

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Mindaugas Gaurilčikas

**OWL 2 ontologijų taikymo interneto portaluose
metodika**

Magistro darbas

Darbo vadovas
prof. L. Nemuraitė

Kaunas, 2012

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Mindaugas Gaurilčikas

OWL 2 ontologijų taikymo interneto portaluose metodika

Magistro darbas

Recenzentas

dr. A. Janavičiūtė

2012-05-28

Vadovas

prof. dr. Lina Nemuraitė

2012-05-28

Atliko

IFM 0/4- gr. stud.

Mindaugas Gaurilčikas

2012-05-28

Kaunas, 2012

1. Įvadas	6
2. Ontologijų ir jomis grindžiamų portalų analizė	9
2.1 Tyrimo sritis, objektas ir problema.....	9
2.2 Ontologijų vaidmuo Pasaulinio tinklo kontekste.....	9
2.2.1 Ontologijos apibrėžimas.....	9
2.2.2 Ontologijų aprašymo kalbų analizė.....	10
2.2.3 Ontologijų kūrimo įrankių analizė	14
2.3 Ontologijomis grindžiamų portalų kūrimo metodų, architektūros ir galimų įgyvendinimo priemonių analizė	15
2.3.1 Ontologijos kūrimo metodikų analizė	15
2.3.2 Semantinių portalų kūrimo karkasų analizė	20
2.3.3 Egzistuojančių ontologijomis grindžiamų portalų analizė	21
2.4 Siekiamos metodikos kūrimo uždavinio apibrėžimas.....	26
2.4.1 Galima semantinio portalo struktūra.....	26
2.4.2 Portalo kūrimo procesas naudojant ontologiją	29
2.5 Analizės išvados	30
3. Ontologijomis grindžiamų portalų kūrimo metodika.....	31
3.1 Metodo eiga	31
3.2 Ontologijos reikalavimų suformulavimas	31
3.3 Ontologijos sudarymas.....	32
3.4 Klasių egzempliorių kūrimas.....	33
3.5 Kompetencijos klausimų formalizavimas	33
3.6 Semantinis ontologijos pilnumo tikrinimas.....	33
3.7 Portalo architektūros projektavimas ir portalo realizacija	34
3.8 Semantinis portalo tikrinimas	34
3.9 Ontologijos ir portalo vystymas	35
4. Mokslinių publikacijų portalo reikalavimų specifikacija ir projektas	36
4.1 Reikalavimų specifikacija	36
4.1.1 Panaudojimo atvejų diagrama.....	36
4.1.2 Panaudojimo atvejų specifikacijos.....	37
4.1.3 Dalykinės srities ontologijos modelis	40
4.2 Sistemos architektūra	41
4.3 Vartotojo navigacijos planas	42
4.4 Sistemos elgsenos modelis	42
4.4.1 Skaitytojo naršymas portale.....	42
4.4.2 Publikacijų citavimo indekso atnaujinimas	44
4.4.3 Citavimo indekso skaičiavimo algoritmo schema	44
4.5 Duomenų bazės schema	46
4.6 Diegimo diagrama.....	47
5. Mokslinių publikacijų portalo realizacija	48
5.1 Realizacijos ir veikimo aprašymas	48
5.1.1 Portalo ontologijos struktūra	48
5.1.2 Portalo struktūra.....	48
6. Eksperimentinis metodo įvertinimas.....	54

6.1	Eksperimento planas	54
6.2	Eksperimentas.....	54
6.2.1	Kompetencijos klausimų sąrašo sudarymas	54
6.2.2	Bazinės ontologijos sukūrimas	54
6.2.3	Klasės egzempliorių sukūrimas	56
6.2.4	Kompetencijos klausimų formalizavimas	56
6.2.5	Ontologijos tikrinimas.....	57
6.2.6	Semantinis portalo tikrinimas	58
6.3	Eksperimento rezultatai.....	60
7.	Išvados.....	61
8.	Literatūra	62
9.	Terminų žodynas.....	64
10.	Priedai	65
10.1	Portalo duomenų bazės lentelių aprašymas.....	65
10.2	Portalo duomenų bazės lentelių sukūrimo SQL kodas	66
10.3	Mokslinių publikacijų OWL ontologijos funkcinė sintaksė	67

Summary

Methodology for applying OWL 2 ontologies in Web portals

Web portals collect the information scattered on the internet and organize it for easy search and access. Current Web technologies employed to build up these portals present serious limitations regarding information search, access, extraction, interpretation and processing.

The Semantic Web provides a common framework that allows data to be shared, reused and processed directly and indirectly by machines. To facilitate this process, OWL has been developed as standard formats for the sharing and integration of data in the form of rich conceptual schemas called ontologies. An ontology defines a common vocabulary for researchers who need to share information in a specific domain.

There are several methods for facilitating development of ontologies. Although, neither describes how to build a Semantic Web portal step by step, using the latest Web portal development tools and technologies.

The main goal of this work is to propose a development methodology for building Semantic Web portals from the scratch.

1. Įvadas

Interneto portalai yra patogi priemonė leidžianti centralizuotai prieiti prie informacijos internete. Interneto portalai pagreitina informacijos panaudojimą, apjungdami informaciją iš skirtingų šaltinių ir pateikdami ją vieningoje, vartotojui patogioje, sąsajoje.

Aktualumas. Kasdien didėjant internetu prieinamo informacijos kiekiui, surasti norimą informaciją, tokioje jos gausoje, yra tikras iššūkis. Specializuotoms paieškos sistemoms vis sudėtingiau apdoroti augantį informacinį krūvį ir pateikti tikslius rezultatus. Taip yra dėl to, kad didžioji dalis šiuo metu internetu prieinamos informacijos yra neorganizuota ir nesistemiška. Be to, tradicinių interneto paieškos sistemų veikimas yra paremtas paieška pagal raktažodžius, nenagrinėjant bendro konteksto. To pasekoje, paieškos užklausos rezultatų gali būti labai daug, arba jie gali neteikti atsakymų į vartotojui rūpimą klausimą.

Dėl visų šių minėtų priežasčių tolimesni interneto vystymą mokslininkai sieja su semantiniu tinklu. Terminą *semantinis tinklas* sugalvojo šiandieninio interneto tinklo išradėjas ir tarptautinės organizacijos, leidžiančios interneto standartus, „*World Wide Web Consortium*“ (W3C) vadovas *Tim Berners Lee*. Jis ateities interneto tinklo viziją mato kaip globalią duomenų bazę, kurioje kompiuteriai gali patys analizuoti ir suprasti informaciją [1].

Vienas iš esminių semantinio tinklo komponentų yra ontologijos. Ontologija yra tam tikros dalykinės srities sąvokų visumos specifikuojimas išreikštu pavidalu. Semantiniai interneto portalai naudoja ontologijas kaip žinių bazę. Tačiau ontologijų kūrimas nėra toks paprastas, todėl reikia metodikos, kaip aprašyti dalykinę sritį ontologijų kalba, kad ji būtų realizuojama interneto portale, atitiktų vartotojų poreikius ir ontologijų kokybės kriterijus.

Problema. Trūksta literatūros aprašančios ontologija grįstų portalų kūrimą šiuolaikinėmis priemonėmis. Reikalinga išsami metodika, paaiškinanti, kaip teisingai kurti ontologija grįsta portalą.

Darbo tikslas. Šio magistrinio darbo tikslas yra padidinti semantinių portalų kūrimo galimybes sudarant ir praktiškai išbandant metodiką, pagal kurią portalų kūrėjai galėtų kurti semantinius portalus, panaudodami dalykinės srities ontologiją.

Darbe sprendžiami šie uždaviniai.

1. Išanalizuoti ontologijos kalbas ir ontologijomis grįstų portalų kūrimo technologijas ir metodus;

2. Sudaryti ontologijomis grindžiamų portalų kūrimo metodiką, kuri papildytų esamus informacinių sistemų kūrimo metodus taisyklėmis ir kriterijais, reikalingais taikant ontologijas;
3. Sukurti semantinį portalą vadovaujantis sudaryta metodika;
4. Atlikti eksperimentą metodikos tinkamumui įvertinti.
5. Atlikta analizė su šaltinių nuorodomis. Darbo metu remtasi šaltiniais aprašančiais semantinių tinką ir jo technologijas [13],[7], ontologijų kūrimo ir jų teisingumo užtikrinimo metodus [6],[11],[12].

Darbo rezultatai. Sudaryta metodika, nuosekliai aprašanti ontologija grindžiamo portalo kūrimo procesą. Sukurtam metodikai išbandyti ir įvertinti atliktas eksperimentas.

Naujumas. Darbe aprašyta ontologija grindžiamo portalo kūrimo metodika, kuri glaudžiai susijusi su dalykinės srities ontologijos sudarymo procesu; panaudotos ontologijos modelio saugojimo reliacinėje duomenų bazėje galimybės, atliktas ontologijos savybių rodiklių skaičiavimas.

Praktinis taikymas. Pagal sukurtą metodiką realizuotas semantinis mokslinių publikacijų portalas, kuriame skelbiamos išleistos mokslinės publikacijos, žurnalai bei publikuoti straipsniai. Sukurtas publikacijų reitingavimas pagal tai, kiek dažnai publikacija yra cituojama kitų autorių darbuose. Šis taikymas galėtų būti pavyzdys kitiems praktiniams taikymams kurti.

Ši sritis pasirinkta ne tik kaip pavyzdys metodikai tirti, bet ir dėl to, kad mokslinių publikacijų paieška, vertinimas ir analizė vis dar yra pakankamai komplikotas ir daug laiko reikalaujantis uždavinys. Pavyzdžiui, vykdant mokslinį tyrimą, labai svarbi yra literatūros, kurios pagrindu buvo remtasi tyrimo metu, kokybė. Literatūros darbo kokybei įvertinti ir išmatuoti mokslinėse institucijose yra atliekamas literatūros publikacijų vertinimas apskaičiuojant jų citavimo rodiklį (angl. *impact factor*). Citavimo rodiklis atspindi, kaip dažnai publikaciją cituoja kiti autoriai. Todėl, citavimo rodiklis yra vienas iš būdų, pagal kurį galima įvertinti publikacijos turinio kokybę.

Citavimo rodikliai pateikiami pripažintose duomenų bazėse, pavyzdžiui, *ISI Web of Science*. Tačiau ten cituojami tik tam tikro lygio žurnalai ir konferencijų straipsniai; be to, jų turinys neprieinamas. Tuo tarpu daugeliui universitetų ar tyrimo organizacijų reikia savos citavimo informacinės sistemos, kuri realiau atspindėtų tos aplinkos mokslininkų publikacijų tematiką, jų tarpusavio ryšius, publikacijų svarbumą. Be to, tokia citavimo sistema galėtų

padaryti prieinamą ir publikacijų turinį, kurį būtų galima atsirinkti pagal kokybinius rodiklius.

Darbo struktūra. Darbo analitinėje dalyje aptariama semantinio tinklo samprata ir jo veikimo principai. Pateikiamas ontologijos apibrėžimas, analizuojamos ontologijos aprašymo kalbos, jų sudarymo metodai ir įrankiai. Atliekama panašių semantinių sistemų analizė. Analitinės dalies pabaigoje pateikiamos atliktos analizės išvados.

Koncepcinėje dalyje nuosekliai aprašomi sukurtos metodikos, skirtos ontologija grindžiamo portalo kūrimui, žingsniai ir eiga.

Projektinėje darbo dalyje specifikuojami reikalavimai, kurių pagrindu, kuriamas mokslinių publikacijų portalas ir sudaromas jo modelis.

Realizacijos dalyje aprašoma kaip realizuotas mokslinių publikacijų portalas ir kaip jis veikia.

Eksperimentinėje darbo dalyje atliekamas sukurtos metodikos patikrinimas atliekant eksperimentą. Eksperimento metu, pagal sudarytą metodiką kuriamas mokslinių publikacijų portalas. Atliekamas sukurtos metodikos įvertinimas

2. Ontologijų ir jomis grindžiamų portalų analizė

Šioje dalyje aptariama semantinio tinklo samprata ir jo veikimo principai. Pateikiamas ontologijos apibrėžimas, sąvokos, analizuojamos ontologijos aprašymo kalbos, jų sudarymo metodai ir įrankiai. Siekiant pasirinkti tinkamiausias mokslinių publikacijų portalo realizacijos priemones, atliekamas ontologijų aprašymo kalbų, įrankių ir palyginimas.

2.1 Tyrimo sritis, objektas ir problema

Tyrimo objektas – ontologijomis grindžiamų portalų kūrimo procesas.

Tyrimo sritis – ontologijų aprašymo kalbos, ontologijos kūrimo ir panaudojimo interneto portale metodai.

Tyrimo problema – dalykinės srities ontologijos panaudojimas kuriant informacines sistemas ir portalus ne tik padeda išnaudoti kompiuterius informacijos žvalgyje, bet ir sutaupo laiko projektuotojams bandant suprasti dalykinę sritį. Reikalinga išsami metodika, paaiškinanti, kaip kurti ontologija grįsta portalą, kaip jį tikrinti ir plėsti.

Semantinis tinklas (angl. *Semantic Web*) – tai aibė metodų ir technologijų, leidžiančių automatams dalintis informacija ir suprasti jos prasmę internete [7]. Semantinio tinklo idėja yra ta, kad kompiuteriai suprastų apdorojamos informacijos prasmę ir padėtų vartotojui surasti, susisteminti ir integruoti reikiamą informaciją. Pagrindiniai semantinio pasaulinio tinklo komponentai yra resursai, juos aprašantys metaduomenys, ontologijos bei pasaulinio tinklo paslaugos.

2.2 Ontologijų vaidmuo Pasaulinio tinklo kontekste

Šiame skyriuje pateikta ontologijų aprašymo kalbų ir ontologijų projektavimo įrankių analizė ir palyginimas.

2.2.1 Ontologijos apibrėžimas

Žodis "ontologija" yra iš filosofijos mokslo kilęs terminas, turintis prasmę "būties teorija". Ontologija yra pagrindinis filosofijos skyrius – būties mokslas, nagrinėjantis būties teoriją, tikrovės pobūdį ir struktūrą.

Nuo 1990 m. ontologijos terminas pradėtas naudoti kompiuterijos ir dirbtinio intelekto srityse, nagrinėjančiose žinių bendrą naudojimą, programinių agentų tarpusavio sąveiką, visuotinai pripažįstamų žinių aprašymą, natūralios kalbos apdorojimą ir kt.

Kompiuterijoje ontologija apibrėžia nagrinėjamos srities esybių (daiktų bei reiškinių) tipus, jų tarpusavio sąryšius, galiojančias taisykles, dėsningumus ir aksiomas [13].

Sąvoka „ontologija“ dažnai suprantama kaip bazinis semantinio pasaulinio tinklo komponentas. Bendru atveju ontologija yra realaus pasaulio dalies modelis. Galima teigti, kad ontologijoje saugoma informacija modeliuoja realaus pasaulio struktūrą.

Kadangi viso pasaulio ontologija būtų be galo sudėtinga, dažnai nagrinėjama tik specifinės (dalykinės) srities ontologija (angl. *domain ontology*), pavyzdžiui, mokslinių publikacijų sistema, žmogaus anatomija, teisė ir pan. Dalykinės srities ontologijos sudarymas ir nagrinėjimas padeda įsisavinti, suvokti jos struktūrą, ryšį su aplinka.

Pagrindinis ontologijų naudojimo informacinėse sistemose privalumas yra pakartotinis sukauptų žinių panaudojimas. Tai padeda išvengti netikslumų, kylančių dėl dalykinės srities nesupratimo ar subjektyvumo. Pakartotinis ontologijų naudojimas informacinėse sistemose, kai jos kuriamos nuo nulio, sumažina galimų klaidų skaičių.

Kitas svarbus ontologijos privalumas yra tas, kad jau sudarytos ontologijos žiniomis visada gali pasinaudoti kiti vartotojai ar programiniai moduliai [4]. Tai reiškia, kad ontologijomis paremti portalai gali keistis duomenimis – integruoti duomenis iš įvairių šaltinių ir juos analizuoti. Be to, vienos ontologijos gali papildyti kitas, jungtis kartu, taip sudarydamos didelį tinklą.

Šios savybės yra pagrindinis ontologijomis grįstų (semantinių) portalų privalumas lyginant su įprastais portalais, turinčiais individualią duomenų bazę.

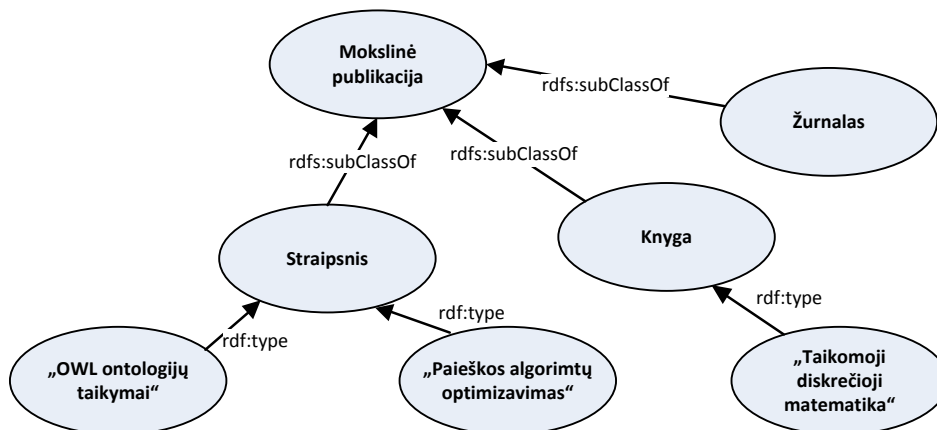
2.2.2 Ontologijų aprašymo kalbų analizė

2.2.2.1 RDF kalbos analizė

Semantinio tinklo resursams aprašyti konsorciumas W3C rekomenduoja RDF (angl. *Resource Description Framework*) kalbą [17]. Tai yra XML ženklavimo kalbos standartu paremtas metaduomenų aprašymo formatas. RDF kalboje kiekvienas teiginys yra sudarytas iš 3 elementų: objekto, objekto savybės ir objekto savybės reikšmės. Dažnai šis trejetas suvokiami kaip veiksnys (angl. *subject*), tarinys (angl. *predicate*) ir papildinys (angl. *object*).

Kiekvienas iš *RDF* trejeto elementų identifikuojami unikaliu *URI* (*Uniform Resource Identifier*) identifikatoriumi. Norint reprezentuoti faktų, galiojančių dalykinėje srityje, visumą, šiuos trejetus reikia kombinuoti (jungti į grafą). *RDF* schema (*RDFS*) praplečia *RDF* kalbos žodyną iki galimybės aprašyti specifines klases, hierarchijos ryšius ir savybes.

RDF kalbą, papildyta *RDF* schema, galima, vadinti riboto išraiškingumo žinių ontologija. Paveiksle 2.1 pavaizduota supaprastinta mokslinių publikacijų klasifikavimo ontologija panaudojant bazinius *RDF* klasifikavimo ir priskyrimo atributus: *rdfs:subClassOf*, *rdf:type*.



2.1 pav. Mokslinių publikacijų ontologijos klasių taksonomija RDFS kalba

2.2.2.2 OWL kalbos analizė

Šiuo metu populiariausia semantinio portalo ontologijų aprašymo kalba yra *OWL* (angl. *Web Ontology Language*) [14] kuri yra *RDFS* plėtinys. *OWL* 2004 metais rekomendavo *W3C* konsorciumas.

Tai vieningas žinių apie tai, kas egzistuoja realiame pasaulyje, pateikimo portalo formatas. *OWL* buvo sukurta tam, kad būtų galimybė aprašyti ontologijos klases bei santykius tarp šių klasių. *OWL* kaip ir *RDF* aprašo elektroninio dokumento turinio struktūrą, bet ne jo pateikimo būdą vartotojui, nes neturi grafinės notacijos.

OWL turi tris dialektus, besiskiriančius išraiškos galimybėmis: *OWL Lite*, *OWL DL* ir *OWL Full*.

1. *OWL Lite* skirta sudarinėti paprastiems žinių aprašams, turi griežtus reikalavimus, todėl jos išraiškos galimybės yra ribotos. Dėl savo griežtų apribojimų *OWL Lite* ontologijos apdorojimui nereikia sudėtingos programinės įrangos.
2. *OWL DL* yra *OWL Lite* plėtinys ir skirtas vartotojams, norintiems plačių išraiškos galimybių ir jų programinio išsprendžiamumo (angl. *decidability*) per baigtinį laiką. *OWL DL* apima visą *OWL* kalbos konstravimo žodyną. Apribojimų atžvilgiu ji yra laisvesnė negu *OWL Lite* dialektas.

3. **OWL Full** yra dialekto **OWL DL** plėtinys, kuris užtikrina maksimalias žinių išraiškos galimybes. Tačiau dėl savo sudėtingos struktūros vargu šiuo metu nėra sukurta jokia programinė įranga gebanti pilnai apdoroti **OWL Full** ontologiją.

2009 metais konsorciumas **WC3** patvirtino ontologijos aprašymo kalbos standartą **OWL 2**. Tai yra 2004 metais patvirtinto **OWL** (pastarąją šiandien priimtą vadinti **OWL 1**) standarto plėtinys.

OWL 2 turi tris naujus dialektus, kurie yra pritaikyti specifinėms taikomųjų programų užduotims spręsti:

1. **OWL 2 EL** standartiniams išvedimo užduotims spręsti. Yra naudojami polinominiai laiko algoritmai, todėl naudojami didelėse ontologijose. **OWL 2 EL** naudingas programose, kuriose nereikia didelių išraiškų galimybių, bet reikia greitų išvedimo rezultatų.
2. **OWL 2 QL** skirtas programoms, naudojančioms sąlyginai nedideles ontologijas ir tiesiogines užklausas į reliacines duomenų bazes (pavyzdžiui **SQL**). Čia išvedimo algoritmų prioritetą yra teisingi užklausų rezultatai.
3. **OWL 2 RL** skirta programoms kurioms reikia plačios išvedimų logikos, neprarandant išraiškos galimybių.

Klasės yra pagrindiniai **OWL** ontologijos blokai. Klasės skirtos grupuoti elementus pagal jų panašumo charakteristikas. Klasės sudaro taksonominę hierarchiją ir yra apibrėžiamos panaudojant elementą *owl:Class*. **OWL** kalboje egzistuoja dvi iš anksto apibrėžtos klasės: *owl:Thing* ir *owl:Nothing*. Pirma yra bendra ir įtraukia viską, antra – tai tuščia klasė. Kiekviena **OWL** klasė yra *owl:Thing* poklasė, o *owl:Nothing* kiekvienos klasės viršklasė. **OWL** klasių pavyzdžiai:

- Šakninės klasės „*fakultetas*“ sukūrimas:

```
<owl:Class rdf:ID="fakultetas"/>
```
- Poklasės „*informatikosFakultetas*“ priklausančios viršklasei „*fakultetas*“ sukūrimas:

```
<owl:Class rdf:ID="informatikosFakultetas">  
  <rdfs:subclassOf rdf:resource="#fakultetas"/>  
</owl:Class>
```

Klasės aprašas apibūdina **OWL** klasę pagal jos pavadinimą arba išreiškiant santykiu su kitomis klasėmis (sankirta, sąjunga ir pan.)

Taigi, iš viso skiriami šie **OWL** klasių aprašymo būdai:

1. **Klasės pavadinimas.** Klasės pavadinimą apibrėžia ištekliaus identifikatorius *URI* adresas.
2. **Elementų išvardijimas.** Detaliai užrašomi visi klasei priklausantys individualūs elementai. Pavyzdys:

```
<owl:Class>
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Vilnius"/>
    <owl:Thing rdf:about="#Kaunas"/>
    <owl:Thing rdf:about="#Klaipeda"/>
  </owl:oneOf>
</owl:Class>
```

3. **Elementų savybių apribojimas.** Tai neįvardinta klasė, kurią sudaro visi elementai tenkinantys nustatytus apribojimus. Apribojimų klasė gali turėti tik vieną savybę, pagal kurią bus atrenkamos ją sudarančios klasės. Taip pat turi būti tik viena reikšmė pagal kurią yra lyginama *onProperty* dalyje aprašyta savybė. Pavyzdys:

```
<owl:Restriction>
  <owl:onProperty rdf:resource="#priklausoKategorijai" />
  <owl:allValuesFrom rdf:resource="#mokslinis" />
</owl:Restriction>
```

Šis pavyzdys aprašo bevarde *OWL* klasę, sudarytą iš visų individualinių elementų, kurių savybė „*priklausoKategorijai*“ turi vienintelę reikšmę „*mokslinis*“.

4. **Dviejų ar daugiau klasių sankirta, sąjunga arba papildymas.** Elementai apibrėžiami aibių teorijos principais. Atitinkamai naudojamos tokios išraiškos:
owl:intersectionOf, *owl:unionOf*, *owl:complementOf*.

Kitas *OWL* ontologijos komponentas – savybė. Savybė tai yra binarinis ryšys, jungiantis dvi klases. Savybės gali būti dviejų tipų:

- **objektų savybės** (angl. *object properties*), kurios suriša individualinius objektus tarpusavyje. Paprasčiausiu atveju objekto savybės aksioma parodo, kokia savybė priklauso objektui. Pavyzdžiui: `<owl:ObjectProperty rdf:ID="yraParasytas"/>`
- **duomenų tipų savybės** (angl. *datatype properties*), ryšys tarp klasės ir *RDF* aprašo arba *XML* schemos. Savybė suriša individualius elementus su duomenų tipu (sveikieji skaičiai, skaičiai su kableliu, simboliai ir t.t) reikšmėmis. Duomenų tipų nustatymui *OWL* naudojama *XML* schema.

```
<owl:DatatypeProperty rdf:ID="yearValue">
  <rdfs:domain rdf:resource="#VintageYear" />
  <rdfs:range rdf:resource="xsd:positiveInteger"/>
</owl:DatatypeProperty>
```

2.2.3 Ontologijų kūrimo įrankių analizė

Ontologijų kūrimo įrankių palyginimo analizei buvo pasirinkti tik tie įrankiai, kurie yra nemokami ir vystomi (t.y. kūrėjai vis dar teikia jų atnaujinimus).

Vienas iš populiariausių ontologijos kūrimo įrankių yra „Protégé“ [15] parašytas *Java* kalba. „Protégé-OWL“ yra grafinis *OWL* kalbos redaktorius, kuris leidžia sukurti ontologiją nuo pirmųjų žingsnių, kai tik formuojama hierarchija. Be to, galima vystyti jau sukurtą ontologiją, įvedant papildomus apribojimus, ryšius.

Ontologijos gali būti eksportuojamos įvairiais formatais, įskaitant *RDF(S)*, *OWL*, ir *XML* schemą. Yra galimybė ontologiją peržiūrėti vizualiai. Dėl savo lanksčios architektūros, *Protégé* turi daug papildinių, kurie išplečia įrankio funkcionalumą.

Semantinio tinklo paslaugoms (angl. *web services*) kurti *Protégé* turi integruotą *Jena* karkasą [9]. Tai *Java* kalba parašytas karkasas turintis: *RDF/OWL* paslaugų programavimo sąsaja, *RDF* trejetų skaitymo, rašymo galimybę, *SPARQL* užklausų procesorių.

Kitas analizei pasirinktas ontologijų kūrimo įrankis yra kompanijos „TopQuadrant“ komercinis įrankis „TopBraid Composer“ turintis tris leidimus: nemokama *Free* ir mokamas *Standard* ir *Maestro* versijas. Mokami leidimai turi tokias funkcijas kaip *XML* schemas generavimą, suderinamumą su reliacinėmis duomenų bazėmis, *HTML* dokumentų generavimą iš ontologijos ir kitas funkcijas. Lyginamajai analizei buvo pasirinkta tik nemokama „TopBraid Comporser Free“ versija, turinti mažiau galimybių.

Abiejų ontologijų projektavimo įrankių palyginimas pateiktas 2 lentelėje.

1 lentelė. Ontologijų projektavimo įrankių palyginimas

Įrankis	<i>Protégé 4.1</i>	<i>TopBraid Composer Free</i>
Lyginamoji savybė		
Ontologijų aprašymo kalbos	RDF, RDFS, OWL, Turtle	RDF, RDFS, OWL, N3, N-Triple, Turtle
Grafinis ontologijos peržiūros, redagavimo režimas	Taip	Taip
Duomenų saugojimo formatai	CLIPS, OWL, RDFS, JDBC DB, XML, Latex, Turtle	OWL, RDFS, XML, XLS
SWRL taisyklių kūrimas ir vykdymas	Taip	Ne
SPARQL užklausų vykdymas	Taip	Taip

Ontologijų suliejimo (<i>merge</i>) galimybė	Taip	Taip
Galimybė suformuoti ontologiją iš UML diagramos	Taip	Ne
Išvedimo logikos mechanizmai (<i>reasoners</i>)	Pellet, Fact++, RacerPro KAON2	Ne
Ontologijos atvaizdavimas grafo pavidalu.	Taip	Ne
Plečiamumas	Taip	Ne
Licencija	Visiškai nemokamas	Nemokama bandomoji 30-ies dienų versija
Dokumentacija ir literatūra.	Išsami dokumentacija. Daug literatūros aprašančios ontologijos kūrimą <i>Protégé</i> aplinkoje.	Dokumentacija akcentuota į mokamus leidimus. Yra literatūros padedančios įsisavinti įrankio naudojimą.

Abu ontologijų kūrimo įrankiai yra galingi ir palaiko populiariausią ontologijų aprašymo *OWL* kalbą. Tačiau „*TopBraid Composer Free*“ turi tik 30 dienų nemokamą bandomąją versiją. Pateikiama naudotojo dokumentacija yra orientuota į mokamus įrankio leidimus.

Tuo tarpu įrankis „*Protégé*“ yra ne tik kad visiškai nemokamas, atviro kodo, bet ir plačiai dokumentuotas, lengvai plečiamas plėtinių pagalba. Remiantis šiais kriterijais nuspręsta ontologijos kūrimui naudoti „*Protégé*“ įrankį.

2.3 Ontologijomis grindžiamų portalų kūrimo metodų, architektūros ir galimų įgyvendinimo priemonių analizė

2.3.1 Ontologijos kūrimo metodikų analizė

Yra du dalykinės srities ontologijos sudarymo būdai 1) **rankinis** ontologijos kūrimas 2) **automatinis** ontologijos išgavimas iš egzistuojančių duomenų šaltinių.

Pirmasis ontologijų sudarymo būdas apima rankinį ontologijos kūrimą, ontologiją pradedant konstruoti nuo nulio. Toks ontologijos sudarymo būdas reikalauja nemažų laiko išteklių, gerų dalykinės srities žinių, bei praktinės ontologijų modelių kūrimo patirties. Kaip bebūtų, rankiniu būdu sudarytos ontologijos kokybė priklauso nuo jos sudarančiojo žinių ir dalykinės srities suvokimo. Kita problema yra ta, kad praktikoje ontologijos dažnai kuriamos nepaisant taisyklių, kurių laikantis, užtikrinama eilė ontologijoms privalomų savybių – pilnumo, nepertekliškumo, plečiamumo ir kt.

Antrasis būdas apima automatinius ontologijų išgavimo mechanizmus [10], kai pradinė

ontologija sudaroma konvertuojant įvairių tipų šaltinių (pvz. duomenų bazių, žodynų, teksto) duomenis. Tokios ontologijų struktūros pagal poreikį vėliau gali būti koreguojamos ranka.

Analizės dalyje nagrinėsime neautomatizuotas ontologijų sudarymo metodus.

2.3.1.1 „Uschold ir King“ metodas.

Tai ontologijų kūrimo metodika [19], sudaryta pagal patirtį, kuriant verslo įmonių aprašymo ontologiją „*Enterprise Ontology*“. Metodika apima 4 esminius etapus 1) ontologijos tikslo apibrėžimą 2) ontologijos sudarymą 3) kitų ontologijų integravimą 4) ontologijos vertinimą 5) ontologijos dokumentavimą.

Pradedant ontologijos kūrimą prioritetiniu uždaviniu metodikos autoriai įvardija ontologijos tikslo nustatymą. Šiame žingsnyje reikia apibrėžti kam yra kuriama ontologija ir kas bus jos vartotojai. Ontologijos kūrimo procesas susideda iš dalykinės srities konceptų ir ryšių tarp jų identifikavimo; vienareikšmio identifikuotų konceptų ir jų sąvokų žodyno sudarymo; formalizuoto jų aprašymo pasirinkta ontologijos kalba. Metodikoje siūlomos trys strategijos pagrindiniams konceptams identifikuoti:

1. Iš viršaus žemyn. Nustatomi patys abstrakčiausi konceptai, kurie vėliau detalizuojami gilyn.
2. Iš apačios žemyn. Pirma nustatomi patys smulkiausi konceptai, kurie vėliau grupuojami.
3. Svarbiausi pirma. Pirma identifikuojami dalykinei sričiai svarbiausi konceptai, kurie vėliau generalizuojami arba specializuojami į kitus konceptus. Šis metodas geras tuo, kad ontologija plečiame tik mums reikiama linkme.

Metodas yra apibendrintas ir nepriklausomas nuo pasirinktų ontologijos technologijų. Metodo trūkumas yra, tas kas procesas nėra plačiai aprašytas. Taip pat, metodas neaprašo ontologijos palaikymo ir vystymo stadijų.

2.3.1.2 „Grüniger and Fox“ metodas

Metodika sukurta remiantis praktika kuriant „*TOVE*“ projekto ontologiją, kuri apima verslo įmonių procesus ir esybės. Siūlomas ontologijos sudarymo metodas [5] remiasi tuo, kad bandoma identifikuoti pagrindinius scenarijus, vykstančius aplinkoje, kurioje bus naudojama ontologija (pavyzdžiui, kompiuterinės sistemos taikymo srityje). Pagal surašytus scenarijus sudaroma neformalių kompetencijos klausimų aibė. Šie kompetencijos klausimai yra pagrindiniai ontologijos reikalavimų šaltinis, ir sudaryta ontologija turi teikti atsakymus į šiuos klausimus. Kompetencijos klausimai naudojami ne tik ontologijos įvertinimui, bet ir pagrindinių konceptų ir jų savybių nustatymui.

Šis, kaip ir prieš tai nagrinėtas metodas neturi ontologijos vystymo stadijų.

2.3.1.3 „Methontology“ metodas.

Metodas tinka kuriant ontologijas nuo pat pradžių arba integruojant jau egzistuojančias ontologijas. Metodas [11] remiasi iteraciniu ontologijos vystymo požiūriu, turi projekto valdymo ir kokybės užtikrinimo mechanizmus

Metodikos autoriai, ontologijos kūrimo veiklas suskirstė i 3 skirtingas kategorijas: projekto valdymą, ontologijos vystymą ir ontologijos palaikymą.

Projekto valdymas apima veiklų planavimą, kontrolę ir kokybės užtikrinimą. Planavimo metu nustatoma kokia yra užduočių atlikimo tvarka, kiek laiko ir resursų jiems skirti. Šis planavimo žingsnis yra svarbus kuriant ontologijas, kurios įtraukia jau panaudotas ontologijas. Kokybės užtikrinimas reikalingas tam, kad rezultatai (ontologija, programinė įranga ir dokumentacija) būtų tinkamos kokybės.

Ontologijos kūrimo veiklos susideda iš reikalavimų specifikavimo, konceptualizacijos, formalizavimo ir realizacijos.

- **Reikalavimų specifikavimo** metu aprašoma, kam yra kuriama ontologija, kas bus jos vartotojai, surašomi kompetencijos klausimai. Kompetencijos klausimų formalizmo lygis nėra griežtai ribojamas.
- **Konceptualizacijos** metu, sudaromas terminų žodynas ir konceptų modelis. Terminams surinkti gali būti panaudota reikalavimų specifikacija.
- **Formalizavimo** metu konceptų modelis transformuojamas i formalų arba pusiau formalų modelį.
- **Integravimo** metu gali būti integruojamos jau sukurtos ontologijos.
- **Realizavimas** apima konceptų modelio perkėlimą į formalią ontologijos kalbą.
- Pabaigoje priežiūra ir korekcijos.

Visus šiuos veiksmus lydi **ontologijos palaikymo** veiklos – nuolatinis žinių papildymas, dokumentavimas ir įvertinimas.

2.3.1.4 Metodas pagal L. Nemuraitę ir B. Paradauską

Tai apibendrinta metodika [12] sudaryta remiantis egzistuojančiomis ontologijų kūrimo metodikomis ir praktinė patirtimi. Metodas susideda iš 6 žingsnių:

1. **Kompetencijos klausimų suformulavimas.** Ontologijai keliami reikalavimai apibrėžiami kaip kompetencijos klausimai, į kuriuos ontologija turi gebėti atsakyti. Kompetencijos klausimai turi būti sudaromi taip, kad atsakymus į kompleksinis klausimus būtų galima gauti iš atskirų, paprastesnių kompetencijos klausimų.

2. **Bazinės ontologijos sudarymas.** Sudarant ontologiją, kaip ir sudarant ir konceptų ar duomenų bazės modelį, yra svarbu užtikrinti jos paprastumą, minimalumą, o kartu ir plečiamumą, tam kad ontologiją būtų galima nesunkiai papildyti. Ontologijos klases galima suskirstyti į *primityvias klases*, kurios neišvedamos iš kitų, ir *išvedamas klases*, kurios išvedamos iš primityvų klasių, naudojant ontologijų aksiomas arba taisyklių išvedimo mechanizmus. Primityvų klasės, yra bazinės ontologijos struktūros dalis, todėl jos turi išlaikyti griežtą klasių taksonominę hierarchiją, kurioje kiekviena klasė gali turėti tik vieną tėvinę klasę. Primityvioms klasėms šis apribojimas netaikomas, nes jų tėvinės klasės, gali kisti, priklausomai nuo apibrėžtų ontologijos aksiomų ir išvedimo mechanizmo taisyklių. Ontologijos vaikinės klasės turi būti atskiriamos (angl. *disjoint*). Objektų ir duomenų tipų savybės turi turėti apibrėžimo sritį (angl. *domain*) ir jos reikšmių intervalą (angl. *range*). Tam, kad užtikrinti ontologijos minimalumą, ontologijoje reikia naudoti minimalų klasių ir išvedimo taisyklių skaičių, užtikrinančių atsakymus į kompetencijos klausimu.
3. **Klasių egzempliorių kūrimas.** Tam kad patikrinti ar ontologijoje nėra prieštaravimų, rekomenduojama kiekvienai klasei sukurti po vieną ar daugiau klasės egzempliorių. Klasių egzemplioriai, dar vadinami individais, taip pat yra ontologijos dalimi. Klasių egzemplioriai taip pat yra analizuojami ontologijos taisyklių išvedimo mechanizmu.
4. **Ontologijos papildymas išvedimo klasėmis.** Nors išvedimo klasės padeda atsakyti į kompetencijos klausimus jos neturi būti įtraukiamos į bazinę ontologiją. Taip yra dėl to, kad išvedimo klasės yra kintanti ontologijos dalis, priklausanti nuo požiūrio, todėl jos turi būti išvedimos. Tačiau praktikoje pasitaiko situacijų, kai bazinį ontologijos modelį reikia papildyti išvedimo klasėmis tam tikro konteksto sprendimams pavaizduoti. Tokiu atveju autoriai siūlo išvedimo klasėmis papildytą ontologiją atskirti nuo bazinės ontologijos, ir šį modelį naudoti kaip atskirą ontologijos modelio atvejį tam tikrame kontekste. Galima sukurti neribotą tokių modelių skaičių, tuo tarpu bazinę ontologiją reikia atskirti ir išlaikyti vieną.
5. **Kompetencijos klausimų formalizavimas.** Kai kurie kompetencijos klausimai gali būti atsakyti remiantis ontologijos aksiomomis, kurios yra paremtos klasių klasifikacijos principais. Kitiems kompetencijos klausimams atsakyti, aksiomomis grįstų išvedimų, gali neužtekti. Sudėtingesniems klausimams atsakyti reikia taikyti

semantinio žiniatinklio taisyklių kalba *SWRL* parašytas taisyklės. Taisyklės gražinti rezultatai papildomai gali būti atrenkami *SPARQL* užklausomis.

6. **Ontologijos validavimas.** Ontologijos normalizavimas ontologijų redaktoriuose nėra analizuojamas. Ontologijos normalizavimas turi būti nagrinėjamas keliais aspektais. Normalizuota ontologija gali būti sukurta naudojant formalią konceptų *FCA* (angl. *Formal Concept Analysis*) analizę [21]. *FCA* yra dažnai taikoma žmogaus sukurtai (ne generuotos automatiškai) ontologijai validuoti. *FCA* ontologijoje leidžia turėti daugiau negu vieną tėvą, todėl tokios ontologijos tinka daugialypiam kontekstui apibūdinti. Naudojant *FCA* bazinės ontologijos validavimui, turėtų būti taikomo *FCA* reikalavimai leidžiantys vieno tėvo klasių hierarchiją. Norint patikrinti ontologijos neprieštarumą ir normalizavimą turi būti užtikrintas pakankamas klasės egzempliorių kiekis – kiekviena ontologijos klasė turi bent po vieną individą.

Visų šių metodų palyginimas pateiktas 2 lentelėje.

2 lentelė. Ontologijų kūrimo metodų palyginimas

Metodologija	„Uschold and King“	„Grüniger and Fox“	„Methontology“	Metodas pagal L.Nemuraitę ir B.Paradauską
Paaškina konceptų identifikavimo žingsnį	Taip	Taip	Taip	Taip
Aiški metodo proceso eiga	Ne	Taip	Taip	Taip
Turi ontologijos vystymo ir palaikymo stadijas	Ne	Taip	Taip	Taip
Sąsaja su šiuolaikinėmis technologijomis	Ne	Ne	Ne	Ne

Iš palyginamosios analizės matyti, kad nei vienas iš ontologijos kūrimo metodų nėra susieti su šiuolaikinėmis ontologijų aprašymo ir panaudojimo technologijomis. Daugiausiai trūkumų turi „*Uschold and King*“ metodas, kadangi jis nėra plačiai aprašytas, neturi ontologijos palaikymo ir vystymo stadijų.

Iš visų metodų labiausiai išsiskiria L.Nemuraitės ir B. Paradausko aprašytas ontologijos sudarymo metodas, kadangi jų siūloma metodika yra apibendrinta ir detalai aprašo kiekvieną iš metodo žingsnių. Be to akcentuojamas iteracinis ontologijos kūrimo procesas, kiekviename iš žingsnių atliekant ontologijos kokybės tikrinimo veiksmus. Dėl šių priežasčių magistrinio darbo realizacijos metu ontologijos sudarymui nutarta naudotis

L.Nemuraitės ir B. Paradausko siūlomų ontologijos sudarymo metodu.

2.3.2 Semantinių portalų kūrimo karkasų analizė

Programų kūrimo karkasas yra taikomųjų aplikacijų programavimo sąsaja (API), suteikianti programuotojui galimybę naudotis jau paruoštomis programinio kodo bibliotekomis. Tokių karkasų naudojimas palengviną programavimą, kadangi dalis reikalingų funkcijų jau yra realizuotos pačiame karkase.

Išanalizuoti du semantinių portalų kūrimui naudojami karkasai – *Jena* ir *RAP*. Abu karkasai yra nemokami ir atviro kodo.

Jena karkasas [9]. Tai yra populiarus *Java* platformoje veikiantis karkasas, skirtas semantinių portalų kūrimui. *Jena* karkasas turi integruotus taisyklėmis grindžiamus išvedimo formavimo mechanizmus. Jis suteikia programavimo aplinką darbui su *RDF*, *OWL* ir *SPARQL*.

Pagrindinės *Jena* karkaso ypatybės:

- Ontologijų, aprašytų *RDF*, *RDFS*, *OWL* kalba, programavimo aplinka;
- *SPARQL* užklausų kalba paremtas užklausų variklis;
- skaitymas ir rašymas *RDF* bylų parašytų *RDF/XML*, *N3*, *N-Triple* formatu;
- Galimybė integruoti taisyklėmis grindžiamus išvadų formavimo mechanizmus;
- Palaikomas ontologijos modelio saugojimas duomenų bazių valdymo sistemose *MySQL*, *PostgreSQL* ir *Oracle*.
- Galimybė integruoti į kitas atviro kodo *Java* programas.

RAP karkasas [16]. *RAP (Rdf API)* yra *PHP* kalba parašytas karkasas aplikacijų programavimo sąsajai (API) ir veiksams su *RDF* bei *OWL* atlikti. Veiksams su ontologija atlikti, modulis turi integruotą reliacinę duomenų bazę.

Pagrindinės *RAP* karkaso ypatybės:

- Ontologijų, aprašytų *RDF*, *OWL* bylose, programavimo aplinka;
- Ontologijos modelio saugojimas kompiuterio darbinėje atmintyje arba reliacinėje duomenų bazėje.
- *SPARQL* užklausų kalba paremtas užklausų variklis;
- Atviras *PHP* kodas.
- Veikia populiarių *LAMP (Linux, Apache, MySQL, PHP)* technologijų aplinkoje;

3 lentelė. Semantinių portalo kūrimo metodų palyginimas

<i>Savybė</i>	<i>Jena</i>	<i>RAP</i>
Vykdymo platformos kalba	Java	PHP
Atviras kodas	+	+
Palaikomos ontologijų aprašymo kalbos	RDF, OWL	RDF, OWL
Užklausų kalbos	RDQL, SPARQL	SPARQL
Loginio išvedimo mechanizmai	Pellet, Racer, FaCT++	Nėra
Ontologijos saugojimas DBVS	MySQL, PostgreSQL, Oracle	ADODB* DBVS
Plečiamumas	+	+
Kūrėjai teikia atnaujinimus	+	–
Dokumentacija	+	+

* *ADODB* yra duomenų bazių integracijų biblioteka parašyta *PHP* ir *Python* programavimo kalbomis, skirta palaikyti tokias duomenų bazių valdymo sistemas kaip *MySQL*, *PostgreSQL*, *Access*, *DB2*, *Foxpro*, *Informix*, *Oracle*, *MsSQL* ir kitas.

Iš lyginamosios analizės matyti, kad *Jena* karkasas siūlo didesnes ontologijų apdorojimo ir integravimo galimybes nei *RAP*. Tačiau karkaso įsisavinimas ir integravimas portale yra sudėtingesnis, dėl *Java* kalbos. Todėl, nuspręsta mokslinių publikacijų portalui realizuoti pasirinkti *RAP* portalo kūrimo karkasą, kuris yra lengviau įsisavinamas, turi visus reikalingus bazinius *OWL* apdorojimo mechanizmus ir pakankamai plačią dokumentaciją.

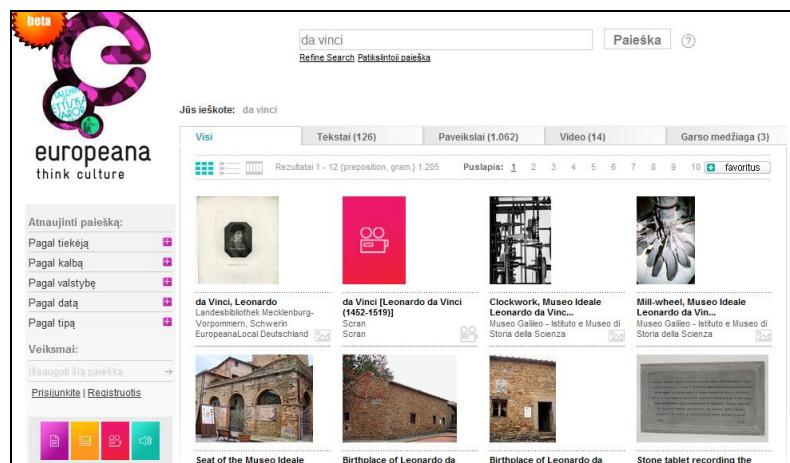
2.3.3 Egzistuojančių ontologijomis grindžiamų portalų analizė

Buvo atlikta egzistuojančių semantinių portalų analizė. Analizės metu buvo siekiama išsiaiškinti, koks duomenų pateikimo būdas yra taikomas portale, ar jis patogus vartotojui, kaip ir iš kur yra integruojami duomenys.

2.3.3.1 Virtualus muziejus „Europeana“

„Europeana“ – virtuali įvairialypės informacijos skaitmeninė biblioteka [3] skirta Europos paveldo pažinimui. Portalo vartotojai gali naudotis daugiau nei dviem milijonais knygų, žemėlapių, įrašų, fotografijų, archyvinių dokumentų, paveikslų ir filmų iš 27 Europos Sąjungos valstybių narių nacionalinių bibliotekų ir kultūros įstaigų. „Europeana“ suteikia galimybę Europos paveldą tyrinėti naudojantis visomis ES kalbomis. Portalo turinys atrenkamas iš, su portalą kuriančia organizaciją bendradarbiaujančių institucijų, žinių bazių.

„Europeana“ paieškos rezultatų langas parodytas 2.2 paveikslėlyje.



2.2 pav. Portalo „Europeana“ paieškos rezultatų langas

Paieškos kriterijus atitinkantys rezultatai suskirstomi pagal jų tipą. Rezultatų paiešką galima susiaurinti nurodant papildomas sąlygas – tiekėją, kalbą, valstybę, datą ir tipą.

Kiekvienas elementas yra semantinio tinklo mazgas, todėl galima peržiūrėti su juo susijusius meta duomenis, šie duomenys priklauso nuo ontologijos aprašo, todėl skirtingų elementų skiriasi ir metaduomenų tipai. Portalo semantiniai elementai (*ESE*) yra sudaryti iš metaduomenų standarto „*Dublin Core*“ elementų, išraiškų poaibio ir iš papildomo dvylikos elementų rinkinio, kuris yra reikalingas patenkinti „*Europeana*“ funkcijoms.

Portale pateikiama metaduomenų semantikos specifikaciją kurios privalu laikytis apipavidalinant turinį, kurį norima skelbti „*Europeana*“ portale.

„*Dublin Core*“, tiek „*Europeana*“ elementai gali turėti kalbos atributą. *XML:lang* nurodo metaduomenų kalbą, kuri reikia naudoti visuose elementuose, kur tik jis yra įmanomas. Pavyzdžiui:

```
<dc:subject xml:lang="en">weather</dc:subject>
```

```
<dc:subject xml:lang="lt">oras</dc:subject>
```

Kiekvienas „*Europeana*“ elementas yra specifikuojamas tokiais atributais:

- **Element Name** (elementas): formalus elemento pavadinimas.
- **Namespace** (kilmė): elemento šaltinio kilmė: dc, dcterms arba Europeana.
- **IRI**: internacionalizuotas išteklius identifikatorius naudojamas unikaliai elemento identifikacijai.
- **Label** (reikšmė): elemento paaiškinimas.
- **Definition** (apibrėžimas): elemento apibrėžimas, remiantis *DC* elemento apibrėžimu, jei tinkamas.
- **Europeana note** (pastaba): pastabos, susijusios su elemento naudojimo

„Europeana“ sistemoje.

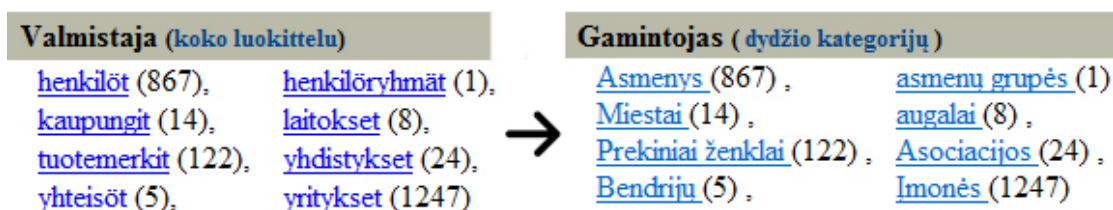
- **Obligation & Occurrence** (privalomumas/būtinumas): iš anksto numatytų reikalavimų arba būtinumų nurodymai elementui ir elemento dažnumas (kaip dažnai elementas gali būti pakartojamas) metaduomenyse.
- **Simple search** (paprasta paieška): nurodo, kurie elementai yra naudojami paprastosios paieškos funkcijai.
- **Facet** (aspektas): nurodo, ar elementas yra naudojamas išplėstinėje paieškoje ir kuriuose laukuose: *Date, Type, Provider, Country* ir *Language*.
- **Timeline** (laiko juosta): nurodo, ar elementas naudojamas laiko juostos naršymo funkcijoje.
- **Advanced search** (sudėtinė paieška): nurodo, kurie elementai yra naudojami sudėtinėje paieškoje: *Title, Creator, Date* ir *Subject*.
- **Full search result display** (rezultatų vaizdavimas): nurodo, ar elemento turinys yra įtrauktas į „Europeana“ skaitmenizuotų objektų išsamios paieškos rezultatų vaizdavimą.
- **Example** (pavyzdys): XML sintaksės pavyzdys.

2.3.3.2 Suomijos istorijos virtualus muziejus „MuseumFinland“

Tai portalas [8] suomių kalba, kuriame galima susipažinti su Suomijos istoriniu paveldu. Portalas agreguoja ir atrenka duomenis apie įvairių tipų artefaktus – piešinius, skulptūras, literatūros darbus, biografijas.

Portalo žinių bazę, sudaryta iš įvairių Suomijos muziejų, bibliotekų ir kitų archyvų transformuotų duomenų bazių ir ontologijų.

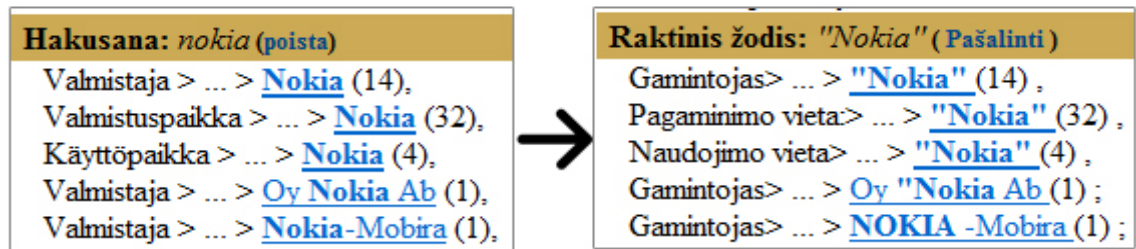
Tituliniame „MuseumFinland“ puslapyje pateikiamos kategorijos, kurias galima naršyti (2.3 pav.). Kategorijos elementų skliaustuose nurodoma kiek jis turi žemesnio lygio kategorijų.



2.3 pav. Kategorijų grupavimas (kairėje suomių kalba, dešinėje lietuviškas vertimas)

Paieškos laukelyje įvedus raktažodį, paieška vykdoma tikrinant kategorijos, kas atitinka ontologijos klasę, pavadinimo sutapimą, o taip pat, elementų meta duomenų

aprašymus. Visi paieškos rezultatai yra grupuojami į kategorijas. Pavyzdžiui, raktažodžio „Nokia“ paieškos rezultatų kategorijų struktūra parodyta 2.4 paveikslėlyje.



2.4 pav. Paieškos rezultatų grupavimas pagal kategorijas (kairėje suomių kalba, dešinėje lietuviškas vertimas)

Elemento peržiūros lange (2.5 pav.) pateikiama aprašomoji jo informacija. Dešinėje portalo lango pusėje yra sugeneruotos semantinės nuorodos, susijusios su peržiūrimu turiniu pvz. „*Taip pat naudojamas vietose*“ (suom. *Sama käyttöpaikka*) ir pan. Šių nuorodų tikslas yra suteikti vartotojui galimybę rasti objektus, įvairiais ryšiais susijusius su peržiūrimu elementu.

2.5 pav. Elemento peržiūros langas

Kad suprasti, kaip veikia portalas „*MuseumFinland*“, reikia susipažinti su jo struktūra. Portalas sukurtas naudojantis semantinių portalų kūrimo karkasu *OntoViews*. Šis karkasas skirtas interaktyviam darbui su *RDF* kalba išreikštomis ontologijomis.

Semantinių paslaugų karkasas *OntoViews* buvo sukurtas tam, kad semantinis portalas būtų lengvai pritaikomas naujoms ontologijoms, lengvai pritaikomas naujam vartotojo

interfeisui bei atvaizdavimui, lengvai plečiamas, panaudojant nepriklausomus funkcinius modulius.

OntoViews karkasas turi tik kelis *HTML* tipo failus, iš kurių nei vienas neformuoja portalo vaizdo. Gavęs užklausa iš vartotojo naršyklės *OntoViews* transformuoja duomenis iš *RDF* rinkmenų į *HTML* dokumentą per *XLST* transformacijas. Todėl galima teigti, kad visas *OntoView* karkasų grindžiamas portalas yra dinamiškas.

Karkasas OntoViews susideda iš trijų pagrindinių komponentų:

1. *OntoViews-C* grafinės vartotojo sąsajos generatoriaus, grįsto *XLST* transformacijomis;
2. *Ontogator* semantinės paieškos algoritmo;
3. *Ontodella* semantinių sąsajų generavimo logikos.

OntoViews-C struktūrinis elementas yra atsakingas už portalo grafinę vartotojo sąsają ir interaktyvius veiksmus tarp vartotojo ir sistemos. *OntoViews-C* gauna *RDF* kodo fragmentus iš kitų sistemos elementų (*Ontogator* ir *Ontodella*) ir jį transformuoja į *HTML* dokumentą. Šias transformacijas vykdo *Apache Cocoon XML* duomenų apdorojimo karkasas. Tai yra specialus programinės įrangos karkasas *XML* duomenų apdorojimui ir atvaizdavimui interneto puslapiuose skirtingais formatais. Dėl to *OntoViews* portalą galima naršyti naudojantis mobiliuoju telefonu.

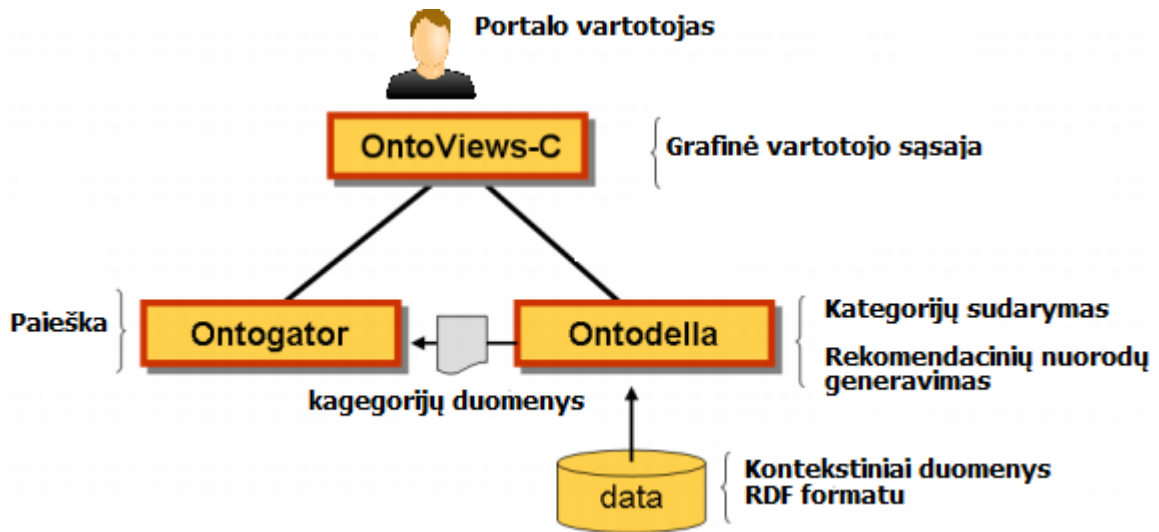
Ontogator yra semantinės paieškos procesorius atskiriantis paieškos logiką nuo vartotojo interfeiso. *Ontogator* vykdo paiešką semantiniame turinyje, o jos rezultatus *RDF* formatu perduoda *OntoViews-C* moduliui, kur jie yra transformuojami ir atvaizduojami *HTML* dokumentu. Šitaip vartotojas naudojantis grafine vartotojo sąsaja gali vykdyti paiešką semantiniuose duomenyse.

Išvedimų procesorius *Ontodella* reikalingas ryšių tarp objektų nustatymui. *Ontodella* atsakinga už šias funkcijas:

Kategorijų vaizdavimas. *OntoViews* portale turinio kategorijos yra hierarchiškai struktūrizuotos. Kiekviena kategorija yra susieta su kitų kategorijų poaibiu ir elementais. Kai kategorijos sugeneruotos jomis galima naviguoti modulio *Ontogator* pagalba.

Rekomendacinių nuorodų generavimas paremtas susiejimo taisyklėmis (angl. *linking rules*). Kiekviena tokia taisyklė gali turėti daug sąlygų, kurias nustato dalykinės sirties specialistas. Susiejimo taisyklė yra aprašoma tokia forma: *predicate(SubjectURI, TargetURI, Explanation)*. Rekomendacinės nuorodos yra generuojamos, kai gaunama *HTTP GET* užklausa iš pagrindinio *OntoViews* modulio.

Karkaso *OntoViews* struktūra pavaizduota 2.6 paveikslėlyje.



2.6 pav. Portalo *OntoViews* struktūra [8]

Visi šie trys komponentai yra sukurti taip, kad būtų kuo mažiau priklausomi vienas nuo kito. Tai reiškia, kad, pavyzdžiui, paieškos komponentas *Ontogator* gali būti panaudotas bet kokioje portale, turinčiame *RDF* kalba aprašytą ontologiją.

Viena iš problemų, su kuria gali susidurti *OntoViews* vartotojai, yra kalba. *OntoViews* metaduomenys ir pavyzdinis portalas „*MuseumFinland*“ pateikiami suomių kalba, o jos nesuprantantiems sunku analizuoti *OntoViews* karkaso veikimą.

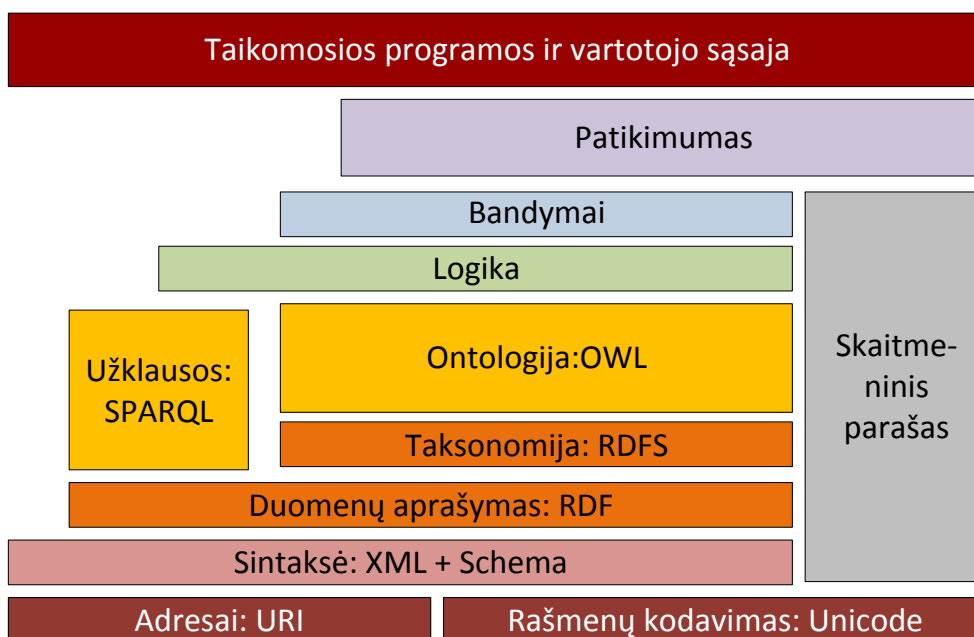
OntoViews pakete yra visi reikalingi duomenys ir dokumentacija analogiško „*MuseumFinland*“ portalo paleidimui savo kompiuteryje. Tačiau nėra dokumentacijos aprašančios *OntoViews* karkaso pritaikymą darbui su savo turiniu. *OntoViews* reikalauja specialių taisyklių rinkinių, kuriomis remiantis užtikrinama portalo veikimo logika. Tokių taisyklių rašymas jau egzistuojančiam semantiniam turiniui reikalautų daug laiko.

2.4 *Siekiamos metodikos kūrimo uždavinio apibrėžimas*

Sudaroma metodika turi apimti portalo struktūrinius elementus (architektūrą, ontologiją) ir kūrimo procesą.

2.4.1 *Galima semantinio portalo struktūra*

Semantinio portalo realizacijos technologijų architektūrą galima pavaizduoti diagrama (2.7 pav.).

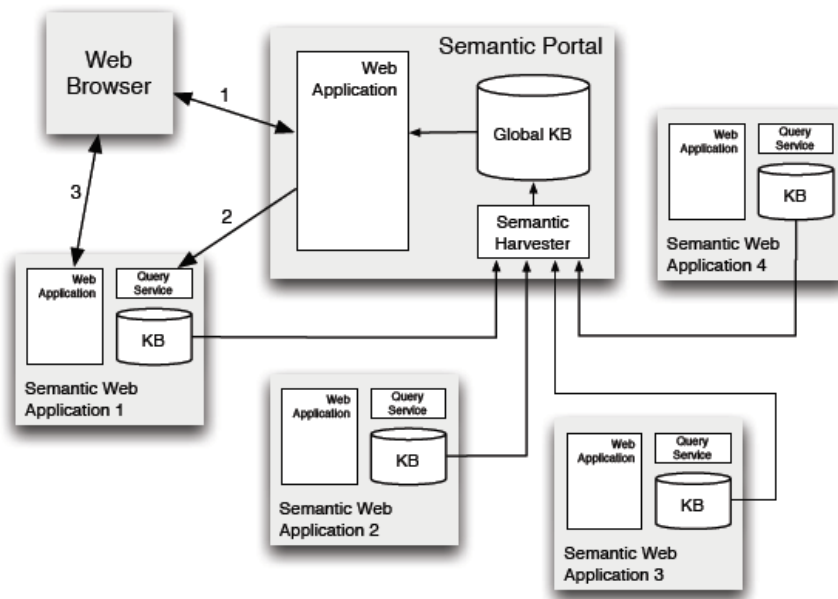


2.7 pav. Semantinio portalo technologijų architektūra

Semantinio portalo technologijų pamatą sudaro rašmenų kodavimo standartas *Unicode* ir *URI* adresai. Standartas *Unicode* naudojamas užtikrinti, kad duomenims būtų naudojami standartiniai simbolių rinkiniai, o *URI* adresas skirtas identifikuoti semantinio tinklo objektą. Sintaksės sluoksnis užtikrina, kad semantiniams aprašams būtų naudojama *XML* bendros paskirties duomenų struktūrų, bei jų turinio aprašomoji kalba. Skaitmeninio parašo lygmuo skirtas dokumentų pakeitimams aptikti. *RDFS* sluoksnis skirtas bazinėms semantinėms žinioms, apie resursą, išreikšti. Ontologijos *OWL* sluoksnis praplečia *RDFS* sluoksnį sudėtingesnėmis išraiškos struktūroms. Šiame sluoksnyje slypi pagrindinė semantinio tinklo galia. Iš ontologijos *RDFS* ir *OWL* sluoksnių informacija išgaunama *SPARQL* užklausų pagalba. Logikos lygmuo skirtas išvedimo taisyklių aprašymui, o bandymų lygmuo šių taisyklių vykdymui. Patikimumo sluoksnis įvertina – pasitikėti bandymų rezultatais ar ne.

Technologijų architektūros piramidę užbaigia vartotojo taikomosios programos ir grafinė vartotojo sąsaja. Šis sluoksnis reikalingas tam, kad vartotojas galėtų naudotis semantiniu portalu.

Vienas iš galimų semantinio portalo loginės architektūros pavyzdžių pateiktas 2.8 paveikslėlyje.



2.8 pav. Semantinio portalo architektūros pavyzdys [8]

Semantinis portalas savo globalią žinių bazę sudaro iš dalykinės srities viešai prieinamų ontologijų. Portalas surenka dalykinės srities informacijos fragmentus (pavyzdžiui kultūros renginių antraštes) ir išsaugoja juos savo duomenų bazėje.

Vartotojas atsidaręs tokį portalą gali ne tik peržiūrėti iš įvairių šaltinių surinktą informaciją, bet ir vykdyti juose paiešką. Vartotojui rezultatai gali būti pateikiami semantiniam portalui būdingu stiliumi, todėl jis net nesupras, kad tikrasis informacijos šaltinis yra visai kitas portalas (2.8 pav., 1 rodyklė).

Informacijos fragmentų surinkimą atlieka semantinio portalo agregatorius (2.8 pav. *Semantic Harvester*), kuris kas tam tikrą laiką patikrina pirminius informacijos šaltinius ir atnaujiną savo duomenų bazę. Jeigu pirminiuose šaltiniuose informacija atnaujinama labai dažnai (pavyzdžiui renginio likusių bilietų skaičius) semantiniame portale duomenys gali būti jau pasenę.

Vienas iš galimų būdų šiai problemai spręsti yra nuo *SPARQL* užklausų vykdymas pirminiuose šaltiniuose (2.8 pav., 2 rodyklė). Tokios užklausos turi būti vykdomos visada, kai vartotojas pareikalauja tokios informacijos. *SPARQL* užklausas vykdo semantinis portalas ir gautus rezultatus pateikia vartotojui.

Vartotojui norint susižinoti tikrąjį informacijos šaltinį ir gauti daugiau informacijos, portalas iš savo sukauptos žinių bazės suranda šį adresą ir pateikia jį lankytojui (2.8 pav., 3 rodyklė). Tai, pavyzdžiui, galėtų būti vartotojo noras užsisakyti bilietus į pasirinktą kultūros renginį.

Tai yra tik vienas iš galimų semantinio portalo infrastruktūros pavyzdžių. Semantiniai

portaluose galima naudoti semantines tinklo paslaugas, kurios išplečia portalo funkcionalumą.

2.4.2 Portalo kūrimo procesas naudojant ontologiją

Nuo įprasto portalo ontologija grįstas portalas skiriasi tuo, kad yra reikalinga dalykinės srities ontologija. Ontologija pati savaime yra statinis sistemos elementas. Todėl reikalingi mechanizmai galintys panaudoti ontologijoje saugomas žinias. Tai galima atlikti dvejais būdais:

1. Ontologijoje vykdyti *SPARQL* užklausas ir jų pagalba gauti struktūrizuotus duomenis;
2. Pasitelkti išvedimo variklius (angl. *reasoning services*) ir gauti faktus paremtus apibūdinančia logika (ang. *description logic*).

Pirmasis variantas labai panašus į *SQL* užklausų vykdymą reliacinėse duomenų bazėse siekiant gauti susietus duomenis.

Antrasis yra paremtas ontologijos taisyklėmis grindžiamais išvedimais. Tai yra automatizuoti mechanizmai, kurie išveda faktus, apie ontologijoje saugomą dalykinės srities struktūrą, pagal galiojančias aksiomas. Pavyzdžiui, jei ontologijoje užfiksuota, kad asmuo yra straipsnio autorius, išvedimo variklis galėtų konstatuoti faktą, kad šis asmuo yra rašytojas. Šių mechanizmų panaudojimas gali būti plačiai pritaikomas ontologijų analizei ir teikia didelių pranašumų ontologija grindžiamų portalų panaudojime.

Toliau pateikti ontologija grindžiamo portalo kūrimo žingsniai:

- Reikalavimų apibrėžimas
- Dalykinės srities analizė ir ontologijos sudarymas.
- Faktinės informacijos surinkimas ir talpinimas dalykinės srities ontologijoje arba atskiroje žinių bazėje.
- Užklausų, kuriose naudojami dalykinės srities ontologijos elementai, realizavimas.
- Vartotojo sąsajos projektavimas ir realizavimas.
- Kitų programinės sistemos dalių projektavimas ir realizavimas.

Pirmas ir trečias žingsniai yra būdingi ontologija grįstoms informacinėms sistemoms. Pirmame žingsnyje atliekamas ontologijos sudarymas, o trečiame – realizuojami ontologijos panaudojimo mechanizmai.

2.5 *Analizės išvados*

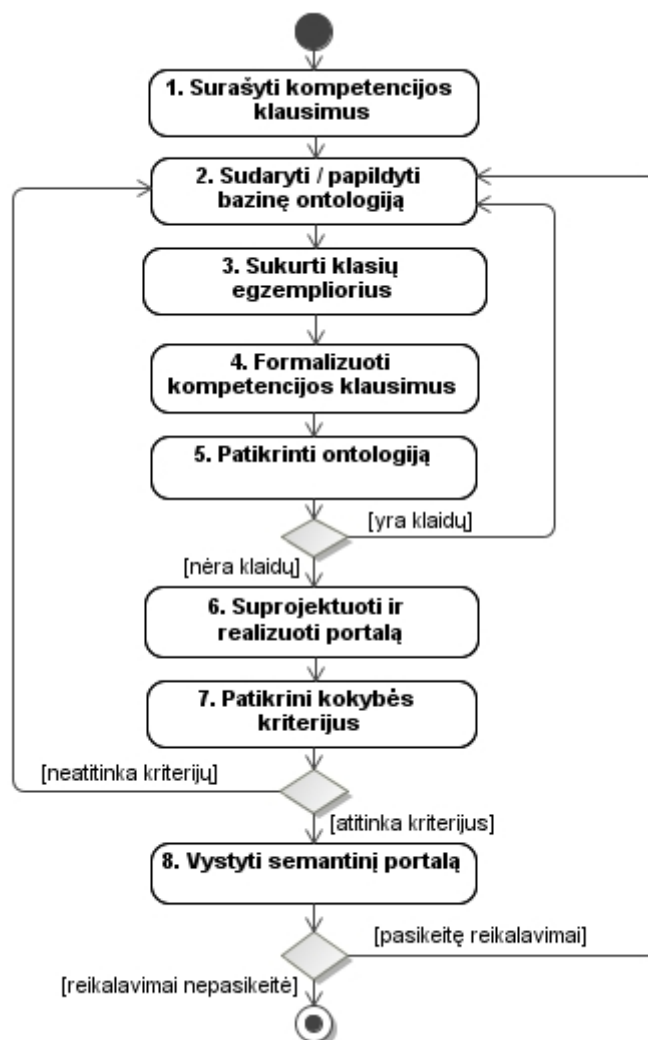
1. Ontologijų panaudojimas interneto portaluose padeda kompiuteriams suprasti informacijos prasmę taip prisidėdami prie informacijos žvalgymo ir analizavimo.
2. Išanalizavus ontologijų aprašymo kalbas ir technologijas, prieita išvada naudoti *OWL* ontologijų aprašymo kalbą, kadangi ji užtikrina žinių išraiškingumą; nemokamą *Protégé* ontologijų redaktorių ir atviro kodo *RAP* karkasą.
3. Atlikus literatūroje aprašytų ontologijų kūrimo metodų analizę pastebėta, kad nėra metodo nuosekliai aprašančio procesą, nuo ontologijų sukūrimo iki jų panaudojimo šiuolaikiniuose interneto portaluose.

3. Ontologijomis grindžiamų portalų kūrimo metodika

3.1 Metodo eiga

Ontologija grindžiamo portalų kūrimo metodika, remiasi analitinėje dalyje išanalizuota apibendrinta ontologijų kūrimo metodika [12]. Ši metodika pasirinkta dėl to, kad apima aibę kriterijų ir taisyklių, kurios leidžia užtikrinti sudarytos ontologijos kokybę. Siūlomas metodas pasižymi tuo, kad ontologija grįsto portalų kūrimo procesas yra glaudžiai susijęs su ontologijos sudarymo procesu. Taip yra dėl to, kad sudarant ontologiją, reikia atlikti nuolatinį jos teisingumo, pilnumo ir kitus patikrinimus. Priklausomai nuo pasirinktų ontologijos kūrimo priemonių, dalis ontologijos patikrinimų yra įmanomi tik portale.

Metodikos aprašomas procesas pavaizduotas 3.1 paveiksle.



3.1 pav. Semantinio portalų kūrimo procesas

3.2 Ontologijos reikalavimų suformulavimas

Ontologijai keliamus reikalavimus galima apibrėžti kaip kompetencijos klausimus, į kurios ta ontologija turi gebėti atsakyti [5], [12].

Kompetencijos klausimai yra suvokiami kaip neformalūs klausimai, pavyzdžiui „*kokie publikacijų autoriai turi mokslų daktaro laipsnį?*“. Šie kompetencijos klausimai vėliau turi būti formalizuotai aprašomi ontologijoje, panaudojant aksiomas ir išvedimo mechanizmų taisykles. Kompetencijos klausimai turi būti sudaromi taip, kad atsakymus į kompleksinius klausimus būtų galima gauti iš atskirų, paprastesnių klausimų. Pavyzdžiui, anksčiau pateiktą klausimą, galime išskaidyti į du paprastesnius kompetencijos klausimus „*kas yra publikacijų autoriai?*“ ir „*kokie autoriai turi mokslų daktaro laipsnį?*“. Mums rūpimas atsakymas bus šių kompetencijos klausimų atsakymų aibės sankirta.

Pavyzdžiui, mokslinių publikacijų portalas turi atsakyti į klausimus 1) kokie asmenys yra knygų autoriai? 2) kokie asmenys turi mokslų daktaro laipsnį? 3) Kas yra X publikacijos autoriai?

Kompetencijos klausimų sudarymo etapą galima prilyginti su įprastų informacinių sistemų reikalavimų specifikavimo žingsniu, kai yra renkami ir specifikuojami veiklos reikalavimai, sistemos funkcijoms ir apribojimams nustatyti.

3.3 Ontologijos sudarymas

Šio žingsnio esmė yra sudaryti ontologiją, kuri atsakytų į iškeltus kompetencijos klausimus.

Sudarant ontologiją, kaip ir sudarant ir konceptų ar duomenų bazės modelį, yra svarbu užtikrinti jos *paprastumą*, *minimalumą*, o kartu ir *plečiamumą*. Tai padaryti nėra lengva, tačiau lengviausias kelias tai pasiekti yra šių taisyklių laikymasis:

1. Sudaryti bazinę ontologiją be išvedamų klasių. Ontologijos klases galima suskirstyti į *primityvias klases*, kurios neišvedamos iš kitų, ir *išvedamas klases*, kurios išvedamos iš primityvių klasių, naudojant ontologijų aksiomas arba taisyklių išvedimo mechanizmus. Nors išvedimo klasės padeda atsakyti į kompetencijos klausimus, jos neturi būti įtraukiamos į bazinę ontologiją. Taip yra dėl to, kad išvedamos klasės yra nuolat kintanti ontologijos dalis, priklausanti nuo išvedimo mechanizmo taisyklių. Be to, išvedimo klasė gali turėti daugiau negu vieną tėvinę klasę, kas savaime prieštarauja bazinės ontologijos sudarymo taisyklėms. Primityvų klasės yra bazinės ontologijos struktūros dalis, todėl jos turi išlaikyti griežtą klasių taksonominę hierarchiją, kurioje kiekviena klasė gali turėti tik vieną tėvinę klasę. Išvedamoms klasėms

šis apribojimas netaikomas, nes jų tėvinės klasės, gali kisti, priklausomai nuo apibrėžtų ontologijos aksiomų ir išvedimo mechanizmo taisyklių.

2. Atskirti ontologijos vaikinės klasės sukuriant atskyrimo (angl. *disjoint*) aksiomas.
3. Nustatyti objektų ir duomenų tipų savybių apibrėžimo sritis (angl. *domain*) ir jų reikšmių intervalus (angl. *range*).
4. Užtikrinant ontologijos minimalumą, ontologijoje naudoti minimalų klasių ir išvedimo taisyklių skaičių, užtikrinančių atsakymus į kompetencijos klausimus.

3.4 Klasių egzempliorių kūrimas

Klasių egzemplioriai, dar vadinami individualais, taip pat yra ontologijos dalimi. Klasių egzemplioriais naudojasi ontologijos taisyklių išvedimo mechanizmai. Norint patikrinti ontologijos ar ontologijoje nėra prieštaravimų, reikia nuolat peržiūrėti išvedamus klasių egzempliorius. Tam turi būti užtikrintas pakankamas klasių egzempliorių skaičius – kiekviena ontologijos klasė turi turėti bent po vieną egzempliorių.

Ontologijoms, kurios turi didelį skaičių individų, juos rekomenduoja saugoti atskirai nuo pačios ontologijos, pavyzdžiui, reliacinėje duomenų bazėje [20].

3.5 Kompetencijos klausimų formalizavimas

Kompetencijos klausimai, ontologijoje turi virsti formalizuotomis aksiomų iš išvedimo taisyklių išraiškomis, kurios teikia atsakymus į šiuos klausimus. Kai kurie iš kompetencijos klausimų gali būti lengvai formalizuojami remiantis ontologijos klasių klasifikacijos principais. Pavyzdžiui $Rašytojas \equiv Asmuo \sqcap \exists yra_parases.publikacija$

Kitai kompetencijos klausimų daliai atsakyti aksiomomis grįstų išvedimų gali neužtekti. Sudėtingesniems klausimams atsakyti reikia taikyti semantinio portalo taisyklių kalba *SWRL* parašytas taisykles. Pavyzdžiui, norit išvesti autorius, savo publikacijoje cituojančius kitus, rašoma tokia *SWRL* taisyklė:

$cituojama_publikacijoje(?y, ?z) \sqcap paraše_publikacija(?x, ?y) \sqcap parašyta_autorius(?z, ?v) \Rightarrow cituoja_autoriu(?v, ?x)$
--

3.6 Semantinis ontologijos pilnumo tikrinimas

Reikia nepamiršti, kad semantinio ontologijos pilnumo negalima prilyginti ontologijos, kaip koncepcinio modelio, užbaigtumui. Kaip ir minėta, pagrindiniai kriterijai semantiniam pilnumui nustatyti yra ontologijos gebėjimas atsakyti į iškeltus kompetencijos klausimus.

Čia galioja ir atvirkštinė įvertinimo taisyklė – ontologija ir jos individai turi kelti kompetencijos klausimus.

3.7 Portalų architektūros projektavimas ir portalų realizacija

Šiame žingsnyje atliekamas semantinio portalų architektūros projektavimas ir portalų realizacija. Šis žingsnis nėra esminė metodikos dalis. Portalų architektūros sudarymui galima naudoti įvairius metodus ir priemones (pvz. *UML* modelius). Yra naudinga akcentuoti tik bendrą portalų architektūros sudarymą, paremtą sluoksniuose trijų lygių architektūra:

- Informacijos lygmuo
- Veiklos logikos lygmuo
- Vartotojo sąsajos lygmuo

Ši sluoksniuose architektūros schema tinka taikyti daugumoje programinės įrangos sistemų, kadangi tiek vartotojo sąsaja, tiek duomenys yra praktiškai bet kurioje programinėje įrangoje. Toks architektūros organizavimas leidžia realizuoti nesunkiai praplečiamą ir palaikomą programinę įrangą bei lengvai paskirstyti darbus.

Informacijos lygmuo. Šiame lygmenyje ontologija grindžiamo portalų architektūroje turi būti analizuojama ontologija. Taip pat aprašomos reikalingos duomenų bazių struktūros. Pavyzdžiui, atliekant mokslinių publikacijų portalų realizaciją, buvo panaudota duomenų bazė ontologijos savybėms apskaičiuoti.

Veiklos logikos lygmuo. Šis lygmuo skirtas informacijos srautų valdymui. Realizuojami visi reikalingi duomenų išvedimo iš ontologijos mechanizmai.

Pavyzdžiui, atliekant mokslinių publikacijų portalų realizaciją, pasirinktas ontologijų skaitymo ir analizės įrankis *RAP* nepalaikė ontologijų *SWRL* taisyklių. Todėl vietoje jos teko parašyti *SPARQL* užklausą.

Grafinės vartotojo sąsajos lygmuo. Projektuojant portalų grafinę vartotojo sąsają reikia įvertinti ontologijos struktūrą ir jose saugomų duomenų apimtį. Pavyzdžiui, jei planuojama portalų navigacijos planą pagrįsti ontologijos klasėmis ar individualiais, turi būti įvertinta, ar tokia vartotojo sąsaja nebus perkrauta, ar viską pavyks pavaizduoti ekrane.

3.8 Semantinis portalų tikrinimas

Realizavus portalą, atliekamas semantinis portalų tikrinimas pagal jo teikiamus atsakymus į kompetencijos klausimus. Kaip ir ontologijos semantinio tikrinimo metu, šiame žingsnyje taip pat tikrinama, ar portalas atsako į kompetencijos klausimus. Skirtumas yra tas, kad ontologijos kūrimo metu ontologijos kūrimo priemonėmis, ne visada gali pavykti

atsakyti iškart į visus kompetencijos klausimus. Semantinio portalo tikrinimo metu turi būti atsakyti visi kompetencijos klausimai.

Jei portalas neatsako į visus kompetencijos klausimus, turi būti grįžtama į ontologijos koregavimo žingsnį (3.1 pav.).

3.9 Ontologijos ir portalo vystymas

Aplinka neišvengiamai keičiasi, todėl natūralu, kad gali prireikti koreguoti sudarytą ontologiją. Atlikus ontologijos koregavimą, reikia įvertinti, kiek tai įtakos portalą. Gali būti, kad po ontologijos koregavimo reikės perdaryti portalą ir iš naujo atlikti patikrinimo veiksmus.

Portalo ontologijos plečiamumui svarbią įtaką turi jo moduliškumas (angl. *modularity*). Nuo ontologijos moduliškumo priklauso ontologijos atnaujinimo galimybės, norint ją papildyti naujomis klasėmis ir savybėmis, nepažeidžiant sudarytos struktūros. Teisingai sudaryta bazinė ontologija užtikrina, kad ją bus galima lengvai koreguoti. Ši savybė ypač svarbi didelėse ontologijose ir ontologijose, su kurių atskiromis dalimis dirba skirtingi žmonės.

4. Mokslinių publikacijų portalo reikalavimų specifikacija ir projektas

4.1 Reikalavimų specifikacija

Skyrelyje pateikiami funkciniai reikalavimai, apimantys panaudojimo atvejų diagramas, jų specifikacijas ir sudarytą dalykinės srities ontologijos modelis.

Kadangi bus kuriama metodika, padėsianti projektuotojams nuo pat pradžių sukurti, semantinį portalą, pagrindiniai vartotojai bus veiklos analitikas, ontologijos ekspertas, portalo projektuotojas ir jo vartotojas.

Veiklos analitikas – analizuoja dalykinę sritį, specifikuoja reikalavimus.

Ontologijos ekspertas – ontologijos sudarytojas, analitikas.

Projektuotojas – ontologija grįsto portalo kūrėjas.

Portalo vartotojas – vartotojas naudojantis ontologijos pagrindu sukurta portalą informacijos peržiūrai, paieškai ir kitiems veiksams atlikti

Veiklos analitiko užduotis yra detalai išanalizuoti veiklos sritį ir sudaryti reikalavimus projektuojamai sistemai.

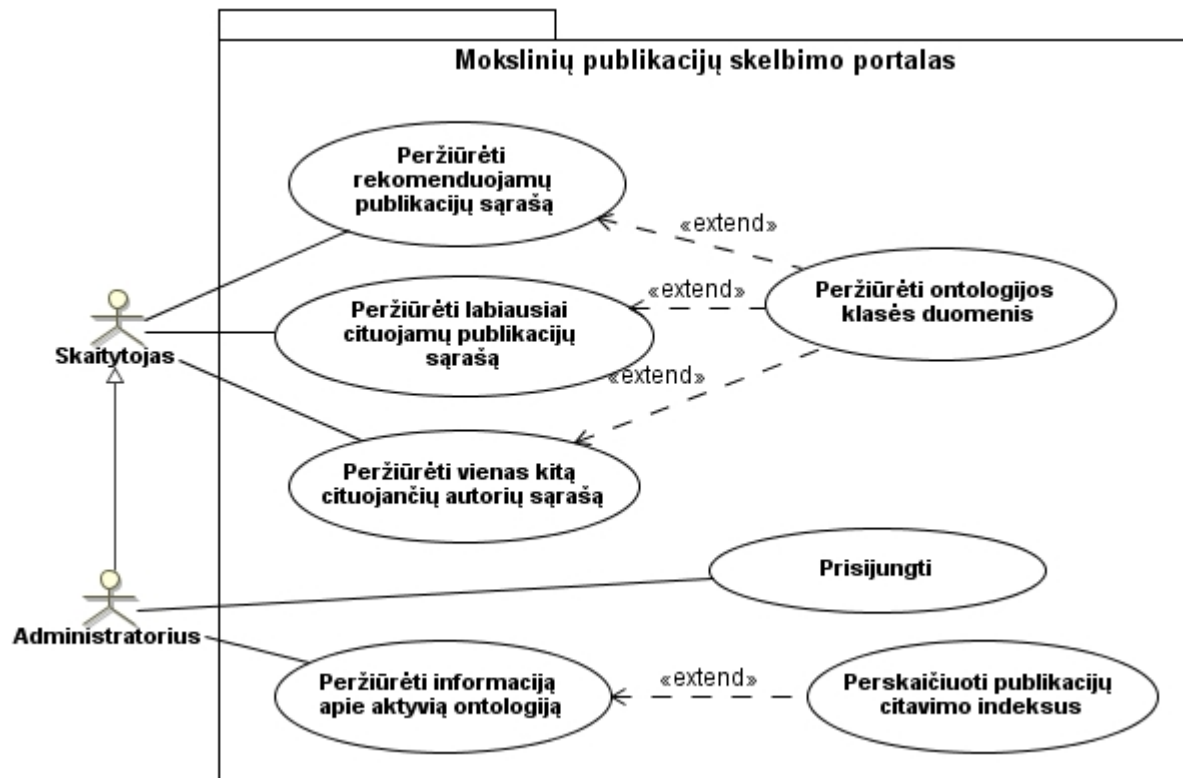
Ontologijos ekspertas pagal dalykinės srities analizę turi sugebėti sukurti ontologiją, kuri integruosis į projektuojamą sistemą. Ontologijos ekspertui gali būti neaišku, kada verta kurti savo ontologiją, o kada naudotis jau sukurta. Taip pat kokie yra ontologijų kūrimo metodai, standartai ir įrankiai.

Projektuotojo tikslas yra sukurti efektyvią ontologija grįsta informacinę sistemą, kuria galėtų naudotis vartotojas. Projektuotojui gali būti neaiškų kokia yra semantinių portalų architektūra ir jų realizacijos metodai.

Sistemos vartotojo tikslas – efektyvus darbas naudojantis portalu. Tai gali būti informacijos paieška, jos apdorojimas ir kitokie veiksmai. Vartotojas gali būti visiškai neįsigilinęs į dalykinę sritį, todėl naudojimasis tokiu portalu bus komplikuoatas. Vartotojui problemos kyla ir dėl to, kad portalo veikimo logika dažnai yra realizuotas remiantis portalo kūrėjo supratimu.

4.1.1 Panaudojimo atvejų diagrama

Projektuojamos sistemos panaudojimo atvejų diagrama pateikta 4.1 paveiksle.



4.1 pav. Portalo panaudojimo atvejų diagrama

4.1.2 Panaudojimo atvejų specifikacijos

Sistemos panaudojimo atvejų specifikacijos pateiktos 4 – 8 lentelėse.

Lentelė 4. PA „Peržiūrėti rekomenduojamų publikacijų sąrašą“ specifikacija

PA „Peržiūrėti rekomenduojamų publikacijų sąrašą“		
Tikslas. Pateikti rekomenduojamų publikacijų sąrašą portalo lankytojui.		
Prieš sąlyga	–	
Aktorius	Skaitytojas, Administratorius	
Sužadinimo sąlyga	Vartotojas atsidaro pagrindinį (namų) portalo langą	
Susiję panaudojimo atvejai	Išplečia PA	–
	Apima PA	–
	Specializuoja PA	–
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. Vartotojas interneto naršyklėje įveda portalo adresą		Sistema užkrauna numatytą ontologiją į atmintį. Sistema prisijungia prie reliacinės duomenų bazės Sistema sugeneruoja navigacijos meniu struktūrą ir bendrą portalo vaizdą
Po sąlyga:		Vartotojo interneto naršyklės lange pateiktas interaktyvus portalo pagrindinio lango vaizdas
Alternatyvūs scenarijai		
1.a Vartotojas interneto naršyklėje įveda teisingą portalo šakninį		Sistema sugeneruoja navigacijos meniu struktūrą ir bendrą portalo vaizdą, bei išveda pranešimą apie

adresą, nurodydamas neteisingą portalo resurso adresą	neegzistuojantį portalo puslapį ar resursą
---	--

Lentelė 5. PA „Peržiūrėti labiausiai cituojamų publikacijų sąrašą“ specifikacija

PA „Peržiūrėti labiausiai cituojamų publikacijų sąrašą“		
Tikslas. Portalo lankytoji pateikti didžiausią citavimo indeksą turinčių publikacijų sąrašą, išrikiuotą pagal citavimo indekso rodiklį.		
Prieš sąlyga	Sistemoje atliktas publikacijų citavimo indekso skaičiavimo algoritmas.	
Aktorius	Skaitytojas, Administratorius	
Sužadavimo sąlyga	Vartotojas portalo pagrindiniame lange pasirenka meniu punktą labiausiai cituojamų publikacijų sąrašui peržiūrėti	
Susiję panaudojimo atvejai	Išplečia PA	–
	Apima PA	–
	Specializuoja PA	–
Pagrindinis įvykių srautas		
Sistemos reakcija ir sprendimai		
1. Vartotojas peržiūri labiausiai cituojamų publikacijų sąrašą	Sistema pateikia visų ontologijoje saugomų publikacijų sąrašą, išsirikiuota citavimo indekso mažėjimo tvarka	
2. Vartotojas iš sąrašo pasirenka norimą publikacijų kategoriją ir paspaudžia [Filtruoti]	Sistema pateikia filtruotą labiausiai cituojamų publikacijų sąrašą, atrenkant tik tas publikacijų kategorijas, kurias nurodė vartotojas	
Po sąlyga:	Pateiktas labiausiai cituojamų publikacijų sąrašas, atitinkantis vartotojo nustatytą filtrą.	
Alternatyvūs scenarijai		
2a. Vartotojas iš sąrašo filtre nurodo kategoriją(-as), pagal kurią(-ias) nėra nei vienos publikacijos	Išvedamas pranešimas, kad pagal pasirinktą filtrą, nerasta nei viena publikacija	

Lentelė 6. PA „Peržiūrėti vienas kitą cituojančių autorių sąrašą“ specifikacija

PA „Peržiūrėti vienas kitą cituojančių autorių sąrašą“		
Tikslas. Portalo lankytoji pateikti autorių, savo kūriniuose cituojančių vienas kitą, sąrašą.		
Prieš sąlyga	Sistemoje atliktas publikacijų citavimo indekso skaičiavimo algoritmas.	
Aktorius	Skaitytojas, Administratorius	
Sužadavimo sąlyga	Vartotojas portalo pagrindiniame lange pasirenka meniu punktą autorių, cituojančių vienas kitą, sąrašo peržiūrai	
Susiję panaudojimo atvejai	Išplečia PA	–
	Apima PA	–
	Specializuoja PA	–
Pagrindinis įvykių srautas		
Sistemos reakcija ir sprendimai		
1. Vartotojas peržiūri vienas kitą cituojančių autorių sąrašą	Sistema pateikia visų ontologijoje saugomų autorių ir jų publikacijų sąrašą, kuriuose autoriai cituoja vieni kitus	
2. Vartotojas iš sąrašo pasirenka norimą publikacijų kategoriją ir	Sistema pateikia vienas kitą cituojančių autorių sąrašą	

paspaudžia [Filtruoti]	
Po sąlyga:	Pateiktas vienas kita cituojančių autorių ir jų publikacijų sąrašas, atitinkantis vartotojo nustatytą filtrą.
Alternatyvūs scenarijai	
2a. Vartotojas iš sąrašo filtre nurodo kategoriją(-as), pagal kurią(-ias) nėra nei vieno autoriaus	Išvedamas pranešimas, kad pagal pasirinktą filtrą, nerastas nei vienas autorius

Lentelė 7. PA „Prisijungti administratoriumi“ specifikacija

PA „Prisijungti administratoriumi“		
Tikslas. Autentifikuoti ir autorizuoti vartotoją, suteikiant jam portalo administratoriaus rolę		
Prieš sąlyga	Vartotojas yra atsidaręs portalo langą ir nėra prisijungęs	
Aktorius	Skaitytojas	
Sužadinimo sąlyga	Vartotojas į administratorius prisijungimo duomenų laukelį įrašo vartotojo duomenis ir paspaudžia mygtuką [Prisijungti]	
Susiję panaudojimo atvejai	Išplečia PA	–
	Apima PA	–
	Specializuoja PA	–
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. Vartotojas įveda teisingus prisijungimo duomenis		Sistema autorizuoja vartotoją, ir priskiria jam administratoriaus rolei skirtas teises
Po sąlyga:	Atlikta portalo vartotojo, kaip administratoriaus, autentifikacija ir autorizacija	
Alternatyvūs scenarijai		
2.a Vartotojas įveda neteisingus prisijungimo duomenis.	Sistema informuoja apie neteisingus prisijungimo duomenis.	

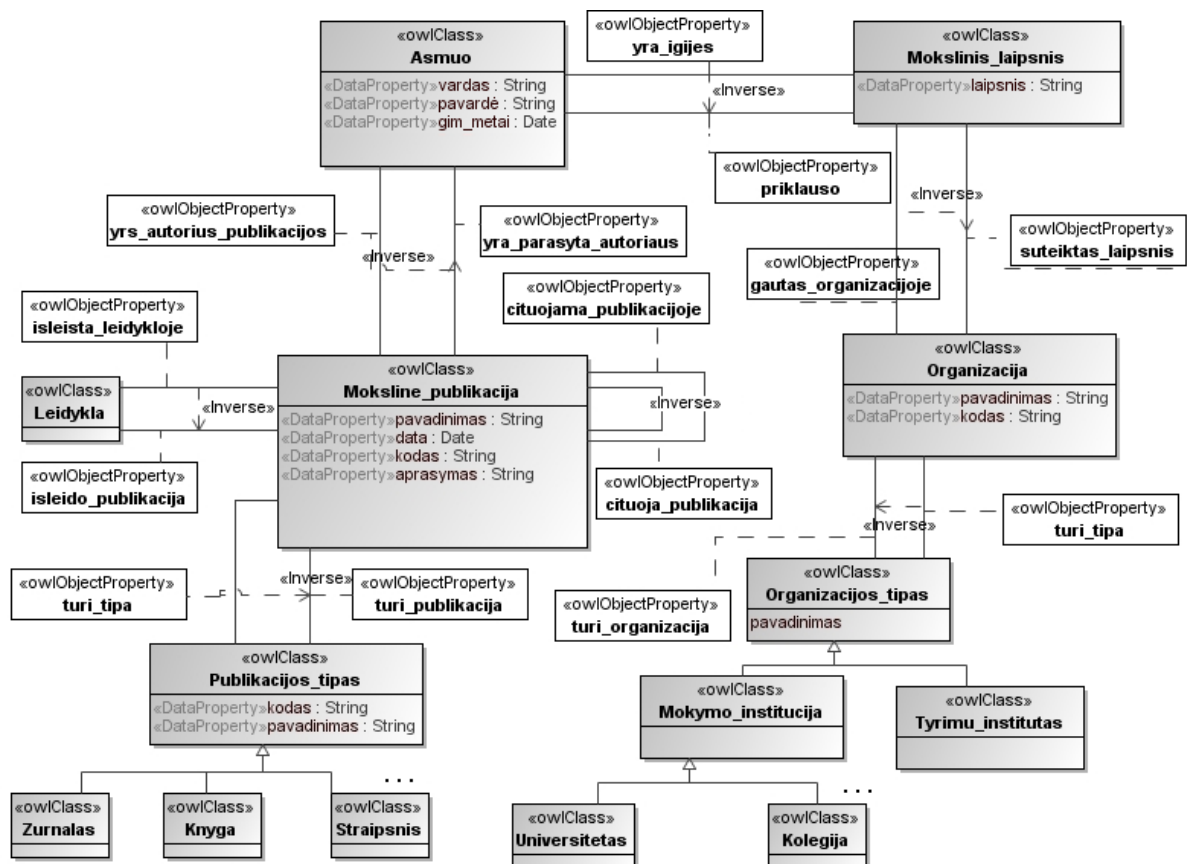
Lentelė 8. PA „Perskaičiuoti publikacijų citavimo indeksus“ specifikacija

PA „Perskaičiuoti publikacijų citavimo indeksus“		
Tikslas. Atlikti aktyvios ontologijos publikacijų citavimo indekso skaičiavimo algoritmą		
Prieš sąlyga	Vartotojas yra atsidaręs publikacijos peržiūros langą	
Aktorius	Administratorius	
Sužadinimo sąlyga	Vartotojas portalo ontologijos lange nurodo citavimo indekso skaičiavimo parametrus ir paspaudžia mygtuką [Suskaičiuoti]	
Susiję panaudojimo atvejai	Išplečia PA	–
	Apima PA	–
	Specializuoja PA	–
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. Vartotojas nurodo citavimo indekso skaičiavimo parametrus ir paspaudžia mygtuką [Suskaičiuoti]		Sistema atlieka ontologijos publikacijų citavimo indekso skaičiavimą, pagal vartotojo nurodytus parametrus.

Po sąlyga:	Suskaičiuoti publikacijų indeksai yra išsaugoti reliacinėje duomenų bazėje.
Alternatyvūs scenarijai	
1a. Vartotojas nenurodė, arba nurodė netinkamus citavimo indekso skaičiavimo parametrus	Sistema skaičiavimo nevykdo ir išveda pranešimą apie netinkamus parametrus

4.1.3 Dalykinės srities ontologijos modelis

Žemiau pateikta (4.2 pav.) mokslinių publikacijų ontologijos schema, kuri išreikšta UML klasių diagrama, įvedus specialius stereotipus, objektų ir duomenų tipų savybėms pavaizduoti.



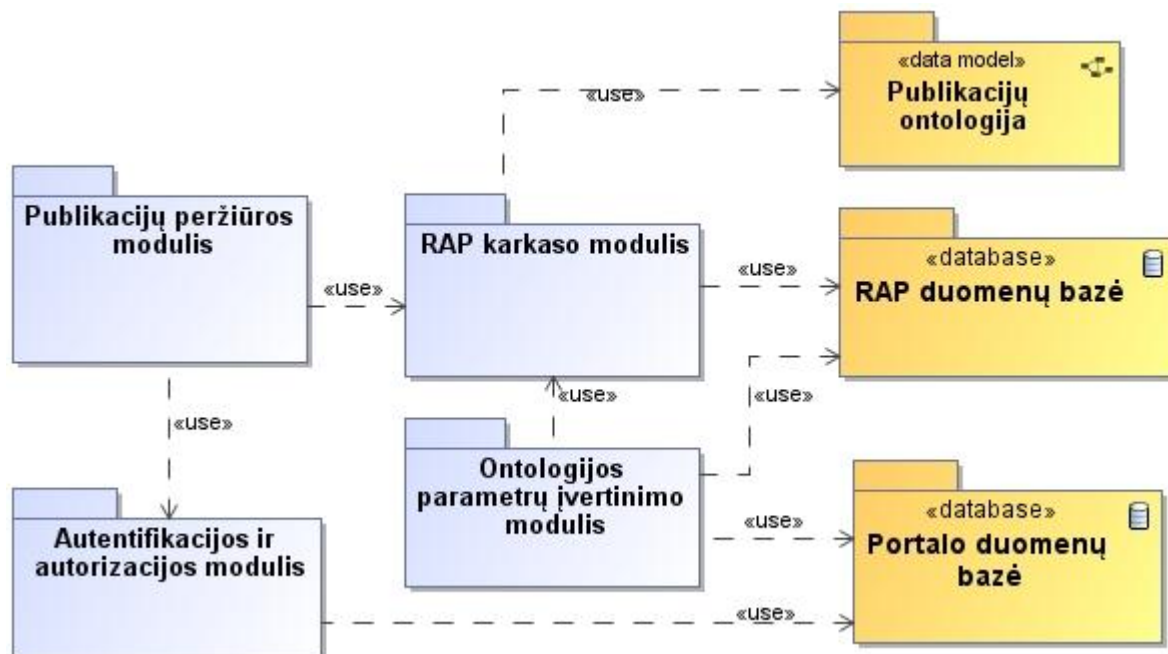
4.2 pav. Mokslinių publikacijų ontologijos vaizdavimas UML klasių diagrama

Ontologijoje kiekvienas žurnalas, knyga ar straipsnis priskiriami mokslinės publikacijos esybei „Mokslinė publikacija“. „Zurnalas“, „Knyga“ ir „Straipsnis“ yra publikacijos tipai. Publikacija taip pat turi savo leidyklą „Leidykla“ ir autorių „Asmuo“. Publikacija gali būti cituojama kitose publikacijose, arba ši gali cituoti kitas. Todėl ši citavimo ryšys jungia tą pačią „Mokslinė publikacija“ esybę. Autoriai gali turėti mokslinį laipsnį „Mokslinis laipsnis“, kuris buvo įgytas mokslo organizacijoje „Organizacija“. Organizacijų tipų gali

būti du – mokymo institucija „*Mokymo_institucija*“ arba tyrimų institutas „*Tyrimu_institutas*“. Mokymų institucija savo ruožtu skirstoma į universitetą „*Universitetas*“ ir kolegiją „*Kolegija*“.

4.2 Sistemos architektūra

Pateikiamas portalo loginės architektūros modelis. Portalo loginę architektūrą galima suskirstyti į 7 funkcinis modulius (4.3 pav).



4.3 pav. Loginė sistemos architektūra

Publikacijų peržiūros modulis yra skirtas portalo grafinei sąsajai realizuoti, naršymui ir navigacijai portale užtikrinti.

Autentifikacijos ir autorizacijos modulis skirtas sistemos administratoriaus prisijungimui prie portalo ir vartotojų teisių valdymui užtikrinti

Ontologijos parametrų įvertinimo modulis atlieka mokslinių publikacijų citavimo indekso apskaičiavimą. Apskaičiuoti publikacijų citavimo indeksai yra išsaugomi portalo duomenų bazėje. Publikacijų talpinimas, koregavimas bei kitos administravimo funkcijos atliekamos šio modulio pagalba.

Ontologijos nuskaitymo ir apdorojimo veiksmams atlikti skirtas **RAP karkaso modulis**. Kiti modeliai su ontologija gali sąveikauti tik per šį modulį.

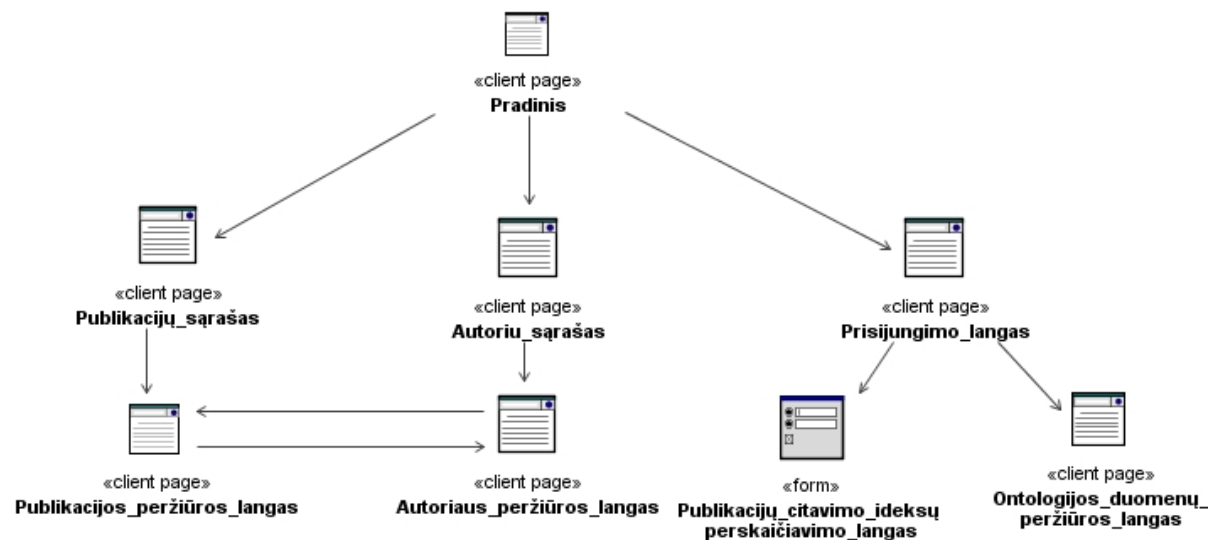
Publikacijų ontologija atitinka portalo mokslinių publikacijų *OWL* ontologiją. Ontologija yra pasiekama tik per *RAP* modulį, naudojanti karkaso apibrėžtoms

Ontologijos modeliui saugoti reliacinėje duomenų bazėje reikalinga **RAP karkaso duomenų bazė**. Ši duomenų bazė yra automatiškai sukuriama pasirinktoje duomenų bazių valdymo sistemoje, parengiant ir nustatant *RAP* karkasą darbui.

Portalo duomenų bazė yra reikalinga publikacijų citavimo indekso skaičiavimo algoritmui realizuoti. Šioje duomenų bazėje taip pat saugoma visa su vartotojais susijusi informacija, bei kiti dinaminiai sistemos duomenys.

4.3 Vartotojo navigacijos planas

Paveiksle 4.4 pavaizduotas navigacijos planas, kuris padeda suprasti, kaip realizuotas portalo navigacijos mechanizmas.



4.4 pav. Portalo navigacijos schema

Publikacijų ir autorių sąrašo langai turi filtrus, todėl juose pateikiama informacija priklauso nuo pasirinkto filtro reikšmės.

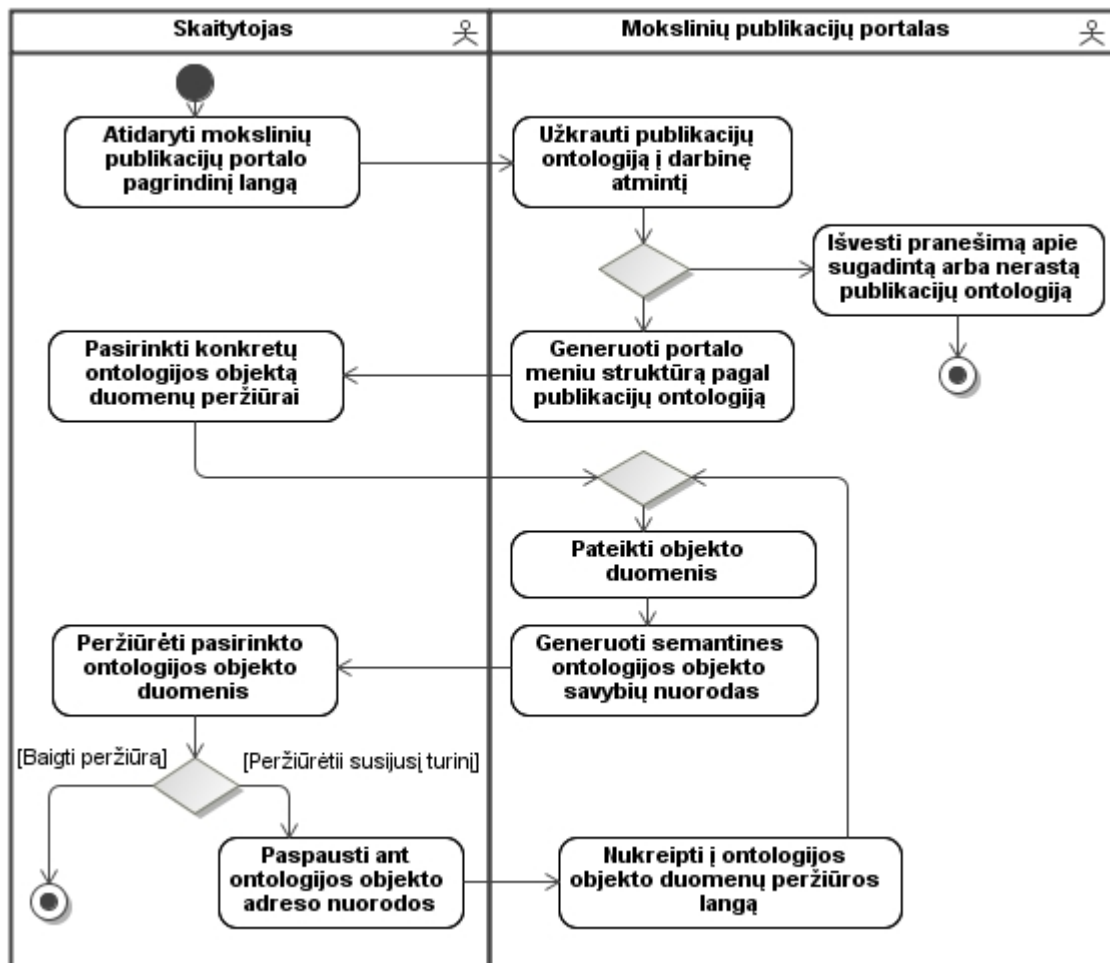
4.4 Sistemos elgsenos modelis

4.4.1 Skaitytojo naršymas portale

Portalo skaitytojo naršymas portale pavaizduotas veiklos diagrama paveiksle 4.5 paveiksle. Skaitytojui savo interneto naršyklėje atsidarius publikacijų portalą, portalo *web* serveris atlieka numatytosios ontologijos duomenų užkrovimą į operatyviają atmintį. Jeigu ontologija numatytu adresu neegzistuoja, arba jos nuskaityti nepavyko, išvedamas klaidos pranešimas. Tokiu atveju, portalo veikimo logiką bus apribota iki administratoriaus veiksmų ir pagrindinio portalo lango rodymo. Jei numatyta ontologija į operatyviają atmintį nuskaityta sėkmingai, *web* serveris atlieka portalo grafinės sąsajos generavimą ir pateikimą

skaitytojui. Skaitytojas naudodamasis interaktyvia portalo grafine sąsaja, gali naršyti mokslinių publikacijų portalą. Pasirinkus ontologijos objektą, atidaromas to objekto duomenų langas. Ontologijos objektą šiuo atveju reikia suprasti, kaip ontologijos klasės egzempliorių pvz.: konkreti publikacija, autorius ir pan.

Pasirinkus norimo ontologijos objekto peržiūrą, sistema generuoja to objekto semantines savybių nuorodas. Šių nuorodų generavimas remiasi peržiūrimos ontologijos klasės objekto savybėmis. Todėl, priklausomai nuo to, koks objektas peržiūrimas, generuojamos vis kitokios semantinės objektinių savybių nuorodos. Paspaudus tokios savybės, portalo skaitytojas yra nukreipiamas į atitinkamą objekto peržiūros langą.

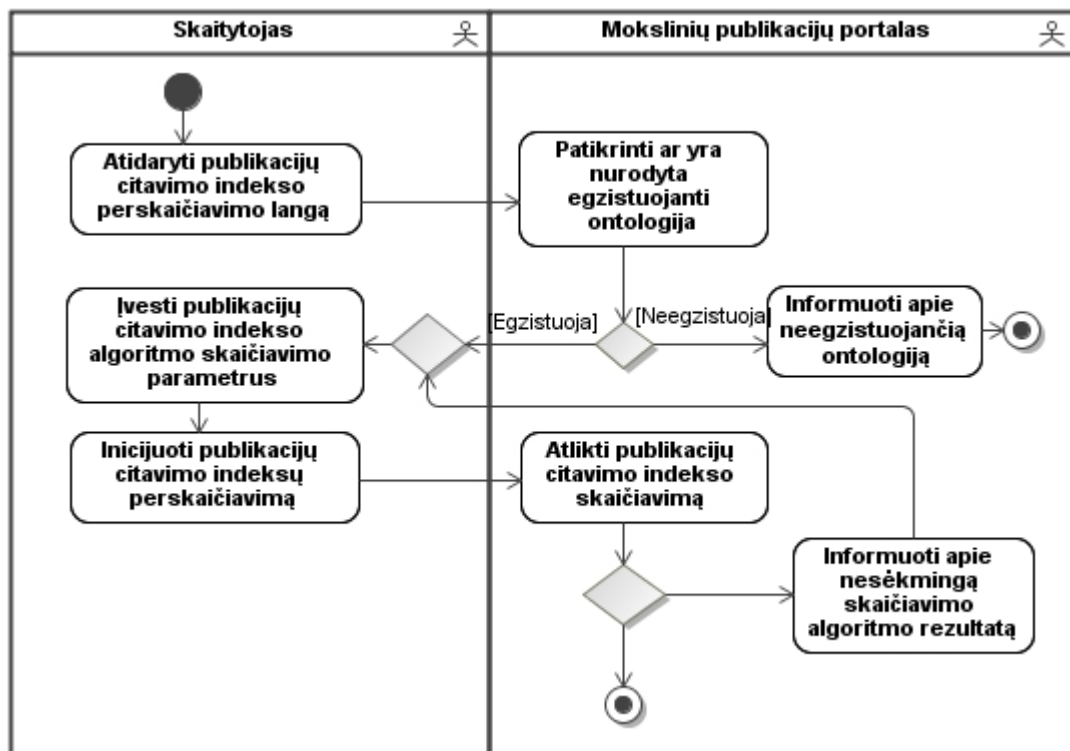


4.5 pav. Portalo skaitytojo naršymas portale

4.4.2 Publikacijų citavimo indekso atnaujinimas

Paveiksle 4.6 pateikta veiklos diagrama, iliustruojanti portalo administratoriaus inicijuojamą publikacijų citavimo indeksų perskaičiavimą. Citavimo indeksų perskaičiavimas yra inicijuojamas administratoriaus, ir galimas tik tada, kai ontologijos modelis yra užkrautas į *web* serverio darbinę atmintį.

Prieš inicijuojamas citavimo indeksų perskaičiavimą, administratorius formoje turi įvesti skaičiavimams reikalingus parametrus.

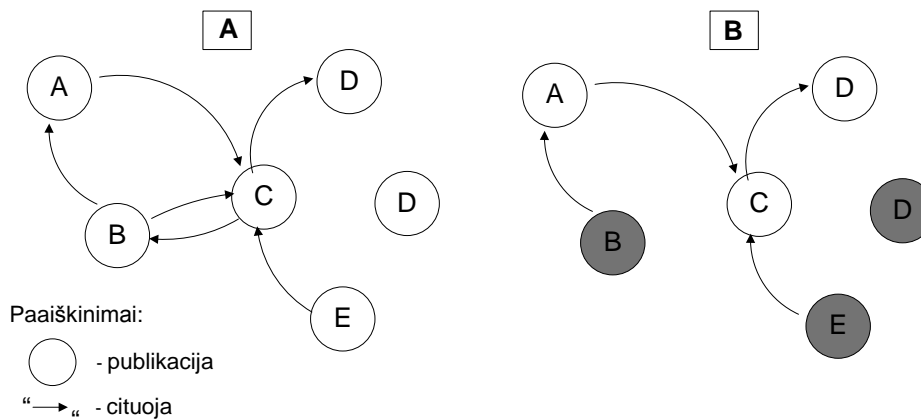


4.6 pav. Mokslinių publikacijų citavimo indekso atnaujinimo proceso veiklos diagrama

Teisingai nurodžius parametrus, pradedamas vykdyti mokslinių publikacijų citavimo indeksų apskaičiavimo algoritmas.

4.4.3 Citavimo indekso skaičiavimo algoritmo schema

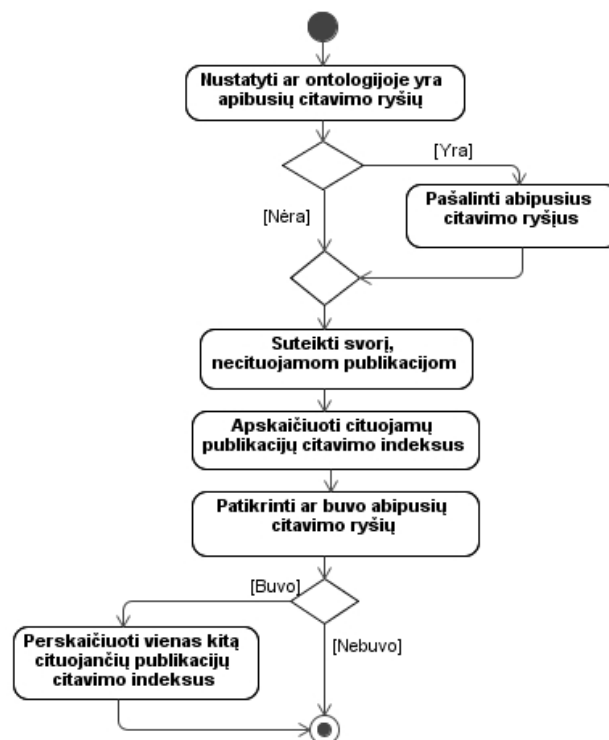
Algoritmo skaičiavimo idėja yra ta, kad visų pirma yra eliminuojami vienas kitą cituojančių publikacijų ryšiai. Jeigu šie ryšiai nebūtų pašalinti, ir jei publikacijų citavimo ryšius įvaizduotumėme grafo struktūra, grafe gautume uždarus ciklus (4.7 pav., A dalis), dėl kurių algoritmo skaičiavimai niekada nesibaigtų.



4.7 pav. Abipusių citavimo ryšių eliminavimas (A – pradinis grafas su abipusiais citavimo ryšiais; B- grafas po abipusių citavimo ryšių panaikinimo)

Panaikinus vienas kitą cituojančius ryšius (4.7 pav., B dalis) toliau pažingsniui atliekamas citavimo indekso skaičiavimas kiekvienai ontologijos publikacijai. Algoritmas visų pirma patikrina ar publikacija yra cituojama kituose publikacijose. Jei ji nėra niekur cituojama, jai yra suteikiamas pradinis citavimo indeksas. Pavyzdžiui, paveiksle 4.7 pav., B dalyje pateiktam atvejui, jau galima apskaičiuoti publikacijų B, D, E citavimo indeksus, kadangi jos neturi įeinančių citavimo ryšių.

Algoritmo, apskaičiuojančio visų ontologijos publikacijų citavimo indeksą, principinė schema pateikta 4.8 paveiksle.

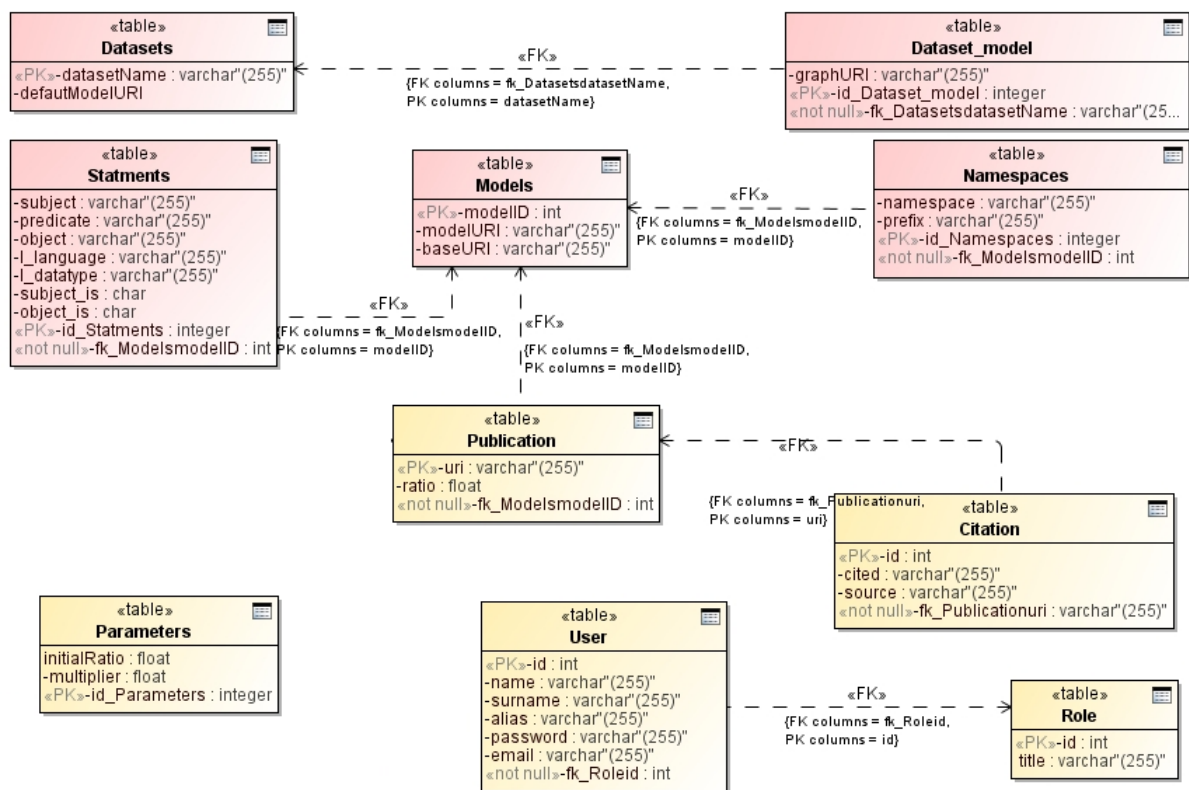


4.8 pav. Publikacijų citavimo indekso skaičiavimo algoritmo principinė schema

Algoritmo veiksmas realizuoti ir citavimo indeksui saugoti, naudojami reliacinės duomenų bazės lentelės.

4.5 Duomenų bazės schema

Portalui realizuoti panaudota 4.9 paveiklė pateikta duomenų bazės schema. Svarbu yra paminėti tai, kad RAP karkasas leidžia ontologijos modelį išsaugoti reliacinėje duomenų bazėje. Tam naudojamos 5 duomenų bazės lentelės (4.9 pav. raudona spalva), kurios turi būti sukurtos pasirinktoje duomenų bazių valdymo sistemoje. Papildomai buvo sukurtos dar 5 duomenų bazės lentelės (4.9 pav. geltona spalva), kurios reikalingos portalo vartotojų autentifikacijai ir citavimo indekso apskaičiavimui.

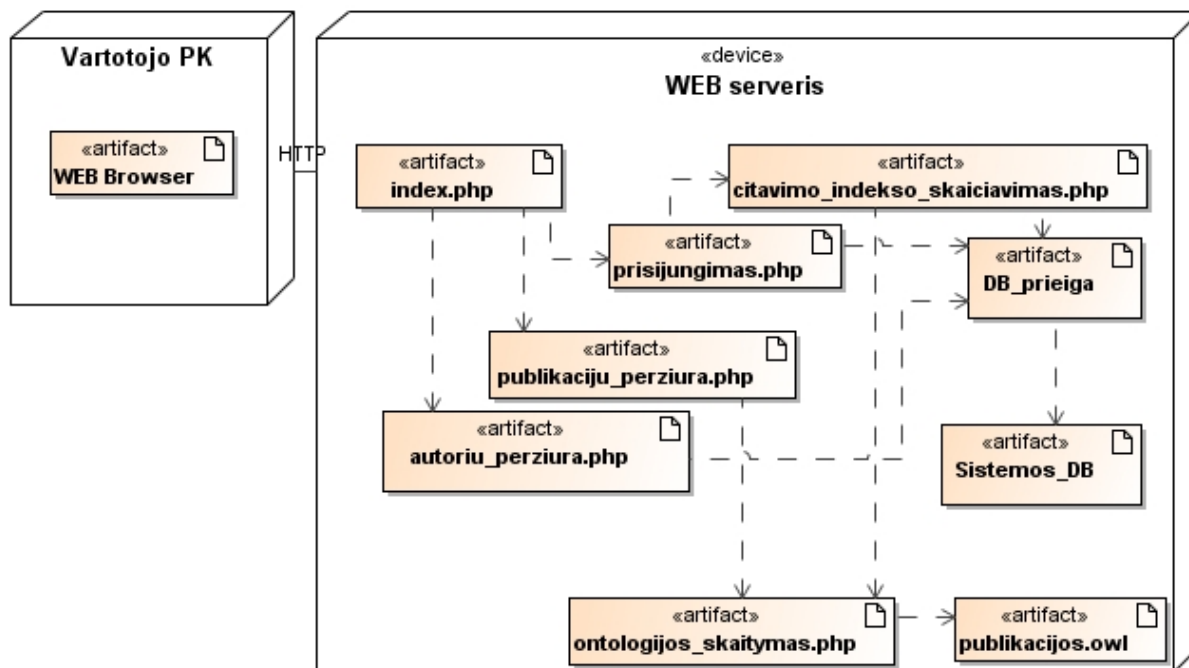


4.9 pav. Portalu duomenų bazės schema (RAP karkaso duomenų bazės lentelės pažymėtos rausvai, portalo - geltonai).

Visų duomenų bazių lentelių ir jų laukų aprašai pateikti prieduose, skyriuje „10.1 Portalu duomenų bazės lentelių aprašymas“

4.6 Diegimo diagrama

4.10 paveiksle pateikta diegimo diagrama, kuri vaizduoja komponentų išskirstymą techniniuose įrenginiuose. Sistemos duomenų bazė ir ontologija yra talpinama tame pačiame *web* serveryje.



4.10 pav. Sistemos diegimo modelis

RAP karkasas gali naudoti ir nutolusią, kitame serveryje esančią ontologiją. Tokiu atveju vietoje lokalaus ontologijos adreso reikia nurodyti globalų ontologijos adresą, pavyzdžiui *http://adresas.com/publikacijos.owl*. Tokiu atveju, nurodyta ontologija bus užkrauta į operatyviąją atmintį ir portalo veikimas nieko nesiskirs nuo veikimo, kai ontologija saugoma vidiniame serveryje.

5. Mokslinių publikacijų portalo realizacija

Mokslinių publikacijų portalo prototipas realizuotas naudojant *PHP/CSS* programavimo technologijas, *RAP* ontologijų apdorojimo karkasą, *MySQL* duomenų bazę ir *Protege* ontologijų redagavimo įrankį.

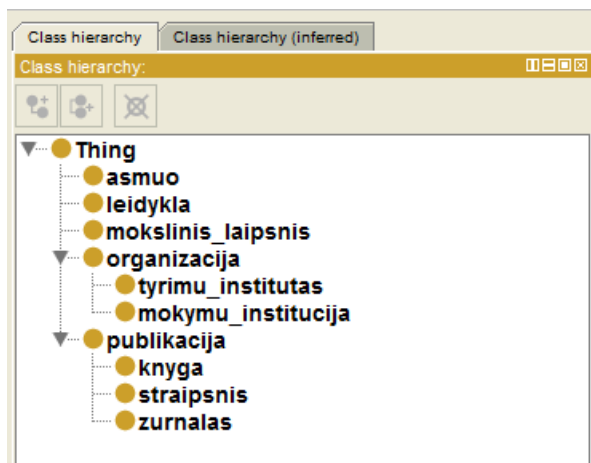
Sistema buvo išbandoma lokaliame kompiuteryje įdiegus *Wamp Server* virtualų *Apache* serverį.

5.1 Realizacijos ir veikimo aprašymas

5.1.1 Portalo ontologijos struktūra

Pagal dalykinės srities ontologijos modelį buvo sudaryta mokslinių publikacijų ontologija, sukuriant klases, duomenų ir objekto tipo savybes. Po to ontologija užpildyta klasių egzemplioriais. Visa tai buvo atlikta naudojant *Protege* ontologijų redaktorių (5.1 pav.).

Ontologijos kūrimo metu yra naudinga įjungti ontologijų analizės ir išvedimo mechanizmą (angl. *reasoner*), kuris realiu laiku atlieka ontologijos analizę pagal aksiomas ir ontologijos taisykles bei išveda išvedimo klases ir individus. Populiariausi ontologijų analizės ir išvedimo mechanizmai yra *FaCT++*, *HermiT*, *Pellet*, *RacerPro*. Ontologijų redaktoriuje *Protege* galima naudoti visus šiuos išvedimo mechanizmus.



5.1 pav. Sukurtos ontologijos klasių hierarchija *Protege* įrankyje

5.1.2 Portalo struktūra

Paveiksle 5.2 pateiktas bendras portalo vaizdas. Dešinėje esantis meniu blokas yra automatiškai generuojami pagal ontologijos klases.

Mokslinių publikacijų skelbimo portalas

Prisijungimo vardas:

 Slaptažodis:

Sveiki

Mokslinių publikacijų skelbimo portalas skirtas kaupti informaciją apie išleistas mokslines publikacijas (knygas, straipsnius, žurnalus) ir sistemingai pateikti šią informaciją portalo lankytojams



Publikacija

Knyga
 Diskrečioji matematika
 Duomenų bazės ir semantiniai modeliai
 Ekonomikos teorijos pagrindai
 Taikomoji matematika
 Skaitmeninės komunikacijos pagrindai

Straipsnis
 Business Rules as Part of Information Systems Life Cycle
 Informatikos istorija
 Ontologijų taikymas portaluose

Žurnalas
 Molekulinė teorija

Asmuo

Marius Balčiūnaitis
 Algirdas Jakutis
 Tadas Lenkevičius
 Mindaugas Gaurilčikas
 Lina Nemuraite
 Bronius Paradauskas

Leidykla

Technologija
 Vaga

Mokslinis laipsnis

Daktaras
 Magistrantas
 Profesorius

Organizacija

Vytauto Didžiojo universitetas
 Kauno technologijos universitetas

Išleidimo šalis

Danija
 Lietuva

Mindaugas Gaurilčikas 2012 (C) KTU

5.2 pav. Pagrindinio portalo lango vaizdas

Portalo meniu struktūra suprogramuota taip, kad būtų dinamiškai generuojama iš ontologijos klasių. Vienas meniu blokas atitinką vieną ontologijos klasę. Meniu punktas atitinką vieną ontologijos klasės individą. Jei klasė turi poklasių (pvz. klasė „Publikacijos“), meniu elementai yra sugrupuojami pagal šiuos poklasius (pvz. „Knyga“, „Straipsnis“ ir pan.)

Žemiau pateiktas pavyzdys su paaiškinimais, kaip atrodo ontologijos klasės „Publikacija“ meniu bloko elementas.



5.3 pav. Meniu struktūra ir jos paaiškinimas

Ontologijos klasių atrinkimui, naudojama SPARQL užklausa:

```
SELECT * WHERE {
  ?x rdf:type owl:Class .
  ?x rdfs:label ?label .
  OPTIONAL {
    ?x rdfs:subClassOf ?sc .
  }
  FILTER ( !bound(?sc) && ?x != owl:Thing ) }
```

Iš publikacijų meniu pasirinkus konkrečią publikaciją, pateikiami jos duomenys (5.4 pav.)

Pagrindinis Publikacijos Autoriai Ontologija	
Publikacija: Business Rules as Part of Information Systems Life Cycle	
Išleidimo data	2011-06-19
Publikacijos pavadinimas	Business Rules as Part of Information Systems Life Cycle
URI	http://www.promind.lt/publikacijos#business_rules
cituoja publikaciją (1)	
Duomenų bazės ir semantiniai modeliai	
išleista leidykloje (1)	
Technologija	
parasyta autoriaus (2)	
Marius Balčiūnaitis	
Lina Nemuraite	
Publikacija	
Knyga	
Diskrečioji matematika	
Duomenų bazės ir semantiniai modeliai	
Ekonomikos teorijos pagrindai	
Taikomoji matematika	
Skaitmeninės komunikacijos pagrindai	
Straipsnis	
Business Rules as Part of Information Systems Life Cycle	
Informatikos istorija	
Ontologijų taikymas portaluose	
Žurnalas	
Molekulinė teorija	
Asmuo	
Marius Balčiūnaitis	
Algirdas Jakutis	
Tadas Lenkevičius	
Mindaugas Gaurilčikas	
Lina Nemuraite	
Bronius Paradauskas	
Leidykla	

5.4 pav. Klasės „Publikacija“ individo peržiūra

Kiekvieno ontologijos egzemplioriaus duomenų peržiūros langas yra sudaromas pagal tokius principus – lango viršuje pateikiamos egzemplioriaus duomenų tipų savybės (*datatype properties*), o žemiau – objektų savybės (*object properties*).

Individo duomenų tipų savybių atrinkimo *SPARQL* užklausa ($\$URI$ – *php* kintamasis saugantis individo *URI* adresą):

```
SELECT * WHERE {
  " .\$URI ." ?prop ?obj .
  ?prop rdf:type owl:DatatypeProperty .
  ?prop rdfs:label ?label }
```

Individo objekto savybių atrinkimo *SPARQL* užklausa ($\$URI$ – *php* kintamasis saugantis individo *URI* adresą):

```
SELECT DISTINCT ?label ?prop WHERE {
  <".\$URI ."> ?prop ?obj .
  ?prop rdf:type owl:ObjectProperty .
  ?prop rdfs:label ?label }
```

Paveikslėlyje žemiau (5.5 pav.) raudona spalva pažymėta dalis, užpildoma iš egzemplioriaus duomenų tipų savybių.

Duomenų bazės ir semantiniai modeliai

Aprašymas	Semantinis modeliavimas – tai duomenų loginis struktūrizavimas pagal jų prasmę. Dažnai duomenys tvarkomi neatsižvelgiant į prasmę ar interpretaciją, pavyzdžiui, pagal abėcėlę; tuo tarpu semantiniai modeliai asocijuojasi su realiais objektais ir jų savybėmis, tokiomis, kokias jas suvokia žmogus, todėl yra lengvai suprantami. Semantinis modeliavimas yra pagrįstas loginėmis sąvokų kategorijomis. Sąvokos mąstyme fiksuojamos, įgauna apibrėžtumo terminų dėka, t.y., operuojame tik tomis sąvokomis, kurios gali būti išreikštos kalba.
Publikacijos kodas	ISBN 9955-09-436-2
Pavadinimas	Duomenų bazių semantiniai modeliai
URI	http://www.promind.lt/publikacijos#duomenu_bazes

Duomenų tipo savybės

5.5 pav. Egzemplioriaus duomenų tipų savybių peržiūra

Egzemplioriaus objektų savybės pateikiamas 5.6 pav., po duomenų tipų savybių.

išleista leidykloje (1)
[Technologija](#)

yra parasyta autoriaus (2)
[Lina Nemuraite](#)
[Bronius Paradauskas](#)

Objekto tipo savybės

5.6 pav. Egzemplioriaus objekto tipų savybių peržiūra

Objekto savybės pateikiamos nurodant savybės pavadinimą (pvz.: „*turi laipsnį*“) ir objekto pavadinimą (pvz.: „*Profesorius*“). Objekto pavadinimas yra aktyvi nuoroda, todėl paspaudus ant jos, pereinama į to objekto duomenų peržiūros langą.

Pagrindinis Publikacijos Autoriai Ontologija

Bronius Paradauskas

Pavardė	Paradauskas
Vardas	Bronius
URI	http://www.promind.lt/publikacijos#paradauskas

yra publikacijos autorius/bendraautorius (1)
[Duomenų bazės ir semantiniai modeliai](#)

turi laipsnį (1)
[Profesorius](#)

5.7 pav. Ontologijos klasės „Asmuo“ individo peržiūra

Paveiksle (5.8 pav.) pateikta klasės „Leidykla“ individo „Technologija“ peržiūros langas.

Pagrindinis Publikacijos Autoriai Ontologija

Technologija

Pavadinimas	Technologija
URI	http://www.promind.lt/publikacijos#technologija

išleido publikaciją (2)
[Business Rules as Part of Information Systems Life Cycle](#)
[Duomenų bazės ir semantiniai modeliai](#)

5.8 pav. Ontologijos klasės „Leidykla“ individo peržiūra

Analogiškų principu peržiūrimi visi kiti ontologijos klasių individai. Paveiksle 5.9 pateiktas ontologijos informacijos peržiūros ir ontologijos publikacijų citavimo indekso perskaičiavimo langas. Jis yra matomas tik portalo administratoriui. Parametrai „svoris“ ir „koeficientas“ perduodami citavimo indekso skaičiavimo algoritmui.

5.9 pav. Publikacijų citavimo indekso perskaičiavimo langas

Ontologijos apdorojimo karkasas *RAP* turi specialų metodą *writeAsHtmlTable()*, kurį iškvietus, darbinėje atmintyje saugomas ontologijos modelis yra atspausdinamas *HTML* formato lentelės pavidalu. Žemiau pateikta (5.10 pav.) atspausdinta mokslinių publikacijų ontologija, kaip *RDF* trejetų rinkinys.

Analogiška struktūra gali būti išsaugoma ir reliacinėje duomenų bazėje.

Pagrindinis Publikacijos Autoriai Ontologija			
Ontologijo URI adresas: http://www.promind.lt/publikacijos#			Size: 287
Prefix:	Namespace:		
rdfs	http://www.w3.org/2000/01/rdf-schema#		
owl2xml	http://www.w3.org/2006/12/owl2-xml#		
owl	http://www.w3.org/2002/07/owl#		
xsd	http://www.w3.org/2001/XMLSchema#		
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#		
No.	Subject	Predicate	Object
1.	Resource: http://www.promind.lt/publikacijos	RDF Node: rdf:type	RDF Node: owl:Ontology
2.	Resource: xsd:date	RDF Node: rdf:type	RDF Node: rdfs:Datatype
3.	Resource: http://www.promind.lt/publikacijos#cituoja_publicacija	RDF Node: rdf:type	RDF Node: owl:ObjectProperty
4.	Resource: http://www.promind.lt/publikacijos#cituoja_publicacija	RDF Node: rdfs:label	Literal: cituoja publikaciją (rdf:datatype="http://www.w3.org/2001/XMLSchema#string")
5.	Resource: http://www.promind.lt/publikacijos#cituoja_publicacija	RDF Node: rdfs:domain	Resource: http://www.promind.lt/publikacijos#citata
6.	Resource: http://www.promind.lt/publikacijos#cituoja_publicacija	RDF Node: rdfs:range	Resource: http://www.promind.lt/publikacijos#publikacija
7.	Resource: http://www.promind.lt/publikacijos#cituoja_publicacija	RDF Node: owl:ObjectProperty	RDF Node: owl:ObjectProperty

5.10 pav. Ontologijos modelio spausdinimas lentelė

6. Eksperimentinis metodo įvertinimas

6.1 Eksperimento planas

Eksperimentiniui tyrimui atlikti buvo sukurtas nesudėtingas mokslinių publikacijų portalas, sukuriant ir panaudojant mokslinių publikacijų ontologiją. Eksperimento eiga bus vykdoma pagal siūlomo semantinio portalo sudarymo metodo žingsnius.

6.2 Eksperimentas

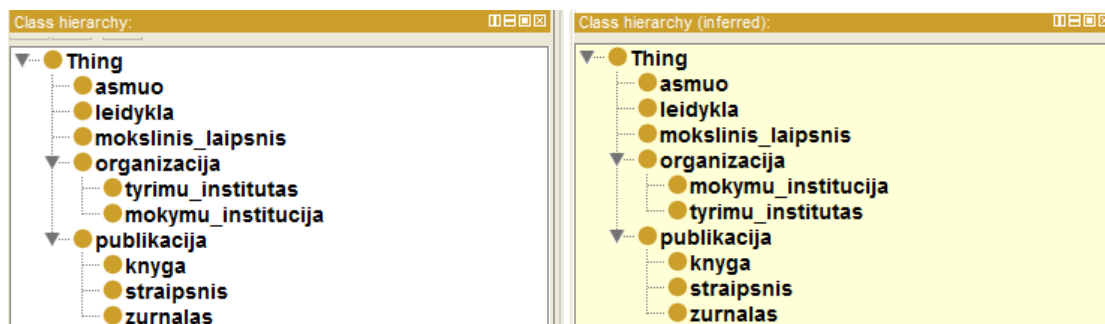
6.2.1 Kompetencijos klausimų sąrašo sudarymas

Pagal darbe aprašytą metodiką, ontologijos reikalavimai yra kompetencijų klausimai. Buvo sudarytas sąrašas kompetencijos klausimų, į kuriuos atsakymus turime rasti iš busimos ontologijos:

- Kokios yra X tipo publikacijos?
- Kokios publikacijos yra labiausiai cituojamos?
- Kas yra X tipo publikacijų autoriai?
- Kokie publikacijų autoriai cituoja vieni kitus?
- Kokie publikacijų autoriai turi X mokslinį laipsnį?

6.2.2 Bazinės ontologijos sukūrimas

Dalykinė ontologija turi atsakyti į sudarytus kompetencijos klausimus. Išnagrinėjus kompetencijos klausimus, pradedama konstruoti bazinę ontologiją. Ontologija kuriama *Protege* įrankiu. Kad ontologija užtikrintu ontologijoms keliamus reikalavimus, ontologijos vystymo metu buvo naudojami taisyklėmis grindžiami ontologijų tikrinimo mechanizmai *Hermit 1.3.5* ir *FaCT++*. Sudarius ontologiją ir palyginus primityvių ir išvedimo klasių hierarchiją (6.1 pav.), pastebėta, kad jos sutampa t.y. ontologija neturi nei vienos išvedimo klasės.



6.1 pav. Ontologijos klasių hierarchija (kairėje sudaryta klasių struktūra, dešinėje – struktūra papildyta išvedimo klasėmis)

Sukuriamos objekto savybės kiekvienai objekto savybei nurodoma – kokia klasė ją gali įgyti (*domains*), ir kokia yra savybės reikšmė (*ranges*). Pavyzdyje 6.2 pav. pateiktas objekto savybės „*parase_publicacija*“ apibrėžimas – savybė gali būti įgyjama klasės „*asmuo*“ ir gali įgyti klasės „*publikacija*“ reikšmes. Naudojamas ontologijos analizės ir išvedimo mechanizmas išveda ekvivalenčias objekto savybes – objekto savybė *parase_publicacija* yra ekvivalenti objekto savybei „*parašyta_atoriaus*“ su inversijos ženklu.



6.2 pav. Objekto tipo savybės kūrimas

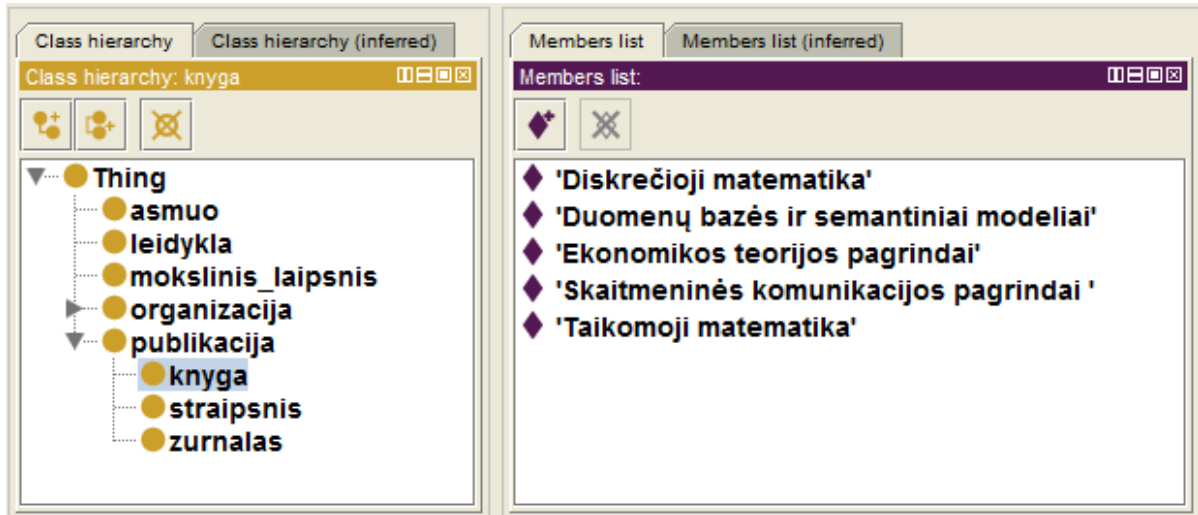
Panašiu principu kuriamos ir duomenų tipų savybės. Pavyzdyje 6.3 pav. pateiktas duomenų tipo savybės „*Publikacijos pavadinimas*“ apibrėžimas – savybė gali būti įgyta klasės „*publikacija*“ ir turėti „*string*“ tipo reikšmę.



6.3 pav. Duomenų tipo savybių kūrimas

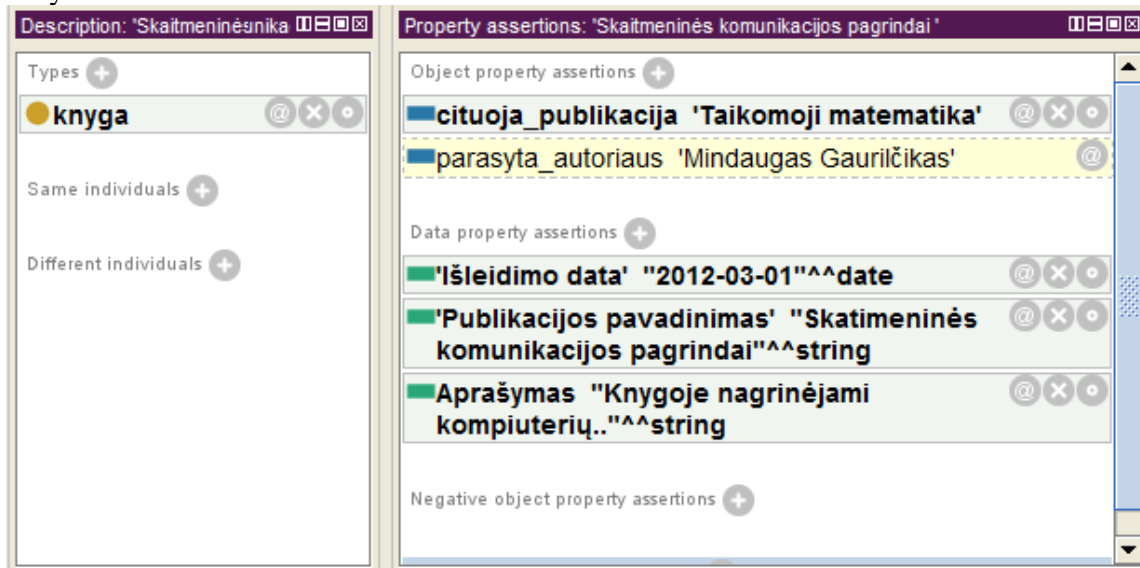
6.2.3 Klasės egzempliorių sukūrimas

Kiekvienai iš ontologijų klasių sukuriama klasių egzemplioriai (6.4 pav.). Egzemplioriai yra kuriami nurodant klasės ir egzemplioriaus *URI* adresą.



6.4 pav. Klasės „publikacija“ poklasei „knyga“ priskirti klasių egzemplioriai

Sukurtiems klasės egzempliorius reikia priskirti tiek objekto ir duomenų tipo savybes.



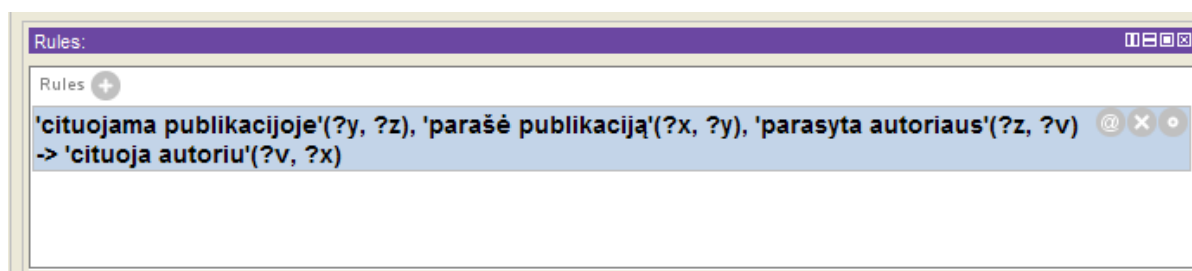
6.5 pav. Klasės „publikacija“ poklasei „knyga“ priskirti klasių egzemplioriai

Pavyzdyje pateiktas pavyzdys, kai išvedimo mechanizmas „Hermit 1.3.5“ išveda savybę „parasyta autoriaus Mindaugas Gaurilčikas“. Šis išvedimas nustatomas pagal ontologijoje apibrėžtas aksiomas.

6.2.4 Kompetencijos klausimų formalizavimas

Šiame žingsnyje svarbiausia yra nustatyti kokias papildomas taisykles reikia aprašyti, norint, kad ontologija atsakytų į klausimus, kurių atsakymui gauti nepakanka *OWL* aksiomų. Mokslinių publikacijų eksperimentiniam portalo sukūrimui pasirinktas *OWL* ontologijų skaitymo karkasas *RAP* nepalaiko *SWRL* taisyklių. Todėl taisyklių rašymas būtų beprasmis. Tačiau jeigu naudojamas pažangesnis įrankis, pavyzdžiui *Jena*, galima aprašyti taisykles, pagal kurias būtų išvedami faktai.

Protege redaktoriuje taisykles rašomos skydelio „Rules“ dalyje (6.6 pav.). Galima aprašyti neribota skaičių taisyklių. Taisyklių surašymo eilės tvarka yra nesvarbi.

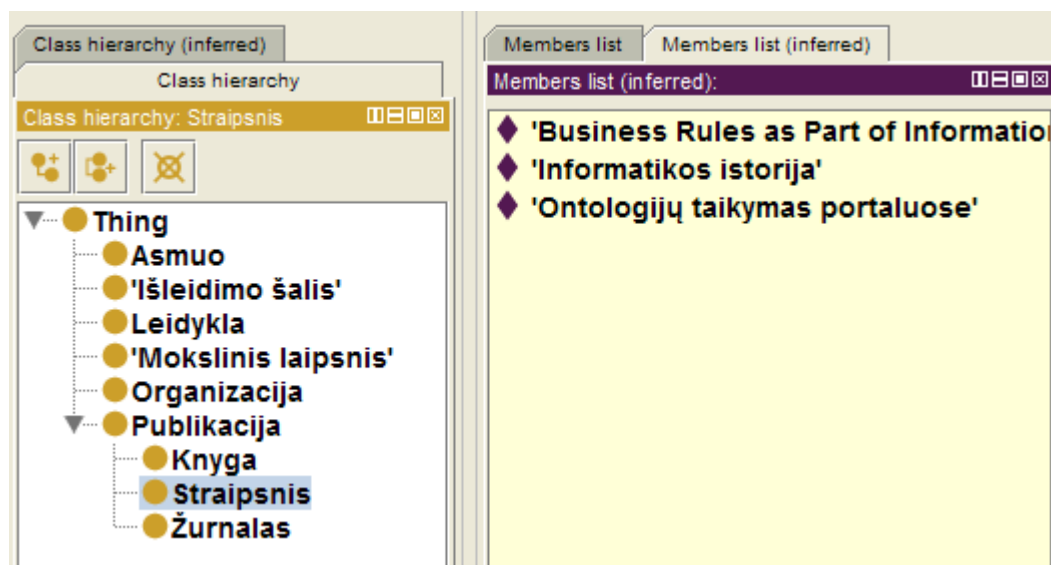


6.6 pav. *SWRL* taisyklės pavyzdys *Protege* įrankyje

6.2.5 Ontologijos tikrinimas

Šiame žingsnyje patikrinama, ar galima gauti atsakymus į kompetencijos klausimus. Pirmasis klausimas buvo „Kokios yra *X* tipo publikacijos?“. Atsakymą į šį klausimą galime nesunkiai rasti pasirinkę norimo *X* tipo publikacijų klasę ir peržiūrėję šiai klasei priklausančius individus.

Paveikslėlyje 6.7 pateiktas pavyzdys kaip *Protege* įrankyje gauname atsakymą į kompetencijos klausimą „kokios yra straipsnio tipo publikacijos?“.



6.7 pav. Klasės „Publikacija“ poklasei „Straipsnis“ priskirti klasių egzemplioriai

Į kompetencijos klausimą „Kas yra X tipo publikacijų autoriai?“ atsakymą galime gauti tik iš dalies t.y. galime atsakyti į paprastesnį klausimą „kas yra publikacijų autoriai?“. Atsakymą gautume peržiūrėję klasės „Asmuo“ individų sąrašą, kadangi asmenys šiuo atveju atitinką autorius.

Į kitus kompetencijos klausimus (*Kokios publikacijos yra labiausiai cituojamos? kokie publikacijų autoriai cituoja vieni kitus? Kokie publikacijų autoriai turi X mokslinį laipsnį?*) šiame žingsnyje taip pat galime atsakyti tik iš dalies arba negalime atsakyti visai. Nežinodami ar ontologija atsako į visus klausimus negalime užtikrinti ontologijos semantinio pilnumo. Todėl, reikia pereiti į kitą žingsnį – semantinio portalo tikrinimą.

6.2.6 Semantinis portalo tikrinimas

Semantinis portalas nuo pat pradžių buvo realizuojamas taip, kad atsakytų į keliamus kompetencijos klausimus. Tai reiškia, kad uždavus naują kompetencijos klausimą, atsakymo į jį portale galime negauti, kadangi klausimas nebuvo numatytas ontologijos ir portalo kūrimo metu.

Atsakymai į kompetencijos klausimus semantiniame portale realizuoti kaip atskiri portalo langai. Paveiksle 6.8 pateiktas autorių sąrašas.

Vardas Pavardė	Mokslinis laipsnis
Algirdas Jakutis	
Bronius Paradauskas	Profesorius
Lina Nemuraitė	Daktaras
Marius Balčiūnaitis	
Mindaugas Gaurilčikas	Daktaras
Tadas Lenkevičius	Profesorius

6.8 pav. Visų publikacijos autorių sąrašas

Paveiksle 6.9 pateiktas autorių savo publikacijose cituojantys vieni kitus. Šis sąrašas teikia atsakymą į kompetencijos klausimą „Kokie publikacijų autoriai cituoja vieni kitus?“

Pagrindinis Publikacijos Autoriai Ontologija		
Visi autoriai	Autoriai cituojantys vieni kitus	Autoriai pagal mokslinį laipsnį
Autoriai cituojantys vieni kitus		
Vardas Pavardė		Vardas Pavardė
Algirdas Jakutis		Tadas Lenkevičius
Marius Balčiūnaitis		Mindaugas Gaurilčikas

6.9 pav. Autorių, cituojančių vienas kitus, sąrašas

Paveiksle 6.10 pateiktas autorių sąrašas, kurie yra įgiję tam tikrą mokslinį laipsnį. vŠis sąrašas teikia atsakymą į kompetencijos klausimą „Kokie publikacijų autoriai turi X mokslinį laipsnį“. X – tai mokslinis laipsnis pasirenkamas iš sąrašo.

Pagrindinis Publikacijos Autoriai		
Visi autoriai	Autoriai cituojantys vieni kitus	Autoriai pagal mokslinį laipsnį
Profesorius	<input type="button" value="Filtruoti"/>	
Autoriai pagal laipsnį		
Vardas Pavardė		Mokslinis laipsnis
Bronius Paradauskas		Profesorius
Tadas Lenkevičius		Profesorius

6.10 pav. Autorių, pagal turimą mokslinį laipsnį, sąrašas

Paveiksle 6.11 pateiktas autorių mokslinių publikacijų sąrašas, išrikiuotas citavimo indekso mažėjimo linkme. Šis sąrašas teikia atsakymą į kompetencijos klausimą „Kokios publikacijos yra labiausiai cituojamos?“. Publikacijos citavimo indeksas yra apskaičiuojamas portale, todėl gauti atsakymą į šį klausimą ontologijos lygyje, panaudojant aksiomas ir taisykles, buvo neįmanoma.

Pagrindinis Publikacijos Autoriai	
Publikacijos pagal citavimo indeksą	
Visos ▼	Filtruoti
Publikacijos pavadinimas	Citavimo ind.
Duomenų bazės ir semantiniai modeliai	1.72
Informatikos istorija	1.39
Taikomoji matematika	1.30
Business Rules as Part of Information Systems Life Cycle	1.00
Diskrečioji matematika	1.00
Ekonomikos teorijos pagrindai	1.00
Molekulinė teorija	1.00
Ontologijų taikymas portaluose	1.00
Skaitmeninės komunikacijos pagrindai	1.00

6.11 pav. Labiausiai cituojamų publikacijų sąrašas

6.3 Eksperimento rezultatai

Eksperimento metu pagal sukurta metodiką pavyko realizuoti semantinį mokslinių publikacijų portalą, kuris atsako į iškeltus kompetencijos klausimus. Sudaryta metodika reikalauja nuoseklaus portalo kūrimo proceso, norit kad sukurtas semantinis portalas atitiktų visus keliamus reikalavimus.

Semantinio portalo ontologija gali būti modifikuojama, tačiau tuomet tenka iš naujo modifikuoti patį portalą atsižvelgiant į sprendžiamą uždavinį ir į būtinas sąlygas semantiniam pilnumui užtikrinti.

7. Išvados

1. Atlikus skirtingų metodų, aprašančių ontologijų ir jomis grindžiamų portalų kūrimą, nustatyta, kad nėra metodikos, pažingsniui aprašančios, kaip reikia sukurti semantinį portalą, nuo ontologijos sukūrimo iki pat portalo realizavimo, panaudojant šiuolaikines portalų kūrimo technologijas;
2. Kelių literatūros šaltiniuose aprašytų ontologijų sudarymo metodų jungimas ir pritaikymas metodikai semantiniams portalams kurti užtikrina, jog sukurta metodika apima visus reikalingus žingsnius ontologijos kokybei užtikrinti ir tuo pačiu yra tinkama semantiniams portalams kurti; sukurtasis metodas papildo informacinių sistemų kūrimo metodus taisyklėmis ir kriterijais, reikalingais kuriant sistemas taikant ontologijas;
3. Pagal sudarytą metodiką praktiškai realizuotas semantinis mokslinių publikacijų skelbimo portalas patvirtino, kad metodika tinkama portalui realizuoti, o DBVS priemonių taikymas leidžia realizuoti ontologijos charakteristikų skaičiavimo algoritmą, kurio nepavyktų realizuoti ontologijų apdorojimo priemonėmis.
4. Atliktas eksperimentinis sukurto metodo įvertinimas parodė, kad sukurta metodika leidžia išpildyti ontologijoms keliamus reikalavimus ir gali būti naudojamas semantiniams portalams kurti.

8. Literatūra

1. BERNERS-LEE, T.; HENDLER, J; LASSILA, O.; *The Semantic Web: Scientific American*. May 2001, 284(5):34-43
2. *Dublin Core Metadata Initiative (DCMI)*. [interaktyvus]. [žiūrėta 2012-05-21] prieiga per internetą: <<http://dublincore.org/>>.
3. *Europeana – Homepage* [interaktyvus]. [žiūrėta 2012-05-21] prieiga per internetą: <<http://www.europeana.eu/portal/>>.
4. FENSEL D.: *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*. Springer-Verlag, 2001, Berlin.
5. 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.
6. GRUBER, T. R. *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*. Technical report, KSL-93-04. – Knowledge Systems Laboratory, Stanford University, 1993. – P. 1–23.
7. HARTMANN, J.; SURE, Y. *Infrastructure for Scalable, Reliable Semantic Portals* [interaktyvus]. Institute AIFB, University of Karlsruhe, June 2004. [žiūrėta 2012-02-10].
8. HYVONEN, E. et al., *Museums and the Web 200, Finnish Museums on the Semantic Web, Figure 9* [interaktyvus]. [žiūrėta 2012-25-21] prieiga per internetą: <<http://www.archimuse.com/mw2004/papers/hyvonen/hyvonenFig9.html>>.
9. *Jena Semantic Web Framework* [interaktyvus]. [žiūrėta 2012-05-21] prieiga per internetą: <<http://jena.sourceforge.net/index.html>>
10. KARPOVIČ, J.; *Semantinei paieškai naudojamų ontologijų generavimo pagal duomenų bazes schemą procesas*: magistro darbas. KTU Informatikos fakultetas. [Kaunas], 2010
11. LOPEZ M. F., *et al.* Building a chemical ontology using Methontology and the Ontology Design Environment. Feb 1999. Univ. Politecnica, Madrid.
12. NEMURAITĖ, L; PARADAUSKAS, B; A methodology for engineering OWL 2 ontologies in practise considering their semantic normalisation and completeness . ISSN 1392 – 1215. 2012. No. 4(120).

13. NOY N. F., McGUINNESS D. L. *Ontology Development : A Guide to Creating Your First Ontology*. [žiūrėta 2012-05-21]. Prieiga per Internetą: <<http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html>>
14. *OWL Web Ontology Language* [interaktyvus]. W3C Recommendation 10 February 2004 [žiūrėta 2012-05-21] prieiga per internetą: <<http://www.w3.org/TR/owl-features/>>
15. *Protégé Ontology Editor and Knowledge Acquisition System* [interaktyvus]. [žiūrėta 2012-05-21] prieiga per internetą: <<http://Protégé.stanford.edu/>>
16. *RAP - RDF API for PHP V0.9.6* [interaktyvus]. [žiūrėta 2012-05-21] prieiga per internetą: <<http://www4.wiwiss.fu-berlin.de/bizer/rdfapi/>>.
17. *Resource Description Framework (RDF): Concepts and Abstract Syntax* [interaktyvus]. W3C Recommendation, 10 February 2004. [žiūrėta 2012-05-21] prieiga per internetą: <<http://www.w3.org/TR/rdf-concepts/>>.
18. *TopBraid Composer* [interaktyvus] [žiūrėta 2012-05-21] prieiga per internetą <http://www.topquadrant.com/products/TB_Composer.html>
19. USCHOLD, M.; KING M; *Towards a Methodology for Building Ontologies*. 1995
20. VYŠNIAUSKAS E.; NEMURAITĖ L.; PARADAUSKAS, B. Hybrid method for storing and querying ontologies in databases. *Electronics and Electrical Engineering*. [Kaunas] Technologija, 2011. – Vol. 115, No. 9. –P. 67–72.
21. Wille. R. Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies. In B. Ganter et al. *Formal Concept Analysis*. – Berlin Heidelberg: Springer-Verlag, 2005. – LNAI, Vol. 3626. –P. 1–33.

9. Terminų žodynas

SANTRUMPA	PAAIŠKINIMAS
Portalas	Interneto svetainė leidžianti centralizuotai prieiti prie informacijos internete, per vartotojui patogią grafinę sąsają.
Informacinė sistema	Informacijos apdorojimo sistemos ir organizacijos išteklių visuma, skirta informacijai apdoroti, formuoti (kurti), skleisti (siųsti ir gauti).
Apache	Populiarus atvirojo kodo HTTP serveris, naudojamas <i>Unix</i> ir <i>Windows</i> aplinkose.
Wamp Server	Virtualus lokalus <i>HTTP</i> serveris apimantis <i>Apache</i> , <i>MySQL</i> , <i>PHP</i> technologijas
URI	Vieningas išteklių identifikatorius internete. <i>URI</i> yra pagrindas ontologijų aprašymo kalbose identifikuojant resursus. <i>OWL 2</i> standartas vietoje <i>URI</i> rekomenduojama naudoti <i>IRI</i> .
SPARQL	Konsorciumo <i>W3C</i> rekomenduotina ontologijų užklausų kalba skirta informacijos išgavimui iš <i>RDF</i> duomenų rinkinių.
SIOC	Ontologijos žodynas apjungiantis diskusijas, dienoraščius, elektroninius susirašinėjimus ir kitas socialines bendruomenes.
RuleML	Taisyklių žymėjimo kalbos standartas (<i>XML</i> pagrindu) skirtas taisyklių sintaksei tinkle palaikyti.
SWRL	Konsorciumo <i>W3C</i> rekomenduotina taisyklių kalba apjungianti <i>OWL-DL</i> kalbos ontologiją su <i>RuleML</i> taisyklėmis.
XSL	Scenarijų kalba skirta transformuoti duomenis iš <i>XML</i> dokumento į kitus formatus.
XLST (XSL Transform)	<i>XSL</i> priemonėmis užrašomas taisyklių rinkinys, susiejantis transformacijos šaltinio metamodelį su transformacijos rezultato metamodeliu.
PROLOG	Loginio programavimo kalba. Programuojama predikatais ir taisyklių aibėmis.
OMG	Tarptautinė standartų kūrimo organizacija.
Dublin Core (DC)	Metaduomenų standartas skirtas aprašyti įvairiems elektroniniams ištekliams. <i>Dublin Core</i> sudaro 15 elementų.

10. Priedai

10.1 Portalo duomenų bazės lentelių aprašymas

models		
modelID	int	id-ref, indexed
modelURI	varchar(255)	used at reopening a persistent model
baseURI	varchar(255)	ffects creating of new resources and serialization syntax

statements		
modelID	int	id-ref, indexed
subject	varchar(255)	used at reopening a persistent model
predicate	varchar(255)	ffects creating of new resources and serialization syntax
object	varchar(255)	blob, indexed
l_language	varchar(255)	only if object is a literal, otherwise null
l_datatype	varchar(255)	only if object is a literal, otherwise null
subject_is	varchar(1)	flags whether subject is a resource or blanknode
object_is	varchar(1)	flags whether object is a resource, blanknode or literal

namespaces		
modelID	int	id-ref, indexed
namespace	varchar(255)	namespace (domain)
prefix	varchar(255)	prefix

datasets		
datasetName	varchar(255)	dataset name
defaultModelURI	varchar(255)	modeluri

dataset_model		
datasetName	int	name
modelID	int	model foreign key
graphURI	varchar(255)	graph uri

publication		
uri	varchar(255)	publikacijos uri adresas
modelID	int	ontologijos modelio id
ratio	float	publikacijos citavimo indeksas

citation		
-----------------	--	--

id	int	citavimo rysio id
cited	varchar(255)	cituojamos publikacijos id
source	varchar(255)	publikacijos, kurioje yra citata, id
publication_uri	varchar(255)	publikacijos id adresas

user		
id	int	virtotojo id
name	varchar(255)	virtotojo vardas
surname	varchar(255)	virtotojo pavardė
alias	varchar(255)	virtotojo paskyros vardas
password	varchar(255)	virtotojo paskyros slaptažodis
email	varchar(255)	virtotojo el. pašto adresas
role	int	virtotojo rodės id

role		
id	int	rolės id
title	varchar(255)	rolės pavadinimas

parameters		
initialratio	float	pradinis publikacijos citavimo indeksas
multiplier	float	citavimo ryšio daugiklis

10.2 Portalo duomenų bazės lentelių sukūrimo SQL kodas

```
-- phpMyAdmin SQL Dump
-- version 3.2.0.1

-- Table structure for table `citation`
CREATE TABLE IF NOT EXISTS `citation` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `cited` varchar(255) NOT NULL,
  `source` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=531 ;

-- Table structure for table `publication`
CREATE TABLE IF NOT EXISTS `publication` (
  `uri` varchar(255) NOT NULL,
  `ratio` double DEFAULT NULL,
  `type` varchar(255) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- Table structure for table `role`
CREATE TABLE IF NOT EXISTS `role` (
  `role_id` int(11) NOT NULL AUTO_INCREMENT,
```

```

    `role_name` varchar(30) NOT NULL,
    `role_desc` varchar(100) DEFAULT NULL,
    PRIMARY KEY (`role_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;

-- Dumping data for table `role`
INSERT INTO `role` (`role_id`, `role_name`, `role_desc`) VALUES
(1, 'Administratorius', NULL),
(2, 'Astorius', NULL);

-- Table structure for table `user`
CREATE TABLE IF NOT EXISTS `user` (
  `user_id` int(11) NOT NULL,
  `user_role` int(11) NOT NULL,
  `user_nick` varchar(30) NOT NULL,
  `user_name` varchar(50) NOT NULL,
  `user_surname` varchar(50) NOT NULL,
  `user_pass` varchar(50) NOT NULL,
  `user_email` varchar(30) NOT NULL,
  PRIMARY KEY (`user_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- Dumping data for table `user`
INSERT INTO `user` (`user_id`, `user_role`, `user_nick`, `user_name`, `user_surname`,
`user_pass`, `user_email`) VALUES
(0, 1, 'Mindaugas', 'Gaurilcikas', 'Administratorius', 'adml23', 'admin@admin.com!');

```

10.3 Mokslinių publikacijų OWL ontologijos funkcinė sintaksė

```

Prefix(xsd:<http://www.w3.org/2001/XMLSchema#>)
Prefix(owl2xml:<http://www.w3.org/2006/12/owl2-xml#>)
Prefix(swrlb:<http://www.w3.org/2003/11/swrlb#>)
Prefix(owl:<http://www.w3.org/2002/07/owl#>)
Prefix(:<http://www.promind.lt/publikacijos#>)
Prefix(xml:<http://www.w3.org/XML/1998/namespace>)
Prefix(rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
Prefix(swrl:<http://www.w3.org/2003/11/swrl#>)
Prefix(rdfs:<http://www.w3.org/2000/01/rdf-schema#>)

Ontology(<http://www.promind.lt/publikacijos>

Declaration(Class(:asmuo))
AnnotationAssertion(rdfs:label :asmuo "Asmuo"^^xsd:string)
Declaration(Class(:knyga))
AnnotationAssertion(rdfs:label :knyga "Knyga"^^xsd:string)
SubClassOf(:knyga :publikacija)
Declaration(Class(:leidykla))
AnnotationAssertion(rdfs:label :leidykla "Leidykla"^^xsd:string)
Declaration(Class(:mokslinis_laipsnis))
AnnotationAssertion(rdfs:label :mokslinis_laipsnis "Mokslinis laipsnis"^^xsd:string)
Declaration(Class(:mokymo_institucija))
AnnotationAssertion(rdfs:label :mokymo_institucija "Mokymo institucija"^^xsd:string)
SubClassOf(:mokymo_institucija :organizacija)
Declaration(Class(:organizacija))
AnnotationAssertion(rdfs:label :organizacija "Organizacija"^^xsd:string)
Declaration(Class(:publikacija))
AnnotationAssertion(rdfs:label :publikacija "Publikacija"^^xsd:string)
Declaration(Class(:salis))
AnnotationAssertion(rdfs:label :salis "Išleidimo šalis"^^xsd:string)
Declaration(Class(:straipsnis))

```

```

AnnotationAssertion(rdfs:label :straipsnis "Straipsnis"^^xsd:string)
SubClassOf(:straipsnis :publikacija)
Declaration(Class(:tyrimu_institutas))
AnnotationAssertion(rdfs:label :tyrimu_institutas "Tyrimu institutas"^^xsd:string)
SubClassOf(:tyrimu_institutas :organizacija)
Declaration(Class(:zurnalas))
AnnotationAssertion(rdfs:label :zurnalas "Žurnalas"^^xsd:string)
SubClassOf(:zurnalas :publikacija)
Declaration(Class(owl:Thing))
Declaration(ObjectProperty(:cituoja_autoriu))
AnnotationAssertion(rdfs:label :cituoja_autoriu "cituoja autoriu"^^xsd:string)
InverseObjectProperties(:cituojamas_autoriaus :cituoja_autoriu)
ObjectPropertyDomain(:cituoja_autoriu :asmuo)
ObjectPropertyRange(:cituoja_autoriu :asmuo)
Declaration(ObjectProperty(:cituoja_publikacija))
AnnotationAssertion(rdfs:label :cituoja_publikacija "cituoja publikacija"^^xsd:string)
InverseObjectProperties(:cituojama_publikacijoje :cituoja_publikacija)
ObjectPropertyDomain(:cituoja_publikacija :publikacija)
ObjectPropertyRange(:cituoja_publikacija :publikacija)
Declaration(ObjectProperty(:cituojama_publikacijoje))
AnnotationAssertion(rdfs:label :cituojama_publikacijoje "cituojama publikacijoje"^^xsd:string)
InverseObjectProperties(:cituojama_publikacijoje :cituoja_publikacija)
ObjectPropertyDomain(:cituojama_publikacijoje :publikacija)
ObjectPropertyRange(:cituojama_publikacijoje :publikacija)
Declaration(ObjectProperty(:cituojamas_autoriaus))
AnnotationAssertion(rdfs:label :cituojamas_autoriaus "cituojamas autoriaus"^^xsd:string)
InverseObjectProperties(:cituojamas_autoriaus :cituoja_autoriu)
ObjectPropertyDomain(:cituojamas_autoriaus :asmuo)
ObjectPropertyRange(:cituojamas_autoriaus :asmuo)
Declaration(ObjectProperty(:gautas_organizacijoje))
AnnotationAssertion(rdfs:label :gautas_organizacijoje "gautas organizacijoje"^^xsd:string)
InverseObjectProperties(:suteiktas_laipsnis :gautas_organizacijoje)
ObjectPropertyDomain(:gautas_organizacijoje :mokslinis_laipsnis)
ObjectPropertyRange(:gautas_organizacijoje :organizacija)
Declaration(ObjectProperty(:isleido_publikacija))
AnnotationAssertion(rdfs:label :isleido_publikacija "išleido publikacija"^^xsd:string)
InverseObjectProperties(:isleido_publikacija :isleista_leidykloje)
ObjectPropertyDomain(:isleido_publikacija :leidykla)
ObjectPropertyRange(:isleido_publikacija :publikacija)
Declaration(ObjectProperty(:isleista_leidykloje))
AnnotationAssertion(rdfs:label :isleista_leidykloje "išleista leidykloje"^^xsd:string)
InverseObjectProperties(:isleido_publikacija :isleista_leidykloje)
ObjectPropertyDomain(:isleista_leidykloje :publikacija)
ObjectPropertyRange(:isleista_leidykloje :leidykla)
Declaration(ObjectProperty(:parase_publikacija))
AnnotationAssertion(rdfs:label :parase_publikacija "yra publikacijos autorius/bendraautorius"^^xsd:string)
InverseObjectProperties(:parasyta_autoriaus :parase_publikacija)
ObjectPropertyDomain(:parase_publikacija :asmuo)
ObjectPropertyRange(:parase_publikacija :publikacija)
Declaration(ObjectProperty(:parasyta_autoriaus))
AnnotationAssertion(rdfs:label :parasyta_autoriaus "parasyta autoriaus"^^xsd:string)
InverseObjectProperties(:parasyta_autoriaus :parase_publikacija)
ObjectPropertyDomain(:parasyta_autoriaus :publikacija)
ObjectPropertyRange(:parasyta_autoriaus :asmuo)
Declaration(ObjectProperty(:priklauso_asmeniui))
AnnotationAssertion(rdfs:label :priklauso_asmeniui "priklauso asmeniui"^^xsd:string)
InverseObjectProperties(:turi_laipsni :priklauso_asmeniui)
ObjectPropertyDomain(:priklauso_asmeniui :mokslinis_laipsnis)
ObjectPropertyRange(:priklauso_asmeniui :asmuo)
Declaration(ObjectProperty(:suteiktas_laipsnis))
AnnotationAssertion(rdfs:label :suteiktas_laipsnis "suteiktas laipsnis"^^xsd:string)
InverseObjectProperties(:suteiktas_laipsnis :gautas_organizacijoje)
ObjectPropertyDomain(:suteiktas_laipsnis :organizacija)
ObjectPropertyRange(:suteiktas_laipsnis :mokslinis_laipsnis)
Declaration(ObjectProperty(:turi_citavimo_lygi))
AnnotationAssertion(rdfs:label :turi_citavimo_lygi "turi citavimo lygi"^^xsd:string)
Declaration(ObjectProperty(:turi_laipsni))
AnnotationAssertion(rdfs:label :turi_laipsni "turi laipsni"^^xsd:string)
InverseObjectProperties(:turi_laipsni :priklauso_asmeniui)
ObjectPropertyDomain(:turi_laipsni :asmuo)
ObjectPropertyRange(:turi_laipsni :mokslinis_laipsnis)
Declaration(DataProperty(:asmuo_pavarde))
AnnotationAssertion(rdfs:label :asmuo_pavarde "Pavardė"^^xsd:string)
DataPropertyDomain(:asmuo_pavarde :asmuo)

```

```

DataPropertyRange(:asmuo_pavarde xsd:string)
Declaration(DataProperty(:asmuo_vardas))
AnnotationAssertion(rdfs:label :asmuo_vardas "Vardas"^^xsd:string)
DataPropertyDomain(:asmuo_vardas :asmuo)
DataPropertyRange(:asmuo_vardas xsd:string)
Declaration(DataProperty(:cit_puslapiai))
AnnotationAssertion(rdfs:label :cit_puslapiai "Puslapiai"^^xsd:string)
DataPropertyRange(:cit_puslapiai xsd:int)
Declaration(DataProperty(:la_pavadinimas))
AnnotationAssertion(rdfs:label :la_pavadinimas "Mokslinis laipsnis"^^xsd:string)
DataPropertyDomain(:la_pavadinimas :mokslinis_laipsnis)
DataPropertyRange(:la_pavadinimas xsd:string)
Declaration(DataProperty(:leid_adresas))
AnnotationAssertion(rdfs:label :leid_adresas "Adresas"^^xsd:string)
DataPropertyDomain(:leid_adresas :leidykla)
DataPropertyRange(:leid_adresas xsd:string)
Declaration(DataProperty(:leid_pavadinimas))
AnnotationAssertion(rdfs:label :leid_pavadinimas "Pavadinimas"^^xsd:string)
DataPropertyDomain(:leid_pavadinimas :leidykla)
DataPropertyRange(:leid_pavadinimas xsd:string)
Declaration(DataProperty(:org_adresas))
AnnotationAssertion(rdfs:label :org_adresas "Adresas"^^xsd:string)
DataPropertyDomain(:org_adresas :organizacija)
DataPropertyRange(:org_adresas xsd:string)
Declaration(DataProperty(:org_pavadinimas))
AnnotationAssertion(rdfs:label :org_pavadinimas "Pavadinimas"^^xsd:string)
DataPropertyDomain(:org_pavadinimas :organizacija)
DataPropertyRange(:org_pavadinimas xsd:string)
Declaration(DataProperty(:p_aprasymas))
AnnotationAssertion(rdfs:label :p_aprasymas "Aprašymas"^^xsd:string)
DataPropertyDomain(:p_aprasymas :publikacija)
DataPropertyRange(:p_aprasymas xsd:string)
Declaration(DataProperty(:p_isleidimoData))
AnnotationAssertion(rdfs:label :p_isleidimoData "Išleidimo data"^^xsd:string)
DataPropertyDomain(:p_isleidimoData :publikacija)
DataPropertyRange(:p_isleidimoData xsd:date)
Declaration(DataProperty(:p_kodas))
AnnotationAssertion(rdfs:label :p_kodas "Publikacijos kodas"^^xsd:string)
DataPropertyDomain(:p_kodas :publikacija)
DataPropertyRange(:p_kodas xsd:string)
Declaration(DataProperty(:p_pavadinimas))
AnnotationAssertion(rdfs:label :p_pavadinimas "Publikacijos pavadinimas"^^xsd:string)
DataPropertyDomain(:p_pavadinimas :publikacija)
DataPropertyRange(:p_pavadinimas xsd:string)
Declaration(DataProperty(:p_paveikslelioURL))
AnnotationAssertion(rdfs:label :p_paveikslelioURL "Paveikslėlio URL"^^xsd:string)
Declaration(NamedIndividual(:Vytauto_Didžiojo_universitetas))
AnnotationAssertion(rdfs:label :Vytauto_Didžiojo_universitetas "Vytauto Didžiojo universitetas"^^xsd:string)
ClassAssertion(:organizacija :Vytauto_Didžiojo_universitetas)
DataPropertyAssertion(:org_adresas :Vytauto_Didžiojo_universitetas "Donelaičio g. 58, LT-44248"^^xsd:string)
DataPropertyAssertion(:org_pavadinimas :Vytauto_Didžiojo_universitetas "Vytauto Didžiojo universitetas"^^xsd:string)
Declaration(NamedIndividual(:balciunaitis))
AnnotationAssertion(rdfs:label :balciunaitis "Marius Balčiunaitis"^^xsd:string)
ClassAssertion(:asmuo :balciunaitis)
ObjectPropertyAssertion(:parase_publicacija :balciunaitis :informatika)
DataPropertyAssertion(:asmuo_pavarde :balciunaitis "Balciunaitis"^^xsd:string)
DataPropertyAssertion(:asmuo_vardas :balciunaitis "Marius"^^xsd:string)
Declaration(NamedIndividual(:business_rules))
AnnotationAssertion(rdfs:label :business_rules "Business Rules as Part of Information Systems Life Cycle"^^xsd:string)
ClassAssertion(:straipsnis :business_rules)
ObjectPropertyAssertion(:cituoja_publicacija :business_rules :duomenu_bazes)
ObjectPropertyAssertion(:isleista_leidykloje :business_rules :technologija)
ObjectPropertyAssertion(:parasyta_autorias :business_rules :nemuraite)
ObjectPropertyAssertion(:parasyta_autorias :business_rules :balciunaitis)
DataPropertyAssertion(:p_isleidimoData :business_rules "2011-06-19"^^xsd:date)
DataPropertyAssertion(:p_pavadinimas :business_rules "Business Rules as Part of Information Systems Life Cycle"^^rdfs:Literal)
Declaration(NamedIndividual(:daktaras))
AnnotationAssertion(rdfs:label :daktaras "Daktaras"^^xsd:string)
ClassAssertion(:mokslinis_laipsnis :daktaras)
ObjectPropertyAssertion(:gautas_organizacijoje :daktaras :ktu)
ObjectPropertyAssertion(:priklauso_asmeniui :daktaras :nemuraite)

```

```

DataPropertyAssertion(:la_pavadinimas :daktaras "Daktaras"^^xsd:string)
Declaration(NamedIndividual(:danija))
AnnotationAssertion(rdfs:label :danija "Danija"^^xsd:string)
ClassAssertion(:salis :danija)
Declaration(NamedIndividual(:diskrecioji_matematika))
AnnotationAssertion(rdfs:label :diskrecioji_matematika "Diskrečioji matematika"^^xsd:string)
ClassAssertion(:knyga :diskrecioji_matematika)
ObjectPropertyAssertion(:cituoja_publicacija :diskrecioji_matematika :matematika)
ObjectPropertyAssertion(:cituojama_publicacijoje :diskrecioji_matematika :matematika)
Declaration(NamedIndividual(:duomenu_bazes))
AnnotationAssertion(rdfs:label :duomenu_bazes "Duomenų bazės ir semantiniai modeliai"^^xsd:string)
ClassAssertion(:knyga :duomenu_bazes)
ObjectPropertyAssertion(:cituojama_publicacijoje :duomenu_bazes :business_rules)
ObjectPropertyAssertion(:cituojama_publicacijoje :duomenu_bazes :informatika)
ObjectPropertyAssertion(:isleista_leidykloje :duomenu_bazes :technologija)
ObjectPropertyAssertion(:parasyta_autoriaus :duomenu_bazes :nemuraite)
DataPropertyAssertion(:p_aprasymas :duomenu_bazes "Semantinis modeliavimas - tai duomenų loginis struktūrizavimas pagal jų prasmę. Dažnai duomenys tvarkomi neatsižvelgiant į prasmę ar interpretaciją, pavyzdžiui, pagal abėcėlę; tuo tarpu semantiniai modeliai asocijuojasi su realiais objektais ir jų savybėmis, tokiais, kokias jas suvokia žmogus, todėl yra lengvai suprantami. Semantinis modeliavimas yra pagrįstas loginėmis savokų kategorijomis. Savokos mąstyme fiksuojamos, įgauna apibrėžtumo terminų dėka, t.y., operuojame tik tomis savokomis, kurios gali būti išreikštos kalba."^^xsd:string)
DataPropertyAssertion(:p_kodas :duomenu_bazes "ISBN 9955-09-436-2"^^xsd:string)
DataPropertyAssertion(:p_pavadinimas :duomenu_bazes "Duomenų bazių semantiniai modeliai"^^xsd:string)
Declaration(NamedIndividual(:ekonomika))
AnnotationAssertion(rdfs:label :ekonomika "Ekonomikos teorijos pagrindai"^^xsd:string)
ClassAssertion(:knyga :ekonomika)
ObjectPropertyAssertion(:parasyta_autoriaus :ekonomika :jakutis)
ObjectPropertyAssertion(:parasyta_autoriaus :ekonomika :lenkevicius)
DataPropertyAssertion(:p_aprasymas :ekonomika "Šioje knygoje ekonomikos teorijos pradžios. Knygos turinys - tai tas minimumas, kuris bus pamatas, perteikiant ir perimant esmines žinias apie ūkį šių dienų sąlygomis. Mokslas, tiriantis visuomenės ūkinę v e i k l a, seniai išsirutuliojo į išsikerojusią ekonomikos mokslų sistemą; ekonomikos teorijos kursas liko jos įvadinė integruota mokomoji disciplina, teikianti pirmuosius įgūdžius norintiems tapti profesionaliais šių laikų ekonomikos žinovais"^^xsd:string)
DataPropertyAssertion(:p_kodas :ekonomika "ISBN 9986-708-39-7"^^xsd:string)
DataPropertyAssertion(:p_pavadinimas :ekonomika "Ekonomikos teorijos pagrindai"^^xsd:string)
Declaration(NamedIndividual(:informatika))
AnnotationAssertion(rdfs:label :informatika "Informatikos istorija"^^xsd:string)
ClassAssertion(:straipsnis :informatika)
ObjectPropertyAssertion(:cituoja_publicacija :informatika :matematika)
ObjectPropertyAssertion(:cituojama_publicacijoje :informatika :matematika)
ObjectPropertyAssertion(:isleista_leidykloje :informatika :vaga)
ObjectPropertyAssertion(:parasyta_autoriaus :informatika :balciunaitis)
DataPropertyAssertion(:p_isleidimoData :informatika "2011-07-14"^^xsd:date)
DataPropertyAssertion(:p_pavadinimas :informatika "Informatika"^^xsd:string)
Declaration(NamedIndividual(:jakutis))
AnnotationAssertion(rdfs:label :jakutis "Algirdas Jakutis"^^xsd:string)
ClassAssertion(:asmuo :jakutis)
ObjectPropertyAssertion(:parase_publicacija :jakutis :ekonomika)
DataPropertyAssertion(:asmuo_pavarde :jakutis "Jakutis"^^xsd:string)
DataPropertyAssertion(:asmuo_vardas :jakutis "Algirdas"^^xsd:string)
Declaration(NamedIndividual(:ktu))
AnnotationAssertion(rdfs:label :ktu "Kauno technologijos universitetas"^^xsd:string)
ClassAssertion(:organizacija :ktu)
ObjectPropertyAssertion(:suteiktas_laipsnis :ktu :profesorius)
ObjectPropertyAssertion(:suteiktas_laipsnis :ktu :daktaras)
DataPropertyAssertion(:org_adresas :ktu "Studentų g 55"^^xsd:string)
ObjectPropertyAssertion(:turi_laipsni :lenkevicius :profesorius)
DataPropertyAssertion(:asmuo_pavarde :lenkevicius "Lenkevičius"^^xsd:string)
DataPropertyAssertion(:asmuo_vardas :lenkevicius "Tadas"^^xsd:string)
Declaration(NamedIndividual(:lietuva))
AnnotationAssertion(rdfs:label :lietuva "Lietuva"^^xsd:string)
ClassAssertion(:salis :lietuva)
Declaration(NamedIndividual(:magistrantas))
AnnotationAssertion(rdfs:label :magistrantas "Magistrantas"^^xsd:string)
ClassAssertion(:mokslinis_laipsnis :magistrantas)
ObjectPropertyAssertion(:priklauso_asmeniui :magistrantas :mindaugas)
DataPropertyAssertion(:la_pavadinimas :magistrantas "Maistras"^^xsd:string)
Declaration(NamedIndividual(:matematika))
AnnotationAssertion(rdfs:label :matematika "Taikomoji matematika"^^xsd:string)
ClassAssertion(:knyga :matematika)

```

```

ObjectPropertyAssertion(:cituojama_publicacijoje :matematika :diskrecioji_matematika)
ObjectPropertyAssertion(:cituojama_publicacijoje :matematika :telekomunikacija)
ObjectPropertyAssertion(:parasyta_autoriaus :matematika :balciunaitis)
DataPropertyAssertion(:p_aprasymas :matematika "SPARQL knows nothing about OWL restrictions,
or the distinction between object and datatype properties, for example, and completely
ignores"^^xsd:string)
DataPropertyAssertion(:p_isleidimoData :matematika "2010-01-03"^^xsd:date)
DataPropertyAssertion(:p_kodas :matematika "EN-458544"^^xsd:string)
DataPropertyAssertion(:p_pavadinimas :matematika "Matematika"^^xsd:string)
DataPropertyAssertion(:p_paveikslelioURL :matematika
"http://i47.tinypic.com/54em8l.jpg"^^xsd:string)
Declaration(NamedIndividual(:mindaugas))
AnnotationAssertion(rdfs:label :mindaugas "Mindaugas Gaurilčikas"^^xsd:string)
ClassAssertion(:asmuo :mindaugas)
ObjectPropertyAssertion(:parase_publicacija :mindaugas :telekomunikacija)
ObjectPropertyAssertion(:parase_publicacija :mindaugas :ontologiju_taikymas)
ObjectPropertyAssertion(:turi_laipsni :mindaugas :magistrantas)
DataPropertyAssertion(:asmuo_pavarde :mindaugas "Gaurilčikas"^^xsd:string)
DataPropertyAssertion(:asmuo_vardas :mindaugas "Mindaugas"^^xsd:string)
Declaration(NamedIndividual(:molekuline_teorija))
ObjectPropertyAssertion(:parase_publicacija :nemuraite :business_rules)
ObjectPropertyAssertion(:parase_publicacija :nemuraite :duomenu_bazes)
ObjectPropertyAssertion(:turi_laipsni :nemuraite :daktaras)
DataPropertyAssertion(:asmuo_pavarde :nemuraite "Nemuraitė"^^xsd:string)
DataPropertyAssertion(:asmuo_vardas :nemuraite "Lina"^^xsd:string)
Declaration(NamedIndividual(:ontologiju_taikymas))
AnnotationAssertion(rdfs:label :ontologiju_taikymas "Ontologiju taikymas
portaluose"^^xsd:string)
ClassAssertion(:straipsnis :ontologiju_taikymas)
ObjectPropertyAssertion(:parasyta_autoriaus :ontologiju_taikymas :mindaugas)
DataPropertyAssertion(:p_aprasymas :ontologiju_taikymas "aprasymas"^^xsd:string)
DataPropertyAssertion(:p_isleidimoData :ontologiju_taikymas "2012-02-11"^^xsd:date)
DataPropertyAssertion(:p_kodas :ontologiju_taikymas "ESB-884845-5DF77"^^xsd:string)
DataPropertyAssertion(:p_pavadinimas :ontologiju_taikymas "Ontologiju taikymas
portaluose"^^xsd:string)
Declaration(NamedIndividual(:paradauskas))
AnnotationAssertion(rdfs:label :paradauskas "Bronius Paradauskas"^^xsd:string)
ClassAssertion(:asmuo :paradauskas)
ObjectPropertyAssertion(:parase_publicacija :paradauskas :duomenu_bazes)
ObjectPropertyAssertion(:turi_laipsni :paradauskas :profesorius)
DataPropertyAssertion(:asmuo_pavarde :paradauskas "Paradauskas"^^xsd:string)
DataPropertyAssertion(:asmuo_vardas :paradauskas "Bronius"^^xsd:string)
Declaration(NamedIndividual(:profesorius))
AnnotationAssertion(rdfs:label :profesorius "Profesorius"^^xsd:string)
ClassAssertion(:mokslinis_laipsnis :profesorius)
ObjectPropertyAssertion(:gautas_organizacijoje :profesorius :ktu)
ObjectPropertyAssertion(:priklauso_asmeniui :profesorius :paradauskas)
DataPropertyAssertion(:la_pavadinimas :profesorius "Profesorius"^^xsd:string)
Declaration(NamedIndividual(:technologija))
AnnotationAssertion(rdfs:label :technologija "Technologija"^^xsd:string)
ClassAssertion(:leidykla :technologija)
ObjectPropertyAssertion(:isleido_publicacija :technologija :business_rules)
ObjectPropertyAssertion(:isleido_publicacija :technologija :duomenu_bazes)
DataPropertyAssertion(:leid_pavadinimas :technologija "Technologija"^^xsd:string)
Declaration(NamedIndividual(:telekomunikacija))
AnnotationAssertion(rdfs:label :telekomunikacija "Skaitmeninės komunikacijos pagrindai
"^^xsd:string)
ClassAssertion(:knyga :telekomunikacija)
ObjectPropertyAssertion(:cituoja_publicacija :telekomunikacija :matematika)
DataPropertyAssertion(:p_aprasymas :telekomunikacija "Komunikacijos pagrindai Komunikacijos
pagrindai Komunikacijos pagrindai Komunikacijos pagrindai Komunikacijos pagrindai
Komunikacijos pagrindai Komunikacijos pagrindai Komunikacijos pagrindai Komunikacijos
pagrindai"^^xsd:string)
DataPropertyAssertion(:p_isleidimoData :telekomunikacija "2012-03-01"^^xsd:date)
DataPropertyAssertion(:p_pavadinimas :telekomunikacija "Skatimeninės komunikacijos
pagrindai"^^xsd:string)
Declaration(NamedIndividual(:vaga))
AnnotationAssertion(rdfs:label :vaga "Vaga"^^xsd:string)
ClassAssertion(:leidykla :vaga)
ObjectPropertyAssertion(:isleido_publicacija :vaga :informatika)
DataPropertyAssertion(:leid_adresas :vaga "Sukilėlių g.78, LT-78001 Vilnius"^^xsd:string)
DataPropertyAssertion(:leid_pavadinimas :vaga "Vaga"^^xsd:string)
Declaration(Datatype(xsd:date))

```