



Kauno technologijos universitetas

Informatikos fakultetas

Natūraliosios kalbos sąsaja ontologijoms

Baigiamasis magistro projektas

Ugnė Valčiukaitė

Projekto autorė

Lekt. Algirdas Šukys

Vadovas

Kaunas, 2022



Kauno technologijos universitetas

Informatikos fakultetas

Natūraliosios kalbos sąsaja ontologijoms

Baigiamasis magistro projektas

Informacinių sistemų inžinerija (6211BX009)

Ugnė Valčiukaitė

Projekto autorė

Lekt. Algirdas Šukys

Vadovas

Lekt. Jaroslav Karpovič

Recenzentas

Kaunas, 2022



Kauno technologijos universitetas

Informatikos fakultetas

Ugnė Valčiukaitė

Natūraliosios kalbos sąsaja ontologijoms

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;
2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąrašė;
3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);
4. suprantu, kad išaiškėjus nesažiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Ugnė Valčiukaitė

Patvirtinta elektroniniu būdu

Valčiukaitė Ugnė. Natūraliosios kalbos sąsaja ontologijoms. Magistro projektas / vadovas lekt. Algirdas Šukys; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informacinių sistemų inžinerija, Informatikos mokslai.

Reikšminiai žodžiai: natūraliosios kalbos sąsaja, ontologija, žodynas, objektinė savybė, duomenų savybė, klasė, tripletų šablonas.

Kaunas, 2022. 90 p.

Santrauka

Kasdien didėjant saugomų duomenų kiekiui tiek internete, tiek ir organizacijose, naudotojui informacijos paieškos procesas tampa vis sudėtingesnis. Tradicinės, raktiniais žodžiais grindžiamos, paieškos sistemos negali suprasti pateikiamų duomenų semantikos ir pateikti pakankamai tikslius atsakymus į naudotojo užklausas. Siekiant paiešką padaryti tikslesnę, naudojamos semantinės technologijos ir paieška. Semantinė paieška vykdoma ontologijose, kuriose apibrėžiamos tam tikros dalykinės srities sąvokos, esybės, jų tipai, tarpusavio ryšiai, priklausomybės ir kt. Nors ontologijų tipų yra įvairių, šiame darbe koncentruojamasi į taikomąsias ontologijas, skirtas konkrečios dalykinės srities duomenims saugoti. Kuriant semantinės paieškos sistemas siekiama sukurti naudotojo sąsajas, kurios būtų patogios ir lengvai suprantamos naudotojams. Viena iš tokių yra natūraliosios kalbos sąsaja. Šiame darbe buvo apsibrėžti pagrindiniai natūraliosios kalbos sąsajos ontologijoms reikalavimai: pritaikymas lietuvių kalbai ir žodyno paruošimas įdedant kuo mažiau rankinio darbo. Šio tiriamojo projekto metu buvo nagrinėjamos jau sukurtos natūraliosios kalbos sąsajos ir nustatyta, kad nei vienoje išanalizuotoje naudotojo sąsajoje nėra realizuoti visi tiriamojo projekto metu nustatyti reikalavimai. Nustatyta, kad vienintelė *SBVR* pagrįsta NKS yra pritaikoma lietuvių kalbai, tačiau jai reikalingas papildomas žodyno paruošimas, kuris reikalauja nemažai rankinio darbo. Dėl šios priežasties yra poreikis sukurti naują sprendimą, kuris leistų naudotojui vykdyti semantinę paiešką ontologijoje lietuvių kalba. Atlikus būsimos natūraliosios kalbos sąsajos reikalavimų analizę ir specifikaciją buvo formaliai aprašytas metodas, skirtas analizuoti klausimus ir juos transformuoti į užklausas. Metodu patikrinti buvo realizuotas natūraliosios kalbos sąsajos prototipas, kurio veikimas buvo ištirtas atlikus eksperimentą. Atliktas eksperimentas padėjo nustatyti, kad metodas sudaro sąlygas vykdyti semantinę paiešką pateikiant klausimus lietuvių kalba. Nors atsakymų tikslumas yra vidutiniškas, plečiant metodo galimybes yra įmanoma jį padidinti.

Valčiukaitė Ugnė. Natural Language Interface for Ontologies. Master's Final Degree Project / supervisor lect. Algirdas Šukys; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Information Systems Engineering, Computing.

Keywords: ontology, vocabulary, object property, data property, class, triple pattern.

Kaunas, 2022. 90.

Summary

As the amount of data stored, both online and in organizations, increases, the process of searching for information has become very difficult for the user. Traditional, keyword-based search engines cannot understand the semantics of the data presented and provide sufficiently accurate answers to user queries. Semantic technology and search are used to make the search more accurate. Semantic search is performed in ontologies, which define the concepts of certain subject areas, entities, their types, interrelationships, dependencies, and so on. Although there are different types of ontologies, this work focuses on applied ontologies for storage data in a specific subject area. The development of semantic search engines is also greatly influenced by the user interface, which must be not only functional, but also user-friendly and easy to understand for the system user. In this work, the main requirements of the natural language interface for ontologies were defined: adaptation to the Lithuanian language and preparation of the dictionary with as little manual work as possible. During this research project, the already developed natural language interfaces were examined and it was found that none of the analyzed user interfaces fulfilled all the requirements established during the research project. It has been established that the only *SBVR* based NKS is applicable to the Lithuanian language, but it requires additional dictionary preparation, which requires a lot of manual work. For this reason, there is a need to create a new solution that would allow the user to perform a semantic search in the ontology in Lithuanian. An analysis and specification of the requirements for a future natural language interface has formally described a method for analyzing questions and transforming them into queries. To test the method, a prototype of a natural language interface was realized, the operation of which was investigated in an experiment. The performed experiment helped to determine that the method allows to perform semantic search by asking questions in Lithuanian language. Although the accuracy of the answers is mediocre, it is possible to increase it by expanding the capabilities of the method.

Turinys

Lentelių sąrašas	8
Paveikslų sąrašas	10
Santrumpų ir terminų sąrašas	12
Įvadas	14
1. Probleminės srities analizė	16
1.1. Analizės tikslas	16
1.2. Tyrimo objektas, sritis ir problema	16
1.3. Tyrimo objekto analizė	16
1.3.1. Semantinio tinklo apibrėžimas	16
1.3.2. Ontologijos apibrėžimas	18
1.3.3. Ontologijų aprašymo kalbos	20
1.3.4. Semantinio tinklo technologijos	23
1.3.5. Semantinės paieškos apibrėžimas	25
1.3.6. Natūraliosios kalbos sąsajos apibrėžimas	27
1.4. Tyrimo objekto naudotojų analizė	32
1.5. Esamų problemos sprendimo metodų analizė	32
1.6. Siekiamo sprendimo apibrėžimas	35
1.7. Analizės išvados	35
2. Natūraliosios kalbos sąsajos ontologijoms reikalavimų specifikacija ir projektas	36
2.1. Natūraliosios kalbos sąsajos ontologijoms reikalavimų specifikacija	36
2.1.1. Funkciniai reikalavimai	36
2.1.2. Nefunkciniai reikalavimai	41
2.2. Natūraliosios kalbos sąsajos ontologijoms dalykinės srities modelis	42
2.2.1. Natūraliosios kalbos sąsajos ontologijoms dalykinės srities esybių klasių diagrama	42
2.2.2. Natūraliosios kalbos sąsajos ontologijoms egzempliorių diagrama	42
2.3. Natūraliosios kalbos sąsajos ontologijoms naudotojų sąsajos modelis	43
2.4. Formalus natūraliosios kalbos sąsajos ontologijoms aprašas	44
2.4.1. Klausimo analizavimas	45
2.4.2. Klausimo transformavimas	48
2.4.3. Ontologijos kūrimo taisyklės	60
2.4.4. Klausimų sudarymo taisyklės	60
2.5. Reikalavimų apibendrinimas	60
3. Natūraliosios kalbos sąsajos ontologijoms eksperimentinės realizacijos projektas	62
3.1. Natūraliosios kalbos sąsajos ontologijoms loginė architektūra	62
3.2. Natūraliosios kalbos sąsajos ontologijoms kompiuterizuojamų panaudojimo atvejų realizacija seku diagramomis	62
3.3. Natūraliosios kalbos sąsajos ontologijoms realizavimo ir diegimo modeliai	65
4. Natūraliosios kalbos sąsajos ontologijoms realizacija ir testavimas	66
4.1. Natūraliosios kalbos sąsajos ontologijoms realizacijos ir veikimo aprašymas	66
4.1.1. Klausimo formulavimas	66
4.1.2. Klausimo analizavimas	67
4.1.3. Klausimo transformavimas	68
4.2. Testavimo modelis, duomenys, rezultatai	70
5. Eksperimentinis natūraliosios kalbos sąsajos ontologijoms tyrimas	72
5.1. Eksperimento planas	72

5.2. Eksperimento vykdymas	72
5.2.1. Ontologijos kūrimas.....	72
5.2.2. Klausimų formulavimas ir paieškos vykdymas	73
5.3. Eksperimento rezultatų analizė	86
5.4. Eksperimento išvados	87
5.5. Natūraliosios kalbos sąsajos ontologijoms veikimo ir savybių analizė	87
5.6. Natūraliosios kalbos sąsajos ontologijoms taikymo rekomendacijos	87
Išvados.....	88
Literatūros sąrašas.....	89

Lentelių sąrašas

1 lentelė. Ontologijų užklausų vykdymo lyginamoji analizė	25
2 lentelė. Natūraliosios kalbos sąsajų lyginamoji analizė	34
3 lentelė. Panaudojimo atvejo „Vykdyti paiešką“ specifikacija.....	37
4 lentelė. Panaudojimo atvejo „Įvesti ir tikrinti klausimą“ specifikacija	38
5 lentelė. Panaudojimo atvejo „Gauti žodyną iš ontologijos“ specifikacija.....	39
6 lentelė. Panaudojimo atvejo „Analizuoti klausimą“ specifikacija.....	40
7 lentelė. Panaudojimo atvejo „Transformuoti klausimą į užklausą“ specifikacija	41
8 lentelė. Neatpažinto žodžio testavimo scenarijus	70
9 lentelė. Gramatinės klaidos testavimo scenarijus	70
10 lentelė. Teisingo klausimo (be patikslinamojo dialogo) testavimo scenarijus	70
11 lentelė. Teisingo klausimo (su patikslinamuoju dialogu) testavimo scenarijus	71
12 lentelė. Teisingo klausimo (naudojant sinonimus) testavimo scenarijus	71
13 lentelė. Klausimų, remiantis viena role be apribojimų, paieška (Elektroninė prekyba).....	74
14 lentelė. Klausimų, remiantis viena role be apribojimų, neteisingų rezultatų vertinimas (Elektroninė prekyba)	74
15 lentelė. Klausimų, remiantis dviem rolėm be apribojimų, paieška (Elektroninė prekyba)	75
16 lentelė. Klausimų, remiantis dviem rolėm be apribojimų, neteisingų rezultatų vertinimas (Elektroninė prekyba)	75
17 lentelė. Klausimų, naudojant agregavimo operatorių remiantis viena role, paieška (Elektroninė prekyba)	76
18 lentelė. Klausimų, naudojant agregavimo operatorių remiantis viena role, neteisingų rezultatų vertinimas (Elektroninė prekyba).....	76
19 lentelė. Klausimų, naudojant agregavimo operatorių remiantis dviem rolėm, paieška (Elektroninė prekyba)	77
20 lentelė. Klausimų, naudojant agregavimo operatorių remiantis dviem rolėm, neteisingų rezultatų vertinimas (Elektroninė prekyba).....	78
21 lentelė. Klausimų, naudojant aritmetinį operatorių, paieška (Elektroninė prekyba).....	78
22 lentelė. Klausimų, naudojant aritmetinį operatorių, neteisingų rezultatų vertinimas (Elektroninė prekyba)	79
23 lentelė. Klausimų, kurių rolė apribota konkrečiu egzemplioriumi, paieška (Elektroninė prekyba)	79
24 lentelė. Klausimų, kurių rolė apribota konkrečiu egzemplioriumi, neteisingų rezultatų vertinimas (Elektroninė prekyba)	80
25 lentelė. Klausimų, remiantis viena role be apribojimų, paieška (Universiteto veikla).....	80
26 lentelė. Klausimų, remiantis viena role be apribojimų, neteisingų rezultatų vertinimas (Universiteto veikla).....	81
27 lentelė. Klausimų, remiantis dviem rolėm be apribojimų, paieška (Universiteto veikla)	81
28 lentelė. Klausimų, remiantis dviem rolėm be apribojimų, neteisingų rezultatų vertinimas (Universiteto veikla)	82
29 lentelė. Klausimų, naudojant agregavimo operatorių remiantis viena role, paieška (Universiteto veikla).....	82
30 lentelė. Klausimų, naudojant agregavimo operatorių remiantis viena role, neteisingų rezultatų vertinimas (Universiteto veikla).....	82
31 lentelė. Klausimų, naudojant agregavimo operatorių remiantis dviem rolėm, paieška (Universiteto veikla).....	83

32 lentelė. Klausimų, naudojant agregavimo operatorių remiantis dviem rolėm, neteisingų rezultatų vertinimas (Universiteto veikla).....	84
33 lentelė. Klausimų, naudojant aritmetinį operatorių, paieška (Universiteto veikla).....	84
34 lentelė. Klausimų, naudojant aritmetinį operatorių, neteisingų rezultatų vertinimas (Universiteto veikla).....	85
35 lentelė. Klausimų, kurių rolė apribota konkrečiu egzemplioriumi, paieška (Universiteto veikla).	85
36 lentelė. Klausimų, kurių rolė apribota konkrečiu egzemplioriumi, neteisingų rezultatų vertinimas (Universiteto veikla)	86
37 lentelė. Eksperimento rezultatai	86

Paveikslų sąrašas

1 pav. Semantinio tinklo struktūrinė architektūra	17
2 pav. Grafinis ontologijos pavyzdys	19
3 pav. RDF grafo pavyzdys	21
4 pav. OWL ontologijos pavyzdys	22
5 pav. <i>SPARQL</i> užklauso struktūra	23
6 pav. Saugomos duomenų struktūros pavyzdys.....	30
7 pav. <i>SPARQL</i> duomenų išraiškos pavyzdys	31
8 pav. <i>SBVR</i> pagrįstos NKS komponentai.....	34
9 pav. Natūraliosios kalbos sąsajos ontologijoms panaudojimo atvejų diagrama.....	36
10 pav. Panaudojimo atvejo „Vykdėti paiešką“ veiklos diagrama	38
11 pav. Panaudojimo atvejo „Įvesti ir tikrinti klausimą“ veiklos diagrama.....	39
12 pav. Panaudojimo atvejo „Gauti žodyną iš ontologijos“ veiklos diagrama	40
13 pav. Natūraliosios kalbos sąsajos ontologijoms sprendimo dalykinės srities esybių klasių modelis	42
14 pav. Egzempliorių specifikacijos diagrama.....	43
15 pav. Pagrindinis sąsajos langas.....	43
16 pav. Patikslinamojo dialogo langas	44
17 pav. Rezultatų langas.....	44
18 pav. Panaudojimo atvejo „Analizuoti paiešką“ veiklos diagrama.....	45
19 pav. Veiklos „Susieti klausimą su žodynu“ veiklos diagrama	46
20 pav. Veiklos „Susieti klausimą su operatoriumi“ veiklos diagrama	47
21 pav. Taisyklių rinkinys klausimo tipo nustatymui.....	48
22 pav. Taisyklių rinkinys klausimo, kuris remiasi viena role be apribojimų, nustatymui.....	49
23 pav. Klausimo, kuris remiasi viena klase be apribojimų, transformavimo algoritmas	49
24 pav. <i>SPARQL</i> užklauso klausimo, kuris remiasi viena klase be apribojimų, transformavimui.....	50
25 pav. Klausimo, kuris remiasi viena duomenų savybe be apribojimų, transformavimo algoritmas	50
26 pav. <i>SPARQL</i> užklauso klausimo, kuris remiasi viena duomenų savybe be apribojimų, transformavimui	50
27 pav. Taisyklių rinkinys klausimo, kuris remiasi dviem rolėm be apribojimų, nustatymui	51
28 pav. Klausimo, kuris remiasi objektine savybe be apribojimų, transformavimo algoritmas	51
29 pav. <i>SPARQL</i> užklauso klausimo, kuris remiasi dviem rolėm be apribojimų, transformavimui ...	52
30 pav. Taisyklių rinkinys klausimo, kuris naudoja agregavimo operatorių ir remiasi viena role, nustatymui.....	52
31 pav. Klausimo, kuris naudoja agregavimo operatorių remiantis viena klase, transformavimo algoritmas.....	53
32 pav. <i>SPARQL</i> užklauso klausimo, kuris naudoja agregavimo operatorių remiantis viena klase, transformavimui	53
33 pav. Klausimo, kuris naudoja agregavimo operatorių remiantis viena duomenų savybe, transformavimo algoritmas	54
34 pav. <i>SPARQL</i> užklauso klausimo, kuris naudoja agregavimo operatorių remiantis viena klase, transformavimui	54
35 pav. Taisyklių rinkinys klausimo, kuris naudoja agregavimo operatorių ir remiasi dviem rolėm, nustatymui.....	55
36 pav. Klausimo, kuris naudoja agregavimo operatorių remiantis objektine savybe, transformavimo algoritmas.....	55

37 pav. SPARQL užklausa klausimo, kuris naudoja agregavimo operatorių remiantis objektine savybe, transformavimui	56
38 pav. Taisyklių rinkinys klausimo, kuris naudoja aritmetinį operatorių, nustatymui	56
39 pav. Klausimo, kuris naudoja aritmetinį operatorių, transformavimo algoritmas	57
40 pav. SPARQL užklausa klausimo, kuris naudoja aritmetinį operatorių, transformavimui	57
41 pav. Taisyklių rinkinys klausimo, kuris naudoja konkretų egzempliorių, nustatymui	58
42 pav. Klausimo, kurio rolė apribota konkrečiu egzemplioriumi, transformavimo algoritmas	59
43 pav. SPARQL užklausa klausimo, kurio rolė apribota konkrečiu egzemplioriumi, transformavimui	59
44 pav. Natūraliosios kalbos sąsajos ontologijoms loginė architektūra	62
45 pav. Panaudojimo atvejo „Vykdyti paiešką“ sekų diagrama	63
46 pav. Panaudojimo atvejo „Gauti žodyną iš ontologijos“ sekų diagrama	63
47 pav. Panaudojimo atvejo „Įvesti ir tikrinti klausimą“ sekų diagrama	64
48 pav. Panaudojimo atvejo „Analizuoti klausimą“ sekų diagrama	64
49 pav. Panaudojimo atvejo „Transformuoti klausimą į užklausą“ sekų diagrama	64
50 pav. Natūraliosios kalbos sąsajos ontologijoms komponentų diagrama	65
51 pav. Natūraliosios kalbos sąsajos ontologijoms diegimo diagrama	65
52 pav. Natūraliosios kalbos sąsajos ontologijoms komponentai	66
53 pav. Patikslinamojo dialogo pavyzdys	67
54 pav. SPARQL užklauskos šablonas klausimo, kuris remiasi dviem rolėmis be apribojimų, transformavimui	69
55 pav. SPARQL užklausa klausimo, kuris remiasi dviem rolėmis be apribojimų, transformavimui ...	69
56 pav. Įvykdytos užklauskos rezultatų sąrašas	69
57 pav. Eksperimentinio natūraliosios kalbos sąsajos tyrimo planas	72
58 pav. Elektroninės prekybos dalykinės srities esybių ir ryšių modelis	72
59 pav. Universiteto veiklos dalykinės srities esybių ir ryšių modelis	73

Santrumpų ir terminų sąrašas

Santrumpos:

DB (angl. *Database*) – duomenų bazė.

IT (angl. *Information Technology*) – informacinės technologijos.

NKS (angl. *Natural Language Interface*) – natūraliosios kalbos sąsaja.

PA (angl. *Use Case*) – panaudojimo atvejis.

Terminai:

Domenas (angl. *Domain*) – *RDFS* aksioma, kuri nusako klasę, atliekančią subjekto vaidmenį ontologijos savybėje.

GraphDB – ontologijos saugykla (duomenų bazė).

Hunspell – morfologinės analizės biblioteka.

Java – objektinė programavimo kalba, dažnai naudojama kuriant bibliotekas, susijusias su semantinėmis technologijomis.

Ontologija (angl. *Ontology*) – tam tikros dalykinės srities naudojamų sąvokų, esybių tipų, jų tarpusavio sąryšių, priklausomybių, aksiomų, dėsnų ir kt. specifikuojamas išreikštas tam tikru pavidalu.

OWL (angl. *Web Ontology Language*) – tinklo ontologijų aprašymo kalba.

Saitynas (angl. *World Wide Web*) – interneto dalis, ištekliai, kuriuos internete galima pasiekti naudojant *URL* tarpusavyje galimai susietus hipertekstu.

RDF (angl. *Resource Description Framework*) – ontologijų sudarymo duomenų modelis.

RDFS (angl. *Resource Description Framework Schema*) – ontologijų sudarymo duomenų modelio schema.

RIF (angl. *Rule Interchange Format*) – taisyklių keitimosi formatas.

SPARQL (angl. *Protocol and RDF Query Language*) – semantinio tinklo ir ontologijų užklausų kalba.

Sritis (angl. *Range*) – *RDFS* aksioma, kuri nusako klasę, atliekančią objekto vaidmenį ontologijos savybėje.

SQA (angl. *Semantic Question Answering*) – semantinis klausimų atsakinėjimas.

SQL (angl. *Structured Query Language*) – struktūrizuota užklausų kalba.

SWRL (angl. *The Semantic Web Rule Language*) – semantinio tinklo taisyklių kalba.

URI (angl. *Uniform Resource Identifier*) – vienodų resursų identifikuojamo adreso struktūra.

UML (angl. *Unified Modeling Language*) – modeliavimo ir specifikacijų kalba.

W3C (angl. *World Wide Web Consortium*) – konsorciumas, leidžiantis programinės įrangos standartus (kaip jie vadina „rekomendacijas“) saitynui.

XML (angl. *Extensible Markup Language*) — *W3C* rekomenduojama duomenų žymėjimų bei jų turinio aprašomoji kalba.

Įvadas

Natūraliosios kalbos sąsajos ontologijoms tiriamasis magistro projektas priklauso informacinių sistemų inžinerijos studijų programai.

Darbo problematika ir aktualumas

Kasdien didėjant saugomų duomenų kiekiui tiek internete, tiek ir organizacijose, naudotojui informacijos paieškos procesas tampa vis sudėtingesnis. Tradicinės, raktiniais žodžiais grindžiamos, paieškos sistemos negali suprasti pateikiamų duomenų semantikos ir grąžina ne tikslius atsakymus į naudotojo užklausas, bet sąrašą rezultatų. Tokiu atveju, naudotojas atsakymus į jį dominančius klausimus turi susirasti pats.

Semantinės paieškos sistemos leidžia gauti tikslesnius rezultatus į pateikiamas užklausas, nes jos geba geriau suprasti saugomų duomenų ir naudotojo užklausų prasmę. Tai yra įmanoma dėl to, nes semantinė paieška vykdoma ontologijose, kuriose apibrėžiamos tam tikros dalykinės srities sąvokos, esybės, jų tipai, tarpusavio ryšiai, priklausomybės ir kt. [16]. Semantinės paieškos galimybes padidina ir tai, kad ontologijoms galima naudoti išvedimo bibliotekas, kurios pagal ontologijoje aprašytas aksiomas ir taisykles gali išvesti naujus faktus. Ontologijos gali būti naudojamos plačiai. Jos gali būti skirtos paieškai internete, kai tarpusavyje sujungiamos ir didelius duomenų grafus sudaro įvairių sričių ontologijos, publikuojamos skirtingų organizacijų ar asmenų. Taip pat ontologijos gali būti naudojamos siauresnei dalykinei sričiai, aprašant konkrečius tos srities duomenis. Tarkime, įmonė gali naudoti ontologiją savo dalykinės srities duomenims saugoti ir analizuoti. Ontologijos gali būti naudingos, kai prireikia integruoti duomenis iš įvairių šaltinių, tarkime, realiųjų duomenų bazių bei tekstinių failų. Šio tiriamojo projekto metu koncentruojamasi į taikomas ontologijas, kurios skirtos konkrečiai dalykinei sričiai.

Semantinių paieškos sistemų kūrimui taip pat didelę įtaką turi ir naudotojo sąsaja, kuri turi būti ne tik funkcionali, bet ir patogi bei lengvai suprantama sistemos naudotojui. Šiame darbe buvo nustatyti pagrindiniai natūraliosios kalbos sąsajos ontologijoms reikalavimai: pritaikymas lietuvių kalbai ir žodyno paruošimas įdedant kuo mažiau rankinio darbo. Tiriamojo projekto metu buvo nagrinėjamos jau sukurtos natūraliosios kalbos sąsajos, kurios analizuoja ir transformuoja klausimus remiantis aprašytais taisyklėmis. Analizės metu buvo lyginamos tokios semantinės paieškos naudotojų sąsajos kaip *QuestIO*, *FREyA*, *QRAKEL*, *Querix* ir *SBVR* pagrįsta NKS. Nustatyta, kad vienintelė *SBVR* pagrįsta NKS yra pritaikyta lietuvių kalbai, tačiau pagrindinis šios sąsajos trūkumas – dalykinės srities aprašymui yra reikalingas ne tik ontologijos sukūrimas, bet ir *SBVR* žodynas, kuriame aprašomos sąvokos turi būti suderintos su ontologija. Taigi, norint pritaikyti tokią naudotojo sąsają dalykinei sričiai, reikia atlikti nemažai rankinio darbo.

Atliktos lyginamosios analizės metu buvo nustatyta, kad nei vienoje išanalizuotoje naudotojo sąsajoje nėra realizuoti visi tiriamojo projekto metu nustatyti reikalavimai. Dėl šios priežasties yra poreikis sukurti naują sprendimą, kuris leistų naudotojui vykdyti semantinę paiešką ontologijoje lietuvių kalba.

Darbo tikslas ir uždaviniai

Darbo tikslas – sudaryti sąlygas vykdyti semantinę paiešką pateikiant klausimus lietuvių kalba.

Darbo uždaviniai:

1. atlikti semantinės paieškos ir su ja susijusių duomenų saugojimo ir užklausų vykdymo technologijų analizę;
2. išanalizuoti natūraliosios kalbos sąsajas ir pagal pasirinktus kriterijus atlikti esamų naudotojo sąsajų lyginamąją analizę;
3. sukurti ir formaliai aprašyti metodą, leidžiantį natūraliąja kalba pateiktus klausimus ontologijoms transformuoti į užklausas;
4. pagal sukurtą metodą realizuoti natūraliosios kalbos sąsajos ontologijoms prototipą;
5. atlikti sukurto metodo eksperimentinį tyrimą;
6. apibendrinti sukurto metodo eksperimentinio tyrimo rezultatus.

Darbo rezultatai ir jų svarba

Sukurtas ir formaliai aprašytas metodas, leidžiantis naudotojui vykdyti paiešką, pateikiant klausimus lietuvių kalba, ir transformuoti juos į ontologijų užklausas. Metode aprašyti algoritmai realizuoti sukūrus natūraliosios kalbos sąsajos prototipą, skirtą įvesti klausimą, jį išanalizuoti remiantis ontologijos schemas konceptais, transformuoti į užklausą ir, ją įvykdžius, pateikti atliktos paieškos rezultatus. Metodas skirtas lietuvių kalbai, jo privalumas yra tai, kad nereikia aprašyti žodyno konkrečiai dalykinei sričiai.

Darbo struktūra

Tiriamą magistro studijų projektą sudaro penki skyriai. Pirmame skyriuje analizuojamas semantinis tinklas, ontologijos, jų aprašymo kalbos, išvedimo ir paieškos technologijos bei semantinė paieška. Analizuojamos jau sukurtos natūraliosios kalbos sąsajos, jų privalumai ir trūkumai bei naudojami būdai siekiant palengvinti klausimų formulavimo ir transformavimo į semantinę užklausą procesą. Antrame skyriuje aprašoma kuriamo natūraliosios kalbos sąsajos ontologijoms prototipo reikalavimų specifikacija, pateikiamas formalus sprendimo aprašymas. Trečiame skyriuje suprojektuota kuriamos sąsajos loginė architektūra, realizacijos modelis bei su diegimu ir testavimo procedūromis susijusi informacija. Ketvirtame skyriuje aprašomi pagrindiniai sukurtos natūraliosios kalbos sąsajos etapai: klausimų formulavimas, analizavimas ir transformavimas į užklausą. Penktame skyriuje aprašomas sukurto metodo eksperimentinis tyrimas bei tyrimo rezultatų apibendrinimas.

1. Probleminės srities analizė

1.1. Analizės tikslas

Analizės tikslas – atlikti semantinio tinklo, ontologijų, jų aprašymo kalbų, išvedimo ir paieškos technologijų bei semantinės paieškos analizę. Išnagrinėti jau sukurtas natūraliosios kalbos sąsajas, jų privalumus ir trūkumus. Pagal pasirinktus kriterijus atlikti esamų natūraliosios kalbos sąsajų ontologijoms lyginamąją analizę.

1.2. Tyrimo objektas, sritis ir problema

Tyrimo objektas – semantinės paieškos vykdymo procesas, pateikiant klausimus lietuvių kalba.

Tyrimo sritis:

- semantinis tinklas;
- ontologijos ir jų aprašymo kalbos: *OWL*, *OWL 2*, *RDF*;
- ontologijų išvedimo ir paieškos technologijos: *SPARQL*, *Jena*, *HermiT*, *FaCT++*, *GraphDB*;
- semantinė paieška;
- natūraliosios kalbos sąsajos, skirtos ontologijoms: *QuestIO*, *FREyA*, *QRAKEL*, *Querix* ir *SBVR* pagrįsta NKS.

Tyrimo problema:

Natūraliosios kalbos sąsajų, skirtų ontologijoms ir pritaikytų lietuvių kalbai, paruošimas reikalauja nemažai rankinio darbo. Viena didžiausių problemų yra tai, kad joms reikalingas ne tik ontologijos sukūrimas, bet ir papildomai paruošiamas žodynas, kuriame aprašomos sąvokos turi būti suderintos su ontologijos schemas konceptais.

1.3. Tyrimo objekto analizė

1.3.1. Semantinio tinklo apibrėžimas

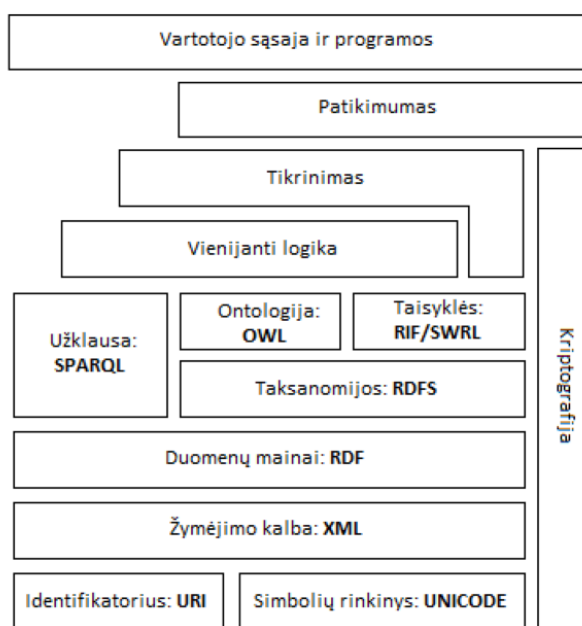
Nuo pat pasaulinio saityno (angl. *World Wide Web*, WWW) atsiradimo pradžios jame pateikiamas turinys buvo skirtas interneto naudotojams, kurie pateikiamą informaciją gali perskaityti ir suprasti. Tačiau kompiuteriai suprasti šios informacijos tradiciškai negalėjo, jie gebėjo tik tinkamai atvaizduoti pateiktą vaizdinę ar tekstinę informaciją saityno naudotojui. Dėl šios priežasties vykdant paiešką buvo susiduriama su tokiomis problemomis [3]:

1. **informacijos paieška** (angl. *Retrieval*) – vykdant paiešką, grindžiamą raktiniais žodžiais, negali būti randami visi atsakymai, nes, apdorojant pateiktus klausimus, yra neįvertinamos tokios kalbos dalys kaip sinonimai, apibendrinimai, metaforos, kontekstas ir kt.;
2. **informacijos ištraukimas** (angl. *Extraction*) – informacija yra pateikiama neapibrėžta struktūros forma, todėl korektiškai ją apdoroti ir interpretuoti gali tik žmogus, nes kompiuterinės programos neturi tam pakankamai žinių;

3. **informacijos priežiūra** – virtualioje erdvėje saugomi duomenys yra pateikiami nestruktūrizuota forma, todėl sudėtinga užtikrinti saugomų duomenų sintaksinę ir semantinę darną bei jų korektiškumą;
4. **personalizavimo problema** – susiduriama su tokiomis problemomis, kai reikalingas pateikiamos informacijos adaptavimas asmeniniams poreikiams.

Tikslesniam informacijos apdorojimui internete buvo sukurtas semantinis tinklas (angl. *Semantic Web*), kurio terminą sugalvojo pirmojo pasaulinio tinklo kūrėjas Timas Bernersas Lee. Šią sąvoką mokslininkas apibrėžė kaip pasaulinio saityno išplėtimą, kuriame saugomi duomenys turi aiškiai apibrėžtą struktūrą. Pasak Timo Bernerso Lee, semantinis tinklas nėra nauja technologija – tai dabartinio saityno pratęsimas, kuriantis bendrą jo pateikiamo turinio struktūrą, suteikdamas jam prasmę ir skatindamas darnų žmogaus bei kompiuterio tarpusavio bendradarbiavimą [3]. Pagrindinė semantinio tinklo vizija orientuota į tai, kad internete saugomi duomenys būtų lengviau suprantami naudotojui ir apdorojami semantinės paieškos sistemoms, kurios pateiktų kuo tikslesnius atliktos paieškos rezultatus.

Dažniausiai semantinio tinklo struktūrinė architektūra yra atvaizduojama semantinio tinklo kūgiu (angl. *Semantic Web Stack*), kuris pavaizduotas 1 pav. Kiekvienas aukštesnis architektūros hierarchijos lygis naudoja žemiau esančio sluoksnio technologijas.



1 pav. Semantinio tinklo struktūrinė architektūra

Semantinio tinklo struktūrą sudaro trys skirtingi sluoksniai:

- **žemiausias architektūros sluoksnis** – hipertekstinio tinklo technologijos (identifikatorius *URI*, unikodas ir žymėjimo kalba *XML*);
- **vidurinis architektūros sluoksnis** – standartizuoto semantinio tinklo technologijos (duomenų modelis *RDF*, tinklo ontologijų kalba *OWL* ir ontologijų užklauso kalba *SPARQL*);
- **aukščiausias architektūros sluoksnis** – nerealizuotos semantinio tinklo technologijos, kurios dar nėra standartizuotos arba yra tik idėjos, kurias reikėtų įgyvendinti (kriptografija, *RIF* arba *SWRL* taisyklės ir kt.).

Kadangi semantinė paieška leidžia gauti tikslesnius paieškos rezultatus, šių technologijų taikymas tampa vis paklausesnis įvairiose srityse: inžinerijoje, medicinoje, geografiijoje, biologijoje, produktų gamyboje ir kt. Informacijos apdorojimas, paremtas semantinėmis technologijomis, yra nuolat plėtojamas ir pritaikomas vis daugiau verslo procesų. Pagrindiniai semantinių technologijų pranašumai, užtikrinantys paieškos rezultatų tikslumą, išskiriami tokie [5]:

- **loginis srities pagrindimas** – semantinės technologijos grindžiamos formalia logika, kuri leidžia išvesti naujus faktus;
- **paieška ir atradimas** – semantinės technologijos leidžia atlikti tikslingesnę paiešką, išrinkdamos ir apdorodamos konkrečius rezultatus;
- **adaptyvumas ir tvirtumas** – semantinės technologijos apdoroja užklausas, pateiktas natūraliąja kalba, įvertinant bendrą turinio kontekstą;
- **žinių grafas** – semantinės technologijos aprašo tam tikros dalykinės srities žinias duomenų modeliu (grafu);
- **verslo kontrolė** – semantinės technologijos sudaro sąlygas įgyvendinti ir pritaikyti tam tikrus sprendimus įvairiuose verslo procesuose.

1.3.2. Ontologijos apibrėžimas

Semantinė paieška vykdoma ontologijose, kurios aprašo dalykinės srities konceptus, jų klasifikaciją ir tarpusavio ryšius. Ontologijos terminas yra kilęs iš dviejų graikų kalbos sąvokų: „būtis“ ir „žodis“. Filosofijoje ontologija minima metafizikoje, nagrinėjant prigimtį, realųjį pasaulį ir egzistencinius būties klausimus. Pagrindinis klausimas į kurį metafizikoje bandoma atsakyti – „Kokia yra būties prasmė?“ [2]. Šis tradicinis ontologijos termino apibūdinimas yra pritaikomas daugelyje abstrakčių mokslo šakų, tokių kaip fizika, chemija, biologija, matematika ir kt.

Informatikos moksluose esminį ontologijos termino apibrėžimą mokslininkai svarstė kurį laiką:

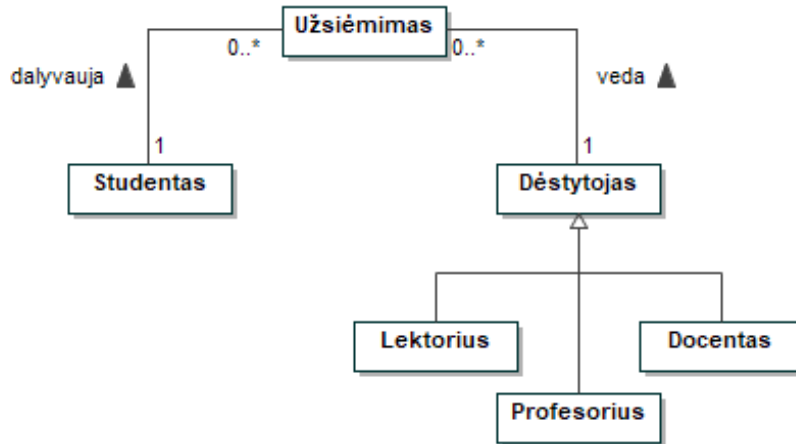
- T. Gruberis 1993 metais jį apibūdino taip: „*Ontologija – tai aiškiai išreikšta konceptualizacijos specifikacija.*“;
- W. Borst 1997 metais patikslino, kad: „*Ontologija – tai formali bendros konceptualizacijos specifikacija.*“;
- R Studer, R. Benjamins, D Fensel 1998 metais apibrėžė taip: „*Ontologija – tai formali ir aiškiai išreikšta bendros konceptualizacijos specifikacija.*“.

Šis ontologijos apibūdinimas nusako, kad ontologija yra tam tikros dalykinės srities formalus naudojamų sąvokų, jų savybių bei tarpusavio ryšių visumos žinių modelis. Pagrindinės ontologijos savybės tam tikros dalykinės srities kontekste yra:

- **formalumas** – formaliai aprašyta ontologija yra lengviau apdorojama kompiuteriams ir tokiu būdu žmogus bei programinė įranga ją gali interpretuoti vienodai;
- **tikslumas** – aiškiai išreikšta ontologija padeda nagrinėjamą sritį interpretuoti visiems vienodai;

- **pasidalinimas** – aprašyti ryšiai tarp kalbos elementų padeda suvokti jų prasmę vienodai ir pakartotinai juos panaudoti.

Ontologijų naudojimas IT srityje padeda apibrėžti ne tik tam tikros dalykinės srities komponentus, bet taip pat palengvina duomenų struktūrizavimo procesą. Žemiau pateikiamas vienas iš ontologijos grafinio vaizdavimo būdų (esybių ir ryšių diagrama) universiteto dalykinėje srityje (žr. 2 pav.).



2 pav. Grafinis ontologijos pavyzdys

Sukurta esybių ir ryšių diagrama apibrėžia dėstytojo vedamų užsiėmimų universitete veiklą. Modelyje apibrėžiama duomenų struktūra apie universiteto studentą, dalyvaujantį užsiėmime, kurį veda universiteto dėstytojas. Naudojama ir klasifikacija – dėstytojas gali užimti lektoriaus, profesoriaus arba docento pareigas.

Ontologijos gali būti aprašomos įvairia forma, jos gali būti įvairių paskirčių ir skirtingo išraiškingumo. Žemiau pateikiamas ontologijų skirstymas pagal įvairius požymius [5]:

1. **išraiškingumą** – ontologijoms aprašyti gali būti pasirenkamos skirtingos kalbos, nuo kurių priklauso jų išraiškingumo lygis, pavyzdžiui:
 - glosarijus – žodynas, pateikiantis tam tikrų žodžių paaiškinimus;
 - taksonomija – objektų klasifikavimas, pagal tam tikrą požymį;
 - tezaurus – žodynas, aprašantis žodžių junginių tarpusavio ryšius;
 - aksiomos – teiginiai, aprašantys tam tikros dalykinės srities taisykles.
2. **formalumą** – ontologijos gali būti aprašomos tam tikra kalbos rūšimi:
 - neformalios – neformaliai aprašyti teiginiai (pvz., terminų katalogai);
 - formalios – formaliai aprašyti teiginiai (pvz., grįstos prototipais). Į šią grupę patenka ir formalia ontologijų kalba, t.y. *OWL*, aprašytos ontologijos.
3. **paskirtį** – ontologijos gali būti aprašomos pagal paskirtį:
 - aukščiausio lygio ontologijos (ALO) aprašo bendras sąvokas, kurios nepriklauso nuo konkrečios dalykinės srities (pvz., DOLCE, GFO, SUMO);

- taikomųjų sričių ontologijos (TSO) aprašo konkrečių dalykinių sričių žodynus, specializuodamos ALO terminus;
- užduočių ontologijos aprašo (UO) konkrečių užduočių ar veiklų žodynus, specializuodamos ALO terminus;
- taikymų ontologijos (TO) aprašo sąvokas, kurios specializuoja tiek TSO, tiek ir UO.

4. **galią** – ontologijos gali būti aprašomos tam tikra išraiška:

- „lengvasvorės“ ontologijos aprašo sąvokas, tipus, hierarchiją ir sąryšius;
- „sunkiasvorės“ ontologijos papildomai aprašo kardinalumo apribojimus, sąryšių klasifikaciją ir kt. Taip pat leidžia aprašyti aksiomas ir išvedimo taisykles, naudojant logikos formalizmus ir loginio išvedimo sistemas.

Taigi, ontologijos suteikia galimybę kompiuterinėms programoms sunkiau apdorojamą informaciją išsaugoti tokiu formatu, kad ji būtų suprantama bei interpretuojama kompiuterių. Ontologijos taip pat gali jungtis tarpusavyje ir sudaryti duomenų tinklą, kuriame semantinės paieškos sistemos gali vykdyti paiešką ne viename, bet daugelyje šaltinių.

1.3.3. Ontologijų aprašymo kalbos

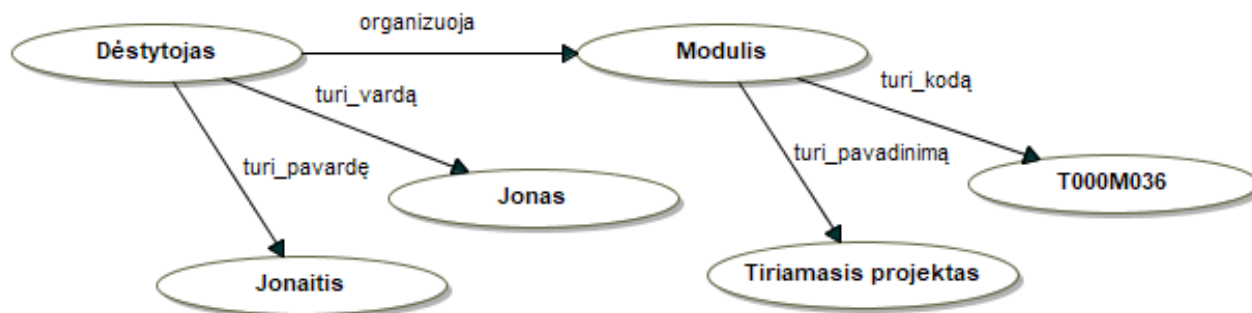
Semantinio tinklo kūrimą ir plėtojimą vykdo W3C konsorciumas, kuris sukūrė ir palaiko ontologijų aprašymo kalbas: *RDF*, *OWL* ir *OWL 2*. Šios kalbos gali būti naudojamos tam tikros dalykinės srities žinių aprašymui, turinio apdorojimui, remiantis tam tikra apibrėžta ontologijos struktūra. Tolimesniuose skyreliuose pateikiami šių kalbų aprašymai (žr. 1.3.3.1 – 1.3.3.3 skyrius).

1.3.3.1. RDF duomenų modelis

Viena pagrindinių semantinio tinklo dalių yra *RDF* (angl. *Resource Description Framework*) standartas, skirtas aprašyti resursus ir jų ryšius internetiniame tinkle. Tai viena pirmųjų žemiausio lygmens standartizuotų kalbų, kuri skirta aprašyti ir mainyti duomenimis semantiniame tinkle. Remiantis šia specifikacija buvo sukurti tokie standartai, kaip *RDFS* (angl. *Resource Description Framework Schema*), *OWL*, *OWL 2* ir kt.

Pagrindinis *RDF* duomenų modelio uždavinys – aprašyti duomenis grafu, kuris sudarytas iš viršūnių, aprašančių resursus, ir lankų tarp viršūnių, kurie apibrėžia to resurso savybes. *RDF* koncepciniame modelyje ryšiai tarp duomenų suskirsto pateikiamą informaciją į skirtingus komponentus: objektą (angl. *Object*), subjektą (angl. *Subject*) ir predikatą (angl. *Predicate*). Aprašomas duomenų modelio formatas yra vadinamas tripletu (angl. *Triple*), kur objektas ir subjektas apibūdina resursus, o predikatas vaizduoja jų tarpusavio ryšį.

Remiantis *RDF* grafu gali būti aprašoma tam tikros dalykinės srities visuma, kurios resursai tarpusavyje yra susiejami tam tikrais ryšiais. *RDF* duomenų modelyje aprašomi resursai gali būti išreiškiami ir keliais tripletais, kurių visuma sudaro orientuotą grafą. Žemiau pateikiamas *RDF* duomenų modelio pavyzdys (žr. 3 pav.).



3 pav. RDF grafo pavyzdys

Modelyje apibrėžiama dėstytojo organizuojamų modulių universitete vykdoma veikla. Jame universiteto dėstytojas aprašomas objektu – „*Dėstytojas*“, kurio predikatas „*organizuoja*“ identifikuoja organizuojamą modulį, subjektą – „*Modulis*“. Objektas „*Dėstytojas*“ turi tokius predikatus kaip „*vardas*“ (Jonas) ir „*pavardė*“ (Jonaitis). Subjektas „*Modulis*“ turi tokius predikatus kaip „*kodas*“ (T000M036) ir „*pavadinimas*“ (Tiriamasis projektas).

RDF yra konceptualus duomenų modelis, kuris remiasi teiginiais, aprašančiais resursų subjekto, predikato ir objekto išraiškų tripletą. *RDFS* turi mechanizmą, kuris leidžia aprašyti susijusių resursų tripletus (*RDF*) ir ryšį tarp jų (pvz., klasės, poklasiai ir kt.). Taigi, galima teigti, kad *RDFS* yra semantinis *RDF* plėtinys, skirtas aprašyti susijusius duomenis. Pavyzdžiui, asmens klasė gali turėti dėstytojo ir lektoriaus *RDF* egzempliorių, tuo tarpu *RDFS* gali nurodyti, kad dėstytojas yra klasė, o lektorius yra dėstytojo poklasis [4].

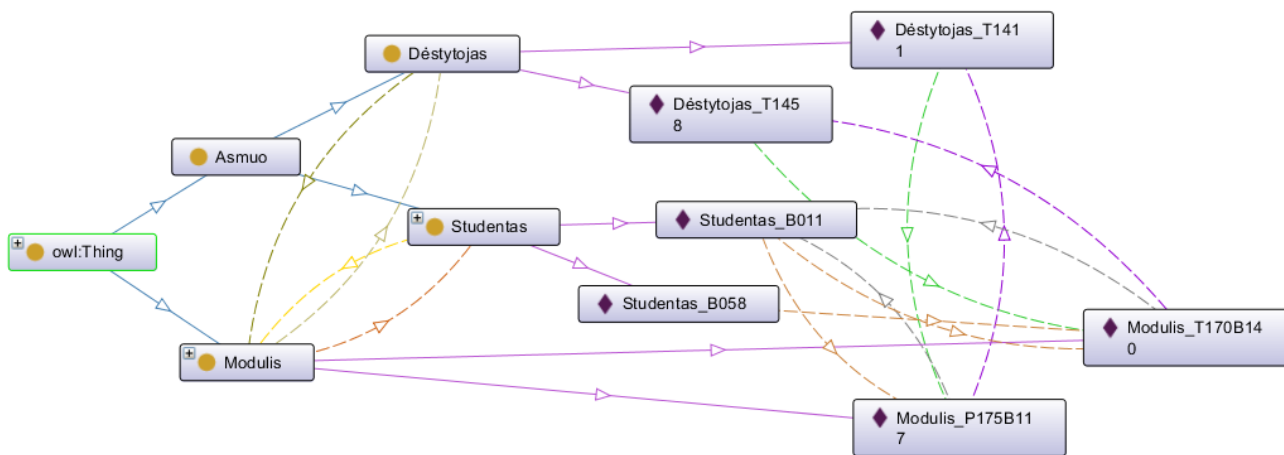
1.3.3.2. Tinklo ontologijų kalba OWL

Viena populiariausių semantinio tinklo paieškos sistemų ontologijų kūrimui naudojamų kalbų yra *OWL* (angl. *Web Ontology Language*). Tai ontologijų kalba, leidžianti aprašyti klases, jų savybes bei tarpusavio ryšius, kurie grindžiami *RDF* duomenų modeliais.

OWL ontologijų kalba turi tris skirtingus išraiškos lygius (dialektus):

- ***OWL Lite dialektas*** pritaikytas paprastų žinių aprašų sudarymui, kuriems užtenka hierarchinės klasifikacijos ir ribotų išraiškos galimybių. Ši kalba palaiko paprastus apribojimus, tokius kaip kardinalumas, ekvivalentiškumas, klasifikavimas ir kt.;
- ***OWL DL (OWL Lite plėtinys)*** suteikia maksimalų įmanomą išraiškingumą išlaikant skaičiavimų (išvedimo) įvykdymą per baigtinį laiko tarpą. Šis dialektas tinkamas daugumai kuriamų ontologijų, nes visus *OWL* kalbos konstruktus (tačiau kai kurie gali būti naudojami su apribojimais) ir yra suderinamas su destruktiviaja logika (angl. *Description Logic*);
- ***OWL Full (OWL DL plėtinys)*** turi maksimalų išraiškingumą, tačiau negali užtikrinti atliekamų skaičiavimų (išvedimo) per baigtinį laiko tarpą.

Naudojant *OWL* ontologijų kalbą gali būti aprašomos žinios apie tam tikrą dalykinę sritį. Žemiau pateikiamas *OWL* ontologijos pavyzdys, kuriame saugoma informacija apie universitetą (žr. 4 pav.).



4 pav. OWL ontologijos pavyzdys

Modelyje pateikiami duomenys apie universitetą, kuriame dėstytojai organizuoja modulius, skirtus universiteto studentams. Pavyzdyje matoma dalykinės srities klasių hierarchija, kurioje klasės ir poklasės yra susietos tarpusavio ryšiais. Grafe aukščiausio lygmens klasė aprašoma „*owl:Thing*“ – tai ontologijos aprašo tėvinė klasė, apimanti visą aprašytą dalykinę sritį. Ši klasė turi dvi poklases „*Asmuo*“ ir „*Modulis*“, kurios taip pat turi savas poklases. Modelyje taip pat matyti klasių egzemplioriai ir jų ryšiai, pvz., *Dėstytojas_T1458* organizuoja *Modulis_T170B140*.

1.3.3.3. Antroji tinklo ontologijų kalbos versija OWL 2

Nors ankstesniame skyrelyje aptartas pirmasis *OWL* kalbos standartas buvo pakankamai sėkmingas, vėliau buvo pastebėta keletas trūkumų, tokių kaip tinkamų integruotų duomenų tipų neapibrėžimas, sintaksės, išraiškingumo problemos ir kt. Dėl šios priežasties 2009 metais *W3C* konsorsiumas išleido antrąją ontologijų kūrimo kalbos versiją *OWL 2* (angl. *Web Ontology Language*), kuri buvo *OWL* ontologijų kalbos tęsinys su keliomis naujomis funkcijomis.

Kartu su naujai išleistu *OWL 2* standartu buvo sukurti trys skirtingi dialektai (profiliai):

- ***OWL 2 EL* dialektas** leidžia atlikti išvedimo užduotis naudojant polinominius laiko algoritmus. Dialektas tinkamas didelių ontologijų palaikymui, siekiant užtikrinti programinės įrangos našumą;
- ***OWL 2 QL* dialektas** naudoja standartinę reliacinių duomenų bazių (DB) technologiją, kuri leidžia atsakyti į jungtines užklausas (angl. *LogSpace*). Dialektas tinkamas lengvoms ontologijoms, saugančioms didelį duomenų (egzempliorių) kiekį. Šis profilis suteikia galimybę vykdyti paiešką tada, kai egzemplioriai saugomi reliacinėse duomenų bazėse. Tokiu atveju užklausa gali būti transformuojama į reliacinių duomenų bazių užklausa, t.y. *SQL*;
- ***OWL 2 RL* dialektas** naudojamas veiklos taisyklėmis grindžiamose sistemos (angl. *business rules engines*). Tai yra *OWL 2 EL* ir *OWL 2 QL* dialektų plėtinys. Šis dialektas tinkamas ontologijoms, kurioms reikalingas didelis išplečiamumas ir būtinumas tiesiogiai valdyti duomenis.

1.3.4. Semantinio tinklo technologijos

1.3.4.1. Ontologinių užklausų kalba SPARQL

Išpopuliarėjus *RDF* duomenų modeliui, skirtam pateikti informaciją internete, atsirado poreikis sukurti kalbą, skirtą vykdyti paiešką tokiose duomenų struktūrose. Buvo sukurta nemažai tam skirtų kalbų, tačiau bėgant laikui labiausiai išpopuliarėjo *SPARQL* (angl. *Protocol and RDF Query Language*), kuri tapo *de-facto* standartine ontologijų užklausų kalba. 2008 metais *SPARQL* kalba tapo oficialiai rekomenduojama *W3C* konsorciumo [24].

SPARQL – tai *RDF* užklausų kalba, grindžiama *RDF* grafo atitikimu (angl. *mapping*) tripletų šablonui (angl. *Triple Pattern*). Užklausos gražinami rezultatai priskiriami kintamiesiems.

SPARQL užklausos struktūrą sudaro šios dalys (žr. 5 pav.) [28]:

- **priešdėlių** (angl. *PREFIX*) **aprašas** – skirtas aprašyti užklausoje naudojamų ontologijų *URI* adresus;
- **užklausos resursas** – skirtas nurodyti, kuriam *RDF* grafiui reikia pateikti užklausą;
- **rezultato sąlyga** – skirta nurodyti, kokia informacija turi būti gražinama pateiktoje užklausoje. Šioje dalyje išvardinami, kokie kintamieji turi būti išvedami ir kaip turi atrodyti užklausos rezultatų stulpeliai;
- **užklausos kriterijus** – skirtas nurodyti, ko ieškoti pagrindiniame duomenų rinkinyje. Šioje dalyje aprašomi užklausos tripletų šablonai ir suformuojamos užklausos sąlygos;
- **užklausos modifikatorius** – skirtas apdoroti užklausos rezultatus.

```
# Priešdėlio aprašas
PREFIX foo: <http://example.com/resources/>
...
# Užklausos resursas
FROM ...
# Rezultato sąlyga
SELECT ...
# Užklausos kriterijus
WHERE {
    ...
}
# Užklausos modifikatorius
ORDER BY ...
```

5 pav. *SPARQL* užklausos struktūra

1.3.4.2. Ontologijų užklausų vykdymo sprendimai

Ontologijų išvedimo bibliotekos leidžia patikrinti ontologijoje aprašytų aksiomų neprieštarinumą bei suteikia galimybę pagal aprašytas išvedimo taisykles ar aksiomas išvesti tam tikrus faktus. Todėl naudojant šias bibliotekas galima gauti išsamesnius paieškos rezultatus.

Ontologijų išvedimo mechanizmai, orientuoti į anksčiau aptartą tinklo ontologijų kalbą *OWL*, gali būti apibūdinami keliais skirtingais aspektais:

1. pirmasis aspektas apima pagrindinius duomenų išvedimo ypatumus, tokius kaip naudojamas išvedimo metodas bei palaikomos kalbos išraiškingumas;
2. antrasis aspektas orientuotas į praktišką naudojimą: ar išvedimo biblioteka palaiko *OWL API* ir *OWL Link API*, ar jis gali būti prieinamas kaip *Protégé* papildinys, kokiose platformose veikia ir kt.

Detalesniam ontologijų išvedimo mechanizmų analizavimui ir jų palyginimui buvo pasirinkti keturi skirtingi mechanizmai:

- **Jena** – viena labiausiai paplitusių bibliotekų, veikiančių semantinio tinklo *Java* programavimo kalbos platformoje. Ši biblioteka skirta informacinių sistemų, kurios grindžiamos *RDF* ontologijomis, palaikymui. *Jena* leidžia nuskaityti bei išsaugoti ontologijas, aprašytas skirtingais formatais: *RDF/XML*, *N3* bei *N-Triple*. *Jena* biblioteka *RDF* standartą pateikia kaip abstraktų duomenų modelį, kurio resursai gali būti gaunami iš *RDF* failų, duomenų bazių (DB) ir kt. [22];
- **Hermit** – *Java* programavimo kalbai skirta biblioteka orientuota į *OWL* ontologijas. Tai pirmoji viešai prieinama biblioteka, apimanti daugybę duomenų optimizavimo būdų. Bibliotekoje taip pat naudojamos įvairios ontologijų klasifikavimo optimizavimo technologijos, kas leidžia ženkliai sumažinti klasifikavimo laiką. Šis įrankis pirmą kartą leido įvykdyti išvedimą daugybėje ontologijų, kurių klasifikavimas būdavo per sudėtingas anksčiau sukurtoms išvedimo bibliotekoms [25];
- **FaCT++** – tai atvirojo kodo *C++* programavimo kalbos biblioteka, pagrįsta optimizuotais lentelių algoritmais. Ši biblioteka palaiko *OWL* ontologijų kalbos išraiškos lygį *OWL DL* bei antrosios tinklo ontologijų kalbos versiją *OWL 2*. Taip pat *FaCT++* palaiko tokias sąsajas, kaip *Protégé*, *Jena*, komandinės eilutės (angl. *Command Line*), *OWL API* (biblioteka, skirta nuskaityti ontologijas į atmintį, iš atminties įrašyti į failą įvairiais formatais ir kt.). Orientuota į tokias išvedimų galimybes: realizavimas, klasifikavimas, priskyrimas, nuoseklumas ir kt. Palaikomos sintaksės: *NA Download*, *Core publication* [11];
- **GraphDB** – tai semantinė saugykla, kurios veikimas paremtas iš anksto aprašytais įvairiais išraiškingumo taisyklių rinkiniais: *RDFS* ar įvairių variacijų *OWL*. Remiantis *GraphDB* taisyklių rinkiniais duomenų išvedimas atliekamas duomenų įrašymo metu (angl. *total materialization*). Tokiu atveju, kai į saugyklą įkeliami didesni duomenų kiekiai, šis procesas padidina duomenų apdorojimo laiką. Tačiau pagrindinis to pranašumas – vykdant užklausas paieška yra greitesnė. Duomenis į saugyklas galima įrašyti tiesiogiai įkeliant *OWL* failus arba sukuriant virtualias saugyklas, kuriose publikuojami duomenys nuskaitomi iš reliacinių duomenų bazių. Taip pat duomenys gali būti įkeliami ir iš tekstinių failų, tokių kaip *CSV*, *JSON* [14].

Analizuojant visus aptartus ontologijų užklausų vykdymo sprendimus buvo nustatyti pagrindiniai natūraliosios kalbos sąsajos ontologijoms reikalavimai duomenų paieškai, kuriais remiantis atliktas esamų sprendimų palyginimas, pateikiamas 1 lentelė.

1 lentelė. Ontologijų užklausų vykdymo lyginamoji analizė

Išvedimo mechanizmas	<i>Jena</i>	<i>HermiT</i>	<i>FaCT++</i>	<i>GraphDB</i>
Savybės				
Programavimo kalba	<i>Java</i>	<i>Java</i>	<i>Java</i>	<i>Java</i>
Išvedimo vykdymo laikas	Užklausos vykdomo metu	Užklausos vykdomo metu	Užklausos vykdomo metu	Duomenų įrašymo metu
<i>OWL</i> užklausų kalbos palaikymas	+	+	+	+
<i>SPARQL</i> palaikymas	+	+	+	+
Metodologija	<i>RDF</i> pagrindu	Hiperlentelės pagrindu	Lentelės pagrindu	<i>RDF</i> pagrindu
Išraiškos lygis	RL	DL	DL	DL, QL, RL
<i>OWL API</i> palaikymas	–	+	+	+
<i>OWL Link API</i> palaikymas	–	+	+	+
<i>Jena</i> palaikymas	+	+	+	+
Laipsniškas klasifikavimas (pridedant/šalinant)	–	–	–	+
Išbaigtumas	–	+	+	+
Duomenų saugojimo būdas	Saugojimas į failą	Saugojimas į failą	Saugojimas į failą	Saugojimas saugykloje

Apibendrinant galima teigti, kad ne visi išanalizuoti užklausų vykdymo mechanizmai yra skirti palaikyti visą *OWL* ontologijų kalbos išraiškos lygį *OWL 2 DL*. Tokios bibliotekos, kaip *HermiT* ir *FaCT++* palaiko visus *OWL DL* standartus, tačiau *Jena* orientuota į *OWL 2 RL*. Visuose sprendimuose duomenų išvedimas yra atliekamas užklausos vykdomo metu, išskyrus *GraphDB* mechanizmą, kuriame išvedimas atliekamas duomenų įrašymo metu. Tokiu būdu, vykdant užklausas *GraphDB* paieška yra greitesnė. Remiantis atlikta analize nustatyta, kad įvykdytos užklausos rezultatai visuose išvedimo mechanizmuose yra išsaugomi į failą, išskyrus *GraphDB*, kuris kartu yra ir saugykla. Taip pat *GraphDB* leidžia nuskaityti duomenis iš įvairių šaltinių, todėl toks sprendimas suteikia daugiau galimybių vykdant duomenų paiešką konkrečių organizacijų duomenyse. Apibendrinant atliktos lyginamosios analizės rezultatus galima teigti, kad daugiausiai pranašumų turi *GraphDB* semantinė saugykla, kuri palaiko pagrindinius reikalavimus, suteikia galimybes vykdyti greitesnę paiešką ir kartu gali būti naudojama kaip saugykla.

1.3.5. Semantinės paieškos apibrėžimas

Informacijos paieškai pagal jos prasmę internete yra naudojama semantinė paieška (angl. *Semantic Search*). IT kontekste semantikos sąvoka yra apibrėžiama kaip lingvistikos šaka, tirianti kalbos arba

jos simbolių prasmę. Semantika nagrinėja kaip sudėtingesnių sąvokų reikšmės interpretuojamos iš paprastų sąvokų, remiantis tam tikromis sintaksės taisyklėmis. Taigi, semantikoje kalbos elementų ir jų darinių prasmė priklauso nuo jų konteksto, kuris apima sąvokų aplinką, ir pragmatikos, kuri nagrinėja, kaip kontekstas įtakoja žinutės prasmę.

Pagrindinis semantinės paieškos tikslas – pagerinti ieškomos informacijos tikslumą, stengiantis suprasti naudotojo ketinimus ir kokį rezultatą jis siekia gauti vykdydamas paiešką. Tokiu būdu siekiama duomenų apdorojimo metu išnagrinėti ne tik pateiktos paieškos užklauso atskirų žodžių prasmę arba jų raktinius žodžius, bet taip pat ir jų tarpusavio ryšį. Semantinėmis technologijomis paremta paieška analizuoja duomenų prasmę įvairiais aspektais: užklauso kontekstas, siekiamas paieškos tikslas ir paties klausimo prasmė, sinonimų, apibendrinimų bei žodžių variacijų atpažinimas.

Didžiausias semantinėmis technologijomis paremtos paieškos pranašumas yra tai, kad ji suteikia ieškomos informacijos rezultatų tikslumą, analizuodama duomenų prasmę įvairiais pjūviais. Naudojant semantinės paieškos sistemas siekiama visapusiškai susieti svarbią informaciją ir pateikti kuo aktualesnius paieškos rezultatus naudotojui. Semantinės paieškos sistemos „*Hakia*“ kūrėjai apibrėžia dešimt pagrindinių semantinės paieškos požymių ir privalumų, lyginant jas su tradicinėmis, raktiniais žodžiais grindžiamomis sistemomis [10]:

1. **užklauso nagrinėjimas morfologiškai** – užklauso apdorojimas pagal morfologines variacijas, sutvarkant tekstą pagal kalbos dalis, giminę, linksniuotę, skaičių ir kt. Pavyzdžiui, užklauso, kuriose naudojami žodžiai „*lyti*“, „*lyja*“ ir „*lijo*“ turėtų būti interpretuojamos vienodai;
2. **sinonimų atpažinimas** – sąvokų sinonimų supratimas bendrame užklauso kontekste. Pavyzdžiui, užklausoje žodis „*negalavimas*“ turėtų būti suprantamas kaip „*liga*“ sinonimas;
3. **apibendrinimų palaikymas** – sąvokų apibendrinimų atpažinimas pateiktoje užklausoje. Pavyzdžiui, užklausoje pateiktas klausimas „*Kuri liga turi kosulio simptomą?*“ turėtų pateikti visas ligas, kuriose kosulys yra bent vienas iš simptomų;
4. **sąvokų atitikimas** – sąvokų atpažinimas sukuriant jų tarpusavio ryšį. Pavyzdžiui, užklausoje žodžių junginys „*galvos skausmas*“ turėtų būti suprantamas kaip „*migrena*“;
5. **žinių atitikimas** – žinių atpažinimas ir panaudojimas atitinkamiems rezultatams gauti. Pavyzdžiui, užklausoje „*kiaulių gripas*“ = „*H1N1*“;
6. **natūraliosios kalbos atpažinimas** – užklauso, pateiktos natūraliąja kalba, supratimas bendrame užklauso kontekste. Pavyzdžiui, klausimai, kuriuose vartojami žodžiai „*kas*“, „*kur*“ ir „*kodėl*“ turėtų būti tinkamai įvertinami užklauso kontekste;
7. **svarbiausių dokumento dalių išrinkimas** – užklauso rezultatas turi pateikti ne tik dokumento nuorodą su tam tikrais raktiniais žodžiais, bet taip pat ir aktualiausių dokumento pastraipų ar skyrių nuorodas;
8. **užklauso pateikimas natūraliąja kalba** – naudotojas gali įvesti klausimą natūraliąja kalba, jam nereikia naudoti specialios užklauso kalbos. Pavyzdžiui, loginių sąlygų ar kabučių;
9. **dokumentų turinio analizė** – užklauso rezultatai pateikiami analizuojant dokumentų turinį, nesiremiant naudotojų elgsena, tam tikra statistika, sąsajomis ir kt.;

10. **pateikiamų rezultatų analizė** – pateikiant užklauso rezultatus remiamasi ne puslapio populiarumu, o tuo, kaip tiksliai jie atitinka pateiktą klausimą.

Išanalizavus šiuos požymius galima teigti, kad semantinės paieškos sistemose labai svarbu tinkamai suprasti užklausoje minimų objektų ir bendro jos konteksto prasmę. Aiškiai apibrėžta užklauso struktūra turi didelę įtaką paieškos rezultatų tikslumui ir naudotojo sugaišto laiko trukmei.

Remiantis šiais požymiais semantinėje paieškoje siekiama tinkamai suprasti užklauso objektų ir bendro jos konteksto prasmę analizuojant pateiktą naudotojo klausimą. Rezultatų tikslumui įtakos turi ir aiškiai apibrėžta užklauso struktūra. Jeigu užklauso struktūra yra griežtesnė ir ribota, naudotojui yra sunkiau įvesti klausimą, tačiau įvedus gaunami tikslesni rezultatai. Jeigu leidžiama klausimą vesti laisvesne forma, yra didesnė tikimybė, kad sistema klausimo nesupras ir negalės atsakyti.

1.3.6. Natūraliosios kalbos sąsajos apibrėžimas

Natūraliosios kalbos sąsaja (angl. *Natural Language Interface*, NKS) – tai paieškos sistema, leidžianti naudotojui įvykdyti paiešką pateikiant klausimus natūraliąja kalba (pvz., lietuvių, anglų, vokiečių). Vienas didžiausių NKS privalumų yra tai, kad naudotojas gali įvesti klausimą jam lengvai suprantama natūraliąja kalba. Tam nereikia mokintis specialių užklauso kalbų. Taigi, pagrindinis NKS uždavinys – informacijos paieškos procesą ontologijoje padaryti kuo paprastesnį ir lengviau suprantamą sistemos naudotojui.

Atsiradus NKS buvo manoma, kad tai viena perspektyviausių naudotojo sąsajų, suteikianti galimybę naudotojui komunikuoti su sistema. Pirmasis NKSDDB (angl. *Natural Language Interface Database*) prototipas pasirodė jau 1960-ųjų pabaigoje ir 1970-ųjų pradžioje, vėliau įvairių tyrimų ir bandymų istorija šioje srityje tęsėsi net penkis dešimtmečius. Dauguma per šį laikotarpį sukurtų įrankių buvo labai efektyvūs ir teikė dideles viltis, tačiau galiausiai jų naudojimas buvo pritaikytas tik konkrečiai dalykinei sričiai. Nors NKSDDB tyrimai vis dar yra vykdomi, tokių sistemų naudojimas nėra plačiai paplitęs ir tai nelaikoma tinkamiausiu ir patogiausiu būdu pateikti užklauso duomenų bazėms [27].

Siekiant tiksliau apdoroti informaciją internete ir sukūrus semantinį tinklą, NKS tyrimai buvo ir vėl atnaujinti. Semantinės paieškos sistemose duomenys saugojami naudojant *RDF* tinklo ontologijų kalbą. Norint vykdyti paiešką tokioje duomenų struktūroje reikėjo mokytis specialių užklauso kalbų arba naudotis įvairių rūšių naudotojo sąsajomis, tokiomis kaip raktiniais žodžiais grindžiamos paieškos sąsajos, grafinės užklauso kalbos sąsajos, meniu valdiklio sąsajos ir kt. [10].

Viena dažniausių problemų naudojant natūraliosios kalbos sąsajas yra semantinis neatitikimas (angl. *Semantic gap*), kai saugomi duomenys žinių bazėje ne visada aprašomi tokia struktūra, kokia sistemos naudotojas formuluoja klausimus. Tarkime, naudotojas gali formuluoti klausimus naudodamas sinonimus ar net ryšius, kurių žinių bazėje nėra, todėl jų tiesiogiai susieti su žinių bazėje saugomais duomenimis negalima. Siekiant išspręsti šią problemą naudojami įvairia forma aprašyti žodynai, kurie leidžia susieti naudotojo poreikių ir tai, kaip jie formuluoja klausimus su žinių bazės struktūra [19].

1.3.6.1. Natūraliosios kalbos sąsajų problemos

Natūraliosios kalbos sąsajos leidžia patogiai formuluoti semantines užklauso, tačiau realizuojant tokio tipo paieškos sistemas susiduriama su nemažai iššūkių. Juos galima skirstyti į septynias grupes, kurios detalčiau aprašomos šiame skyriuje.

Semantinis klausimų atsakinėjimas (angl. *Semantic Question Answering*, SQA) natūraliaja kalba padeda išspręsti du pagrindinius priegios prie semantinio tinklo reikalavimus [18]:

- **formali užklausų kalbų** (pvz., *SPARQL*);
- **konkreto žodyno išmanymas.**

SQA sistemos dažniausiai pateiktus natūraliaja kalba klausimus interpretuoja ir transformuoja į semantines užklausas. Tačiau sudėtingesnių ar rečiau pasitaikančių klausimų atveju gautos užklauso ne visada atitinka naudotojo ketinimus. Atlikus klausimų atsakinėjimo semantiniame tinkle apklausą buvo nustatytos pagrindinės problemos su kuriomis susiduriama interpretuojant klausimus natūraliaja kalba [17]:

1. **leksikos neatitikimas** (angl. *Lexical Gap*)

Natūraliaja kalba pateikto klausimo žodžių prasmė gali būti interpretuojama skirtingai, nes ta pati sąvoka gali turėti daug konkrečių arba abstrakčių reikšmių. Taip pat ir *RDF* duomenų modelis, skirtas aprašyti resursus, apibrėžiant jų savybes ir suformuojant grafą. Kadangi natūraliaja kalba pateikto klausimo struktūra gali neatitikti *RDF* grafo struktūros ir negali būti tiesiogiai susieta su klausimu natūraliosios kalbos *RDF* resursai gali būti aprašomi naudojant `rdfs:label` savybę. Nors klausimo žodžių sinonimai tam pačiam *RDF* resursui gali būti sumodeliuoti naudojant kelias skirtingas to resurso savybes, bet žinių bazėse dažniausiai nėra visų galimų terminų, apibrėžiančių tą subjektą. Išskiriamos kelios pagrindinės problemos, sukeliančios sąvokų neatitikimus:

- resursai, kurių išraiška ontologijoje sutampa, gali naudoti tą pačią *RDF* duomenų modelio savybę;
- sąvoka gali turėti keletą skirtingų išraiškų, o *RDF* duomenų modelio savybė gali skirtis nuo klausimo naudojamo žodžio. Pavyzdžiui, lietuvių žodyne įvedus žodį „žemė“ pateikiama net penki sinonimai;
- žodžiai gali būti kaitomi morfologiškai. Pavyzdžiui, pagal laiką „bėga, bėgo“;
- rašybos klaidos, kai skiriasi vienas ar daugiau simbolių.

Taigi, jei klausime naudojamas žodis skiriasi nuo to, kuris naudojamas žinių bazių savybėse, tai vadinama leksikos neatitikimu. Šis neatitikimas padidina riziką į naudotojo pateiktą klausimą atsakyti neteisingai, nes natūraliaja kalba pateiktos sąvokos gali būti identifikuojamos neteisingai ontologijoje.

2. **dviprasmybė** (angl. *Ambiguity*)

Priešingai nei leksikos neatitikimo problema, klausime naudojami dviprasmiški žodžių junginiai gali neigiamai paveikti jo atsakymo tikslumą. Natūraliaja kalba pateikto klausimo žodžių prasmė gali būti skirtingai suprantama dėl dviprasmybių, kurios skirstomos į dvi rūšis:

- **sintaksės** dviprasmybės gali atsirasti tada, kai klausimuose žodžiai pateikti netinkama seka ar netaisyklingai vartojami žodžių junginiai. Vienas klausimo sintaksės neaiškumo pavyzdžių yra „Ar skrydis lėktuvu gali būti pavojingas?“. Šį klausimą galima suprasti dviem skirtingais būdais dėl jo gramatinės struktūros. Žodžių junginys „skrydis lėktuvu“ gali apibūdinti lėktuvo skridimo veiksmą arba transporto priemonę, kuri gali skraidyti;

- **leksikos** dviprasmybės gali kilti dėl atskirų klausimo žodžių interpretavimo, o ne dėl jo bendros struktūros. Pagrindinė priežastis yra tokia, kad dauguma natūraliosios kalbos žodžių gali turėti tokią pačią išraišką, tačiau jų reikšmė gali atitikti visiškai kitą prasmę. Vienas iš leksikos dviprasmybių žodžių pavyzdžių yra „*austrės*“, kuris lietuvių kalbos žodyne gali reikšti jūros moliuskų pavadinimą, gurmanišką patiekalą arba Austrijos šalies moteriškos giminės tautybės apibūdinimą.

Naudojant nedidelės apimties žinių bazes, dviprasmybių poveikis gali būti mažai pastebimas klausimo atsakymo tikslumui. Tačiau kai naudojami didesnės apimties žodynai, tai gali tapti aktualia problema.

3. **daugiakalbiškumas** (angl. *Multilingualism*)

Informacija žiniatinklyje yra išreiškiama įvairiomis skirtingomis kalbomis. *RDF* duomenų modelio resursai gali būti aprašomi keliomis skirtingomis kalbomis. Kadangi nėra numatytos vienos kalbos, naudojamos žiniatinklyje interneto naudotojai naršydami internete gali ieškoti informacijos įvairiomis kalbomis. Dėl šios priežasties ir semantinės klausimų-atsakymų sistemos turi turėti lankstesnę požiūrį į kelių kalbų įvestį. Daugiakalbės *SQA* sistemos turėtų leisti pateikti klausimus įvairiomis kalbomis, kurios gali skirtis nuo kalbos, kuria saugomi duomenys ontologijose.

4. **kompleksinės užklausos** (angl. *Complex Queries*)

Dažniausiai į paprastus naudotojo klausimus natūraliąja kalba galima atsakyti transformuojant juos į *SPARQL* užklausos sąlygos dalį, kitaip vadinamą tripleto šablonu. Jį sudaro trys skirtingos dalys: objektas, subjektas ir predikatas. Tačiau didesnė problema atsiranda tada, kai sudėtingesnio klausimo rezultatu pateikti turi būti naudojami ir kiti užklausos elementai. Pavyzdžiui, sukurta užklausa turi atitikti tam tikrus apribojimus, naudojant sudėtingesnius operatorius rezultatų grupavimui, rikiavimui, filtravimui ir kt.

5. **paskirstytos žinios** (angl. *Distributed Knowledge*)

Natūraliąja kalba pateiktą sudėtingesnę klausimą transformavus į užklausą joje naudojamos sąvokos gali būti pateikiamos paskirstytais *RDF* duomenų modeliais. Tokiu atveju, jei *RDF* resursui sumodeliuoti žinių bazėse randamos ne visos reikalingos elementų savybės, gali trūkti reikalingos informacijos atsakymui į pateiktą klausimą. Tačiau atskiruose duomenų rinkiniuose, kuriuose randamas bent vienas atitinkamas šaltinis naudojant sujungtų duomenų rinkinių atvejį šią problemą būtų galima išspręsti sukuriant atitinkamas nuorodas (pvz., „*sameAs*“, „*equivalentClass*“ arba „*equivalentProperty*“).

6. **procedūriniai, laikini ir erdviniai klausimai** (angl. *Procedural, Temporal and Spatial Questions*)

Procedūriniai klausimai

Faktiniai klausimai į kuriuos atsakymas gali būti pateikiamas teiginiu „*taip*“ arba „*ne*“ yra patys paprasčiausi, nes jie gali tiesiogiai atitikti *SPARQL* užklausos struktūrą. Tačiau kartais naudotojo pateikiami klausimai nėra tokie paprasti ir atsakymai į juos turi būti detalesnis, reikalaujantys papildomo klausimo apdoravimo ir gilesnio jo supratimo. Atsakant į tokius klausimus sistema galėtų pateikti instrukciją, pavyzdžiui, surinkti baldus ar pagaminti patiekalą.

Laikini klausimai

Laikiniai klausimai, kuriems atsakyti reikia apdoroti užuominas apie laiką. Tai galėtų būti klausimai apie įvykius, kurie vyko tam tikru laikotarpiu arba faktus, kurie egzistavo tam tikru laikotarpiu, pavyzdžiui, kokias pareigas tam tikras asmuo užėmė prieš metus. Vienas šių klausimų apdorojimo metodų yra *TEQUILA*, kuris turi keturis skirtingus etapus:

- nustatoma ar klausimas yra laikinas;
- klausimas išskaidomas ir perrašomas į paprastą klausimą, nesiremiant pateiktais laiko apribojimais;
- atsakymai į antrinius klausimus gaunami iš pagrindinio *KB-QA* variklio;
- naudojamas laiko apribojimų samprotavimas, kad būtų apskaičiuojamas galutinis atsakymas į pateiktą klausimą.

Erdviniai klausimai

Objektų lokacija *RDF* duomenų modelyje gali būti išreikšta dviejų dimensijų koordinatėmis – platumu ir ilguma. Taip pat ontologijose gali būti modeliuojami objektų santykiai erdvėje. Į tokius klausimus atsakyti yra lengviau, naudotojai taip pat dažniausiai prašo ne tikslių koordinatčių, bet santykio, pavyzdžiui, tarp kurių miestų atstumas yra didesnis. Tokios sąsajos kaip *Younis* naudoja inversinį indeksą, skirtą įvardytų esybių atpažinimui, kurios papildo semantinius duomenis erdviniais santykiais, tokiais kaip kryžminimas (angl. *crossing*), įtraukimas ir artumas (angl. *nearness*). Tada šią informaciją galima panaudoti ir išrenkant duomenis *SPARQL* užklausomis.

7. Šablonai (angl. *Templates*)

Sudėtingesniuose klausimuose, kai gautoje *SPARQL* užklausoje yra daugiau nei vienas grafo šablonas, užfiksuoti pagrindinės užklaustos struktūrą reikia taikyti sudėtingesnius metodus. Dabartiniai tyrimai remiasi vienu iš šių būdų:

- **šablonais pagrįstas**, kuris susieja įvestą klausimą su rankiniu ar automatiniu būdu sukurtu *SPARQL* užklaustos šablonu;
- **nenaudojant šablonų**, kuris siekia sukurti *SPARQL* užklausas, atsižvelgiant į klausimo sintaksinę struktūrą.

Taigi, klausimo natūraliąja kalba susiejimas su pateikiamos užklaustos struktūra yra viena didžiausių problemų.

1.3.6.2. Užklausų sudarymo metodai

Įvairiose naudotojo sąsajose, skirtose pateikti klausimus natūraliąja kalba, siekiant užtikrinti rezultatų teisingumą transformuojant klausimą į *SPARQL* užklausą, taikomi įvairūs kokybės užtikrinimo metodai. Viena dažniausių problemų naudojant NKS yra semantinis neatitikimas (angl. *Semantic gap*), kai saugoma duomenų struktūra neatitinka naudotojo suformuluoto klausimo [12].

```
dbr:Graikija dbp:narys dbr:Europos_Sajunga .  
dbr:Prancūzija dbp:narys dbr:Europos_Sajunga .
```

6 pav. Saugomos duomenų struktūros pavyzdys

Pavyzdyje pateikiamas tam tikros saugomos duomenų struktūros pavyzdys, apibrėžiantis tokius faktus, kaip „Graikija yra Europos sąjungos narė“ ir „Prancūzija yra Europos sąjungos narė“. Tačiau SPARQL užklausoje šį informacija yra išreiškiama kitaip (žr. 7 pav.).

```
dbp:Graikija dct:subjektas dbc:Europos_Sajungos_narė .  
dbp:Prancūzija dct: subjektas dbc:Europos_Sajungos_narė .
```

7 pav. SPARQL duomenų išraiškos pavyzdys

Taigi, vietoje savybės „*dbp: narys*“ informacijai išreikšti klausime naudojama „*dbc: Europos_Sajungos_narė*“ klasė. Tai rodo, kad apskritai neįmanoma suformuoti teisingos SPARQL užklauso formos, turint tik klausimą. Žemiau pateikiami įvairūs užklauso kūrimo metodai, kurie klasifikuojami pagal tai, kaip suformuojama SPARQL užklausa [12].

1. Užklauso kūrimas naudojant šablonus (angl. *Query construction using templates*)

Kai kurios sistemos naudoja šablonus SPARQL užklauso generuoti, t.y. iš anksto nustatyti užklauso rinkiniai, kuriuose reikia užpildyti tam tikras tuščias vietas. Pavyzdžiui, tokia sąsaja kaip QAKiS gali pasirinkti užklauso tik su vienu tripletu. ISOFT sąsaja naudoja taip pat nedidelį šablonų rinkinį, kurį sudaro: užklauso forma ASK, naudojanti vieną tripletą, keletas paprastų SELECT užklauso formų su vienu ar daugiau tripleto šablonų ir tam tikri agregavimo operatoriai, tokie kaip COUNT, ORDER BY arba FILTER, kuriuose naudojamas tik vienas tripleto šablonas. Didžiausias šio metodo trūkumas yra tai, kad ne visus klausimus galima apdoroti naudojant šablonus.

2. Užklauso sudarymas vadovaujantis klausimo analizės informacija (angl. *Query construction guided by information from the question analysis*)

Tokio tipo sistemose pirmiausia klausimas yra išanalizuojamas ir pagal tai suformuojama SPARQL užklausa. Tarkime, tokios NKS kaip FREyA ir Intui3 pradeda nuo klausimo segmentavimo ir vėliau frazes susieja su atitinkamais resursais. Tuomet šie resursai yra sujungiami į tripletus, atsižvelgiant į klausimo segmentų tvarką. Vienas didžiausių šio metodo trūkumų yra tai, kad dažniausiai daroma prielaida, jog iš klausimo struktūros galima sugeneruoti SPARQL užklauso nežinant, kaip duomenys yra saugomi.

3. Užklauso sudarymas naudojant semantinę analizę (angl. *Query construction using semantic parsing*)

Semantiniai analizatoriai yra tam tikras užklauso analizės tipas, susiejantis sintaksines taisykles su semantine klausimo struktūra. Išanalizavus klausimą gaunama jo prasminė interpretacija. Pagrindiniai šio metodo privalumai yra tai, kad galima gauti tiesioginę semantinio klausimo išraišką, kuri gali apimti konkrečias ir abstrakčias klausimo interpretacijas. Trūkumas yra toks, kad klausimai turi būti tinkamai suformuluoti, t.y. klausime vartojamos sąvokos neturi iškreipti jo prasmės. Viena didžiausių problemų yra tai, kad kiekvienas leksikos elementas turi turėti atitinkamą semantinę išraišką.

4. Užklauso sudarymas naudojant mašininį mokymąsi (angl. *Query construction using machine learning*)

Tokia natūraliosios kalbos sąsaja kaip CASIA naudoja mašininio mokymosi metodą visam kokybės užtikrinimo procesui. Klausimo analizės etape klausimui segmentuoti ir frazėms išskirti naudojamos tokios funkcijos kaip sąvokos padėtis, sąvokos žyma, priklausomybės tipas priklausomybių medyje ir

kt. Sąvokų susiejimo etape resursai yra susiejami su segmentais ir sukuriamos naujos funkcijos: resurso tipas ir resurso panašumas. Gautos savybės naudojamos rasti labiausiai tikėtiną ryšį tarp klausimo segmento ir resurso žinių bazėje. Surasti ryšiai naudojami *SPARQL* užklausai generuoti.

5. Užklausos sudarymas semantinės informacijos perdavime (angl. *Query construction relaying on semantic information*)

Kokybės užtikrinimo sistema *SINA* buvo sukurta spręsti raktinių žodžių užklausas, t.y. vietoje klausimo įterpimo, pavyzdžiui, „*Kas yra Belgijos sostinė?*“ naudotojas gali įterpti raktinius žodžius tokius, kaip „*sostinė Belgija*“. Tai reiškia, kad santykis tarp skirtingų resursų nėra toks aiškus, kaip natūraliosios kalbos klausime. *SINA* pirmiausia susegmentuoja klausimą ir suranda susijusius resursus, kuriuos nustačius sukuriami užklausa. Kiekvienam egzemplioriui ar klasei sukuriami viršūnė, o kiekvienai savybei sukuriamos briaunos, kurios naudojamos viršūnėms sujungti. Jei briaunos negali būti sujungiamos dėl tam tikrų ribojimų, tada sukuriami viena arba dvi naujos viršūnės, atitinkančios tuos kintamuosius. Kombinuojant duomenis gali būti sukuriama ne vienas duomenų grafai. Visi galimi duomenų modeliai yra išreiškiami *SPARQL* užklausa ir vykdomi. Šio metodo privalumas yra tai, kad grafai sudaromas nenaudojant klausimo sintaksės. Tačiau tokių metodų trūkumas toks, kad šis procesas yra paremtas sudėtingais skaičiavimais ir nėra atsižvelgiama į klausimo sintaksę.

6. Užklausų sudarymas nenaudojant *SPARQL* (angl. *Approach not using SPARQL*)

Tokios natūraliosios kalbos sąsajos kaip *Treo* ir *SemSek* nesukuria *SPARQL* užklausos, o vietoj to ieško informacijos žinių bazėse sekant nuorodomis. Pavyzdžiui, pateikus klausimą „*Kuriame mieste gimė Europos Sąjungos lyderis?*“, *SemSek* sistema pirmiausia nustato klausime „centrinę sąvoką“, šiuo atveju „*Europos Sąjunga*“. Naudojant priklausomybių medį ir pradėdant nuo „centrinės sąvokos“, sudaromas rastų atitikimų sąrašas. Remiantis pateiktu pavyzdžiu atitinkančių terminų sąrašas būtų toks: „*Europos Sąjunga*“, „*lyderis*“, „*gimė*“, „*miestas*“. Tada ieškoma pirmojo termino atitikčių, pavyzdžiui, resursas *dbp: Europos_Sajunga*, o atitinkamam šio resurso *URI* http://dbpedia.org/resource/Europos_Sajunga nurodoma ieškoti visų jam atitinkamų savybių. Jei viena iš surastų savybių yra panaši į antrąją sąrašo sąvoką, tuomet sukuriami nauja kryptis algoritmo paieškai. Priešingu atveju, ši kryptis toliau neanalizuojama. Tokiu būdu paieška vykdoma, kol surandamas atsakymas į klausimą.

1.4. Tyrimo objekto naudotojų analizė

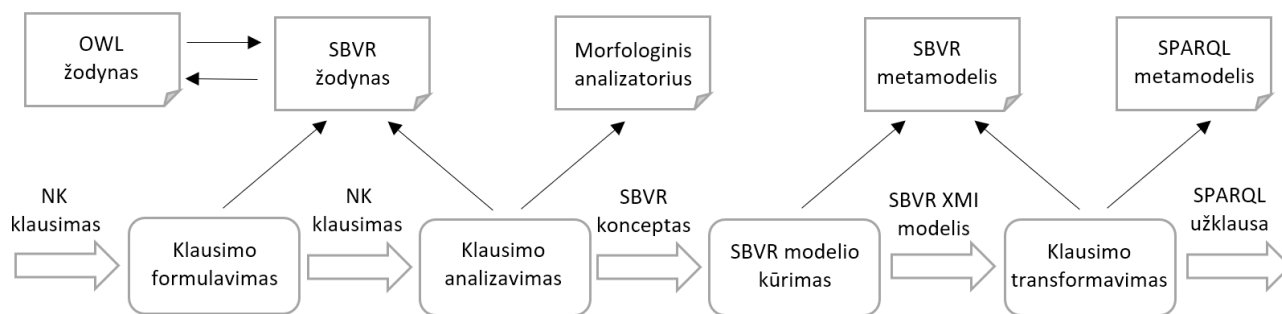
Naudotojais gali būti bet kurios organizacijos ar įmonės darbuotojai, kurie naudoja ontologijas duomenų saugojimui ar analizei. Pagrindinis šių naudotojų tikslas – rasti norimą informaciją ontologijoje pateikiant klausimą natūraliąja kalba. Natūraliosios kalbos sąsajos naudotojų aibę gali sudaryti tiek didelę, tiek mažą kompetenciją naudotis tokiomis paieškos sistemomis turintys asmenys. Dėl šios priežasties tyrimo metu yra orientuojamasi į tai, kad sąsajos naudotojas galėtų pateikti klausimus lietuvių kalba.

1.5. Esamų problemos sprendimo metodų analizė

Siekiant informacijos paieškos procesą ontologijoje padaryti kuo paprastesnį ir lengviau suprantamą, turėtų būti naudojamos kuo patogesnės naudotojo sąsajos. Vienos iš tokių yra sąsajos, leidžiančios pateikti klausimus natūraliąja kalba. Nors natūraliosios kalbos sąsajų užklausų formulavimas yra lengvai išmokstamas ir greitai įsimenamas, sistemos negali interpretuoti bet kokio klausimo – jis turėtų atitikti ontologijoje naudojamas sąvokas ir ryšius. Dėl to NKS naudojami įvairūs metodai, kurie padeda

lengviau suformuluoti klausimą, kurį sistema galėtų suprasti ir į jį atsakyti. Ankstesniame skyrelyje buvo analizuojami įvairūs užklausų kūrimo metodai (žr. 1.3.6.2 skyrių). Šio tiriamojo projekto metu koncentruojamasi į užklausos sudarymą vadovaujantis klausimo analizės informacija, kai pirmiausia klausimas yra išanalizuojamas ir pagal tai suformuojama *SPARQL* užklausa. Šiame skyrelyje analizuojamos jau esamos natūraliosios kalbos sąsajos ir jose naudojami metodai. Analizuojamos pasirinktos sistemos, kuriose užklausos sudaromos vadovaujantis klausimo analizės informacija.

- ***QuestIO*** – tai natūraliosios kalbos sąsaja, leidžianti pateikti įvairios struktūros klausimus anglų kalba. Vienas didžiausių šios sąsajos privalumų – lengvai suprantama naudotojui ir nereikalaujanti papildomų apmokymų ja naudotis. Taip pat *QuestIO* yra lengvai perkeliama, kurios žodynas kuriamas automatiškai generuojant žodžių junginių sąrašą iš morfologiškai normalizuotų ontologinių leksikalizacijų. *QuestIO* trūkumas – negali susieti natūraliosios kalbos žodžių junginių su ontologijos išteklių deriniais [9];
- ***FREyA*** – tai natūraliosios kalbos sąsaja, leidžianti lanksčiau formuluoti klausimus anglų kalba, netaikant griežtos turinio struktūros. *FREyA* sąsaja buvo sukurta *QuestIO* autorių, siekiant lengviau suprasti pateikto klausimo semantiką ir išvesti aiškius bei trumpus jo atsakymus. *FREyA* yra perkeliama NKS, nereikalaujanti rankinio konfigūravimo. Žodyno sukūrimui vykdomos *SPARQL* užklausos semantinėje saugykloje. Pagrindinis trūkumas – *FREyA* negali susieti žodžių junginių natūraliąja kalba su ontologijų resursais [8];
- ***ORAKEL*** – tai natūraliosios kalbos sąsaja, skirta suprasti sudėtinės semantines konstrukcijas, tokias kaip kiekybiniai dydžiai, jungtukai ar neiginiai. *ORAKEL* yra perkeliama, tačiau, skirtingai nei *QuestIO* ir *FREyA*, tam reikalingas rankinis konfigūravimas. Natūraliosios kalbos sąvokų susiejimas su ontologija apibrėžiamas kuriant kalbos struktūras, vadinamas „*subkategorizavimo rėmais*“, t.y. veiksmažodžiai su argumentais. Šioje NKS dalis žodyno automatiškai sukuriama iš pagrindinės ontologijos. *ORAKEL* leidžia susieti kalbos struktūras tarpusavio ryšiais ontologijoje ir atsakyti į klausimus, kurie tiesiogiai neatitinka tik vienos ontologijos struktūros [7];
- ***Querix*** – tai natūraliosios kalbos sąsaja, skirta atsakyti į klausimus anglų kalba. *Querix* yra lengvai perkeliama ir nereikalaujanti konfigūracijos rankiniu būdu. Žodynas sudaromas automatiškai iš ontologijų ir yra papildomas naudojant *WordNet* žodyną. *Querix* neturi priemonių natūraliosios kalbos sąvokoms atvaizduoti susiejant ontologijos resursus. Taip pat šis sprendimas nesistengia automatiškai išspręsti natūraliosios kalbos neatitikimų, tačiau naudotojui pateikiami patikslinimų dialogai [20];
- ***SBVR pagrįstą NKS*** – tai *SBVR* (angl. *Semantics of Business Vocabulary and Business Rules*) žodynas pagrįstas natūraliosios kalbos sąsaja. Vienas didžiausių šios sąsajos privalumų – tai pritaikomumas skirtingoms kalboms ir gebėjimas atsakyti į klausimus, kurie nėra tiesiogiai susiję su ontologija [26]. Žemiau pateikiamas *SBVR* pagrįsto NKS pagrindiniai komponentai (žr. 8 pav.).



8 pav. SBVR pagrįstos NKS komponentai

Remiantis išanalizuotu procesu buvo nustatytas pagrindinis šio sprendimo trūkumas – norint naudoti šią sąsają jai reikalingas papildomas *SBVR* žodyno paruošimas, kuris reikalauja nemažai rankinio darbo. Šis žodynas naudojamas klausimo formulavimui ir analizavimui.

Analizuojant visas aptartas natūraliosios kalbos sąsajas ontologijoms buvo nustatyti pagrindiniai reikalavimai, kuriais remiantis atliktas esamų sprendimų palyginimas, pateikiamas 2 lentelė.

2 lentelė. Natūraliosios kalbos sąsajų lyginamoji analizė

Sprendimas / Savybė	<i>QuestIO</i>	<i>FREyA</i>	<i>QRAKEL</i>	<i>Querix</i>	SBVR paremta NKS	Kuriamas metodas
Palaikomas klausimo įvedimas natūraliąja kalba	+	+	+	+	+	+
Pritaikymas lietuvių kalbai	-	-	-	-	+	+
Sąsaja gali pateikti naudotojui dialogą apie klausimo patikslinimą	-	-	-	+	-	+
Kontekstinio meniu naudojimas	-	-	-	+	+	+
Naudotojui nereikia mokintis kaip naudotis sąsaja	-	+	+	+	+	+
Sąsaja gali interpretuoti pateiktą klausimą	-	+	+	+	+	+
Perkeliamumas	+	+	+	+	+	+
Automatinė konfigūracija	+	+	-	+	-	+

Atliktos lyginamosios analizės metu buvo nustatyta, kad nei vienoje išanalizuotoje naudotojo sąsajoje nėra realizuoti visi pagrindiniai tiriamojo projekto metu nustatyti reikalavimai. Vienas pagrindinių tiriamojo projekto reikalavimų – tai naudotojo sąsajos pritaikymas lietuvių kalbai. Analizuojant jau sukurtus sprendimus nustatyta, kad vienintelė *SBVR* pagrįsta NKS yra pritaikyta lietuvių kalbai. Taip pat nustatyta, kad visos analizuotos sistemos yra perkeliamos ir gali būti pritaikomos skirtingoms dalykinėms sritims. Tačiau toks pritaikymas ne visose sistemose yra patogus. Dalis jų leidžia žodyną

išgauti automatiškai. Žodynas gali būti išgaunamas vykdant *SPARQL* užklausas ontologijoje. Kitiems metodams žodyną reikia parengti rankiniu būdu, t.y. tam reikalinga rankinė konfigūracija. Vienas iš tokių yra *SBVR* grindžiamas sprendimas, kuriam reikia parengti *SBVR* žodyną, atitinkantį ontologijos schemą. Siekiant padėti naudotojams lengviau suformuluoti klausimus, daugumoje sistemų naudojami kontekstiniai meniu. Kita naudinga savybė yra patikslinantis dialogas. Jeigu sprendimas negali vienareikšmiškai interpretuoti klausimo, tuomet sistema gali paprašyti naudotojo klausimą patikslinti. Apibendrinant atliktos lyginamosios analizės rezultatus galima teigti, kad nei viena išanalizuota naudotojo sąsaja neatitinka tiriamojo projekto metu nustatytų reikalavimų.

1.6. Siekiamo sprendimo apibrėžimas

Sukurtas ir formaliai aprašytas metodas, leidžiantis naudotojui vykdyti paiešką pateikiant klausimą lietuvių kalba ir transformuoti jį į užklausą ontologijoje. Metodą sudaro algoritmai, skirti išanalizuoti natūraliąją kalbą pateiktą klausimą remiantis *OWL* žodyno sąvokomis ir transformuoti jį į *SPARQL* užklausą. Metodo veikimui patikrinti realizuojamas natūraliosios kalbos sąsajos ontologijoms prototipas, kuris leidžia įvesti klausimą lietuvių kalba, įvykdyti paiešką ir peržiūrėti gautus paieškos rezultatus. Šiame darbe sukurtas sprendimas turėtų būti pranašesnis už egzistuojantį *SBVR* paremtą metodą tuo, kad jam nereikėtų rankiniu būdu parengti žodyno.

Pagrindiniai kuriamos natūraliosios kalbos sąsajos ontologijoms privalumai:

1. natūraliosios kalbos sąsaja leidžia vykdyti semantinę paiešką lietuvių kalba;
2. naudotojo sąsajai nereikia papildomai ruošti žodyno, reikalingas tik ontologijos sukūrimas, kuris nereikalauja daug rankinio darbo.

1.7. Analizės išvados

1. Semantinė paieška, palyginti su įprasta, raktiniais žodžiais grindžiama paieška, leidžia gauti tikslesnius rezultatus, nes ieškoma ne tik pagal sintaksinį atitikimą, bet ir pagal prasmę ontologijoje, kurioje aprašomi resursai ir jų ryšiai atitinka realaus pasaulio objektus.
2. Išanalizavus esamus semantinės paieškos vykdymo sprendimus buvo nustatyta, kad daugiausiai galimybių turi *GraphDB* semantinė saugykla, kuri suteikia galimybes vykdyti greitesnę paiešką, nes išvedimai yra vykdomi duomenų įrašymo metu. Taip pat *GraphDB* kartu gali būti naudojama ir kaip saugykla.
3. Viena iš semantinės paieškos savybių yra galimybė suprasti natūraliąją kalbą pateiktus klausimus ir į juos atsakyti. Naudotojams tai yra vienas iš patogiausių būdų vykdyti paiešką. Tačiau kuriant natūraliosios sąsajos sistemas yra susiduriama su įvairiomis problemomis, tokiomis kaip dviprasmybės, klausime naudojamų terminų ir ontologijos schemos neatitikimas ir kt.
4. Viena iš svarbių natūraliosios kalbos sąsajų dalių yra žodynas, kuris naudojamas formuluojant klausimus ir juos transformuojant į užklausas. Jame taip pat gali būti aprašomi sinonimai, kurie padeda formuluoti klausimus lanksčiau. Žodynas tokiose sistemose gali būti sudaromas automatiškai arba rankiniu būdu. Atlikus esamų natūraliosios kalbos sąsajų analizę, buvo nustatyta, kad dauguma tokių sistemų pritaikytos anglų kalbai. Tik vienintelė *SBVR* pagrįstą NKS yra pritaikoma lietuvių kalbai, tačiau pagrindinis trūkumas yra tai, kad jos paruošimui papildomai reikia aprašyti žodyną, kuris atitiktų ontologijos schemoje aprašytus terminus.

2. Natūraliosios kalbos sąsajos ontologijoms reikalavimų specifikacija ir projektas

2.1. Natūraliosios kalbos sąsajos ontologijoms reikalavimų specifikacija

Atlikus jau esamų natūraliosios kalbos naudotojo sąsajų, skirtų ontologijoms, analizę buvo nuspręsta, kad šio darbo metu bus kuriamas naujas metodas, leidžiantis naudotojui vykdyti paiešką pateikiant klausimus lietuvių kalba ir transformuojant juos į ontologijos užklausą. Šio sprendimo veikimas realizuojamas sukuriant NKS ontologijoms prototipą, skirtą įvesti klausimą, įvykdyti paiešką ir peržiūrėti pateiktus atliktos paieškos rezultatus. Pagrindinis šio etapo tikslas – apibrėžti kuriamo metodo, skirto lietuvių kalba pateiktų klausimų apdorojimui ontologijoje, pagrindinius aspektus ir juos struktūrizuoti. Pagal surinktus kuriamo inžinerinio sprendimo reikalavimus aprašoma jų specifikacija ir atliekamas būsimos sistemos projektavimas, naudojant *UML* (angl. *Unified Modeling Language*) kalbą. Remiantis sukurtu NKS ontologijoms projektu tolimesniuose tiriamojo projekto etapuose kuriamas jo funkcionavimui realizuoti reikalingas prototipas.

2.1.1. Funkciniai reikalavimai

Atlikus esamų NKS ontologijoms analizę buvo nustatyta, kokie panaudojimo atvejai (PA) reikalingi aprašyti kuriamo sprendimo funkcinis reikalavimus. PA modeliavimas leidžia aprašyti tai, ką turi daryti kuriamas sprendimas: identifikuojami analizuojamoje veikloje dalyvaujantys aktoriai ir pagrindinės jų atliekamos funkcijos bei nustatomi jų tarpusavio ryšiai. Kuriamai NKS buvo išskirti dviejų tipų aktoriai:

1. **sąsajos naudotojas** – šio tipo aktorius gali įvesti klausimą ir inicijuoti jo paieškos vykdymą bei peržiūrėti gautus rezultatus;
2. **semantinė saugykla** – šio tipo aktorius yra atsakingas už ontologijų, kurios naudojamos klausimų formulavimui ir paieškos vykdymui bei apdorojimui, valdymą ir saugojimą.

Atsižvelgiant į kiekvieno tipo aktoriaus poreikius ir galimybes, kuriamame sprendime buvo identifikuoti pagrindiniai PA ir jų sąsajos su aktoriais. Žemiau pateikiama kuriamos natūraliosios kalbos sąsajos ontologijoms prototipo PA diagrama, aprašanti pagrindinius kuriamo sprendimo funkcinis reikalavimus ir juose dalyvaujančius aktorius (žr. 9 pav.).



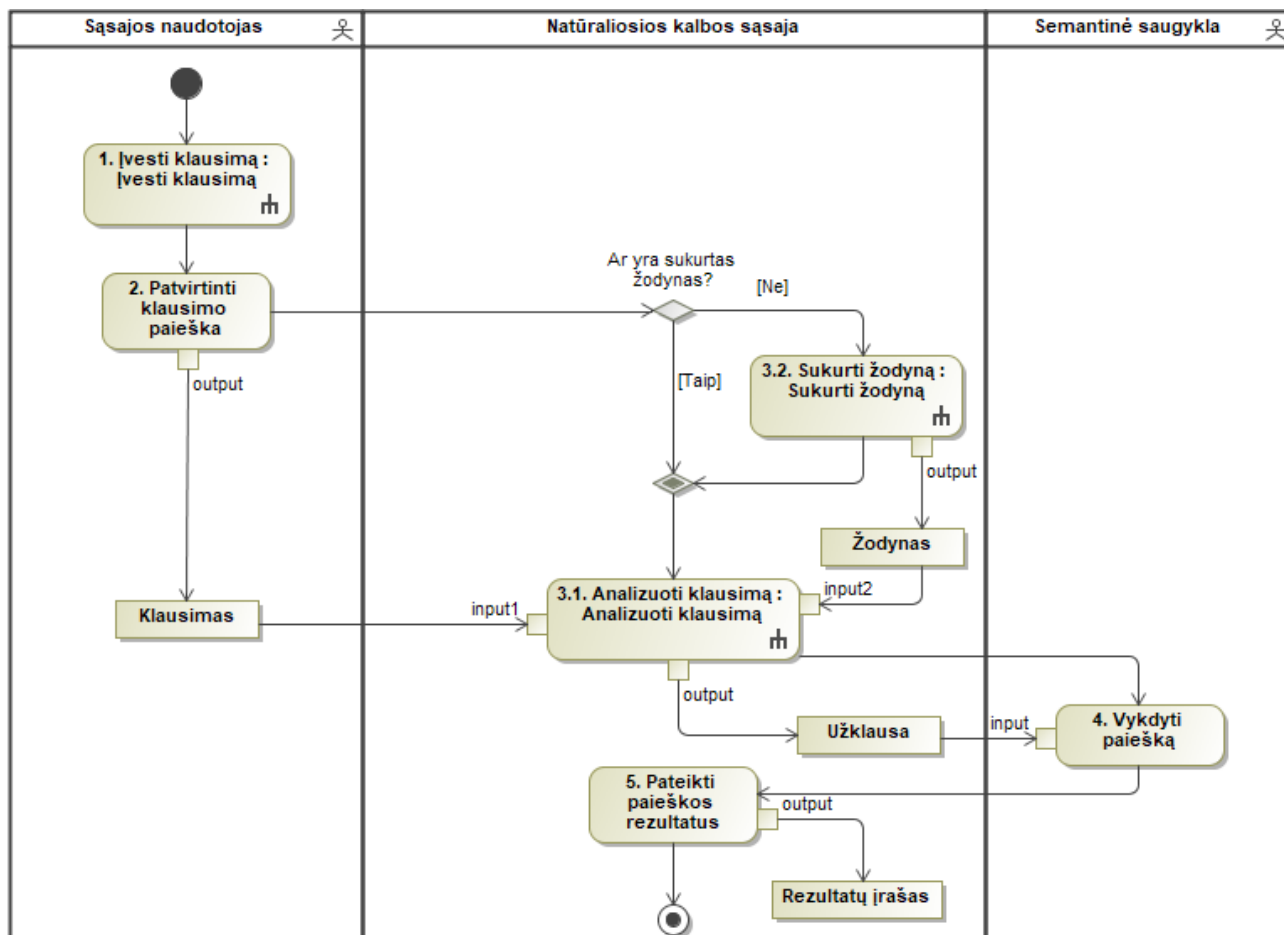
9 pav. Natūraliosios kalbos sąsajos ontologijoms panaudojimo atvejų diagrama

2.1.1.1. Panaudojimo atvejų specifikacijos ir veiklos modeliai

Pagrindinį kuriamos sąsajos funkcionalumą apibūdina PA „*Vykdyti paiešką*“ (žr. 9 pav., PA1), kuri inicijuoja sąsajos naudotojas. Vykdamas šį panaudojimo atvejį dalyvauja ir semantinės saugyklos aktorius. Šioje veikloje sąsajos naudotojas turi įvesti klausimą ir patvirtinti paiešką. Tai atlikus, sukurtas NKS algoritmas tikrina įvestą klausimą, remiantis semantinės saugyklos žodynu ir transformuoja jį į užklausą. Semantinėje saugykloje įvykdžius užklausą, sukurta NKS pateikia atliktos paieškos rezultatų sąrašą. Šį funkcinį reikalavimą išplečia PA „*Gauti žodyną iš ontologijos*“ (žr. 9 pav., PA4) ir apima PA „*Įvesti ir tikrinti klausimą*“ ir „*Analizuoti klausimą*“ (žr. 9 pav., PA2 ir PA3). Žemiau pateikiama PA „*Vykdyti paiešką*“ specifikacija (žr. 3 lentelė) ir grafiškai jį aprašanti veiklos diagrama (žr. 10 pav.).

3 lentelė. Panaudojimo atvejo „Vykdyti paiešką“ specifikacija

PA1 „Vykdyti paiešką“		
Aprašymas: Naudotojas vykdo paiešką pateikęs klausimą natūraliąja kalba.		
Prieš sąlyga:	Naudotojas turi būti pradiniam sąsajos lange.	
Sužadinimo sąlyga:	Naudotojas turi įvesti klausimą ir patvirtinti.	
Aktorius:	Sąsajos naudotojas Semantinė saugykla	
Susiję PA	Išplečiantys PA	PA4 „Gauti žodyną iš ontologijos“
	Apimami PA	PA2 „Įvesti ir tikrinti klausimą“ PA3 „Analizuoti klausimą“
	Specializuoti PA	Nėra
Po sąlyga:	Paieška įvykdyta, naudotojui pateikiami rezultatai.	

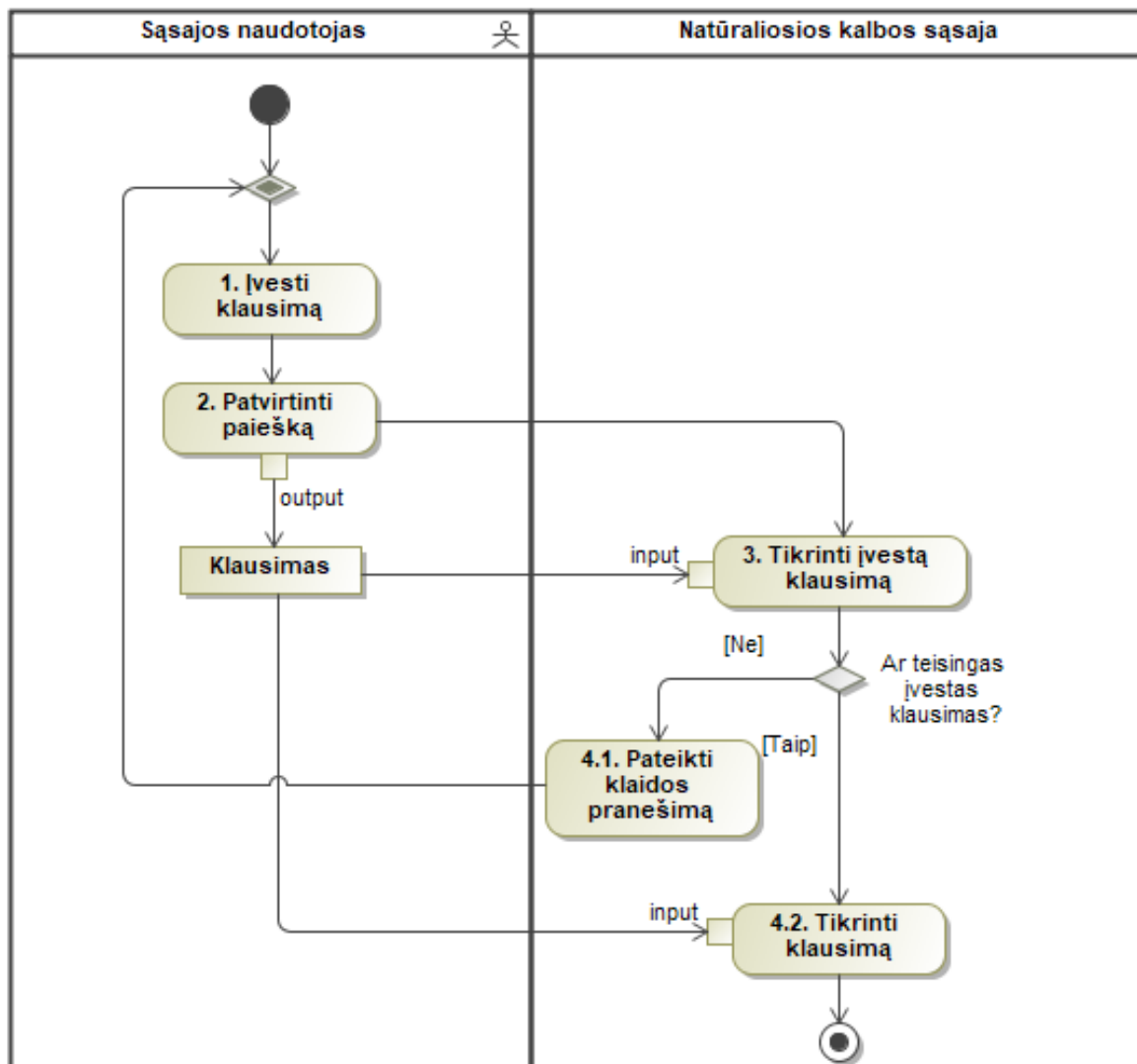


10 pav. Panaudojimo atvejo „Vykdyti paiešką“ veiklos diagrama

Semantinės paieškos vykdymo procesas pradedamas nuo klausimo įvedimo, kurį atlieka sąsajos naudotojas. Teisingai pateiktas klausimas yra pradedamas analizuoti natūraliosios kalbos sąsajoje. Žemiau pateikiama PA „Įvesti ir tikrinti klausimą“ specifikacija (žr. 4 lentelė) ir grafiškai jį aprašanti veiklos diagrama (žr. 11 pav.).

4 lentelė. Panaudojimo atvejo „Įvesti ir tikrinti klausimą“ specifikacija

PA2 „Įvesti ir tikrinti klausimą“		
Aprašymas: Naudotojas įveda klausimą.		
Prieš sąlyga:	Naudotojas turi būti pradiniame sąsajos lange.	
Sužadinimo sąlyga:	Naudotojas nori vykdyti paiešką.	
Aktorius:	Sąsajos naudotojas	
Susiję PA	Išplečiantys PA	Nėra
	Apimami PA	Nėra
	Specializuoti PA	Nėra
Po sąlyga:	Klausimas įvestas ir patikrintas.	



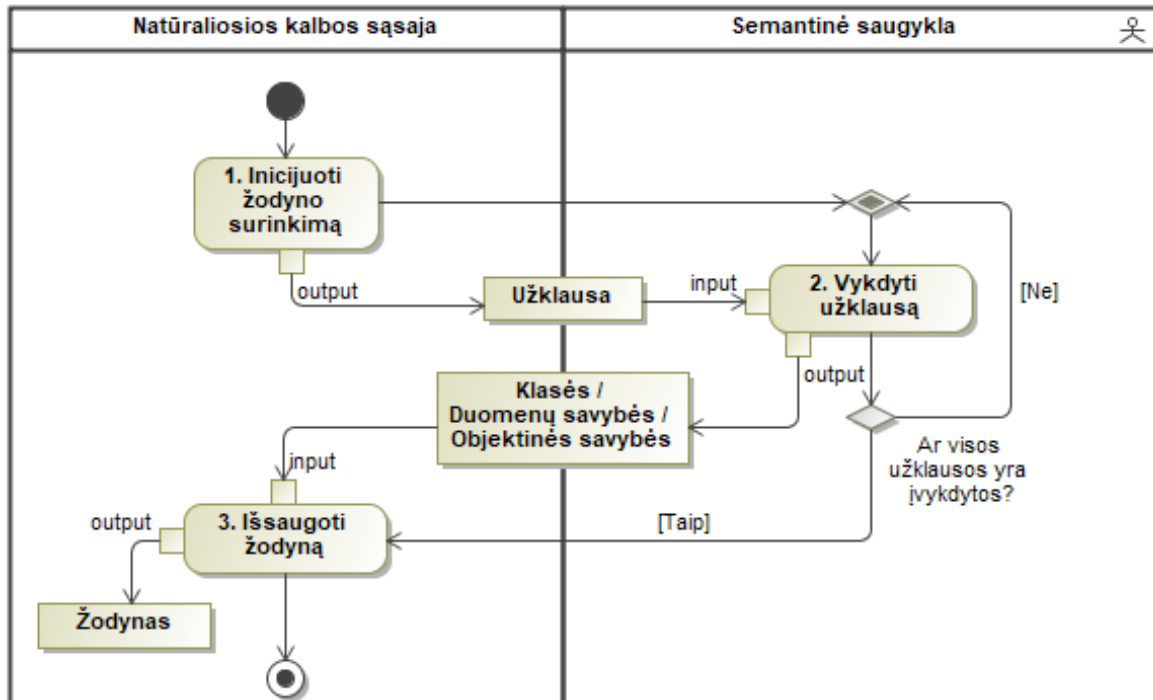
11 pav. Panaudojimo atvejo „Įvesti ir tikrinti klausimą“ veiklos diagrama

Klausimų analizavimas paremtas semantinės saugyklos žodynu, kuris gaunamas vykdant *SPARQL* užklausas (skirtas išrinkti klases, savybes ir sinonimus) ontologijoje. Šis PA įjungus programą vykdomas vieną kartą, prieš pradėdant vykdyti paiešką, t.y. prieš atsakant į pirmąjį klausimą. Žemiau pateikiama PA „Gauti žodyną iš ontologijos“ specifikacija (žr. 5 lentelė) ir grafiškai jį aprašanti veiklos diagrama (žr. 12 pav.).

5 lentelė. Panaudojimo atvejo „Gauti žodyną iš ontologijos“ specifikacija

PA4 „Gauti žodyną iš ontologijos“		
Aprašymas: Natūraliosios kalbos sąsaja vykdo <i>SPARQL</i> užklausas ontologijoje ir pagal gautus rezultatus sukuria žodyną.		
Prieš sąlyga:	Įjungus natūraliosios kalbos sąsają dar nebuvo vykdyta paieška ir žodynas negautas.	
Sužadinimo sąlyga:	Naudotojas turi būti įvedęs ir patvirtinęs klausimą.	
Aktorius:	Sąsajos naudotojas	
Susiję PA	Išplečiantys PA	Nėra

	Apimami PA	Nėra
	Specializuoti PA	Nėra
Po sąlyga:	Žodynas gautas ir sistema parengta analizuoti klausimą.	



12 pav. Panaudojimo atvejo „Gauti žodyną iš ontologijos“ veiklos diagrama

Natūraliosios kalbos sąsajoje pateikus klausimą naudotojas jį patvirtina ir inicijuoja paieškos vykdymą. Teisingai suformuluotas klausimas turi būti išanalizuojamas. Žemiau pateikiama PA „Analizuoti klausimą“ specifikacija (žr. 6 lentelė) ir grafiškai jį aprašanti veiklos diagrama (žr. 18 pav.).

6 lentelė. Panaudojimo atvejo „Analizuoti klausimą“ specifikacija

PA3 „Analizuoti klausimą“		
Aprašymas: Natūraliosios kalbos sąsaja tikrina klausimą.		
Prieš sąlyga:		Pateiktas klausimas turi būti teisingas.
Sužadinimo sąlyga:		Naudotojas įveda klausimą, jį patvirtina ir klausimas yra teisingas.
Aktorius:		Sąsajos naudotojas
Susiję PA	Išplečiantys PA	PA5 „Transformuoti klausimą į užklausą“
	Apimami PA	Nėra
	Specializuoti PA	Nėra
Po sąlyga:		Klausimas išanalizuotas.

Išanalizavus pateiktą klausimą ir nustatčius jo tipą, klausimas transformuojamas į užklausą, kurią įvykdžius naudotojui pateikiami gauti rezultatai. Žemiau pateikiama PA „Transformuoti klausimą į užklausą“ specifikacija (žr. 7 lentelė). Grafiškai jį aprašančios veiklos diagramos pateikiamos 2.4.2 skyriuje.

7 lentelė. Panaudojimo atvejo „Transformuoti klausimą į užklausą“ specifikacija

PA5 „Transformuoti klausimą į užklausą“		
Aprašymas: Natūraliosios kalbos sąsaja transformuoja klausimą į užklausą.		
Prieš sąlyga:		Nustatytas palaikomas klausimo tipas.
Sužadinimo sąlyga:		Natūraliosios kalbos sąsaja išanalizavo klausimą.
Aktorius:		Sąsajos naudotojas
Susiję PA	Išplečiantys PA	Nėra
	Apimami PA	Nėra
	Specializuoti PA	Nėra
Po sąlyga:		Klausimas transformuotas į užklausą.

2.1.2. Nefunkciniai reikalavimai

Nefunkciniai reikalavimai apibrėžia kuriamos natūraliosios kalbos sąsajos savybes, kuriomis turi pasižymėti kuriamas sprendimas:

1. klausimai turi būti pateikiami lietuvių kalba;

Testavimas – žmogus, gramatiškai ir sintaksiškai išmanantis lietuvių kalbą, patikrina ar naudotojo sąsaja tikrina pateikiamų klausimų taisyklumą.

2. grafinė naudotojo sąsaja turi būti sudaryta iš vieno paieškos lauko ir patvirtinimo mygtuko;

Testavimas – žmogus, atliekantis sąsajos testavimą, patikrina, ar naudotojo sąsajos paprasta ir lengvai suprantama.

3. klaidos pranešimai turi būti spausdinami raudonos spalvos šriftu;

Testavimas – žmogus, atliekantis sąsajos testavimą, patikrina, ar duomenų įvedimo formose įvedus neteisingus duomenis arba atliekant sistemos veikimui prieštaraujančius veiksmus atspausdintas klaidos pranešimas yra pateikiama raudonos spalvos šriftu.

4. paieška turi būti vykdomas naudojant *SPARQL* užklausų kalbą;

Testavimas – žmogus, atliekantis sąsajos testavimą, peržiūri programinį kodą ir patikrina, ar sukurtų užklausų šablonai yra aprašyti *SPARQL* kalba.

5. NKS neturi būti susieta su konkrečia dalykine sritimi, bet turi būti pritaikoma skirtingoms dalykinėms sritims.

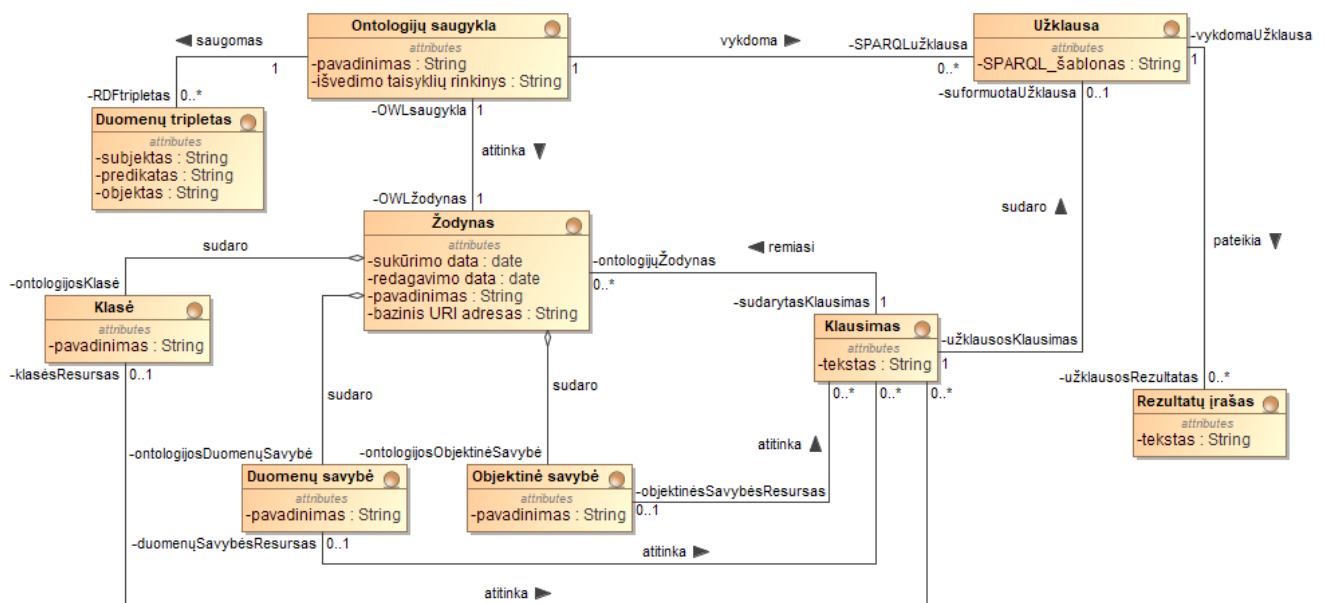
Testavimas – žmogus, atliekantis sąsajos testavimą, patikrina, ar yra galimybė pasirinkti skirtingų dalykinių sričių žodynus.

2.2. Natūraliosios kalbos sąsajos ontologijoms dalykinės srities modelis

Pagal aprašytą kuriamo inžinerinio sprendimo funkcinių ir nefunkcinių reikalavimų specifikaciją buvo identifikuojami pagrindiniai atliekamo tyrimo srities objektai ir jų tarpusavio ryšiai. Kuriamos NKS ontologijoms dalykinės srities modeliui aprašyti buvo analizuojami kiekvieno PA aprašymai, operacijų argumentai ir juose dalyvaujantys objektai. Nustačius pagrindines esybes, atributus ir jų tarpusavio sritis buvo atliktas esybių klasių ir egzempliorių diagramų modeliavimas (žr. 2.2.1 ir 2.2.2 skyrius).

2.2.1. Natūraliosios kalbos sąsajos ontologijoms dalykinės srities esybių klasių diagrama

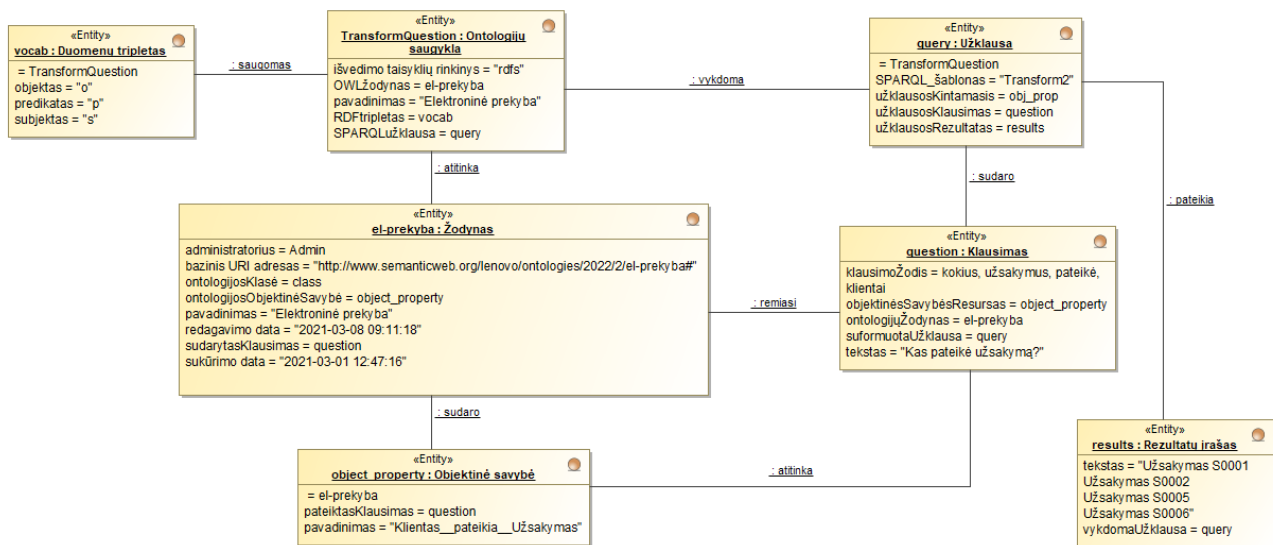
NKS esybių klasių modelis apibūdina pagrindines su kuriama sprendimu susijusias sąvokas, tokias kaip klausimas, užklausa, klasė, objektinė savybė, duomenų savybė, duomenų tripletas ir kt. Klausimas rašomas remiantis ontologijos schema, kuri sudaryta iš klasių, savybių ir naudojama kaip žodynas analizuojant klausimus. Pateiktas klausimas paremtas vienu iš jį atitinkančiais resursų (klasė arba savybės). Išanalizuotas klausimas transformuojamas į *SPARQL* užklausą, kurią įvykdžius pateikiami gauti rezultatai (žr. 13 pav.).



13 pav. Natūraliosios kalbos sąsajos ontologijoms sprendimo dalykinės srities esybių klasių modelis

2.2.2. Natūraliosios kalbos sąsajos ontologijoms egzempliorių diagrama

Remiantis sukurtu NKS esybių klasių modeliu buvo sukurta egzempliorių diagrama. Pavyzdinėje egzempliorių diagramoje apibrėžiamas naudotojo pateikto klausimo „Kokius užsakymus pateikė klientai?“ paieškos procese naudojami objektai (žr. 14 pav.).

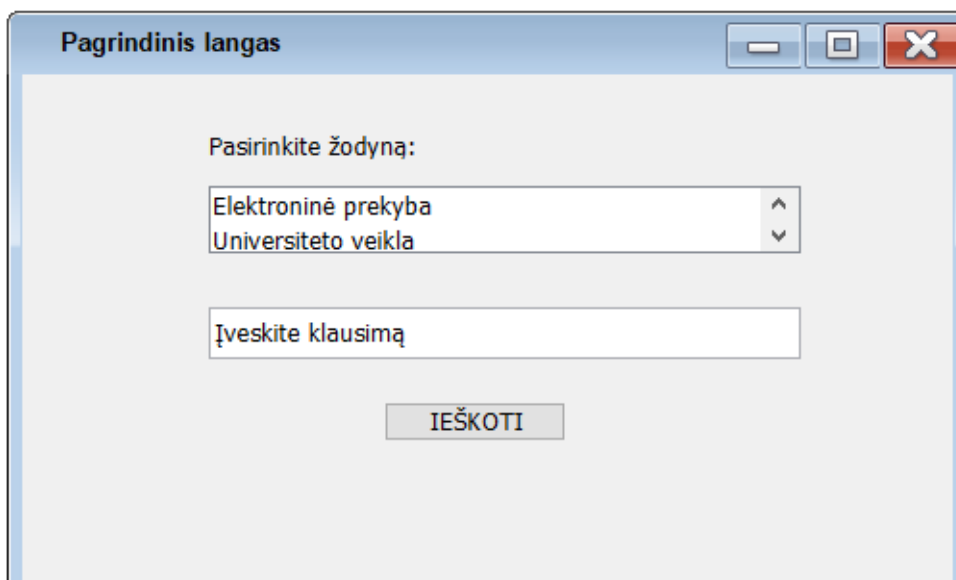


14 pav. Egzempliorių specifikacijos diagrama

Vykdam įvesto klausimo paiešką visų pirma klausimas yra išskaidomas į atskirus žodžius: „*kokius*“, „*užsakymus*“, „*pateikė*“, „*klientai*“. Kiekvienas klausimo žodis yra išnagrinėjamas morfologiškai, randama žodžio pagrindinė forma ir nustatoma kalbos dalis: „*koks*“ (įvardis), „*užsakymas*“ (daiktavardis), „*pateikti*“ (veiksmažodis), „*klientas*“ (daiktavardis). Išanalizuoti klausimo žodžiai susiejami su žodyno sąvokomis: klasėmis „*Užsakymas*“ ir „*Klientas*“ bei objektine savybe „*Klientas_pateikė_Užsakymas*“. Nustačius klausimo tipą, jis transformuojamas į užklausą, kurią įvykdžius pateikiami gauti rezultatai, t.y. randami užsakymai, kuriuos pateikė klientai.

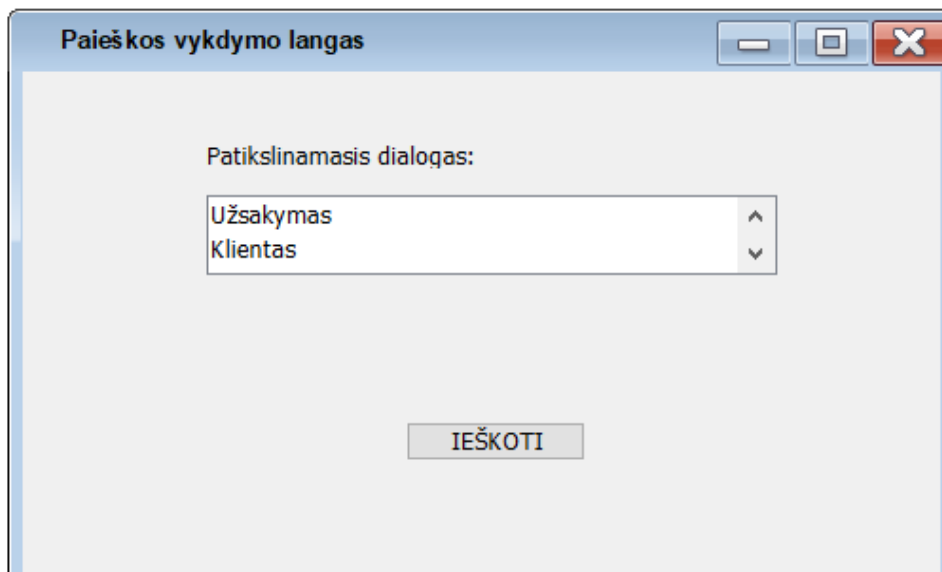
2.3. Natūraliosios kalbos sąsajos ontologijoms naudotojų sąsajos modelis

Šiame skyrelyje pateikiami pagrindiniai natūraliosios kalbos sąsajos grafiniai naudotojo langų prototipai. Pagrindiniame sąsajos lange pateikiamas žodyno pasirinkimo laukas, skirtas pasirinkti dalykinę sritį, kurioje turi būti vykdoma paieška. Žemiau pateikiamas klausimo įvedimo laukas ir patvirtinimo mygtukas, skirtas vykdyti paiešką (žr. 15 pav.).



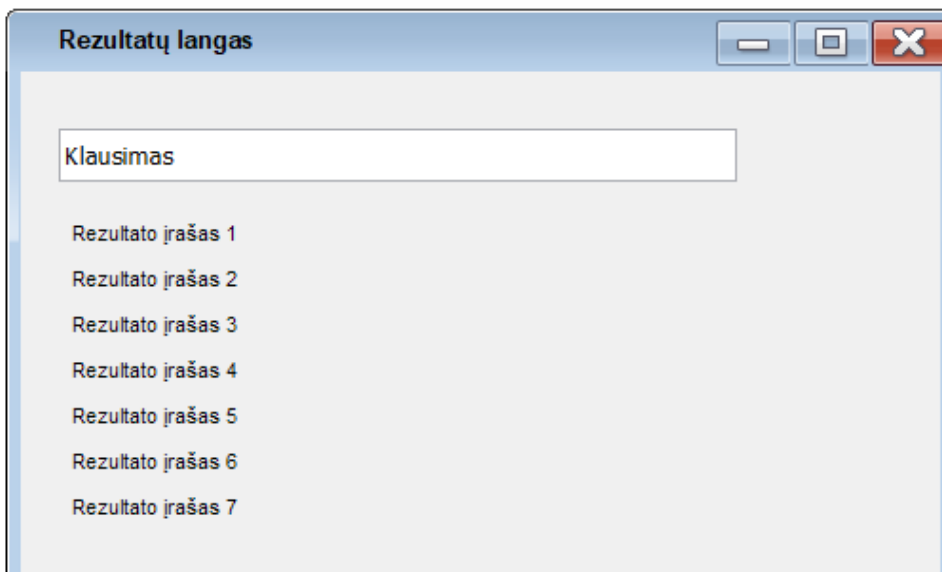
15 pav. Pagrindinis sąsajos langas

Tokiais atvejais, kai klausimas interpretuojamas dviprasmiškai, NKS pateikia naudotojui patikslinamojo dialogo langą, kuriame naudotojas turi pasirinkti vieną iš pateiktų pasirinkimo variantų ir patvirtinti paiešką mygtuko paspaudimu (žr. 16 pav.).



16 pav. Patikslinamojo dialogo langas

Įvykdžius klausimo paiešką, NKS pateikia rezultatų langą, kuriame pateikiamas gautų rezultatų sąrašas (žr. 17 pav.).



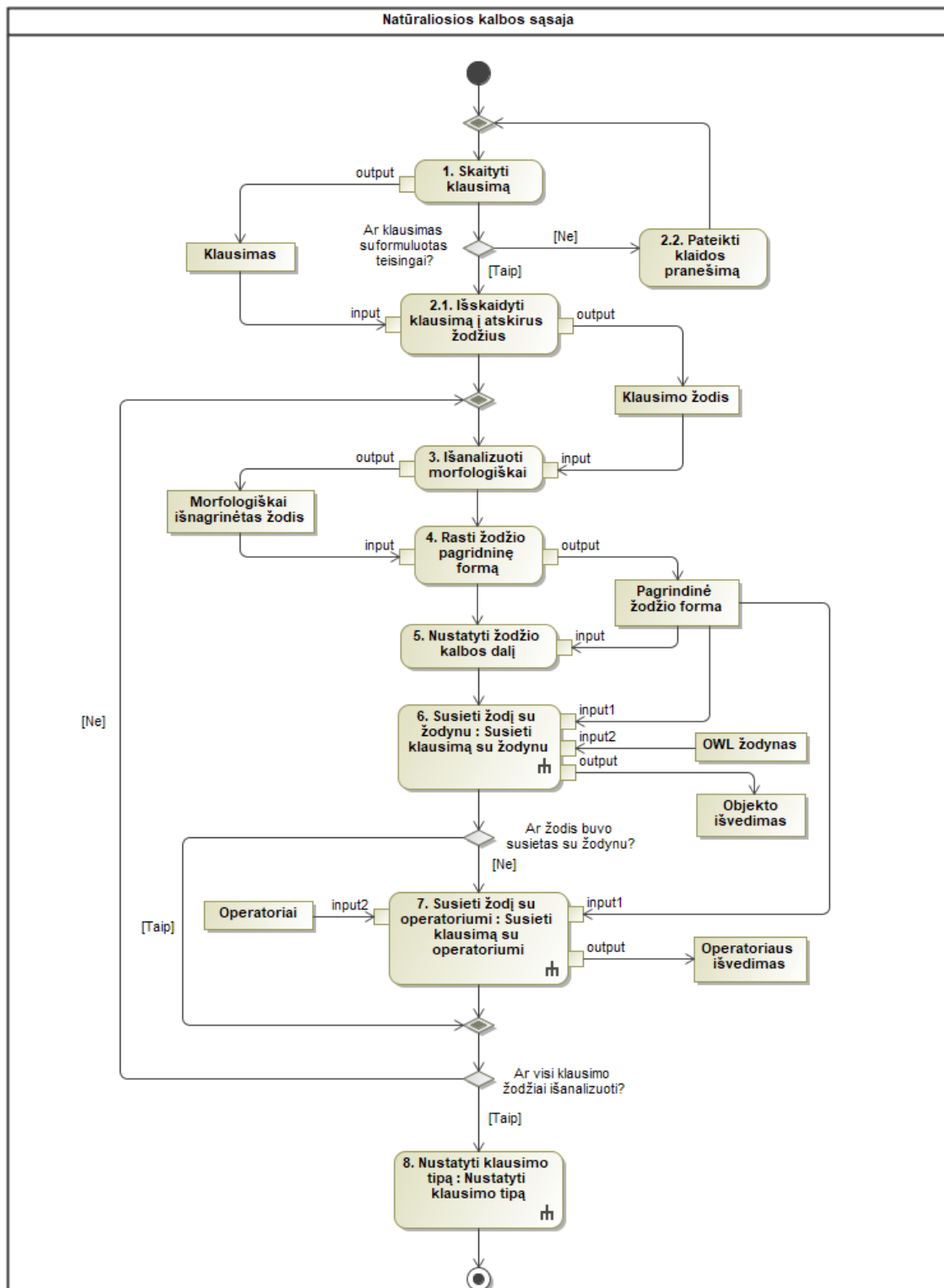
17 pav. Rezultatų langas

2.4. Formalus natūraliosios kalbos sąsajos ontologijoms aprašas

Kuriamas metodas, leidžiantis naudotojui vykdyti paiešką pateikiant klausimą lietuvių kalba ir transformuoti jį į užklausą ontologijoje, yra išskaidytas į du pagrindinius etapus: analizavimas ir transformavimas į užklausą. Metodas realizuojamas sukurtu algoritmu, kuris išanalizuoja natūraliąja kalba pateiktą klausimą, remiantis iš anksto aprašytomis taisyklėmis nustato klausimo tipą ir transformuoja jį į *SPARQL* užklausą. Šie etapai detalčiau aprašomi sekančiuose skyriuose (žr. 2.4.1 ir 2.4.2 skyrius). Taip pat 2.4.3 ir 2.4.4 skyriuose aprašomos ontologijų kūrimo ir klausimų sudarymo taisyklės.

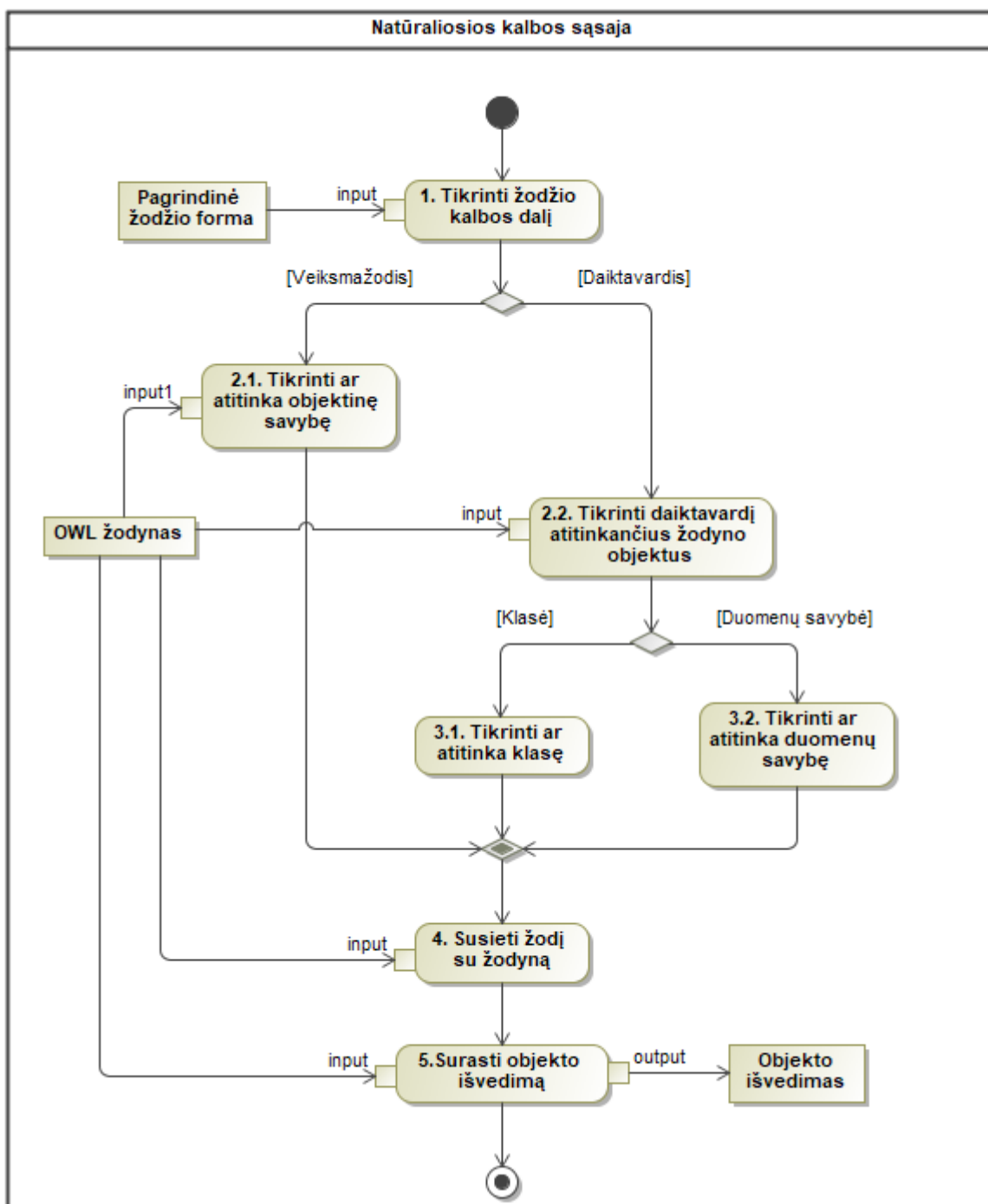
2.4.1. Klausimo analizavimas

Metodas realizuojamas sukurtu algoritmu, kuris išanalizuoja natūraliąja kalba pateiktą klausimą. Pirmiausia tikrinimo metu įvestas naudotojo klausimas yra išskaidomas į atskirus žodžius. Kiekvienas žodis yra išnagrinėjamas morfologiškai naudojant morfoliginį analizatorių. Tokiu būdu kiekvienam žodžiui yra surandama pagrindinė jo forma ir nustatoma kalbos dalis. Remiantis nustatyta kalbos dalimi, klausimo žodžiai yra susiejami su žodyno, išgauto iš OWL schemos sąvokomis arba aprašytais operatoriais (aritmetiniai ir agregavimo). Žemiau pateikiama PA „Analizuoti klausimą“ veiklos diagrama (žr. 18 pav.).



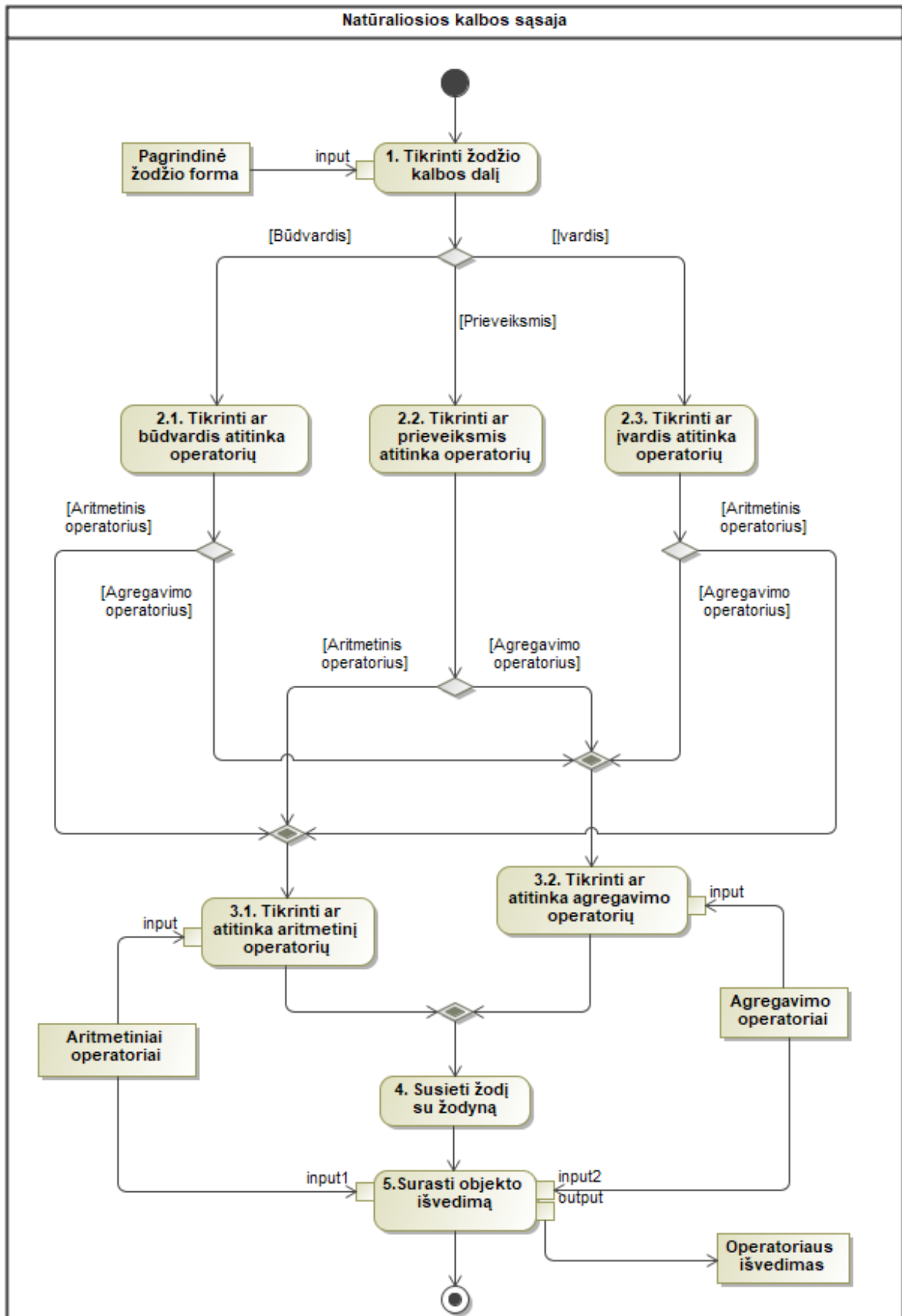
18 pav. Panaudojimo atvejo „Analizuoti paiešką“ veiklos diagrama

Vienas pagrindinių žingsnių analizuojant pateikto klausimo žodžius yra klausimo susiejimas su *OWL* žodyno sąvokomis. Žodyne ieškomi rastų klausimo žodžių atitikmenys remiantis nustatyta kalbos dalimi. Tokiu būdu žodis susiejamas su vienu iš ontologijos elementų – klase, duomenų ar objektine savybe. Siekiant, kad susiejimas būtų sėkmingesnis, randama ir pagrindinė žodžio forma (lema). Žemiau pateikiama proceso „Susieti klausimą su žodynu“ veiklos diagrama (žr. 19 pav.).



19 pav. Veiklos „Susieti klausimą su žodynu“ veiklos diagrama

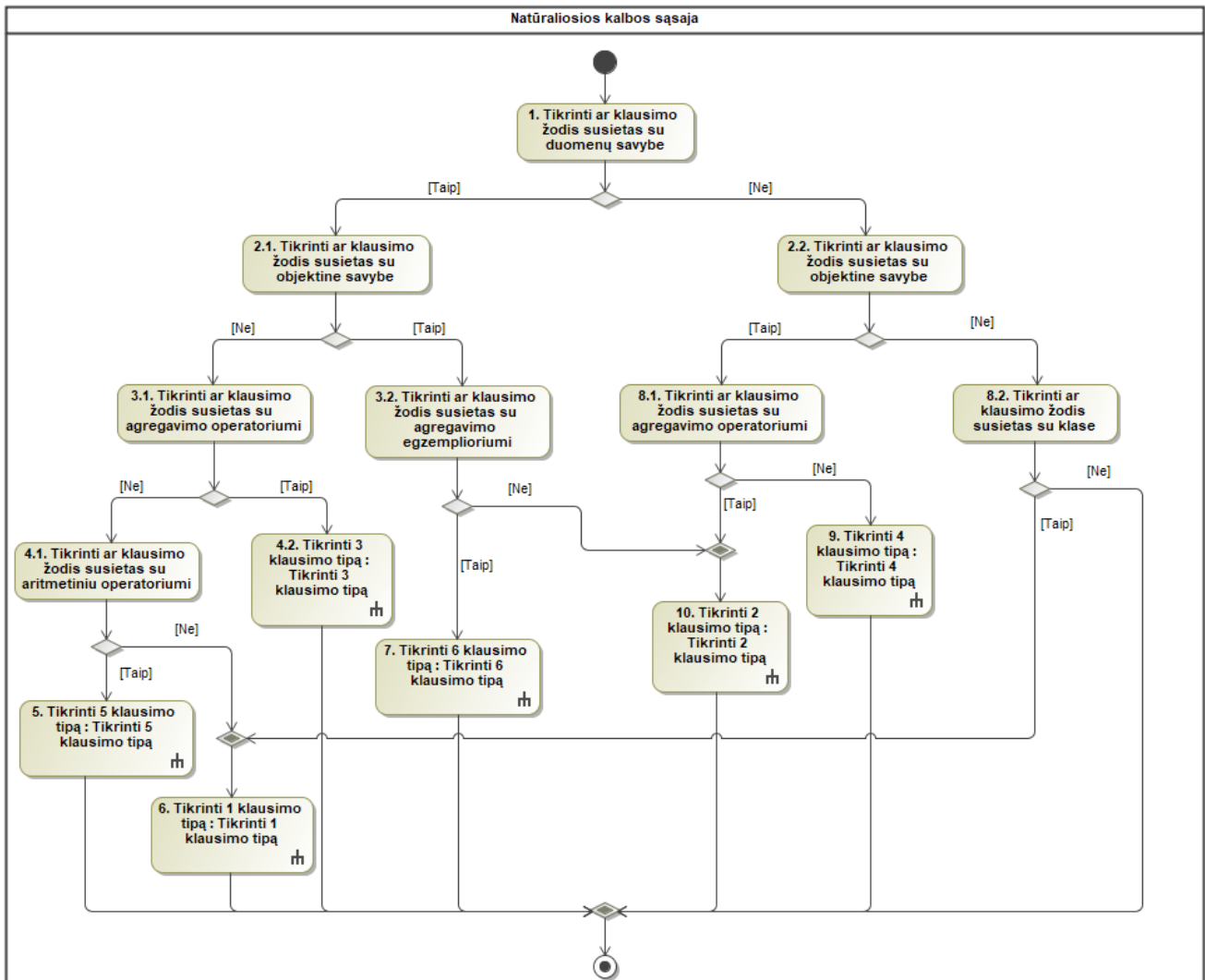
Remiantis nustatyta išanalizuotų klausimo žodžių kalbos dalimi yra tikrinami ir aprašytų operatorių atitikmenys. Jeigu atitikmuo randamas, žodis susiejamas su operatoriumi (pvz. agregavimo operatoriumi – *COUNT*). Žemiau pateikiama proceso „Susieti klausimą su operatoriumi“ veiklos diagrama (žr. 20 pav.).



20 pav. Veiklos „Susieti klausimą su operatoriumi“ veiklos diagrama

2.4.2. Klausimo transformavimas

Išanalizavus pateikto klausimo žodžius ir susiejus juos su *OWL* žodynu, turi būti nustatoma, kokio klausimo tipo transformacija į užklausą turi būti vykdoma. Klausimo transformavimo taisyklės yra sukurtos remiantis šešiais klausimų tipais. Kiekvienam klausimų tipui jo nustatymo taisyklės ir transformavimo į *SPARQL* užklausą algoritmai. Žemiau pateikiamas bendras taisyklių rinkinys, kuriuo remiantis yra nustatomas klausimo tipas (žr. 21 pav.).

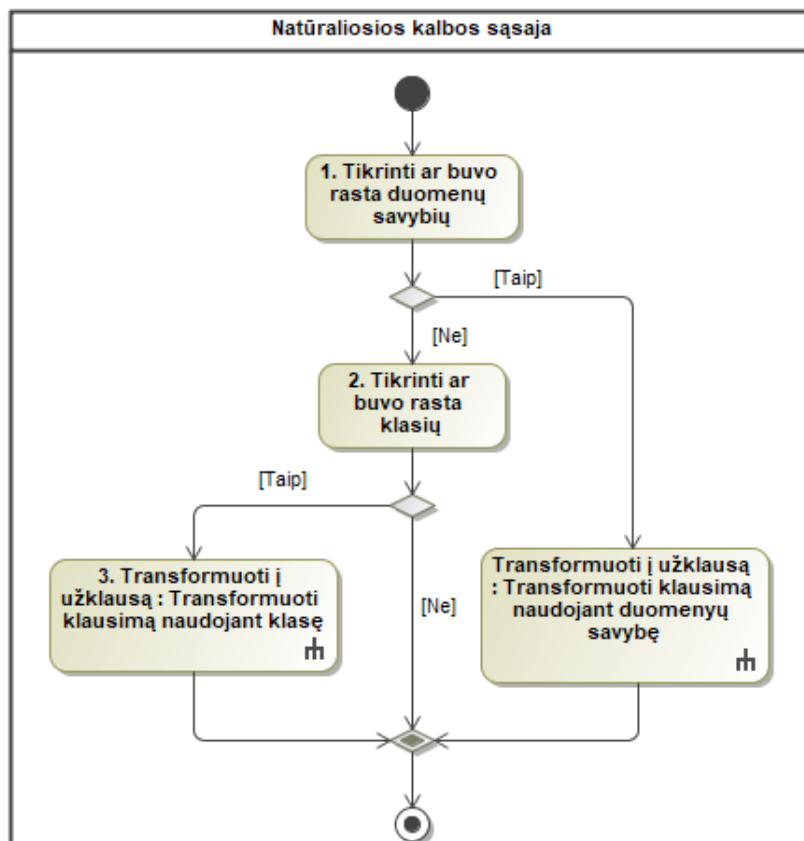


21 pav. Taisyklių rinkinys klausimo tipo nustatymui

Diagramoje aprašomos sąlygos, kada yra tikrinama, ar klausimas atitinka tam tikrą tipą. Toliau skyrelyje pateikiamos diagramos, kuriose matyti klausimų tipų nustatymo ir transformavimo algoritmai. Transformavimas vykdomas naudojant *SPARQL* užklausos šablonus, aprašytus konkrečiam klausimo tipui.

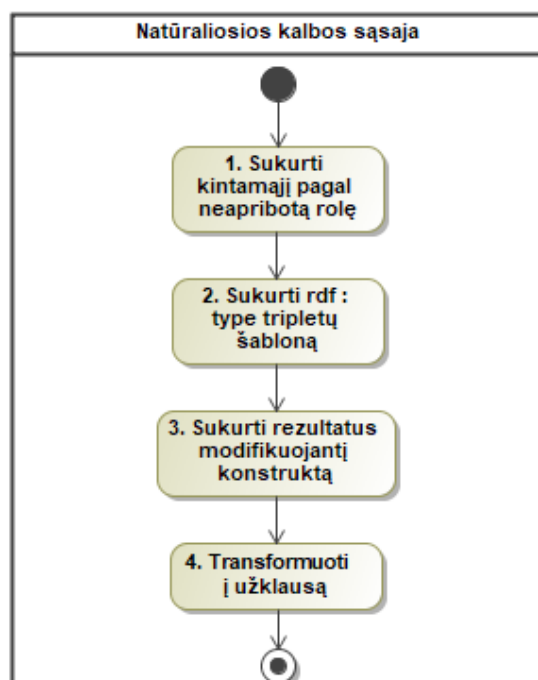
1. Klausimas remiantis viena role be apribojimų (1 klausimo tipas)

Šiame skyrelyje aprašomas paprastų klausimų, kurių atsakymams naudojama viena rolės, susieta su konkrečiu ontologijos resursu, t.y. klase ar duomenų savybe, transformavimas. Žemiau pateikiamas taisyklių rinkinys, kuriuo remiantis nustatomas pirmasis klausimų tipas (žr. 22 pav.).



22 pav. Taisyklių rinkinys klausimo, kuris remiasi viena role be apribojimų, nustatymui

Klausimą transformuojant į užklausą, naudojamas šiam klausimo tipui aprašytas *SPARQL* užklauso šablonas. Klausimams, kurie remiasi viena role be apribojimų, gali būti taikomi dviejų tipų šablonai: remiantis klase arba duomenų savybe. Pavyzdinis klausimas, kai remiamasi klase galėtų būti „*Kokių yra produktų?*“. Tokio klausimo atsakymui naudojama klasė „*Produktas*“, kuriai nėra taikomi apribojimai. Žemiau pateikiamas transformavimo algoritmas, kai klausimas paremtas viena klase (žr. 23 pav.), ir sukurta *SPARQL* užklausa (žr. 24 pav.).



23 pav. Klausimo, kuris remiasi viena klase be apribojimų, transformavimo algoritmas

```

# Užklauso kintamasis
SELECT ?s
WHERE {
# Užklauso rdf:type tripletų šablonas
  ?s rdf:type vocab:Produktas.
}
GROUP BY ?s

```

24 pav. SPARQL užklausa klausimo, kuris remiasi viena klase be apribojimų, transformavimui

Pavyzdinis klausimas, kai remiamasi duomenų savybe galėtų būti „Kokių yra produktų rūšių?“. Tokio klausimo atsakymui naudojama duomenų savybė „Produktas__turi_rūši“, kuriai nėra taikomi apribojimai. Žemiau pateikiamas transformavimo algoritmas, kai klausimas paremtas viena duomenų savybe (žr. 25 pav.) ir sukurta SPARQL užklausa (žr. 26 pav.).



25 pav. Klausimo, kuris remiasi viena duomenų savybe be apribojimų, transformavimo algoritmas

```

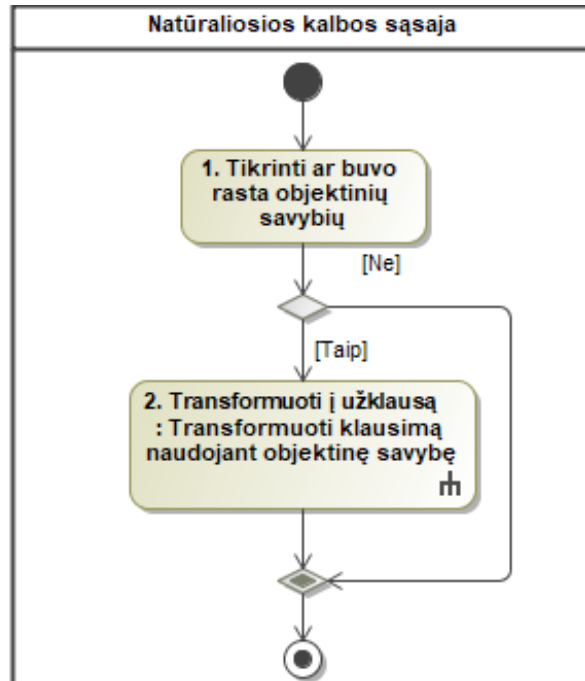
# Užklauso kintamasis
SELECT ?o
WHERE {
# Užklauso tripletų šablonas
  ?s vocab:Produktas__turi_rūši ?o.
}
GROUP BY ?o

```

26 pav. SPARQL užklausa klausimo, kuris remiasi viena duomenų savybe be apribojimų, transformavimui

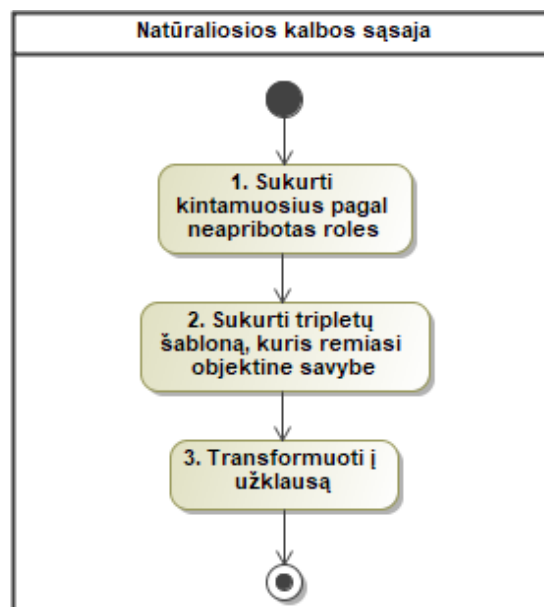
2. Klausimas remiantis dviem rolėm be apribojimų (2 klausimo tipas)

Šiame skyrelyje aprašomas paprastų klausimų, kurių atsakymams naudojamos dvi rolės, susietos su objektinės savybės *domain* ir *range* klasėmis, transformavimas. Rolėms nėra taikomi jokie papildomi apribojimai. Žemiau pateikiamas taisyklių rinkinys, kuriuo remiantis nustatomas antrasis klausimų tipas (žr. 27 pav.).



27 pav. Taisyklių rinkinys klausimo, kuris remiasi dviem rolėm be apribojimų, nustatymui

Klausimą transformuojant į užklausą naudojamas šiam klausimo tipui aprašytas *SPARQL* užklausos šablonas. Klausimams, kurie remiasi dviem rolėm be apribojimų, taikomas šablonas naudojant objektinę savybę. Pavyzdžiui, klausimo „Kokie užsakymai gauti iš klientų?“ atsakymui remiamasi objektine savybe „Užsakymas_gautas_iš_Klientas“, kuriai nėra taikomi apribojimai. Žemiau pateikiamas šio klausimo tipo transformavimo algoritmas (žr. 28 pav.) ir sukurta *SPARQL* užklausa (žr. 29 pav.).



28 pav. Klausimo, kuris remiasi objektine savybe be apribojimų, transformavimo algoritmas

```

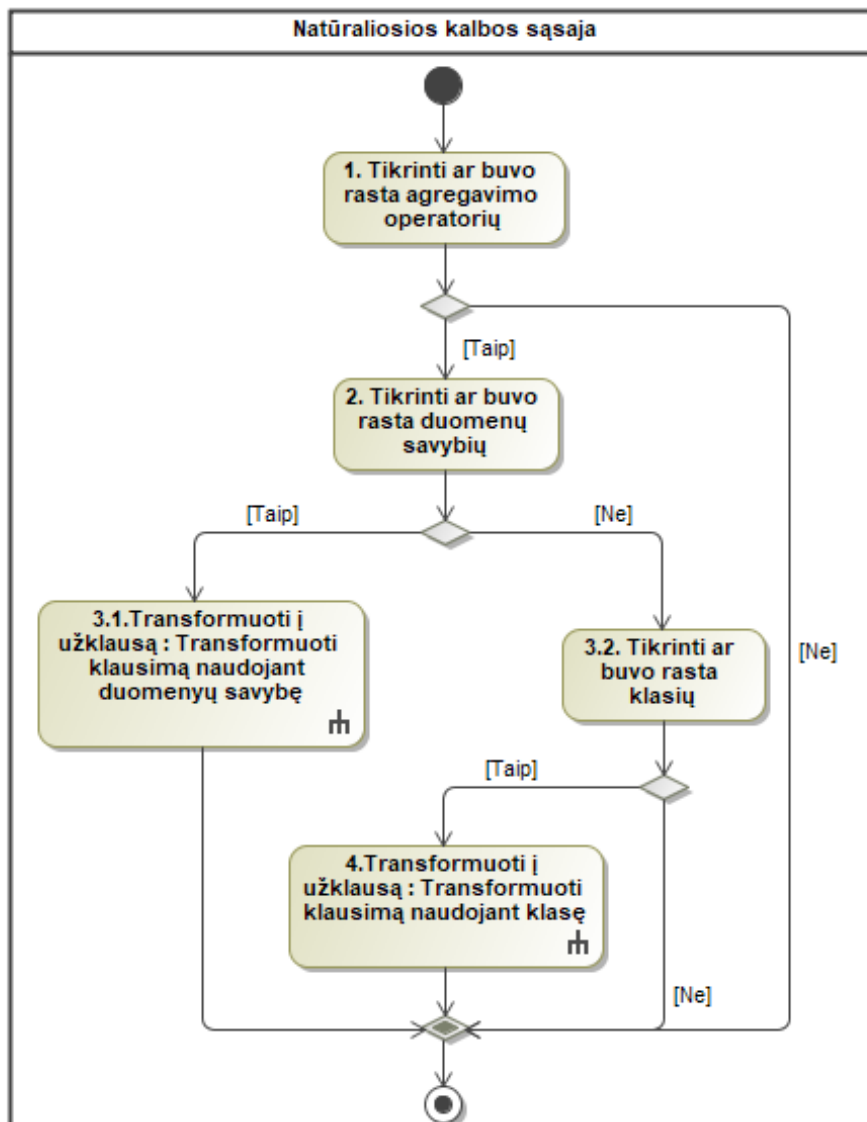
# Užklauso kintamasis
SELECT ?s ?o
WHERE {
# Užklauso tripletų šablonas
?s vocab:Klientas__pateikia__Užsakymas ?o.
}

```

29 pav. SPARQL užklausa klausimo, kuris remiasi dviem rolėm be apribojimų, transformavimui

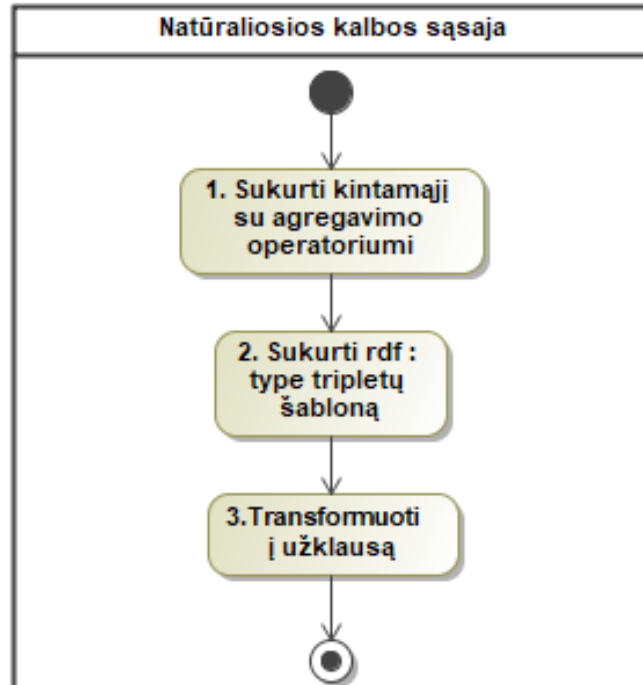
3. Klausimas naudojant agregavimo operatorių remiantis viena role (3 klausimo tipas)

Šiame skyrelyje aprašomas klausimų, kurių atsakymams naudojami agregavimo operatoriai (*SUM*, *AVG*, *COUNT*, *MIN*, *MAX*) remiantis vienu ontologijos resursu, t.y. klase ar duomenų savybe, transformavimas. Žemiau pateikiamas taisyklių rinkinys, kuriuo remiantis nustatomas trečiasis klausimo tipas (žr. 30 pav.).



30 pav. Taisyklių rinkinys klausimo, kuris naudoja agregavimo operatorių ir remiasi viena role, nustatymui

Klausimą transformuojant į užklausą, naudojamas šiam klausimo tipui aprašytas *SPARQL* užklausos šablonas. Klausimams, kurie naudoja agregavimo operatorių ir remiasi viena role, gali būti taikomi dviejų tipų šablonai: remiantis klase arba duomenų savybe. Pavyzdinis klausimas, kai remiamasi klase galėtų būti „*Kiek yra produktų?*“. Tokio klausimo atsakymui naudojamas agregavimo operatorius *COUNT* ir klasė „*Produktas*“. Žemiau pateikiamas transformavimo algoritmas, paremtas viena klase (žr. 31 pav.) ir sukurta *SPARQL* užklausa (žr. 32 pav.).



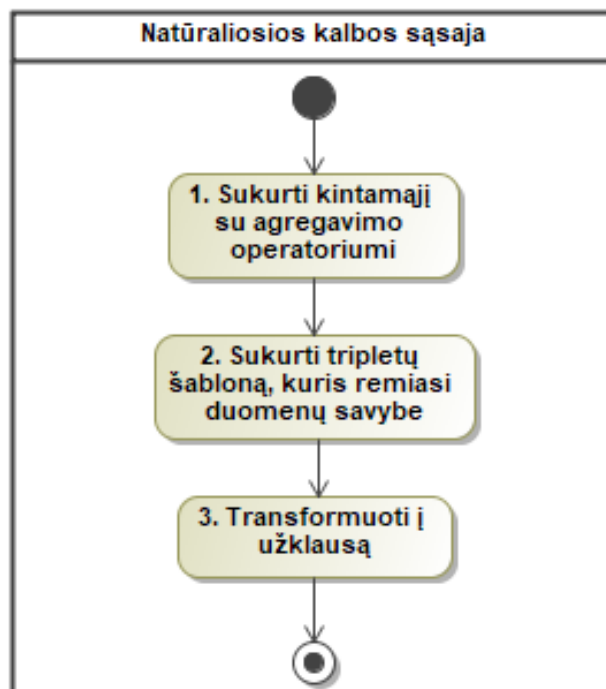
31 pav. Klausimo, kuris naudoja agregavimo operatorių remiantis viena klase, transformavimo algoritmas

```

# Užklausos kintamasis su agregavimo operatoriumi
SELECT (COUNT (?s) as ?o)
WHERE {
# Užklausos rdf:type tripletų šablonas
  ?s rdf:type vocab:Produktas.
}
  
```

32 pav. *SPARQL* užklausa klausimo, kuris naudoja agregavimo operatorių remiantis viena klase, transformavimui

Pavyzdinis klausimas, kai remiamasi duomenų savybe galėtų būti „*Kokio produkto kaina yra didžiausia?*“. Tokio klausimo atsakymui naudojamas agregavimo operatorius *MAX* ir duomenų savybė „*Produktas__ turi __ kainą*“. Žemiau pateikiamas transformavimo algoritmas, paremtas viena duomenų savybe (žr. 33 pav.) ir sukurta *SPARQL* užklausa (žr. 34 pav.).



33 pav. Klausimo, kuris naudoja agregavimo operatorių remiantis viena duomenų savybe, transformavimo algoritmas

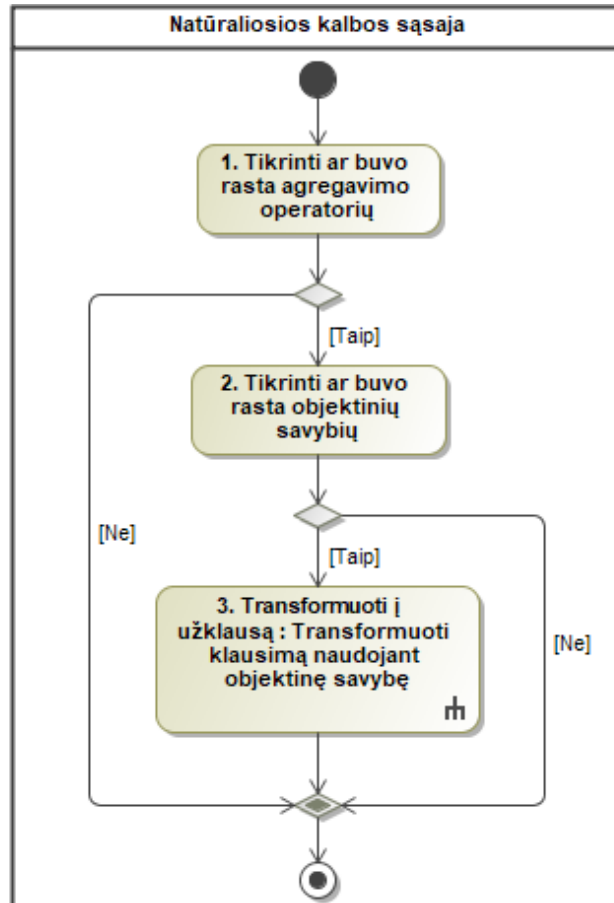
```

# Užklausos kintamasis su agregavimo operatoriumi
SELECT (MAX (?s) as ?o)
WHERE {
# Užklausos tripletų šablonas
  ?s vocab:Produktas__turi_kainą ?o.
}
  
```

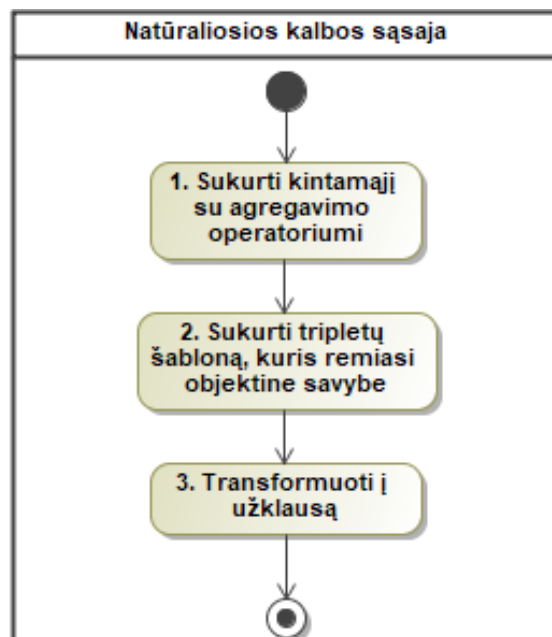
34 pav. SPARQL užklausa klausimo, kuris naudoja agregavimo operatorių remiantis viena klase, transformavimui

4. Klausimas naudojant agregavimo operatorių remiantis dviem rolėm (4 klausimo tipas)

Šiame skyrelyje aprašomas klausimų, kurių atsakymams naudojamos dvi rolės, susietos su objektinės savybės *domain* ir *range* klasėmis ir vienai iš rolių naudojamas agregavimo operatorius (*SUM*, *AVG*, *COUNT*, *MIN*, *MAX*). Žemiau pateikiamas ketvirtojo klausimų tipo nustatymo algoritmas (žr. 35 pav.).



35 pav. Taisyklių rinkinys klausimo, kuris naudoja agregavimo operatorių ir remiasi dviem rolėm, nustatymui Klausimą transformuojant į užklausą, naudojamas šiam klausimo tipui aprašytas *SPARQL* užklauso šablonas, kuriame naudojama objektinė savybė. Pavyzdžiui, klausimo „*Kiek užsakymų pateikė klientai?*“ atsakymui gauti naudojamas agregavimo operatorius *COUNT* ir objektinė savybė „*Klietas_pateikia_Užsakymas*“. Žemiau pateikiamas transformavimo algoritmas, paremtas objektine savybe (žr. 36 pav.) ir sukurta *SPARQL* užklausa (žr. 37 pav.).



36 pav. Klausimo, kuris naudoja agregavimo operatorių remiantis objektine savybe, transformavimo algoritmas

```

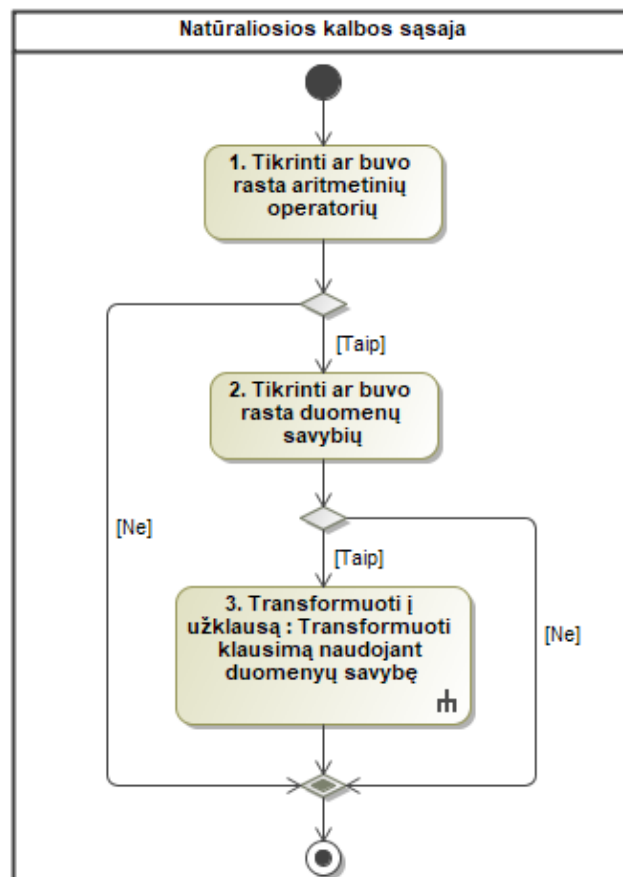
# Užklausa kintamasis su agregavimo operatoriumi
SELECT (COUNT (?o) as ?p)
WHERE {
# Užklausa tripletų šablonas
  ?s vocab:Klientas__pateikia__Užsakymas ?o.
}

```

37 pav. SPARQL užklausa klausimo, kuris naudoja agregavimo operatorių remiantis objektine savybe, transformavimui

5. Klausimas naudojant aritmetinį operatorių (5 klausimo tipas)

Šiame skyrelyje aprašomas klausimų, kurių atsakymams gauti naudojamas aritmetinis palyginimo operatorius (*FILTER*), transformavimas. Žemiau pateikiamas taisyklių rinkinys, kuriuo remiantis nustatomas penktasis klausimų tipas (žr. 38 pav.).



38 pav. Taisyklių rinkinys klausimo, kuris naudoja aritmetinį operatorių, nustatymui

Klausimą transformuojant į užklausą, naudojamas šiam klausimo tipui aprašytas SPARQL užklauso šablonas, turintis operatorių *FILTER*, kuris pritaikomas duomenų savybės reikšmei. Pavyzdžiui, klausimo „Kokių produktų kaina didesnė nei 50 eurų?“ atsakymui gauti naudojamas aritmetinis operatorius *FILTER*, turintis palyginimo apribojimą su pateikta reikšme „50“, remiantis duomenų savybe „Produktas__turi_kainą“. Žemiau pateikiamas transformavimo algoritmas (žr. 39 pav.) ir sukurta SPARQL užklausa (žr. 40 pav.).



39 pav. Klausimo, kuris naudoja aritmetinį operatorių, transformavimo algoritmas

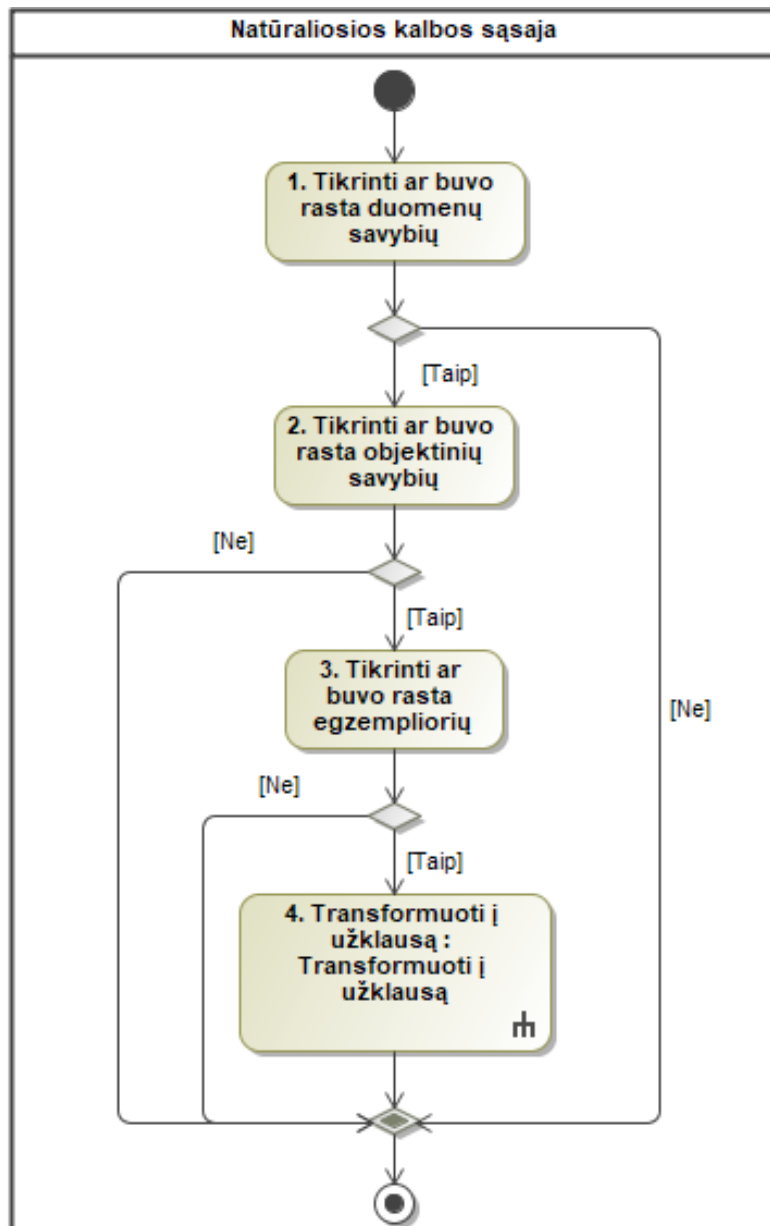
```

# Užklausos kintamieji
SELECT ?o
WHERE {
# Užklausos tripletų šablonas, kuris remiasi objektine
savybe
    ?s vocab:Produktas__turi_kaina ?o.
# Užklausos aritmetinis palyginimo operatorius rolei
    FILTER(?o > 50)
}
# Užklausos rezultatų modifikatorius
ORDER BY DESC (?o)
  
```

40 pav. SPARQL užklausa klausimo, kuris naudoja aritmetinį operatorių, transformavimui

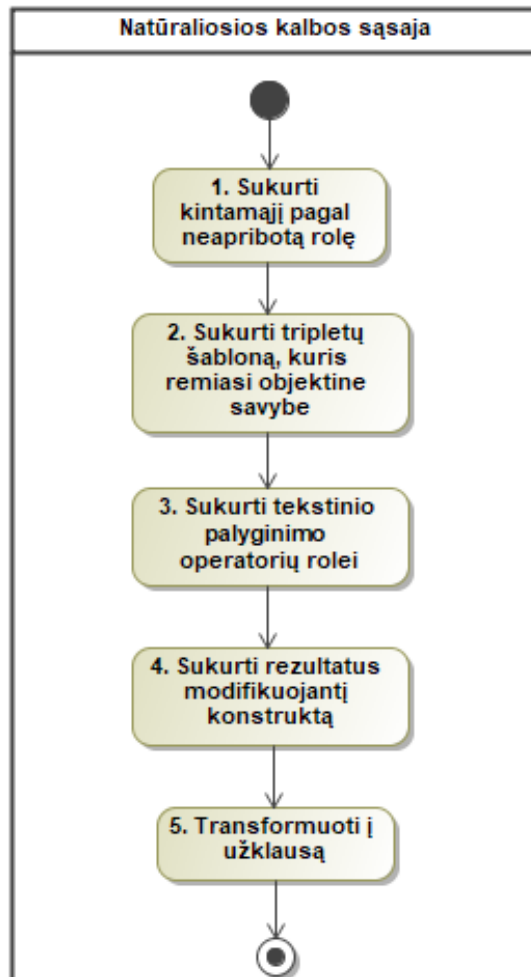
6. Klausimas, kurio rolė apribota konkrečiu egzemplioriumi (6 klausimo tipas)

Šiame skyrelyje aprašomas klausimų, kurių atsakymams naudojamas konkretus egzempliorius, transformavimas. Žemiau pateikiamas taisyklių rinkinys, kuriuo remiantis nustatomas šeštasis klausimų tipas (žr. 41 pav.).



41 pav. Taisyklių rinkinys klausimo, kuris naudoja konkretų egzempliorių, nustatymui

Klausimą transformuojant į užklausą, naudojamas šiam klausimo tipui aprašytas *SPARQL* užklauso šablonas, turintis objektinę ir duomenų savybę bei egzemplioriaus tekstinę išraišką (pvz.: vardą). Pavyzdžiui, klausimo „Kokius užsakymus pateikė klientas vardu Jonas?“ atsakymui gauti naudojama objektinė savybė „Klientas_pateikia_Užsakymas“, duomenų savybė „Klientas_turi_vardą“ ir konkretus egzempliorius „Jonas“. Žemiau pateikiamas transformavimo algoritmas (žr. 42 pav.) ir sukurta *SPARQL* užklausa (žr. 43 pav.).



42 pav. Klausimo, kurio rolę apribota konkrečiu egzemplioriumi, transformavimo algoritmas

```

# Užklauso kintamasis
SELECT ?s
WHERE {
# Užklauso tripletų šablonas, kuris remiasi
objektine savybe
    ?s vocab:Klientas__pateikia__Užsakymas ?o.
# Užklauso tekstinio palyginimo operatorius
rolei
    ?o vocab:Klientas__turi_varda `Jonas`.
}
# Užklauso rezultatų modifikatorius
GROUP BY ?s
  
```

43 pav. SPARQL užklausa klausimo, kurio rolę apribota konkrečiu egzemplioriumi, transformavimui

2.4.3. Ontologijos kūrimo taisyklės

Sukurtam metodui ontologijos turi būti kuriamos laikantis tam tikrų taisyklių:

1. ontologija turi būti parengta lietuvių kalba;
2. ontologijos resursų pavadinimai negali kartotis;
3. ontologijai kuriamų klasių pavadinimai turi būti apibrėžti vienu arba dviem žodžiais, atskiriant juos apatiniu brūkšniu (pvz., „*Studentas*“, „*Studijų programa*“);
4. ontologijai kuriamų objektinių savybių pavadinimai turi būti veiksmažodinė frazė (pvz., „*pateikia*“, „*suformuoja*“);
5. kiekvienai objektinei savybei turi būti nustatytos domeno (angl. *domain*) ir srities (angl. *range*) reikšmės (pvz., „*Klientas __pateikia __Užsakymas*“);
6. ontologijai kuriamų duomenų savybių pavadinimai turi būti apibrėžti vienu arba dviem žodžiais pridėdant žodį „*turi*“ ir atskiriant juos apatiniu brūkšniu (pvz., „*turi_vardą*“, „*turi_asmens_kodas*“);
7. kiekvienai duomenų savybei turi būti nustatytos domeno (angl. *domain*) reikšmės (pvz., „*Klientas __turi_vardą*“);
8. sinonimai aprašomi naudojant klasių ar savybių ekvivalentiškumo aksiomas.

2.4.4. Klausimų sudarymo taisyklės

Kiekvienoje dalykinėje srityje klausimų sudarymas natūraliąja kalba turi būti atliekamas remiantis šiomis taisyklėmis:

1. klausimai turi būti formuluojami lietuvių kalba;
2. klausimuose negali būti vartojami žodžių trumpiniai (pvz. vietoje „*cm*“ turi būti vartojama „*centimetrai*“);
3. skaičiai turi būti nurodomi skaitine forma (pvz. *100*);
4. data turi būti nurodoma remiantis tokiu formatu „*YYYY-MM-DD*“ (pvz. „*2022-02-22*“);
5. klausimas negali prasidėti egzemplioriumi (pvz. vietoje „*Jono pateikti užsakymai?*“ galėtų būti vartojama „*Kokius užsakymus pateikė Jonas?*“);
6. klausime nurodomi egzemplioriai turi būti pradedami iš didžiosios raidės (pvz. „*Kokių užsakymų būseną yra Atlikta?*“).

2.5. Reikalavimų apibendrinimas

Atlikus būsimo NKS ontologijoms prototipo reikalavimų analizę buvo nustatyti pagrindiniai kuriamo inžinerinio sprendimo funkciniai ir nefunkciniai reikalavimai. Kuriamos sąsajos naudotojas galės įvesti klausimus ontologijai natūralia lietuvių kalba ir gauti rezultatus. Kuriamą naudotojo sąsajos funkcionalumui aprašyti buvo įtraukta ir semantinė saugykla, kuri atsakinga už ontologijų valdymą ir saugojimą. Pagal aprašytą kuriamo inžinerinio sprendimo funkcinių ir nefunkcinių reikalavimų

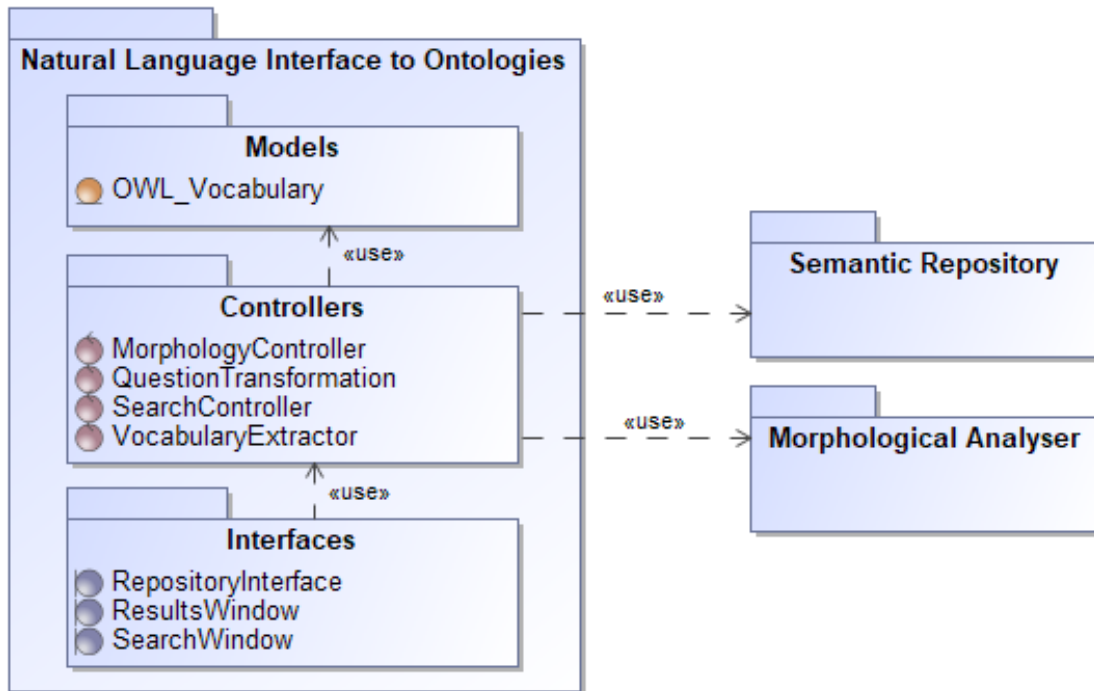
specifikaciją buvo identifikuoti pagrindiniai atliekamo tyrimo srities objektai ir jų tarpusavio ryšiai bei sumodeliuota esybių klasių diagrama.

Kuriamam metodui, skirtam vykdyti paiešką pateikiant klausimą lietuvių kalba ir transformuoti jį į užklausą ontologijoje, buvo nustatyti du pagrindiniai klausimo apdorojimo etapai: analizavimas ir transformavimas. Metodas realizuojamas sukurtu algoritmu, kuris išanalizuoja pateiktą klausimą, nustato klausimo tipą ir transformuoja jį į užklausą. Klausimo transformavimo taisyklės yra grindžiamos šešiais klausimų tipais: (1) klausimas remiantis viena role be apribojimų; (2) klausimas remiantis dviem rolėmis be apribojimų; (3) klausimas naudojant agregavimo operatorių remiantis viena role; (4) klausimas naudojant agregavimo operatorių remiantis dviem rolėmis; (5) klausimas naudojant aritmetinį operatorių ir (6) klausimas, kurio rolė apribota konkrečiu egzemplioriumi. Nustačius klausimo tipą, jis transformuojamas į užklausą, kuri sudaroma remiantis *SPARQL* užklausos šablonu, aprašytu konkrečiam klausimo tipui.

3. Natūraliosios kalbos sąsajos ontologijoms eksperimentinės realizacijos projektas

3.1. Natūraliosios kalbos sąsajos ontologijoms loginė architektūra

Natūraliosios kalbos sąsajos ontologijoms loginė architektūra vaizduojama trijų lygių MVC (angl. *Model-View-Controller*) architektūra. Realizuotas sprendimas apibrėžiamas trimis paketais: natūraliosios kalbos sąsaja ontologijoms („*Natural Language Interface to Ontologies*“), semantinė saugykla („*Semantic Repository*“) ir morfologinis analizatorius („*Morphological Analyser*“). Žemiau pateikiama sumodeliuota NKS loginė architektūra (žr. 44 pav.).



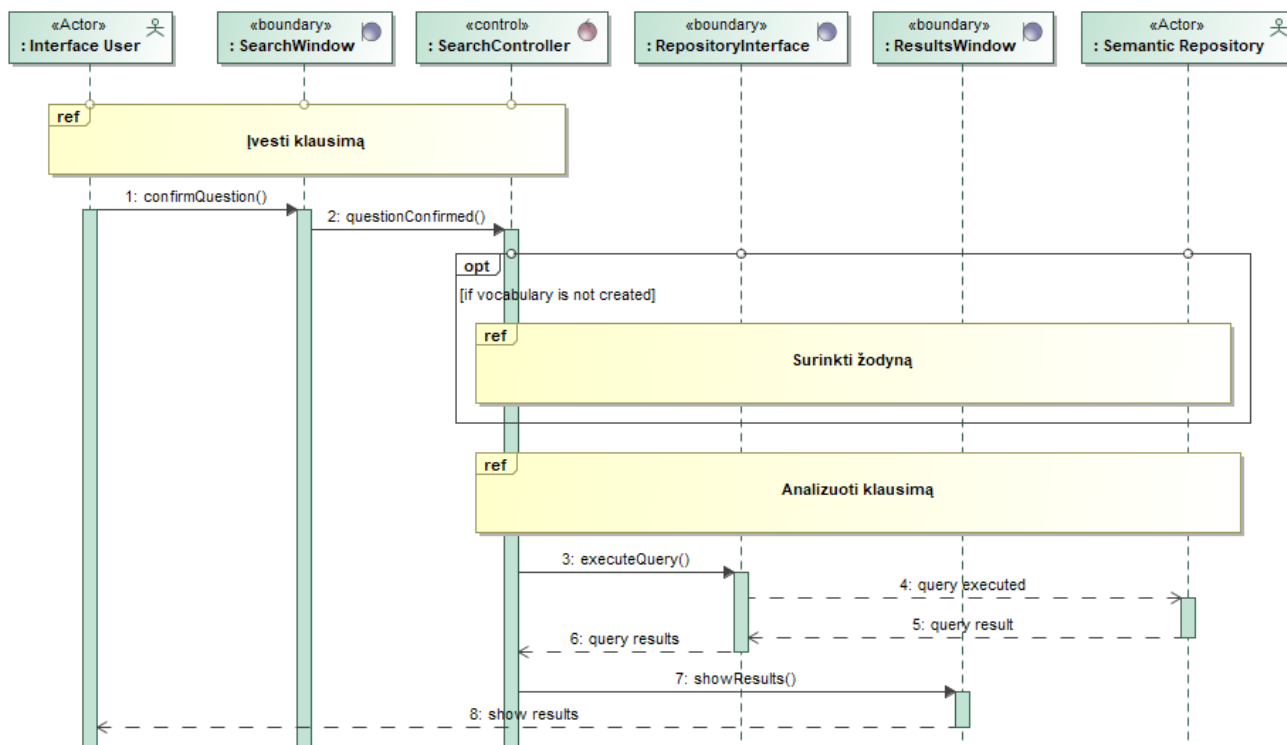
44 pav. Natūraliosios kalbos sąsajos ontologijoms loginė architektūra

NKS ontologijoms komponente modelių pakete („*Models*“) vaizduojama esybė *OWL_Vocabulary*, kuri apibendrina klausimų analizei ir transformavimui naudojamą žodyną. Šis žodynas yra sudaromas automatiškai pagal ontologijos schemą. Valdiklių pakete („*Controllers*“) vaizduojami valdikliai, kuriuose realizuojami visi kuriamo metodo algoritmai ir kurie užtikrina esybės, morfologinio analizatoriaus, semantinės saugyklos ir naudotojo sąsajos komunikavimą. Sąsajų pakete („*Interfaces*“) vaizduojamos grafinės naudotojo sąsajos, atsakingos už duomenų pateikimą sąsajos naudotojui bei programinė sąsaja, skirta prisijungti prie semantinės saugyklos. Morfologinio analizatoriaus paketas („*Morphological Analyser*“) yra skirtas išanalizuoti pateiktą klausimą morfologiškai. Semantinės saugyklos paketas („*Semantic Repository*“) atsakingas už ontologijų saugojimą.

3.2. Natūraliosios kalbos sąsajos ontologijoms kompiuterizuojamų panaudojimo atveju realizacija sekų diagramomis

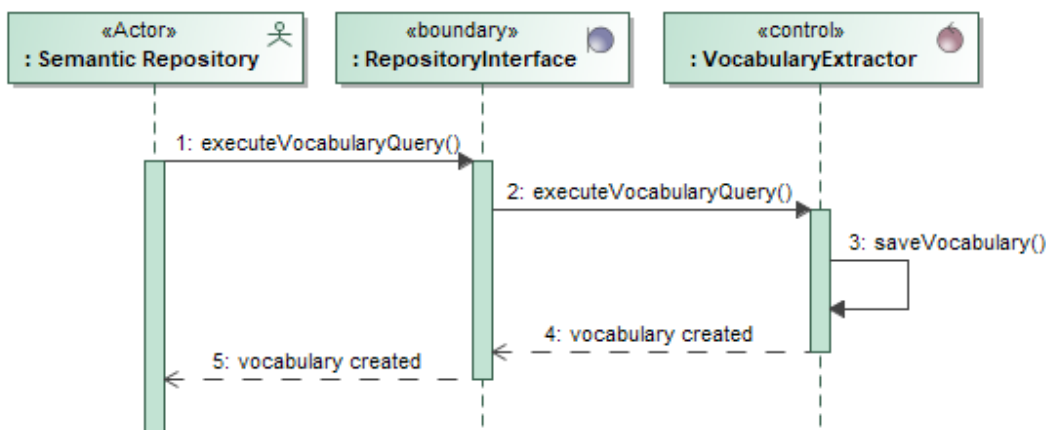
Remiantis atlikta NKS kompiuterizuojamų PA specifikacija ir suprojektuota logine architektūra buvo atlikta kiekvieno PA realizacija sekų diagramomis. Sekų diagramos vaizduoja kiekvieno specifikuoto kompiuterizuojamo PA veiklos vykdymo seką, kurią sudaro esybės, valdiklio ir naudotojo sąsajos elementai bei pranešimai. Žemiau pateikiami suprojektuoti kiekvieno kompiuterizuojamo PA scenarijai (žr. 45 pav. – 49 pav.).

Pagrindinį sukurto sprendimo funkcionalumą apibūdina PA „Vykdėti paiešką“, kuriame sąsajos naudotojas įveda klausimą ir patvirtina paieškos vykdymą. Natūraliojoje kalbos sąsajoje paieškos vykdymas ir klausimo analizavimas paremtas *OWL* žodynu, kuris turi būti išgaunamas iki klausimo tikrinimo pradžios. Išanalizuotas klausimas yra transformuojamas į *SPARQL* užklausą, kurią įvykdžius semantinėje saugykloje, sąsajos naudotojui pateikiami gauti rezultatai. Žemiau pateikiama PA „Vykdėti paiešką“ sekų diagrama (žr. 45 pav.).



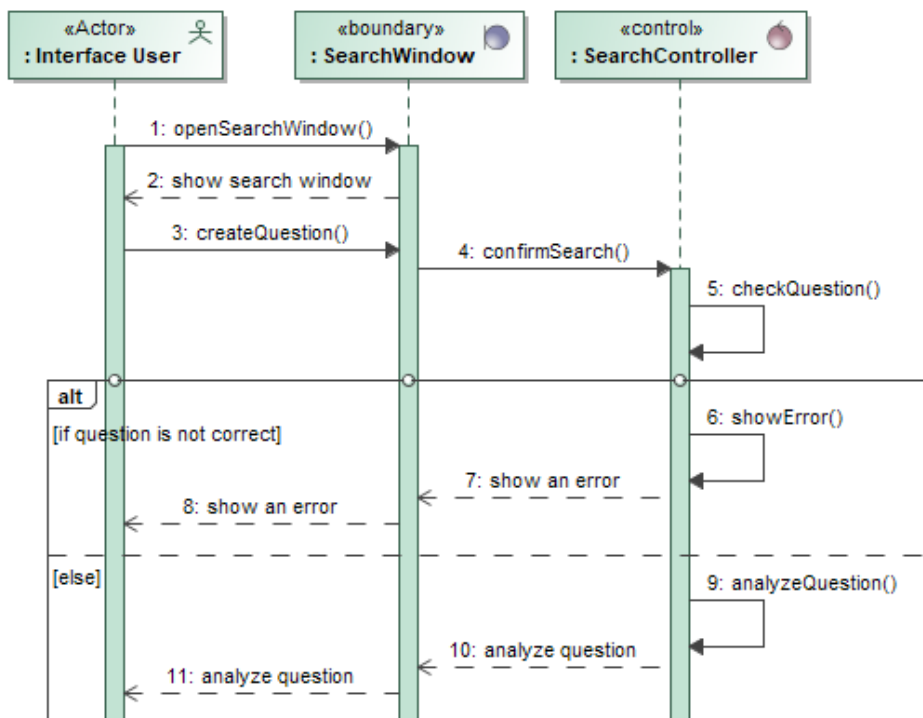
45 pav. Panaudojimo atvejo „Vykdėti paiešką“ sekų diagrama

Klausimų paieškos vykdymui naudojama sukurta ontologijos schema, kuria remiantis NKS išgauna *OWL* žodyną vykdant užklausas semantinėje saugykloje. Žemiau pateikiama PA „Gauti žodyną iš ontologijos“ sekų diagrama (žr. 46 pav.).



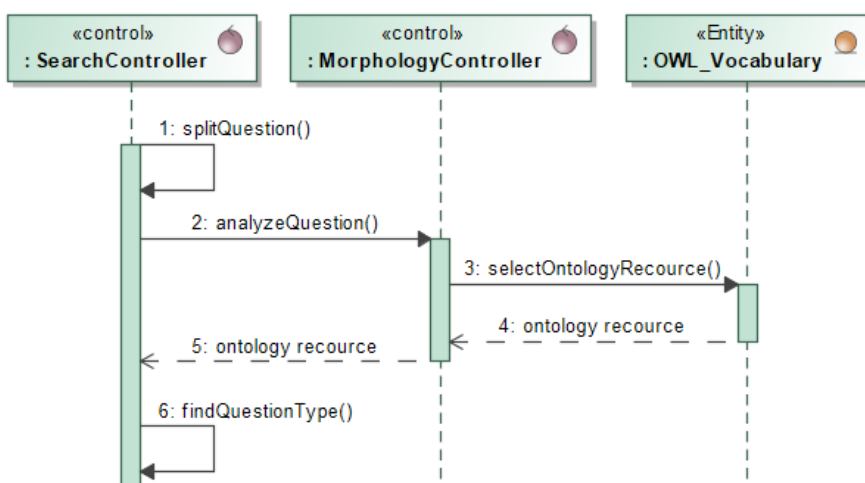
46 pav. Panaudojimo atvejo „Gauti žodyną iš ontologijos“ sekų diagrama

Realizuotoje natūraliosios kalbos sąsajoje paieškos vykdymo procesas prasideda nuo klausimo įvedimo, kurį atlieka sąsajos naudotojas. Žemiau pateikiama PA „Įvesti ir tikrinti klausimą“ sekų diagrama (žr. 47 pav.).

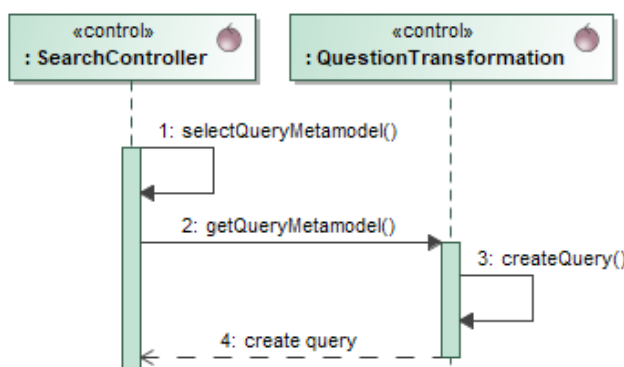


47 pav. Panaudojimo atvejo „Įvesti ir tikrinti klausimą“ sekų diagrama

Teisingai suformuluotas klausimas yra išanalizuojamas ir nustatomas klausimo tipas, kuriuo remiantis sukuriama transformavimo užklausa. Žemiau pateikiamos PA „Analizuoti klausimą“ ir PA „Transformuoti klausimą į užklausą“ sekų diagrama (žr. 48 pav. ir 49 pav.).



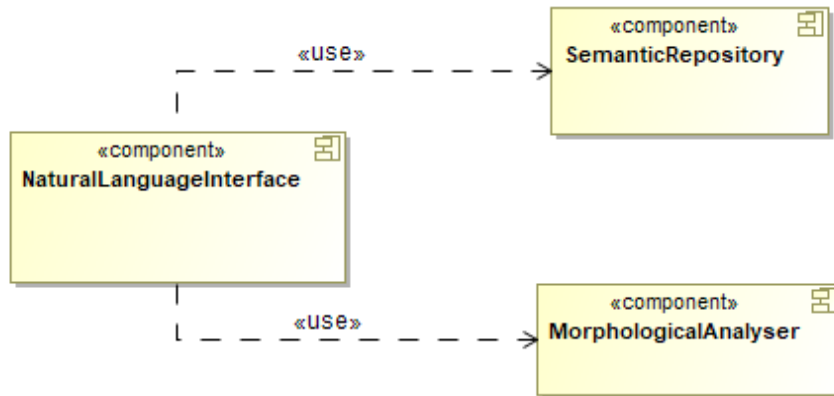
48 pav. Panaudojimo atvejo „Analizuoti klausimą“ sekų diagrama



49 pav. Panaudojimo atvejo „Transformuoti klausimą į užklausą“ sekų diagrama

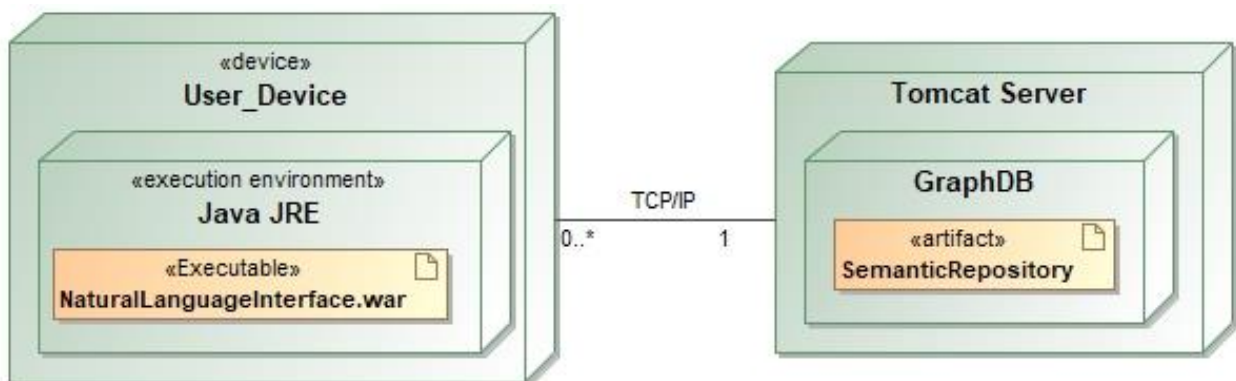
3.3. Natūraliosios kalbos sąsajos ontologijoms realizavimo ir diegimo modeliai

Šiame skyrelyje pateikiamos sukurtos NKS komponentų ir diegimo diagramos. Komponentų diagramą (žr. 50 pav.) sudaro trys komponentai. *NaturalLanguageInterface* komponente yra realizuoti visi šiame darbe aprašyti klausimų analizės ir transformavimo algoritmai. Komponentas naudoja išorinę morfologinės analizės biblioteką. Žodynui gauti ir užklausoms vykdyti kreipiamasi į semantinę saugyklą.



50 pav. Natūraliosios kalbos sąsajos ontologijoms komponentų diagrama

Diegimo diagramoje (žr. 51 pav.) vaizduojami sukurtai natūraliosios kalbos sąsajai veikti reikalingi įrenginiai, jų tarpusavio ryšiai bei programiniai paketai ir elementai, veikiantys tuose aparatinuose įrenginiuose. Sąsaja realizuota *Java* programavimo kalba ir sukompilijuojama į *NaturalLanguageInterface.jar* failą, kuris gali būti paleidžiamas kompiuteryje, kuriame įdiegta *Java JRE* aplinka. Į šį archyvą įtraukiamas ir morfologinės analizės bibliotekos failas (*hunspell.dll*) ir lietuvių kalbos morfologinis žodynas. Šių artefaktų atskirai diegti nereikia ir diagramoje jie nerodomi. Ontologijoms saugoti ir užklausoms vykdyti naudojama *Tomcat* serveryje veikianti *GraphDB* ontologijų duomenų bazė, kurioje sukuriamos atskiros semantinės saugyklos skirtingoms dalykinėms sritims. Ontologijos (schemos ir egzemplioriai) į *GraphDB* saugyklą gali būti importuojamos paruošus jas su *Protégé* įrankiu.

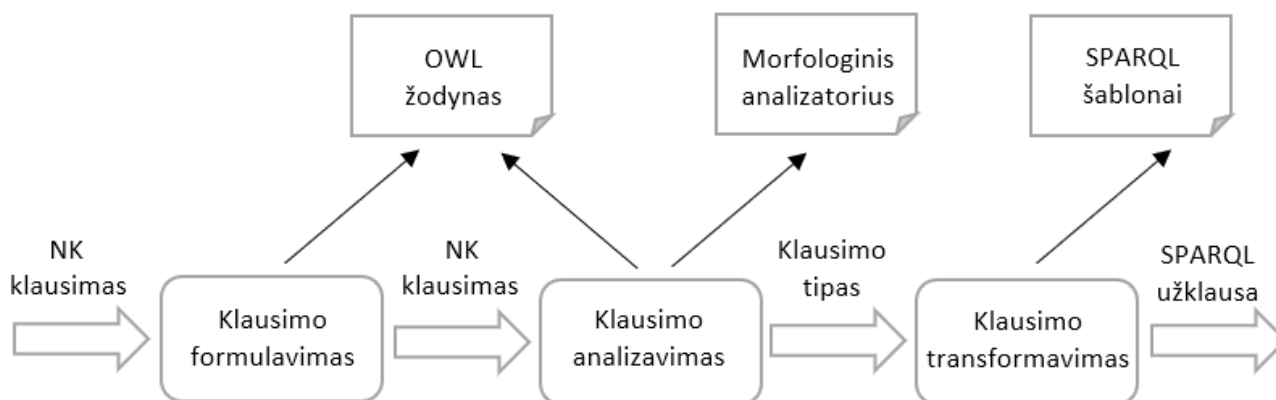


51 pav. Natūraliosios kalbos sąsajos ontologijoms diegimo diagrama

4. Natūraliosios kalbos sąsajos ontologijoms realizacija ir testavimas

4.1. Natūraliosios kalbos sąsajos ontologijoms realizacijos ir veikimo aprašymas

Siekiant iširti tiriamojo projekto metu sukurtą metodą buvo realizuotas natūraliosios kalbos sąsajos ontologijoms prototipas, kuris leidžia įvesti klausimą lietuvių kalba, transformuoti į *SPARQL* užklausą, ją įvykdyti ir peržiūrėti paieškos rezultatus. Remiantis apibrėžta NKS ontologijoms reikalavimų specifikacija ir atliktų panašių sistemų analize buvo nustatyti pagrindiniai pateikto klausimo apdorojimo etapai: klausimo formulavimas, analizavimas ir transformavimas į *SPARQL* užklausą. Šie etapai ir juose naudojami artefaktai pateikti žemiau esančioje diagramoje (žr. 52 pav.).



52 pav. Natūraliosios kalbos sąsajos ontologijoms komponentai

Apibrėžto metodo veikimas realizuojamas algoritmu, kuris apima pagrindinius pateikto klausimo apdorojimo etapus. Semantinės paieškos vykdymo procesas pradedamas nuo klausimo įvedimo natūraliaja kalba. Teisingai pateiktas klausimas yra pradedamas analizuoti naudojant morfologinį analizatorių ir susiejant klausimo žodžius su *OWL* žodyno sąvokomis. Nustačius, kokius ontologijos resursus atitinka pateiktas klausimas, yra tikrinamas klausimo tipas, remiantis aprašytais jų nustatymo taisyklėmis. Nustačius klausimo tipą, jis transformuojamas į užklausą, kuri sudaroma naudojant *SPARQL* šablonus, aprašytus konkrečiam klausimo tipui. Šie etapai yra realizuoti kaip konsolinė programa naudojant *Java* programavimo kalbą. Jie detaliau aprašomi sekančiuose skyreliuose (žr. 4.1.1 – 4.1.3 skyrius).

4.1.1. Klausimo formulavimas

Siekiant sukurti lengvai suprantamą NKS, naudotojui pateikiamas vienas paieškos laukelis, skirtas klausimo įvedimui ir paieškos vykdymui. Tokiu būdu sąsajos naudotojas gali formuluoti klausimą natūraliaja kalba ir stebėti, ką į jo veiksmus „*atsako*“ sistema. NKS geba interpretuoti ir atsakyti tik į tokius klausimus, kurie pagrįsti *OWL* žodyno sąvokomis. Dėl šios priežasties klausimus formuluoti sistemoje galima naudojant žodyno terminus, parašytus taisyklinga lietuvių kalba bei sinonimus. Jeigu klausimas yra pateikiamas netaisyklingai (pvz., formuluojant padaroma rašybos klaida), sistemoje rodomas klaidos pranešimas. Tokiais atvejais, kai klausime yra naudojami neįprasti simboliai (pvz., klausimo viduryje parašytas kablelis), kurių NKS negali interpretuoti kaip klausimo žodžių, yra sukurta funkcija jiems pašalinti. Tam naudojamas iš anksto aprašytas nenaudotinių simbolių sąrašas, kuriuo remiantis gali būti tikrinami neatpažintų simbolių atitikimai.

Tokiais atvejais, kai klausimas interpretuojamas dviprasmiškai, NKS pateikia naudotojui patikslinamąjį dialogą ir paprašo naudotojo patikslinti, kaip klausimas turi būti interpretuojamas.

Pavyzdžiui, klausime „*Kokia prekė yra didžiausia?*“ NKS negali interpretuoti, pagal kokią savybę turi būti nustatomas prekės dydis, t.y. ilgį, plotį ar aukštį. Tokiu atveju sistema pateikia patikslinamąjį dialogą, kuriame naudotojas turi pasirinkti jam tinkamą interpretaciją. Žemiau pateikiamas nurodyto pavyzdinio klausimo patikslinamasis dialogas (žr. 53 pav.).

```
PATIKSLINIMAS. Kokių duomenų ieškote?  
1 ilgis  
2 plotis  
3 aukštis  
Įveskite eilutės numerį:  
...
```

53 pav. Patikslinamojo dialogo pavyzdys

Naudotojui pateiktas patikslinamasis dialogas rodo klasės „*Prekė*“ rastų duomenų savybių sąrašą. Pateiktame lange sąsajos naudotojas turi pasirinkti duomenų savybę, pagal kurią bus nustatomas prekės dydis. Patikslinus duomenis, NKS geba interpretuoti klausimą ir pateikti teisingus paieškos rezultatus.

Suformulavus klausimą ir jį patvirtinus, pradedamas paieškos vykdymas. Sistema gali atsakyti tik į tokius klausimus, kuriuose naudojami žodžiai gali būti susiejami su žodynu, kuris sukuriamas vykdant užklausas *GraphDB* semantinėje saugykloje saugojamoje ontologijos schemoje (žr. 12 pav.). Ontologijos schemoje naudojant ekvivalentiškumo klasių ir savybių aksiomas gali būti aprašomi sinonimai, kurie suteikia naudotojui galimybę pateikti įvairesnius klausimus. Kai klausimuose yra naudojami sinonimai, juos analizuojant pirmiausia yra randama pagrindinė forma. Pavyzdžiui, klausime vartojamas žodis „*Pirkėjas*“ ontologijoje atitinka klasę „*Klientas*“ ir yra jos sinonimas. Tam, kad naudotojo formuluojamas klausimas būtų suprantamas paieškos sistemai, rastai klasei naudojama pagrindinė jos forma „*Klientas*“. Tokiu būdu analizuojant klausimo prasmę NKS gali nesudėtingai rasti ontologijoje aprašytus resursus, kurie yra susieti su rasta klase.

4.1.2. Klausimo analizavimas

Naudotojui patvirtinus įvesto klausimo paiešką atliekama klausimo analizė, kurios pagalba NKS gali interpretuoti pateikto klausimo prasmę. Pagrindinis analizės tikslas – nustatyti klausimo tipą, kuriuo grindžiamas pateiktas naudotojo klausimas. Analizavimas realizuotas *Java* programavimo kalba naudojant morfologinį analizatorių ir remiantis sukurtu *OWL* žodynu. Klausimo analizė vykdoma šiais žingsniais:

1. pradinis teksto apdorojimas

Naudojant *replace()* funkciją pašalinami analizei nereikalingi simboliai (kableliai, brūkšniai, taškai ir kt.). Jeigu parašyti du ar daugiau tarpo simbolių, jie pakeičiami į vieną. Taip pat naudojant *trim()* funkciją pašalinami tarpo simboliai iš klausimo pradžios ir pabaigos.

2. tokenizavimas

Pateiktas naudotojo klausimas yra išskaidomas į atskirus žodžius. Skaidymas yra atliekamas naudojant *split()* funkciją, kurios pagalba klausimas atskiriamas į žodžių sąrašą, remiantis nurodytu skyrikliu (pvz. tarpu).

3. morfologinė analizė

Klausimo žodžiai yra išnagrinėjami morfologiškai naudojant morfologinės analizės biblioteką *Hunspell* ir morfologinį lietuvių kalbos žodyną. Pirmiausia žodžiui yra surandama pagrindinė jo forma (lema) ir kalbos dalis, kurių pagalba žodis gali būti susiejamas su žodyno sąvokomis.

4. sąsaja su žodynu

Kiekvienas klausimo žodis, remiantis nustatyta kalbos dalimi, programiškai yra susiejamas su žodynu (žr. 19 pav.). Tikrinimo metu identifikuotas ontologijos resursas yra priskiriamas atpažintų resursų sąrašui, kurie vėliau naudojami klausimo tipo nustatymui.

5. nustatomas klausimo tipas

Identifikavus klausimo žodžius, atitinkančius ontologijos resursus (pvz., klases, objektines savybes, duomenų savybes ir kt.), tikrinamas pateikto klausimo tipas. Klausimo tipo nustatymas vykdomas programiškai tikrinant atpažintus resursus ir remiantis aprašytomis taisyklėmis (žr. 21 pav.).

4.1.3. Klausimo transformavimas

Nustačius klausimo tipą, pateiktas klausimas yra transformuojamas į *SPARQL* užklausą. Klausimo transformavimas realizuotas pagal aprašytus šešių klausimų tipų algoritmus, kurie pateikti 2.4.2 skyriuje. Klausimo transformavimas buvo vykdomas *Java* programavimo kalba, naudojant *SPARQL* užklausų šablonus kiekvienam klausimo tipui. Sukurtoje NKS užklausos aprašytos tekstiniais *SPARQL* šablonais su įterptiniais kintamaisiais (angl. *placeholder*), skirtais įterpti klausime identifikuotus ontologijos resursus (pvz., klasė, objektinė savybė, duomenų savybė), atitinkančius tripletų šablonus, į jiems skirtas vietas. Žemiau pateikiamas pavyzdinio klausimo paieškos vykdymo procesas, apibūdinantis visus pagrindinius transformavimo į užklausą etapus:

1. pateikiamas klausimas – „*Kokius užsakymus pateikė klientai?*“;
2. klausimas išskaidomas į atskirus žodžius: „*kokius*“, „*užsakymus*“, „*pateikė*“, „*klientai*“. Identifikavus klausimo žodžius daiktavardžiai „*klientas*“ ir „*užsakymas*“ atpažįstami kaip klasės, o veiksmažodis „*pateikti*“ kaip objektinė savybė. Tokiu būdu žodyne rasti žodžiai yra sujungiami į vieną objektinę savybę „*Klientas__pateikia__Užsakymas*“;
3. remiantis klausimų tipų tikrinimo taisyklėmis nustatomas klausimo tipas – 2. *Klausimas, remiantis dviem rolėm be apribojimų*. Klausimo transformavimui *SPARQL* užklausos šablone naudojamas tripletų šablonas, paremtas objektine savybe (žr. 54 pav.);

```

queryString = "PREFIX : <http://localhost:8080/mydataset/>" + "\n"
+ "PREFIX vocab: " + uri + "\n"
+ "PREFIX owl: <http://www.w3.org/2002/07/owl#>" + "\n"
+ "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>" + "\n"
+ "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>" + "\n"
+ "PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>" + "\n"
+ "SELECT ?s ?o" + "\n"
+ "WHERE {" + "\n"
+ "?s vocab:" + obj_prop + " ?o." + "\n"
+ "}";

```

54 pav. SPARQL užklauso šablonas klausimo, kuris remiasi dviem rolėm be apribojimų, transformavimui

4. pateikiama galutinė SPARQL užklausa, kuri vykdoma ontologijoje (žr. 55 pav.);

```

PREFIX : <http://localhost:8080/mydataset/>
PREFIX vocab: <http://www.semanticweb.org/lenovo/ontologies/2022/2/el-prekyba#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?s ?o
WHERE {
?s vocab:Klientas__pateikia__Uzsakymas ?o.
}

```

55 pav. SPARQL užklausa klausimo, kuris remiasi dviem rolėm be apribojimų, transformavimui

5. įvykdžius sukurtą užklausą ontologijoje pateikiamas gautų rezultatų sąrašas (žr. 56 pav.).

```

UŽKLAUSOS REZULTATAI:

| Klientas_37811165722 | Uzsakymas_S0001
| Klientas_37811165722 | Uzsakymas_S0022
| Klientas_37811165722 | Uzsakymas_S0020
| Klientas_37811165722 | Uzsakymas_S0024
| Klientas_37912224530 | Uzsakymas_S0023
| Klientas_37912224530 | Uzsakymas_S0002
| Klientas_38202096798 | Uzsakymas_S0025
| Klientas_38202096798 | Uzsakymas_S0003
| Klientas_38208075499 | Uzsakymas_S0004
| Klientas_38401181234 | Uzsakymas_S0005
| Klientas_38409140932 | Uzsakymas_S0006
| Klientas_38409171134 | Uzsakymas_S0007
| Klientas_38512014587 | Uzsakymas_S0008
| Klientas_38807080955 | Uzsakymas_S0009

```

56 pav. Įvykdytos užklauso rezultatų sąrašas

4.2. Testavimo modelis, duomenys, rezultatai

Realizuoto sprendimo veikimui patikrinti buvo taikomas naudotojų funkcionalumo testavimas, apimantis pagrindinį panaudojimo atvejį „Vykdėti paiešką“. Semantinės paieškos proceso testavimui buvo naudojami pavyzdiniai klausimai. Žemiau pateikiami išanalizuoti kiekvieno atvejo metu gauti rezultatai (žr. 8 lentelė – 12 lentelė).

8 lentelė. Neatpažinto žodžio testavimo scenarijus

Panaudojimo atvejis	PA1 „Vykdėti paiešką“		
Aprašymas	Scenarijus, kai klausimas panaudoti neatpažįstami žodžiai.		
Aktorius	Sąsajos naudotojas		
Prieš-sąlyga	Naudotojas turi būti pagrindiniame sistemos lange		
Po-sąlyga	Paieška įvykdyta, naudotojui pateikiami rezultatai.		
Nr.	Naudotojo veiksmai	Laukiamas rezultatas	Gautas rezultatas
1.	Pateikiamas klausimas – „Kokius pirkimus pateikė pirkėjai?“ ir patvirtina.	Sistema pateikia klaidos pranešimą.	Teisingas

9 lentelė. Gramatinės klaidos testavimo scenarijus

Panaudojimo atvejis	PA1 „Vykdėti paiešką“		
Aprašymas	Scenarijus, kai klausimas parašytas su gramatine klaida.		
Aktorius	Sąsajos naudotojas		
Prieš-sąlyga	Naudotojas turi būti pagrindiniame sistemos lange		
Po-sąlyga	Paieška įvykdyta, naudotojui pateikiami rezultatai.		
Nr.	Naudotojo veiksmai	Laukiamas rezultatas	Gautas rezultatas
1.	Pateikiamas klausimas – „Kokis užsakymus pateikė klientai?“ ir patvirtina.	Sistema pateikia klaidos pranešimą.	Teisingas

10 lentelė. Teisingo klausimo (be patikslinamojo dialogo) testavimo scenarijus

Panaudojimo atvejis	PA1 „Vykdėti paiešką“		
Aprašymas	Scenarijus, kai klausimas parašytas teisingai ir nenaudojamas patikslinamasis dialogas.		
Aktorius	Sąsajos naudotojas		
Prieš-sąlyga	Naudotojas turi būti pagrindiniame sistemos lange		
Po-sąlyga	Paieška įvykdyta, naudotojui pateikiami rezultatai.		
Nr.	Naudotojo veiksmai	Laukiamas rezultatas	Gautas rezultatas

1.	Pateikiamas klausimas – „Kokius užsakymus pateikė klientai?“ ir patvirtina.	Sistema pateikia gautus paieškos rezultatus.	Teisingas
----	---	--	-----------

11 lentelė. Teisingo klausimo (su patikslinamuoju dialogu) testavimo scenarijus

Panaudojimo atvejis	PA1 „Vykdėti paiešką“		
Aprašymas	Scenarijus, kai klausimas parašytas teisingai ir naudojamas patikslinamasis dialogas.		
Aktorius	Sąsajos naudotojas		
Prieš-sąlyga	Naudotojas turi būti pagrindiniame sistemos lange		
Po-sąlyga	Paieška įvykdyta, naudotojui pateikiami rezultatai.		
Nr.	Naudotojo veiksmai	Laukiamas rezultatas	Gautas rezultatas
1.	Pateikiamas klausimas – „Kokia prekė yra didžiausia?“ ir patvirtina.	Sistema pateikia patikslinamąjį dialogą.	Teisingas
2.	Naudotojas pasirenka vieną iš pateiktų pasirinkimo variantų.	Sistema pateikia gautus paieškos rezultatus.	Teisingas

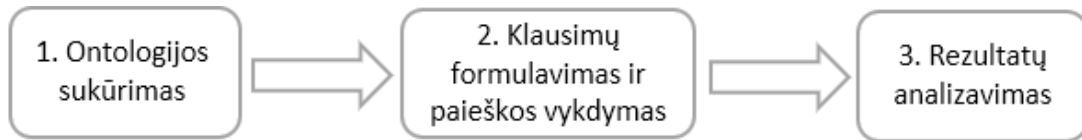
12 lentelė. Teisingo klausimo (naudojant sinonimus) testavimo scenarijus

Panaudojimo atvejis	PA1 „Vykdėti paiešką“		
Aprašymas	Scenarijus, kai klausime parašytas sinonimas.		
Aktorius	Sąsajos naudotojas		
Prieš-sąlyga	Naudotojas turi būti pagrindiniame sistemos lange		
Po-sąlyga	Paieška įvykdyta, naudotojui pateikiami rezultatai.		
Nr.	Naudotojo veiksmai	Laukiamas rezultatas	Gautas rezultatas
1.	Pateikiamas klausimas – „Kokios yra medienos?“ ir patvirtina.	Sistema pateikia gautus paieškos rezultatus.	Teisingas

5. Eksperimentinis natūraliosios kalbos sąsajos ontologijoms tyrimas

5.1. Eksperimento planas

Eksperimento tikslas – įvertinti, ar sukurtas metodas leidžia formuluoti klausimus natūraliaja kalba ir vykdyti semantinę paiešką. Žemiau pateikiamas eksperimento vykdymo planas, kurį sudaro šie etapai: ontologijos sukūrimas, klausimų formulavimas ir paieškos vykdymas bei rezultatų analizavimas (žr. 57 pav.). Kitame skyriuje šie etapai aprašyti detaliau (žr. 5.2 skyrių).



57 pav. Eksperimentinio natūraliosios kalbos sąsajos tyrimo planas

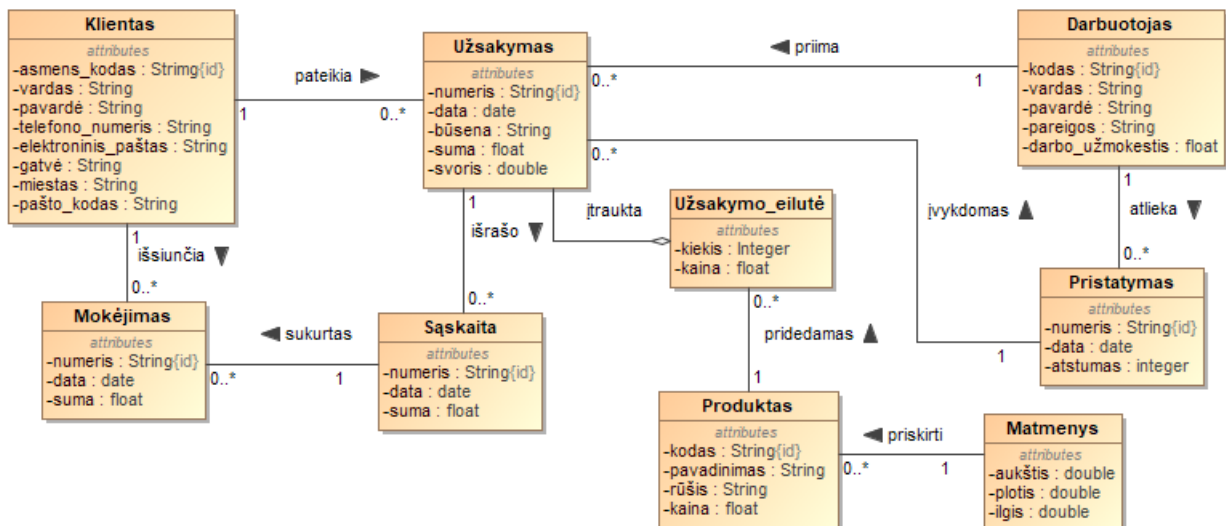
5.2. Eksperimento vykdymas

5.2.1. Ontologijos kūrimas

Viena pagrindinių atliekamo eksperimentinio tyrimo dalių yra ontologijų kūrimas, aprašant skirtingas dalykines sritis. Kiekvienos dalykinės srities ontologijos kūrimui naudojamas atviro kodo ontologijų redaktorius *Protégé*. Sukurta ontologijos schema kartu su egzemplioriais įkeliama į *GraphDB* saugyklą. Eksperimentui buvo pasirinktos dvi dalykinės sritys: elektroninės prekybos ir universiteto veiklos (žr. 5.2.1.1 ir 5.2.1.2 skyrius). Joms buvo suprojektuotos ontologijos schemas laikantis taisyklių, aprašytų 2.4.3 skyriuje.

5.2.1.1. Elektroninės prekybos ontologija

Pirmoji eksperimentiniam tyrimui pasirinkta dalykinė sritis – elektroninė prekyba medienos gaminiais. Šiai sričiai sukurta ontologija aprašo pagrindinius elektroninės prekybos procesus, nuo užsakymo pateikimo iki jo apmokėjimo ir pristatymo klientui. Šioje dalykinėje srityje bus siekiama atsakyti į tokius pavyzdinius klausimus: „Kokių yra prekių?“, „Kokio produkto kaina yra didžiausia?“, „Kokius užsakymus pateikė klientas Jonas?“, „Kiek iš viso yra užsakymų?“ ir kt. Žemiau pateikiamas šios dalykinės srities esybių ir ryšių modelis (žr. 58 pav.). Pagal šį modelį buvo sukurta ontologijos schema įrankiu *Protégé*.

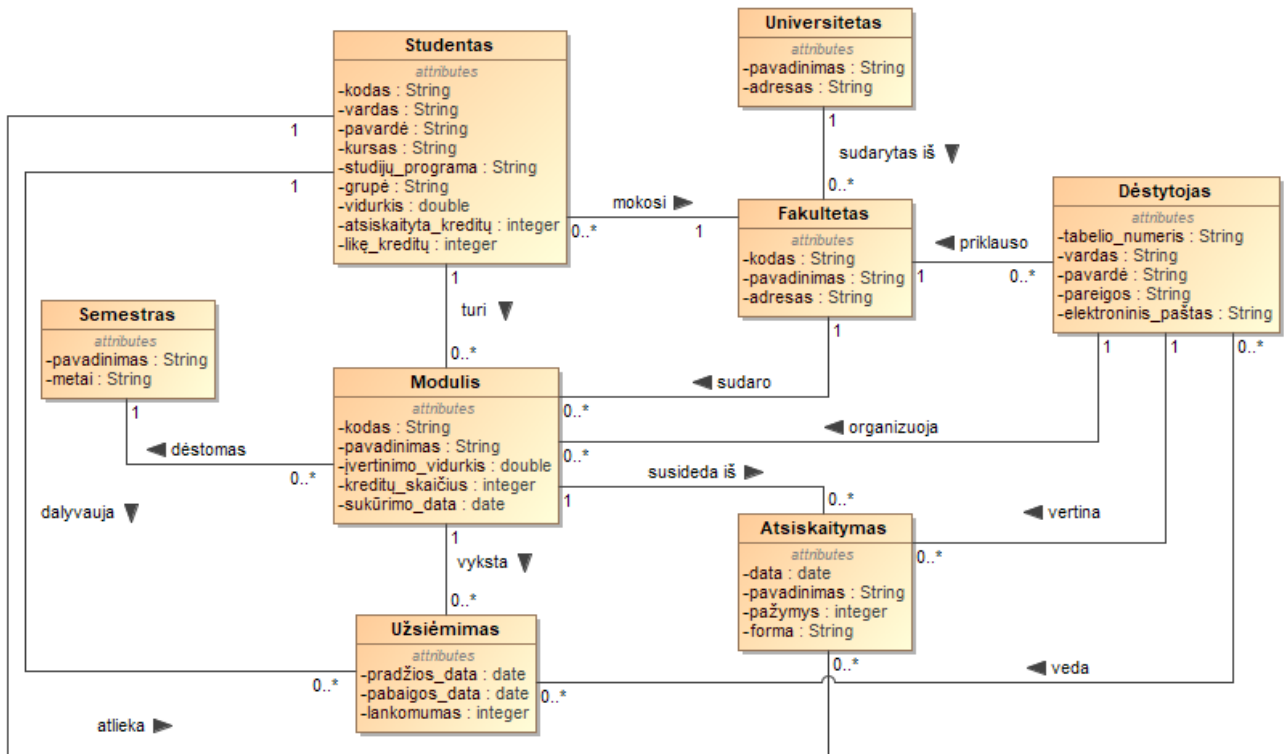


58 pav. Elektroninės prekybos dalykinės srities esybių ir ryšių modelis

5.2.1.2. Universiteto veiklos ontologija

Antroji eksperimentiniam tyrimui pasirinkta dalykinė sritis – universiteto studijų veikla. Šiai sričiai sukurta ontologija aprašo pagrindines universitetui aktualias duomenų struktūras: fakultetus, studentus, dėstytojus, vedamus modulius, organizuojamus atsiskaitymus ir kt. Šioje dalykinėje srityje bus siekiama atsakyti į tokius pavyzdinius klausimus: „Kurio studento pažymys yra aukščiausias?“, „Kokius modulius veda dėstytojas pavarde Jonaitis?“, „Kiek studentų mokosi universitete?“, „Kokie studentai priklauso grupei IFM/0-4?“ ir kt.

Žemiau pateikiamas šios dalykinės srities esybių ir ryšių modelis (žr. 59 pav.). Pagal šį modelį buvo sukurta ontologijos schema įrankiu *Protégé*.



59 pav. Universiteto veiklos dalykinės srities esybių ir ryšių modelis

5.2.2. Klausimų formulavimas ir paieškos vykdymas

Sukurto metodo veikimas analizuojamas vykdant paiešką, suskirsčius klausimus pagal jų tipus. Eksperimentinio tyrimo metu buvo sudarytas klausimų sąrašas konsultuojantis su kiekvienos dalykinės srities ekspertais bei remiantis asmenine patirtimi. Klausimai buvo sudaromi remiantis taisyklėmis, aprašytomis 2.4.4 skyriuje. Buvo atrinkta po dešimt klausimų kiekvienam tipui dalykinėse srityse. Vykdant paiešką buvo tikrinami du pagrindiniai kriterijai:

- **atsakymo gavimas** – nurodo, ar sistema sugebėjo interpretuoti klausimą, suformuoti užklausą ir pateikti atsakymą;
- **atsakymo teisingumas** – nurodo, ar į klausimą buvo atsakyta teisingai.

Žemiau pateikiami eksperimentinio tyrimo metu vykdomos paieškos rezultatai kiekvienoje dalykinėje srityje (žr. 5.2.2.1 ir 5.2.2.2 skyriai).

5.2.2.1. Elektroninės prekybos ontologija

Elektroninės prekybos medienos gaminių dalykinės srities eksperimentiniam tyrimui buvo sudarytas pavyzdinių klausimų rinkinys kiekvienam klausimo tipui. Žemiau pateikiami kiekvieno klausimo vykdomos paieškos rezultatai. Taip pat pateikiami klausimai, į kuriuos sistema nesugebėjo atsakyti teisingai, ir to priežastys (žr. 13 lentelė –24 lentelė).

1. Klausimas remiantis viena role be apribojimų (1 klausimo tipas)

13 lentelė. Klausimų, remiantis viena role be apribojimų, paieška (Elektroninė prekyba)

Nr.	Klausimas	Atsakymas gautas	Atsakymas teisingas
1.	Kokių yra medienos rūšių?	+	+
2.	Koks yra sandėlio likutis?	+	+
3.	Raskite užsakymų datas?	+	+
4.	Raskite įmonės krovėjus?	–	–
5.	Kokios medienos kainos?	+	+
6.	Raskite darbuotojų komandas?	+	–
7.	Kokios yra užsakymų sumos?	+	+
8.	Kokie darbuotojų atlyginimai?	+	+
9.	Kokių yra gaminių sandėlyje?	+	+
10.	Koks yra pristatymo svoris?	+	–
Iš viso:		9	7

Vykdamas pirmojo klausimo tipo klausimų paiešką, iš dešimt klausimų atsakymas buvo gautas į devynis pateiktus klausimus, tačiau tik septynių klausimų rezultatai buvo teisingi. Žemiau pateikiamas neteisingų rezultatų vertinimas:

14 lentelė. Klausimų, remiantis viena role be apribojimų, neteisingų rezultatų vertinimas (Elektroninė prekyba)

Klausimas	Priežastis	Sprendimas
4. Raskite įmonės krovėjus?	Klausime naudojami žodžiai nebuvo atpažinti sukurtoje ontologijoje.	Papildyti ontologiją naujomis sąvokomis (sinonimais).
6. Raskite darbuotojų komandas?	Ontologijoje darbuotojo klasei nėra sukurta tokia duomenų savybė kaip komanda.	Papildyti ontologijos schemą nauja duomenų savybe.

10. Koks yra pristatymo svoris?	Ontologijoje pristatymo klasei nėra sukurta tokia duomenų savybė kaip svoris.	Papildyti ontologijos schemą nauja duomenų savybe.
---------------------------------	---	--

2. Klausimas remiantis dviem rolėm be apribojimų (2 klausimo tipas)

15 lentelė. Klausimų, remiantis dviem rolėm be apribojimų, paieška (Elektroninė prekyba)

Nr.	Klausimas	Atsakymas gautas	Atsakymas teisingas
1.	Kokius užsakymus pateikė klientai?	+	+
2.	Kokias prekes užsakė klientai?	+	-
3.	Kokias sąskaitas apmokėjo klientai?	+	-
4.	Kokie vadybininkai priėmė užsakymus?	+	-
5.	Kokie vairuotojai pristatė užsakymus?	+	-
6.	Kokiems užsakymams priklauso pristatymas?	+	+
7.	Kokiems užsakymams išrašyta sąskaita?	+	+
8.	Kokia mediena įtraukta į užsakymus?	+	+
9.	Kokius pirkimus atliko pirkėjai?	-	-
10.	Kokie darbuotojai išrašo sąskaitas?	+	-
Iš viso:		9	4

Vykdam antrojo klausimo tipo klausimų paiešką, iš dešimt klausimų atsakymas buvo gautas į devynis pateiktus klausimus, tačiau tik keturių klausimų rezultatai buvo teisingi. Žemiau pateikiamas neteisingų rezultatų vertinimas:

16 lentelė. Klausimų, remiantis dviem rolėm be apribojimų, neteisingų rezultatų vertinimas (Elektroninė prekyba)

Klausimas	Priežastis	Sprendimas
2. Kokias prekes užsakė klientas?	Ontologijoje kliento ir užsakymo prekės klasės neturi jas tarpusavyje siejančio ryšio, t. y. objekcinės savybės.	Papildyti ontologijos schemą nauja objektine savybe.
3. Kokias sąskaitas apmokėjo klientai?	Ontologijoje kliento ir sąskaitos klasės neturi jas tarpusavyje siejančio ryšio, t. y. objekcinės savybės.	Papildyti ontologijos schemą nauja objektine savybe.
4. Kokie vadybininkai priėmė užsakymus?	NKS pateikia rezultatus remiantis abiem objekcinės savybės „Darbuotojas__ priėmė__Užsakymas“ klasėmis.	Sukurti naują klausimo tipą, klausimų išvedimui naudojant vieną klasę ir modifikavimo konstrukta.

5. Kokie vairuotojai pristatė užsakymus?	Ontologijoje darbuotojo ir užsakymo klasės neturi jas tarpusavyje siejančio ryšio, t. y. objektinės savybės.	Papildyti ontologijos schemą nauja objektine savybe.
9. Kokius pirkimus atliko pirkėjai?	Klausime naudojami žodžiai nebuvo atpažinti sukurtoje ontologijoje.	Papildyti ontologiją naujomis sąvokomis (sinonimais).
10. Kokie darbuotojai išrašo sąskaitas?	Ontologijoje darbuotojo ir sąskaitos klasės neturi jas tarpusavyje siejančio ryšio, t. y. objektinės savybės.	Papildyti ontologijos schemą nauja objektine savybe.

3. Klausimas naudojant agregavimo operatorių remiantis viena role (3 klausimo tipas)

17 lentelė. Klausimų, naudojant agregavimo operatorių remiantis viena role, paieška (Elektroninė prekyba)

Nr.	Klausimas	Atsakymas gautas	Atsakymas teisingas
1.	Kiek įmonėje dirba krovėjų?	–	–
2.	Kiek yra medienos rūšių?	+	–
3.	Koks yra vidutinis darbo užmokestis?	+	+
4.	Koks maksimalus leistinas pristatymo svoris?	+	–
5.	Kuris užsakymas yra lengviausias?	+	+
6.	Kokia ilgiausia mediena yra sandėlyje?	+	–
7.	Kuris užsakymas yra naujausias?	+	+
8.	Koks vidutinis užsakymų svoris?	+	+
9.	Kokia medienos rūšis yra brangiausia?	+	–
10.	Kuris pristatymas yra tolimiausias?	+	+
Iš viso:		9	5

Vykdamas trečiojo klausimo tipo klausimų paiešką, iš dešimt klausimų atsakymas buvo gautas į devynis pateiktus klausimus, tačiau tik penkių klausimų rezultatai buvo teisingi. Žemiau pateikiamas neteisingų rezultatų vertinimas:

18 lentelė. Klausimų, naudojant agregavimo operatorių remiantis viena role, neteisingų rezultatų vertinimas (Elektroninė prekyba)

Klausimas	Priežastis	Sprendimas
1. Kiek įmonėje dirba krovėjų?	Klausime naudojami žodžiai nebuvo atpažinti ontologijoje.	Papildyti ontologiją naujomis sąvokomis (sinonimais).

2. Kiek yra medienos rūšių?	NKS rastą duomenų savybę „Produktas_turi_rūšį“ interpretavo kaip atskirą įrašą, kuriam buvo apskaičiuotas bendras kiekis.	Sukurti naują klausimo tipą, panaudojant agregavimo operatorių ir papildomą duomenų modifikavimo konstrukta.
4. Koks maksimalus leistinas užsakymo svoris?	Ontologijoje darbuotojo klasei nėra sukurta tokia duomenų savybė kaip leistinas svoris.	Papildyti ontologijos schemą nauja duomenų savybe.
6. Kokia ilgiausia mediena yra sandėlyje?	Ontologijoje prekės klasei nėra sukurta tokia duomenų savybė kaip ilgis.	Papildyti ontologijos schemą nauja duomenų savybe.
9. Kokia medienos rūšis yra brangiausia?	NKS negali apdoroti šio klausimo taip, kad rezultate būtų pateikiamas duomenų savybių išvedimas.	Sukurti naują klausimo tipą, kuriame būtų pakoreguojamas rezultatų išvedimas trūkstama duomenų savybe.

4. Klausimas naudojant agregavimo operatorių remiantis dviem rolėm (4 klausimo tipas)

19 lentelė. Klausimų, naudojant agregavimo operatorių remiantis dviem rolėm, paieška (Elektroninė prekyba)

Nr.	Klausimas	Atsakymas gautas	Atsakymas teisingas
1.	Kiek iš viso gauta užsakymų iš klientų?	+	+
2.	Kokios medienos yra nupirkta daugiausiai?	+	+
3.	Kiek medienos nupirko klientai?	+	+
4.	Kokia gautų užsakymų bendra apyvarta?	+	-
5.	Kokia vidutinė į užsakymus įtrauktų prekių kaina?	+	+
6.	Kokia klientų pateiktų užsakymų suma?	+	+
7.	Kiek vidutiniškai pristatymų įvykdė darbuotojai?	+	-
8.	Kiek iš viso prekių buvo užsakyta?	+	-
9.	Kokio kliento pateiktas užsakymas yra brangiausias?	+	+
10.	Kiek į pristatymą įtraukta lentų?	-	-
Iš viso:		9	6

Vykdamas ketvirtojo klausimo tipo klausimų paiešką, iš dešimt klausimų atsakymas buvo gautas į devynis pateiktus klausimus, tačiau tik šešių klausimų rezultatai buvo teisingi. Žemiau pateikiamas neteisingų rezultatų vertinimas:

20 lentelė. Klausimų, naudojant agregavimo operatorių remiantis dviem rolėm, neteisingų rezultatų vertinimas (Elektroninė prekyba)

Klausimas	Priežastis	Sprendimas
4. Kokia gautų užsakymų bendra apyvarta?	NKS nepalaiko klausimo turinčio objektinę savybę ir agregavimo operatorių SUM vykdymo.	Patobulinti metodą sukuriant apribojimus.
7. Kiek vidutiniškai pristatymų įvykdė darbuotojai?	NKS nepalaiko filtravimo pagal agregavimo funkcijos suskaičiuotą reikšmę.	Sukurti naują klausimo tipą panaudojant vidinę užklausą.
8. Kiek iš viso prekių buvo užsakyma?	NKS nepalaiko filtravimo pagal agregavimo funkcijos suskaičiuotą reikšmę.	Sukurti naują klausimo tipą panaudojant vidinę užklausą.
10. Kiek į pristatymą įtraukta lentų?	Klausime naudojami žodžiai nebuvo atpažinti ontologijoje.	Papildyti ontologiją naujomis sąvokomis (sinonimais).

5. Klausimas naudojant aritmetinį operatorių (5 klausimo tipas)

21 lentelė. Klausimų, naudojant aritmetinį operatorių, paieška (Elektroninė prekyba)

Nr.	Klausimas	Atsakymas gautas	Atsakymas teisingas
1.	Kokie kroviniai pristatyti iki 2022-01-31?	+	+
2.	Kokių medienos gaminių plotis mažesnis nei 25 centimetrai?	+	+
3.	Kurių prekių kaina didesnė nei 10 eurų?	+	+
4.	Kurie užsakymai viršija 100 kilogramų?	+	+
5.	Kuriuose užsakymuose yra daugiau nei 10 prekių?	+	-
6.	Kurių sąskaitų data yra tokia pati kaip atlikto mokėjimo data?	+	-
7.	Kurie mokėjimai viršija 500 eurų?	+	+
8.	Kokie pirkimai buvo pateikti iki 2022-03-01?	+	+
9.	Kokie pardavimai atlikti 2022 metais?	-	-
10.	Kokie mokėjimai buvo atlikti nuo 2022-04-01 iki 2022-04-31?	+	-
Iš viso:		9	6

Vykdamas penktojo klausimo tipo klausimų paiešką, iš dešimt klausimų atsakymas buvo gautas į devynis pateiktus klausimus, tačiau tik šešių klausimų rezultatai buvo teisingi. Žemiau pateikiamas neteisingų rezultatų vertinimas:

22 lentelė. Klausimų, naudojant aritmetinį operatorių, neteisingų rezultatų vertinimas (Elektroninė prekyba)

Klausimas	Priežastis	Sprendimas
5. Kuriuose užsakymuose yra daugiau nei 10 prekių?	NKS nepalaiko filtravimo pagal agregavimo funkcijos suskaičiuotą reikšmę.	Sukurti naują klausimo tipą, panaudojant HAVING operatorių.
6. Kurių sąskaitų data yra tokia pati kaip atlikto mokėjimo data?	NKS neturi galimybių apdoroti palyginimo klausimų, kuriuose nėra nurodyta tikslios reikšmės.	Sukurti naują klausimo tipą, kuris leistų palyginti dviejų duomenų savybių reikšmes.
9. Kokie pardavimai atlikti 2022 metais?	NKS neturi galimybių apdoroti palyginimo klausimų, kuriuose nėra nurodytas palyginimo operatorius.	Patikslinti klausimą.
10. Kokie mokėjimai buvo atlikti nuo 2022-04-01 iki 2022-04-31?	NKS neturi galimybių apdoroti klausimų, kuriuose turi būti vykdomas palyginimas su daugiau nei viena reikšme.	Sukurti naują klausimo tipą, kuris leistų ieškoti duomenų pagal pateiktą reikšmių intervalą.

6. Klausimas, kurio rolė apribota konkrečiu egzemplioriumi (6 klausimo tipas)

23 lentelė. Klausimų, kurių rolė apribota konkrečiu egzemplioriumi, paieška (Elektroninė prekyba)

Nr.	Klausimas	Atsakymas gautas	Atsakymas teisingas
1.	Kokius užsakymus pateikė Jonas?	+	+
2.	Kokius užsakymus atliko Rolandas?	+	+
3.	Kokius užsakymus sudaro medienos prekė Ažuolas?	+	-
4.	Kas apmokėjo faktūrą INV0001?	-	-
5.	Kokia gaminio P0001 rūšis?	+	+
6.	Kokie kroviniai buvo pristatyti į Kauną?	+	-
7.	Koks medienos P0002 sandėlio likutis?	+	-
8.	Kiek užsakymų atsiuntė Jonas?	+	-
9.	Kokius pristatymus įvykdė Martynas?	+	+
10.	Kokios medienos rūšis Uosis?	+	+
Iš viso:		9	6

Vykdamas penktojo klausimo tipo klausimų paiešką, iš dešimt klausimų atsakymas buvo gautas į devynis pateiktus klausimus, tačiau tik šešių klausimų rezultatai buvo teisingi. Žemiau pateikiamas neteisingų rezultatų vertinimas:

24 lentelė. Klausimų, kurių rolė apribota konkrečiu egzemplioriumi, neteisingų rezultatų vertinimas (Elektroninė prekyba)

Klausimas	Priežastis	Sprendimas
4. Kas apmokėjo sąskaitą INV0001?	Ontologijoje kliento ir užsakymo prekės klasės neturi jas tarpusavyje siejančio ryšio, t. y. objektinės savybės.	Papildyti ontologijos schemą nauja objektine savybe.
6. Kokie kroviniai buvo pristatyti į Kauną?	Morfologinio analizatoriaus trūkumas, kai morfologiškai išnagrinėtas žodis neatitinka egzemplioriaus reikšmės.	Morfologinio generatoriaus panaudojimas egzemplioriams.
7. Koks medienos P0002 sandėlio likutis?	NKS negali apdoroti šio klausimo taip, kad rezultate būtų pateikiamas duomenų savybių išvedimas.	Sukurti naują klausimo tipą, kuriame būtų pakoreguojamas rezultatų išvedimas trūkstama duomenų savybe.
8. Kiek užsakymų atsiuntė Jonas?	NKS negali apdoroti šio klausimo taip, kad jame būtų panaudotas agregavimo operatorius ir egzempliorius.	Sukurti naują klausimo tipą, kuriame būtų panaudojamas agregavimo operatorius ir egzempliorius.

5.2.2.2. Universiteto veiklos ontologija

Universiteto veiklos dalykinės srities eksperimentiniam tyrimui buvo sudarytas pavyzdinių klausimų rinkinys kiekviena klausimo tipui. Žemiau pateikiami kiekvieno klausimo vykdomos paieškos rezultatai. Taip pat pateikiami klausimai, į kuriuos sistema nesugebėjo atsakyti teisingai, ir to priežastys (žr. 25 lentelė – 36 lentelė).

1. Klausimas remiantis viena role be apribojimų (1 klausimo tipas)

25 lentelė. Klausimų, remiantis viena role be apribojimų, paieška (Universiteto veikla)

Nr.	Klausimas	Atsakymas gautas	Atsakymas teisingas
1.	Kokie yra padalinių pavadinimai?	–	–
2.	Kokių yra modulių?	+	+
3.	Koks yra kiekvieno modulio įvertinimas?	+	+
4.	Kokių yra studijų programų?	+	+
5.	Kiek kiekvienas studentas turi atsiskaitytų ir likusių kreditų?	+	+
6.	Koks yra studentų pažangumas?	+	–
7.	Kokios galimos atsiskaitymo formos?	+	+
8.	Kokie yra dėstytojų įvertinimai?	+	–

9.	Kokie studentų kodai?	+	+
10.	Kokios paskaitų datos?	+	+
Iš viso:		9	7

Vykdamas pirmojo klausimo tipo klausimų paiešką iš, dešimt klausimų atsakymas buvo gautas į devynis pateiktus klausimus, tačiau tik septynių klausimų rezultatai buvo teisingi. Žemiau pateikiamas neteisingų rezultatų vertinimas:

26 lentelė. Klausimų, remiantis viena role be apribojimų, neteisingų rezultatų vertinimas (Universiteto veikla)

Klausimas	Priežastis	Sprendimas
<i>1. Kokie yra padalinių pavadinimai?</i>	Klausime naudojami žodžiai nebuvo atpažinti ontologijoje.	Papildyti ontologiją naujomis sąvokomis (sinonimais).
<i>6. Koks yra studentų pažangumas?</i>	Ontologijoje studento klasei nėra sukurta duomenų savybė pažangumas.	Papildyti ontologijos schemą nauja duomenų savybe.
<i>8. Kokie yra dėstytojų įvertinimai?</i>	Ontologijoje dėstytojo klasei nėra sukurta duomenų savybė įvertinimas.	Papildyti ontologijos schemą nauja duomenų savybe.

2. Klausimas remiantis dviem rolėmis be apribojimų (2 klausimo tipas)

27 lentelė. Klausimų, remiantis dviem rolėmis be apribojimų, paieška (Universiteto veikla)

Nr.	Klausimas	Atsakymas gautas	Atsakymas teisingas
1.	Kokiuose fakultetuose mokosi studentai?	+	+
2.	Kokie dėstytojai dirba universitete?	+	+
3.	Kokie atsiskaitymai sudaro modulius?	+	+
4.	Kokios paskaitos vedamos universitete?	+	+
5.	Kokie dėstytojai organizuoja modulius?	+	+
6.	Kokie dėstytojai vertina atsiskaitymus?	+	+
7.	Kokie dėstytojai veda paskaitas?	+	+
8.	Kokiais semestrais dirba dėstytojai?	+	–
9.	Kokie studentai vertina modulius?	+	+
10.	Kokie moduliai turi atsiskaitymus?	+	+
Iš viso:		10	9

Vykdam antrojo klausimo tipo klausimų paiešką, iš dešimt klausimų atsakymas buvo gautas į visus pateiktus klausimus, tačiau tik devynių klausimų rezultatai buvo teisingi. Žemiau pateikiamas neteisingų rezultatų vertinimas:

28 lentelė. Klausimų, remiantis dviem rolėm be apribojimų, neteisingų rezultatų vertinimas (Universiteto veikla)

Klausimas	Priežastis	Sprendimas
8. Kokiais semestrais dirba dėstytojai?	Ontologijoje semestro ir dėstytojo klasės neturi jas tarpusavyje siejančio ryšio, t. y. objektinės savybės.	Papildyti ontologijos schemą nauja objektine savybe.

3. Klausimas naudojant agregavimo operatorių remiantis viena role (3 klausimo tipas)

29 lentelė. Klausimų, naudojant agregavimo operatorių remiantis viena role, paieška (Universiteto veikla)

Nr.	Klausimas	Atsakymas gautas	Atsakymas teisingas
1.	Koks vidutinis paskaitų lankomumas?	+	+
2.	Koks studentų atsiskaitymų vidurkis?	+	+
3.	Kiek universitete yra fakultetų?	+	+
4.	Koks vidutinis modulių įvertinimas?	+	+
5.	Koks mažiausias atsiskaitymo pažymys?	+	+
6.	Koks vidutinis pamokos įvertis?	–	–
7.	Kurioje grupėje daugiausia mokinių?	+	–
8.	Koks vidutinis dėstytojo įvertinimas?	+	–
9.	Koks maksimalus kreditų skaičius semestru?	+	–
10.	Koks didžiausias studento vidurkis?	+	+
Iš viso:		9	6

Vykdam trečiojo klausimo tipo klausimų paiešką, iš dešimt klausimų atsakymas buvo gautas į devynis pateiktus klausimus, tačiau tik šešių klausimų rezultatai buvo teisingi. Žemiau pateikiamas neteisingų rezultatų vertinimas:

30 lentelė. Klausimų, naudojant agregavimo operatorių remiantis viena role, neteisingų rezultatų vertinimas (Universiteto veikla)

Klausimas	Priežastis	Sprendimas
6. Koks vidutinis pamokos įvertis?	Klausime naudojami žodžiai nebuvo atpažinti ontologijoje.	Papildyti ontologiją naujomis sąvokomis (sinonimais).

7. Kurioje grupėje daugiausia mokinių?	NKS negali apdoroti šio klausimo taip, kad rezultate būtų pateikiamas duomenų savybių išvedimas.	Sukurti naują klausimo tipą, kuriame būtų pakoreguojamas rezultatų išvedimas trūkstama duomenų savybe.
8. Raskite vidutinį dėstytojo įvertinimą?	Ontologijoje dėstytojo klasei nėra sukurta tokia duomenų savybė kaip įvertis.	Papildyti ontologijos schemą nauja duomenų savybe.
9. Koks maksimalus kreditų skaičius semestru?	NKS negali apdoroti šio klausimo taip, kad rezultate būtų pateikiamas duomenų savybių išvedimas.	Sukurti naują klausimo tipą, kuriame būtų pakoreguojamas rezultatų išvedimas trūkstama duomenų savybe.

4. Klausimas naudojant agregavimo operatorių remiantis dviem rolėm (4 klausimo tipas)

31 lentelė. Klausimų, naudojant agregavimo operatorių remiantis dviem rolėm, paieška (Universiteto veikla)

Nr.	Klausimas	Atsakymas gautas	Atsakymas teisingas
1.	Kiek modulių dėstoma semestro metu?	+	–
2.	Kiek studentų mokosi universitete?	+	+
3.	Kuriame fakultete mokosi daugiausiai studentų?	+	–
4.	Kiek iš viso vienas dėstytojas veda modulių?	+	–
5.	Kiek modulių turi studentas?	+	+
6.	Kiek atsiskaitymų priklauso moduliui?	+	–
7.	Kiek dėstytojų organizuoja modulį?	+	+
8.	Kiek dėstytojų dirba universitete?	+	+
9.	Kokios paskaitos vedamos vėliausiai?	+	+
10.	Kiek studentų mokosi kiekvienoje grupėje?	+	–
Iš viso:		10	5

Vykdamas ketvirtojo klausimo tipo klausimų paiešką, iš dešimt klausimų atsakymas buvo gautas į visus pateiktus klausimus, tačiau tik keturių klausimų rezultatai buvo teisingi. Žemiau pateikiamas neteisingų rezultatų vertinimas:

32 lentelė. Klausimų, naudojant agregavimo operatorių remiantis dviem rolėm, neteisingų rezultatų vertinimas (Universiteto veikla)

Klausimas	Priežastis	Sprendimas
1. Kiek modulių dėstoma semestro metu?	NKS nepalaiko filtravimo pagal agregavimo funkcijos suskaičiuotą reikšmę.	Sukurti naują klausimo tipą panaudojant vidinę užklausą.
3. Kuriame fakultete mokosi daugiausiai studentų?	NKS nepalaiko filtravimo pagal agregavimo funkcijos suskaičiuotą reikšmę.	Sukurti naują klausimo tipą panaudojant vidinę užklausą.
4. Kiek iš viso vienas dėstytojas veda modulių?	NKS nepalaiko filtravimo pagal agregavimo funkcijos suskaičiuotą reikšmę.	Sukurti naują klausimo tipą panaudojant vidinę užklausą.
6. Kiek atsiskaitymų priklauso moduliui?	NKS nepalaiko filtravimo pagal agregavimo funkcijos suskaičiuotą reikšmę.	Sukurti naują klausimo tipą panaudojant vidinę užklausą.
10. Kiek studentų mokosi kiekvienoje grupėje?	NKS nepalaiko filtravimo pagal agregavimo funkcijos suskaičiuotą reikšmę.	Sukurti naują klausimo tipą panaudojant vidinę užklausą.

5. Klausimas naudojant aritmetinį operatorių (5 klausimo tipas)

33 lentelė. Klausimų, naudojant aritmetinį operatorių, paieška (Universiteto veikla)

Nr.	Klausimas	Atsakymas gautas	Atsakymas teisingas
1.	Kokie studentai gavo pažymį mažesnę už 5?	+	+
2.	Kokie studentai turi vidurkį didesnę už 8?	+	+
3.	Kokie studentai turi skolų daugiau nei 1?	+	-
4.	Kokie moduliai turi lankomumą mažesnę už 50(%)?	+	+
5.	Kokie moduliai turi studentų daugiau nei 100?	+	+
6.	Kokius modulius dėsto daugiau nei 2 dėstytojai?	+	+
7.	Kokie moduliai turi kreditų daugiau nei 4?	+	+
8.	Kokie moduliai turi sukūrimo metus didesnius už 2020?	+	-
9.	Kokie moduliai turi atsiliepimą mažesnę už 7?	+	+
10.	Kiek studentų mokosi informatikos fakultete?	+	-
Iš viso:		10	7

Vykdamy penktojo klausimo tipo klausimų paiešką, iš dešimt klausimų atsakymas buvo gautas į visus pateiktus klausimus, tačiau tik septynių klausimų rezultatai buvo teisingi. Žemiau pateikiamas neteisingų rezultatų vertinimas:

34 lentelė. Klausimų, naudojant aritmetinį operatorių, neteisingų rezultatų vertinimas (Universiteto veikla)

Klausimas	Priežastis	Sprendimas
3. Kokie studentai turi skolų daugiau nei 1?	Ontologijoje darbuotojo klasei nėra sukurta tokia duomenų savybė kaip leistinas svoris.	Papildyti ontologijos schemą nauja duomenų savybe.
8. Kokie moduliai turi sukūrimo metus didesnius už 2020?	NKS nepalaiko nepilno datos formato vykdant užklausas panaudojant aritmetinį operatorių.	Sukurti naują klausimo tipą, kuris naudotų datos reikšmių atpažinimą.
10. Kokiame fakultete mokosi daugiau nei 300 studentų?	NKS nepalaiko filtravimo pagal agregavimo funkcijos suskaičiuotą reikšmę.	Sukurti naują klausimo tipą panaudojant vidinę užklausą.

6. Klausimas, kurio rolė apribota konkrečiu egzemplioriumi (6 klausimo tipas)

35 lentelė. Klausimų, kurių rolė apribota konkrečiu egzemplioriumi, paieška (Universiteto veikla)

Nr.	Klausimas	Atsakymas gautas	Atsakymas teisingas
1.	Kokius įvertinimus gavo Jonas?	+	+
2.	Kokie dėstytojai veda Matematika?	+	+
3.	Koks modulis turi kodą T170B140?	+	+
4.	Kokie moduliai yra dėstomi semestre Pavasaris?	+	+
5.	Kokie dėstytojai įvertino Joną?	+	+
6.	Kokius modulius lanko Jonas?	+	+
7.	Kokie moduliai sudaro Informatikos fakultetą?	+	+
8.	Kiek kreditų turi modulis T170B140?	+	+
9.	Kurių atsiskaitymų forma yra Testas?	+	+
10.	Kurie moduliai turi 6 kreditus?	+	+
Iš viso:		10	9

Vykdamy šeštojo klausimo tipo klausimų paiešką, iš dešimt klausimų atsakymas buvo gautas į visus pateiktus klausimus, tačiau tik devynių klausimų rezultatai buvo teisingi. Žemiau pateikiamas neteisingų rezultatų vertinimas:

36 lentelė. Klausimų, kurių rolė apribota konkrečiu egzemplioriumi, neteisingų rezultatų vertinimas (Universiteto veikla)

Klausimas	Priežastis	Sprendimas
5. Kokie dėstytojai įvertino Joną?	Morfologinio analizatoriaus trūkumas, kai morfologiškai išnagrinėtas žodis neatitinka egzemplioriaus reikšmės.	Morfologinio generatoriaus panaudojimas.

5.3. Eksperimento rezultatų analizė

Atliktas eksperimentas skirtas įvertinti sukurtos natūraliosios kalbos sąsajos ontologijoms pateiktų rezultatų kokybę kiekvienoje dalykinėje srityje. Gautų rezultatų kokybę gali būti matuojama tikslumu (angl. *precision*), kuris leidžia įvertinti, kiek buvo pateikta teisingų atsakymų ir išsamumu (angl. *recall*), kuris leidžia įvertinti, kiek klausimų sistema interpretavo [21].

Tikslumas (PQ) apibrėžiamas teisingai atsakytų klausimų (CQ) ir klausimų, kuriems buvo gautas atsakymas (AQ) santykiu:

$$PQ = \frac{CQ}{AQ}$$

Išsamumas (RQ) apibrėžiamas teisingai atsakytų klausimų (CQ) ir visų klausimų (TQ) skaičiaus santykiu:

$$RQ = \frac{CQ}{TQ}$$

F-matas (FQ) – bendras klausimų kokybės įvertinimas:

$$FQ = 2 \times \frac{PQ \times RQ}{PQ + RQ}$$

Žemiau pateikiamas rezultatų vertinimas remiantis atliktos paieškos rezultatais kiekvienoje dalykinėje srityje (žr. 37 lentelė).

37 lentelė. Eksperimento rezultatai

Dalykinė sritis	Visi klausimai (TQ)	Atsakymas gautas (AQ)	Atsakymas teisingas (CQ)	Tikslumas (PQ)	Išsamumas (RQ)	Bendras įvertinimas (FQ)
Elektroninė prekyba	60	54	34	0,63	0,56	0,59
Universiteto veikla	60	58	43	0,74	0,72	0,73

5.4. Eksperimento išvados

Atlikto eksperimentinio tyrimo gautų rezultatų kokybės vertinimas parodo, kad sukurtas metodas sudaro sąlygas vykdyti semantinę paiešką lietuvių kalba. Remiantis bendru kokybės įvertinimu galima teigti, kad realizuoto sprendimo naudojimui kaip realaus produkto turėtų būti atliekami tam tikri patobulinimai. Pagrindinė priežastis, dėl kurios nepavyko pasiekti aukštesnio kokybės įvertinimo – klausimo tipų ribotumas. Dėl šios priežasties metodo patobulinimui reikalingas galimų klausimų tipų aibės išplėtimas, suteikiant galimybę suprasti skirtingesnius naudotojo klausimus.

5.5. Natūraliosios kalbos sąsajos ontologijoms veikimo ir savybių analizė

Tiriamąjį projekto metu formaliai aprašytas ir sukurtas metodas turi tokius privalumus:

1. natūraliosios kalbos sąsaja leidžia vykdyti semantinę paiešką lietuvių kalba;
2. naudotojo sąsajai nereikia papildomai ruošti žodyno, reikalingas tik ontologijos sukūrimas, kuris nereikalauja daug rankinio darbo.

Analizuojant gautus eksperimentinio tyrimo rezultatus buvo nustatyta ir daugiau trūkumų, kuriems išspręsti reikia tokių patobulinimų:

1. **kontekstinis meniu** – siekiant naudotojui palengvinti klausimo formulavimą ir padėti įvesti teisingą klausimą, turėtų būti pateikiamas kontekstinis meniu, kuris sudaromas naudojant morfologinį generatorių;
2. **rezultatų pasirinkimas** – siekiant, kad naudotojui pateikti rezultatai būtų aiškesni, išvedimo metu naudotojas turėtų turėti galimybę patikslinti, kokie stulpeliai turėtų būti atvaizduojami rezultatų sąrašė;
3. **rezultatų rikiavimas** – siekiant, kad naudotojas galėtų patogiai peržiūrėti gautus rezultatus, jų išvedimo lange turėtų būti galimybė rikiuoti duomenis pagal pasirinktus kriterijus;
4. **grafinė naudotojo sąsaja** – sukurta paprasta ir lengvai suprantama grafinė naudotojo sąsaja;
5. **klausimo tipų išplėtimas** – siekiant, kad klausimų įvairovė būtų didesnė turi būti papildomi arba sukuriami nauji klausimų tipai.

5.6. Natūraliosios kalbos sąsajos ontologijoms taikymo rekomendacijos

Sukurtas metodas, leidžiantis vykdyti klausimų paiešką lietuvių kalba ir transformuoti juos į užklausas ontologijoje. Sprendimas gali būti pritaikomas tam tikrai dalykinei sričiai, pavyzdžiui, elektroninės prekybos įmonei, prekiaujančiai elektroninėmis prekėmis internetu. Įmonė savo užsakymų valdymo sistemoje galėtų integruoti sukurtą komponentą, kuris leistų vykdyti paiešką norint peržiūrėti informaciją apie užsakymus, klientus, prekes, sąskaitas ir kt. Naudota semantinė saugykla *GraphDB* turi galimybę dirbti ne tik su ontologijomis, parengtomis *Protégé* įrankiu, bet ir leidžia sukurti virtualias saugyklas, kai ontologijos egzemplioriai sukuriami skaitant duomenis iš reliacinės duomenų bazės. Taip pat duomenis gali integruoti iš failų, tarkime *CSV* ar *JSON* failų. Tai suteikia dar daugiau galimybių išnaudoti ontologijas duomenų paieškai ir analizei organizacijos veikloje.

Šiame darbe sukurtas prototipas veikia kaip konsolinė programa. Norint pritaikyti sprendimą praktikoje, reikėtų sukurti grafinę naudotojo sąsają, kuri, priklausomai nuo organizacijos poreikių, galėtų veikti tiek kaip taikomoji programa, tiek kaip interneto puslapis.

Išvados

1. Semantinė paieška, palyginti su įprasta, raktiniais žodžiais grindžiama paieška, leidžia gražinti tikslesnius rezultatus, nes ieškoma ne tik pagal sintaksinį atitikimą, bet ir pagal prasmę ontologijoje, kurioje aprašomi resursai ir jų ryšiai atitinka realaus pasaulio objektus.
2. Išanalizavus esamus semantinės paieško vykdymo sprendimus buvo nustatyta, kad daugiausiai galimybių turi *GraphDB* semantinė saugykla, kuri suteikia galimybes vykdyti greitesnę paiešką, nes išvedimai yra vykdomi duomenų įrašymo metu. Taip pat *GraphDB* kartu gali būti naudojama ir kaip saugykla.
3. Viena iš semantinės paieškos savybių yra galimybė suprasti natūraliąja kalba pateiktus klausimus ir į juos atsakyti. Naudotojams tai yra vienas iš patogiausių būdų vykdyti paiešką. Tačiau kuriant natūraliosios sąsajos sistemas yra susiduriama su įvairiomis problemomis, tokiomis kaip dviprasmybės, klausime naudojamų terminų ir ontologijos schemos neatitikimas ir kt.
4. Viena iš svarbių natūraliosios kalbos sąsajų dalių yra žodynas, kuris naudojamas formuluojant klausimus ir juos transformuojant į užklausas. Jame taip pat gali būti aprašomi sinonimai, kurie padeda formuluoti klausimus lanksčiau. Žodynas tokiose sistemose gali būti sudaromas automatiškai arba rankiniu būdu. Atlikus esamų natūraliosios kalbos sąsajų analizę, buvo nustatyta, kad dauguma tokių sistemų pritaikytos anglų kalbai. Tik vienintelė *SBVR* pagrįstą NKS yra pritaikoma lietuvių kalbai, tačiau pagrindinis trūkumas yra tai, kad jos paruošimui papildomai reikia aprašyti žodyną, kuris atitiktų ontologijos schemoje aprašytus terminus.
5. Kuriamo sprendimo pagrindinis reikalavimas – vykdyti semantinę paiešką ontologijoje pateikiant klausimus lietuvių kalba. Nuspręsta, kad sąsajos žodynas turėtų būti sudaromas automatiškai pagal ontologijos schemą.
6. Remiantis analize ir asmenine patirtimi buvo nustatyti šeši klausimų tipai, į kuriuos sistema turės atsakyti. Klausimams atpažinti, analizuoti ir transformuoti į *SPARQL* užklausas buvo aprašyti algoritmai. Siekiant, kad sistema lengviau interpretuotų klausimus buvo nuspręsta naudoti patikslinantį dialogą. Pagal nustatytus suprojektuoto sprendimo reikalavimus ir jų specifikaciją buvo sukurtas sprendimas, sudarantis sąlygas įvykdyti semantinę paiešką lietuvių kalba.
7. Pagal sukurtą projektą buvo sukurtas metodo prototipas. Prototipas realizuotas kaip konsolinė programa, naudojant *Java* programavimo kalbą. Morfologinei analizei panaudota *Hunspell* biblioteka ir lietuvių kalbos morfologinis žodynas.
8. Siekiant įvertinti sukurtą metodą buvo atliktas eksperimentinis tyrimas, kurio metu buvo siekiama iširti, ar korektiškai sistema atsako į klausimus. Buvo nustatyta, kad pasiūlytas metodas leidžia vykdyti semantinę paiešką, gauti tikslumo ir išsamumo įvertinimai svyruoja nuo 0,6 iki 0,7. Norint metodą pritaikyti praktikoje, šie parametrai turėtų būti padidinami. Viena iš pagrindinių krypčių – klausimų analizės algoritmų tobulinimas ir palaikomų klausimų tipų aibės išplėtimas.
9. Pagrindinis sukurto metodo privalumas – galimybė vykdyti semantinę paiešką pateikiant klausimus lietuvių kalba ir tai, kad pritaikymas dalykinei sričiai nereikalauja žodyno parengimo rankiniu būdu. Metodas gali būti pritaikomas organizacijų informacinėse sistemose siekiant palengvinti duomenų paiešką net ir tais atvejais, kai duomenys saugomi reliacinėse duomenų bazėse ar failuose.

Literatūros sąrašas

- [1] Androustopoulos, Ion, Graeme D. Ritchie, and Peter Thanisch. "Natural language interfaces to databases—an introduction." *Natural language engineering* 1.1 (1995): 29-81.
- [2] Benjamin, P., et al. "IDEF5 Method Report. College Station: Knowledge Based Systems." (1994).
- [3] Berners-Lee, Tim, James Hendler, and Ora Lassila. "The semantic web." *Scientific american* 284.5 (2001): 34-43.
- [4] Broekstra, Jeen, Arjohn Kampman, and Frank van Harmelen. "Sesame: A generic architecture for storing and querying rdf and rdf schema." *International semantic web conference*. Springer, Berlin, Heidelberg, 2002.
- [5] Butkienė Rita. Kauno technologijos universitetas. Modulio „Duomenų semantiniai modeliai“ paskaitų skaidrės.
- [6] Calero, Coral, Francisco Ruiz, and Mario Piattini, eds. *Ontologies for software engineering and software technology*. Springer Science & Business Media, 2006.
- [7] Cimiano, Philipp, et al. "Orakel: A portable natural language interface to knowledge bases." (2007).
- [8] Damljanić, Danica, et al. "Improving habitability of natural language interfaces for querying ontologies with feedback and clarification dialogues." *Journal of web semantics* 19 (2013): 1-21.
- [9] Damljanić, Danica, Valentin Tablan, and Kalina Bontcheva. "A text-based query interface to owl ontologies." *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. 2008.
- [10] De Keyser, Piet. *Indexing: from thesauri to the semantic web*. Elsevier, 2012.
- [11] Dentler, Kathrin, et al. "Comparison of reasoners for large ontologies in the OWL 2 EL profile." *Semantic Web 2.2* (2011): 71-87.
- [12] Diefenbach, Dennis, et al. "Core techniques of question answering systems over knowledge bases: a survey." *Knowledge and Information systems* 55.3 (2018): 529-569.
- [13] Guarino, Nicola, Daniel Oberle, and Steffen Staab. "What is an ontology?." *Handbook on ontologies*. Springer, Berlin, Heidelberg, 2009. 1-17.
- [14] Güting, Ralf Hartmut. "GraphDB: Modeling and querying graphs in databases." *VLDB*. Vol. 94. 1994.
- [15] Goutte, Cyril, and Eric Gaussier. "A probabilistic interpretation of precision, recall and F-score, with implication for evaluation." *European conference on information retrieval*. Springer, Berlin, Heidelberg, 2005.
- [16] Hitzler, Pascal, et al. "OWL 2 web ontology language primer." *W3C recommendation* 27.1 (2009): 123.
- [17] Höffner, Konrad, et al. "Survey on challenges of question answering in the semantic web." *Semantic Web* 8.6 (2017): 895-920.
- [18] Höffner, Konrad, et al. "Overcoming challenges of semantic question answering in the semantic web." *Semantic Web Journal* (2016): 1-21.
- [19] Kaufmann, Esther, and Abraham Bernstein. "Evaluating the usability of natural language query languages and interfaces to Semantic Web knowledge bases." *Journal of Web Semantics* 8.4 (2010): 377-393.
- [20] Kaufmann, Esther. "Talking to the semantic web—query interfaces to ontologies for the casual user." *International Semantic Web Conference*. Springer, Berlin, Heidelberg, 2006.

- [21] Lee, Tae Jun, et al. "Precision and recall for range-based anomaly detection." arXiv preprint arXiv:1801.03175 (2018).
- [22] Matentzoglou, Nicolas, et al. "A Survey of Current, Stand-alone OWL Reasoners." ORE. 2015.
- [23] Nihalani, Neelu, Sanjay Silakari, and Mahesh Motwani. "Natural language interface for database: a brief review." International Journal of Computer Science Issues (IJCSI) 8.2 (2011): 600.
- [24] Seaborne, Andy, et al. "SPARQL/Update: A language for updating RDF graphs." W3c member submission 15 (2008).
- [25] Shearer, Rob, Boris Motik, and Ian Horrocks. "Hermit: A Highly-Efficient OWL Reasoner." Owled. Vol. 432. 2008.
- [26] Šukys, Algirdas, Lina Nemuraitė, and Rita Butkienė. "SBVR based natural language interface to ontologies." Information Technology and Control 46.1 (2017): 118-137.
- [27] Wang, Wenlu, et al. "A natural language interface for database: Achieving transfer-learnability using adversarial method for question understanding." 2020 IEEE 36th International Conference on Data Engineering (ICDE). IEEE, 2020.
- [28] World Wide Web Consortium. "SPARQL 1.1 overview." (2013).