitateEstandarrak.id
```

EH baliabidea

```
for      $iEntry in document("eh_A_letra.xml")//entry
where    $iEntry/orth = "arrazoi"
return   <ema>{ for $ad in $iEntry//sense
         return <obj> {
            $ad//form[@type="variant"]/orth/text()
            $ad/@n
            } </obj>
         }</ema>
```

V.2 Taula: EDBL eta EH iturrietan egingo diren galderak

EH-forma	EH-adieraId
arrazoin	EHad1
razoin	EHad1
arrazio	EHad1
arrazoa	EHad1
razoe	EHad1

EDBL-sarrera	EDBL-homId
errazoe	EDBLad1
arrazoin	EDBLad1
errazoi	EDBLad1

Sistemak iturrietatik datozen erantzunak jaso ondoren, behin-behineko datuak *DB lokala* izeneko datu-basean gordetzen ditu. Orduan, datu-arazketa prozesua burutuko du erantzunen gainean, baliabide desberdinetatik jasotako datuak *garbituz* —ikus IV.6.1 atala—, eta erantzunen artean munduko objektu bera erreferentziatzen duten objektuak identifikatuz —ikus IV.6.2 atala.

Ondoren erantzun bateratua izango du, iturrietatik heldu direnetatik sortua, eta erabiltzaileari itzuliko dio. V.3 taulan *DB lokalean* datu-basean gordeko dena ikus daiteke.

Estandarrak	Kodea	hitzForma
	s1	arrazoi

Adierak	kodea	id	sarreraKodea
	ad1	EHad1	s1
	ad2	EDBLad1	s1

EzEstandarrak	kodea	hitzForma
	ald1	arrazio
	ald2	arrazoa
	ald3	arrazoin
	ald4	errazoe
	ald5	errazoi
	ald6	razoe
	ald7	razoin

estandarra	EEkod	ADkod
	ald1	ad1
	ald2	ad1
	ald3	ad1
	ald3	ad2
	ald4	ad2
	ald5	ad2
	ald6	ad1
	ald7	ad1

V.3 Taula: "arrazoi" hitzaren forma ez-estandarrak.

2. Erabilera konposatua.

Aurreko adibidean erabiltzaileak, galdera bakar baten bidez, iturri anitzetik jaso du informazioa. Oraingoan, berriz, erabiltzaileak hainbat iturri kontsultatuko du, ELHISAren interfaze bateratuaren bidez. Horrela, erabiltzaileak bere galderak era inkrementalean egingo dizkio ELHISAri, alegia, galdera batekin hasi, eta, erantzunek sortzen dioten jakin-mina asetzeko edo, beste zenbait galdera egingo dituela suposatuko dugu.

Jo dezagun, hasteko, erabiltzaileak euskarazko determinatzaileen singularreko formak nahi dituela. Honako galdera hau jarriko luke:

$$\begin{aligned} q(\mathbf{forma}) \quad :\!\!- \quad & Sarrerak(sar), hitzForma(sar, \mathbf{forma}), \\ & adierak(sar, ad), gram(ad, gr), \\ & katBalioa(gr,'' determiner''), DetEzaugGram(gr), \\ & mugatasuna(gr,'' singular''). \end{aligned}$$

Galdera honek iturri batera baino ez du joko, hots, EDBL datu-base lexikalera. Izan ere, integratu ditugun gainerako iturriek ez baitute eskatutako informazioari buruzko noziorik. Hona hemen EDBL iturrirako sortutako berridazketa:

$$\begin{aligned} q(\mathbf{forma}) \quad :\!\!- \quad & EDBL\_sarrera(ad, \mathbf{forma}), \\ & EDBL\_Determinatzaileak(ad), \\ & EDBL\_numeroaMugatasuna(ad,'' S''). \end{aligned}$$

eta, emaitza honakoa da:

EDBL-sarrera
bat
bera
berbera
hau
hori
hura
berori

Sarrerak	kodea	hitzForma
	s1	bat
	s2	bera
	s3	berbera
	s4	hau
	s5	hori
	s6	hura
	s7	zero
	s8	berori

Adierak	kodea	sarreraKodea
	ad1	s1
	ad2	s2
	ad3	s3
	ad4	s4
	ad5	s5
	ad6	s6
	ad7	s7
	ad8	s8

V.4 Taula: Determinatzaile singularrak.

Demagun, orain, erabiltzaileak "hori" hitzaren definizioak nahi dituela, hitzak esanahi anitz dituelako susmoa egiaztatu nahi duelako edo. Hitzak hainbat definizio izan ditzakeela eta, hitzak izan ditzakeen adieren identifikatzaileak ere nahi ditu. Horrela adieraziko lioke galdera sistemari[18]:

$$q(\mathbf{d}, \mathbf{adId}) \text{ :- } Sarrerak(sar), hitzForma(sar,'' hori''), adierak(sar, ad), id(ad, \mathbf{adId}), def(ad, defk), definizioTestua(defk, \mathbf{d}).$$

Galdera honetarako, EH hiztegiak soilik du zeresanik. Honako berridazketa sortuko litzateke, bada:

$$q(\mathbf{d}, \mathbf{adId}) \text{ :- } EH\_Sarrerak(sar), EH\_forma(sar,'' hori''), EH\_adierak(sar, ad), EH\_adieraId(ad, \mathbf{adId}), EH\_definizioa(ad, defk), EH\_def(defk, \mathbf{d}).$$

Emaitza hau jasoko duelarik:

[18]Berriro ere, interfazearen laguntzarekin. Esan bezala, interfazeak aukera eman behar baitio erabiltzaileari erantzun-datuen gainean galderak birformulatzeko.

EH-id	EH-def
h1s1	Entzuten ari denaren inguruko pertsonei eta gauzei ezartzen zaien erakuslea.
h1s2	Bigarren pertsonaren indargarri gisa
h2s0	Sufrearen, limoiaren, urrearen kolorekoa, ostadarren bigarren kolorea.

Sarrerak

kodea	hitzForma
s5	hori

Adierak

kodea	id	sarreraKodea
ad5	h1s1	s5
ad9	h1s2	s5
ad10	h2s0	s5

Definizioa

kodea	testua	defAdierak
D1	Entzuten ari ...	ad5
D2	Bigarren pertsona ...	ad9
D3	Sufrearen, limoiaren ...	ad10

V.5 Taula: "hori"hitzaren definizioak.

Erabiltzailearen usteak berretsi dira, definizioetan arreta ipiniz ikus daitekeen bezala. Orain, demagun erabiltzaileak "hori" formaren itzulpenak nahi dituela, itzulpenaren hizkuntzarekin batera. Hona hemen galdera:

$$q(\mathbf{f}, \mathbf{hiz}) \text{ :- } Sarrerak(sar), hitzForma(sar,''hori''), adierak(sar, ad), hizkuntza(ad, "EU"), ordainakDitu(ad, itzKon), ordainakDitu(itzAd, itzKon), hizkuntza(itzAd, \mathbf{hiz}), adierak(itzSar, itzAd), hitzForma(itzSar, \mathbf{f}).$$

Integratu ditugun iturrien artean, *EuroWordNet*ek soilik gordetzen du hizkuntza anitzetako informazioa. Hortaz, galdera berari helaraziko zaio, honako berridazketa baten bidez:

$$q(\mathbf{f}, \mathbf{hiz}) \text{ :- } EWN\_form(ad, "hori"), EWN\_lang(ad, ``EU''), EWN\_toILI(ad, itzKon), EWN\_toILI(itzKon, itzAd), EWN\_lang(itzAd, \mathbf{hiz}), EWN\_forma(itzAd, \mathbf{f}).$$

eta emaitzak honako itxura hau du:

EWN-forma	EWN-lang
amarillo	ES
amarillo_limón	ES
gamboge	EN
groc	CA
groguenc	CA
hori	EU
horitasun	EU
lemmon	EN
llimona	CA
maize	EN
yellow	EN
yellowness	EN

Sarrerak	kodea	hitzForma
	s5	hori
	s9	yellow
	s10	yellowness
	s11	amarillo
	s12	groc
	s13	groguenc
	s14	horitasun
	s15	gamboge
	s16	maize
	s17	lemmon
	s18	amarillo_limón
	s19	llimona

Adierak	kodea	hizk	sarreraKodea
	ad11	EU	s5
	ad12	EN	s9
	ad13	EN	s10
	ad14	ES	s11
	ad15	CA	s12
	ad16	CA	s13
	ad17	EU	s14
	ad19	EN	s15
	ad20	EN	s16
	ad21	EN	s17
	ad22	ES	s18
	ad23	CA	s19

baliokideak	adKodea	kKod
	ad11	ad12
	ad11	ad13
	ad11	ad14
	ad11	ad15
	ad11	ad16
	ad11	ad17
	ad11	ad18
	ad11	ad19
	ad11	ad20
	ad11	ad21
	ad11	ad22
	ad11	ad23

V.6 Taula: "hori"hitzaren itzulpenak.

Erabiltzaileak, deteminatzaileei buruz galdetzen hasita, izen baten itzulpenak eskatzen bukatu du. Eskaerak egiteko, gainera, EKOak eskaintzen duen atzibide bateratua izan du, eta sistema galderak erantzuteko zeresanik duten baliabideetara baino ez da joan. Horrela, adibide honen bidez ELHISAz eskura daitekeen informazioa antzemateko aukera izan dugu.

VI. KAPITULUA

Baliabide berriak integratzen.

ELHISA sistemaren funtsezko ezaugarria bere hedgarritasunean dugu, eta hasiera-hasieratik diseinatu dugu bere baitan baliabide berriak onar ditzan. Sistemaren arkitektura ahalik eta malguen definitu dugu, eta, bereziki, ez dugu iturri lexiko berezietarako mugatu. Hala ere, iturri berri bat ELHISAn integratu nahi bada, hainbat buruhausteri egin behar zaio aurre. Kapitulu labur honetan, baliabideen integrazio-prozesua aztertuko dugu.

Sarrera kapituluko I.4 atalean ikusi dugun bezala, iturri berriak integratzeko arazoak hiru mailatan sailka ditzakegu: integrazio semantikoa, integrazio estrukturala eta datu-mailako integrazioa.

- Integrazio semantikoa gauzatu behar da baldin eta baliabideek kontzeptualizazio desberdina erabiltzen badute informazio komuna adierazteko. Horrela, iturrietako eredu kontzeptualen arteko izen-gatazkak —kontzeptu bera erreferentziatzeko izen desberdinak erabiltzea— domeinu-gatazkak —izen bereko kontzeptuen domeinuak desberdinak izatea— , zein moten arteko gatazkak —mota desberdinak erabiltzea kontzeptu bera adierazteko— integrazio semantikoan koka ditzakegu.
- Integrazio estrukturalak garrantzia du iturrien barne-antolaketak desberdinak direnean.
- Datu-mailako integrazioa da maila estentsioalean burutzen dena.

Atal honetan, arazo horiei guztiei buruz hitz egingo dugu, eta horretarako kasu praktiko batean oinarrituko gara. Iturri berri bat ELHISAn inte-

gratzeak hainbat urrats hartzen ditu: iturriaren modelizazioaren garapenetik datu fisikoen normalizazioraino, hainbat auzi eta buruhauste ebatzi behar ditu sistemaren diseinatzaileak.

ELHISAn iturri berri bat integratzeko bete behar diren urratsak azalduko ditugu, eta urrats bakoitzean egin beharrekoak argitu. Eskuartean izango dugun adibidea oso konplexua ez bada ere, ELHISAk integrazio-arazo desberdinei aurre egiteko eskaintzen dituen irtenbideei buruz aritzeko parada emango digu.

VI.1 Kasu praktikoa. Lematizazio datu-base baten integrazioa.

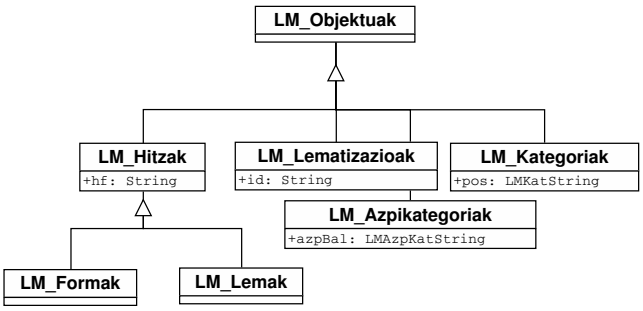
Egin dezagun kontu iturri lexikal berri bat ELHISAn integratu nahi dugula, eta iturria lematizazioak biltzen dituen datu-base bat dela. Datu-baseak —LM izenarekin ezagutuko duguna—, euskarazko hitz-formak gordeko ditu, eta, horiekin batera, formaren lemak (hitz-erroa), kategoriak eta azpikategoriak. Adibidez, LM datu-baseak "daramat" hitzerako bere lema gordeko du ("eraman"), eta, baita ere, bere kategoria ("aditza") eta azpikategoria (ADT, "aditz trinkoa"). Datu-base honen sorburua corpus bat lematizatzearen emaitza izan litzateke, esate baterako.

Gure nahia da LM datu-basea ELHISAn integratzea, bere baitan duen informazioaz baliatzeko, eta gainerako iturrietako informazioarekin erkatu ahal izateko. Integrazioa helburu, lau urrats beteko ditugu: iturria modelizatu, iturriaren eta ELHISAren —EKOaren— arteko mapaketa semantikoak adierazi, datu-mailako normalizazioa burutuko duen datu-arazketa zehaztu eta LM gainean *wrapper* bat garatu.

Iturria modelizatzen.

Diseinatzaileak egin behar duen lehenengo urratsa datu-basea bera modelizatzea da, hots, bere baitan gordetako informazioaren antolamendua formalizatzea, kontzeptu, erlazio eta atributuen bidez.

Esan bezala, LM iturriak hitz-forma eta lemak lotzen ditu, eta, horiekin batera, baita kategoria eta azpikategoria ere. Horrela, bada, lau kontzeptu nagusi izango ditu LM iturriak: `LM_Hitzak`, `LM_Lemak`, `LM_Kategoriak` eta `LM_Azpikategoriak`.



VI.1 Irudia: LM iturriaren modelizazioa (BEK).

VI.1 irudian LM iturriaren modelizazioa ikus daiteke, UML formalismoari jarraiki osatutakoa. Ikus daitekeenez, identifikatu berri ditugun kontzeptu oro agertzen da. Gainera, hitz-formen inguruan hierarkia bat osatu dugu. `LM_Hitzak` kontzeptua dago, hitz orokorrak errepresentatzen dituenak, eta bi ume dituenak: `LM_Lemak`, lemen formak gordeko dituen kontzeptua, eta `LM_Formak`, forma flexionatuak gordeko dituenak.

Horretaz gain, `LM_Lematizazioak` izeneko bosgarren kontzeptu bat ere agertzen da, lematizazioak adieraziko dituenak. LM iturriak hitz-forma anitz lotzen ditu lema bakar batekin. Konparazio batera, "daramat" eta "zeneraman" hitz-forma biek —eta beste askok— "eraman" lema dute. Bestalde, baina, hitz-forma batek ez du zertan lema bakarra izan, desanbiguatzen ez den bitartean. Adibidez, "batera" hitz-formak hiru lema desberdin jaso ditzake: "bat", "batera" edo "bateratu". Lema desberdinak jasotzen dituenean, halaber, kategoria eta azpikategoria desberdinak izango ditu hitz-formak:

Hitz-forma	Lema	Kategoria	Azpikategoria
batera	bat	DET	DZH
batera	batera	ADB	ALGARR
batera	bateratu	ADI	SIN

Informazio hori guztia `LM_Lematizazioak` izeneko kontzeptuaren pean gordeko dugu. Hartara, `LM_Lematizazioak` kontzeptuak lematizazioak adieraziko ditu, hau da, hitz-forma, lema, kategoria eta azpikategoria laukoteak.

Iturriko kontzeptuak azalarazi ondoren, beren arteko erlazioak finkatu behar ditugu. Erlazioek lematizazio bat bere osagaiekin jartzen dute harremanetan. LM baliabidean, lematizazioak lau osagaiz daude osatuta, hots,

Erlazioa	Domeinua	Heina	Parte-hartzea (dom : heina)
`lemakDitu`	`LM_Formak`	`LM_Lematizazioak`	`(1:n) : (1:1)`
`formakDitu`	`LM_Lemak`	`LM_Lematizazioak`	`(1:n) : (1:1)`
`kategoria`	`LM_Lematizazioak`	`LM_Kategoriak`	`(1:1) : (1:n)`
`azpikategoria`	`LM_Lematizazioak`	`LM_Azpikategoriak`	`(1:1) : (1:n)`

VI.1 Taula: LM iturriko erlazio lokalak

hitz-forma, lema, kategoria eta azpikategoria. `formakDitu` erlazioak lematizazioak eta hitz-forma orokorrak lotuko ditu. Era berean, `lemakDitu` erlazioak hitz-formak bere lemekin lotuko ditu, eta abar. VI.1 irudian LM iturrirako definitutako erlazioak ikus daitezke.

Modelazio honen bidez, hitz-forma jakin baten lema bat eskuratu nahi badugu, lortu behar dugu `LM_Lematizazioak` instantzia bat, zeinaren `formakDitu` erlazioa eskatutako hitz-formarekin bat datorren. Instantziaren `lemakDitu` erlazioak emango digu hitz-formaren lema.

Azkenik, LM iturriaren gainean zehaztutako modelizazioa ELHISAk ulertzen duen formalismoan adierazi behar dugu. VI.2 irudian LM iturriaren BEKaren definizioa ikus dezakegu, *NeoClassic*-ez. Iturrietako kontzeptu eta erlazioak *NeoClassic*-ez adierazterakoan, sistemak berak ezartzen dituen zenbait murriztapenetan jarri behar dugu arreta. Batez ere, honako auzi hauek ebatzi behar dira:

- **Erlazioen aritatea**. Iturri baten eredua modelizatzerakoan biko aritatea baino handiagoa duten erlazioekin topa gaitezkeen arren, *NeoClassic*-en errepresentazio-sistemak biko aritatea duten erlazioak baino ez ditu onartzen. *Reification* delako eragile ezagunaren bidez, baina, edozein aritatetako hainbat erlazio erlazio bitarren bidez adierazten ahal da, erlazioak kontzeptu berrien bidez adieraziz.

- **Alderantzizko erlazioak**. Alderantzizko erregelak oso baliagarriak izan daitezke iturri baten modelizazioan, iturria atzitzeko bide malguagoak eskaintzen baitizkio erabiltzaileari. *NeoClassic*-ek, ordea, ez du kontzeptuetako *rol*en alderantzizkorik errepresentatzeko gaitasunik. Hala ere, sistemak eskaintzen dituen erregelez balia gaitezke, eta, horrela, kontzeptu baten instantzia bat sortu bezain pronto, kontzeptuak dituen *rol*en alderantzizkoak automatikoki sortuko dituzten erregelak idatzi. Alderantzizko *rol*en informazioa, erregelen bidez sortua denez,

```
(createConcept LM_Objektuak ClassicThing true)
(createConcept LM_Kategoriak
       (and LM_Objektuak
            (all LM_pos LM_KatString)
            )
       true)
;; antzeko definizioa LM_Azpikategoriak kontzepturako
(createConcept LM_Lematizazioak
       (and LM_Objektuak
            (all LM_kategoria LM_Kategoriak)
            (atMost 1 LM_kategoria)
            (atLeast 1 LM_kategoria)
            ;; gainerako rol-ak
            )
       true)

(createDisjointGroup hitzDg) ;; Disjoint group:
                             ;; LM_Formak eta LM_Hitzak disjuntuak direla
                             ;; adierazteko.

(createConcept LM_Hitzak
       (and LM_Objektuak
            (all LM_hf String)
            )
       true)

(createConcept LM_Formak
       (and LM_Hitzak
            (all LM_lemakDitu LM_Lematizazioak)
            (atLeast 1 LM_kat))
       hitzDg)

(createConcept LM_Lemak
       (and LM_Hitzak
            (all LM_formakDitu LM_Lematizazioak)
            )
       hitzDg)
```

VI.2 Irudia: LM iturriaren BEKaren zati bat, NeoClassic-ez

ez da ustiatzen *NeoClassic*-ek automatikoki burututako sailkapenean, ezta subsuntzioa kalkulatzerakoan ere.

- **Zirkularitateak.** *NeoClassic*-en ezin da definizio zirkularrik adierazi. Badago, hala ere, *NeoClassic*-eko zirkularitateak ebazteko teknika ezagun bat[1], eta horixe bera erabil daiteke —eta erabili dugu— BEKak adierazi ahal izateko.

Mapaketa semantikoak.

Integrazio semantikoa gauzatzeko, iturriko modelizazioa bat etorrarazi behar da ELHISAk informazio lexikal orokorra errepresentatzeko duen ereduarekin, hots, EKOarekin. Mapaketa semantikoak BED erregelen bidez gauzatzen dira, IV kapituluan ikusi dugun legez.

Mapaketa semantikoen bidez, iturri lokaleko kontzeptu eta erlazioek esanahia hartzen dute EKOan. Mapaketa semantikoa gauzatzeko, iturriko kontzeptu eta erlazio ororentzat erregela bat idatzi behar da, kontzeptu zein erlazio lokalak EKOarekin duten harremana azalaraziz. Aurreko ataletan esan dugun bezala, mapaketak adierazteko LAV hurbilpena jarraitu behar da, hau da, iturriko kontzeptu zein erlazio oro EKOko kontzeptu eta erlazioen arabera adierazi behar da. Jo beza irakurleak IV.2.3 atalera[2] BEDeko erregelei buruzko xehetasunen bila.

Eskuartean darabilgun LM iturriaren adibidean, iturriaren modelizazioa burutu dugularik, BEKaren eta EKOaren arteko mapaketa semantikoak gauza ditzakegu. VI.3 irudian LMko BEKa EKOarekin harremanetan jartzen duten BED erregelak ikus daitezke.

Ikus daitekeenez, LMko kontzeptu eta erlazio orok du erregela bat atxikita. Zenbait erregela nahiko sinpleak dira. Adibidez, `LM_pos` atributuak ja-

[1] Demagun honako kontzeptuak definitu behar ditugula:

– PERTSONA

– UME: PERTSONA bat zeinaren gurasoak GURASOkoak diren.

– GURASO: gutxienez UMEkoa den ume bat dituzten PERTSONAk

Zirkularitatea apurtzeko, zera egin daiteke: GURASO kontzeptua PERTSONA bat da, gutxienez PERTSONAkoa den ume bat duena. UME kontzeptua PERTSONA bat da, zeinaren gurasoak GURASO kontzeptukoak diren. PERTSONA kontzeptuari erregela bat erantsi, zeinek adierazten duen PERTSONAko instantzia oro UMEkoa ere badela.

[2] 185. orrian.

$$
\begin{array}{lcl}
LM\_Hitzak(ff) & \leftarrow & Hitzak(ff). \\
LM\_hf(ff,f) & \leftarrow & Hitzak(ff), hitzForma(ff,f). \mid LM\_Hitzak. \\
LM\_Formak(ff) & \leftarrow & Flexionatuak(ff). \\
LM\_hf(ff,f) & \leftarrow & Flexionatuak(ff), \\
 & & hitzForma(ff,f). \mid LM\_Formak. \\
LM\_lemakDitu(ff,ad) & \leftarrow & Flexionatuak(ff), \\
 & & lemakDagozkio(ff,ad), \\
 & & Adierak(ad). \mid LM\_Formak. \\
LM\_lemakDitu(ff,ad) & \leftarrow & Flexionatuak(ff), \\
 & & adierak(ff,ad), Adierak(ad). \\
 & & \mid LM\_Formak. \\
LM\_Lemak(lm) & \leftarrow & Kanonikoak(lm). \\
LM\_hf(lm,f) & \leftarrow & Kanonikoak(lm), hitzForma(lm,f). \mid \\
 & & LM\_Lemak. \\
LM\_formakDitu(lm,ad) & \leftarrow & Kanonikoak(lm), \\
 & & lemaDa(lm,ad), Adierak(ad). \mid LM\_Lemak. \\
LM\_formakDitu(lm,ad) & \leftarrow & Kanonikoak(lm), \\
 & & adierak(lm,ad), Adierak(ad). \\
 & & \mid LM\_Lemak. \\
LM\_Lematizazioak(an) & \leftarrow & Adierak(an). \\
LM\_kategoria(ad,kt) & \leftarrow & Adierak(ad), kat(ad,kt), \\
 & & Kat(kt). \mid LM\_Lematizazioak. \\
LM\_azpikategoria(ad,azpkt) & \leftarrow & Adierak(ad), \\
 & & azpikat(ad,kt), \\
 & & AzpiKat(azpkt). \mid LM\_Lematizazioak. \\
LM\_Kategoriak(kt) & \leftarrow & Kat(kt). \\
LM\_pos(kt,pos) & \leftarrow & Kat(kt), katBalioa(kt,pos). \mid LM\_Kategoriak. \\
LM\_Azpikategoriak(azpkt) & \leftarrow & AzpiKat(azpkt). \\
LM\_azpBal(apkt,azp) & \leftarrow & AzpiKat(azpkt), \\
 & & azpikatBalioa(azpkt,azp). \mid LM\_Azpikategoriak.
\end{array}
$$

VI.3 Irudia: LM baliabidearen BEDa

sotakoa honako hau da:

$$LM\_pos(kt, pos) \leftarrow Kat(kt), katBalioa(kt, pos). \mid LM\_Kategoriak$$

Erregela honek dio, ezen baldin eta EKOan `Kat` kontzeptuko `kt` instantzia bat badago, zeinaren `katBalioa` atributuaren balioa `pos` den, orduan LMko BEKean `kt` elementua `LM_Kategoriak` kontzeptuko instantzia izango dela, eta bere `LM_pos` atributuaren balioa `pos` izango dela. Laburbilduz, erregelak dio LMko kategoria-balioak EKOko `Kat` kontzeptuko kategoriekin mapatu behar direla.

ELHISAk iturriaren informazioa ustiatu ahal izateko —azken finean, LM iturriak gordetako hitz-formen lemak edo erroak atzitzeko—, LMtik hitz baten lema nola eskura daitekeen adierazi behar da. Horrela dio `LM_lemakDitu` erlazioaren erregelak:

$$LM\_lemakDitu(ff, an) \leftarrow Flexionatuak(ff), lemakDagozkio(ff, an), Adierak(an). \mid LM\_Formak$$

Hau da, EKOan `Flexionatuak` kontzeptuko `ff` instantzia bat badago, zeina `lemakDagozkio` erlazioaren bitartez lotuta dagoen `Adierak` kontzeptuko `an` instantzia bati, orduan `ff` elementua, LMko BEKean, `LM_Hitzak` kontzeptuko instantzia izango dela, `an`, berriz, `LM_Adierak` kontzeptukoa[3], eta biak `LM_lemakDitu` erlazioaren bidez loturik egongo direla.

Iturriaren modelizazioa eta mapaketa semantikoak zehaztu ondoren, aurrerapauso handia egin dugu LMren integraziorantz. Izan ere, une honetan sistema gai baita iturriak gordetzen duen informazioa ulertu eta ustiatzeko. ELHISAren EKOa eta iturriaren BEKa elkartzean integrazio semantikoa lortu dugu.

Zenbaitetan, baina, arazo saihestezinekin topa gaitezke, eta gerta daiteke iturriko BEKa —edo BEKaren zati bat— ezin erlazionatu ahal izatea EKOarekin. Informazio lexikalaren domeinua hain heterogeneoa delarik, aise topa gaitezke EKOak aurreikusi ez duen kontzeptu —edo erlazio— batekin. EKOaren gainean aldaketak burutzea komenigarria ez bada ere —sinetsiak egon behar dugu EKOa aldatzeko zio sendo bat dagoela—, aldaketak sistema osoaren birdiseinu osoa ez du zertan erakarri behar, LAV eredua lagun:

[3] `an` instantzia `LM_Adierak` kontzeptukoa da, zeren `lemakDagozkio` *rol*aren heina `LM_Adierak` baita.

LM iturriak EKOan kontzeptu berri bat txertatzea ekarriko balu, gainerako iturrien BEDak ukitzeke utz daitezke.

LM iturriaren mapaketa semantikoa —BEDa— zehaztu delarik, galderen itzultzaileak aukera du informazioa ustiatzeko, eta ondoren, EKOaren arabera jarritako galderak LM iturriaren BEKera itzultzeko. Jo dezagun erabiltzaileak "dagokio" hitz-formaren lemak eskuratu nahi dituela, honako galdera konjuntiboa idatziz:

$$q(\mathbf{lforma}) \text{ :- } Flexionatuak(fh), hitzForma(fh, "dagokio"), lemaDagozkio(fh, ad), Adierak(ad), adierak(lh, ad), Sarrerak(lh), hitzForma(lh, \mathbf{lforma}).$$

Hona hemen galderen itzultzaileak lortutako berridazketa:

$$q(\mathbf{lforma}) \text{ :- } LM\_hf(fh, "dagokio"), LM\_Formak(fh), LM\_lemakDitu(fh, ad), LM\_lematizazioak(ad), LM\_Lemak(lh), LM\_formakDitu(lh, ad), LM\_hf(lh, \mathbf{lforma}).$$

Ikus daitekeenez, galderak ondo egingo du bere lana. Bestalde, lematizazioekin zerikusirik ez duten galderak itzultzeko beste ere bada. Demagun erabiltzaileak kategoria jakin bat duten hitz-forma guztiak nahi dituela, esaterako:

$$q(\mathbf{pos}, \mathbf{f}) \text{ :- } Kat(kt), katBalioa(kt, \mathbf{pos}) Adierak(ad), kat(ad, kt), Hitzak(h), adierak(h, ad), hitzForma(h, \mathbf{f}).$$

honako berridazketa sortuko da LM iturrirako:

$$q(\mathbf{pos}, \mathbf{f}) \text{ :- } LM\_hf(h, \mathbf{f}), LM\_Formak(h), LM\_lemakDitu(h, ad), LM\_Lematizazioak(ad), LM\_kategoria(ad, kt), LM\_Kategoriak(kt), LM\_pos(kt, \mathbf{pos}).$$

$$q(\mathbf{pos}, \mathbf{f}) \text{ :- } LM\_hf(h, \mathbf{f}), LM\_Lemak(h), LM\_formakDitu(h, ad), LM\_Lematizazioak(ad), LM\_kategoria(ad, kt), LM\_Kategoriak(kt), LM\_pos(kt, \mathbf{pos}).$$

Datu-arazketa.

Datu-mailako integrazioa ere gauzatu behar dugu, LMtik datozen erantzunak ELHISAk uler ditzan, eta beste iturrietatik etorritakoekin erlazionatu ahal izateko. IV kapituluko IV.6 atalean ikusi dugu ELHISAk datu-arazketarako duen estrategia: iturri batetik jasotako datuak *garbitu* behar dira lehen, eta, ondoren, datu *garbien* gainean objektu-identifikazioari ekin, munduko entitate bera erreferentziatzen duten objektuak identifikatu eta bateratzeko.

Objektu-identifikazio urratsa bateratua da ELHISAn, alegia, iturri berri bat integratzerakoan ez gara arduratu behar objektu-identifikazioaz: ELHISAren planifikatzaileak hartuko du lan hori. Hala ere, datu-garbiketa baliabide bakoitzak bideratu behar du, eta prozesua burutzeko iturriko informazio propioa behar du.

Datuen garbiketa-prozesuan hiru eginkizun betetzen dira, hots, atributuak erauztea —atributu batean balio atomiko bat baino gehiago agertzen direnean—, datuak egiaztatzea —akats-erroreak badituzte, kasu—, eta balioak normalizatzea —iturriek informazio lexikala gordetzeko erabiltzen dituzten notazio desberdinak bateratuz.

LM iturriko datu-garbiketa burutzeko, bertan gordetako informazioan jarri behar dugu arreta, eta hiru galdera hauei erantzun:

1. Atributuek balio ez-atomikoak gorde ditzakete? Edo, bestela esanda, gerta daiteke atributu batean balio bat baino gehiago egotea?

2. Atributuen balioak akastunak dira?

3. Normalizatu behar den informaziorik gordetzen da?

Galdera horietatik, hirugarrenak soilik jasoko du baiezkoa LM iturrian. Izan ere, LM iturriko atributu orok balio atomikoa adierazten baitu (hitzforma, lema, kategoria edo azpikategoria); bestalde, bertan gordetako informazioak akatsik ez duela suposatuko dugu. Izan ere, LM prozesu automatiko baten emaitza baita, eta akatsik gabekotzat har daiteke. Hala ere, zenbait kontzepturen balioak, hots, kategoria eta azpikategoriak, notazio propioa jarraituko dute.

LM gainean datu-garbiketari ekiteko, bada, kategoria eta azpikategoria atributuen balioak ELHISAk ulertzen duen notaziora itzuli behar dira. ELHISA, datu-arazketaren auziak ebazteko, domeinu abstraktuetan oinarritzen

da. Horiei esker, besteak beste, datu-arazketarako egikaritu behar diren prozesuak era deklaratiboan adieraz daitezke. Halaber, sistema osoan era trinkoan txerta daitezke domeinu abstraktuak, *NeoClassic* sistemak eskaintzen duen funtzionalitateari esker.

LM iturria modelizatzerakoan bi domeinu abstraktu definitu ditugu, hots, `LM_KatString` eta `LM_AzpString`, kategoria eta azpikategoria balioak gordeko dituztenak, hurrenez hurren. Balio horiek normalizatzeko, *normalizazio-erregela* bat atxiki behar zaio kontzeptu bakoitzari. Adibidez, `LM_KatString` kontzeptuari honako erregela erantsi behar zaio:

```
(createRule LMKatNorm
            LM_KatString
            (computedFillers mapaketaEDF LM_balNorm LM_bal KatLMHash))
```

Erregela honek `mapaketaEDF(obj,KatLMHash)` funtzioari deituko dio `LM_KatString` kontzeptuko `obj` instantzia bat gehitzen den bakoitzean. Funtzioaren emaitza `LM_balNorm` izeneko *rol*ean utziko du. `mapaketaEDF` funtzioa dagoeneko idatzia dagoen funtzio bat da. *Hash* bat jaso, eta funtzioak `obj` instantziak `bal` atributuan duen balioa *hash*aren arabera mapatuko du, eta emaitza `balNorm` *rol*ean txertatuko du. Hortaz, LM iturriko kategorien balioak PAROLEk onartzen dituenetarako mapaketa adierazten duen *hash* bat definitzearekin aski da. VI.2 taulan kategoria eremua normalizatzeko erabilitako *hash* bat ikus daiteke.

LM		ELHISA
ADB	→	Adverb
ADI	→	Verb
ADJ	→	Adjective
DET	→	Determiner
IOR	→	Pronoun
IZE	→	Noun
ITJ	→	Interjection

VI.2 Taula: LM eta ELHISAren arteko hash taula bat. Kategoriak.

`LM_AzpString` kontzepturako antzeko erregela idatzi ondoren, LM iturriko datuak ELHISAk ulertuko ditu, eta, hortaz, datu-mailako integrazioa lortua dugu.

Wrapper-a.

Iturri baten integrazioaren azkeneko urratsa iturria bera atzituko duen *wrapper* baten garapenean datza. *Wrapper*ak LM iturri lokalaren eta ELHISAren arteko komunikazioa bermatuko du. LMren BEKarekin bat datorren galdera konjuntiboa jaso, eta iturriak ulertzen duen kontsulta-lengoaiara itzuliko du, bertako datuak eskuratzeko. Galdera iturri lokalera igorri ondoren, emaitzak jaso, OEM formatura bihurtu, eta ELHISAri itzuliko dizkio.

LM iturriak datuak gordetzeko duen antolamendua dela eta, *wrapper* berezi bat garatu beharko da informazioa berreskuratzeko. Izan ere, iturriaren barne-antolamenduak zeharo baldintzatzen du *wrapper*aren diseinua eta garapena.

LM datu-base erlazionala izango balitz, *wrapper*ak jasotako galdera SQL lengoaiara itzuli beharko luke. Galdera konjuntiboak SQLra bihurtzea zuzena den neurrian, *wrapper*aren lana errazten da nabarmen. Adibidez, aurrean ikusi dugun "dagokion" hitz-formaren lema eskatzen duen honako galderarako:

$$q(\mathbf{lforma}) \text{ :- } LM\_hf(fh, \text{"}dagokio\text{"}), LM\_Formak(fh), LM\_lemakDitu(fh, ad), LM\_lematizazioak(ad), LM\_Lemak(lh), LM\_formakDitu(lh, ad), LM\_hf(lh, \mathbf{lforma}).$$

*wrapper*ak honako SQL galdera lortu beharko luke:

```
SELECT   LM_Lemak.hf
FROM     LM_Lemak, LM_Hitzak
         LM_Lematizazioak
WHERE    LM_Hitzak.hf = "dagokio"
and      LM_Hitzak.lemakDitu = LM_Lematizazioak.id
and      LM_Lemak.formakDitu = LM_Lematizazioak.id
```

LM iturria, hala ere, hamaika era desberdinez egon daiteke egituratua. Are gehiago, LMk ez du zertan datu-base estatikoa izan. Jo dezagun, esaterako, LM iturria komando-lerroko aplikazio bat dela, *lemati* izenekoa. Aplikazioak honako exekuzio-aukerak izan ditzake:

```
lemati [-f hitz-forma] [-l lema] [-kat kategoria] [-azp azpikat]
```

non:

- `-f` forma bat emanda bere lema(k) eta kategoria(k) itzultzen ditu.
- `-l` lema bat emanda, bere formak eta kategoriak itzultzen ditu.
- `-kat` soilik kategoria bateko formak/lemak itzuli.
- `-azp` soilik azpikategoria bateko formak/lemak itzuli.

Programaren irteera, bestalde, laukoteak izango dira (forma, lema, kategoria, azpikategoria). Gauzak horrela, ikusi-berri dugun galdera konjuntiboaren itzulpenak beste formulazio arras desberdina izango luke, hots,

```
lemati -f dagokio
```

Dena dela, ELHISAk ez du zertan jakin LM iturriaren barne-antolamendua nolakoa den. Dela LM datu-base erlazionala, dela aplikazio berezitu bat, atxikita duen *wrapper*ak egingo ditu LMren eta ELHISAren arteko bitartekari lanak.

VII. KAPITULUA

Ondorioak eta aurrera begirakoak.

Kapitulu honetan, egindako lanari buruzko gogoetak bilduko ditugu. Deskribapenari eta azalpenari baino gehiago, hausnarketari egin nahi diogu tartea orain. Ikuspuntu kritiko batetik, ELHISAk zer ekarri duen, eta zer utzi duen egiteke aztertuko dugu.

VII.1 Ondorioak.

Lan honetan ezagutza lexikalaren integrazio-sistema bat diseinatu eta garatu da. Sistemak informazio lexikala eskuratzen laguntzen du, eta zeregin horretan atzibide bateratua eskaintzen dio erabiltzaileari.

Ingenieritza linguistikoaren arloan kokatu behar da hemen aurkeztutako lana. Informazio lexikala integratzearen arazoaren irtenbide bat proposatu dugularik, informazioaren integrazioa eta datu-base lexikalen arloak bildu ditugu auzi horretan, eta, gure ustez, oso aberasgarria suertatu da bi arloon elkarrenganatzea.

Informazio-integrazioaren arloak iturri heterogeneoen artean informazioa trukatzeak sortzen dituen arazoak ikertu ditu sakon, eta ikerlerroak eman ditu bere fruituak, terminologia eta teknika propioak ezarriz. Gure integrazio-proposamena informazioaren integrazioko teknikez baliatzen da esparru jakin bateko informazioa —alegia, hitzei eta hitz-adierei buruzkoa den informazio lexikala— integratzeko.

Lexikoietan —eta iturri lexikaletan, oro har— gordetako informazioa he-

terogeneoa da, izaeraz, eta, informazio lexikala bateratzearen xedearekin hainbat ekimen eta saiakera egon diren —eta dauden— arren, emaitzak ez dira nahi bezain arrakastatsuak izan. Beste ikuspuntu bat jarraitu dugu guk iturri anitzetako informazio lexikalaz baliatzeko: baliabide lexikaletan zehar gordetako informazioa formalismo estandar batera bihurtu ordez, baliabideak "federazio" moduko batean integratzea da guk proposatutakoa.

Gure hurbilpenean oso kontuan izan dugu iturrien berezitasuna eta independentzia. ELHISAn integratutako iturri lexikalen autonomia-maila erabatekoa da, eta, hartara, datuak gehitzeko, aldatzeko edo ezabatzeko askatasun osoa dute. Iturriek, bestalde, ez dute beren errepresentazio-formalismo propioa zertan aldatu ELHISAren hornitzaile lexikalak izateko.

Informazio lexikala adierazteko eredu kontzeptual orokorra (EKO) diseinatu dugu. EKOa informazio lexikala adierazteko eredu komuna eta irekia da. Sistema diseinatzeko LAV hurbilpenari jarraitu izanaren ondorioz, EKOak *bitarteko eskemaren* papera jokatzen du ELHISAn. EKOaren garapenean arreta berezia jarri diogu informazioaren independentzia eta berrerabilgarritasuna bermatzeari, eta, auzi horretan, informazio lexikalaren estandarizazio- eta integrazio-ekimenek izan dituzten emaitzak geureganatu ditugu. Hitz-formen, adieren zein kontzeptuen, eta ezaugarrien arteko bereizketa garbiari esker, EKOan errepresentatutako informazioa hainbat dimentsio zein ikuspuntutatik ikus daiteke. Objektuei zuzendutako paradigma bat jarraitu dugunez, bestalde, informazio lexikala era txukun eta garbian modelizatzeko aukera izan dugu, informazio-moten arteko orokortasunak antzemanez. Ereduaren hedagarritasunak ere garrantzi berezia izan du guretzat, eta sinetsiak gaude EKOaren diseinu garbiak izugarri lagunduko duela bertan aurreikusi ez dugun informazio-mota berria gehitzeko garaian.

EKOa garatzeko bi norabideko metodologia jarraitu dugu. Batetik, IXA taldean egunero erabiltzen ditugun iturri lexikalak aztertu ditugu, eta elkarren arteko informazio komuna erauzi. Horretaz gain, beste hainbat iturri lexikal ere aztertu ditugu. Bestetik, baina, oso aintzat hartu dugu EKOaren bidez informazioa eskatu eta eskuratuko duena, alegia, ELHISAren erabiltzailea. IXA taldeak badu eskarmenturik iturri lexikal anitzetatik informazioa eskuratzen, eta kontsulta usuenak edota interesgarrienak gure sistemarekin egin ahal izateak garrantzi berezia izan du.

Iturriek informazioa granularitate desberdinez errepresentatzen dutelarik, konpromiso batera iritsi behar izan dugu informazio hori jatorrian bezain zehatz adieraztearen edo EKOa ahalik eta sinpleen mantentzearen arteko auzian. Maila ertaineko irtenbidea hartu dugu guk, bertatik informazioa modu

intuitiboan eskuratu ahal izatea oso kontuan hartuz. Iturriek duten informazio oso zehatzak edo bereziak, baina, ez du leku naturala aurkituko EKOan, eta zenbait informazio espezifiko edo teoria linguistiko zehatz batekiko dependente ezin izango da taxuz integratu.

EKOaren egokitasuna frogatzeko, 5 iturri desberdin integratu ditugu gure sisteman: giza-erabiltzaileari zuzendutako hiztegi bat (EH), datu-base lexikal bi (EDR eta EDBL), eta ezagutza-base bi (Hiztsua eta EuskalWordNet). Aipatzekoa da integratu ditugun iturriak ez direla jostailu-sistemak, alegia, milaka sarrera lexikal gordetzen dituzten neurri errealeko iturriak integratu ditugula ELHISAn. Bestalde, VI kapituluan bestelako iturri bat, lematizazio-base bat, nola integratu beharko litzatekeen ere erakutsi dugu.

Iturrien arteko heterogeneotasun semantikoari heltzeko, iturri bakoitzaren gainean modelizazio bat burutu dugu (BEK), eta, ondoren, EKOarekin parekatu, erregelen bidez. Erregelek baliabideen eduki-deskribapena (BED) adierazten dute, hau da, iturrien edukia errepresentatzen dute, EKOaren gaineko bistak balira bezala. Informazio-integrazioa gauzatzeko teknika sendoa den LAV hurbilpenari jarraitu diogu eduki-erregelak adierazteko.

LAV hurbilpena jarraitzeak malgutasun handia eskaini digu integrazio-sistema osatzeko. Besteak beste, informazio lexikala adierazterakoan askatasun osoa izan dugu goitik beherako estrategia jarraitzeko, eta, horrela, informazioa era orokor eta abstraktuan antolatzeko. Bestalde, sistema osoaren hedagarritasuna bermatzeko aukera eman digu, iturri berrien edukia deskribatzeak ez baitu zertan jadanik integratuta daudenena ezeztatu.

Ereduen artean galderen itzulpena gauzatzen duen *Minicon* algoritmoari gure datu-eredura egoki dadin zenbait hobekuntza egin dizkiogu, eta algoritmoa dugu. Algoritmoaren egokitasuna egiaztatu dugu, erregela kopuru handi eta BEK konplexuekin txukun eta arin betetzen baitu bere lana, V.2 atalean ikusi dugun legez. Algoritmoak gehiegizko sorkuntza duen arren —berridazketa posible anitz sor ditzake jatorrizko galdera bakar baterako—, optimizatzaileak gainsorkuntza horietatik berridazketa *hoberena* diskriminatzeko aukera eman beharko luke.

Wrapper-en teknologiari esker, iturrien egiturari buruzko heterogeneotasuna —heterogeneotasun estrukturala— ebazteko aukera izan dugu. Iturri bakoitzaren datu-eredu, kontsulta-lengoaia eta biltegi fisiko bereziak estaliz, *wrapper*-ek bidea ireki digute iturriotan dauden datuak ELHISArekin atzitzeko. Zeregin horretan, eta Donostiako Informatika Fakultateko karrera-bukaerako proiektu gisa, bi *wrapper* garatu ditugu, bata datu-base erlazionalen gainean, eta, bestea XMLz kodeturiko hiztegi baten gainean lan egiten due-

na. Batak SQLra itzultzen ditu ELHISAk igorritako galderak, eta besteak, berriz, XQuery lengoaiara. Erantzunak sistemari helarazteko, bestalde, OEM lengoaian oinarrituriko XML formatu bat erabili dugu.

Iturri anitzetatik jasotako erantzun-datuen gainean *datu-arazketa* prozesu bat diseinatu dugu, datuen arteko erlazioak hautemateko. Iturrietako datuak *garbitu* egiten ditugu lehen, datuek izan ditzaketen arazoak ebatzi eta balio komunak normalizatzeko, eta, ondoren, *objektuaren identifikazioari* ekiten diogu, datuak erkatu ahal izateko.

EKOa zein iturrietako BEKak errepresentatzeko *NeoClassic* deskribapen-logikaren formalismoa jarraitu dugu. Integrazioa bera gauzatzeko erabili ez dugun arren —ELHISAri eginiko galderak ez dira *NeoClassic* espresioak—, sistema oso baliagarria egokitu zaigu integrazio-prozesu osoan. Batetik, esan bezala, iturrien ereduak zein eredu kontzeptual orokorra adierazteko formalismo bateratua izan dugu. Bestetik, galderen itzulpenaren prozesua bideratzeko ere erabili dugu. Horrela, erabiltzailearen jatorrizko galdera sinplifikatzeko, galderarekin zerikusia duten iturrien erlazioak erabakitzeko, edo prozesuaren ostean zilegi ez diren berridazketak baztertzeko, *NeoClassic*ek eskainitako arrazoibide-mekanismoaz baliatu gara. Inplementatu ez den arren, gorago aipaturiko optimizatzailea ere, galderen barne-hartzearen arazoa ebatzi beharko duela kontuan izanik, deskribapen-logiketan oinarritutako teknikez balia daiteke. Azkenik, *NeoClassic* sistemak eskaintzen dituen erregelek bide dotorea eskaini digute *datu-arazketako* prozesua burutzeko.

VII.2 Aurrera begirakoak eta zabaldutako ikerlerroak.

Tesi-proiektu honen etorkizuneko perspektibak azalduko ditugu azpiatal honetan, labur-labur. Perspektiba horiek bi alde nagusitan bana ditzakegu, alegia, sistemaren hobekuntzak eta etorkizuneko ikerlerroak.

VII.2.1 Sistemaren hobekuntzak.

- Integrazioa taxuz burutzeko, hamaika gauza geratu da egiteke. Izan ere, ELHISAren jokabide zuzena frogatzeko erabat beharrezko diren hainbat modulu inplementatzeke geratu dira lan honetan. Hala ere, modulu hauen zeregina finkatu dugu, eta beren diseinua aztertu. Hona hemen inplementatu gabe geratu diren modulu nagusiak:

- Optimizatzailea. Bitartekoak sortutako berridazketa kopurua handia izan daiteke, eta berridazketa horien artean erreduntanterik ere agertuko da, alegia, berak erantzungo duen datu multzoa beste berridazketa bateko erantzunen azpimultzoa dena. Horiek baztertzeko, optimizatzaileak galdera konjuntiboen barne-hartzearen arazoa ebazten duen algoritmo bat inplementatu behar du, eta, horretarako, *NeoClassic* sistemak eskainitako funtzionalitateaz balia daiteke.
- Planifikatzailea. Iturrietara doazen berridazketak jasota, planifikatzailearen esku dago iturrietara doazen planak antolatzea, planak exekutatzea, eta erantzunak jasotzea. Tarteko urratsak ere bete beharko ditu, plan bakoitzak sortutako emaitza-datuak besteek sortutakoekin elkartu beharko baititu, azken emaitza lortzeko. Planifikatzailearen eginbeharrak finkatu ditugu, eta, betebeharrak burutzeko beharrezkoak diren atazak azpi-moduluetan antolatu: *Plan-Osatzailea*, *Plan-Igortzailea*, *Erantzunen Jasotzailea* eta *Datuen Araztailea*.
- Datu-arazketarako prozesua burutzeko *datuen garbiketari* eta *objektuen identifikazioari* ekin behar zaio. Bi eginkizun horiek planifikatzaileko Datuen Araztailea izeneko azpimoduluak burutuko ditu. Lehenengoari dagokionez, garbiketarako beharrezkoak diren erregela-motak identifikatu ditugu, eta mota bakoitzeko erregelak garatu ditugu. Bigarren eginkizuna, *objektuen identifikazioa*, alegia, garatzeke geratu da, baina IV.6.2 atalean bere inplementaziorako zenbait hausnarketa aurkeztu ditugu.
- Interfaze grafikoa. Giza-erabiltzaileak interfaze grafikoaren bidez egingo ditu bere galderak ELHISAn. Inplementatu gabeko modulua da hau ere, baina berari buruzko hainbat gogoeta egin ditugu, eta interfazeak beharko lituzkeen ezaugarri nagusiak finkatu —hala nola, eragiketa grafikoetan oinarritzea, galdeketarako ikussistemen antzera— V.1 atalean.

Edonola ere, sinetsiak gaude modulu hauen garapenak integrazioari zein portaera banatuari buruzko beste hainbat buruhauste jarriko dituela agerian. Izan ere, hain da konplexua informazioaren integrazio arloa, non prozesuko urrats orok sortzen dituen zailtasun eta erronka berriak.

- Alde ahul bat ELHISAren datu-ereduari dagokio. Galdera konjuntiboek erabiltzaile baten galdera arruntenak adierazteko aukera ematen badute ere, komenigarria litzateke adierazpen-ahalmen handiagoa duen formalismoa erabiltzea. Esaterako, ukapen-eragilerik erabili ezinak zenbait murriztapen ezartzen du galderak egiteko. Adibidez, erabiltzaileak galde dezake "Eman x baldintza betetzen duten elementuak". Ez, ordea, mota honetakoak: "Eman x baldintza betetzen *ez* duten elementuak". Ukapen-eragileak, halaber, datu-iturriak modelatzeko aukera berriak eskainiko dizkigu, eta iturrien arteko bereizketak finago egiteko bidea emango digu. Hala ere, oso kontu handiarekin aztertu beharreko auzia dugu, III.3.3.1 atalean ikusi baitugu zein ondorio izan dezakeen, LAV hurbilpenari jarraiki, galdera konjuntiboak baino espresibotasun aberatsagoak dituzten formalismoak erabiltzeak, galderen itzulpenaren prozesuan.

- EKOaren modelizazio aberatsagoa. Diseinatutako eredu kontzeptual orokorra findu eta aberastu nahi dugu, bere baitan informazio lexikal gehiago har dezan. Aberasketa bi hurbilpeni jarraituz etorriko da. Batetik, informazio-iturri gehiago ELHISAn integratzean, iturri horiek aurreikusita ez dagoen informazioa eskain baitezakete. Bestalde, ereduaren aberasketa sistemaren erabilera-azterketatik etor daiteke, eta, bide horretan, nahi genuke azterlana datu estatistikoekin hornitu: zein informazio eskuratzeko erabiltzen den ELHISA maizen, erabiltzaileak zein galderatan duen interes gehien eta abar.

- Eredu aberastua edo interfaze-eredua. Erabiltzaileak egun galdera bat egin nahi badio ELHISAri, jakin behar du galdera horren adierazpen zehatza egiten. Interfaze grafikoak laguntza estimagaitza eskain diezaiokeen arren, oso interesgarria litzateke EKOaren gainean interfaze-ereduak definitu ahal izatea, EKOko informazioaren gainean ikuspuntu desberdinak errepresentatzeko aukera emanez. Horrela, erabiltzaileari erraztasunak eskainiko dizkiogu, zenbait galdera era oso erraz batean egiteko aukera emanez, eta ELHISAk eskaintzen duen informazio zabalaren gaineko zatiren bat eskuratzen. Interfaze-eredua, finean, EKOaren gainean definitutako bistez osatua egongo lirateke. Gure ustez, bista hauen definizioa integrazio-erregelen bidez gauzatu beharko litzateke, era deklaratibo batean. Gainera, erregelak definitzeko GAV ereduari jarrai dakioke, EKOa bera aldakorra izateko bokazioarekin eratu ez

dugunez. Zalantzarik gabe, bisten erabilpena informazio lexikalaren integraziorantz eginiko beste urrats bat izango da, informazioaren dimentsio-aniztasuna adierazteko aukera ematen baitu.

- Eredu kontzeptualak —EKOa zein iturrienak— errepresentatzeko adierazpen-formalismo aberatsagoak beharko lituzke ELHISAk. Izan ere, egun erabilitako *NeoClassic* sistema, oso eraginkorra bada ere, aukera urriak eskaintzen ditu modelizaziorako, VI.1 atalean ikusi dugun bezala. Izan ere, alderantzizko erlazioak adierazi ahal izatea, edota erlazioen gainean hierarkiak osatu ahal izatea, iturrien eta EKOaren modelizazio txukunago batera eramango baikintuzkete. Horrela, *NeoClassic* baino espresibotasun handiagoa eskaintzen duten sistemak aztertu behar ditugu. Auzi horretan, *deskribapen-logiken* familikoa den FaCT (Horrocks *et al.*, 2000) sistema dugu hautagai nagusi.

VII.2.2 Internetera begira.

Interneten dagoen informazio lexikalaren kopurua handitzen doa etengabe, direla *on-line* hiztegi eleanitzak, direla LDC, CLR edo ELRA[1] erakundeek sortu edo publiko egindako biltegiak, eta abar. Erabiltzailea ELHISAren bidez informazio horretaz baliatu ahal izatea biziki interesgarria litzateke, gure ustez, eta horretara bideratuko ditugu etorkizuneko ikerketak.

Egun sistemak Interneteko iturriak onartzen baditu ere, iturriak datubase arruntak balira bezala ikusten ditu. Interneteko *web* orrietan dagoen informazioa taxuz baliatzeko, baina, zenbait hobespen egin behar ditugu sisteman.

- Internet orrietako interfazeak, datu-base tradizionalenak ez bezala, murritzak dira maiz, hots, atributu jakin batzuen arabera soilik galdetu ahal izaten da. III.3.4 atalean ikusi dugun bezala, integrazio-sistemak galdeketa-gaitasunez baliatzen dira egoera hau adierazteko, eta uste dugu geureak ere galdeketa-gaitasun horiek deskribatzeko moduak eskaini beharko lituzkeela. Ikertu beharko dugu, horrela, galdeketa-gaitasunak onartzeak dakartzan ondorioak, batik bat, galderen itzulpenaren prozesuan (ikus, adibidez, Rajaraman *et al.*, 1995; Florescu *et al.*, 1998a; Li eta Chang, 2000).

[1] LDC (Linguistic Data Consortium): http://www.ldc.upenn.edu/
CLR (Consortium for Lexical Research): http://crl.nmsu.edu/clr/CLR.html
ELRA (European Language Resources Association): http://www.icp.grenet.fr/ELRA

- *Wrapper*-en garapenak dimentsio berri bat hartzen du Interneteko orriak integratu behar badira. ELHISAn *wrapper*-ak garatzeak, egun, eskulangintzarekin du zerikusi handia, eta, horrela, iturri berrien *wrapper*-ak huts-hutsetik sortu behar dira, aurretik garatutako *wrapper*-ek laguntza eskasa ematen dutelarik. Estrategia berriak beharko dira Internet orrietarako *wrapper*-en garapenean zeren, batetik, orrien kopurua oso handia izan baitaiteke, eta badakigu *wrapper*-en garapena lan neketsua izaten dela; bestetik, baina, Interneteko orriek izan dezakete hainbat ezaugarri komun; esaterako, denak daude sasi-egituratutako paradigma batez adierazita. Horrela, *web* orrien gainean sortutako *wrapper*-ek informazioa erauzi egin behar dute, orrian islatutako informazioak datu-baseko erlazioen antza har dezan. Horrela, aztertu behar dira bideak informazio-iturrien gainean *wrapper*-ak automatikoki, edo, behintzat, erdi-automatikoki garatzeko (Kushmerick *et al.*, 1997; Ashish eta Knoblock, 1997; Knoblock *et al.*, 2001).

- Internetetik etorriko den informazioa ez da datu-base lexikaletakoa bezain zehatza izango, eta, seguruenik, akatsak izango ditu bere baitan. Horrela, datu-arazketako prozesu nabarmenki konplexuago batera eramango gaitu Interneteko *web* orrien integrazioak. Oso interesgarria litzateke datu-arazketako prozesua bideratzeko erregelak era automatiko edo erdi-automatiko batean ikastea, "Active Atlas" objektu-identifikaziorako sistemak egiten duen antzera (Tejada *et al.*, 2001).

VII.2.3 LNPko aplikazioak integratzen.

Datu linguistikoen integrazioaz gain, LNPko aplikazioen integrazioak ere arreta berezia hartu du azken boladan. Leidner-ek dioen legez, LNPko aplikazioak ez dira berrerabilgarriak, batik bat, aplikazioak integratzea zaila delako oso, eta sortzen dituen arazoak maiz gutxietsi egiten direlako (Leidner, 2003). Egun dagoen "tresna azoka"(Bird *et al.*, 2000) ingurune bateratu batean integratzea proposatzen du egileak, eta integrazioa burutzeko softwarearen ingeniaritzaren arloko "osagaietan oinarritutako garapenaren" antzeko paradigma bati jarraitu behar zaiola azpimarratzen du.

IXA taldean arazo horri heldu nahi diogu, eta norabide horretan zenbait urrats eman ditugu, adibidez, taldeko aplikazioen arteko komunikazioa bermatzen duen anotazio-sistema bat garatuz (Artola *et al.*, 2000). Anotazio-

sistemak hainbat tresnaren sarrera-irteerak estandarizateko eta bateratzeko bidea eskaintzen digu, *ad hoc* egindako formatu bereziak saihestuz.

Errepresentazio bateratuak azpiegitura egokia eskaintzen digu osagaietan oinarrituriko ingurune bat garatzeko, zeinean integra daitezkeen software-modulu —osagai— berriak era erraz batean, GATE (Bontcheva *et al.*, 2002) edo ATLAS (Bird *et al.*, 2000) sistemen antzera. Horretarako, eginkizun desberdinak burutzeko zerbitzuak eskaini nahi dizkiegu aplikazioei, API bateratu baten bidez. Software-moduluek, eginkizun berezi bat burutzen dutelarik, zenbait zerbitzu linguistiko beharko dituzte, eta beste zenbait eskaini.

Testuinguru horretan, gure ELHISA sistema zerbitzu lexikalak eskaintzeko atzibide paregabea delakoan gaude, datu lexikal kopuru handiak eskaintzeko aukera ematen baitu. Era honetan, LNPko aplikazioen integrazioan buruhauste larria den datu lexikalen integrazioaren arazoari irtenbide bat emango genioke.

Zerbitzu lexikalen hornitzailea izateko oso komenigarria izango da gorago aipaturiko eredu aberastua lantzea, EKOaren gainean bista desberdinak definituz, eta, hartara, zerbitzu lexikalen eskaintza finagoak osatuz.

Bibliografia

Abiteboul S. Querying semi-structured data. In Afrati F.N. and Kolaitis P., editors, *Database Theory—ICDT'97, 6th International Conference*, volume 1186 of *Lecture Notes in Computer Science*, pages 1–18, Delphi, Greece, 8–10 January 1997. Springer.

Abiteboul S. and Duschka O.M. Complexity of answering queries using materializated views. In *Proc. of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'98)*, pages 254–263, 1998.

Abiteboul S., Hull R., and Vianu V. *Foundations of Databases*. Addison Wesley, 1995.

Abiteboul S., Quass D., McHugh J., Widom J., and Wiener J.L. The Lorel query language for semistructured data. *International Journal on Digital Libraries*, 1(1):68–88, 1997.

Agirre E. *Kontzeptuen arteko erlazio-izaeraren formalizazioa ontologiak erabiliaz: Dentsitate Kontzeptuala*. PhD thesis, Euskal Herriko Unibertsitatea, Donostia, 1999.

Agirre E., Ansa O., Arregi X., Artola X., Díaz de Ilarraza A., and Lersundi M. A conceptual schema for a Basque lexical-semantic framework. In *Conference on Computational Lexicography and Text Research (Complex 2003)*, Budapest, 2003.

Agirre E., Ansa O., Arriola J.M., Díaz de Ilarraza A., Pociello E., and Uria L. Methodological issues in the building of the Basque WordNet: quantitative and qualitative analysis. In *Proceedings of the first International WordNet Conference*, pages 21–25, Mysore (India), January 2002.

Agirre E., Arregi X., Artola X., Díaz de Ilarraza A., Edvard F., and Sarasola K. Lexical knowledge representation in an intelligent dictionary help system. In *Proc. of COLING'94*, pages 544–550, Kyoto (Japan), 1994.

Agirre E., Arregi X., Artola X., Díaz de Ilarraza A., Sarasola K., and Soroa A. An intelligent dictionary help system. *Encyclopedia of Library Information Science*, 6:242–259, 2000.

Agirre E., Arregi X., Artola X., Díaz de Ilarraza A., Sarasola K., and Soroa A. MLDS: A translator-oriented multilingual dictionary system. *Natural Language Engineering*, 5(4), 2001.

Albano A. and Attardi G. Issues in data base and knowledge base integration. In Schmidt J.W. and Thanos C., editors, *Foundations of Knowledge Base Management: Contributions from Logic, Databases, and Artificial Intelligence*, pages 283–291. Springer, Berlin, Heidelberg, 1989.

Alderson A. and Shah H. Viewpoints on legacy systems. *Communications of the ACM*, 42(3):115–116, 1999.

Aldezabal I. *Aditz-azpikategorizazioaren azterketa sintaxi partzialetik sintaxi osorako bidean. 100 aditzen azterketa, Levin-en (1993) lana oinarria hartuta, eta metodo automatikoak baliatuz.* PhD thesis, Euskal Herriko Unibertsitatea, Gasteiz, 2004.

Aldezabal I., Ansa O., Arrieta B., Artola X., Ezeiza A., Hernandez G., and Lersundi M. EDBL: a general lexical basis for the automatic processing of Basque. In *Proceedings of IRCS Workshop on linguistic databases*, Philadelphia. USA, 2001.

Amsler R.A. A taxonomy for english nouns and verbs. In *Proc. of the 19th Annual Meeting of the Association for Computational Linguistics (ACL'81)*, pages 133–138, Stanford, California, 1981.

Arens Y., Hsu C.N., and Knoblock C.A. Query processing in the SIMMS information mediator. In *Advanced Planning Technology*. AAAI Press, California, USA, 1996.

Arens Y. and Knoblock C.A. Planning and reformulating queries for semantically-modeled multidatabase systems. In *Proc. of the 1st International Conference On Information and Knowledge Management (CIKM'92)*, Baltimore, USA, 1992.

Arregi X. *ANHITZ: Itzulpenean laguntzeko hiztegi-sistema eleanitza*. PhD thesis, Euskal Herriko Unibertsitatea, Donostia, 1995.

Arregi X., Arriola J.M., Artola X., Díaz de Ilarraza A., García E., Lascurain V., Sarasola K., Soroa A., and Uria L. Semiautomatic conversion of the Euskal Hiztegia Basque dictionary to a queryable electronic form. *Traitement automatique des langues (TAL)*, 44(2):107–124, 2003.

Arriola J.M. *EUSKAL HIZTEGIA-ren Azterketa eta Egituratzea Ezagutza Lexikalaren Eskuratze Autoimatikoari Begira*. PhD thesis, Euskal Herriko Unibertsitatea, Donostia, 2000.

Arriola J.M., Artola X., and Soroa A. Automatic extraction of lexical information from an ordinary dictionary. In *Proc. of EURALEX*, Göteborg, Sueden, 1996.

Arriola J.M. and Soroa A. Lexical information extraction for Basque. In *Proc. of the CLIM'96*, Montreal,Canada, 1996.

Artola X. *HIZTSUA: Hiztegi-sistema Urgazle Adimendunaren Sorkuntza eta Eraikuntza*. PhD thesis, Euskal Herriko Unibertsitatea, Donostia, 1993.

Artola X., Díaz de Ilarraza A., Ezeiza N., Gojenola K., Maritxalar A., and Soroa A. A proposal for the integration of NLP tools using SGML-tagged documents. In *Proc. of the Second Int. Conf. on Language Resources and Evaluation*, Athens. Greece, 2000.

Artola X. and Soroa A. An architecture for a federation of highly heterogeneous lexical information sources. In *IRCS Workshop on linguistic databases.*, Philadelphia. USA, 2001a.

Artola X. and Soroa A. Using data integration techniques in a federation of heterogeneus lexical databases. In *Proceedings of NAACL. Workshop on "Wordnet and Other Lexical Resources: Applications, Extensions and Customizations".*, Pittsburgh. USA, 2001b.

Ashish N. and Knoblock C.A. Semi-automatic wrapper generation for internet information sources. In *Conference on Cooperative Information Systems*, pages 160–169, 1997.

Baader F., Calvanese D., McGuinness D., Nardi D., and Schneider P.P., editors. *The Description Logics Handbook*. Cambridge University Press, 2001.

Baeza-Yates R.A. and Navarro G. XQL and proximal nodes. *JASIST*, 53(6): 504–514, 2002.

Batini C., Lenzerinio M., and Navathe S.B. A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys*, pages 323–364, 1986.

Becker J.D. Multilingual word processing. *Scientific American*, 251(1):96–107, 1984.

Beery C., Levy A.Y., and Rousset M.C. Rewritting queries using views in description logics. In *Proc. of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'97)*, pages 99–108, Tucson, AZ. USA, 1997.

Bel N., Busa F., Calzolari N., Gola E., Lenci A., Monachini M., Ogonowski A., Peters I., Peters W., Ruimy N., Villegas M., and Zampolli A. SIMPLE: A general framework for the development of multilingual lexicons. In *2nd International Conference on Language Resources and Evaluation (LREC2000)*, Athens. Greece, 2000.

Bermúdez J. *Una lógica de descripciones en un nivel meta-ontológico para la gestión de sistemas de información globales*. PhD thesis, Euskal Herriko Unibertsitatea, Donostia, 2001.

Bird S., Day D., Garofolo J., Henderson J., Laprun C., and Liberman M. ATLAS: A flexible and extensible architecture for linguistic annotation. pages 1699–1706, July 13 2000.

Blanco J.M., Illarramendi A., and Goñi A. Building a federated relational database system: An approach using a knowledge-based system. *Int. Journal of Intelligent and Cooperative Information Systems*, 3(4):415–455, 1994.

Boguraev B. Building a lexicon: The contibution of computers. *Internazional journal of Lexicography*, 4:227–260, 1991.

Boguraev B. and Briscoe T., editors. *Computational Lexicography for Natural Language Processing*. Longman, 1989.

Bohannon P., Freire J., Roy P., and Simeon J. From XML schema to relations: A cost-based approach to XML storage. In *ICDE*, 2002.

Bontcheva K., Cunningham H., Maynard D., Tablan V., and Saggion H. Developing reusable and robust language processing components for information systems using GATE. In *Proceedings of the 13th International Workshop on Database and Expert Systems Applications*, pages 223–227. IEEE Computer Society, 2002. ISBN 0-7695-1668-8.

Bordiga A., Brachman R.J., McGuinness D.L., and Resnick L.A. CLASSIC: A structural data model for objects. In *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pages 59–67, 1989.

Borgida A. Description logics in data management. *Knowledge and Data Engineering*, 7(5):671–682, 1995.

Brachman R. What IS-A is and isn't: An analysis of taxonomic links in semantic networks. *IEEE Computer*, 16(10):30–36, 1983.

Brachman R., Borgida A., McGuinness D., Patel-Schneider P., and Resnick L. The CLASSIC knowledge representation system of, KL-ONE: The next generation. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, pages 1036–1043, ICOT, Japan, 1992. Association for Computing Machinery.

Brachman R. and Schmoke J. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, 1985.

Brachman R.J. and Levesque H.J. The tractability of subsumption in frame-based description languages. In *Proc. of the 4th National Conference of the American Association for Artificial Intelligence*, pages 34–37, Austin, Texas, 1985.

Briscoe T. Lexical issues in natural language processing. In Klein E. and Veltman F., editors, *Natural Language and Speech*, pages 39–68. Springer-Verlag, 1991.

Cahill L., Doran C., Evans R., Mellish C., Paiva D., Reape M., and Scott D. Achieving theory-neutrality in reference architectures for NLP: to what extent is it possible/desirable. In *Proceedings of the AISB'99 workshop on reference architectures and data standards for NLP*, pages 32–35, 1999.

Calvanese D., De Giacomo G., and Lenzerini M. Answering queries using views in description logics. In *Proc. of the 1999 Description Logics Workshop (DL'99), CEUR Workshop*, volume 2, pages 9–13, 1999a.

Calvanese D., De Giacomo G., Lenzerini M., Nardi D., and Rosati R. Information integration: Conceptual modeling and reasoning support. In *Proc. of the 6th Int. Conf. on Cooperative Information Systems (CoopIŚ98)*, pages 280–291, 1998a.

Calvanese D., De Giacomo G., Lenzerini M., and Vardi M.Y. Rewriting of regular expressions and regular path queries. *J. of Computer and System Sciences*, 64(3):443–465, 2002.

Calvanese D., De Giacomo G., and Lenzerini M. Description logics for information integration. In *Computational Logic: From Logic Programming into the Future (In honour of Bob Kowalski)*. Lecture Notes in Computer Science.Springer Verlag, argitaratzeke.

Calvanese D., Giacomo G.D., Lenzerini M., Nardi D., and Rosati R. A principled approach to data integration and reconciliation in data warehousing. In *Design and Management of Data Warehouses*, page 16, 1999b.

Calvanese D., Lembo D., and Lenzerini M. Survey on methods for query rewriting and query answering using views. Technical report, Integration, Warehousing and Mining of Heterogeneous Data Sources Project, April 2001.

Calvanese D., Lenzerini M., and Nardi D. Description logics for conceptual data modeling. In *Logics for Databases and Information Systems*, pages 229–263. 1998b.

Calzolari N. Issues for lexicon building. In Zampolli A., Calzolari N., and Palmer M., editors, *Current Issues in Computational Linguistics: Essays in Honour of Don Walker*, pages 267–281. Giardini Editori e Stampatori and Kluwer Academic Publishers, Pisa and Dordrecht, 1994.

Calzolari N., Zampolli A., and Lenci A. Towards a standard for a multilingual lexical entry: The EAGLES/ISLE initiative. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, pages 264–279. Springer-Verlag, 2002.

Carpenter B. Typed feature structures: A generalization of first-order terms. In Saraswat V. and Ueda K., editors, *Logic Programming: Proc. of the 1991 International Symposium*, pages 187–201. MIT Press, Cambridge, MA, 1991.

Carpenter B. and Penn G. ALE: The Atribute Logic Engine. User's guide. Version 2.0.3. Technical report, Computational Linguistic Program. Philosophy Department, Carnegie Mellon University, 1997.

Catarci T., Costabile M.F., Levialdi S., and Batini C. Visual Query Systems for databases: a survey. *Journal of Visual Languages & Computing*, 8(2): 215–260, 1997a.

Catarci T. and Lenzerini M. Representing and using interschema knowledge in cooperative information systems. *Journal for Intelligent and Cooperative Information Systems*, 2(4):375–399, 1993.

Catarci T., Santucci G., and Cardiff J. Graphical interaction with heterogeneous databases. *VLDB J.*, 6(2):97–120, 1997b.

Chamberlin D. XQuery: An XML query language. *IBM Systems Journal*, 41(4), 2002. ISSN 0018-8670. URL `http://www.research.ibm.com/journal/sj/414/chamberlin.pdf`.

Chan E.P.F. Containment and minimization of positive conjunctive queries in OOSB's. In *Proc. of the Eleventh ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 202–211, San Diego, CA, 1992.

Chandra A.K. and Merlin P.M. Optimal implementation of conjunctive queries in relational data bases. In *Conference Record of the Ninth Annual ACM Symposium on Theory of Computing (STOC'77)*, pages 77–90, Boulder, CO. USA, 1977.

Chawathe S., Garcia-Molina H., Hammer J., Ireland K., Papakonstantinou Y., Ullman J., and Widom J. The TSIMMIS project: Integration of heterogeneous information sources. In *Proc. of IPSJ Conference*, pages 7–18, Tokio, Japan, 1994.

Copestake A. An approach to building the hierarchical element of a lexical knowledge base from a machine readable dictionary. In *Proc. of First International Workshop Inheritance in NLP*, pages 19–29, Tilburg, Netherlands, 1990.

Cunningham H., Bontcheva K., Peters W., and Wilks Y. Uniform language resource access and distribution in the context of a General Architecture for Text Engineering (GATE). In *Proceedings of the Workshop on Ontologies and Language Resources (OntoLex'2000)*, Sozopol, Bulgaria, September 2000.

Cunningham H. A definition and short history of language engineering. *Journal of Natural Language Engineering*, (5):1–16, 1999.

Deutsch A., Fernandez M., Florescu D., Levy A., and Suciu D. A query language for XML. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1155–1169, 1999.

Duineveld A.J., Stoter R., Weiden M.R., Kenepa B., and Benjamins V.R. Wondertools ? a comparative study of ontological engineering tools. In *Proc. of the 12th Workshop on Knowledge Acquisition, Modeling and Management (KAW'99)*, Banff, Alberta, Canada, 1999.

Duschka O.M. and Genesereth M.R. Answering recursive queries using views. In *Proc. of the ACM SIGACT-SIGMOD-SIGART symposium on Principles on Database Systems*, Tucson, AZ. USA, 1997a.

Duschka O.M. and Genesereth M.R. Query planning in Infomaster. In *Proc. of the ACM symposium on Applied Computing*, San Jose, CA. USA, 1997b.

Duschka O.M., Genesereth M.R., and Levy A.Y. Recursive query plans for data integration. *Journal of Logic Programming*, 43(1):49–73, 2000. URL `citeseer.nj.nec.com/duschka99recursive.html`.

Evans R. and Gazdar G. DATR: A language for lexical knowledge representation. *Computational Linguistics*, 22(2):167–216, 1996.

Farquhar A., Fikes R., and Rice J. The Ontolingua Server: A tool for collaborative ontology construction. In *Proc. of the 10th Knowledge Acquisition for KBS Workshop (KAW'96)*, Banff, Alberta, Canada, 1996.

Florescu D. and Kossmann D. Storing and querying XML data using an RDMBS. *IEEE Data Engineering Bulletin*, 22(3):27–34, 1999.

Florescu D., Kossmann D., and Manolescu I. Integrating keyword search into XML query processing. *Computer Networks*, 33(1–6):119–135, 2000.

Florescu D., Levy A., and Suciu D. Query containment for conjunctive queries with regular expressions. In ACM, editor, *PODS '98. Proceedings of the ACM SIGACT–SIGMOD–SIGART Symposium on Principles of Database Systems, June 1–3, 1998, Seattle, Washington*, pages 139–148, New York, NY 10036, USA, 1998a. ACM Press. ISBN 0-89791-996-3.

Florescu D., Levy A.Y., and Mendelzon A. "Database Techniques for the World-Wide Web: A Survery". *ACM SIGMOD Record*, 27(3):59–74, September 1998b.

Franconi E. Description logics for natural language processing. In *Proc. of the 1994 AAAI Fall Symposium on Knowledge Representation for Natural Language Processing in Implemented Systems*, New Orleans, US, 1994.

Friedman M. and Weld D.S. Efficiently executing information-gathering plans. In *15th International Joint Conference on Artificial Intelligence*, pages 785–791, Nagoya, Japan, 1997.

Galhardas H., Florescu D., and Shasha D. Declarative data cleaning: Language, model, and algorithms. In *Proc. of 27th International Conference on Very Large Data Bases*, pages 371–380, Rome, Italy, September 2001.

Galhardas H., Florescu D., Shasha D., and Simon E. Declarative data cleaning: Language, model, and algorithms. *Proc. of 27th International Conference on Very Large Data Bases*, pages 371–380, October 2000.

Garcia-Molina H., Labio W.J., and Yang J. Expiring data in a warehouse. In *Proc. 24th Int. Conf. Very Large Data Bases, VLDB*, pages 500–511, 24–27 1998.

Garcia-Molina H., Papakonstantinou Y., Quass D., Rajaraman A., Sagiv Y., Ullman J.D., Vassalos V., and Widom J. The TSIMMIS approach to mediation: Data models and languages. *Journal of Intelligent Information Systems*, 8(2):117–132, 1997.

Gazdar G. Paradigm merger in natural language processing. In Milner R. and Wand I., editors, *Computing Tomorrow: Future Research Directions in Computer Science*, pages 88–109. Cambridge University Press, 1996.

Gazdar G. and Mellish C. *Natural Language Processing in Prolog.* Addison-Wesley Publishing Company, Cambridge, 1989.

Genesereth M. and Ketchpel S.P. Software agents. *Comunications of the ACM*, 37(7):48–53, 1994.

Genesereth M.R., Keller A.M., and Duschka O.M. Infomaster: an information integration system. In Peckman J.M., editor, *Proceedings, ACM SIGMOD International Conference on Management of Data: SIGMOD 1997: May 13–15, 1997, Tucson, Arizona, USA*, volume 26(2) of *SIGMOD Record (ACM Special Interest Group on Management of Data)*, pages 539–542, New York, NY 10036, USA, 1997. ACM Press. ISBN 0-89791-911-4.

Gibbon D., Peters W., and Wittenburg P. Metadata elements for lexicon descriptions. Technical report, NPI Nijmegen, 2001.

Goh C.H. *Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources.* PhD thesis, MIT, 1997.

Goldman R. and Widom J. DataGuides: Enabling query formulation and optimization in semistructured databases. In Jarke M., Carey M.J., Dittrich K.R., Lochovsky F.H., Loucopoulos P., and Jeusfeld M.A., editors, *VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece*, pages 436–445. Morgan Kaufmann, 1997. ISBN 1-55860-470-7.

Goñi A., Illarramendi A., Mena E., and Blanco J. An optimal cache for a federated database system. *Journal of Intelligent Information Systems (JIIS), ISSN 0925-9902*, 9(2):125–156, September/October 1997.

Grisham R., Macleod C., and Meyers A. Comlex syntax: building a computational lexicon. In *Proc. of the 15th Annual Meeting of the Association for*

Computational Lunguistics (COLING'94), pages 268–272, Kyoto, Japan, 1994.

Gruber T.R. Ontolingua. a mechanism to support portable ontologies. Technical report, Stanford University, Knowledge Systems Laboratory, March 1992.

Gruber T.R. A translation approach to portable ontology specifications. *Knowledge Acquisition*, pages 199–220, 1993.

Guarino N. Understanding, building and using ontologies. a commentary to 'using explicit ontologies in kbs development' by heijst, schreiber and wielinga. *International Journal of Human and Computer Studies*, 43(2/3): 293–310, 1997.

Heid U. EAGLES computational lexicons working group - reading guide. Technical Report EAG-CLWG-FR-2, 1996.

Heimbigner D. and McLeod D. A federate architecture for information managment. *ACM Transactions on Office Information Systems*, 3(3):253–278, 1985.

Hernandez M. and Stolfo S. Real-world data is dirty: Data cleansing and the merge/purge problem. *Journal of Data Mining and Knowledge Discovery*, 2(1):9–37, 1995.

Horrocks I., Sattler U., and Tobies S. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL*, 8(3):239–264, May 2000.

Ide N., Maitre J.L., and Véronis J. Outline of a model for lexical databases. In Zampolli A., Calzolari N., and Palmer M., editors, *Current Issues in Computational Linguistics: Essays in Honour of Don Walker*, pages 283–320. Giardini Editori e Stampatori and Kluwer Academic Publishers, Pisa and Dordrecht, 1994.

Ide N. and Véronis J. Refining taxonomies extracted from machine-readable dictionaries. In Hockey S. and Ide N., editors, *Research in Humanities Computing 2*. Oxford University Press, Pisa and Dordrecht, 1993.

Ide N. and Véronis J. Knowledge extraction from machine-readable dictionaries: An evaluation. In Steffens P., editor, *Machine Translation and the Lexicon*, pages 19–34. Springer-Verlag, 1994.

Ide N. and Véronis J. *Text Encoding Initiative. Background and Context.* Kluwer Academic, Dordrecht, 1995.

Ives B. and Jarvenpaa S.L. Applications of global information technology: key issues for management. *MIS Q.*, 15(1):33–49, 1991. ISSN 0276-7783.

Ives Z.G., Florescu D., Friedman M., Levy A.Y., and Weld D.S. An adaptive query execution system for data integration. In Delis A., Faloutsos C., and Ghandeharizadeh S., editors, *SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, June 1-3, 1999, Philadelphia, Pennsylvania, USA*, pages 299–310. ACM Press, 1999. ISBN 1-58113-084-8.

Jarke M., Lenzerini M., Vassiliou Y., and Vassiliadis P. *Fundamentals of Data Warehouses*. Springer-Verlag, 2000.

Jarke M. and Vassiliou Y. Foundations of data warehouse wuality: An overview of the DWQ project. In *Proceedings of the 2nd International Conference on Information Quality*, pages 199–313, Cambridge, MA, 1997.

Jing H. and McKeown K. Combining multiple, large-scale resources in a reusable lexicon for natural language generation. In *36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL'98)*, pages 607–613, Montreal, Quebec, Canada, 1998.

Jones D., Bench-Capon T., and Visser P. Methodologies for ontology development. In *Proc. IT&KNOWS Conference of the 15th IFIP World Computer Congress*, Budapest, 1998. Chapman-Hall.

Kanngießer S. Zwei Printzipen des Lexikonimport und Lexikonexport. *Lexikonimport, Lexikonexport - Studien zur Wiederverwertung lexikalischer Ressourcen*, 20:90–110, 1996.

Kashyap V. and Seth A. Semantic heterogeneity in global information systems: The role of metadata, context and ontologies. In Papazoglou M. and Schlageter G., editors, *Cooperative Information Systems: Current Trends and Applications*. 1996.

Kay M. Functional grammar. In et al. C.C., editor, *Proceedings of the Fifth Annual Meeting of the Berkeley Linguistic Society*, 1979.

Kirk T., Levy A.Y., Sagiv Y., and Srivastava D. The Information Manifold. In Knoblock C. and Levy A., editors, *Information Gathering from Heterogeneous, Distributed Environments*, AAAI Spring Symposium Series, Stanford University, Stanford, California, March 1995.

Knoblock C., Minton S., Ambite J.L., Ashish N., Muslea I., Philpot A.G., and Tejada S. The ARIADNE approach to web-based information integration. *International the Journal on Cooperative Information Systems (IJCIS) Special Issue on Intelligent Information Agents: Theory and Applications*, 10(1/2):145–169, 2001.

Kushmerick N., Weld D.S., and Doorenbos R.B. Wrapper induction for information extraction. In *Intl. Joint Conference on Artificial Intelligence (IJCAI)*, pages 729–737, 1997.

Kwok C.T. and Weld D.S. Planning to gather information. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, pages 32–39, Menlo Park, August 4–8 1996. AAAI Press / MIT Press. ISBN 0-262-51091-X.

Lapp J., Robbie J., and Schac D. XML query language (XQL). In *The query languages workshop*, 1998.

Lattes F.G.V. and Rousset M.C. The use of CARIN language and algorithms for information integration: The PICSEL system. *International Journal of Cooperative Information Systems*, 9(4):383–401, 2000.

Leidner J.L. Current issues in software engineering for natural language processing. In *Proceedings of the Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS) held at the Joint Conference for Human Language Technology and the Annual Meeting of the Noth American Chapter of the Association for Computational Linguistics 2003 (HLT/NAACL'03)*, pages 45–50, Edmonton, Alberta, Canada, May 2003.

Levin B. Building a lexicon: The contribution of linguistics. 1991.

Levy A.A. and Rousset M.C. CARIN: A representation language combining Horn rules and description logics. *Artificial Intelligence*, 104(1–2):165–209, 1998.

Levy A.Y. The Information Manifold approach to data integration. *IEEE Intelligent Systems*, 13:12–16, September/October 1998.

Levy A.Y. Answering queries using views: A survey. Technical report, Computer Science Dept. Washington Univ., 2000.

Levy A.Y., Mendelzon A.O., Sagiv Y., and Srivastava D. Answering queries using views. In *Proc. of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'95)*, pages 95–104, San Jose, CA. USA, 1995.

Levy A.Y., Rajaraman A., and Ordille J.J. Querying heterogeneous information sources using source descriptions. In *Proc. of the 1996 Conference on Very Large Data Bases (VLDB'96)*, pages 251–262, 1996.

Li C. and Chang E. Query planning with limited source capabilities. In *16th International Conference on Data Engineering (ICDE' 00)*, pages 401–412, Washington - Brussels - Tokyo, March 2000. IEEE. ISBN 0-7695-0506-6.

Litwin W., Mark L., and Roussoupoulos N. Interoperability of multiple autonomous databases. *ACM Computing Surveys*, 22(3):267–293, Sep 1990.

MacNaugth J. Reusability of lexical and terminological resources; steps towards the independence. In *Proc. of Int. Workshop on Electronic Dictionaries*, pages 97–107, Kanagawa, Japan, 1990.

McGregor R. and Bates R. The LOOM knowledge representation language. Technical report, University of Southern California. Information Science Institute, 1991.

Mena E., Kashyap V., Seth A.P., and Ilarramendi A. OBSERVER: And approach for query processing in global information systems based on interoperoperability between pre-existing ontologies. In *Proc. of the 1st IFCIS International Conference on Cooperative Information Systems (CooplS'96)*, Brussels, Belgium, 1996.

Mena E., Kashyap V., Seth A.P., and Ilarramendi A. OBSERVER: And approach for query processing in global information systems based on interoperation across pre-existing ontologies. *International Journal of Distributed and Parallel Databases (DAPD)*, 8(2):223–271, 2000.

Meng W. and Yu C. Query processing in multidatabase systems. In Kim W., editor, *Modern Database Systems*, pages 551–572. ACM Press, Addison-Wesley, New York, 1995.

Miike S., Amano S., Uchida H., and Yokoi T. The structure and function of the EDR concept dictionary. *Terminology and Knowledge Engineering*, 1, 1990.

Miller G. Five papers on WordNet. *Special Issue of International Journal of Lexicography*, 3(4), 1990.

Mitra P. An algorithm for efficiently answering queries using views. Technical report, Infolab, Stanford University, 1999.

Monge A. An efficient domain-independent algorithm for detecting approximately duplicate database records. In *Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'97) in cooperation with ACM-SIGMOD97*, Tucson, USA, may 1997.

Nakamura J. and Nagao M. Extraction from semantic information from an ordinary english dictionary and its evaluation. In *Proc. of the 12th International Conference of Computational Linguistics, COLING'88*, pages 459–464, Budapest, Hungary, 1988.

Neff M.S., Blaser B., Lange J.M., Lehmann H., and Dominguez I.Z. Get it where you can: Acquiring and maintaining bilingual lexicons for machine-translation. In *AAAI Spring Simposium on Building Lexicons for Machine Translation*. 1993.

Normier B. and Nossim M. GENELEX project: EUREKA for linguist engineering. In *Proc. of Int. Workshop on Electronic Dictionaries*, pages 63–70, Kanagawa, Japan, 1990.

Özsu M.T. and Valduriez P. *Principles of distributed database systems*. Prentice Hall, 1999.

Papakonstantinou Y., Garcia-Molina H., and Widom J. Object exchange across heterogeneous information sources. In Yu P.S. and Chen A.L.P., editors, *11th Conference on Data Engineering*, pages 251–260, Taipei, Taiwan, 1995a. IEEE Computer Society.

Papakonstantinou Y., Gupta A., Garcia-Molina H., and Ullman J. A query translation scheme for rapid implementation of wrappers. *Lecture Notes in Computer Science*, 1013:161–191, 1995b.

Pottinger R. and Levy A.Y. A scalable algorithm for answering queries using views. In *Proc. of the 26th International Conference on Very Large Data Bases (VLDB'2000)*, 2000.

Preece A.D., Hui K.J., Gray W.A., Marti P., Bench-Capon T.J.M., Jones D.M., and Cui Z. The kraft architecture for knowledge fusion and transformation. In *Proc. of the 19th SGES International Conference on Knowledge-Based System and Applied Artificial Intelligence (ES'99)*, 1999.

Quantz J. and Royer V. Implementation of BACK system. version 4. KIT-report 78. Technical report, FB Informatik, Technische Universität Berlin, Germany, 1990.

Rahm E. and Do H.H. Data cleaning: Problems and current approaches. *IEEE Bulletin of the Technical Committee on Data Engineering*, 23(4), December 2000.

Rajaraman A., Sagiv Y., and Ullman J.D. Answering queries using templates with binding patterns. In *Proceedings of the 14th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS '95)*, pages 105–112, New York, May 1995. ACM. ISBN 0-89791-730-8.

Ramsay A. Can a neutral dictionary be useful? In ming Guo C., editor, *Machine Tratable Dictionaries: Design and Construction*, pages 65–74. Ablex Publishing Corporation, Norwood, N.J:Ablex, 1995.

Resnick L.A., Patel-Schneider P.F., McGuinness D.L., Weixelbaum E., Abrahams M.K., Borgida A., Brachman R., Isbell C.L., and Zalondek K.C. NeoClassic user's guide: Version 1.0. Technical report, Artificial Intelligence Principles Research Department, AT&T Bell Labs, 1996.

Ribeiro R., Mamede N., and Trancoso I. Reusing linguistic resources: a case study in morphosyntactic tagging. In *TASHA'2003 - Workshop on Tagging and Shallow Processing of Portuguese,*, 2003.

Roth M.T. and Schwartz P. Don't Scrap It, Wrap it ! a wrapper architecture for legacy data sources. In *Proc. of the 23rd International Conference on Very Large Data Bases (VLDB'97)*, pages 266–275, 1997.

Ruimy N., Corazzari O., Elisabetta G., Spanu A., Calzolari N., and Zampolli A. The european LE-PAROLE project and the italian lexical instantiation. In *ALLC/ACH*, pages 149–153, Lajos Kossuth University, Debrecen, Hungary, 1998.

Sagiv Y. and Yannakakis M. Equivalences among relational expressions with the union and difference operators. *Journal of the ACM*, 27(4):633–655, 1980.

Sarasola I. *Euskal Hiztegia*. Kutxa Fundazioa, 1996.

Schieber S.M., Uszkoreit H., Pereira F., Robinson J., and Tyson M. The formalism and implementation of PATR-II. Report, SRI International, Menlo Park, CA, 1983.

Sérasset G. Recent trends of electronic dictionary research and development in europe. Technical report, Electronic Dictionary Research (EDR), Japan, 1993.

Seth A., Gala S., and Navathe S. On automatic reasoning for schema integration. *Int. Journal of Intelligent and Cooperative Information Systems*, 2(1):23–50, 1993.

Seth A.P. and Larson J.A. Federated database systems for managing distributed, heterogeneous and autonomous databases. *ACM Computing Surveys*, 22(3), 1990.

Shanmugasundaram J., Tufte K., Zhang C., He G., DeWitt D.J., and Naughton J.F. Relational databases for querying XML documents: Limitations and opportunities. In Atkinson M.P., Orlowska M.E., Valduriez P., Zdonik S.B., and Brodie M.L., editors, *VLDB'99, Proceedings of 25th International Conference on Very Large Data Bases, September 7-10, 1999, Edinburgh, Scotland, UK*, pages 302–314. Morgan Kaufmann, 1999. ISBN 1-55860-615-7.

Shieber S.M. An introduction to unification-based approaches to grammar. Lecture Notes 4, Center for the Study of Language and Information, 1986.

Shieber S. *Constraint-Based Grammer Formalisms*. MIT Press, Cambridge/ MA, 1992.

Shmueli O. Decidability and expresiveness aspects of logic queries. In *Proc. of the 6th ACM Symposium on Principles of Database Systems*, pages 237–249, 1987.

Simmons G.F. The nature of linguistic data and the requirements of a computing environment for linguistic research. In Lawler J. and Dry A., editors, *Using Computers in Linguistics – A practical guide*, pages 10–25. Routledge, London, 1998.

Simons G.F. Conceptual modeling versus visual modeling: a technological key to building consensus. *Computers and the Humanities*, (30):303–319, 1997.

Sowa J.F. *Conceptual structures: information processing in mind and machine*. Addison-Wesley Longman Publishing Co., Inc., 1984. ISBN 0-201-14472-7.

Sperber-McQeen C.M. and Burnard L., editors. *TEI P4: Guidelines for Electronic Text Encoding and Interchange*. Oxford, 4 edition, 2002. URL `http://www.tei-c.org/P4X/`.

Sperberg-McQueen C.M. and Burnard L. *Guidelines for Electronic Text Encodings and Interchange*. Chicago & Oxford, 1995.

Studer R., Benjamins V.R., and Fensel D. Knowledge engineering, principles and methods. *Data and Knowledge Engineering*, 25(1-2):161–197, 1998.

Suciu D. Web data and the resurrection of database theory. In *Eleventh International Workshop on Research Issues in Data Engineering (RIDE' 01)*, pages 3–3, Washington - Brussels - Tokyo, April 2001. IEEE. ISBN 0-7695-0957-6.

Sycara K., Lu J., and Klusch M. Interoperability among heterogeneous software agents on the internet. Technical report, Carnegie-Mellon University, Pittsburgh, USA, 1998.

Tejada S., Knoblock C.A., and Minton S. Learning object identification rules for information integration. *Special Issue on Data Extraction, Cleaning,*

and Reconciliation Information Systems Journal, pages 607–633, December 2001.

Tsichritzis D. and Klug A. The ANSI/X3/SPARC DBMS frameword. *Information Systems*, 3(4), 1978.

Ullman J.D. *Principles of Database & Knowledge-Base Systems Vol. 2:The New Technologies*. Computer Science Press, 1989.

Ullman J.D. Information integration using logical views. In Afrati F.N. and Kolaitis P., editors, *Database Theory—ICDT'97, 6th International Conference*, volume 1186 of *Lecture Notes in Computer Science*, pages 19–40, Delphi, Greece, 8–10 January 1997. Springer.

Uszkoreit H., Backofen R., Calder J., Capstick J., Dini L., Dörre J., Erbach G., Estival D., Manandhar S., Mineur A.M., and Oepen S. The EAGLES formalisms working group - final report expert advisory group on language engineering standards. Technical Report LRE 61-100, 1996.

Valverde A. Integración de la información: Una arquitectura basada en wrappers. *Karrera bukaerako Proiektua. Informatika Fakultatea. Euskal Herriko Unibertsitatea.*, 2003.

Vassalos V. and Papakonstantinou Y. Expressive capabilities description languages and query rewriting algorithms. *Journal of Logic Programming*, 43(1):75–122, 2000.

Vossen P. EuroWordNet: a multilingual database for information retrieval, 1997.

Wache H., Scholz T., Stieghahn H., and König-Ries B. An integration method for the specification of rule-oriented mediators. In Kambayashi Y. and Takakura H., editors, *Proc. of the International Symposium on Database Applicatons in Non-Traditional Environments (DANTE'99)*, pages 109–112, Kyoto, Japan, 1999.

Wache H. Towards rule-based context transformation in mediators. In Conrad S. and anf G Saake W.H., editors, *International Workshop on Engineering Federated Information Systems (EFIS 99)*. Infix-Verlag, Kühlungsborn, Germany, 1999.

Wache H., Vögele T., Visser U., Stuckernschmidt H., Schuster G., Neumann H., and Hübner S. Ontology-based integration of information — a survey of existing approaches. In *Proc of IJCAI-01 Workshop: Ontologies and Information Sharing*, pages 108–117, Seattle, USA, 2001.

Walker D., Zampolli A., and Calzolari N. *Automating the Lexicon: Research and Practice in a Multilingual Environment*. Oxford University Press, 1994.

Weber D.J. Reference grammars for the computational age. *Notes on Linguistics*, 33:28–38, 1986.

Weld D.S. Recent advances in planning. *AI Magazine*, 20(2):93–123, 1999.

Whitelock P., Somers H., Bennet P., Johnson R., and Wood M. *Linguistic Theory and Computer Applications*. Academic Press, New York, 1987.

Wickler G. and Tate A. Capability representations for brokering: A survey. Avaiable at http://www.aiai.ed.ac.uk/~oplan/cdl/, 1998.

Wiederhold G. Knowledge and database management. *IEEE Software*, 1(1): 63–73, January 1984. ISSN 0740-7459.

Wiederhold G. Knowledge versus data. In *Brodie, Mylopoulos, and Schmidt (eds) 'On Knowledge Base Management Systems: Integrating Artificial Intelligence and Database Technologies,' Springer Verlag (Heidelberg, FRG and NewYork NY, USA)*. February 1986.

Wiederhold G. Mediators in the architecture of future information systems. *IEEE Computer*, 25(3):38–49, 1992.

Wiederhold G., editor. *Intelligent Inegration of Information*. Kluwer Academic Publisher, Boston, USA, 1996.

Wilks Y., Slator B., and Guthrie L. *Electronic Words: Dictionaries, Computers and Meanings*. The MIT Press. Cambridge, MA., 1998.

Wittenburg P., Peters W., and Drude S. Analysis of lexical structures from field linguistics and language engineering. In *LREC2002*, Las Palmas, Spain, 2002.

WWW Consortium. XML path language (XPath) version 1.0 – W3C recommendation. Available at http://www.w3.org/TR/xpath.html, 2000. URL `http://www.w3.org/TR/xpath.html`.

Yang H.Z. and Larson P.A. Query transformation for psj-queries. In *Proc. of the International Conference on Very Large Data Bases (VLDB)*, pages 245–254, Brighton, England, 1987.

Yokoi T. The EDR electronic dictionary. *Communications of the ACM*, 38 (11):42–44, November 1995.

Zajac R. Inheritance and constraint-based grammar formalisms. *Computational Linguistics*, 18(2):159–182, June 1992.

Zajac R. On some aspects of lexical standardization. In *ACL/SIGLEX99 - Standardizing Lexical Resources*, University of Maryland, June 21,22 1999.

Zhou G., Hull R., King R., and Franchitti J.C. Supporting data integration and warehousing using H2O. *IEEE Data Engineering Bulletin*, 18(2):29–40, June 1995.