

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Informacijos sistemų katedra

Evaldas Mureika

**Duomenų, surinktų iš HTML failų, publikavimo RDF
formate metodika**

Magistro darbas

Darbo vadovas: Doc. R. Butkienė

KAUNAS, 2012

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Informacijos sistemų katedra

Evaldas Mureika

**Duomenų, surinktų iš HTML failų, publikavimo RDF
formate metodika**

Magistro darbas

Recenzentas
dr. lekt. A.Janavičiūtė
2012-05-

Vadovas
Doc. R. Butkienė
2012-05-

Atliko
IFM-0/4 gr. stud.
Evaldas Mureika
2012-05-24

Kaunas, 2012

Turinys

TURINYS	3
1. TERMINŲ IR SANTRUMPŲ ŽODYNAS	5
2. ĮVADAS	6
3. ANALIZĖ	8
3.1. ANALIZĖS TIKSLAS.....	8
3.2. ANALIZĖS METODAI.....	8
3.3. VARTOTOJŲ ANALIZĖ.....	8
3.3.1. Vartotojų aibė, tipai ir savybės.....	8
3.3.2. Vartotojų tikslai ir problemos.....	9
3.4. TYRIMO OBJEKTO ANALIZĖ.....	10
3.4.1. HTML rinkmenos.....	10
3.4.1. Semantinio tinklo apibrėžimas.....	10
3.4.2. Semantinio tinklo technologijos bei architektūra.....	11
3.4.3. Semantiniotinklo saugojimas ir paieška.....	11
3.5. SEMANTINIO TINKLO ARCHITEKTŪROS APŽVALGA.....	11
3.5.1. URI/IRI.....	11
3.5.1. XML.....	12
3.5.1. RDF resursų aprašymo formatas.....	12
3.5.1. RDF-S žymų kalba.....	12
3.5.2. OWL.....	13
3.5.1. RIF – taisyklių apsikeitimo formatas.....	13
3.5.1. Apjungianti logika.....	14
3.5.1. Validavimas.....	14
3.5.2. Lygiagretinimas su patikrintais duomenimis.....	14
3.6. ŽODYNAI.....	15
3.6.1. SIOC.....	15
3.6.2. The Friend of a Friend (FOAF) projektas.....	15
3.6.3. vCards žodynas.....	17
3.7. RELIACINĖS DUOMENŲ BAZĖS IR SEMANTIKA.....	17
3.8. SEMANTINIO TINKLO SAUGOJIMAS IR PAIEŠKA.....	18
3.8.1. SPARQL - užklausų kalba.....	18
3.8.2. Joseki - HTTPvariklis SPARQL protokolui palaikyti.....	19
3.8.3. Jena.....	20
3.8.4. Sesame.....	21
3.9. ESAMI PUBLIKAVIMO ĮRANKIAI BEI RDF DUOMENŲ SAUGYKLOS.....	21
3.9.1. Oracle Database 11g Semantic Technologies.....	21
3.9.2. Openlink Virtuoso.....	22
3.9.3. D2RQ ir D2R Server.....	23
3.9.4. SquirrelRDF.....	25
3.9.5. METAmorphoses.....	28
3.10. ĮVAIRIŲ SEMANTINIO TINKLO KŪRIMO ĮRANKIŲ PALYGINIMAS.....	32
3.10.1. Palyginimo kriterijai.....	32
3.10.2. Spartos palyginimas.....	34
3.11. ONTOLOGIJŲ KŪRIMO ĮRANKIAI.....	36
3.11.1. TopBraidComposer.....	36
3.11.2. Prot'eg'é.....	37
3.12. TAIKYMO PAVYZDŽIAI.....	37
3.12.1. DBpedia.....	38
3.12.2. Watson.....	39
3.12.3. Muzikos duomenų publikavimas semantiniame tinkle.....	40
3.12.4. Knygų aprašymų pateikimas semantiniame tinkle RDF kalba.....	41
3.13. RELIACINIŲ DUOMENŲ BAZIŲ PUBLIKAVIMAS SEMANTINIAME TINKLE.....	41
3.14. ANALIZĖS IŠVADOS.....	43
4. SISTEMOS REIKALAVIMAI	45
4.1. REIKALAVIMŲ SPECIFIKACIJA.....	45
4.1.1. Funkciniai reikalavimai:.....	46
4.1.2. Panaudojimo atvejų aprašas.....	47

4.1.	PRINCIPINĖ DUOMENŲ BAZĖS UŽPILDYMO DUOMENIMIS BEI JOS TURINIO PUBLIKAVIMO SCHEMA.....	53
4.2.	DALYKINĖS SRITIES ESYBIŲ MODELIS	54
4.3.	NEFUNKCINIAI REIKALAVIMAI	56
5.	SISTEMOS PROJEKTAS	57
5.1.	SISTEMOS ARCHITEKTŪRA	57
5.2.	PROJEKTAVIMO ETAPO KLASĖS	58
5.3.	PANAUDOJIMO ATVEJUS REALIZUOJANČIOS KLASĖS	60
5.4.	PANAUDOJIMO ATVEJUS DETALIZUOJANČIOS SEKŲ DIAGRAMOS.....	64
5.5.	SISTEMOS REALIZACIJOS MODELIS	67
6.	SPRENDIMO REALIZACIJA	69
6.1.	RELIACINĖ DUOMENŲ BAZĖ	69
6.2.	HTML RINKMENŲ NUSKAITYMAS	69
6.3.	ONTOLOGIJOS GENERATORIUS	70
6.4.	DUOMENŲ PUBLIKAVIMAS	72
6.5.	GRAFINĖS VARTOTO SAŠAJOS MODELIS:	73
6.5.1.	<i>GUI puslapis: Pasirinkimo meniu:</i>	73
6.5.2.	<i>GUI puslapis: Degalų duomenų registravimo programa:</i>	73
6.5.3.	<i>GUI puslapis: Informacija semantiniame tinkle:</i>	74
6.5.4.	<i>GUI puslapis: Statistika:</i>	75
7.	EKSPERIMENTINIS SISTEMOS TYRIMAS.....	76
7.1.	EKSPERIMENTO PLANAS.....	76
7.2.	EKSPERIMENTO REZULTATAI	77
7.2.1.	<i>HTML rinkmenos.....</i>	77
7.2.2.	<i>Duomenų bazė.....</i>	79
7.2.3.	<i>Ontologija.....</i>	81
7.2.4.	<i>Ontologijos pervadinimo eksperimentinis tikrinimas.....</i>	83
7.2.5.	<i>Metamorphoses žymėjimo rinkmena</i>	84
7.2.6.	<i>Metamorphoses šablono rinkmena.....</i>	85
7.2.7.	<i>RDF Tikrinimas.....</i>	85
7.2.8.	<i>Duomenų apdorojimo sparta.....</i>	89
7.3.	SISTEMOS VEIKIMO IR SAVYBIŲ ANALIZĖ, KOKYBĖS KRITERIJŲ ĮVERTINIMAS.....	90
7.4.	SISTEMOS TAIKYMO REKOMENDACIJOS	91
8.	IŠVADOS	92
9.	SUMMARY	93
10.	LITERATŪRA.....	96

1. Terminų ir santrumpų žodynas

Terminas ar santrumpa	Paaškinimas
RDF	Semantinio tinklo infrastruktūrai aprašyti yra naudojama Resursų Aprašymo Struktūra (angl. <i>Resource Description Framework</i> arba RDF).
HTML	(angl. <i>Hyper text Markup Language</i>) Hiperteksto žymėjimo kalba – tai kompiuterinė žymėjimo kalba, naudojama pateikti turinį internete. Kalbą standartizuoja W3 konsorciumas.
DB	Duomenų bazė
XML	XML (angl. <i>Extensible Markup Language</i>) yra W3C rekomenduojama bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba.
XML S	angl.Extensible Markup Language (XML) Tai yra rinkinys taisyklių, kuriomis galima aprašyti dokumentus kompiuteriui suprantama forma
RDF S	Tai kalba suteikianti bazinius elementus skirtus aprašyti ontologijas, kurios dar yra vadinamos RDF žodynais
OWL	Tai grupė kalbų skirtų aprašyti žinioms.
SPARQL	Tai užklausų kalba skirta užklausti RDF duomenis
SIOC	(angl. <i>Semantically-Interlinked Online Communities</i>), tai žodynas nusakantis semantiškai susijusiais žiniatinklio bendruomenes.
RDF-S	Laisvu formatu aprašomas žodynas
FOAF	Žodynas skirtas aprašyti žmones, sąsajas tarp žmonių ir dalykus, kurios žmonės sukuria ar veikia
vCards	Žodynas skirtas aprašyti elektroninėse knygose, švieslentėse ar kitose taikomuosiose programose kaupiamą informaciją.
N3	
RDBMS SQL	(angl. <i>A relational database management system</i>) Reliacinės duomenų bazės valdymo sistema
D2RQ ir D2R	Server Tai yra duomenų bazių ir RDF susiejimo įrankis
GUI	Grafinė vartotojo sąsaja

2. Įvadas

Semantinis tinklas – tai duomenų tinklas, leidžiantis apjungti informaciją esančią įvairiuose šaltiniuose ir pavaizduoti visą surinktą informaciją kaip vientisą turinį. Semantiniame tinkle duomenys susieti ir prieinami per bendrąją tinko architektūrą, panaudojant URI adresus. Didžioji dalis duomenų yra neapjungti ir prienami atskiroms taikomosioms programoms. Apjungus šiuos įvairiuose šaltiniuose laikomus duomenis būtų sukurtas semantinis duomenų tinklas.

Duomenims panaudoti, aprašyti bei susieti kuriami įvairūs karkasai, leidžiantys pakartotinai dalintis duomenimis visose taikymo srityse, įmonėse bei bendrijose. Struktūrizuota semantinė informacija naudojama automatiškai arba rankiniu būdu ir apdorojama kompiuterių. Semantinio tinklo metodai ir technologijos leidžia kompiuteriams suprasti pasauliniame tinkle (angl. *World Wide Web*) laikomos informacijos reikšmę arba kitaip semantiką.

Semantinio tinklo technologijos panaudojamos:

- 1) duomenų integracijai, duomenų laikomų skirtingose vietose ir saugomų skirtingais formatais susiejimui;
- 2) resursų aptikimui ir klasifikacijai, sukuriant geresnį informacijos paieškos variklį;
- 3) tinklapių, svetainių ar skaitmeninių bibliotekų katalogavimui aprašant turinį, turinio ryšius;
- 4) sumaniųjų programinės įrangos agentų žinių mainams palengvinti;
- 5) turinio reitingavimui;
- 6) aprašant intelektualinės nuosavybės teises (tinklapiuose);

Semantiniotinklų taikymas bei technologijos technologijos yra dar labai naujos, neištirtos, neišbandytos praktiškai ir joms trūksta metodinės medžiagos. Tiriamajame darbe išanalizuota kiek šios technologijos yra pažengusios, išanalizuoti semantinių tinklų kūrimo įrankiai bei naudojamos technologijos. Įrankių bei technologijų taikymo galimybės pademonstruoti sukurta taikomoji programa, galinti automatiškai atrinkti informaciją ir ją semantiškai praturtinti.

Demonstracinė taikomoji programa surenka informaciją iš tinklapių kopijų, kurios saugomos HTML formatu. Nuskaityta informacija talpinama į reliacinę duomenų bazę. Sukaupti duomenys panaudojami OWL ontologijos generavimui, bei duomenų išpublikavimui semantiniame tinkle.

Uždaviniai:

- 1) Išanalizuoti kaip gali būti išpublikuota HTML formate saugomi duomenys;
- 2) Išanalizuoti semantinio tinklo kūrimo technologijas;
- 3) Sukurti taikomąją programą automatiškai nuskaitančią duomenis iš HTML rinkmenų;
- 4) Sukurti taikomąją programą kuri sugeneruotų pirminį OWL ontologijos variantą iš reliacinės duomenų bazės;
- 5) Pritaikyti semantinį duomenų publikavimąduomenims;
- 6) Eksperimentiškai ištirti sukurtą sistemą;

3. Analizė

Semantiniams tinklams plėtoti yra naudojama labai daug ir įvairių įrankių bei technologijų, taigi yra būtina iširti jau sukurtus semantinio tinklo sprendimus ir išnagrinėti technologijas, kurios gali būti taikomos

Taip pat bus aptariami semantinio tinklo vartotojai aptartas tyrimo objektas - semantinis tinklas - semantinio tinklo architektūra, HTML rinkmenose saugoma informacija, bei naudojamos technologijos

3.1. Analizės tikslas

Analizėje siekiama:

- Išsiaiškinti tyrimo objektą. Kokios problemos kyla diegiant semantinio tinklo sistemas.
- Išanalizuoti esamus sprendimus, semantinės informacijos publikavimo technologijas;

3.2. Analizės metodai

Tyrimui atlikti ir semantinių tinklų galimybes įvertinti bus analizuojami įvairūs šaltiniai. Šaltinius siekiama suklasifikuoti ir priskirti tam tikroms sritims, kurios apibūdina nagrinėjamą objektą. Vėliau pagal atliktos literatūros analizę, bus lyginamos skirtingų objektų savybės ir siekiama įvertinti galimus pasirinkimus ir pasirinkti sprendimus gilesniam semantinių tinklų galimybių tyrimui atlikti. Pasirinktus sprendimus bus siekiama praktiškai pritaikyti.

3.3. Vartotojų analizė

3.3.1. Vartotojų aibė, tipai ir savybės

Ateityje semantinis tinklas taps prieinamas visiems pasaulio vartotojams ir taps kiekvieno žmogaus kasdienybe.

Semantinių tinklų sistemų vartotojus galima suskirstyti į 2 grupes:

- Paprastus vartotojus, kurie naudosis semantinių tinklų galimybėmis paieškos sistemose, tokiose kaip Watson, muzikiniuose tinklapiuose ar kitose sistemose kuriuose visa informacija bus susieta.
- Semantinių tinklų kūrėjus, kurie taikys sukurtas technologijas ir susies jau sukurtas sistemas, kurios naudoja semantinių tinklų technologijas.

3.3.2. Vartotojų tikslai ir problemos

Lentelė 1 Vartotojų tikslai ir problemos

Semantinių tinklų kūrėjai	
Problema	Kaip viskas vyksta dabar?
Nėra paprasto sprendimo nuo duomenų surinkimo iki išpublikavimo semantiniame tinkle.	Skirtingiems uždaviniams ieškoma vis kitokių architektūrinių bei technologinių sprendimų.
Nėra prieigos prie duomenų bazių, kuriose saugoma įvairi informacija	Reikalingi duomenys surenkami apdorojant informaciją prieinamą per URL adresus.
Paprastus vartotojus	
Informacija yra saugoma įvairiuose duomenų šaltiniuose ir nesusiteminta.	Vartotojui sunku surasti informacija, kadangi šios informacijos prieiga yra necentralizuota, informacija išmėtyta po skirtingus šaltinius.

Naudojant semantinių tinklų technologijas galima integruoti skirtingas funkcijas atliekančias sistemas, susieti susijusią informaciją. Semantinių tinklų technologijos suteikia galimybę mašinai suprasti užklauso semantiką, todėl paieškos sistemose ir kitose programose galima naudoti semantines užklauso.

3.4. Tyrimo objekto analizė

3.4.1. HTML rinkmenos

HTML formatu aprašomi svetainių puslapiai. Informacijai pasiekti sudaromos *regex* užklausos, leidžiančios rasti tikslią informacijos vietą dokumente. Apdorojus informacija kaupiama reliacinėje duomenų bazėje bei pateikiama semantiniame tinkle.

HTML tinklapio fragmentas:

```
<table cellpadding="0" cellspacing="0">
  <tr>
    <td><strong>Adresas:</strong></td>
    <td>Šilutės pl. 96, Klaipėda</td>
  </tr>
  <tr>
    <td><strong>Telefonas:</strong></td>
    <td>+370 46 30 14 96</td>
  </tr>
  <tr>
    <td><strong>El. paštas:</strong></td>
    <td><strong>mail@gmail.com</strong></td>
  </tr>
</table>
```

HTML tinklapio fragmente pateikiama HTML kalba aprašyta lentelė. Lentelėje nurodyta informacija adresas, telefonas, elektroninis paštas yra prasminga ir todėl gali būti atrenkama ir naudojama.

3.4.1. Semantinio tinklo apibrėžimas

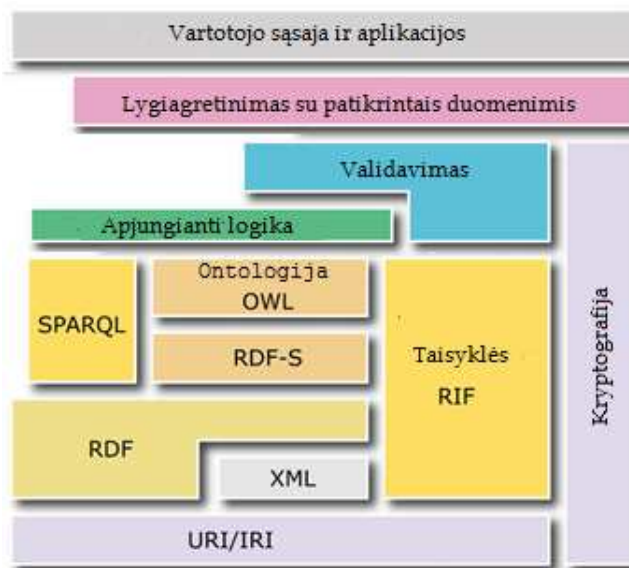
Tim Berners-Lee, pasaulinio tinklo (angl. *World Wide Web*) išradėjas, nusako Semantinį tinklą kaip duomenų tinklą, kuris gali būti tiesiogiai ir netiesiogiai apdorojamas kompiuterių. Semantiniame tinkle duomenys apdorojami, transformuojami, surenkami, panaudojami kompiuterių.

3.4.2.

Semantinio tinklo technologijos bei architektūra

Semantinis tinklas sudarytas iš įvairių standartų ir įrankių XML, XML S, RDF, RDF S ir OWL, kurie yra susieti semantinio tinklo sistemos architektūroje.

1. XML kalba suteikia sintaksę turinio struktūrai semantinio tinklo dokumentuose aprašyti, tačiau nesusieja semantinių duomenų. XML S yra kalba, kuri suteikia ir atskiria XML dokumentų struktūrą bei praplečia ją duomenų tipais.



Pav. 1 Semantinio tinklo architektūra, šaltinis [29]

2. RDF yra paprasta kalba išreiškianti duomenų modelius, kurie nurodo į objektus (resursus) ir jų ryšius. RDF kalba paremtas modelis gali būti aprašytas XML sintaksėje
3. RDF S praplečia RDF ir kartu tai yra žodynas skirtas aprašyti RDF kalba aprašytas klases ir ypatybes.
4. OWL kalba prideda platesnį žodyną, kuris gali plačiau aprašyti klases ir detales.
5. SPARQL tai yra semantinio tinklo ir informacijos protokolas ir užklausų kalba.

3.4.3. Semantiniotinklo saugojimas ir paieška

Semantinio tinklo informacijai kaupti yra naudojam ontologijos OWL rinkmenos, RDF formatu aprašomi informacijos resursai ir kiti duomenų saugojimo formatai. Minėtais formatais išreikšti duomenys yra talpinami į duomenų saugyklas, o patalpinta informacija gali būti užklausiama panaudojant SPARQL užklausas. Naudodami SPARQL užklausas galime ieškoti informacijos skirtinguose šaltiniuose bei nepriklausomai nuo duomenų kaupimo vietas.

3.5. Semantinio tinklo architektūros apžvalga

3.5.1. URI/IRI

URI/IRI - tai nuorodos į duomenų šaltinius, kur pasauliniame tinkle yra saugoma informacija. Naudodami šiuos duomenų adresus galime susieti visą pasauliniame tinkle laikomą informaciją. Šis lygmuo yra svarbus semantiniame tinkle, kadangi visa informacija semantiniam tinklui jau yra, reikia tik rasti būdus kaip šią informaciją apjungti.

3.5.1. XML

Extensible Markup Language arba XML aprašo duomenų objektų, XML dokumentų, klases ir dalinai aprašo kompiuterio programų, apdorojančių XML duomenis, elgesį. XML dokumentai sudaryti iš subjektų, kuriuose saugoma informacija. Informacija yra sudaroma iš ženklų nusakančių duomenis bei XML dokumento žymėjimą. Dokumento žymėjimas nusako duomenų išdėstymą bei loginę struktūrą. XML kalba suteikia duomenų išdėstymo bei loginės struktūros apribojimo mechanizmą. Programinės įrangos modulis vadinamas XML procesoriumi naudojamas XML dokumentų nuskaitymui ir apdorojimui.

3.5.1. RDF resursų aprašymo formatas

Semantinio tinklo infrastruktūrai aprašyti yra naudojama Resursų Aprašymo Struktūra (angl. *Resource Description Framework* arba *RDF*). RDF resursų aprašymo struktūra suteikia paprastą duomenų aprašymo modelį, kuris yra skirtas aprašyti semantinio tinklo duomenis. Duomenys aprašomi naudojant trilypes struktūras, kurios susideda iš subjekto, teiginio ir reikšmės aprašymo.

Trilypės struktūros subjektas ir reikšmė gali būti nusakoma tame pačiame dokumente arba gali nurodyti į kitą dokumento šaltinį esantį internete. Teiginiu gali būti bet koks XML kalboje nusakomas vardas. Naudodami šią technologiją galime aprašyti ir duomenis, kurių šaltinis yra kiti RDF duomenys. RDF modelyje naudojamas įvairių išraiškų rinkinys vartoja konkretų žodyną, kuris nusako ypatybes ir duomenų tipus. Šis žodynas gali būti panaudotas taikomojoje programoje.

3.5.1. RDF-S žymų kalba

Žodynas gali būti aprašomas RDF Schema. RDF Schema kalboje, kaip dalis RDF Schema kalbos yra aprašomi iš anksto nusakyti koncepcijas nusakančias primityvus skirtus klasių / sub klasių, hierarchijos, santykių (ypatybių) tarp klasių, srities arba grupės apribojimui nusakyti. RDF modelis aprašo tik rinkiny trilypių struktūrų, o RDF-s kalba

pateikia pakankamai trilypių struktūrų skirtų nusakyti klasių hierarchiją, kurios leidžia užklausti ir argumentuoti.

3.5.2. OWL

OWL tinklo ontologijos kalba sukurta naudoti aplikacijoms, kurioms reikia apdoroti įvairią informaciją. Naudodami OWL kalbą galime praplėsti semantinį žodyną. Yra 3 OWL kalbos atmainos:

1. OWL Lite – skirtams vartotojams norintiems klasifikuoti hierarchiją bei supaprastinti apribojimus. Tai yra paprasčiausia OWL ontologijos kalbos atmaina.
2. OWL DL – skirta vartotojams norintiems gauti maksimalią ekspresiją išlaikant skaičiavimo išbaigtumą ir išsprendžiamumą. OWL DL kalboje naudojami visi OWL kalbos konstruktai, tačiau šių konstrukčių naudojimas yra ribojamas.
3. OWL Full – skirta vartotojams norintiems gauti maksimalią ekspresiją ir RDF sintaksės lasivę nesuteikdama jokių skaičiavimų garantijų.

Semantinis tinklas – tai ateities tinklo vizija, kurioje palengvinamas automatinis pasaulinio tinklo informacijos apdorojimas ir integravimas kompiuteriais suteikiant duomenims tikslią reikšmę. Šis tinklas kuriamas panaudojant XML dokumentų savybę, leidžiančia aprašyti individualias žymėjimo schemas, bei RDF dokumentų lankstumą atvaizduojant duomenis.

OWL Ontologija leidžia aprašyti tinklo dokumentuose naudojamų terminų prasmę praplečia duomenų aprašymo žodyną aprašant klases, ypatybes, ryšius tarp klasių, kardinalumus, lygybes, ypatybių charakteristikas.

3.5.1. RIF – taisyklių apsikeitimo formatas

Taisyklių apsikeitimo formatas arba RIF (angl. *Rule Interchange Format*) yra semantinio tinklo infrastruktūros dalis, dažniausiai naudojama kartu su SPARQL, RDF ir OWL. RIF lygmuo yra naudojamas taisyklių apsikeitimui, kadangi yra labai daug semantinio tinklo taisyklėms aprašyti naudojamų kalbų.

Tokios taisyklės pavyzdys galėtų būti JEI – TAI konstruktas. Galime sudaryti taisyklę įvertinti įvairioms savybėms, tarkim santuokai. Tarkim

SUSITUOKĖ(MARIUS,JOANA).SUSITUOKĖ

tai predikatas nusakantis jog šie 2 žmonės yra susituokę. Įrašę kintamuosius galime suformuluoti štai tokią taisyklę:

JEI *SUSITUOKE*(?x, ?y) TAI *MYLI*(?x, ?y), kur (?x, ?y) yra MARIUS ir JOANA.

Taisyklės - tai parpasčiausias būdas žinioms užkoduoti, diegti į informaciją apdorojančias sistemas ir apdorojus gauti teisingas išvadas.

3.5.1. Apjungianti logika

Šiame lygmenyje yra naudojama FOL (angl. *First Order Logic*) pirmojo lygmens logika. Dėl FOL palaikymo semantiniame tinkle yra pakankamai palaikoma logikos galimybių ir kartu išlaikoma pagrįsta skaičiavimo resursų kaina.

3.5.1. Validavimas

Šiame lygmenyje diegiami semantinio tinklo logikos išvadas formuojantys varikliai, sudaromi išvadų patvirtinimai arba paneigimai. Taip pat diegiami begalinių logikos užklausų supaprastinimo ir išsprendimo mechanizmai, kur įmanoma pateikiamos jau validuotos išvados, kurios gali būti lengvai patikrintos. Patikrintos išvados yra kaupiamos ir vėliau gali būti panaudojamos naujų, neišspręstų logikos užduočių validavimui.

3.5.2. Lygiagretinimas su patikrintais duomenimis

Šiame lygmenyje yra panaudojami visi kiti lygmenys, naudojami skaitmeninio parašo, kodavimo mechanizmai. Semantiniame tinkle yra intensyviai naudojamos viešojo rakto struktūros, jau naudojamos kaip skaitmeniniai sertifikatai indentifikuojantys šalis, pasirašančias kontraktus.

3.6. Žodynai.

Duomenų prasmei aprašyti kuriami žodynai. Norint bet kokius duomenis praturtinti semantiškai pirmiausiai reikia turėti betkokios srities ontologiją. Tam tikroms sritims jau egzistuoja ontologijos ir jos gali būti naudojamos kaip žodynai aprašyti tam tikrą duomenų semantiką. Tačiau ne visuomet šių žodynų pakanka, kuomet norima aprašyti semantinio tinklo duomenis, ir yra reikalinga susidaryti nagrinėjamą dalykinę sritį apibūdinantį žodyną.

Be ontologijų yra sukurti tam tikrų sričių visuotinai pripažinti žodynai ir gali būti panaudoti semantikai aprašyti. SIOC, FOAF, vCards – tai tik keletas jau naudojamų žodynų pavyzdžių.

3.6.1. SIOC

SIOC (angl. *Semantically-Interlinked Online Communities*), tai žodynas nusakantis semantiškai susijusiais žiniatinklio bendruomenes. Pagrindinė ontologija suteikia konceptus ir ypatybes, kurie yra būtini žiniatinklio bendruomenių (įvairios skelbimų lentos, blogai, žinynai ir kiti) turiniu semantiniame tinkle aprašyti.

Įvairiuose bloguose žinyuose ar skelbimų lentose yra pateikiama daug naudingos informacijos. Skirtinguose žiniatinklio bendruomenėse pateikiama įvairi informacija, kuri pateikiama skirtinguose šaltiniuose gali papildyti viena kita. Naudojant SIOC žodyną bei kitas semantinių tinklų technologijas galima susieti šiose bendruomeninėse organizacijose pateikiamą informaciją ir aprašyti informacijos struktūrą, turinį bei sąsajas. SIOC žodynas sukurtas remiantis informacija, kuri yra pateikiama žiniatinklio bendruomenėse pateikiama mašinai suprantamu formatu.

3.6.2. The Friend of a Friend (FOAF) projektas

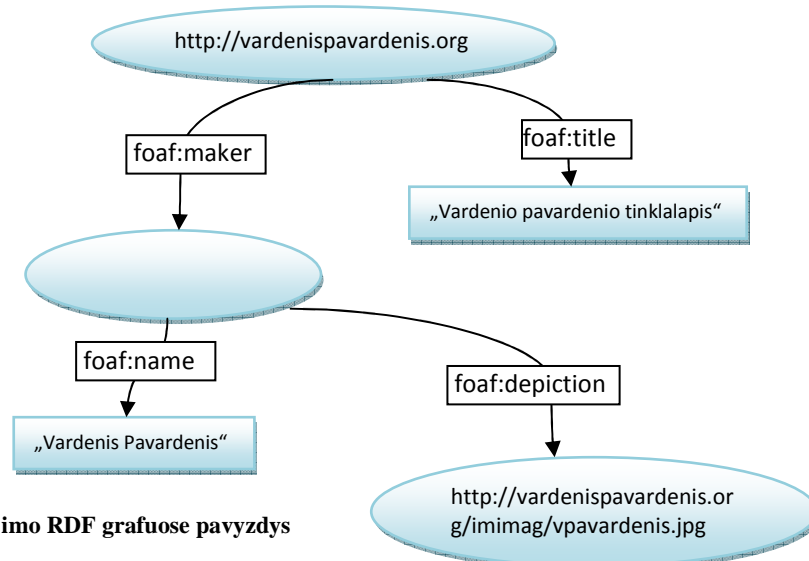
Friend of a Friend (FOAF) žodynas buvo sukurtas, kuomet buvo vykdomas projektas, kurio tikslas buvo sukurti tinklą, kuris būtų suprantamas mašinoms ir kartu nusakytu žmones, sąsajas tarp žmonių ir nusakytu dalykus, kurios žmonės sukuria ar veikia.

Šis projektas susieja žmones ir informaciją, kuri yra saugoma žmonių galvose, fiziniuose ar skaitmeniniuose dokumentuose, ar faktinių duomenų formoje. FOAF integruoja trijų rūšių tinklus: socialinius, žmonių bendradarbiavimo, draugystės ir asociacijų.

Jei žmonės publikuos informaciją naudodami FOAF dokumentų formatą, tuomet kompiuterija galės suprasti ir panaudoti tuose dokumentuose saugomą informaciją. Jei tie dokumentai yra sudaryti iš nuorodų „Taip pat pažiūrėkite“ į kitus tokios pačios paskirties dokumentus, kompiuterines programas galės pačios surasti ir kaupti informaciją ir atlikti įvairias operacijas su duomenimis.

FOAF dokumentai tai yra tekstiniai dokumentai, kurie perima RDF modeliavimo kalbos formalias išraiškas. Šie dokumentai gali būti užrašomi XML ar kokia kita RDF modelio sintakse, tokia kaip RDF ar N3. FOAF tai yra žodynas, kuris nusako žmonių ryšius, veiklas ar kitus dalykus. Šis žodynas apibrėžia keletą naudingų konstrukčių, kurie naudojami FOAF bylose kartu su RDF modeliu apibrėžiamais žodynais. FOAF žodynas nusako klases (foaf:Person, foaf:Document, foaf:Image, kartu nusakydamas ypatybes foaf:name, foaf:mbox, foaf:homepage etc.,) ir kartu nusako keletą naudingų informacijos rūšių, kurios susieja narius. Pavyzdžiui FOAF žodyne naudojamas santykiu tipas foaf:depiction susieja žmones (foaf:Person) su nuotraukom (foaf:Image). FOAF žodynas leidžia susieti nuotraukas ir žmones ir nusakyti kas yra kuriame paveikslėlyje. Visa tai priklauso tik nuo to, kokia programinė įranga ir įrankiai yra parenkami nagrinėti RDF dokumentams ir dokumentų ypatybių panaudojimui. FOAF žodyno turinys yra aprašomas FOAF vardų erdvės dokumente, kuris yra pateikiama adresu internete: <http://xmlns.com/foaf/spec/>.

Šis dokumentas aprašo, kaip naudojant XML/RDF sintaksę (bylų formatą) ir sąlygas keliamas FOAF žodyne, publikuoti ir interpretuoti aprašus, kurie yra aprašomi dabartiniame pasauliniame tinkle. FOAF naudojamų trilypių struktūrų pavyzdys:



Pav. 2 FOAF žodyno naudojimo RDF grafuose pavyzdys

URI adresas adresui (<http://vardenispavardenis.org>) priskiriamos FOAF žodynu nusakomos ypatybės. Grafa aprašančios ypatybės:

- *dc:title* ypatybė aprašoma meta antraštė „Vardenio Pavardenio tinklalapis“.
- *foaf:maker* nusako asmenį kuris sukūrė tinklapį, kuris yra prieinamas šiuo uri adresu. Kūrėją gali nusakyti 2 ypatybės: jo vardas ir paveikslėlis.
- *foaf:name* ypatybė nusako tinklapio kūrėjo vardą „Vardenis Pavardenis“.
- *foaf:depiction* nusako ypatybę paveikslas kuri yra prieinama URI adresu iš Joe Smith sukurto tinklapio.

3.6.3. vCards žodynas

Šiuo metu yra paplitę elektroninės adresų knygos. Šios knygos naudojamos įvairiuose mobiliuose įrenginiuose. Naudodami vCards žodyną galime netik aprašyti informaciją, kuri yra laikoma adresų knygutėje ar aplikacijos byloje, bet ir palengvina informacijos keitimąsi tarp šių įrenginių.

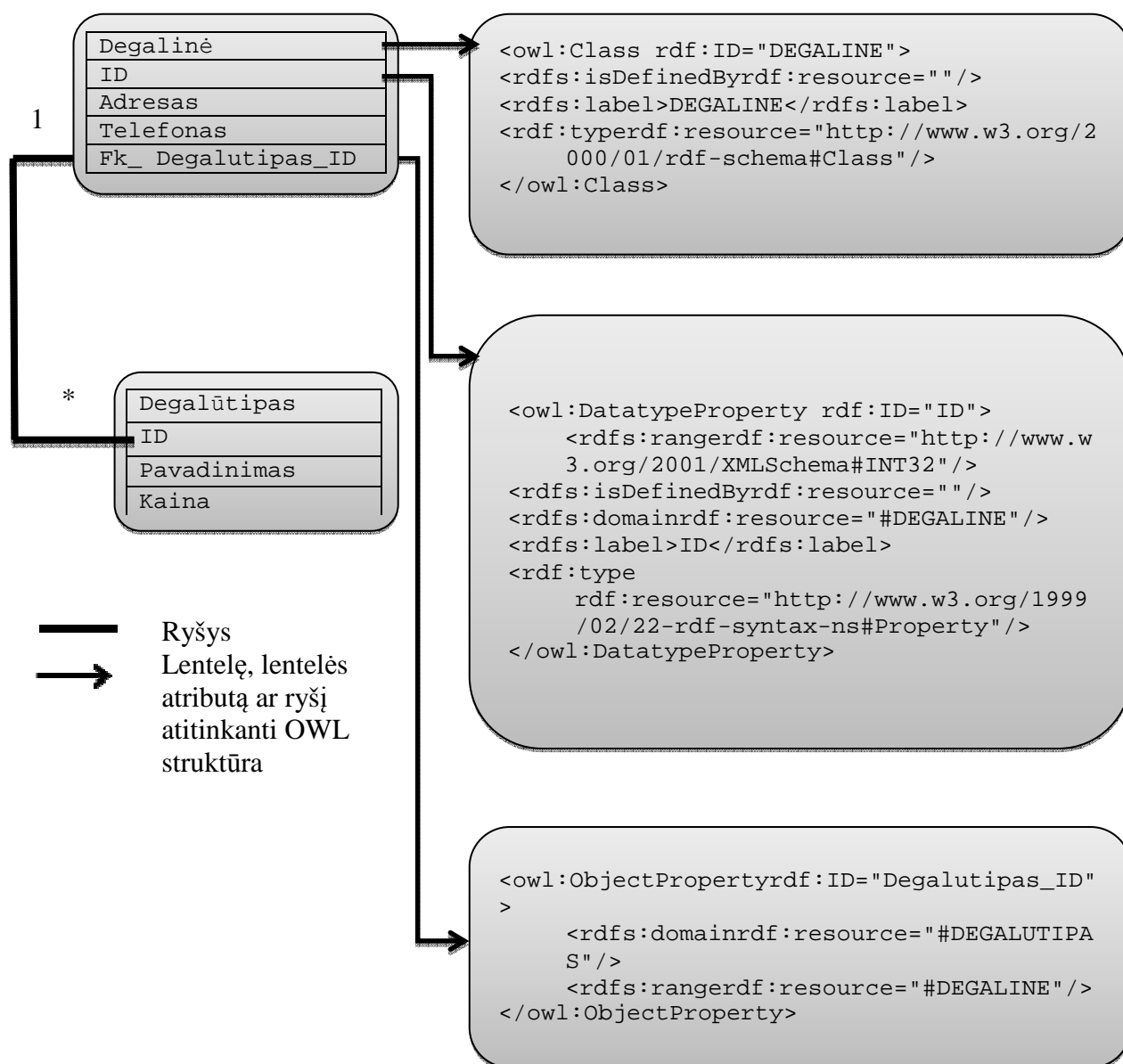
vCards žodynas skirtas pateikti informaciją apie asmenis ir kitus subjektus (pvz. struktūrizuotą vardą ar pristatymo adresą, elektroninio pašto adresą, telefono numerius, fotografijas, paveikslėlius, muziką arba vaizdo klipus).

3.7. Reliacinės duomenų bazės ir semantika

Dalį semantikos duomenų galima gauti išnagrinėjus duomenų bazės schemą. Reliacinės duomenų bazės lentelės atitinka realaus pasaulio objektų ryšius ir yra pasiūlyta sprendimų ontologijai iš dalykinės srities išgauti.

Reliacinės duomenų bazės lentelės galima aprašyti ontologija, kiekvienai lentelei sukuriant naują ontologijos klasę *owl:Class*, lentelės atributui sukuriant *owl:DatatypeProperty* ypatybę, o lentelės ryšiam nusakyti panaudojant *owl:ObjectProperty* ypatybę. Pav. 3 paveikslėlyje pateikiamas pavyzdys ontologijos sukūrimo iš reliacinės duomenų bazės pavyzdys.

Lentelei Degalinė sukuriama OWL ontologijos klasė degalinė, atributui ID sukuriama OWL ontologijos savybė *DatatypeProperty*, o ryšiui *Fk_DegaluTipas_ID* sukuriama ontologijos savybė *ObjectProperty*. Analogiškai ontologijoje aprašomi ir kitos lentelės, atributai bei ryšiai.



Pav. 3 OWL ontologijos sudarymas iš reliacinės duomenų bazės

3.8. Semantinio tinklo saugojimas ir paieška

3.8.1. SPARQL - užklausų kalba

RDF yra grafinis duomenų formatas, skirtas nusakyti pasauliniame tinkle laikomą informaciją. Naudodami SPARQL kalbą galime išreikšti užklausas skirtingiems duomenų šaltiniams ir nesvarbu ar duomenys yra laikomi vietinėje RDF duomenų saugykloje ar peržiūrima naudojant tarpinę programinę įrangą. SPARQL užklausų kalba yra pajėgi atlikti užklausas reikalaujantiems ir pasirinktiniems grafiniams šablonams kartu su šių šablonų konjunkcijomis ir disjunkcijomis. SPARQL užklausų kalba taip pat nusako pratęsimas

reikšmių tikrinimo ir apribojimo užklausa nurodydamos šaltinio RDF grafinį šabloną. SPARQL užklausa rezultatu gali būti RDF grafinių šablonų rinkinys.

SPARQL kalbos pavyzdys:

Pavyzdyje parodoma SPARQL užklausa skirta duotame duomenų grafe surasti knygos pavadinimą. Užklausa susideda iš dviejų dalių: SELECT išlygos, kuri nusako kokie kintamieji turėtų atsirasti užklausoje rezultate ir WHERE išlygos, kuri nusako bazinį grafa, kuris turi sutapti su duomenų grafu. Bazinis grafo šablonas susideda iš vieno trilypės struktūros šablono, kuris objektui nusakyti turi vienintelį kintamąjį (?title).

Duomenys:

```
<http://example.org/book/book1>  
<http://purl.org/dc/elements/1.1/title> "SPARQL".
```

Užklausa:

```
SELECT ?title  
WHERE  
{  
  <http://example.org/book/book1><http://purl.org/dc/elements/1.1/  
  title> ?title .  
}
```

Užklausoje rezultatas:

title
"SPARQL"

3.8.2. Joseki - HTTPvariklis SPARQL protokolui palaikyti

Joseki tai yra Jena RDF serveris. Šis serveris skirtas RDF modelių publikavimui pasauliniame tinkle. Modeliai turi priskirtus URI adresus, kurie gali būti prieinami naudojant HTTP GET funkcijas. Šis serveris yra Jena struktūrinė dalis.

Joseki serveris yra konfigūruojamas naudojant servisus. Servisai yra taikomi duomenų apdorojimo metu, kuomet yra vykdomos užklausoje fiksuotiems, iš anksto

nusakyties duomenų rinkiniams arba šis serveris gali dinamiškai surinkti duomenis iš užklausų.

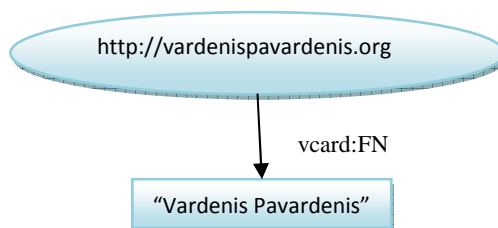
Publikuojant duomenis, jei jau yra anksčiau išpublikuotų duomenų, yra geriau jei yra naudojamos procesorius, kuris neleidžia duomenų rinkiniams būti specifikuojamiems užklausoje ar užklausų protokoluose.

3.8.3. Jena

Ryšiams tarp RDF resursų ir ypatybių nusakyti jena naudoja vCards elektroninio verslo kortelių profilį, kuris apibrėžtas vCards specifikacija. RDF dokumentai struktūrizuoti naudojant vCards specifikaciją taip pat gali būti vadinami 'RDF vCard' dokumentais.

Jena naudojimo pavyzdys

Pavyzdyje naudojamas žmogaus aprašymas. RDF duomenų grafas galėtų atrodyti štai taip:



Pav. 4 RDF grafų aprašymas vCard žodynu jena aplinkoje.

Resursas Vardenis Pavardenis yra vaizduojamas kaip elipsė ir yra identifikuojamas URI aprašu. Šiuo atveju tai yra "http://vardenis pavardenis.org". Su įprasta naršykle šis URI adresas yra neprieinamas. Resursai turi ypatybes. Šiame pavyzdyje labiausia domina tokios ypatybės, kurios galėtų būti naudojamos Vardenio Pavardenio vizitinėje kortelėje. Ypatybės nusakomos vCard žodynu. Ypatybės taip pat yra URI adresai, tačiau šie adresai yra priskirti XML qname formai. Šiuo atveju ypatybė vcard:FN nusako resurso <http://vardenis pavardenis.org> vardą Vardenis Pavardenis. Suprogramuoti šį grafą Java kalboje galima šitaip:

```

// Apibrėžimai
staticStringpersonURI    = "http://somewhere/JohnSmith";
staticStringfullName     = "John Smith";
// Tuščio modelio sukūrimas
Model model = ModelFactory.createDefaultModel();
// Resurso sukūrimas
ResourcejohnSmith = model.createResource(personURI);
// Ypatybės priskyrimas
johnSmith.addProperty(VCARD.FN, fullName);

```

3.8.4. Sesame

Sesame tai yra atvirojo kodo Java struktūra skirta RDF bei RDF-S duomenų kaupimui, užklausimui ir apdorojimui. Ši struktūra gali būti naudojama kaip RDF arba RDF-S duomenų bazė arba kaip Java biblioteka skirta aplikacijoms, kurioms reikia dirbti su RDF duomenimis. Sesame gali būti naudojama kaip reliacinė RDF duomenų saugykla.

3.9. Esami publikavimo įrankiai bei RDF duomenų saugyklos.

Šiame skyriuje nagrinėjamos semantinio tinklo duomenų Kaupyklos. Sugeneruota semantinio tinklo informacija gali būti kaupiama ir saugoma nagrinėjamose duomenų kaupyklose, o vėliau užklausiamas naudojant SPARQL užklausas.

3.9.1. Oracle Database 11g Semantic Technologies

Oracle Database 11g Semantic Technologies yra atvira, standartais paremta saugi, patikima ir naši RDF modelių administravimo platforma. RDF trilypės struktūros yra saugomos indeksuojamos ir užklausiamos lygiai taip pat, kaip ir kiti objektiniai - reliaciniai duomenų tipai.

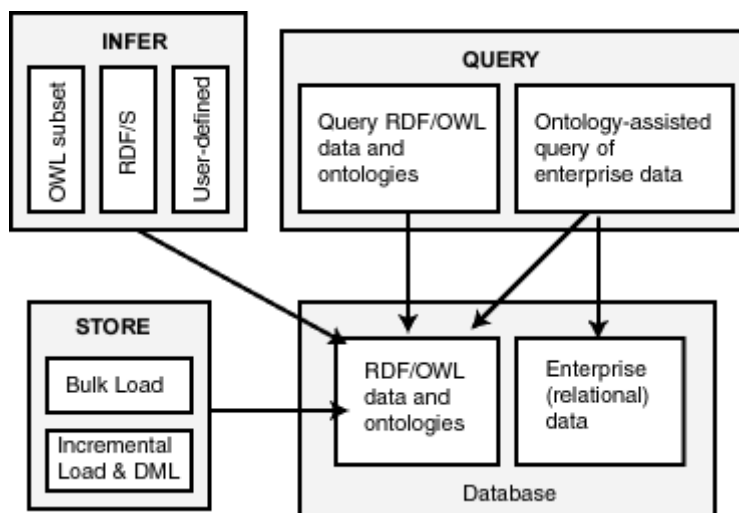
Taikomųjų programų kūrėjai naudoja Oracle 11g duomenų bazę plataus spektro taikomųjų semantinio tinklo aplikacijų kūrimui. Įrankis naudojamas kuriant programas centrinei žvalgybai, teisėtvarakai, integruotai bioinformatikai, sveikatos priežiūrai, finansams, socialiniams tinklams, masinės informacijos priemonėms, žaidimams ir turinio valdymui.

Įrankis naudodamas ontologijas, RDFS, SKOS, OWL semantikas ir vartotojo nusakytas taisykles padeda atrasti naujus duomenų ryšius. Naudodami SPARQL užklausas (Jena, Sesame arba Joseki) galime užklausti Oracle duomenų bazes.

Oracle 11g duomenų bazė palaiko Wc3 standartus nusakytus semantinio tinklo žodynams ir užklausoms (RDF, SKOS, RDFS, OWL, and SPARQL.), taip pat atvirojo kodo

technologijas, tokias kaip Jena, Joseki, ARQ, Sesame, Pellet, D2RQ, Jetty, Cytoscape, GATE, ir Protégé .

Naudodami oracle semantines technologijas galime užklausti reliacines duomenų bazes ir susiedami duomenų bazių duomenis su RDFS, OWL ar vartotojo apibrėžtais žodynais paversti duomenis rdf/owl duomenimis ir ontologijomis.



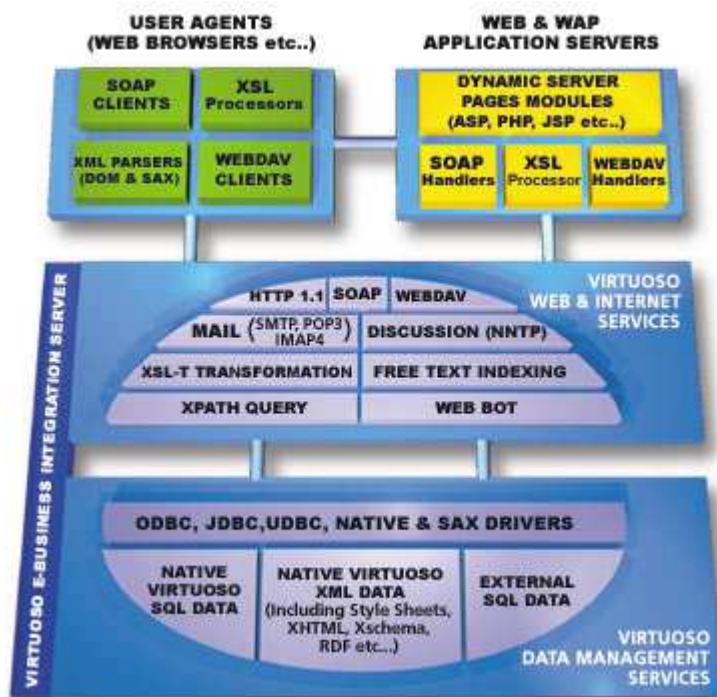
Pav. 5 Oracle sematinių technologijų galimybės. Šaltinis [28]

3.9.2. Openlink Virtuoso

Tai yra multi-modelio duomenų serveris skirtas organizacijoms ir pavieniems asmenims. Šis serveris suteikia galimybę valdyti integruoti duomenis bei suteikia priėjimą prie šių duomenų.

Unikali ir hibridinė serverio architektūra pateikia įprastinį serverio funkcionalumą viename produkte. Įrankio funkcijos padengia šias sritis:

- Reliacinės duomenų bazės;
- RDF duomenų administravimas;
- XML duomenų administravimas;
- Laisvojo teksto turinio valdymas ir indeksavimas;
- Dokumentų Web serveris;
- Susietųjų duomenų (angl.*Linked data*) serveris;
- Web aplikacijų serveris;
- Web servisų (SOAP arba REST) dislokavimo serveris



Pav. Open link virtuoso architektūra, šaltinis [30]

3.9.3. D2RQ ir D2R Server

Tai yra duomenų bazių ir RDF susiejimo įrankis, kuris yra sukurtas java kalba. Šis įrankis, nemodifikuodamas reliacinės duomenų bazės gali susieti bet kokią JDBC duomenų bazę su RDF duomenimis. Įrankis pradėtas kurti 2004 metais Chris Bizer, FU, Berlyne. Įrankis paremtas N3 sintakse

Žymėjimo procesas D2RQ kalba

Reliacinės duomenų bazės sužymėjimui atlikti turime:

- Nusakyti duomenų bazės prisijungimą.
- Apibrėžti subjektus
- Subjektams priskirti ypatybes
- Susieti ypatybes
- Nusakyti sąlygas, sąjungas reikšmes ir pavertimus.

Prisijungimui nusakyti reikia nurodyti duomenų bazės adresą, naudojamą tvarkyklę, vartotojo vardą bei slaptažodį.

```
map:MyDatabase a d2rq:Database;  
  d2rq:jdbcDSN "jdbc:mysql://localhost/mydb";  
  d2rq:jdbcDriver "com.mysql.jdbc.Driver";  
  d2rq:username "user";  
  d2rq:password "password".
```

Objektams apibrėžti reikia nusakyti reliacinės duomenų bazės sritį, kuri yra susiejama, nurodant SQL užklausas (pažymėta raudonai).

```
map:People a d2rq:ClassMap;  
  d2rq:uriPattern http://.../people/@@User.ID@@  
  d2rq:condition "User.deleted=0".
```

Subjektams reikia priskirti ypatybes iš RDF/OWL žodyno (gali būti priskirtas FOAF žodynas):

```
map:People a d2rq:ClassMap;  
  d2rq:uriPattern "http://.../people/@@User.ID@";  
  d2rq:condition "User.deleted=0";  
  d2rq:class foaf:Person.
```

```
map:People a d2rq:ClassMap  
map:name a d2rq:PropertyBridge;  
  d2rq:belongsToClassMap map:People;  
  d2rq:property foaf:nick;  
  d2rq:column "User.name".
```

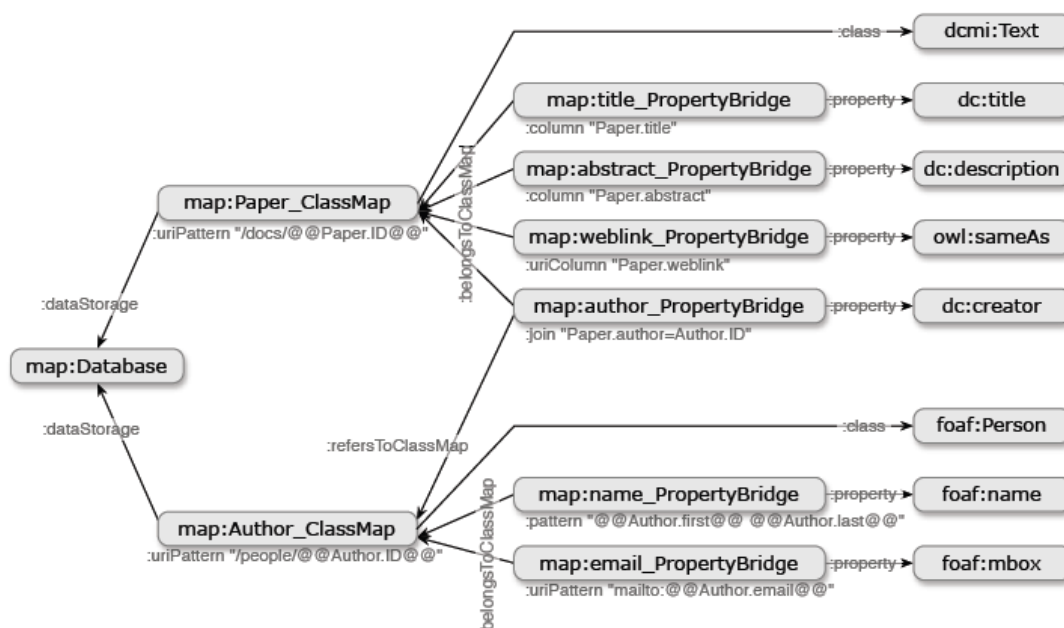
```
map:mbox a d2rq:PropertyBridge;  
  d2rq:belongsToClassMap map:People;  
  d2rq:property foaf:mbox;  
}  d2rq:uriPattern "mailto:@@User.email@".
```

```
map:Photos a d2rq:ClassMap;  
  d2rq:uriPattern "http://.../photo/@@Photo.ID@";  
  d2rq:class foaf:Image .
```

```
map:photo a d2rq:PropertyBridge;  
  d2rq:belongsToClassMap map:People;  
  d2rq:property foaf:made;  
  d2rq:uriPattern "http://.../photo/@@Photo.UserID@".
```


Čia Photo.UserID yra svetimas raktas User.ID.

```
map:photo a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:People;
  d2rq:property foaf:made;
  d2rq:join "User.ID = Photo.UserID";
  d2rq:refersToClassMap map:Photos.
```



Pav. 1 Naudojamos žymėjimo šablono struktūra, šaltinis [26]

Paveikslėlyje parodyta, kaip yra susiejama duomenų bazė, subjektai ir ypatybės.

3.9.4. SquirrelRDF

Internete yra labai daug struktūrizuotos informacijos, tačiau ji nėra saugoma RDF formate. Ne visada yra leidžiama ar norima šią informaciją parsisiųsti ir paversti į RDF formatą. Neapdoroto duomenys dažniausiai yra neprieinami, ir nuolat atnaujinti RDF duomenis būtų sudėtinga. Šiais atvejais ir yra naudojamas SquirrelRDF įrankis, kuris leidžia ne-RDF duomenų saugyklas užklausti naudojant SPARQL užklausų kalbą. Reliacines duomenų bazes galima užklausti naudojant JDBC tvarkyklę ir LDAP serverius, naudojant JNDI tvarkyklę. Priėjimui prie duomenų šaltinių java kalba skirtas ARQ QueryEngine, komandinės eilutės įrankis ir servletas skirtas priėjimui prie SPARQL per http protokolą. Taip gaunama nuolat prieinama RDF informacija.

Modelio žymėjimas SquirrelRDF įrankiu

Naudodami SquirrelRDF įrankį galime publikuoti sužymėtą saugyklą greičiau apdorojamoje formoje. Įrankis nesuteikia galimybės nusakyti numanomus ryšius, kurie yra nusakomi išoriniais raktais arba normalizuoti nenormalizuotus duomenis. Tai supaprastina SquirrelRDF įrankį, kadangi yra susikoncentruojama į RDF žymėjimą ir yra ignoruojamas sudėtingas žodynų ir ontologijų pertvarkymo uždavinys. Įvairiems įrankiams apdoroti nepertvarkytas ontologijas ar žodynus yra žymiai lengviau.

- Duomenų bazių žymėjimas atliekamas naudojant konfiguraciones bylas, kurios nusako labai ribotą ypatybės ar klasės žymėjimą.
- Naudodami *CONSTRUCT* funkciją galime keisti gaunamų RDF duomenų formą.
- Naudojamos N3 ar jena taisyklės.
- Galima transformuoti įeinančią užklausą.

SquirrelRDF naudojimo pavyzdys

Prisijungti prie duomenų bazės galima naudojant komandą:

```
lewis:~/ pldms$ javasquirrelrdf.ExtractConfig
\jdbc:mysql://localhost/conference\com.mysql.jdbc.Driver\user password
http://example.com/db/ >dbmap.ttl
```

Prisijungus atliekamas vienos klasės žymėjimas *ex:people*.

```
ex:people
  a rdfs:Class;
  db:primaryKey ex:people_id;
  db:database <jdbc:mysql://localhost/conference>;
  db:table "people" .
```

Ši žymima klasė atitinka reliacinės duomenų bazės lentelę "people" ir ši lentelė turi pirminį raktą. Taip yra nusakoma *ex:people* klasės ypatybė. Ši ypatybė sužymi lentelės stulpelį „*id*“. Stulpelio tipas yra nenaudojamas, tačiau mes galime matyti kad tai yra *integer* tipo kintamasis.

```
ex:people_name
  a rdf:Property ;
  rdfs:domain ex:people ;
  db:col "name";
  db:colType "varchar".
```

Kita `ex:people` klasės ypatybė yra `ex:people_name`. Klasės ir ypatybės eilutės yra atsitiktinai panašios, tačiau taip funkcija `ExtractConfig` sugeneruoja RDF duomenis.

Sugeneruotas rezultatas

Įrankio naudojama reliacinės duomenų bazės lentelė:

People	
id	name
1	Damian
2	Libby

Sugeneruojami RDF duomenys:

```
ex:people;id=1 a ex:people ;
  ex:people_id 1 ;
  ex:people_name "Damian" .
ex:people;id=2 a ex:people ;
  ex:people_id 2 ;
  ex:people_name "Libby" .
```

Apibendrinus:

- Lentelės tampa klasėmis;
- Stulpeliai kartu su lentelės klasėmis tampa savo srities ypatybėmis;
- Subjektai yra eilutės, kuriuos yra konstruojamos naudojant pirminį raktą ir lentelės klasę. Jei yra nenurodytas joks raktas, tuomet yra naudojamas tuščias mazgas.\

3.9.5. METAmorphoses

Žymėjimo kalba ir naudojamas šablonas:

METAmorphoses įrankio kūrėjai semantinio tinklo žymėjimui ir duomenų transformavimui pateikia dvi XML pagrindu sukurtas kalbas. Kalbos skirtos schemos žymėjimo dokumentams aprašyti ir naudojamo šablono dokumentams aprašyti.

Reliacinės duomenų bazės ir semantinio tinklo duomenų susiejimui nusakyti naudojama RDFS ontologija, kuri nusako žodyną, kurio pagrindu bus sugeneruota RDF byla.

```
<rdfs : Classrdf :ID =" Person ">
  <rdfs :label >Person </ rdfs :label >
</ rdfs :Class >

<rdf : Property rdf :ID =" surname ">
  <rdfs : range rdf : resource =" http :// www .w3.org /2001/ XMLSchema #
    string " />
  <rdfs : domain rdf : resource ="# Person "/>
</rdf : Property >
```

Šioms kalboms pasirinkta XML sintaksė, kadangi:

- XML kalba buvo sukurta tokio tipo dokumentams pateikti. Ši kalba yra įprasta pasaulinio tinklo bei semantinio tinklo bendruomenėms.
- XML sintaksė gali būti analizuojama or apdorojama naudojant Dokumentų Objektų Modelį (angl.*Document Object Model* arba *DOM*), kuris gali būti nagrinėjamas ir valdomas keletu nemokamų arba komercinių bibliotekų.
- XML sintaksė yra suprantama žmogui. Žmogus gali laisvai redaguoti XML dokumentus
- RDF publikavimo formatas taip pat pateikiamas XML kalba. Jei šablono dokumentas yra šablonas RDF dokumentui, žmogui yra intuityvu kurti XML kalba nusakytą duomenų medį, kadangi RDF modelis gali būti įsivaizduojamas irgi kaip medis.

METAmorphoses Schemas žymėjimo kalba

Schemas žymėjimo kalba yra atsakinga už duomenų sužymėjimą tarp duomenų bazės schemas ir ontologijos. Ji gali susieti duomenų bazės schemas dalis su klasėmis ir ontologijos ypatybėmis. Vėliau šie žymėjimai naudojami šablono lygmenyje.

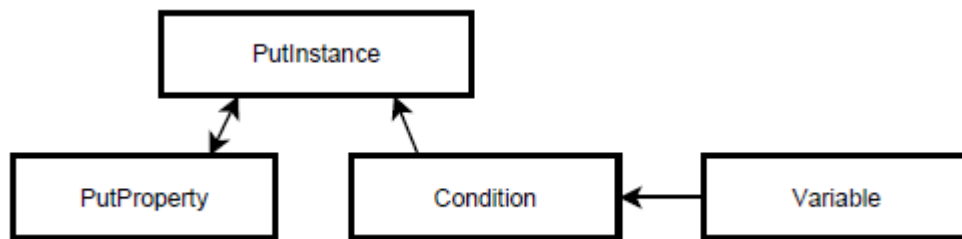
Žymėjimo dokumento kalba sukurta:

- Nusakyti kaip duomenų bazės schema ir ontologija nusakyta struktūra turētu būti susietos;
- fragmentų valdymui;
- Sukurti išbaigtą RDF dokumento struktūra;
- Prisijungimui prie bet kokios duomenų bazės sudaryti;

METAmorphoses Schemas šablono kalba

Šablono dokumento kalba yra paremta maža grupe elementų:

PutInstance, PutProperty, Condition ir Variable



Pav. 6 Šablono kalbos elementų hierarchija, šaltinis [21]

Visi šie elementai per atributus „name“, kurie susieja šablono atributą „templateName“, nurodo atitinkamus elementus iš konkretaus schemas žymėjimo dokumento. Žymėjimo elementuose naudojant SQL užklausas galima sudaryti šablono dokumentą, kuris gali būti laikomas dalies šaltinio reliacinės duomenų bazės RDF vaizdu.

Šablono dokumentu galime nusakyti ypatybes. Pavyzdžiui RDF ypatybei *surname* galime priskirti atitinkamą reliacinės duomenų bazės reikšmę kiekvienai ontologijos klasei *person*:

```
<putInstance name = " person " >
<putProperty name = " surname " />
</ putInstance >
```

Naudodami *DatabaseConnection* elementą galime nusakyti duomenų bazę, prie kurios jungiamės:

```
<DatabaseConnection jdbcURL = " jdbcURL "
jdbcDriver = " jdbcDriver "
username = "username "
password="password " />
```

Class elementas skirtas nusakyti ryšį tarp vienos ar daugiau duomenų bazės lentelių ir atitinkamų ontologijos klasių:

```
<ClassTemplateName="templateName "  
  rdfLabel="rdfLabel "  
  sql="sqlQuery ">  
  Content  
</Class>
```

- *templateName* - nusako unikalų žymėjimo fragmento indentifikatorių.
- *rdfLabel* - nusako atitinkamos ontologijos klasės, nurodomos šio fragmento, vardą.
- *sqlQuery* - nusako specifinę RDBMS SQL užklausą, kuri sugražina žymimą duomenų bazės turinį.
- *content* – čia gali būti nurodomas bet koks santykius nusakantis elementas

Taip pat galima nusakyti **atributus**:

```
<Attribute rdfLabel="rdfLabel " [prefix="prefix "  
 [sqlName="sqlName "][sufix="sufix " ] />
```

Ypatybės:

```
<PropertyTemplateName="templateName "  
  rdfLabel="rdfLabel "  
  [sqlName="sqlName "  
  [containerType="BAG | ALT | SEQ | COL" ] >  
  Content  
</Property>
```

Sąlygas:

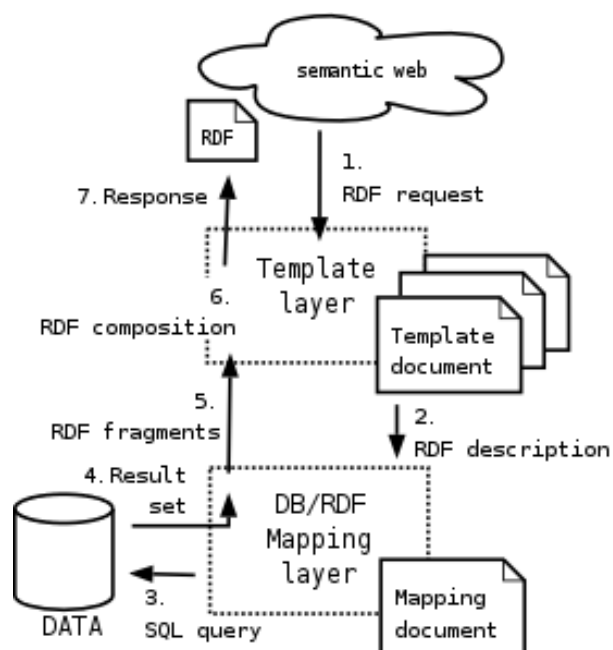
```
<ClassConditionTemplateName="templateName "  
  whereString="whereString "  
  [tableString="tableString " ] />
```

Kintamuosius:

```
<VariableTemplateName="templateName "  
  sqlName="sqlName " />
```

METAmorphoses procesoriaus architektūra

METAmorphoses procesorius yra paremtas 2 sluoksnių transformavimo modeliu. Tam kad būtų galima lanksčiai paversti duomenis ir juos našiai panaudoti įrankio logika yra padalinta į dvi atskiras dalis, kurios yra vadinamos žymėjimo ir šablono lygmenimis (angl. *the mapping layer* ir *template layer*). Duotos duomenų bazės schema yra sužymima į duotos ontologijos struktūrą. Šablono sluoksnis naudoja šį žymėjimą ir sukuria RDF dokumentus. Žymėjimo sluoksnis sudarytas iš kompleksinio ir lakstaus žymėjimo o šablono



Pav. 2 METAmorphoses procesoriaus architektūra, šaltinis [21] lygmuo yra paprastas programuotojo interfeisas sus duomenų transformavimo modeliu.

METAmorphoses žymėjimo redaktorius

Reliacinės duomenų bazės duomenims išpubliuoti semantiniame tinkle galime naudoti METAmorphoses žymėjimo redaktorių, kuris yra sukurtas drag'n'drop principu. Įrankis leidžia prisijungti prie nurodytos duomenų bazės. Programos languose yra matoma duomenų bazės struktūra bei ontologija. Naudodami programos lange pateikiamus duomenis galime sudaryti duomenų bazės žymėjimo dokumentą, kurio privalomieji laukai gali būti redaguojami. Sudarytas dokumentas gali būti išsaugojamas XML formatu ir tiesiogiai naudojamas METAmorphoses processor programoje.

Žymėjimo redaktorius parašytas java kalba, todėl yra nepriklausomas nuo naudojamos platformos. Programoje prisijungimui prie domenų bazės yra naudojama JDBC tvarkyklė ir ontologijų valdymui sukurta aplinka Jena2 RDF API.

METAmorphoses žymėjimo redaktoriaus ypatybės:

- Palaikomi visi žymėjimo kalbos konstruktai;
- Programa realizuota drag'n'drop principu;
- Darbui gali būti naudojama bet kokia OWL ar RDFS ontologija;
- Gali būti naudojama bet kuri RDBMS naudojanti JDBC tvarkyklę;
- Suteikiamas darbo nuoseklumas kuriant žymėjimo dokumentus;

- Sukurti žymėjimo dokumentai gali būti saugomi XML formatu;
- Vienu metu gali būti naudojamas tik viena ontologijos byla. Jei yra planuojama naudoti daugiau ontologijų jos turi būti sujungtos į vieną bylą.

3.10. Įvairių semantinio tinklo kūrimo įrankių palyginimas

Šiame skyriuje pateiktas įrankių palyginimas, kuris remiasi tyrimo autorių, Chris Bizer, Daniel Westphal, atliktu tyrimu. Pateikiami 2007 metų duomenys. Bus palyginti įrankiai, kurie buvo aptikti atlikus analizę (D2R server, Squirrel RDF, JENA, Sesame, Oracle RDF, OpenLink Virtuoso, METAmorphoses);

3.10.1. Palyginimo kriterijai

- Licencija. Naudojama įrankio licencija.
- API-Paradigma. Šis kriterijus nusako, kuri API paradigma yra naudojama RDF duomenų manipuliavimui.
 - **Duomenys arba grafai** – API leidžia pavaizduoti ir užklausti RDF duomenis.
 - **Modeliu paremtas** - API leidžia tik užkrauti, saugoti ir ištrinti visus RDF modelius.
 - **Teiginiais paremtas** - API, RDF duomenys yra manipuluojami kaip rinkinys RDF trilypių struktūrų, kurių kiekviena susideda iš subjekto, teiginio ir objekto.
 - **Resursais paremtas** - API nusako RDF modelį, kaip resursą, kuriam yra priskirtos ypatybės.
 - **Pagal ontologiją** - vaizdas yra resursas yra pridedamas tiesioginis įvairiems ontologijos objektams palaikymas.
- Užklausių kalbos. Sąrašas užklausių kalbų, kurios yra palaikomos įrankio.
- Modelio saugykla. RDF duomenų saugojimo alternatyvų Sąrašas.
 - **Atmintis** – naudojamas jei duomenys yra saugomi tik atmintyje ir tik turi būti išpublikuoti naudojant XML/RDF arba N3 kalbas.
 - **DB** – jei įrankis gali kaupti duomenis standartinėje reliacinėje ar objektų-reliacinėje duomenų bazėje.
 - **Rinkmena** - jei duomenys yra laikomi atskirame bylos formate.
- Palaikomos duomenų bazės. Palaikomų duomenų bazių arba naudojamo abstrakcijos lygmens sąrašas.

- Palaikomi publikavimo formatai. Palaikomų duomenų publikavimo formatai skirti rašyti ir skaityti RDF modeliais
- Samprotavimo palaikymas. Ontologijos arba kalbos palaikomos įrankio.
- RDF Serveris Nurodo jei įrankis yra sudarytas iš RDF serverio, kuris leidžia modelius užklausti per žiniatinklį naudojant HTTP ar SOAP protokolus.

Lentelė 2|vairių semantinio tinklo kūrimo įrankių palyginimas

	D2R	Squirrel RDF	JENA 2.4	Sesame 2.0	Oracle RDF Data Model	OpenLink Virtuoso	METArphoses
Licencija	LGPL	Jena licencija	BSD	LGPL	Komercinė licencija, nemokama kūrimo reikmėms	Apribota komercinė Ir atvirojo kodo licencijos	LGPL
API-Paradigma	Teiginiais paremtas Modeliu paremtas	Modeliu paremtas	Teiginiais paremtas Resursais paremtas Pagal ontologiją	Teiginiais paremtas Resursais paremtas Pagal ontologiją	Teiginiais paremtas	Query-centric	Teiginiais paremtas Pagal ontologiją Modeliu paremtas
Užklausių kalbos	SPARQL	SPARQL	RDQL SPARQL	SeRQL SPARQL	Į SPARQL panaši užklausių kalba	SPARQL	SPARQL
Modelio saugykla	DB		Atmintis DB Rinkmena	Atmintis DB Rinkmena	DB	DB Rinkmena	Atmintis Rinkmena
Palaikomos duomenų bazės	Oracle MySQL PostgreSQL SQL Server HSQLDB Interbase/Firebird, ODBC duomenų šaltiniai	DB palaikančios JDBC sąsają	MySQL PostgreSQL Oracle BerkeleyDB MS SQLServer DB2 HSQLDB	MySQL	Oracle Database	JDBC/ODBC	MySQL PostgreSQL
Palaikomi Publikavimo formatai	RDF/XML	N3, RDF/XML	RDF/XML N-Triples N3 Turtle	RDF/XML N-Triples N3 Turtle TriX JSON skirtas SPARQL protokolui	N-Triples	RDF/XML, RSS, ATOM, GData, SPARQL-P	RDF/XML
Samprotavimo palaikymas	RDFS	RDFS OWL	RDFS OWL-Lite DIG 1.1	RDFS OWL-Lite	RDFS SWRL-like	-	RDFS OWL
RDF Serveris	SPARQL protokolas		SPARQL protokolas	SPARQL protokolas	+	SPARQL protocol ir SIMILE Semantic Bank API	-

3.10.2. Spartos palyginimas

Martin Svihla ir Ivan Jelinek tyrinėdami reliacinių duomenų bazių transformavimo į RDF duomenis bei RDF saugyklas galimybes atliko įvairius spartos testus skirtus įvertinti RDF duomenų sukūrimo greitį bei semantinių duomenų kūrimo efektyvumą. Tyrimo metu buvo netik ištirta bet ir palyginta šių mokslininkų sukurtas METAmorphoses processor įrankis, bet ir buvo lyginamas su kitais reliacinių duomenų bazių turinio transformavimo įrankiais bei RDF saugyklomis, kuriose yra saugomi sugeneruotas reliacinių duomenų bazių turinys.

Tyrimo metu naudoti XML duomenys parsisti iš DBLP informatikos bibliografijos. Reliacinė duomenų versija sudaryta iš 6 nesuindeksuotų lentelių ir yra sudaryta iš 881,876 įrašų. Testuojant METAmorphoses, D2RQ, SquirrelRDF duomenys buvo patalpinti MySQL duomenų bazėje. Kuomet buvo testuojama Jena bei Sesame greitaveika sugeneruoti DBLP informatikos bibliografijos RDF duomenys, turinys 1,608,344 įrašų, buvo patalpinti Jena bei Sesame saugyklose.

Tyrimo aplinka

Testai buvo atliekami naudojant Intel Pentium M procesorių 1400MHz su 1536 MB RAM, bei Linux(i386 operacinę sistemą) su 2.6.12 branduolio versija. Tyrimui naudojama Java Virtual Machine *1.5.0 01-b08* versija. Testai buvo atlikti naudojant JBenCh₃ Java testavimo aplinką.

Testavimas

Užklausų gražinamo rezultato tyrimas:

Lentelė 3 Užklausos gražinamo rezultato tyrimas, šaltinis [21]

Nr.	Užklausa	Gautų trilypių struktūrų skaičius
1	CONSTRUCT * WHERE {?r rdf:type d:Series.}	24
2	CONSTRUCT * WHERE {?r rdf:type d:Publisher.}	64
3	CONSTRUCT * WHERE {?r rdf:type d:Proceeding500.}	500
4	CONSTRUCT * WHERE {?r rdf:type d:Proceeding1500.}	1500
5	CONSTRUCT * WHERE {?r rdf:type d:Proceeding.}	3007

Rezultato gavimo laiko tyrimas, pateikiama milisekundėmis

Lentelė 4 Užklausos rezultatų gražinimo laiko tyrimas, šaltinis [21]

Sistema	Testo numeris. (Gautų trilypių struktūrų skaičius)				
	1 (20)	2 (64)	3 (500)	4 (1500)	5(3007)
METAmorphoses	84	230	1840	5692	13124
SquirrelRDF	830	1314	5180	16228	42504
D2RQ SPARQL	522	1332	7730	25530	45704
Jena SPARQL	482	1444	8296	27478	49968
D2RQL Graph API	366	1134	6434	20922	38253
Jena Graph API	368	1028	6902	22874	42588
Sesame1 SeRQL	194	423	2144	6624	12826

Užklausų gražinimo laikas. tiriama semantinių duomenų grafų sudėtingumo įtaka gražinamų rezultatų laikui nustatyti.

Lentelė 5 RDF grafo sudėtingumo įtakos rezultatų gražinimo laikui tyrimas, šaltinis [21]

Sistema	Testo numeris. (Gautų trilypių struktūrų skaičius)			
	1 (2)	2 (4)	3 (6)	4 (8)
METAmorphoses	28	76	110	124
SquirrelRDF	640	678	768	808
D2RQ SPARQL	252	426	674	850
Jena SPARQL	212	360	456	552
D2RQL Graph API	106	224	434	506
Jena Graph API	94	150	204	262
Sesame1 SeRQL	100	198	272	324

Užklausų būsenos sudėtingumo tyrimas, rezultatų gražinimo laikas milisekundėmis:

Lentelė 6 Užklausų būsenos sudėtingumo tyrimas, rezultatų gražinimo laikas milisekundėmis, šaltinis [21]

Sistema	Testo numeris. (Gautų trilypių struktūrų skaičius)			
	1 (8)	2 (2)	3 (2)	4 (2)
METAmorphoses	78	36	30	32
SquirrelRDF	636	582	9670	598
D2RQ SPARQL	618	336	17480	360
Jena SPARQL	544	240	30794	232
Sesame1 SeRQL	238	124	110	126

Taigi lentelėje Lentelė 2, matome jog nėra konkretaus programinio paketo lyderio, todėl remdamiesi šia lentele negalime nustatyti, kuris paketas yra tinkamiausias, tačiau remdamiesi atliktu spartos tyrimo matome, jog daugeliui spartos palyginimo tyrinėtų atvejų METAmorphoses paketas užklausas duomenims atlikdavo ir rezultatus sugražindavo greičiausiai, todėl darome išvadą jog šis paketas yra tinkamiausias semantinio tinklo taikymo demonstracijai atlikti. Oracle ir OpenLink Virtuoso siūlomas sprendimas yra netinkamas, kadangi realizacijai reikalingos papildomos komercinės licencijos.

3.11. Ontologijų kūrimo įrankiai.

Ontologijoms kurti naudojami įrankiai supaprastina ontologijos kūrimą, bei padeda patikrinti jau turimas OWL ir RDF rinkmenas ir įvertinti ar šios yra korektiškos.

3.11.1. TopBraidComposer

TopBraidComposer yra aplinka skirta kurti ir plėtoti WC3 semantinio tinklo standartus: RDFs, OWL tinklo ontologijas, Sparql užklausų kalbą. Šia aplinką kūrėjai pateikia trejomis versijomis:

- Free Editon
- Standart editon
- Maestro editon

Freediton versijoje yra pateikiamos priemonės skirtos kurti ir redaguoti RDF/OWL bylas ir atlikti SPARQL užklausas šioms byloms. Taip pat, naudojant SPARQL užklausų kalbą, galima apibrėžti verslo taisykles ir vientisumo apribojimus.

Standart Etition versijoje yra žymiai praplėčia FreeEdition versijos funkcionalumą. Šioje versijoje suteikiamas grafinis redaktorius skirtas RDF grafų bei clasių diagramų

kūrimui, taip pat pateikiami įrankiai sistemų logikos kūrimui (*Jena SDB/TDB, AllegroGraph, Oracle 11g and Sesame*). Taip pat pateikiamos priemonės įvairių duomenų formatų importavimui (*XML, Excel, RDBM*) taip pat daugelį kitų priemonių skirtų kurti semantius tinklus palaikančias aplikacijas.

Maestro Edition versija yra skirta žiniatinklio aplikacijų bei paslaugų kūrimui. Ši versija turi žiniatinklio serverį skirtą kuriamų aplikacijų testavimui ir gali būti naudojama SPARQLMotion skriptų kūrimui, duomenų grandinių kūrimui ir integruotų duomenų servisų kūrimui.

3.11.2. Protégé

Protégé tai yra atvirojo kodo įrankis sukurtas Stanford Medical Informatics centre. Pagrinde šis įrankis buvo kurtas biomedicininėms aplikacijoms tačiau jis yra nepriklausomas nuo sistemos domeno, todėl gali būti naudojamas įvairiose srityse.

Protégé modelis yra paremtas paprastu bei lanksčiu metamodeliu, kuris gali būti palyginamas su objektiškai orientuotomis ir karkasais valdomomis sistemomis.

Naudodamį įrankį galime pavaizduoti ontologijas, kurios susideda iš klasių, ypatybių ar ypatybių charakteristikų.

Naudojamos klasės vadinamos :STANDARD-CLASS ir gali turėti tokias ypatybes, kaip :NAME ar :DIRECT-SUPERCLASSES. Tokia metamodelio struktūra leidžia lengvai praplėsti ir pritaikyti Protégé siūlomą funkcionalumą kitoms priemonėms. Metamodelis buvo praplėstas palaikyti UML ir OWL apdorojimą.

Protégé-OWL OWL redaktorius leidžia:

- Užkrauti ir saugoti bei redaguoti OWL ir RDF ontologijas;
- Redaguoti bei vizualizuoti klases, ypatybes ar SWRL taisykles;
- Aprašyti logines klasių charakteristikas kaip OWL išraiškas;

3.12. Taikymo pavyzdžiai

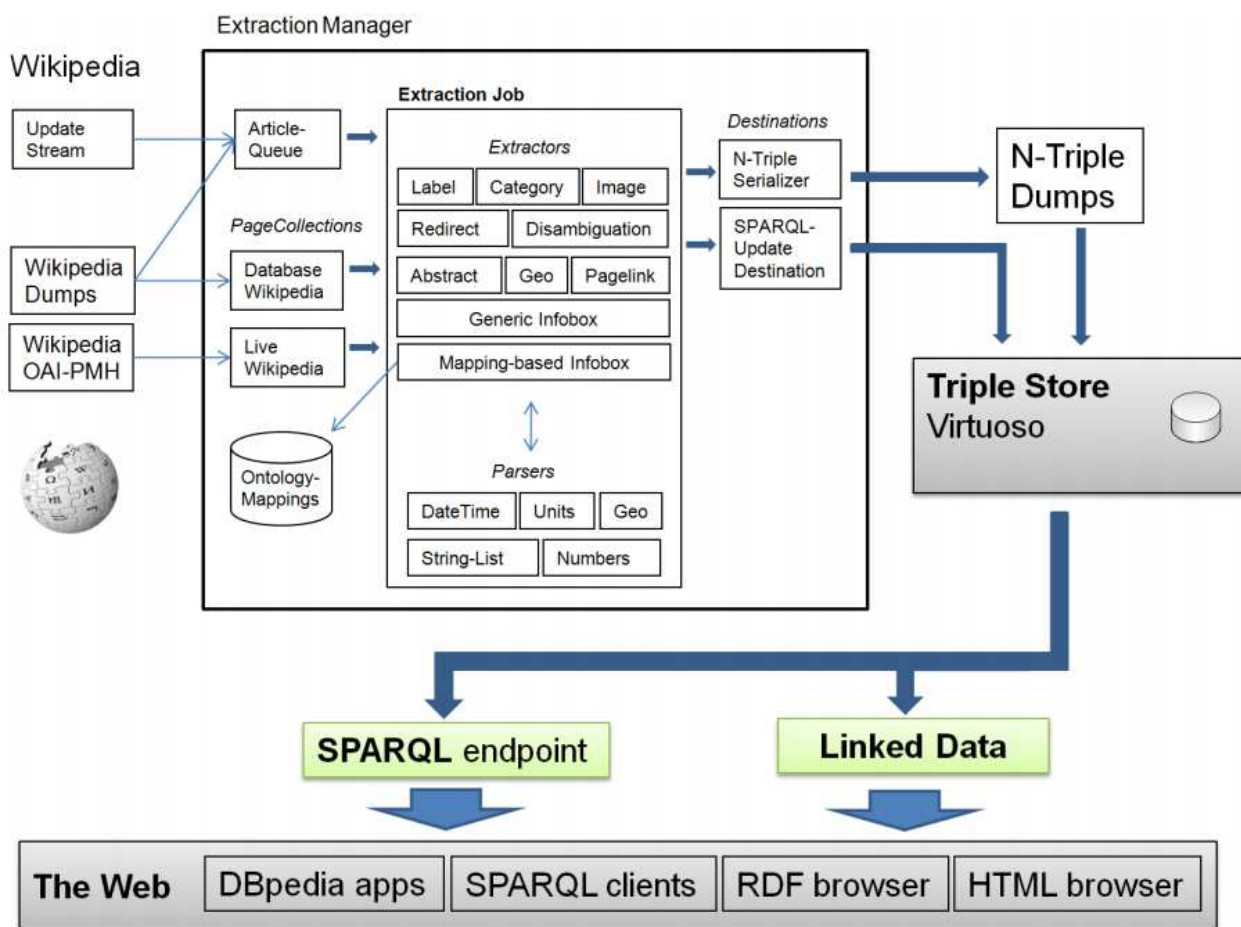
Nagrinėjant jau sukurtus semantinio tinklo taikymo sprendimus galima įvertinti ar yra naudojamos jau žinomos technologijos ar įrankiai bei atrasti dar nematytus būdus bei nežinomas technologijas naudojimas semantiniame tinkle. Tai pat, taikymo pavyzdžių

analizė yra puikus būdas išvelgti sprendimo privalumus, taikymą įvertinti padarytas klaidas bei šių išvengti.

Šie sprendimai pasirinkti, kadangi juose naudojamos semantinio tinklo duomenų publikavimo technologijos, o informacija publikavimui surenkama iš įvairių šaltinių ir susiejama.

3.12.1. DBpedia

<http://dbpedia.org/> - šioje svetainėje pateikiama struktūrizuota *wikipedia.org* informacija, kuri yra pateikiama pasauliniame tinkle. DBpedia leidžia užduoti sudėtingas užklausas į wikipedia.org ir susieti kitus duomenų rinkinius su wikipedia.org serveriuose laikoma informacija. DBpedijoje yra aprašyta daugiau nei 2.6 milijonų esybių, kurios yra aprašytos žmogui suprantama RDF kalba trisdešimčia kalbų. DBpedia pateikia geografinę informaciją, duomenis apie žmones, kompanijas, filmus, muziką, genus, vaistus, knygas, ir mokslines publikacijas.



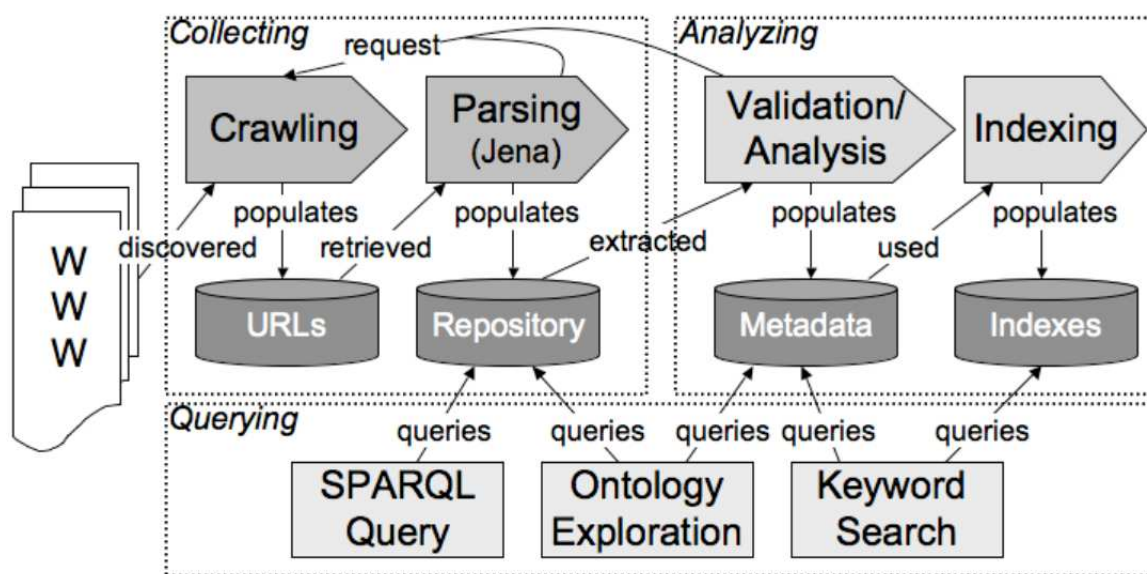
Pav. 7DBpedia struktūra [10]

DBpedia turinio išgavimo mechanizmas sudarytas iš:

- **Puslapių kolekcijų** (angl. *Page collections*), kurios yra vietinių nutolusių wikipedijos straipsnių abstrakcija;
- **Paskyrimo vietų** (angl. *Destinations*), kuriose laikoma išgauta RDF trilypė (angl. *triples*) informacija;
- **Išgavėjo** (angl. *extractor*), kuris paverčia specifinę wikipedia žymėjimo dalį į trilypę informaciją;
- **Analizatoriaus** (angl. *parser*), kuris padeda išgauti duomenis iš wikipedia, paversdamas skirtingų tipų duomenis ir skaldydamas pažymėtas wikipedia sritis į sąrašus;
- **Išgavimo užduočių grupės** (angl. *Extraction Jobs group*), kurioje surinkta informacija, išgavėjai ir priskyrimo vietos sugrupuojami į darbo eigą;
- **Išgavimo valdymo sistemos** (angl. *Extraction manager*), kuris valdo wikipedijos straipsnių perdavimą išgavėjams ir perduoda išgautą informaciją.

3.12.2. Watson

Watson yra dar vienas semantinio tinklo galimybių taikymo pavyzdys. Šį tinklą galime pasiekti adresu: <http://kmi-web05.open.ac.uk/WatsonWUI/>. Šis semantinis tinklapis surenka, išanalizuoja ir suteikia prieigą prie ontologijų ir semantinių duomenų per internetą. Watson tinklapis paskirtis yra padėti kurti semantinių tinklų taikomas programas, kurios dinamiškai pasirenka sujungia ir išnaudoja žinias, kurios yra publikuojamos semantiniame tinkle.



Pav. 8 Watson sistemos architektūra. Šaltinis [11]

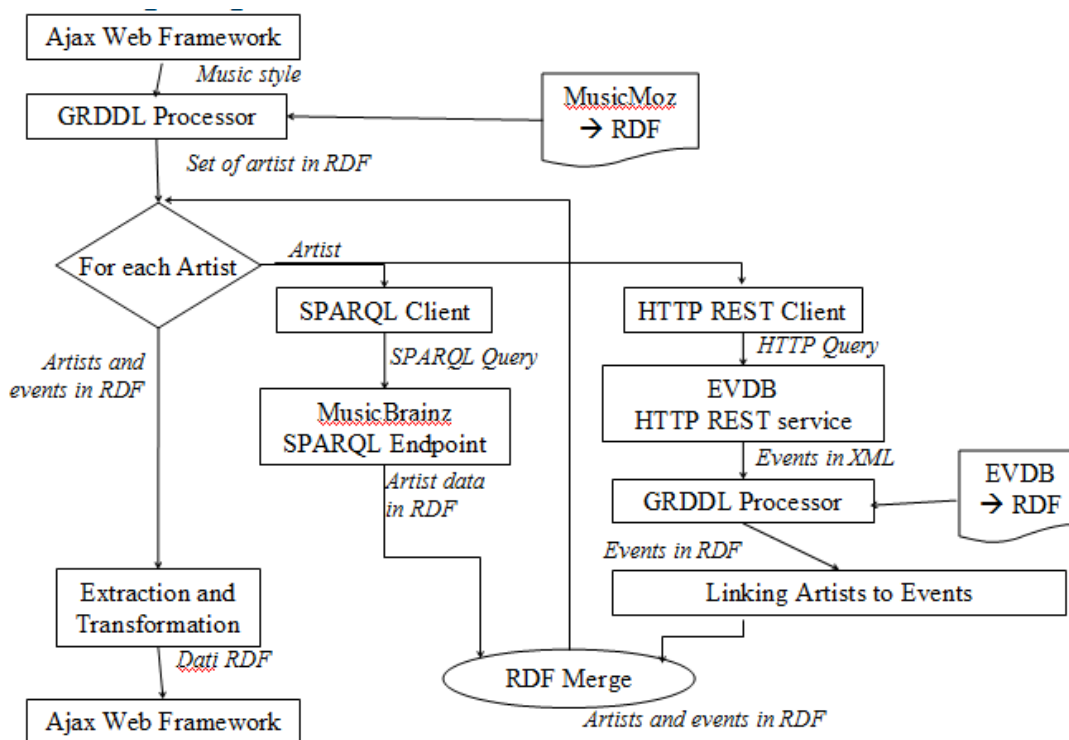
Watson sistema sudaryta iš 3 sluoksnių:

- **Ontologijos skanavimo ir aptikimo sluoksniu** (Collecting sritis), kuriame tiriant ontologijomis paremtas nuorodas, surenkamas internetu prieinamas semantinis turinys;
- **Validavimo ir analizavimo sluoksniu** (Analyzing sritis), kuris yra architektūros šerdis. Šis sluoksnis užtikrina, kad duomenys apie renkamos semantinės informacijos kokybę yra apskaičiuojami, saugomi ir indeksuojami;
- **Užklausų ir navigacijos sluoksniu** (Querying sritis), kuris užtikrina prieigą prie suindeksuotų duomenų per daugybę mechanizmų, kurie leidžia tyrinėti įvairias semantinių duomenų galimybes.

Gilesniame techniniame lygyje šie 3 sluoksniai yra talpinami tinklo serveryje, naudojant RDMS (MySQL) komunikacijai arba surinktų semantinių duomenų panaudojimui. Visi Watson sistemos komponentai yra aprašyti Java kalba.

3.12.3. Muzikos duomenų publikavimas semantiniame tinkle.

Emanuele DellaValle dirbantis profesoriaus darbą elektronikos ir informacijos skyriuje, Politecnico di Milano įstaigoje, siūlo tokią semantinės taikomosios programos struktūrą, kuri parodytų informaciją apie Muzikos atlikėjus, muzikos atlikėjų koncertus, vietovę, kurioje vyksta koncertai:



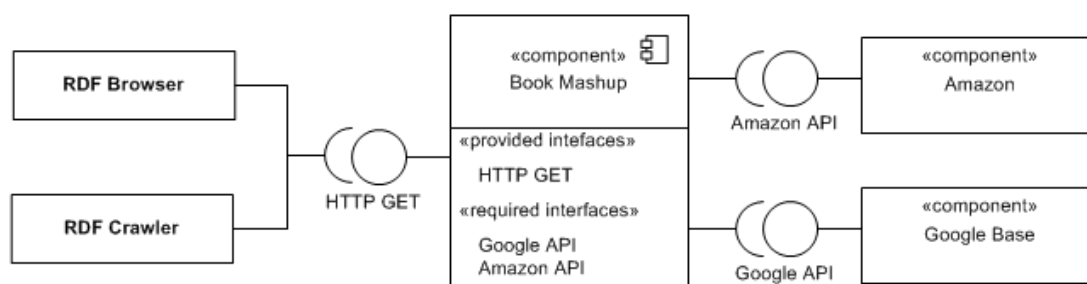
Pav. 9 Semantinio tinklo taikomoji programa apie muzikos atlikėjus, šaltinis [31].

Ši taikomoji programa nuskaityto duomenis iš MusicMoz, EVDB ir MusicBrainz tinklapių ir paverčia nuskaitytą informaciją į semantinio tinklo duomenis. Duomenys pateikiami RDF formatu ir vėliau nuskaityta informacija yra susiejama ir atvaizduojama naudojant AJAX Framework.

3.12.4. Knygų aprašymų pateikimas semantiniame tinkle RDF kalba

Naudojant semantinio tinklo technologijas galima pateikti informaciją apie knygas, autorius, atsiliepimus ir knygytus RDF formatu.

Kiekvienas HTTP URI adresas yra priskiriamas knygai pagal ISBN numerį. Kuomet yra kreipiamasi pagal priskirtą URI adresą yra gražinami duomenys naudojant [Amazon API](#) ir [Google Base API](#).



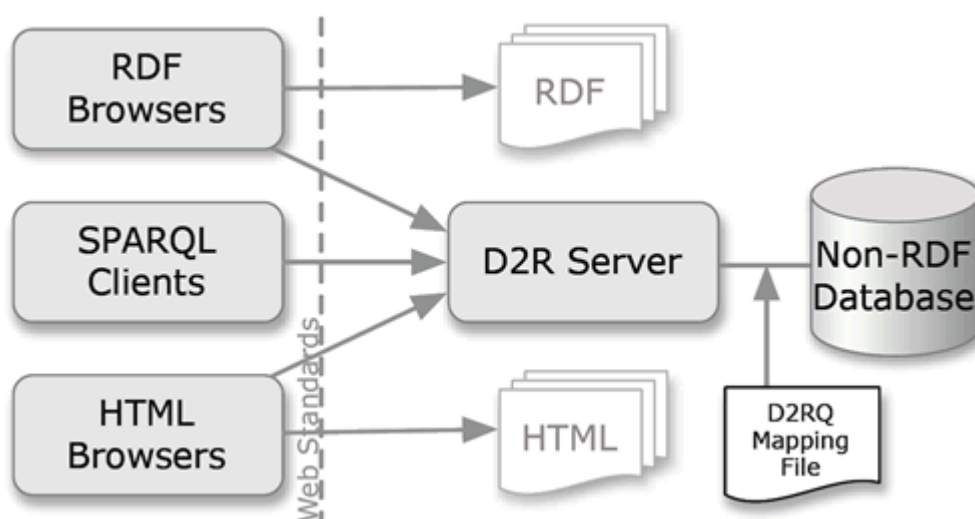
Pav. 10 Knygų aprašymų pateikimas semantiniame tinkle RDF kalba

3.13. Reliacinių duomenų bazių publikavimas semantiniame tinkle

Jei duomenys yra laikomi reliacinėse duomenų bazėse, galima publikuoti su egzistuojančia duomenų baze susijusius duomenis (angl. *linked data*). Reliacinės duomenų bazės turinį galiam pateikti naudojant D2R Server arba OpenLink Virtuoso serverius. Šiuose serveriuose galima laikyti sąsajas tarp reliacinės duomenų bazės turinio ir semantiniame tinkle naudojamų duomenų, kurie yra pateikiami RDF formatu.

D2R Server arba OpenLink serverių privalumai:

- Serveriai suteikia galimybę RDF ir HTML naršyklėms pasiekti nereliacinėse duomenų bazėse publikuojamą turinį;
- Naudojant SPARQL Užklausų kalbą suteikia galimybę programoms pasauliniu tinklu užklausti duomenų bazes;
- Duomenų bazių turinys kartografuojamas naudojant deklaratyvaus kartografavimo principą, kuris aprašo kaip yra identifikuojami resursai ir kaip yra sugeneruojamos duomenų bazės turinio nuosavybės vertės;
- Serveriai gali SQL užklausas paversti SPARQL užklausomis, todėl nereikia dubliuoti duomenų bazių turinio.



Pav. 11 Reliacinių duomenų bazių publikavimas, šaltinis [26]

Jau yra egzistuojančių D2R serverių, kurie publikuoja reliacinių duomenų bazių turinį:

- 1) Berlin DBLP Bibliography Server, prieinamas adresu: <http://www4.wiwiss.fu-berlin.de/dblp/>
- 2) Hannover DBLP Bibliography Server, prieinamas adresu: <http://dblp.l3s.de/d2r/>
- 3) Web-basedSystems @ FU Berlin Group Server, prieinamas adresu: <http://www4.wiwiss.fu-berlin.de/is-group/>
- 4) EuroStatCountriesandRegions Server, prieinamas adresu: <http://www4.wiwiss.fu-berlin.de/eurostat/>

3.14. Analizės išvados

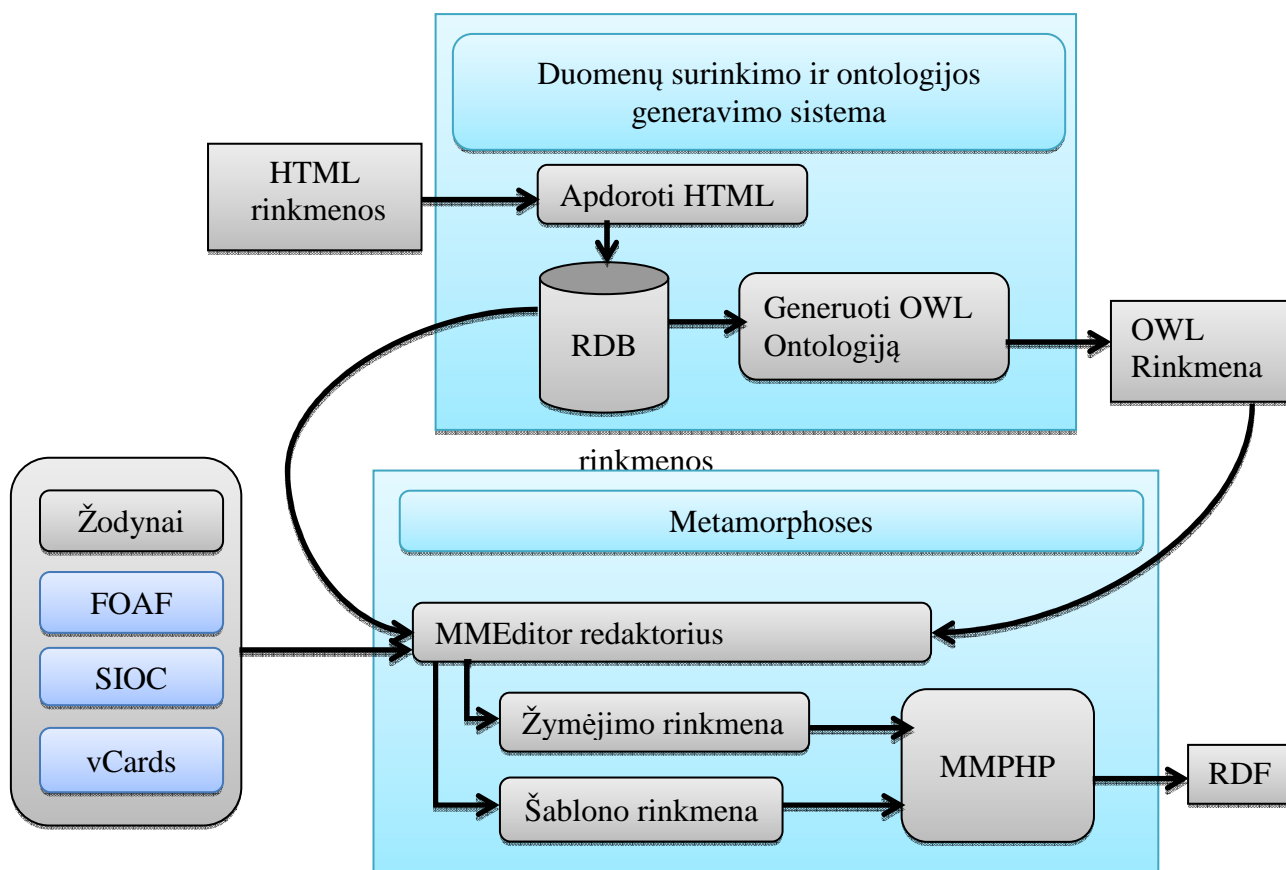
- Naudojant publikavimo įrankius (OpenLink Virtuoso, D2R Server, SPARQL, RDF ir kitas) galima publikuoti tiek reliacines duomenų bazes turinį, tiek publikuoti duomenis semantiniame tinkle. Tačiau duomenų publikavimui yra būtina turėti OWL ontologiją, yra poreikis šią ontologiją susikurti.
- Jena bei Sesame įrankiai yra plačios paskirties įrankiai ir šiems įrankiams pritaikyti konkrečiai paskirčiai reikia labai daug programuoti, kad būtų galima išpublikuoti reliacinės duomenų bazės turinį, taigi šie įrankiai netinka užduočiai atlikti.
- METAmorphoses ir D2R įrankiai yra tinkamiausi semantinio tinklo galimybių tyrimui bei taikymui atlikti, kadangi šie įrankiai yra pateikiami su atvirojo kodo licencija. Atliktas įvairių įrankių spartos palyginimas (3.10 skyrius) parodė, jog METAmorphoses įrankis yra sparčiausias, taigi kartu ir pranašesnis už D2R. Naudodami METAmorphoses galime pademonstruoti praktinį taikymą ir išpublikuoti reliacinės duomenų bazės turinį.
- Yra keletas jau sukurtų reliacinių duomenų bazių turinio semantiniame tinkle publikavimo įrankių. Duomenų išpublikavimui tinka OpenLink Virtuoso, D2R server, METAmorphoses įrankiai. OpenLink Virtuoso pateikiama duomenų publikavimo galimybė kuri yra prieinama tik komercinėje versijoje. D2R server bei METAmorphoses yra įrankiai sukurti pagal atvirojo kodo licenciją todėl gali būti lengvai prieinami ir naudojami reliacinių duomenų bazių publikavimo semantiniame tinkle galimybių tyrimui bei taikomųjų semantinio tinklo programų kūrimui.
- Duomenų publikavimui iš reliacinės duomenų bazės yra reikalinga tam tikros dalykinės srities ontologija, tačiau nevisų sričių ontologijos yra sukurtos, todėl yra poreikis sukurti priemones, kurios automatiškai sugeneruotų OWL iš reliacinės duomenų bazės schemas.

- Kadangi dalis informacijos pateikiama HTML puslapiuose, o tiesioginės prieigos prie informacijos duomenų bazių nėra, todėl yra poreikis sukurti priemones automatiškai nuskaityti HTML puslapyje publikuojamus duomenis ir patalpinti juos į reliacinę duomenų bazę semantiniam išpublikavimui.

4. Sistemos reikalavimai

4.1. Reikalavimų specifikacija

Sistemos darbui pademonstruoti kuriama sistema nuskaitanti HTML formatu saugomus duomenis, šiuos duomenis apdorojanti bei talpinanti į reliacinę duomenų bazę. Nuskaičius duomenų bazę sugeneruojama OWL ontologija. Panaudojus MMEditor programą apjungiami OWL Ontologija bei RDB ir sudaromos Metamorphoses šablono bei žymėjimo rinkmenos. Šias rinkmenas, aprašomas XML kalba, galima sudaryti rankiniu arba automatiškai būdu. Rinkmenos apjungiamos panaudojant MMPHP programą ir gaunami sugeneruoti RDF duomenys. Paveikslėlyje pavaizduojama sistemos architektūra.



Pav. 12 Sistemos architektūra

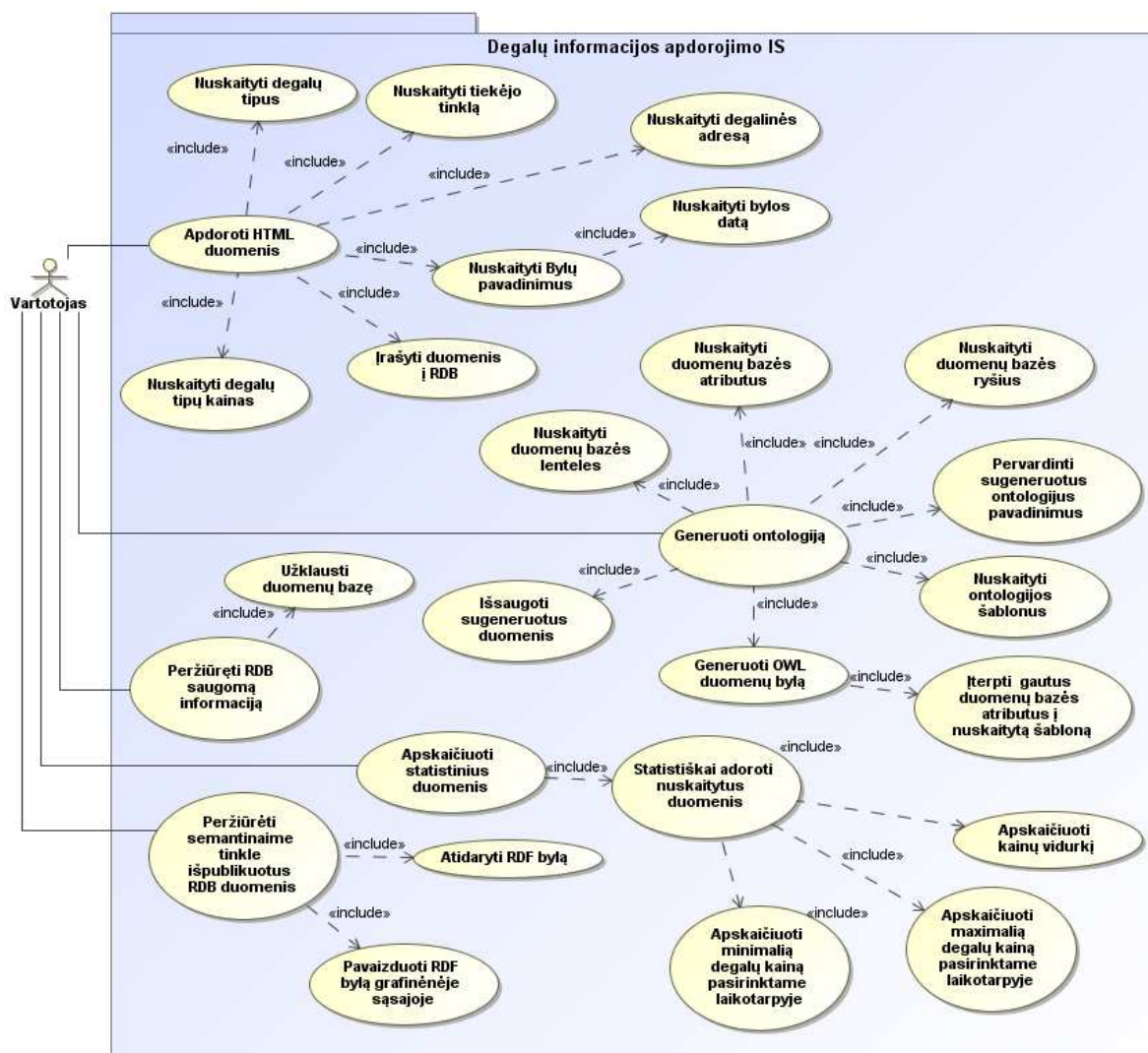
4.1.1. Funkciniai reikalavimai:

Specifikuojami panaudojimo atvejai:

HTML duomenų išsaugojimo programos funkcionalumas nusakomas Pav. 13 paveikslėlyje specifikuotais sistemos panaudojimo atvejais.

Sistemos funkcionalumui realizuoti reikia sukurti komponentus, kurie realizuotų pagrindinius sistemos panaudojimo atvejus:

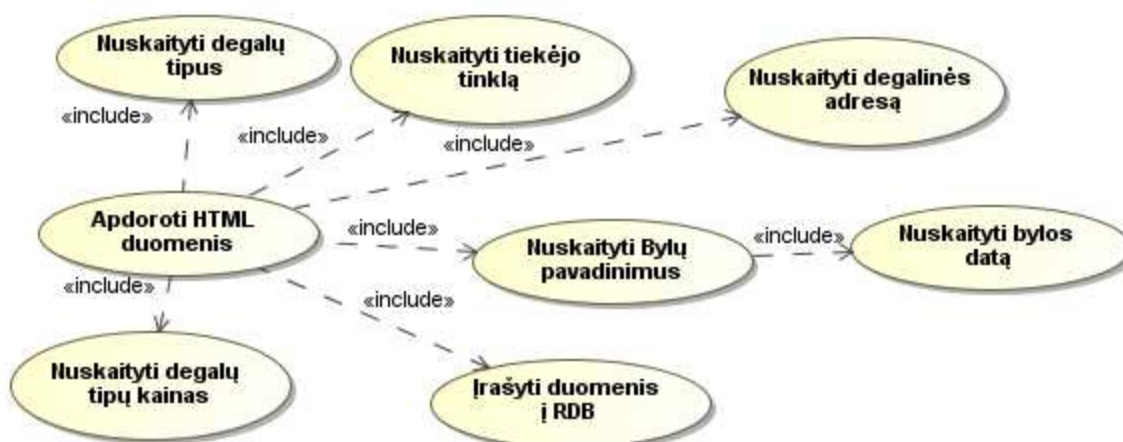
- Apdoroti HTML duomenis
- Peržiūrėti RDB saugomą informaciją
- Generuoti ontologiją
- Apskaičiuoti statistinius duomenis
- Peržiūrėti semantiniame tinkle išpublikuotus RDB duomenis



Pav. 13 Sistemos panaudojimo atvejai

4.1.2. Panaudojimo atvejų aprašas

Panaudojimo atvejis "Apdoroti HTML duomenis"



Pav. 14 Panaudojimo atvejis "Apdoroti HTML duomenis"

1. PANAUDOJIMO ATVEJIS: Apdoroti HTML duomenis

Vartotojas/Aktorius: Sistemos vartotojas

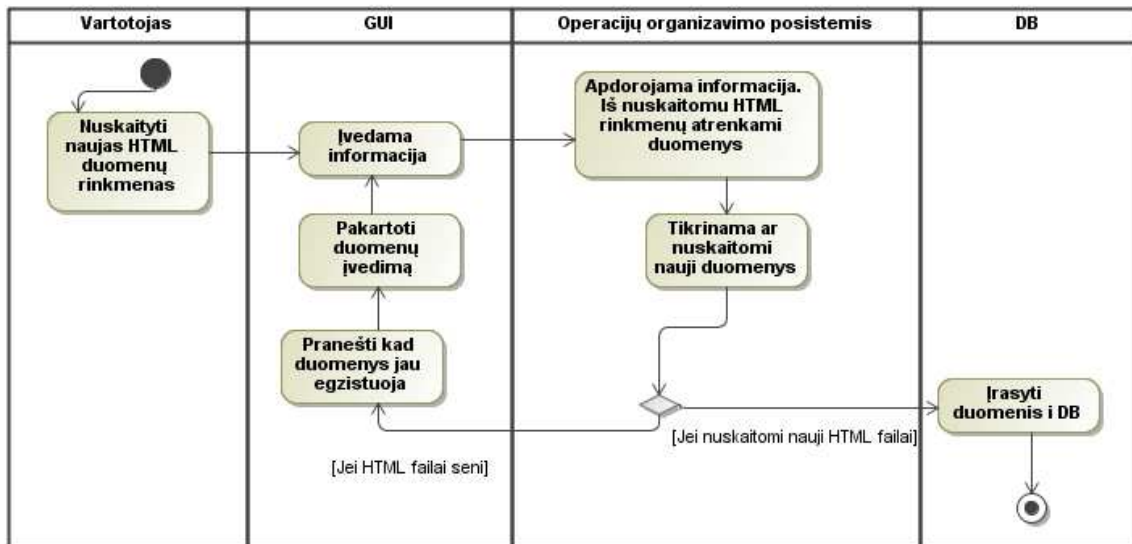
Aprašas: Apima duomenų nuskaitymo procesą

Prieš sąlyga: Duomenys nenuskaityti iš HTML rinkmenų arba nuskaityti nauji duomenys.

Sužadinimo sąlyga: Duomenys nuskaityti pradėjus darbą su programa arba nuskaityti naujus duomenis.

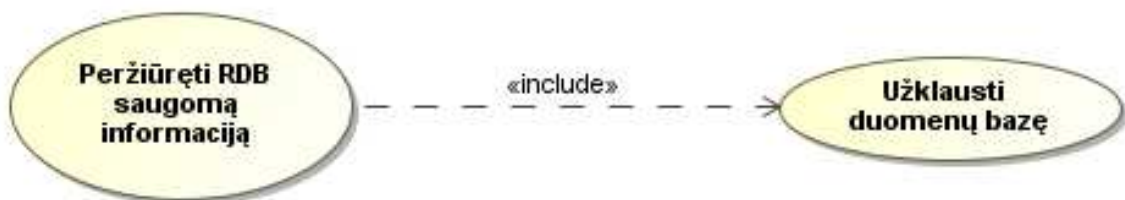
Po-sąlyga: Duomenys nuskaityti ir išsaugoti duomenų bazėje.

Panaudojimo atvejo „HTML rinkmenų nuskaitymas, informacijos įvedimas“ specifikacija.



Pav. 15 HTML rinkmenų nuskaitymas, informacijos įvedimas

Panaudojimo atvejis „Peržiūrėti RDB saugomą informaciją“



Pav. 16 Panaudojimo atvejis "Peržiūrėti RDB saugomą informaciją"

2. PANAUDOJIMO ATVEJIS: Peržiūrėti RDB saugomą informaciją

Vartotojas/Aktorius: Sistemos vartotojas

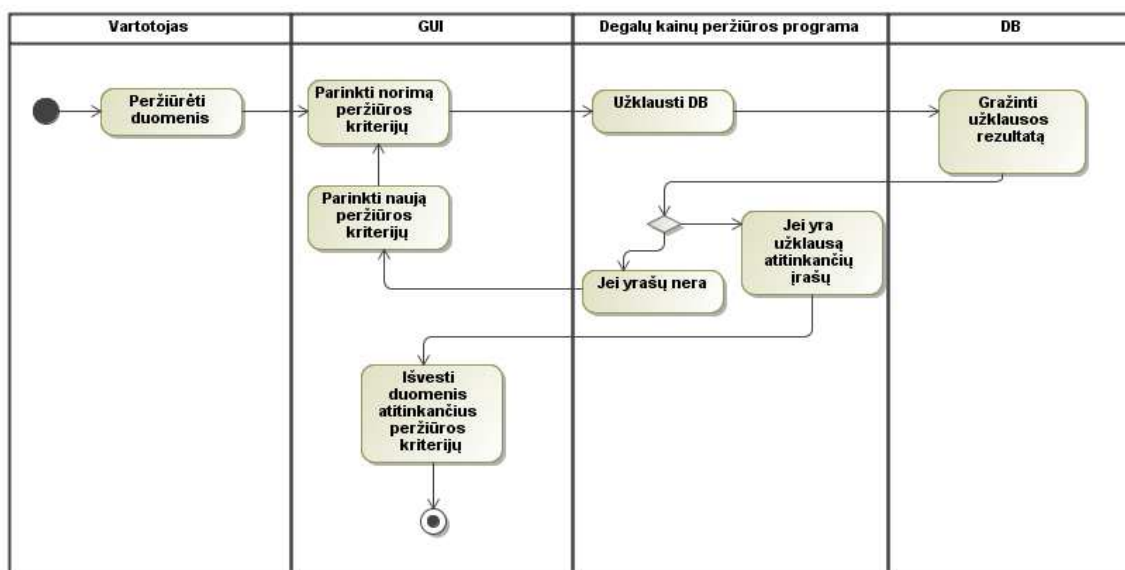
Aprašas: Apima duomenų peržiūros procesą

Prieš sąlyga: Duomenys nuskaityti iš HTML rinkmenų ir patalpinti reliacinėje duomenų bazėje.

Sužadinimo sąlyga: Duomenys išvedami į sistemos grafinę sąsają nuskaičius reliacinės duomenų bazės turinį.

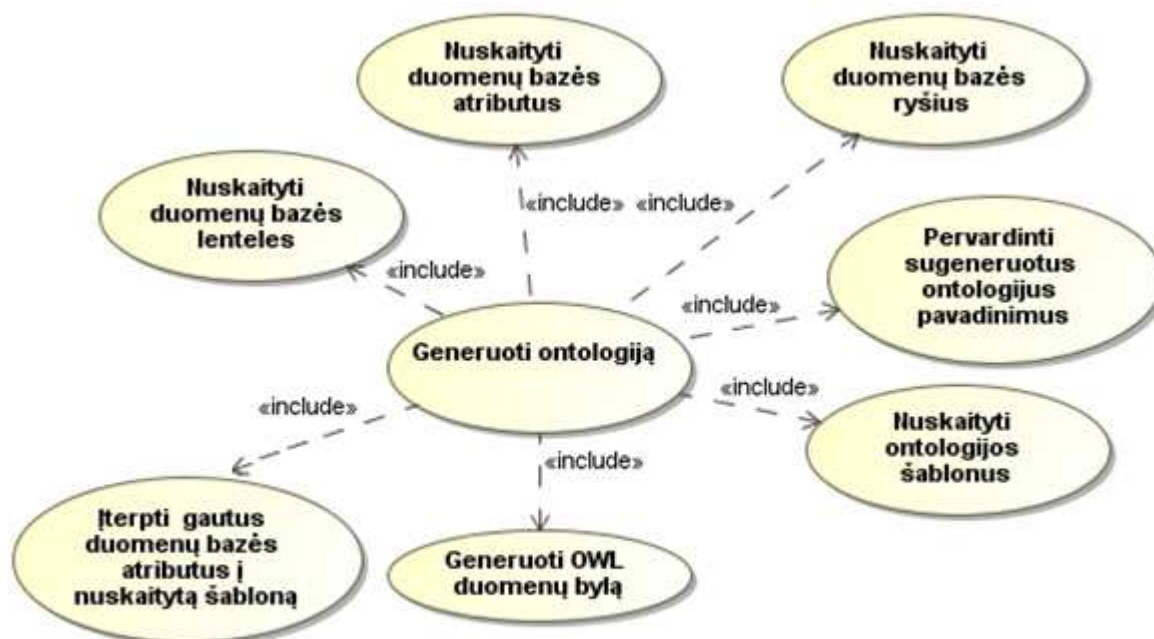
Po-sąlyga: Duomenys išvesti į grafinę vartotojo sąsają ir pateikiami vartotojo peržiūrai

Panaudojimo atvejo „Peržiūrėti RDB saugomą informaciją“ specifikacija.



Pav. 17 Panaudojimo atvejo „Peržiūrėti RDB saugomą informaciją“ specifikacija.

Panaudojimo atvejis „Generuoti ontologiją“



Pav. 18 Panaudojimo atvejis „Generuoti Ontologiją“

3. PANAUDOJIMO ATVEJIS: Panaudojimo atvejis „Generuoti ontologiją“

Vartotojas/Aktorius: Sistemos vartotojas

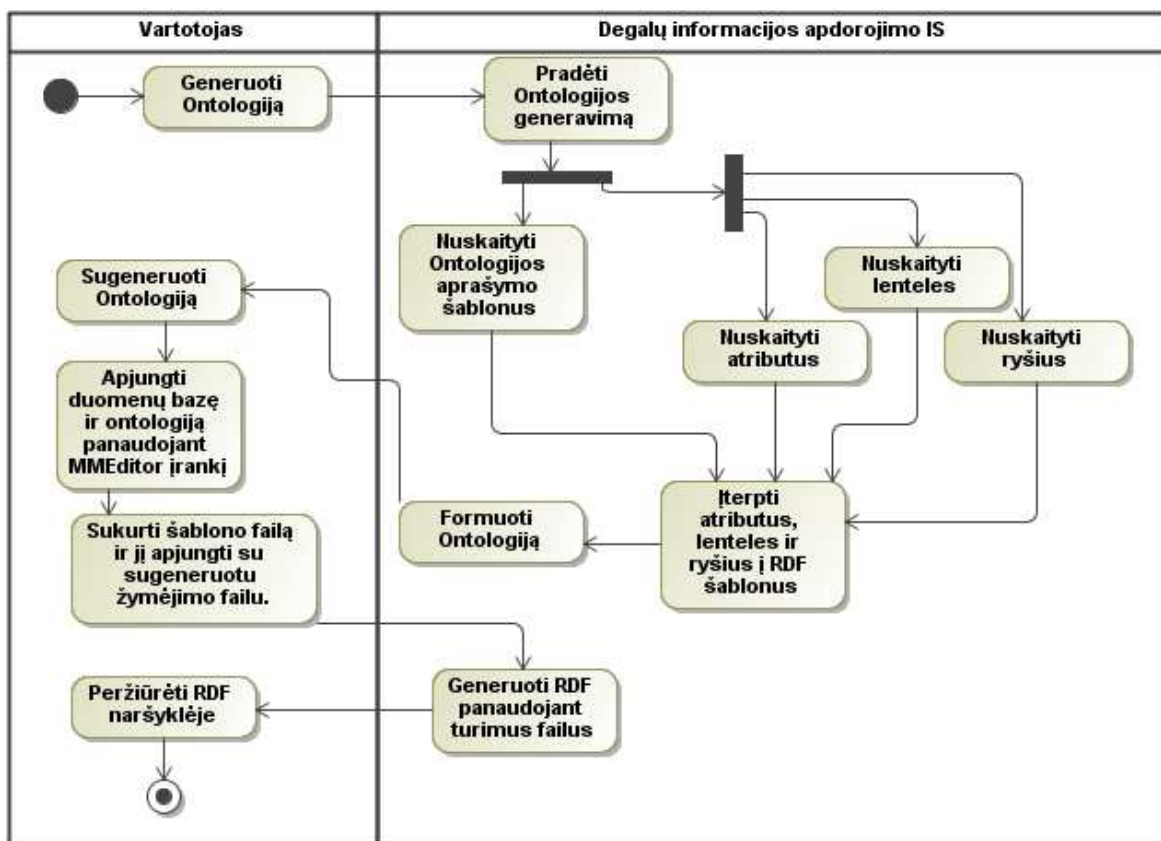
Aprašas: Apima RDF duomenų generavimo procesą.

Prieš sąlyga: Duomenys nuskaityti iš HTML rinkmenų ir patalpinti reliacinėje duomenų bazėje.

Sužadinimo sąlyga: Vartotojas nori sugeneruoti OWL ontologiją.

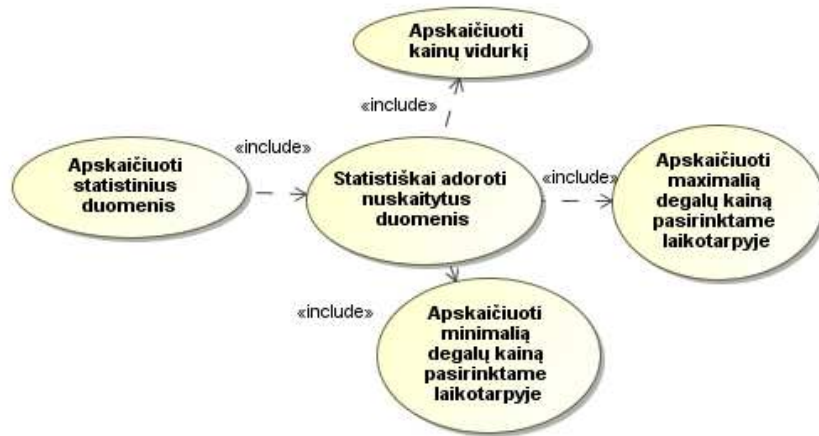
Po-sąlyga: Nuskaityta reliacinė duomenų bazė, sugeneruota OWL ontologija.

Panaudojimo atvejo „Generuoti ontologiją“ specifikacija.



Pav. 19 Panaudojimo atvejo „Generuoti ontologiją“ specifikacija

Panaudojimo atvejis „Apskaičiuoti statistinius duomenis“



Pav. 20 Panaudojimo atvejis „Apskaičiuoti statistinius duomenis“

4. PANAUDOJIMO ATVEJIS: Apskaičiuoti statistinius duomenis

Vartotojas/Aktorius: Sistemos vartotojas

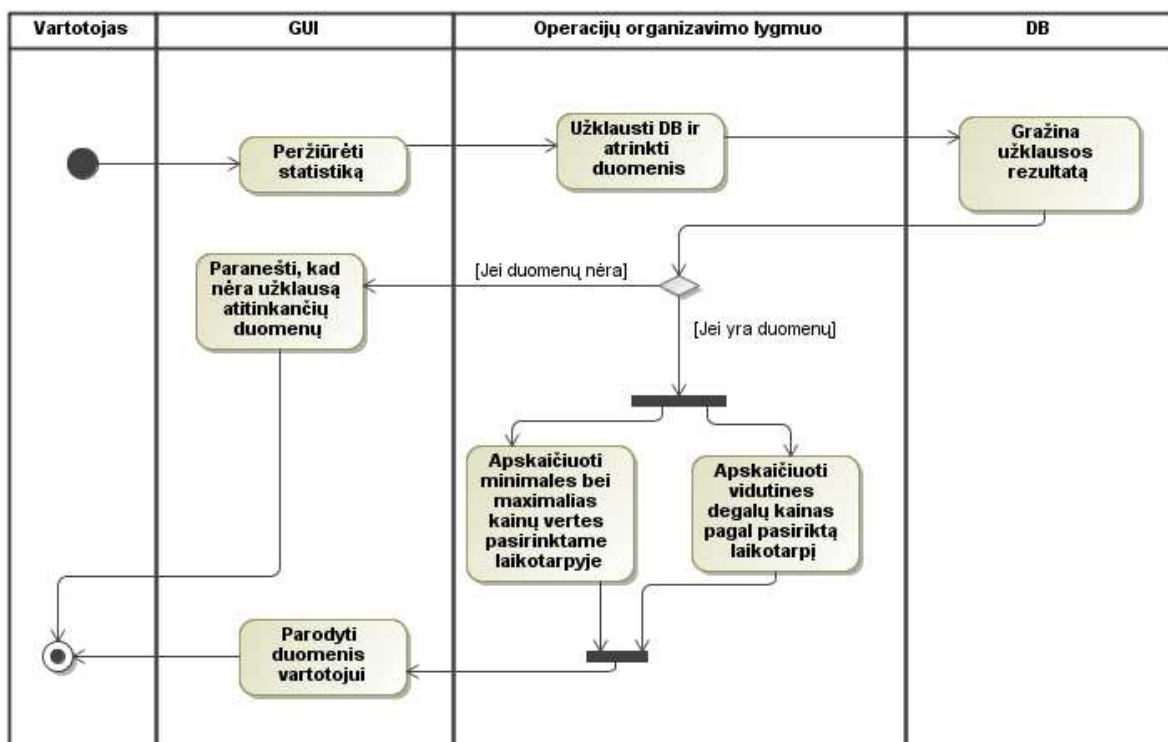
Aprašas: Apima statistikos formavimo procesą.

Prieš sąlyga: Duomenys nuskaityti iš HTML rinkmenų ir patalpinti reliacinėje duomenų bazėje.

Sužadinimo sąlyga: Vartotojas nori peržiūrėti statistinius duomenis

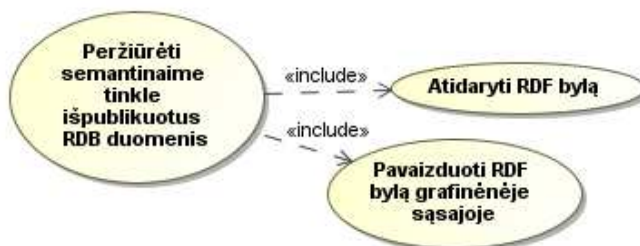
Po-sąlyga: Statistiniai duomenys apskaičiuoti

Panaudojimo atvejo „Apskaičiuotų statistinius duomenis“ specifikacija.



Pav. 21 Panaudojimo atvejo „Apskaičiuotų statistinius duomenis“ specifikacija.

Panaudojimo atvejis „Peržiūrėti semantiniame tinkle išpublikuoti RDF duomenis“



Pav. 22 Panaudojimo atvejis „Peržiūrėti semantiniame tinkle išpublikuoti RDF duomenis“

4. PANAUDOJIMO ATVEJIS: Peržiūrėti semantiniame tinkle išpublikuoti RDF duomenis

Vartotojas/Aktorius: Sistemos vartotojas

Aprašas: Apima RDF duomenų peržiūros procesą

Prieš sąlyga: Sugeneruota RDF duomenų rinkmena

Sužadinimo sąlyga: Vartotojas nori peržiūrėti RDF duomenis, kurie yra išpubliuojami sistemoje

Po-sąlyga: RDF duomenys išpublikuoti

Panaudojimo atvejo „Peržiūrėti semantiniame tinkle išpubliuotus RDB duomenis“ specifikacija.

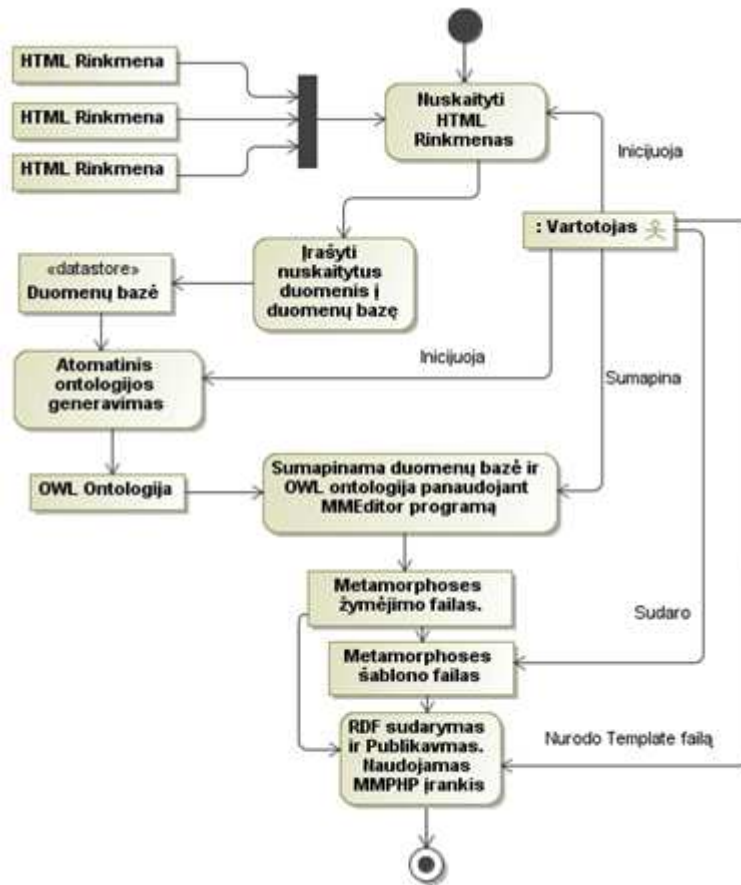


Pav. 23 Panaudojimo atvejo „Peržiūrėti semantiniame tinkle išpublikuotus RDB duomenis“ specifikacija

4.1. Principinė duomenų bazės užpildymo duomenimis bei jos turinio publikavimo schema.

Duomenys yra nuskaityti iš HTML Rinkmenų. HTML rinkmenos – tai degalų tiekėjų svetainių kopijos rinkmenų, kurios buvo reguliariai daromos siekiant kaupti informaciją apie kintančias degalų kainas. Vartotojas inicijuoja šių rinkmenų nuskaitymą taip duomenys perkeliama iš HTML rinkmenų į releacinę duomenų bazę.

Releacinės duomenų bazės turiniui išpublikuoti reikia sudaryti šios duomenų bazės ontologiją. Ontologija yra sudaroma automatiškai. Sugeneruota ontologija turi būti apjungiamas (sumapinama) su releacine duomenų baze. Apjungimui naudojama Metamorphoses MMEditor programa. Apjungus sugeneruojama Metamorphoses žymėjimo rinkmena, kurioje XML kalba aprašyta, kaip ir kokie duomenys yra susiejami. Vartotojas turi pats nurodyti XML kalba aprašytą šablono rinkmeną. Žymėjimo bei šablono rinkmenas galima sudaryti rankiniu būdu arba automatiškai generuoti. Apjungus žymėjimo bei šablono rinkmenas ir nurodžius šablono rinkmeną MMPHP programoje sugeneruojami RDF formatu aprašyti duomenys.



Pav. 24Principinė duomenų bazės užpildymo duomenimis bei jos turinio publikavimo schema.

4.2. Dalykinės srities esybių modelis

Demonstraciniai sistemai sukurti pasirinkta degalinių informacijos dalykinė sritis. Šis dalykinė sritis bus naudojama demonstracijai. Dalykinės srities informacijai kaupti reikia susikurti duomenų bazę, kurios lentelės apibūdina lentelėje Lentelė 7 aprašytos duomenų bazės lentelės bei atributai.

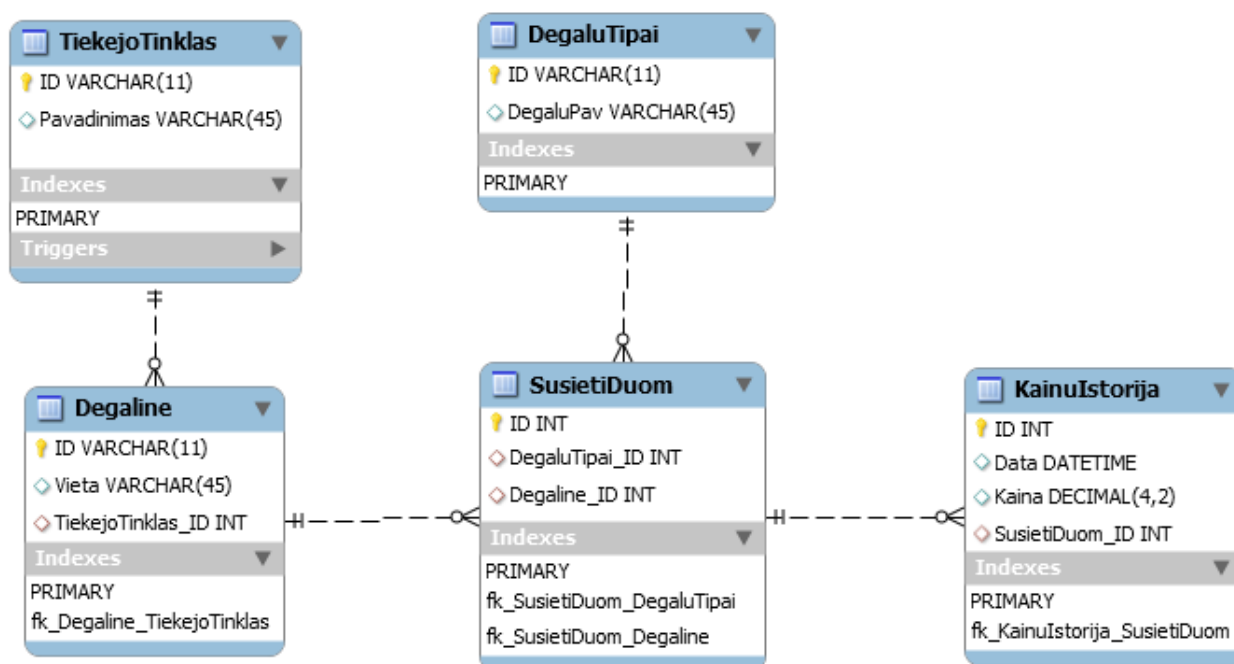
Iš HTML failų nuskaitytiems duomenims saugoti reikalinga duomenų bazė turinti 5 lenteles:

- **TiekėjoTinklas** - aprašomi degalų tiekėjų tinklai (Pvz. Statoil. Orlean, LukOil);
- **DegaluTipai** - aprašomi degalų tipai (Pvz. 95,98, D);
- **Degalinė** - aprašomas degalinės kuro tiekėjo tinklas bei vieta kur yra pastatyta degalinė;
- **SusietiDuom** - lentelė naudojama duomenų susiejimui;
- **KainuIstorija** - Saugoma degalų kainų istorija.

Lentelė 7 Duomenų bazės struktūra

Lentelės pavadinimas	Lauko pavadinimas	Lauko tipas	Apibūdinimas
TiekėjoTinklas	ID	INT	Tiekėjo tinklo identifikatorius
	Pavadinimas	VARCHAR	Tiekėjo pavadinimas
Degalinė	ID	INT	Degalinės identifikatorius
	TiekėjoTinklas_ID	INT	TiekėjoTinklas lentelės identifikatorius
	Vieta	VARCHAR	Vieta kur yra degalinė
SusietiDuom	ID	INT	Duomenų susiejimo identifikatorius
	Degalinė_ID	INT	Lentelės Degalinė identifikatorius
	DegaluTipai_ID	INT	Lentelės DegaluTipai identifikatorius
KainuIstorija	ID	INT	Kainų istorijos identifikatorius
	SusietiDuom_ID	INT	SusietiDuom lentelės identifikatorius
	Data	DATE	Data
	Kaina	DECIMAL	Kaina
DegaluTipai	ID	INT	Degalu tipo identifikatorius
	DegaluPav	VARCHAR	Degalu pavadinimas

Duomenų bazėje saugomas tiekėjo tinklas, tiekėjo tinklui priklausančios degalinės ir degalinių degalų tipai. Skirtingi duomenys susiejami naudojant lentelę „*SusietiDuom*“. Taip pat yra kaupiama kainų istorija.



Pav. 25 Dalykinę sritį atitinkanti klasių diagrama

4.3. **Nefunkciniai reikalavimai**

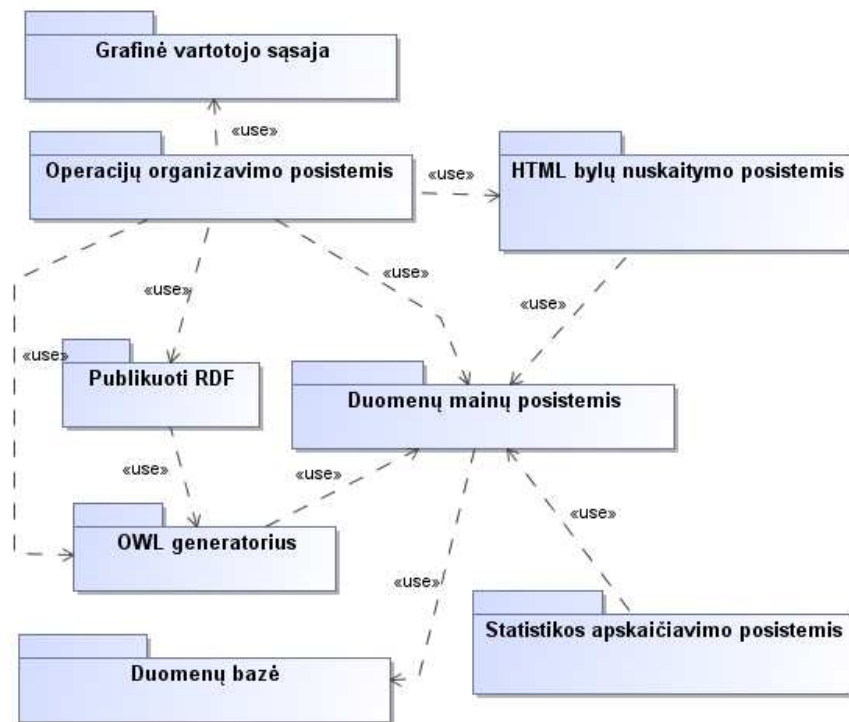
Sistemos kūrimui naudojamos C# kalba visual studio arba SharpDevelop aplinkoje, duomenų publikavimui naudojama PHP kalba ir šia kalba realizuota MMPHP programa, duomenims laikyti naudojama MySQL duomenų bazė. Sąsajai tarp C# visual studio aplinkos ir MySQL duomenų sudaryt i naudojamas MySQL connector bei Apache serveris.

5. Sistemos projektas

5.1. Sistemos architektūra

Projektuojama sistema susideda iš šių pagrindinių komponentų:

- Grafinė vartotojo sąsaja;
- Operacijų organizavimo posistemis;
- Duomenų mainų posistemis;
- Duomenų bazė;
- HTML bylų nuskaitymo posistemis;
- RDF generavimo posistemis;
- Statistikos apskaičiavimo posistemis;



Pav. 26 Sistemos architektūra

Grafinės vartotojo sąsajos komponentas perduoda vartotojo vykdomas operacijas operacijų organizavimo posistemiiui, kuriame yra organizuojama visa sistemos veikla. Čia išskviečiami metodai skirti duomenų nuskaitymui iš HTML bylų, gautų duomenų įrašymui į

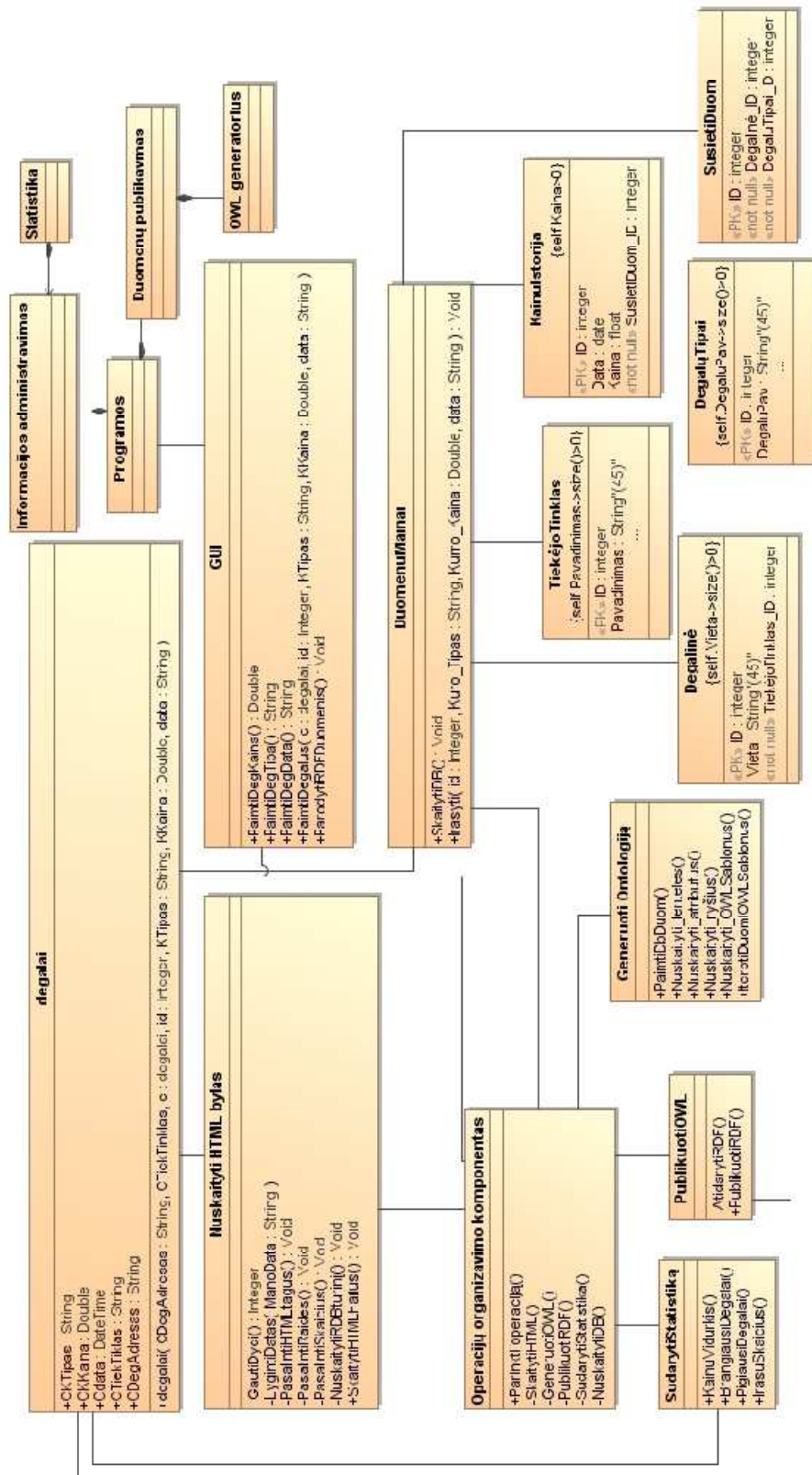
duomenų bazę. Pastaroji operacija atliekama naudojant duomenų mainų posistemį. Taip pat naudojamas OWL duomenų generavimo posistemis, kuriame yra atliekamas duomenų ir duomenų bazės schemas nuskaitymas iš RDB ir OWL dokumento generavimas. Naudodami RDF publikavimo posistemį galime išpublikuoti sugeneruotus RDF duomenis semantiniame tinkle. Statistikos paskaičiavimo posistemis skirtas apskaičiuoti ir suformuoti statistinius duomenis panaudojant reliacinėje duomenų bazėje saugomus duomenis.

5.2. Projektavimo etapo klasės

Degalų informacijai nuskaityti bei apdoroti reikia sukurti tai galinčią atlikti sistemą.

Sistemai realizuoti reikalingos klasės:

- ***Operaciju_Organizavimo***komponentas – tai pagrindinė klasė, kurioje bus atliekamas atliekamų operacijų organizavimas;
- Klasė ***degalai*** – tai klasė skirta kaupti duomenims, ir perduoti duomenis sąrašo pavidalu tarp kitų komponentų;
- ***Operaciju_Organizavimo***komponentas – tai pagrindinė klasė, kurioje bus atliekamas atliekamų operacijų organizavimas;
- Operacijų klasės:
 - ***NuskaitytiHTMLbylas***– klasė skirta HTML duomenims nuskaityti;
 - ***SudarytiStatistiką***– Statistikos sudarimo klasė;
 - ***PublikuotiRDF***- RDF publikavimo klasė;
 - ***GeneruotiRDF***– RDF generavimo klasė;
 - ***DuomenuMainai***– klasė skirta duomenų mainams tarp aplikacijos ir RDB;
- Duomenų bazės klasės:
 - ***Degalinė***;
 - ***SusietiDuom***;
 - ***TiekejoTinklas***;
 - ***DegaluTipai***;
 - ***KainuIstorija***;



Pav. 27Sistemos klasių diagrama

5.3. Panaudojimo atvejus realizuojančios klasės

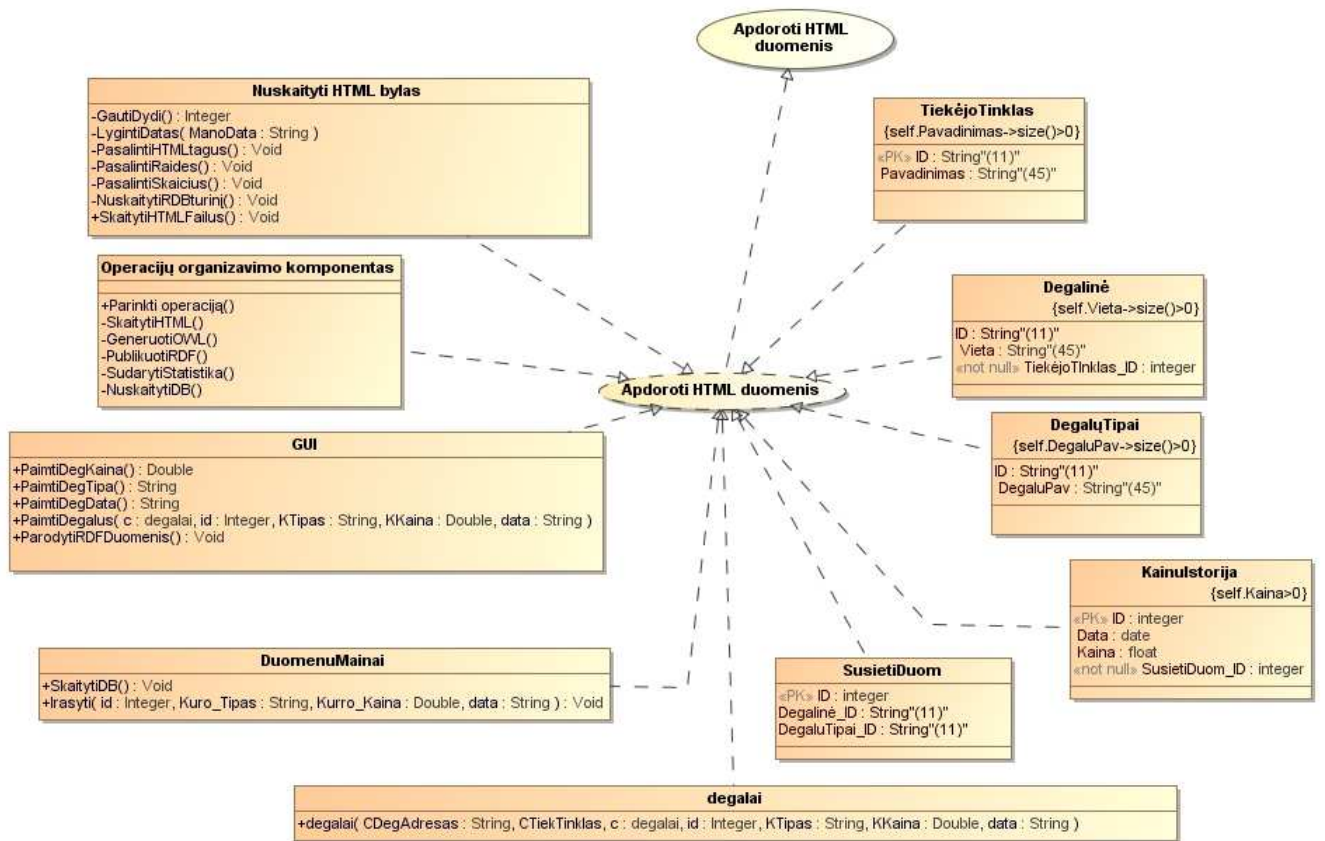
Kiekvienam, pagal sistemos reikalavimus aprašytam, panaudojimo atvejui turime sukurti klases, kurios realizuos panaudojimo atvejuose numatytą sistemos funkcionalumą.

- Apdoroti HTML duomenis;
- Apskaičiuoti statistinius duomenis;
- Peržiūrėti RDB saugomą informaciją;
- Peržiūrėti semantiniame tinkle išpublikuotus duomenis;
- Generuoti Ontologiją;

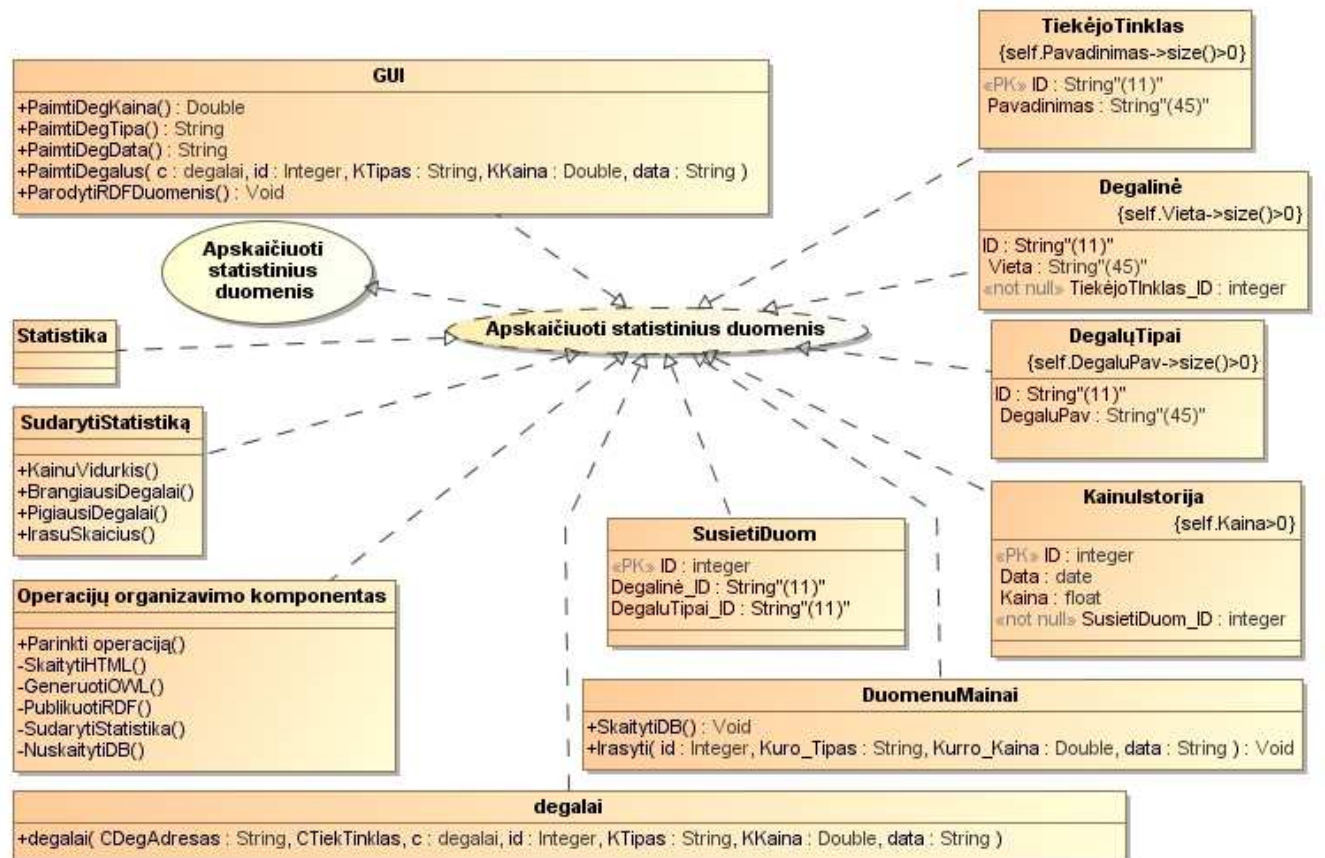
Šie panaudojimo atvejai realizuojami panaudojant sistemos klases, kurios yra aprašytos skyrelyje projektavimo etapo klasės.

Operacijų metu vyksta informacijos mainai tarp sistemos ir DB. Naudojamos DB lentelės:

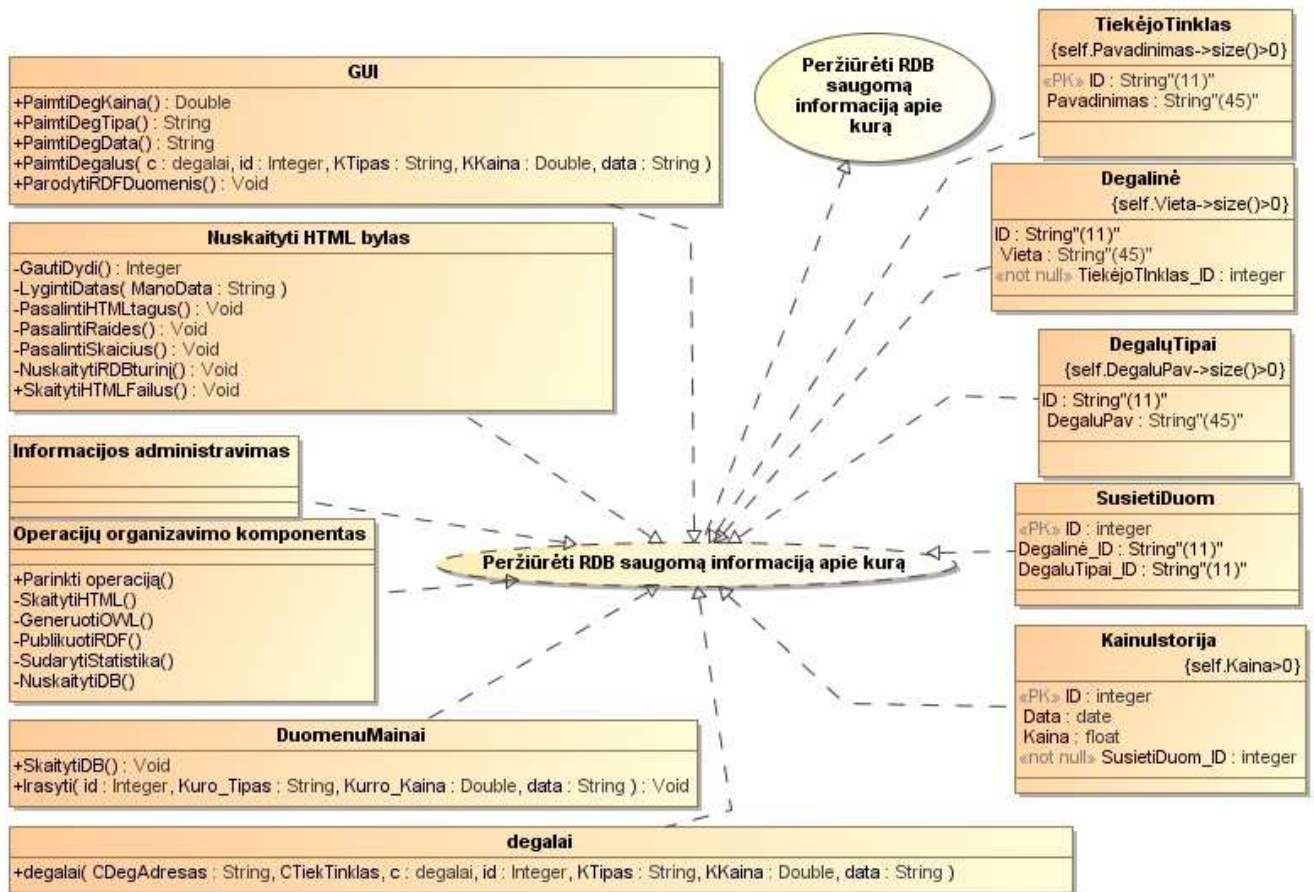
- Degalinė,
- Tiekėjo tinklas,
- Kainųistorija,
- DegalųTipai,
- SusietiDuom;



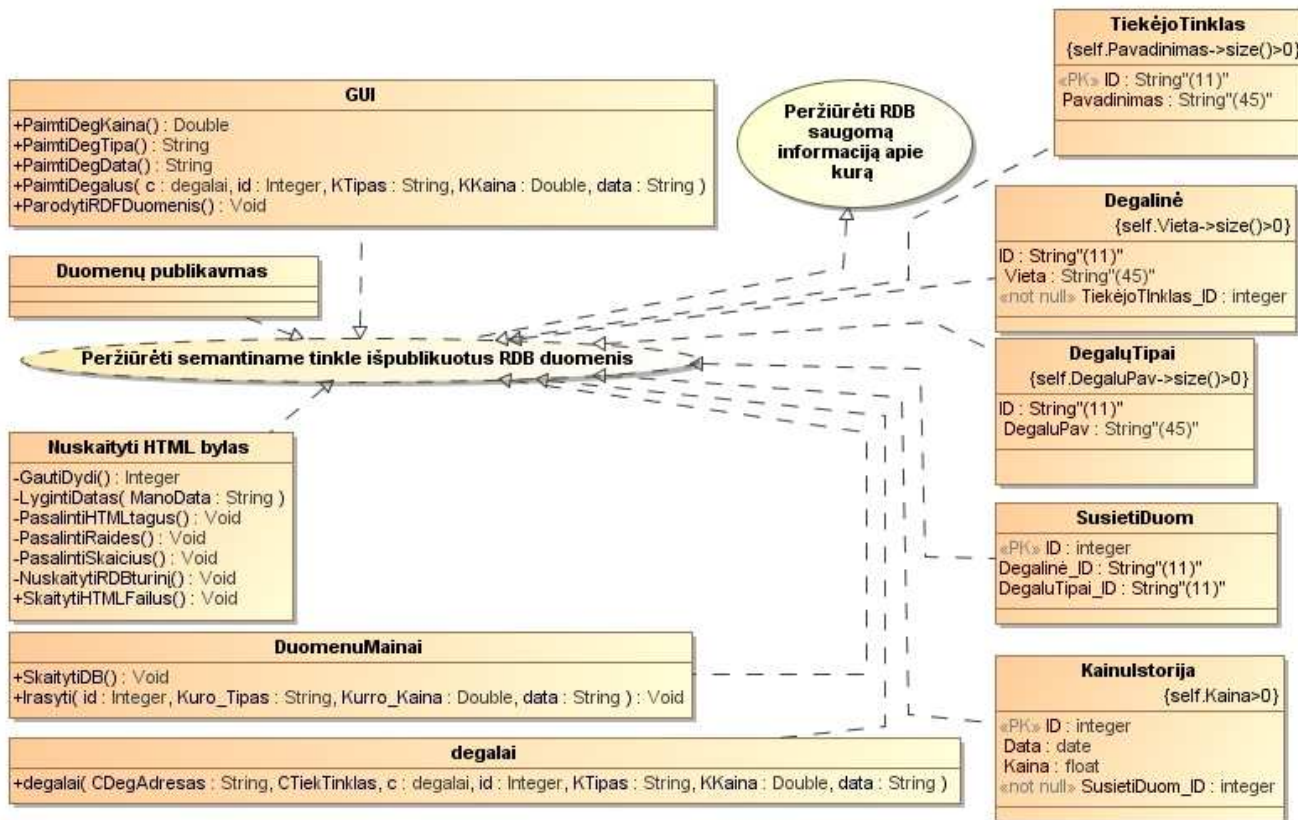
Pav. 28 Panaudojimo atvejį "Apsdoroti HTML duomenis" realizuojančios klasės



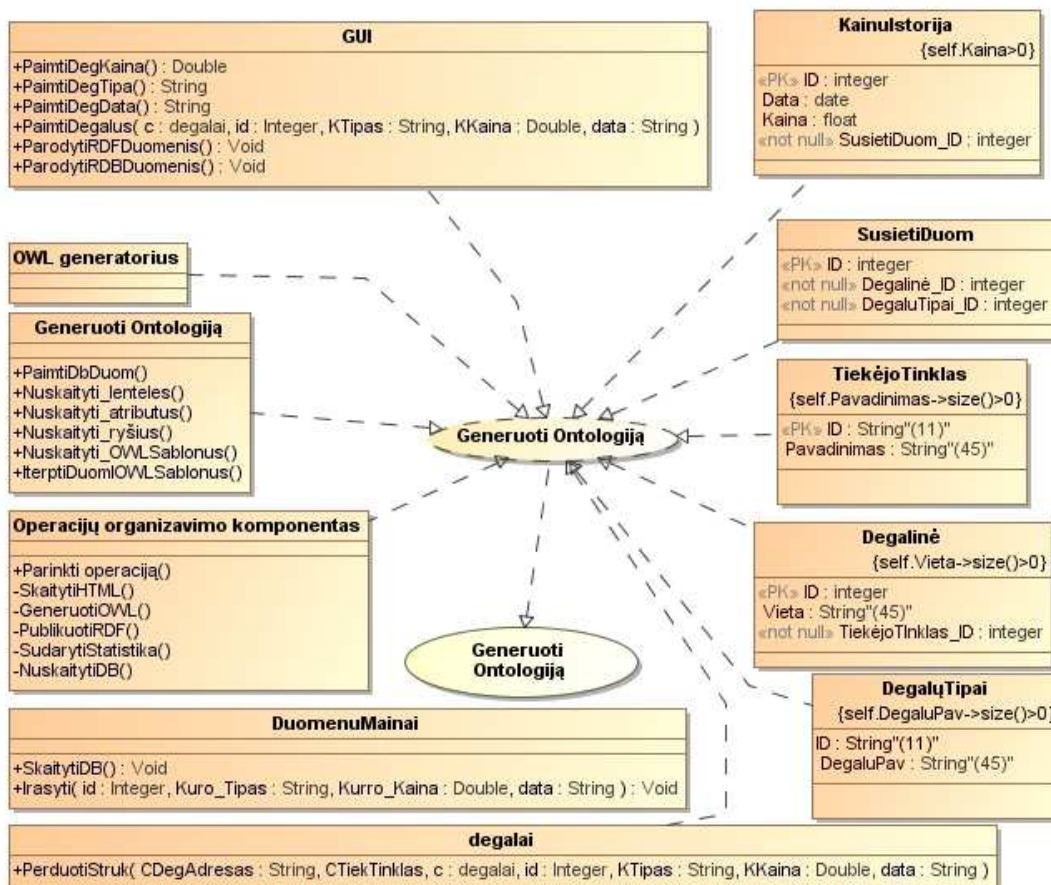
Pav. 29 Panaudojimo atvejį "Apskaičiuoti statistinius duomenis" realizuojančios klasės



Pav. 30 Panaudojimo atvejį "Peržiūrėti RDB saugomą informaciją apie kurą" realizuojančios klasės



Pav. 31 Panaudojimo atveji "Peržiūrėti semantiniame tinkle išpublikuotus RDB duomenis" realizuojančios klasės



Pav. 32 Panaudojimo atveji "Peržiūrėti semantiniame tinkle išpublikuotus RDB duomenis" realizuojančios klasės

5.4. Panaudojimo atvejus detalizuojančios sekų diagramos

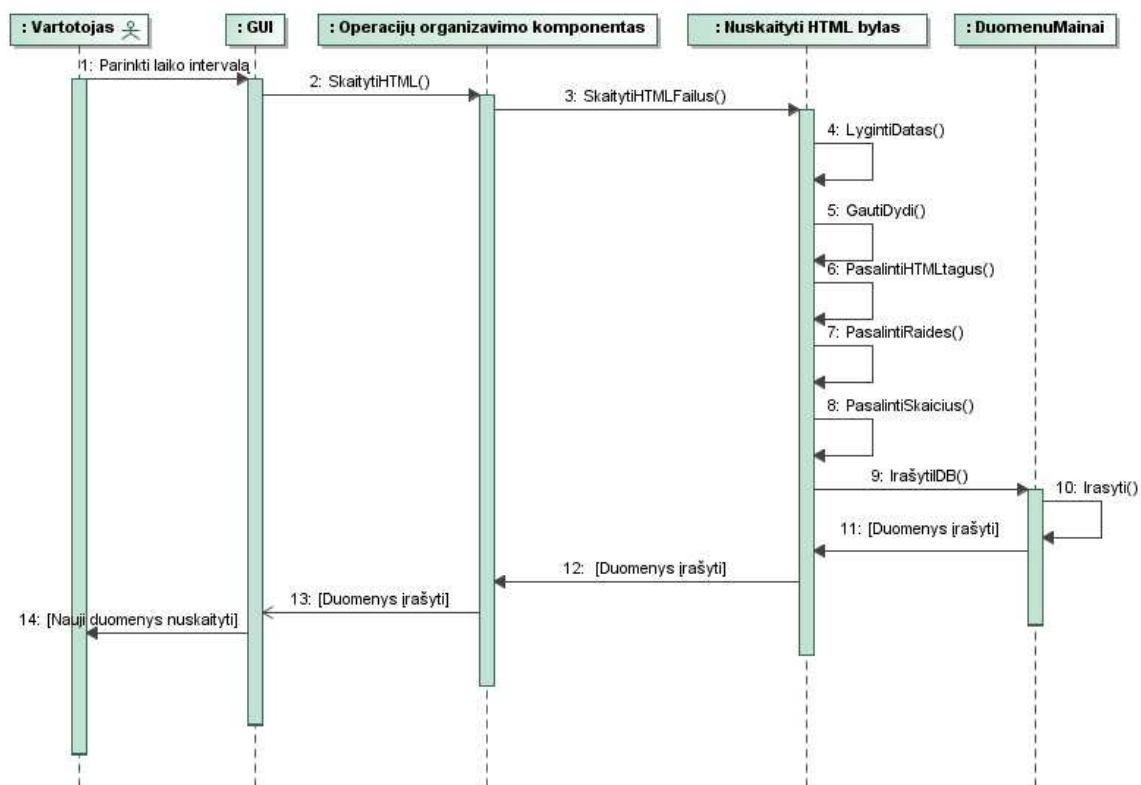
Sistemos funkcionalumui aprašyti bei detalizuoti naudojamos sekų diagramos.

Panaudojimo atvejo “Apdoroti HTML duomenis“ detalizavimas:

Vartotojas nuskaitydamas naują informaciją pasirenka norimą laiko intervalą kuriame bus rodomi duomenys. Nuskaitant HTML failus reikia atrinkti reikiamas eilutes ir iš šių eilučių atrinkti tik tas, kuriose yra saugoma reikalinga informacija apie degalus:

- Degalų kaina;
- Degalų tipas;
- Tiekėjo tinklas;
- Nuskaitymo data;
- Degalinės adresas;

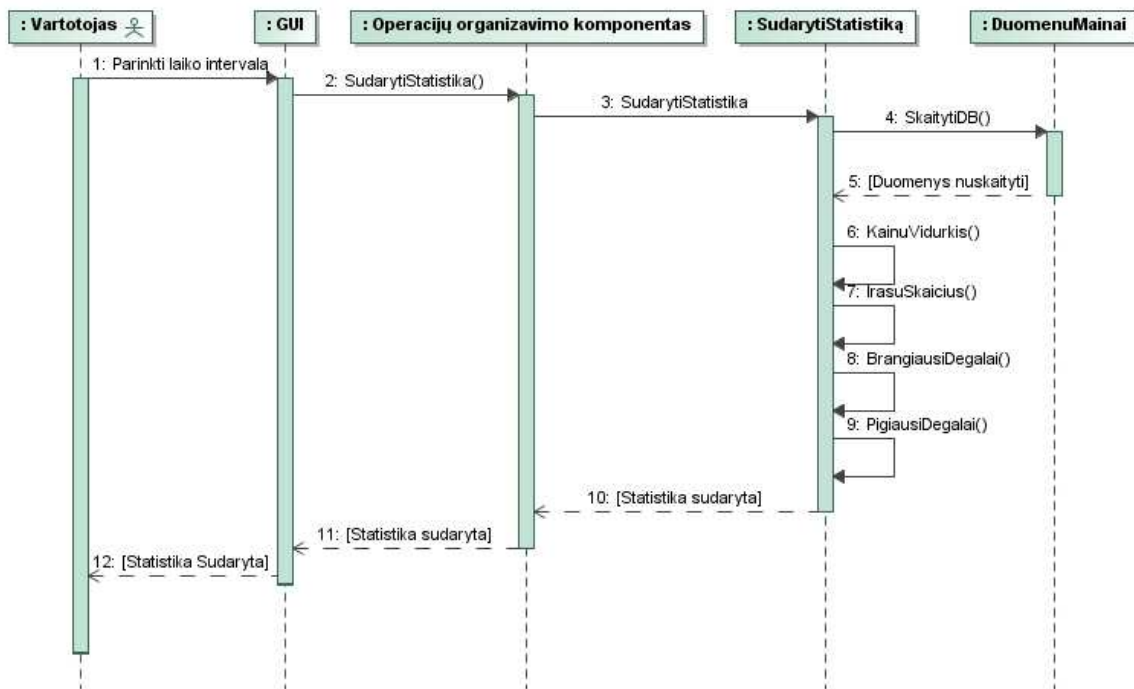
Iš atrinktų eilučių yra atrenkami tik reikalingi duomenys ir yra pašalinamos nereikalingos HTML kalbos žymos: nereikalingos raidės, skaičiai ir tarpai tarp žodžių bei ženklų;



Pav. 33 Panaudojimo atvejį "Apdoroti HTML duomenis" detalizuojanti sekų diagrama

Panaudojimo atvejo "Apskaičiuoti statistinius duomenis" detalizavimas:

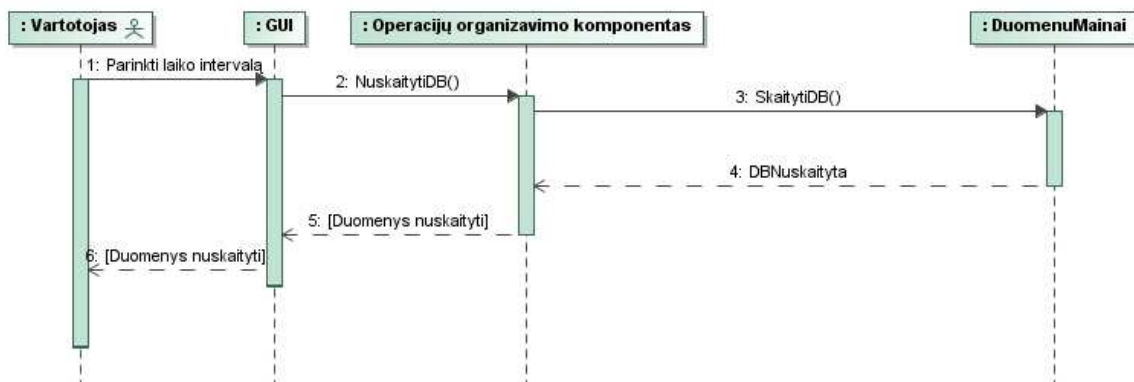
Nuskaitomas RDB turinys, ir apskaičiuojami reikalingi statistiniai parametrai. Tam yra panaudojami sistemoje aprašyti metodai. Apskaičiavus pigiausias, brangiausias degalus bei kainų vidurkį vartotojui išvedama statistika.



Pav. 34 Panaudojimo atvejį "Apskaičiuoti statistinius duomenis" detalizuojanti sekų diagrama

Panaudojimo atvejo "Peržiūrėti RDB saugomą informaciją apie kurą" detalizavimas:

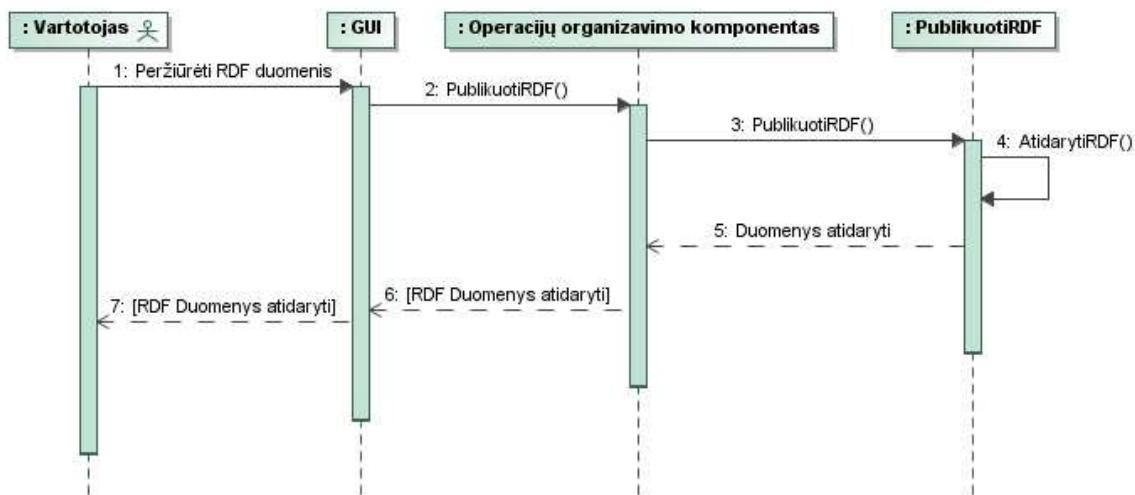
Nuskaitant sukauptą informaciją reikia nuskaityti duomenų bazės turinį ir šį turinį perduoti vartotojo sąsajai. Atvaizduojamą turinį gali peržiūrėti vartotojas.



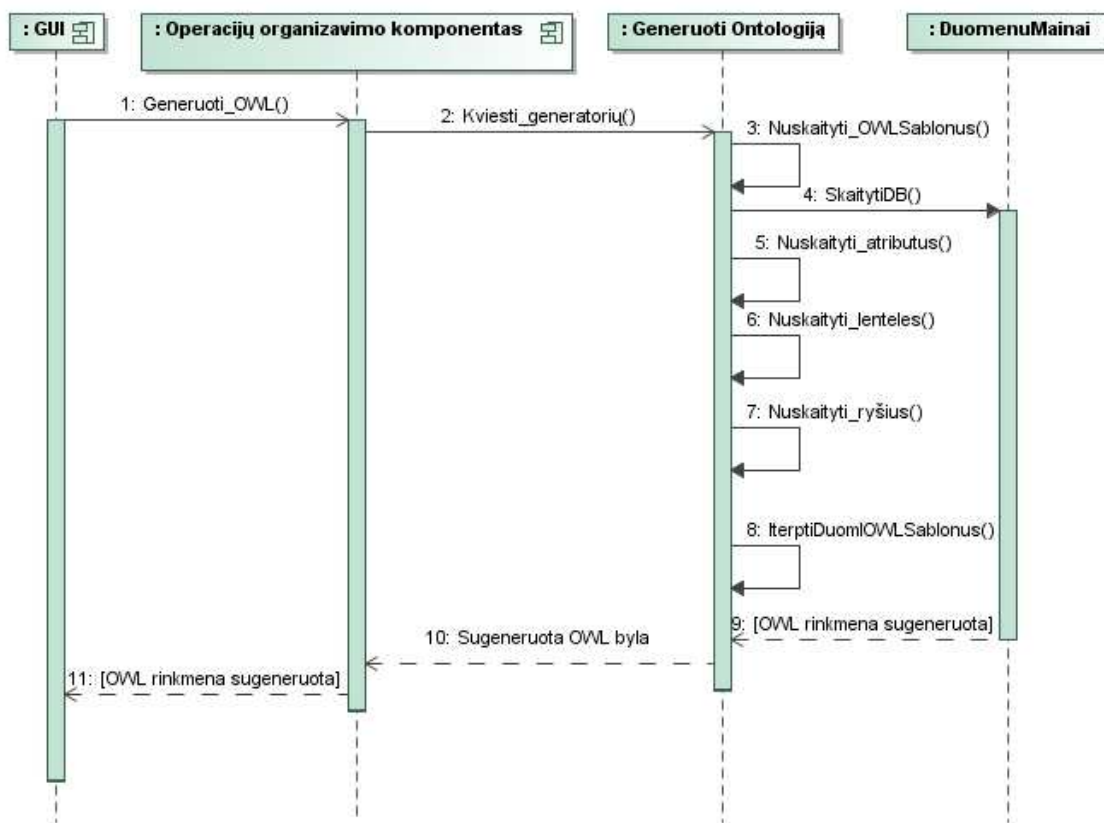
Pav. 35 Panaudojimo atvejį "Peržiūrėti RDB saugomą informaciją apie kurą" detalizuojanti sekų diagrama

Panaudojimo atvejo "Peržiūrėti RDB saugomą informaciją apie kurą" detalizavimas:

Informacijos pateikimui semantinio tinklo RDF duomenimis reikalinga nuskaityti duomenų bazės turinį. Vėliau duomenų bazės turinio aprašymo šabloną sugeneruojamas RDF failas kuris gali būti pateikiamas vartotojui kaip semantinio tinklo duomenys;



Pav. 36 Panaudojimo atveji "Peržiūrėti semantiniame tinkle išpublikuotus duomenis" detalizuojanti sekų diagrama



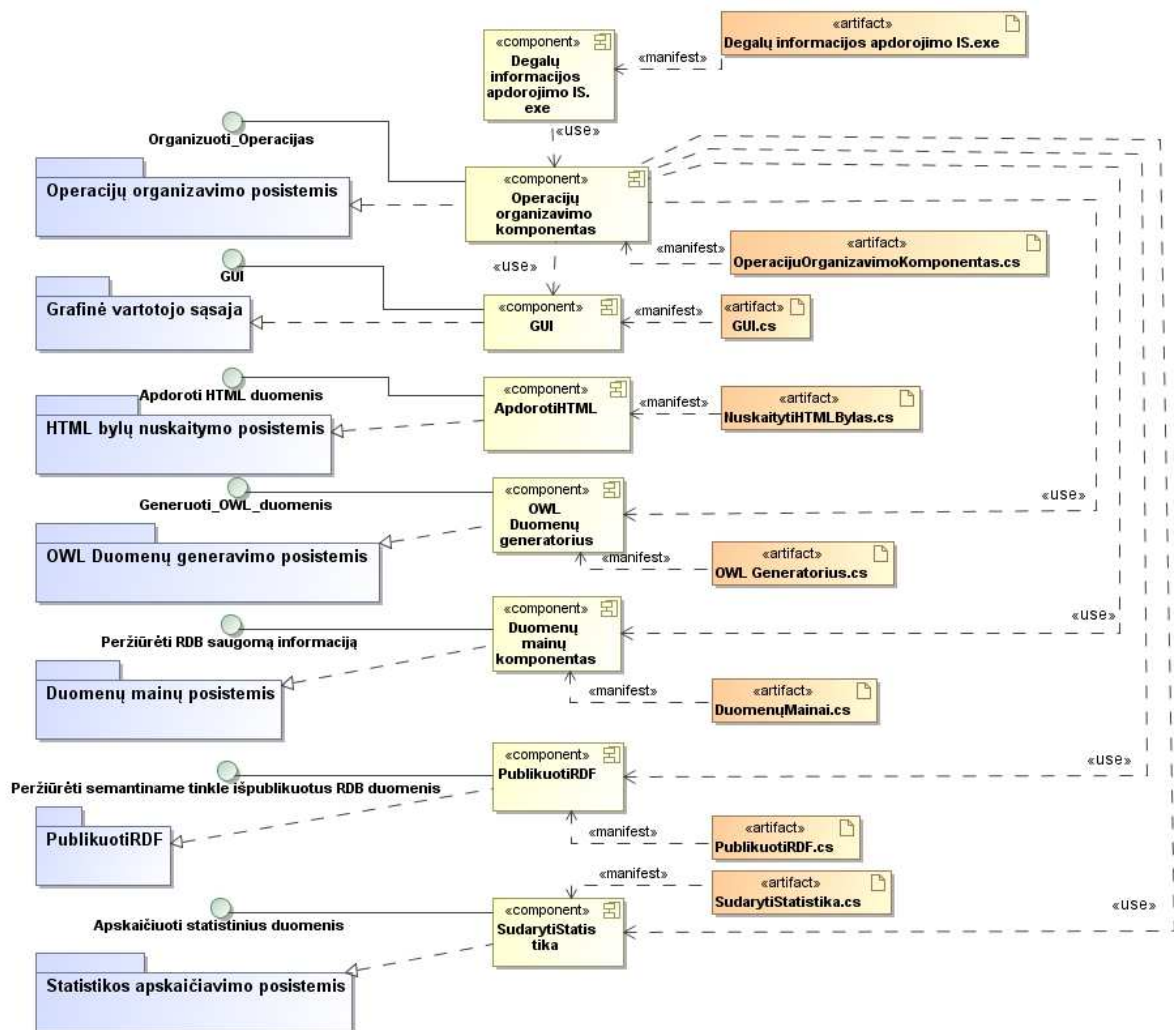
Pav. 37 Panaudojimo atveji "Generuoti RDF duomenis" detalizuojanti sekų diagrama

5.5. Sistemos realizacijos modelis

Sistema realizuojama naudojant 7 pagrindinius komponentus:

- Sudaryti statistiką;
- GUI;
- ApdorotiHTML;
- OWL Duomenų generatorius;
- Duomenų mainų komponentas;
- PublikuotiRDF;

Šie komponentai yra naudojami *Operacijų organizavimo komponento*, kuriame yra valdomas komponentų iškvietimas pagal vykdomas operacijas. Operacijų organizavimo komponentas yra naudojamas kaip aukčiausio lygmens valdiklis ir yra naudojamas *Degalų apdorojimo informacijos IS.exe* komponento.



Pav. 38 Sistemos realizacijos modelis;

6. Sprendimo realizacija

6.1. Reliacinė duomenų bazė

Ontologijai sudaryti pasirinkta duomenų bazė, turinti 3000 įrašų, sudaryta iš 5 lentelių. Panaudojama duomenų bazė aprašyta skyriuje 4.1.

6.2. HTML rinkmenų nuskaitymas

```
Regex g = new Regex(@"<tr><th nowrap="); Regex adresas = new Regex(@"Adresas:");
int ind = 0;
System.IO.DirectoryInfo dir = new System.IO.DirectoryInfo(@"C:\Data\");
foreach (System.IO.DirectoryInfo BylosPav in dir.GetDirectories())
{
    //Nuskaityti reikšmes
    Console.WriteLine(BylosPav.FullName);
    string[] filePaths = Directory.GetFiles(BylosPav.FullName);
    int failuskaicius = filePaths.Length;
    for (int fi = 0; fi < failuskaicius; fi++)
    {
        using (StreamReader r = new StreamReader(filePaths[fi]))
        {
            string line="";
            string adresline;
            string Kuro_Tipas="";
            double Kuro_Kaina = 0;
            string Degadresas = "";
            Degalai Dg = null;
            List<Degalai> DgList = new List<Degalai>();
            while ((line = r.ReadLine()) != null)
            {
                Match m = g.Match(line);
                if (m.Success)
                {
                    Dg = new Degalai();
                    string html = line;
                    string input = HtmlRemoval.StripTagsRegex(html);

                    Kuro_Tipas = ReiksmeSutampa(input);
                    string trim = Regex.Replace(input, @"\s", "");
                    int ilgis = trim.Length;

                    string beraidziu = PasalintiRaides.StripTagsRegex(trim);
                    beraidziu = Pasalintiskaicius.StripTagsRegex(beraidziu);
                    beraidziu = beraidziu.Replace(".", "");
                    Kuro_Kaina = double.Parse(beraidziu);

                    Dg.CKKaina = Kuro_Kaina; // Set field
                    Dg.CKTipas = Kuro_Tipas;
                    Dg.Cdata = BylosPav.Name;
                    DgList.Add(Dg);
                }
                adresline = line;

                Match adrM = adresas.Match(adresline);
                if (adrM.Success)
                {
                    adresline = r.ReadLine();
                    Degadresas = HtmlRemoval.StripTagsRegex(adresline);
                    Degadresas = Regex.Replace(Degadresas, @"\s\s", "");
                    Console.WriteLine("Degadresas: " + Degadresas);
                }
            }
        }
        foreach(Degalai d in DgList){
            d.Cadresas = Degadresas;
        }
        DgLists.Add(DgList);
    }
}
}
```

HTML rinkmenoms nuskaityti nuskaitomas visas direktorijų bei rinkmenų medis nurodytame aplanke. Panaudojant *regex* užklaudas reikalinga informacija yra atrenkama ir apdorojama pašalinant nereikalingus tusčius tarpus ir kitus simbolius. Atrinkta informacija talpinama į objektų sąrašą ir gali būti panaudota tolimesniam apdorojimui.

6.3. Ontologijos generatorius

Ontologijai sudaryti reikia nusiskaityti duomenų bazės schema. Schemoje yra saugomos duomenų bazės lentelės, lentelės sudarantys atributai, atributų duomenų tipai, bei ryšiai, nusakantys kaip skirtingos lentelės yra susijusios tarpusavyje.

Ontologijai sudaryti kuriamas Ontologijos generatorius. Generatoriui sukurti panaudota .NET Framework aplinka. Duomenų bazė laikoma lokaliajame MySQL duomenų bazėje ir Apache serveryje.

Nuskaityta duomenų bazės schema, .NET aplinkoje, talpinama į duomenų struktūras. Šios struktūros yra panaudojamos generuojant RDF/XML ontologiją. Generavimui naudojami iš anksto paruošti ontologijos šablonai.

Nuskaitytų lentelių pavadinimai naudojami užpildant šablonus, nusakančius duomenų bazės lenteles. Kiekvienai lentelei sukuriamas atskira klase. Reikšmės „*LentelesVardas*“ pakeičiamos nuskaitytos lentelės vardu.

```
<owl:Class rdf:ID="LentelesVardas">  
<rdfs:isDefinedBy rdf:resource="" />  
<rdfs:label>LentelesVardas</rdfs:label>  
<rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class" />  
</owl:Class>
```

Lentelių atributams aprašyti naudojamas *owl:DatatypeProperty* duomenų tipas. Šiame duomenų tipe nurodomas atributo vardas, lentelė, kuriai šis atributas priklauso bei Atributo tipas.

```
<owl:DatatypeProperty rdf:ID="AtributoVardas">  
<rdfs:range  
rdf:resource="http://www.w3.org/2001/XMLSchema#AtributoTipas" />  
<rdfs:isDefinedBy rdf:resource="" />  
<rdfs:domain rdf:resource="#LentelesVardas" />  
<rdfs:label>AtributoVardas</rdfs:label>  
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-  
ns#Property" />  
</owl:DatatypeProperty>
```

Lentelės ryšiams užpildyti naudojamas owl:ObjectProperty ontologijos duomenų tipas. Šablonas užpildomas įrašant tėvinės klasės, kuriai priklauso, vardą nusakant domeną bei sritį.

```
<owl:ObjectProperty rdf:ID="TevoVardas">
  <rdfs:domain rdf:resource="#Domenas"/>
  <rdfs:range rdf:resource="#Sritis"/>
</owl:ObjectProperty>
```

Klasėms, ryšiams bei atributams aprašyti panaudojami Class, ObjectProperty bei DatatypeProperty duomenų tipai aprašyti oficialioje OWL ontologijos specifikacijoje <http://www.w3.org/TR/owl-guide/>.

Naudojant šiuos tris ontologijos duomenų tipus nusakanti šablonus sudaroma duomenų bazės ontologija:

Generuojamų ontologijos elementų pavyzdžiai:

Lentelės atitinkančios klasės:

```
<owl:Class rdf:ID="DEGALINE">
<rdfs:isDefinedBy rdf:resource="" />
<rdfs:label>DEGALINE</rdfs:label>
<rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class" />
</owl:Class>
```

```
<owl:Class rdf:ID="DEGALUTIPAI">
<rdfs:isDefinedBy rdf:resource="" />
<rdfs:label>DEGALUTIPAI</rdfs:label>
<rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class" />
</owl:Class>
```

Lentelės atributus atitinkančios ontologijos struktūros:

```
<owl:DatatypeProperty rdf:ID="ID">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#INT32" />
<rdfs:isDefinedBy rdf:resource="" />
<rdfs:domain rdf:resource="#DEGALINE" />
<rdfs:label>ID</rdfs:label>
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:ID="VIETA">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#STRING" />
<rdfs:isDefinedBy rdf:resource="" />
<rdfs:domain rdf:resource="#DEGALINE" />
<rdfs:label>VIETA</rdfs:label>
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
</owl:DatatypeProperty>
```

```
<owl:ObjectProperty rdf:ID="TIEKEJOTINKLAS_ID">
  <rdfs:domain rdf:resource="#TIEKEJOTINKLAS"/>
  <rdfs:range rdf:resource="#DEGALINE"/>
</owl:ObjectProperty>
```

```

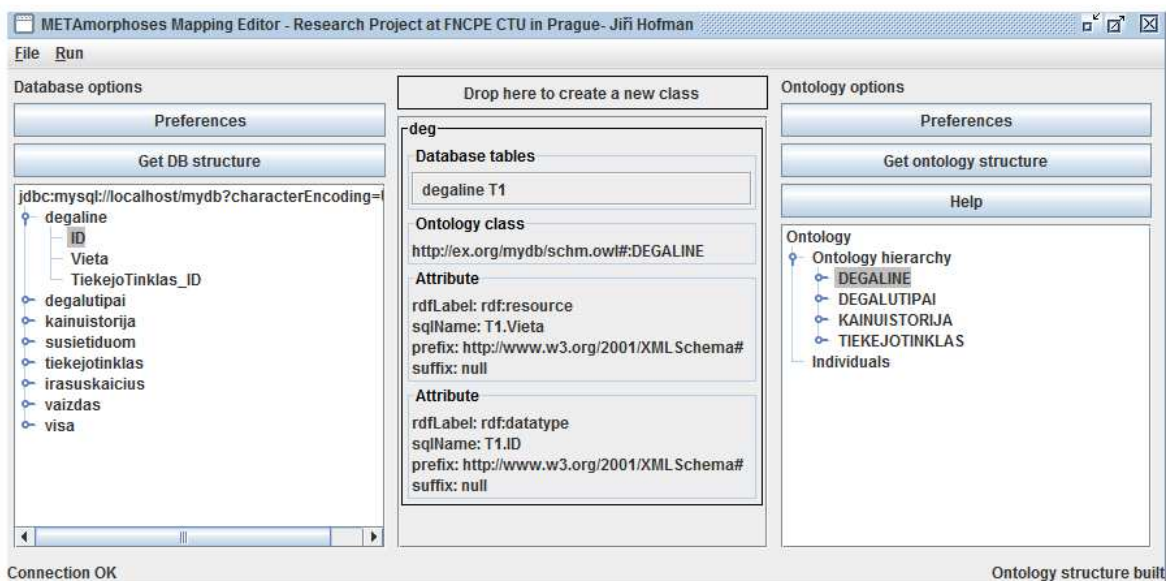
<owl:ObjectProperty rdf:ID="DEGALUTIPAI_ID">
    <rdfs:domain rdf:resource="#DEGALUTIPAI" />
    <rdfs:range rdf:resource="#DEGALINE" />
</owl:ObjectProperty>

```

6.4. Duomenų Publikavimas

Sudaryta ontologija panaudojama žymėjimo rinkmenoms sudaryti, kurie yra naudojami METAmorphoses processor ir kitų METAmorphoses įrankių. Panaudojus šiuos įrankius bei sudarytas žymėjimo rinkmenas duomenų bazė išpublikuojama i RDF rinkmeną.

Žymėjimo rinkmenoms sudaryti naudojamas METAmorphoses MappingEditor įrankis. Šis įrankis padeda susieti reliacinę duomenų bazę bei šios duomenų bazės ontologiją. Susiejus ontologiją su reliacine duomenų baze gaunama žymėjimo rinkmena.



Pav. 39 METAmorphoses MappingEditor

Duomenims išpubliuoti galima panaudoti MMPHP, RDF-Shout arba METAmorphoses procesor programas. Šios programos RDF duomenims generuoti naudoja žymėjimo bei šablono rinkmenas.

6.5. Grafinės vartoto sąsajos modelis:

6.5.1. GUI puslapis: Pasirinkimo meniu:

Šiame GUI puslapyje pateikiamas meniu norimai operacijai pasirinkti. Galima rinktis „**Informacijos administravimas**“ (Iškviečiama Degalų duomenų registravimo programa) arba „**Duomenų publikavimas**“ (Iškviečiamas ontologijos generavimo meniu).



Pav. 40 Pasirinkimo meniu

6.5.2. GUI puslapis: Degalų duomenų registravimo programa:

Šiame vartotojo sąsajos puslapyje galima nuskaityti duomenų failus, redaguoti duomenų įrašus, juos šalinti arba pridėti naujus. Taip pat iš čia prieinamas ataskaitų meniu, kuriame pateikiama informacija apie duomenų bazėje saugomą informaciją.

Vartotojui paspaudus mygtuką „*Nuskaityti Rinkmenas*“ sistema iš nurodytų HTML failų nuskaitys duomenis.

Vartotojui paspaudus mygtuką „*Skaityti*“, nuskaito duomenų bazės turinį ir išveda DataGrid komponente.

Vartotojui paspaudus mygtuką „*Irašyti*“, įrašomas naujas įrašas į duomenų bazę.

Vartotojui paspaudus mygtuką „*Redaguoti*“ aktyvuojami neaktyvūs mygtukai „**Trinti**“ – ištrinamas DataGrid komponente parinktas įrašas, „**Atnaujinti**“ – Atnaujinamas DataGrid komponente parinktas įrašas, „**Atšaukti**“ – atšaukiamas duomenų redagavimas.

Nuspaudus mygtuką „**Ataskaitos**“, iškviečiamas ataskaitų meniu.

Degalus galima filtruoti pagal degalų tipą. Srityje „**Ieškoti pagal pavadinimą**“ įvedamas ieškomas degalų tipas ir nuspaudžiamas Mygtukas „**Paieška**“.

ID	Tiekėjas	Adresas	Degalų Pavadinimas	Kaina
1	Lukoil	Šilutes pl. 96, Klai...	95	4,36
2	Lukoil	Šilutes pl. 96, Klai...	VERVA 95	4,46
3	Lukoil	Šilutes pl. 96, Klai...	D	3,00
4	Lukoil	Šilutes pl. 96, Klai...	VERVA D	4,08
5	Lukoil	Šilutes pl. 96, Klai...	LPG DUJOS	2,39
6	Lukoil	Šilutes pl. 96, Klai...	ADBLUE	1,99
7	Lukoil	Skuodo g. 14, M...	95	4,38
8	Lukoil	Skuodo g. 14, M...	VERVA 95	4,48
9	Lukoil	Skuodo g. 14, M...	98	4,52
10	Lukoil	Skuodo g. 14, M...	D	4,01
11	Lukoil	Skuodo g. 14, M...	VERVA D	4,11
12	Lukoil	Skuodo g. 14, M...	LPG DUJOS	2,34
13	Lukoil	Žemaitijos g. 42, ...	95	4,38
14	Lukoil	Žemaitijos g. 42, ...	VERVA 95	4,48
15	Lukoil	Žemaitijos g. 42, ...	98	4,52

Pav. 41 GUI puslapis: Kainų peržiūra

6.5.3. GUI puslapis: Informacija semantiniame tinkle:

Šiame lange išpublikuojami iš HTML tinklapio kopijų nuskaityti duomenys. Duomenys yra saugomi duomenų bazėje ir yra išpublikuojami šiame grafinės sąsajos puslapyje. Duomenims išpublikuoti sudaroma (nuspaudus mygtuką „Generuoti“) duomenų bazės ontologija. Kuomet yra generuojama puslapio ontologija, programoje galima parinkti duomenų bazės lenteles (*Visos Lentelės*), kurios yra generuojamos. Sarašuose - Generuojamos lentelės, atributai, ryšiai – galima keisti lentelės pavadinimą, atributų bei ryšių pavadinimus. Pakeistosios reikšmės generuojamos ir ontologijos rinkmenoje.

Visos Lentelės

- R: degaline
- R: degalutipai
- R: dtipai
- R: irasuskaicius
- R: kainuistorija
- R: prisijungimas
- R: susietiduum
- R: tiekėjotinklas
- R: vaizdas
- R: visa

Generuojamos lentelės

- 0: degaline
- 1: degalutipai
- 2: dtipai
- 3: irasuskaicius
- 4: kainuistorija
- 5: prisijungimas
- 6: susietiduum
- 7: tiekėjotinklas
- 8: vaizdas
- 9: visa

Atributai

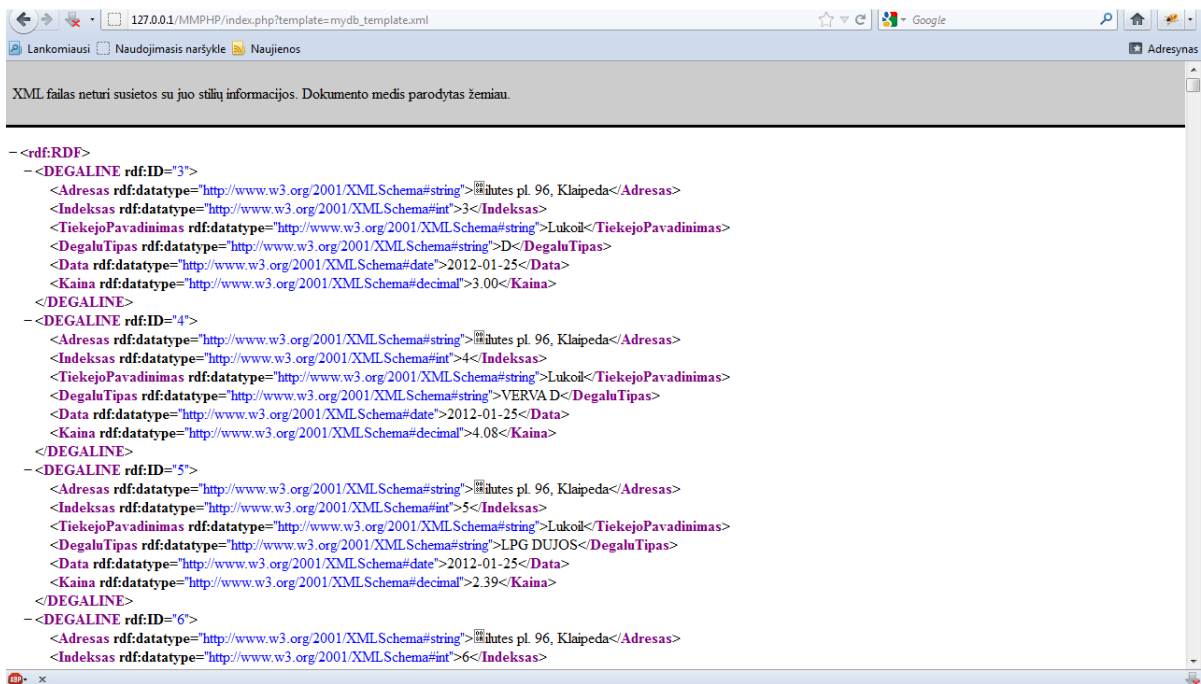
Ryšiai

Tipas

Pavadinimas

Ontologija sugeneruota

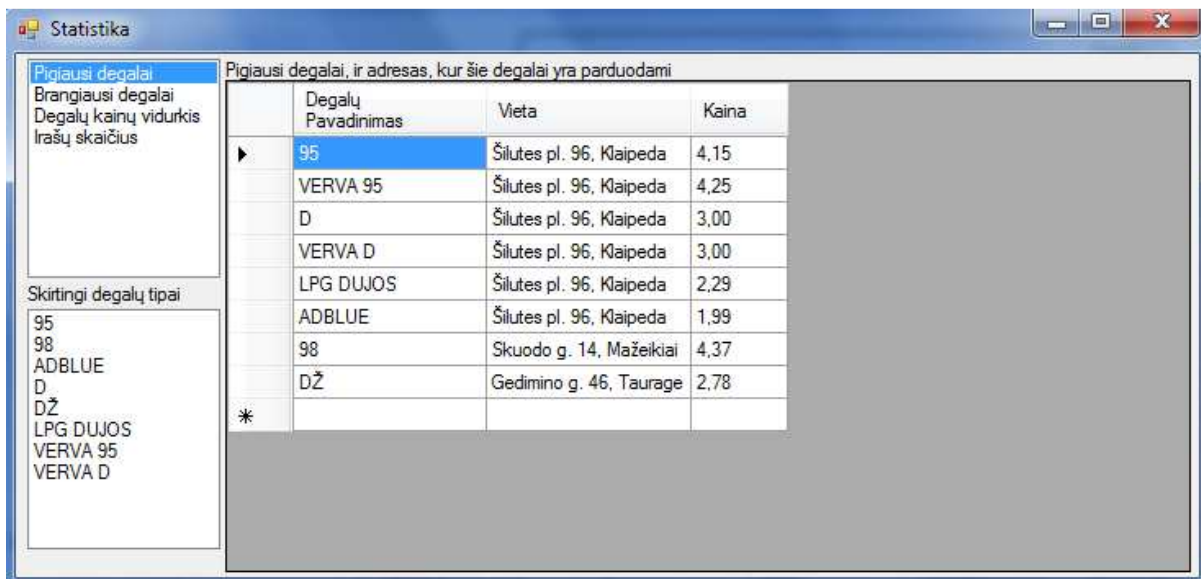
Pav. 42 GUI, ontologijos generavimas



Pav. 43 GUI puslapis: Informacija semantiniame tinkle

6.5.4. GUI puslapis: Statistika:

Šiame grafinės sąsajos puslapyje yra pateikiama degalų kainų statistika. Išvedami skirtingi degalų tipai saugomi duomenų bazėje, apskaičiuojami pigiausi bei brangiausi degalai, surandamas kiekvieno tipo degalų kainų vidurkis, pateikiamas kiekvieno degalų tipo įrašų skaičius.



Pav. 44 GUI puslapis: Statistika

7. Eksperimentinis sistemos tyrimas.

7.1. Eksperimento planas.

Eksperimentui atlikti reikia:

- 1) Užpildyti duomenų bazę duomenimis.
- 2) Iš duomenų bazės sugeneruoti ontologiją.
- 3) Apjungti ontologiją bei duomenų bazę ir sugeneruoti Metamorphoses žymėjimo rinkmeną.
- 4) Sukurti Metamorphoses šablono rinkmeną.
- 5) Panaudoti Metamorphoses MMPHP tinklinę aplikaciją ir apjungti šablono bei žymėjimo rinkmenas.
- 6) Palyginti sugeneruotą RDF rinkmeną su RDF pavyzdžiu, kuris pateikiamas kitame lape.
- 7) Aptarti duomenų nuskaitymo iš HTML rinkmenų trukmę, OWL rinkmenos generavimo trukmę, RDF generavimo trukmę.

Kokybės kriterijai:

- 1) Ontologijoje turi būti generuojami dalykinės srities duomenų bazėje esantys lentelių, lentelių ryšių bei atributų pavadinimai.
- 2) Ontologijoje lentelės turi būti aprašomos *owl:Class* savybe.
- 3) Ontologijoje ryšiai turi būti aprašomos *owl:ObjectProperty* savybe.
- 4) Ontologijoje atributai turi būti aprašomi *owl:DatatypeProperty* savybe.
- 5) Sugeneruota RDF rinkmena turi tenkinti RDF dokumentu reikalavimus bei struktūrą. Rinkmeną turi būti galima atidaryti su Protege (ar kitu RDF redaguojančiu) įrankiu.

7.2.2. Duomenų bazė

Šiame skyriuje pateikiamas duomenų bazės lentelių turinys po duomenų importavimo. Pateikiamos lentelės tiekėjų tinklas, susietiduum, degalutipai, degaline, kainuistorija ir apibendrinta lentelė su visais duomenimis.

Lentelė 8 Turinys lentelėje Tiekėjų tinklas

ID	Pavadinimas
t1	Orlean

Lentelė 9 Turinys lentelėje Degalutipai

ID	DegaluPav
dt1	95
dt2	VERVA 95
dt3	D
dt4	VERVA D
dt5	LPG DUJOS
dt6	ADBLUE
dt7	98

Lentelė 10 Turinys lentelėje degaline

ID	Vieta	TiekėjoTinklas_ID
d1	Šilutes pl. 96, Klaipėda	t1
d2	Vilimiškes k., Vydmantai, Kretingos r., Klaipėda - Liepoja 24 km.	t1
d3	Dirvupiuk., Gargždu r.	t1

Lentelė 11 Turinys lentelėje kainų istorija

ID	Data	Kaina	SusietiDuom_ID
1	2012-05-20	4.36	1
2	2012-05-20	4.46	2
3	2012-05-20	3.00	3
4	2012-05-20	4.08	4
5	2012-05-20	2.39	5
6	2012-05-20	1.99	6
7	2012-05-20	4.36	7
8	2012-05-20	4.46	8
9	2012-05-20	4.50	9
10	2012-05-20	3.99	10
11	2012-05-20	4.09	11
12	2012-05-20	2.39	12
13	2012-05-20	4.36	13
14	2012-05-20	4.46	14
15	2012-05-20	3.00	15
16	2012-05-20	4.08	16
17	2012-05-20	2.39	17

Lentelė 12 Turinys lentelėje susietiduum

ID	DegaluTipai_ID	Degaline_ID
1	dt1	d1
2	dt2	d1
3	dt3	d1
4	dt4	d1
5	dt5	d1
6	dt6	d1
7	dt1	d2
8	dt2	d2
9	dt7	d2
10	dt3	d2
11	dt4	d2
12	dt5	d2
13	dt1	d3
14	dt2	d3
15	dt3	d3
16	dt4	d3
17	dt5	d3

Lentelė 13 Duomenys gauti apjungus lenteles degalutipai, tiekijotinklas, susietiduum, kainuistorija, degaline

ID	Pavadinimas	DegaluPav	Vieta	Data	Kaina
1	Orlean	95	Šilutes pl. 96, Klaipeda	2012-05-20	4.36
2	Orlean	VERVA 95	Šilutes pl. 96, Klaipeda	2012-05-20	4.46
3	Orlean	D	Šilutes pl. 96, Klaipeda	2012-05-20	3.00
4	Orlean	VERVA D	Šilutes pl. 96, Klaipeda	2012-05-20	4.08
5	Orlean	LPG DUJOS	Šilutes pl. 96, Klaipeda	2012-05-20	2.39
6	Orlean	ADBLUE	Šilutes pl. 96, Klaipeda	2012-05-20	1.99
7	Orlean	95	Vilimiškes k., Vydmantai, Kretingos r., Klaipeda - Liepoja 24 km.	2012-05-20	4.36
8	Orlean	VERVA 95	Vilimiškes k., Vydmantai, Kretingos r., Klaipeda - Liepoja 24 km.	2012-05-20	4.46
9	Orlean	98	Vilimiškes k., Vydmantai, Kretingos r., Klaipeda - Liepoja 24 km.	2012-05-20	4.50
10	Orlean	D	Vilimiškes k., Vydmantai, Kretingos r., Klaipeda - Liepoja 24 km.	2012-05-20	3.99
11	Orlean	VERVA D	Vilimiškes k., Vydmantai, Kretingos r., Klaipeda - Liepoja 24 km.	2012-05-20	4.09
12	Orlean	LPG DUJOS	Vilimiškes k., Vydmantai, Kretingos r., Klaipeda - Liepoja 24 km.	2012-05-20	2.39
13	Orlean	95	Dirvupiuk., Gargždu r.	2012-05-20	4.36
14	Orlean	VERVA 95	Dirvupiuk., Gargždu r.	2012-05-20	4.46
15	Orlean	D	Dirvupiuk., Gargždu r.	2012-05-20	3.00
16	Orlean	VERVA D	Dirvupiuk., Gargždu r.	2012-05-20	4.08
17	Orlean	LPG DUJOS	Dirvupiuk., Gargždu r.	2012-05-20	2.39

7.2.3. Ontologija

Sugeneruojama duomenų bazės lentelės, atributus bei ryšius atitinkanti ontologija. Panaudojamos owl:Class, owl:DatatypeProperty, owl:ObjectProperty ypatybės. Ontologijoje esantys lentelių, atributų bei ryšių pavadinimai visiškai sutampa su duomenų bazėje esančiais pavadinimais. Taip pat ontologijoje nurodoma vardų sritis: *xml:base="http://localhost/Degalines#">*.

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns="http://example.org/mydb/mySchema.owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://localhost/Darbiniai/Studentai/Mureika/Degalines#">

  <owl:Class rdf:ID="TIEKEJOTINKLAS">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  </owl:Class>
  <owl:Class rdf:ID="DEGALUTIPAI">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  </owl:Class>
  <owl:Class rdf:ID="DEGALINE">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  </owl:Class>
  <owl:Class rdf:ID="KAINUISTORIJA">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  </owl:Class>

  <owl:Class rdf:ID="DEGALINE">
    <rdfs:isDefinedBy rdf:resource="" />
    <rdfs:label>DEGALINE</rdfs:label>
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class" />
  </owl:Class>
  <owl:DatatypeProperty rdf:ID="ID">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#STRING" />
    <rdfs:isDefinedBy rdf:resource="" />
    <rdfs:domain rdf:resource="#DEGALINE" />
    <rdfs:label>ID</rdfs:label>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="VIETA">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#STRING" />
    <rdfs:isDefinedBy rdf:resource="" />
    <rdfs:domain rdf:resource="#DEGALINE" />
    <rdfs:label>VIETA</rdfs:label>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
  </owl:DatatypeProperty>
  <owl:ObjectProperty rdf:ID="TIEKEJOTINKLAS_ID">
    <rdfs:domain rdf:resource="#TIEKEJOTINKLAS" />
    <rdfs:range rdf:resource="#DEGALINE" />
  </owl:ObjectProperty>
  <owl:Class rdf:ID="DEGALUTIPAI">
    <rdfs:isDefinedBy rdf:resource="" />
    <rdfs:label>DEGALUTIPAI</rdfs:label>
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class" />
  </owl:Class>
  <owl:DatatypeProperty rdf:ID="ID">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#STRING" />
    <rdfs:isDefinedBy rdf:resource="" />
    <rdfs:domain rdf:resource="#DEGALUTIPAI" />
    <rdfs:label>ID</rdfs:label>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
  </owl:DatatypeProperty>
```

```

<owl:DatatypeProperty rdf:ID="DEGALUPAV">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#STRING"/>
  <rdfs:isDefinedBy rdf:resource=""/>
  <rdfs:domain rdf:resource="#DEGALUTIPAI"/>
  <rdfs:label>DEGALUPAV</rdfs:label>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</owl:DatatypeProperty>
<owl:Class rdf:ID="KAINUISTORIJA">
  <rdfs:isDefinedBy rdf:resource=""/>
  <rdfs:label>KAINUISTORIJA</rdfs:label>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
</owl:Class>
<owl:DatatypeProperty rdf:ID="ID">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#STRING"/>
  <rdfs:isDefinedBy rdf:resource=""/>
  <rdfs:domain rdf:resource="#KAINUISTORIJA"/>
  <rdfs:label>ID</rdfs:label>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="DATA">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#DATETIME"/>
  <rdfs:isDefinedBy rdf:resource=""/>
  <rdfs:domain rdf:resource="#KAINUISTORIJA"/>
  <rdfs:label>DATA</rdfs:label>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="KAINA">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#DECIMAL"/>
  <rdfs:isDefinedBy rdf:resource=""/>
  <rdfs:domain rdf:resource="#KAINUISTORIJA"/>
  <rdfs:label>KAINA</rdfs:label>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="SUSIETIDUOM_ID">
  <rdfs:domain rdf:resource="#SUSIETIDUOM"/>
  <rdfs:range rdf:resource="#KAINUISTORIJA"/>
</owl:ObjectProperty>
<owl:Class rdf:ID="TIEKEJOTINKLAS">
  <rdfs:isDefinedBy rdf:resource=""/>
  <rdfs:label>TIEKEJOTINKLAS</rdfs:label>
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
</owl:Class>
<owl:DatatypeProperty rdf:ID="ID">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#STRING"/>
  <rdfs:isDefinedBy rdf:resource=""/>
  <rdfs:domain rdf:resource="#TIEKEJOTINKLAS"/>
  <rdfs:label>ID</rdfs:label>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="PAVADINIMAS">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#STRING"/>
  <rdfs:isDefinedBy rdf:resource=""/>
  <rdfs:domain rdf:resource="#TIEKEJOTINKLAS"/>
  <rdfs:label>PAVADINIMAS</rdfs:label>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</owl:DatatypeProperty>

```

7.2.4. Ontologijos pervadinimo eksperimentinis tikrinimas

Pavyzdinė ontologija apibūdinanti lentelę DEGALUTIPAI;

```
<owl:Class rdf:ID="DEGALUTIPAI">
<rdfs:isDefinedBy rdf:resource="" />
<rdfs:label>DEGALUTIPAI</rdfs:label>
<rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class" />
</owl:Class>
<owl:DatatypeProperty rdf:ID="ID">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#INT32" />
<rdfs:isDefinedBy rdf:resource="" />
<rdfs:domain rdf:resource="#DEGALUTIPAI" />
<rdfs:label>ID</rdfs:label>
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="DEGALUPAV">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#STRING" />
<rdfs:isDefinedBy rdf:resource="" />
<rdfs:domain rdf:resource="#DEGALUTIPAI" />
<rdfs:label>DEGALUPAV</rdfs:label>
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="DTID">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#INT32" />
<rdfs:isDefinedBy rdf:resource="" />
<rdfs:domain rdf:resource="#DEGALUTIPAI" />
<rdfs:label>DTID</rdfs:label>
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
</owl:DatatypeProperty>
```

Pavyzdinė ontologija pervadinama į:

DEGALUTIPAI -> DT

DEGALUPAV -> DPAVADINIMAS

DTID -> DEGALUTIPUID

```
<owl:Class rdf:ID="DT">
<rdfs:isDefinedBy rdf:resource="" />
<rdfs:label>DT</rdfs:label>
<rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class" />
</owl:Class>
<owl:DatatypeProperty rdf:ID="ID">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#INT32" />
<rdfs:isDefinedBy rdf:resource="" />
<rdfs:domain rdf:resource="#DT" />
<rdfs:label>ID</rdfs:label>
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="DPAVADINIMAS">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#STRING" />
<rdfs:isDefinedBy rdf:resource="" />
<rdfs:domain rdf:resource="#DT" />
<rdfs:label>DPAVADINIMAS</rdfs:label>
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="DEGALUTIPUID">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#INT32" />
<rdfs:isDefinedBy rdf:resource="" />
<rdfs:domain rdf:resource="#DT" />
<rdfs:label>DEGALUTIPUID</rdfs:label>
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property" />
</owl:DatatypeProperty>
```

Gaunama teisinga ontologija su atitinkamai pakeistomis ontologijos reikšmėmis. Galime teigti, jog ontologijos pervadinimas veikia.

7.2.5. Metamorphoses žymėjimo rinkmena

```
<?xml version="1.0" encoding="iso-8859-1"?>
<mmm:Mappingxmlns:mmm="http://www.svihla.net/metamorphoses/mapping.dtd">

  <mmm:DatabaseConnection
    mmm:driver="mysql"
    mmm:host = "localhost"
    mmm:port = ""
    mmm:databaseName = "mydb"
    mmm:username = "root"
    mmm:password="">
  </mmm:DatabaseConnection>

  <mmm:DocumentHeader>
    <![CDATA[<?xml version="1.0" encoding="iso-8859-1"?>
    <rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:owl="http://www.w3.org/2002/07/owl#"
      xmlns:Degalai="http://localhost/Degalines.owl#"
      xml:base="http://localhost/Degalines_duom#"
    ]]>
  </mmm:DocumentHeader>

  <mmm:DocumentFoot>
    <![CDATA[</rdf:RDF>]]>
  </mmm:DocumentFoot>

  <Classmmm:templateName = "KAINUISTORIJA" mmm:rdfLabel = "Degalai:KAINUISTORIJA"
  sql ="SELECT s.ID AS SID,Concat('#', s.Degaline_ID) AS DID, Concat('#', s.DegaluTipai_ID) AS
  DTID, k.Data AS data,k.Kaina AS kaina,k.IDasKid FROM susietiduum s
  LEFT JOIN degalutipaidt ON dt.ID = s.DegaluTipai_ID
  LEFT JOIN kainuistorija k ON k.SusietiDuom_ID = s.ID;">
  <mmm:Attributemmm:rdfLabel="rdf:ID" mmm:sqlName="Kid" />
  <mmm:Propertymmm:templateName="DegData" mmm:rdfLabel="Data" mmm:sqlName="data">
    <mmm:Attributemmm:rdfLabel="rdf:datatype"
  mmm:prefix="http://www.w3.org/2001/XMLSchema#date" />
  </mmm:Property>
  <mmm:Propertymmm:templateName="DegKaina" mmm:rdfLabel="Kaina" mmm:sqlName="kaina">
    <mmm:Attributemmm:rdfLabel="rdf:datatype"
  mmm:prefix="http://www.w3.org/2001/XMLSchema#decimal" />
  </mmm:Property>
  <mmm:Propertymmm:templateName="DEGALINE_ID" mmm:rdfLabel="DEGALINE_ID"
  mmm:containerType="Bag">
    <mmm:Attributemmm:rdfLabel="rdf:resource" mmm:sqlName="DID" />
  </mmm:Property>
  <mmm:Propertymmm:templateName="DEGALUTIPAI_ID" mmm:rdfLabel="DEGALUTIPAI_ID"
  mmm:containerType="Bag">
    <mmm:Attributemmm:rdfLabel="rdf:resource" mmm:sqlName="DTID" />
  </mmm:Property>
</Class>

  <Classmmm:templateName = "TIEKEJOTINKLAS" mmm:rdfLabel = "Degalai:TIEKEJOTINKLAS"
  sql ="SELECT * FROM tiekejotinklas ">
  <mmm:Attributemmm:rdfLabel="rdf:ID" mmm:sqlName="ID" />
  <mmm:Propertymmm:templateName="Tiekejas" mmm:rdfLabel="Tiekejas" mmm:sqlName="Pavadinimas">
  <mmm:Attributemmm:rdfLabel="rdf:datatype"
  mmm:prefix="http://www.w3.org/2001/XMLSchema#string" />
  </mmm:Property>
</Class>

  <Classmmm:templateName = "DEGALUTIPAI" mmm:rdfLabel = "Degalai:DEGALUTIPAI"
  sql ="SELECT * FROM degalutipai">
  <mmm:Attributemmm:rdfLabel="rdf:ID" mmm:sqlName="ID" />
  <mmm:Propertymmm:templateName="DegPavadinimas" mmm:rdfLabel="DegaluTipas"
  mmm:sqlName="DegaluPav">
    <mmm:Attributemmm:rdfLabel="rdf:datatype"
  mmm:prefix="http://www.w3.org/2001/XMLSchema#string" />
  </mmm:Property>
</Class>
```

```

<mmm:Classmmm:templateName="deg" mmm:rdfLabel="Degalai:DEGALINE"
mmm:sql="SELECTConcat('#', s.DegaluTipai_ID) asDTid, Concat('#', d.TiekejoTinklas_ID)
asTTid, d.Vietaas Adresas, d.IDasDid FROM susietiduom s LEFT JOIN degaline d ON
d.ID=s.Degaline_ID">
<mmm:Attributemmm:rdfLabel="rdf:ID" mmm:sqlName="Did" />
<mmm:Propertymmm:templateName="DegVieta" mmm:rdfLabel="Adresas" mmm:sqlName="Adresas">
  <mmm:Attributemmm:rdfLabel="rdf:datatype"
mmm:prefix="http://www.w3.org/2001/XMLSchema#string"/>
</mmm:Property>
<mmm:Propertymmm:templateName="DegID" mmm:rdfLabel="Indeksas" mmm:sqlName="Did">
  <mmm:Attributemmm:rdfLabel="rdf:datatype"
mmm:prefix="http://www.w3.org/2001/XMLSchema#int"/>
</mmm:Property>
<mmm:Propertymmm:templateName="DTipas" mmm:rdfLabel="DegaluTipas_ID"
mmm:containerType="Bag">
  <mmm:Attributemmm:rdfLabel="rdf:resource" mmm:sqlName="DTid" />
</mmm:Property>
<mmm:Propertymmm:templateName="TTinklas" mmm:rdfLabel="TiekejoTinklas_ID"
mmm:containerType="Bag">
  <mmm:Attributemmm:rdfLabel="rdf:resource" mmm:sqlName="TTid" />
</mmm:Property>
<mmm:PropertytemplateName="esamasTipas" rdfLabel="esamasTipas" mmm:sqlName=""
mmm:containerType="Bag"/>
</mmm:Class>
</mmm:Mapping>

```

Metamorphozes žymėjimo failo klasėse panaudojama sugeneruota OWL ontologija, nurodant vardų sritį Degalai ypatybėje `mmm:rdfLabel="Degalai:DEGALINE"`.

7.2.6. Metamorphozes šablono rinkmena

```

<?xml version="1.0" encoding="iso-8859-1"?>
<mmt:Template xmlns:mmt="http://www.svihla.net/metamorphoses/template.dtd">
  <mmt:Mapping mmt:url="C:/Program
Files/EasyPHP5.2.10/www/MMPHP/mappings/mydb_mapping2.xml"/>
  <mmt:PutInstance mmt:name="deg" mmt:id='1'>
    <mmt:PutProperty mmt:name="DegVieta"/>
    <mmt:PutProperty mmt:name="DegID"/>
    <mmt:PutProperty mmt:name="DTipas"/>
    <mmt:PutProperty mmt:name="TTinklas"/>
  </mmt:PutInstance>
  <mmt:PutInstance name="DEGALUTIPAI">
    <mmt:PutProperty name="DegPavadinimas"/>
  </mmt:PutInstance>
  <mmt:PutInstance name="TIEKEJOTINKLAS">
    <mmt:PutProperty name="Tiekejas"/>
  </mmt:PutInstance>
  <mmt:PutInstance name="KAINUISTORIJA">
    <mmt:PutProperty mmt:name="DegData"/>
    <mmt:PutProperty mmt:name="DegKaina"/>
    <mmt:PutProperty mmt:name="DEGALINE_ID"/>
    <mmt:PutProperty mmt:name="DEGALUTIPAI_ID"/>
  </mmt:PutInstance>
</mmt:Template>

```

7.2.7. RDF Tikrinimas


```

<Degalai:DEGALINE rdf:ID="d1">
<Adresas rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Šilutes pl. 96,
Klaipeda</Adresas>
<Indeksas rdf:datatype="http://www.w3.org/2001/XMLSchema#int">d1</Indeksas>
<DegaluTipas_ID rdf:resource="#dt4"/>
<TiekejoTinklas_ID rdf:resource="#t1"/>
</Degalai:DEGALINE >
<Degalai:DEGALINE rdf:ID="d1">
<Adresas rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Šilutes pl. 96,
Klaipeda</Adresas>
<Indeksas rdf:datatype="http://www.w3.org/2001/XMLSchema#int">d1</Indeksas>
<DegaluTipas_ID rdf:resource="#dt3"/>
<TiekejoTinklas_ID rdf:resource="#t1"/>
</Degalai:DEGALINE >
<Degalai:DEGALINE rdf:ID="d1">
<Adresas rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Šilutes pl. 96,
Klaipeda</Adresas>
<Indeksas rdf:datatype="http://www.w3.org/2001/XMLSchema#int">d1</Indeksas>
<DegaluTipas_ID rdf:resource="#dt2"/>
<TiekejoTinklas_ID rdf:resource="#t1"/>
</Degalai:DEGALINE >
<Degalai:DEGALINE rdf:ID="d1">
<Adresas rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Šilutes pl. 96,
Klaipeda</Adresas>
<Indeksas rdf:datatype="http://www.w3.org/2001/XMLSchema#int">d1</Indeksas>
<DegaluTipas_ID rdf:resource="#dt1"/>
<TiekejoTinklas_ID rdf:resource="#t1"/>
</Degalai:DEGALINE >
<Degalai:DEGALINE rdf:ID="d3">
<Adresas rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Dirvupiu k., Gargždu
r.</Adresas>
<Indeksas rdf:datatype="http://www.w3.org/2001/XMLSchema#int">d3</Indeksas>
<DegaluTipas_ID rdf:resource="#dt5"/>
<TiekejoTinklas_ID rdf:resource="#t1"/>
</Degalai:DEGALINE >
<Degalai:DEGALUTIPAI rdf:ID="dt7">
<DegaluTipas rdf:datatype="http://www.w3.org/2001/XMLSchema#string">98</DegaluTipas>
</Degalai:DEGALUTIPAI>
<Degalai:DEGALUTIPAI rdf:ID="dt6">
<DegaluTipas rdf:datatype="http://www.w3.org/2001/XMLSchema#string">ADBLUE</DegaluTipas>
</Degalai:DEGALUTIPAI>
<Degalai:DEGALUTIPAI rdf:ID="dt5">
<DegaluTipas rdf:datatype="http://www.w3.org/2001/XMLSchema#string">LPG DUJOS</DegaluTipas>
</Degalai:DEGALUTIPAI>
<Degalai:DEGALUTIPAI rdf:ID="dt4">
<DegaluTipas rdf:datatype="http://www.w3.org/2001/XMLSchema#string">VERVA D</DegaluTipas>
</Degalai:DEGALUTIPAI>
<Degalai:DEGALUTIPAI rdf:ID="dt3">
<DegaluTipas rdf:datatype="http://www.w3.org/2001/XMLSchema#string">D</DegaluTipas>
</Degalai:DEGALUTIPAI>
<Degalai:DEGALUTIPAI rdf:ID="dt2">
<DegaluTipas rdf:datatype="http://www.w3.org/2001/XMLSchema#string">VERVA 95</DegaluTipas>
</Degalai:DEGALUTIPAI>
<Degalai:DEGALUTIPAI rdf:ID="dt1">
<DegaluTipas rdf:datatype="http://www.w3.org/2001/XMLSchema#string">95</DegaluTipas>
</Degalai:DEGALUTIPAI>
<Degalai:TIEKEJOTINKLAS rdf:ID="t1">
<Tiekejas rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Orlean</Tiekejas>
</Degalai:TIEKEJOTINKLAS>
<Degalai:KAINUISTORIJA rdf:ID="16">
<Data rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2012-05-20</Data>
<Kaina rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">4.08</Kaina>
<DEGALINE_ID rdf:resource="#d3"/>
<DEGALUTIPAI_ID rdf:resource="#dt4"/>
</Degalai:KAINUISTORIJA>
<Degalai:KAINUISTORIJA rdf:ID="15">
<Data rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2012-05-20</Data>
<Kaina rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3.00</Kaina>
<DEGALINE_ID rdf:resource="#d3"/>
<DEGALUTIPAI_ID rdf:resource="#dt3"/>
</Degalai:KAINUISTORIJA>
<Degalai:KAINUISTORIJA rdf:ID="14">
<Data rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2012-05-20</Data>
<Kaina rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">4.46</Kaina>
<DEGALINE_ID rdf:resource="#d3"/>
<DEGALUTIPAI_ID rdf:resource="#dt2"/>
</Degalai:KAINUISTORIJA>

```


7.2.8. Duomenų apdorojimo sparta

Duomenų apdorojimo spartai įvertinti naudojamas Hewlett-Packard kompiuteris, turintis Intel Celeron M420 1.60 Ghz 32 bit procesorių 1,5GB RAM atminies su įdiegta Windows 7 operacine sistema.

Rezultatai pateikiami formatu - valandos:minutės:sekundės.miliskundės

Duomenų nuskaitymo laiko iš HTML rinkmenų testas.

Nusikaitant duomenis sukuriama 3096 skirtingų įrašų. Nuskaitymo ir talpinimo trukmė pavdemonstruojama lentelėje Lentelė 14.

Lentelė 14 Duomenų nuskaitymo laiko iš HTML rinkmenų testas.

Testo numeris	Laikas
1	00:00:56.1835374
2	00:00:47.5997226
3	00:00:47.6587260
4	00:00:46.1696408

OWL ontologijos generavimo testas.

Ontologijai sugeneruoti reikia nuskaityti duomenų bazę, bei sukurti ontologiją. Lentelėje Lentelė 15 pateikiami ontologijos generavimo testo rezultatai. Ontologijos generavimui naudojama 4.2 skyriuje aprašyta duomenų bazė.

Lentelė 15 ontologijos generavimo laiko testas.

Testo numeris	Laikas
1	00:00:00.2340134
2	00:00:00.2150123
3	00:00:00.1790103
4	00:00:00.1790103

RDF generavimo laiko testas

Šio testo metu įvertinamas laikas reikalingas RDF dokumentui sugeneruoti naudojant MMPHP įrankį. Rezultatas pateikiamas sekundėmis

Lentelė 16RDF generavimo laiko testas

Testo numeris	Laikas
1	5.5090019702911
2	4.9346590042114
3	5.5654129981995
4	4.1188299655914

RDF generavimo laiko testo tyrimo metu pastebėta, jog RDF duomenų užkrovimas užtrunka papildomą laiką. Nagrinėjamu atveju duomenys naršyklėje pasirodydavo 1 minute vėliau.

7.3. Sistemos veikimo ir savybių analizė, kokybės kriterijų įvertinimas.

Palygine pateiktų html rinkmenų turinį matome, jog turinys HTML rinkmenose bei reliacinės duomenų bazės lentelėse sutampa, todėl galime teigti, jog duomenys yra talpinami teisingai.

Sugeneruojama duomenų bazės lenteles, atributus bei ryšius atitinkanti ontologija. Panaudojamos owl:Class, owl:DatatypeProperty, owl:ObjectProperty ypatybės. Ontologijoje esantys lentelių, atributų bei ryšių pavadinimai visiškai sutampa su duomenų bazėje esančiais pavadinimais. Taigi galime teigti, jog ontologijos generavimas veikia teisingai

Grafinėje sąsajoje pakeitus ontologijos XML pavadinimus taip pat gaunama teisinga ontologija su atitinkamai pakeistomis ontologijos reikšmėmis, todėl galime teigti, jog ontologijos pervadinimas veikia teisingai.

Šiame RDF rinkmenos pavyzdyje matome, jog RDF rinkmenoje esanti informacija (nuorodos į degalų tipą -**DEGALUTIPAI** - ID reikšmės bei nuorodos į degalų tiekėją -**TIEKEJOTINKLAS** - ID reikšmės) sutampa ir sutampa informacija nurodoma

KAINUISTORIJA taguose (**DEGALINE_ID, DEGALUTIPAI_ID, Data, Kaina**) bei duomenų bazėje esantys duomenys, todėl galima teigti jog rdf duomenys yra generuojami teisingai.

7.4. **Sistemos taikymo rekomendacijos**

Sistema patartina diegti, kuomet norima įdiegti turinio publikavimo sprendimą, kuriame yra reikalinga:

- Nuskaityti duomenis iš HTML rinkmenų.
- Sudaryti duomenų bazės ontologiją.
- Išpublikuoti releacinės duomenų bazės turinį.

8. Išvados

- 1) Kadangi dalis informacijos pateikiama HTML puslapiuose, o tiesioginės prieigos prie informacijos duomenų bazių nėra, todėl yra poreikis sukurti priemones automatiškai nuskaitančias HTML puslapyje publikuojamus duomenis ir patalpinti juos į reliacinę duomenų bazę semantiniam išpublikavimui.
- 2) METAmorphoses yra tinkamiausias semantinio tinklo galimybių taikymui atlikti, kadangi šis įrankis yra pateikiamas su atvirojo kodo licencija, o atliktas įvairių įrankių spartos palyginimas (3.10 skyrius) parodė, jog METAmorphoses įrankis yra sparčiausias.
- 3) Duomenų publikavimui iš reliacinės duomenų bazės yra reikalinga tam tikros dalykinės srities ontologija, tačiau nevisų sričių ontologijos yra sukurtos, todėl yra poreikis sukurti priemones, kurios automatiškai sugeneruotų OWL iš reliacinės duomenų bazės schemas.
- 4) Dalį dalykinės srities semantikos galima rasti duomenų bazės schemoje, todėl ontologijos kūrimas iš reliacinės duomenų bazės yra prasmingas ir nereikia nuo pat pradžių kurti dalykinės semantiką. OWL Ontologija yra reikalinga duomenų publikavimui. Sukurta priemonė automatiškai sugeneruojanti pirminį OWL variantą iš reliacinės duomenų bazės
- 5) Atlikus eksperimentinį tyrimą nustatyta, jog pasiūlytas sprendimas veikia korektiškai:
 - a) Nuskaitoma HTML informacija teisingai patalpinama į reliacinę duomenų bazę.
 - b) Iš reliacinės duomenų bazės sugeneruota ontologija yra tokia kokios buvo tikimasi. Sugeneruojama duomenų bazės ryšius, atributus ir lenteles atitinkantis OWL ontologijos aprašymas.
 - c) Teisingai sugeneruojamas RDF dokumentas.
- 6) Siūlomas nuo iki sprendimas apimantis duomenų nuskaitymą, kaupimą ir publikavimą. Publikuojami duomenys paruošiami iš HTML rinkmenų, nuskaitomi, įrašomi į reliacinę duomenų bazę. Vėliau šie duomenys naudojami ontologijos bei RDF duomenų generavimui. Sugeneruota ontologija bei RDF duomenys gali būti užklaunami SPARQL užklausomis.

9. Summary

Data collected from the HTML files, RDF publishing techniques

Evaldas Mureika

Kaunas University of Technology

Department of Information Systems

Semantic Network - is a data network of the information to combined from various sources and displayed as onesolid content. Data, in the Semantic networks is accessed using URIs, via a common network architecture.

Semantic networks and technologies are still very new, unexplored, not tested inpractice and lacking of reference material. It isnesesry to analyzehow the technology is advanced and what technology and tools are used for semantic web application development.

Objectives of this work:

- 1) To analyze how HTML data can be published;
- 2) To analyze the development of semantic web technologies;
- 3) To create an application automatically scanning the data from the HTML files;
- 4) To create an application that generates the OWL ontology of the relational database;
- 5) Apply semantic publishing for data;
- 6) To analyze experimentaly HTML data colleting application and OWL generator;

10. Literatūra

- [1] An Application Server for the Semantic Web, Daniel Oberle, Steffen Staab, Raphael Volz University of Karlsruhe Institute for Applied Informatics and Formal Description Methods 76128 Karlsruhe, Germany. [Žiūrėta 2010-10-9]. Prieiga internete; <http://citeseerx.ist.psu.edu>, dokumentas 10.1.1.2.390
- [2] Creating Semantic Web Services with the Web, Service Modeling Toolkit (WSMT), Mick Kerrigan, Adrian Mocan, Martin Tanler and Werner Bliem, Digital Enterprise Research Institute (DERI). [Žiūrėta 2010-10-10]. Prieiga internete; <http://sourceforge.net/projects/wsmt/>
- [3] The Semantic Web, kovo 17, 2001, Scientific American.com, By Tim Berners-Lee, James Hendler and Ora Lassila [Žiūrėta 2010-10-9]. Prieiga internete; <http://www.scientificamerican.com/article.cfm?id=the-semantic-web>
- [4] Ivan Herman, Tutorial on Semantic Web Technologies, Torento, italy, on the 14th December, 2005. [Žiūrėta 2010-10-10]. Prieiga internete; <http://www.w3.org/2005/Talks/1214-Trento-IH/#%281%29>
- [5] The semantic web (for Web Developers) [Žiūrėta 2010-10-15]. Prieiga internete; <http://logicerror.com/semanticWeb-webdev>
- [6] SenseLab Semantic Web Development [Žiūrėta 2010-10-15]. Prieiga internete; <http://neuroweb.med.yale.edu/senselab/>
- [7] Semantic Web Development; <http://www.w3.org/2000/01/sw/DevelopmentProposal> [Žiūrėta 2010-10-15]. Prieiga internete;
- [8] W3C Semantic Web Activity, <http://www.w3.org/2001/sw/>, [Žiūrėta 2010-11-14]. Prieiga internete;
- [9] DBpedia, <http://dbpedia.org/About>, [Žiūrėta 2010-11-14]. Prieiga internete;

- [10] DBpedia - A Crystallization Point for the Web of Data, Christian Bizer, Jens Lehmann, Georgi Kobilarov, Soren Auer, Christian Becker, Richard Cyganiak, Sebastian Hellmann, <http://www.wiwiss.fu-berlin.de/en/institute/pwo/bizer/research/publications/Bizer-et-al-DBpedia-CrystallizationPoint-JWS-Preprint.pdf> [Žiūrėta 2010-11-14]. Prieiga internete;
- [11] WATSON: SUPPORTING NEXT GENERATION SEMANTIC WEB APPLICATIONS, Mathieu d'Aquin, Claudio Baldassarre, Laurian Gridinoc, Marta Sabou, Sofia Angeletou, Enrico Motta, *Knowledge Media Institute*, http://kmiweb05.open.ac.uk/Downloads%20and%20Publications_files/www-int07.pdf, [Žiūrėta 2010-11-14]. Prieiga internete;
- [12] GRDDL aprašymas, <http://www.w3.org/2004/01/rdxh/spec> [Žiūrėta 2010-11-14]. Prieiga internete;
- [13] How to Publish Linked Data on the Web, Chris Bizer, Richard Cyganiak, Tom Heath, <http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/#relationaldata>, [Žiūrėta 2010-11-15]. Prieiga internete;
- [14] RDF Book Mashup, Chris Bizer Tobias Gauß, [Žiūrėta 2010-11-15]. Prieiga internete; <http://www4.wiwiss.fu-berlin.de/bizer/bookmashup/index.html>,
- [15] RDF Vocabulary Description Language 1.0: RDF Schema, [Dan Brickley](http://www.w3.org/TR/rdf-schema/), R.V. Guha, <http://www.w3.org/TR/rdf-schema/>, [Žiūrėta 2010-12-13]. Prieiga internete;
- [16] SPARQL Query Language for RDF, W3C Recommendation 15 January 2008, <http://www.w3.org/TR/rdf-sparql-query/>, [Žiūrėta 2010-12-13]. Prieiga internete;
- [17] The Friend of a Friend (FOAF) Project, <http://www.foaf-project.org/> [Žiūrėta 2010-12-13]. Prieiga internete;

- [18] Representing vCard Objects in RDF, W3C Member Submission 20 January 2010 Harry Halpin, Renato Iannella, Brian Suda, Norman Walsh. <http://www.w3.org/TR/vcard-rdf/#Intro>. [Žiūrėta 2010-12-13]. Prieiga internete;
- [19] Scalability Report on Triple Store Applications, <http://simile.mit.edu/reports/stores/>, Žiūrėta [2010-12-13]. Prieiga internete;
- [20] SquirrelRDF aprašymas, <http://jena.sourceforge.net/SquirrelRDF/>, [Žiūrėta 2010-12-13]. Prieiga internete;
- [21] METAmorphoses DB to RDF transformation for the semantic web, [Martin Švihla, http://metamorphoses.sourceforge.net/](http://metamorphoses.sourceforge.net/) [Žiūrėta 2010-12-13]. Prieiga internete;
- [22] TopBraid Composer, ontologijų redaktorius http://www.topquadrant.com/products/TB_Composer.html, [Žiūrėta 2011-01-20]. Prieiga internete;
- [23] Protege ontologijų redaktorius, <http://protege.stanford.edu/>, [Žiūrėta 2011-02-10]. Prieiga internete;
- [24] Sesame RDF Java karkaso specifikacija, <http://www.openrdf.org/about.jsp>, [Žiūrėta 2011-01-20]. Prieiga internete;
- [25] VCard žodyno specifikacija, <http://tools.ietf.org/html/draft-ietf-vcarddav-vcardrev-12#section-6.5.2>, [Žiūrėta 2011-02-10]. Prieiga internete;
- [26] DRQ User manual and language specification, <http://www4.wiwiw.fu-berlin.de/bizer/d2rq/spec/index.htm>, [Žiūrėta 2011-01-20]. Prieiga internete;
- [27] SIOC žodyno specifikacija, <http://rdfs.org/sioc/spec/>, [Žiūrėta 2011-01-20]. Prieiga internete;

- [28] ORACLE semantinio tinklo technologijų specifikacija,
http://download.oracle.com/docs/cd/E11882_01/appdev.112/e11828/sdo_rdf_concepts.htm#CIHHEDAC[Žiūrėta 2011-01-20]. Prieiga internete;
- [29] Semantic web architecture, <http://www.obitko.com/tutorials/ontologies-semantic-web/semantic-web-architecture.html> [Žiūrėta 2011-01-20]. Prieiga internete;
- [30] Key features of virtuoso, <http://docs.openlinksw.com/virtuoso/whatisnewto2x.html>,
[Žiūrėta 2011-01-20]. Prieiga internete;
- [31] Realizing a Semantic Web Application, Emanuele Della Valle, <http://applied-semantic-web.org>, [Žiūrėta 2011-01-20]. Prieiga internete;
- [32] Developers Guide to Semantic Web Toolkits for different Programming Languages,
Chris Bizer, Daniel Westphal (Freie Universität Berlin,
Germany)<http://www4.wiwiw.fu-berlin.de/bizer/toolkits/index.htm#comparison>[Žiūrėta
2011-01-20]. Prieiga internete;
- [33] Semantic Web, <http://rhizomik.net/html/~roberto/thesis/html/SemanticWeb.html>,
[Žiūrėta 2011-03-17]. Prieiga internete;