

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Kęstutis Spudys

**Dinaminio semantinių užklausų
formavimo sąsaja**

Magistro darbas

Darbo vadovas

prof. L. Nemuraitė

Kaunas, 2012

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Kęstutis Spudys

**Dinaminio semantinių užklausų
formavimo sąsaja**

Magistro darbas

Recenzentas

doc. dr. R. Misevičienė

2012-05-25

Vadovas

prof. L. Nemuraitė

2012-05-25

Atliko

IFM-0/4 gr. stud.

Kęstutis Spudys

2012-05-25

Kaunas, 2012

Turinys

SUMMARY	4
TERMINŲ IR TRUMPINIŲ ŽODYNAS.....	5
1. ĮVADAS.....	6
2. SEMANTINĖS PAIEŠKOS SĄSAJOS MODELIO IR JO TAIKymo SEMANTINIAM PAIEŠKOS PORTALUI ANALIZĖ.....	10
2.1. TYRIMO OBJEKTAS, TIKSLAS IR UŽDAVINIAI	10
2.2. SEMANTINIŲ PAIEŠKOS KALBŲ ANALIZĖ	11
2.2.1. <i>OWL kalbos analizė.....</i>	<i>12</i>
2.3. SEMANTINĖS PAIEŠKOS VARTOTOJŲ POREIKIŲ ANALIZĖ	14
2.3.1. <i>Paieškos vartotojų tikslai ir problemos.....</i>	<i>14</i>
2.3.2. <i>Paieškos vartotojo sąsaja</i>	<i>15</i>
2.4. ESAMŲ SEMANTINĖS PAIEŠKOS PORTALŲ ANALIZĖ	16
2.4.1. <i>Paieškos portalų savybės.....</i>	<i>16</i>
2.4.2. <i>Detali esamų paieškos modelių analizė.....</i>	<i>19</i>
2.4.2.1. <i>Užklauskos sudarymas.....</i>	<i>19</i>
2.4.2.2. <i>Paieškos algoritmas.....</i>	<i>21</i>
2.4.2.3. <i>Rezultatų pateikimas.....</i>	<i>21</i>
2.4.2.4. <i>Sąveika su rezultatais ir užklauskų tikslinimas.....</i>	<i>23</i>
2.4.2.5. <i>Semantinės paieškos kokybės vertinimo kriterijai.....</i>	<i>24</i>
2.5. SEMANTINĖS PAIEŠKOS TINKLALAPIŲ REALIZAVIMO ĮRANKIŲ ANALIZĖ.....	25
2.5.1. <i>Programavimo kalbos, karkasai ir ontologijų manipuliavimo bibliotekos.....</i>	<i>25</i>
2.5.2. <i>Ontologijų saugyklos.....</i>	<i>26</i>
2.5.3. <i>Ontologijų kūrimo ir modifikavimo įrankiai.....</i>	<i>27</i>
2.6. SIEKIAMAS SPRENDIMAS	27
2.7. ANALIZĖS IŠVADOS.....	29
3. DINAMINĖS VARTOTOJO SĄSAJOS FORMAVIMO METODAS	30
4. EKSPERIMENTINĖS SISTEMOS KŪRIMAS.....	34
4.1. REIKALAVIMŲ SPECIFIKACIJA	34
4.1.1. <i>Funkciniai reikalavimai.....</i>	<i>34</i>
4.1.2. <i>Nefunkciniai reikalavimai.....</i>	<i>39</i>
4.2. SISTEMOS PROJEKTAS	40
4.2.1. <i>Sistemos architektūros projektas.....</i>	<i>40</i>
4.2.1.2. <i>Vartotojo, veiklos ir duomenų paslaugos.....</i>	<i>41</i>
4.2.1.3. <i>Sistemos klasės</i>	<i>42</i>
4.2.2. <i>Sistemos elgsenos modelis</i>	<i>43</i>
4.2.3. <i>Ontologijos konceptų schema</i>	<i>46</i>
4.2.4. <i>Realizacijos modelis.....</i>	<i>47</i>
4.3. SISTEMOS REALIZACIJA.....	48
4.3.1. <i>Vartotojų sąsajos modelis.....</i>	<i>48</i>
4.3.2. <i>Sistemos veikimo aprašymas.....</i>	<i>49</i>
4.4. TESTAVIMO MODELIS IR DUOMENYS.....	55
5. EKSPERIMENTINIS SUKURTO SEMANTINIŲ UŽKLAUSŲ MODELIO TYRIMAS IR ĮVERTINIMAS	58
5.1. UŽKLAUSŲ KOKYBINIŲ PARAMETRŲ EKSPERIMENTINIS TYRIMAS.....	58
5.1.1. <i>Pirmo tipo užklausa.....</i>	<i>59</i>
5.1.2. <i>Antro tipo užklausa.....</i>	<i>63</i>
5.1.3. <i>Eksperimento rezultatai</i>	<i>68</i>
5.2. UŽKLAUSŲ FORMAVIMO IR PAIEŠKOS PROCESO ĮVERTINIMAS.....	69
6. IŠVADOS.....	71
7. LITERATŪRA	72

Dynamic user interface for semantic queries

Summary

Increasing popularity of Semantic Web raises a question how we could make a simple user interface for building semantic queries while keeping high precision of results returned. This thesis presents a method that helps users to create SPARQL queries by allowing to dynamically add components to user interface. The goal of the work is to improve of user interface model for semantic queries by allowing users to construct and change it dynamically till obtaining the desirable answer results. That model was created on the base of analysis of Semantic Web languages, tools and existing portals, their functions and user interfaces. Algorithms for dynamic user interface generation based on user actions were developed that allow creating queries of various complexities with minimal amount of user interface components. Implementation and testing the prototype of the system using movie and wine ontologies has shown that dynamic construction and generation of query interface has desirable functionality and is easily applicable to various ontologies. Experimental comparison with existing semantic search portals has shown that the proposed dynamic user interface generation method could improve precision and recall of semantic queries and may be applied in semantic search portal applications.

Keywords: ontology; semantic search; semantic query; *OWL*; *SPARQL*; query generation.

Terminų ir trumpinių žodynas

OWL – Web Ontology Language trumpinys. Aukšto lygio ontologijų aprašymo kalba.

SPARQL - SPARQL Protocol and RDF Query Language. Užklausų kalba RDF paremtoms duomenų bazėms.

RDF – Resource Description Framework. Grupė specifikacijų, skirtų aprašyti duomenims grafo pavidalu.

Applet'as – Java kalbos programa, talpinama tinklapyje ir vykdoma kliento kompiuteryje.

1. Įvadas

Semantika informacinių technologijų kontekste yra duomenų apdorojimas pagal prasmę ir kontekstą. Tam įgyvendinti yra taikomas natūralios kalbos apdorojimas, pritaikytas informacijos paieškai, išrinkimui, analizavimui ir panaudojimui.

Šiuo metu labiausiai paplitusi raktiniais žodžiais grįsta paieška gražina visus rezultatus kuriuose yra randami raktiniai žodžiai, tuo tarpu semantinės paieškos tikslas yra išgauti teisingą ir tikslią informaciją, suprantant informacijos ieškotojo tikslus ir užklausoje pateiktų žodžių prasmę, aprašytą ontologijose.

Semantinė paieška gali būti apibrėžta kaip:

- Pagalbinė įprastos raktiniais žodžiais pagrįstos paieškos priemonė, duodanti tikslesnę, geriau vartotojo tikslus atitinkančią informaciją.
- Į klausimus atsakanti sistema, kur vartotojas užduoda klausimą natūralia kalba. Šį variantą yra labai sudėtinga įgyvendinti, šiuo metu vartotojas duoda užklausas pildydamas iš anksto sudarytas formas.
- Užklausoje, kurios gražina prasmingus atsakymus sudarytus iš XML dokumentų fragmentų, o ne pačius dokumentus.
- Užklausoje, panašios į SQL užklausoje, skirtos RDF aprašytoms duomenų bazėms.
- Atsakymo išgavimas sistemoje, kurioje yra visa tam reikalinga RDF aprašyta informacija, bet nėra nurodytas pats atsakymas.

Patogumo dėlei semantinę paiešką galima suskirstyti į vykdomą visame tinkle esančiuose dokumentuose (HTML) ir vykdomą tik pasirinktose ontologijose (OWL/RDF).

Visame tinkle paiešką atliekančiose sistemose dokumentai nėra priskirti kokiam nors ontologijos daliai vadinamos laisvai susietomis (angl. *loosely coupled*).

Dokumentai šiuo atveju yra *HTML* ar panašūs puslapiai kuriuose yra patalpinta informacija, arba *XML* kodo dalys nusakančios informacijos objektus.

Semantinė paieška pasirinktose ontologijose gali atsakyti į sudėtingus klausimus, susijusius su objektais ir ryšiais tarp jų. Tokia sistema būtų glaudžiai susieta (angl. *tightly coupled*). Joje kiekvienas dokumentas yra priskirtas kokiam nors ontologijos konceptui.

Paieška tokiose sistemose susideda iš dviejų dalių: naršymo (angl. *browsing*) ir užklausų (angl. *querying*). Naršymas yra pagrįstas prasminiais konceptų ryšiais, todėl vartotojas negali gauti atsakymų į laisvas semantines užklausas. Jis gali tik suformuoti kelią, vedantį link tam tikrų konceptų.

Kai paieškos modelis naudoja minimalų kiekį formos komponentų, tarkime viena lauką, vartotojas turi tiksliai suformuluoti užklausą. Tai gali būti gana sunku nepatyrusiems vartotojams, kuriems semantinė paieška yra naujovė. Didinant komponentų skaičių sumažinamas vartotojo kvalifikacijos reikalavimas, bet vartotojas apkraunamas pasirinkimais ir teksto įvedimo laukais.

Taip yra todėl, kad semantinė paieška leidžia rasti tikslius atsakymus, bet tam reikia suformuoti tikslias užklausas, dėl to vartotojo sąsajoje, be įprasto raktažodžių įvedimo lauko, atsiranda daugybė papildomų elementų. Taigi portalų vartotojo sąsajas galima suskirstyti į apkrautas elementais ir gražinančias tikslius rezultatus, arba vieno teksto įvedimo lauko sąsajas, kurios gražina grupę rezultatų. Gavus grupę rezultatų vartotojas vis tiek apkraunamas rezultatų filtravimo ir/arba užklausos tikslinimo elementais, kas prilygsta pirmajam variantui.

Šio darbo sritis yra - *OWL* ir ja paremtų portalų paieškos sąsajos modelių taikymas semantinių portalų kūrimo.

Kadangi semantinių portalų sąsajai nėra išbaigtų, vartotojo poreikius atitinkančių sprendimų, siekiama patobulinti semantinio portalų sąsajos modelį remiantis geriausiais esamais semantinės ir struktūrizuotos paieškos sistemų pavyzdžiais. Šiam tikslui pasiekti reikia išspręsti šiuos uždavinius:

1. Atlikti literatūros analizę ir ištirti esančius paieškos sąsajų modelius:
 - 1.1. Išanalizuoti *SPARQL*, *OWL*, *RDF* ir susijusias sąvokas;
 - 1.2. Išanalizuoti literatūroje pateiktus semantinės paieškos modelius;
 - 1.3. Išanalizuoti esamų semantinių portalų architektūras ir paieškos modelius;
 - 1.4. Išanalizuoti tokių sistemų kūrimo įrankius;
2. Sudaryti semantinės paieškos sąsajos modelį ir suprojektuoti pasirinktos dalykinės srities informacinę sistemą su semantine paieška;
3. Realizuoti semantinės paieškos sistemos prototipą;
4. Atlikti eksperimentą ir įvertinti paieškos modelio privalumus ir trūkumus;

Analizės dalyje ištirtos *OWL* (*Web Ontology Language*) kalbos savybės ir iš trijų *OWL* kalbos pokalbių pasirinkta *OWL DL*, nes ji suteikia maksimalų išraiškingumą ir užtikrina skaičiavimų baigtumą (užtikrinama, kad visos išvados yra

garantuotai suskaičiuojamos) ir sprendimo baigtumą (visi skaičiavimai bus atlikti per baigtinį laiką).

Atliekant semantinės paieškos vartotojų poreikių analizę apibrėžti vartotojų tikslai ir problemos, bei iškelti pagrindiniai vartotojo sąsajos reikalavimai. Semantinės paieškos vartotojo sąsaja ji turi:

- būti intuityvi
- suprantamai pateikti paieškos rezultatus
- padėti vartotojui suformuoti užklausą

Tiriant esamus paieškos, kadangi šis darbas orientuotas į glaudžiai susietas sistemas, buvo atrinkti penki glaudžiai susieti portalai. Paieškos procesas buvo suskirstytas į keturis etapus. Tam panaudojamos IKS – Interactive Knowledge Stack projekto tinklapyje pateiktos turinio valdymo sistemų (angl. CMS) semantinių savybių vertinimo lentelės [29]. Lentelėse sistemos nagrinėjamos keturiomis fazėmis: užklausos sudarymo, paieškos algoritmo, rezultatų pateikimo ir sąveikos su rezultatais bei užklausų tikslinimo.

Nuspręsta leisti vartotojui pasirinkti kokio tikslumo paiešką jis norėtų atlikti. Kad vartotojai, kurie žino ko ieško galėtų greičiau rasti dokumentus, pasirinkę tikslią paiešką. Vartotojai kurie ieško bendresnės informacijos galėtų pasirinkti platesnę paiešką ir rinktis iš grąžintų rezultatų.

Atliekant semantinės paieškos tinklapių realizavimo įrankių analizę, buvo lyginti programavimo karkasai, ontologijų manipuliavimo bibliotekos ir jų programavimo kalbos, bei ontologijų kūrimo ir modifikavimo įrankiai.

Kadangi prototipas yra tinklapis, pasirinktos Javascript kalba tinklapio kūrimui ir informacijos atvaizdavimui bei Jena karkasas, dėl išsamios dokumentacijos ir didelio funkcionalumo dirbant su SPARQL užklausomis ir ontologijomis. Jena karkasas sukurtas Java kalba, todėl kuriant prototipą irgi pasirinkta Java programavimo kalba.

Ontologijai saugoti pasirinktas Joseki serveris, nes jis pateikiamas kaip standartinis ontologijų saugojimo serveris Jena karkasui. Dėl didelio mokomosios medžiagos kiekio ir išsamios dokumentacijos ontologijai kurti ir modifikuoti pasirinktas Protégé įrankis.

Prototipo realizacijoje pradžioje vartotojui pateikiami tik keli sąsajos elementai ir po to leidžiama pačiam pridėti kitus elementus norimam užklausos

sudėtingumui pasiekti. Kadangi tokia sąsaja nepatyrusiam vartotojui iš pirmo žvilgsnio gali atrodyti sudėtinga, sistema, skirtingai nuo kitų panašių darbų [1], [2], duoda patarimus, kokius duomenis galima įrašyti į suformuotus elementus. Gavęs paieškos rezultatus, vartotojas gali toliau formuoti užklausą, jei gauti rezultatai jo netenkina. Toks būdas leidžia vartotojui suformuoti *SPARQL* užklausas su neribotu sąlygų skaičiumi ir gauti didelio tikslumo (angl. *precision*) ir gražinimo (angl. *recall*) rezultatus [3].

Gauti eksperimento rezultatai parodė, jog prototipas tikslumu lenkia kitas sistemas arba joms prilygsta atliekant objektų su nustatyta savybe ir konkrečių objektų paiešką naudojant abi eksperimente naudotas ontologijas ir, jog šį vartotojo sąsajos formavimo ir užklausių generavimo metodą galima taikyti portaluose, siekiant padidinti tikslumą ir gražinimą.

Šio darbo rezultatai buvo pristatyti XVII tarpuniversitetinėje magistrantų ir doktorantų konferencijoje ir išspausdintas straipsnis leidinyje „Informacinės technologijos 2012“

Darbo struktūra: antrame skyriuje aprašoma užklausių formavimo būdų analizė, trečiame – užklausių formavimo procesas ir algoritmas, ketvirtame – prototipo projektas bei realizacija, penktame – eksperimentas. Pabaigoje pateikiamos išvados ir naudotos literatūros sąrašas.

2. Semantinės paieškos sąsajos modelio ir jo taikymo semantiniam paieškos portalui analizė

2.1. Tyrimo objektas, tikslas ir uždaviniai

Šio darbo tyrimo objektas yra semantinių užklausų vartotojo sąsajoje formavimo ir vykdymo procesas sistemoje, glaudžiai susietoje su (angl. *tightly coupled*) dalykinės srities ontologija.

Sritis - OWL ir ja paremtų portalų paieškos sąsajos modelių taikymas semantinių portalų kūrime.

Semantinių portalų sąsajai nėra išbaigtų sprendimų atitinkančių vartotojų poreikius.

Semantinės paieškos tikslas yra išgauti teisingą informaciją remiantis konceptų prasmės atitikimu. Semantinė paieška nepakeis esamų interneto paieškos metodų, bet gali juos patobulinti. Šiuolaikiniai paieškos mechanizmai randa atsakymus į daugelį užklausų, tačiau jie remiasi raktinių žodžių paieška. Semantinė paieška siekia rasti tikslus atsakymus pagal semantinėse užklausoje pateiktą konceptų prasmę, kurią aprašo ontologijos. Šiuo metu populiariausia ontologijų aprašymo kalba yra *OWL* [1]. Semantinėms užklausoje naudojamos ontologijų užklausų kalbos, iš kurių labiausiai paplitusi *SPARQL*. Tačiau ši kalba yra tebevystoma, o semantinės paieškos sistemos yra pradinio kūrimo stadijoje. Semantinė paieška susideda iš dviejų dalių: naršymo (angl. *browsing*) ir užklausų (angl. *querying*) [2]. Naršymas yra pagrįstas prasminiais konceptų ryšiais, todėl vartotojas negali gauti atsakymų į laisvas semantines užklausas. Jis gali tik suformuoti kelią, vedantį link tam tikrų konceptų. Literatūroje yra įvairių pasiūlymų, kaip rasti kelius tarp konceptų ir siaurinti paieškos erdvę. Iš kitos pusės, egzistuoja nemažai informacinių sistemų su gerai išvystyta struktūrizuota paieška. Todėl šiame darbe numatoma sukurti tam tikrai dalykinei sričiai pritaiktą semantinės paieškos sistemą, kuri leistų vartotojui gauti tikslus atsakymus į tam tikrų tipų užklausas ir padėtų jam tas užklausas formuluoti. Išanalizavus esamų sistemų ir aprašytų metodų galimybes, tyrimo rezultate tikimasi sudaryti geresnę semantinės paieškos sąsajos modelį, kurio kūrimo principus būtų galima pritaikyti daugeliui sričių.

Darbo tikslas yra patobulinti semantinės paieškos galimybes, sukuriant semantinės paieškos sąsajos modelį, paremtą geriausiai semantinės ir struktūrizuotos paieškos sistemų pavyzdžiais.

Šiam tikslui pasiekti reikia išspręsti šiuos uždavinius:

1. Atlikti literatūros analizę ir ištirti esančius paieškos sąsajų modelius
 - 1.1. Išanalizuoti *SPARQL*, *OWL*, *RDF* ir susijusias sąvokas
 - 1.2. Išanalizuoti literatūroje pateiktus semantinės paieškos modelius
 - 1.3. Išanalizuoti esamų semantinių portalų architektūras ir paieškos modelius
 - 1.4. Išanalizuoti tokių sistemų kūrimo įrankius
2. Sudaryti semantinės paieškos sąsajos modelį ir suprojektuoti pasirinktos dalykinės srities informacinę sistemą su semantine paieška
3. Realizuoti semantinės paieškos sistemos prototipą
4. Atlikti eksperimentą ir įvertinti paieškos modelio privalumus ir trūkumus

Tyrimo metu bus atliekama analizė, kurios tikslas – susipažinti su esama situacija srityje, problemos sprendimo būdais; ištirti alternatyvius sprendimus, semantinės paieškos tinklapių kūrimo įrankius ir pasirinkti tinkamą sprendimo metodą, įrankius; sukurti paieškos modelį, pritaikytą pasirinktai dalykinei sričiai, kurią nusakys ontologija. Modelis negali tiktis visoms sritims, todėl atlikus literatūros ir alternatyvių sprendimų analizę reikia sudaryti bendrą paieškos modelio, jo komponentų vaizdą, kuris vėliau bus pritaikytas pasirinktai sričiai.

Analizei bus taikomi šie metodai: pagrindiniai tyrimo objekto analizėje naudojami metodai yra esančių sprendimų analizė pagal iš anksto pasirinktus kriterijus ir tinkamų tinklapių kūrimo įrankių pasirinkimas lyginant jų savybes. Taip pat analizuojama *OWL* ontologijų aprašymo kalba, jos savybės. Paieškos modelis kuriamas remiantis esančių sprendimų ir literatūros analize. Kadangi literatūroje yra pasiūlyta įvairių modelio įgyvendinimo variantų, kurie nėra realizuoti, tiriama ir tokių pasiūlymų įgyvendinimo galimybė.

2.2. Semantinių paieškos kalbų analizė

Literatūroje ir internete sutinkama keletas semantinės paieškos tipų:

- Raktažodžių paieška - naudojanti semantiką, kad patikslinti užklausą.
- Meta paieškos sistemos – kaip ir tradicinė raktažodžių paieška, naudoja semantiką užklausos tikslinimui, bet po to patikslintą užklausą perduoda

kitoms semantinės paieškos sistemoms ir gautus rezultatus pateikia vartotojui. Tokios sistemos yra laisvai susietos, nes rezultatai yra sunkiai priskiriami kokiai nors vienai ontologijai. Rezultatai gali būti pateikiami tokie kokie gaunami iš kitų sistemų arba apdoroti rikiuojant/grupuojant,

- Semantinė paieška ontologijose – iš vartotojo užklauso suformuoja SPARQL arba atitinkamos kalbos užklauso. Gautų rezultatų pateikimo būdas priklauso nuo sistemos kurioje įgyvendinta paieška.

Kadangi ontologijų taikymas suteikia didesnes semantinės paieškos galimybes, šiame darbe bus tiriama semantinės paieškos vartotojo sąsajos kūrimas glaudžiai su dalykinės srities ontologija susietai sistemai.

2.2.1. OWL kalbos analizė

OWL (*Web Ontology Language*) kalba palengvina mašininį turinio interpretavimą. Kuriant OWL buvo remtasi XML, RDF ar RDF-S kūrimo ir taikymo patirtimi, dėl to OWL turi platesnį žodyną, formalią semantiką ir geriau aprašo informaciją mašininiam interpretavimui.

OWL kalba suskirstyta i pokalbes: OWL Lite, OWL DL ir OWL Full, kurių kiekviena turi vis didesnę išraiškų laisvę.

OWL Lite yra naudojama klasifikavimo hierarchijoms ir paprastiems apribojimams. Pavyzdžiui, nors ji palaiko kardinalumo apribojimus, bet suteikia jiems tik 0 arba 1 reikšmę. Dėl to šiai kalbai yra lengviau kurti įrankius nei kitoms, ir ji turi mažesnę formalų sudėtingumą.

OWL DL suteikia maksimalų išraiškingumą ir užtikrina skaičiavimų baigtumą (užtikrinama, kad visos išvados yra garantuotai suskaičiuojamos) ir sprendimo baigtumą (visi skaičiavimai bus atlikti per baigtinį laiką). OWL DL gali būti naudojami visi OWL kalbos dariniai, bet jie turi apribojimus (pavyzdžiui, nors klasė gali būti kelių klasių poklasė, bet klasė negali būti kitos klasės atskiras atvejis). OWL DL pavadinimas išplaukia iš jo suderinamumo su aprašomąja logika (angl. *description logics*) – tyrinėjimų srities, kuri studijavo logiką nuo pat formalaus OWL įkūrimo.

OWL Full suteikia maksimalų išraiškingumą, bet nesuteikia garantijų dėl skaičiavimų baigtumo. Deja, panašu, kad jokia ryšių išvedimo (angl. *reasoning*) programinė įranga galėtų palaikyti pilną ryšių išvedimą kiekvienai OWL Full savybei. Kiekvienas sudėtingesnis OWL kalbos tipas yra paprastesnio tipo praplėtimas, todėl paprastesnio tipo kalboje aprašyta ontologija galioja sudėtingesnėje. Šiuo metu OWL

Full realizacijų nėra, dėl to palaikymas ir aiškinimas yra menkai išplėtotas palyginus su OWL DL.

OWL Lite naudoja tik kai kurias OWL kalbos galimybes. Joje yra daugelis kalbos darinių, bet jie turi tam tikrų apribojimų.

- OWL Lite iš RDF schemos paveldėtos savybės:

Class – tai yra grupė individų, kurie yra kartu, nes turi kokią nors bendrą savybę;

rdfs:subClassOf – nurodo hierarchinį ryšį, kuris reiškia vienos klasės buvimą kitos poklase;

rdf:Property – savybė nurodo ryšius tarp individų arba jų duomenų reikšmių;

rdfs:subPropertyOf – nurodo hierarchinį ryšį, kuris reiškia vienos savybės buvimą kitos posavybe;

rdfs:domain – nurodo ribojimą, kuriems individams savybė gali būti taikoma;

rdfs:range – nurodo ribojimą, kuriuos individus savybė gali turėti kaip reikšmę;

Individual – individai yra klasių objektai;

- Lygybės:

equivalentClass – ekvivalenčios klasės, kurios turi tuos pačius objektus;

equivalentProperty – ekvivalenčios savybės, kurios tuos pačius objektus susieja tarpusavyje;

sameAs – nurodo, kad du objektai yra vienodi;

differentFrom – nurodo, kad objektas skiriasi nuo kito;

AllDifferent – nurodo kad visi objektai yra skirtingi;

- Savybių charakteristikos:

inverseOf – viena savybė yra deklaruojama kaip priešinga kitai;

TransitiveProperty – nurodo, kad savybės yra tranzityvios;

SymmetricProperty – nurodo, kad savybės yra simetriškos;

FunctionalProperty – nurodo, kad savybė gali turėti tik vieną unikalią reikšmę kiekvienam objektui;

InverseFunctionalProperty – nurodo priešingą funkcinę savybę;

- Savybių ribojimai:

allValuesFrom – nurodo, kad savybės turi visas klasės reikšmes;

someValuesFrom – nurodo, kad savybės apima dalį klasės reikšmių;

- Ribotas kardinalumas:

minCardinality – nurodo kiek mažiausiai kiekvienas objektas turės susietų objektų;

maxCardinality – nurodo kiek daugiausiai kiekvienas objektas turės susietų objektų;

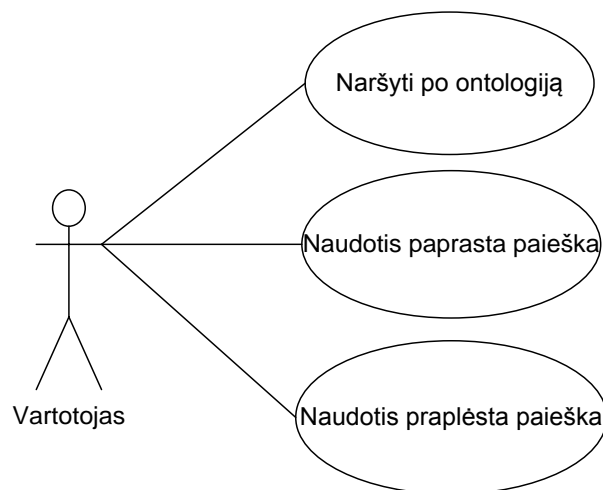
cardinality – nurodomas, kai ir *minCardinality*, ir *maxCardinality* reikšmės yra tokios pat;

2.3. Semantinės paieškos vartotojų poreikių analizė

2.3.1. Paieškos vartotojų tikslai ir problemos

Vartotojas gali būti bet kas, kam reikia surasti informaciją esančią portale ar svetainėje. Išskiriami du vartotojų tipai: vartotojai, neturintys žinių apie semantinę paiešką ir portalo ontologiją, ir vartotojai, turintys patirties semantinėje paieškoje. Pirmieji būtų linkę naudotis paprasta paieška su vienu įvedimo lauku, antrieji naudotų praplėstą paiešką.

Vartotojų tikslai yra gauti teisingą atsakymą į duotą užklausą arba tiesiog naršyti po ontologiją, tyrinėjant objektus, jų ryšius ir savybes. Paprasta paieška turi tik vieną lauką įvesti užklausai ir paieškos modelis sukuria SPARQL užklausą ontologijai pagal nustatytą to lauko prasmę. Praplėsta paieška leidžia vartotojui nurodyti užklausos parametrus, taip išvengiant galimų netikslumų.



2.1. pav. Sistemos panaudojimo atvejai

Šiuolaikinių semantinės paieškos sistemų problema yra netobulas paieškos modelis. Jo sąsaja gali reikalauti per didelio kiekio nustatymų iš vartotojo pusės. Modelis gali gražinti neteisingą atsakymą į užklausą, per daug apibendrinamas

paieškos parametrus arba negražinti jokio atsakymo, jei nėra numatytų parametru sinonimų.

Kad paieškos sistema geriau suprastų, ko vartotojas ieško, reikia pašalinti užklauso žodžių dviprasmiškumą. Pašalinti dviprasmiškumą galima pasirenkant labiausiai tikėtiną žodžio prasmę. Dažniausiai labiausiai tikėtina prasmė būna ta, kuri taikoma ontologijoje, kurioje vykdoma paieška. Jeigu paieška vykdoma keliose ontologijose, naudojančiose tą patį terminą skirtingomis prasmėmis, paieškos sistema turėtų leisti vartotojui pasirinkti, su kuria prasme susiję rezultatai turėtų būti pateikiami.

2.3.2. Paieškos vartotojo sąsaja

Paieškos sąsajai išskelti trys pradiniai reikalavimai, ji turi:

- būti intuityvi
- suprantamai pateikti paieškos rezultatus
- padėti vartotojui suformuoti užklausa

Intuityvumo reikalavimas esamuose sprendimuose išpildomas mažinant aktyvių vartotojo sąsajos komponentų kiekį. Kuo mažiau ekrane pasirinkimo ir pildymo laukų, tuo lengviau vartotojui suprasti, kaip naudotis sistema. Taip pat intuityvumą padidina skirtingomis spalvomis nuspalvinti skirtingi sąsajos elementai kaip Semedico [8] arba Epicurious [10] svetainėse. Vartotojas iš karto gali atskirti paieškos lauką nuo patarimų ar rezultatų laukų.

Esamos realizacijos paieškos rezultatus pateikia lentelėse. Įprastai kiekvieną rezultatą atitinka rasto dokumento pavadinimas užrašytas didesniu šriftu, bei po juo esanti smulkesniu šriftu pateikta trumpa jo ištrauka ir nuoroda į dokumentą. Jei paieška gražina tik vieną dokumentą, jis iš karto pateikiamas vartotojui. Toks paieškos rezultatų atvaizdavimas yra aiškus ir patogus, nes yra panašus į raktažodžių paieškos sistemų rezultatų atvaizdavimą.

Užklausiai tikslinti literatūroje yra siūloma vartotojo sąsaja leidžianti vartotojui sudėlioti užklausa iš blokų, atitinkančių konceptus ir savybes, ir leidžiant nurodyti jų ryšius [2]. Pagal gautą darinį sukuriama SPARQL užklausa ir ontologijoje ieškomi panašūs dariniai. Tokia sąsaja labai tinka lietimui jautrius ekranus naudojančiose sistemose. Yra sukurta tokios vartotojo sąsajos prototipų [31], bet jie nėra panaudoti semantinės paieškos tinklapiuose.

Kitas variantas yra dažnai sutinkamas semantinės interaktyviose paieškos portalų realizacijose. Vartotojui pateikiama forma, kurios laukuose jis gali nurodyti užklausą tikslinančius parametrus arba paieškos vykdymo metu vartotojo prašoma patikslinti užklausą. Tai padeda susiaurinti paieškos sritį ir pašalinti dviprasmiškumą. Dauguma semantinės paieškos portalų naudoja tokias formas kaip išplėstinę paiešką. Toks interaktyvumas reikalauja iš vartotojo daugiau veiksmų, bet leidžia gauti tikslesnius rezultatus.

Panašus variantas yra pavyzdžių pateikimas šalia užklaustos įvedimo lauko ar laukų, kad vartotojas galėtų pats tiksliau parašyti savo užklausą, kaip Semedico [8] ir Exalead [13] portaluose.

Navigacija yra reikalinga kai vartotojas pageidauja peržiūrėti ontologiją nenaudodamas paieškos. Ontologiją galima atvaizduoti kaip sąrašą, medį (jei ontologija nėra sudėtinga) arba grafą. Europeana [9] – Europos kultūros portalas ontologijos turinį atvaizduoja kaip laiko juostą, kurioje pagal datą išdėlioti ontologijoje esantys įrašai.

Literatūroje rekomenduojama, kad paieškos modelis naudotų suprogramuoto (angl. *hard coded*) ir savaimė besimokančio vartotojo konteksto nustatymo metodų mišinį [3]. Suprogramuotas vartotojo konteksto nustatymas leidžia vartotojui nurodyti kokia sritimi jis domisi, taip panaikinant dviprasmiškumą ir negražinant vartotojui nereikalingų rezultatų. Pvz. paieška su žodžiu „plazma“ medicinos srityje negražins atsakymų apie fizikinį terminą. Tokia funkcija nėra būtina siauros srities semantinės paieškos sistemose. Savaimė besimokantis konteksto nustatymas fiksuoja ankstesnes vartotojo užklausas ir nustato kokioje srityje vartotojas ieško informacijos. Tam reikėtų, kad vartotojas sistemoje susikurtų paskyrą ir prieš naudodamasis paieška, prisiregistruotų.

2.4. Esamų semantinės paieškos portalų analizė

2.4.1. Paieškos portalų savybės

Sekančioje lentelėje parodomi semantinės paieškos portalų privalumai ir trūkumai vertinant pagal tai ar jie naudoja ontologijas, ar galima naršyti pateikiamoje informacijoje, ar yra galimybė tikslinti užklausą naudojant paiešką pagal aspektus ir ar paieškos portalai suskirsto rezultatus į grupes pagal randamos informacijos prasmę.

Paskutinėje skiltyje nurodoma ar paieškos portalo informacija glaudžiai susieta su naudojama ontologija.

Lentelė Nr.2.1. Paieškos portalų palyginimas

Paieškos portalas	Taiko ontologijas	Naršymas	Aspektinė paieška	Grupuojami paieška	Sistema glaudžiai susieta
Hakia [6]	+-	-	-	-	-
Lt-world [7]	+	+	+	-	+
Semedico [8]	+	+	+	+	+
Europeana [9]	+	+	+	-	+
Epicurious [10]	+	+	+	-	+
Kosmix [11]	+-	+	-	+	-
Dmoz [12]	+	+	+	-	+
Exalead [13]	-	-	-	-	-
SenseBot [14]	-	-	-	-	-
Cognition [15]	+	-	-	-	-
Swoogle [16]	+	-	-	-	-
Bing [17]	-	-	+	-	-

- Hakia [6] – semantinės paieškos tinklapis, bandantis suprasti vartotojo užklauso prasmę, atliekantis rezultatų analizę ir atvaizduojantis rezultatus išrikiuotus pagal užklauso ir rezultato prasmės atitikimą.
- Lt-world [7] – ontologijomis paremtas tinklapis, turintis paiešką pagal kategorijas.
- Semedico [8] – ontologijas taikantis puslapis ieškantis informacijos medicinos srityje. Turi daug straipsnių, ieško geriausiai užklauso prasmę atitinkančių straipsnių, rikiuoja juos pagal atitikimo tikslumą. Rezultatai grupuojami į sritis kuriuose galima naršyti.
- Europeana [9] – plačioje ontologijoje saugo įrašus apie paveikslus, muziką, filmus ir knygas iš Europos galerijų, bibliotekų, archyvų ir muziejų. Galima kurti užklausą nurodant jos aspektus. Rezultatai suskirstomi į penkias grupes (tekstas, vaizdai, video medžiaga, garsai ir viskas kartu).

- Epicurious [10] – didelė kulinarinė biblioteka, taikanti ontologijas, sinonimus, užklausa galima formuoti aspektų pagalba. Galima naršyti po ontologiją ir užklausų rezultatus.
- Kosmix [11] – paieškos portalas su daliniu ontologijų taikymu. Pateikia didelį informacijos kiekį iš įvairių šaltinių ir suskirsto juos į kategorijas. Kadangi paieškos rezultatai analizuojami, kategorijų pavadinimai skirtingose paieškose gali skirtis.
- Dmoz [12] open directory project – ontologijas taikantis paieškos tinklapis, kuriame patys vartotojai gali tobulinti esančią ontologiją. Labai daug klasių, galima naršyti arba duoti bendrą užklausa ir naršyti tarp pateiktų rezultatų.
- Exalead [13] – tinklapis taikantis semantiką tikslesniems atsakymams rasti. Paiešką galima atlikti tarp vaizdų, vaizdo medžiagos, wikipedijoje arba visame tinkle.
- SenseBot [14] – naudoja semantiką išgauti prasmei iš raktažodžiais pagrįstų užklausų rezultatų. Vartotojams pateikiamas gana rišlus atsakymas, sudarytas iš paieškos rezultatų iškarpu ir nuorodų į tų rezultatų dokumentus.
- Cognition Search [15] – paieškos tinklapis apimantis teisės, sveikatos, biblijos sritis bei atliekantis paiešką wikipedijoje. Cognition turi didžiausią anglų kalbos semantinį žemėlapi, kurtą 24 metus. Tai padeda netik apdoroti užklausas ir rezultatus, bet taip pat gali padėti kitose srityse, kur reikia išgauti žinias iš informacijos.
- Swoogle [16] – paieška skirta semantiniam tinklui (angl. *semantic web*). Indeksuoja dokumentus kurtus remiantis semantikos standartais tokiais kaip RDF ir atlieka jų paiešką.
- Bing [17] – naudoja semantiką tikslesniems rezultatams gauti. Galima patikslinti užklausa pridėdant papildomų parametrų.

Dauguma tinklalapių naudoja semantinę paiešką, kad praplėstų raktažodžiais grįstą paiešką. Ontologijų nenaudojantys portalai (Exalead, Bing) gali pasiūlyti paiešką pagal teisingą raktažodį, jei šis buvo parašytas su klaidomis, ieškoti panašiai skambančių arba panašiai rašomų žodžių, bet neieško raktažodžio sinonimų. Keletas tinklalapių (Europeana, Semedico) paiešką atlieka savo žinių bazėse. Srityse besispecializuojantys tinklapiai gana gerai atsako į užduotus klausimus, tuo tarpu bendrai paiešką tinkle atliekantys tinklapiai atsakymus duoda ne tokius tikslius.

Galimybė įvesti paieškos aspektus gražina tikslesnius rezultatus, o naršymas leidžia surasti tikslų atsakymą į klausimą, jei jis numatytas tinklapio ontologijoje.

Kadangi šis darbas orientuotas į glaudžiai susietas sistemas, reikia plačiau iširti šių sistemų realizacijas.

2.4.2. Detali esamų paieškos modelių analizė

Iš anksčiau paminėtų paieškos portalų, penkių portalų dokumentai glaudžiai susieti su ontologijomis. Juos tiriant detaliau reikia iširti jų grafinės sąsajos sudėtį ir, jei įmanoma, paieškos variklio architektūrą.

Tam panaudojamos IKS – Interactive Knowledge Stack projekto tinklapyje pateiktos turinio valdymo sistemų (angl. *CMS*) semantinių savybių vertinimo lentelės [29]. Lentelėse sistemos nagrinėjamos keturiomis fazėmis: užklausos sudarymo, paieškos algoritmo, rezultatų pateikimo ir sąveikos su rezultatais bei užklausų tikslinimo. Sekančiuose poskyriuose palyginamos šios pasirinktų tinklapių fazės.

2.4.2.1. Užklausos sudarymas

Pirma nagrinėjama fazė – užklausos sudarymas. Lentelėje Nr. 2.2. vaizduojamos portalų užklausų sudarymo galimybės, kokiomis funkcijomis jos įgyvendinamos ir kokie vartotojo sąsajos komponentai yra tam naudojami.

Lentelė Nr.2.2. Paieškos portalų užklausos sudarymo funkcijų ir sąsajos komponentų tyrimas.

Galimybė	Portalas	Funkcijos	Sąsajos komponentai
Teksto įvedimas	Lt-world	Raktažodžių paieška	Vienas įvedimo laukas, Paieškos forma
	Europeana		
	Dmoz		
	Semedico		Vienas įvedimo laukas
	Epicurious		
Operatoriai	Lt-world	Loginiai operatoriai	Nuo realizacijos priklausoma sintaksė
	Europeana		
	Semedico	-	
	Epicurious	Loginiai operatoriai, Reguliaros išraiškos (angl. <i>regexp</i>)	
	Dmoz	Loginiai operatoriai, Reguliaros išraiškos (angl. <i>regexp</i>),	

		Specialios paskirties operatoriai	
Kontroliuojamos sąlygos (angl. <i>controlled terms</i>)	Lt-world	Užklauso tikslinimas,	Norimos reikšmės (angl. <i>value list</i>),
	Epicurious	Rezultatų pateikimo ribojimas	Grupotas naršymas
	Europeana	Užklauso tikslinimas, Iš anksto sudarytos užklauso	Norimos reikšmės (angl. <i>value list</i>), Populiarios užklauso
	Semedico	-	-
	Dmoz	Rezultatų pateikimo ribojimas	Grupotas naršymas
Reakcija į vartotojo veiksmus	Lt-world	-	-
	Epicurious		
	Dmoz		
	Semedico	Užklauso tikslinimas prieš ieškant	Patarimų sąrašas,
	Europeana		Semantinis auto-užbaigimas (angl. <i>auto completion</i>)

Visi tirti portalai naudoja paiešką pagal raktažodžius, įvedamus viename lauke arba išplėstinėje paieškoje-formoje. Lt-world, Europeana ir Dmoz išplėstinei paieškai naudoja formas. Semedico ir Epicurious turi vieną užklauso įvedimo lauką ir formų nenaudoja.

Epicurious formos nebuvimą kompensuoja loginiais operatoriais ir reguliariųjų išraiškų naudojimu, kuris, kaip ir Dmoz portale, atranda panašius atsakymus net jei sutampa tik raktažodžio dalis. Dmoz vienintelis naudoja specialios paskirties operatorius, tokius kaip kabutės - frazių arba „*“ - bet kokio simbolio paieškai.

Portaluose užklauso tikslinamos nurodant norimas reikšmes formose. Rezultatų pateikimas ribojamas naršant grupėse, taip pat Lt-world ir Dmoz naršant pateiktos užklauso turi galimybę atlikti paiešką tik esamoje grupėje. Europeana portale lankytojams pateikiamos populiariausios užklauso, kas yra gana patogiu kai daug vartotojų ieško tos pačios informacijos.

Semedico, nors neturi daugumos anksčiau minėtų funkcijų, turi išsamią užklauso formavimo pagalbą, kuri užklauso rašymo metu vartotojui pateikia patarimų/auto-užbaigimo sąrašą. Europeana taip pat naudoja tokią funkciją, bet patarimų sąrašas sudaromas tik tuo metu rašomam žodžiui, kai Semedico suranda ir pateikia ontologiškai susijusius terminus.

2.4.2.2. Paieškos algoritmas

Antroji tyrimo fazė - portalų naudojami paieškos algoritmai. Kadangi portalai nepateikia išsamios informacijos apie sandarą ir veikimą, apie naudojamus algoritmus sprendžiama tik iš portalų elgsenos.

Lentelė Nr.2.3. Paieškos portalų paieškos algoritmų tyrimas.

Galimybė	Portalas	Funkcijos	Sąsajos komponentai
Sintaksės palyginimas	Lt-world	Tikslus, žodžio pradžios, žodžio dalies lyginimas	Nenaudojami
	Semedico		
	Europeana		
	Epicurious		
	Dmoz		
Semantinis palyginimas	Lt-world	-	Nenaudojami
	Dmoz		
	Semedico	Praplėtimas sinonimais,	
	Europeana	Praplėtimas grafu (angl. <i>graph traversal</i>),	
	Epicurious	RDFS/OWL rezultatų išvedimas (angl. <i>reasoning</i>)	

Sintaksės palyginime visi portalai naudojo tik tikslų žodžių lyginimą, neatsižvelgiant į jų dalis. Užklausų šakojimas (angl. *stemming*) ir minimalus redagavimo atstumas (angl. *minimal edit distance*) paieškoje nenaudojami, bet pastebėti patarimų sąrašo sudaryme.

Tiriant semantinį lyginimą Lt-world ir Dmoz gražino rezultatus tik su užklausoje užduotais raktažodžiais. Semedico, Europeana ir Epicurious portalai gražino susijusius su užklausa rezultatus, kai kurie iš jų neturėjo paieškoje nurodytų žodžių. Pastarieji portalai semantiškai praplečia paieškos sritį, bet negalima tiksliai nustatyti kokie algoritmai naudojami, todėl skiltyje „Funkcijos“ paliekami originalioje palyginimų lentelėje nurodyti algoritmai.

2.4.2.3. Rezultatų pateikimas

Trečioji fazė – paieškos rezultatų pateikimas. Sekančioje lentelėje parodoma kokias funkcijas portalai palaiko išrenkant duomenis, juos rikiuojant, išdėstant ir kaip vartotojas gali sąveikauti su rezultatais. Taip pat nurodomi vartotojo sąsajos komponentai, kurių pagalba suteikiamos šios galimybės.

Lentelė Nr.2.4. paieškos portalų rezultatų pateikimo tyrimas.

Galimybė	Portalas	Funkcijos	Sąsajos komponentai
Duomenų išrinkimas (angl. <i>data selection</i>)	Lt-world	Klasėmis aprašyti šablonai (angl. <i>class specific template</i>)	Tekstas
	Semedico		
	Epicurious		Tekstas, Laiko linija
	Dmoz		
	Europeana		
Rikiavimas	Lt-world	Turiniu ir ryšiais paremtas rikiavimas	Rikiuotas sąrašas
	Semedico		
	Europeana		
	Epicurious		
	Dmoz	-	-
Išdėstymas	Lt-world	Dinamiškas grupavimas	Medis
	Semedico	Grupavimas pagal savybes arba kelias	Sudėtinių langelių struktūra (angl. <i>Nested box structure</i>)
	Europeana	-	-
	Epicurious	-	-
	Dmoz	-	-
Reakcija į vartotojo veiksmus	Lt-world	-	-
	Dmoz		
	Semedico	Užklausos tikslinimas (angl. <i>query refinement</i>), Rezultatų dviprasmiškumo šalinimas (angl. <i>post-query disambiguation</i>)	Grupavimo laukai (angl. <i>facets</i>)
	Europeana		
	Epicurious		

Visi tirti portalai išrinktus duomenis atvaizduoja šablonais, kuriuose nurodomas dokumento ar resurso pavadinimas, adresas ir data ar trumpas aprašymas. Rezultatai atvaizduojami tekstu, nenaudojant grafų, žemėlapių, kalendorių ar debesies atvaizdavimo (angl. *tag cloud*). Europeana be tekstinio rezultatų atvaizdavimo taip pat atvaizduoja rezultatus laiko juostoje.

Lt-world, Semedico, Europeana ir Epicurious rezultatus rikiuoja, pirmiausia pateikiant labiausiai užklausą atitinkančius rezultatus. Dmoz rezultatų rikiavimas nepastebėtas.

Grupuojant Lt-world sudaro dviejų lygių medį žemesniame lygyje pateikiant dokumentų informaciją ir nuorodas, o aukštesniame apibendrinančias antraštes.

Semedico rezultatus pateikia sąrašu ir šone pateikia gautus rezultatus suskirstytus į iš anksto parinktas grupes. Europeana, Epicurious ir Dmoz rezultatus pateikia paprastu sąrašu be grupavimo.

Tirtuose portaluose su ieškoma tema susijusių resursų rekomendavimas nebuvo pastebėtas. Semedico, Europeana ir Epicurious portalai pateikus netikslią arba daug rezultatų gražinusią užklausą siūlo patobulinti užklausą, renkantis norimą grupavimą.

Lt-world ir Dmoz portaluose norint patikslinti užklausą tenka grįžti į pradinį paieškos langą ir rašyti užklausą iš naujo.

2.4.2.4. Sąveika su rezultatais ir užklausų tikslinimas

Ketvirta fazė yra vartotojo sąveika su užklausos rezultatais ir užklausos tobulinimas. Čia tiriama kaip portalai valdo rezultatų pateikimą ir užklausų tikslumą.

Lentelė Nr.2.5. Paieškos portalų sąveikos su rezultatais ir užklausomis tyrimas.

Galimybė	Portalas	Funkcijos	Sąsajos komponentai
Vartotojo modelis	Lt-world	-	-
	Semedico		
	Epicurious		
	Dmoz		
	Europeana	virtotojo profilis/personalizacija (angl. <i>user profiling/personalization</i>)	
Prisitaikantis sistemos turinio atvaizdavimo modelis	Lt-world	-	-
	Semedico		
	Europeana		
	Epicurious		
	Dmoz		
Užklausos teisingumo palaikymas	Lt-world	Paieškos ir užklausos algoritmų valdymas	Užklausos parametrų valdymo skydas (angl. <i>query parameter control panel</i>)
	Europeana		
	Dmoz		
	Semedico	-	
	Epicurious		

Iš tirtų portalų tik Europeana portale užsiregistravus galima nurodyti savo interesus į kuriuos vėliau portalas atsižvelgia pateikiant rezultatus ir naršant. Kituose portaluose vartotojo modeliai nenaudojami. Epicurious portale galima užsiregistruoti, bet tai niekaip neįtakoja rezultatų tikslumo.

Kadangi visi tirti portalai turi tik dokumentų tipo turinį, sistemos turinio atvaizdavimo modelis nėra prisitaikantis ir neturi nuo turinio priklausomo atvaizdavimo (angl. *content type dependent representation*) funkcijos.

Lt-world, Europeana ir Dmoz leidžia valdyti paieškos algoritmus nurodant paieškos sritį ir norimų rezultatų parametrų reikšmes. Nei vienas iš tirtų portalų nenaudoja paieškos paradigmu, tokių kaip paieška pagal pateiktą XML kodą ar pagal suformuotą grafo dalies modelį.

2.4.2.5. Semantinės paieškos kokybės vertinimo kriterijai

Semantinės paieškos efektyvumui vertinti literatūroje randami tikslumo (angl. *precision*) ir grąžinimo (angl. *recall*) parametrai [3]. Idealyje paieškos sistemoje šie parametrai būtų lygūs vienetui.

$$\text{grąžinimas} = \frac{\text{rastų tinkamų dokumentų kiekis}}{\text{tinkamų dokumentų kiekis}}$$

$$\text{tikslumas} = \frac{\text{rastų tinkamų dokumentų kiekis}}{\text{rastų dokumentų kiekis}}$$

Grąžinimui padidinti vartotojo užklauskos sąvokos pakeičiamos bendresnėmis ontologijoje numatytomis sąvokomis. Tai tuo pačiu sumažina tikslumą, nes bendresnės sąvokos apima daugiau dokumentų, o didėjant grąžinamų dokumentų skaičiui daugėja ir dokumentų kurie neturi vartotojui reikalingos informacijos.

Analogiškai, kad padidinti tikslumą, užklauskos sąvokos keičiamos tikslesnėmis, bet tikslinant sąvokas yra mažinamas grąžinimas ir gali būti praleisti vartotojui reikalingi dokumentai.

Šiai problemai spręsti reiktų leisti vartotojui pasirinkti kokio tikslumo paiešką jis norėtų atlikti. Taip vartotojai, kurie žino ko ieško galėtų greičiau rasti dokumentus, pasirinkę tikslią paiešką. Vartotojai kurie ieško bendresnės informacijos galėtų pasirinkti platesnę paiešką ir rinktis iš grąžintų rezultatų.

2.5. Semantinės paieškos tinklalapių realizavimo įrankių analizė

Tinklapio su semantine paieška kūrimas reikalaus ontologijos kūrimo ir modifikavimo įrankio, programavimo aplinkos (kalbos) ir ontologijų manipuliavimo bibliotekų jai. Sukurtas ontologijas taip pat galima saugoti nuosavame serveryje arba panaudojus vieną galimų saugyklų. Galimi įrankiai detaliau išanalizuoti šio skyriaus poskyriuose.

2.5.1. Programavimo kalbos, karkasai ir ontologijų manipuliavimo bibliotekos

Karkasai (angl. *frameworks*) ir bibliotekos suderina programavimo kalbas tokias kaip PHP, Java, .NET su semantinio tinklo informaciją aprašančiomis kalbomis OWL, RDF, RDF-S. Sekančioje lentelėje lyginami karkasai, jų palaikomos kalbos ir papildomos savybės.

Lentelė Nr.2.6. Karkasų palyginimas.

Karkasai	Programavimo kalba	Papildomos savybės
Trice	PHP	Palaiko SPARQL, RDF. Reikalauja PHP ir MySQL serverių. Dar nėra dokumentacijos
Jena	Java	Į paketą įeina duomenų saugykla ir SPARQL užklausų sluoksnis
EasyRDF	PHP	Sukuria RDF grafą iš PHP objektų, po kurį galima naršyti ir paimti informaciją
Sesame	Java	Į paketą įeina duomenų saugykla, turi nuosavą užklausų kalbą SeRQL
SemWeb	.NET, C#	Į paketą įeina duomenų saugykla

Java kalba suteikia daugiau funkcionalumo, nes turi daugiau įrankių nei PHP. .NET ir C# turi mažai įrankių ir labiau tiktų savarankiškomis programoms nei tinklapiui. Kadangi yra kuriamas tinklapis, pasirinktos Javascript kalba tinklapio kūrimui ir informacijos atvaizdavimui bei Java programavimo kalba, dėl didelio funkcionalumo dirbant su SPARQL užklausomis ir ontologijomis.

2.5.2. Ontologijų saugyklos

Semantinio portalo kūrimo įrankių yra labai daug, todėl palyginimui paimta tik keletas įrankių iš www.w3.org [18] puslapyje rekomenduojamų įrankių sąrašo. Renkantis pagrindinis dėmesys buvo skirtas įrankiams palaikantiems Java, PHP, C# kalbas.

Sukurtas ontologijas galima saugoti tinkle arba vietiniame kompiuteryje, instaliavus reikiamą programinę įrangą. Sekančioje lentelėje palyginama keletas teikiamų tinklo paslaugų ir instaliuojamų paketų.

Lentelė Nr. 2.7. Duomenų saugyklų palyginimas.

Duomenų saugyklos	Palaikomos kalbos	Naudojimas
Sesame	Java, Python, PHP	Instaliuojama
TalisPlatform	Prieinama per HTTP	Tinklo paslauga
AllegroGraph	Java, LISP, Python, Prolog	Tinklo paslauga
Knoodl	Prieinama per HTTP	Tinko paslauga
Joseki	Priėjimas per SPARQL prieigos tašką	Instaliuojama
SemWeb	.NET, C#	Instaliuojama
SDB	Priėjimas per SPARQL prieigos tašką	Instaliuojama
TDB	Priėjimas per SPARQL prieigos tašką	Instaliuojama

Visos pasirinktos saugyklos ir saugojimo paslaugos yra nemokamos tik norint naudotis kai kuriomis saugyklų paslaugomis, kaip Knoodl, reikia prisiregistruoti paslaugos tiekėjo tinklapyje.

Paslaugos patogios tuo, kad susikurtą ontologiją galima pasiekti iš įvairių kompiuterių per HTTP nepriklausomai nuo programavimo kalbos, svarbu kad kalba turėtų funkcionalumą HTTP paketų siuntimui, bet tam būtina turėti nuolatinį interneto ryšį.

Kompiuteryje instaliuojamose saugyklose ontologijas galima pasiekti bet kada, nepriklausomai nuo interneto ryšio. Be to vietiniame kompiuteryje galima greičiau ir paprasčiau atlikti pakeitimus ontologijoje, dėl to darbo pradžioje šis variantas yra priimtinesnis.

SDB, Joseki ir TDB yra Jena karkaso komponentai skirti ontologijų saugojimui ir užklausoms per SPARQL prieigos tašką (angl. *SPARQL endpoint*). TDB veikia remiantis vien Java kalba. Joseki serveris gali veikti tiek lokaliame, tiek nutolusiame kompiuteryje, konfigūracijai naudojama Turtle kalba ir ši saugykla šiuo

metu yra standartinė Jena karkasui. SDB veikimui naudoja SQL duomenų bazes tokias kaip MySQL, PostgreSQL Oracle, Microsoft SQL Server ar Apache Derby.

Kadangi buvo pasirinktas Jena karkasas, ontologijai saugoti pasirinktas Joseki serveris.

2.5.3. Ontologijų kūrimo ir modifikavimo įrankiai

Ontologijų kūrimo įrankiai reikalingi aprašyti ir modifikuoti tinklapio žinioms apie dalykinės srities objektus, jų savybes ir santykius.

Lentelė Nr.2.8. Ontologijų kūrimo įrankių palyginimas.

Ontologijų kūrimo įrankiai	Palaikomos technologijos	Kaina	Dokumentacijos kiekis
Model Futures OWL Editor	OWL	Nemokama	Mažas
OntoStudio	OWL, RDF, RDF-S	Mokama	Mažas
Protégé	OWL, RDF, RDF-S	Nemokama	Didelis
Knoodl	OWL, RDF, RDF-S	Nemokama	Vidutinis

Model Futures OWL Editor įrankis yra nemokamas, bet yra skirtas didelėms ontologijoms ir palaiko tik OWL redagavimą. OntoStudio palaiko daugiau semantinio tinklo technologijų, bet yra mokamas. Protégé ir Knoodl palaiko OWL, RDF ir RDF-S technologijas ir yra nemokami. Protégé turi išsamią dokumentaciją, daug mokomosios medžiagos ir patogią grafinę sąsają. Knoodl nėra instaliuojamas, kaip Protégé, visi veiksmai vyksta per vartotojo naršyklę.

Dėl didelio mokomosios medžiagos kiekio ir išsamios dokumentacijos ontologijai kurti ir modifikuoti pasirinktas Protégé įrankis.

2.6. Siekiamas sprendimas

Siekiamas sprendimas – semantinės paieškos vartotojo sąsaja, sukurta naudojant Javascript, HTML, Java kalbas, vykdanči paiešką nuosavoje OWL ontologijoje, kuri yra prieinama per talpyklos SPARQL prieigos tašką (SPARQL endpoint).

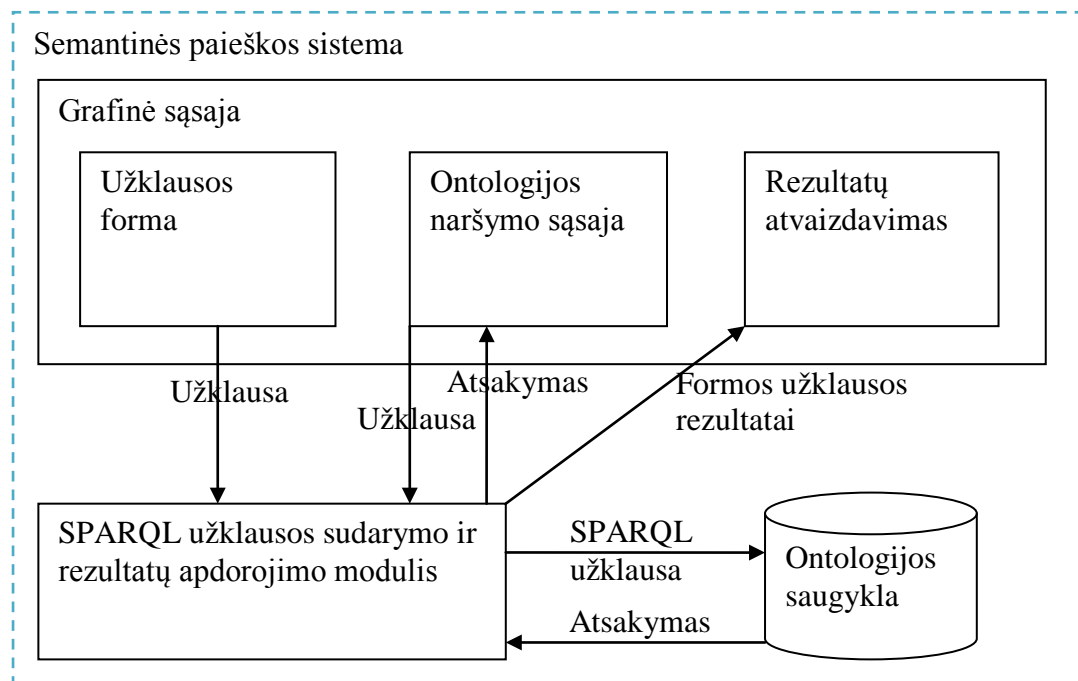
Užklauso sudarymui pasirinktas formos pildymo metodas, bet siekiant sumažinti vartotojui pateikiamų elementų kiekį, forma turi būti kuriama dinamiškai, vartotojui pasirenkant kiek elementų jis pageidauja.

Atlikus detalią paieškos modelių analizę pasirinktos pagrindinės siekiamos sistemos funkcijos ir tam reikalingi vartotojo sąsajos elementai.

Lentelė Nr.29. Siekiamo rezultato funkcijos ir vartotojo sąsajos komponentai.

Funkcija	Sąsajos komponentas
Teksto įvedimas	Paieškos laukai
Užklausos tikslinimas	Paieškos laukai
Rezultatų pateikimo ribojimas	Paieškos laukai
Praplėtimas sinonimais	-
Reakcija į vartotojo veiksmus	Norimos reikšmės (angl. value list),
Turiniu ir ryšiais paremtas rikiavimas	Rikiuotas sąrašas
Paieškos ir užklausos algoritmų valdymas	Papildomų komponentų pridėjimo mygtukai

Pradinė numatomos sistemos architektūra parodyta pav. 2.2.



2.2. pav. Siekiama semantinės paieškos sistemos sudėtis

Rezultatai bus atvaizduojami naudojant HTML lenteles, suformuojant jas Java applet'e ir pateikiant vartotojui per Javascript.

Ontologijų manipuliavimui Java kalboje bus naudojamas Jena karkasas. Ontologija bus saugoma Joseki serveryje, kuris yra pateikiamas kartu su Jena karkasu.

2.7. Analizės išvados

1. Atlikta semantinės paieškos portalų analizė parodė, kad portalai naudoja didelį vartotojo sąsajos elementų kiekį tikslumui palaikyti, o portalai naudojantys mažai vartotojo sąsajos elementų grąžina netikslius rezultatus, todėl buvo nutarta kurti dinaminę paieškos sąsają, kuri suteiktų norimą tikslumo lygį, leidžiant vartotojui dinamiškai formuoti užklauso vartotojo sąsają.
2. Yra didelis panaudojamų vartotojo sąsajos komponentų pasirinkimas, dėl to užklauso formavimas galėtų vykti interaktyviai, pateikiant vartotojui naujus komponentus priklausomai nuo vartotojo veiksmų.
3. Remiantis semantinių portalų analize, suformuluoti pagrindiniai funkciniai semantinės paieškos reikalavimai:
 - paieškos sistema turi padėti vartotojui suformuoti užklauso;
 - pateikti paieškos rezultatus;
 - leisti vartotojui redaguoti užklauso.
4. Literatūros šaltinių semantinės paieškos tema analizė parodė, kad paieškai vykdyti tikslinga naudoti OWL ir SPARQL kalbas, paieškos kokybei nustatyti tinkamiausi tikslumo ir grąžinimo parametrai.
5. Semantinės paieškos realizavimo technologijų analizė parodė, kad tinkamiausi tam įrankiai yra Jena karkasas su susijusiomis Joseki saugykla ir Java kalba manipuluoti užklausoms ir duomenims, Javascript ir HTML kalbos – vaizduoti rezultatus ir vykdyti pagrindines tinklapio funkcijas.
6. Išanalizavus ontologijų kūrimo įrankius, nustatyta, kad tinkamiausias iš jų yra Protégé – jis patikimas, vystomas akademinės institucijos, pateikia daug mokomosios medžiagos, nuolat tobulinamas.

3. Dinaminės vartotojo sąsajos formavimo metodas

Šiame skyriuje aprašomas dinamiškai kuriamos vartotojo sąsajos formavimo metodas, tam panaudojami vartotojo sąsajos elementai, detaliai paaiškinamas užklausos formavimo procesas ir SPARQL užklausos generavimo algoritmas.

Analizės dalyje paieškos procesas suskirstytas į keturias dalis, naudojančias skirtingus sąsajos komponentus:

1. užklausos formavimas
2. paieškos algoritmas
3. rezultatų pateikimas
4. sąveika su rezultatais ir užklausos tikslinimas

Panaudojant dinaminę vartotojo sąsają, pirmas ir ketvirtas proceso etapai įgyvendinami naudojant tuos pačius vartotojo sąsajos komponentus ir funkcijas.

Užklausos formavimo etapas pradedamas vartotojui atidarant paieškos tinklapį(3.1. pav. „Pradėti formuoti užklausa“). Sistema pirmoje eilutėje sukuria pirmą teksto įvedimo lauką su patarimų sąrašu. Patarimų sąrašas yra sudaromas naudojant iš anksto paruoštus SPARQL užklausų šablonus. Užpildžius šabloną ir įvykdžius užklausa, gauti rezultatai išvedami į sukurto elemento patarimų sąrašą.

Vartotojui įvedus paieškos subjektą, sukuriama antras elementas eilutėje. Elemento patarimų sąrašas užpildomas pagal šabloną išrinktomis pirmame elemente įvesto subjekto objektinėmis (angl. *Object Property*) ir duomenų (angl. *Data Property*) savybėmis.

Vartotojui įvedus norimą savybę, sistema sukuria trečią elementą ir užpildo jo patarimų sąrašą šabloninės užklausos rezultatais. Jei vartotojas antrame eilutės elemente pasirinko objektinę savybę, sistema naudoja šabloną išrinkti objektams, kurie su subjektu susieti pasirinktu ryšiu. Jei buvo pasirinkta duomenų savybė, naudojamas šablonas, išrenkantis pasirinktos duomenų savybės galimas reikšmes.

Trečiame eilutės elemente įvedus norimą kintamąjį arba reikšmę, vartotojas gali pradėti paiešką arba toliau formuoti užklausa.

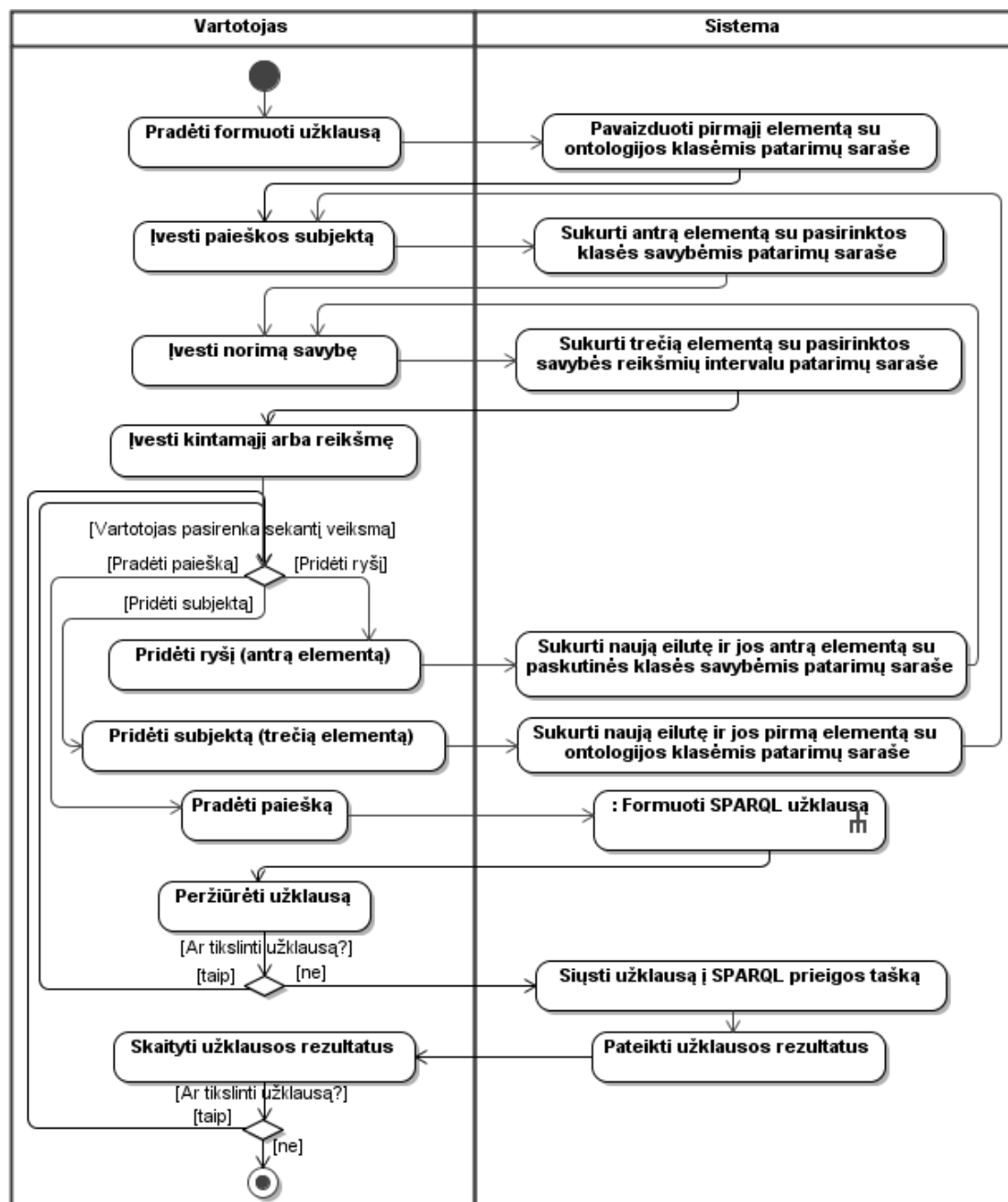
Pridedant naują ryšį sukuriamas nauja eilutė be pirmojo elemento. Taip vartotojas filtruoja gaunamus rezultatus. Antro elemento patarimų sąrašas užpildomas paskutinio įvesto subjekto savybėmis ir grįžtama į atitinkamą eilutės pildymo būseną.

Vartotojui pasirinkus naujo subjekto pridėjimą formuojamas kelias ontologijoje, o vartotojo sąsajoje sukuriamas nauja eilutė su visomis ontologijos

klasėmis patarimų sąrašė. Baigus pildyti eilutę, vartotojas vėl gali rinktis tarp užklauso kūrimo ir paieškos vykdymo.

Įvykdžiusi algoritmą, kuris detaliau paaiškintas toliau šiame skyriuje, sistema suformuoja SPARQL užklauso ir pateikia ją vartotojo peržiūrai. Vartotojui pasirinkus paieškos funkciją, sistema pereina į rezultatų pateikimo etapą.

Sąveikos su rezultatais ir užklauso tikslinimo etape gauti suformuotos SPARQL užklauso rezultatai apdorojami ir išvedami lentele vartotojo peržiūrai. Jei vartotojas nori pakeisti užklauso, jis gali grįžti į ankstesnius užklauso formavimo etapus.



3.1. pav. Užklauso formavimo proceso diagrama

SPARQL užklauso generavimo algoritmo (3.2. pav.) veikimui reikalinga, kad būtų suformuota užklausa iš vartotojo sąsajos elementų. Vartotojui suformavus užklausa ir nurodžius atlikti paiešką, sistema tikrina iš eilės visus suformuotos vartotojo sąsajos elementus ir kiekvienam sukuria SPARQL užklauso dalį.

Kol yra neapdorotų vartotojo sąsajos elementų eilučių, vykdomos šios algoritmo dalys:

1. Tikrinama ar pirmos eilutės elemente yra kintamasis ir ar toks kintamasis yra SPARQL užklauso SELECT dalyje. Jei elemente yra kintamasis ir jo nėra SELECT dalyje, jis pridedamas prie SELECT užklauso dalies galo, jei elementas neturi kintamojo, pereinama prie sekančio žingsnio.
2. Taip pat kaip ir prieš tai buvusiame žingsnyje yra patikrinamas trečias elementas ir jei jame yra nurodytas kintamasis, SELECT dalis papildoma nauju kintamuoju.
3. Jei pirmas ir trečias eilutės elementai neturi kintamųjų, naudojamas šablonas:

```
{%s movies:%s movies:%s} UNION  
{%s movies:%s '%s'^^xsd:dateTime} UNION  
{%s movies:%s '%s'^^xsd:double} UNION  
{%s movies:%s '%s'^^xsd:integer} UNION  
{%s movies:%s '%s'^^xsd:string}
```

„%s“ nurodo šablono vietas, užpildomas tekstu iš vartotojo sąsajos elementų.

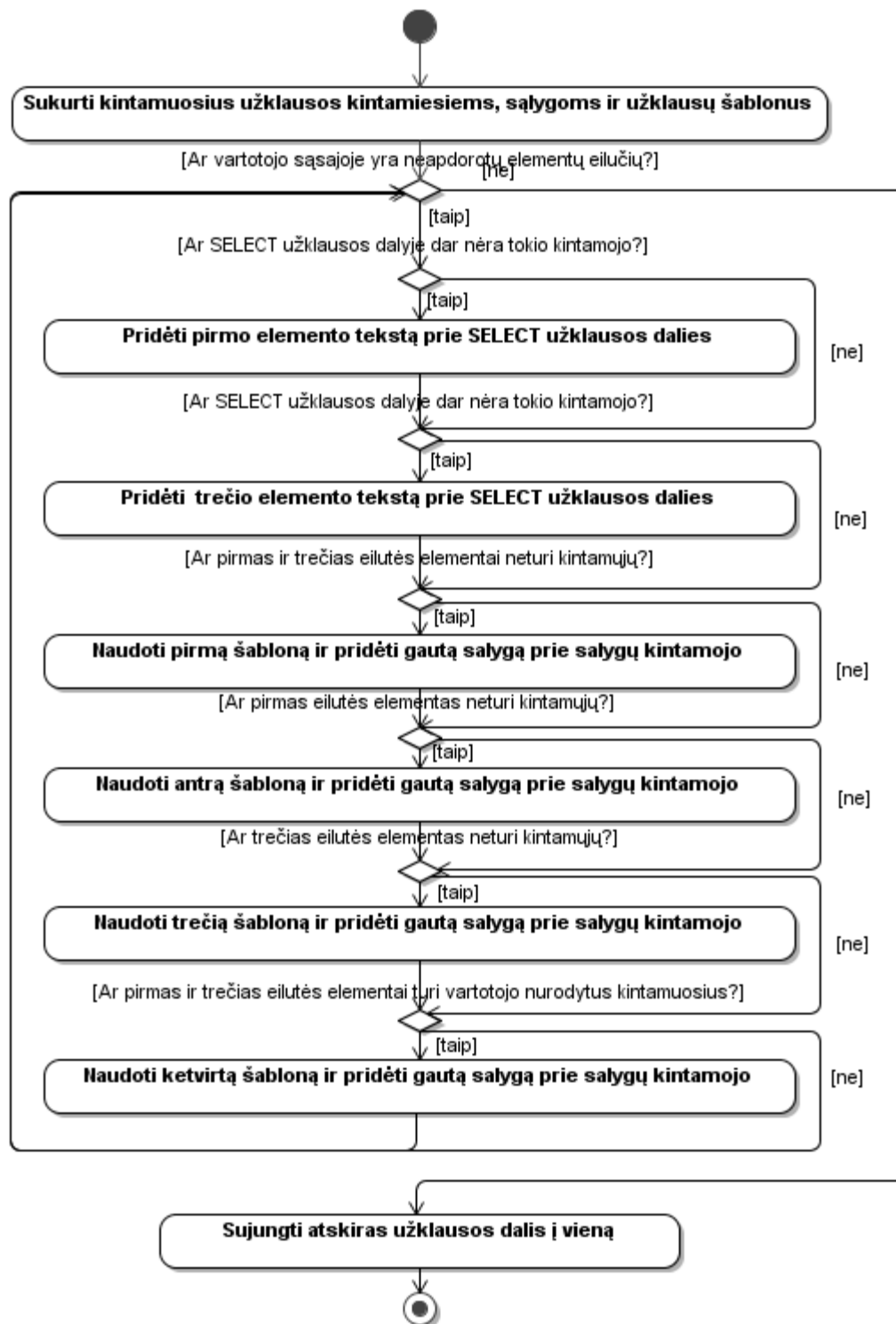
4. Jei kintamasis yra tik trečiame eilutės elemente, naudojamas šablonas:
5. Jei kintamasis yra tik pirmame eilutės elemente, naudojamas šablonas:

```
{%s movies:%s %s}  
  
{%s movies:%s movies:%s} UNION  
{%s movies:%s '%s'^^xsd:dateTime} UNION  
{%s movies:%s '%s'^^xsd:double} UNION  
{%s movies:%s '%s'^^xsd:integer} UNION  
{%s movies:%s '%s'^^xsd:string}
```

6. Jei kintamieji nurodyti ir pirmame ir trečiame eilutės elementuose, naudojamas šablonas:

```
{%s movies:%s %s}
```

Kai visos varotojo sąsajos eilutės baigiamos apdoroti, atskiros užklauso dalys (predikatai, SELECT dalis, formuotos sąlygos) apjungiami į vieną užklausa ir gražinama tolesniam programos darbui.



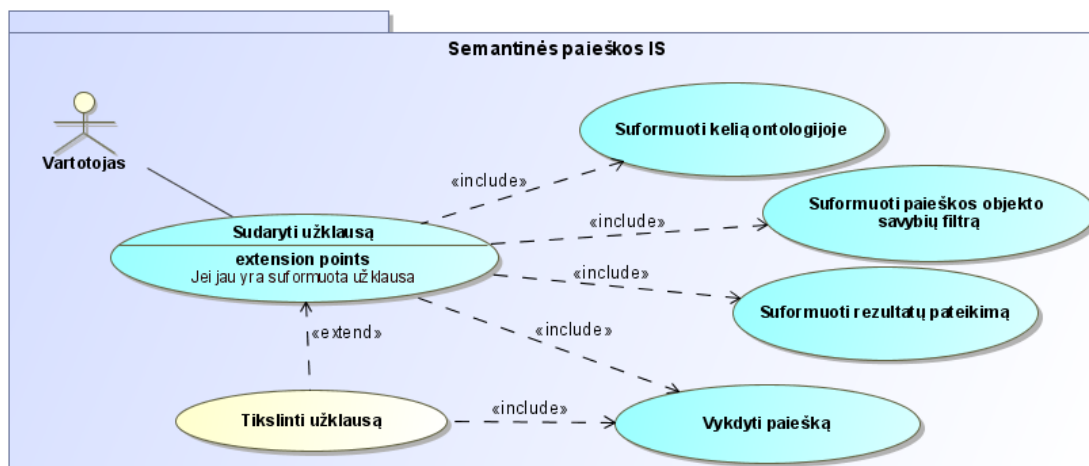
3.2. pav. SPARQL generavimo algoritmas

4. Eksperimentinės sistemos kūrimas

4.1. Reikalavimų specifikacija

4.1.1. Funkciniai reikalavimai

4.1.1.1. Prototipo panaudojimo atvejų diagrama



4.1 pav. Kompiuterizuojamų panaudojimo atvejų diagrama

Užklausos sudarymo funkcija apima kelio ontologijoje formavimą, filtro formavimą, rezultatų formavimą ir paieškos vykdymo pradžią. Užklausos sudarymas susideda iš keturių smulkesnių procesų:

1. Kelio ontologijoje formavimas – vartotojas, nuroydamas kelią ontologijoje, suformuoja ryšius tarp konceptų. Paieškoje bus ieškomi tik taip susiję konceptai, kaip nurodyta kelio formavime. Keliui formuoti naudojami dinamiškai pridedami vartotojo sąsajos elementai.
2. Paieškos objekto savybių filtro formavimas – vartotojas suformuoto kelio konceptams nurodo norimas savybes – paieškos filtrą, pagal kurį bus atliekama paieška. Filtro formavimas įgyvendinamas tais pačiais dinamiškai formuojamais elementais kaip ir kelio formavimas.
3. Rezultatų pateikimo formavimas – vartotojas nurodo kaip bus pateikti paieškos rezultatai. Tai gali būti rikiavimo prioritetai arba kokios ieškomo koncepto savybės bus pateikiamos ekrane.
4. Paieškos vykdymas – Pasirinkęs norimus nustatymus vartotojas nuspaudžia mygtuką pradedantį paiešką.

Užklauso tikslinimo funkciją vartotojas gali pasirinkti kai egzistuoja ankstesnės paieškos rezultatai, tada vykdomi tie patys keturi procesai kaip ir užklauso sudaryme, toliau mažinant gautų rezultatų kiekį, kol randamas vartotoją tenkinantis rezultatas.

4.1.1.1. Panaudojimo atvejų specifikacijos

PA Nr.1 „Sudaryti užklausa“		
Tikslas. Suformuoti užklausa naudojant vartotojo sąsajos elementus		
Aprašymas. Pagrindinis sistemos PA		
Prieš sąlyga	Nėra	
Aktorius	Vartotojas	
Sužadinimo sąlyga	Vartotojas atidarė paieškos tinklapį	
Veiklos taisyklės		
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	PA Nr.2 „Suformuoti kelią ontologijoje“
		PA Nr.3 „Suformuoti paieškos objekto savybių filtrą“
		PA Nr.4 „Suformuoti rezultatų pateikimą“
Specializuoja PA	PA Nr.5 „Vykdyti paiešką“	
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. Vartotojas vykdo panaudojimo atvejį „Formuoti paieškos kelią“		Sistema pereina į PA „Formuoti paieškos kelią“
2. Vartotojas vykdo panaudojimo atvejį „Formuoti paieškos savybių filtrą“		Sistema pereina į PA „Formuoti paieškos savybių filtrą“
3. Vartotojas vykdo panaudojimo atvejį „Formuoti rezultatų pateikimą“		Sistema pereina į PA „Formuoti rezultatų pateikimą“
4. Vartotojas vykdo panaudojimo atvejį „Vykdyti paiešką“ ontologijoje		Sistema pereina į PA „Vykdyti paiešką“
Po sąlyga	Užklausa suformuota	
Alternatyvūs scenarijai		
-		
Pastabos		
1. Vartotojas bet kada gali baigti PA		

PA Nr.2 „Suformuoti kelią ontologijoje“		
Tikslas. Patikslinti sritį kurioje vyks paieška		
Aprašymas. Šis PA yra PA „Sudaryti užklausa“ dalis		
Prieš sąlyga	Nėra	
Aktorius	Vartotojas	
Sužadinimo sąlyga	Vartotojas pasirinko paieškos funkciją	
Veiklos taisyklės		
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	-
	Specializuoja PA	PA Nr.1 „Sudaryti užklausa“
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. Vartotojas suformuoja kelią paieškos pradžia		Sistema suformuoja pradinę užklauskos dalį
2. Vartotojas baigia PA		
Po sąlyga	Paieškos kelias suformuotas	
Alternatyvūs scenarijai		
1. Vartotojas palieka paieškos kelią nesuformuotą		
Pastabos		
1. Vartotojas bet kada gali baigti PA		

PA Nr.3 „Suformuoti paieškos objekto savybių filtrą“		
Tikslas. Patikslinti sritį kurioje vyks paieška		
Aprašymas. Šis PA yra PA „Sudaryti užklausa“ dalis		
Prieš sąlyga	Nėra	
Aktorius	Vartotojas	
Sužadinimo sąlyga	Vartotojas pasirinko paieškos funkciją	
Veiklos taisyklės		
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	-
	Specializuoja PA	PA Nr.1 „Sudaryti užklausa“
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. Vartotojas nurodo kokias savybes turėtų atitikti jo ieškomas objektas		Sistema suformuoja užklauskos filtro dalį
2. Vartotojas baigia PA		
Po sąlyga	Paieškos filtras suformuotas	

Alternatyvūs scenarijai	
1. Vartotojas palieka paieškos filtrą nesuformuotą	
Pastabos	
1. Vartotojas bet kada gali baigti PA	

PA Nr.4 „Suformuoti rezultatų pateikimą“		
Tikslas. Patikslinti sritį kurioje vyks paieška		
Aprašymas. Šis PA yra PA „Sudaryti užklausa“ dalis		
Prieš sąlyga	Nėra	
Aktorius	Vartotojas	
Sužadinimo sąlyga	Vartotojas pasirinko paieškos funkciją	
Veiklos taisyklės		
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	-
	Specializuoja PA	PA Nr.1 „Sudaryti užklausa“
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. Vartotojas nurodo kokia informacija jį domina ir kokia tvarka ji turėtų būti atvaizduota		Sistema suformuoja rezultatų struktūrą ir rūšiavimo kriterijų
2. Vartotojas baigia PA		
Po sąlyga	Rezultatų struktūra ir rūšiavimo kriterijai suformuoti	
Alternatyvūs scenarijai		
1. Vartotojas nenurodo kaip ir kokie rezultatai turėtų būti pateikti		
Pastabos		
1. Vartotojas bet kada gali baigti PA		

PA Nr.5 „Vykdėti paiešką“		
Tikslas. Atlikti paiešką ir atvaizduoti gautus rezultatus pagal vartotojo nustatymus		
Aprašymas. Šis PA yra PA „Sudaryti užklausa“ ir „Tikslinti užklausa“ dalis		
Prieš sąlyga	Nėra	
Aktorius	Vartotojas	
Sužadinimo sąlyga	Vartotojas pasirinko paieškos funkciją ir komandą „leškoti“	
Veiklos taisyklės		
Susiję panaudojimo	Išplečia PA	-

atvejai	Apima PA	-
	Specializuoja PA	PA Nr.1 „Sudaryti užklausą“ PA Nr.6 „Tikslinti užklausą“
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. Vartotojas nurodo pradėti paiešką	Sistema suformuoja SPARQL užklausą pagal vartotojo nustatytus parametrus	
	Sistema vykdo užklausą	
	Sistema pateikia gautus rezultatus pagal vartotojo nustatytus parametrus	
2. Vartotojas baigia PA		
Po sąlyga	Užklausos rezultatai pateikti	
Alternatyvūs scenarijai		
-		
Pastabos		
-		

PA Nr.6 „Tikslinti užklausą“		
Tikslas. Vartotojas nori atlikti paiešką ankstesnės paieškos rezultatuose		
Aprašymas. Šis PA išplečia PA „Sudaryti užklausą“		
Prieš sąlyga	Pateikti anksčiau vykdytos užklausos rezultatai	
Aktorius	Vartotojas	
Sužadinimo sąlyga	Vartotojas nori modifikuoti anksčiau suformuotą užklausą	
Veiklos taisyklės		
Susiję panaudojimo atvejai	Išplečia PA	PA Nr.1 „Sudaryti užklausą“
	Apima PA	PA Nr.5 „Vykdyti paiešką“
	Specializuoja PA	-
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. Vartotojas vykdo panaudojimo atvejį „Formuoti paieškos kelią“		Sistema pereina į PA „Formuoti paieškos kelią“
2. Vartotojas vykdo panaudojimo atvejį „Formuoti paieškos savybių filtrą“		Sistema pereina į PA „Formuoti paieškos savybių filtrą“
3. Vartotojas vykdo panaudojimo atvejį „Formuoti rezultatų pateikimą“		Sistema pereina į PA „Formuoti rezultatų pateikimą“
4. Vartotojas vykdo panaudojimo atvejį „Vykdyti paiešką“ nurodytuose rezultatuose		Sistema pereina į PA „Vykdyti paiešką“
5. Vartotojas baigia PA		
Po sąlyga	-	

Alternatyvūs scenarijai	
-	
Pastabos	
1. Vartotojas bet kada gali baigti PA	

4.1.2. Nefunkciniai reikalavimai

Reikalavimas Nr. 1 „Sistemos išvaizda“	
Aprašymas:	Vartotojo sąsaja turi būti intuityvi, funkcijos pasiekiamos iš pradinio sistemos lango vartotojui prisijungus
Pagrindimas:	Vartotojui turi būti paprasta naudotis sistema
Tinkamumo kriterijus:	Vartotojas lengvai randa visas jam prieinamas funkcijas, meniu pavadinimai aiškūs ir nedviprasmiški, patogiai išdėstyti informacijos įvedimo laukai

Reikalavimas Nr. 2 „Stilius“	
Aprašymas:	Vartotojo sąsaja neturi naudoti ryškių, kontrastuojančių spalvų
Pagrindimas:	Vartotojo neturi erzinti sistemos aplinka
Tinkamumo kriterijus:	Vartotojo neerzina sistemos naudojamas spalvų derinys

Reikalavimas Nr. 3 „Naudojimosi paprastumas“	
Aprašymas:	Grafinė vartotojo sąsaja turi būti aiški, lengvai suprantama
Pagrindimas:	Vartotojui turi būti lengva naudotis sistema
Tinkamumo kriterijus:	Pateikiamą informaciją suskirstoma į atskirus puslapius, į kuriuos galima patekti paspaudus meniu punktus ar atitinkamas nuorodas. Meniu punktų pavadinimai ir nuorodos turi būti aiškūs, kad būtų įmanoma iš pavadinimo orientuotis, kas yra tame puslapyje

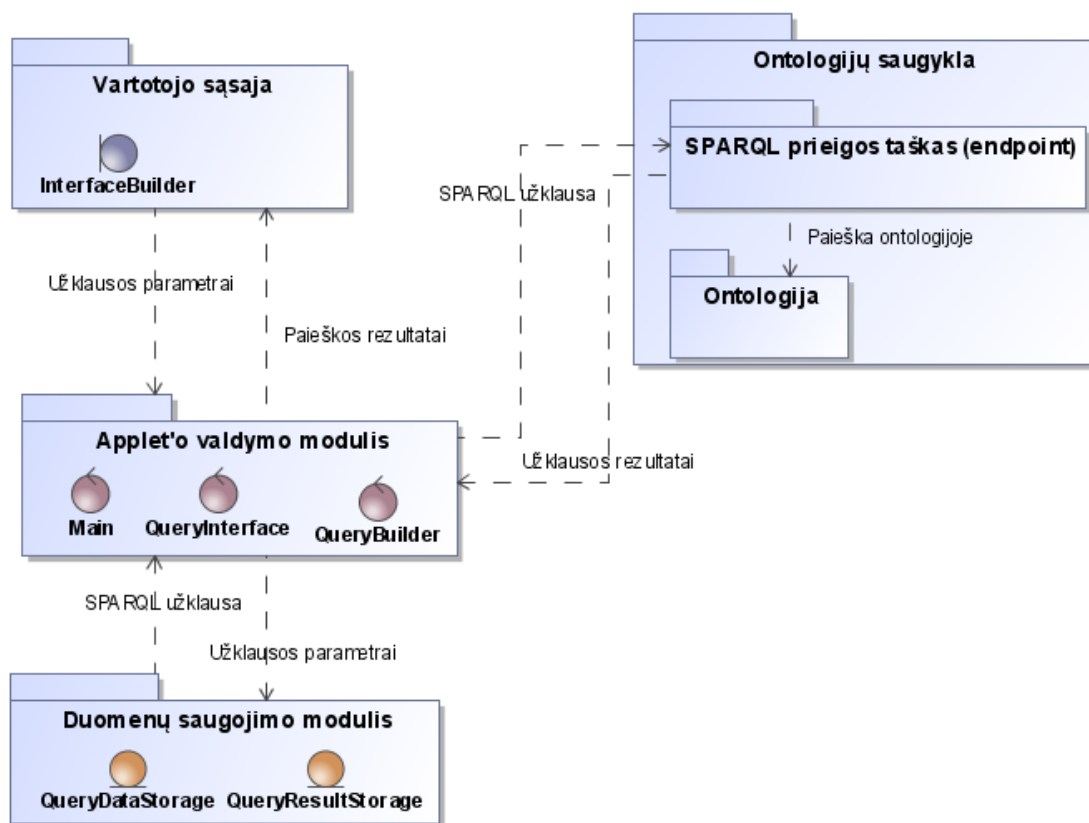
Reikalavimas Nr. 4 „Mokymosi reikalavimai“	
Aprašymas:	Turi būti paprasta išmokti naudotis sistema
Pagrindimas:	Sistema turi būti nesudėtinga
Tinkamumo kriterijus:	Neapmokyti vartotojai išmoksta naudotis sistema per 1 valandą ar greičiau.

4.2. Sistemos projektas

Projekto tikslas – suprojektuoti semantinės paieškos modelį. Tai apima vartotojo sąsajos formavimo, vartotojo įvestų paieškos parametrų transformavimą į SPARQL užklausas, užklausų vykdymo logikos, bei rezultatų saugojimo ir apdorojimo klases.

4.2.1. Sistemos architektūros projektas

4.2.1.1. Sistemos loginė architektūra



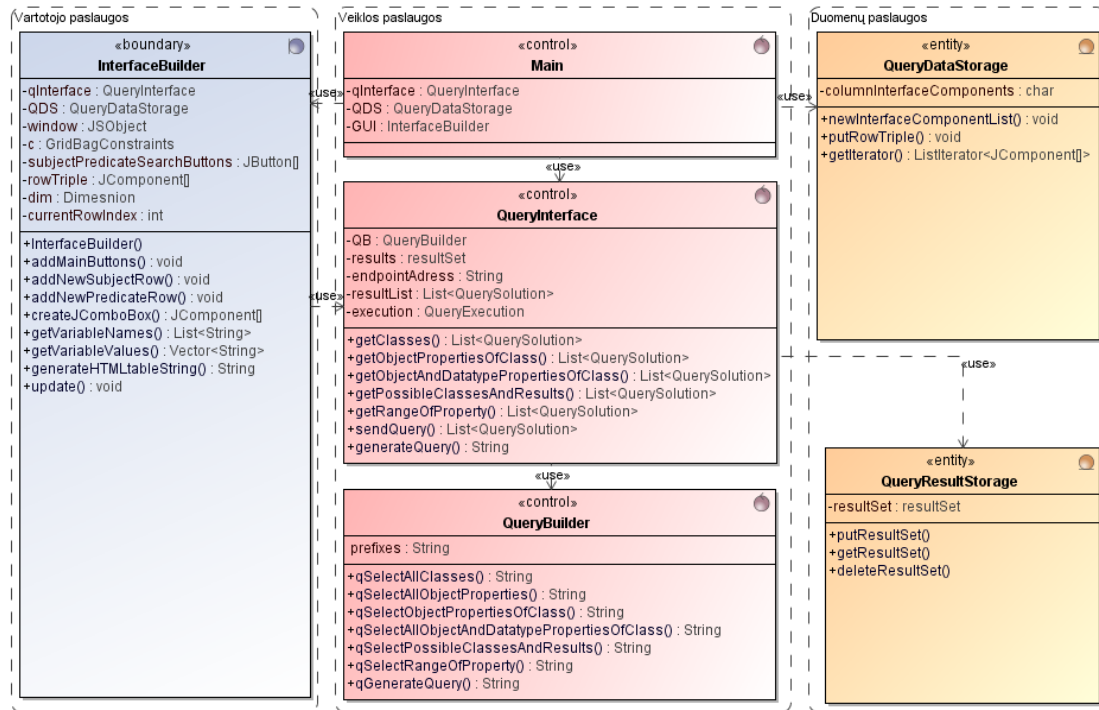
4.2. pav. Sistemos loginės architektūros diagrama

Sistema kurta pagal MVC architektūrą ir ją sudaro keturi pagrindiniai komponentai:

1. Vartotojo sąsaja – atsako už vartotojo sąsajos formavimą ir rezultatų atvaizdavimą.
2. Duomenų saugojimo modulis – jame saugomi sistemos duomenys.
3. Applet'o valdymo modulis – atsako už visų komponentų bendradarbiavimą.
4. Ontologijų saugykla – talpina ontologiją ir priima užklausas per SPARQL prieigos tašką.

Iš vartotojo sąsajoje įvestų paieškos parametrų Applet'o valdymo modulis suformuoja SPARQL užklausą. Užklausą nusiuntus į ontologijų saugyklos prieigos tašką, gražinti rezultatai apdorojami pagal vartotojo sąsajoje įvestus parametrus ir persiunčiami vartotojo sąsajai atvaizduoti.

4.2.1.2. Vartotojo, veiklos ir duomenų paslaugos



4.3. pav. Sistemos paslaugų diagrama

Sistemos klases galima suskirstyti į tris paslaugų grupes (4.3 pav.). Kurių pirmoji yra vartotojo paslaugos. Šios klases formuoja programos išvaizdą.

Veiklos paslaugų klasėse atliekami visi loginiai sistemos veiksmai. Vartotojui sąveikaujant su vartotojo sąsajos elementais, kviečiami veiklos paslaugų klasių metodai.

Duomenų paslaugų klasėse saugomi sistemos duomenys ir duomenų logika. Į šias klases kreipiamasi kai kitoms paslaugoms reikia išsaugoti ar paimti saugomus duomenis.

4.2.1.3. Sistemos klasės

Sistemos klasių modelis pateikiamas 4.4 paveiksle.



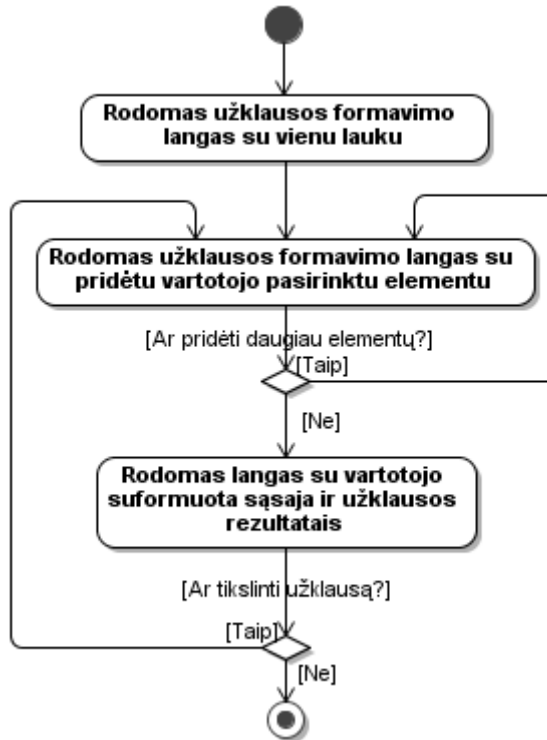
4.4. pav. Sistemos UML klasių diagrama

Aukščiau pateiktoje diagramoje pavaizduotos pagrindinės projekto klasės:

- Main klasė skirta sukurti kitų klasių objektams ir kintamiesiems. Tai yra pagrindinė projekto klasė.
- QueryDataStorage klasės objekte saugomi vartotojo nurodyti užklauso parametrai, dinaminio vartotojo sąsajos elementų sąrašo pavidalu. Klasė turi metodus parametų įvedimui, paėmimui ir ištrynimui.
- QueryResultStorage klasės objekte saugomi atliktų užklauso rezultatai. Rezultatus galima įdėti, paimti arba ištrinti.
- QueryBuilder klasės objektas iš pateiktų užklauso parametų sukuria ir grąžina SPARQL užklauso. Taip pat šioje klasėje saugomi visi reikalingi užklauso šablonai, taigi norint pritaikyti sistemą kitai ontologijai tereikia modifikuoti šią klasę.
- QueryInterface klasės objektas siunčia nurodytą užklauso SPARQL prieigos taškui ir grąžina gautus rezultatus.
- InterfaceBuilder klasės objektas dinamiškai kuria vartotojo sąsają ir pagal vartotojo veiksmus kviečia atitinkamas programos funkcijas.

4.2.2. Sistemos elgsenos modelis

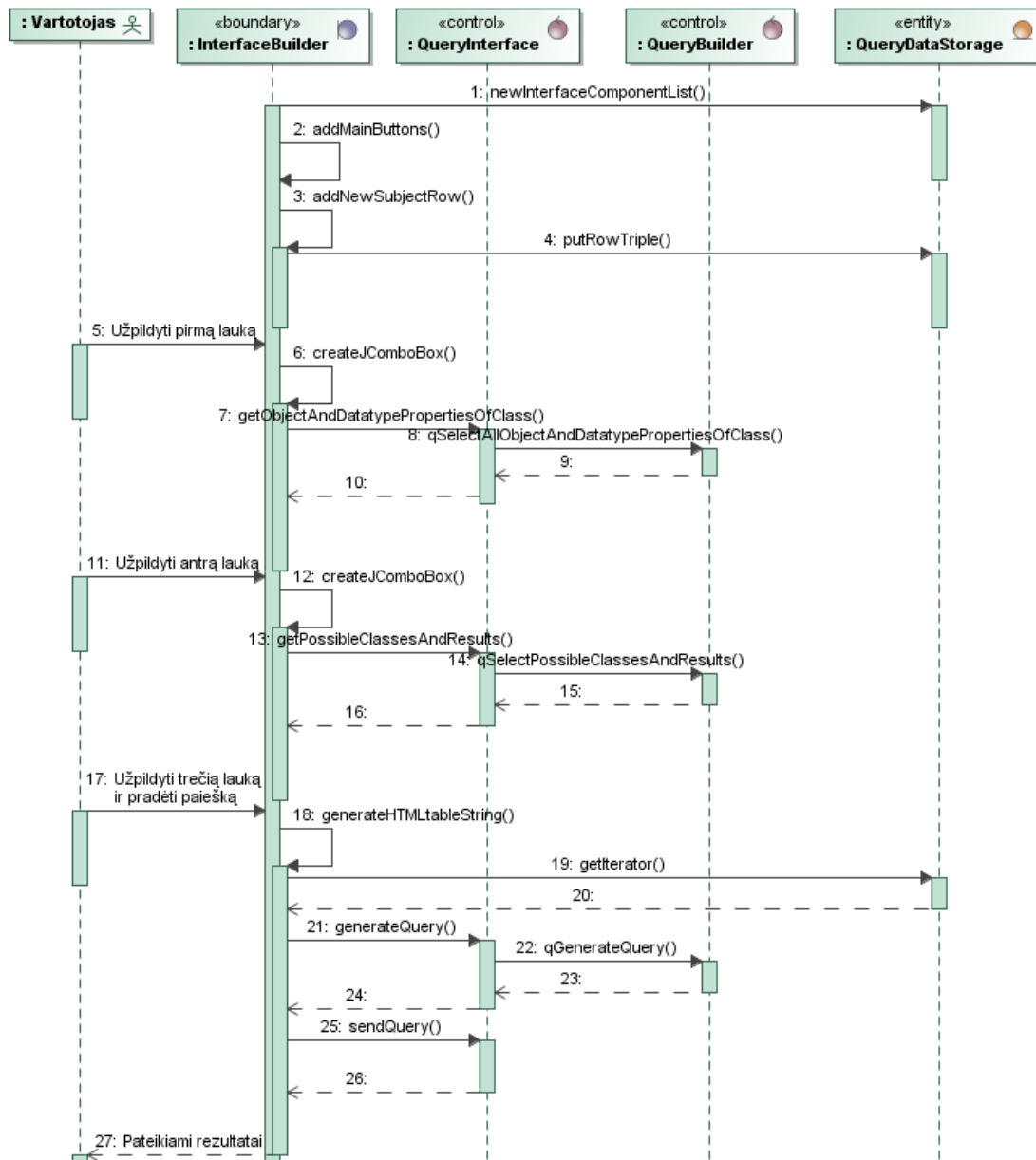
Šiame skyriuje pateikiamos sistemos elgsenos diagramos, vaizduojančios kaip naudojami sistemos elementai atliekant tam tikras funkcijas.



4.5. pav. Vartotojo sąsajos navigacijos planas

Vartotojo sąsajos navigacijos planas (4.5 pav.) vaizduoja sistemos būsenų kaitą. Tik pradėjus veikti programai yra rodomas užklauso formavimo langas su vienu lauku ir trimis algoritmo valdymo mygtukais. Vartotojui pradėjus formuoti užklausa pereinama prie antrosios būsenos, kurioje rodomas užklauso formavimo langas su nauju elementu, kurį vartotojas pasirinko pridėti.

Jei vartotojas pasirenka pridėti daugiau elementų, sistema grįžta į tą pačią būseną. Jei vartotojas pasirenka užklauso vykdymą, sistema pereina į trečią būseną, kurioje yra rodoma vartotojo jau suformuota užklausa ir šalia pateikti įvykdytos užklauso rezultatai. Vartotojas gali baigti darbą arba tikslinti užklausa. Pasirinkus užklauso tikslinimą, sistema grįžta į antrą būseną.



4.6. pav. Ontologijos naršymo sekų diagrama

4.6 pav. pavaizduota diagrama vaizduoja sistemos elementų panaudojimą vykdant paprasčiausią užklausą, susidedančią iš subjekto, predikato ir objekto.

Pirmiausia, pradedant programos veikimą, objektas „InterfaceBuilder“ kreipiasi į „QueryDataStorage“ objektą ir sukuria naują vartotojo sąsajos elementų sąrašą. Sukūrus naują sąrašą kuriami algoritmo valdymo mygtukai (pridėti subjektui, predikatui ir paieškos vykdymui) ir pirmos subjekto-objekto-predikato eilutės pirmas elementas, kuris, su visa eilute, per „putRowTriple()“ metodą patalpinamas „QueryDataStorage“ klasėje.

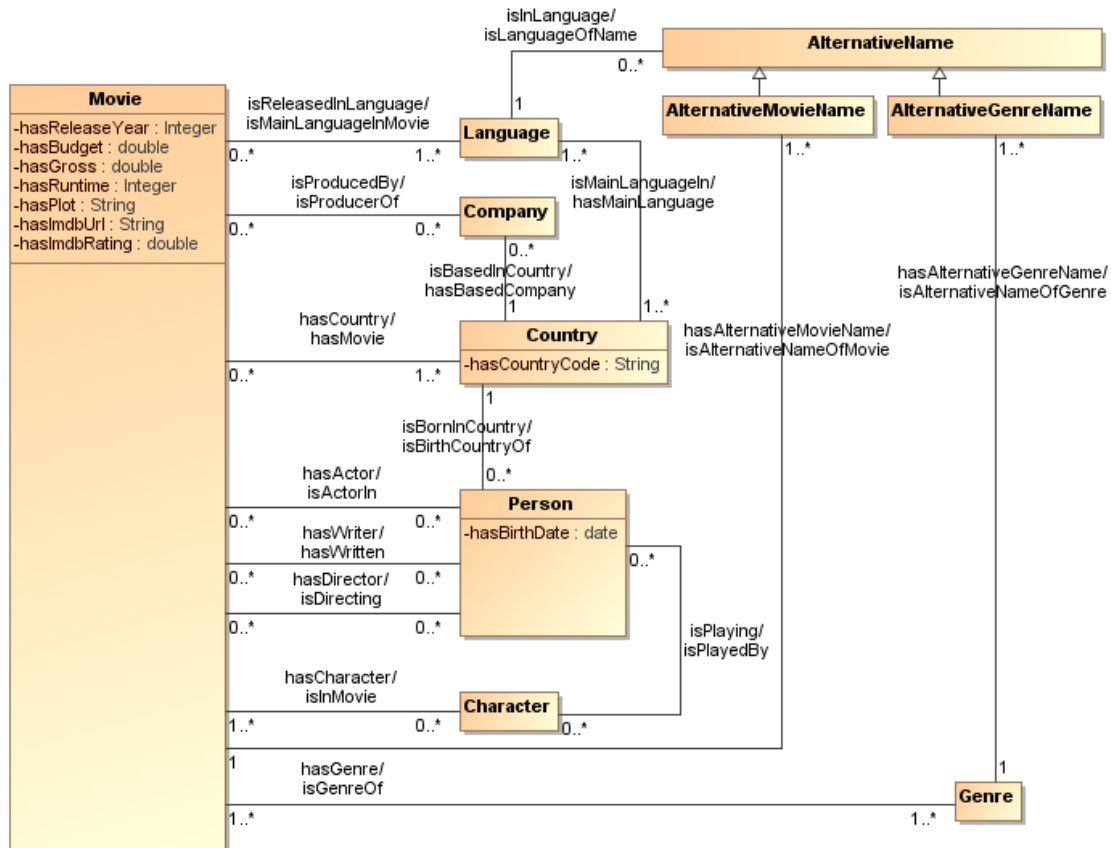
Vartotojui užpildžius pirmos eilutės subjekto lauką, objektas „InterfaceBuilder“ kviečia „createJComboBox()“ metodą, kuriame nustatoma, kad

buvo užpildytas pirmas eilutės laukas ir iškviečiamas objekto „QueryInterface“ „getObjectAndDatatypePropertiesOfClass“ metodas, skirtas išrinkti visoms galimoms pirmame lauke įvesto subjekto savybėms. Kadangi pačioje „QueryInterface“ klasėje užklausų šablonai nesaugomi, šis objektas kreipiasi į „QueryBuilder“ objektą ir gavęs atitinkamą užklausą siunčia ją SPARQL prieigos taškui. Gautų rezultatų sąrašas panaudojamas antro eilutės elemento patarimų sąrašė. Vartotojui užpildžius antrą pirmos eilutės lauką, atliekami analogiški veiksmai, tik aptikus, jog užpildytas antras laukas, kviečiamas metodas, skirtas išrinkti galimoms antrame lauke nurodytos savybės reikšmėms, kurios gali būti tiek objektai, tiek duomenų savybių konkrečios reikšmės.

Vartotojui užpildžius trečią lauką ir nuspaudus paieškos vykdymo mygtuką, kviečiamas „generateHTMLtableString“ metodas skirtas rezultatų lentelei sugeneruoti, kuriame kviečiamas „QueryDataStorage“ vartotojo sąsajos elementų sąrašo iteratorius. Iteratoriaus pagalba „generateQuery“ metodas „QueryBuilder“ objekte generuoja SPARQL užklausą, kuri yra nusiunčiama SPARQL prieigos taškui ir gražinti rezultatai per Javascript pateikiami vartotojui pagrindiniame puslapyje.

4.2.3. Ontologijos konceptų schema

Sistema naudoja ontologiją, kurios UML diagrama pateikta 4.7. paveiksle.



4.7. pav. Ontologijos konceptų diagrama

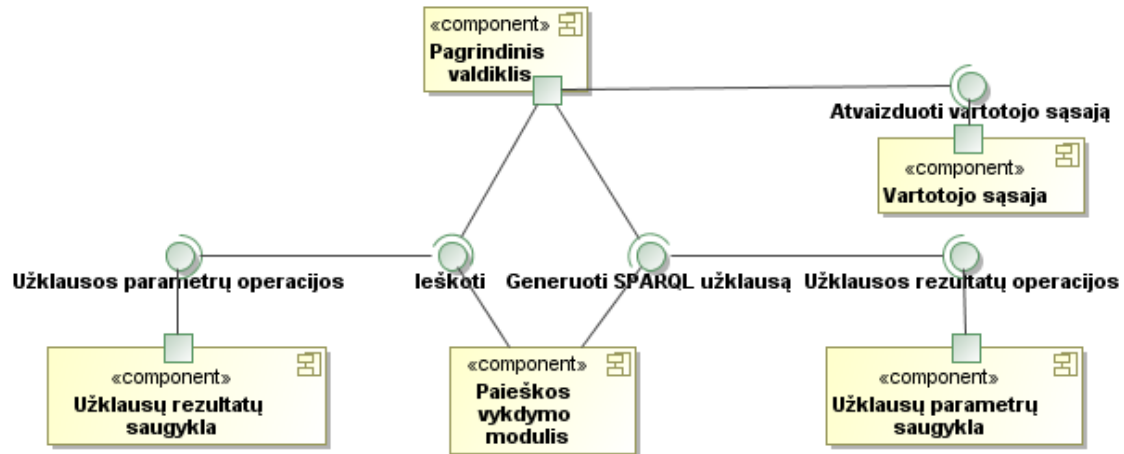
Klasės „AlternativeMovieName“ ir „AlternativeGenreName“ paveldi klasę „AlternativeName“, kas suteikia galimybę pateikti konceptų pavadinimų informaciją keliomis kalbomis, naudojant sinonimus.

Klasės „Movie“ ir „Person“ sujungtos šešiais savybių ryšiais, kas leidžia tam pačiam asmeniui atlikti kelis vaidmenis ar atlikti kelis vaidmenis filmo kūrimo procese (režisierius, rašytojas, aktorius).

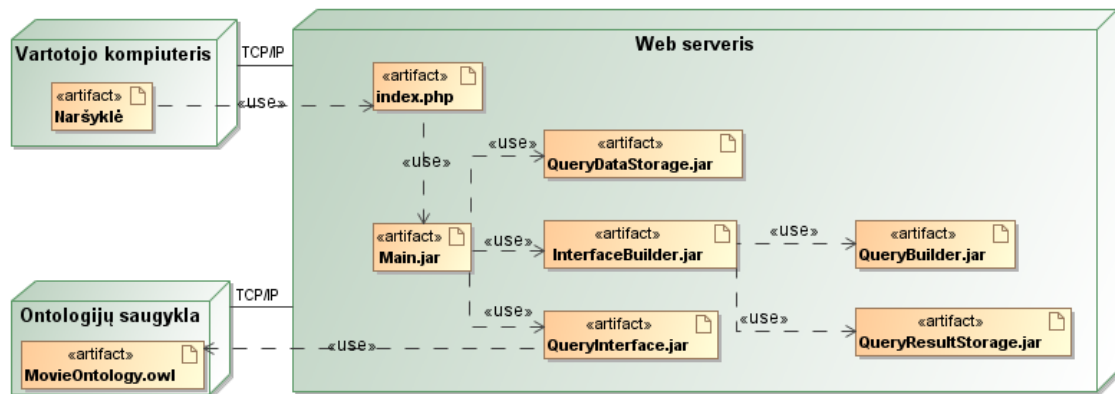
Kiekviena objekto savybė ontologijoje turi atvirkštinę savybę (angl. *inverse object property*). Tai nėra būtina, kad modelis veiktų, bet padeda vartotojui suformuoti daugiau ir įvairesnių užklausų, nes atsiranda daugiau ryšių tarp objektų.

4.2.4. Realizacijos modelis

Sistemos realizacijos modelis, pateikiamas 4.9. paveiksle, susideda iš Web serverio, ontologijų saugyklos ir vartotojo kompiuterio. Web serveryje talpinami index.php ir Java applet'o failai.



4.8. pav. Komponentų diagrama



4.9. pav. Diegimo diagrama

Ontologija yra saugoma ontologijų saugykloje, kuri gali veikti tiek nutolusiame kompiuteryje, tiek tame pačiame kompiuteryje kaip ir Web serveris.

Vartotojui kreipiantis į puslapį, applet'as užkraunamas į vartotojo kompiuterį. Iš ten applet'as gali betarpiškai siųsti užklausas ontologijų saugyklai, nesikreipiant į Web serverį.

4.3. Sistemos realizacija

4.3.1. Vartotojų sąsajos modelis

Vartotojo sąsajos (4.10 pav.) modelį sudaro užklauso formavimo dalis ir rezultatų pateikimo lentelė. Sistemos žinučių išvedimo laukas neįeina į galutinio modelio sudėtį, bet sistemos prototipo stadijoje yra reikalingas sistemos veikimui sekti.

Add subject	Add predicate	Search
Company	isProducerOf	The_Smurfs
		<ul style="list-style-type: none">The_FountainFireflyFight_ClubTotal_RecallSnatchDonnie_DarkoThe_SmurfsThe_Man_from_Earth

Company
Kernel_Entertainment_Company
Columbia_Pictures
Sony_Pictures_Animation

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX movies: <http://www.semanticweb.org/ontologies/2012/0/
movies.owl#>
SELECT ?Company WHERE {
  {?Company movies:isProducerOf movies:The_Smurfs} UNION
  {?Company movies:isProducerOf 'The_Smurfs'^^xsd:dateTime} UNION
  {?Company movies:isProducerOf 'The_Smurfs'^^xsd:double} UNION
  {?Company movies:isProducerOf 'The_Smurfs'^^xsd:integer} UNION
  {?Company movies:isProducerOf 'The_Smurfs'^^xsd:string}
}
```

Clear

4.10. pav. Vartotojo sąsajos modelis

Užklauso formavimo dalį sudaro trys mygtukai skirti užklauso formavimo algoritmui valdyti ir dinamiškai generuojami komponentai. Kiekvienas komponentas turi savo patarimų sąrašą pateikiamą vartotojui pildant atitinkamą lauką.

Pirmas mygtukas „Add subject“ prideda papildomą subjekto lauką vartotojo sąsajoje. Šio lauko patarimų sąrašas yra automatiškai užpildomas ontologijos klasių sąrašu.

Antras mygtukas „Add predicate“ prideda papildomą predikatą. Naujas elementas būna suformuojamas naujoje eilutėje ir jo patarimų sąrašas užpildomas paskutinės klasės, nurodytos subjektų stulpelyje galimais ryšiais ir savybėmis.

Trečias mygtukas „Search“ inicijuoja užklauso formavimo algoritmą ir paieškos vykdymą.

Rezultatų pateikimo lentelės stulpelių antraštėse nurodomi užklauso kintamųjų pavadinimai, stulpeliuose pateikiami patys kintamieji. Į rezultatus įtraukiami visi subjektai ir vartotojo nurodyti SPARQL kintamieji, kurie gali būti nurodomi objekto ir subjekto laukuose.

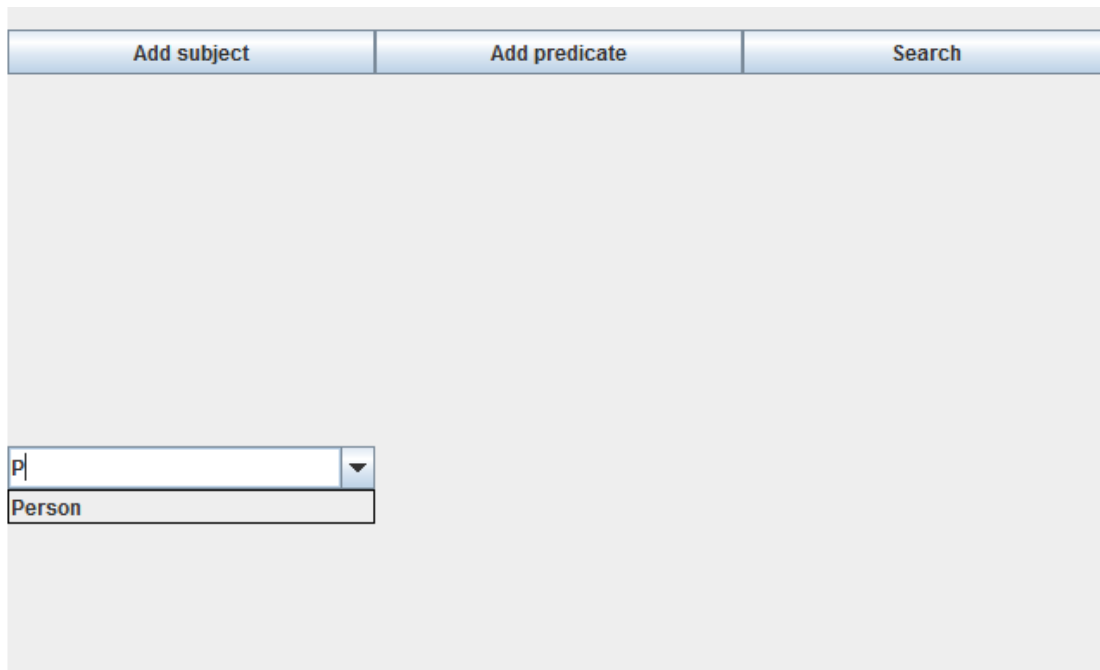
Sistemos žinučių laukas pateikia informaciją apie sistemos veikimą, tokią kaip klaidos, suformuojamos SPARQL užklauso, atsakas į tam tikrus veiksmus vartotojo sąsajoje.

4.3.2. Sistemos veikimo aprašymas

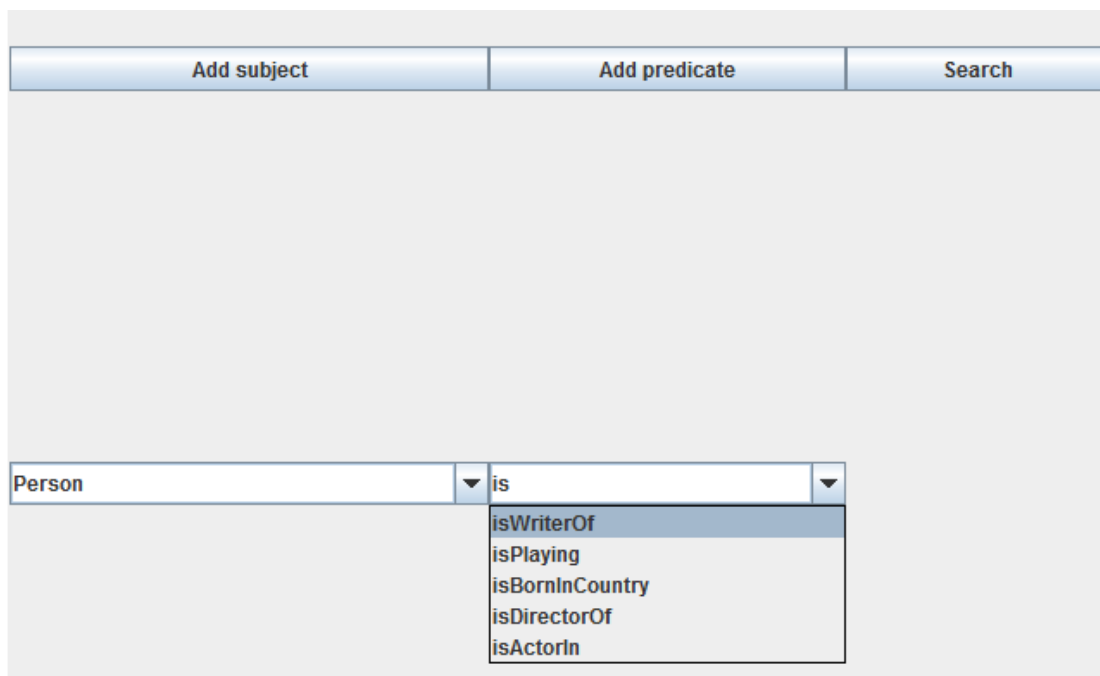
Šiame skyriuje vaizduojamas sistemos prototipo veikimas sudėtingos užklauso formavimo metu pateikiant galima sistemos naudojimo scenarijų.

Sistemos prototipas turi automatiškai generuojamus elementus JComboBox įvesties laukus, algoritmų valdymo mygtukus pridėti subjektui ir predikatui, paieškos mygtuką, HTML elementą „textarea“ SPARQL užklauso atvaizduoti ir mygtuką „clear“, kviečiantį JavaScript funkciją, kuri išvalo „textarea“ lauką.

Applet'ui pradėjus darbą yra vykdoma užklausa, išrenkanti visas klases esančias ontologijoje. Gautų rezultatų sąrašas patalpinamas į pirmos eilutės pirmo elemento duomenų modelį, kuris naudojamas atvaizduojant patarimų sąrašą šiam elementui (4.11 pav.). Vartotojui pradėjus rašyti tekstą į įvesties lauką, patarimų sąrašas filtruojamas pagal žodžio pradžią.



4.11. pav. Vartotojo sąsaja pradėjus darbą



4.12. pav. Vartotojo sąsajos formavimas, antras žingsnis

Vartotojui pasirinkus norimą klasę, šiuo atveju „Person“, vykdoma užklausa gaunanti visus objektų ryšius, kurių domenai yra pasirinkta klasė. Iš gautų rezultatų suformuojamas patarimų sąrašas (4.12 pav.).

Add subject	Add predicate	Search
Person	isWriterOf	
		<ul style="list-style-type: none"> Total_Recall Firefly Fight_Club Snatch The_Man_from_Earth The_Fountain

4.13. pav. Vartotojo sąsajos formavimas, trečias žingsnis

Baigiant formuoti užklausą vartotojas gali rinktis iš trečiam elementui sugeneruoto patarimų sąrašo, nurodant konkrečios reikšmės apribojimą (4.13 pav.). Šiuo atveju vartotojas pasirenka išrinkti visus asmenis, parašiusius scenarijus filmams ir atitinkamus filmus. Tam yra panaudojamas sistemos palaikomas SPARQL kintamasis. SPARQL kintamasis yra bet koks simbolių derinys prasidedantis klaustuku (4.14 pav.).

Add subject	Add predicate	Search
Person	isWriterOf	?Movie

Person	Movie
Philip_K_Dick	Total_Recall
Joss_Whedon	Firefly
Chuck_Palahnuk	Fight_Club
Guy_Ritchie	Snatch
Jerome_Bixby	The_Man_from_Earth
Darren_Aronofsky	The_Fountain

4.14. pav. Suformuota pirma užklausa su rezultatais

Baigus formuoti visus tris eilutės elementus pradeda vykdyti paieška. Sugeneruota SPARQL užklausa pateikta 4.15 pav., jos rezultatai yra visų ontologijos asmenų, parašiusių scenarijus filmams ir tų filmų lentelė (4.14 pav.).

```

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX movies: <http://www.semanticweb.org/ontologies/2012/0/
movies.owl#>
SELECT ?Person ?Movie WHERE {
  {?Person movies:isWriterOf ?Movie}
}

```

4.15. pav. Sugeneruota pirma SPARQL užklausa

Add subject	Add predicate	Search
Person	isWriterOf	?Movie
	isDirectorOf	?Movie

Person	Movie
Guy_Ritchie	Snatch
Darren_Aronofsky	The_Fountain

4.16. pav. Papildyta pirma užklausa su rezultatais

Gavęs užklauso atsakymą, vartotojas, tikslindamas užklausa, prideda apribojimą. Paspaudus „Add predicate“ mygtuką sukuriamas naujas predikato elementas su paskutinio subjekto savybėmis patarimų sąrašė. Į paskutinį lauką įrašęs atitinkamą kintamąjį, vartotojas nurodo, kad asmuo, parašęs filmo scenarijų, taip pat turi būti ir to paties filmo režisierius(4.16 pav.). Nurodžius atlikti paiešką, sistema sugeneruoja naują užklausa(4.17 pav.) ir atnaujina rezultatus.

```

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX movies: <http://www.semanticweb.org/ontologies/2012/0/
movies.owl#>
SELECT ?Person ?Movie WHERE {
  {?Person movies:isWriterOf ?Movie}
  {?Person movies:isDirectorOf ?Movie}
}

```

4.17. pav. Atnaujinta SPARQL užklausa

Add subject	Add predicate	Search
Person	isWriterOf	?Movie
	isDirectorOf	?Movie
?Movie	hasA	
	hasAlternativeMovieName	
	hasActor	

4.18. pav. Papildyta antra užklausa, pirma dalis

Norėdamas dar labiau patikslinti užklausa, vartotojas suformuoja papildomą kelio ontologijos grafe dalį, pridėdamas naują savybę esamam subjektui. Tai įvykdoma pasirinkus „Add subject“ funkciją. Sistema naujoje eilutėje sukuria naują elementą analogišką elementui, sukurtam pirmojoje eilutėje. Vartotojui įrašius anksčiau panaudotą kintamąjį, sistema aptinka kintamojo klasę ir užpildo antrojo eilutės elemento patarimų sąrašą atitinkamu galimų savybių sąrašu(4.18 pav.).

Ädd subject	Ädd predicate	Šearch
Person	isWriterOf	?Movie
	isDirectorOf	?Movie
?Movie	hasActor	B
		Benicio_Del_Toro
		Bryan_Cranston
		Brad_Pitt
		Bill_Nighy

4.19. pav. Papildyta antra užklausa, antra dalis

Taip pat kaip ir ankstesniuose žingsniuose, sistema sugeneruoja trečią eilutės elementą ir atitinkamą patarimų sąrašą, iš kurio vartotojas pasirenka konkretų aktorį(4.19 pav.).

Add subject	Add predicate	Search
Person	isWriterOf	?Movie
	isDirectorOf	?Movie
?Movie	hasActor	Brad_Pitt

Person	Movie
Guy_Ritchie	Snatch

4.20. pav. Vartotojo sąsajoje suformuota galutinė užklausa ir rezultatai

Baigus formuoti užklausa(4.20 pav.) ir pradėjus paiešką, sistema sugeneruoja SPARQL užklausa(4.21 pav.) ir atlikus paiešką pateikia rezultatus, kurie šiuo atveju sudaryti iš vieno subjekto ir vieno objekto(4.20 pav.).

```

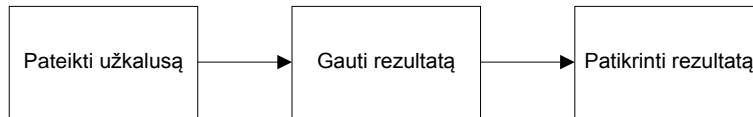
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX movies: <http://www.semanticweb.org/ontologies/2012/0/
movies.owl#>
SELECT ?Person ?Movie WHERE {
  {?Person movies:isWriterOf ?Movie}
  {?Person movies:isDirectorOf ?Movie}
  {?Movie movies:hasActor movies:Brad_Pitt} UNION
  {?Movie movies:hasActor 'Brad_Pitt'^^xsd:dateTime} UNION
  {?Movie movies:hasActor 'Brad_Pitt'^^xsd:double} UNION
  {?Movie movies:hasActor 'Brad_Pitt'^^xsd:integer} UNION
  {?Movie movies:hasActor 'Brad_Pitt'^^xsd:string}
}

```

4.21. pav. Atnaujinta SPARQL užklausa

Vartotojas gali pridėti neribotą kiekį vartotojo sąsajos elementų ir keisti jų įrašus. Pakeitus pirmus elementus eilutėje, sekančių elementų patarimo sąrašai generuojami iš naujo.

4.4. Testavimo modelis ir duomenys



4.22. pav. Testavimo modelis

Pagal testavimo modelį sistemai bus paduodama užklausa ir pagal gautą rezultatą ir ontologijoje esančius duomenis tikrinamas rezultato teisingumas.

SPARQL užklauskos sistemai testuoti:

1. Užklausa išrenkanti visas klases ontologijoje:

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX movies: <http://www.semanticweb.org/ontologies/2012/0/movies.owl#>

SELECT ?class

WHERE { ?class rdf:type owl:Class. }

Laukiamas rezultatas:

<http://www.semanticweb.org/ontologies/2012/0/movies.owl#AlternativeGenreName>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#Person>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#Language>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#Country>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#AlternativeObjectName>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#Movie>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#Genre>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#Company>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#Character>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#AlternativeMovieName>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#AlternativeGenreName>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#Person>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#Language>

<http://www.semanticweb.org/ontologies/2012/0/movies.owl#Country>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#AlternativeObjectName>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#Movie>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#Genre>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#Company>

2. Užklausa išrenkanti visus objektų ryšius ontologijoje:

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX movies: <http://www.semanticweb.org/ontologies/2012/0/movies.owl#>

SELECT ?class

WHERE { ?class rdf:type owl:ObjectProperty. }

Laukiamas rezultatas:

<http://www.semanticweb.org/ontologies/2012/0/movies.owl#isGenreOf>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#isAlternativeNameOfMovie>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#hasAlternativeGenreName>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#hasAlternativeMovieName>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#isWriterOf>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#hasWriter>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#isProducerOf>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#isAlternativeNameOfGenre>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#hasDirector>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#hasBasedCompany>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#isPlaying>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#hasActor>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#isInMovie>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#isReleasedInLanguage>
<http://www.semanticweb.org/ontologies/2012/0/movies.owl#isMainLanguageIn>

5. Eksperimentinis sukurto semantinių užklausų modelio tyrimas ir įvertinimas

5.1. Užklausų kokybinių parametrų eksperimentinis tyrimas

Šio eksperimento tikslas yra įvertinti sukurto dinaminio semantinių užklausų formavimo vartotojo sąsajos modelio efektyvumą tikslumo ir gražinimo parametrų atžvilgiu vartotojo požiūriu interneto aplinkoje.

Eksperimento parametrais pasirinkti analizės dalyje minėti gražinimas ir tikslumas (2.4.2.5. skyrius). Kadangi gražinimui nustatyti reikia informacijos apie visus portalo ontologijoje esančius įrašus, gražinimo parametras bus matuojamas tik prototipo užklausoms.

Tikslumo parametras bus matuojamas gautų užklausą tenkinančių rezultatų kiekį dalinant iš visų gautų rezultatų. Kadangi portalai dažniausiai gražina didelį kiekį rezultatų puslapių, tikslumas bus skaičiuojamas tik pirmam rezultatų puslapiui.

Rezultatai gali būti paskirstyti dviem būdais. Pirmu atveju tinkami ir netinkami rezultatai yra nerikiuoti ir tolygiai pasiskirstę visuose rezultatų puslapiuose, kas leidžia gana tiksliai nustatyti sistemos tikslumą. Antru atveju rezultatai išrikiuoti pagal tinkamumą pirmiausia pateikiant tinkamiausius rezultatus (į galimybę, jog portalas rikiuoja rezultatus, pirmiausia pateikdamas mažiausiai tinkamus neatsižvelgiama). Abiem atvejais pirmas rezultatų puslapis leidžia įvertinti didžiausią sistemos tikslumą.

Matuojant prototipo tikslumą, rezultatai pateikiami naudojant sukurtą kino ontologiją, bei pavyzdinę vynų ontologiją.

Sistemoms bus pateikiamos dviejų tipų užklausos:

1. Objektų su nustatyta savybe radimo užklausa.
2. Konkretaus objekto radimo užklausa.

5.1.1. Pirmo tipo užklausa

Search text

For a simple text search, enter your search term here. Multiple words may be found by combining them with **AND** and **OR**. The text in this field will be matched with items' contents, title and description.

More search options

Item type ▾

Return items of a specific type.

Select All/None

Page File Folder Image ReflectoDirectory Reflecto File Reflector cf:Relationships
dc:DublinCore dc:Formats dc:Relations dc:Subjects_for_Events dc:Subjects_for_News
dc:Subjects_for_Organisations dc:Subjects_for_Patents dc:Subjects_for_Persons
dc:Subjects_for_Projects dc:Types iso:Countries iso:Languages leg:License-Types lis:BibTex-
Types It:Language_Pairs It:Linguality It:Linguistic_Approaches It:Linguistic_Areas
It:Technological_Applications Itw:Academic_Institutions Itw:Associations Itw:Blogs Itw:Calls
Itw:Commercial_Services Itw:Companies Itw:Company_Parts Itw:Concept_Container
Itw:Conferences Itw:Distribution_Agencies_for_Language_Data Itw:Events Itw:Funding_Organisations
 Itw:Funding_Programmes Itw:Glossary Itw:Groups Itw:IPR_and_Products
Itw:Information_Services Itw:LT-World-Resource-Modality-Types Itw:Language_Data
 Itw:Language_Data-Types Itw:Language_Description-Models Itw:Language_Description-Types
Itw:Language_Descriptions Itw:Language_Tool-Types Itw:Language_Tools Itw:Language_Tools_Input-
Types Itw:META Itw:Metadata_Initiatives_for_Language_Data_and_NLP_Tools Itw:Networks
Itw:News Itw:Ontology_Related_Tools Itw:Organisations Itw:Patents Itw:People
Itw:Players_and_Teams Itw:Products Itw:Projects Itw:Public_R_and_D_Administration
Itw:Publication_Repositories Itw:Publications Itw:Regional_Services Itw:Related_Services
Itw:Relevant_Sources Itw:Repositories_for_Language_Resources_or_NLP_Tools Itw:Resource-Entity-
Types Itw:Roadmap_Sources Itw:Schools Itw:Societies Itw:Standard_Initiatives Itw:Systems
Itw:Teaching_Materials Itw:Technologies Itw:Workshops Itw:extra_Domain_Organisations
Itw:extra_Domain_People meta:Event-Relationships nlsr:Currency nlsr:MainSections nlsr:Mimetypes
 tech:CompressionFormats tech:DistributionFormats tech:DistributionMedia
tech:Language_Tools_Input-Formats tech:Language_Tools_Output-Formats tech:ProgrammingLanguages

New items since ▾

Return items added since you were last logged on, the last week, etc.

Ever ▾

Author ▾

Return items created by a particular person.

Any Author ▾

Display options

Sort results

Sort the Search Results based on the selected criterion

Relevance ▾

Results per page

The number of results that is displayed on one page.

30 ▾

5.1. pav. Pirmo tipo užklausa Lt-world portale

Portale Lt-world, naudojant išplėstinę paiešką užduodama užklausa rasti su lietuvių kalba susijusius projektus į paieškos lauką įvedant tekstą „lithuanian“ ir pažymint varnelę ties savybe „Itw:Projects“ (5.1. pav.).

Pirmame rezultatų puslapyje pateikti 11 rezultatų iš kurių 6 yra projektai susiję su lietuvių kalba.

These terms define your query. Click to remove a term.

Diseases / Pathological Processes: [Virus Diseases](#) ▶ [RNA Virus Infections](#)

▶ [Picornaviridae Infections](#) ▶ [Common Cold](#)

Keyword: [treatment](#)

[show review articles only](#)

sort by: [date and relevance](#) ▼

5.2. pav. Pirmo tipo užklausa Semedico portale

Portale Semedico užduota užklausa rasti straipsnius, susijusius su peršalimo gydymu, paieškos lauke įvedant „common cold“ ir parametą „treatment“ (5.2. pav.).

Pirmame puslapyje iš gautų 10 rezultatų 5 rezultatai tenkina užklausą.

Advanced Recipe Search

enter keyword:

 all any word exact word

recipe category
 Chef Recipe Epicurious TV Kid-Friendly Quick & Easy

dietary consideration
 Healthy High Fiber Kosher Low Cal
 Low Carb Low Fat Low Sodium Low/No Sugar
 Raw Vegan Vegetarian Wheat/Gluten Free

cuisine
 ▼

meal / course
 ▼

type of dish
 ▼

season / occasion
 ▼

preparation method
 ▼

source
 ▼

main ingredient

<input type="checkbox"/> Bean	<input type="checkbox"/> Beef	<input type="checkbox"/> Berry	<input type="checkbox"/> Cheese	<input type="checkbox"/> Chicken
<input type="checkbox"/> Chocolate	<input type="checkbox"/> Citrus	<input type="checkbox"/> Coffee	<input type="checkbox"/> Duck	<input type="checkbox"/> Egg
<input type="checkbox"/> Fish	<input type="checkbox"/> Fruit	<input type="checkbox"/> Game	<input type="checkbox"/> Ginger	<input type="checkbox"/> Herb
<input type="checkbox"/> Lamb	<input type="checkbox"/> Mushroom	<input type="checkbox"/> Nut	<input type="checkbox"/> Olive	<input type="checkbox"/> Pasta
<input type="checkbox"/> Pepper	<input type="checkbox"/> Pork	<input checked="" type="checkbox"/> Potato	<input type="checkbox"/> Poultry	<input type="checkbox"/> Rice/Grain
<input type="checkbox"/> Shellfish	<input type="checkbox"/> Tea	<input type="checkbox"/> Tomato	<input type="checkbox"/> Turkey	<input type="checkbox"/> Vegetable
<input type="checkbox"/> Yogurt				

exclude from search
 Dairy Eggs Nuts Shellfish

5.3. pav. Pirmo tipo užklausa Epicurious portale

Portale Epicurious užduota užklausa rasti visus blynus „pancakes“, kurių pagrindinis ingredientas yra bulvės, pažymint elementą „Potato“ skyriuje „main ingredient“ (5.3. pav.).

Pirmame rezultatų puslapyje iš 10 rezultatų visi 10 atitiko užklausą.

Open Directory Advanced Search

simpsons voice actors

Only show results in category: Arts ▼

Search: Categories only Sites only Sites and Categories

Kids and Teens Sites: Kids Teens Mature Teens

[Help on Search](#)

[Web](#) - [AltaVista](#) - [Deja](#) - [Google](#) - [HotBot](#) - [Infoseek](#) - [Northern Light](#) - [Netscape](#) - [Yahoo](#)

5.4. pav. Pirmo tipo užklausa Dmoz portale

Portale Dmoz išplėstinėje paieškoje užduota užklausa rasti visus objektus, susijusius su serialu „Simpsons“, pridėdant parametą „voice actors“, nurodantį, kad ieškoma informacija yra susijusi su serialo balso aktoriais (5.4. pav.).

Iš 20 rezultatų pirmame puslapyje 16 rezultatų buvo susiję su užklausa.

Explore Europe's cultural collections

lithuanian

[Refine your search](#) [Help](#)

Explore Europe's cultural collections

song

[Hide Refine Search](#)

5.5. pav. Pirmo tipo užklausa Europeana portale

Portale Europeana užduota užklausa rasti lietuviškas dainas, paieškos lauke įrašant „lithuanian“ ir „refine search“ skiltyje įrašius papildomą parametą „song“.

Sistema gražina 12 rezultatų iš kurių 4 yra lietuviškos dainos.

Add subject	Add predicate	Search
Movie	hasGenre	Mistery

Movie
Fight_Chub
Donnie_Darko

5.6. pav. Pirmo tipo užklausa prototipe su kino ontologija ir užklauskos rezultatai

Add subject	Add predicate	Search
Wine	hasMaker	Corbans

Wine
CorbansDryWhiteRiesling
CorbansSauvignonBlanc
CorbansPrivateBinSauvignonBlanc

5.7. pav. Pirmo tipo užklausa prototipe su vynu ontologija ir užklauskos rezultatai

Prototipe su kino ontologija užduota užklausa rasti visiems filmams „Movie“, parametre nurodant, jog filmo žanras yra „Mystery“ (5.6. pav.).

Iš gautų 2 rezultatų abu tenkina užklausą.

Prototipe su vynu ontologija nurodyta rasti visus vynus, kurių gamintojas yra „Corbans“ (5.7. pav.).

Iš trijų gautų rezultatų visi yra „Corbans“ vynai.

Lentelė Nr.5.2. Prototipo ir paieškos portalų tikslumo palyginimas.

Portalas	Rezultatai	Tikslumas
www.Lt-world.org	6/11	0,54
www.Semedico.org	5/10	0,5
www.Epicurious.com	10/10	1
www.Dmoz.org	16/20	0,8
www.Europeana.eu	4/12	0,3
Prototipas (kino ontologija)	2/2	1
Prototipas (vynu ontologija)	3/3	1

Suskaičiavus visus gautus rezultatus ir atrinkus užklauskas tenkinančius, gaunama 5.2. lentelė, parodanti kiekvienos sistemos tikslumą.

5.1.2. Antro tipo užklausa

Search text

For a simple text search, enter your search term here. Multiple words may be found by combining them with **AND** and **OR**. The text in this field will be matched with items' contents, title and description.

More search options

Item type

Return items of a specific type.

Select All/None

Page File Folder Image ReflectoDirectory Reflecto File Reflector cf:Relationships
 dc:DublinCore dc:Formats dc:Relations dc:Subjects_for_Events dc:Subjects_for_News
 dc:Subjects_for_Organisations dc:Subjects_for_Patents dc:Subjects_for_Persons
 dc:Subjects_for_Projects dc:Types iso:Countries iso:Languages leg:License-Types lis:BibTex-Types
 It:Language_Pairs It:Linguality It:Linguistic_Approaches It:Linguistic_Areas
 It:Technological_Applications Itw:Academic_Institutions Itw:Associations Itw:Bloggs Itw:Calls
 Itw:Commercial_Services Itw:Companies Itw:Company_Parts Itw:Concept_Container
 Itw:Conferences Itw:Distribution_Agencies_for_Language_Data Itw:Events Itw:Funding_Organisations
 Itw:Funding_Programmes Itw:Glossary Itw:Groups Itw:IPR_and_Products
 Itw:Information_Services Itw:LT-World-Resource-Modality-Types Itw:Language_Data
 Itw:Language_Data-Types Itw:Language_Description-Models Itw:Language_Description-Types
 Itw:Language_Descriptions Itw:Language_Tool-Types Itw:Language_Tools Itw:Language_Tools_Input-Types
 Itw:META Itw:Metadata_Initiatives_for_Language_Data_and_NLP_Tools Itw:Networks
 Itw:News Itw:Ontology_Related_Tools Itw:Organisations Itw:Patents Itw:People
 Itw:Players_and_Teams Itw:Products Itw:Projects Itw:Public_R_and_D_Administration
 Itw:Publication_Repositories Itw:Publications Itw:Regional_Services Itw:Related_Services
 Itw:Relevant_Sources Itw:Repositories_for_Language_Resources_or_NLP_Tools Itw:Resource-Entity-Types
 Itw:Roadmap_Sources Itw:Schools Itw:Societies Itw:Standard_Initiatives Itw:Systems
 Itw:Teaching_Materials Itw:Technologies Itw:Workshops Itw:extra_Domain_Organisations
 Itw:extra_Domain_People meta:Event-Relationships nlsr:Currency nlsr:MainSections nlsr:Mimetypes
 tech:CompressionFormats tech:DistributionFormats tech:DistributionMedia
 tech:Language_Tools_Input-Formats tech:Language_Tools_Output-Formats tech:ProgrammingLanguages

New items since

Return items added since you were last logged on, the last week, etc.

Ever

Author

Return items created by a particular person.

Any Author

Display options

Sort results

Sort the Search Results based on the selected criterion

Relevance

Results per page

The number of results that is displayed on one page.

30

5.8. pav. Antro tipo užklausa Lt-world portale

Portale Lt-world, naudojant išplėstinę paiešką užduodama užklausa rasti projektą „Survey of lithuanian language technical applications“ į paieškos lauką įvedant tekstą „survey of language lithuanian technical applications“ ir pažymint varnelę ties savybe „Itw:Projects“ (5.8. pav.).

Pirmame rezultatų puslapyje pateikti 5 rezultatų iš kurių 1 yra ieškomas projektas.

semedico
search PubMed beta

Green tea attenuates cardiovascular remodelling symptoms search

Select a term to search for synonyms and related terms. Press ESC to ignore suggestions.

Result 1-10 of 3812 (142 ms)

BioMed Immunology Bibliography

Chemicals and Drugs (3775)

- Heterocyclic Compounds (2914)
- Complex Mixtures (2662)
- Chemical Actions and Uses (2457)
- more..

Investigative Techniques (2931)

- Epidemiologic Methods (1249)
- Chemistry, Analytical (816)
- Diet (507)
- more..

Diseases / Pathological Processes (2523)

These terms define your query. Click to remove a term.

Green is a synonym for 25 terms, [refine](#)

Chemicals and Drugs: [Complex Mixtures](#) [Biological Products](#) [Plant Preparations](#) [Tea](#)

Keyword: attenuates

Keyword: cardiovascular

Keyword: remodelling

Keyword: symptoms

[show review articles only](#) sort by: [date and relevance](#)

5.9. pav. Antro tipo užklausa Semedico portale

Portale Semedico užduota užklausa rasti straipsnį „Green tea attenuates cardiovascular remodelling and metabolic symptoms in high carbohydrate-fed rats“ paieškos lauke įvedant „Green tea attenuates cardiovascular remodelling symptoms“ (5.9. pav.). Kadangi paieškos lauko ilgis yra apribotas, pateikiant užklausą negalima įvesti pilno straipsnio pavadinimo, įrašius straipsnio pavadinimo pradžią, sistema neranda ieškomo straipsnio. Rezultatų vertinimui pasirinktas užklaunos variantas, grąžinantis ieškomą rezultatą.

Pirmame puslapyje iš gautų 10 rezultatų 1 rezultatas tenkina užklausą.

Advanced Recipe Search

enter keyword:

Rib-Eye Steaks with Gorgonzola Butter and Crispy Sweet Onion Rings

all any word exact word

recipe category ?

Chef Recipe Epicurious TV Kid-Friendly Quick & Easy

dietary consideration ?

Healthy High Fiber Kosher Low Cal
 Low Carb Low Fat Low Sodium Low/No Sugar
 Raw Vegan Vegetarian Wheat/Gluten Free

cuisine

all cuisines ▼

meal / course

all courses ▼

type of dish

all types ▼

season / occasion

all occasions ▼

preparation method

all methods ▼

source

all sources ▼

main ingredient

Bean Beef Berry Cheese Chicken
 Chocolate Citrus Coffee Duck Egg
 Fish Fruit Game Ginger Herb
 Lamb Mushroom Nut Olive Pasta
 Pepper Pork Potato Poultry Rice/Grain
 Shellfish Tea Tomato Turkey Vegetable
 Yogurt

exclude from search ?

Dairy Eggs Nuts Shellfish

submit

5.10. pav. Antro tipo užklausa Epicurious portale

Portale Epicurious užduota užklausa rasti patiekalą „Rib-Eye Steaks with Gorgonzola Butter and Crispy Sweet Onion Rings“. Detalios paieškos formoje į paieškos lauką įvestas tikslus patiekalo pavadinimas ir kaip pagrindinis ingredientas pažymėta savybė „Beef“ (5.10. pav.).

Pirmame rezultatų puslapyje tarp 10 pateiktų rezultatų nebuvo ieškomo patiekalo.

Open Directory Advanced Search

prometheus movie trailer

Only show results in category: Arts

Search: Categories only Sites only Sites and Categories

Kids and Teens Sites: Kids Teens Mature Teens

[Help on Search](#)

5.11. pav. Antro tipo užklausa Dmoz portale

Portale Dmoz išplėstinėje paieškoje užduota užklausa rasti reklaminių filmo klipą „prometheus movie trailer“, pažymint „Arts“ kategoriją (5.11. pav.).

Iš 20 rezultatų pirmame puslapyje 2 rezultatai tenkino užklausą.

Explore Europe's cultural collections

Gioconda

[Refine your search](#) [Help](#)

5.12. pav. Antro tipo užklausa Europeana portale

Portale Europeana užduota užklausa rasti paveikslą „Gioconda“, „refine search“ skiltyje įrašius papildomą parametą „portrait“ (5.12. pav.).

Sistema gražina vieną rezultatą – ieškomą paveikslą.

Add subject	Add predicate	Search
Person	isBornInCountry	United_States
	hasBirthDate	1928-12-16T00:00:00

Person

Philip_K_Dick

5.13. pav. Antro tipo užklausa prototipe su kino ontologija ir užklaustos rezultatai

Prototipe konkretaus objekto paieška atliekama didinant užklaustos parametru kieki, taip susiaurinant rezultatus iki norimo tikslumo.

Nurodžius, kad ieškoma asmens ir pateikus jo gimimo metus, bei šalį, atrandamas konkretus ieškomas asmuo (5.13. pav.).

Iš gautų 2 rezultatų abu tenkina užklausą.

5.14. pav. Antro tipo užklausa prototipe su vynu ontologija ir užklaustos rezultatai

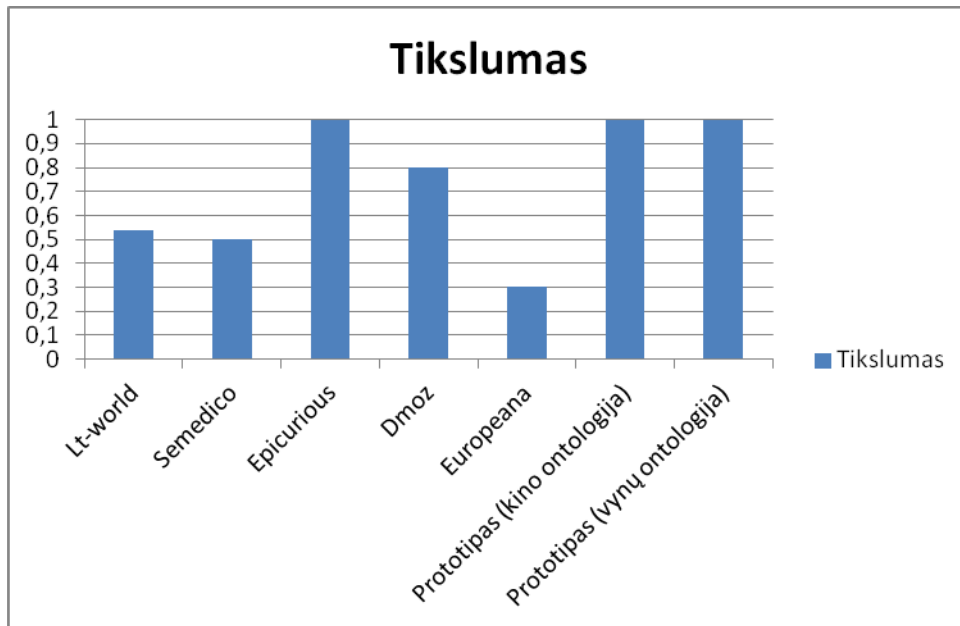
Prototipe nurodžius, kad ieškoma vyno ir nurodžius jo „hasSugar“ ir „hasBody“ savybes, atrandamas konkretus ieškomas vinas (5.14. pav.).

Lentelė Nr.5.3. Prototipo ir paieškos portalų tikslumo palyginimas.

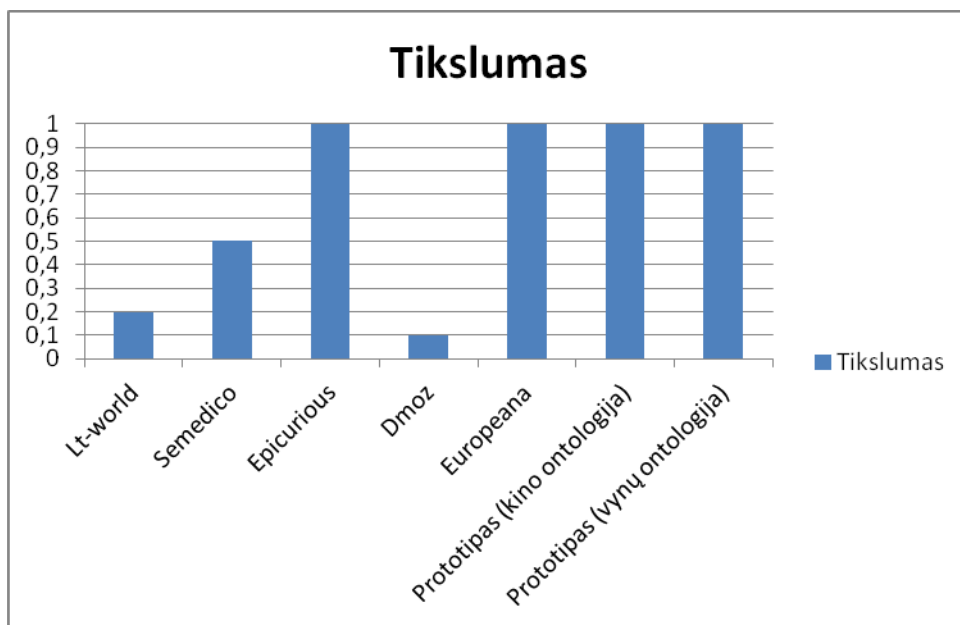
Portalas	Rezultatai	Tikslumas
www.Lt-world.org	1/5	0,2
www.Semedico.org	1/10	0,5
www.Epicurious.com	0/10	1
www.Dmoz.org	2/20	0,1
www.Europeana.eu	1/1	1
Prototipas (kino ontologija)	1/1	1
Prototipas (vynų ontologija)	1/1	1

Suskaičiavus visus gautus rezultatus ir atrinkus užklausas tenkinančius, gaunama 5.3. lentelė, parodanti kiekvienos sistemos tikslumą.

5.1.3. Eksperimento rezultatai



5.15. pav. Pirmo užklauso tipo rezultatų histograma



5.16. pav. Antro užklauso tipo rezultatų histograma

Gauti eksperimento rezultatai 5.15. ir 5.16 paveiksluose parodo, jog prototipas tikslumu lenkia kitas sistemas arba joms prilygsta atliekant objektų su nustatyta savybe ir konkrečių objektų paiešką naudojant abi eksperimente naudotas ontologijas.

5.2. Užklausų formavimo ir paieškos proceso įvertinimas

Šiame skyriuje siekiama išanalizuoti sukurtą prototipą lyginant jo realizuotų funkcijų kiekį kitų sistemų atžvilgiu (5.1. lentelė).

Lentelė Nr.5.1. Prototipo ir paieškos portalų funkcijų palyginimas skirtinguose paieškos etapuose.

Etapas	Savybė	Lt-world	Europeana	Dmoz	Semedico	Epicurious	Prototipas
Užklausos sudarymas	Teksto įvedimas	+	+	+	+	+	+
	Operatoriai	+	+	+	-	+	+
	Reakcija į vartotojo veiksmus	-	+	-	+	-	+
Paieškos algoritmas	Sintaksės palyginimas	+	+	+	+	+	+
Rezultatų pateikimas	Reakcija į vartotojo veiksmus	-	+	-	+	+	+
Sąveika su rezultatais ir užklausos tikslinimas	Užklausos teisingumo palaikymas (angl. <i>truth maintenance</i>)	+	+	+	-	-	+

Užklausos sudarymo etape, kaip ir kitos sistemos, sukurtas prototipas turi teksto įvedimo savybę. Prototipas naudoja operatorių „?“ , kuris pažymi, jog atitinkamame lauke įvestas SPARQL kintamasis. Reakcija į vartotojo veiksmus įgyvendinta panaudojant patarimų sąrašą, generuojamą kiekvienam vartotojo sąsajos elementui, programos veikimo metu.

Paieškos algoritmo veikimo etape prototipas naudoja sintaksės palyginimą, vartotojo įvestiems SPARQL kintamiesiems atpažinti pagal žodžio pradžią.

Rezultatų pateikimo etape prototipas leidžia vartotojui tikslinti užklausą. Tokios savybės kaip rikiavimas ar grupavimas nebuvo realizuotos, nes jos mažai įtakoja sistemos veikimą ir nėra būtinos.

Sąveikos su rezultatais ir užklauso tikslinimo etape prototipo užklauso teisingumo palaikymo savybe įgyvendinta leidžiant vartotojui valdyti prototipo vartotojo sąsajos ir užklauso formavimo algoritmus, pridėdant naujų vartotojo sąsajos elementų.

6. Išvados

1. Atlikta semantinės paieškos portalų analizė parodė, kad:
 - tikslius užklausų rezultatus leidžia gauti tik tie portalai, kurie naudoja ontologijas ir turi didelį vartotojo sąsajos elementų skaičių, o portalai, turintys mažai vartotojo sąsajos elementų, gražina netikslius rezultatus;
 - užklausos formavimas galėtų vykti interaktyviai, pateikiant vartotojui naujus komponentus priklausomai nuo vartotojo veiksmų.

Todėl buvo nutarta kurti dinaminę paieškos sąsają, kuri suteiktų norimą tikslumo lygį, leisdama vartotojui dinamiškai formuoti užklausą vartotojo sąsajoje.

2. Sistemos testavimas parodė, kad sudarytas dinaminės paieškos sąsajos modelis ir formavimo algoritmai atitinka dinaminės sąsajos reikalavimus: padeda vartotojui suformuoti užklausą, pateikia paieškos rezultatus ir leidžia redaguoti užklausą.
3. Eksperimentinis paieškos sistemos, realizuotos taikant Java, SPARQL, OWL, Javascript ir *Protégé* ir Apache web server technologijas, išbandymas su kino ir vynu ontologijomis parodė, kad sistema leidžia dinamiškai formuoti vartotojo sąsają, gražina tikslius užklausų rezultatus ir gali veikti su įvairiomis ontologijomis, kurių neprieštaringumas patikrintas ontologijų išvedimo mechanizmu (angl. *Reasoner*).
4. Sukurtos užklausų formavimo sistemos užklausų rezultatų kokybinių rodiklių (tikslumo ir gražinimo) tyrimas parodė, kad sistema gražina rezultatus, tolygius arba geresnius, negu tyrime nagrinėti žinomi semantinės paieškos portalai.
5. Sukurtą dinaminę paieškos vartotojo sąsajos modelį ir algoritmus galima taikyti semantinės paieškos portalams kurti.
6. Darbo rezultatai buvo pristatyti XVII tarpuniversitetinėje magistrantų ir doktorantų konferencijoje ir išspausdintas straipsnis leidinyje „Informacinės technologijos 2012“.

7. Literatūra

- [1] Noy, N.F., McGuinness, D.L. Ontology Development 101: A Guide to Creating Your First Ontology, 2001, [žiūrėta 2010-11-10] Prieiga per internetą: <http://www.ling.helsinki.fi/kit/2004k/ct1310semw/Protege/WineTutorial.pdf>
- [2] Guan, J., Zhang, X., Deng, J. Qu, Y., An Ontology-Driven Information Retrieval Mechanism for Semantic Information Portals, 2005, [žiūrėta 2010-11-10] http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4125851
- [3] Mangold, C., A survey and classification of semantic search approaches // Int. J. Metadata, Semantics and Ontology, Vol. 2, No. 1, p.23-34.
- [4] Hartmann, J., Sure, Y., An Infrastructure for Scalable, Reliable Semantic Portals // Intelligent Systems, IEEE. – 2004, Vol. 19, No. 3, p. 58-65.
- [5] Semantic Search Survey. [žiūrėta 2010-11-16] Prieiga per internetą: http://swuiwiki.webscience.org/index.php/Semantic_Search_Survey
- [6] Hakia. [žiūrėta 2010-11-11] Prieiga per internetą: <http://www.hakia.com>
- [7] Lt-world. [žiūrėta 2010-11-11] Prieiga per internetą: <http://www.lt-world.org>
- [8] Semedico. [žiūrėta 2010-11-11] Prieiga per internetą: <http://www.semedico.org>
- [9] Europeana. [žiūrėta 2010-11-11] Prieiga per internetą: <http://www.europeana.eu>
- [10] Epicurious. [žiūrėta 2010-11-11] Prieiga per internetą: <http://www.epicurious.com>
- [11] Kosmix. [žiūrėta 2010-11-11] Prieiga per internetą: <http://www.kosmix.com>
- [12] Dmoz open directory project. [žiūrėta 2010-11-11] Prieiga per internetą: <http://www.dmoz.org>
- [13] Exalead. [žiūrėta 2010-11-11] Prieiga per internetą: <http://www.exalead.com/search>
- [14] SenseBot. [žiūrėta 2010-11-11] Prieiga per internetą: <http://www.sensebot.net>
- [15] Cognition. [žiūrėta 2010-11-11] Prieiga per internetą: <http://www.cognition.com>
- [16] Swoogle. [žiūrėta 2010-11-11] Prieiga per internetą: <http://swoogle.umbc.edu>
- [17] Bing. [žiūrėta 2010-11-11] Prieiga per internetą: <http://www.bing.com>
- [18] Semantic web development tools. Prieiga per internetą: <http://www.w3.org/2001/sw/wiki/SemanticWebTools>
- [19] OntoStudio. [žiūrėta 2010-11-13] Prieiga per internetą: <http://www.ontoprise.de/en/home/products/ontostudio>
- [20] Protégé. [žiūrėta 2010-11-13] Prieiga per internetą: <http://protege.stanford.edu>

- [21] Model Futures OWL Editor. [žiūrėta 2010-11-13] Prieiga per internetą:
<http://www.modelfutures.com/owl>
- [22] Knoodl. [žiūrėta 2010-11-13] Prieiga per internetą: <http://knoodl.com>
- [23] SemWeb. [žiūrėta 2010-11-13] Prieiga per internetą:
<http://razor.occams.info/code/semweb>
- [24] Allegro Graph RDF Store. [žiūrėta 2010-11-13] Prieiga per internetą:
<http://www.franz.com/agraph/allegrograph>
- [25] Sesame. [žiūrėta 2010-11-13] Prieiga per internetą: <http://www.openrdf.org>
- [26] EasyRDF. [žiūrėta 2010-11-13] Prieiga per internetą:
<http://www.aelius.com/njh/easyrdf>
- [27] Jena, a Java RDF API and toolkit. [žiūrėta 2010-11-13] Prieiga per internetą:
<http://openjena.org>
- [28] Ying, D., Sun, Y., Chen, Bin., Börner, K., Ding, L., Wild, D., Wu, M., DiFranzo, D., Fuenzalida, A., G., Li, D., Milojević, S., Chen, S., Sankarangarayanan, M., Toma, I., Semantic Web Portal: A Platform for Better Browsing and Visualizing Semantic Data // Proceedings of the 2010 International Conference on Active Media Technology [Toronto, Canada, August 28-30].
- [29] Semantic search benchmark. [žiūrėta 2010-12-12] Prieiga per internetą:
<http://wiki.iks-project.eu/index.php/Semantic-search-benchmark>
- [30] Content-Based Image Retrieval Query Paradigms. [žiūrėta 2010-12-12] Prieiga per internetą:
<http://www.the-crankshaft.info/2010/07/content-based-image-retrieval-query.html>
- [31] Ambrus, O., Möller, K., Handschuh, S. Konduit VQB: a Visual Query Builder for SPARQL on the Social Semantic Desktop // Workshop on Visual Interfaces to the Social and Semantic Web, 2005, Vol. 1, p. 5.
<http://ceur-ws.org/Vol-565/paper4.pdf>
- [32] Hyvönen, E., Viljanen, K., Suominen, O., Hukka, E., HealthFinland—Publishing Health Promotion Information on the Semantic Web