

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Jaroslav Karpovič

**Semantinei paieškai naudojamos ontologijos
generavimo pagal duomenų bazės schemą
procesas**

Magistro darbas

Darbo vadovas
profesorius Rimantas Butleris

Konsultantas
doktorantas Vytautas Taujanskas

Kaunas
2007

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Jaroslav Karpovič

**Semantinei paieškai naudojamos ontologijos
generavimo pagal duomenų bazės schemą
procesas**

Magistro darbas

Recenzentas
doc. dr. Stasys Maciulevičius
2007-01-08

Darbo vadovas
profesorius Rimantas Butleris
2007-01-08

Atliko
IFM-1/4 gr. studentas
Jaroslav Karpovič
2007-01-08

Kaunas
2007

Turinys

IVADAS	5
1. DARBE NAUDOJAMŲ SAŲOKŲ IR SANTRUMPŲ PAAIŠKINIMAS	8
1.1. SAŲOKŲ PAAIŠKINIMAS	8
1.1.1. <i>Ontologija</i>	8
1.1.2. <i>Semantinis žiniatinklis</i>	8
1.1.3. <i>Kalba, programavimo kalba</i>	9
1.2. SUTRUMPINIMAI, TERMINAI.....	10
2. EGZISTUOJANČIŲ SEMANTINIŲ TECHNOLOGIJŲ ANALIZĖ	11
2.1. ANALIZĖS TIKSLAS	11
2.2. SEMANTINIŲ TECHNOLOGIJŲ APLINKOS ANALIZĖ	11
2.2.1. <i>Generatoriaus kontekstas</i>	11
2.2.2. <i>Semantinės paieškos aplinka</i>	12
2.2.3. <i>Generatoriaus bei paieškos naudotojai</i>	13
2.3. EGZISTUOJANTYS SPRENDIMAI IŠKELTAI PROBLEMAI SPREŠTI.....	14
2.3.1. <i>Semantinės paieškos pranašumai</i>	14
2.3.2. <i>Ontologijos panaudojimas paieškoje</i>	14
2.3.3. <i>Ontologijos gavimo būdai</i>	15
2.3.3.1. <i>Ontologijos ištraukimas iš nestruktūrizuoto teksto</i>	16
2.3.3.2. <i>Ontologijos generavimas pagal žodyną</i>	17
2.3.3.3. <i>Ontologijos generavimas iš žinių bazės</i>	18
2.3.3.4. <i>Ontologijos generavimas iš dalinai struktūrizuotų duomenų</i>	18
2.3.3.5. <i>Ontologijos generavimas iš reliacinės duomenų bazės</i>	18
2.3.4. <i>Ontologijos gavimo iš reliacinės duomenų bazės problemos</i>	19
2.3.4.1. <i>Egzistuojančių metodų trūkumai</i>	19
2.3.4.2. <i>Reliacinių DB pateikimas semantiniame žiniatinklyje</i>	19
2.3.4.3. <i>Siūlomas ontologijos generavimo metodas</i>	21
2.3.5. <i>Semantinės paieškos remiantis ontologija problemos</i>	23
2.3.5.1. <i>SWED semantinis tinklalapis</i>	23
2.3.5.2. <i>Geospatial Semantic MediaWiki</i>	26
2.3.5.3. <i>Siūloma semantinės paieškos sistema</i>	27
2.4. ARCHITEKTŪROS IR GALIMŲ ĮGYVENDINIMO PRIEMONIŲ VARIANTŲ ANALIZĖ	28
2.5. METODŲ REALIZUOJANČIŲ PROTOTIPŲ PRADINĖS SPECIFIKACIJOS	29
2.5.1. <i>Funkciniai reikalavimai sistemoms</i>	29
2.5.2. <i>Sistemų prototipų pradinės specifikacijos</i>	30
2.6. REIKALAVIMAI DUOMENIMS	31
2.7. NEFUNKCINIAI REIKALAVIMAI IR APRIBOJIMAI.....	32
2.8. RIZIKOS FAKTORIŲ ANALIZĖ.....	33
2.9. REZULTATO KOKYBĖS KRITERIJAI	33
2.10. ANALIZĖS IŠVADOS	34
3. GENERAVIMO IR PAIEŠKOS PROCESŲ BEI JUOS REALIZUOJANČIŲ PROTOTIPŲ SPECIFIKACIJOS	35
3.1. ONTOLOGIJOS IŠGAVIMAS: PROCESAS IR GENERATORIAUS PROTOTIPAS	35
3.1.1. <i>DB schemas pateikimas ontologijos pagalba</i>	35
3.1.2. <i>Ontologijos architektūra</i>	37
3.1.3. <i>Metaduomenų bazės struktūra</i>	39
3.1.4. <i>Ontologijos generatoriaus funkcijos</i>	40
3.1.5. <i>Ontologijos generatoriaus elgsena</i>	41
3.1.6. <i>Ontologijos generavimo taisyklės</i>	42
3.1.7. <i>Ontologijos generatoriaus architektūra</i>	43
3.2. SEMANTINĖ PAIEŠKA: PROCESAS IR PAIEŠKOS VARIKLIO PROTOTIPAS	43
3.2.1. <i>Semantinės paieškos funkcijos</i>	44
3.2.2. <i>Semantinės paieškos elgsena</i>	44
3.2.3. <i>Semantinės paieškos taisyklės</i>	46
3.2.4. <i>Semantinės paieškos sistemos architektūra</i>	46
3.2.4.1. <i>Apibendrinta paieškos sistemos architektūra</i>	47
3.2.4.2. <i>Paieška portalo reliacinėje duomenų bazėje</i>	47
3.3. SEMANTINIŲ SISTEMŲ VARTOTOJO SAŠAJA.....	50

3.3.1.	<i>Ontologijos generatoriaus vartotojo sąsaja</i>	50
3.3.2.	<i>Semantinės paieškos vartotojo sąsaja</i>	53
3.4.	PROJEKTO IŠVADOS	58
4.	EKSPERIMENTINIS GENERATORIAUS IR PAIEŠKOS TYRIMAS	60
4.1.	ONTOLOGIJOS GENERATORIAUS KOKYBĖS TYRIMAS.....	60
4.2.	SEMANTINĖS PAIEŠKOS SISTEMOS KOKYBĖS TYRIMAS.....	61
4.3.	TOLIMESNIO SISTEMŲ PLĖTOJIMO GALIMYBĖS	62
	IŠVADOS	63
	LITERATŪRA	64
	SUMMARY	67
	PRIEDAI	68
	1 PRIEDAS. RDF ONTOLOGIJA ATITINKANTI 21 PAV. PATEIKTĄ PAVYZDINĘ DB SCHEMĄ	68
	2 PRIEDAS. ONTOLOGIJOS ATITIKIMAS W3 KONSORCIUMO REIKALAVIMAMS BEI REKOMENDACIJOMS	72
	3 PRIEDAS. ONTOLOGIJOS GENERATORIAUS FUNKCIJŲ SPECIFIKACIJA	73
	4 PRIEDAS. SEMANTINĖS PAIEŠKOS SISTEMOS FUNKCIJŲ SPECIFIKACIJA	77
	5 PRIEDAS. PAVYZDINĖ METADUOMENŲ BAZĖ UŽPILDYTA DUOMENIMIS	84
	6 PRIEDAS. STRAIPSNIS „ONTOLOGIJŲ IŠGAVIMO ĮRANKIO ARCHITEKTŪRA“, J. KARPOVIČ, V.TAUJANSKAS.....	86

Įvadas

Semantinės technologijos [29], skelbiančios prasmės (sąvokų reikšmės) saugojimą atskirai nuo duomenų ir nuo taikomųjų programų kodo, įgyja privalumą leisdamos mašinoms lygiai taip pat kaip ir žmonėms suprasti informaciją bei pasidalinti šiuo supratimu realiame laike vykdymo metu. Šios technologijos taip pat įgalina pridėti naują duomenų šaltinį ar ryšį tarp taikomųjų programų taip paprastai kaip nubraižyti naują ryšį modelyje. Deja, šios technologijos iki šiol nėra tiek išvystytos, kad kasdieniame gyvenime pastebėtume ryškia jų naudą.

Semantinės paieškos naudojimas – tai bandymas parodyti semantinių technologijų pranašumus. Tokia paieška yra tikslesnė dėl savo paieškos srities susiaurinimo galimybių, pateikia konkretesnę rezultatą už įprastinę paiešką pagal raktinį žodį, ieškančią visoje duomenų bazėje. Šis privalumas išryškėja ypač tada, kai duomenų bazė yra didelė bei pripildyta dideliu kiekiu duomenų. Didelių indeksų failų panaudojimas neišsprendžia šios problemos.

Darbe siūlomas automatinis ontologijos generavimas pagal duomenų bazės schemą užtikrina, kad šiomis ontologijomis besinaudojanti semantinės paieškos sistema teiks savo paslaugas net tada, kai duomenų bazės struktūra pasikeis.

Darbo tikslas: automatiškai, be žmogaus įsikišimo sugeneruoti galutinę ontologiją, kurią naudojant būtų galima atlikti semantinę paiešką reliacinėje duomenų bazėje, kurios pagrindu ontologija buvo sugeneruota.

Darbe pateikiamas siūlomo generavimo proceso bei prototipinės sistemos, veikiančios šio proceso pagrindu, aprašas. Nagrinėjamu atveju pagrindinis tikslas yra ontologijos generavimas, o semantinė paieška – tai praktinis sugeneruotos ontologijos pritaikymas.

Darbo **tyrimo sritis** susideda iš dviejų giminingų ir dalinai persidengiančių sričių:

- ontologijos ir metaduomenys;
- semantinės paieškos sistemos.

Dėl priežasties, kad darbas susideda iš dviejų persidengiančių tyrimo sričių, galima išskirti du **tyrimo objektus**:

- automatinis ontologijos generavimas pagal duomenų bazės schemas bei papildomų metaduomenų informaciją;

¹ „Properly designed, the Semantic Web can assist the evolution of human knowledge as a whole“ [3]

- sugeneruotos ontologijos panaudojimas semantinėje paieškoje.

Sprendžiama problema. Dažnai įmonių bei pavienių asmenų informacija ir duomenys yra laikomi skirtinguose šaltiniuose. Tai gali būti pasenusios (angl. legacy) duomenų bazės, šiuolaikinės duomenų bazės, skirtingos struktūros bylos, duomenys gaunami per kitus komponentus ir panašiai.

Sudėtinga informacijos iš skirtingų duomenų šaltinių paieška skatina sukurti patogias vartotojui paieškos sistemas. Dažnai vien pagal raktinius žodžius atlikta paieška nesuteikia laukiamo rezultato, todėl siūloma praturtinti esamą informaciją papildoma semantine prasme bei suprojektuoti priemonės, leidžiančias ieškoti informacijos skirtingose šaltiniuose vienu metu, o iškilus būtinybei, pavišinti turimus duomenis pasauliniame tinkle – žiniatinklyje. Pažymėtina, kad paieška turi būti atliekama ne tik pagal raktinius žodžius, bet ir pagal semantinę duomenų prasmę. Semantinis duomenų praturtinimas įgalina pritaikyti paieškoje daugiakalbiškumą. Ontologijos sudarymas turimiems duomenims – vienas iš būdų duomenims suteikti papildomą semantinę prasmę.

Darbe nagrinėjamas duomenų, saugomų reliacinėje duomenų bazėje, semantinis praturtinimas – ontologijos generavimas bei sugeneruotos ontologijos panaudojimas semantinėje paieškoje. Semantinis duomenų praturtinimas bei jo pagrindu atliekama paieška yra semantinių technologijų dalis.

Automatinio galutinės ontologijos generavimo pagal duomenų bazės schemą bei papildomus meta-aprašus proceso ir įrankio architektūros pateikimas, o taip pat sugeneruotos ontologijos pagrindu veikiančios semantinės paieškos proceso ir variklio architektūros apibrėžimas bei specifikavimas sudaro **mokslinį darbo aspektą**.

Pasiūlyto ontologijos generatoriaus **naujumas** yra tas, kad jį naudojant generuojama galutinė (nereikalaujanti rankinio modifikavimo pasibaigus automatiniam generavimui) duomenų bazės schemos ontologija. Tokia generatoriaus realizacija užtikrina ontologijos atitikimą duomenų bazės schemai net tuomet, kai duomenų bazės struktūroje įvyksta pasikeitimai: tereikia tik papildyti (pakoreguoti) meta-aprašus ir inicijuoti generavimo procesą. Pati schemos ontologija tarnauja duomenų, esančių duomenų bazėje, semantiniam praturtinimui – šalia duomenų saugoma ir jų prasmė.

Pateikiamoje semantinėje paieškoje **nauja** yra tai, kad ji realizuoja galimybę patogiai atlikti paiešką skirtinguose šaltiniuose. Sugeneruotos ontologijos panaudojimas įgalina duomenų aprašų daugiakalbiškumo pritaikymą, o tai leidžia vartotojams naršyti ontologiją savo gimtąja (arba geriausiai žinoma iš pateikiamų) kalba nepriklausomai nuo to, kokia kalba (arba kokių santrumpų pagalba) sudaryti duomenų bazės schemos pavadinimai ar joje saugomi duomenys.

Duomenų bazėje saugomų duomenų semantinis praturtinimas bei jų pateikimas semantinėms technologijoms – darbo **praktinė nauda**. Sugeneruota visus reikalavimus ir rekomendacijas [28] (žr. 2 priedas.) atitinkanti ontologija gali būti naudojama ne tik siūlomos semantinės paieškos variklio, bet ir būti kaip „žaliava“ kitoms semantinėms technologijoms. Siūlomų generavimo ir paieškos procesus realizuojančių įrankių naudojimas įgalina dabartines ar net pasenusias (angl. legacy) duomenų saugojimo technologijas be jokių modifikavimų pateikti ateities technologijoms – semantinėms technologijoms.

Šis darbas yra suskirstytas į dalis – skyrius. Pirmajame skyriuje pateikiami darbe naudojamų sąvokų, terminų ir santrumpų paaiškinimai padedantys vienareikšmiškai suprasti aprašymą. Antrajame skyriuje analizuojami realizuoti ar literatūroje aprašyti sprendimai iškeltai problemai spręsti, išryškinami jų privalumai ir trūkumai. Trečiasis skyrius apima siūlomų ontologijos generavimo ir paieškos procesų aprašus ir specifikacijas, o taip pat procesus realizuojančių sistemų architektūrinius sprendimus. Realizuotų generatoriaus ir paieškos variklio prototipų eksperimentinis tyrimas ir tolesnių generavimo ir paieškos procesų ir sistemų tobulinimų galimybės aprašytos ketvirtajame skyriuje. Darbo pabaigoje pateiktos darbo išvados, naudojamos literatūros sąrašas bei priedai.

1. Darbe naudojamų sąvokų ir santrumpų paaiškinimas

Naudojamų sąvokų vienareikšmiškam supratimui būtina pateikti žodyną. Ontologijų ir semantinių žiniatinklų sąvokos yra palyginus naujos, todėl nėra nusistovėjusių jų apibrėžimų. Šiame skyriuje įdėti dažniausiai pasitaikantys sąvokų apibrėžimai.

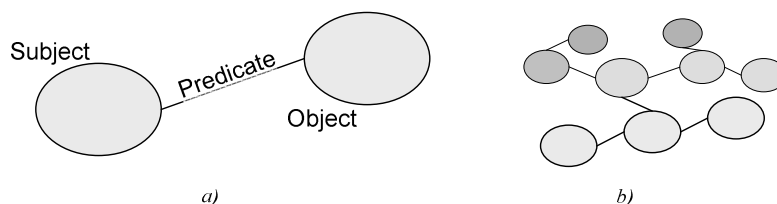
1.1. Sąvokų paaiškinimas

1.1.1. Ontologija

Ontologija – tiksli ir detali tam tikros srities objektų, konceptų, esybių ir jų ryšių vaizdavimo formalioji specifikacija [22].

Ontologija - tai metamodelis, nusakantis dalykinės srities subjektus, jų savybes bei ryšius tarp jų. Ji suteikia mašinoms trūkstamas bazines žinias apie dalykinę sritį [22].

Ontologija – tai bandymo apibrėžti išsamią ir griežtą srities konceptualiąją schemą darbo rezultatas. Dažniausiai tai hierarchinė duomenų struktūra, kurioje yra susijusios esybės su jas siejančiais ryšiais bei nagrinėjamos dalykinės srities taisyklės [29].



1 pav. RDF tripletas bei tripletų tinklas

Pats paprasčiausias ontologijos vaizdavimas yra RDF notacija, siūloma W3C konsorciumo [28]. RDF (Resource Description Framework) – universali informacijos vaizdavimo žiniatinklyje kalba. RDF – tai tripletų (*<subject>* *<predicate>* *<object>*) tinklas (1 pav.). Tripletas – tai dvi klasės (*<subject>*, *<object>*) ir jas siejantis ryšys (*<predicate>*), kuris kartais vadinamas tripleto savybe (*<property>*) [28].

1.1.2. Semantinis žiniatinklis

Žiniatinklis (angl. *Web*, *WWW*, *World Wide Web*) – „hipertekstinės informacijos visuotinis tinklas, svarbiausias interneto komponentas. Po jį galima naršyti naudojantis saitais, naudotis paieškos sistemomis reikalingai informacijai rasti, parsisiųsdinti tinklalapius, rinkmenas, atlikti įvairius kitus darbus, susijusius su duomenų mainais“ – tai Enciklopedinio kompiuterijos žodyno [7] aiškinimas.

Apibendrinant, *žiniatinklis* (pasaulinis tinklas) - interneto dalis, resursai, kuriuos internete galima pasiekti naudojant URI (vieningus resursų identifikatorius). Dėl savo

naudojimo platumo ši sąvoka dabar yra neretai supainiojama su visu internetu, tačiau tai yra tik interneto poaibis. Pasaulinis tinklas daugiausiai remiasi hipertekstu, taigi ir HTTP protokolu bei HTML kalba.

Semantinis žiniatinklis - technologijų visuma, suteikianti žiniatinkliui daugiau prasmės, padarant jį naudingesniu žmonėms. Terminas apibūdina ateities interneto tinklą kaip globalią duomenų bazę. Jį pirmą kartą panaudojo Tim Berners-Lee [3]. Semantinio tinklo infrastruktūra leidžia tiek žmonėms, tiek mašinoms daryti efektyvesnius informacijos panaudojimo sprendimus. Architektūriniai elementai apima semantiką (elementų pavadinimus), struktūrą (elementų hierarchiją) ir sintaksę (bendravinimą).

Semantinis žiniatinklis – globali informacinė erdvė, susidedanti iš resursų, susietų tarpusavio ryšiais [24, 5].

Semantinio žiniatinklio paskirtis - automatizuoti tinklalapio resursų apdorojimo procesus, panaudoti tinkamas technologijas ir standartus, skirtus palengvinti mašinų darbą, t.y. padėti geriau joms suprasti informaciją, esančią tinklalapyje, tokiu būdu užtikrinant efektyvesnę paiešką, duomenų integravimą, navigaciją, užduočių automatizavimą [22].

1.1.3. Kalba, programavimo kalba

Darbe dažnai sutinkama „kalbos“ sąvoka. Nedviprasmiškam supratimui sutariama dėl sąvokų naudojimo:

- kalba – naudojama žmonių kalbai pažymėti (pvz. lietuvių, lenkų, anglų, rusų). Ją bendraudami vieni su kitais naudoja žmonės [29];
- programavimo kalba – dirbtinė kalba, skirta aprašyti aparato (ypač kompiuterio) funkcionavimą [29].

1.2. Sutrumpinimai, terminai

Darbe naudojami sutrumpinimai, sąvokos ir terminai pateikti ir aprašyti 1 lentelėje.

1 lentelė. Sutrumpinimų, sąvokų ir terminų paaiškinimai

Sąvoka, santrumpa	Paaškinimas
API	(angl. Application Programming Interface) - taikomųjų programų programavimo sąsaja
DB	Duomenų bazė
DBVS	Duomenų bazių valdymo sistema
metaDB	Metaduomenų bazė (meta-aprašų bazė)
MRD	(angl. Machine Readable Dictionary) - mašinoms suprantami žodynai
NLP	(angl. Natural Language Processing) – natūralios kalbos apdorojimas
OWL	(angl. Web Ontology Language) – žymių kalba, naudojama duomenims semantiniame žiniatinklyje pateikti ir jais pasidalinti panaudojus ontologija
PA	Panaudojimo atvejis (angl. Use Case)
progr. (taikomoji progr.)	Programa (taikomoji programa)
RDF	(angl. Resource Description Framework) – žinių modeliavimo metodas; universali informacijos vaizdavimo žiniatinklyje kalba
URI	(angl. Uniform Resource Identifier) - vieningas resursų identifikatorius

2. Egzistuojančių semantinių technologijų analizė

Šioje dalyje apibrėžiamas nagrinėjamų sistemų kontekstas, analizuojami egzistuojantys metodai bei sukurtos sistemos, pabrėžiant jų privalumus ir trūkumus, siūlomi patobulinimai bei sudaromos pradinės specifikacijos.

2.1. Analizės tikslas

Pagrindinis darbo tikslas – ontologijos išgavimas, tačiau analizė čia sudėliojama šiek tiek kitokia tvarka tam, kad labiau išryškėtų kiekvieno žingsnio būtinumas. Analizė susideda iš šių punktų:

- semantinės paieškos apibrėžimas bei palyginimas su įprastine (raktažodžiais paremta) paieška;
- semantinės paieškos pritaikymo tam tikrai taikomajai sričiai, apribotai konkrečios reliacinės duomenų bazės ribomis, analizė;
- egzistuojančių semantinės paieškos sistemų bei prototipų analizė;
- ontologijos – semantinės paieškos būtinos dedamosios apibrėžimas bei jos išgavimo būdų analizė;
- ontologijos generavimo pagal srities reliacinės duomenų bazės schemas informaciją ypatybių analizė.

2.2. Semantinių technologijų aplinkos analizė

Darbo pabaigoje bus gautas rezultatas, kuris susideda iš dviejų dalių: generatoriaus sugeneruota ontologija ir tos ontologijos pagrindu veikianti semantinės paieškos sistema. Dėl šios priežasties atskirų dalių aplinkas reikėtų nagrinėti atskirai – jos nėra tapačios ir turi skirtingus vartotojus (vartotojų grupes).

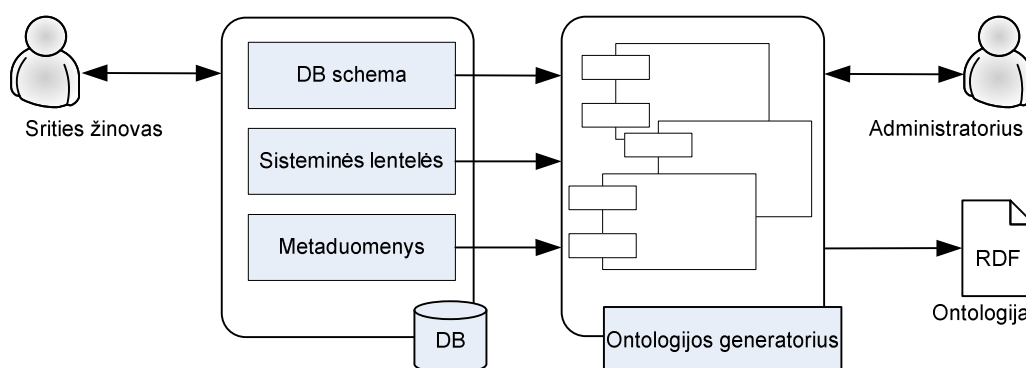
Nėra apibrėžtos srities, kurioje gali būti taikoma siūloma sistema, todėl aplinką geriausiai analizuoti nagrinėjant komponentus (generatorius, paieškos sistema) kaip „juodąsias dėžes“.

2.2.1. Generatoriaus kontekstas

Generatorius neturi jokių sąsajų su kitomis programomis ar komponentais (2 pav.), išskyrus sąsają su duomenų baze (DB) bei išvesties sąsają į rezultatų bylą, kuri ir yra sugeneruota ontologija.

RDF tipo ontologija sudaroma nuskaičius iš duomenų bazės jos struktūrą aprašančius šaltinius: pačią schemą, sisteminėse lentelėse saugomą informaciją bei metaduomenų bazėje

srities žinovo įvestus duomenis, kurie papildo informaciją (2 pav.).



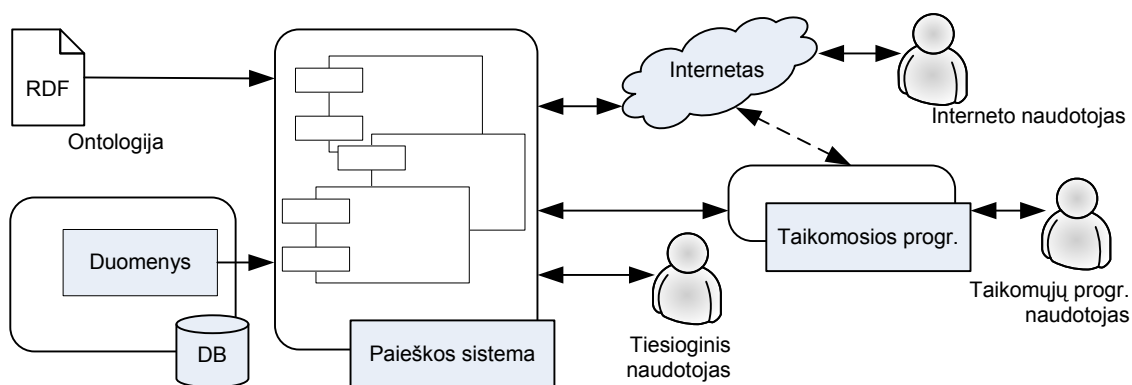
2 pav. Ontologijos generatoriaus kontekstas

Generatoriaus naudotojai detaliam aprašyti 2.2.3 skyrelyje „Generatoriaus bei paieškos naudotojai“.

2.2.2. Semantinės paieškos aplinka

Paieškos sistema naudoja generatoriaus sugeneruotą ontologiją duomenims iš duomenų bazės (tos pačios, pagal kurią buvo sugeneruota naudojama ontologija) semantiniam praturtinimui (3 pav.).

W3 konsorciumo standartus [28] atitinkančios ontologijos panaudojimas (žr. 2 priedas.) taip pat įgalina pateikti turimą informaciją semantiniame žiniatinklyje arba apdoroti išorinių taikomųjų programų, prijungtų tiesiogiai prie paieškos sistemos sąsajų arba panaudojus internetines technologijas (3 pav., brūkšniuota sąsajos rodyklė).



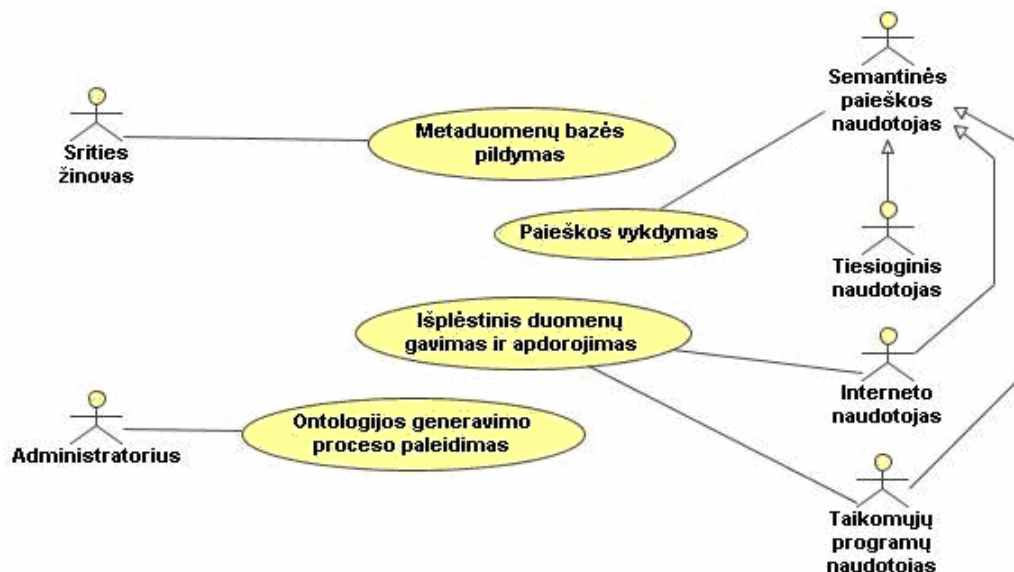
3 pav. Semantinės paieškos kontekstas

3 pav. pavaizduotoje diagramoje sąvoka „Internetas“ reiškia žiniatinklį su semantinėmis technologijomis – semantinį žiniatinklį [3], galintį saugoti, apdoroti ir perteikti pateikti ontologijose saugomą informaciją.

Paieškos sistemos naudotojai detaliam aprašyti 2.2.3 skyrelyje „Generatoriaus bei paieškos naudotojai“.

2.2.3. Generatoriaus bei paieškos naudotojai

Visi sistemų naudotojai yra pateikti sistemų kontekstinėse diagramose (2 pav., 3 pav.) bei apibendrintoje veiklos diagramoje (4 pav.).



4 pav. Dalykinės srities naudotojai bei jų veiklos

Ontologijos generatoriaus bei semantinės paieškos sistemos naudotojai realiai gali būti tie patys, bet jie sudaro skirtingas grupes, todėl šiame skyriuje detalios aprašyta kiekviena naudotojų grupė:

Srities žinovas (2 pav., 4 pav.) – tai naudotojas, daug žinantis apie duomenų bazėje pateiktą sritį, t.y. šios srities ekspertas. Jo pagrindinė funkcija – tiesiogiai duomenų bazėje arba per tam tikras formas užpildyti duomenų bazės aprašų (metaduomenų) bazę. Srities žinovas aprašo duomenų bazę ta kalba, kuria norima naudoti sugeneruotą ontologiją. Pažymėtina, kad ta pati bazė gali būti aprašyta keliomis kalbomis (lietuvių, anglų, vokiečių, rusų ir t.t.). Nuo schemas aprašų kokybės priklauso sugeneruotos ontologijos pilnumas, jos semantinis „turtas“.

Administratorius (2 pav., 4 pav.) – tai naudotojas, žinantis apie visus atliekamus duomenų bazės struktūros pakeitimus ir paleidžiantis generavimo procesą po kiekvieno jų. Ontologijos generavimas – pagrindinė jo funkcija.

Paieškos sistemos **naudotojai** yra skirstomi į tris tipus (3 pav., 4 pav.) pagal prieigą prie duomenų ir prie pačios sistemos. Jie skirstomi į:

- **tiesioginis naudotojas** – naudotojas, turintis tiesioginę prieigą prie paieškos sistemos sąsajos ir gaunantis paieškos rezultatus. Paieškos sistemos sąsaja nėra labai išplėsta ir neturi papildomų duomenų apdorojimo funkcijų;
- **taikomųjų programų naudotojas** (Taikomųjų progr. naudotojas, 3 pav.), prie semantinių žinių (ontologija ir duomenys) prieinantis išorinių taikomųjų

programų pagalba. Taikomosiuose programose gali būti realizuoti ne tik paieškos moduliai, bet ir kitokios duomenų apdorojimo funkcijos (kategorizavimas, įvertinimas ir pan.). Šios programos gali būti prijungtos ir tiesiogiai prie paieškos sistemos sąsajų, ir naudotis semantinio žiniatinklio teikiamomis paslaugomis, kadangi standartus atitinkančių [28] ontologijų panaudojimas įgalina pasidalinti turimomis žiniomis;

- **interneto naudotojas** prie sistemos teikiamų paslaugų prieina pasinaudojęs semantinio žiniatinklio teikiamomis paslaugomis. Iš esmės šis naudotojas yra beveik tapatus taikomųjų programų naudotojui, pasiekiančiam duomenis semantinio žiniatinklio pagalba.

2.3. Egzistuojantys sprendimai iškeltai problemai spręsti

Šiame skyriuje aptariami semantinės paieškos pranašumai prieš raktažodžiais paremtą paiešką, semantinei paieškai būtinos ontologijos gavimo būdai ir problemos, palyginamos egzistuojančios semantinės paieškos sistemos bei pasiūlomi tam tikri patobulinimai.

2.3.1. Semantinės paieškos pranašumai

Semantinė paieška pranašesnė už standartinę paiešką, surandančią informaciją pagal vartotojo įvestus raktinius žodžius, tuo, kad ji pateikia labiau vartotojo reikalavimus atitinkančius rezultatus. Ne tik raktažodžiais, bet ir semantine prasme paremtos paieškos rezultatai ne būtinai yra nuorodos į tekstinį šaltinį - tai gali būti nuoroda į garsinę bylą, video filmą ar paveiksluką, kuris buvo tinkamai aprašytas.

Semantinė paieška gali veikti dvejopai: ji gali pateikti platųjį rezultatą (ieškodama pagal raktinius žodžius bei semantinę prasmę) arba atvirkščiai, leidžia susiaurinti sritį, vartotojui nustačius semantinių žinių apribojimus. Kaip jau buvo minėta, paieškos rezultatas tiesiogiai priklauso nuo semantinių aprašų kokybės.

Vienas iš būdų praturtinti duomenis semantine informacija – sukurti ontologiją.

2.3.2. Ontologijos panaudojimas paieškoje

Ontologijos (sąvoka paaiškinta 1.1.1 skyrelyje) naudojimas leidžia susisteminti duomenis, suteikti jiems papildomą semantinę prasmę. Netgi pačios paprasčiausios ontologijos, išreikštos RDF kalbos tripletais, panaudojimas leidžia patogiai susieti duomenis tarpusavyje, apibrėžti duomenų prasmę bei apribojimus jiems, užrašyti primityvias taisykles tarp duomenų.

Dažnai šiuolaikiniai duomenų „šaltiniai“ turi tarptautinį charakterį – juose realizuotas

daugiakalbiškumas. Ontologija įgalina tokių sistemų paieškoje taip pat įvesti daugiakalbiškumą. Tam reikia tik aprašyti duomenis pageidaujamos kalbomis. Toks daugiakalbis aprašas turi privalumą, kad vartotojas randa informaciją pageidaujama kalba, o ir paieškos parametrų įvedimas gali būti atliekamas naudotojui geriausiai žinoma kalba.

Semantiniai duomenų aprašai ontologijų pavidalu yra pranašesni ir tuo, kad prisilaikant standartais apibrėžtos ontologijos struktūros [28] (žr. 2 priedas.), galima pasidalinti turimais duomenimis išorinių taikomųjų programų arba semantinio žiniatinklio pagalba. Tokiu būdu turima ontologija gali būti prijungta prie jau sukurtų nagrinėjamos dalykinės srities ontologijų arba atvirkščiai – paieška gali būti atliekama ne tik „nuosavoje“ ontologijoje, bet ir pasinaudojus semantinio žiniatinklio technologijomis pateikta ontologija.

Ontologijos nauda matoma ne tik semantinėje paieškoje, ji – kertinis akmuo viso semantinio žiniatinklio, tačiau, ji turi būti sudaryta. Ontologija dažniausiai sudaroma tam tikrai sričiai. Bendrinės, viską apimančios ontologijos praverčia tik tam tikrais, išskirtiniais atvejais.

2.3.3. Ontologijos gavimo būdai

Rankinis ontologijos sudarymas turi tą privalumą, kad sudaroma labai tiksli ontologija – žinių modelis. Tačiau toks ontologijos sudarymas reikalauja labai daug išteklių, tiek laiko, tiek žmogiškųjų. Be to, tokios ontologijos kokybė tiesiogiai priklauso nuo ją sudarančio asmens žinių, jo išprusimo ontologijos aprašomoje srityje.

Praktikoje dažniau yra taikomi automatiniai ontologijų išgavimo būdai su tolesniu gauto rezultato (ontologijos) patobulinimu, pakoregavimu, praplėtimu. Automatiniais būdais sugeneruota ontologija, nors ir ne visada turi rankinių būdu sukurtos ontologijos tikslumą, gaunama daug pigiau ir greičiau.

Egzistuoja skirtingi ontologijos išgavimo būdai. Priklausomai nuo to, iš kokio duomenų (ar informacijos) šaltinio generuojama, skirstoma į:

- ontologijos ištraukimas iš nestruktūrizuoto teksto;
- ontologijos generavimas pagal žodyną;
- ontologijos generavimas iš žinių bazės;
- ontologijos generavimas iš dalinai struktūrizuotų duomenų šaltinių;
- ontologijos generavimas iš reliacinės duomenų bazės.

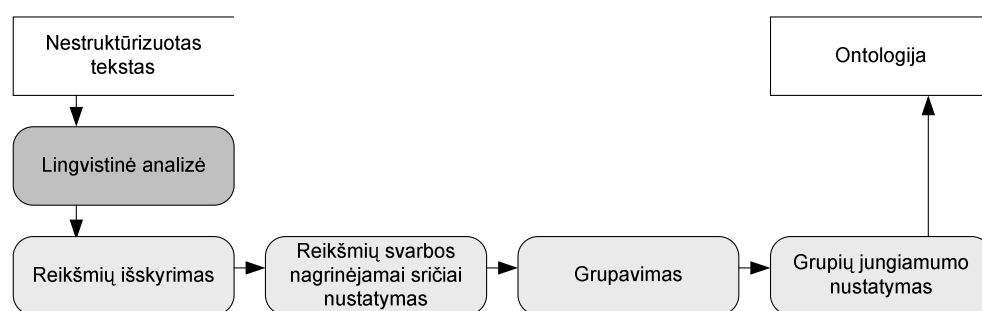
Kiekvieno ontologijos išgavimo (generavimo) būdo ypatumai aptarti sekančiuose poskyriuose.

2.3.3.1. Ontologijos ištraukimas iš nestruktūrizuoto teksto

Išvystyta nemažai automatinio ontologijos išgavimo iš laisvo teksto metodų. Dalį jų bent iš dalies realizuoja tam tikri ontologijos išskyrimo įrankiai, kita dalis – tik teoriniai [17].

Bendru atveju ontologijos išskyrimo procesas yra toks (5 pav.): iš teksto išskiriamos reikšmės, nustatoma jų svarba nagrinėjamai sričiai, išskirtos reikšmės sugrupuojamos bei nustatomas grupių jungiamumas [3, 4, 8, 12, 14, 9].

Išskyrimo proceso priešaky įterptas lingvistinės analizės blokas. Tai sudėtingą vidinę sandarą turintis blokas, dažnai atskiras posistemis ar net sistema. Lingvistinės analizės blokas mums suteikia žinias apie žodžio formą ir leidžia tas žinias „suvienareikšminti“ – gauti nedviprasmišką informaciją.



5 pav. Klasikinis ontologijos išskyrimo procesas

Labiausiai paplitusių metodų pagrindiniai bruožai trumpai aprašyti žemiau. Pažymėtina, kad dažniausiai metodai pavadinti autorių, kurie pasiūlė metodą, pavardėmis.

Alfonseca and Manandhar [8]. Šis metodas paremtas automatinio žodžių, esančių šalia vienas kito sąvokų rinkinyje, kontekstinių savybių išgavimu. Jis gali būti naudojamas jau esamos ontologijos sąvokų grupavimui arba jos papildymui naujomis sąvokomis. Principas yra pagrįstas hipoteze, kad žodžio reikšmė yra labai susijusi su kontekstu, kuriame tas žodis pasirodo. Išskiriami raktiniai konteksto žodžiai, o likusios sąvokos vaizduojamos kaip susijusių žodžių su jų pasirodymo dažnių, rinkiniai. Šiems rinkiniams vėliau gali būti pritaikytas TFIDF arba chi-square statistinis metodas „atstumui“ tarp sąvokų nustatymui.

Aussenac-Gilles metodos [14] skirtas ontologijų generavimui; yra pagrįstas žinių išgavimu iš techninių dokumentų. Jis leidžia kurti srities modelį panaudojus natūralios kalbos apdorojimą (angl. Natural language processing, NLP) bei lingvistinius metodus. Pagrindinis dėmesys skiriamas tekstui, tačiau žinios taip pat gali būti išskiriamos panaudojus jau sukurtas ontologijas. Taip pat šis metodas leidžia išsaugoti nuorodas tarp modelio ir teksto, bet reikalauja žinių inžinieriaus įsikišimo į ontologijos generavimą. Metodas siūlo trijų lygių ontologijos išgavimą: lingvistinę analizę, normalizaciją ir formalų lygį. Pirmas lygis išrenka

sąvokas ir leksinius ryšius tarp jų. Normalizacijos lygyje šios sąvokos yra apjungiamos ir, galiausiai, formalizuojamos paskutiniame lygyje.

Gupta metodas [12] - skirtas išgauti ir palaikyti dalykinės srities WordNet žodyno-ontologijos porūšį. Iš dalykinės srities tekstų yra išrenkama tai sričiai būdinga leksika atmetant kitų dalykinių sričių reikšmes.

Hearst metodas [8]. Metodo esmė: žodžių (hiponimų) šablonų, išreiškiančių leksinius ryšius, panaudojimas naujų ryšių tarp konceptų ontologijoje nustatymui. Čia naudojamos WordNet tipo ontologijos bei teksto analizė žinių bazės išplėtimui.

Missikoff (OntoLearn) metodas [14]. Ontologijų generavimui ir praplėtimui skirtas metodas naudojant natūralios kalbos ir mašininio atpažinimo būdus. Kaip pradinis šaltinis čia yra naudojama bendrinė WordNet ontologija. Pirmiausia yra atmetamos sričiai nebūdingos sąvokos (angl. pruning). Po šio žingsnio naudojami statistiniai įverčiai (sąvokos priklausomybės sričiai nustatyti) ir semantinė interpretacija (tikrų sąvokų reikšmių ir ryšių tarp jų nustatymui).

Nobecourt metodas paremtas ontologijų išskyrimu iš teksto panaudojus natūralios kalbos (NLP) ir eksperto paruoštų srities tekstinių duomenų rinkinių (corpus) apdorojimu [8]. Metodas susideda iš dviejų veiklų: modeliavimo ir vaizdavimo. Modeliavimo veikla savo ruožtu susideda dar iš dviejų veiklų: lingvistinės ir konceptualiosios. Pagrindinis lingvistinės veiklos tikslas yra pagrindinių srities sąvokų išskyrimas, konceptualiosios – pagrindinių srities porūšių identifikavimas. Vaizdavimo veikla skirta modeliavimo schemos transformavimui į įgyvendinimo kalbą.

Atsižvelgus į aukščiau paminėtų metodų stipriausias savybes, buvo pasiūlyta ontologijos ištraukimo iš laisvo teksto architektūra [9] (žr. 6 priedas.).

2.3.3.2. Ontologijos generavimas pagal žodyną

Šis ontologijos išgavimo būdas paremtas mašinoms suprantamų žodynų (angl. MRD – machine readable dictionary) panaudojimu. Ontologijos išgavimas pagal žodyną dažniausiai nėra naudojamas atskirai, o tiesiog kaip papildoma priemonė kitais būdais išgaunamų ontologijų kokybei pagerinti. Mašinoms suprantamais žodynais gali būti ir tinkamai paruošti kompiuteriniai vienakalbiai arba daugiakalbiai žodynai, ir kitos bendrinės ontologijos, tokios kaip WordNet ar EuroWordNet [6, 16].

Pavyzdžiui, anksčiau minėtas ontologijų išgavimo iš laisvo teksto Hearst'o metodas naudoja WordNet žodyną-ontologiją generuojamos ontologijos patikrinimui bei jos kokybės pagerinimui.

Egzistuoja ir kiti metodai, tokie kaip Jannink ir Wiederhold [8, 12] arba Rigau [8],

tačiau į juos nebus gilinama, kadangi bendrinius ontologijas-žodynus turi tik kelios pasaulio kalbos, o šio darbo tikslas – sukurti daugiakalbę ontologiją.

2.3.3.3. Ontologijos generavimas iš žinių bazės

Ontologijos generavimo iš žinių bazės metodai pasinaudodami žinių bazėmis ir statistiniais įverčiais nustato sudaromos ontologijos taksonomiją. Tiksliau šioje vietoje reikėtų kalbėti ne apie metodus, bet apie vieną metodą, nes literatūroje buvo rastas tik vienas pažymėtas šiai kategorijai priskiriamas metodas – Suryanto ir Compton [8] metodas.

Panašiai, kaip ir ontologijos sudarymo pagal žodyną, taip ir generavimo iš žinių bazės metodai nėra paplitę ir naudojami vien dėl to, kad jiems vis dar nėra tinkamo darbo pagrindo – duomenų šaltinio.

2.3.3.4. Ontologijos generavimas iš dalinai struktūrizuotų duomenų

Ontologijų generavimo iš dalinai struktūrizuotų duomenų metodai dažniausiai skirti ne ontologijos sudarymui nuo nulio, bet egzistuojančios ontologijos praturtinimui. Dalinai struktūrizuotais duomenimis gali būti XML bylos arba RDF tipo jau sukurtos ontologijos.

Dalis ontologijų išgavimo metodų remiasi grafų teorija (kaip pvz. Deitel [8]) arba specialiai sukurtais šablonais (pvz. Doan [8] metodas). Papildomus duomenis šių metodų ontologijos generavimui siūloma imti iš nestruktūrizuoto teksto arba iš bendrinių ontologijų, tokių kaip WordNet.

2.3.3.5. Ontologijos generavimas iš reliacinės duomenų bazės

Metodų, pristatytų šiame skyriuje, pagrindinis siekis – sukurti tarpinę schemą, leidžiančią ištraukti žinias iš reliacinės duomenų bazės ir jas apjungti.

Kashyap metodo autorius teigia [10], kad duomenų bazės žinių nešėju yra duomenų bazėje saugomi duomenys bei duomenų bazės schema. Šiuo metodu ontologija generuojama pagal duomenų bazės schemą visą laiką bendradarbiaujant su žmogumi, kuris turi nuspręsti dėl ištraukiamų esybių svarbos nagrinėjamai sričiai. Ontologijos sudarymas susideda iš dviejų žingsnių: DB schemos analizė ir vartotojo specifinių užklausų vykdymas ontologijos pagerinimui.

Pirmajame žingsnyje analizuojami tik pirminiai bei išoriniai duomenų bazės schemos raktai ir ryšiai tarp lentelių. Toliau sukuriama tarpinė schema, kurią vartotojas papildo sukurtomis užklausomis taip sukurdamas ontologiją. Pažymėtina, kad metodas skirtas apjungti kelias panašią sritį aprašančias duomenų bazes, tačiau labai daug tenka nuspręsti vartotojui.

Rubin'o metodas [20] siūlo pildyti jau egzistuojančią kuklią ontologiją duomenimis

iš duomenų bazės pasinaudojus tam tikra atvaizdavimo sąsaja (XML tipo byla) tarp duomenų bazės ir ontologijos. Šis metodas nesiūlo sudarinėti ontologijos, o tik praturtinti jau egzistuojančią.

Stojanovic pasiūlytas metodas [25] siūlo ištraukti iš duomenų bazės schemas tiek informacijos, kiek tik įmanoma, kad vėliau, pasinaudojus sudarytomis atvaizdavimo taisyklėmis, būtų galima sukurti kiek įmanoma platesnę ontologiją [8]. Sudarius pradinę ontologiją, ji yra užpildoma duomenimis iš reliacinės duomenų bazės.

2.3.4. Ontologijos gavimo iš reliacinės duomenų bazės problemos

Ontologijos išgavimas iš reliacinės duomenų bazės – šio darbo tikslas, todėl šiame skyriuje bus aptarti egzistuojančių metodų trūkumai ir netikslumai, apžvelgtos egzistuojančios reliacinių duomenų bazių pateikimo semantiniame žiniatinklyje sistemos.

2.3.4.1. Egzistuojančių metodų trūkumai

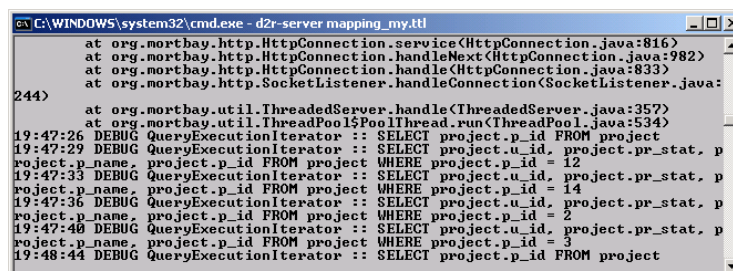
Visi aukščiau išnagrinėti ontologijos generavimo iš reliacinės duomenų bazės metodai siūlo saugoti duomenis tiesiogiai ontologijoje. Taip sudaryta ontologija tampa statinė, t.y. nesikeičia papildžius duomenų bazei naujais duomenimis.

Suprantama, kad galima kas kažkiek laiko pergeneruoti ontologiją, tačiau tada susiduriama su antru trūkumu – visais metodais sugeneruota ontologija yra koreguojama, taisoma ir praturtinama žmogaus, o tai reiškia, kad po kiekvieno generavimo reikės iš naujo atlikti rankinį ontologijos koregavimą.

2.3.4.2. Reliacinių DB pateikimas semantiniame žiniatinklyje

Vienas iš reliacinės duomenų bazės paviešinimo žiniatinklyje būdų pristatytas 2006-aisiais metais „15th World Wide Web Conference“ konferencijoje – *D2R Server* projektas [5]. Jo autoriai siūlo gauti ontologiją pagal duomenų bazės schemą panaudojus informacijos apie duomenų bazę (o tiksliau apie jos schemą) XML byla ir tarpinio sluoksnio (serverio) pagalba. Susietumo (ang. mapping) bylos fragmento vaizdas pateiktas 7 pav.

Serveris pagal vartotojo pasirinktus resursus generuoja paprastas SQL kalbos SELECT užklausas (6 pav.) ir grąžina rezultataž vartotojui HTML pavidalu (8 pav.).



```
C:\WINDOWS\system32\cmd.exe - d2r-server mapping_my.ttl
at org.morthay.http.HttpConnection.service(HttpConnection.java:816)
at org.morthay.http.HttpConnection.handleNext(HttpConnection.java:982)
at org.morthay.http.HttpConnection.handle(HttpConnection.java:833)
at org.morthay.http.SocketListener.handleConnection(SocketListener.java:
244)
at org.morthay.util.ThreadedServer.handle(ThreadedServer.java:357)
at org.morthay.util.ThreadPool$PoolThread.run(ThreadPool.java:534)
19:47:26 DEBUG QueryExecutionIterator :: SELECT project.p_id FROM project
19:47:29 DEBUG QueryExecutionIterator :: SELECT project.u_id, project.pr_stat, p
project.p_name, project.p_id FROM project WHERE project.p_id = 12
19:47:33 DEBUG QueryExecutionIterator :: SELECT project.u_id, project.pr_stat, p
project.p_name, project.p_id FROM project WHERE project.p_id = 14
19:47:36 DEBUG QueryExecutionIterator :: SELECT project.u_id, project.pr_stat, p
project.p_name, project.p_id FROM project WHERE project.p_id = 2
19:47:40 DEBUG QueryExecutionIterator :: SELECT project.u_id, project.pr_stat, p
project.p_name, project.p_id FROM project WHERE project.p_id = 3
19:48:44 DEBUG QueryExecutionIterator :: SELECT project.p_id FROM project
```

6 pav. SQL SELECT užklausų formavimas serveryje

```

mapping_my.ttl - Notepad
File Edit Format View Help
@prefix map: <file:/C:/d2r/mapping_my.ttl#> .
@prefix db: <> .
@prefix vocab: <vocab/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix d2rq: <http://www.wiwiiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#> .

map:database a d2rq:Database;
  d2rq:jdbcDriver "com.mysql.jdbc.Driver";
  d2rq:jdbcDSN "jdbc:mysql://isd.ktu.lt/rezerv";
  d2rq:username "jar1o0Buser";
  d2rq:password "jar5laptazodis";
  .

# Table project
map:project a d2rq:ClassMap;
  d2rq:dataStorage map:database;
  d2rq:uriPattern "project/@@project.p_id@@";
  d2rq:class vocab:project;
  .

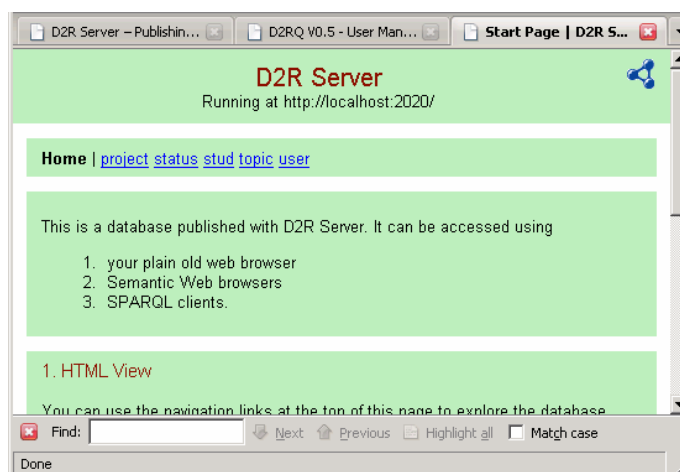
map:project_p_id a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:project;
  d2rq:property vocab:project_p_id;
  d2rq:column "project.p_id";
  d2rq:datatype xsd:short;
  .

map:project_u_id a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:project;
  d2rq:property vocab:project_u_id;
  d2rq:column "project.u_id";
  .

```

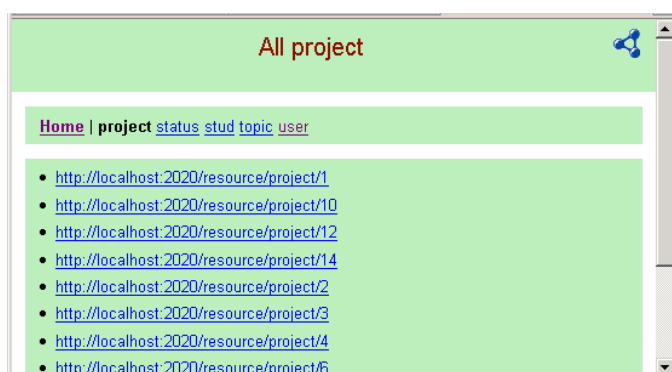
7 pav. Duomenų bazės susietumo bylos fragmentas

Ontologija nėra generuojama, tačiau pasinaudojus serverine dalimi, galima pasiekti kažką panašaus į ontologiją, bei ją naršyti naršyklėje (8 pav.). Serveris atlieka tarpinio sluoksnio tarp duomenų bazės ir vartotojo rolę.



8 pav. D2R serverio duomenų bazės pateikimas semantiniame žiniatinklyje

Naršyklėje atidarius pavišintą duomenų bazę ir pasirinkus vieną iš jos lentelių (nuorodų, atitinkančių pavišintos duomenų bazės lentelių vardus), visi pasirinktos lentelės įrašai pateikiami nuorodų (9 pav.) arba RDF pavidalu (10 pav.).



9 pav. Pasirinktos lentelės įrašai pateikti nuorodų pavidalu

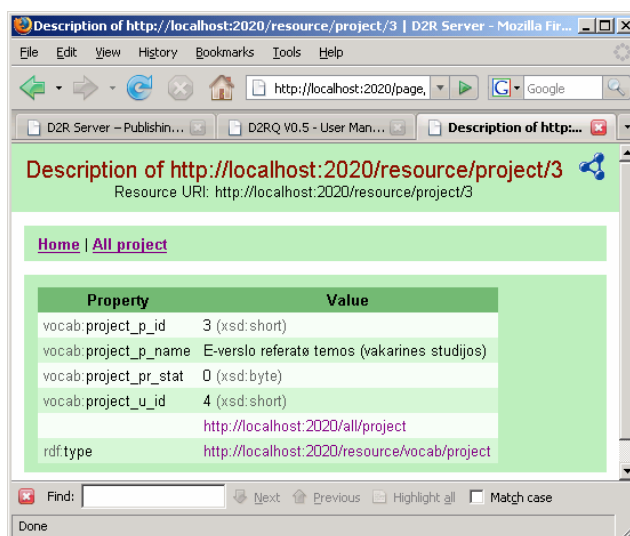
```

Source of: http://localhost:2020/all/project - Mozilla Firefox
File Edit View Help
<rdf:RDF
  xmlns:db="http://localhost:2020/resource/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:vocab="http://localhost:2020/resource/vocab/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >
  <rdf:Description rdf:about="http://localhost:2020/resource/project/9">
  <rdf:type rdf:resource="http://localhost:2020/resource/vocab/project"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost:2020/resource/project/6">
  <rdf:type rdf:resource="http://localhost:2020/resource/vocab/project"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost:2020/resource/project/14">
  </rdf:Description>
  </rdf:RDF>

```

10 pav. Pasirinktos lentelės įrašai pateikti RDF pavidalu

Pasirinkus atitinkamą įrašą atitinkančią nuorodą, pateikiama visa DB lentelės įrašo informacija (11 pav.).



11 pav. Pasirinkto DB lentelės įrašo informacija

Pastebimas šio projekto trūkumas – ryšių tarp duomenų bazės esybių (lentelių) nepalaikymas: visos lentelės ir jų įrašai yra tarsi atskiri, niekuo nesusiję elementai. Dar vienas trūkumas – duomenų bazės schema nepraturtinama jokia semantine informacija.

2.3.4.3. Siūlomas ontologijos generavimo metodas

Siūlomas metodas, generuojantis ontologiją ne tik pagal duomenų bazės schemas informaciją, bet ir pagal metaduomenų bazę, kurioje detaliau aprašytos duomenų bazės esybės, atributai bei ryšiai tarp jų. Automatiškai sugeneruota ontologija rankiniu būdu netaisoma. Jei sugeneravus pastebėti netikslumai, tuomet koreguojami metaduomenys ir paleidžiamas iš naujo ontologijos generavimo procesas.

Dar vienas skirtumas palyginus su prieš tai išnagrinėtais metodais – ontologija nesaugo duomenų. Ji yra tarsi duomenų bazės schema atitinkanti RDF schema. Duomenys gaunami panaudojus tarpinį sluoksnį (API) tarp ontologijos ir duomenų bazės.

Toks ontologijos generavimas turi šiuos privalumus:

- pasikeitus duomenų bazės struktūrai reikia pakeisti tik metaduomenis, aprašančius pasikeitusią vietą, ir pergeneruoti ontologiją iš naujo; generavimo procesas trunka neilgai;
- metaduomenimis galima aprašyti duomenų bazę skirtingomis kalbomis, nepriklausomai nuo to kokia kalba sudaryta duomenų bazės schema (tai ypač aktualu jei duomenų bazės lentelės ar atributai sudaryti iš santrumpų);
- panaudojus tarpinį sluoksnį, ontologija įgauna duomenų bazės dinamiškumo: pasikeitus duomenų bazės įrašams, vartotojui atrodo, kad ontologija automatiškai į juos reaguoja pasikeisdama;
- tarpinis sluoksnis gali atlikti taip pat saugumo posistemio funkcijas – skirtingiems vartotojams leisti prieigą tik prie jiems skirtų duomenų.

Taip sugeneruota ontologija gali būti panaudota semantinės paieškos reliacinėje duomenų bazėje panaudojus metaduomenų bazėje „sukauptą semantiką“.

Aukščiau pateiktų ontologijos gavimo iš reliacinės duomenų bazės metodų ir D2R projekto savybių palyginimas su siūlomu ontologijos generavimo procesu pateiktas 2 lentelėje. Šioje lentelėje „+“ ženklu žymimas teigiamos savybės išpildymas, „-“ ženklu – neišpildymas, o klausukas reiškia, kad literatūroje nerasta žinių apie konkrečios savybės įgyvendinimą.

2 lentelė. Apžvelgtų ontologijos gavimo iš duomenų bazės metodų ir projektų palyginimas

Metodas, projektas	Kashyap	Rubin	Stojanovic	D2R	Siūlomas metodas
Teigiama savybė					
Duomenų semantinis praturtinimas	?	?	?	-	+
Ontologija nereikalauja rankinio taisymo	-	-	-	+	+
Ontologija duomenimis užpildoma pasinaudojus programinėmis priemonėmis (dinamiška ontologija)	-	-	-	+	+
Galimybė įgyvendinti daugiakalbiškumą	?	?	?	-	+
Mažai žmogaus įsikišimo į generavimo procesą	-	?	?	+	+
Nereikia pildyti papildomų meta-aprašų	+	+	+	+	-
Ontologija palaiko ryšius tarp duomenų bazės esybių (lentelių)	+	?	?	-	+
Nereikalauja pradinės ontologijos	+	-	+	+	+

2.3.5. Semantinės paieškos remiantis ontologija problemos

Šiame darbo skyriuje išanalizuotos egzistuojančios semantinės paieškos sistemos bei duomenų bazių pateikimo semantiniame žiniatinklyje sistemos, išryškinti jų trūkumai bei pasiūlytas sprendimas.

2.3.5.1. SWED semantinis tinklalapis

SWED (The Semantic Web Environment Directory) [27] yra Europos Sąjungos remiamas projektas. Jis vykdomas Hewlett Packard laboratorijoje ir ILRT (Institute for Learning and Research Technology) Bristolyje, Didžiojoje Britanijoje. Pagrindinė jo idėja buvo sukurti internetinę sistemą, skirtą bendruomenės informacijos šaltinių (tinklapių) valdymui. Šios sistemos tikslai:

- pagerintas informacijos surinkimas;
- lengvas palaikymas ir praplečiamumas;
- lengvas daugkartinis informacijos panaudojimas (pvz. kituose tinklalapiuose);

SWED semantinio tinklalapio savybės pateikiamos 3 lentelėje.

Vienas iš paieškos privalumų yra tai, kad galima atlikti paiešką tarp prieš tai vykusios paieškos rezultatų (12 pav., „a“ sritis). Tačiau į paieškos laukelį iš karto įvesta paieška niekuo nesiskiria nuo raktiniais žodžiais paremtos paieškos. Vienintelis skirtumas tik tas, kad ieškoma ontologijoje.

3 lentelė. SWED savybių langas

Savybė	Aprašas
Agregavimas informacijos iš necentralizuotų šaltinių (kitų tinklalapių)	SWED siūlo kiekvienai organizacijai bendradarbiauti. Organizacija pateikia informaciją apie save savo tinklalapyje. SWED ją agreguoja ir tokiu būdu pateikia ją savo tinklalapyje.
Informacijos sindikavimas	SWED neapriboja turimos informacijos platinimo. Vis dėlto SWED dar turi keblumų su turimų RDF failų platinimu, ateityje jie tikisi sukurti tinkamą RDF failų pasiėmimo formą.
Semantinė paieška	Paiešką galima atlikti visame tinklalapyje arba jau gautuose rezultatuose. Informaciją apie organizaciją galima matyti skirtingais pjūviais: <ul style="list-style-type: none">• tekstiniu formatu;• pagal semantines nuorodas;• pagal pasirinktas semantines nuorodas.
Ontologijos (RDF failo) atnaujinimas	SWED pateikia redagavimo formą su informacija, prieš tai nurodant kelią iki RDF failo.
Klasifikavimas	Pateikiama informacija yra klasifikuota pagal: interesus, organizacijos tipą, veiklą, projekto tipą, geografinę sritį, pavadinimą.

12 pav. SWED portalo semantinės paieškos dialogas

Taip gautas rezultatas yra labai platus, kadangi pateikiami ne tik tie rezultatai, kuriuose egzistuoja ieškoma sąvoka, bet ir tie, kurie yra semantiškai susiję su ieškoma sąvoka turinčiais rezultatais.

13 pav. Raktažodžio paieška

Taip, pavyzdžiui, paieškos langelyje įvedus žodį „university“, gauname rezultatus skirtingose srityse (13 pav.): „Topic of interest“, „Organisation Type“, „Activity“ ir t.t. Tokiu būdu žmogus gali pasirinkti sau tinkamą sritį ir siaurinti paiešką.

Record 1 of 22 | [Back to Results](#) << | [Next](#) >>

ARKive
Type of Organisation:
URL: <http://www.arkive.org/>

ARKive is a not-for-profit initiative of The Wildscreen Trust (www.wildscreen.org.uk). It is the Noah's Ark for the Internet era - the world's centralised digital library of films, photographs and associated recordings of species, accessible to all via the world wide web. ARKive is leading the virtual conservation effort - finding, sorting, cataloguing and copying the key records of species, and building them into a comprehensive and enduring audio-visual record.

For each species, we are selecting and digitising an average of 10 minutes of moving footage, 6 - 10 stills and sound recordings (where appropriate), to compile a complete profile of the species' characteristic behaviour and appearance.

Founded in:

Contacts

Name: c/o The Wildscreen Trust
fax: +44 (0) 117 915 7105 Anchor Road
tel: +44 (0) 117 915 7100 Harbourside
 Bristol
 BS1 5TT
 England

Classifications

14 pav. Ontologijos naršymas

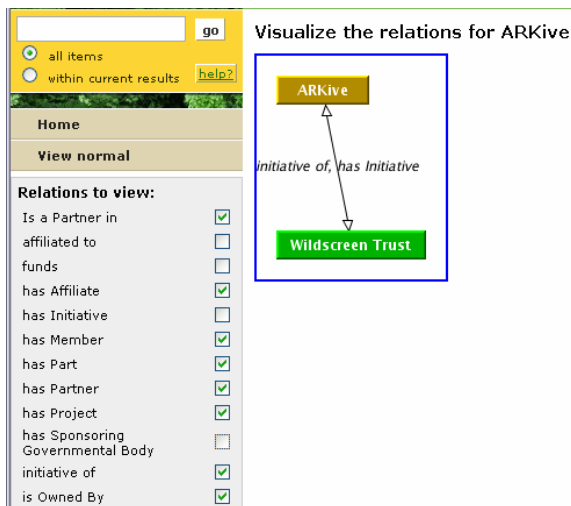
Dešinėje paieškos lango dalyje (12 pav., „b“ sritis) visas portalo turinys yra padalintas į šešias kategorijas. Iš esmės ši sritis atitinka ontologijos naršyklės dalį.

ARKive
 View all the properties of a knowledge item:
 ARKive is a [Project or Organisation](#)

initiative of	Wildscreen Trust
rdf:type	Project or Organisation
rdfs:label	ARKive
swed:has_activity_cl	Campaigning and Lobbying Education and Training Information Provision Information Related Organisational Support Type of Activity swed_at:conservation cultural heritage
swed:has_activity_type	Education and Training Information Provision swed_at:conservation cultural heritage
swed:has_care_of_proq_address	[]
swed:has_logo	http://www.arkive.org/images/page_arkivelogo.gif
swed:has_long_description	ARKive is a not-for-profit initiative of The Wildscreen Trust (www.wildscreen.org.uk). It is the Noah's Ark for the Internet era - the world's centralised digital library of films, photographs and associated recordings of species, accessible to all via the world wide web. ARKive is leading the virtual conservation effort - finding, sorting, cataloguing and copying the key records of species, and building them into a comprehensive and enduring audio-visual record. For each species, we are selecting and digitising an average of 10 minutes of moving footage, 6 - 10 stills and sound recordings (where appropriate), to compile a complete profile of the species' characteristic behaviour and appearance.

15 pav. Tiesioginis RDF vaizdas

Pasirinkus tam tikrą sritį atsiranda specializuotas meniu (14 pav., „a“ sritis). Pasirinkus jo punktus galime pamatyti norimos ontologijos šakos RDF struktūrą (15 pav.) arba vizualizavus ryšius, pasirinkti kurio tipo ryšiuose toliau ieškoti, o kuriuose nebe (16 pav.).

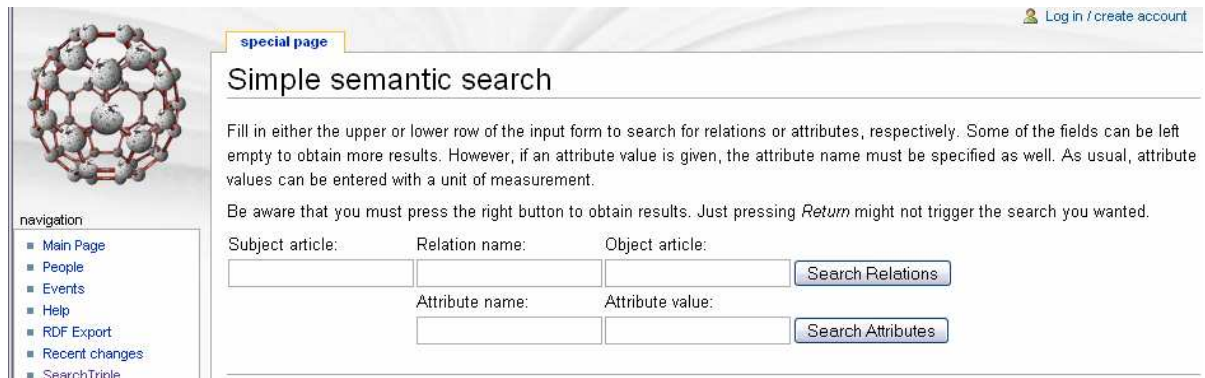


16 pav. RDF ryšių (nuorodų) pasirinkimas

Nėra žinių kaip yra vykdoma paieška ontologijoje. Žinoma tik, kad visi duomenys saugomi tiesiogiai RDF tipo ontologijoje.

2.3.5.2. Geospatial Semantic MediaWiki

Geospatial Semantic MediaWiki [23] – tai paprastas mėgėjiškas Wiki tipo puslapis, kuriame panaudotas ontologijų iš skirtingų semantinio žiniatinklio šaltinių apdorojimas bei sukurta labai paprasta, bet įdomi savo realizacijos ypatumais paieškos forma (17 pav.).



17 pav. Paprasta paieška ontologijos tripletuose

Nėra galimybės matyti visos ontologijos vaizdo, tačiau galima peržiūrėti atskirus ryšius. Vienas iš jų yra „Affiliation“ (18 pav.), nusakantis asmens darbovietę.

Affiliation	
Abraham Bernstein	IFI
Alain Leger	France Telecom
Alan Ruttenberg	Millennium Pharmaceuticals
Alberto Martelli	University of Torino, Italy
Aldo Gangemi	CNR
Aleksander Pivk	JSI
Alex Borgida	Rutgers, USA
Alexander Loeser	IBM Research Almaden
Alexv	University of the Aegean
Alicia Diaz	Lifa

18 pav. Ontologijos vieno ryšio duomenų apžvalga

Žinant ontologijos struktūrą, galima atlikti paiešką joje. 19 pav. pateiktas paieškos „visi žmonės, dirbantys JSI kompanijoje“ rezultatas.

Search results (relations)

Marko Grobelnik	Affiliation	JSI
Dunja Mladenic	Affiliation	JSI
Aleksander Pivk	Affiliation	JSI

19 pav. Paieškos rezultatas

Paieškos technologinis išpildymas nėra aprašytas.

2.3.5.3. Siūloma semantinės paieškos sistema

Apžvelgtos semantinės paieškos sistemos turi savo privalumų ir gerų idėjų, tačiau turi ir trūkumų.

Pavyzdžiui, SWED portalo vartotojo sąsaja yra labai nepatogi. Sunku surasti informacijos pirmą kartą naudojantis portalu, kadangi neaišku ką pažymėti. Taip pat neįmanoma įvesti skirtingų ryšių reikšmių. Šią neigiamą savybę dalinai kompensuoja galimybė ieškoti tarp jau rastų rezultatų, tačiau tai nėra tapatu, kadangi pateikiamos reikšmių aibės gali būti skirtingos.

Geospatial Semantic MediaWiki – tai prototipinis puslapis. Šio puslapio paieškos trūkumas yra tai, kad reikia matyti ontologijos struktūrą, kitaip paieška nepateiks jokie rezultato. Dar vienas trūkumas (kaip ir SWED puslapio paieškos), kad negalima įvesti apytiksliai reikšmių, t.y. žodžio fragmentų.

Siūloma paieškos sistema, naudojanti ontologiją ir tam tikrą duomenų šaltinį su galimybe atlikti paiešką ne tik pagal raktažodžius, bet ir pagal semantinę duomenų prasmę. Turi būti sukurta ontologijos naršyklė su galimybe įvesti tikslias arba apytikslias dominančių parametrų reikšmes, taip pat skaitinėms reikšmėms turi būti realizuota reikšmių palyginimo galimybė.

Ontologijos naršyklė atliks srities susiaurinimo vaidmenį ir, vartotojui benaršant, sudarinės taip vadinamą RDF užklausą. Sudarytą RDF užklausą tarpinio sluoksnio (tarp vartotojo ir duomenų šaltinio) modulis pagal užduotas taisykles (XML bylos formate) transformuos į duomenų šaltiniui suprantamą kalbą.

Išanalizuotų semantinės paieškos sistemų palyginimas su siūloma paieškos sistema pateiktas 4 lentelėje.

4 lentelė. Apžvelgtų semantinės paieškos sprendimų palyginimas

Teigiama savybė	Metodas, projektas	SWED	MediaWiki	Siūlomas sprendimas
Įgyvendinta ontologijos naršyklė (paieškai nereikia žinoti ontologijos struktūros)		-	-	+
Galimybė įvesti skirtingas ryšių ir mazgų reikšmes		-	-	+
Apibendrinta paieška daugiau nei per vieną semantinį ryšį		+	-	+
Galimybė atlikti paiešką daugelyje skirtingų šaltinių		-	-	+
Apytiksliai reikšmių paieškos galimybė		-	-	+
Paieškoje atsižvelgimas į įvedamo parametro tipą (tekstas, skaičius ir t.t.)		-	-	+

2.4. Architektūros ir galimų įgyvendinimo priemonių variantų analizė

Darbo eigoje gaunamo rezultato realizavimui galimos skirtingos platformos ir įvairios įgyvendinimo priemonės.

Generatoriaus prototipo realizacijai būtina pasirinkti tokią duomenų bazės valdymo sistemą (DBVS), kuri būtų plačiai naudojama, turėtų patogius įrankius kitų egzistuojančių duomenų bazių migravimui bei iš kurios būtų patogiu ištraukti sisteminę informaciją.

Nemokamos duomenų bazės, tokios kaip MySQL, nėra plačiai naudojamos didelėms duomenų bazėms, todėl jos automatiškai tampa netinkamomis. Be to, nėra taip paprasta automatiškai (programinėmis priemonėmis) išgauti duomenų bazės sisteminę informaciją: ryšius, lenteles, laukus.

Tokio milžino kaip Oracle DB panaudojimas tikrai nepalengvins darbo pirmiausia

dėl sudėtingo duomenų bazės administravimo, didelės duomenų bazės kainos, sudėtingo ir specifinio valdymo.

Rinkai pateiktas Microsoft kompanijos produktas pasižymi naudojimo paprastumu, turi integruotus įrankius kitų tipų duomenų bazių migravimui (duomenų bazės struktūros arba struktūros su duomenimis perkėlimui). Pažymėtina, kad sugeneravus ontologiją, paieškai bus ir toliau naudojama „gimtoji“ duomenų bazė. Migravimas reikalingas tik ontologijos sugeneravimui. Microsoft SQL Server 2005 turi integruotus įrankius Access, Oracle, Sybase, DB2, Informix duomenų bazių migravimui [15]. Be šių įrankių, nagrinėjama DBVS turi įdiegtą JDBC (santr. Java DataBase Connectivity) tvarkyklę (angl. driver), kurios pagalba į MS SQL Server 2005 duomenų bazę galima perkelti bet kurios duomenų bazės, duomenų apdorojimui ir mainams naudojančios Java kalba parašytas programas, struktūrą [15].

Generatoriaus realizacijai pasirinkta glaudžiausiai su MS SQL Server 2005 duomenų bazių valdymo sistema susijusi Microsoft Visual Studio .Net 2005 programavimo aplinka.

Pačios paieškos sistemos realizacijai gali būti panaudotos tiek Microsoft, tiek Java programavimo aplinkos. Java kalba įgyja tą privalumą, kad jos pagalba yra daugiau parašytų nemokamų įrankių bei taikomųjų programų ontologijų manipuliavimui, kurios gali būti panaudotos darbo metu. Panašius komponentus turi ir Microsoft .Net platforma, tačiau jų yra žymiai mažiau ir jie yra daug kuklesni savo savybėmis.

Kadangi darbui nereikia išplėstinių RDF manipuliavimo funkcijų, pasirinkta nemokama Drive [30] taikomųjų programų programavimo sąsaja (angl. API), skirta Microsoft .Net platformai.

2.5. Metodų realizuojančių prototipų pradinės specifikacijos

Šiame skyriuje apibrėžti sistemai keliami funkciniai reikalavimai bei pateiktos pradinės specifikacijos.

2.5.1. Funkciniai reikalavimai sistemoms

Ontologijų generatoriui keliami funkciniai reikalavimai:

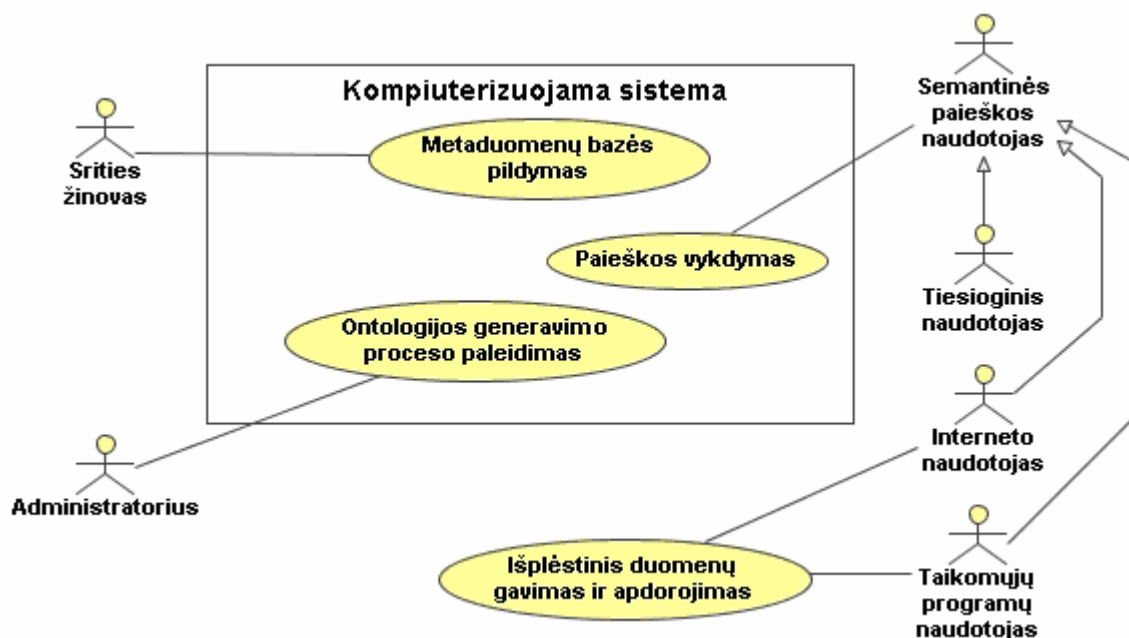
- Ontologijų išskyrimas esant minimaliam vartotojo įsikišimui.
- Generatoriaus darbo rezultato (sugeneruotos ontologijos) atitikimas standartams taip, kad ji galėtų būti panaudota ir apdorota semantinių įrankių.

Paieškos sistemai keliami funkciniai reikalavimai:

- Rasti informaciją ne tik pagal raktažodžius, bet ir pagal semantinę duomenų prasmę.
- Turi būti įgyvendinta galimybė prijungti daugumą duomenų šaltinių.

2.5.2. Sistemų prototipų pradinės specifikacijos

Kompiuterizuojamas sistemas (ontologijos generatorių bei paieškos variklį) specifikuosime panaudodami panaudojimo atvejų (angl. Use Case) diagramas. Apibendrinta panaudojimo atvejų diagrama pateikta 4 pav., o 20 pav. išskirti darbe kompiuterizuojami panaudojimo atvejai.



20 pav. Kompiuterizuojamos sistemos funkcijos

Visi sistemų panaudojimo atvejai aprašyti naudojant specifikavimo šabloną.

5 lentelė. PA „Metaduomenų bazės pildymas“ specifikacija

PA ID	1		
PA pavadinimas	Metaduomenų bazės pildymas		
Autorius	Jaroslav Karpovič	Kas atnaujino?	
Sukūrimo data	2006-01-20	Atnaujinimo data	
Aktoriai	Srities žinovas		
Aprašymas	Srities žinovo pildomi metaduomenys, aprašantys duomenų bazėje pateiktą sritį.		
„Prieš“ sąlygos	Egzistuojanti duomenų bazė, perkelta (migruota) į MS SQL Server 2005 DBVS		
„Po“ sąlygos	Užpildyta papildomos informacijos apie duomenų bazę metaduomenų bazė		
Pagrindinis scenarijus			
Žingsnio Nr.	Vartotojo veiksmas	Sistemos reakcija (jei ji turi būti)	
P-1.	Metaduomenų įvedimas	-	

6 lentelė. PA „Ontologijos generavimo proceso paleidimas“ specifikacija

PA ID	2		
PA pavadinimas	Ontologijos generavimo proceso paleidimas		
Autorius	Jaroslav Karpovič	Kas atnaujino?	

<i>Sukurimo data</i>	2006-01-20	<i>Atnaujinimo data</i>	
<i>Aktorai</i>	Administratorius		
<i>Aprašymas</i>	Administratoriaus paleidžiamas ontologijos pagal duomenų bazės schemą bei įvestus metaduomenis generavimas. Šis procesas vykdomas kiekvieną kartą pakeitus duomenų bazės struktūrą ar atnaujinus duomenų bazės meta-aprašus.		
<i>„Prieš“ sąlygos</i>	Egzistuojanti duomenų bazė, perkelta (migruota) į MS SQL Server 2005 DBVS bei užpildyta metaduomenų bazė.		
<i>„Po“ sąlygos</i>	Sugeneruota ontologija		
<i>Pagrindinis scenarijus</i>			
<i>Žingsnio Nr.</i>	<i>Vartotojo veiksmas</i>	<i>Sistemos reakcija (jei ji turi būti)</i>	
P-1.	Prisijungimas prie duomenų bazės	Pateikiama prisijungimo būseną bei pasirinktos duomenų bazės schemos	
P-2.	Duomenų bazės schemos pasirinkimas	Pateikiamos pasirinktos schemos lentelės	
P-3.	Generavimo metu naudojamų lentelių pasirinkimas	Ontologijos generavimas bei pranešimo apie būseną pateikimas	

7 lentelė. PA „Paieškos vykdymas“ specifikacija

<i>PA ID</i>	3		
<i>PA pavadinimas</i>	Paieškos vykdymas		
<i>Autorius</i>	Jaroslav Karpovič	<i>Kas atnaujino?</i>	
<i>Sukurimo data</i>	2006-01-20	<i>Atnaujinimo data</i>	
<i>Aktorai</i>	Semantinės paieškos naudotojas (tiesioginis, taikomųjų programų, interneto)		
<i>Aprašymas</i>	Semantinės paieškos duomenų bazėje pagal sugeneruotą ontologiją vykdymas		
<i>„Prieš“ sąlygos</i>	1) Sugeneruota ontologija; 2) Duomenų bazė užpildyta duomenimis		
<i>„Po“ sąlygos</i>	Pateikiamas paieškos rezultatas		
<i>Pagrindinis scenarijus</i>			
<i>Žingsnio Nr.</i>	<i>Vartotojo veiksmas</i>	<i>Sistemos reakcija (jei ji turi būti)</i>	
P-1.	Ontologijos naršymas bei pageidaujamų parametru užpildymas	Semantinės užklauskos formavimas bei pasirošimas rezultato pateikimui	
P-2.	Paieškos paleidimas	Suformuotos užklauskos vykdymas bei rezultato pateikimas.	

2.6. Reikalavimai duomenims

Nagrinėjamos sistemos duomenimis bus reliacinės duomenų bazės schema bei metaduomenų bazė. Metaduomenų bazė galės būti bet kokios struktūros, kadangi duomenų nuskaitymui iš jos bus panaudota XML tipo atvaizdavimo (angl. mapping) byla. Lanksčios metaduomenų schemos panaudojimo reikalavimas atsiranda dėl to, kad egzistuojančios duomenų bazės gali jau turėti savo metaduomenų bases, o papildomos metaduomenų bazės naudojimas galėtų būti ne tik nepatogus, bet ir neįmanomas.

Ontologijos generavimui turima duomenų bazė turės būti perkelta (angl. migrating) į

atitinkamos architektūros duomenų bazę.

Ontologijos generavimas pagal duomenų bazę, atitinkančią trečiosios normalinės formos reikalavimus (3NF) yra daug paprastesnis. Šiuo atveju net nebūtina metaduomenimis aprašinėti visų ryšių – elementarūs RDF ryšiai gali būti sugeneruoti automatiškai pagal DB schemas ryšius.

Iš tikrųjų nėra reikalavimo, kad duomenų bazė būtų trečioje normalinėje formoje, ar net tokio, kad tai būtų reliacinė duomenų bazė, tačiau visų kitų bazių meta-aprašai turi būti daug detalesni ir kruopščiau patikrinti.

2.7. Nefunkciniai reikalavimai ir apribojimai

Sistemai keliami nefunkciniai reikalavimai aptariami 8 lentelėje.

8 lentelė. Sistemai keliami nefunkciniai reikalavimai

Reikalavimas	Išpildymas
Reikalavimai standartams	Sistemos darbo rezultatas turi atitikti ontologijų bylų standartus (OWL, RDF ar kt.), nustatytus W3 konsorciūmo (w3c) [28].
Reikalavimai veikimui	Ontologijos generatoriaus darbo kokybė svarbesnė už darbo laiką. Semantinės paieškos sistemos greitimeika – labai svarbus parametras, nuo kurio priklauso vartotojo požiūris į sistemą.
Reikalavimai sąveikai/suderinamumui su kitomis sistemomis	Generatoriaus sugeneruotos ontologijos atitikimas standartams yra būtina sąlyga, kadangi taip sudaryta ontologija gali būti panaudota kitų semantinio žiniatinklio teikiamų duomenų apdorojimo paslaugų. Semantinis tinklalapis nėra integruojamas, tačiau palaiko glaudžius ryšius su kitais semantiniiais tinklalapiais, keisdamasis su jais informacija.
Reikalavimai vartotojo sąsajai	Generatoriaus ir semantinės paieškos sistemos vartotojo sąsaja privalo būti intuityvi. Generatoriaus sąsaja privalo turėti vedlio tipo dialogo su vartotoju langus.
Reikalavimai plėtimui	Sistema privalo turėti sąsajas (interfeisus) papildomų modulių prijungimui ateityje. Taip pat turi būti galimybė prie paieškos sistemos prijungti kitokius duomenų šaltinius.
Saugumo reikalavimai	Turi būti užtikrinta, kad slapta informacija nebūtų išplatinta.

2.8. Rizikos faktorių analizė

Egzistuoja keli rizikos faktoriai, galintys sukliudyti pasiekti norimus rezultatus:

- metaduomenų bazės pildymas gali būti labai sudėtingas ir reikalaujantis nemažai laiko (ypač didesnėms duomenų bazėms). Tai gali užgožti sistemų teikiamus privalumus;
- vien semantinės paieškos naudojimas gali pasirodyti nepakankamas prie paieškos pagal raktažodžius įpratusiam naudotojui, kadangi semantinė paieška pateikia tuo tikslesnį rezultatą, kuo labiau yra žinoma, kas turi būti surasta.

2.9. Rezultato kokybės kriterijai

Darbo kokybės kriterijai nustatomi atsižvelgus į sistemai išskeltus reikalavimus (žr. 2.6 ir 2.7 darbo skyrius) bei darbo rezultato – sugeneruotos ontologijos ir jos panaudojimo semantinėje paieškoje – kokybę.

Iki šiol nėra sukurtų įrankių ar netgi apibrėžtos metodikos ontologijų kokybei įvertinti. Ontologijos, generuojamos nagrinėjant laisvą, nestruktūrizuotą tekstą, dalinį įvertį nusako esybių bei ryšių skaičius, tačiau kiekybinės charakteristikos mažiau nusako negu kokybinės – sugeneruotos ontologijos teisingumas yra daug svarbesnis.

Ontologijos, generuojamos iš reliacinės duomenų bazės, teisingumas patikrinamas daug paprasčiau – ontologija šiuo atveju yra tik kitoks duomenų bazės pateikimas:

- ontologijoje neturi būti esybių (atributų) ir ryšių, neesančių duomenų bazėje;
- ontologija privalo turėti visus norimus paviešinti atributus ir jų ryšius;
- ontologijos struktūra turi atitikti W3 konsorciumo reikalavimus ir rekomendacijas [28] (žr. 2 priedas.).

Be abejo, įmanomas automatinis ontologijos, sudarytos pagal duomenų bazės schemą, patikrinimas, tačiau tai nėra šio darbo dalis, taigi, kol nėra įrankių, ontologiją tikrins srities žinovas-ekspertas (2 pav.).

Sudarytos ontologijos struktūros bei atitikimo reikalavimams patikrinimui naudojami tam skirti įrankiai ir paslaugos (žr. 4.1 skyrių „Ontologijos generatoriaus kokybės tyrimas“).

Semantinės paieškos kokybės kriterijus – kuo tikslesnė paieška. Šioje vietoje didesnis paieškos pateiktų rezultatų skaičius (skirtingai nuo raktažodžiais paremtos paieškos) ne tik nėra geros paieškos kokybės kriterijus, o netgi atvirkščiai, nurodo, kad buvo nepakankamai susiaurinta sritis.

2.10. Analizės išvados

Apibendrinus analizės rezultatus gautos šios išvados:

- semantinė paieška yra pranašesnė už raktažodžiais paremtą paiešką;
- iki šiol egzistuoja tik prototipinės semantinės paieškos sistemos;
- semantinei paieškai atlikti privalu turėti kokybišką ontologiją;
- rankinis ontologijos kūrimas reikalauja daug resursų;
- organizacijų ir kitokių subjektų duomenys šiuo metu dažniausiai saugomi duomenų bazėse (labiausiai paplitusios reliacinės);
- galimas automatinis ontologijos generavimas pagal duomenų bazės schemą bei schemą paaiškinančius metaduomenis.

Taip pat analizės metu:

- specifikuoti sistemai keliami reikalavimai bei atliekamos funkcijos, nustatyti vartotojai ir pagrindiniai veiklos proceso žingsniai, kuo remiantis bus paruoštas sistemos projektas;
- generatoriaus ir paieškos sistemos prototipų realizacijai pasirinkta Microsoft SQL server 2005 DBVS bei Microsoft Visual Studio .Net 2005 programavimo aplinka dėl DBVS didelio paplitimo, patogumo naudoti bei galimybės migruoti daugelyje kitokių DBVS sukurtų duomenų bazių.

3. Generavimo ir paieškos procesų bei juos realizuojančių prototipų specifikacijos

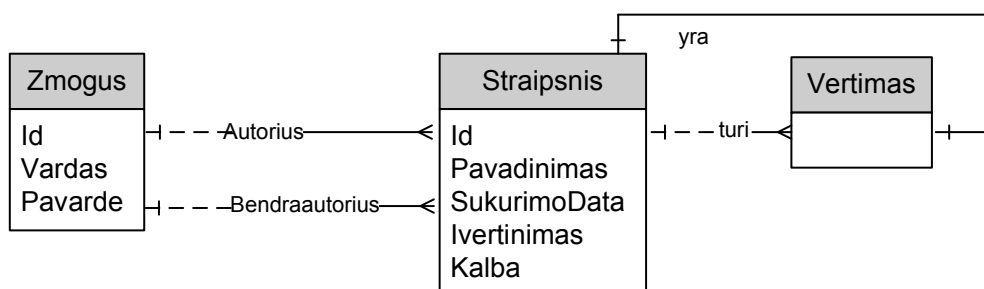
Šioje darbo dalyje apibrėžiama ontologijos architektūra, specifikuojamas ontologijos išgavimo procesas bei ontologijos generatoriaus prototipo projektas, taip pat aprašomi semantinės paieškos veikimo principai, pateikiamas jos prototipo modelis.

3.1. Ontologijos išgavimas: procesas ir generatoriaus prototipas

Šiame darbo skyriuje apibrėžiamos galimybės duomenų bazės schemas pateikimui ontologijos pagalba, pateikiama siūloma RDF tipo ontologijos architektūra bei vienas iš galimų metaduomenų (naudojamų domenų, esančių duomenų bazėje, semantiniam praturtinimui) saugojimo variantų. Be to, skyriaus gale pateikiama RDF generatoriaus architektūra.

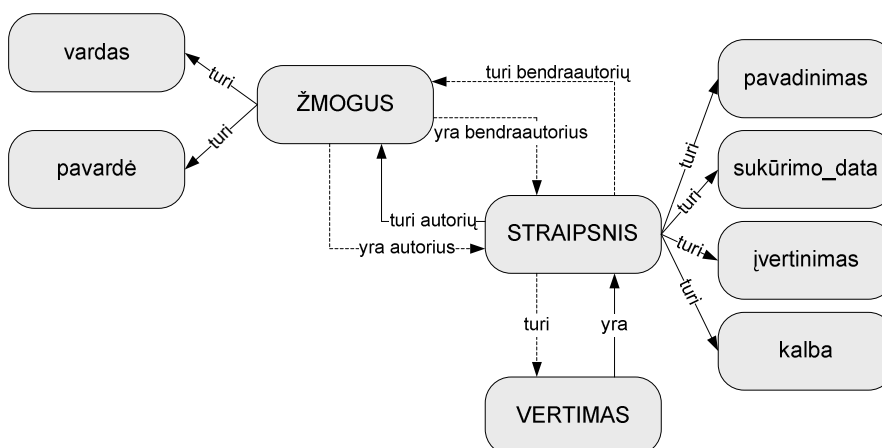
3.1.1. DB schemas pateikimas ontologijos pagalba

Tarkime, tam tikros sistemos reliacinė duomenų bazė atrodo taip, kaip pavaizduota 21 pav.



21 pav. Pavyzdinės sistemos reliacinė DB

Ontologija, atitinkanti 21 pav. duomenų bazės schemą, galėtų atrodyti taip, kaip pavaizduota 22 pav.



22 pav. Reliacinę DB atitinkanti ontologija

Pateiktas duomenų bazės fragmentas aprašo straipsnius, jų autorius bei bendraautorius ir vertimus (21 pav.). Straipsnis turi vieną autorių bei gali turėti vieną bendraautorį. Taip pat straipsnis gali turėti vertimų, kurie taip pat yra laikomi straipsniais (dėmesio: šis pavyzdys ne būtinai yra teisingas, čia jis pateiktas tiesiog kaip vaizdinė medžiaga).

21 pav. pateiktos reliacinės DB schemos ontologijos grafinis vaizdas galėtų atrodyti panašiai pateiktam 22 pav., o pavaizduotą struktūrą atitinkančios RDF bylos fragmentas pateiktas žemiau (visa RDF byla pateikta prieduose – 1 priedas.):

```

<db:Database rdf:about="&res;">
  <db:Label xml:lang="LT">Straipsnių IS</db:Label>
  <db:Label xml:lang="EN">Article IS</db:Label>
  <db:HasIdentifier>straipsniuIS</db:HasIdentifier>
</db:Database>
<db:Table rdf:about="&res;Straipsnis">
  <db:Label xml:lang="LT">Straipsnis</db:Label>
  <db:Label xml:lang="EN">Article</db:Label>
  <db:PartOfDB rdf:resource="&res;" />
  <db:HasColumn rdf:nodeID="Straipsnis.Id_rel" />
  <db:HasColumn rdf:nodeID="Straipsnis.Pavadinimas_rel" />
  <db:HasColumn rdf:nodeID="Straipsnis.SukurimoData_rel" />
  <db:HasColumn rdf:nodeID="Straipsnis.Autorius_rel" />
  <db:HasColumn rdf:nodeID="Straipsnis.Bendraautorius_rel" />
  <db:HasColumn rdf:nodeID="Straipsnis.Ivertinimas_rel" />
  <db:HasColumn rdf:nodeID="Straipsnis.Kalba_rel" />
  <db:HasChild rdf:nodeID="FK_Vertimas_Straipsnis_rel" />
  <db:HasChild rdf:nodeID="FK_Vertimas_Straipsnis1_rel" />
  <db:HasParent rdf:nodeID="FK_Straipsnis_Zmogus_rel" />
  <db:HasParent rdf:nodeID="FK_Straipsnis_Zmogus1_rel" />
  <db:HasIdentifier>dbo.Straipsnis</db:HasIdentifier>
</db:Table>
<db:Table rdf:about="&res;Vertimas">
  ***
</db:Table>

```

Pateiktame pavyzdyje matome, kad RDF ryšiai yra dvipusiai, o tai įgalina ypač patogų RDF naršymą. Bendru atveju ryšiai yra „yra_dalis“ („įeina“) ir „turi_dalį“ („turi“) [11].

Konceptų susietumo RDF byla gali būti automatiškai sugeneruota iš duomenų bazės schemos. Tokio tipo bylos generavimas – viena šio darbo dalių. Panaudojus vien duomenų bazės schemos informaciją, gaunama labai kukli ontologija, turinti tik dviejų tipų ryšius. Ontologijos praturtinimui galima naudoti du būdus:

- Sugeneravus paprastą ontologiją, generatoriaus pagalba, koreguoti ją bei papildyti rankiniu būdu ontologijų sudarymo įrankių pagalba (Protege, Altova ir pan.).
- Prieš generavimą suteikti duomenų bazės ryšiams tarp lentelių tinkamus pavadinimus, nusakančius jų rolę, aprašyti kiekvieną paieškoje dalyvaujantį duomenų lentelių lauką metaduomenimis, kurie atskleistų to lauko reikšmę.

Antrasis iš pateiktų variantų yra efektyvesnis, kadangi pasikeitus duomenų bazės

schemai ir vėl pergeneravus ontologiją, gaunama pilnoji ontologija, tuo tarpu pirmu variantu reikės iš naujo ją koreguoti.

3.1.2. Ontologijos architektūra

RDF notacija – pats paprasčiausias ontologijos vaizdavimas. Ontologijos RDF tripletai pasirinkti dėl šių priežasčių:

- tokią struktūrą atitinkančiomis XML bylomis patogiau apibrėžti priklausomybes tarp objektų dalykinėje srityje [17];
- jos suteikia papildomą prasmę duomenims (papildomą metainformaciją), pagal kurią lengviau rasti norimus duomenis;
- ontologijos panaudojimas įgalina panaudoti semantinės paieškos metodus išplėstinei paieškai atlikti [18];
- panaudojus RDF schemas (RDFS) galima detaliai apibrėžti dalykinės srities ontologijos struktūrą, suteikiant vartotojui bendrą sąsają prie paieškos iš skirtingų duomenų šaltinių taip, kad vartotojas net nepastebėtų, kad gauna apjungtą skirtingų šaltinių informaciją;
- yra sukurtos bibliotekos RDF tripletų manipuliavimui [13] (paieškos sistemoje bus panaudotos Drive RDF API bibliotekos).

Paieškos sistemoje, priklausomai nuo jos sudėtingumo bei išplečiamumo reikalavimų, gali būti iki kelių RDF arba kitokio tipo ontologijų. Jos gali saugoti duomenų bazės struktūras, užklausų struktūras, taikomųjų programų teikiamų paslaugų asortimentą bei bendrą priėjimo prie duomenų sąsają.

RDF struktūra, XML požiūriu, nėra hierarchinė:

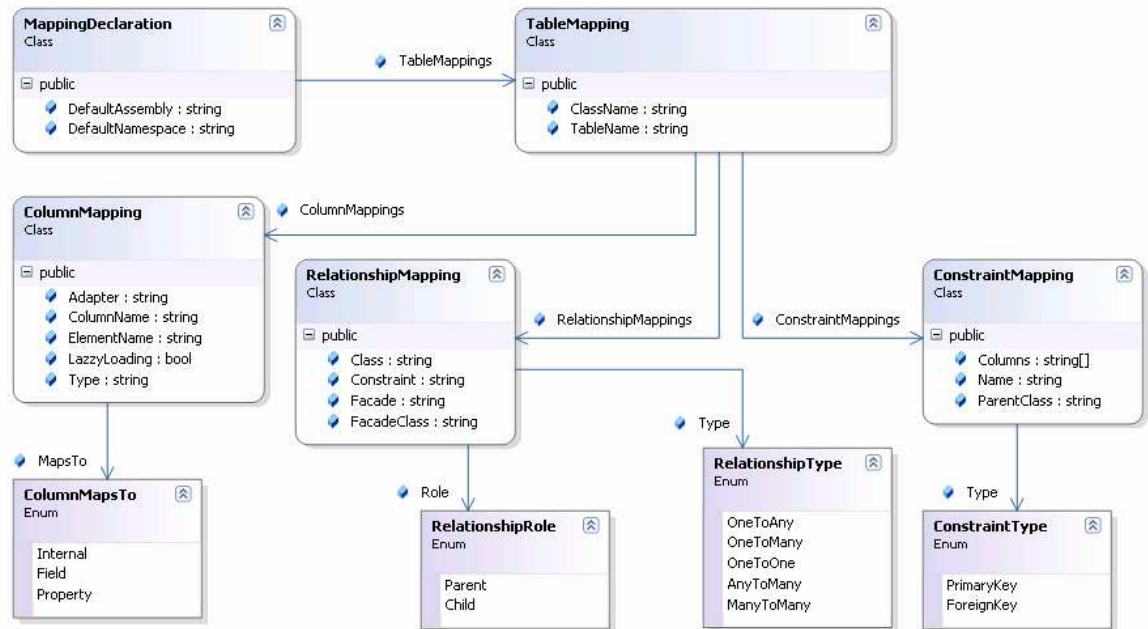
- nenaudojama įprasta reliacinėms DB hierarchija: DB → lentelė → stulpelis;
- nenaudojami priimti XML apjungiantieji elementai (pvz. LENTELES: lentelė_1, lentelė_2);
- visi RDF bylos elementai priklauso šakniniam <rdf:RDF> elementui;
- hierarchija pasiekama tik RDF struktūriniame lygyje – ontologijos elementai sudaro logines hierarchines struktūras.

Nėra paprasta manipuluoti informacija neįprastinėje formoje (kai ją tarpusavyje sieja tik šios informacijos reikšmės: elementų ir atributų pavadinimai), todėl visą duomenų bazės schemas konvertavimo į RDF procesą geriausia suskaldyti į du etapus:

- pirmiausia surinkti informaciją į hierarchinę struktūrą;

- turint hierarchinę informacijos struktūrą, ją dar kartą konvertuoti į RDF atitinkančią struktūrą.

Surenkamos informacijos apie DB struktūrą pateikiama 23 pav.



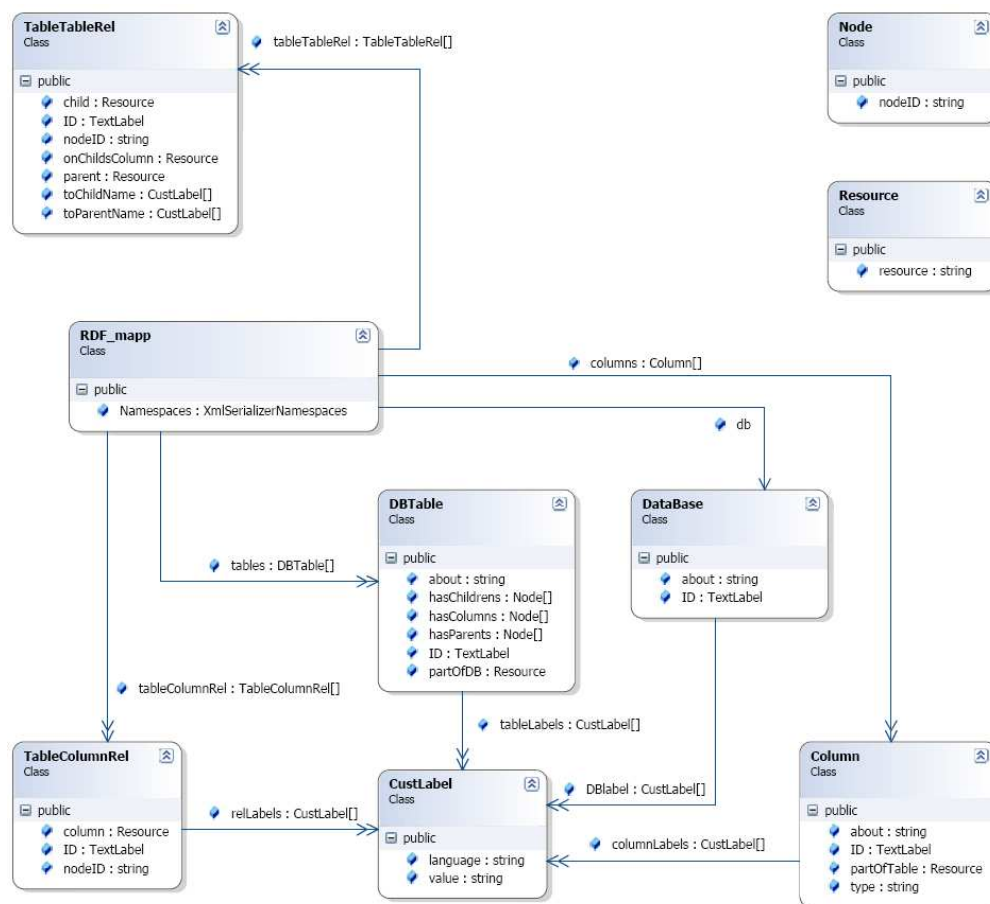
23 pav. Surenkamos informacijos apie DB lenteles struktūra

Pagrindinė klasė *MappingDeclaration*, saugoja vardų srities (*namespace*) ir schemą sudarančių lentelių kolekcijos informaciją (*TableMapping*). Kiekvienai lentelei aprašomi visi jos laukai (laukų masyvas *ColumnMapping*), ryšiai su kitomis lentelėmis (ryšių masyvas *RelationshipMapping*) bei pirminiai ir išoriniai raktai (kolekcija *ConstraintMapping*). Kiekvienas ryšys apibrėžiamas nurodant ar tai ryšys su tėvu ar vaiku ir nurodant kito ryšio galo klasės informaciją.

RDF bylos generavimui 23 pav. pateikta struktūra pertvarkoma į RDF standartus atitinkančią klasių struktūrą (24 pav.) tam, kad panaudojus *XML serializaciją* sugeneruoti ontologiją (žr. 1 priedas.).

Pažymėtina, kad ontologijos conceptams (elementai, atributai) panaudoti anglų kalbos pavadinimai dėl šių priežasčių:

- toje pačioje RDF byloje *rdf* vardų srities atributai ir elementai yra vardinami anglų kalba (pagal standartizuotą vardų sritį: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>);
- anglų kalbos pavadinimų panaudojimas įgalina panaudoti ontologiją ne tik savo tikslams (kaip, pvz., žemiau aprašomoje semantinėje paieškoje), bet ir pateikti ją kitoms semantinėms technologijoms.



24 pav. RDF ontologijos struktūrą atitinkanti klasių diagrama

3.1.3. Metaduomenų bazės struktūra

Pažymėtina, kad metaduomenų bazės struktūra – nėra šio darbo tikslas. Darbo atlikimui parinkta struktūra, pakankama sistemos veikimui palaikyti.

Realioje aplinkoje paieškos variklis prijungiamas prie egzistuojančių sistemų, kurios dažnai turi savo metaduomenų bazę. Egzistuojančią metaduomenų bazę būtų patogiausia išnaudoti semantiniam duomenų praturtinimui, tačiau dėl priežasties, kad šio darbo tikslas nėra sukurti universalios metaduomenų bazės modelį, žemiau pateikiama paprasčiausia (paprasčiausiai pakankama) struktūra.

Ontologijos generatoriaus naudojama metaduomenų bazė sutelpta vienoje duomenų bazės lentelėje (25 pav.).

metaDB
object_id
object_type
objects_discr_element
parameter
value
initial
visible

25 pav. Metaduomenų bazės-lentelės struktūra

Informacijos apie duomenų bazės schemą praplėtimo (metaduomenų, 25 pav.) lentelės laukų aprašai pateikti 9 lentelėje. Pavyzdinės ontologijos (22 pav.) meta-aprašai pateikti prieduose (žr. 5 priedas.).

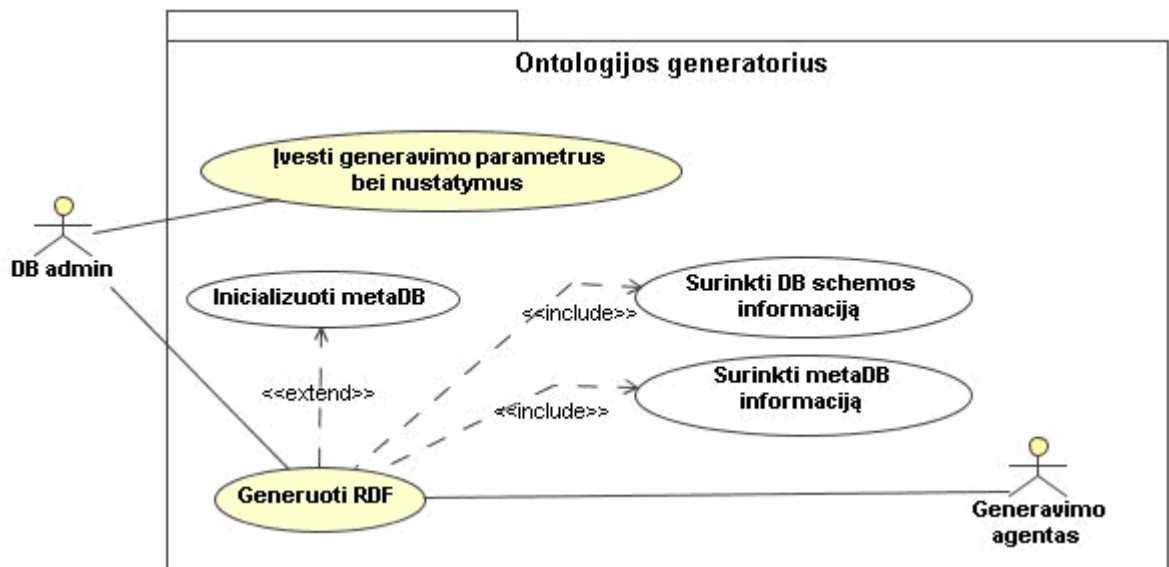
9 lentelė. Metaduomenų bazės laukų aprašai

Atributas	Tipas	Būtinasis	Atributo aprašas
object_id	nvarchar(max)	Taip	Ontologijos mazgo identifikatorius, mazgą identifikuojanti sąvoka.
object_type	nvarchar(50)	Taip	Mazgo tipas (metaduomenų bazės pildymą palengvinantis atributas). Mazgo tipas gali būti: <ul style="list-style-type: none"> • database (duomenų bazė); • db table (duomenų bazės lentelė); • table column (lentelės stulpelis); • table-column relation (ryšys tarp lentelės ir stulpelio); • table-table relation (ryšys tarp lentelių).
objects_discr_element	nvarchar(50)	Taip	Mazgo aprašomas elementas. Pradinėje versijoje papildomai aprašomos tik tekstinės etiketės (Label, ToChildName, ToParentName).
parameter	nvarchar(50)	Taip	Mazgo aprašomo elemento parametras. Pradinėje versijoje parametru yra tik kalba.
value	nvarchar(max)	Ne	Mazgo elemento (su nustatytu parametru) reikšmė.
initial	bit	Taip	Nurodo, kurie mazgai (lentelės) yra pradinės (pagrindinės, šakninės). Paieška ontologijoje bus vykdoma būtent nuo šakninių elementų.
visible	bit	Taip	Nurodo ar elementas yra matomas ontologijoje.

3.1.4. Ontologijos generatoriaus funkcijos

Ontologijos generatoriaus funkcijos apibrėžiamos vartotojų panaudojimo atvejų diagramomis bei jų specifikacijomis.

Ontologijos generatorius turi vieną išorinį vartotoją – duomenų bazės administratorių. Tokia aukšta vartotojo rolė yra būtina tam, kad duomenų bazės informacija būtų apsaugota nuo nesankcionuotos prieigos paieškos sistemos naudojimo metu. Už generavimo eigą atsakingas generavimo agentas.



26 pav. Ontologijos generatoriaus funkcijos

Iš panaudojimo atvejų diagramos pastebima, kad generatoriaus funkcijų nėra daug. Vienu sakiniu tas funkcijas galime įvardinti taip: vedlio tipo vartotojo sąsajos pagalba pagal vartotojo nurodytą DB schemų lenteles, pasinaudojus pasirinktų schemų bei metaduomenų bazės (metaDB) saugoma informacija, sugeneruojama XML tipo RDF susietumo (angl. mapping) byla.

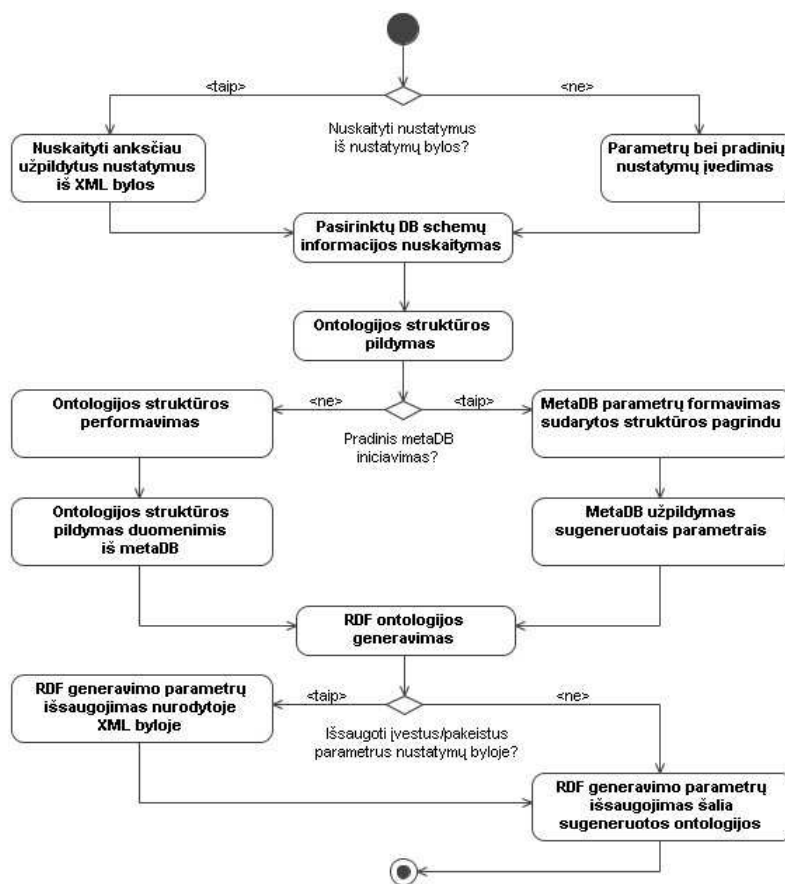
Panaudojimo atvejų specifikacijos pateiktos darbo prieduose (žr. 3 priedas.).

3.1.5. Ontologijos generatoriaus elgsena

Ontologijos generavimo procese yra trys vietos, kuriose galimas generavimo proceso pakeitimas:

- ontologijos generavimo pradžioje pasirenkama ar nustatymus ir generavimo parametrai (prisijungimas prie DB, prisijungimas prie metaDB, ontologijos vardų sritys ir pan.) bus įvedami visiškai iš naujo, ar bus koreguojami anksčiau išsaugoti XML tipo byloje nustatymai;
- antras pasirinkimas – pradinis metaduomenų iniciavimas arba skaitymas iš jau užpildytos metaduomenų bazės; pirmas variantas pasirinktoje duomenų bazės lentelėje sudaro visą metaduomenų bazės struktūrą su tuščiais „value“ laukais (paliktais duomenų įvedimui, žr. 5 priedas.);
- ontologijos generavimo pabaigoje yra galimybė išsaugoti įvestus nustatymus XML tipo byloje. Pažymėtina, kad nepriklausomai nuo to, ar buvo pasirinktas nustatymų saugojimas, ar ne, generatorius automatiškai išsaugo visus nustatymus ir parametrus tame pačiame kataloge, kuriame pasirinkta saugoti RDF ontologiją.

Ontologijos generatoriaus aukšto lygio veiklos diagrama pateikta 27 pav.



27 pav. Ontologijos generavimo procesas

3.1.6. Ontologijos generavimo taisyklės

Egzistuoja apibrėžti apribojimai ir taisyklės ontologijos pagal duomenų bazės schemą generavime:

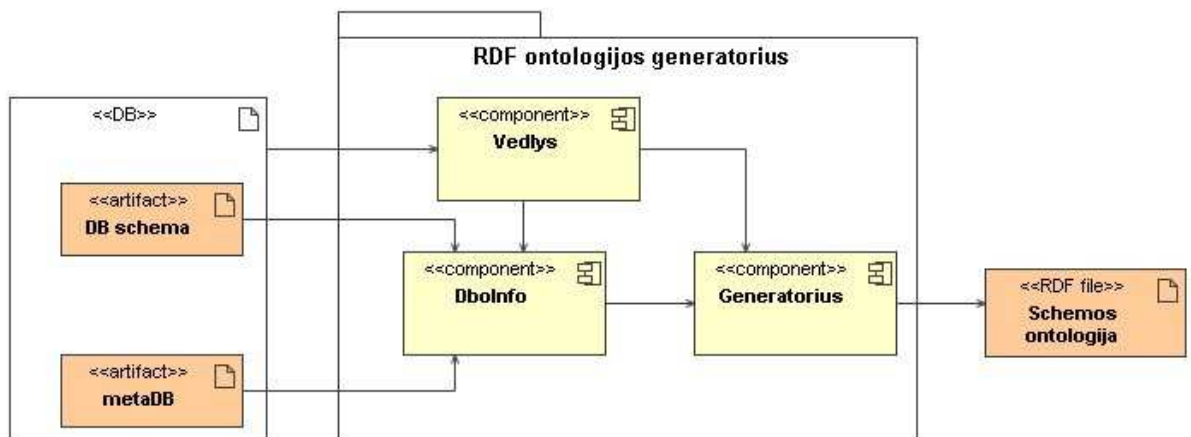
1. Atvaizduojami dviejų tipų ryšiai:
 - a. ryšiai tarp DB lentelės ir jos atributų;
 - b. ryšiai tarp DB lentelių;
2. Reliaciniai ryšiai (tarp lentelių ir tarp lentelės ir atributu) atvaizduojami tuščiu RDF mazgu (angl. blank RDF node), suteikiant visas ryšio savybes (angl. predicate) šiam mazgui.
3. RDF ryšiai, vaizduojantys ryšius tarp lentelių privalo nurodyti, kuri lentelė ryšyje yra tėvinė, o kuri vaikinė.
4. Ryšiai ontologijoje privalo būti dvipusiai taip, kad iš kiekvieno mazgo galima būtų pasiekti šio mazgo tėvinius ir vaikinius ontologijos mazgus.
5. Daugiakalbiškumas pasiekiamas RDF mazgams suteikus „*xml:lang*“ atributą,

kurio reikšmė – ISO 639-1 standartas – dviejų raidžių kalbos kodas.

6. Kiekvienas RDF mazgas privalo turėti identifikatorių.
7. Lentelės atributus atvaizduojantys RDF mazgai saugo informaciją apie vaizduojamo atributo tipą.
8. Apsisaugojimui nuo pavadinimų dubliavimosi skirtingose DB schemose, sudaromi pilni vardai kiekvienam ontologijos mazgui. Pilną vardą sudaro: `<db_schemas_vardas>.<lentelės_vardas>.<atributo_vardas>` (atitinkamai lentelei tas pats, tik be atributo vardo).

3.1.7. Ontologijos generatoriaus architektūra

Nagrinėjamas generatorius – pilnai savarankiškai egzistuojantis modulis. Šio modulio architektūra pateikta 28 pav. Diagramoje rodyklių kryptis parodo duomenų judėjimo kryptį.



28 pav. Ontologijų generatoriaus pagal DB schemą architektūra

Ontologijų generatorius sudarytas iš trijų komponentų:

Vedlys – vedlio tipo vartotojo sąsają turintis komponentas, kuris surenka iš vartotojo informaciją apie duomenų ir metaduomenų bazes bei papildomus pradinius nustatymus;

DboInfo – komponentas, surenkantis informaciją apie duomenų bazės lenteles iš DB schemas bei metaduomenų bazės;

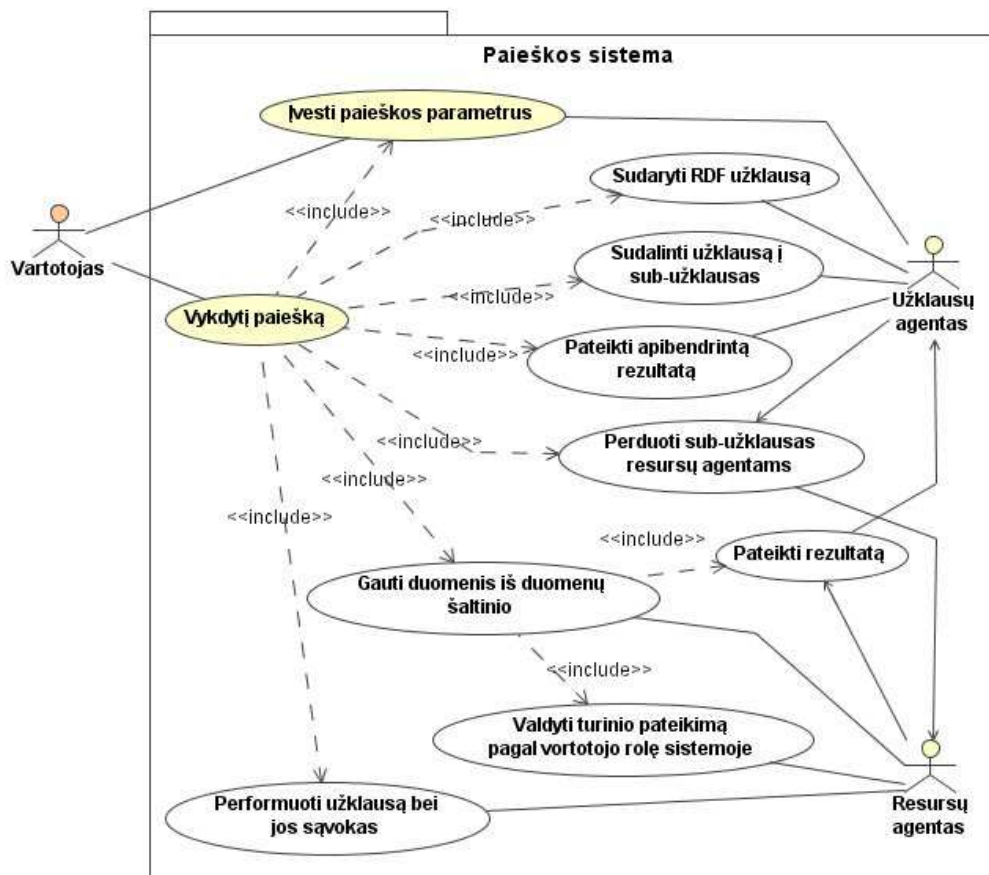
Generatorius – pagal surinktą informaciją generuojantis RDF bylą (prieš tai pertvarkius *DboInfo* komponento surinktą hierarchinę informaciją).

3.2. Semantinė paieška: procesas ir paieškos variklio prototipas

Šis skyrius detaliam aprašo semantinės paieškos esmę, jos funkcijas bei paieškos variklio prototipo architektūrą.

3.2.1. Semantinės paieškos funkcijos

Semantinės paieškos sistemos funkcijos apibrėžiamos vartotojų panaudojimo atvejų diagramomis bei jų specifikacijomis.



29 pav. Paieškos sistemos funkcijos

Semantinės paieškos sistema turi vieną išorinį vartotoją, paliekant tam tikriems (saugumo) duomenų šaltinio komponentams nurodyti resursų agentams, koks turinys gali būti pateiktas atitinkamai vartotojo rolei sistemoje (29 pav.).

Sistemos kontekstinė diagrama bei jos sąveika su kitomis sistemomis pateikta architektūros apžvalgos skyriuje. Semantinės paieškos panaudojimo atvejų detalios specifikacijos pateiktos prieduose (žr. 4 priedas.).

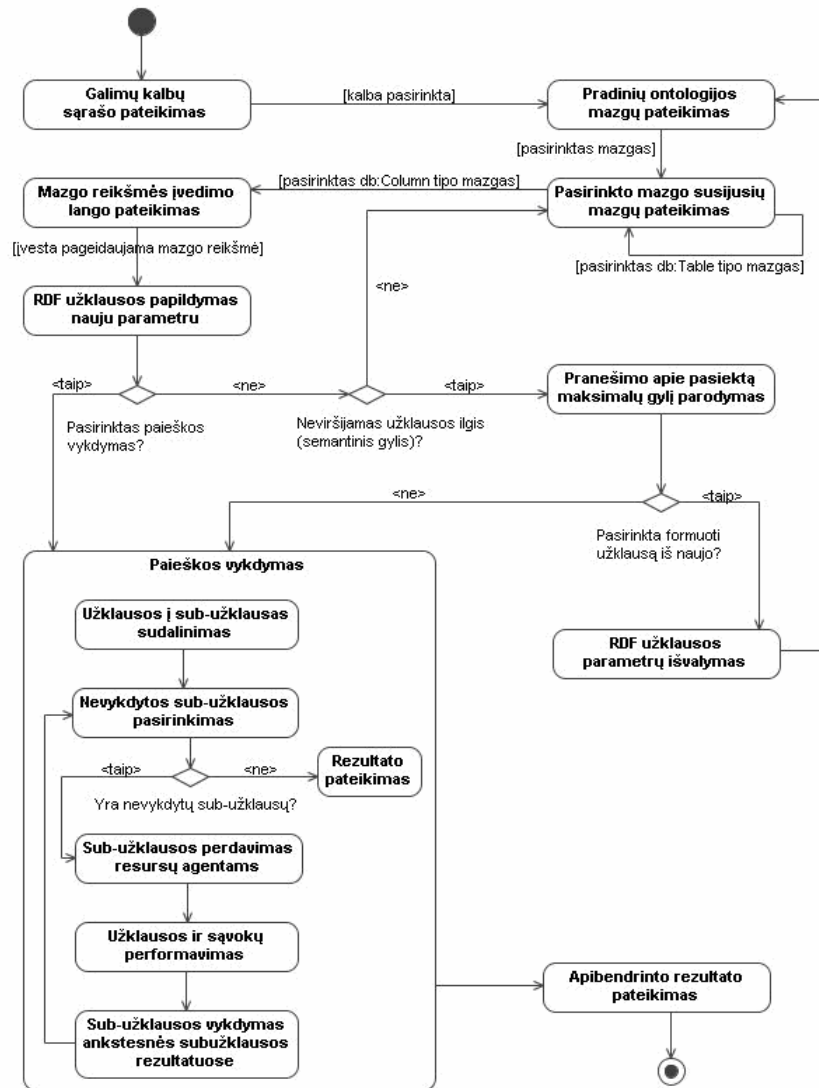
3.2.2. Semantinės paieškos elgsena

Semantinės paieškos dinamiką vaizduoja 30 pav. pateikta veiklos diagrama.

Paieška yra vykdoma sugeneruotos ontologijos pagrindu. Vartotojui iniciavus paieškos procesą (30 pav.), pirmiausia pateikiamas ontologijos klasių skirtingų kalbų sąrašas. Pasirinkus kalbą, toliau visos sąvokos yra vaizduojamos būtent pasirinkta kalba.

Pradėjus paiešką, vaizduojami tik mazgai, kurių RDF tipas yra *db:Table* ir kurie yra pradiniai (metaduomenų bazėje (25 pav.) pažymėti kaip inicializuojantys). Toliau

žingsniais, priklausomai nuo vartotojo pasirinkimo, pateikiami su pasirinktu mazgu susiję mazgai (susieti RDF ryšiais *db:TableColumnRel* ir *db:TableTableRel*). Taip pat, jei pasirinktas mazgas atvaizduoja duomenų bazės lentelės stulpelį, pateikiama galimybė įvesti pageidaujamą šio mazgo tikslią arba apytikslių reikšmę. Pageidaujamų reikšmių visuma sudaro RDF užklausą.



30 pav. Semantinės paieškos procesas

RDF užklausa gali aprėpti vartotojo pageidaujamą mazgų skaičių – taip vadinamą semantinę užklausos gylį. Gylis realioje sistemoje gali būti apribotas programiškai arba pasirenkamas vartotojo. Užklausos semantinis gylis nustato paieškos klasterio dydį (žr. 3.2.3 skyrių „Semantinės paieškos taisyklės“).

Semantinė paieška vykdoma vartotojui surinkus RDF užklausą (pasirinkus mazgus ir įvedus jų pageidaujamą reikšmę). Pirmiausia užklausa sudalinama į sub-užklauskas, t.y. vienetines užklauskas. Kitų žingsnių užklauskos yra pateikiamos resursų agentams, kur jos yra verčiamos atitinkamam šaltiniui suprantama kalba ir vykdomos. Pažymėtina, kad sub-

užklausa yra vykdoma prieš tai įvykdytos sub-užklaustos rezultatų aibėje (užklausių rezultatai sujungiami logine IR (angl. AND) sąlyga).

Apibendrinto rezultato pateikimo žingsnis apjungia rezultatus, gautus iš skirtingų šaltinių (skirtingų resursų agentų).

3.2.3. Semantinės paieškos taisyklės

Egzistuoja keletas semantinės paieškos veikimo taisyklių:

1. RDF klasėmis vadinami RDF subjektas ir objektas (angl. subject, object) – ontologijos mazgai.
2. Esybėmis vadinami duomenų šaltinių informacijos vienetai – RDF klasės egzemplioriai (pvz. lentelės eilutė duomenų bazės saugykloje).
3. Esybės privalo turėti identifikatorius.
4. Klasterį sudaro viena ar kelios esybės, susietos semantiniiais ryšiais.
5. Klasteris identifikuojamas šaknine klasterio esybe.
6. Šaknine klasterio esybe laikoma pradinės (angl. initial) RDF klasės esybė (jei ji įeina į klasterį) arba esybė, į kurią yra daugiausia paieškos pataikymų (jei į klasterį įeina daugiau nei viena pradinė RDF klasė arba neįeina nei viena).
7. Semantinis užklaustos gylis nusako klasterio dydį ir yra matuojamas pilnų RDF ryšių (angl. predicate) skaičiumi.
8. RDF ryšys yra laikomas pilnu, jei jis jungia netuščius RDF mazgus (angl. nonblank RDF nodes); ryšys, einantis per tuščius RDF mazgus yra laikomas nesudalintu, vienu pilnu ryšiu.
9. Yra du paieškos kokybės kriterijai:
 - a. pataikymų skaičius – nurodo kiek kartų paieška grąžino tam tikro klasterio identifikatorių;
 - b. pilnumas – procentinis santykis į klasterį rodančių užklaustos parametrų su visų paieškai pateiktų parametrų skaičiumi.
10. Persidengiantys paieškos atrinkti klasteriai yra laikomi praplėsto pataikomumo klasteriais.

3.2.4. Semantinės paieškos sistemos architektūra

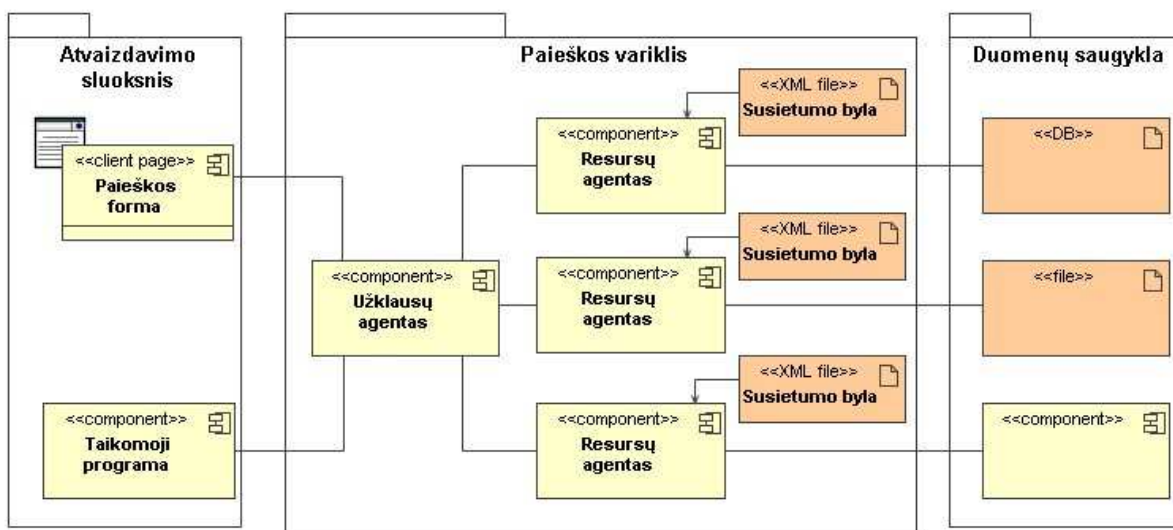
Semantinės paieškos sistema, kai paieška vykdoma daugelyje šaltinių, aprašyta 3.2.4.1 poskyryje „Apibendrinta paieškos sistemos architektūra“. Darbe apsiribojama siauresne paieškos sistemos funkcija – paieška reliacinėje duomenų bazėje (žr. 3.2.4.2 skyrių „Paieška portalo reliacinėje duomenų bazėje“).

3.2.4.1. Apibendrinta paieškos sistemos architektūra

Standartiškai [26] paieškos sistema susideda iš 3 dalių, kurios atitinka trijų lygių architektūros modelį. Žiūrint iš vartotojo pozicijos, pirmas yra vartotojo lygis (31 pav.), neturintis logikos ir atliekantis tik vaizdavimo funkcijas. Kitame logikos lygyje atliekamas užklausų formavimas, paskirstymas bei vertimas į atitinkamos duomenų saugyklos kalbą. Duomenų saugyklomis tokioje sistemoje laikomos duomenų bazės, bylos, per kitus komponentus gaunami duomenys.

Užklausų agentas komponentas (31 pav.) atsakingas už vartotojo įvestų užklausų apdorojimą. Šis komponentas priima užklausas iš vartotojų, padalina sudėtines užklausas į paprastas sub-užklausas bei perduoda sub-užklausų seką atitinkamų duomenų šaltinių resursų agentams (*Resursų agentas*).

Resursų agentas – komponentas, verčiantis užklausas duomenų saugyklos suprantama kalba. Reliacinės DB atveju, tokia kalba būtų SQL. Kalbos vertimas taip pat apima ontologijoje naudojamų sąvokų vertimą į atitinkamus saugyklos konceptus (tokius kaip duomenų bazė, schema, lentelė). Užklauso atitikimas tam tikros duomenų saugyklos suprantamai kalbai pasiekiamas panaudojant XML tipo susietumo (angl. mapping) bylas.

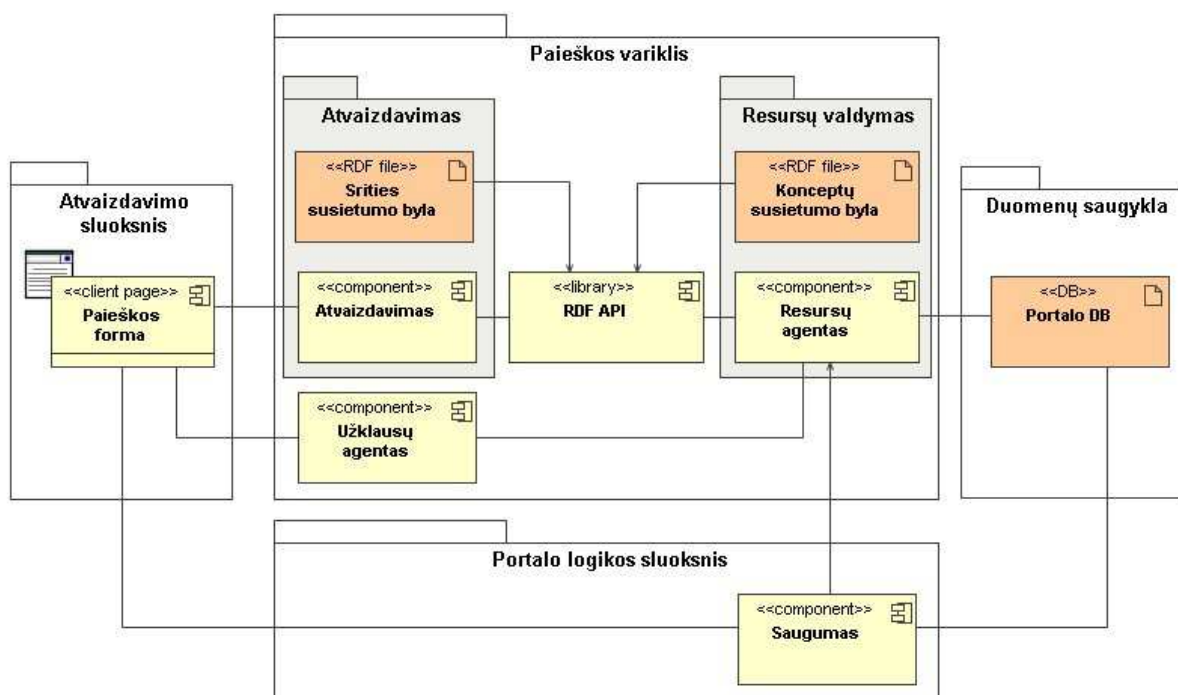


31 pav. Apibendrinta paieškos sistemos architektūra

Naujo duomenų šaltinio prijungimas tokioje sistemoje yra nesudėtingas: sukurtas šaltinio resurso agentas (*Resursų agentas*) su atitinkamomis susietumo (angl. mapping) bylomis prieregistruojamas užklausų agento (*Užklausų agentas*) registre [1].

3.2.4.2. Paieška portalo reliacinėje duomenų bazėje

Šiame darbe nagrinėjama viena iš aukščiau apibrėžtų paieškos sistemos šakų – paieška internetinio portalo reliacinėje duomenų bazėje (32 pav.).



32 pav. Paieška portalo reliacinėje duomenų bazėje

Portalo paieškos sistemoje komponentas *Resursų agentas* atliks dar papildomus veiksmus: bendraudamas per sąsają su portalo saugumo mechanizmo komponentu (*Saugumas*), užtikrins, kad vartotojui būtų pateiktas tik jo portale užimamą rolę atitinkantis turinys. *Resursų agentas*, priklausomai nuo vartotojo paieškos tipo pasirinkimo, užklausų agentui grąžina skirtingus duomenis. Vartotojui pasirinkus įprastinę paiešką [14] pagal įvestus raktinius žodžius, sąsaja iš *Resursų agento* perduodama nuoroda į atitinkamus raktinius žodžius turintį objektą bei trumpas kontekstas, kuriame rasti atitinkami žodžiai.

Vartotojui pasirinkus semantinę paiešką, visų vienos užklausos sub-užklausų rezultatai perduodami užklausų agentui, kuris juos apjungia, apibendrina, išrikiuoja pagal sutapimo laipsnį ir pateikia vartotojui.

Užklausų bei srities (atitinkamos duomenų saugyklos) susietumui (angl. mapping) realizuoti pasirinkti RDF tipo ontologijų tripletai.

DRF API – atviro kodo (apimtos *GNU Lesser General Public License (LGPL) Version 2.1* arba *Apache License V2.0* licencijomis) bibliotekos, suteikiančios sąsają RDF (Resource Description Framework) tripletų grafui valdyti bei juo manipuluoti. Vienas iš tokių bibliotekų rinkinių (komponentų) yra Drive RDF (apimtas *GNU Lesser General Public License licenzija*). Tai patogi priemonė tripletų valdymui, parašyta C# (Visual Studio .Net) kalba.

Konceptų susietumo RDF byla tarnauja kaip tarpinis sluoksnis [19], nurodantis, kur ieškoti tam tikros (paieškos užklausoje prašomos) informacijos. Šios XML bylos pagalba užklausos sąvokos paverčiamos konkrečiai duomenų saugyklai suprantamomis sąvokomis.

Galima teigti, kad ši byla sieja (angl. mapp) tarpusavyje duomenų bazės bei atvaizdavimo sluoksniu (o konkrečiai užklausų) sąvokas ir praturtina paieškos parametrus metaduomenimis, pagal kuriuos lengviau rasti reikiamą informaciją.

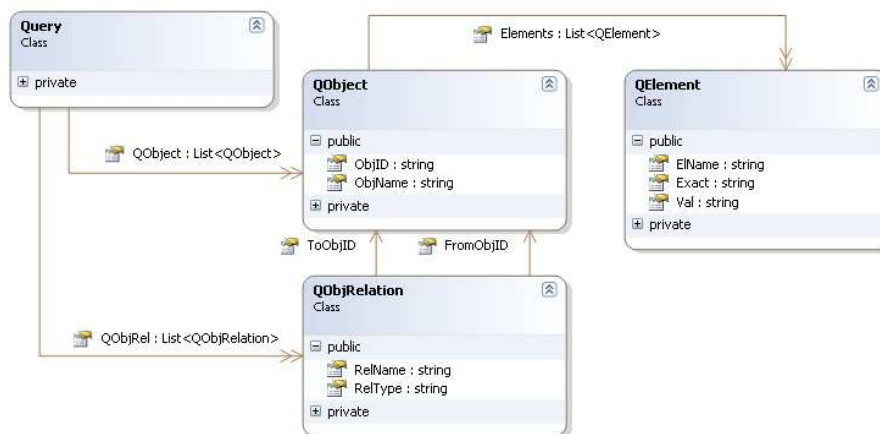
Srities susietumo RDF byla turi prasmę tik tuomet, kai paieška atliekama iš skirtingų šaltinių. Esant tik vienam šaltiniui, ši byla yra analogiška konceptų susietumo bylai ir nėra naudojama (naudojama vien konceptų RDF ontologija). Jos pagrindinė paskirtis – reprezentacijos lygyje pateikti vartotojui vieningą srities vaizdą.

Bendru atveju (esant keliems duomenų šaltiniams) šios bylos esmė – kuo tiksliau atvaizduoti dalykinę sritį, apjungiant visų sistemoje esančių konceptų ontologijų sąvokas. Sąvokų apjungimas nėra tik paprastas ontologijos prirašymas prie kitos ontologijos. Tokia situacija gaunama tik tuomet, kai skirtingi šaltiniai neturi bendrų poabių (skirtingos, nepersidengiančios sritys). Priešingu atveju konceptų ontologijos persipina vienoje ar keliose vietose.

Srities ontologija, turėdama informacijos apie tai, kokiame šaltinyje kokių duomenų galime tikėtis, pagreitina paiešką, nukreipdama užklausos agento sudarytas sub-užklausas tik tiems resursų agentams, kurie yra prisiregistravę ir nurodę, kad turi atitinkamos informacijos.

Kadangi šiame darbe apsiribojama paieška tik iš vieno šaltinio, srities susietumo ir konceptų susietumo bylos bus vienodos.

RDF užklausa atitinka 33 pav. pateiktą struktūrą.



33 pav. Užklausos duomenų struktūra

Užklausa (*Query*, žr. 33 pav.) susideda iš objektų sąrašo (*QObject*) ir ryšių tarp objektų (*QObjRelation*). Objektas (DB atveju objektą atitinka DB lentelė) identifikuojamas identifikatoriumi (*ObjID*) ir turi pavadinimą (*ObjName*), taip pat objektas turi atributų sąrašą (*QElements*). Atributai charakterizuojami vardu (*ElName*), reikšme (*Val*) ir paieškos tikslumu (*Exact*). Paieškos tikslumas priklausomai nuo atributo tipo gali būti „tiksliai“, „panašų į“, „daugiau“, „mažiau“ ir t.t. Ryšys tarp objektų turi pavadinimą (*RelName*), tipą (*RelType*) ir

saugo informaciją apie objektus, tarp kurių yra ryšys (*FromObjID*, *ToObjID*). Ryšio tipas nurodo ar tai ryšys iš tėvo į vaiką, ar atvirkščiai.

3.3. Semantinių sistemų vartotojo sąsaja

Semantinių sistemų vartotojo sąsajos skyriuje detalai aprašomos ontologijos generatoriaus ir semantinės paieškos variklio grafinės vartotojo sąsajos.

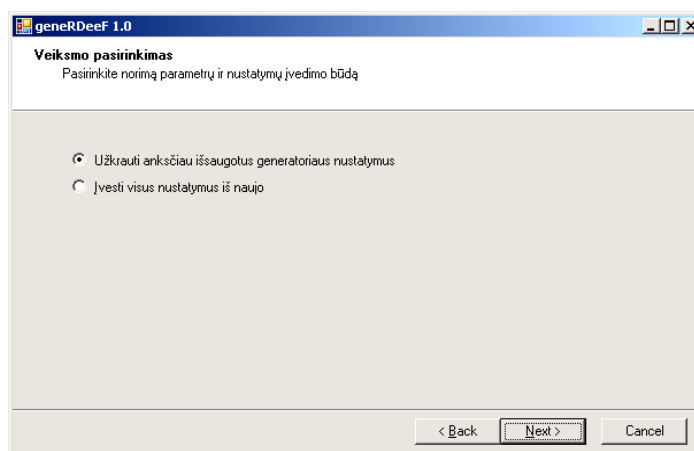
3.3.1. Ontologijos generatoriaus vartotojo sąsaja

Vedlio (angl. wizard) tipo ontologijos generatoriaus vartotojo sąsaja leidžia greitai ir patogiai sugeneruoti ontologiją iš MS SQL Server 2005 duomenų bazėje esančios DB schemos. Paleidus generatorių startuoja vedlio tipo aplikacija. Pirmajame vedlio lange (34 pav.) pateikiama trumpa informacija apie generatorių.



34 pav. Pradinis generatoriaus vedlio langas

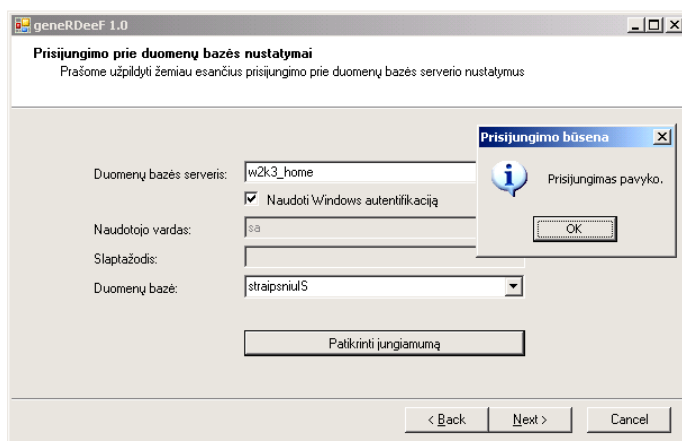
Kitame vedlio lange pasirenkamas pradinių nustatymų ir parametrų įvedimo būdas: užkrovimas iš ankščiau išsaugotos bylos su koregavimo galimybe arba įvedimas iš naujo (35 pav.)



35 pav. Nustatymų ir parametrų įvedimo būdo pasirinkimas

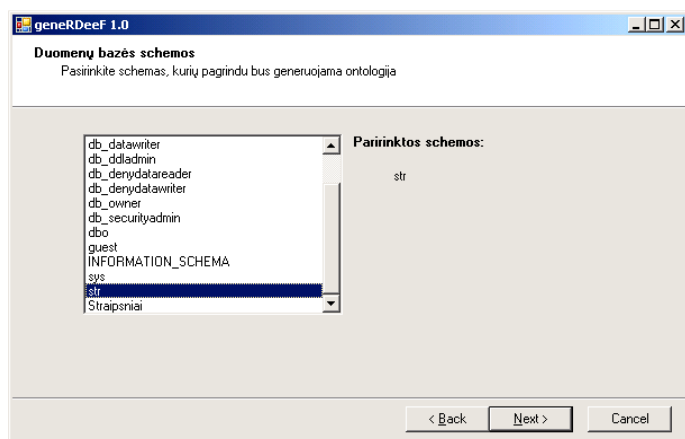
Tolesniu žingsniu įvedami/koreguojami prisijungimo prie duomenų bazės parametrai

(36 pav.). Pažymėtina, kad prie kito vedlio lango pereiti galima tik patikrinus jungiamumą su duomenų bazės serveriu.



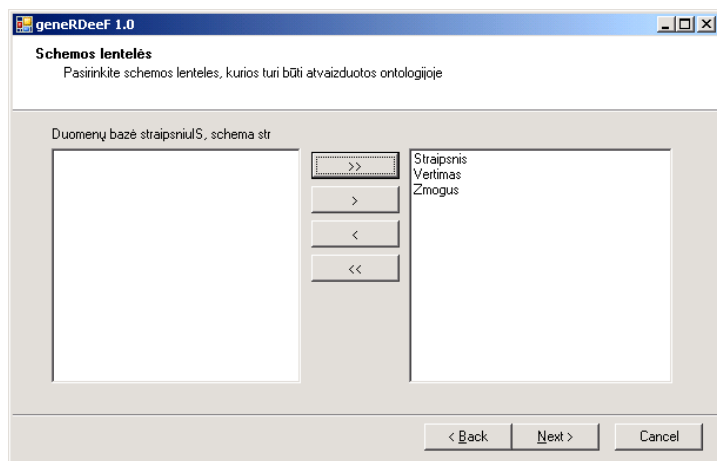
36 pav. Prisijungimo prie duomenų bazės parametrų pildymas

Paspaudus mygtuką *Next*, pereinama prie sekančio lango, kuriame pasirenkamos duomenų bazės schemas, iš kurių bus generuojama ontologija (37 pav.). Darbe naudojamas prototipas palaiko ontologijos generavimą tik iš vienos schemas.



37 pav. DB schemų pasirinkimo langas

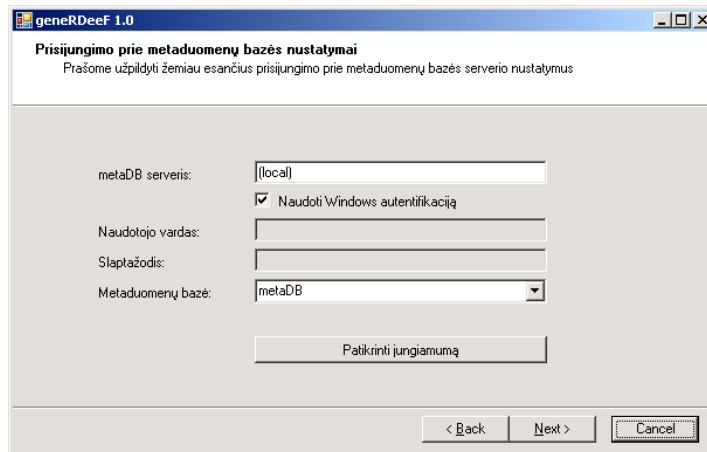
Parinkus schemas, leidžiama pasirinkti šių schemų atskiras lenteles (38 pav.).



38 pav. Schemų lentelių parinkimas generavimui

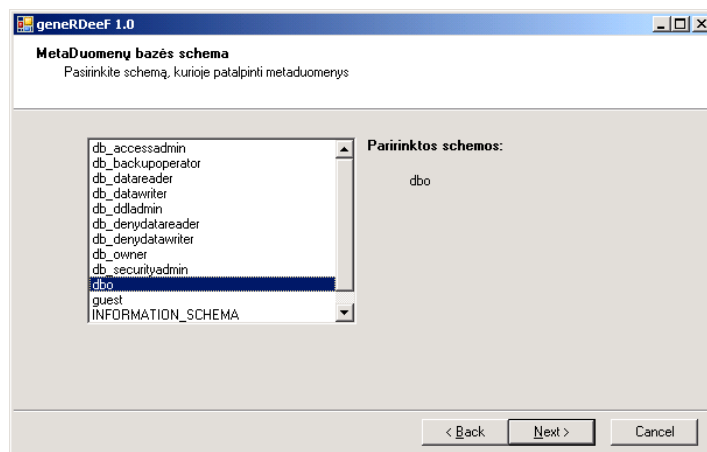
Parinkus generavime dalyvaujančias lenteles, kitame žingsnyje įvedami prisijungimo

prie metaduomenų bazės parametrai (36 pav.).



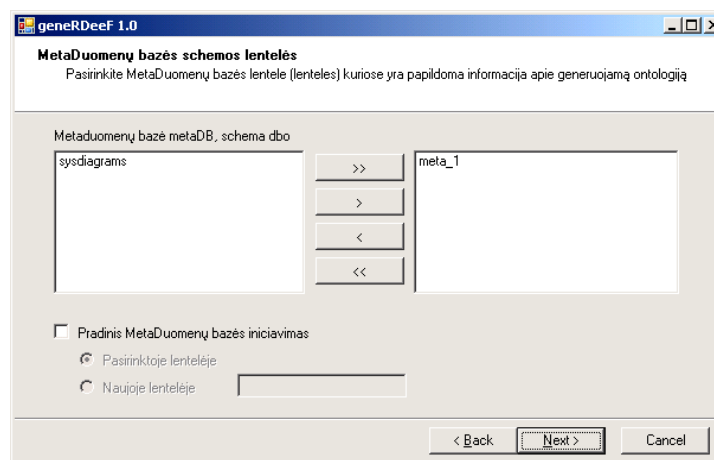
39 pav. Prisijungimo prie metaduomenų bazės parametrų pildymas

Patikrinus galimybę prisijungti ir paspaudus mygtuką *Next*, pereinama prie sekančio lango, kuriame pasirenkama metaduomenų bazės schema (skirtingai nuo duomenų bazės, čia pasirenkama tik viena schema), kurioje yra metaduomenų bazės lentelė (lentelės) (31 pav.).



40 pav. metaDB schemas pasirinkimo langas

Parinkus schemą, leidžiama pasirinkti metaduomenų lenteles (41 pav.).

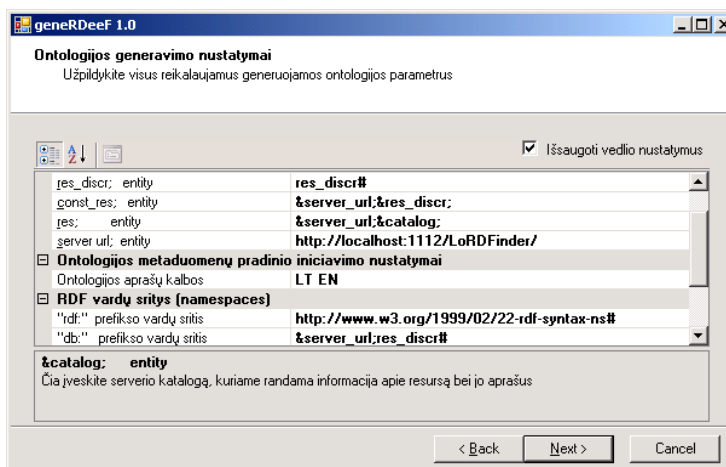


41 pav. Schemas lentelių parinkimas

41 pav. pateiktame lange taip pat nurodoma ar tai yra pradinis metaduomenų

iniciavimas. Pradinis iniciavimas – tai metaduomenų bazės (pasirinktoje lentelėje) užpildymas generavime dalyvaujančių lentelių šakniniais atributais, kurie galės būti aprašyti detaliau.

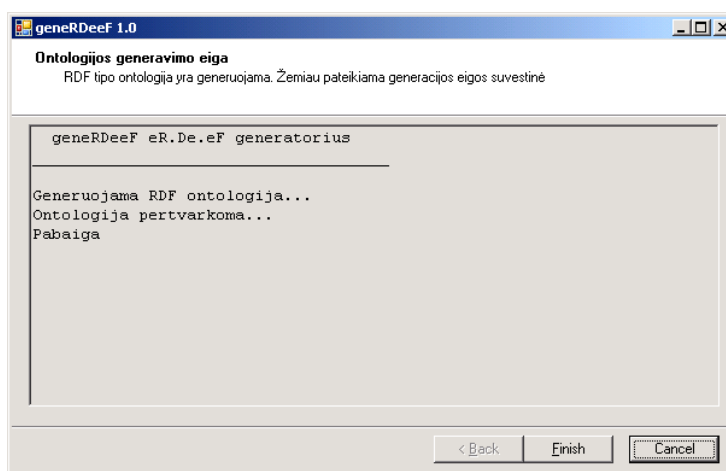
Nustačius prisijungimo prie duomenų bazių parametrus, nustatomi papildomi generavimo parametrai.



42 pav. Ontologijos generavimo parametrų pasirinkimo langas

Papildomais parametrais yra RDF vardų sritys (angl. namespaces), sugeneruotos ontologijos saugojimo vieta, pirminio metaduomenų iniciavimo kalbos ir pan. (28 pav.).

Generavimo metu matomas darbo progresas (43 pav.).



43 pav. Generavimo eigos suvestinės langas

3.3.2. Semantinės paieškos vartotojo sąsaja

Paieškos sistemos vartotojo sąsaja turi vieną langą, kuris susideda iš kelių komponentų (44 pav.).

Langas yra suskirstytas į 3 dalis: katalogo naršymo, parametrų įvedimo ir užklauso vaizdavimo. Paskutinė vartotojo sąsajos dalis (užklauso vaizdavimo) nėra būtina ir gali būti nerodoma arba rodoma tik tam tikro lygio vartotojams.

Katalogo naršymas:					
Parametrai:					
<input type="text"/>	<input type="button" value="Įvesti"/>				
<table border="1"> <tr> <td>Tikslumas:</td> <td>Tik skaičiams:</td> </tr> <tr> <td> <input checked="" type="radio"/> Tiksliai <input type="radio"/> Panašu į... </td> <td> <input type="radio"/> Daugiau negu... <input type="radio"/> Mažiau negu... </td> </tr> </table>		Tikslumas:	Tik skaičiams:	<input checked="" type="radio"/> Tiksliai <input type="radio"/> Panašu į...	<input type="radio"/> Daugiau negu... <input type="radio"/> Mažiau negu...
Tikslumas:	Tik skaičiams:				
<input checked="" type="radio"/> Tiksliai <input type="radio"/> Panašu į...	<input type="radio"/> Daugiau negu... <input type="radio"/> Mažiau negu...				
Užklausa:					
<opcionalu, ne būtina turi būti rodoma>					
<input type="button" value="Nauja paieška"/> <input type="button" value="Ieškoti"/>					

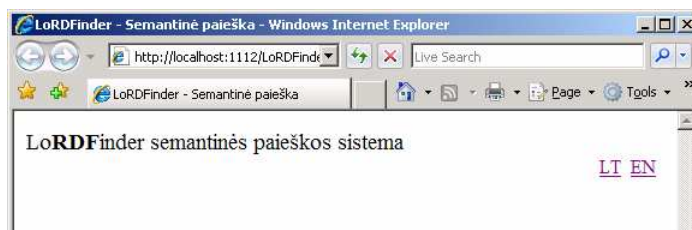
44 pav. Semantinės paieškos langas

Paieškos sistema realizuoja dvigubą funkcionalumą:

- Vartotojui nepažymėjus jokio mazgo iš katalogo naršymo srities (arba paspaudus mygtuką „Nauja paieška“), įvedus tam tikrą sąvoką (arba sąvokų rinkinį) ir spragtelėjus mygtuką „Ieškoti“ (arba nuspaudus „Enter“ klavišą) – atliekama įprastinė paieška pagal įvestus raktinius žodžius visoje duomenų saugykloje. Kadangi šio darbo sritis yra semantinė paieška, tai įprastinė paieška čia nebus nagrinėjama.
- Vartotojui parinkus (pelės pagalba) tam tikrą sritį iš katalogo ir parametruose įvedus šios srities reikšmę – atliekama semantinė paieška (paieška pagal prasmę).

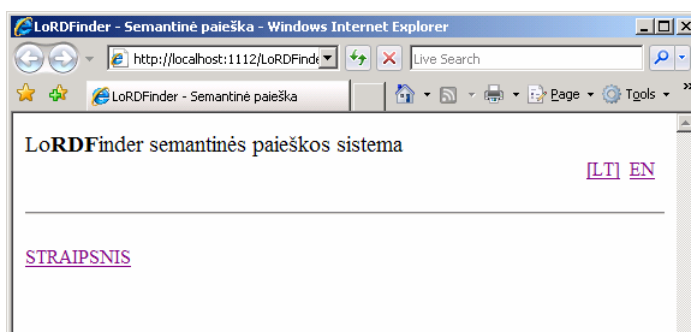
Katalogo naršymo sritis - tai sritis, kurioje vartotojui pateikiama ontologija su galimybe ją naršyti.

Jeigu nagrinėjamo pavyzdžio sistema yra „Straipsnių informacinė sistema“, tai iniciavęs paiešką vartotojas, pasirinkęs kalbą (45 pav.), ontologijos (katalogo) naršymo srityje mato ontologijos pradines (metaduomenų bazėje pažymėtas kaip „Initial“, žr. 25 pav., 5 priedas.) viršūnes (46 pav.)



45 pav. Kalbos parinkimas

Pažymėtina, kad galimų kalbų sąrašas pateikiamas iš ontologijoje esančių galimų kalbų (pagal tai, kokiomis kalbomis aprašytos duomenų bazės etiketės ontologijoje).

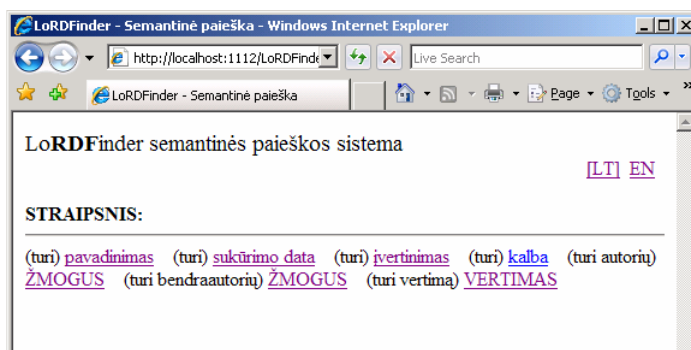


46 pav. Pradinis ontologijos mazgas ontologijos naršymo srityje

Pradinis ontologijos mazgas – tai sistemos šakninė sąvoka, nuo kurios prasideda visos paieškos. Šakninė sąvoka gali būti ne viena ir dažniausiai atitinka sistemos posistemių (loginių vienetų) pavadinimus.

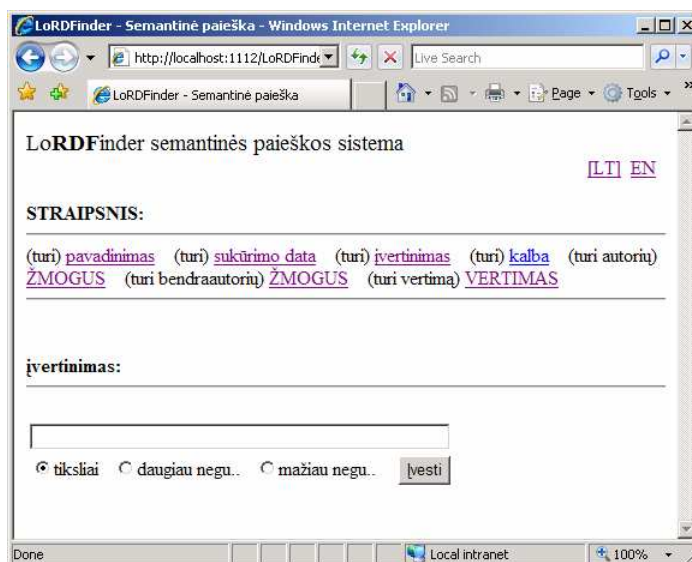
Spragtelėjus pele šakninį elementą, vartotojui pateikiamos su šią RDF ontologijos klase susijusios klasės (47 pav.).

Paieškos sistemoje didžiosiomis raidėmis rašomos tos klasės, kurios savyje turi subklasių (pasirinkus pele tokią klasę, matomos su ja susijusios klasės), tuo tarpu mažosiomis raidėmis rašomos galinės (neturinčios vaikinių) RDF klasės.



47 pav. Straipsnio susijusios klasės

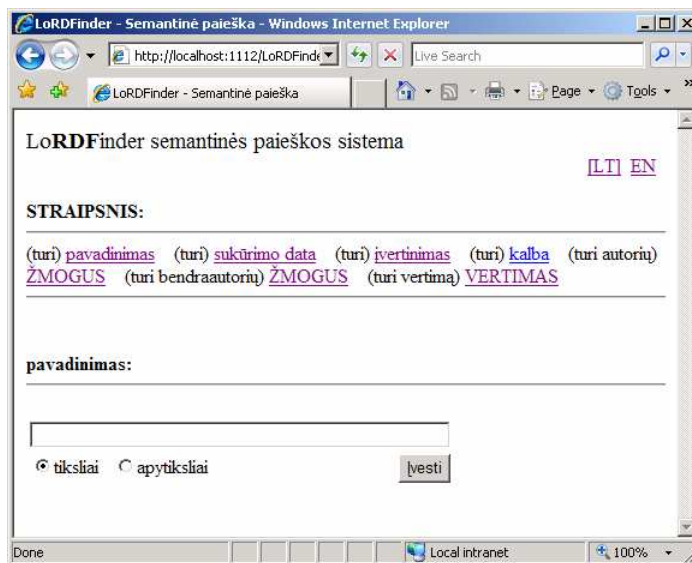
Pelės pagalba pasirinkus ontologijos mazgą, atvaizduojanti duomenų bazės stulpelį, priklausomai nuo mazgo tipo (tekstinė informacija ar skaičius, data), pageidaujamos reikšmės įvedimui pateikiamas skirtingų parametrų įvedimo langas (48 pav., 49 pav.).



48 pav. Skaitinių reikšmių įvedimas

Skaitinėms reikšmėms yra palyginimo galimybė (48 pav.). Plečiant sistemą reikėtų nepamiršti ir kitokių duomenų tipų, tačiau nereikia pamiršti, kad sudėtingų tipų naudojimas padarytų sistemą sunkiau suprantamą naudotojams.

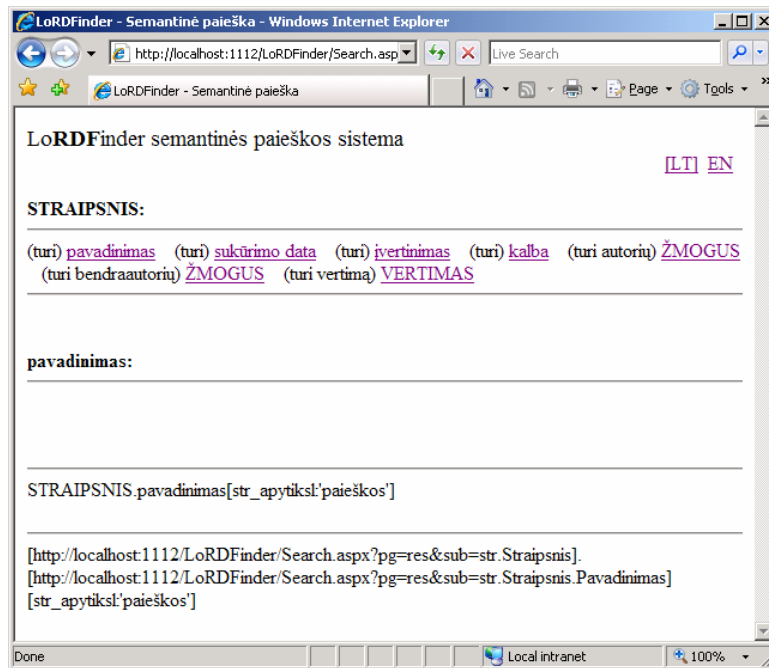
Tekstinio tipo RDF mazgų reikšmėms įvesti galimi du pasirinkimai (49 pav.): tikslus sutapimas arba reikšmės, panašios į įvestą, paieška.



49 pav. Tekstinių reikšmių įvedimas

Spragtelėjus nuorodą „pavadinimas“ (49 pav.), katalogo naršymo langas nepasikeičia, tačiau paieškos lango parametrų sritis pasiruošus priimti straipsnio pavadinimo parametą. Paieškos laukelyje įvedus žodį „paieškos“, pasirinkus paieškos būdą „apytiksliai“ ir spragtelėjus pele mygtuką „Įvesti“ arba paspaudus klavišą „Enter“, pradedama pildyti RDF užklausa, ką galima pamatyti užklauskos lango dalyje (50 pav.):

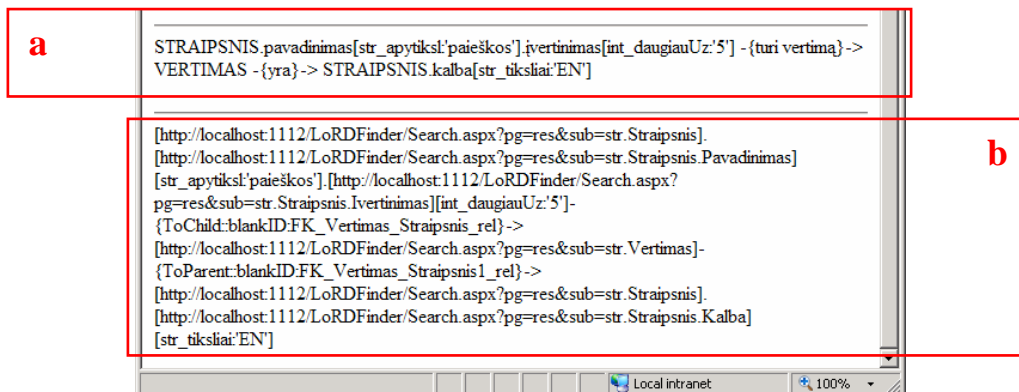
```
STRAIPSNIS.pavadinimas[str_apytiksl:'paieškos']
```

50 pav. Užklauso parametrų pildymas

Pavyzdžiui, norint rasti straipsnį, kurio pavadinime yra žodis „paieškos“, įvertintą daugiau negu 5 ir turintį vertimą į anglų kalbą, reikėtų atlikti papildomai šiuos žingsnius: katalogo naršymo lango dalyje spragtelėti pele nuorodą „(turi) įvertinimas“, įvesti norimą reikšmę ir norimą kryptingumą (tiksliai įvestą reikšmę, daugiau už įvestą ar mažiau). Sekančiu žingsniu spragtelėjame nuorodą „(turi vertimą) VERTIMAS“.

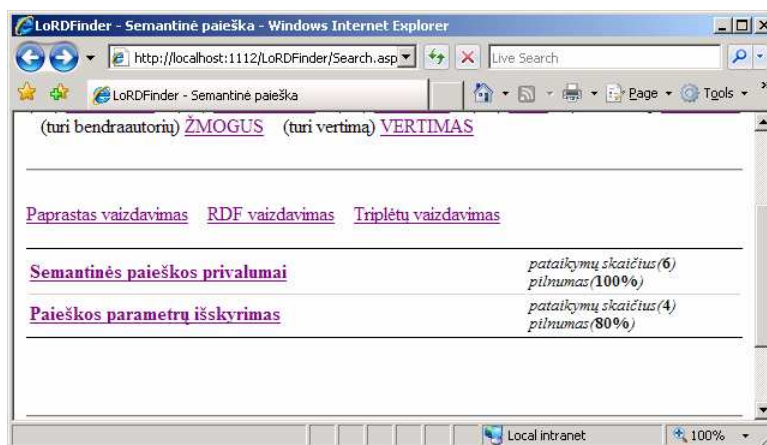
Langelyje, iš su vertimu susijusių klasių pasirenkamas „(yra) STRAIPSNIS“ (kas reiškia, kad vertimas – tai irgi straipsnis). Spragtelėjus nuorodą „kalba“ ir parametrų langelyje įvedus „EN“, spaudžiamas įvedimo mygtukas (51 pav.).



51 pav. Užpildyta užklausa su parametrais

Ankščiau pateikiamos užklauso pavyzdys – tai tik taip vadinama vartotojo užklausa (51 pav., a), skirta vartotojui stebėti įvedamus ar įvestus parametrus. Kartu sistema pildo sisteminę užklausa (51 pav., b), skirta semantinei paieškai. Tam, kad apsiriboti nuo tekstinės užklauso apdorojimo, kartu sisteminė užklausa yra pildoma specifinėje užklauso duomenų struktūroje (33 pav.).

Paspaudus mygtuką „Ieškoti“, sistema pateikia paieškos rezultatą. Galimas vienas iš trijų paieškos rezultatų vaizdavimo variantų. Įprastinis vaizdavimas – tai vaizdavimas nurodant nuorodą į šakninį klusterio elementą (52 pav.).



52 pav. Paprastas paieškos rezultatų vaizdavimas

RDF vaizdavimas – tai paieškos rezultato pateikimas grynai RDF formatu (53 pav.).



53 pav. Paieškos rezultato pateikimas RDF formatu

Paieškos rezultatų pateikimas tripletais nebuvo įgyvendintas semantinės paieškos sistemos prototipe, tačiau dažnai jis yra naudingas ir daug akivaizdesnis už RDF vaizdavimą XML struktūros pagalba.

3.4. Projekto išvados

Analizės metu išnagrinėti automatiniai ontologijų išgavimo metodai iš skirtingų šaltinių, susiaurinus sritį kristalizavosi į ontologijos išgavimo iš reliacinės duomenų bazės struktūros metodą.

Išanalizavus egzistuojančius paieškos reliacinėje duomenų bazėje metodus, nustatyta,

kad visi jie yra pakankamai sudėtingi ir dažnai neefektyvūs laiko bei pateikiamų rezultatų atžvilgiu didesnės struktūros duomenų bazėse.

Apibrėžtas procesas bei suprojektuotas įrankis automatiškai sudarantis srities ontologiją pagal MS SQL Server 2005 duomenų bazės schemą bei metaduomenų bazės informaciją.

Apibrėžtas procesas ir suprojektuota ontologijom pagrįstos semantinės paieškos portalo reliacinėje duomenų bazėje sistema – vienas iš ontologijos panaudojimo pavyzdžių.

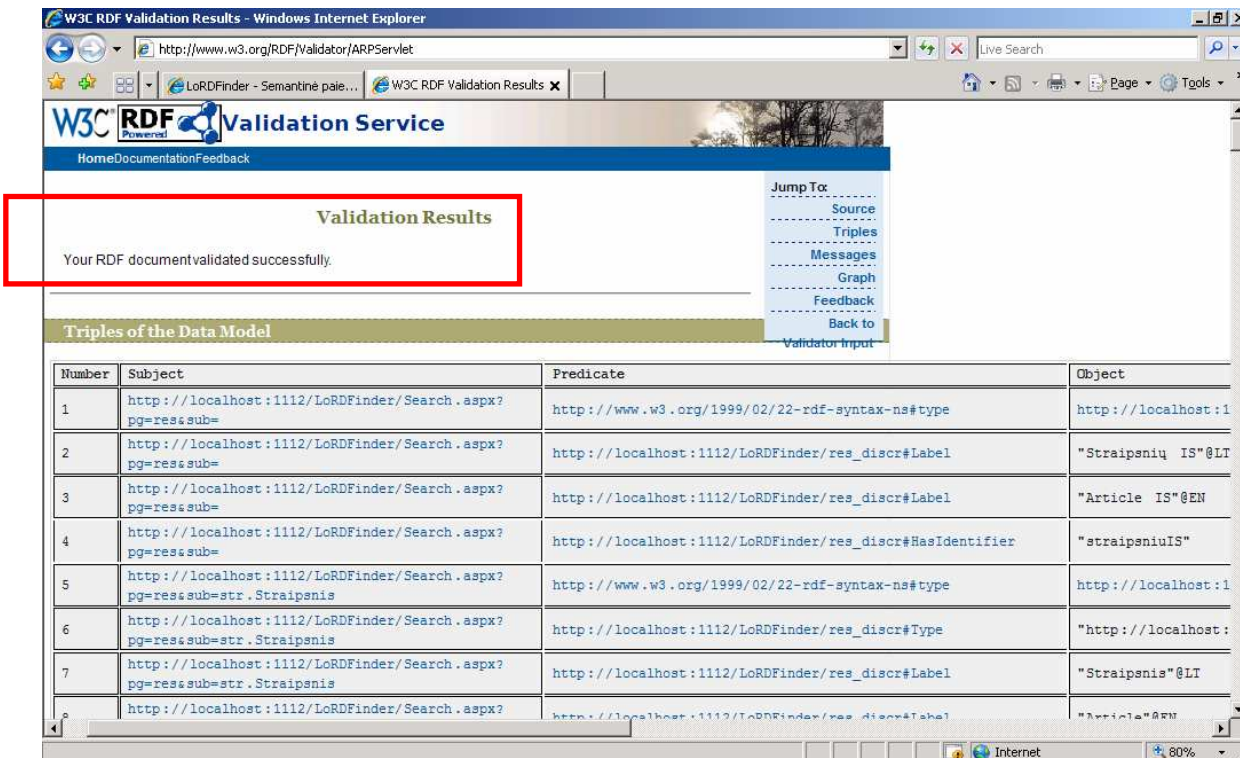
4. Eksperimentinis generatoriaus ir paieškos tyrimas

Eksperimentinis semantinių technologijų tyrimas apima pasiūlyto metodo pagrindu veikiančių generatoriaus ir semantinės paieškos variklio prototipų kokybės tyrimą, jų vertinimą, trūkumų išryškirimą ir tolimesnio plėtojimo galimybes.

4.1. Ontologijos generatoriaus kokybės tyrimas

Sukurta generatoriaus prototipas pilnai atitinka metodo apibrėžtus reikalavimus ir ontologijos generavimo proceso taisykles. Generatoriaus sugeneruota ontologija atitinka visus W3 konsorciumo reikalavimus ir ontologijos struktūros rekomendacijas, todėl ji gali būti panaudota ne tik siūlomoje semantinėje paieškoje, bet ir būti išnaudota kitų semantines paslaugas teikiančių taikomųjų programų.

Ontologijos atitikimą standartams garantuoja W3 konsorciumo sukurta ontologijos patikrinimo (validavimo) internetinė paslauga (<http://www.w3.org/RDF/Validator/>, 54 pav.)

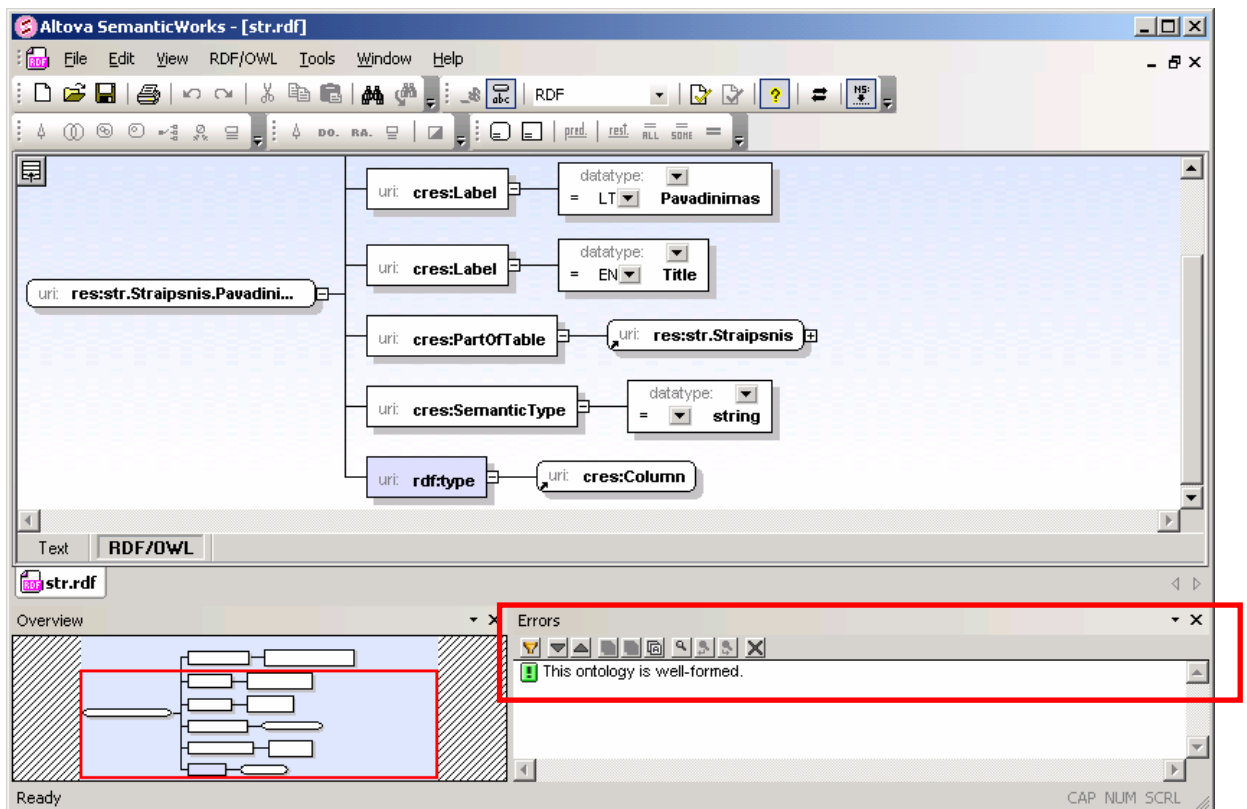


The screenshot shows the W3C RDF Validation Service interface in a Windows Internet Explorer browser. The page title is "W3C RDF Validation Results". The main heading is "Validation Results" and the message states "Your RDF document validated successfully." Below this, there is a section titled "Triples of the Data Model" which contains a table with 8 rows of data. The table has four columns: Number, Subject, Predicate, and Object. The subject column contains URIs pointing to a local search page. The predicate column contains various RDF predicates. The object column contains various values, including URIs and literals.

Number	Subject	Predicate	Object
1	http://localhost:1112/LorDFinder/Search.aspx?pg=res&sub=	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://localhost:1112/LorDFinder/Search.aspx?pg=res&sub=
2	http://localhost:1112/LorDFinder/Search.aspx?pg=res&sub=	http://localhost:1112/LorDFinder/res_discr#Label	"Straipsnių IS"@LT
3	http://localhost:1112/LorDFinder/Search.aspx?pg=res&sub=	http://localhost:1112/LorDFinder/res_discr#Label	"Article IS"@EN
4	http://localhost:1112/LorDFinder/Search.aspx?pg=res&sub=	http://localhost:1112/LorDFinder/res_discr#HasIdentifier	"straipsniuIS"
5	http://localhost:1112/LorDFinder/Search.aspx?pg=res&sub=str.Straipsnis	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://localhost:1112/LorDFinder/Search.aspx?pg=res&sub=str.Straipsnis
6	http://localhost:1112/LorDFinder/Search.aspx?pg=res&sub=str.Straipsnis	http://localhost:1112/LorDFinder/res_discr#Type	"http://localhost:1112/LorDFinder/Search.aspx?pg=res&sub=str.Straipsnis"
7	http://localhost:1112/LorDFinder/Search.aspx?pg=res&sub=str.Straipsnis	http://localhost:1112/LorDFinder/res_discr#Label	"Straipsnis"@LT
8	http://localhost:1112/LorDFinder/Search.aspx?pg=res&sub=str.Straipsnis	http://localhost:1112/LorDFinder/res_discr#Label	"Article"@EN

54 pav. W3 konsorciumo RDF patikrinimo paslauga

Taip pat ontologijos patikrinimą ir grafinį vaizdą galima pamatyti atidarius sugeneruotą ontologiją Altova kompanijos SemanticWorks įrankiu (55 pav.)



55 pav. Altova SemanticWorks ontologijos naršymas ir patikrinimas

Ontologijos generavimas – nėra realaus laiko uždavinys. Ontologija sugeneruojama vieną kartą ir pergeneruojama tik duomenų bazės schemai pasikeitus, arba papildžius meta-aprašus, todėl veikimo laikas nėra tokio tipo sistemų kritinis faktorius. Dvigubas struktūros transformavimas (žr. 23 pav., 24 pav.) trunka ilgiau, tačiau taip išvengiama klaidų bei toks generavimo būdas įgalina RDF ontologijos kūrimui panaudoti standartines Visual Studio .Net XML sudarymo (*xmlSerializer*) priemonės, kurios tikrina sudaromą struktūrą jos generavimo metu. Toks generavimo būdas yra geras tuo, kad, esant būtinybei, sugeneruotą ontologiją kelių kodo eilučių pagalba (pasinaudojus *xmlDeSerializer* bibliotekomis) galima paversti atgal į objektinę struktūrą.

4.2. Semantinės paieškos sistemos kokybės tyrimas

Sukurtas semantinės paieškos variklio prototipas teikia galimybę naršyti ontologiją pageidaujamos kalbomis, sudaryti RDF užklausas bei gauti paieškos rezultatus. Paieškos rezultatai gali būti vaizduojami dviem būdais: įprastinės nuorodos į klasterio šakninį mazgą (šis rezultato pateikimo būdas yra įprastas šiuolaikinių paieškos sistemų) ir RDF pavidalu.

Darbe nebuvo realizuotas dar vienas vizualią prasmę turintis vaizdavimo būdas – triplėtų pavidalu.

Semantinės paieškos sistemos prototipas, dėl to, kad kodas nebuvo optimizuotas

našumo atžvilgiu, nepasižymi ypatingu greičiu. Taip yra dėl to, kad kuriant prototipą, nebuvo reikalavimo našumui.

Prototipe realizuota patogi RDF naršyklė. Naršyklės ir parametrų įvedimo lauko pagalba patogiu sudarinėti RDF užklausas, tačiau pastebėta, kad kartais pritrūksta įprastinės, raktažodžiais paremtos paieškos funkcionalumo. Taip atsitinka ypač tada, kai visiškai nežinoma kur reikia ieškoti.

Atsižvelgus į pastebėtus sistemos privalumus ir trūkumus, apibrėžtos galimos tolesnės sistemų plėtojimo galimybės ir kryptis.

4.3. Tolimesnio sistemų plėtojimo galimybės

Visų pirmą reikėtų pažymėti, kad siūlomi prototipai nėra tobulinti našumo atžvilgiu (kai kurios vietos atlieka perteklinius ciklus ir pan.), todėl jų veikimo laikas gali būti žymiai sumažintas atlikus logiškus kodo pertvarkymus.

Semantinės paieškos vartotojo sąsaja yra realizuota tik lietuvių kalba, todėl, net parinkus ontologijos vaizdavimą kita kalba (tarkime anglų ar rusų), vartotojo sąsajos elementai išlieka neišversti, keičiasi tik ontologijos elementų kalba.

Viena silpniausių generatoriaus vietų – meta-aprašų (taip vadinama metaduomenų) bazė. Pildyti joje duomenis nėra patogiu. Reikėtų suprojektuoti bazę bei generatorių taip, kad generatorius automatiškai rastų ir pakeistų tik duomenų bazės schemas pasikeitusias vietas.

Ontologijos struktūra nėra pilnai užbaigta. Siūloma RDF ontologija tik nedaug praturtina aprašomus duomenis semantinėmis žiniomis. Norint atvaizduoti sąlygas ir pateikti išplėstą priklausomumą tarp duomenų, reikėtų pereiti nuo RDF prie OWL ontologijos.

Semantinės paieškos sistemą reikėtų integruoti su įprastinės (raktažodžiais paremtos) sistemos varikliu, kadangi tik taip galima pasiekti didesnės naudos. Semantinė paieška – tai lyg elektrinis variklis. Visiems žinomi elektrinio variklio privalumai (ekologija, tylus darbas ir pan.) lyginant su vidaus degimo varikliu (raktažodžiais remtos paieškos metafora), tačiau iki šiol dėl tam tikrų priežasčių automobiliuose daugiausia naudojami būtent vidaus degimo varikliai. Tačiau geriausi rezultatai ekonomiškumo ir taršos mažinimo srityje šiuo metu yra pasiekti panaudojus mišrųjį vidaus degimo su elektriniu variklį. Lygiai taip pat yra ir su paieška – negalima iškart išmesti tai, prie ko žmonės yra įpratę, kuo jie naudojami, ir pereiti prie kažko naujo.

Išvados

1. Atlikus semantinių technologijų analizę, nustatyta, kad šios technologijos daugelyje sričių pranoksta iki šiol plačiai naudojamas įprastines paslaugas (pvz. raktažodžiais paremta paieška, įprastinis žiniatinklis ir kt.), kadangi jos leidžia mašinoms lygiai taip pat kaip ir žmonėms suprasti apdorojamą informaciją bei pasidalinti supratimu. Taip pat nustatyta, kad iki šiol sukurtos tik prototipinės semantines paslaugas teikiančios sistemos, kurių privalumai nepastebimi kasdieniame gyvenime.
2. Nuodugniai išnagrinėta viena iš semantinių paslaugų – semantinė paieška ir pagrindinė šios paslaugos dedamoji – kokybiška ontologija bei jos gavimo būdai. Atlikus analizę, nustatyta, kad ontologijos generavimas pagal plačiausiai naudojamą duomenų šaltinį – duomenų bazę (o tiksliau jos schemą) – vienas patikimiausių ir pigiausių automatinio ontologijos gavimo būdų.
3. Atlikus analizę nustatyti iki šiol aprašytų ontologijos generavimo pagal duomenų bazės schemą metodų trūkumai, kurių pagrindiniai yra: visų generavimo metu duomenų bazėje esančių duomenų saugojimas ontologijoje ir ontologijos taisymo būtinumas pasibaigus generavimui. Remiantis analize, darbe pasiūlyti galutinės ontologijos (nereikalaujančios rankinio taisymo) generavimo pagal duomenų bazės schemą ir papildomus meta-aprašus procesas bei šio proceso pagrindu veikiančio automatinio ontologijos generatoriaus architektūra.
4. Remiantis analize, darbe siūlomas semantinės paieškos reliacinėje duomenų bazėje procesas ir prototipo architektūra, kurie turi pagrindinį privalumą prieš egzistuojančias semantinės paieškos sistemas, nes suteikia galimybę įvesti atskirų semantinių ryšių ir mazgų reikšmes, taip susiaurinant paieškos sritį. Semantinės paieškos variklis – praktinis sugeneruotos ontologijos pritaikymo pavyzdys.
5. Suprojektuotas ir Visual Studio .Net 2005 priemonėmis realizuotas ontologijos generatorius, veikiantis pagal Microsoft SQL Server 2005 DBVS patalpintą duomenų bazės schemą, ir semantinės paieškos variklio prototipas įgalino praktiškai patikrinti siūlomus procesus.
6. Generatoriaus darbo metu generuojama RDF reikalavimus atitinkanti ontologija gali būti naudojama tiek siūlomos semantinės paieškos sistemos, tiek kitų taikomųjų programų, veikiančių semantinių technologijų pagrindu.
7. Tarpuniversitetinėje magistrantų doktorantų konferencijoje „Informacinės technologijos 2006“ pristatytas straipsnis ir paruoštas pranešimas „Ontologijų išgavimo įrankio architektūra“ tema [9] (žr. 6 priedas.).

Literatūra

1. Baret T., Jones D., Yuan J., Sawaya J., Uschold M., Adams T., Folger D. RDF Representation of Metadata for Semantic Integration of Corporate Information Resources, The Boeing Company, Philadelphia, 2003
2. Barrasa J., Corcho Ó., Gómez-Pérez A. R2O, an Extensible and Semantically Based Database-to-ontology Mapping Language, Ontology Engineering Group, Departamento de Inteligencia Artificial, Madrid, 2004
3. Berners-Lee T., Hendler J., Lassila O. The Semantic Web [interaktyvus], 2001 [žiūrėta 2005-12-15] (Scientific Am., May 2001, pp. 34–43). Prieiga per internetą: <http://www.cs.nyu.edu/rgrimm/teaching/readings/semantic-web.pdf>
4. Buitelaar P., Olejnik D., Sintek M. A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis. Germany, Saarbruecken, 2004
5. Cyganiak R., Bizer Ch. D2R Server: Publishing databases on the Web as SPARQL endpoints, 15th World Wide Web Conference, 2006
6. Cognitive Science Laboratory, Princeton university, WordNet project [ineraktyvus], 2000-2006. Prieiga per internetą: <http://wordnet.princeton.edu>
7. Dagienė V., Grigas G., Jevsikova T. Enciklopedinis kompiuterijos žodynas. ISBN: 9955-680-19-9, Vilnius: leidykla TEV, 2005
8. Gómez-Pérez A., Manzano-Macho D. A survey of ontology learning methods and techniques. Universidad Politécnica de Madrid, 2003
9. Karpovič J., Taujanskas V. Ontologijų išgavimo įrankio architektūra. Informacinės technologijos 2006, Tarpuniversitetinė doktorantų magistrantų konferencija. Kaunas, 2006
10. Kashyap V. Design and Creation of Ontologies for Environmental Information Retrieval. Twelfth Workshop on Knowledge Acquisition, Modelling and Management Voyager Inn, Banff, Alberta, Canada. October, 1999
11. Kawata Y. Tagsets for Morphosyntactic Corpus Annotation: the idea of a 'reference tagset' for Japanese. Essex, 2005
12. Kietz J.U., Maedche A., Volz R. A Method for Semi-Automatic Ontology Acquisition from a Corporate Intranet" [interaktyvus], 2000 [žiūrėta 2005-12-15]. Prieiga per internetą: <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS//Vol-51/Maedche.pdf>

13. Lin C.-Y., Hovy E. The Automated Acquisition of Topic Signatures for Text Summarization. Information Science Institute, University of Southern California, USA, 2002
14. Maedche A., Volz R. The Ontology Extraction & Maintenance Framework: Text-To-Onto. Forschungszentrum Informatik FZI, Karlsruhe, 2001
15. Microsoft kompanijos produktų apžvalgos portalas [ineraktyvus], 2006. Prieiga per internetą: <http://www.microsoft.com>
16. Multilingual Database with WordNets for Several European Languages [ineraktyvus], 2001-2006. Prieiga per internetą: <http://www.illc.uva.nl/EuroWordNet>
17. OntoGroup – Kauno technologijos universiteto Informacijos sistemų katedros darbo grupės wiki [ineraktyvus], 2005-2006. Prieiga per internetą: <http://isk.ktu.lt:8081/wiki/>
18. O'Shaughnessy D. D. Parsing with a Small Dictionary for Applications such as Text To Speech. INRS-Telecommunications, 3 Place du Commerce, Nuns' Island, Quebec H3E 1H6 Canada, 2002
19. Pedersen T., Patwardhan S., Michelizzi J. WordNet::Similarity - Measuring the Relatedness of Concepts. Department of Computer Science University of Minnesota, Duluth, 2004
20. Rubin D.L., Hewett M., Oliver D.E., Klein T.E., Altman R.B. Automatic data acquisition into ontologies from pharmacogenetics relational data sources using declarative object definitions and XML. In: Proceedings of the Pacific Symposium on Biology, Lihue, HI, 2002
21. Seki K., Mostafa J. An Application of Text Categorization Methods to Gene Ontology Annotation. Laboratory of Applied Informatics Research, Indiana University, Bloomington, 2005
22. Semantinių technologijų tyrimo grupė [interaktyvus], 2006. Prieiga per internetą: <http://semantical.org>
23. Semantinių wiki tipo puslapių populiarinimo portalas [ineraktyvus], 2005-2006. Prieiga per internetą: <http://www.geospatialsemanticweb.com/2006/08/16/learn-to-edit-semantic-mediawiki-in-10-minutes>
24. Sheth A. P. “Semantic Web & Semantic Web Processes” [interaktyvus], 2005 [žiūrėta 2006-01-21]. Prieiga per internetą: <http://lsdis.cs.uga.edu/courses>

/madeira/MadeiraCourse-I.ppt

25. Stojanovic L., Stojanovic N., Volz R. Migrating data-intensive Web Sites into the Semantic Web. Proceedings of the 17th ACM symposium on applied computing (SAC), ACM Press, 2002, pp. 1100-1107
26. Svihla M., Jelinek I. Two Layer Mapping from Database to RDF, Department of Computer Science and Engineering Czech Technical University, Prague, 2004
27. The Semantic Web Environment Directory, [ineraktyvus], 2005-2006. Prieiga per internetą: <http://www.swed.org.uk>
28. W3C (World Wide Web Consortium) rekomendacijos [interaktyvus]. Prieiga per internetą: <http://www.w3c.org>. (detaliau žr. 2 priedas.)
29. Wikipedia, The Free Encyclopedia, “Ontology (computer science)” [ineraktyvus], 2006. Prieiga per internetą: <http://www.wikipedia.org/>
30. Singh R. Drive RDF Parser, [interaktyvus]. Prieiga per internetą: <http://www.driverdf.org/>

The Process of the Ontology Generation for the Semantic Search Engine on the Basis of Database Scheme

“Properly designed, the Semantic Web can assist the evolution of human knowledge as a whole“

– Tim Berners-Lee [3]

Summary

Data storing semantic technologies separate it from applications code and gives availability for computers as well as people understand and share semantics in real time. These technologies also enable to add new data source or link between software applications as easy as to draw new link in the model.

Unfortunately these technologies are yet not developed and popular as we could notice strong benefits of them in daily life.

Introduction of semantic search system is an attempt to show the strong points of semantic technologies. Semantic search is more precise because of its opportunities to narrow handled domain down, it gives more exact result than usual, keyword based search. This advantage is clearly shown when database is very large and is filled with plenty of data. It also gives possibility to retrieve results from multiple distant data sources and form custom or predefined result sets as a central hub for some data domain.

Automatic ontology generation based on database schema and metadata is suggested in this work. Such solution ensures that semantic search, which uses generated ontology, serves up-to-date search services even when structure of database is changed.

Priedai

1 priedas. RDF ontologija atitinkanti 21 pav. pateiktą pavyzdinę DB schema

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE rdf:RDF [
  <!ENTITY server_url 'http://localhost:1112/LoRDFfinder/'>
  <!ENTITY catalog 'Search.aspx?pg=res&sub='>
  <!ENTITY res '&server_url;&catalog;'>
  <!ENTITY res_discr 'res_discr# '>
  <!ENTITY const_res '&server_url;&res_discr;'>
]>

<rdf:RDF
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:db="&server_url;res_discr#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:res="&server_url;&catalog;"
  xmlns:cres="&server_url;&res_discr;" >

  <db:Database rdf:about="&res;">
    <db:Label xml:lang="LT">Straipsnių IS</db:Label>
    <db:Label xml:lang="EN">Article IS</db:Label>
    <db:HasIdentifier>straipsniuIS</db:HasIdentifier>
  </db:Database>

  <db:Table rdf:about="&res;str.Straipsnis" db:Type="&const_res;InitialTable">
    <db:Label xml:lang="LT">Straipsnis</db:Label>
    <db:Label xml:lang="EN">Article</db:Label>
    <db:PartOfDB rdf:resource="&res;" />
    <db:HasColumn rdf:nodeID="str.Straipsnis.Id_rel" />
    <db:HasColumn rdf:nodeID="str.Straipsnis.Pavadinimas_rel" />
    <db:HasColumn rdf:nodeID="str.Straipsnis.SukurimoData_rel" />
    <db:HasColumn rdf:nodeID="str.Straipsnis.Autorius_rel" />
    <db:HasColumn rdf:nodeID="str.Straipsnis.Bendraautorius_rel" />
    <db:HasColumn rdf:nodeID="str.Straipsnis.Ivertinimas_rel" />
    <db:HasColumn rdf:nodeID="str.Straipsnis.Kalba_rel" />
    <db:HasChild rdf:nodeID="FK_Vertimas_Straipsnis_rel" />
    <db:HasChild rdf:nodeID="FK_Vertimas_Straipsnisl_rel" />
    <db:HasParent rdf:nodeID="FK_Straipsnis_Zmogus_rel" />
    <db:HasParent rdf:nodeID="FK_Straipsnis_Zmogusl_rel" />
    <db:HasIdentifier>str.Straipsnis</db:HasIdentifier>
  </db:Table>

  <db:Table rdf:about="&res;str.Vertimas">
    <db:Label xml:lang="LT">Vertimas</db:Label>
    <db:Label xml:lang="EN">Translation</db:Label>
    <db:PartOfDB rdf:resource="&res;" />
    <db:HasColumn rdf:nodeID="str.Vertimas.Originalo_id_rel" />
    <db:HasColumn rdf:nodeID="str.Vertimas.Vertimo_id_rel" />
    <db:HasColumn rdf:nodeID="str.Vertimas.Id_rel" />
    <db:HasParent rdf:nodeID="FK_Vertimas_Straipsnis_rel" />
    <db:HasParent rdf:nodeID="FK_Vertimas_Straipsnisl_rel" />
    <db:HasIdentifier>str.Vertimas</db:HasIdentifier>
  </db:Table>

  <db:Table rdf:about="&res;str.Zmogus">
    <db:Label xml:lang="LT">Žmogus</db:Label>
    <db:Label xml:lang="EN">Person</db:Label>
    <db:PartOfDB rdf:resource="&res;" />
    <db:HasColumn rdf:nodeID="str.Zmogus.Id_rel" />
    <db:HasColumn rdf:nodeID="str.Zmogus.Vardas_rel" />
    <db:HasColumn rdf:nodeID="str.Zmogus.Pavarde_rel" />
    <db:HasChild rdf:nodeID="FK_Straipsnis_Zmogus_rel" />
    <db:HasChild rdf:nodeID="FK_Straipsnis_Zmogusl_rel" />
  </db:Table>
</rdf:RDF>
```

```

    <db:HasIdentifier>str.Zmogus</db:HasIdentifier>
</db:Table>
<db:Column rdf:about="&res;str.Straipsnis.Id" db:Type="&const_res;TableID">
  <db:PartOfTable rdf:resource="&res;str.Straipsnis" />
  <db:SemanticType>string</db:SemanticType>
  <db:HasIdentifier>str.Straipsnis.Id</db:HasIdentifier>
</db:Column>
<db:Column rdf:about="&res;str.Straipsnis.Pavadinimas">
  <db:PartOfTable rdf:resource="&res;str.Straipsnis" />
  <db:Label xml:lang="LT">Pavadinimas</db:Label>
  <db:Label xml:lang="EN">Title</db:Label>
  <db:SemanticType>string</db:SemanticType>
  <db:HasIdentifier>str.Straipsnis.Pavadinimas</db:HasIdentifier>
</db:Column>
<db:Column rdf:about="&res;str.Straipsnis.SukurimoData">
  <db:PartOfTable rdf:resource="&res;str.Straipsnis" />
  <db:Label xml:lang="LT">Sukūrimo data</db:Label>
  <db:Label xml:lang="EN">creation date</db:Label>
  <db:SemanticType>DateTime</db:SemanticType>
  <db:HasIdentifier>str.Straipsnis.SukurimoData</db:HasIdentifier>
</db:Column>
<db:Column rdf:about="&res;str.Straipsnis.Autorius" db:Type="&const_res;FK">
  <db:PartOfTable rdf:resource="&res;str.Straipsnis" />
  <db:Label xml:lang="LT">Autorius</db:Label>
  <db:Label xml:lang="EN">Author</db:Label>
  <db:SemanticType>string</db:SemanticType>
  <db:HasIdentifier>str.Straipsnis.Autorius</db:HasIdentifier>
</db:Column>
<db:Column rdf:about="&res;str.Straipsnis.Bendraautorius"
  db:Type="&const_res;FK">
  <db:PartOfTable rdf:resource="&res;str.Straipsnis" />
  <db:Label xml:lang="LT">Bendraautorius</db:Label>
  <db:Label xml:lang="EN">Co-Author</db:Label>
  <db:SemanticType>string</db:SemanticType>
  <db:HasIdentifier>str.Straipsnis.Bendraautorius</db:HasIdentifier>
</db:Column>
<db:Column rdf:about="&res;str.Straipsnis.Ivertinimas">
  <db:PartOfTable rdf:resource="&res;str.Straipsnis" />
  <db:Label xml:lang="LT">Įvertinimas</db:Label>
  <db:Label xml:lang="EN">Rating</db:Label>
  <db:SemanticType>int</db:SemanticType>
  <db:HasIdentifier>str.Straipsnis.Ivertinimas</db:HasIdentifier>
</db:Column>
<db:Column rdf:about="&res;str.Straipsnis.Kalba">
  <db:PartOfTable rdf:resource="&res;str.Straipsnis" />
  <db:Label xml:lang="LT">Kalba</db:Label>
  <db:Label xml:lang="EN">Language</db:Label>
  <db:SemanticType>string</db:SemanticType>
  <db:HasIdentifier>str.Straipsnis.Kalba</db:HasIdentifier>
</db:Column>
<db:Column rdf:about="&res;str.Vertimas.Originalo_id" db:Type="&const_res;FK">
  <db:PartOfTable rdf:resource="&res;str.Vertimas" />
  <db:SemanticType>string</db:SemanticType>
  <db:HasIdentifier>str.Vertimas.Originalo_id</db:HasIdentifier>
</db:Column>
<db:Column rdf:about="&res;str.Vertimas.Vertimo_id" db:Type="&const_res;FK">
  <db:PartOfTable rdf:resource="&res;str.Vertimas" />
  <db:SemanticType>string</db:SemanticType>
  <db:HasIdentifier>str.Vertimas.Vertimo_id</db:HasIdentifier>
</db:Column>
<db:Column rdf:about="&res;str.Vertimas.Id" db:Type="&const_res;TableID">
  <db:PartOfTable rdf:resource="&res;str.Vertimas" />
  <db:SemanticType>string</db:SemanticType>
  <db:HasIdentifier>str.Vertimas.Id</db:HasIdentifier>
</db:Column>
<db:Column rdf:about="&res;str.Zmogus.Id" db:Type="&const_res;TableID">
  <db:PartOfTable rdf:resource="&res;str.Zmogus" />
  <db:SemanticType>string</db:SemanticType>
  <db:HasIdentifier>str.Zmogus.Id</db:HasIdentifier>
</db:Column>
<db:Column rdf:about="&res;str.Zmogus.Vardas">

```

```

    <db:PartOfTable rdf:resource="&res;str.Zmogus" />
    <db:Label xml:lang="LT">Vardas</db:Label>
    <db:Label xml:lang="EN">Name</db:Label>
    <db:SemanticType>string</db:SemanticType>
    <db:HasIdentifier>str.Zmogus.Vardas</db:HasIdentifier>
</db:Column>
<db:Column rdf:about="&res;str.Zmogus.Pavarde">
    <db:PartOfTable rdf:resource="&res;str.Zmogus" />
    <db:Label xml:lang="LT">Pavardė</db:Label>
    <db:Label xml:lang="EN">Surname</db:Label>
    <db:SemanticType>string</db:SemanticType>
    <db:HasIdentifier>str.Zmogus.Pavarde</db:HasIdentifier>
</db:Column>
<db:TableColumnRel rdf:nodeID="Straipsnis.Id_rel">
    <db:Column rdf:resource="&res;str.Straipsnis.Id" />
    <db:HasIdentifier>tcrel_str.Straipsnis.Id</db:HasIdentifier>
</db:TableColumnRel>
<db:TableColumnRel rdf:nodeID="Straipsnis.Pavadinimas_rel">
    <db:Column rdf:resource="&res;str.Straipsnis.Pavadinimas" />
    <db:Label xml:lang="LT">turi</db:Label>
    <db:Label xml:lang="EN">has</db:Label>
    <db:HasIdentifier>tcrel_str.Straipsnis.Pavadinimas</db:HasIdentifier>
</db:TableColumnRel>
<db:TableColumnRel rdf:nodeID="Straipsnis.SukurimoData_rel">
    <db:Column rdf:resource="&res;str.Straipsnis.SukurimoData" />
    <db:Label xml:lang="LT">turi</db:Label>
    <db:Label xml:lang="EN">has</db:Label>
    <db:HasIdentifier>tcrel_str.Straipsnis.SukurimoData</db:HasIdentifier>
</db:TableColumnRel>
<db:TableColumnRel rdf:nodeID="Straipsnis.Autorius_rel">
    <db:Column rdf:resource="&res;str.Straipsnis.Autorius" />
    <db:Label xml:lang="LT">turi</db:Label>
    <db:Label xml:lang="EN">has</db:Label>
    <db:HasIdentifier>tcrel_str.Straipsnis.Autorius</db:HasIdentifier>
</db:TableColumnRel>
<db:TableColumnRel rdf:nodeID="Straipsnis.Bendraautorius_rel">
    <db:Column rdf:resource="&res;str.Straipsnis.Bendraautorius" />
    <db:Label xml:lang="LT">turi</db:Label>
    <db:Label xml:lang="EN">has</db:Label>
    <db:HasIdentifier>tcrel_str.Straipsnis.Bendraautorius</db:HasIdentifier>
</db:TableColumnRel>
<db:TableColumnRel rdf:nodeID="Straipsnis.Ivertinimas_rel">
    <db:Column rdf:resource="&res;str.Straipsnis.Ivertinimas" />
    <db:Label xml:lang="LT">turi</db:Label>
    <db:Label xml:lang="EN">has</db:Label>
    <db:HasIdentifier>tcrel_str.Straipsnis.Ivertinimas</db:HasIdentifier>
</db:TableColumnRel>
<db:TableColumnRel rdf:nodeID="Straipsnis.Kalba_rel">
    <db:Column rdf:resource="&res;str.Straipsnis.Kalba" />
    <db:Label xml:lang="LT">turi</db:Label>
    <db:Label xml:lang="EN">has</db:Label>
    <db:HasIdentifier>tcrel_str.Straipsnis.Kalba</db:HasIdentifier>
</db:TableColumnRel>
<db:TableColumnRel rdf:nodeID="Vertimas.Originalo_id_rel">
    <db:Column rdf:resource="&res;str.Vertimas.Originalo_id" />
    <db:HasIdentifier>tcrel_str.Vertimas.Originalo_id</db:HasIdentifier>
</db:TableColumnRel>
<db:TableColumnRel rdf:nodeID="Vertimas.Vertimo_id_rel">
    <db:Column rdf:resource="&res;str.Vertimas.Vertimo_id" />
    <db:HasIdentifier>tcrel_str.Vertimas.Vertimo_id</db:HasIdentifier>
</db:TableColumnRel>
<db:TableColumnRel rdf:nodeID="Vertimas.Id_rel">
    <db:Column rdf:resource="&res;str.Vertimas.Id" />
    <db:HasIdentifier>tcrel_str.Vertimas.Id</db:HasIdentifier>
</db:TableColumnRel>
<db:TableColumnRel rdf:nodeID="Zmogus.Id_rel">
    <db:Column rdf:resource="&res;str.Zmogus.Id" />
    <db:HasIdentifier>tcrel_str.Zmogus.Id</db:HasIdentifier>
</db:TableColumnRel>
<db:TableColumnRel rdf:nodeID="Zmogus.Vardas_rel">
    <db:Column rdf:resource="&res;str.Zmogus.Vardas" />

```

```

    <db:Label xml:lang="LT">turi</db:Label>
    <db:Label xml:lang="EN">has</db:Label>
    <db:HasIdentifier>tcrel_str.Zmogus.Vardas</db:HasIdentifier>
  </db:TableColumnRel>
  <db:TableColumnRel rdf:nodeID="Zmogus.Pavarde_rel">
    <db:Column rdf:resource="&res;str.Zmogus.Pavarde" />
    <db:Label xml:lang="LT">turi</db:Label>
    <db:Label xml:lang="EN">has</db:Label>
    <db:HasIdentifier>tcrel_str.Zmogus.Pavarde</db:HasIdentifier>
  </db:TableColumnRel>
  <db:TableTableRel rdf:nodeID="FK_Straipsnis_Zmogus_rel">
    <db:Child rdf:resource="&res;str.Straipsnis" />
    <db:Parent rdf:resource="&res;str.Zmogus" />
    <db:ToParentName xml:lang="LT">turi autoriu</db:ToParentName>
    <db:ToParentName xml:lang="EN">has author</db:ToParentName>
    <db:ToChildName xml:lang="LT">yra autorius</db:ToChildName>
    <db:ToChildName xml:lang="EN">is author</db:ToChildName>
    <db:HasIdentifier>FK_Straipsnis_Zmogus</db:HasIdentifier>
  </db:TableTableRel>
  <db:TableTableRel rdf:nodeID="FK_Straipsnis_Zmogus1_rel">
    <db:Child rdf:resource="&res;str.Straipsnis" />
    <db:Parent rdf:resource="&res;str.Zmogus" />
    <db:ToParentName xml:lang="LT">turi bendraautoriu</db:ToParentName>
    <db:ToParentName xml:lang="EN">has co-author</db:ToParentName>
    <db:ToChildName xml:lang="LT">yra bendraautorius</db:ToChildName>
    <db:ToChildName xml:lang="EN">is co-author</db:ToChildName>
    <db:HasIdentifier>FK_Straipsnis_Zmogus1</db:HasIdentifier>
  </db:TableTableRel>
  <db:TableTableRel rdf:nodeID="FK_Vertimas_Straipsnis_rel">
    <db:Child rdf:resource="&res;str.Vertimas" />
    <db:Parent rdf:resource="&res;str.Straipsnis" />
    <db:ToChildName xml:lang="LT">turi vertima</db:ToChildName>
    <db:ToChildName xml:lang="EN">has translation</db:ToChildName>
    <db:HasIdentifier>FK_Vertimas_Straipsnis</db:HasIdentifier>
  </db:TableTableRel>
  <db:TableTableRel rdf:nodeID="FK_Vertimas_Straipsnis1_rel">
    <db:Child rdf:resource="&res;str.Vertimas" />
    <db:Parent rdf:resource="&res;str.Straipsnis" />
    <db:ToParentName xml:lang="LT">yra</db:ToParentName>
    <db:ToParentName xml:lang="EN">is a</db:ToParentName>
    <db:HasIdentifier>FK_Vertimas_Straipsnis1</db:HasIdentifier>
  </db:TableTableRel>
</rdf:RDF>

```

2 priedas. Ontologijos atitikimas W3 konsorciumo reikalavimams bei rekomendacijoms

1. RDF Core Working Group, *Resource Description Framework (RDF)*, <http://www.w3.org/RDF/>, 2004.
2. W3C Recommendation 10 February 2004, *RDF/XML Syntax Specification (Revised)*, <http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/>, 2004.
3. W3C Recommendation 16 August 2006, *Namespaces in XML 1.0 (Second Edition)*, <http://www.w3.org/TR/2006/REC-xml-names-20060816>, 2006.
4. W3C Recommendation 10 February 2004, *Resource Description Framework (RDF): Concepts and Abstract Syntax*, <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>, 2004.
5. W3C Recommendation 10 February 2004, *RDF Semantics*, <http://www.w3.org/TR/2004/REC-rdf-mt-20040210/>, 2004.
6. W3C Recommendation 10 February 2004, *RDF Vocabulary Description Language 1.0: RDF Schema*, <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>, 2004.
7. W3C Recommendation 10 February 2004, *RDF Primer*, <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>, 2004.

Ontologijos atitikimą aukščiau išvardintiems reikalavimams bei rekomendacijoms garantuoja W3 konsorciumo RDF validavimo paslauga: <http://www.w3.org/RDF/Validator/>.

3 priedas. Ontologijos generatoriaus funkcijų specifikacija

Ontologijos generatoriaus panaudojimo atvejai (26 pav.) detaliai aprašomi specifikavimo šablonais (10 lentelė - 12 lentelė).

10 lentelė. PA „Įvesti generavimo parametrus bei nustatymus“ specifikacija

PA ID	4		
PA pavadinimas	PA „Įvesti generavimo parametrus bei nustatymus“		
Autorius	Jaroslav Karpovič	Kas atnaujino?	
Sukūrimo data	2006-06-20	Atnaujinimo data	
Aktoriai	DB Admin (duomenų bazės administratorius)		
Aprašymas	Informacijos apie duomenų bazę (prisijungimas, vartotojo vardas, prisijungimo tipas ir t.t.) bei generavimo nustatymus įvedimas		
Trigeriai	Vartotojas inicijuoja ontologijos generavimą		
„Prieš“ sąlygos	<ul style="list-style-type: none"> • Duomenų bazės schema sudaryta MS SQL 2005 serveryje; • Ryšiai tarp lentelių įvardinti; • Lentelių laukeliai, pagal kuriuos yra įmanoma paieška, pažymėti metaduomenų modelyje • Metaduomenų modelyje yra papildomos informacijos apie laukelio paskirtį 		
„Po“ sąlygos	Sistema surinkus visus generavimui reikalingus parametrus. Galima pradėti generavimą.		
<i>Pagrindinis scenarijus</i>			
Žingsnio Nr.	Vartotojo veiksmas	Sistemos reakcija (jei ji turi būti)	
P-1.	1. Vartotojas paleidžia generavimo modulį	1. Pateikiamas pirminis vedlio langas, kuriame vartotojas gali įvesti prisijungimo prie duomenų bazės parametrus.	
P-2.	1. Vartotojas įveda prisijungimo prie MS SQL 2005 DB serverio duomenis: <ul style="list-style-type: none"> • serverio vardas; • autentifikacijos tipas; • vartotojo vardas; • slaptažodis; 	1. Sistema patikrina jungiamumą su pasirinktu duomenų bazės serveriu; 2. Sistema pateikia egzistuojančių pasirinkto serverio duomenų bazių sąrašą	
P-3.	1. Vartotojas pasirenką generuojamos DB pavadinimą	1. Pateikiamas egzistuojančių DB schemų sąrašas su galimybe pasirinkti visas, kurių lentelės turi būti įtrauktos į generuojamą ontologiją.	
P-4.	1. Vartotojas pasirenka į generavimą įeinančias schemas	1. Pateikiamos visos pasirinktų schemų lentelės su galimybe vartotojui nurodyti, kurios dalyvaus generavimo procese	
P-5.	1. Vartotojas nurodo generavime dalyvaujančias lenteles 2. Vartotojas nurodo kur išsaugoti sugeneruota RDF bylą	1. Sistema pasiruošus generavimo procesui	
<i>Išimtys (klaidos)</i>			

<i>Klaidos sąlyga</i>	<i>Sprendimas</i>
-	<p>Visos klaidos turi būti registruojamos administratoriui pasiekiamoje byloje. Klaidos įrašą sudaro:</p> <ol style="list-style-type: none"> 1) veiksmas, kurio metu įvyko klaida; 2) vartotojas, kuris vykdė šį veiksmą; 3) klaidos tekstas. <p>Vartotoją taip pat būtina informuoti apie įvykusią klaidą, išvedant į ekraną pranešimą.</p>

11 lentelė. PA „Generuoti RDF“ specifikacija

<i>PA ID</i>	5		
<i>PA pavadinimas</i>	PA „Generuoti RDF“		
<i>Autorius</i>	Jaroslav Karpovič	<i>Kas atnaujino?</i>	
<i>Sukūrimo data</i>	2006-06-20	<i>Atnaujinimo data</i>	
<i>Aktoriai</i>	DB Admin (duomenų bazės administratorius), generavimo agentas		
<i>Aprašymas</i>	Pagal pasirinktos DB lentelės sugeneruojama RDF tipo ontologija		
<i>Trigeriai</i>	Surinkta visa prisijungimo prie DB serverio informacija, vartotojas paleidžia generavimo procesą		
<i>„Prieš“ sąlygos</i>	Sėkmingai praėjo PA „Įvesti generavimo parametrus bei nustatymus“ aprašytas procesas		
<i>„Po“ sąlygos</i>	Sugeneruota RDF tipo ontologija pasirinktai duomenų bazei		
<i>Pagrindinis scenarijus</i>			
<i>Žingsnio Nr.</i>	<i>Vartotojo veiksmas</i>	<i>Sistemos reakcija (jei ji turi būti)</i>	
P-1.	1. Vartotojas paleidžia generavimo procesą	<p>1. FOR EACH lentelė iš nurodytų DB lentelių sąrašo</p> <p>PA „Surinkti informaciją iš DB schemos sisteminių „view“ bei metaDB“</p> <p>Užpildyti duomenimis specialią duomenų struktūrą</p> <p>2. Iš sudarytos struktūros sugeneruoti XML tipo RDF ontologiją.</p> <p>3. Pateikti vartotojui informaciją apie generavimo proceso eigą</p>	
<i>Išimtys (klaidos)</i>			
<i>Klaidos sąlyga</i>	<i>Sprendimas</i>		
-	<p>Visos klaidos turi būti registruojamos administratoriui pasiekiamoje byloje. Klaidos įrašą sudaro:</p> <ol style="list-style-type: none"> 1) veiksmas, kurio metu įvyko klaida; 2) vartotojas, kuris vykdė šį veiksmą; 3) klaidos tekstas. <p>Vartotoją taip pat būtina informuoti apie įvykusią klaidą, išvedant į ekraną pranešimą.</p>		
<<include>> PA	<ul style="list-style-type: none"> • PA „Surinkti DB schemos informaciją“ • PA „Surinkti metaDB informaciją“ 		

12 lentelė. PA „Surinkti DB schemos informaciją“ specifikacija

<i>PA ID</i>	6		
<i>PA pavadinimas</i>	PA „Surinkti DB schemos informaciją“		
<i>Autorius</i>	Jaroslav Karpovič	<i>Kas atnaujino?</i>	

<i>Sukūrimo data</i>	2006-06-20	<i>Atnaujinimo data</i>	
<i>Aktoriai</i>	Generavimo agentas		
<i>Aprašymas</i>	Surenkama reikiama informacija iš pasirinktos DB schemų, šių schemų sisteminių lentelių.		
<i>Trigeriai</i>	Vartotojas paleidžia generavimo procesą		
<i>„Prieš“ sąlygos</i>	Sėkmingai praėjo PA „Įvesti generavimo parametrus bei nustatymus“ aprašytas procesas		
<i>„Po“ sąlygos</i>	Užpildyta reikiamų duomenų apie duomenų bazės schemą struktūra		
<i>Pagrindinis scenarijus</i>			
<i>Žingsnio Nr.</i>	<i>Vartotojo veiksmas</i>	<i>Sistemos reakcija (jei ji turi būti)</i>	
P-1.	-	1. Išgaunama reikiama informacija bei užpildoma informacijos apie DB duomenų struktūra	
<i>Išimtys (klaidos)</i>			
<i>Klaidos sąlyga</i>	<i>Sprendimas</i>		
-	<p>Visos klaidos turi būti registruojamos administratoriui pasiekiamoje byloje. Klaidos įrašą sudaro:</p> <ol style="list-style-type: none"> 1) veiksmas, kurio metu įvyko klaida; 2) vartotojas, kuris vykdė šį veiksmą; 3) klaidos tekstas. <p>Vartotoją taip pat būtina informuoti apie įvykusią klaidą, išvedant į ekraną pranešimą.</p>		

13 lentelė. PA „Surinkti metaDB informaciją“ specifikacija

<i>PA ID</i>	7		
<i>PA pavadinimas</i>	PA „Surinkti DB schemas informaciją“		
<i>Autorius</i>	Jaroslav Karpovič	<i>Kas atnaujino?</i>	
<i>Sukūrimo data</i>	2006-06-20	<i>Atnaujinimo data</i>	
<i>Aktoriai</i>	Generavimo agentas		
<i>Aprašymas</i>	Surinkta informacija apie DB schemą praturtinama duomenimis iš metaduomenų bazės.		
<i>Trigeriai</i>	Vartotojas paleidžia generavimo procesą		
<i>„Prieš“ sąlygos</i>	Sėkmingai praėjo PA „Surinkti DB schemas informaciją“ aprašytas procesas		
<i>„Po“ sąlygos</i>	Užpildyta RDF atitinkanti duomenų struktūra.		
<i>Pagrindinis scenarijus</i>			
<i>Žingsnio Nr.</i>	<i>Vartotojo veiksmas</i>	<i>Sistemos reakcija (jei ji turi būti)</i>	
P-1.	-	1. Suirinkta informacija apie DB schema praturtinama semantinėmis žiniomis, paimtomis iš meta-aprašų bazės.	
<i>Išimtys (klaidos)</i>			
<i>Klaidos sąlyga</i>	<i>Sprendimas</i>		
-	<p>Visos klaidos turi būti registruojamos administratoriui pasiekiamoje byloje. Klaidos įrašą sudaro:</p> <ol style="list-style-type: none"> 1) veiksmas, kurio metu įvyko klaida; 2) vartotojas, kuris vykdė šį veiksmą; 3) klaidos tekstas. <p>Vartotoją taip pat būtina informuoti apie įvykusią klaidą, išvedant į ekraną pranešimą.</p>		

14 lentelė. PA „Inicializuoti metaDB“ specifikacija

<i>PA ID</i>	8		
<i>PA pavadinimas</i>	PA „ Inicializuoti metaDB“		
<i>Autorius</i>	Jaroslav Karpovič	<i>Kas atnaujino?</i>	
<i>Sukūrimo data</i>	2006-06-20	<i>Atnaujinimo data</i>	
<i>Aktoriai</i>	Generavimo agentas		
<i>Aprašymas</i>	Pirminio ontologijos pagal pasirinktą DB schemą generavimo metu (kai dar nėra šios DB meta-aprašų), galimas automatinis metaduomenų bazės iniciavimas. Iniciavimo metu užpildoma visa metaDB išskyrus reikšmių atribūtą (žr. 25 pav., “Value”).		
<i>Trigeriai</i>	Vartotojas pažymi, kad paleistas generavimo procesas – pirminis (angl. initial)		
<i>„Prieš“ sąlygos</i>	Sėkmingai praėjo PA „Surinkti DB schemas informaciją “ aprašytas procesas		
<i>„Po“ sąlygos</i>	Užpildyta pasirinkta meta-aprašų bazė		
<i>Pagrindinis scenarijus</i>			
<i>Žingsnio Nr.</i>	<i>Vartotojo veiksmas</i>	<i>Sistemos reakcija (jei ji turi būti)</i>	
P-1.	<ol style="list-style-type: none"> Pažymima, kad paleistas generavimo procesas – pradinis Pasirenkama nauja arba jau egzistuojanti DB meta-aprašų lentelė, kurioje bus pildoma metainformacija. 	1. Pasirinktoje lentelėje užpildoma	
<i>Išimtys (klaidos)</i>			
<i>Klaidos sąlyga</i>	<i>Sprendimas</i>		
-	<p>Visos klaidos turi būti registruojamos administratoriui pasiekiamoje byloje. Klaidos įrašą sudaro:</p> <ol style="list-style-type: none"> veiksmas, kurio metu įvyko klaida; vartotojas, kuris vykdė šį veiksmą; klaidos tekstas. <p>Vartotoją taip pat būtina informuoti apie įvykusią klaidą, išvedant į ekraną pranešimą.</p>		

4 priedas. Semantinės paieškos sistemos funkcijų specifikacija

Paieškos sistemos panaudojimo atvejai (29 pav.) detaliai aprašomi specifikavimo šablonais (15 lentelė- 24 lentelė).

15 lentelė. PA „Vykdėti paiešką“ specifikacija

PA ID	9		
PA pavadinimas	PA „Vykdėti paiešką“		
Autorius	Jaroslav Karpovič	Kas atnaujino?	
Sukūrimo data	2006-06-19	Atnaujinimo data	
Aktoriai	Vartotojas		
Aprašymas	Vykdoma paieška		
Triggeriai	Vartotojas inicijuoja paiešką		
„Prieš“ sąlygos	Vartotojas pasirinkęs paieškos sistemos sąsają (atidaręs paieškos sistemos langą); Vartotojas naršymo ontologijoje (RDF trioletų tinkle) bei suvedęs raktinius paieškos žodžius sudarė vidinę RDF užklausa;		
„Po“ sąlygos	Vartotojui pateikiamas paieškos rezultatas		
<i>Pagrindinis scenarijus</i>			
Žingsnio Nr.	Vartotojo veiksmas		Sistemos reakcija (jei ji turi būti)
P-1.	-		Pateikiamas paieškos langas
P-2.	<ol style="list-style-type: none"> 1. Vartotojas naršydamas srities ontologijoje suveda užklausoje parametrus 2. Vartotojas įveddamas raktinius žodžius papildo užklausoje parametrus 3. Vartotojas paleidžia paieškos mechanizmą 		<ol style="list-style-type: none"> 1. PA „Įvesti paieškos parametrus“ 2. PA „Sudaryti RDF užklausa“ 3. WHILE užklausa yra ne atominė PA „Sudalinti užklausa į sub-užklausas“ PA „Perduoti sub-užklausas resursų agentams“ PA „Performuoti užklausa bei jos sąvokas“ PA „Gauti duomenis iš duomenų šaltinio“ 4. PA „Pateikti apibendrintą rezultatą“
<i>Išimtys (klaidos)</i>			
<i>Klaidos sąlyga</i>		<i>Sprendimas</i>	
-		<p>Visos klaidos turi būti registruojamos administratoriui pasiekiamoje byloje. Klaidos įrašą sudaro:</p> <ol style="list-style-type: none"> 1) veiksmas, kurio metu įvyko klaida; 2) vartotojas, kuris vykdė šį veiksmą; 3) klaidos tekstas. <p>Vartotoją taip pat būtina informuoti apie įvykusią klaidą, išvedant į ekraną pranešimą.</p>	
<<include>> PA	<ul style="list-style-type: none"> • PA „Įvesti paieškos parametrus“ • PA „Sudaryti RDF užklausa“ • PA „Sudalinti užklausa į sub-užklausas“ • PA „Perduoti sub-užklausas resursų agentams“ • PA „Performuoti užklausa bei jos sąvokas“ • PA „Gauti duomenis iš duomenų šaltinio“ • PA „Pateikti apibendrintą rezultatą“ 		

16 lentelė. PA „Įvesti paieškos parametrus“ specifikacija

PA ID	10		
PA pavadinimas	PA „Įvesti paieškos parametrus“		
Autorius	Jaroslav Karpovič	Kas atnaujino?	
Sukūrimo data	2006-06-19	Atnaujinimo data	
Aktoriai	Vartotojas, užklausų agentas		
Aprašymas	Užklausų agentas, pagal vartotojo suvestą (tiesiogiai: suvesti raktiniai žodžiai; netiesiogiai: naršymas srities (Domain) RDF ontologijoje) informaciją sudaro užklausos parametrus, kurie perduodami tolesnei veiklai ir lemia pateikiamą paieškos rezultatą		
Trigeriai	Vartotojas iniciavo paiešką (paspaudė „Ieškoti“ mygtuką)		
„Prieš“ sąlygos	Vartotojas pasirinkęs paieškos sistemos sąsają (atidaręs paieškos sistemos langą); Vartotojas naršymo ontologijoje (RDF trioletų tinkle) bei suvedęs raktinius paieškos žodžius sudarė vidinę RDF užklausą;		
„Po“ sąlygos	Suformuoti užklausos parametrai		
Pagrindinis scenarijus			
Žingsnio Nr.	Vartotojo veiksmas	Sistemos reakcija (jei ji turi būti)	
P-1.	1. Vartotojas, naršydamas srities (Domain) ontologijoje pasirinko tam tikrą jos šaką bei įvedė paieškos raktinius žodžius 2. Vartotojas paspaudė „Ieškoti“ mygtuką	1. Parametrai surenkami 2. Parametrai susisteminami	
Išimtytys (klaidos)			
Klaidos sąlyga		Sprendimas	
-		Visos klaidos turi būti registruojamos administratoriui pasiekiamoje byloje. Klaidos įrašą sudaro: 1) veiksmas, kurio metu įvyko klaida; 2) vartotojas, kuris vykdė šį veiksmą; 3) klaidos tekstas. Vartotoją taip pat būtina informuoti apie įvykusią klaidą, išvedant į ekraną pranešimą.	
<<include>> PA			

17 lentelė. PA „Sudaryti RDF užklausą“ specifikacija

PA ID	11		
PA pavadinimas	PA „Sudaryti RDF užklausą“		
Autorius	Jaroslav Karpovič	Kas atnaujino?	
Sukūrimo data	2006-06-19	Atnaujinimo data	
Aktoriai	Užklausų agentas		
Aprašymas	Užklausų agentas pagal vartotojo pateiktus parametrus sudaro RDF užklausą		
Trigeriai	Vartotojas iniciavo paiešką (paspaudė „Ieškoti“ mygtuką)		
„Prieš“ sąlygos	Suformuoti užklausos parametrai (užklausų agento)		
„Po“ sąlygos	Suformuota RDF užklausą		
Pagrindinis scenarijus			
Žingsnio Nr.	Vartotojo veiksmas	Sistemos reakcija (jei ji turi būti)	
P-1.	-	1. Užkraunamas RDF užklausos šablonas	
P-2.	-	1. Šablonas užpildomas parametrais	

<i>Išimtys (klaidos)</i>	
<i>Klaidos sąlyga</i>	<i>Sprendimas</i>
-	<p>Visos klaidos turi būti registruojamos administratoriui pasiekiamoje byloje. Klaidos įrašą sudaro:</p> <ol style="list-style-type: none"> 1) veiksmas, kurio metu įvyko klaida; 2) vartotojas, kuris vykdė šį veiksmą; 3) klaidos tekstas. <p>Vartotoją taip pat būtina informuoti apie įvykusią klaidą, išvedant į ekraną pranešimą.</p>
<<include>> PA	

18 lentelė. PA „Sudalinti užklausa į sub-užklausas“ specifikacija

<i>PA ID</i>	12		
<i>PA pavadinimas</i>	PA „Sudalinti užklausa į sub-užklausas“		
<i>Autorius</i>	Jaroslav Karpovič	<i>Kas atnaujino?</i>	
<i>Sukūrimo data</i>	2006-06-19	<i>Atnaujinimo data</i>	
<i>Aktoriai</i>	Užklausių agentas		
<i>Aprašymas</i>	Sudėtinga RDF užklausa sudalinama į paprastas (atomines) sub-užklausas		
<i>Trigeriai</i>	Sudaryta RDF užklausa		
<i>„Prieš“ sąlygos</i>	Paruošta RDF užklausa		
<i>„Po“ sąlygos</i>	Užklausa sudalinta į atomines (paprastas) sub-užklausas		
<i>Pagrindinis scenarijus</i>			
<i>Žingsnio Nr.</i>	<i>Vartotojo veiksmas</i>	<i>Sistemos reakcija (jei ji turi būti)</i>	
P-1.	-	1. Išskirti atominį parametą	
P-2.	-	1. Sudaryti užklausa vien išskirtam parametrai	
<i>Išimtys (klaidos)</i>			
<i>Klaidos sąlyga</i>	<i>Sprendimas</i>		
-	<p>Visos klaidos turi būti registruojamos administratoriui pasiekiamoje byloje. Klaidos įrašą sudaro:</p> <ol style="list-style-type: none"> 1) veiksmas, kurio metu įvyko klaida; 2) vartotojas, kuris vykdė šį veiksmą; 3) klaidos tekstas. <p>Vartotoją taip pat būtina informuoti apie įvykusią klaidą, išvedant į ekraną pranešimą.</p>		
<<include>> PA			

19 lentelė. PA „Perduoti sub-užklausas resursų agentams“ specifikacija

<i>PA ID</i>	13		
<i>PA pavadinimas</i>	PA „Perduoti sub-užklausas resursų agentams“		
<i>Autorius</i>	Jaroslav Karpovič	<i>Kas atnaujino?</i>	
<i>Sukūrimo data</i>	2006-06-19	<i>Atnaujinimo data</i>	
<i>Aktoriai</i>	Užklausių agentas, resursų agentas		
<i>Aprašymas</i>	Užklausių agentas perduoda vieną sub-užklausa resursų agentams (vykdymui)		
<i>Trigeriai</i>	Sudaryta sub-užklausa		
<i>„Prieš“ sąlygos</i>	Resursų agentas sudalino užklausa į atomines		
<i>„Po“ sąlygos</i>	Paprasta (atominė) užklausa perduota visiems sistemoje esantiems resursų agentams vykdymui		
<i>Pagrindinis scenarijus</i>			
<i>Žingsnio Nr.</i>	<i>Vartotojo veiksmas</i>	<i>Sistemos reakcija (jei ji turi būti)</i>	

P-1.	-	1. Atpažįstamas priregistruotas resursų agentas, galintis vykdyti užklausas
P-2.		1. Atominė užklausa perduodama resursų agentui
<i>Išimtys (klaidos)</i>		
<i>Klaidos sąlyga</i>		<i>Sprendimas</i>
-		Visos klaidos turi būti registruojamos administratoriui pasiekiamoje byloje. Klaidos įrašą sudaro: 1) veiksmas, kurio metu įvyko klaida; 2) vartotojas, kuris vykdė šį veiksma; 3) klaidos tekstas. Vartotoją taip pat būtina informuoti apie įvykusią klaidą, išvedant į ekraną pranešimą.
<<include>> PA		

20 lentelė. PA „Performuoti užklausą bei jos sąvokas“ specifikacija

PA ID	14		
PA pavadinimas	PA „Performuoti užklausą bei jos sąvokas“		
Autorius	Jaroslav Karpovič	Kas atnaujino?	
Sukūrimo data	2006-06-19	Atnaujinimo data	
<i>Aktoriai</i>	Resursų agentas		
<i>Aprašymas</i>	Pagal tam tikro duomenų šaltinio resursų agente (Resource Agent) apibrėžtas taisykles [2], užklausa paverčiama konkrečiam šaltiniui suprantama kalba (tarkime, reliacinei DB tai būtų SQL kalbos SELECT komanda) o RDF parametrai paverčiami tos kalbos parametrais (sąvokomis)		
<i>Trigeriai</i>	Gauta atominė užklausa		
„Prieš“ sąlygos	Užklausų agentas pateikė resursų agentui vieną sub-užklausą		
„Po“ sąlygos	Užklausa „išverčiama“ į duomenų šaltiniui suprantamą kalbą		
<i>Pagrindinis scenarijus</i>			
<i>Žingsnio Nr.</i>	<i>Vartotojo veiksmas</i>	<i>Sistemos reakcija (jei ji turi būti)</i>	
P-1.	-	1. Išversti užklausą į duomenų šaltiniui suprantamą kalbą pagal vidines taisykles;	
P-2.		1. Pagal konteksto susietumo (Concept Mapping) RDF bylos taisykles pakeisti užklausos parametrus į duomenų saugyklai suprantamus	
P-3.		1. Įstatyti duomenų saugyklai suprantamus parametrus į suformuotą užklausą	
<i>Išimtys (klaidos)</i>			
<i>Klaidos sąlyga</i>		<i>Sprendimas</i>	
-		Visos klaidos turi būti registruojamos administratoriui pasiekiamoje byloje. Klaidos įrašą sudaro: 1) veiksmas, kurio metu įvyko klaida; 2) vartotojas, kuris vykdė šį veiksma; 3) klaidos tekstas. Vartotoją taip pat būtina informuoti apie įvykusią klaidą, išvedant į ekraną pranešimą.	
<<include>> PA			

21 lentelė. PA „Gauti duomenis iš duomenų šaltinio" specifikacija

PA ID	15		
PA pavadinimas	PA „Gauti duomenis iš duomenų šaltinio“		
Autorius	Jaroslav Karpovič	Kas atnaujino?	
Sukūrimo data	2006-06-19	Atnaujinimo data	
Aktoriai	Resursų agentas		
Aprašymas	Vykdančią sudarytą užklausa, gaunami duomenys iš duomenų saugyklos		
Trigeriai	Sudaryta duomenų šaltiniui suprantama užklausa		
„Prieš“ sąlygos	Resursų agentas suformavo duomenų šaltiniui suprantamą užklausa		
„Po“ sąlygos	Resursų agentas gauna užklauskos rezultata		
Pagrindinis scenarijus			
Žingsnio Nr.	Vartotojo veiksmas	Sistemos reakcija (jei ji turi būti)	
P-1.	-	1. IF PA „Valdyti turinio pateikimą pagal vartotojo rolę sistemoje“ su parametrais (vartotojo rolę, duomenų šaltinio informacijos mazgas) pranešė apie vartotojo tinkamas teises pageidaujama turiniui gauti THEN 1.1. Vykdyti užklausa 1.2. PA „Pateikti rezultata“	
Išimtys (klaidos)			
Klaidos sąlyga		Sprendimas	
-		Visos klaidos turi būti registruojamos administratoriui pasiekiamoje byloje. Klaidos įrašą sudaro: 1) veiksmas, kurio metu įvyko klaida; 2) vartotojas, kuris vykdė šį veiksmą; 3) klaidos tekstas. Vartotoją taip pat būtina informuoti apie įvykusią klaidą, išvedant į ekraną pranešimą.	
<<include>> PA	<ul style="list-style-type: none"> • PA „Valdyti turinio pateikimą pagal vartotojo rolę sistemoje“ • PA „Pateikti rezultata“ 		

22 lentelė. PA „Valdyti turinio pateikimą pagal vartotojo rolę sistemoje" specifikacija

PA ID	16		
PA pavadinimas	PA „Valdyti turinio pateikimą pagal vartotojo rolę sistemoje“		
Autorius	Jaroslav Karpovič	Kas atnaujino?	
Sukūrimo data	2006-06-19	Atnaujinimo data	
Aktoriai	Resursų agentas		
Aprašymas	Pasinaudojus vidiniais duomenų saugyklos komponentais bei jų apibrėžtomis taisyklėmis, nustatyti, ar paiešką atliekantis vartotojas turi teisę matyti tam tikrą turinį (jo elementą)		
Trigeriai	Inicijuojamas duomenų gavimas iš saugyklos		
„Prieš“ sąlygos	<ul style="list-style-type: none"> • Sudaryta duomenų šaltiniui suprantama užklausa • Įmanoma pasiekti duomenų šaltinio saugumo posistemio mechanizmą 		
„Po“ sąlygos	Pateikiamas atsakymas į klausimą ar gali būti vartotojui pateiktas ieškomas turinys		
Pagrindinis scenarijus			
Žingsnio Nr.	Vartotojo veiksmas	Sistemos reakcija (jei ji turi būti)	

P-1.	-	1. Perduoti duomenų šaltinio saugumo mechanizmui vartotojo rolę (vienas iš užklauso parametru)
P-2.	-	1. Perduoti duomenų šaltinio saugumo mechanizmui reikalaujamos informacijos mazgą (reliacinės DB atveju tai būtų schema, lentelė, lentelės laukas)
P-3.	-	1. IF teisės suteiktos THEN 1.1. Gražinti teigiamą atsakymą ELSE 1.2. Gražinti neigiamą atsakymą
<i>Išimtys (klaidos)</i>		
<i>Klaidos sąlyga</i>		<i>Sprendimas</i>
-		Visos klaidos turi būti registruojamos administratoriui pasiekiamoje byloje. Klaidos įrašą sudaro: 1) veiksmas, kurio metu įvyko klaida; 2) vartotojas, kuris vykdė šį veiksmą; 3) klaidos tekstas. Vartotoją taip pat būtina informuoti apie įvykusią klaidą, išvedant į ekraną pranešimą.
<<include>> PA		

23 lentelė. PA „Pateikti rezultata“specifikacija

PA ID	17		
PA pavadinimas	PA „Pateikti rezultata“		
Autorius	Jaroslav Karpovič	Kas atnaujino?	
Sukūrimo data	2006-06-19	Atnaujinimo data	
Aktoriai	Resursų agentas, užklauso agentas		
Aprašymas	Perduoti užklauso agentui sub-užklauso vykdymo rezultata: nuoroda į šaltinį (šaltinius), trumpas kontekstas, kuriame rasta sąvoka bei metaduomenys (iš konceptų ontologijos)		
Trigeriai	Vykdoma užklausa gražino bent vieną rezultata		
„Prieš“ sąlygos	Įvykdyta užklausa		
„Po“ sąlygos	Užklauso (sub-užklauso) vykdymo rezultatas perduotas užklauso agentui		
<i>Pagrindinis scenarijus</i>			
Žingsnio Nr.	Vartotojo veiksmas	Sistemos reakcija (jei ji turi būti)	
P-1.	-	1. Resursų agentas užklauso agentui perduoda užklauso vykdymo rezultata sekančiu pavidalu: nuoroda į šaltinį (šaltinius), trumpas kontekstas, kuriame rasta sąvoka bei metaduomenys (iš konceptų ontologijos)	
<i>Išimtys (klaidos)</i>			
<i>Klaidos sąlyga</i>		<i>Sprendimas</i>	
-		Visos klaidos turi būti registruojamos administratoriui pasiekiamoje byloje. Klaidos įrašą sudaro: 1) veiksmas, kurio metu įvyko klaida; 2) vartotojas, kuris vykdė šį veiksmą;	

	3) klaidos tekstas. Vartotoją taip pat būtina informuoti apie įvykusią klaidą, išvedant į ekraną pranešimą.
<<include>> PA	

24 lentelė. PA „Pateikti apibendrintą rezultatą“ specifikacija

PA ID	18		
PA pavadinimas	PA „Pateikti apibendrintą rezultatą“		
Autorius	Jaroslav Karpovič	Kas atnaujino?	
Sukūrimo data	2006-06-19	Atnaujinimo data	
Aktoriai	Užklausų agentas		
Aprašymas	Surenkamas visų sub-užklausų veikimo rezultatas, apjungiamas, surūšiuojamas, išrikiuojamas ir pateikiamas vartotojui		
Triggeriai	Įvykdyta paskutinė užklausa sub-užklausa		
„Prieš“ sąlygos	Įvykdytos visos užklausa sub-užklausa bei gautas kiekvienos sub-užklausa rezultatas		
„Po“ sąlygos	Paieškos lange pateikiamas apjungtas, surūšiuotas ir išrikiuotas užklausa veikimo rezultatas		
<i>Pagrindinis scenarijus</i>			
<i>Žingsnio Nr.</i>	<i>Vartotojo veiksmas</i>	<i>Sistemos reakcija (jei ji turi būti)</i>	
P-1.	-	1. Sub-užklausų įvykdymo rezultatai išrikiuojami pirmiausia pagal atitikimo visiems paieškos parametrų (raktiniai žodžiai, pasirinkta srities ontologijos šaka) skaičių, sudarant grupes	
P-2.		1. Sub-užklausų įvykdymo rezultatai grupės viduje išrikiuojami pagal ieškomų parametrų pasikartojimo dažnį	
P-3.		1. IF kelių sub-užklausų parametro paieškos gražintas kontekstas sutampa (tarkime dvi sąvokos pasirodo tame pačiame sakinyje) – apjungti sąvokų kontekstą (rodyti vieną).	
<i>Išimtys (klaidos)</i>			
<i>Klaidos sąlyga</i>	<i>Sprendimas</i>		
-	Visos klaidos turi būti registruojamos administratoriui pasiekiamoje byloje. Klaidos įrašą sudaro: 1) veiksmas, kurio metu įvyko klaida; 2) vartotojas, kuris vykdė šį veiksma; 3) klaidos tekstas. Vartotoją taip pat būtina informuoti apie įvykusią klaidą, išvedant į ekraną pranešimą.		
<<include>> PA			
<i>Veiklos taisyklės</i>			
<i>Atviri klausimai</i>			

5 priedas. Pavyzdinė metaduomenų bazė užpildyta duomenimis

object_id	object_type	objects_ discr_element	parameter	value	initial	visible
straipsniuIS	database	Label	LT	Straipsnių IS	FALSE	TRUE
str.Straipsnis	db table	Label	LT	Straipsnis	TRUE	TRUE
str.Vertimas	db table	Label	LT	Vertimas	FALSE	TRUE
str.Zmogus	db table	Label	LT	Žmogus	FALSE	TRUE
str.Straipsnis.Id	table column	Label	LT		FALSE	TRUE
str.Straipsnis.Pavadinimas	table column	Label	LT	Pavadinimas	FALSE	TRUE
str.Straipsnis.SukurimoData	table column	Label	LT	Sukūrimo data	FALSE	TRUE
str.Straipsnis.Autorius	table column	Label	LT	Autorius	FALSE	TRUE
str.Straipsnis.Bendraautorius	table column	Label	LT	Bendraautorius	FALSE	TRUE
str.Straipsnis.Ivertinimas	table column	Label	LT	Įvertinimas	FALSE	TRUE
str.Straipsnis.Kalba	table column	Label	LT	Kalba	FALSE	TRUE
str.Vertimas.Originalo_id	table column	Label	LT		FALSE	TRUE
str.Vertimas.Vertimo_id	table column	Label	LT		FALSE	TRUE
str.Vertimas.Id	table column	Label	LT		FALSE	TRUE
str.Zmogus.Id	table column	Label	LT		FALSE	TRUE
str.Zmogus.Vardas	table column	Label	LT	Vardas	FALSE	TRUE
str.Zmogus.Pavarde	table column	Label	LT	Pavardė	FALSE	TRUE
terel_str.Straipsnis.Id	table-column relation	Label	LT		FALSE	TRUE
terel_str.Straipsnis.Pavadinimas	table-column relation	Label	LT	turi	FALSE	TRUE
terel_str.Straipsnis.SukurimoData	table-column relation	Label	LT	turi	FALSE	TRUE
terel_str.Straipsnis.Autorius	table-column relation	Label	LT	turi	FALSE	TRUE
terel_str.Straipsnis.Bendraautorius	table-column relation	Label	LT	turi	FALSE	TRUE
terel_str.Straipsnis.Ivertinimas	table-column relation	Label	LT	turi	FALSE	TRUE
terel_str.Straipsnis.Kalba	table-column relation	Label	LT	turi	FALSE	TRUE
terel_str.Vertimas.Originalo_id	table-column relation	Label	LT		FALSE	TRUE
terel_str.Vertimas.Vertimo_id	table-column relation	Label	LT		FALSE	TRUE
terel_str.Vertimas.Id	table-column relation	Label	LT		FALSE	TRUE
terel_str.Zmogus.Id	table-column relation	Label	LT		FALSE	TRUE
terel_str.Zmogus.Vardas	table-column relation	Label	LT	turi	FALSE	TRUE
terel_str.Zmogus.Pavarde	table-column relation	Label	LT	turi	FALSE	TRUE
FK_Straipsnis_Zmogus	table-table relation	ToChildName	LT	yra autorius	FALSE	TRUE
FK_Straipsnis_Zmogus	table-table relation	ToParentName	LT	turi autorių	FALSE	TRUE
FK_Straipsnis_Zmogus1	table-table relation	ToChildName	LT	yra bendraautorius	FALSE	TRUE
FK_Straipsnis_Zmogus1	table-table relation	ToParentName	LT	turi bendraautorių	FALSE	TRUE
FK_Vertimas_Straipsnis	table-table relation	ToChildName	LT	turi vertimą	FALSE	TRUE
FK_Vertimas_Straipsnis	table-table relation	ToParentName	LT		FALSE	TRUE
FK_Vertimas_Straipsnis1	table-table relation	ToChildName	LT		FALSE	TRUE
FK_Vertimas_Straipsnis1	table-table relation	ToParentName	LT	yra	FALSE	TRUE
straipsniuIS	database	Label	EN	Article IS	FALSE	TRUE
str.Straipsnis	db table	Label	EN	Article	FALSE	TRUE
str.Vertimas	db table	Label	EN	Translation	FALSE	TRUE
str.Zmogus	db table	Label	EN	Person	FALSE	TRUE
str.Straipsnis.Id	table column	Label	EN		FALSE	TRUE
str.Straipsnis.Pavadinimas	table column	Label	EN	Title	FALSE	TRUE
str.Straipsnis.SukurimoData	table column	Label	EN	creation date	FALSE	TRUE
str.Straipsnis.Autorius	table column	Label	EN	Author	FALSE	TRUE
str.Straipsnis.Bendraautorius	table column	Label	EN	Co-Author	FALSE	TRUE
str.Straipsnis.Ivertinimas	table column	Label	EN	Rating	FALSE	TRUE
str.Straipsnis.Kalba	table column	Label	EN	Language	FALSE	TRUE
str.Vertimas.Originalo_id	table column	Label	EN		FALSE	TRUE
str.Vertimas.Vertimo_id	table column	Label	EN		FALSE	TRUE
str.Vertimas.Id	table column	Label	EN		FALSE	TRUE
str.Zmogus.Id	table column	Label	EN		FALSE	TRUE
str.Zmogus.Vardas	table column	Label	EN	Name	FALSE	TRUE
str.Zmogus.Pavarde	table column	Label	EN	Surname	FALSE	TRUE

object_id	object_type	objects_discr_element	parameter	value	initial	visible
terel_str.Straipsnis.Id	table-column relation	Label	EN		FALSE	TRUE
terel_str.Straipsnis.Pavadinimas	table-column relation	Label	EN	has	FALSE	TRUE
terel_str.Straipsnis.SukurimoData	table-column relation	Label	EN	has	FALSE	TRUE
terel_str.Straipsnis.Autorius	table-column relation	Label	EN	has	FALSE	TRUE
terel_str.Straipsnis.Bendraautorius	table-column relation	Label	EN	has	FALSE	TRUE
terel_str.Straipsnis.Ivertinimas	table-column relation	Label	EN	has	FALSE	TRUE
terel_str.Straipsnis.Kalba	table-column relation	Label	EN	has	FALSE	TRUE
terel_str.Vertimas.Originalo_id	table-column relation	Label	EN		FALSE	TRUE
terel_str.Vertimas.Vertimo_id	table-column relation	Label	EN		FALSE	TRUE
terel_str.Vertimas.Id	table-column relation	Label	EN		FALSE	TRUE
terel_str.Zmogus.Id	table-column relation	Label	EN		FALSE	TRUE
terel_str.Zmogus.Vardas	table-column relation	Label	EN	has	FALSE	TRUE
terel_str.Zmogus.Pavarde	table-column relation	Label	EN	has	FALSE	TRUE
FK_Straipsnis_Zmogus	table-table relation	ToChildName	EN	is author	FALSE	TRUE
FK_Straipsnis_Zmogus	table-table relation	ToParentName	EN	has author	FALSE	TRUE
FK_Straipsnis_Zmogus1	table-table relation	ToChildName	EN	is co-author	FALSE	TRUE
FK_Vertimas_Straipsnis	table-table relation	ToChildName	EN	has translation	FALSE	TRUE
FK_Vertimas_Straipsnis	table-table relation	ToParentName	EN		FALSE	TRUE
FK_Vertimas_Straipsnis1	table-table relation	ToChildName	EN		FALSE	TRUE
FK_Vertimas_Straipsnis1	table-table relation	ToParentName	EN	is a	FALSE	TRUE
FK_Straipsnis_Zmogus1	table-table relation	ToParentName	EN	has co-author	FALSE	TRUE

**6 priedas. Straipsnis „Ontologijų išgavimo
įrankio architektūra“, J. Karpovič,
V. Taujanskas**

ONTOLOGIJŲ IŠGAVIMO ĮRANKIO ARCHITEKTŪRA

Jaroslav Karpovič, Vytautas Taujanskas

Kauno technologijos universitetas, Informacijos sistemų katedra

Studentų 50-311, 51368 Kaunas

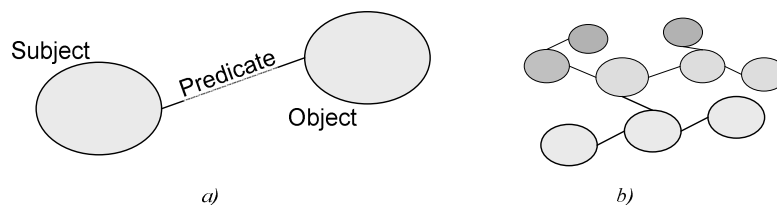
Straipsnyje pristatomos ontologijos kaip informacijos formalizavimo būdas. Plačiau aprašomos galimybės ir problemos formalų aprašą kuriant automatinio būdu. Pateikiami esminiai tokio metodo žingsniai bei galima architektūra jiems įgyvendinti.

1. Įvadas

Internete esančios informacijos formalizavimo idėja materializavosi į priemones semantiniams žiniatinklams sukurti. Vien tik žmogui suprantama informacija yra sunkiai pakartotinai panaudojama, tad strategijos žinių formalizavimui leidžia tą pačią informaciją pasiekti ne tik žmogui, bet ir kompiuteriui. Webservisai suteikia galimybę pasinaudoti API (application programming interface – programinės bibliotekos) kuriant programas, o ontologijos – perimti žinias ir jas pakartotinai panaudoti be ypatingų pastangų.

Ontologija – tai bandymo apibrėžti išsamią ir griežtą srities konceptualiąją schemą darbo rezultatas. Dažniausiai tai hierarchinė duomenų struktūra, kurioje yra susijusios esybės su jas siejančiais ryšiais bei nagrinėjamos dalykinės srities taisyklės [13].

Pats paprasčiausias ontologijos vaizdavimas – RDF notacijoje siūlomoje W3C konsorciumo [12]. RDF (Resource Description Framework) – universali informacijos vaizdavimo žiniatinklyje kalba. RDF – tai tripletų (<subject> <predicate> <object>) tinklas (1 pav.). Tripletas – tai dvi klasės (<subject>, <object>) ir jas siejantis ryšys (<predicate>), kuris kartais vadinamas tripleto savybe (<property>) [12].

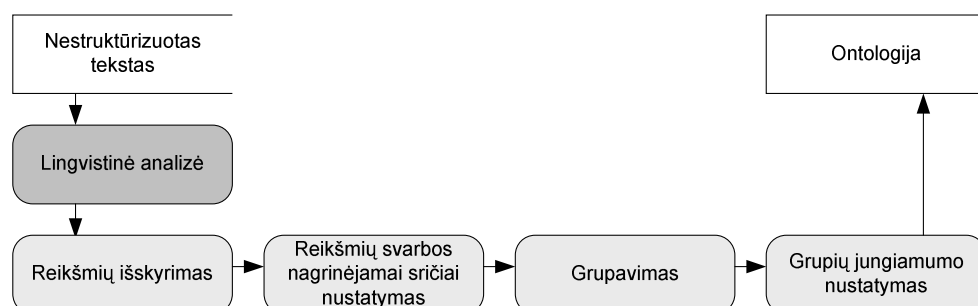


56 pav. RDF tripletas bei tripletų tinklas

Patogus vaizdinis informacijos pateikimas bei galimybės automatizuoti srities sąvokų ištraukimą leidžia nagrinėti automatinį ontologijos išskyrimą. Šiame straipsnyje bendrais bruožais aptariamos tokio ontologijos kūrimo būdo problemos ir aprašomas galimas ontologijos sudarymo procesas.

2. Klasikinis ontologijos išskyrimo procesas panaudojus lingvistinę analizę

Išvystyta nemažai automatinio ontologijos išgavimo iš laisvo teksto metodų. Dalį jų bent iš dalies realizuoja tam tikri ontologijos išskyrimo įrankiai, kita dalis – tik teoriniai [7]. Bendru atveju ontologijos išskyrimo procesas yra toks (5 pav.): iš teksto išskiriamos reikšmės, nustatoma jų svarba nagrinėjamai sričiai, išskirtos reikšmės sugrupuojamos bei nustatomas grupių jungiamumas [1, 5].



57 pav. Klasikinis ontologijos išskyrimo procesas

Išskyrimo proceso priešaky įterptas lingvistinės analizės blokas. Tai sudėtingą vidinę sandarą turintis blokas – dažnai atskiras posistemis ar net sistema. Lingvistinės analizės blokas mums suteikia žinias apie žodžio formą ir leidžia tas žinias suvienareikšminti – gauti nedviprasmišką informaciją. Detaliau kiekvienas žingsnis bei jame sprendžiamos problemos aptariamos tolesniuose skyreliuose.

2.1 Lingvistinė analizė bei jos lygiai

Proceso automatizavimui bei išskiriamų ontologijų kokybei pagerinti reikalingas mechanizmas, mokantis išnagrinėti sakinio struktūrą, o tiksliau, atliekantis sakinio lingvistinę analizę. Tam tikslui naudojamas lingvistinės analizės blokas (5 pav.).

Analizuodamas laisvą, nestruktūrizuotą tekstą lingvistinės analizės blokas suteikia vertingų žinių tolimesniems proceso etapams. Šio žingsnio išėiga – struktūrizuotas, lingvistiškai sužymėtas tekstas. Galimas trijų lygių sužymėjimas [1, 5, 6]:

- žodžių lygis: kalbos dalies bei žodžių formų struktūros informacija (morfologinė analizė);
- frazės lygis: žodžių junginių struktūra, pažymint pagrindinį bei pažymimąjį žodį (head-modifier);
- sakinio lygis: semantinė sakinio analizė.

Kalbos dalies informacija (daiktavardis, veiksmažodis, būdvardis ir t.t.) turi būti tiksliai nustatytas iš konteksto, kadangi tas pats žodis priklausomai nuo konteksto gali būti skirtinga kalbos dalimi. Kai kurių kalbų morfologinė informacija (žodžių formų struktūros) yra labai turtinga, todėl svarbu tiksliai nustatyti sąvokų bendrines formas [6, 11]. Šie uždaviniai sprendžiami žodžių lygio analizėje.

Frazės lygyje nagrinėjamos sintaksinės sakinio struktūros (taisyklės, apibrėžiančios sakinio struktūrą). Sakinys yra sudalinamas į kelių žodžių grupes, kurių kiekviena būtinai susideda iš būdvardžio, daiktavardžio arba prielinksnio priešakyje [1, 6]. Be sudalinimo, nagrinėjama vidinė sudarytos frazės struktūra nurodant pagrindinį bei jį pažyminčiuosius žodžius.

Sakiniai dažnai nebūna pavieniai, bet susideda iš pagrindinio bei vieno ar kelių šalutinių sakinių. Sakinio lygio lingvistinėje analizėje nagrinėjami sakiniai (pagrindinis, šalutiniai) su būtina jų (nors kartais ir paslėpta, kas dar labiau apsunkina analizę) sudėtine dalimi – tariniu ir jo argumentais: veiksmu (subjektas), papildiniu (objektas).

2.2 Reikšmių išskyrimas

Ontologijos išgavimo proceso reikšmių išskyrimo žingsnio įeiga – lingvistiškai sužymėtas tekstas, kuriame iš žodžių visumos atrenkami esminiai terminai atmetant nereikšminius žodžius.

Reikšmių išskyrimas arba atmetimas dažnai tapatinamas ne su fiziniu reikšmių perkėlimu į tam tikrą sąrašą ir nereikalingų reikšmių išmetimu, bet su atitinkamu teksto sužymėjimu, priskiriant žodžiams tam tikras žymes. Toks požiūris turi savo privalumų, kadangi kitiems žingsniams nereikia analizuoti iš naujo teksto ieškant išskirtų reikšmių konteksto (tarkime šio proceso grupių jungiamumo nustatymo žingsnyje, 5 pav.). Kai kurie autoriai teigia, kad iš viso negalima at mesti jokios informacijos, nes taip prarandama esmė, prarandamas tikslumas [10]. Šio požiūrio šalininkai tvirtina, kad geriau turėti daugiau informacijos, net ir tuo atveju, kai atrodo, jog ta informacija yra mažai verta.

Išskiriant reikšmes didžiausias dėmesys turi būti skiriamas veiksmui (daiktavardis) bei jo požymius (veiksma, būseną, jų abiejų pokytį, ypatybę ir pan.) nusakantiems tariniams. Žingsnio išėiga – išskirtų reikšmių sąrašas (galimos būsimos ontologijos klasės) su jas apibūdinančiais požymiais (klasių ryšio komponentė, <predicate>) [1, 2].

Tarkime kompiuterinės klaviatūros naudojimosi instrukcijoje (neilgas tekstas, pora pastraipų) „kompiuterinė klaviatūra“ ir „mygtukas“ yra daiktavardžiai, o tai reiškia, kad jie ontologijoje bus esybėmis (klasėmis). Šioje vietoje reikalingas mechanizmas galintis iš viso teksto išskirti tik daiktavardžius (arba daiktavardžių junginius) kaip potencialiai galimas būsimas ontologijos klases bei juos apibūdinančius žodžius – ryšius tarp klasių. Šią funkciją atlieka lingvistinės analizės blokas: lingvistinis teksto sužymėjimas leidžia be žmogaus įsikišimo atskirti esybes nuo ryšių ir esybių apibūdinimo.

2.3 Reikšmių svarbos nagrinėjimai sričiai nustatymas

Tai statistinis procesas, kuris nustato reikšmės tiesioginį ryšį nagrinėjamoje srityje. Proceso išėiga sudaro anksčiau išskirtos esminės reikšmės su statistiniais įverčiais, nurodančiais jų aktualumą (svarbą) nagrinėjamuose tekstuose. Paprastai ontologija sudaroma tam tikrai sričiai (medicina, statyba ir t.t.). Tam tikslui yra nagrinėjami dominančios srities tekstai. Šiame žingsnyje pagal žodžio (termino) pasikartojimo dažnį tekstuose nustatoma jo svarba nagrinėjamai sričiai. Teigiama, kad žodžio reikšmė yra tiesiogiai susijusi su kontekstu, kuriame tas žodis naudojamas [2].

Šiame žingsnyje kartais apsiribojama tiesiog sąvokos pasikartojimo dažnio tekste įvertinimu, tačiau įmanoma daug tiksliau išskirti tas sąvokas, kurios suteikia daugiau nagrinėjamos srities informacijos, tokiu būdu ne tik sumažinant duomenų kiekį, bet ir pagerinant kitame žingsnyje atliekamą klasifikaciją [21]. Egzistuoja statistiniai metodai, pvz. TFIDF (angl. term frequency-inverse document frequency), chi kvadrato ir kt. [21], kurie dar yra vadinami panašumo matavimo metrikomis, kadangi statiškai išmatuoja atstumą tarp sąvokų teksto vienetė (straipsnis, paragrafas, sakiny) bei suteikia atitinkamus svorius [2, 21].

2.4 Reikšmių grupavimas

Numatomų klasių sistematikai (taksonomijai) nustatyti prieš tai išskirtos reikšmės yra grupuojamos. Identiškos arba labai artimos reikšmės yra apjungiamos, taip sudarant vieną galimą klasę su daugeliu ją apibūdinančių požymių. Būtent šios galimos klasės su požymių aibe ir sudaro proceso darbo rezultatą.

Šiame žingsnyje lingvistinės žinios naudojamos to paties žodžio leksemoms (linksniai, laikai ir panašiai) apjungti į vieną grupę. Aukščiau nagrinėtame pavyzdyje apie kompiuterinę klaviatūrą nebus visą laiką naudojamas žodis „mygtukas“ (bendrinėje formoje), bet pasitaikys „mygtuką“ (pvz., „paspausti mygtuką“), „mygtukų“ (pvz., „mygtukų kombinacija“), kurių bendrinė forma yra ta pati, bet statistiniai metodai to neatpažins.

Jeigu leksemų sutraukimo į vieną grupę būtinumas yra nediskutuotinas, tai dėl sinonimų apjungimo nėra tokios vieningos nuomonės. Tai yra grupavimo žingsnio problema, nes nėra aiškiai nubrėžtos reikšmių sutraukimo į grupę ribos. Suprantama, kad, jei nagrinėsime anksčiau paminėtą klaviatūros aprašą kartu su kito gamintojo aprašu, kur vietoje žodžio „mygtukas“ panaudotas jo sinonimas „klavišas“, neapjungus šiuos sinonimus į vieną grupę, gausime ontologiją panašią į: „klaviatūra turi klavišus ir mygtukus“. Šiuo atveju apjungimas yra reikalingas, bet egzistuoja atvejų kai toks apjungimas lems informacijos praradimą arba iškraipymą.

Visai priešinga situacija su homonimais. Dėl savo vienodos rašybos (nors ir skirtingos reikšmės) šie žodžiai tikrai bus įtraukti į vieną grupę. Toks apjungimas lems ontologijos netikslumą arba netgi prieštaravimą.

2.5 Grupių jungiamumo nustatymas

Tai statistinis procesas, nustatantis ryšius tarp šio proceso grupavimo žingsnio (žr. 2.4 skyrelį) sudarytų reikšmių grupių – numatomų ontologijos klasių. Šioje vietoje, vadovaudamiesi taisykle, kad žodžio reikšmė yra labai susijusi su kontekstu, kuriame ta reikšmė pasirodo [2], visas kontekstas traktuojamas susijusių sąvokų vektorių aibe su susietumo statistiniais įverčiais [13]. Iš vektorių aibės atrenkami tie, kurie sujungia tam tikrą reikšmingiausių nagrinėjamai sričiai sąvokų skaičių (žr. 2.3 skyrelį).

Šis procesas vainikuoja visą darbą ir gaunamas viso proceso darbo rezultatas – turima srities žinių ontologija.

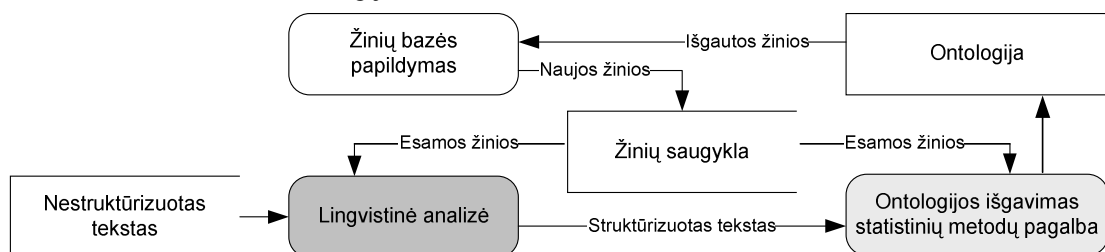
2.6 Proceso sunkumai

Klasikinis ontologijos išskyrimo procesas remiasi statistiniais metodais, kurių patikimiems rezultatams gauti imties dydis turi būti pakankamai didelis. Teksto apimtis privalo būti tokia, kad žodžiai pasikartotų bent po keletą kartų, nes tik iš pasikartojančios informacijos galima spręsti apie žodžio ryšį, jo vertę nagrinėjamame tekste, o kartu ir vertę nagrinėjamai sričiai.

Nuodugni lingvistinė analizė praktiškai eliminuoja žmogaus įsikišimą į ontologijos išgavimo procesą: žodžių lygyje sužymimi daiktavardžiai – galimos ontologijos klasės, išskirtų reikšmių grupavimo žingsnis gali apjungti sąvokas pagal jų bendrines formas, frazės bei sakinių lygio analizė įgalina tiksliau apibrėžti ontologijos ryšius, automatiškai nustatant tripletų (<subject> <predicate> <object>) sandarą [1, 5, 6]. Tačiau atlikti tokią analizę yra pakankamai sudėtinga, o kai kurioms kalboms – beveik neįmanoma. Plačiai naudojamos kalbos, tokios kaip anglų, rusų, vokiečių ar prancūzų turi išvystytus natūralios kalbos apdorojimo (angl. NLP – natural language processing) įrankius, tačiau ir jie vis dar turi trūkumų.

3. Įrankio architektūra

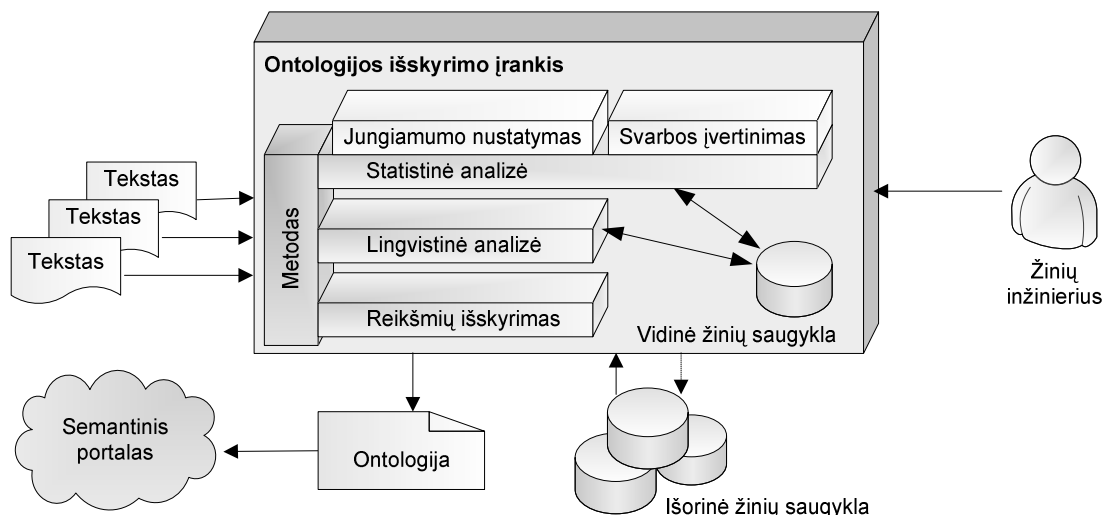
Dažnai darbo supaprastinimui bei rezultato kokybės pagerinimui ontologijos išskyrimo procese naudojami reikšmių vidiniai arba išoriniai (viešieji) žodynai – žinių saugyklos (58 pav.). Priklausomai nuo siekiamo tikslo, tokia žinių saugykla gali būti tiesiog dažniausiai pasitaikančių žodžių sąrašas su pasirodymo dažniniu įverčiu arba bendrinė ontologija [1, 5, 2].



58 pav. Žinių saugyklos panaudojimas ontologijos išskyrimo procese

Tokią žinių saugyklą naudoja ir lingvistinė analizė, ir statistiniai metodai, tačiau šiems blokams iš saugyklos reikalinga skirtinga informacija. Lingvistinės analizės blokas iš tokios saugyklos gauna reikšmių bendrines formas, gali patikrinti kalbos dalį, gauti reikšmės sinonimus ir panašią informaciją, padedančią lengviau sužymėti tekstą. Statistiniai metodai palygina išskirtų srities reikšmių pasikartojimo dažnius su tų pačių reikšmių bendrinėje saugykloje pasikartojimo dažniais. Toks palyginimas leidžia išskirti itin specifines srities reikšmes: jei srities tekstuose dažnai pasikartoja žodis retai pasitaikanti bendriniuose tekstuose, reiškia, kad tai yra specifinis nagrinėjamos srities žodis. Tokį žodį tikslinga įtraukti į kuriamos ontologijos klasę.

Ontologijos išskyrimo įrankio, realizuojančio aukščiau aprašytą klasikinių ontologijos išskyrimo metodą praplėstą lingvistine analize bei naudojančią žinių saugyklą, supaprastinta architektūrinė schema bei kontekstas pavaizduoti 59 pav.



59 pav. Ontologijos išskyrimo įrankio architektūra bei kontekstas

Panaudojus žinių saugyklos žinias išskiriamos naujos žinios, kurių gali nebūti saugykloje, todėl tikslinga papildyti esamas žinias naujomis žinių bazės praplėtimui. Taip sudaromos didžiulės bendrinės ontologijos-žodynai. Tokio žodyno pavyzdys – anglų kalbos žinių bazė WordNet – plačiausia bendrinė ontologija-žodynas [5, 2]. Vidinės saugyklos taip pat gali būti naudojamos vartotojo (žinių inžinieriaus) pasirinkimams saugoti. Įvykus situacijai, kurioje sistema automatiškai negalės priimti sprendimo ir reikės vartotojo įsikišimo, įrankis išsaugos vartotojo pasirinkimą. Kitą kartą susidūrus su analogiška arba labai panašia situacija, sistema galės įveikti ją neužklausius vartotojo. Suprantama, kad ne visada toks požiūris pasiteisina, nes dažnai net ir labai panašios situacijos iš tikrųjų gali būti labai išskirtinės, tačiau, jei sistemoje bus išsaugota daugiau tokių pasirinkimų, jas apibendrinus gali būti padarytos tam tikros išvados.

Toks įrankio suskirstymas į atskirus, tam tikrus rezultatus atliekančius blokus, turi savo privalumą: kiekvienas jų (lingvistinė analizė, reikšmių išskyrimas, svarbos įvertinimas, jungiamumo nustatymas) gali būti realizuoti nuotolinėse sistemose kaip atitinkamas paslaugas suteikiantis webservisas. Toks suskirstymas galimas dėl to, kad apibrėžta kiekvieno bloko įeiga bei išeiga.

4. Išvados

Straipsnyje apžvelgtos automatinio ontologijų išgavimo proceso problemos, pateiktas jo skaidymas į etapus – lingvistinio, grupavimo ir kt. Kiekviename etape iškyla daug svarbių specifinių problemų, kurias reikia išspręsti automatiškai interpretuojant tekstą.

Ontologijos sudarymui statistiškai įvertinamos iš teksto išskirtos reikšmės nustatant jų svarbą nagrinėjamai sričiai. Reikšmių išskyrimas be lingvistinių teksto žinių reikalautų žinių eksperto pagalbos – taip pat ir reikšmių grupavimo bei ryšių tarp grupių nustatymo proceso žingsniuose. Lingvistinės analizės etape tekstas yra praturtinamas lingvistinėmis žiniomis, kurios suteikia informacijos apie sakinio struktūrą bei žodžio formas. Visi proceso žingsniai einantys po lingvistinės analizės, išskyrus reikšmių išskyrimą, remiasi statistiniais metodais.

Straipsnyje pasiūloma architektūra, leidžianti išskaidyti uždavinių sprendimą atskiroms paprogramėms bei įvertinanti išorinių saugyklų ir eksperimentų metu sukauptą informaciją. Sudalinimas į nepriklausomas paprogrames leidžia įrankio architektūrinius blokus suskaidyti paskirstytos sistemos mazgams.

Naudojama literatūra

31. Buitelaar P., Olejnik D., Sintek M. A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis. Germany, Saarbruecken, 2004.
32. Gómez-Pérez A., Manzano-Macho D. A survey of ontology learning methods and techniques. Universidad Politécnica de Madrid, 2003.
33. Kawata Y. Tagsets for Morphosyntactic Corpus Annotation: the idea of a 'reference tagset' for Japanese. Essex, 2005.
34. Lin C.-Y., Hovy E. The Automated Acquisition of Topic Signatures for Text Summarization. Information Science Institute, University of Southern California, USA, 2002.
35. Maedche A., Volz R. The Ontology Extraction & Maintenance Framework: Text-To-Onto. Forschungszentrum Informatik FZI, Karlsruhe, 2001.
36. Navarro B., Civit M., Martí M., Marcos R., Fernández B. Syntactic, semantic and pragmatic annotation in Cast3LB. Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante, Spain, 2003.
37. OntoGroup – Kauno technologijos universiteto Informacijos sistemų katedros darbo grupės wiki. <http://isk.ktu.lt:8081/wiki/>, 2005-2006.
38. O'Shaughnessy D. D. Parsing with a Small Dictionary for Applications such as Text To Speech. INRS-Telecommunications, 3 Place du Commerce, Nuns' Island, Quebec H3E 1H6 Canada, 2002.
39. Pedersen T., Patwardhan S., Michelizzi J. WordNet::Similarity - Measuring the Relatedness of Concepts. Department of Computer Science University of Minnesota, Duluth, 2004.
40. Rimkutė E. Lietuvių kalbos tekstyno morfologinės analizės automatizavimas. 6-toji magistrantų ir doktorantų konferencija Informacinė visuomenė ir universitetinės studijos. Kaunas, 2001: pp. 60-66.
41. Seki K., Mostafa J. An Application of Text Categorization Methods to Gene Ontology Annotation. Laboratory of Applied Informatics Research, Indiana University, Bloomington, 2005.
42. W3C (World Wide Web Consortium) rekomendacijos, <http://www.w3.org/>, 2004
43. Wikipedia, The Free Encyclopedia, "Ontology (computer science)", <http://www.wikipedia.org/>, 2006.

ARCHITECTURE OF ONTOLOGY EXTRACTION TOOL

Ontology extraction method is introduced in the publication. Possible steps of such method are introduced and the list of problems for each step is build. The architecture of ontology extraction tool and its components are introduced according to this review. Decentralization is taken into account when designing the architecture.