

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACINIŲ SISTEMŲ INŽINERIJOS STUDIJŲ PROGRAMA

SIGITAS ŠERNIUS

OWL 2 ONTOLOGIJŲ MODULINIO KŪRIMO IR JUNGIMO
METODIKA

Magistro darbas

Darbo vadovas:
prof. dr. L. Nemuraitė

KAUNAS, 2013

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACINIŲ SISTEMŲ INŽINERIJOS STUDIJŲ PROGRAMA

SIGITAS ŠERNIUS

OWL 2 ONTOLOGIJŲ MODULINIO KŪRIMO IR JUNGIMO METODIKA

Magistro darbas

Darbo vadovas:
prof. dr. L. Nemuraitė
2013-05-22

Recenzentas:
dr. A. Janavičiūtė
2013-05-24

Atliko:
IFM-1/4 gr. studentas
Sigitas Šernius
2013-05-22

KAUNAS, 2013

Santrauka

OWL 2 ontologijų modulinio kūrimo ir jungimo metodika

Šiuo metu yra plačiai naudojamos ontologijos įvairiems uždaviniams spręsti. Dažnai susiduriama su problema kuriant naujas ontologijas, tai užima daug laiko.

Darbo tikslas sukurti metodiką, kuri nusakytų principus ontologijų jungimui bei moduliniam kūrimui. Analizuojant mokslinę literatūrą buvo nustatyti esami ontologijų jungimo bei kūrimo procesai. Tačiau, šie procesai neužtikrina korektiško ontologijų jungimo bei kūrimo. Todėl buvo detalizuoti konkretesni procesai, kuriais buvo parengta metodika.

Siekiant praktiškai pritaikyti ir palyginti metodiką su kitomis analizės metu nagrinėtomis metodikomis, buvo sukurtas prototipas, kurio veikimas grįstas parengta metodika. Eksperimento metu buvo jungiamos bei kuriamos naujos ontologijos. Įvertinus gautus rezultatus buvo pastebėta jog, parengta ontologijų modulinio kūrimo ir jungimo metodika leidžia patogiau kurti bei jungti ontologijas naudojanti kitas ontologijas ar jų dalis.

Summary

Methodology for Modular Development and Linking OWL 2 Ontologies

At present, ontologies are widely used for various tasks. The main problem is that takes a lot of time of creating new ontologies. There are no methodologies which allow to merge ontologies or reuse various parts of ontology.

The aim of research is to create the methodology of modular development and linking OWL 2 ontologies. This work covers analysis how to create and link the ontologies. During the research, a process was specified for creating and linking ontologies. This process allows creating modular ontologies for better maintenance and reuse.

The experiment was accomplished in order to compare the new methodology whit existing one by creating a prototypical tool for merging ontologies. Three ontologies were created and merged using the prototype tool and Protégé. A comparison of results has shown the advantages of the created prototypical tool and methodology behind it against existing ones. The new methodology allows easily merging ontologies and creating new ontologies by reusing various parts of existing ontologies.

Turinys

Lentelių sąrašas	7
Paveikslų sąrašas	8
Terminų žodynas	9
1. Įvadas	10
2. OWL 2 ontologijų kūrimo ir jungimo analizė	13
2.1 Ontologijų kūrimo principų analizė	13
2.2 Ontologijų modulinio kūrimo principai	14
2.3 Ontologijų jungimo metodai	15
2.3.1 Ontologijų importavimas	16
2.3.2 Ontologijų suliejimas	17
2.3.3 Ontologijų jungimo sprendimo pagrindimas ir esmės išdėstymas	17
2.3.4 Ontologijų jungimo metodų apibendrinimas	21
2.4 Problemos sprendimo metodų literatūros šaltiniuose analizė	21
2.5 Ontologijų kūrimo ir jungimo įrankių analizė	22
2.5.1 <i>Protege</i> ontologijų kūrimo įrankis	23
2.5.2 <i>TopBride Composer</i> ontologijų kūrimo įrankis	24
2.6 Darbo tikslas ir uždaviniai	25
2.7 OWL ontologijų kūrimo ir jungimo analizės apibendrinimas	26
3. Ontologijų jungimo įrankio reikalavimai	28
3.1 Siekiamos sistemos apibrėžimas	28
3.2 Ontologijų jungimo reikalavimų specifikacija ir analizė	30
3.2.1 Panaudos atvejai	30
3.2.1.1 PA 1 Pasirinkti ontologiją	31
3.2.1.2 PA 1.1 Pasirinkti plečiamą ontologiją	31
3.2.1.3 PA 2 Pasirinkti elementus	32
3.2.1.4 PA 2.1 Pasirinkti klasę	33
3.2.1.5 PA 2.2 Pasirinkti klasės egzempliorių	33
3.2.1.6 PA 3 Peržiūrėti ontologiją	34
3.2.1.7 PA 3.1 Redaguoti ontologiją	35
3.2.1.8 PA 3.2 Peržiūrėti klases	36
3.2.1.9 PA 3.3 Peržiūrėti klasių egzempliorius	36
3.2.1.10 PA 3.4 Peržiūrėti ontologijos elementų ryšius	37
3.2.1.11 PA 4 Išsaugoti ontologiją	38
3.2.2 Sisteminiai panaudos atvejai	39
3.2.2.1 PA S1 Nuskaityti ontologiją	39
3.2.2.2 PA S1.1 Atvaizduoti ontologiją	40
3.2.2.3 PA S2 Sujungti ontologijas	41
3.2.2.4 PA S2.1 Atrinkti ontologijos elementus	41
3.2.2.5 PA S2.2 Susieti klasių egzempliorius	42
3.2.2.6 PA S2.3 Sukurti ontologijos elementus	43
3.2.2.7 PA S2.4 Patikrinti ontologiją	43
3.3 Dalykinės srities modelis	44
3.4 Rizikos faktorių analizė	45
4. Ontologijų modulinio kūrimo ir jungimo metodika	46
4.1 Ontologijų modulinio kūrimo procesas	47
4.2 Ontologijų jungimo procesas	49
5. Ontologijų jungimo metodikos pritaikymas	53
6. Eksperimentinis metodikos įvertinimas	57
6.1 Dalykinės srities specifikavimas	57
6.2 Ontologijos sukūrimas	58
6.3 Ontologijų sujungimas	60

6.4	Prototipo testavimas.....	65
6.4.1	Dviejų ontologijų sujungimas.....	65
6.4.2	Modulinis ontologijų kūrimas.....	67
6.5	Sukurtos metodikos vertinimas.....	69
7.	Išvados	71
8.	Literatūra.....	72
9.	Priedai	74
9.1	Jungiamos ontologijos.....	74
9.1.1	Organizacijų ontologija organizacion.owl.....	74
9.1.2	Vietovių ontologija location.owl	74
9.2	Sujungta ontologija	76

Lentelių sąrašas

Lentelė 1. Terminų ir santrumpų paaiškinimai	9
Lentelė 2. Ontologijų jungimo proceso aprašas	19
Lentelė 3. Ontologijos suliejimo proceso aprašas	20
Lentelė 4. Veiklos proceso aprašai	29
Lentelė 5. Ontologijų modulinio kūrimo proceso aprašai	48
Lentelė 6. Ontologijų suliejimo proceso aprašai	51
Lentelė 7. Ontologijų suliejimo proceso aprašai	62
Lentelė 8. Klasių identifikavimo proceso aprašai	64
Lentelė 9. Sukurtos metodikos palyginimas	70

Paveikslų sąrašas

1 pav. Ontologijų medžiai.....	15
2 pav. Sukurta ontologija.....	15
3 pav. Ontologijų sujungimas	16
4 pav. Ontologijos jungimo esamas procesas	18
5 pav. Ontologijos suliejimo procesas.....	20
6 pav. Ontologijų jungimas.....	21
7 pav. PROMPT algoritmas.....	22
8 pav. <i>Protege</i> programos vartotojo sąsaja.....	23
9 pav. <i>TopBride Composer</i> pagrindinis langas	25
10 pav. Veiklos procesų diagrama	28
11 pav. Panaudos atvejų diagrama	30
12 pav. Sisteminiai panaudos atvejai.....	39
13 pav. Ontologijos metamodelis.....	44
14 pav. Ontologijų modulinio kūrimo veiklos procesas.....	48
15 pav. Ontologijų suliejimo veiklos procesas	51
16 pav. Ontologijos kūrimo procesas	53
17 pav. Vartotojo sąsajos loginė architektūra.....	54
18 pav. Sistemos loginė architektūra.....	54
19 pav. Valdymo klasės (moduliai).....	55
20 pav. Vartotojo sąsajos klasės.....	55
21 pav. Vartotojo sąsajos prototipas.....	56
22 pav. Sistemos komponentų diagrama	56
23 pav. <i>Vietos</i> ontologijos medis	57
24 pav. <i>Organizacijų</i> ontologijos medis.....	58
25 pav. <i>Produktų</i> ontologijos medis.....	58
26 pav. Ontologijos kūrimas	58
27 pav. Ontologijos klasės	59
28 pav. Ontologijos sub-klasės	59
29 pav. Ontologijos savybė.....	59
30 pav. Susietos ontologijos klasių egzemplioriai	59
31 pav. Ontologijos klasių egzemplioriai	59
32 pav. Konkrečios ontologijos klasės egzemplioriai	60
33 pav. Pagrindinis sistemos langas	61
34 pav. Ontologijų suliejimas	61
35 pav. Savybių kūrimo langas	63
36 pav. Klasių identifikavimas.....	64
37 pav. Ontologijos vaizdavimo langas.....	65
38 pav. Ontologijos klasių vaizdavimas	65
39 pav. Sukurtos ontologijos vaizdavimas	66
40 pav. Sujungta ontologija	67
41 pav. Sukurta nauja ontologija.....	68
42 pav. Sulieta ontologija	68

Terminų žodynas

Lentelė 1. Terminų ir santrumpų paaiškinimai

Terminas/santrumpa	Paaškinimas
Suliejimas (angl. <i>merge</i>)	Jungimo būdas, kuomet sujungiami vienodų tipų elementai
Ontologija	Ontologija apibrėžia nagrinėjamos srities esybių (daiktų bei reiškinių) tipus, jų tarpusavio sąryšius, galiojančias taisykles, dėsningumus ir aksiomas.
Importavimas (angl. <i>import</i>)	Vienos ontologijos elementų įkėlimas į kitą.
Individas	Ontologijos klasės egzempliorius
Protege	Ontologijų kūrimo įrankis
TopBaid	Ontologijų kūrimo įrankis
PROMPT	Ontologijų kūrimo įrankio <i>Protege</i> papildinys, skirtas ontologijų jungimui
Protege	Ontologijų kūrimo įrankis
<i>TopBride Composer</i>	Ontologijų kūrimo įrankis
Import	Ontologijų importavimo funkcija
Merge	Ontologijų suliejimo funkcija
Prototipas	Bandomoji sistemos versija
Naudotojas	Galutinis sistemos naudotojas dirbantis su sistema.
OWL 2	Web Ontology Language
RDF	Resource Description Framework
XML	Extensible Markup Language

1. Įvadas

Ontologijos terminas yra naudojamas septyniuose skirtinguose kontekstuose: filosofijos mokslo, konceptualizuotos informacijos apie sistemos visumą, objekto semantines reikšmės, koncepcijos specifikacijos, sistemos koncepcijos paremtos logikos teorija, logikos teorijos žodyno, meta-lygio logikos teorijos specifikacijose [1].

Kompiuterijoje ontologija apibrėžia nagrinėjamos srities esybių (daiktų bei reiškinių) tipus, jų tarpusavio sąryšius, galiojančias taisykles, dėsningumus ir aksiomas. Dažniausiai ontologijos naudojamos konkrečios dalykinės srities aprašymui. Viena ontologija apibrėžia vieną konkrečią dalykinę sritį, tačiau dažnai praktikoje pasitaiko atvejų kuomet reikia sujungti dvi ar daugiau ontologijų siekiant detalizuoti aprašomą dalykinę sritį.

Ontologijos yra naudojamos įvairiems uždaviniams spręsti bei aprašyti dalykines sritis:

- įmonių duomenų integravimui – ontologijomis sprendžiama integravimo problema, jos pateikia bendrą įmonės duomenų supratimą;
- duomenų bazės modeliui kurti;
- informacinės sistemos taisyklėms nustatyti;
- gauti informaciją ir ja apsikeisti plačiame pasauliniame žiniatinklyje;
- palengvina informacinės sistemos ir žmogaus komunikavimą. Pavyzdžiui, remiantis ontologija, galima surasti ieškomos informacijos raktinių žodžių kategorijas, sinonimus, o taip didėja tikimybė rasti reikiamą informaciją;
- leidžia bendrauti kompiuterinėms sistemoms.

Aktualumas. Kadangi viso pasaulio ontologija būtų be galo sudėtinga, dažnai nagrinėjama tik specialios (dalykinės) srities ontologija (angl. *Domain Ontology*), pavyzdžiui, mokslinių publikacijų skelbimo sistema, žmogaus anatomija ir pan. Dalykinės srities ontologijos sudarymas ir nagrinėjimas padeda įsisavinti, suvokti realaus pasaulio dalies struktūrą ir jos ryšius [9].

Ontologijos, kaip ir kiti informacinių sistemų komponentai, gali būti daugkartinio naudojimo, nes tos pačios sąvokos vartojamos skirtingoms dalykinėms sritims detalizuoti. Dažniausiai daugkartinio panaudojimo metu ontologijų kūrimui ar jų jungimui naudojami sujungimo arba suliejimo metodai. Šių metodų taikymo metu iš dviejų ontologijų gaunama viena. Gauta ontologija aprašo dalykinę sritį arba tam tikro uždavinio sprendimą.

Kuriant ontologiją, dalykinės srities surinktiems terminams saugoti siūloma naudoti formalų kontekstą, kad vėliau terminus būtų galima analizuoti, tikslinti bei vaizduoti. taip išsaugotus formaliame kontekste terminus galima vartoti įvairioms užduotims atlikti. Pavyzdžiui, paieškai, informacijos apdorojimui ir t. t.

Rengiama metodika padės plačiau pažvelgti į ontologijų ar jų dalių pakartotinį panaudojimą. Tokiu būdu bus palengvintas didelių ontologijų kūrimas.

Tyrimo problema. Nėra parengtos metodikos, kuri nurodytų ontologijų modulinio kūrimo principus, leidžiančius konstruoti naujas ontologijas pakartotinai panaudojant mažesnes ontologijas ar jų dalis. Parengta metodika leistų užtikrinti patogų ir tikslų būdą sujungti ontologijas. Jungimo metu būtų gauta nauja ontologija, turinti kelių ontologijų elementus bei savybes.

Protege įrankis turi galimybę sujungti dvi ontologijas naudojant *import* ir *merge* funkcijas. Tačiau jei pradinės ontologijos buvo kuriamos laisvai, nenumatant pakartotinio panaudojimo ateityje, šių funkcijų taikymas neduos laukiamų rezultatų. Importavimo ar suliejimo funkcijos surašo abiejų ontologijų informaciją į vieną, tik importavimo atveju jungimas yra virtualus (abi ontologijos išlieka savarankiškos ir jas galima modifikuoti nepriklausomai), o jungimo atveju gaunama nauja ontologija, kuri modifikuojama nepriklausomai nuo sujungtų ontologijų. Jei ontologijos netinkamai parengtos, dubliuojami vienodi ontologijų įrašai. Parengiant ontologijas jungimui, reikia redaguoti ontologijų specifikacijas ir suvienodinti jų *IRI* adresus, kurie naudojami ontologijoms sujungti.

Tyrimo objektas – OWL 2 ontologijų modulinio kūrimo ir jungimo procesas.

Tyrimo tikslas – padidinti ontologijų kūrėjų galimybes kurti ir pakartotinai panaudoti modulines ontologijas sukuriant metodiką, kuri leistų pakartotinai panaudoti kuriamas ontologijas kitų ontologijų kūrime.

Tyrimo sritis yra ontologijų inžinerija t.y. ontologijų modulinio kūrimo bei jungimo principų analizė bei metodologijos rengimas praktiškai pritaikant egzistuojančius sprendimus ontologijų kūrimui bei jungimui. Taikant šių įrankių galimybes, turi būti galima automatiškai sujungti dvi pasirinktas ontologijas taip, kad nenukentėtų gautos ontologijos kokybė, t. y., sujungus nesidubliuotų įrašai bei nebūtų prarasta pradinė informacija.

Darbai apribuoti siekiama parengti pavyzdinį ontologijų jungimo įrankį, kuris būtų realizuotas remiantis parengtos ontologijų jungimo metodikos principais.

Darbo uždaviniai.

1. Išanalizuoti:
 - 1.1. ontologijų kūrimo principus;
 - 1.2. ontologijų jungimo principus;
 - 1.3. egzistuojančius ontologijų kūrimo ir jungimo įrankius ir jų galimybes.
2. Remiantis atlikta analize, sudaryti ontologijų modulinio kūrimo ir jungimo metodiką, nusakančią jungimo taisykles ir apribojimus.
3. Realizuoti ontologijų jungimo įrankio prototipą, kuris leistų pasirinkti jungiamas klases, klasių egzempliorius bei savybes.

4. Atlikti eksperimentą, kuris leistų įvertinti metodikos tinkamumą ir pritaikymą realioms sprendimams.
5. Įvertinti tyrimo rezultatus.

Darbo rezultatai. Sudaryta metodika, aprašanti ontologijų modulinio kūrimo ir jungimo procesą. Metodikai išbandyti ir įvertinti buvo sukurtas prototipas bei atliktas eksperimentas.

Naujumas. Parengta ontologijų kūrimo ir jungimo metodika, kuri aprašo kriterijus ontologijų kūrimui, kad sukurtas ontologijas būtų galima pakartotinai panaudoti kitų ontologijų kūrimui.

Praktinis pritaikymas. Siekiant įvertinti sukurtą metodiką, buvo sukurtas prototipas. Taikant metodiką ir prototipą buvo kuriamos naujos ontologijos. Pagal metodikos nurodymus buvo sukurtos trys ontologijos, kurios gali būti pakartotinai panaudojamos. Pasitelkus prototipą, sukurtos ontologijos buvo naudojamos naujų ontologijų kūrimui. Tokios pat ontologijos buvo kuriamos naudojantis *Protege* ontologijų kūrimo įrankiu. Atlikus ontologijų kūrimą *Protege* įrankiu buvo gautos perteklinės ontologijos t.y., sukurtos ontologijos turėjo nenaudojamų elementų, kurie apkrauna ontologiją ir tokiu atveju ontologija gali tapti nekorektiška.

Atlikus ontologijų kūrimo veiksmus, buvo lyginami gauti rezultatai. Tai leido įvertinti parengtą ontologijų modulinio kūrimo ir jungimo metodiką. Vertinimo metu buvo nustatyta, kad sukurta metodika yra leidžia efektyviau kurti naujas ontologijas, kai tam naudojamos esamos ontologijos ar jų dalys.

Darbo struktūra.

Antrame skyriuje aprašoma darbo metu atlikta ontologijų kūrimo ir jungimo metodų analizė, apžvelgti egzistuojantys įrankiai ir jų galimybės ontologijų jungimui bei kūrimui.

Trečiame skyriuje pateikiama automatizuojamų funkcijų analizė, kurios metu buvo analizuojamos galimybės realizuoti programinius įrankius kuriamos metodikos pagrindu.

Ketvirtame skyriuje pateikiama sukurta metodikos, skirtos ontologijų moduliniam kūrimui ir jungimui, žingsniai ir eiga.

Penktame skyriuje pateikiamas sukurta ontologijos pritaikymas, specifikuojami reikalavimai, kurių pagrindu bus kuriamas prototipas.

Šeštame skyriuje pateikiamas eksperimentas. Šioje dalyje parašomas sukurta prototipas bei atliekamo eksperimento eiga bei rezultatai. Taip pat pateikiamas parengtos metodikos vertinimas.

2. *OWL 2* ontologijų kūrimo ir jungimo analizė

Šiame skyriuje pateikiama *OWL 2* ontologijų kūrimo ir jungimo analizė, kurios metu apžvelgiami ontologijų jungimo, kūrimo principai bei galimybės iš dalies automatizuoti atskirų ontologijų jungimo procesą, šio proceso metu būtų automatizuotas ontologijų klasių ir savybių susiejimas į vieną ontologiją. Išanalizavus ontologijų kūrimo bei jungimo principus bus siekiama detalizuoti ontologijų modulinio kūrimo principus bei taisykles, kad būtų paprasčiau ir patogiau sukurti dideles ontologijas.

Taip pat pateikti ontologijų modulinio kūrimo reikalavimus siekiant palengvinti ontologijų kūrimą iš atskirų ontologijų ar jų dalių. Reikalavimai apibrėš ontologijų kūrimo principus, kad kuriamos ontologijos būtų tinkamos moduliniam ontologijų kūrimui.

Analizuojant mokslinėje literatūroje aprašytus ontologijų kūrimo ir jungimo metodus ir galimybes, taip pat bus nagrinėjami ir egzistuojantys sprendimai. Nagrinėjami sprendimai yra skirti būtent ontologijų kūrimui, tačiau turi galimybę dalinai apjungti kelias ontologijas.

Taigi **analizės tikslai** – išsiaiškinti ontologijų kūrimo principus ir jų pritaikymą, išanalizuoti problemas, kurios kyla jungiant ontologijas ir tolimesnių analizių metu priimti sprendimą šioms problemoms spręsti. Pagal analizės metu gautus rezultatus bus parengta ontologijų jungimo metodiką pateikiant tenkinimo sąlygas ir apribojimus, kurie turi būti tenkinami jungiant ontologijas.

2.1 Ontologijų kūrimo principų analizė

Ontologijos yra naudojamos įvairiems uždaviniams spręsti bei aprašyti dalykinės sritis. Dažniausiai naudojamos detalizuoti duomenų bazės modeliui, kurti informacinės sistemos taisyklėms nustatyti bei gauti informaciją ir ja apsieisti plačiame pasauliniame žiniatinklyje. Taip pat ontologijos plačiai naudojamos semantinio žiniatinklio uždavinių sprendimui, kurių metu yra detalizuojami terminų žodynai.

Siekiant sukurti ontologiją, reikia apibrėžti dalykinę sritį. Analizuojant dalykinę sritį kuriamai ontologijai suformuluojami konkretūs reikalavimai. Ontologijai keliamus reikalavimus galima apibrėžti kaip kompetencijos klausimus, į kuriuos, kuriama ontologija, turi atsakyti [22].

Siekiant sukurti kokybiškas ontologijas, kurioms būtų paprasta pritaikyti jungimo ir importavimo galimybes, reikia remtis kokybiniais kriterijais. Šie kriterijai nusako kaip teisingai sukurti ir detalizuoti ontologiją, kad ši būtų korektiška ir pilna. Remiantis šiais kriterijais sukuriama ontologija, kuri leidžia atvaizduoti tikslią informaciją.

Būtinai ontologijų kokybiniai kriterijai:

Minimalumas – ontologija turi apimti minimalią ir būtiną klasių, savybių aibę, t.y. kuriama ontologija turi būti aprašyta panaudojus terminus ar žodynus, kad būtų galima suprasti ką norima atvaizduoti ontologijoje [14].

Pilnumas - ontologija turi apimti visą būtiną klasių, savybių ir aksiomų aibę, kuri leidžia atsakyti į kompetencijos klausimus. Sukurta ontologija turi būti suprantama ir atvaizduoti visą informaciją, kuriai atvaizduoti ir buvo kuriama ontologija [14].

Korektiškumas – ontologijos turi būti kuriamos korektiškai aprašant ir detalizuojat klasių bei savybių aibes. Taip pat detalizuojant ontologijos hierarchinius ryšius tarp sukurtų klasių [14].

Neprieštarinumas – aksiomos turi būti neprieštaringos. Kuriamų ontologijų klasių ir savybių sąryšiai neturi prieštarauti vieni kitiems. Taip pat turi būti detalizuoti ryšiai tarp jų. Tai galima patikrinti pasinaudojus *Protege* galimybe *Reasoning*. Ši funkcija detaliau bus apžvelgiama tolimesniuose tyrimo etapuose, kuomet bus rengiama ontologijų modulinio kūrimo metodika[14].

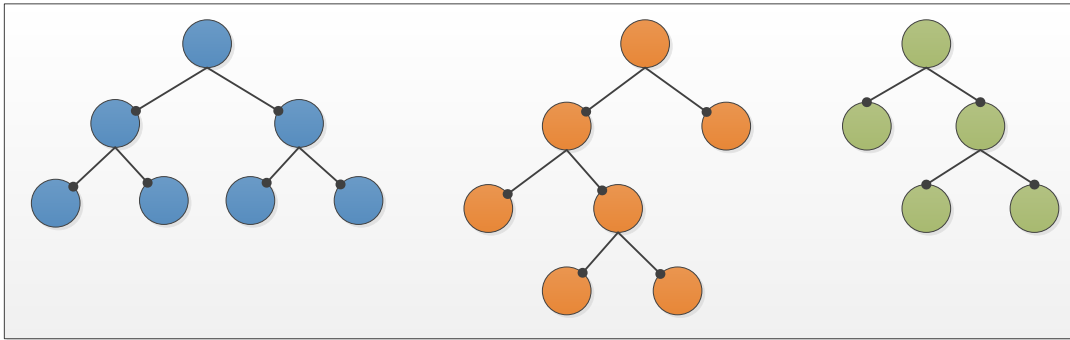
Normalizavimas – kuriama ontologija turi atitikti normalizavimo taisykles, kurios yra dvi: ontologinis normalizavimas (angl. *ontological normalisation*) ir įgyvendinimo normalizavimas (angl. *implementation normalisation*). Ontologinis normalizavimas tai – tai žmogaus supratimas apie pristatomus konceptus. Tai priklauso nuo technologijų, kurios yra naudojamos realizuoti ontologijas. Įgyvendinimo normalizavimas – tai logika pagal kurią yra realizuojamos ontologijos, t.y. nusakomi ontologijos hierarchiniai ryšiai tarp klasių ir savybių aibių [15].

2.2 Ontologijų modulinio kūrimo principai

Modulinis ontologijų kūrimas yra pagrįstas ontologijų sujungimo principais, kadangi ontologijos kūrimo metu iš atskirų ontologijų ar jų dalių yra konstruojama nauja ontologija, kuri apima vieną platesnę arba kelias dalykines sritis.

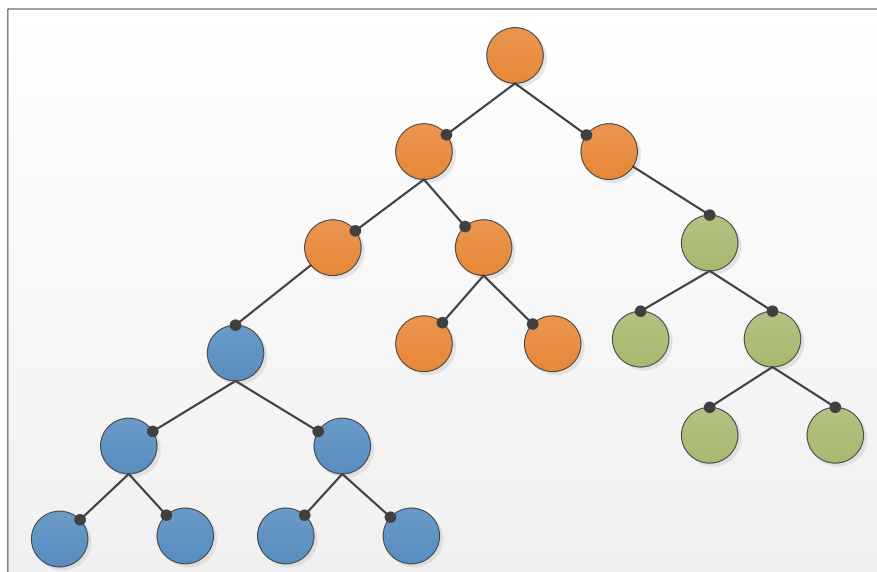
Ontologijų modulinio kūrimas leidžia turimas ontologijas pakartotinai panaudoti kotų ontologijų kūrimui, kadangi naudojamos ontologijos yra neredaguojamos, o tik naudojamos kaip informacijos ar duomenų šaltiniai.

Siekiant vaizdžiai parodyti ontologijų modulinį jungimą, toliau yra pateikiamas ontologijų modulinio kūrimo pavyzdys. Iš trijų ontologijų, kurios suskurtos remiantis rekomendacijomis. Iš pateiktų ontologijų (1 pav. Ontologijų medžiai) yra sudaroma viena didelė ontologija, kuri gali būti naudojama specifiniams uždaviniams spręsti.



1 pav. Ontologijų medžiai

Moduliniu kūrimo metu sujungus aukščiau pateiktas ontologijas gaunama viena (2 pav. Sukurta ontologija). Tokiu būdu kuriant ontologijas galima pakartotinai panaudoti jau turimas ontologijas ar jų dalis.



2 pav. Sukurta ontologija

Modulinis ontologijų kūrimas palengvina darbą kadangi nėra būtina iš naujo kurti atskirų ontologijų arba iš naujo kurti vieną didelę ontologiją apimančia kelias ontologijas.

Kaip buvo minėta anksčiau, modulinis ontologijų jungimas leidžia taikyti ontologijų daugkartinio panaudojimo principą. Tokiu būdu yra palengvinamas ontologijų kūrimo procesas, nes nereikia kurti ontologijos iš pagrindų. Kūrimo metu užtenka pasirinkti kelių ontologijų turimas dalis ar visas ontologijas ir sulipdyti vientisą didelę ontologiją.

2.3 Ontologijų jungimo metodai

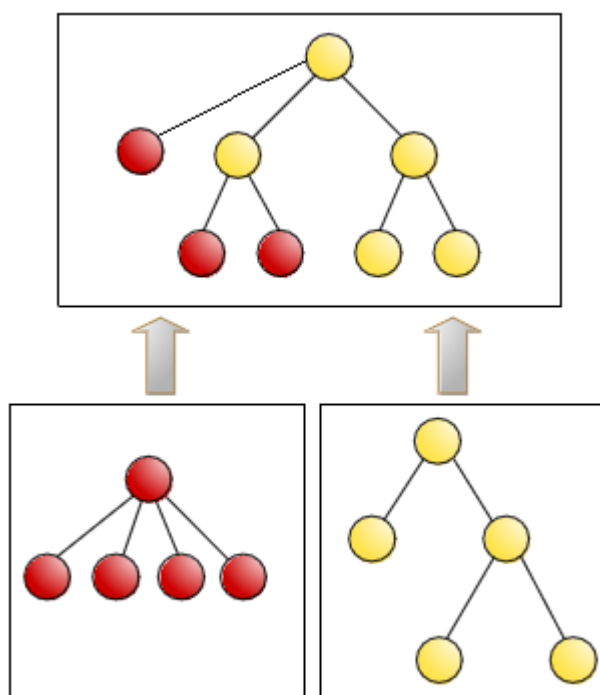
Yra keli ontologijų jungimo būdai, kuriais naudojantis galima iš dviejų ontologijų sudaryti vieną ontologiją. Taip pat yra galimybė paruoti šablonus [8], kuriais naudojantis bus sukuriamos ontologijos ir pagal šablonų turimas savybes jos bus sujungiamos.

Ontologijas sujungti galima dviem kompiuterizuotais būdais, t.y. suliejimo (anlg. *merge*) ir importavimo (anlg. *import*). Ontologijų jungimo metu galima sujungti dvi lokalias ontologijas. Taip

pat yra galimybė prie lokaliios ontologijos prijungti ir nutolusią ontologiją, t.y. galima naudoti ontologijos, kuri yra talpinama serveryje ar kitoje nutolusioje vietoje, elementus.

Tiek siejimo (angl. *merge*) tiek importavimo (angl. *import*) funkcijos yra naudingos atliekant ontologijų jungimo ar siejimo veiksmus. Kiekviena funkcija naudojama skirtingiems rezultatams gauti ar veiksmams atlikti. Suliejimo funkcija geriausia taikyti kuomet norime sujungti dvi skirtingas ontologijas. Importavimo galimybė naudojama kuomet norime sujungti dvi vienodas ontologijas, t.y. papildyti vieną ontologiją kitos ontologijos savybėmis.

Ontologijų jungimas naudojamas tam, kad būtų galima lengviau ir patogiau sukurti tam tikros dalykinės srities ontologiją. Taip pat ontologijų jungimas naudojamas siekiant papildyti ontologiją naujais elementais, juos tiesiog importavus į pasirinktą ontologiją.



3 pav. Ontologijų sujungimas

Svarbiausias dalykas, jungiant dvi ontologijas, tai klasių vardų suderinimas. Žinant kad siejamų ontologijų klasių pavadinimai atitinka pagal prasmę, būtų galima susieti klases, kurios turi tuos pačius pavadinimus. Tuomet siejimo metu vienos klasės elementai būtų priskirti kitai. Tokiu atveju gautųsi viena ontologija, klasės turėtų dviejų skirtingų ontologijų savybes.

2.3.1 Ontologijų importavimas

Ontologijų importavimo metu į vieną ontologiją importuojama kita t.y. viena ontologija papildoma kitos ontologijos elementais neatsižvelgiant į jau egzistuojančius ontologijos elementus. Importavimo metu tiesiog perkeliama ontologijų elementai.

Šio ontologijų jungimo metu nėra tikrinami ontologijų turimi elementai, todėl jungimo metu atsiranda duomenų dubliavimas. Importuojant ontologiją, kuri turi vienodų elementų, reiktų

suvienodinti *IRI* adresus, kad vienodi elementai būtų sujungiami arba prijungiami prie kitų elementų. Importavimo funkcija naudojama sujungti dvi vienodus aprašus turinčias ontologijas. Šios funkcijos metu pagrindinė ontologija (ontologija į kurią importuojama kita ontologija) papildoma kitos ontologijos savybėmis.

Naudojant importavimo funkciją patogiauiausia sujungti dvi ontologijas, kurios apibrėžia vienodą dalykinę sritį tačiau yra sudarytos iš skirtingų elementų. Tokiu atveju nauja ontologija būtų papildyta naujais elementais, kurie yra naudojami detalizuoti tą pačią dalykinę sritį.

Ontologijų importavimą geriausia naudoti kuomet reikia iš dviejų skirtingų ir nesusijusių ontologijų sukurti vieną. Kadangi, abiejų ontologijų elementai yra sudedami į vieną ontologiją. Tokiu būdu gauta ontologija yra sudaryta iš dviejų ontologijų, kurios aprašo skirtingas dalykines sritis ir sprendžia skirtingus uždavinius.

2.3.2 Ontologijų suliejimas

Ontologijų suliejimo metu yra sukuriama nauja ontologija, kuri turi abiejų ontologijų elementus. Suliejimo metu yra susiejami vienodi ontologijų elementai ir išvengiama duomenų dubliavimo. Ontologijų suliejimo metu vienodi elementai užrašomi vienas ant kito, tokiu atveju galimas duomenų praradimas. Siekiant išvengti duomenų praradimo reikia atkreipti dėmesį į jungiamų ontologijų struktūrą ir jų elementų aprašymo detalumą.

Naudojant ontologijų suliejimo metodus papildomai reikia atlikti ontologijos patikrinimą bei jos elementų detalizavimą. Atlikus šiuos veiksmus galima sulieti ontologijas. Kadangi suliejimo metu yra per rašomi esami ontologijų elementai. Tokiu būdu gali būti prarasta informacija. Siekiant išvengti informacijos praradimo ontologijų susiejimą reikia atlikti pažingsniui ir siejant kiekviena ontologijos elementą atskirai.

Ontologijų suliejimą geriausias naudoti kuomet norima sujungti vienodas arba panašias ontologijas. Kadangi siejimo metu bus papildyti ontologijų elementai naujomis savybėmis. Tokiu būdu bus gauta nauja ontologija, kur detalizuota ir labiau pritaikyta konkrečiai dalykiniai sričiai.

2.3.3 Ontologijų jungimo sprendimo pagrindimas ir esmės išdėstymas

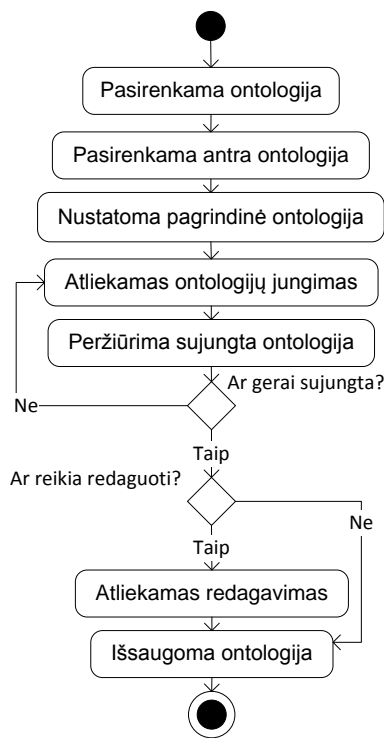
Šiuo metu susiduriama su problemomis jungiant dvi skirtingas ontologijas, t.y. ontologijų jungimo metu realizuoti sprendimai negali atskirti vienodų ontologijų klasių bei savybių, todėl atsiranda duomenų dubliavimas. Parengta ontologijų jungimo metodika leis apibrėžti reikalingus veiksmus siekiant išvengti jungimo metu dubliuojamų duomenų bei leis sukurti korektišką ontologiją, t.y. sukurtoje ontologijoje nebus duomenų dubliavimo bei nesusietų klasių ir savybių.

Taip pat siekiama, kad parengta ontologijų siejimo metodika būtų pritaikyta ontologijų moduliniam kūrimui. Tokiu būdu ontologijos būtų kuriamos iš mažesnių dalių konstruojant vieną didelę ontologiją.

Siejimo metodika pagrįsta dviejų skirtingų ontologijų susijimu t.y. siejimo metu yra siejamos dvi skirtingos ontologijos, kurios detalizuoja viena kitą ir tokiu būdu sukuriama nauja ontologija, kuri skirta aprašyti konkrečią dalykinę sritį.

Ontologijų jungimo metu, siekiant pabrėžti modulinį ontologijų jungimą, jungiamose ontologijose turi būti aprašyta tik viena dalykinė sritis t.y., vienoje ontologijoje gali būti tik viena medžio viršūnė. Pvz. ontologijoje detalizuojamas *laikas*. Tai šioje ontologijoje aprašoma visa informacija susijusi su laiko momentu. Siekiant taikyti modulinį ontologijų jungimą turimoje ontologijoje negali būti elementų nesusijusių su *laiku*. Ir kitos jungiamos ontologijos turi būti sukurtos taikant panšius principus. Tokiu būdu jungiamos ontologijos išlaikys vienodą struktūrą. Tai užtikrins patogesnę ir lengvesnę ontologijų jungimą.

Rengiant ontologijų jungimo metodiką, buvo detalizuotas esamas procesas, kurio metu yra atliekami ontologijų jungimo veiksmai, detalizuoti siejimo ir jungimo procesai. Pastarieji ontologijų jungimo procesai tinkami tik konkrečiais atvejais, todėl daugiau orientuojamasi į naujo ontologijų jungimo procesą. Toliau paveiksle pateikiamas ontologijų jungimo procesas, kuris detalizuoja pagrindinius ontologijų jungimo veiksmus.

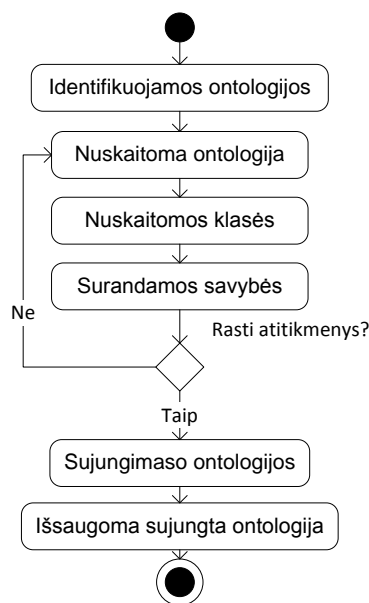


4 pav. Ontologijos jungimo esamas procesas

Lentelė 2. Ontologijų jungimo proceso aprašas

Veiksmas	Aprašymas
Pasirenkama ontologija	Pasirenkama ontologija, kurios elementai bus naudojami kitos ontologijos praplėtimui. Pasirenkama ontologija turi būti tinkamo formato ir struktūros.
Pasirenkama antra ontologija	Pasirenkama ontologija, kuri bus plečiama naujais elementais ir savybėmis. Pasirenkama ontologija taip pat turi būti tinkamo formato ir struktūros.
Nustatoma pagrindinė ontologija	Sistema antrą ontologiją naudoja kaip pagrindą naujos ontologijos sukūrimui. Tokiu būdu ontologija papildoma reikiamais elementais ir savybėmis.
Atliekamas ontologijų jungimas	Sistema analizuoja parinktas ontologijas ir atlieka ontologijų jungimą. Jungimo metu yra perkeliama bei sukuriama nauji elementai taip pat sukuriama sistemos naudotojo nurodytos ontologijos savybės.
Peržiūrima sujungta ontologija	Pateikiama ontologija peržiūrai siekiant įsitikinti ar jungimo veiksmas buvo atliktas korektiškai. Jei jungimo metu buvo atlikti klaidingi veiksmas, tai galima koreguoti gautą ontologiją. Po kiekvienų atliktų koregavimo veiksmų turi būti tikrinama ontologija siekiant sukurti korektišką ontologiją.
Ar gerai sujungta?	Tikrinama ar elementai ir savybės yra teisingai sukurtos naujoje ontologijoje. Norint sukurti korektišką ontologiją reikia patikrinti ar buvo sukurti reikiami elementai, savybės bei ryšiai tarp reikiamų ontologijos elementų.
Ar reikia redaguoti?	Tikrinant ontologija patikus netikslumų galima atlikti ontologijos koregavimą siekiant pašalinti ontologijos trūkumus.
Atliekamas redagavimas	Atliekami ontologijos redagavimo veiksmas. Pakeičiami ontologijos elementų pavadinimai, aprašai, sukuriama bei redaguojami ryšiai. Redagavimo metu sukuriama korektiška ontologija.
Išsaugoma ontologija	Išsaugoma sukurta ontologija. Nauja ontologija yra saugoma remiantis pasirinktų ontologijų struktūra. Ontologija išsaugoma faile, kuris atitinka pasirinktų ontologijų formatą bei struktūrą.

Jungiant lokalią ir nutolusia ontologiją atsiranda sunkumu suvienodinant *IRI* adresą, kadangi nutolusios ontologijos nėra galimybės redaguoti ir keisti jos *IRI* adresą. Tačiau naudojant importavimo funkciją galima turimą ontologiją papildyti naujais elementais iš nutolusios ontologijos.



5 pav. Ontologijos suliejimo procesas

Lentelė 3. Ontologijos suliejimo proceso aprašas

Veiksmas	Aprašymas
Identifikuojamos ontologijos	Patikrinamos parinktos ontologijos. Ar jos yra vienodo formato ir struktūros.
Nuskaitomos ontologijos	Analizuojamos ontologijos nuskaitant visus turimus elementus. Apdoroja kiekvieną ontologijos elementą atskirai identifikuojant turimas savybes, ryšius bei aprašus.
Nuskaitomos klasės	Identifikuojamos ontologijų klasės, kurios turi būti sujungiamos. Kadangi siejimo metu siejamos vienodos klasės papildant jas naujais lementais ar naujomis savybėmis.
Surandamos savybės	Identifikavus ontologijų klases ir radus vienodas klases yra identifikuojamos jų savybės, kurios bus susiejamos. Siejimo metu savybės bus perrašomos jei jos bus vienodos, kitu atveju bus sukuriamos naujos.
Sujungiamos ontologijos	Pagal rastus atitikmenis sujungiamos dvi pasirinktos ontologijos. Jungimo metu vienodi ontologijų elementai yra papildomi naujomis savybėmis.
Išsaugoma sujungta ontologija	Išsaugoma sukurta ontologija. Nauja ontologija yra saugoma remiantis pasirinktų ontologijų struktūra. Ontologija išsaugoma faile, kuris atitinka pasirinktų ontologijų formatą bei struktūrą.

Naudojant siejimo (angl. *merge*) funkciją reikia redaguoti ontologiją aprašantį failą. Šio failo viduje reikia pakeisti unikalius ontologijos *IRI* adresus. Šis adresas turi būti pakeistas į pagrindinės ontologijos adresą, kadangi pagal tai klasės ir jų įrašai bus prijungiami prie egzistuojančių klasių. Naudojant siejimo (angl. *merge*) funkciją yra sukuriama nauja ontologija, kuri yra sudaryta iš siejamų ontologijų.

2.3.4 Ontologijų jungimo metodų apibendrinimas

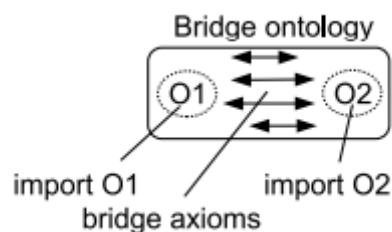
Importavimo funkcija prijungė siejamos ontologijos klases ir jų turimus įrašus prie pagrindinės ontologijos. Taip atsirado klasių pavadinimų dubliavimas ir esamos klasės nepasipildė naujais įrašais. Atsižvelgiant į tai, importavimo funkcija netenkina ontologijų jungimui keliamų reikalavimui, t.y. esamos klasėse nesukuriami nauji įrašai, sukuriamos analogiškos klasės, ko turėtų nebūti. Importavimo galimybė gerai tinka tuo atveju, jei norima vieną ontologiją papildyti naujomis klasėmis su skirtingais įrašais.

Ontologijų suliejimo metodas yra labiau tinkamas kuomet norima sujungti dvi vienodas ar panašias ontologijas į vieną, kadangi siejimo metu yra perrašomi esantys vienodi elementai. Siejant ontologijas gali atsirasti netikslumu kadangi perrašant elementus galimas duomenų praradimas ar jų nekorektiškumas sukurtos ontologijos atžvilgiu.

Sujungiant ontologijas reikia konkrečiai detalizuoti, kokius ontologijų elementus norima sujungti, kokias savybes norima sukurti ar detalizuoti. Tik tokiu atveju pavyks sukurti korektišką ontologiją, kuri aprašys tam tikrą dalykinę sritį ir bus naudojama konkrečioms uždaviniais spręsti.

2.4 Problemos sprendimo metodų literatūros šaltiniuose analizė

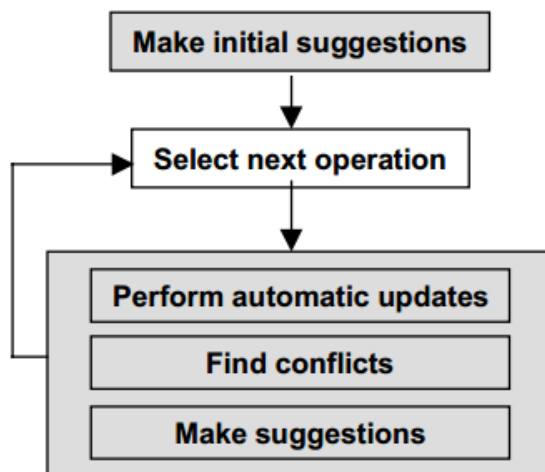
Šiuo metu naudojami sprendimai negali pasiūlyti visiškai sujungtų ontologijų tikslumo. Dauguma sprendimų nusako tik sąsajas tarp vienodų ar panašių ontologijų elementų. Naudojantis šiais sprendimais sujungus ontologijas tenka redaguoti gautą ontologiją, kad būtų galima ją pritaikyti ir naudoti praktiškai.



6 pav. Ontologijų jungimas

Aukščiau pateiktas paveiklas iliustruoja sąsajas tarp dviejų importuojamų ontologijų. Importavimo metu yra sukuriami ryšiai tarp atitinkamų ontologijų elementų. Tokiu būdu reikia naudoti dvi ontologijas norimiems uždaviniais spręsti ar detalizuoti norimą dalykinę sritį.

Ontologijų suliejimui galima naudoti *PROMPT* algoritmą, kuris paima dvi siejamas ontologijas ir sukuria naują ontologiją. Naudojant šį algoritmą naudotojui yra pasiūlomas siejamų ontologijų klasių sąrašas iš kurio naudotojas pasirenka atitinkamas ontologijas ir jas susieja. Tuomet *PROMPT* automatiškai atieka pakeitimus ontologijų klasėse.



7 pav. *PROMPT* algoritmas

PROMPT algoritmu grįstas ir *Protege* ontologijų jungimo funkcionalumas. Tam tikslui yra specialiai sukurti *Protege* įrankiai, leidžiantys atlikti ontologijų jungimo veiksmus. Tačiau naudojant šį algoritmą nėra visiškai užtikrintas susietos ontologijos korektiškumas. Dažnu atveju reikia atlikti papildomus ontologijos redagavimo veiksmus.

2.5 Ontologijų kūrimo ir jungimo įrankių analizė

Šiuo metu Lietuvoje nėra žinoma panašių įrankių, kurie patogiai leistų sujungti dvi ontologijas, taip pat nėra parengtos konkrečios ontologijų jungimo metodikos, kuri nurodytų tikslius veiksmus ontologijų jungimo procese. Užsienio valstybėse yra žinomi keli įrankiai, kurie tik iš dalies leidžia korektiškai sujungti ontologijas.

Detalesniam esamų sprendimų apžvalgai pasirinkti du ontologijų kūrimo įrankiai tai:

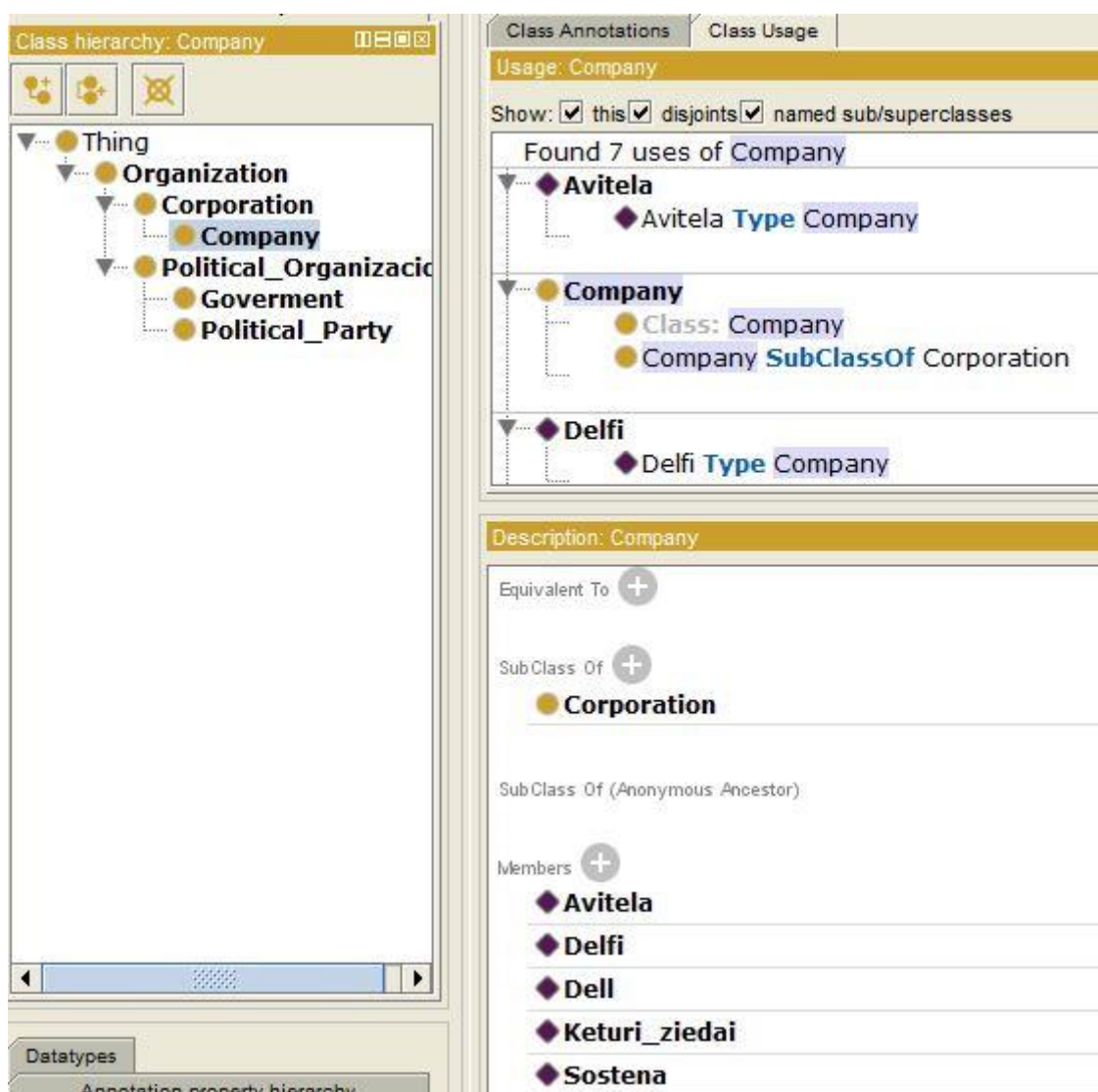
- *Protege* – atviro kodo ir nemokamai platinamas ontologijų kūrimo įrankis, kuris leidžia atlikti ontologijų importavimo bei jungimo funkcijas, tačiau tai nėra visiškai priimtinas būdas dviejų ontologijų jungimui.
- *TopBride Composer* – komerciniais tikslais sukurtas įrankis [12]. Tačiau turi ir nemokamą versiją, kuri palaiko *OWL* bei *RDF* formatus.

Sekančiuose skyriuose bus apžvelgiami minėti įrankiai. Apžvalgos metu bus nuspręsta kurį ontologijų kūrimo įrankį pasirinkti tolimesniems tyrimams.

2.5.1 Protege ontologijų kūrimo įrankis

Atlikus įrankių tinkamų ontologijos kūrimui analizę, pasirinktas nemokamas atviro kodo *Protege* programinis įrankis, kuris sukurtas ir vystomas Stanfordo universiteto (JAV) biomedicinos informatikos fakulteto bendruomenės.

Protege yra grafinis *OWL* kalbos redaktorius, leidžia sukurti ontologiją nuo pirmųjų žingsnių, kai tik formuojama hierarchija. Be to, galima vystyti jau sukurtą ontologiją, įvedant papildomus apribojimus bei ryšius [5]. *Protege* įrankis leidžia sulieti (angl. *merge*) ir importuoti dvi ontologijas. Importuojant vieną ontologiją į kitą, sukuriama trečia ontologija, kuri turi abiejų ontologijų elementus ir savybes. Vykdam importavimą, tų pačių klasių duomenys yra dubliuojami ir tai neleidžia sukurti vientisos ontologijos, kuri būtų paruošta naudojimui.



8 pav. *Protege* programos vartotojo sąsaja

8 paveiksle matome *Protege* programos langu, kuriuose sukuriamos klasės ir jų įrašai. Taip pat sukuriamos įrašų savybės bei pateikiama informacija apie sukurtas klases ir jų įrašus. *Protege* įrankis yra patogus ontologijų kūrimui. Sukūrus ontologiją galima vizualiai pažiūrėti jos hierarchiją.

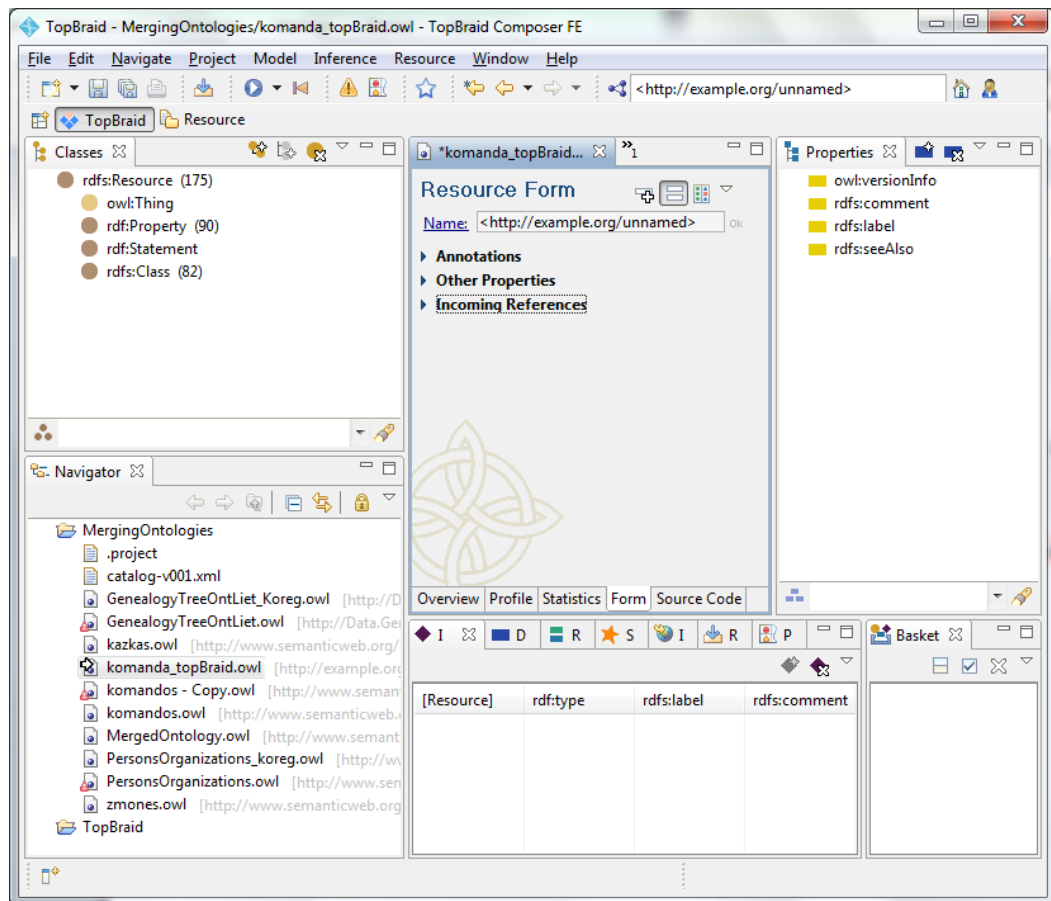
Naudojantis suliejimo (angl. *merge*) funkcija sukuriamą ontologiją papildoma naujais įrašais nedubliuojant esamų. Ši funkcija vykdoma automatiškai, tačiau norint susieti dvi ontologijas reikia rankiniu būdu suvienodinti ontologijų *IRI* adresus, kurie yra naudojami identifikuoti suliejamus ontologijų elementus.

Taip pat šis įrankis turi ontologijų importavimo funkciją. Ši funkcija į turimą ontologiją importuoja pasirinktą ontologiją. Importavimo metu bazinėje ontologijoje sukuriama visos klasės ir aprašai iš importuojamos ontologijos. Jei importuojama ontologija turi vienoda klases, vis tiek sukuriamą naują tokia pat klasė, tokiu atveju atsiranda duomenų dubliavimas.

Ontologijos gali būti eksportuojamos įvairiais formatais, įskaitant *RDF(S)* ir *OWL*. Yra galimybė ontologiją peržiūrėti vizualiai. Dėl savo lanksčios architektūros *Protege* turi daug papildinių, kurie išplečia funkcionalumą. Vienas toks yra *PROMP* papildinys, kuris leidžia sujungti dvi ontologijas.

2.5.2 *TopBride Composer* ontologijų kūrimo įrankis

Dar vienas ontologijų kūrimo įrankis yra *TopBride Composer*. Šis įrankis savo savybėmis panašus į *Protege*. *TopBride Composer* pagalba galime kurti ontologijas įvairiais formatais, taip pat ir *OWL/RDF*. Šis įrankis yra komercinis. Jį platina ir plėtoja *TopQuadrant, Inc.*[13] Už atitinkamą kainą galima įsigyti skirtingas programos versijas, kurios turi papildomas funkcijas. Tačiau galima rinktis ir nemokamą versiją, kuri leidžia kurti ontologijas *OWL* formatu. Tokia versija ir buvo pasirinkta analizei.



9 pav. *TopBride Composer* pagrindinis langas

Ontologijų kūrimas panašus į *Protege* siūlomas galimybes. Tačiau šis įrankis turi didesnes galimybes detalizuoti klasių savybes. Šis įrankis yra mokamas ir palaikomas komerciniais tikslais todėl, *TopBride Composer* turi platesnes galimybes darbui su ontologijomis. Įrankio pagalba galima atlikti ontologijų sujungimą naudojantis importavimo funkcija.

Bandant šį įrankį buvo naudojama siūloma importavimo (angl. *import*) funkcija. Nagrinėjant importavimo galimybes buvo pastebėta, jog ši funkcija nesiskiria nuo *Protege* siūlomo funkcionalumo. Šis įrankis yra specializuotas ir plačiai pritaikytas ontologijų kūrimui. *TopBride Composer* įrankis yra pritaikytas komandiniam darbui, t.y. įrankis turi tokias galimybes kaip *SVN* duomenų versijavimo kontrolę bei komandinio darbo galimybes *Team Synchronizing*.

Kadangi bandytas įrankis yra komercinis ir yra ribotos galimybės naudotis nemokamai, jis nėra atviro kodo ir nėra galimybės jo tobulinti, tai jis tolimesniuose tyrimuose nebus detalčiau nagrinėjamas.

2.6 Darbo tikslas ir uždaviniai

Tyrimo sritis yra ontologijų inžinerija t.y. ontologijų modulinio kūrimo bei jungimo principų analizė bei metodologijos rengimas praktiškai pritaikant egzistuojančius sprendimus ontologijų kūrimui bei jungimui. Taikant šių įrankių galimybes, turi būti galima automatiškai sujungti dvi

pasirinktas ontologijas taip, kad nenukentėtų gautos ontologijos kokybė, t. y., sujungus nesidubliuotų įrašai bei nebūtų prarasta pradinė informacija.

Tyrimo objektas – OWL 2 ontologijų modulinio kūrimo ir jungimo procesas.

Tyrimo tikslas – padidinti ontologijų kūrėjų galimybes kurti ir pakartotinai panaudoti modulines ontologijas sukuriant metodiką, kuri leistų pakartotinai panaudoti kuriamas ontologijas kitų ontologijų kūrime.

Darbai aprobuoti siekiama parengti prototipinį ontologijų jungimo įrankį, kuris būtų realizuotas remiantis parengtos ontologijų jungimo metodikos principais.

Darbo uždaviniai:

6. Išanalizuoti:
 - 6.1. ontologijų kūrimo principus;
 - 6.2. ontologijų jungimo principus;
 - 6.3. egzistuojančius ontologijų kūrimo ir jungimo įrankius ir jų galimybes.
7. Remiantis atlikta analize, sudaryti ontologijų modulinio kūrimo ir jungimo metodiką, nusakančią jungimo taisykles ir apribojimus.
8. Realizuoti ontologijų jungimo įrankio prototipą, kuris leistų pasirinkti jungiamas klases, klasių egzempliorius bei savybes.
9. Atlikti eksperimentą, kuris leistų įvertinti metodikos tinkamumą ir pritaikymą realioms sprendimams.
10. Įvertinti tyrimo rezultatus.

2.7 OWL ontologijų kūrimo ir jungimo analizės apibendrinimas

1. Siekiant sukurti ontologiją, reikia apibrėžti dalykinę sritį, kurią analizuojant kuriamai ontologijai suformuluojami konkretūs reikalavimai, kuriuos galima apibrėžti kaip kompetencijos klausimus, į kuriuos kuriama ontologija turi atsakyti.
2. Siekiant taikyti modulinį ontologijų kūrimą, reikia turėti tam tikros struktūros ontologijas, kurios užtikrintų tinkamą kuriamos ontologijos struktūrą, kadangi nauja ontologija yra kuriama iš atskirų dalių.
3. Ontologijas sujungti galima dviem kompiuterizuotais būdais, t.y. suliejimo (angl. *merge*) ir importavimo (angl. *import*). Ontologijų jungimo metu galima sujungti dvi lokalias arba prijungti nutolusią ontologiją, t.y. galima naudoti ontologijas, kurios yra savo kompiuteryje ar nutolusiame serveryje.
4. Apžvelgti du įrankiai, leidžiantys kurti bei sujungti ontologijas. Šie įrankiai yra labiau orientuoti į ontologijų kūrimą, o ne į jų siejimą. Todėl jų siūlomos ontologijų jungimo

galimybės yra ribotos ir neužtikrina korektiško ontologijų sujungimo, t. y. , nagrinėti įrankiai netenkina ontologijų jungimo poreikių.

5. Abu apžvelgti įrankiai neužtikrina ontologijos vientisumo, t. y., tiek importavimo, tiek suliejimo funkcijomis atlikus ontologijų sujungimą nepavyksta išvengti vienodų duomenų dubliavimo. Tokiu atveju reikia papildomai analizuoti ir koreguoti naują ontologiją, o tai reikalauja papildomų darbo sąnaudų.

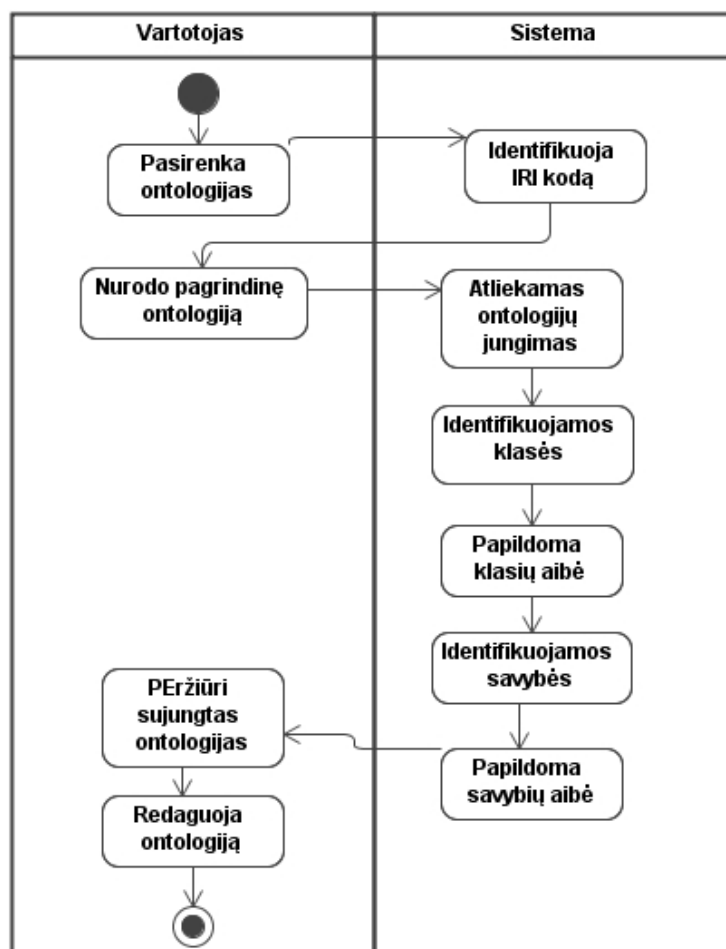
3. Ontologijų jungimo įrankio reikalavimai

Pritaikant sukurtą ontologijų jungimo ir modulinio kūrimo metodiką, galima realizuoti įrankį, kuris automatizuotų dalį veiksmų atliekant ontologijų sujungimą. Realizuojant turėtų būti sukurtas įrankis, kuris turėtų kelis aktyvius langus. Šiuose languose būtų pavaizduotos jungiamos ontologijos bei gautas rezultatas. Atliekant jungimo veiksmus, vedlys padėtų atlikti ontologijų klasių bei savybių susiejimą (angl. *mapping*) nurodant susiejamas ontologijų klases ir savybes. Pagal nurodytus susiejimo požymius sistema sukurtų naują ontologiją, sudarytą iš abiejų ontologijų.

3.1 Siekiamos sistemos apibrėžimas

Parengtos metodikos bei realizuoto sprendimo pagalba bus galima sujungti dvi ontologijas, t.y. bus galima sujungti dvi skirtingas ontologijas, kurios bus sudarytos iš skirtingų elementų. Taip pat bus galima sujungti ontologijas, kurios turės vienodus arba panašius elementus.

10 paveiksle pateiktoje procesų diagramoje pavaizduoti vartotojo ir sistemos atliekami veiksmai ontologijų jungimo metu.



10 pav. Veiklos procesų diagrama

Lentelė 4. Veiklos proceso aprašai

Veiksmas	Vykdytojas	Aprašymas
Pasirenka ontologijas	Naudotojas	Pasirenkamos ontologijos, kurių elementai bus naudojami ontologijų praplėtimui. Pasirenkamos ontologijos turi būti tinkamo formato ir struktūros.
Identifikuoja IRI kodą	Sistema	Sistema nuskaito parinktas ontologijas ir identifikuoja IRI kodus. Šie kodai reikalingi ontologijų suvienodinimui. Pagal tai bus susiejami ontologijų elementai.
Nurodoma pagrindinė ontologija	Naudotojas	Naudotojas nurodo pagrindinę ontologiją ir į ją talpina papildomus ontologijų elementus, kuriais yra praplečiama pasirinkta ontologija.
Atliekamas ontologijų jungimas	Sistema	Sistema susieja ontologijas pagal naudotojos nurodytus kriterijus. Siejimo metu sistema susieja vienodus elementus bei jei reikia sukuria naujus ontologijos elementus.
Identifikuojamos klasės	Sistema	Sistema analizuoja ontologijas ir identifikuoja jų klases, kurios bus susiejamos arba kuriamos naujoje ontologijoje.
Papildomas klasių aibė	Sistema	Sistema papildo pagrindinės ontologijos klasių aibę naujomis klasėmis, kurios yra perkeliamos iš kitos ontologijos arba sukuriamos naujos.
Identifikuojamos savybės	Sistema	Sistema analizuoja ontologijas ir identifikuoja jų klasių savybes, kurios bus perkeliamos arba kuriamos naujoje ontologijoje.
Papildoma savybių aibė	Sistema	Sistema papildo pagrindinės ontologijos klasių savybių aibes naujomis savybėmis, kurios yra perkeliamos iš kitos ontologijos arba sukuriamos naujos.
Peržiūrėti sujungtas ontologijas	Naudotojas	Naudotojas peržiūri sujungtas ontologijas taip jas patikrindamas ar visi jungimo veiksmai buvo sėkmingai atlikti. Jei jungimas buvo atliktas gerai, naudotojas išsaugo ontologiją.
Redaguoti ontologiją	Naudotojas	Esant reikalui naudotojas atlieka ontologijos redagavimo darbus ir išsaugo ontologiją.

Kuriamas prototipas veiks atsižvelgdamas į parengtą ontologiju jungimo metodiką. Pagal nustatytus principus sistema vartotojui leis atlikti ontologijų elementų kūrimą ir siejimą. Sistema

pagal parengtus apribojimus naudotojui leis atlikti tik tam tikrus veiksmus, kurie būti ontologijų jungimui siekiant sukurti korektišką ontologiją.

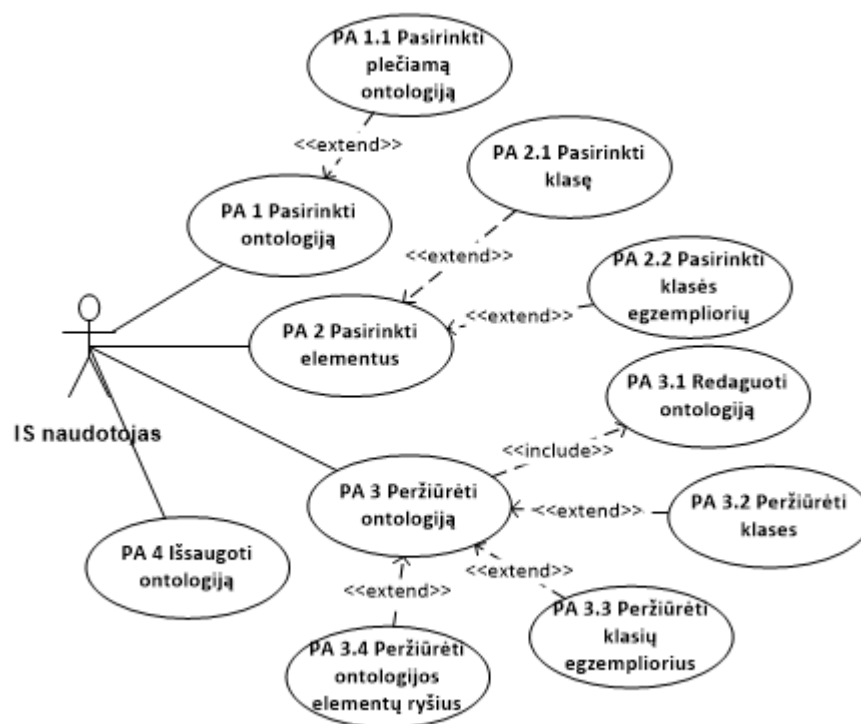
Sistema leis kurti ontologijas naudojant ontologijų modulinio kūrimo principus. Ontologijų kūrimo metu sistema iš kelių ontologijų naudotojui leis sukurti vieną didelę ontologiją. Kūrimui sistema leis naudoti tik tam tikros struktūros ontologijas, kurios užtikrintų tinkamą kuriamos ontologijos struktūrą, nes nauja ontologija yra kuriama iš atskirų dalių siekiant užtikrinti tinkamą modulinį ontologijų kūrimą,

3.2 Ontologijų jungimo reikalavimų specifikacija ir analizė

Darbo metu bus sukurta metodika, kuria remiantis bus palengvintas ontologijų jungimas. Jungimo metu be didelio vargu bus susiejami dviejų ontologijų duomenys. Šiuo metu esantys įrankiai siūlo ontologijų jungimo galimybes, tačiau neužtikrina korektiško jų sujungimo. Parengta metodika apibrėš konkrečius veiksmus, kurių dėka bus jungiamos ontologijos išvengiant duomenų pasikartojimo ir užtikrins ontologijų korektiškumą ir tikslumą.

Pagrindinis tikslas yra detalizuoti veiksmus ir jų seką, kad atliekant paeiliui konkrečius veiksmus būtų sujungiamos dvi ontologijos. Turi būti išanalizuotas ir detalizuotas kiekvienas veiksmas ir jo rezultatas, kad būtų galima apibrėžti ontologijų jungimo metodiką.

3.2.1 Panaudos atvejai



11 pav. Panaudos atvejų diagrama

3.2.1.1 PA 1 Pasirinkti ontologiją

Panaudojimo atvejo pavadinimas	PA 1 Pasirinkti ontologiją
Susiję panaudojimo atvejai	PA 1.1 Pasirinkti plečiamą ontologiją
Santrauka	Sistemos naudotojas apsirenka dvi ontologijas, kurios bus jungiamos. Iš pasirinktų ontologijų bus sukurta nauja ontologija.
Naudotojų grupės	IS naudotojas
Pirminiai reikalavimai	Turi būti atsidaryta ontologijų jungimo programa. Turi būti sukurti bent du ontologijų failai.
Pagrindinis scenarijus:	
1. Naudotojas spaudžia [Pasirinkti ontologiją].	
2. Atsidariusiame lange naudotojas pasirenka norimą ontologiją.	
3. Naudotojas spaudžia mygtuką [Gerai].	
4. Sistema nuskaito parinktą ontologiją.	
5. Pabaiga	
Rezultatai	Į sistemą įkeliamas ontologijos failas, kuris naudojamas jungimo veiksmams.
Alternatyvūs scenarijai	
<i>Pasirinktas netinkamo formato failas</i>	
1. Vykdomi pirmi 3 pagrindinio scenarijaus žingsniai.	
2. Sistema parodo klaidos pranešimą „ <i>Pasirinktas neteisingo formato failas</i> “.	
3. Sistema atveria ontologijos pasirinkimo langą.	
4. Vykdomas pagrindinio scenarijaus 3 žingsnis.	
5. Pabaiga.	
<i>Pasirinkta ontologija turi cikliškus ryšius</i>	
1. Vykdomi pirmi 3 pagrindinio scenarijaus žingsniai.	
2. Sistema parodo klaidos pranešimą „ <i>Pasirinkta ontologija turi cikliškus ryšius</i> “.	
3. Sistema atveria ontologijos pasirinkimo langą.	
4. Vykdomas pagrindinio scenarijaus 3 žingsnis.	
5. Pabaiga.	
Pastabos	Pasirinkimo metu pasirenka

3.2.1.2 PA 1.1 Pasirinkti plečiamą ontologiją

Panaudojimo atvejo pavadinimas	PA 1.1 Pasirinkti plečiamą ontologiją
Susiję panaudojimo atvejai	PA 1 Pasirinkti ontologiją
Santrauka	Sistemos naudotojas pasirenka ontologiją, kuri bus praplečiama, t.y. prie šios pasirinktos ontologijos bus prijungiami kitos ontologijos elementai.
Naudotojų grupės	IS naudotojas
Pirminiai reikalavimai	Turi būti pasirinktas jungiamos ontologijos failas.
Pagrindinis scenarijus:	

Panaudojimo atvejo pavadinimas	PA 1.1 Pasirinkti plečiamą ontologiją
	1. Naudotojas spaudžia [Pasirinkti plečiamą ontologiją].
	2. Atsidariusiame lange naudotojas pasirenka norimą ontologiją.
	3. Naudotojas spaudžia mygtuką [Gerai].
	4. Sistema nuskaito parinktą ontologiją.
	5. Pabaiga.
Rezultatai	Pasirinkta ontologija praplečiama naujais elementais, savybėmis.
Alternatyvūs scenarijai	
	1. <i>Pasirinktas netinkamo formato failas</i>
	2. Vykdomi pirmi 3 pagrindinio scenarijaus žingsniai.
	3. Sistema parodo klaidos pranešimą „ <i>Pasirinkta ontologija turi ciklišių ryšių</i> “.
	4. Sistema atveria ontologijos pasirinkimo langą.
	5. Vykdomas pagrindinio scenarijaus 3 žingsnis.
	6. Pabaiga.
Pastabos	Pasirinktos ontologijos pagrindu bus sukuriama nauja praplėsta ontologija.

3.2.1.3 PA 2 Pasirinkti elementus

Panaudojimo atvejo pavadinimas	PA 2 Pasirinkti elementus
Susiję panaudojimo atvejai	PA 2.1 Pasirinkti klasę PA 2.2 Pasirinkti klasės egzempliorių
Santrauka	Sistemos naudotojas pasirenka ontologijų elementų tipą, kuriais vėliau bus praplėsta pasirinkta ontologija.
Naudotojų grupės	IS naudotojas
Pirminiai reikalavimai	Turi būti pasirinktos dvi ontologijos.
Pagrindinis scenarijus:	
	1. Sistemos naudotojas pasirenka ontologijos elementų tipą.
	2. Sistema pateikia ontologijos elementų sąrašą.
	3. Pabaiga.
Rezultatai	Nurodomi elementų tipas, kuriais bus praplečiama pasirinkta ontologija.
Alternatyvūs scenarijai	
Nėra	
Pastabos	Pasirinktą ontologiją galima praplėsti klasėmis, egzemplioriais bei sukurti ryšius tarp jų.

3.2.1.4 PA 2.1 Pasirinkti klasę

Panaudojimo atvejo pavadinimas	PA 2.1 Pasirinkti klasę
Susiję panaudojimo atvejai	PA 2 Pasirinkti elementus
Santrauka	Sistemos naudotojas pasirenka norimas klases, kurios bus perkeliomos į kitą ontologiją. Tokiu būdu bus praplėsta pasirinkta ontologija.
Naudotojų grupės	IS naudotojas
Pirminiai reikalavimai	Turi būti pasirinktos dvi ontologijos. Turi būti atidarytas ontologijų klasių sąrašas.
Pagrindinis scenarijus:	
1. Sistemos naudotojas spaudžia mygtuką [Klasės].	
2. Sistema atvaizduoja visas ontologijos klases.	
3. Sistemos naudotojas pasirenka norimas klases ir jas perkelia į kitą ontologiją.	
4. Sistema atvaizduoja perkeltas klases naujoje ontologijoje.	
5. Pabaiga.	
Rezultatai	Nurodomos perkeliomos klasės.
Alternatyvūs scenarijai	
<i>Tokios klasės pavadinimas jau yra</i>	
1. Vykdomas pagrindinio scenarijaus 3 žingsnis.	
2. Sistema pateikia elementų redagavimo langą.	
3. Sistemos naudotojas pervadina norimas klases.	
4. Sistemos naudotojas spaudžia mygtuką [Gerai].	
5. Sistema sukuria naujas klases ontologijoje.	
6. Sistema atvaizduoja klases ontologijoje.	
7. Pabaiga.	
<i>Elementų redagavimo atšaukimas</i>	
1. Vykdomas <i>Tokios klasės pavadinimas jau yra</i> scenarijaus 2 žingsnis.	
2. Sistemos naudotojas spaudžia mygtuką [Atšaukti].	
3. Sistema uždaro elementų redagavimo langą.	
4. Pabaiga.	
Pastabos	Nurodyti galima tiek klases, tiek jų aibes, t.y. klases su visais poklasiais.

3.2.1.5 PA 2.2 Pasirinkti klasės egzempliorių

Panaudojimo atvejo pavadinimas	PA 2.2 Pasirinkti klasės egzempliorių
Susiję panaudojimo atvejai	PA 2 Pasirinkti elementus

Panaudojimo atvejo pavadinimas	PA 2.2 Pasirinkti klasės egzempliorių
Santrauka	Sistemos naudotojas pasirenka klasių egzempliorius, kuriais bus papildyta plečiama ontologija.
Naudotojų grupės	IS naudotojas
Pirminiai reikalavimai	Turi būti pasirinktos dvi ontologijos. Turi būti atidarytas egzempliorių sąrašo langas.
Pagrindinis scenarijus:	
1. Sistemos naudotojas spaudžia mygtuką [Egzemplioriai].	
2. Sistema atvaizduoja visas ontologijos klasių egzempliorius.	
3. Sistemos naudotojas pasirenka norimas klasės egzempliorius ir juos perkelia į kitą ontologiją.	
4. Sistema atvaizduoja perkeltus klasės egzempliorius naujoje ontologijoje.	
5. Pabaiga.	
Rezultatai	Nurodomi perkeliama klasių egzemplioriai.
Alternatyvūs scenarijai	
<i>Tokios klasės egzempliorių pavadinimas jau yra</i>	
1. Vykdomas pagrindinio scenarijaus 3 žingsnis.	
2. Sistema pateikia elementų redagavimo langą.	
3. Sistemos naudotojas pervadina norimus klasių egzempliorius.	
4. Sistemos naudotojas spaudžia mygtuką [Gerai].	
5. Sistema sukuria naujus klasės egzempliorius ontologijoje.	
6. Sistema atvaizduoja klasės egzempliorius ontologijoje.	
7. Pabaiga.	
<i>Elementų redagavimo atšaukimas</i>	
1. Vykdomas <i>Tokios klasės pavadinimas jau yra</i> scenarijaus 2 žingsnis.	
2. Sistemos naudotojas spaudžia mygtuką [Atšaukti].	
3. Sistema uždaro elementų redagavimo langą.	
4. Pabaiga.	
Pastabos	Perkeliant egzempliorius perkeliama tik konkretūs ontologijos elementai be ryšių.

3.2.1.6 PA 3 Peržiūrėti ontologiją

Panaudojimo atvejo pavadinimas	PA 3 Peržiūrėti ontologiją
Susiję panaudojimo atvejai	PA 3.1 Redaguoti ontologiją PA 3.2 Peržiūrėti klases PA 3.3 Peržiūrėti klasių egzempliorius PA 3.4 Peržiūrėti ontologijos elementų ryšius
Santrauka	Sistemos naudotojas atlikęs jungimo veiksmus gali peržiūrėti sujungtą ontologiją.
Naudotojų grupės	IS naudotojas

Panaudojimo atvejo pavadinimas	PA 3 Peržiūrėti ontologiją
Pirminiai reikalavimai	Turi būti pasirinktos dvi ontologijos. Turi būti įvykdyti jungimo veiksmai. Turi būti išsaugota ontologija.
Pagrindinis scenarijus:	
1. Sistemos naudotojas spaudžia mygtuką [Peržiūrėti ontologiją].	
2. Sistema pateikia ontologiją naujame lange.	
3. Sistemos naudotojas spaudžia mygtuką išskleisti viską.	
4. Sistema parodo pilną ontologijos medį.	
5. Pabaiga.	
Rezultatai	Sistemos lange pateikiama nauja ontologija medžio pavidalu.
Alternatyvūs scenarijai	
<i>Uždaryti peržiūros langą</i>	
1. Vykdomas pagrindinio scenarijaus 2 žingsnis.	
2. Sistemos naudotojas spaudžia mygtuką [Uždaryti].	
3. Sistema uždaro ontologijos peržiūros langą.	
4. Pabaiga.	
Pastabos	Nėra.

3.2.1.7 PA 3.1 Redaguoti ontologiją

Panaudojimo atvejo pavadinimas	PA 3.1 Redaguoti ontologiją
Susiję panaudojimo atvejai	PA 3 Peržiūrėti ontologiją
Santrauka	Sistemos naudotojas gali atlikti sujungtos ontologijos redagavimą, t.y. galima papildyti klases egzemplioriais bei sukurti atitinkamus ryšius.
Naudotojų grupės	IS naudotojas
Pirminiai reikalavimai	Turi būti sujungta ontologija. Turi būti atidarytas ontologijos elementų peržiūros langas.
Pagrindinis scenarijus:	
1. Sistemos naudotojas pasirenka norimų ontologijos elementų tipą.	
2. Sistema pateikia tam tikro tipo elementų sąrašą.	
3. Sistemos naudotojas atlieka koregavimo veiksmus	
4. Priskiria elementus kitiems ontologijos elementams.	

Panaudojimo atvejo pavadinimas	PA 3.1 Redaguoti ontologiją
	5. Ištrina nereikalingus elementus.
	6. Sukuria reikiamus elementų ryšius.
	7. Sistema atvaizduoja redaguotą ontologiją.
	8. Pabaiga.
Rezultatai	Koreguojama sujungta ontologija.
Alternatyvūs scenarijai	
Nėra	
Pastabos	Nėra.

3.2.1.8 PA 3.2 Peržiūrėti klases

Panaudojimo atvejo pavadinimas	PA 3.2 Peržiūrėti klases
Susiję panaudojimo atvejai	PA 3 Peržiūrėti ontologiją
Santrauka	Sistemos naudotojas gali peržiūrėti visas ontologijos turimas klases. Klases pateikiamos medžio pavidalu.
Naudotojų grupės	IS naudotojas
Pirminiai reikalavimai	Turi būti pasirinkta ontologija. Ontologija turi turėti bent vieną klasę.
Pagrindinis scenarijus:	
1. Sistemos naudotojas spaudžia mygtuką [Klasės].	
2. Sistema pateikia ontologijos klasių sąrašą.	
3. Pabaiga.	
Rezultatai	Sistemos lange pateikiamas ontologijos klasių medis.
Alternatyvūs scenarijai	
Ontologijoje nėra deklaruotų klasių	
1. Sistemos naudotojas spaudžia mygtuką [Klasės].	
2. Sistema pateikia tuščia ontologijos klasių sąrašą.	
3. Pabaiga.	
Pastabos	Nėra.

3.2.1.9 PA 3.3 Peržiūrėti klasių egzempliorius

Panaudojimo atvejo pavadinimas	PA 3.3 Peržiūrėti klasių egzempliorius
Susiję panaudojimo atvejai	PA 3 Peržiūrėti ontologiją

Panaudojimo atvejo pavadinimas	PA 3.3 Peržiūrėti klasių egzempliorius
Santrauka	Sistemos naudotojas gali peržiūrėti visų ontologijos turimų klasių egzempliorius. Klasių egzemplioriai pateikiami sugrupuoti pagal klases.
Naudotojų grupės	IS naudotojas
Pirminiai reikalavimai	Turi būti pasirinkta ontologija. Ontologija turi turėti bent vieną klasę. Ontologijos klasė turi turėti bent vieną egzempliorių.
Pagrindinis scenarijus:	
1. Sistemos naudotojas spaudžia mygtuką [Egzemplioriai].	
2. Sistema pateikia ontologijos klasių egzempliorių sąrašą.	
3. Pabaiga.	
Rezultatai	Sistemos lange pateikiamas klasių egzempliorių sąrašas sugrupuotas pagal klases.
Alternatyvūs scenarijai	
<i>Ontologijoje nėra deklaruotų klasių</i>	
1. Sistemos naudotojas spaudžia mygtuką [Egzemplioriai].	
2. Sistema pateikia tuščią ontologijos klasių egzempliorių sąrašą.	
3. Pabaiga.	
Pastabos	Pateikiami tik konkrečios ontologijos klasių egzemplioriai.

3.2.1.10 PA 3.4 Peržiūrėti ontologijos elementų ryšius

Panaudojimo atvejo pavadinimas	PA 3.4 Peržiūrėti ontologijos elementų ryšius
Susiję panaudojimo atvejai	PA 3 Peržiūrėti ontologiją
Santrauka	Sistemos naudotojas gali peržiūrėti visų ontologijos turimų klasių egzempliorių ryšius.
Naudotojų grupės	IS naudotojas
Pirminiai reikalavimai	Turi būti pasirinkta ontologija. Ontologija turi turėti susietų elementų.
Pagrindinis scenarijus:	
1. Sistemos naudotojas spaudžia mygtuką [Savybės].	
2. Sistema pateikia ontologijos ryšių sąrašą.	
3. Pabaiga.	
Rezultatai	Sistemos naudotojui pateikiamas ontologijos turimų ryšių sąrašas.
Alternatyvūs scenarijai	
<i>Ontologijoje nėra deklaruotų klasių</i>	
1. Sistemos naudotojas spaudžia mygtuką [Savybės].	
2. Sistema pateikia tuščią ontologijos ryšių sąrašą.	

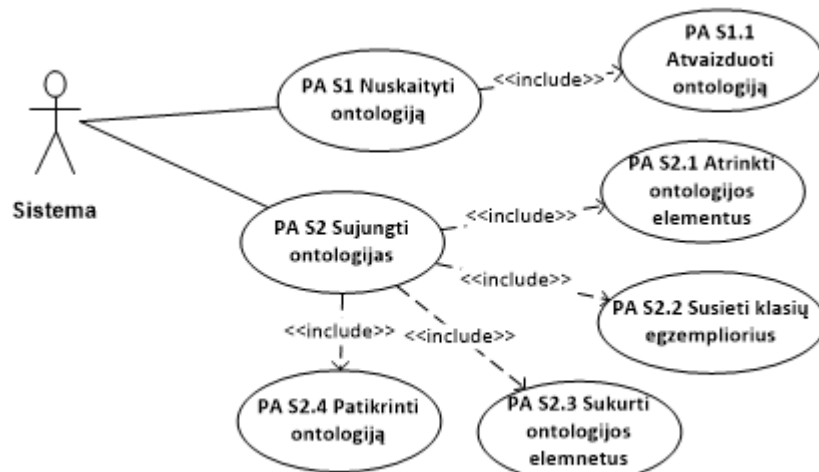
Panaudojimo atvejo pavadinimas	PA 3.4 Peržiūrėti ontologijos elementų ryšius
3. Pabaiga.	
<i>Sukurti savybę</i>	
1. Sistemos naudotojas spaudžia mygtuką [Pridėti savybę].	
2. Sistema atveria ryšių kūrimo langą.	
3. Sistemos naudotojas pasirenka du jungiamus elementus.	
4. Sistemos naudotojas pasirenka savybę.	
5. Sistemos naudotojas spaudžia mygtuką [Pridėti]	
6. Sistema sukuria ryšius tarp pasirinktų elementų.	
7. Pabaiga.	
<i>Savybės kūrimo atšaukimas</i>	
1. Vykdomas scenarijaus <i>Sukurti savybę</i> 2 žingsnis.	
2. Sistemos naudotojas spaudžia mygtuką [Atšaukti].	
3. Sistema uždaro savybių kūrimo langą.	
4. Pabaiga.	
Pastabos	Pateikiama tik konkrečios ontologijos elementų ryšių sąrašas.

3.2.1.11 PA 4 Išsaugoti ontologiją

Panaudojimo atvejo pavadinimas	PA 4 Išsaugoti ontologiją
Susiję panaudojimo atvejai	Nėra
Santrauka	Sistemos naudotojas atlikęs ontologijų jungimo veiksmus išsaugo naują ontologiją.
Naudotojų grupės	IS naudotojas
Pirminiai reikalavimai	Turi būti pasirinktos dvi ontologijos. Turi būti atlikti ontologijos jungimo veiksmai.
Pagrindinis scenarijus:	
1. Sistemos naudotojas spaudžia mygtuką [Saugoti ontologiją].	
2. Sistema pateikia failo saugojimo langą.	
3. Sistemos naudotojas nurodo kur išsaugoti failą.	
4. Sistemos naudotojas spaudžia mygtuką [Save].	
5. Sistema išsaugo sukurtą ontologiją.	
6. Pabaiga.	
Rezultatai	Naujame faile išsaugoma nauja ontologija.
Alternatyvūs scenarijai	
<i>Failo saugojimo atšaukimas</i>	
1. Vykdomas pagrindinio scenarijaus 2 žingsnis.	
2. Sistemos naudotojas spaudžia mygtuką [Cancel].	
3. Sistema uždaro failo saugojimo langą.	

Panaudojimo atvejo pavadinimas	PA 4 Išsaugoti ontologiją
4. Pabaiga.	
Pastabos	Išsaugota ontologija gali būti redaguojama kitomis programomis bei pakartotinai praplečiama.

3.2.2 Sisteminiai panaudos atvejai



12 pav. Sisteminiai panaudos atvejai

3.2.2.1 PA S1 Nuskaityti ontologiją

Panaudojimo atvejo pavadinimas	PA S1 Nuskaityti ontologiją
Susiję panaudojimo atvejai	PA S1.1 Atvaizduoti ontologiją
Santrauka	Sistema analizuoja sistemos naudotojo parinktą ontologijos failą ir jį išsaugo atmintyje.
Naudotojų grupės	Sistema
Pirminiai reikalavimai	Turi būti parinktas ontologijos failas.
Pagrindinis scenarijus:	
1. Sistema nuskaito sistemos naudotojo parinktą failą.	
2. Sistema atlieka failo analizę:	
a. Identifikuoja ontologijos klases t.y., atrenka <Class/> elementus, kurie yra aprašyti <Declaration></Declaration> sekcijoje.	
b. Identifikuoja sub-klases klases t.y., atrenka <Class/> elementus, kurie yra aprašyti <SubClass></SubClass> sekcijoje.	
c. Identifikuoja ontologijos klasių elementus t.y. atrenka <NameIndividual/> elementus, kurie yra aprašyti <Declaration></Declaration> sekcijoje.	
d. Identifikuoja konkrečių klasių turimus elementus t.y. atrenka <NameIndividual/> elementus, kurie yra aprašyti <ClassAssertion></ClassAssertion> sekcijoje.	

Panaudojimo atvejo pavadinimas	PA S1 Nuskaityti ontologiją
	e. Identifikuoja ontologijos turimus ryšius t.y., atrenka <ObjecProperties/> elementus, kurie yra aprašyti <Declaration></Declaration> sekcijoje.
	f. Identifikuoja konkrečius ryšius tarp ontologijos elementų t.y., atrenka <ObjecProperties/> elementus, kurie yra aprašyti <ObjectPropertiesAssertion></ObjectPropertiesAssertion> sekcijoje.
	3. Sistema išsaugo failą atmintyje.
	4. Pabaiga.
Rezultatai	Sistema paruošia failą ontologijų jungimui.
Alternatyvūs scenarijai	
<i>Parinktas neteisingas failas</i>	
	1. Sistema nuskaityto sistemos naudotojo parinktą failą.
	2. Sistema pateikia klaidos pranešimą „ <i>Pasirinktas neteisingo formato failas.</i> “.
	3. Pabaiga.
Pastabos	Nėra

3.2.2.2 PA S1.1 Atvaizduoti ontologiją

Panaudojimo atvejo pavadinimas	PA S1.1 Atvaizduoti ontologiją
Susiję panaudojimo atvejai	PA S1 Nuskaityti ontologiją
Santrauka	Sistema išanalizavusį ontologijos failą pateikia ekrane, kuriame atvaizduojami faile aprašyta ontologija.
Naudotojų grupės	Sistema
Pirminiai reikalavimai	Turi būti parinktas ontologijos failas. Turi būti išanalizuotas ontologijos failas.
Pagrindinis scenarijus:	
	1. Sistema nuskaityto failą.
	2. Sistema pateikia ontologiją medžio pavidalu.
	3. Pabaiga.
Rezultatai	Sistemos lange atvaizduojama ontologija. Ontologija pateikiama medžio pavidalu.
Alternatyvūs scenarijai	
Nėra	
Pastabos	Nėra

3.2.2.3 PA S2 Sujungti ontologijas

Panaudojimo atvejo pavadinimas	PA S2 Sujungti ontologijas
Susiję panaudojimo atvejai	PA S2.1 Atrinkti ontologijos elementus PA S2.2 Susieti klasių egzempliorius PA S2.3 Sukurti ontologijos elementus PA S2.4 Patikrinti ontologiją
Santrauka	Sistema sujungia dviejų ontologijų elementus, kuriuos nurodė sistemos naudotojas.
Naudotojų grupės	Sistema
Pirminiai reikalavimai	Turi būti parinktos dvi ontologijos. Turi būti išanalizuotos ontologijos. Turi būti parinkti elementai. Turi būti nurodyti ryšiai.
Pagrindinis scenarijus:	
1. Sistema atrenka nurodytus ontologijų elementus.	
2. Sistema patikrina plečiamą ontologiją.	
3. Sistema perkelia elementus į plečiamą ontologiją.	
4. Pabaiga.	
Rezultatai	Sujungiamos dvi ontologijos ir sukuriama viena bendra ontologija.
Alternatyvūs scenarijai	
<i>Rasta vienodų elementų</i>	
1. Vykdomas pagrindinio scenarijaus 2 žingsnis.	
2. Sistema randa vienodų ontologijų elementų.	
3. Sistema susieja vienodus elementus.	
4. Pabaiga.	
Pastabos	Nėra.

3.2.2.4 PA S2.1 Atrinkti ontologijos elementus

Panaudojimo atvejo pavadinimas	PA S2.1 Atrinkti ontologijos elementus
Susiję panaudojimo atvejai	PA S2 Sujungti ontologijas
Santrauka	Sistema analizuoja parinktą ontologiją ir atrenka sistemos naudotojo nurodytus ontologijos elementus.
Naudotojų grupės	Sistema
Pirminiai reikalavimai	Turi būti parinkta ontologija. Turi būti nuskaityta ontologija.
Pagrindinis scenarijus:	
1. Sistema analizuoja parinktą ontologijos failą.	

Panaudojimo atvejo pavadinimas	PA S2.1 Atrinkti ontologijos elementus
	2. Sistema atrenka nurodytus ontologijos elementus.
	3. Sistema atrinktus elementus išsaugo atmintyje.
	4. Pabaiga.
Rezultatai	Sistema atrenka reikiamus ontologijos elementus.
Alternatyvūs scenarijai	
Nėra	
Pastabos	Naudotojas gali nurodyti sistemai atrinkti ontologijos klases, subklases bei jų egzempliorius ir ryšius.

3.2.2.5 PA S2.2 Susieti klasių egzempliorius

Panaudojimo atvejo pavadinimas	PA S2.2 Susieti klasių egzempliorius
Susiję panaudojimo atvejai	PA S2 Sujungti ontologijas
Santrauka	Sistema susieja nurodytus ontologijų klasių turimus egzempliorius. Siejami tik tie klasių egzemplioriai, kuriuos nurodė sistemos naudotojas.
Naudotojų grupės	Sistema
Pirminiai reikalavimai	Turi būti parinktos dvi ontologijos. Turi būti nurodyti jungiamų klasių egzemplioriai.
Pagrindinis scenarijus:	
	1. Sistema atrenka nurodytus elementus.
	2. Sistema susieja elementus nurodytu ryšiu.
	3. Sistema papildo ontologija naujomis savybėmis.
	4. Pabaiga.
Rezultatai	Susiejami dviejų skirtingu ontologijų klasių egzemplioriai.
Alternatyvūs scenarijai	
<i>Nenurodyti ryšiai</i>	
	1. Sistema atrenka nurodytus elementus.
	2. Sistema sukuria naujus ryšius.
	3. Sistema susieja elementus sukurtais ryšiais.
	4. Sistema papildo ontologija naujomis savybėmis.
	5. Pabaiga.
Pastabos	Vienu metu sujungiama tik du egzemplioriai.

3.2.2.6 PA S2.3 Sukurti ontologijos elementus

Panaudojimo atvejo pavadinimas	PA S2.3 Sukurti ontologijos elementus
Susiję panaudojimo atvejai	PA S2 Sujungti ontologijas
Santrauka	Sistema sukuria naujus ontologijų elementus, kuriu reikia ontologijos sujungti.
Naudotojų grupės	Sistema
Pirminiai reikalavimai	Turi būti pasirinktos dvi ontologijos. Ontologijoje neturi būti tokio elemento.
Pagrindinis scenarijus:	
1. Sistema patikrina ontologiją.	
2. Sukuria ontologijos elemento aprašą.	
3. Sistema užpildo aprašą duomenimis.	
4. Pabaiga.	
Rezultatai	Kuriamoje ontologijoje sistema sukuria naujus elementus.
Alternatyvūs scenarijai	
<i>Elementas jau egzistuoja</i>	
1. Sistema patikrina ontologiją.	
2. Sistema randa vienodą elementą.	
3. Sistema pateikia klaidos pranešimą „Toks elementas jau egzistuoja“.	
4. Pabaiga.	
Pastabos	Neleidžiama sukurti dviejų ontologijos elementų vienodais pavadinimais.

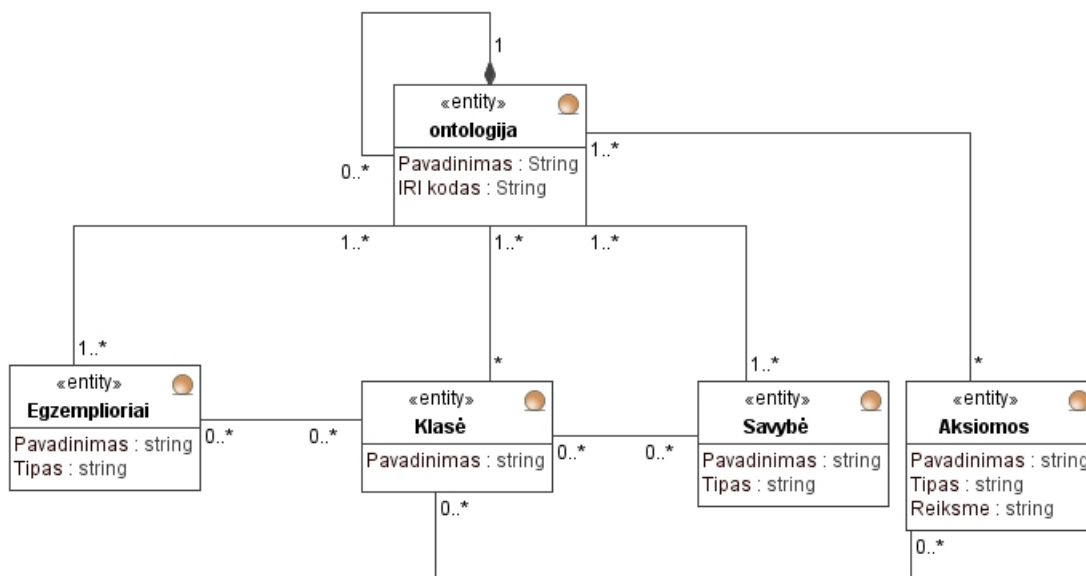
3.2.2.7 PA S2.4 Patikrinti ontologiją

Panaudojimo atvejo pavadinimas	PA S2.4 Patikrinti ontologiją
Susiję panaudojimo atvejai	PA S2 Sujungti ontologijas
Santrauka	Sistema patikrina visą ontologiją ar nėra vienodų elementų ir ar nėra blogai atliktų jungimo veiksmų.
Naudotojų grupės	Sistema
Pirminiai reikalavimai	Turi būti sujungtos dvi ontologijos
Pagrindinis scenarijus:	
1. Sistema analizuoja sujungtą ontologiją.	
2. Sistema tikrina ar nėra vienodų elementų.	
3. Sistema patikrina ar nėra vienodų ryšių.	
4. Pabaiga.	
Rezultatai	Patikrinama ar teisinga ontologija. Kitu atveju pateikiamas klaidos pranešimas.

Panaudojimo atvejo pavadinimas	PA S2.4 Patikrinti ontologiją
Alternatyvūs scenarijai	
<i>Rasta vienodų elementų</i>	
1. Vykdomas pagrindinio scenarijaus 2 žingsnis.	
2. Sistema randa vienodus elementus.	
3. Sistema pateikia klaidos pranešimą.	
4. Pabaiga.	
<i>Rasta vienodų ryšių</i>	
1. Vykdomas pagrindinio scenarijaus 3 žingsnis.	
2. Sistema randa vienodus ryšius.	
3. Sistema pateikia klaidos pranešimą.	
4. Pabaiga.	
Pastabos	Nėra.

3.3 Dalykinės srities modelis

Šiame skyriuje yra apibrėžiamas ontologijos meta modelis. Remiantis šiuo modeliu detalizuojami ontologijų jungimo veiksmai, kurie nusako kokie elementai galimi kiekvienoje ontologijoje ir kokie ontologijos elementai turi būti jungiami. Norinti sujungti ontologijas reikia identifikuoti metamodelyje matomus ontologijos elementus, t.y. egzempliorius, klases, savybes ir aksiomas.



13 pav. Ontologijos metamodelis

3.4 Rizikos faktorių analizė

Ontologijų jungimo metu gali būti blogai suprasta ontologijų panaudojimo dalykinė sritis. Susietų ontologijų redagavimas gali iškreipti jungiamų ontologijų dalykinę sritį, kadangi jungimo metu įvykus klaidai gali būti sujungiami duomenys aprašantys skirtingas klases ar savybes. Tokiu būdu gali atsirasti loginių klaidų.

Visi netikslumai atsiranda dėl ontologijų pavadinimų ar aprašymų, kadangi tą pačią reikšmę turintys elementai skirtingose ontologijose gali būti aprašomi arba vadinami skirtingai.

Taip pat kyla problemų su ontologijų elementų turimomis aksiomomis, kurios ontologijų jungimo metu nesujungiamos ar papildomos, todėl jas reikia realizuoti rankiniu būdu. Tokiu būdu gali kilti dėl ontologijų validavimo. Dėl žmogiškojo faktoriaus gali būti taip, kad bus realizuotos ne visos jungiamų ontologijų turimos aksiomos, dėl kurių gali nukentėti ontologijos korektiškumas.

Ryšiai tarp ontologijos klasių egzempliorių gali būti realizuojami klaidingai arba gali būti taip, kad jie visai nebus sukurti ar perkelti sujungiant ontologijas. Tokiu atveju reikalingas papildomas ontologijos tikrinimas arba ontologijos elementų jungimas turi būti vykdomas žingsniais jungiant vieną elementą po kito.

4. Ontologijų modulinio kūrimo ir jungimo metodika

1. Dalykinės srities specifikuojimas – kuriant ontologiją reikia specifikuoti konkrečią dalykinę sritį. Kadangi modulinio kūrimo principas yra ontologijos kūrimas iš atskirų dalių t.y. ontologijų. Specifikavus konkrečias dalykines sritis galima jas apjungti ir sukurti naują ontologiją, kuri apimtų kitą dalykinę sritį arba kelias dalykines sritis.
2. Ontologijos sukūrimas – specifikuojus dalykinę sritį galima kurti ontologiją, kuri aprašys dalykinės srities taisykles.
 - 2.1. Ontologijoje turi būti tik su ta dalykine sritimi susiję duomenys bei informacija.
 - 2.2. Ontologijoje klasių hierarchija turi būti kuriama medžio pavidalu, t. y., reikia išskirti pagrindines klases, kurių negalima išvesti iš kitų klasių.
 - 2.3. Kuriamos ontologijos klasės turi būti kuo labiau nepriklausomos nuo išorės. Tokiu būdu ontologijas bus galima pakartotinai panaudoti naujų ontologijų kūrimui.
 - 2.4. Savybės turi būti kuriamos paskutinėje ontologijų jungimo stadijoje, kadangi savybėmis nurodoma, kaip susisieja reikalingi ontologijų elementai. Savybė nurodo ryšį tarp dviejų ontologijos elementų, t. y., klasių egzempliorių.
3. Ontologijos tikrinimas – sukūrus ontologiją, reikia patikrinti ar ji tenkina šiuos kriterijus:
 - 3.1. **Minimalumas** – ontologija turi apimti minimalią ir būtiną klasių, savybių aibę, t. y., kuriama ontologija turi būti aprašyta panaudojus terminus ar žodynus, kad būtų galima suprasti, ką norima pavaizduoti ontologijoje [14].
 - 3.2. **Pilnumas** - ontologija turi apimti visą būtiną klasių, savybių ir aksiomų aibę, kuri leidžia atsakyti į kompetencijos klausimus. Sukurta ontologija turi būti suprantama ir vaizduoti visą informaciją, kuriai vaizduoti ir buvo kuriama ontologija [14].
 - 3.3. **Korektiškumas** – ontologijos turi būti kuriamos korektiškai aprašant ir detalizuojat klasių bei savybių aibes ir hierarchinius ryšius tarp sukurtų klasių [14].
 - 3.4. **Neprieštaringumas** – aksiomos turi būti neprieštaringos. Kuriamų ontologijų klasių ir savybių sąryšiai neturi prieštarauti vieni kitiems. Taip pat turi būti detalizuoti ryšiai tarp jų. Tai galima patikrinti pasinaudojus *Protege* analizės galimybe *Reasoning*. Ši funkcija detaliau bus apžvelgiama tolimesniuose tyrimo etapuose, kuomet bus rengiama ontologijų modulinio kūrimo metodika [14].
4. Dalykinės srities specifikacijos detalizavimas – specifikuojant vienos srities ontologiją reikia nustatyti, kokiomis ontologijomis ar jų dalimis bus detalizuojama ontologija. Tokiu atveju išvengiama perteklinės informacijos ontologijoje. Detalizavus reikiamus duomenis nesudėtingai galima juos paimti iš kitų ontologijų.

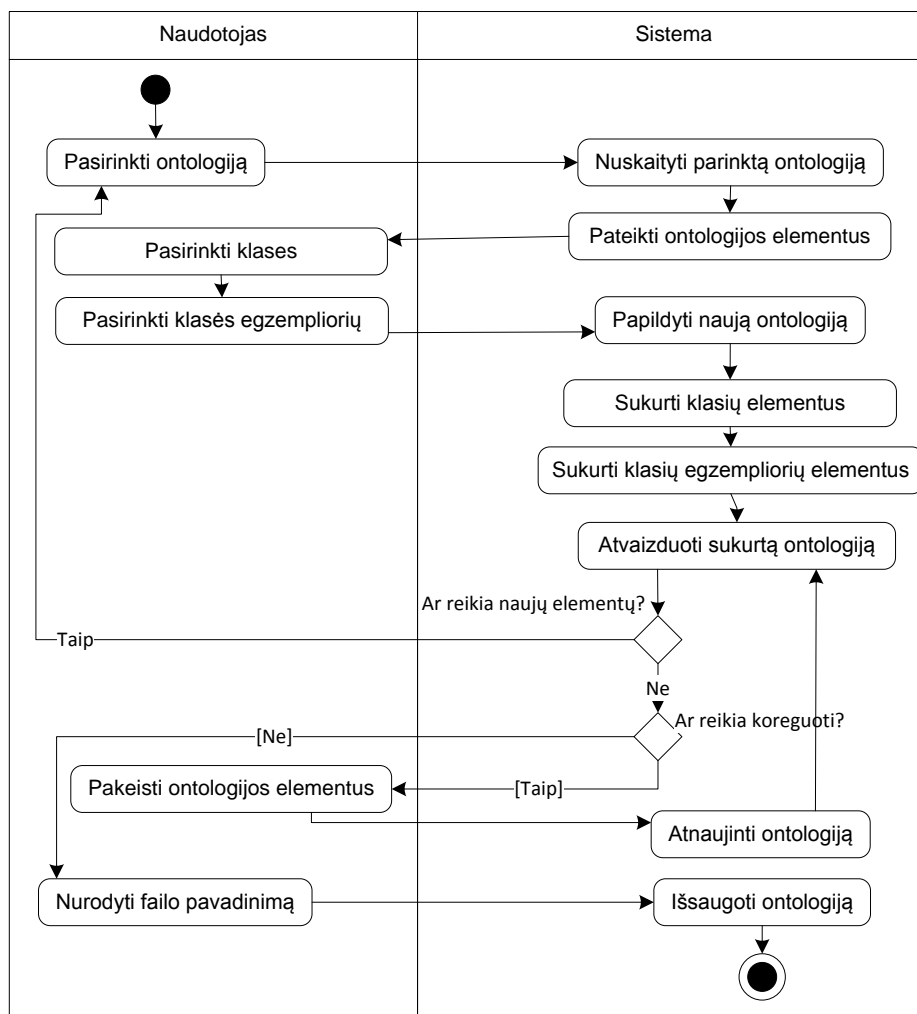
5. Ontologijų jungimas – jungiant ontologijas būtina turėti vienodos struktūros ontologijas. Ontologijos turi būti sukurtos remiantis 2 punkte aprašytais nurodymais. Sujungus ontologiją reikia papildomai atlikti jos tikrinimą, ar ji tenkina 3 punkte pateiktus kriterijus.

4.1 Ontologijų modulinio kūrimo procesas

Siekiant taikyti modulinį ontologijų kūrimą, reikia turėti tam tikros struktūros ontologijas, kurios užtikrintų tinkamą kuriamos ontologijos struktūrą. Tai turi būti padaryta, nes nauja ontologija yra kuriama iš atskirų dalių. Siekiant užtikrinti tinkamą modulinį ontologijų kūrimą, sudedamosios ontologijos ar jų dalys turi tenkinti šias sąlygas:

- Prijungiama ontologija negali turėti cikliškų ryšių. Ontologijos gali turėti ryšius tarp elementų tačiau, ontologijų modulinio kūrimo metu ryšiai turės būti kuriami iš naujo. Taip yra todėl kad, dalis ontologijų turimų ryšių gali būti nelogiški ir tokiu būdu bus klaidinga ontologija. Patartina modulinio ontologijų kūrimo metu naudoti ontologijas be ryšių.
- Ontologijoje turi būti aprašyta tik viena dalykinė sritis. Modulinio ontologijų kūrimo metu būtina naudoti ontologijas, kuriose būtų aprašyta tik viena dalykinė sritis. Kitu atveju tektų skaidyti ontologiją į smulkesnes ontologijas, o tai gali sukelti nesklandumų detalizuojant atskiras ontologijų dalis.
- Ontologija turi būti medžio struktūros. Naudojamos ontologijos turi būti medžio pavidalo t.y., ontologija gali turėti tik vieną viršūnę. Jei ontologijos medis turi daugiau viršūnių, patartina ontologiją skaidyti į mažesnes dalis.

Pateiktame ontologijų modulinio kūrimo veiklos procese (14 pav.) nurodoma, kokie veiksmai atlieka ontologijų kūrimo metu. Šio proceso metu iš atskirų ontologijos dalių ar ontologijų sukuriama viena ontologija.



14 pav. Ontologijų modulinio kūrimo veiklos procesas

Lentelė 5. Ontologijų modulinio kūrimo proceso aprašai

Veiksmas	Vykdytojas	Aprašymas
Pasirinkti ontologiją	Naudotojas	Sistemos naudotojas pasirenka ontologiją, kurios elementais bus praplečiama kita ontologija.
Nuskaityti parinktą ontologiją	Sistema	Analizuojamos ontologijos nuskaitant visus turimus elementus. Sistema apdoroja kiekvieną ontologijos elementą atskirai identifikuodama turimas savybes, ryšius bei aprašus. Sistema antrą ontologiją naudoja kaip pagrindą naujos ontologijos sukūrimui. Tokiu būdu ontologija papildoma reikiamais elementais ir savybėmis.
Pateikti ontologijos elementus	Sistema	Sistema atlikus pasirinktos ontologijos analizę pateikia ontologijos elementus sistemos naudotojui.

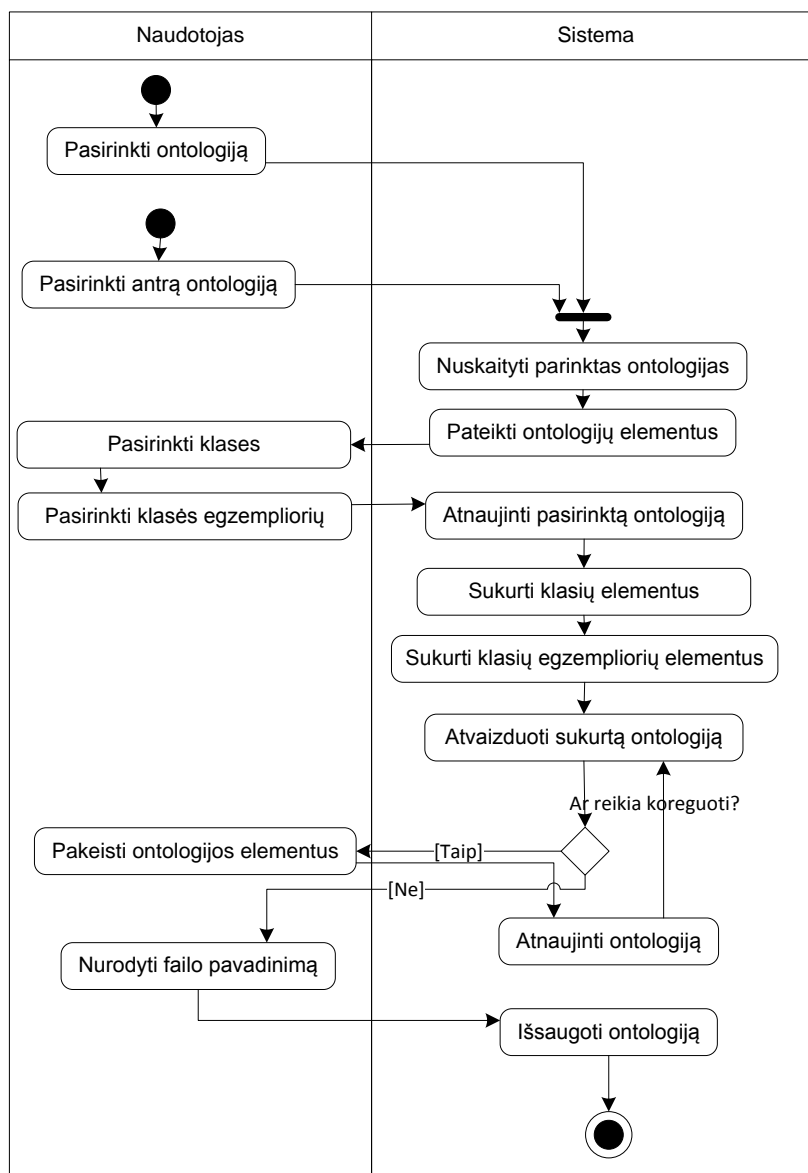
Veiksmas	Vykdytojas	Aprašymas
Pasirinkti ontologijos elementus	Naudotojas	Sistemos naudotojas nurodo kokius ontologijos elementus nori perkelti į jungiamą ontologiją. Taip pat nurodo kokius ryšius tarp elementų norima sukurti.
Papildyti naują ontologiją	Sistema	Sistema papildo kuriamą ontologiją sistemos naudotojo nurodytais elementais. Taip yra kuriama nauja ontologija iš atskirų ontologijos dalių.
Atvaizduoti sukurta ontologiją	Sistema	Pateikiama ontologija peržiūrai siekiant įsitikinti ar jungimo veiksmai buvo atlikti korektiškai. Jei jungimo metu buvo atlikti klaidingi veiksmai, tai galima koreguoti gautą ontologiją. Po kiekvienų atliktų koregavimo veiksmų turi būti tikrinama ontologija siekiant sukurti korektišką ontologiją.
Pakeisti klases	Naudotojas	Sistemos naudotojas nurodo ontologijos klases, kurios bus perkeliamos į naują ontologiją.
Pasirinkti klasės egzempliorių	Naudotojas	Sistemos naudotojas pasirenka klasės egzempliorius, kurie bus perkeliami į naują ontologiją.
Atnaujinti ontologiją	Sistema	Sistema pateikia naują ontologijos medį pagal atliktus koregavimo veiksmus.
Sukurti klasių elementus	Sistema	Sistema naujoje ontologijoje sukuria klasių elementus, kuriuos pasirinko sistemos naudotojas.
Sukurti klasių egzempliorių elementus	Sistema	Sistema naujoje ontologijoje sukuria klasių egzempliorių elementus, kuriuos pasirinko sistemos naudotojas.
Nurodyti failo pavadinimą	Naudotojas	Naudotojas išsaugodamas nurodo ontologijos pavadinimą. Nurodant ontologijos pavadinimą patartina, kad jis atspindėtų ontologijos apibrėžiamą dalykinę sritį.
Išsaugoti ontologiją	Sistema	Išsaugoma sukurta ontologija. Nauja ontologija yra saugoma remiantis pasirinktų ontologijų struktūra. Ontologija išsaugoma faile, kuris atitinka pasirinktų ontologijų formatą bei struktūrą.

4.2 Ontologijų jungimo procesas

Jungiant ontologijas būtina atsižvelgti į šiuos kriterijus ir apribojimus:

- Naujai sudaromoje ontologijoje negali būti vienodų elementų, t.y. atliekamas tikrinimas ar pasirinkti ontologijos elementai neegzistuoja jungiamoje ontologijoje.
- Sujungtos ontologijos turi būti saugomos *XML* formato faile bei atitikti *OWL 2* struktūrą.
- Prijungiama ontologija negali turėti cikliškų ryšių. Ontologijos gali turėti ryšius tarp elementų, tačiau ontologijų modulinio kūrimo metu ryšiai turės būti kuriami iš naujo. Taip yra todėl, kad dalis ontologijų turimų ryšių gali būti nelogiški ir tokiu būdu bus klaidinga ontologija. Patartina modulinio ontologijų kūrimo metu naudoti ontologijas be ryšių.
- Ontologijoje turi būti aprašyta tik viena dalykinė sritis. Modulinio ontologijų kūrimo metu būtina naudoti ontologijas, kuriose būtų aprašyta tik viena dalykinė sritis. Kitu atveju tekstą skaidyti ontologiją į smulkesnes ontologijas, o tai gali sukelti nesklandumų detalizuojant atskiras ontologijų dalis.
- Ontologija turi būti medžio struktūros. Naudojamos ontologijos turi būti medžio pavidalo t.y., ontologija gali turėti tik vieną viršūnę. Jei ontologijos medis turi daugiau viršūnių, tai patartina ontologiją skaidyti į mažesnes dalis.

Pateiktame veiklos procese (15 pav.) nurodoma, kokie veiksmai atlieka ontologijų sujungimo metu. Šio proceso metu iš dviejų pasirinktų ontologijų sukuriama viena ontologija.



15 pav. Ontologijų suliejimo veiklos procesas

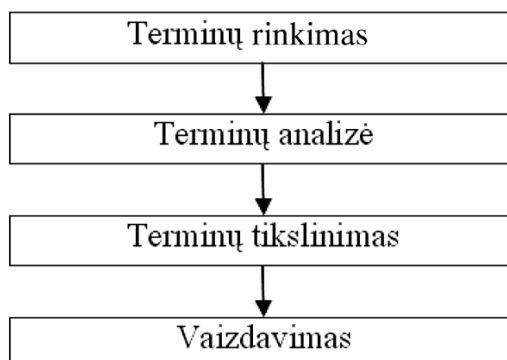
Lentelė 6. Ontologijų suliejimo proceso aprašai

Veiksmas	Vykdytojas	Aprašymas
Pasirinkti ontologiją	Naudotojas	Sistemos naudotojas pasirenka ontologiją, kurios elementais bus praplečiama kita ontologija.
Pasirinkti antrą ontologiją	Naudotojas	Sistemos naudotojas pasirenka antrą ontologiją, kuri bus plečiama norimais ontologijos elementais.
Nuskaityti parinktas ontologijas	Sistema	Analizuojamos ontologijos nuskaitant visus turimus elementus. Sistema apdoroja kiekvieną ontologijos elementą atskirai identifikuodama turimas savybes, ryšius bei aprašus. Sistema antrą ontologiją naudoja kaip pagrindą naujos ontologijos

Veiksmas	Vykdytojas	Aprašymas
		sukūrimui. Tokiu būdu ontologija papildoma reikiamais elementais ir savybėmis.
Pasirinkti ontologijos elementus	Naudotojas	Sistemos naudotojas nurodo kokius ontologijos elementus nori perkelti į jungiamą ontologiją. Taip pat nurodo kokius ryšius tarp elementų norima sukurti.
Atnaujinti pasirinktą ontologiją	Sistema	Sistema analizuoja parinktas ontologijas ir atlieka ontologijų jungimą. Sistema papildo plečiamos ontologijos elementų aibę naujais elementais bei ryšiais, kurie yra perkeliama iš kitos ontologijos arba sukuriama nauji.
Atvaizduoti sukurtą ontologiją	Sistema	Pateikiama ontologija peržiūrai siekiant įsitikinti ar jungimo veiksmai buvo atlikti korektiškai. Jei jungimo metu buvo atlikti klaidingi veiksmai, tai galima koreguoti gautą ontologiją. Po kiekvienų atliktų koregavimo veiksmų turi būti tikrinama ontologija siekiant sukurti korektišką ontologiją.
Pakeisti klases	Naudotojas	Sistemos naudotojas nurodo ontologijos klases, kurios bus perkeliama į naują ontologiją.
Pasirinkti klasės egzempliorių	Naudotojas	Sistemos naudotojas pasirenka klasės egzempliorius, kurie bus perkeliama į naują ontologiją.
Atnaujinti ontologiją	Sistema	Sistema pateikia naują ontologijos medį pagal atliktus koregavimo veiksmus.
Sukurti klasių elementus	Sistema	Sistema naujoje ontologijoje sukuria klasių elementus, kuriuos pasirinko sistemos naudotojas.
Sukurti klasių egzempliorių elementus	Sistema	Sistema naujoje ontologijoje sukuria klasių egzempliorių elementus, kuriuos pasirinko sistemos naudotojas.
Nurodyti failo pavadinimą	Naudotojas	Naudotojas išsaugodamas nurodo ontologijos pavadinimą. Nurodant ontologijos pavadinimą patartina, kad jis atspindėtų ontologijos apibrėžiamą dalykinę sritį.
Išsaugoti ontologiją	Sistema	Išsaugoma sukurta ontologija. Nauja ontologija yra saugoma remiantis pasirinktų ontologijų struktūra. Ontologija išsaugoma faile, kuris atitinka pasirinktų ontologijų formatą bei struktūrą.

5. Ontologijų jungimo metodikos pritaikymas

Šiuo metu nėra žinoma panašių sprendimų, kurie leistų sujungti dvi ontologijas, taip pat nėra parengtos konkrečios ontologijų jungimo metodikos, kuri nurodytų tikslus veiksmus ontologijų jungimo procese.



16 pav. Ontologijos kūrimo procesas

Paprastai ontologijos kūrimo procesą sudaro keturi etapai: terminų rinkimas, jų analizė, tikslinimas ir vaizdavimas. Šio proceso schema pavaizduota 16 paveiksle [16]. Terminų surinkimo etape yra identifikuojami visi nagrinėjamos srities vartojami terminai, jų tarpusavio ryšiai ir apibrėžimai. Analizės etape surinkti terminai semantiškai analizuojami: ieškoma skirtingų terminų, apibūdinančių vienodus objektus ar reiškinius. Radus šiuos terminus, turi būti pasiektas susitarimas dėl vieno termino bendro vartojimo. Tai daroma trečiajame ontologijos sudarymo etape. Peržiūrėjus visus tokius terminus pereinama prie paskutinio etapo – ontologijos vaizdavimo pasirinkta ontologijų vaizdavimo kalba. Paskutinis etapas gali būti įgyvendinamas taikant pasirinktą ontologijų kūrimo priemonę. Tai lemia ir ontologijos vaizdavimo kalbos pasirinkimą. Taikant ontologijų kūrimo priemones, ontologijas gali kurti žemesnės kvalifikacijos darbuotojai, pavyzdžiui, įmonės informacinių sistemų specialistai.

Magistrinio darbo metu bus siekiama sukurti prototipą pagrįstą parengta ontologijų jungimo metodika. Kuriamas prototipas vizualiai parodys, kaip yra panaudojama parengta ontologijų jungimo metodika. Taip pat bus stengiamasi vizualiai pavaizduoti ontologijų modulinio kūrimo galimybę. Šios galimybės dėka bus galima iš dviejų ontologijų sukurti vieną ontologiją. Tokiu būdu bus išvengiama dažnai daromų klaidų, kurios atsiranda kuomet nėra tiksliai detalizuojama kuriamos ontologijos dalykinė sritis.

Toliau pateikiamas įrankio prototipo projektas. Realizuojant dalis pateiktų komponentų bei vartotojo sąsaja gali kisti priklausomai nuo realizuojamų prototipo funkcijų.

Toliau pateikiama projektuojamo sprendimo loginė architektūra, vartotojo sąsajos ir sistemos funkcijų valdikliai.



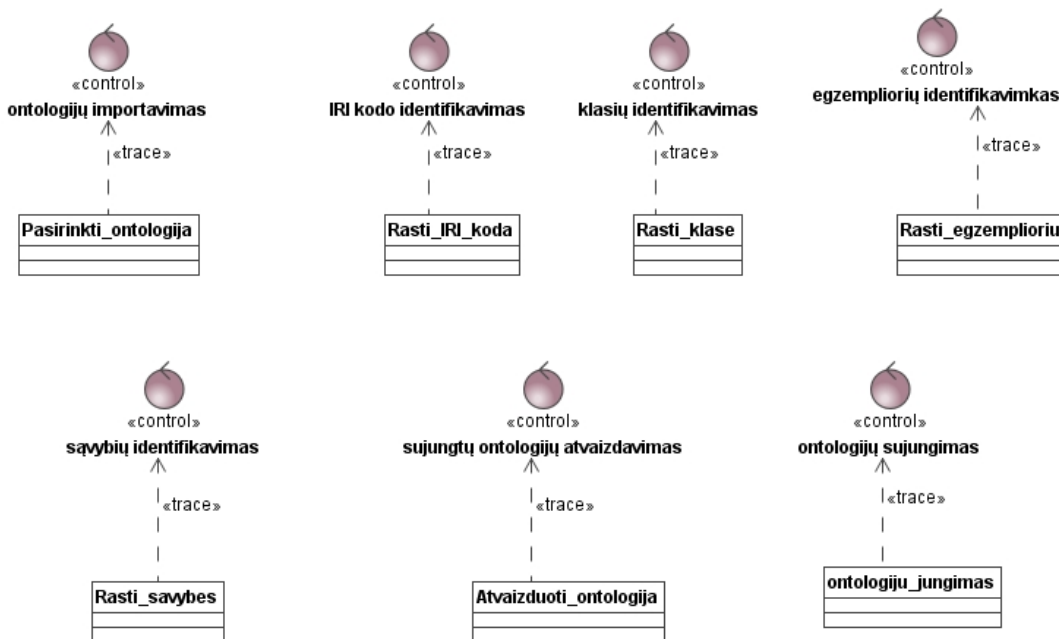
17 pav. Vartotojo sąsajos loginė architektūra

17 paveiksle pateikiami vartotojo sąsajos valdikliai. Pagrindiniai sprendimo sąsajos valdikliai yra ontologijų jungimo langas ir ontologijų peržiūros langas. Abiejuose languose bus pateikiamas ontologijų atvaizdavimas. Jungimo lange bus pavaizduojamos jungiamos ontologijos ir reikalingi nustatymų pasirinkimo komponentai, kurių pagalba į sistemą bus įvedami ontologijų jungimo kriterijai, t.y. pagrindinė ontologija, vienodi ontologijų elementai, esant poreikiui ir *IRI* adreso suvenodinimas.



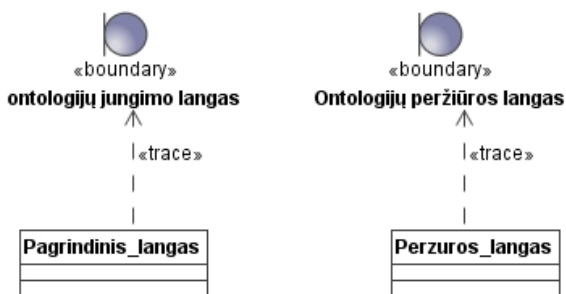
18 pav. Sistemos loginė architektūra

19 paveiksle pateikiami sistemos funkcijų valdikliai, kurie atlieka pagrindinius ontologijų jungimo veiksmus. Kiekvinas valdiklis skirtas atlikti tam tikras funkcijas susijusias su ontologijų jungimu.



19 pav. Valdymo klasės (moduliai)

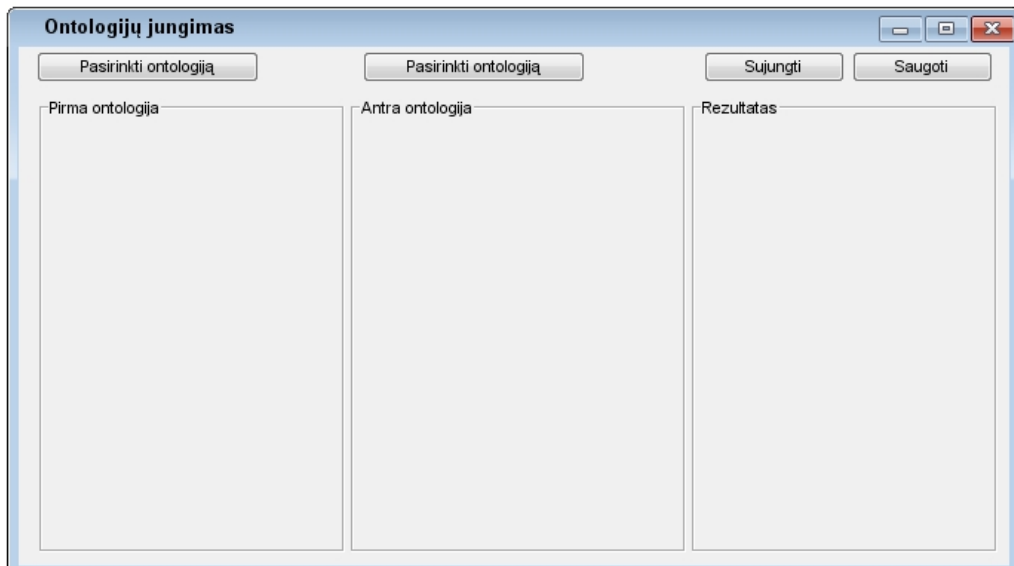
Valdymo klasės realizuotos remiantis valdikių projektavimu. Trasų diagrama parodo kaip susiję valdikliai ir valdymo klasės.



20 pav. Vartotojo sąsajos klasės

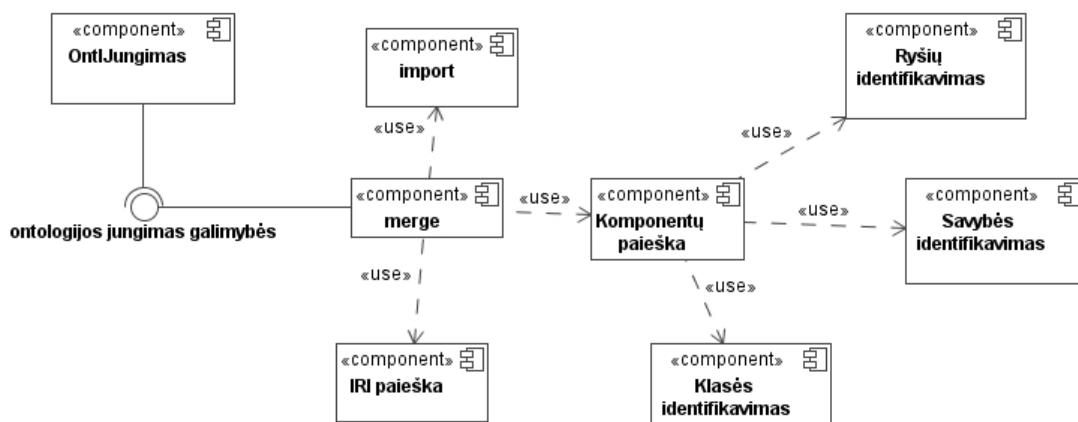
Vartotojo sąsajos klasės taip pat atvaizduojamos trasų diagramos pagalba, kuri nurodo kaip susiję vartotojo sąsajos valdikliai ir klasės.

21 paveiksle parodomas realizuojamos sistemos vartotojo sąsajos prototipas. Lange pateikiami pasirinktų ontologijų atvaizdavimo laukai bei jungimo rezultatų atvaizdavimas.



21 pav. Vartotojo sąsajos prototipas

Sistemos komponentų diagrama apibrėžia, iš kokių komponentų susideda realizuojama sistema. Kiekviename sistemos komponente bus aprašomos skirtingas funkcijas atliekančios sistemos klasės. Ryšiai tarp komponentų parodo kurie komponentai kuriuos naudoja.



22 pav. Sistemos komponentų diagrama

Kadangi prototipas bus kaip darbalaukio programa, t.y. prototipas bus diegiamas į kompiuteri, todėl duomenų bazė nebus reikalinga. Tačiau yra galimybė ontologijas pasiekti nuotoliniu būdu, t.y. galima naudotis ontologija, kuri yra patalpinta nutolusiame serveryje. Tokiu būdu nėra reikalinga ontologijos turėti kompiuteryje, reikia turėti jos adresą.

6. Eksperimentinis metodikos įvertinimas

Eksperimentiniams tyrimams atlikti buvo sukurtas prototipas, kurio veikimas grįstas ontologijų modulinio kūrimo ir jungimo metodikos principais. Eksperimentas bus vykdomas remiantis ontologijų modulinio kūrimo ir jungimo metodikos žingsniais.

Prototipo pagalba sistemos naudotojas pasirinkęs norimas ontologijos elementus galės importuoti į pasirinktą ontologiją. Programos lange bus pateiktos pasirinktos ontologijos.

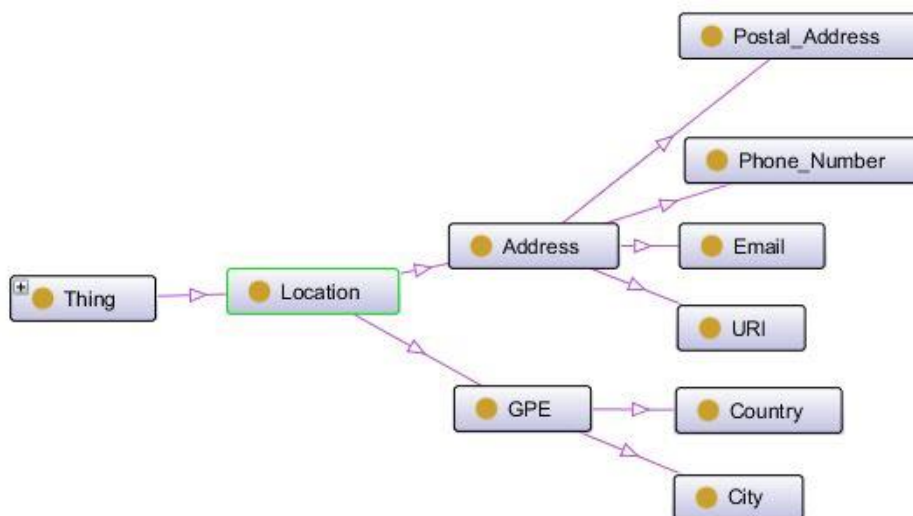
Iš pateiktų ontologijų elementų sistemos naudotojas galės sukurti naują ontologiją ir susieti reikiamus jos elementus.

Eksperimento metu sukuriamos kelios ontologijos remiantis analizės metu nustatytais principais ir apribojimais. Ontologijos buvo kuriamos siekiant pabrėžti modulinį ontologijų kūrimą. Todėl kiekvienoje ontologijoje aprašoma tik viena konkreti dalykinė sritis.

6.1 Dalykinės srities specifikavimas

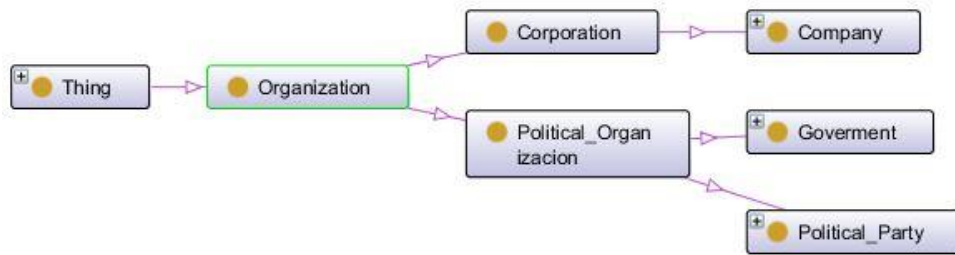
Sukurtos trys ontologijos, kuriose aprašomos skirtingos dalykinės sritys:

- Vietos ontologija – ontologijoje detalizuojama vietos informacija t.y. šalis, miestas, adresas.



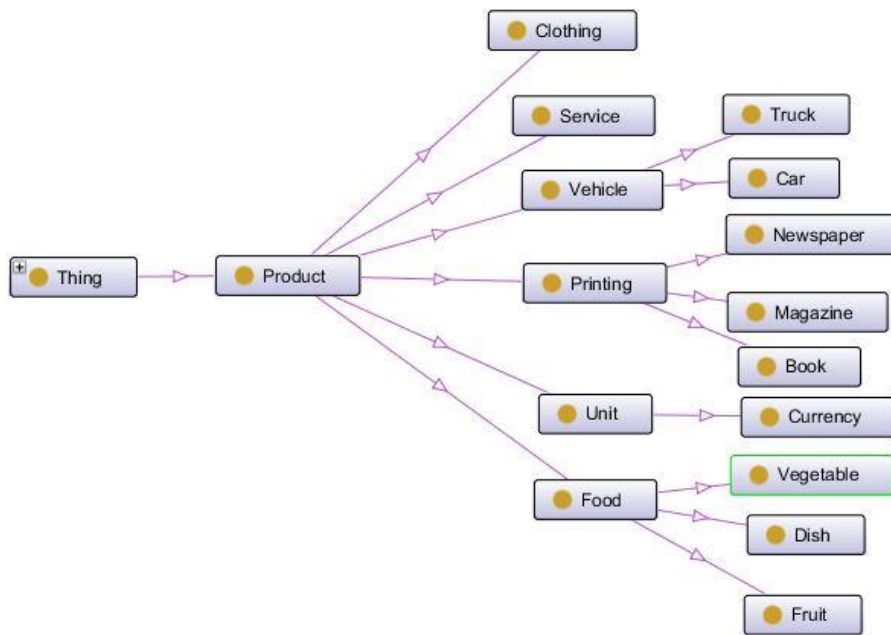
23 pav. Vietos ontologijos medis

- Organizacijų ontologija – ontologijoje detalizuojamos organizacijos, įmonės bei jų rūšys.



24 pav. Organizacijų ontologijos medis

- Produktų ontologija – ontologijoje detalizuojami produktai, gaminai, daiktai ir t.t..



25 pav. Produktų ontologijos medis

6.2 Ontologijos sukūrimas

Ontologijų kūrimui buvo naudojamas *Protege* ontologijų kūrimo įrankis. Naudojantis šiuo įrankiu buvo sukurtos trys ontologijos, kurios vėliau naudojamos eksperimente.



26 pav. Ontologijos kūrimas

Ontologijų jungimas ir kūrimas vykdomas atsižvelgiant į ontologijos sudedamąsias dalis. Atitinkamos ontologijos dalys bus naudojamos ontologijų jungimui ar naujų ontologijų kūrimui. Toliau pateiktame 27 paveiksle parodytas ontologijos klasės vaizdas. Tokio tipo klasės bus importuojamos į kitas ontologijas arba sukuriamos naujos ontologijos.

```
<Declaration>
  <Class IRI="#Book"/>
</Declaration>
<Declaration>
  <Class IRI="#Car"/>
</Declaration>
```

27 pav. Ontologijos klasės

```
<SubClassOf>
  <Class IRI="#Book"/>
  <Class IRI="#Printing"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Car"/>
  <Class IRI="#Vehicle"/>
</SubClassOf>
```

28 pav. Ontologijos sub-klasės

Taip pat ontologija susideda iš klasės apibūdinančių elementų. Šie elementai (29 – 32 pav.) apibūdina klases, kurios turi papildomų parametrų ar duomenų.

```
<Declaration>
  <ObjectProperty IRI="#capital_of" />
</Declaration>
```

29 pav. Ontologijos savybė

```
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#capital_of" />
  <NamedIndividual IRI="#Vilnius" />
  <NamedIndividual IRI="#Lietuva" />
</ObjectPropertyAssertion>
```

30 pav. Susietos ontologijos klasių egzemplioriai

```
<Declaration>
  <NamedIndividual IRI="#Stoholmas" />
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Vilnius" />
</Declaration>
```

31 pav. Ontologijos klasių egzemplioriai

```

<ClassAssertion>
  <Class IRI="#Vegetable"/>
  <NamedIndividual IRI="#Pomidoras"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Book"/>
  <NamedIndividual IRI="#Putino_nafta"/>
</ClassAssertion>

```

32 pav. Konkrečios ontologijos klasės egzemplioriai

6.3 Ontologijų sujungimas

Ontologijų jungimas bus paremtas teksto analizės metodais, kurie leis analizuoti ontologijos failą ir išrinkti reikiamas ontologijos dalis. Atrinkus ontologijos dalis sistema pateiks jų sąrašą naudotojui, kuris pasirinktą norimas dalis. Pasirinktos ontologijos elementai bus įrašomi į failą iš laikant ontologijos struktūrą. Tokiu būdu bus sukuriamas nauja ontologija.

Pagrindiniai elementai, kurie naudojami ontologijų jungimui yra:

- Class – apsiraso <declaration> </declaration> dalyje

```

<Declaration>
  <Class IRI="#Komanda"/>
</Declaration>

```

- Subclass – apsiraso <subclassof></subclassof> dalyje

```

<SubClassOf>
  <Class IRI="#Komanda"/>
  <Class IRI="#Asmuo"/>
</SubClassOf>

```

- NamedIndividual – apsiraso <declaration> </declaration> dalyje

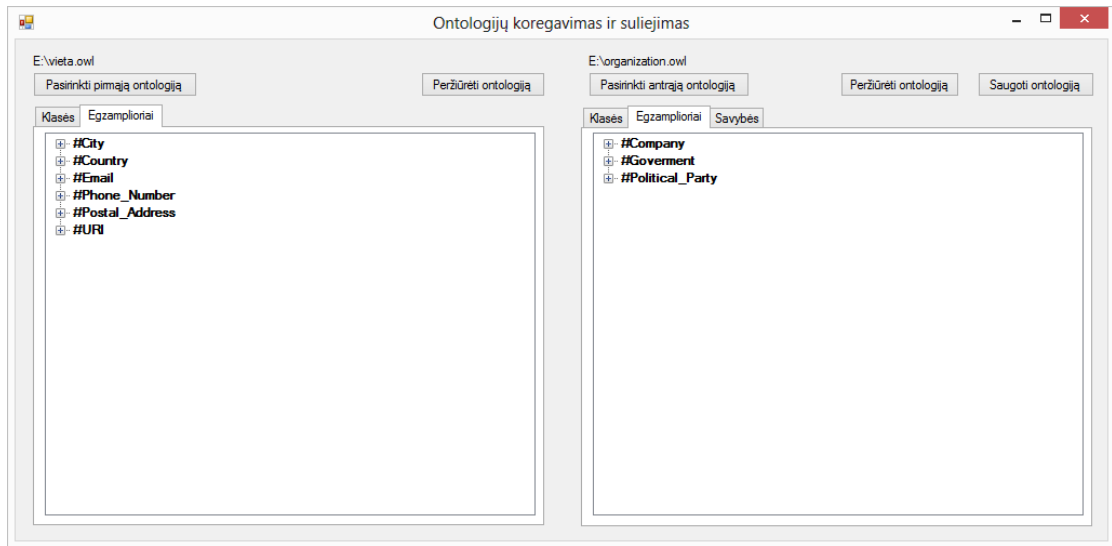
```

<Declaration>
  <NamedIndividual IRI="#Devils"/>
</Declaration>

```

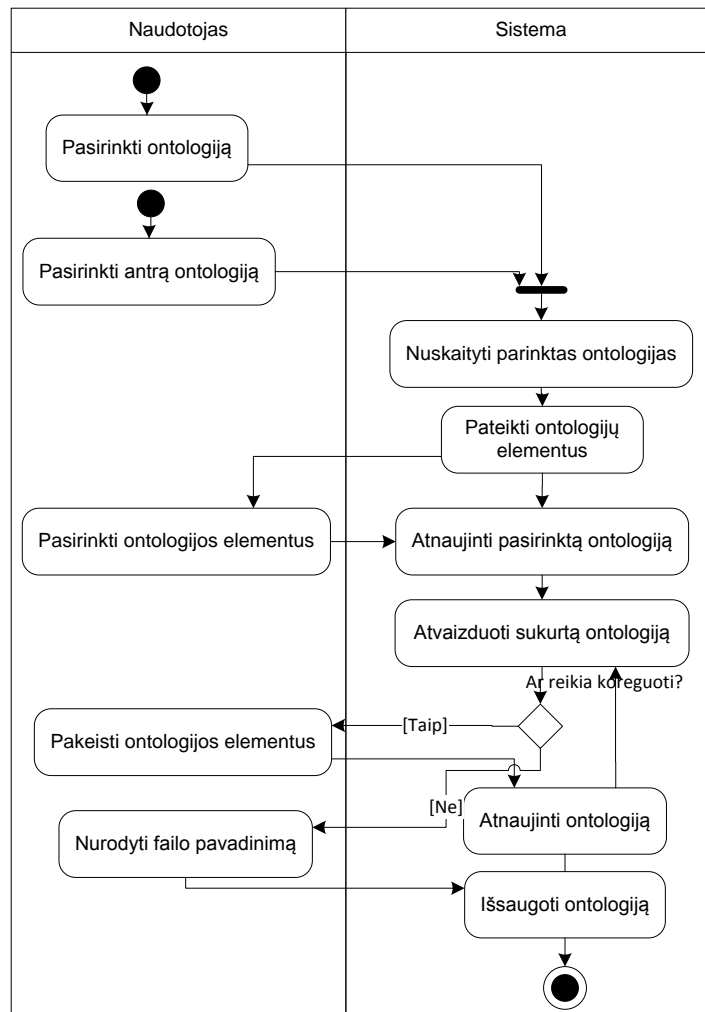
- ObjectProperty – yra naudojamas ryšiams. Ryšiai turėtų būti kuriami ranka. Jie nusako kaip susijusios *class* ir *subclass* elementai.

Pagal programos naudotojo pasirinktą ontologiją ir parinktus parametrus programa pateikia rezultatus, kurie leidžia naudotojui sukurti naują ontologiją iš norimų ontologijos klasių. 33 paveiksle pavaizduota pagrindinis sistemos langas, kuriame pateikiama ontologijų klasės.



33 pav. Pagrindinis sistemos langas

Veiklos procese 34 paveiksle nurodoma, kokie veiksmai atlieka ontologijų sujungimo metu. Šio proceso metu iš dviejų pasirinktų ontologijų sukuriama viena ontologija.



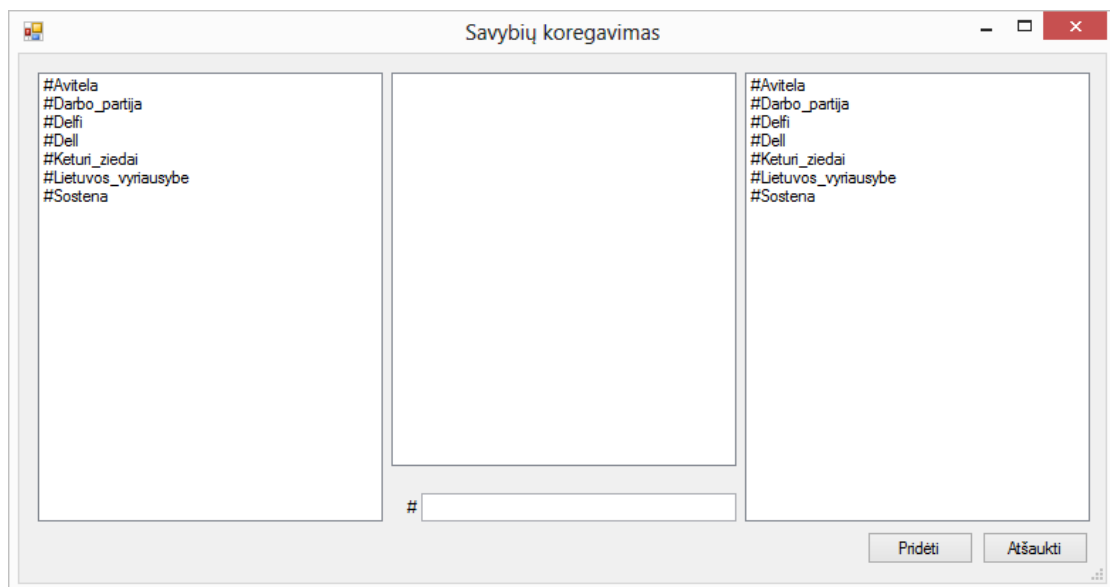
34 pav. Ontologijų sujiejimas

Lentelė 7. Ontologijų suliejimo proceso aprašai

Veiksmas	Vykdytojas	Aprašymas
Pasirinkti ontologiją	Naudotojas	Sistemos naudotojas pasirenka ontologiją, kurios elementais bus praplečiama kita ontologija.
Pasirinkti antrą ontologiją	Naudotojas	Sistemos naudotojas pasirenka antrą ontologiją, kuri bus plečiama norimais ontologijos elementais.
Nuskaityti parinktas ontologijas	Sistema	Analizuojamos ontologijos nuskaitant visus turimus elementus. Sistema apdoroja kiekvieną ontologijos elementą atskirai identifikuodama turimas savybes, ryšius bei aprašus. Sistema antrą ontologiją naudoja kaip pagrindą naujos ontologijos sukūrimui. Tokiu būdu ontologija papildoma reikiamais elementais ir savybėmis.
Pasirinkti ontologijos elementus	Naudotojas	Sistemos naudotojas nurodo kokius ontologijos elementus nori perkelti į jungiamą ontologiją. Taip pat nurodo kokius ryšius tarp elementų norima sukurti.
Atnaujinti pasirinktą ontologiją	Sistema	Sistema analizuoja parinktas ontologijas ir atlieka ontologijų jungimą. Sistema papildo plečiamos ontologijos elementų aibę naujais elementais bei ryšiais, kurie yra perkelti iš kitos ontologijos arba sukuriama nauji.
Atvaizduoti sukurtą ontologiją	Sistema	Pateikiama ontologija peržiūrai siekiant įsitikinti ar jungimo veiksmai buvo atlikti korektiškai. Jei jungimo metu buvo atlikti klaidingi veiksmai, tai galima koreguoti gautą ontologiją. Po kiekvienų atliktų koregavimo veiksmų turi būti tikrinama ontologija siekiant sukurti korektišką ontologiją.
Pakeisti ontologijos elementus	Naudotojas	Atliekami ontologijos redagavimo veiksmai. Pakeičiami ontologijos elementų pavadinimai, aprašai, sukuriama bei redaguojami ryšiai. Redagavimo metu sukuriama korektiška ontologija.
Atnaujinti ontologiją	Sistema	Sistema pateikia naują ontologijos medį pagal atliktus koregavimo veiksmus.
Nurodyti failo pavadinimą	Naudotojas	Naudotojas išsaugodamas nurodo ontologijos pavadinimą. Nurodant ontologijos pavadinimą patartina, kad jis atspindėtų ontologijos apibrėžiamą dalykinę sritį.

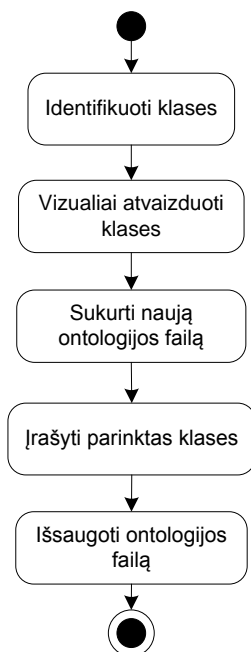
Veiksmas	Vykdytojas	Aprašymas
Išsaugoti ontologiją	Sistema	Išsaugoma sukurta ontologija. Nauja ontologija yra saugoma remiantis pasirinktų ontologijų struktūra. Ontologija išsaugoma faile, kuris atitinka pasirinktų ontologijų formatą bei struktūrą.

35 paveiksle parodyta, kaip yra jungiami konkretūs klasių egzemplioriai. Pasirenkami du egzemplioriai ir nurodoma savybės pavadinimas, pagal kurį sujungiami pasirinkti elementai.



35 pav. Savybių kūrimo langas

Bendrame veiklos procese 36 paveiksle parodyta, kokie veiksmai atliekami ontologijos klasių identifikavimo metu. Kiekvienas procese pateiktas veiksmas yra atskiras programos metodas atliekantis tik jam būdingas funkcijas. Daliai vykdomų veiksmų reikia ir programos naudotojo veiksmų tam, kad būtų įvykdyti visi programos veiksmai.

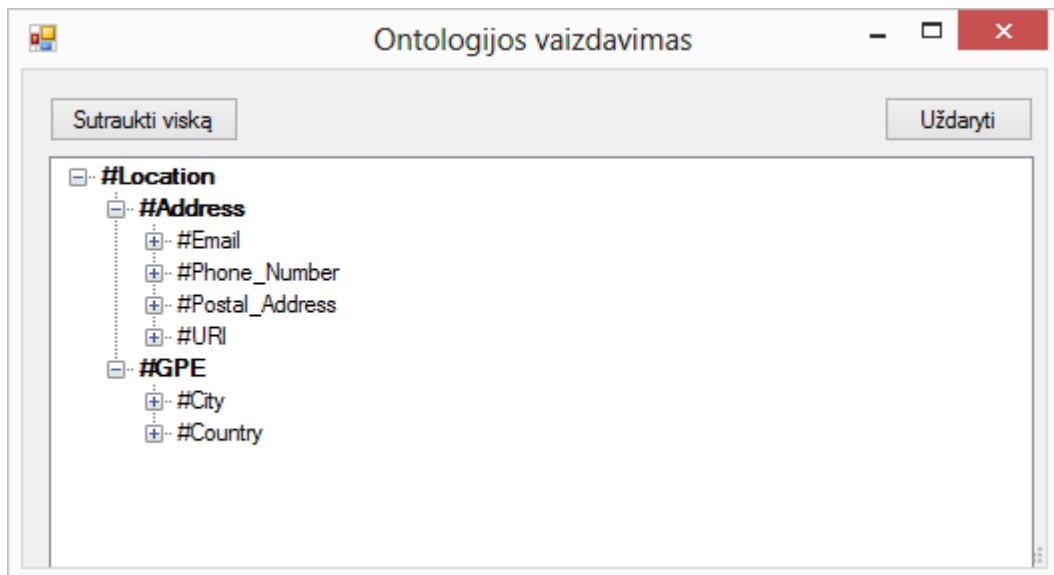


36 pav. Klasių identifikavimas

Lentelė 8. Klasių identifikavimo proceso aprašai

Veiksmas	Aprašymas
Identifikuoti klases	Identifikuojamos ontologijų klasės, kurios turi būti sujungiamos. Kadangi siejimo metu siejamos vienodos klasės papildant jas naujais lementais ar naujomis savybėmis.
Vizualiai atvaizduoti klases	Pateikiama ontologija peržiūrai siekiant įsitikinti ar jungimo veiksmai buvo atlikti korektiškai.
Sukurti naują ontologijos failą	Nauja ontologija yra saugoma remiantis pasirinktų ontologijų struktūra.
Įrašyti parinktas klases	Pagal rastus atitikmenis sujungiamos dvi pasirinktos ontologijos. Jungimo metu vienodi ontologijų elementai yra papildomi naujomis savybėmis.
Išsaugoti ontologijos failą	Išsaugoma sukurta ontologija. Nauja ontologija yra saugoma remiantis pasirinktų ontologijų struktūra. Ontologija išsaugoma faile, kuris atitinka pasirinktų ontologijų formatą bei struktūrą.

37 paveiksle pateikiamas ontologijos atvaizdavimo langas, kuriame pateikiama visa ontologija medžio pavidalu.

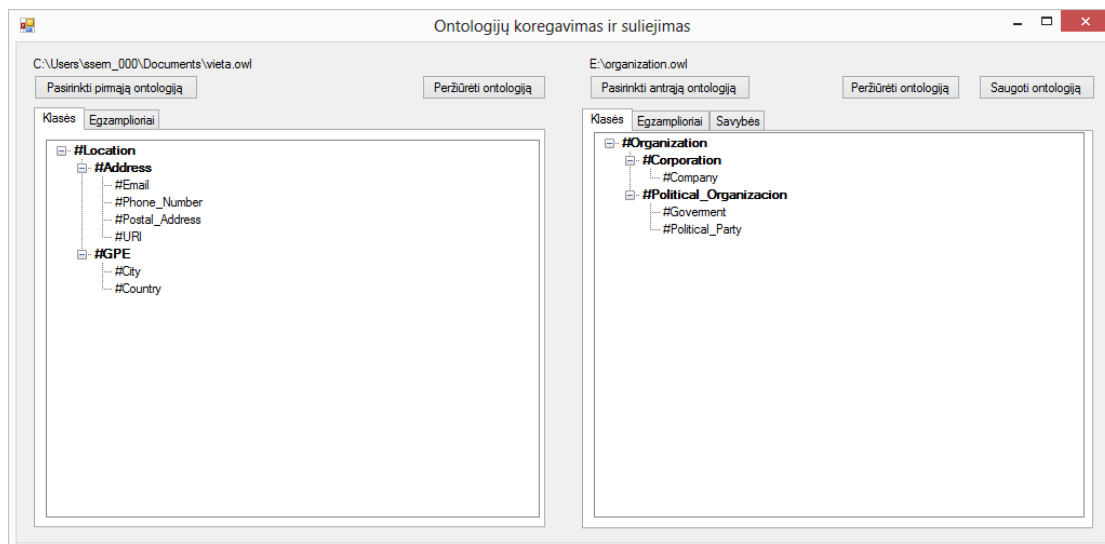


37 pav. Ontologijos vaizdavimo langas

6.4 Prototipo testavimas

6.4.1 Dviejų ontologijų sujungimas

Pasirinkus du ontologijų failus sistema atvaizduoja jų turinį medžio pavidalu (38 pav.). Vaizduojant ontologijas, pateikiami visi jos turi elementai, kurie naudojami ontologijų jungimui t.y., klasės, klasių egzemplioriai bei savybės.

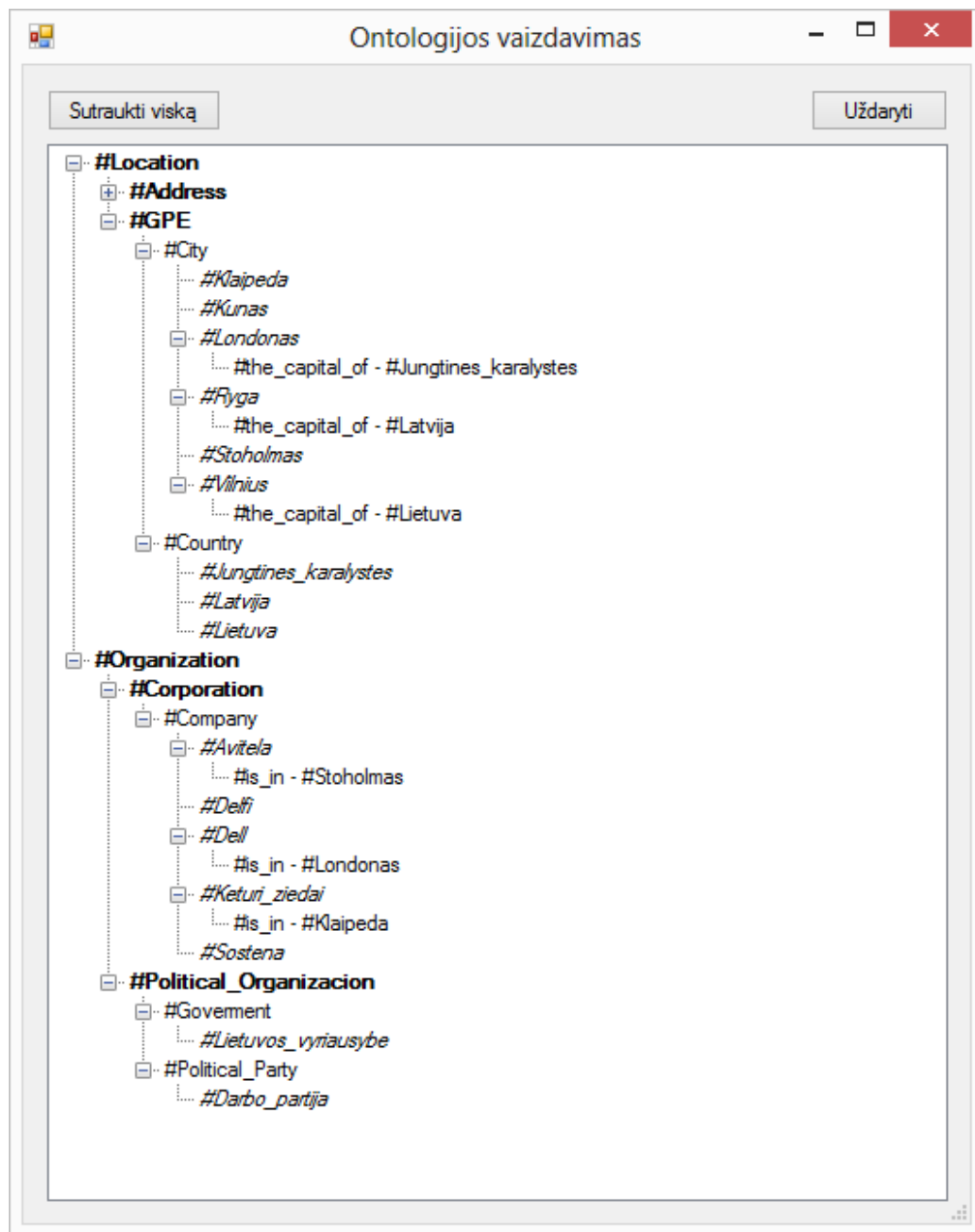


38 pav. Ontologijos klasių vaizdavimas

Atlikus ontologijų jungimo veiksmus, sistema pateikė sujungtos ontologijos vaizdą, kuriame pateikiama visa ontologija medžio pavidalu (39 pav.). Vaizduojant ontologiją pateikiami ontologijos elementai:

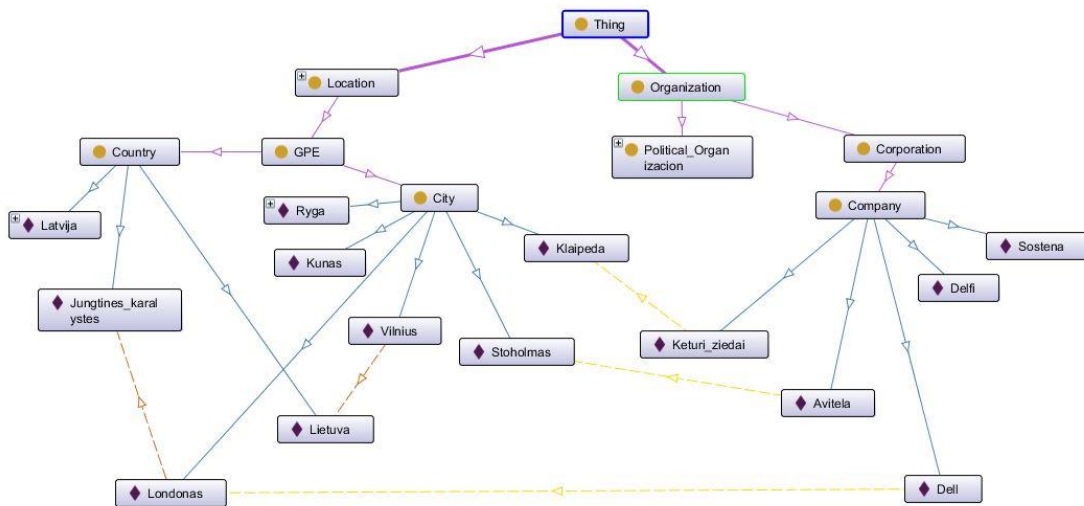
- **Klasė**
 - Sub-klasė

- *Klasės egzempliorius*
 - savybė



39 pav. Sukurtos ontologijos vaizdavimas

Atlikus ontologijų jungimą ir išsaugojus sujungtą ontologiją faile, ontologijos korektiškumas buvo patikrintas naudojant ontologijų kūrimo įrankį *Protege*. Gautas rezultatas pateikiamas 40 paveiksle. Į vieną ontologiją buvo sujungtos dvi skirtingos ontologijos, kurios aprašė skirtingas dalykines sritis.



40 pav. Sujungta ontologija

40 paveiksle pateiktos punktyrinės linijos rodo ryšius tarp klasių egzempliorių. Ryšiai buvo sukurti ontologijų jungimo metu.

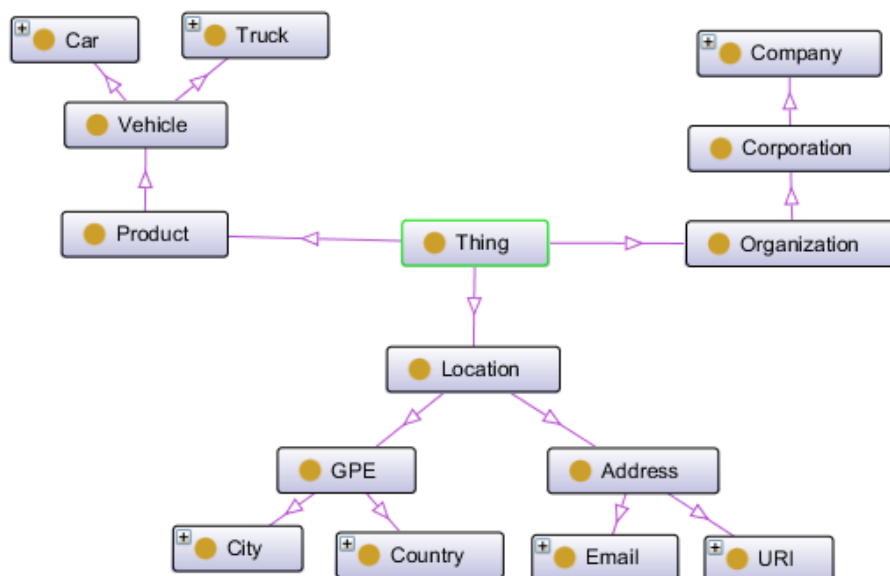
Nagrinėjant gautus rezultatus galime spręsti jos, ontologijų jungimas buvo atliktas sėkmingai. Tai galima spręsti iš to jog, atidarius sukurtą ontologiją su *Protege* sistema nepateikė jokių klaidų ar perspėjimų jog ontologija yra klaidinga ar blogai sukurta.

6.4.2 Modulinis ontologijų kūrimas

Remiantis parengta metodika buvo atliekamas eksperimentas, kurio metu iš kelių ontologijų skirtingų dalių buvo sukurta viena bendra ontologija. Ontologijos kūrimui buvo naudojamos vietos, organizacijų ir produktų ontologijos (23 – 25 pav.), parengtos pagal modulinio ontologijų kūrimo reikalavimus.

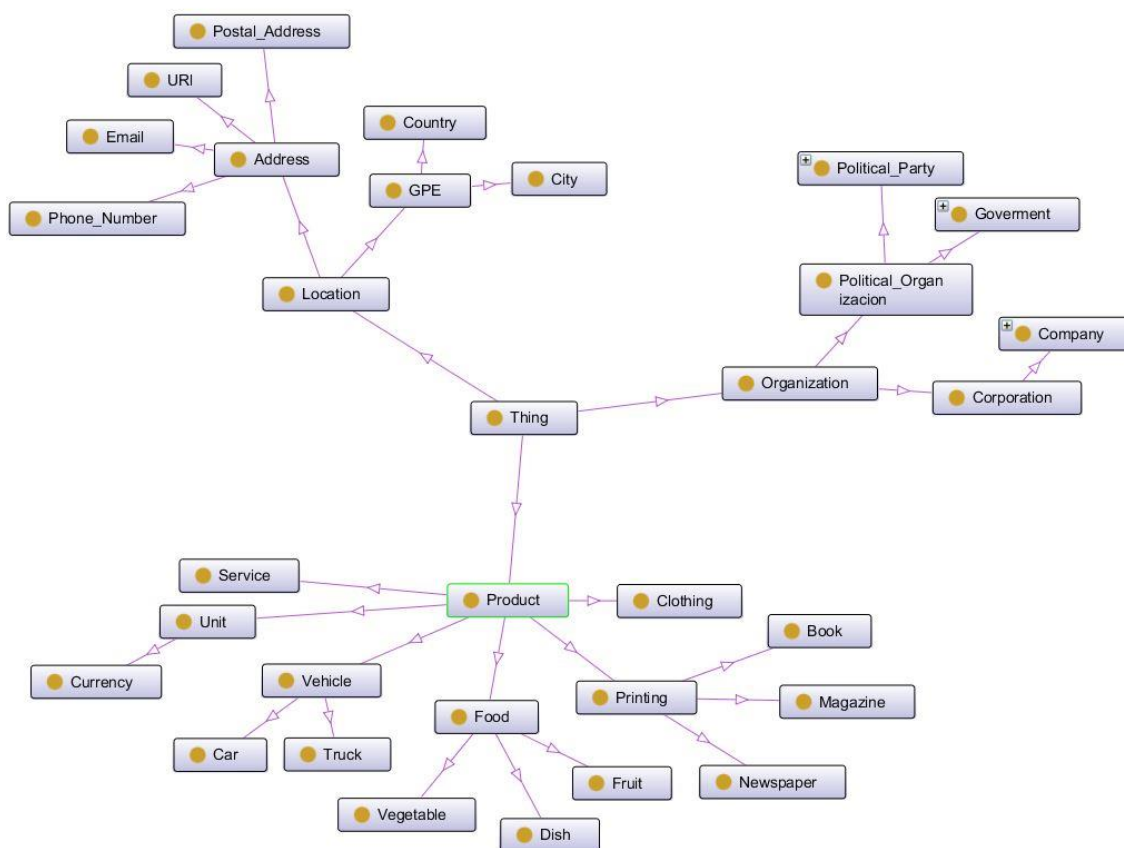
Eksperimento metu buvo gauti du rezultatai:

- Vieni rezultatai buvo gauti vadovaujantis parengta metodika ir naudojantis sukurto prototipu. Naudojantis prototipu buvo detalizuota *organization* ontologija. Į šią ontologiją buvo įkeltos atitinkamos ontologijų *location* ir *product* dalys. Tokiu atveju buvo gauta nauja ontologija (41 pav.). Gautai ontologijai peržiūrėti buvo naudojama *Protege* įrankis.



41 pav. Sukurta nauja ontologija

- Tokie patys veiksmai buvo bandomi atlikti naudojantis *Protege* ontologijų kūrimo ir jungimo įrankį. Siekiamiems rezultatams gauti buvo naudojama suliejimo funkcija. Į naują ontologiją buvo sudėtos trijų naudotų ontologijų dalys. Atlikus suliejimo veiksmus buvo gauta nauja ontologija (42 pav.). Gautai ontologijai peržiūrėti buvo naudojama *Protege* įrankis.



42 pav. Sulieta ontologija

Analizuojant gautus rezultatus galima pastebėti, jog sukurta ontologija (41 pav.) turi mažiau elementų. Tačiau visi ontologijoje esantys elementai yra naudojami detalizuojant organizaciją.

Pažvelgus į *Protege* įrankiu gautą ontologiją (42 pav.), pastebime joje visus naudotų ontologijų elementus. Tokiu atveju gauta ontologija yra perteklinė, kadangi tik dalis ontologijos elementų yra aktualūs.

Apibendrinant gautus rezultatus galime teigti jog, naudojantis metodika buvo gauti tikslesni rezultatai. Siekiant gauti panašius rezultatus naudojantis *Protege* įrankiu tenka papildomai redaguoti ontologiją ištrinant nereikalingas jos dalis. Sukurta ontologija gali būti papildomai papildyta naujomis ontologijomis ar jų dalimis.

Abiem atvejais kuriant ontologiją buvo nekeičiamos naudojamos ontologijos t.y., buvo sukuriamas naujas ontologijos failas, o naudojami failai buvo nekoreguojami.

6.5 Sukurtos metodikos vertinimas

Siekiant įvertinti eksperimento metu gautus rezultatus, sukurtos ontologijos buvo jungiamos naudojant du įrankius. T.y. ontologijos buvo jungiamos naudojant sukurta prototipą bei ontologijų kūrimo įrankį *Protege*. Pastaruoju įrankiu buvo naudota suliejimo (ang. *merge*) funkcija.

Atlikus gautų rezultatų palyginamąją analizę buvo pastebėta:

- Tiek prototipo, tiek *Protege* pagalba buvo korektiškai perkelti ontologijų klasių įrašai. Tačiau, naudojant *Protege* buvo perkeltos visos ontologijos klasės. Tokiu atveju buvo gauta perteklinė ontologija, kurioje buvo saugoma dalis nereikalingos informacijos.
- Suliejimo metu *Protege* įrankis tik perkėlė esamas ontologijų klases ir jas aprašančias savybes. Tačiau, ryšiai t.y. klasių egzemplioriai nebuvo susieti, kadangi siejimo veiksmai vyko automatiškai. Naudojat prototipą perkeliat norimas klases galima nurodyti papildomai ryšius tarp reikiamų ontologijos elementų. Tokiu būdu nereikia papildomai redaguoti sukurtos ontologijos.
- Prototipo naudojimas yra patogesnis ir geresnis, kadangi naudotojui suteikiama galimybė pasirinkti norimus ontologijos elementus ir jais praplėsti pasirinktą ontologiją. Taip pat plėtimo metu naudotojas gali atlikti ontologijos elementų jungimą. Tai nereikalauja papildomo ontologijos redagavimo, to negali pasiūlyti *Protege* įrankio suliejimo funkcionalumas.

Atlikus sukurtos metodikos ir įrankio palyginimą su *Protege* ontologijų kūrimo įrankiu buvo gauti rezultatai, kurie patiekti žemiau esančioje lentelėje.

Lentelė 9. Sukurtos metodikos palyginimas

Palyginimo kriterijus	Metodika, kurią taiko <i>Protege</i>	Sukurta metodika, kuria palaiko prototipas
Galimybė matyti abi ontologijas. (palengvina jungimą, ryšių kūrimą bei reikiamų elementų išrinkimą)	-	+
Galimybė pasirinkti ontologijų elementus (leidžia išvengti pertekliškumo ir dubliavimo)	-	+
Pagalba keičiant ontologijų elementų vardus (leidžia išvengti dubliavimo)	-	+
Galimybė vieną ontologiją naudoti kitų ontologijų kūrimui.	+	+
Galimybė, jungiant ontologijas, iš karto sukurti savybes. (nereikia papildomo ontologijos redagavimo)	-	+

Apibendrinat palyginimo rezultatus (Lentelė 9. Sukurtos metodikos palyginimas) galime teigti, jog sukurta metodika palengvina ontologijų kūrimą bei jungimą. Taip pat sukurta metodiką leidžia lanksčiau taikyti modulinį ontologijų kūrimą. Tokiu būdu yra padidintos ontologijų pakartotinio panaudojimo galimybės – galima kurti ontologijas iš atskirų dalių ar atskirų ontologijų.

7. Išvados

1. Atlikta mokslinės literatūros, ontologijų kūrimo įrankių ir ontologijų kūrėjų poreikių analizė parodė, kad ontologijų jungimo metu dažnai reikia jungti tik dalį ontologijos elementų, pakeisti jų vardus, užtikrinti neprieštarumą, tačiau tam nėra tinkamos metodikos ir esami ontologijų kūrimo įrankiai šių galimybių neturi.
2. Analizės metu buvo detalizuotas egzistuojantis ontologijų jungimo bei kūrimo procesai, kurie leido daryti prielaidą, jog šie procesai neužtikrina korektiško ontologijų jungimo bei kūrimo, todėl buvo reikalinga išskirti naujus procesus, kurie remtųsi sukurta metodika.
3. Atliktos analizės pagrindu sudaryta metodika leido daryti prielaidą, kad tokiai metodikai palaikyti galima sukurti ontologijų jungimo įrankį.
4. Sudaryta projektinė specifikaciją leido realizuoti siekiamo įrankio prototipą, padedantį atlikti ontologijų jungimo bei modulinio kūrimo funkcijas, kurio dėka galima lanksčiau taikyti ontologijų modulinį kūrimą bei pakartotinį ontologijų panaudojimą.
5. Atliktas eksperimentas, kurio metu pagal sudarytą metodiką buvo sukurtos trys modulinės ontologijos ir sujungtos, naudojant *Protege* ir sukurto įrankio prototipą. Eksperimentas parodė, kad prototipo jungimo galimybės didesnės ir geriau atitinka ontologijų kūrėjų poreikius.

8. Literatūra

1. Baader F., Horrocks I., Sattler U. Description logics./ Staab S., Studer R. *Handbook on Ontologies (International Handbooks on Information Systems)*. New York, 2004,
2. Sanjay Kumar Malik, Nupur Prakash, S.A.M. Rizvi, *Ontology Merging using Prompt plug-in of Protégé in Semantic Web*, 2010
3. Sven Abels, Liane Haak, Axel Hahn “*Identification of Common Methods Used for Ontology Integration Tasks*”, IHIS’05, November 4, 2005, Bremen, Germany. 2005 ACM 1-59593-184-5/05/0011
4. Jos de Bruijn, Marc Ehrig, Cristina Feier, *Ontology mediation, merging and aligning*, May 20, 2006
5. M. Mohsenzadeh, F. Shams, M. Teshnehlab, *A New Approach for Merging Ontologies*, Engineering and Technology 4, 2005
6. Eliza Sach, *Getting Started with Protege-Frames*, June 6, 2006
7. Managing imports in protégé-owl. Prieiga per internetą: < <http://protege.stanford.edu/doc/owl/owl-imports.html> > [žiūrėta 2012-04-13]
8. François Scharffe, Jos de Bruijn, Douglas Foxvog, *D4.3.2 Ontology Mediation Patterns Library V2*, University of Innsbruck, February 4, 2006, SEKT EU-IST-2003-506826
9. Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, *Knowledge Representation for the Semantic Web Part I: OWL 2*, 2009 Paderborn. Prieiga per internetą: < <http://semantic-web-book.org/page/KI2009> > [žiūrėta 2012-05-20]
10. Pascal Hitzler and Markus Krötzsch and Marc Ehrig and York Sure. *What Is Ontology Merging?* Institute AIFB, University of Karlsruhe, Germany.
11. François Scharffe, Ying Ding, and Dieter Fensel, *Towards Correspondence Patterns for Ontology Mediation*, University of Innsbruck, Austria
12. E. Jiménez-Ruiz, R. Berlanga, *A View-based Methodology for Collaborative Ontology Engineering: an Approach for Complex Applications (VIMethCOE)*, Proceedings of the 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 0-7695-2623-3/06, 2006
13. *TopBride Composer* programinės įrangos tinklapis. Prieiga per internetą: < <http://www.topquadrant.com/index.html> > [žiūrėta 2012-06-10]
14. TopQuadrant, Inc. *Getting Started Guide Version 3.0*, July 18, 2011
15. Alan L Rector, *Modularisation of Domain Ontologies Implemented in Description Logics and related formalisms including OWL*, ACM 1-58113-583-1/03/0010 University of Manchester, 2003 UK.

16. Alans Rector, *Normalisation of ontology implementations: Towards modularity, re-use, and maintainability*, Department of Computer Science, University of Manchester
17. Darius Jurkevičius, Olegas Vasilecas, Algirdas Laukaitis, *Formalių konceptų analizė, kuriant taikomosios srities ontologijas*. ISSN 1392–0561. Informacijos mokslai. 2008 46
18. Natalya F. Noy ir Deborah L. McGuinness. *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford University, Stanford, CA, 94305. Prieiga per internet http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html
19. OWL Web Ontology Language Semantics and Abstract Syntax. Prieiga per internetą. <http://www.w3.org/TR/owl-semantics/>
20. OWL Web Ontology Language Reference. Prieiga per internetą <http://www.w3.org/TR/owl-ref/>
21. GRÜNINGER, M. FOX M., 1995. *Methodology for the Design and Evaluation of Ontologies*. In Proceedings of the IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing. – Menlo Park Calif.: AAAI Press, 1995. –P. 1–10.

9. Priedai

9.1 Jungiamos ontologijos

9.1.1 Organizacijų ontologija organizacion.owl

```
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
xml:base="http://www.semanticweb.org/sigser0/ontologies/2013/3/untitled-ontology-16"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xml="http://www.w3.org/XML/1998/namespace"

ontologyIRI="http://www.semanticweb.org/sigser0/ontologies/2013/3/untitled-ontology-16">
  <Prefix name="" IRI="http://www.w3.org/2002/07/owl#" />
  <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#" />
  <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
  <Prefix name="xsd"
IRI="http://www.w3.org/2001/XMLSchema#" />
  <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#" />
  <Declaration>
    <Class IRI="#Company" />
  </Declaration>
  <Declaration>
    <Class IRI="#Corporation" />
  </Declaration>
  <Declaration>
    <Class IRI="#Government" />
  </Declaration>
  <Declaration>
    <Class IRI="#Organization" />
  </Declaration>
  <Declaration>
    <Class IRI="#Political_Organizacion" />
  </Declaration>
  <Declaration>
    <Class IRI="#Political_Party" />
  </Declaration>
  <Declaration>
    <Class IRI="#Company" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#Avitela" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#Darbo_partija" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#Delfi" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#Dell" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#Keturi_ziedai" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#Lietuvos_vyriausybe" />
  </Declaration>
```

9.1.2 Vietovių ontologija location.owl

```
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
xml:base="http://www.semanticweb.org/sigser0/ontologies/2013/3/untitled-ontology-11"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xml="http://www.w3.org/XML/1998/namespace"

ontologyIRI="http://www.semanticweb.org/sigser0/ontologies/2013/3/untitled-ontology-11">
  <Prefix name="" IRI="http://www.w3.org/2002/07/owl#" />
  <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#" />
```

```
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Sostena" />
</Declaration>
<SubClassOf>
  <Class IRI="#Company" />
  <Class IRI="#Corporation" />
</SubClassOf>
<SubClassOf>
  <Class IRI="#Corporation" />
  <Class IRI="#Organization" />
</SubClassOf>
<SubClassOf>
  <Class IRI="#Government" />
  <Class IRI="#Political_Organizacion" />
</SubClassOf>
<SubClassOf>
  <Class IRI="#Political_Organizacion" />
  <Class IRI="#Organization" />
</SubClassOf>
<SubClassOf>
  <Class IRI="#Political_Party" />
  <Class IRI="#Political_Organizacion" />
</SubClassOf>
<ClassAssertion>
  <Class IRI="#Company" />
  <NamedIndividual IRI="#Avitela" />
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Political_Party" />
  <NamedIndividual IRI="#Darbo_partija" />
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Company" />
  <NamedIndividual IRI="#Delfi" />
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Company" />
  <NamedIndividual IRI="#Dell" />
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Company" />
  <NamedIndividual IRI="#Keturi_ziedai" />
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Government" />
  <NamedIndividual IRI="#Lietuvos_vyriausybe" />
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Company" />
  <NamedIndividual IRI="#Sostena" />
</ClassAssertion>
</Ontology>
```

```
<Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
  <Prefix name="xsd"
IRI="http://www.w3.org/2001/XMLSchema#" />
  <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#" />
  <Declaration>
    <Class IRI="#Address" />
  </Declaration>
  <Declaration>
    <Class IRI="#City" />
  </Declaration>
  <Declaration>
```

```

    <Class IRI="#Country"/>
  </Declaration>
<Declaration>
  <Class IRI="#Email"/>
</Declaration>
<Declaration>
  <Class IRI="#GPE"/>
</Declaration>
<Declaration>
  <Class IRI="#Location"/>
</Declaration>
<Declaration>
  <Class IRI="#Phone_Number"/>
</Declaration>
<Declaration>
  <Class IRI="#Postal_Address"/>
</Declaration>
<Declaration>
  <Class IRI="#URI"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#8_37_123456"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#8_5_1234567"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#8_800_11111"/>
</Declaration>
<Declaration>
  <NamedIndividual
IRI="#Architetu_g_121,_Vilnius,_LT_234765"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Jungtines_karalystes"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Klaipeda"/>
</Declaration>
<Declaration>
  <NamedIndividual
IRI="#Kreves_pr_4,_Kaunas,_LTU_345435"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Kunas"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Latvija"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Lietuva"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Londonas"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Ryga"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Stoholmas"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Vilnius"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#info@ktu.lt"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#info@lrt.lt"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#naujienos@delfi.lt"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#www.delfi.lt"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#www.ktu.lt"/>

```

```

</Declaration>
<Declaration>
  <NamedIndividual IRI="#www.lrt.lt"/>
</Declaration>
<SubClassOf>
  <Class IRI="#Address"/>
  <Class IRI="#Location"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#City"/>
  <Class IRI="#GPE"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Country"/>
  <Class IRI="#GPE"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Email"/>
  <Class IRI="#Address"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#GPE"/>
  <Class IRI="#Location"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Phone_Number"/>
  <Class IRI="#Address"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Postal_Address"/>
  <Class IRI="#Address"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#URI"/>
  <Class IRI="#Address"/>
</SubClassOf>
<ClassAssertion>
  <Class IRI="#Phone_Number"/>
  <NamedIndividual IRI="#8_37_123456"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Phone_Number"/>
  <NamedIndividual IRI="#8_5_1234567"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Phone_Number"/>
  <NamedIndividual IRI="#8_800_11111"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Postal_Address"/>
  <NamedIndividual
IRI="#Architetu_g_121,_Vilnius,_LT_234765"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Country"/>
  <NamedIndividual IRI="#Jungtines_karalystes"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#City"/>
  <NamedIndividual IRI="#Klaipeda"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Postal_Address"/>
  <NamedIndividual
IRI="#Kreves_pr_4,_Kaunas,_LTU_345435"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#City"/>
  <NamedIndividual IRI="#Kunas"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Country"/>
  <NamedIndividual IRI="#Latvija"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Country"/>
  <NamedIndividual IRI="#Lietuva"/>
</ClassAssertion>

```

```

<ClassAssertion>
  <Class IRI="#City"/>
  <NamedIndividual IRI="#Londonas"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#City"/>
  <NamedIndividual IRI="#Ryga"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#City"/>
  <NamedIndividual IRI="#Stoholmas"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#City"/>
  <NamedIndividual IRI="#Vilnius"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Email"/>
  <NamedIndividual IRI="#info@ktu.lt"/>
</ClassAssertion>
<ClassAssertion>

```

```

  <Class IRI="#Email"/>
  <NamedIndividual IRI="#info@lrt.lt"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Email"/>
  <NamedIndividual IRI="#naujienos@delfi.lt"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#URI"/>
  <NamedIndividual IRI="#www.delfi.lt"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#URI"/>
  <NamedIndividual IRI="#www.ktu.lt"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#URI"/>
  <NamedIndividual IRI="#www.lrt.lt"/>
</ClassAssertion>
</Ontology>

```

9.2 Sujungta ontologija

```

<Ontology xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
ontologyIRI="http://www.semanticweb.org/sigser0/ontologies/2
013/3/untitled-ontology-16"
xml:base="http://www.semanticweb.org/sigser0/ontologies/201
3/3/untitled-ontology-16"
xmlns="http://www.w3.org/2002/07/owl#">
  <Prefix name=""
IRI="http://www.w3.org/2002/07/owl#" />
  <Prefix name="owl"
IRI="http://www.w3.org/2002/07/owl#" />
  <Prefix name="rdf"
IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
  <Prefix name="xsd"
IRI="http://www.w3.org/2001/XMLSchema#" />
  <Prefix name="rdfs"
IRI="http://www.w3.org/2000/01/rdf-schema#" />
  <Declaration>
    <Class IRI="#Company" />
  </Declaration>
  <Declaration>
    <Class IRI="#Corporation" />
  </Declaration>
  <Declaration>
    <Class IRI="#Government" />
  </Declaration>
  <Declaration>
    <Class IRI="#Organization" />
  </Declaration>
  <Declaration>
    <Class IRI="#Political_Organizacion" />
  </Declaration>
  <Declaration>
    <Class IRI="#Political_Party" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#Avitela" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#Darbo_partija" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#Delfi" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#Dell" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#Keturi_ziedai" />
  </Declaration>
  <Declaration>
    <NamedIndividual
IRI="#Lietuvos_vyriausybe" />
  </Declaration>

```

```

  <Declaration>
    <NamedIndividual IRI="#Sostena" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#info@ktu.lt" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#info@lrt.lt" />
  </Declaration>
  <Declaration>
    <NamedIndividual
IRI="#naujienos@delfi.lt" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#8_37_123456" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#8_5_1234567" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#8_800_11111" />
  </Declaration>
  <Declaration>
    <NamedIndividual
IRI="#Architetu_g_121,_Vilnius,_LT_234765" />
  </Declaration>
  <Declaration>
    <NamedIndividual
IRI="#Kreves_pr_4,_Kauanas,_LTU_345435" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#www.delfi.lt" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#www.ktu.lt" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#www.lrt.lt" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#Klaipeda" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#Kunas" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#Londonas" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#Ryga" />
  </Declaration>
  <Declaration>
    <NamedIndividual IRI="#Stoholmas" />
  </Declaration>

```



```

        <Class IRI="#Postal_Address" />
        <NamedIndividual
IRI="#Architetu_g_121,_Vilnius,_LT_234765" />
        </ClassAssertion>
        <ClassAssertion>
        <Class IRI="#Postal_Address" />
        <NamedIndividual
IRI="#Kreves_pr_4,_Kaunas,_LTU_345435" />
        </ClassAssertion>
        <ClassAssertion>
        <Class IRI="#URI" />
        <NamedIndividual IRI="#www.delfi.lt" />
        </ClassAssertion>
        <ClassAssertion>
        <Class IRI="#URI" />
        <NamedIndividual IRI="#www.ktu.lt" />
        </ClassAssertion>
        <ClassAssertion>
        <Class IRI="#URI" />
        <NamedIndividual IRI="#www.lrt.lt" />
        </ClassAssertion>
        <ClassAssertion>
        <Class IRI="#City" />
        <NamedIndividual IRI="#Klaipeda" />
        </ClassAssertion>
        <ClassAssertion>
        <Class IRI="#City" />
        <NamedIndividual IRI="#Kunas" />
        </ClassAssertion>
        <ClassAssertion>
        <Class IRI="#City" />
        <NamedIndividual IRI="#Londonas" />
        </ClassAssertion>
        <ClassAssertion>
        <Class IRI="#City" />
        <NamedIndividual IRI="#Ryga" />
        </ClassAssertion>
        <ClassAssertion>
        <Class IRI="#City" />
        <NamedIndividual IRI="#Stoholmas" />
        </ClassAssertion>
        <ClassAssertion>
        <Class IRI="#City" />
        <NamedIndividual IRI="#Vilnius" />
        </ClassAssertion>
        <ClassAssertion>
        <Class IRI="#Country" />
        <NamedIndividual
IRI="#Jungtines_karalystes" />
        </ClassAssertion>

```

```

<ClassAssertion>
  <Class IRI="#Country" />
  <NamedIndividual IRI="#Latvija" />
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Country" />
  <NamedIndividual IRI="#Lietuva" />
</ClassAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#the_capital_of" />
  <NamedIndividual IRI="#Vilnius" />
  <NamedIndividual IRI="#Lietuva" />
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#the_capital_of" />
  <NamedIndividual IRI="#Ryga" />
  <NamedIndividual IRI="#Latvija" />
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#the_capital_of" />
  <NamedIndividual IRI="#Londonas" />
  <NamedIndividual
IRI="#Jungtines_karalystes" />
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#is_in" />
  <NamedIndividual IRI="#Avitela" />
  <NamedIndividual IRI="#Stoholmas" />
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#is_in" />
  <NamedIndividual IRI="#Dell" />
  <NamedIndividual IRI="#Londonas" />
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#is_in" />
  <NamedIndividual IRI="#Keturi_ziedai" />
  <NamedIndividual IRI="#Klaipeda" />
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#contacts" />
  <NamedIndividual IRI="#www.delfi.lt" />
  <NamedIndividual IRI="#8_5_1234567" />
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#contacts" />
  <NamedIndividual IRI="#www.lrt.lt" />
  <NamedIndividual IRI="#8_37_123456" />
</ObjectPropertyAssertion>
</Ontology>

```