

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

PROGRAMŲ INŽINERIJOS KATEDRA

Tomas Bandza

**Duomenų bazėje realizuoto semantinio paslaugų
registro paieškos algoritmo tyrimas**

Magistro darbas

Darbo vadovė
prof. Lina Nemuraitė

Kaunas, 2010

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

PROGRAMŲ INŽINERIJOS KATEDRA

Tomas Bandza

**Duomenų bazėje realizuoto semantinio paslaugų
registro paieškos algoritmo tyrimas**

Magistro darbas

Recenzentas

doc. dr. Vytautas Pilkauskas
2010-05-31

Vadovė

prof. Lina Nemuraitė
2010-05-31

Atliko

IFM-4/2 gr. stud.
Tomas Bandza
2010-05-31

Kaunas, 2010

Summary	5
1. Įvadas	6
2. Semantinio registro modelio ir paieškos algoritmų analizė	8
2.1. Darbo tikslas ir aktualumas	8
2.2. Standartinio registro ir internetinių paslaugų taikymas ir aplinka	9
2.2.1. Internetinių paslaugų architektūra	9
2.2.2. Interneto paslaugų registruose naudojimo pavyzdys	10
2.2.3. Procesų ir aplinkos analizė	11
2.3. Paslaugų aprašo WSDL struktūros analizė	13
2.4. Standartinio (UDDI) registro analizė	15
2.5. Semantinių technologijų analizė	19
2.6. Semantinio registro kūrimo būdai pritaikant semantines technologijas	21
2.6.1. UDDI registro išplėtimas semantika	21
2.6.2. WSDL išplėtimas semantika	23
2.7. Semantinės paieškos algoritmų analizė	23
2.7.1. Semantinės paieškos formavimas	23
2.7.2. Semantinės paieškos algoritmo veikimo principas	24
2.7.3. Standartinis semantinis panašumo metodas	25
2.7.4. Semantinio panašumo išreiškimas reikšme	26
2.7.5. Kiti semantinio panašumo metodai	27
2.7.6. Semantinių panašumų metodų palyginimas	30
2.8. Standartinio registro modelio realizacijos duomenų bazėje analizė	30
2.9. Esamų semantinių registro modelių analizė	33
2.9.1. OWL-S	33
2.9.2. SAWSDL	34
2.9.3. Semantinio registro modelių palyginimas	35
2.10. Analizės išvados	36
3. Semantinio registro reikalavimų specifikacija ir analizė	38
3.1. Siekiamo registro modelio ir jo algoritmo tyrimo sprendimų apibrėžimas	38
3.2. Reikalavimų specifikacija	39
3.2.1. Veiklos kontekstas	39
3.2.2. Panaudojimo atvejai	40
3.2.3. Funkciniai reikalavimai	45
3.3. Nefunkciniai reikalavimai	49
3.3.1. Reikalavimai saugumui	49
3.3.2. Reikalavimai sąsajai	50
3.3.3. Reikalavimai panaudojamumui	51
3.3.4. Reikalavimai vykdymo charakteristikoms	52
3.4. Semantinio registro duomenų modelis	52
3.5. Reikalavimų analizės išvados	54
4. Semantinio registro programų architektūra	55
4.1. Bendra komponentų išsidėstymo schema	55
4.2. Projekto loginė architektūra	56
4.2.1. Registrų administravimo programos loginė architektūra	56
4.2.2. Paieškos programos loginė architektūra	60
4.3. Programų dinaminiai procesų vaizdai	62
4.3.1. Paieškos programos dinaminiai vaizdai	63
4.3.2. Internetinių paslaugų paieškos procesai	65
4.3.3. Registrų administravimo programos dinaminiai vaizdai	67

4.3.4.	Registru administravimo programos veiklos diagramos.....	71
4.4.	Registro duomenų bazė.....	72
4.5.	Semantinės paieškos algoritmas.....	76
5.	Tyrimui realizuotos programinės priemonės.....	80
5.1.	Registru administravimo programa.....	80
5.1.1.	Naujo registro sukūrimas.....	81
5.1.2.	Interneto paslaugų įtraukimas į registrą.....	82
5.1.3.	Semantikos priskyrimas.....	83
5.2.	Registru paieškos programa.....	85
5.2.1.	Paieškos programos sąsaja.....	85
5.2.2.	Semantinė paieška.....	85
5.3.	Testavimas.....	87
6.	Eksperimentinis semantinės paieškos algoritmo tyrimas.....	92
6.1.	Eksperimentinio tyrimo tikslas ir uždaviniai.....	92
6.2.	Semantinės paieškos algoritmo greیتaveikos tyrimas.....	92
6.3.	Eksperimento išvados.....	95
7.	Išvados.....	95
8.	Literatūra.....	96
9.	Terminų ir santrumpų žodynelis.....	98
10.	Priedai.....	99
10.1.	Testavime ir eksperimente naudotos viešbučių ontologijos schema.....	99
10.2.	Eksperimento duomenys.....	99

Summary

Investigation of discovery algorithm for semantic registry implemented in database

Efficiently finding Web services on the Web is a challenging issue in service-oriented computing. Currently, UDDI is a standard for publishing and discovery of Web services, and UDDI registries also provide keyword searches for Web services. However, the search functionality is very simple and fails to account for relationships between Web services. Firstly, users are overwhelmed by the huge number of irrelevant returned services. Due to the lack of semantic descriptions of the Web services, the results returned by registries are effectively inadequate. Semantic web services discovery based on what model of semantic registry is used and architecture of search algorithm.

Semantic registry models according articles are based on UDDI registry XML elements mapping with the enhanced semantics such OWL or RDF ontology. Ontology describes the information domain about web services operation type, inputs and outputs. The search algorithm tries to find a match for user query inputs or outputs. Executing matchmaking method results the some sort similarity value by which can be decided what web service operation are more similar to users query parameters.

To perform the investigation was needed to create a simple registry model enhanced with semantics mappings with OWL ontology and realize software that works with this model. To investigate search algorithm's performance based on Greedy matchmaking method. The registry model was implemented in MySQL database.

Keywords: Web service, Semantic registry, matchmaking, similarity, UDDI, OWL

1. ĮVADAS

Interneto paslaugos (angl. Web services) yra pažangi ir besivystanti programinės įrangos koncepcija, kai programinė įranga teikiama kaip paslauga. Interneto ryšiu teikiamos specializuotos paslaugos (toliau vadinsime jas servisais) pasiekiamos per registrus, kurie gali būti prieinami viešai arba privačiame įmonių tinkle [5].

Išaugus paslaugų paklausai ir pasiūlai, atsiranda problema, kaip geriau atlikti paslaugų paiešką. Egzistuoja paprastos paieškos sistemos, kurios nelabai praktiškos servisų paieškai, tad kyla reikalingumas sukurti patogesnę ir kokybiškesnę servisų paieškos sistemą. Paslaugų registre galima aprašyti paslaugų teikėją pateikiant veiklos aprašymą, kontaktinius duomenis, užregistruoti paslaugas ir aprašyti jas. Registras saugo techninius paslaugų duomenis: iškvietimo sąsajas, paslaugų veikimo protokolus. Didžiausiais trūkumais yra tas, kad paslaugų paieška registruose vykdoma pagal raktinius žodžius. Tokiu atveju paieškos rezultatų skaičius yra didelis ir dažnai neatspindintis realaus paslaugos vykdomos operacijos pobūdžio. Vienas iš sprendimo būdų interneto paslaugų tikslesnei paieškai atlikti yra semantinių aprašų sukūrimas ir susiejimas su paslaugomis. Tuomet ieškant paslaugos užtenka nurodyti jos parametrų ar kategoriją. Paieškos mechanizmas atlieka ieškomų parametrų ir paslaugoje aprašytų parametrų panašumo lyginimą ir nusprendžia, ar paslaugoje aprašyti elementai panašūs į ieškomuosius. Tokiu atveju rezultatų skaičius sumažinamas ir paslaugos randamos tiksliau. Egzistuojantys semantiniai paslaugų aprašai OWL-S, WSMO, SAWSDL yra gana sudėtingi. Šiame darbe norėta parodyti, kad semantinį registrą galima sukurti paprasčiau ir saugoti paslaugų aprašus duomenų bazėse, kurios suteikia patogias ir patikimas duomenų saugojimo priemones.

Šio darbo **tyrimo sritis** yra semantinių registrų modeliai, technologijos ir paieškos metodai. **Tyrimo objektas** – paslaugų paieškos algoritmo semantiniame registre vykdymo procesas.

Šio magistrinio **darbo tikslas** – ištirti sukurto paieškos algoritmo veiksmingumą semantiniame registre, kuris yra realizuotas reliacinėje duomenų bazėje.

Darbo uždaviniai:

- Išanalizuoti semantinių registrų modelius ir architektūrą.
- Išanalizuoti semantinės paieškos algoritmus.
- Realizuoti semantinį registro modelį duomenų bazėje.
- Sukurti programinius įrankius darbui su semantinių registru.

- Atlikti semantinės paieškos algoritmo tyrimo eksperimentą ir įvertinti darbo rezultatus.

Veiksmingumu šiame darbe suprantamas algoritmo funkcionalumas (gebėjimas rasti paslaugas pagal jų parametrų semantiką) ir priimtinas paieškos vykdymo laikas.

Tyrimo metodika. Darbe buvo taikoma literatūros ir lyginamoji analizė, sprendimui konstruoti buvo taikomas evoliucinis kūrimo procesas, atliktas eksperimentinis tyrimas.

Siekiant sukurti semantinio registro modelį, buvo išanalizuoti [1]-[10], nagrinėjant semantinių technologijų specifiką [10] ir [12] šaltinius. Registro modelio sukūrimui ir jo realizacijos būdus naudotasi [2]-[4], [13] šaltiniais. Buvo nuspręsta sudaryti kuo paprastesnį modelį, atitinkantį semantinio registro koncepciją. Išanalizavus semantinės paieškos algoritmus šaltiniuose [15],[20]-[22] buvo sudarytas algoritmas, paremtas Greedy metodu [20]. Semantinis registras buvo realizuotas, ištestuotas, atliktas paieškos algoritmo tyrimas patvirtino jo veiksmingumą.

Darbo struktūra.

- 2 skyriuje semantinio registro modelio architektūros analizė. Apžvelgti įvairių modelių semantikos pritaikymo būdai, modelio kūrimo principai. Išnagrinėti semantinės paieškos principai, paieškos algoritmo projektavimo būdai, semantinio panašumo nustatymo metodai. Registro modelio realizacijos duomenų bazėje būdai.
- 3 skyriuje aprašomi reikalavimai, specifikacijos programinėms priemonėms skirtoms šio darbo tyrimui. Aprašytas projektuojamo registro duomenų modelis
- 4 skyriuje aprašomas eksperimento tyrimui skirtų priemonių architektūros. Pateiktos paieškos ir registro valdymo programų panaudojimo atvejų veiklos ir sekos diagramos. Aprašyta registro modelio struktūros realizacija duomenų bazėje.
- 5 skyriuje pateikiamos algoritmo tyrimui realizuotos programos veikimo pavyzdžiai
- 6 skyriuje aprašytas paieškos algoritmo sukurtame semantiniame registro modelyje eksperimentinis tyrimas
- 7 skyriuje suformuluotos darbo išvados

2. SEMANTINIO REGISTRO MODELIO IR PAIEŠKOS ALGORITMŲ ANALIZĖ

2.1. Darbo tikslas ir aktualumas

Darbo tikslas yra sukurti semantinį registro modelį, kuris leistų atlikti semantinę interneto paslaugų (angl. web services) paiešką (toliau patogumo dėlei tinklo paslaugas vadinsime servisais). Tam reikia išanalizuoti registro modelio kūrimo variantus, jų realizacijos būdus, semantikos pritaikymo būdus registrams, semantinės paieškos algoritmus.

Dabar dauguma informacinių paslaugų teikiama ne vien tiesiogiai per interneto puslapius. Projektuojami serveriai, kurioje yra talpinamos specialūs servisai, kurie netiesiogiai, pateikia norimą paslaugos informacija. Tai vadinamieji internetinių paslaugų (angl. web services) serveriai [6].

Siūlomos paslaugos – tai programinės įrangos moduliai, gali būti projektuojami, bet kokia programavimo kalba (paketais): C++, Java, .NET. Jie veikia serveryje ir esant tokių paslaugų užklausimui iš kliento pusės, jie aktyvuojasi ir siunčia informaciją klientui.

Esant dideliame servisų skaičiui, iškyla norimo serviso paieškos problemos. Dabar esamos servisų registravimo sistemos turi paprastas paslaugų aprašymo galimybes, kurios aprašomos paprastai, kaip ir naudojant interneto puslapių paieškos sistemose, kaip „Google“ ar „Yahoo“. Randamas puslapyje raktinis žodis (angl. keyword), jis įtraukiamas į paieškos duomenų bazę. Ieškant paieškos sistemoje puslapių, pagal įvestus raktinius žodžius, sistema randa atitinkamus žodžius, kurie figūravo puslapyje. Dažnai tokios paieškos sistemos, ne visada išveda tikrai norimos informacijos puslapį, arba išveda puslapį, kur reikšminiai žodžiai, būna skirtingose pastraipose, arba vienoje eilutėje, bet puslapio turinys skirtingas pagal ieškoma informaciją [7]. Tas pats aktualu, ir ieškant servisų paslaugų.

Kadangi, vis daugiau atsiranda servisų technologijomis integruojamų verslo įmonių sistemų. Jos diegiamos vis į daugiau serverių, o ir patys serveriai jungiami į dar didesnius informacinių servisų tinklus, tai reikalinga patogesnė ir kokybiškesnė servisų paieškos sistema.

Semantinis registras – tai sudėtinga sistema, kurios sudėtingumas priklauso nuo registro ypatybių: duomenų modelio struktūros, programavimo ir semantinių technologijų pritaikymo semantinės paieškos algoritmų charakteristikų. Todėl jam sukurti reikia atlikti išsamią analizę:

- Išanalizuoti registro ir internetinių paslaugų architektūrą ir taikymą
- Išanalizuoti pagrindinius principus, kaip kuriamas semantinis registras
- Apžvelgti esamų semantinių registrų modelius

- Išanalizuoti, kokie yra semantinio registro realizacijos būdai
- Nustatyti semantinio registro realizacijai naudojamą technologijas
- Apžvelgti semantinės paieškos algoritmus

2.2. Standartinio registro ir internetinių paslaugų taikymas ir aplinka

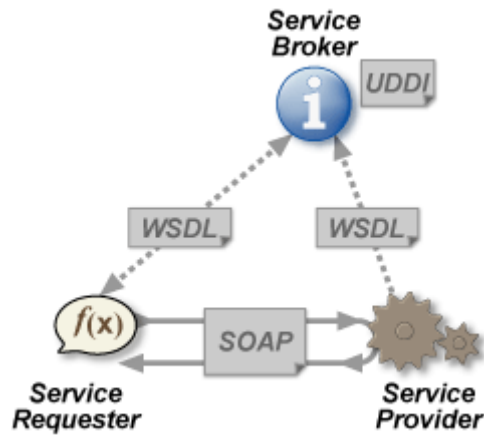
2.2.1. Internetinių paslaugų architektūra

Šio darbo tyrimas, grindžiamas internetinių paslaugų (toliau servais) paieška. Visu pirma reikia išanalizuoti aspektus, kaip veikia patys servais, kaip jie projektuojami. Apžvelgti esamas semantinės paieškos specifikas, ir aptarti panašių egzistuojančius projektus.

Internetinės paslaugos (angl. web services) – tai protokolų ir standartų rinkinys, naudojamas duomenų apsikeitimui tarp aplikacijų ir sistemų. Įvairiomis programavimo kalbomis parašytos programinės įrangos moduliai, veikiančios skirtingose operacinėse sistemose, naudojami servisų duomenų mainams kompiuteriniuose tinkluose [6].

Technologijas, naudojamas įgyvendinant servisų projektavimą, galima suskirstyti į keletą lygmenų. Tai perdavimo (angl. Transportation), pranešimų (angl. Messaging), apibrėžimo (angl. Definition), atradimo (angl. Discovery), kokybės (angl. Quality of Service) ir sujungimo lygmenys (angl. Service Aggregation). Žemiausiame, perdavimo lygmenyje, gali būti panaudoti keli protokolai, tačiau dažniausiai naudojamas HTTP bei jo atmainos. Kiti galimi perdavimo protokolai yra JMS, SMTP ir IIOP.

Paprastai tariant, servais tai dalis programinės įrangos, kuri sudaro sąsajos interfeisas (angl.interface) ir internetinės jungtys (angl. Binding) gali būt aprašytos XML kalba. Tai suteikia galimybę vienam servisui bendrauti su kitomis programinės įrangos moduliais naudojant XML pagrindu, siunčiamus sisteminius pranešimus per Interneto protokolus. Dažniausiai naudojamas HTTP protokolas. Paprastas serviso pavyzdys gali būti, užsakomų vietų rezervacija, bilieto pirkimas ar prekyba akcijų biržoje internetu ir t.t.[2].

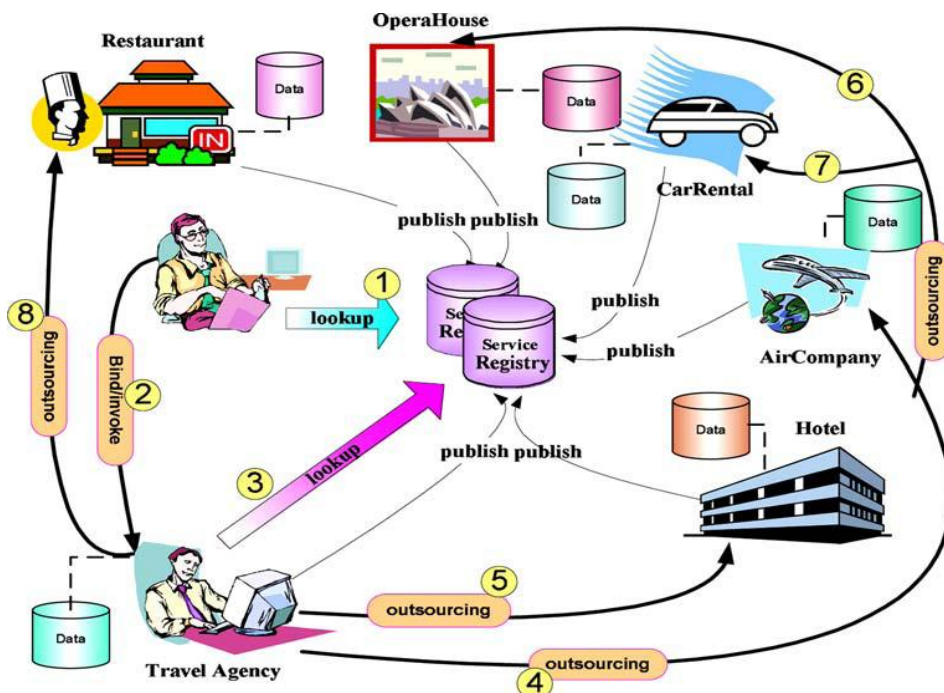


1 pav. Servisų architektūra [1]

Projektuojant interneto paslaugas, dažniausiai naudojamos technologijos: SOAP, WSDL, UDDI. Pateiktame paveikslėlyje (1 pav.), matome internetinių paslaugų veikimą. Klientas (Service Requester) kviečia paslaugą (servisą), jį radęs paslaugų aprašuose, kurie laikomi registrais serverio UDDI kataloge (Service Broker). Iškvietęs jį, automatiškai siunčiamas pranešimas, paslaugą teikiančiam serveriui (Service provider), kuris gražina rezultatus klientui.

2.2.2. Interneto paslaugų registruose naudojimo pavyzdys

Pateiktame paveikslėlyje (2 pav.) matome internetinių paslaugų naudojimo pavyzdį. Laikykime, kad esame kelionių organizavimo kompanija ("Travel agency"), kurie teikia kelionių planavimo ir organizavimo paslaugas: transporto parinkimas, vykstant į kitą šalį, viešbučių užsakymą ir kt.



2 pav. Paslaugų panaudojimo pavyzdys [2]

Sakykime, yra profesorius Jonas. Jis keliauja į tarptautinę konferenciją užsienyje. Jam reikalingos paslaugos: lėktuvo bilieto užsakymas, viešbučio registracija reikiamu laiku, ir papildomas laisvalaikio praleidimas – apsilankymas teatre, tai pat užsakant bilietus į spektaklį. Taigi profesoriui Jonui, reikia sudaryti jo norimus dalykus atitinkanti kelionės paslaugų paketą. Jonas kelionių kompanijos tinklapyje, išsirenka reikiamų paslaugų paketą. Jonas išsiunčia užklausimą internetu. Gavusi užklausimą kelionių kompanija, formuoja jo norimų paslaugų paketą. Formuodami jie gauna iš kitų kompanijų, verslo partnerių, duomenis apie atitinkamą paslaugą ir pasinaudodami teikiamais servisiais jie užsako viešbutį, lėktuvo, teatro bilietus.

Išorinės kompanijos savo teikiamas paslaugas buvo pateikusios serverio registruose, jog kelionių organizatorių kompanijos galėtų laisvai rasti reikiamą paslaugą, gauti duomenis, atlikti užsakymą. Užsakymas pagal servisų registrus, nukeliauja į kompanijų duomenų serverius. Atliekamos finansinės operacijos. Galiausiai paruošiamas paketas klientui, ir išsiunčiamas jam.

Magistrinio darbo sritis specializuojasi semantinės paieškos technologijos pritaikymo interneto paslaugų paieškai. Semantiškai veikianti paieška, leidžia pagreitinti paiešką ir padidinti ieškomų objektų radimo kokybę. To sekoje sumažinamas paieškos rezultatų skaičius iki minimumo, mažiau sugaištama laiko.

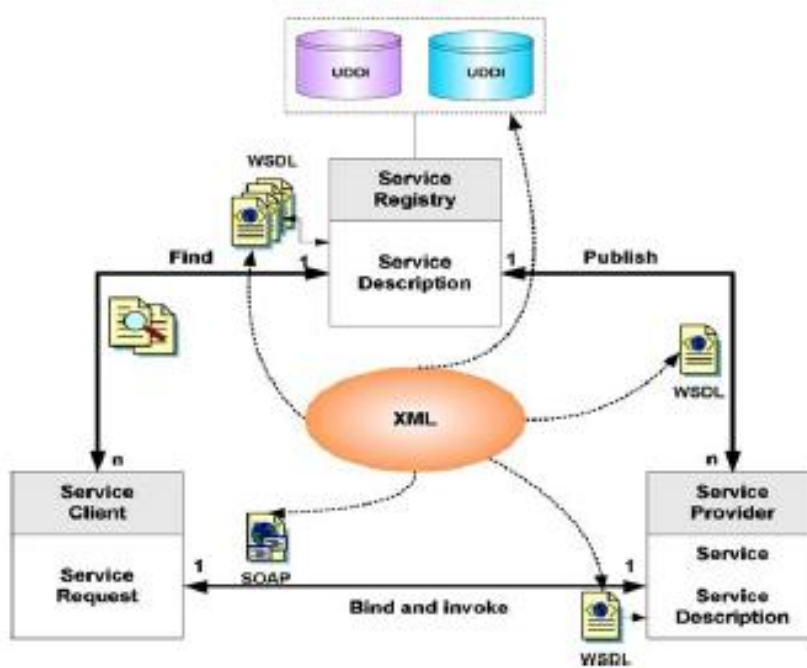
2.2.3. Procesų ir aplinkos analizė

Šiame skyriuje nagrinėsime detaliau registrų sistemos procesus ir veikiančius objektus. Žemiau pateikiama bendra procesų schema (3 pav.). Galime išskirti trys bendrus procesus:

- Serviso paieškos („FIND“) registro saugykloje
- Serviso užregistravimo, paskelbimo („PUBLISH“) registro saugykloje
- Serviso iškvietimo ir vykdymo („BIND AND INVOKE“)

Tai pat iš schemos galima nusakyti, kokie duomenų objektai dalyvauja su šiais procesais:

- UDDI registro bazė - XML sintaksės failų saugykla.
- WSDL failas - XML sintaksės pagrindu aprašomas serviso aprašymo failas.
- SOAP duomenys – tai HTTP protokolo duomenų perdavimo tipas, paremtas XML failo sintakse.



3 pav. Registro ir internetinių paslaugų procesų schema [2]

Internetinių paslaugų(servisų) paieškos proceso metu vartotojo programa jungiasi prie registų bazės. Atliekama paieška esamuose registruose pagal vartotojo įvestus duomenis. Radus atitinkamą paslaugą, ją galima įvykdyti. Prieš vykdant servisą vartotojo programa parsisiunčia serviso atitinkamą WSDL failą, iš analizuoja. Analizės metu randamos servise esamos operacijos, jų parametrai. Toliau programa, žinodama operacijų pavadinimus ir parametrus suformuoja SOAP pranešimą, kurį jį išsiunčia į paslaugos tiekėjo serverį. Tiekėjo serveris apdoroja atsiųstą pranešimą, įvykdo pasirinktą operaciją su vartotojo įvestais duomenimis ir grąžina rezultatą vartotojo programai.

Paslaugos registravimo proceso metu, servisą administruojantis dalyvis prisijungia prie registų bazės. Sukuria registrą, aprašo serviso tiekėjo duomenis: verslo organizaciją, kontaktinę informaciją ir t.t. Toliau registų valdymo procesui pateikia paslaugos WSDL failą, kurį iš analizavus rasti duomenis įtraukiami registro XML aprašymą.

Tai pat nepamirėta paieškos dalis. Šiame pavyzdyje pateikiamas standartinis registras UDDI pavyzdžiu. Šio registro paieška galima vykdyti raktinio žodžio „keyword“ būdu. Su įvestais duomenimis atliekamas tekstinis žodžio suliginimas su registre rasta žodžiais.

Pagrindiniai akcentai projektuojant semantinį registro modelį yra WSDL ir UDDI technologijos. WSDL ir UDDI plačiau apžvelgiami tolimesniuose skyriuose.

2.3. Paslaugų aprašo WSDL struktūros analizė

Internetinių paslaugų (toliau servais) sąsajos apibrėžimą WSDL galima suprasti kaip susitarimą tarp serviso ir jo naudotojo, užtikrinantį, kad servisas atliks tam tikrus veiksmus, gaudamas tam tikrus parametrus. Tai sąsajos (angl. Inteface) analogas objektinio programavimo kalbose [2]. Tokiam serviso apibrėžimui yra sukurta WSDL (angl. Web Services DefinitionLanguage) kalba. WSDL aprašams yra numatyti du panaudojimo atvejai. Vienu atveju, aprašymas naudojamas serviso kvietimui, kitu – standartinio serviso dokumentacijai. Toks WSDL panaudojimo pavyzdys galėtų būti situacija, kai keli servisų tiekėjai nori sukurti savo servisas, turinčius vieningą sąsają. WSDL aprašas paremtas XML sintakse, tačiau XML schemas apibrėžimo kalbą galima pasirinkti laisvai. Tai gali būti tiek „W3C XML Schema“, tiek RelaxNG, tiek ir kitos kalbos [2]. Taip pat WSDL numatyta galimybė naudoti ne tik SOAP protokolą [9].

Siekiant suteikti galimybę išplėsti kalbą, į WSDL buvo įtraukti tik būtiniausi internetinių paslaugų aspektai. Tai yra informacinių pranešimų aprašymai, serviso atliekamos operacijos bei serviso adresas ir palaikomas protokolas. Reikėtų paminėti tai, jog pirmi du elementai vadinami abstrakčia aprašo dalimi, o kiti du – konkrečiai. Abstrakti dalis turėtų būti naudojama standartinio serviso dokumentacijai. Taip pat reikėtų atkreipti dėmesį į tai, jog WSDL neatspindi servisų semantikos t.y. iš aprašo negalime tiksliai pasakyti nei ką daro internetinių paslaugų metodai, nei kokia tvarka juos reikėtų kviesti [3].

WDSL struktūra

Toliau pateikta tipinė WDSL failo struktūra, naudojant XML sintaksę [8].

Pagrindiniai elementai:

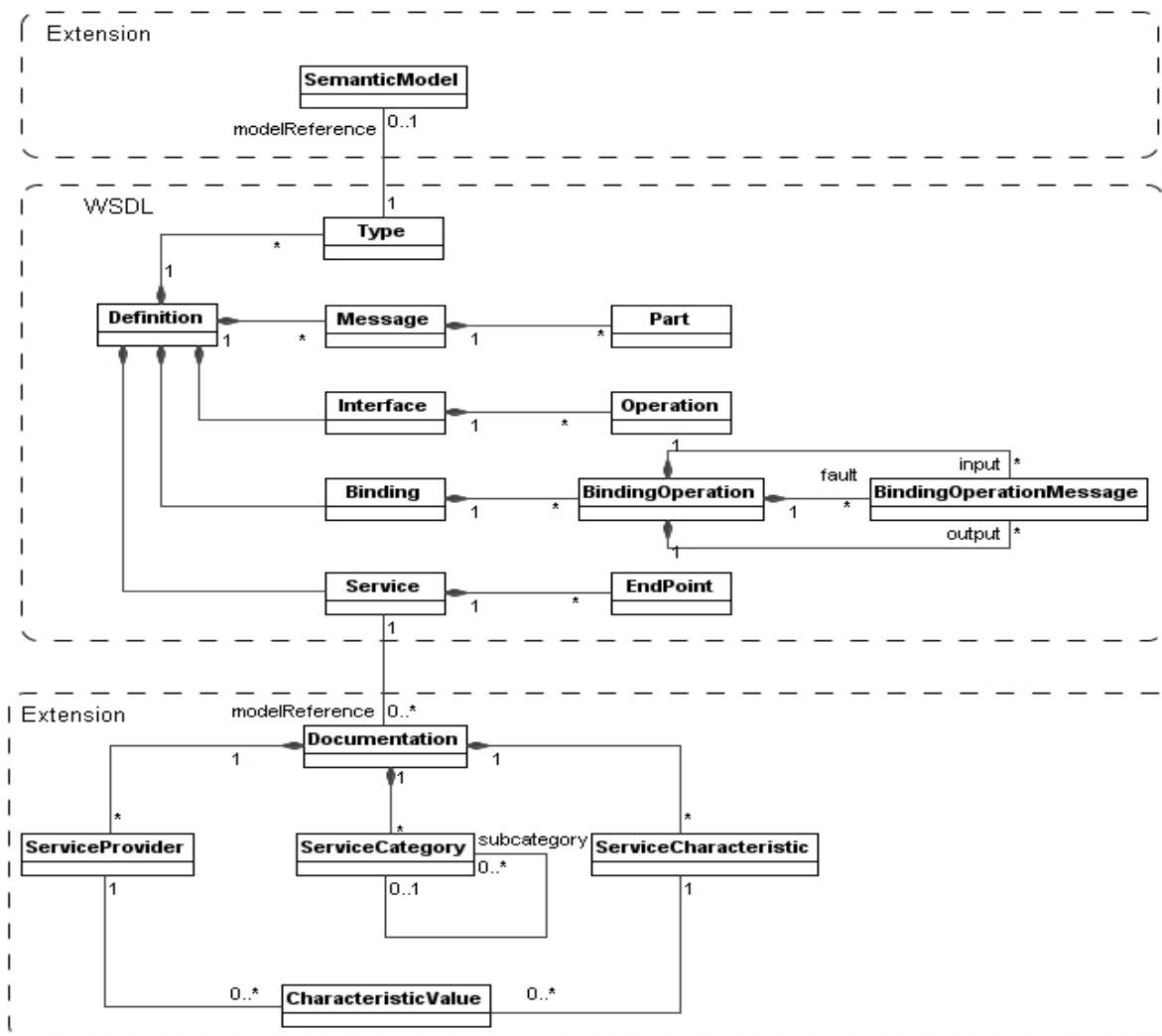
- *Types* - Aprašo duomenų tipus kuriuos naudoja serviso operacijos
- *Message* - Aprašomi sisteminiai pranešimai dirbant su servisu, kokie parametrai perduodami ir t.t.
- *portType* - Aprašo operacijos tipą: operacija vykdoma, be rezultatų grąžinimo ar su grąžinimu
- *binding* - Aprašomi protokolai, kuriais dirba servisas

Žemiau pateikiamas WSDL failo XML sintaksės išraiška. Joje aprašytas servisas – "UzupioViesbucioWebServisas". Iš aprašymo galima suprasti, kad servisas turi vieną operaciją – "ViesbutisUzupisOperation". Operacija turi po vieną įėjimą ir išėjimą. Message dalyje matome, kad įėjimo parametras nesuformuotas. Reiškia, jog

operacija vykdoma viena kryptimi t.y. vartotojas iškvietęs serviso operaciją, gauna iš karto jo atsakymą.

```
<definitions name="ViesbutisUzupis"
targetNamespace="http://j2ee.netbeans.org/wsdl/ManoWSDLFailas/ViesbutisUzupis"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://j2ee.netbeans.org/wsdl/ManoWSDLFailas/ViesbutisUzupis"
xmlns:ns0="http://xml.netbeans.org/schema/XandY"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  <types>
    <xsd:schema
targetNamespace="http://j2ee.netbeans.org/wsdl/ManoWSDLFailas/ViesbutisUzupis">
      <xsd:import schemaLocation="XandY.xsd"
namespace="http://xml.netbeans.org/schema/XandY"/>
    </xsd:schema>
  </types>
  <message name="ViesbutisUzupisOperationRequest"/>
  <message name="ViesbutisUzupisOperationResponse">
    <part name="Adresas" type="xsd:string"/>
    <part name="Miestas" type="xsd:string"/>
    <part name="telefono_nr" type="xsd:string"/>
  </message>
  <portType name="GautiViesbucioiInformacija">
    <operation name="ViesbutisUzupisOperation">
      <input name="input1"
message="tns:ViesbutisUzupisOperationRequest"/>
      <output name="output1"
message="tns:ViesbutisUzupisOperationResponse"/>
    </operation>
  </portType>
  <binding name="GautiViesbucioiInformacijaBinding"
type="tns:GautiViesbucioiInformacija">
    <soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="ViesbutisUzupisOperation">
      <soap:operation/>
      <input name="input1">
        <soap:body use="literal"/>
      </input>
      <output name="output1">
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="UzupioViesbucioWebServisas"/>
</definitions>
```

Paveikslėlyje 4 pateiktas WSDL aprašo duomenų modelis.



4 pav. WSDL duomenų modelis [2]

2.4. Standartinio (UDDI) registro analizė

Interneto paslaugų paieškos lygmenyje, naudojama UDDI (angl. Universal Description, Discovery, and Integration) technologija. UDDI specifikacija aprašo registrą, skirtą kaupti informacijai apie servिसus bei juos teikiančias organizacijas. Toks registras yra būtinas, norint pasinaudoti vienu iš SOA (angl. Service Oriented Architecture) privalumų – daugkartiniu komponentų panaudojimu. Programinės įrangos kūrėjams reikėjo vietos, kur būtų saugomos nuorodos į servिसus bei informacija, kaip su jais elgtis [3]. Tokia informacija suteikia galimybę dinamiškai surasti ir iškviešti reikiamą servिसą tiek sistemos kūrimo, tiek jos veikimo metu.

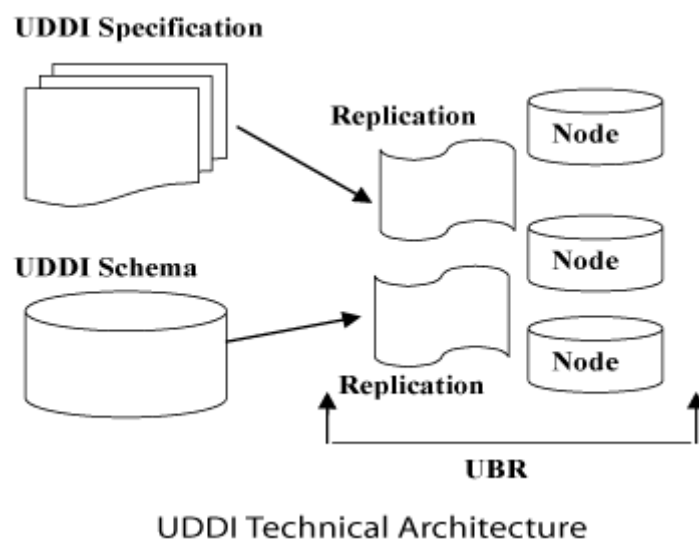
Pradžioje buvo planuojama, kad internete bus globalus viešas registras, kuriame visi galės talpinti informaciją apie save ir savo teikiamas paslaugas. Globalaus registro idėją palaikė tokios kompanijos kaip IBM, Microsoft, SAP ir kitos. Tačiau ši idėja nepasiteisino,

nes atsiradus naujoms interneto technologijoms, esamų registų aprašų nebeužteko. Vis dėlto, privatūs registrai, veikiantys kompanijų vidiniuose tinkluose, paplito gana plačiai. [3].

Registų sistemos architektūra

UDDI architektūra susideda iš trijų dalių:

- 1) UDDI duomenų modelio (angl. UDDI data model) – tai XML struktūros schema, kurioje įrašoma informacija apie paslaugą ir(arba) tą paslaugą teikiančią verslo įmonę: paslaugos pavadinimas, aprašymas, parametrai.
- 2) UDDI interfeiso specifikacijos (angl. UDDI API Specification) – tai funkcijos darbui su UDDI sistemos baze, publikuoti ar ieškoti registų
- 3) UDDI „debesies“ servisas (angl. UDDI cloud services) – tai administruojamoji dalis, kurioje atliekamos registų atnaujinimo, sinchronizacija su verslo įmonių duomenimis. 5 pav. pateikta techninė UDDI architektūra. Iš paveikslėlio matome, kad schemas ir specifikacijos sudaro tam tikrus pasikartojančius registrus, kurie kartojasi įvairiuose UDDI registų bazėse. Tai sudaro mazgus („node“), kurie tarpusavyje sujungti sudaro UDDI registų tinklą.



5 pav. UDDI techninė architektūra [3]

Registro struktūra ir duomenų modelis

UDDI įrašo struktūrą sudaro trys dalys. Tai baltieji, geltonieji ir žalieji puslapiai. Baltieji puslapiai – tai bendro pobūdžio informacija apie organizaciją. Pavyzdžiui: organizacijos pavadinimas, adresas, apibūdinimas. Geltonieji puslapiai skirti konkretesniam verslo aprašymui. Tai gali būti teikiamų paslaugų sąrašas. Taip pat čia saugomi verslo klasifikatoriai. Klasifikatorius identifikuoja verslą pagal tam tikrą klasifikavimo sistemą. Tai

gali būti geografinis indeksas ar kita standartinė sistema. Klasifikavimo sistemą taip pat gali aprašyti ir pats vartotojas. Žaliuosiuose puslapiuose saugoma techninė informacija apie servisus. Tai nuoroda į patį servisą, jo WSDL aprašą ir panašiai

UDDI duomenų modelio (angl. UDDI data model) – tai XML struktūros schema, kurioje įrašoma informacija apie paslaugą ir(arba) tą paslaugą teikiančią verslo įmonę: paslaugos pavadinimas, aprašymas, parametrai [4,10].

Pagrindinės 5 duomenų struktūros:

businessEntity

Aprašo paslaugas teikiančios verslo įmonės duomenis, kaip pavadinimas, kategorija, kontaktinę informaciją, verslo identifikacinius numerius (paryškintos dalys).

Aprašo pavyzdys:

```
<businessEntity businessKey="uuid:C0E6D5A8-C446-4f01-99DA-70E212685A40"
  operator="http://www.ibm.com"
  authorizedName="John Doe">
  <name>Acme Company</name>
  <description>
    We create cool Web services
  </description>
  <contacts>
    <contact useType="general info">
      <description>General Information</description>
      <personName>John Doe</personName>
      <phone>(123) 123-1234</phone>
      <email>jdoe@acme.com</email>
    </contact>
  </contacts>
  <businessServices>
  ...
  </businessServices>
  <identifierBag>
    <keyedReference
      tModelKey="UUID:8609C81E-EE1F-4D5A-B202-3EB13AD01823"
      name="D-U-N-S"
      value="123456789" />
  </identifierBag>
  <categoryBag>
    <keyedReference
      tModelKey="UUID:C0B9FE13-179F-413D-8A5B-5004DB8E5BB2"
      name="NAICS"
      value="111336" />
  </categoryBag>
</businessEntity>
```

businessService

Aprašoma teikiamos paslaugos charakteristikos: pavadinimas, aprašas ir t.t.

```
<businessService serviceKey="uuid:D6F1B765-BDB3-4837-828D-8284301E5A2A"
  businessKey="uuid:C0E6D5A8-C446-4f01-99DA-70E212685A40">
  <name>Hello World Web Service</name>
  <description>A friendly Web service</description>
  <bindingTemplates>
  ...
  </bindingTemplates>
  <categoryBag />
</businessService
```

Kitos duomenų struktūros:

bindingTemplate – tai techninis aprašas, panašus į WSDL aprašymą. Įrašoma informaciją apie servisą, jo pasiekiamumą.

publisherAssertion – skirtas, kelių verslo įmonių paslaugų susiejimui.

tModeldata – svarbi struktūra, kurioje galima išplėsti kitomis schemomis esamą registrą. Ši dalis dažniausiai naudojama registro semantikos pritaikymo procese.

UDDI funkcijos

Pats UDDI registras turi savo funkcijų API. Pagrindinės funkcijos pateiktos žemiau pateiktoje lentelėje (1 lent.)

1 lentelė

UDDI API funkcijos

Inquiry APIs	Publishing APIs
find_business	save_business
find_service	save_service
find_binding	save_binding
find_tModel	save_tModel
get_businessDetail	delete_business
get_serviceDetail	delete_service
get_bindingDetail	delete_binding
get_tModelDetail	delete_tModel
get_registeredInfo	get_authToken

Inquiry API's – tai paieškos UDDI registruose, funkcijos. *Publishing API's* – UDDI registru paskelbimui (įrašymui), šalinimui.

Pagal funkcijų pavadinimus galima atskirti jų naudojimo paskirtį:

- *find_xxx* – funkcijos rasti iš UDDI registro bazės reikalingo elementą, gražinamas registro identifikacinis numeris, pagal kurį galimos *get_xxx* funkcijų operacijos. Pradiniai parametrai funkcijoms gali būti, ir pats identifikacinis numeris, arba paprasčiausias eilutės tipo įmonės ar paslaugos pavadinimas.
- *get_xxx* – funkcijos skirtos gauti informaciją. Pradinis parametras gali būti identifikacinis numeris. Rezultatas: XML sintakse gauti duomenys. Jose sintaksiškai aprašyti elementai, su jais esamais duomenimis.
- *save_xxx* – funkcijos, skirtos įkelti naujus registro duomenis. Suteikiamas identifikacinis numeris, įterpiami elementai.
- *delete_xxx* – pašalinami elementai.

Magistro darbe suprojektuotame modelyje UDDI funkcijos nenaudojamos.

2.5. Semantinių technologijų analizė

Semantika dažnai laikomas ontologijos panaudojimas. Ontologija (Graikiškai $\omega\nu$ „būtis“, $\lambda\acute{o}\gamma\omicron\varsigma$ „žodis“ ar „kalba“) – kompiuterijoje šiuo termino daugiskaitine forma ontologijos vadinamas tam tikros srities sąvokų visumos specifikuojamas išreikštu pavidalu. Semantikoje tai tam tikrų žodžių susiejimas su kitais žodžiais pagal prasmę. Dabar projektuojamos semantinės technologijos daugiau remiasi ontologijomis.

Ontologija galima aprašyti tam tikroje realioje srityje esančių objektų klasifikacija. Pavyzdžiui, 6 paveikslėlyje pateikiama tokia pavyzdinė klasifikacija. Turime *World ontology* ontologiją. Iš jos galima išskirti tam tikras realias sritis: kelionės (*Travel*), Maistas (*Food*), Finansai (*Finance*) ir t.t. Atitinkamai išskiriame žemesnes vaiko klases, vėliau iš jų dar smulkesnes klases. Taip galima sudaryti tam tikros srities ontologijas, aprašyti siejamas klases.



6 pav. Ontologijos pavyzdys [15]

RDF

RDF (angl. The Resource Description Framework) – resursų aprašymo paketas [17]. Paketas skirtas interneto puslapiuose esančių resursų reprezentacijai. Resursais gali būti asmens vardas, pavarde ar automobilio modelio tipas, kurie aprašomi interneto puslapiuose tekstine išraiška. Kadangi pradėjus tobulėti internetinėm paieškos sistemom, reikalinga buvo gebėti atskirti žodžius, kurie tam tikrame kontekste turi skirtingas prasmes. Pavyzdžiui: žodžiu „vilnius“ galima suprasti, kad čia turima omenyje Lietuvos sostinę Vilnius arba viešbutis pavadinimu „Vilnius“, arba gatvės pavadinimas Šiaulių mieste.

RDF paketu galima aprašyti resursų traktuojamas reikšmes nuo atitinkamo resurso naudojimo konteksto. Toks aprašymo naudojimas – paprasčiausias semantikos taikymo būdas.

OWL

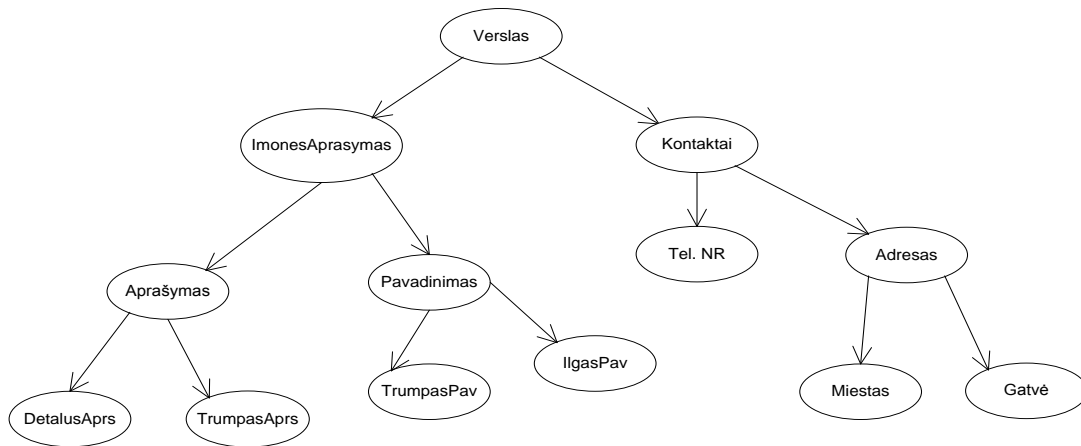
OWL ontologijos kalba [16] buvo sukurta naudoti OWL palengvina mašininį turinio interpretavimą palyginus su XML, RDF ar RDF-S, nes praplečia žodyną ir formalią semantiką. OWL taip pat daug sukauptos patirties įtraukė iš DAML+OIL ontologijos kalbos kūrimo ir taikymo.

OWL palaikomas XML sintaksės pagrindu, kurio galima aprašyti klases, jų atributus, klasių ryšius. OWL - papildytas ir išplėstas RDF variantas.

OWL kalba yra išskaidyta į tris poaibius: OWL Lite, OWL DL ir OWL Full, kurių kiekviena turi vis didesnę išraiškų laisvę.

- OWL Lite yra naudojama tokioms reikmėms, kaip klasifikavimo hierarchijos ir paprasti apribojimai. Dėl to šiai kalbos rūšiai yra žymiai lengviau kurti įrankius nei kitoms, ir ji turi mažesnę formalų sudėtingumą.
- OWL DL suteikia maksimalų ekspresyvumą, išlaikant skaičiavimų baigtumą (užtikrinama, kad visos išvados yra garantuotai suskaičiuojamos) ir sprendimo baigtumą (visi skaičiavimai bus atlikti per baigtinį laiką). OWL DL gali būti naudojami visi OWL kalbos dariniai, bet juos galima naudoti su kai kuriais ribojimais (pavyzdžiui, nors klasė gali būti kelių klasių poklasė, bet klasė negali būti kitos klasės atskiras atvejis). OWL DL pavadinimas išplaukia iš jo suderinamumo su apibūdinimo logika (angl. description logics) – tyrinėjimų srities, kuri studijavo logiką nuo pat formalaus OWL įkūrimo.
- OWL Full suteikia maksimalų panaudojimą, bet nesuteikia garantijų dėl skaičiavimų baigtumo.

Kiekviena kalbos rūšis yra ankstesnės praplėtimas, todėl paprastesnėje kalbos rūšyje aprašyta ontologija taip pat bus galiojanti sudėtingesnėje. 7 paveikslėlyje pateikiamas OWL ontologijos aprašo pavyzdys:



7 pav. Pavyzdinė verslo ontologija

Ontologija aprašo verslo organizaciją: pavadinimą, veiklos aprašymą, kontaktinius duomenis. Ontologija išskaidoma į klases su ryšiais, nurodoma kas nuo kuo priklauso. Semantiškai žiūrint galima suprasti, kad jei kalbama apie „Kontaktus“, tai turima omenyje ne vien kontaktai, bet ir telefono numeris, adresas. Jei paieškos sistemoje ieškomas gatvės pavadinimas („Gatvė“ klasė), tai neradus tikslios reikšmės - galima ieškoti kitu objektu „Adresas“, kuris turės tą pačią prasmę, nes adreso dalyje yra gatvė.

Toks paprasčiausias ontologijos struktūros išdėstymas leidžia atlikti paprastą semantišką objektų panašumų įvertinimą.

2.6. Semantinio registro kūrimo būdai pritaikant semantines technologijas

Šiame skyriuje apžvelgiami semantinio registro kūrimo būdai panaudojus semantines technologijas. Dabar labiausiai varijuoja standartinio registro(UDDI) arba WSDL failo aprašo išplėtimas semantikomis.

2.6.1. UDDI registro išplėtimas semantika

UDDI registro išplėtimas semantikomis remiasi struktūrų papildymu naujais XML sintaksės elementais [19]. Šiais elementais nurodomos semantikos: klasės, objektai iš ontologijos. Žemiau pateikiama registro UDDI struktūra *BusinessEntity* aprašanti pavadinimą, pavadinimą, kontaktus:

```
<businessEntity businessKey=..."
```

```

        operator="http://www.ibm.com"
        authorizedName="John Doe">
<name>Acme Company</name>
<description>
    apie verslo įmonę ...
</description>
<contacts>
    <contact useType="general info">
        <description> kontaktinė informacija</description>
        <personName> Asmuo atsakingas </personName>
        <phone>(123) 123-1234</phone>
        <email>jdoe@acme.com</email>
    </contact>
</contacts>
</businessEntity>

```

Iš skyriaus 2.5 naudosime pavyzdinės ontologijos struktūrą (7 pav.). Iš ontologijos ištrauksime ontologijos klases, kurios nurodytų struktūros elementų semantiką. Įtraukti elementai paryškinti.

```

<businessEntity businessKey=..."
    operator="http://www.ibm.com"
    authorizedName="John Doe">
    <name > <sem klase = "verslas#TrumpasPav">Acme Company </sem></name>
    <description>
        <sem klase = "verslas#Aprasymas"> apie verslo įmonę ... </sem>
    </description>

    <contacts>
        <sem klase= „verslas#Kontaktai">
        <contact useType="general info">
            <description> kontaktinė informacija</description>
            <personName> Asmuo atsakingas </personName>
            <phone>(123) 123-1234</phone>
            <sem klase= „verslas#Tel_NR"><email>jdoe@acme.com</email></sem>
        </contact>
        </sem >
    </contacts>
</businessEntity>

```

Įtrauktas elementas <sem>, kuris turi klasės atributą „klase“. Atributui nurodoma ontologija – „verslas“, ir ontologijos klasė.

Tai dažniausias būdas atlikti registro semantikos pritaikymą UDDI registru. Kai atliekama semantinė paieška, analizuojant išskiriamas <sem> elementas. Su šiais elementais vėliau galima atlikti tolesnius paieškos veiksmus. Šis būdas tinkamas magistro darbo tyrimui kuriant modelį su papildomais semantiniais aprašais.

2.6.2. WSDL išplėtimas semantika

Analogiškas principas vyksta, kaip ir praeitam skyriuje aprašytam UDDI registro išplėtimui [18]. Tik priskirimas vyksta jau WSDL failo elementuose. Pateikiamas pavyzdinis išplėtimas papildant tokiais pačiais semantiniiais aprašais 2.6.1 skyriuje. Pavyzdyje įtrauktas <sem> elementas:

```
<message name="manoWSDLOperationResponse">
  <part name="resultatas" type="xsd:string"/>
</message>
<portType name="manoWSDLPortType">
  <sem klase= „verslas#Kontaktai“>
    <operation name="sudetis">
      <input name="input1" message="tns:sudetisRequest"/>
      <output name="output1" message="tns:sudetisResponse"/>
    </operation>
  </sem >
  <operation name="labas">
    <input name="input2" message="tns:manoWSDLOperationRequest"/>
    <output name="output2"
message="tns:manoWSDLOperationResponse"/>
  </operation>
</portType>
```

2.7. Semantinės paieškos algoritmų analizė

Šiame skyriuje analizuojami semantinės paieškos algoritmai semantiškai sudarytam registrai. Aprašomas algoritmo veikimas ir integravimas.

2.7.1. Semantinės paieškos formavimas

Semantinės paieškos formavimas priklauso nuo realizuoto semantinio registro modelio. Priklausomai nuo kaip modelyje semantiškai aprašyta paslaugos struktūra - kiekviena paslauga turi operacijas. Jos tuo tarpu turi įėjimus ir išėjimus. Tai pat įėjimų ir išėjimų skaičiai gali variuoti. Pavyzdžiui: operacija turi viena parametą ir grąžina du parametrus. Kita operacija atitinkamai 3 įėjimus ir 1 išėjimą. Jei modelyje ontologijos klasėmis yra susieti visi paslaugoje esantys struktūros elementai - tai paieškos formavimas galimas pateikus parametrus paieškos programoje įėjimams, išėjimams arba prie išvardintų papildomai pridėjus operacijos parametro įvedimą [15], [20].

Laikykite, kad ieškome internetinės paslaugos iš viešbučių kategorijos. Paslauga turi turėti operaciją – viešbučio paieška ir pagal įvestą pavadinimą grąžintų viešbučio adresą. Paieškos formavimas būtų: pateikti operacijos įvedimui - semantinę klasę „viešbutis“, įėjimams – „pavadinimas“, išėjimams – „adresas“. Toks paieškos formavimas aktualus magistro darbo paieškos algoritmo vykdymui. Reikia nuspręsti, kokiais paieškos parametrais

galima paieška. Tai priklauso nuo kaip registro modelyje semantiškai aprašomos paslaugos. Jei semantiniame modelyje ontologijos klasėmis susieti tik įėjimai ir išėjimai, tai paieškos formavime nereikalinga naudoti funkcijos parametro.

2.7.2. Semantinės paieškos algoritmo veikimo principas

Semantinės paieškos parametrai yra semantiniai žodžiai arba atitinkamos klasės pavadinimas iš tam tikros srities ontologijos. Semantinio paieškos tikslas - atlikti įvesto parametro radimą ir panašumo nustatymą su registruose priskirtais internetinių paslaugų(servisų) semantinių žodžių(klasių). Rastus semantinius žodžius palyginti – atlikti panašumo nustatymą. Išreikšti nustatyto panašumą – skaičiaus koeficientu, ar logine išraiška (*true, false*). Galiausiai pateikti rezultatus [15].

Semantinės paieškos veikimą galima aprašyti šiais žingsniais:

1. Vartotojas įveda į paieškos programą ieškomus parametrus(semantiniai žodžiai)
2. Programa apdoroja įvestus duomenis ir pateikia semantinės paieškos algoritmui
3. Algoritmas kreipiasi į registru bazę
4. Parenkama pirma registruota internetinė paslauga
5. Iš internetinės paslaugos išskiriamos susietos semantinės klasės
6. Atliekamas įvestų parametrų panašumo nustatymas su paslaugos semantiniiais žodžiais
7. Panašumo įvertinimas (nustatomas skaitinis koeficientas)
8. Įvertinimo išsaugojimas
9. Toliau atliekami iš naujo žingsniai 5 – 8 su sekančia rasta paslauga
10. Atlikus veiksmus su visomis paslaugomis, vykdomas rastų įvertinimų pateikimas vartotojui (pavyzdžiui pateikiamas surikiuotas sąrašas pagal koeficientą)

Pateikti žingsniai yra bendriniai, įvairiose semantinės paieškos realizacijose gali skirtis. Svarbiausi išskiriami yra 5 ir 6 žingsniai. Panašumo nustatymas ir įvertinimas priklauso nuo pasirinkto semantinių panašumų metodo.

2.7.3. Standartinis semantinis panašumo metodas

Semantinių panašumų metodas[20] - atlieka vartotojo įvestos ontologijos klasės palyginimą su interneto paslaugos susieta ontologine klase. Palyginimo rezultata galime išreikšti tam tikra reikšme. Klasių palyginimo procedūra:

```
1: if outA = outQ then
2:   return Exact
3: else if outQ superclass of outA then
4:   return Plugin
5: else if outQ subsumes outA then
6:   return Plugin
7: else if outA subsumes outQ then
8:   return Subsumes
9: else
10:  return Fail
11: end if
```

Turime ontologines klases *outA* ir *ouQ* – atitinkamai paslaugos klasė ir ieškomoji klasė.

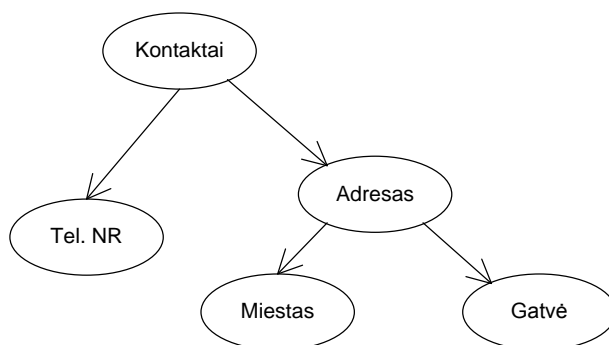
Jei *outA* lygi *outQ* – tai laikoma, kad šios klasės vienodos t.y. pagal prasmę identiški objektai.

Jei *outQ* yra poaibis (subklasė) iš klasės *outA* – tai ieškotoji klasė panaši tam tikromis savybėmis į paslaugos klasę – tai dalinis panašumas.

Jei *outQ* yra pagrindinė klasė ir *outA* yra jos poaibis – tai ieškotoji klasė laikoma dalinai identiška paslaugos klasei.

Jei *outQ* klasė nėra nei viename *outA* klasės poaibyje ir atvirkščiai – laikoma, kad klasės visiškai skirtingos.

Toliau pateiksime pavyzdį su realia ontologija. Laikome, kad turime kontaktų ontologiją (8 pav.). Joje aprašytos klasės *Tel. NR*, *Adresas*, *Miestas*, *Gatvė*.



8 pav. Kontaktų ontologija

Interneto paslaugoje esančią operaciją, kuri turi po vieną įėjimo ir išėjimo parametą. Įėjimas susietas su *Adresas* klase, išėjimas su *Tel .NR* .Vartotojas įveda įėjimo parametro klasę *Miestas*. Panašumo nustatymas, kai lyginami įėjimų parametrai t.y. *Miestas* su *Adresas* . *Miestas* priklauso *Adreso* poaibiui, tai tada operacija dalinai panaši. Palyginimas – *Tel. NR* ir *Adresas* – operacija nepanaši pagal paiešką.

2.7.4. Semantinio panašumo išreiškimas reikšme

Semantinio panašumo išreiškimas tam tikra verte priklauso nuo metodo atliekamų sąlygų. Laikykime, kad ieškomoji klasė(Q) lyginama su serviso(S) klase.

- 1) jei $S=Q$, tai ieškoma klasė semantiškai vienoda serviso klasei t.y. jos nusako tą patį objektą. Tuomet panašumo koeficientas lygus 1.
- 2) $S > Q$, tai laikoma, jog viena iš serviso tėvo klasės išeinančių vaiko klasių semantiškai vienoda su ieškomąja t.y. išvestoji vaiko klasė turi tokias pačias savybes, kaip ir nurodytoji tėvinė klasė. Tuomet taikomas panašumo koeficientas kaip ir pirmoje sąlygoje – 1.
- 3) Jei $S < Q$, tai ieškoma klasė ontologijoje yra tėvinė serviso klasės atžvilgiu. Tada laikoma, jog serviso klasė dalinai turi tas pačias savybes, kaip ir ieškomoji. Semantišku požiūriu jos nėra vienodos. Panašumo koeficientas apskaičiuojamas remiantis, kokiu santykiu prarandamas semantinis panašumas. Skaičiavimo formulė:

$$koef = \frac{SN}{QN}$$

koef – panašumo koeficientas; *SN* – iš serviso išvedamų visų vaiko klasių skaičius ontologijos atžvilgiu; *QN* – iš ieškomosios klasės išvedamų visų vaiko klasių skaičius tos pačios kaip ir serviso ontologijos atžvilgiu.

- 4) Jei netenkinamos pirmosios trys sąlygos, bet abi klasės priklauso tai pačiai ontologijai, laikoma jog klasės ontologijos atžvilgiu labai skiriasi semantiškai, beveik nevienodos. Tuomet panašumo koeficientas nustatomas pasirinktinai. Galima laikyti – 0, arba jei reikia vis tiek nurodyti, jog klasės priklauso tai pačiai sričiai t.y. ontologijai, bet semantiškai labai skiriasi, remiamasi geometrinio atstumo apskaičiavimu ontologijos klasių hierarchijoje. Apskaičiavimo formulė:

$$koef = \sqrt{\frac{SN \cap QN}{SN \cup QN} * \frac{SN \cap QN}{QN}}$$

koef – panašumo koeficientas; *SN* – iš serviso išvedamų visų vaiko klasių skaičius ontologijos atžvilgiu; *QN* – iš ieškomosios klasės išvedamų visų vaiko klasių skaičius tos pačios kaip ir serviso ontologijos atžvilgiu.

2.7.5. Kiti semantinio panašumo metodai

Toliau šiame poskyryje aprašomi metodai yra standartinio panašumo metodo išvestiniai variantai. Dažniausiai išskiriami trys metodai:

- Greedy metodas
- False metodas
- Biparte metodas

Greedy metodas

Metodas remiasi visų paieškos parametrų klasių sulyginimu su interneto paslaugoje susietomis semantinėmis klasėmis atliekant sąlyginį tikrinimą. Klasių panašumo nustatymo procedūra vyksta kaip ir standartinio. Skirtumas atsiranda, kai pasirenkamas panašumo nustatymo laipsnis t.y. klasių sąlygų skaičius [15],[20].

Šiame metode dažniausiai taikomas 4 sąlygų laipsnis, laikome *A* – paslaugos klasė, *B* – ieškomoji klasė:

- $A = B$, klasės semantiškai vienodos
- $A > B$, tai yra *A* klasei priklauso *B* klasė – semantiškai vienodos
- $A < B$, tai *B* klasei priklauso *A* klasė, bet semantišku atžvilgiu paslaugos klasė tik dalinai atitinka ieškomąją
- *A* ir *B* nenusisijusios

Šis panašumo metodas remiasi visų įmanomų variantų klasių palyginimu, tad jei lyginame daugiau negu vieną klasę padidėja skaičiavimų skaičius, kol gaunamas apibendrintas panašumo įvertinimas t.y. atrenkamos didžiausios reikšmės iš variantų ir skaičiuojamas vidurkis.

False metodas

False metodas yra paremtas standartiniu panašumo metodu [20]. Naudoja 3 sąlygų tikrinimą. Skirtingai negu **Greedy** metodas esant kelių parametrų lyginimui metodas po pirmų ieškomos ir paslaugos klasių variantų lyginimo – atmeta šį variantą, jei didelis panašumas. Toliau jie nebenagrinėjami, tęsiamas panašumo tikrinimas su likusiomis klasėmis Gaunami įvertinimai ir vėl atmetama, imamas sekantis t.t.

Atmetimo požymis yra, kai tenkinama (A – paslaugos klasė, B – ieškomoji klasė):

- $A = B$, klasės semantiškai vienodos
- $A < B$, tai yra A klasė priklauso B klasė – semantiškai vienodos

Galiausiai iš gautų atmetų variantų įvertinimų reikšmių parenkamas didžiausias ir toliau laikomas pagrindiniu panašumo koeficientu.

Metodo panašumo koeficiento apskaičiavimas priklauso nuo ieškomų klasių eiliškumo. Šio metodo principas sumažinti panašumų tikrinimo skaičių, kai didėja lyginamųjų klasių skaičius.

Pateikimas metodo veikimo pavyzdys. Semantines klases naudosime iš ankstesnio skyriaus aprašytos kontaktų ontologijos (8 pav.).

Turime aprašytą paslaugą klasėmis: *KONTAKTAI*, *TEL. NR*, *GATVĖ*. Paieškos klasės: *TEL. NR.*, *ADRESAS*.

- Atlikus lyginimą $\{TEL. NR ; KONTAKTAI\}$, tenkinama sąlyga $A = B$, koeficientas – 1.
- Toliau atliekame lyginimą $\{ADRESAS, TEL. NR\}$ - netenkinamos sąlygos, su tuo pačiu *ADRESAS* klase atliekame tolesnį lyginimą
- Lyginimas $\{ADRESAS, GATVĖ\}$ tenkinama sąlyga $A < B$, koeficientas – 1.
- Ieškomos klasės panaudotos, baigtas lyginimas.
- Rasta didžiausia koeficiento reikšmė - 1

Verta paminėti, jo šio metodo yra kelios variacijos kurios priklauso nuo lyginamosios klasės atžvilgio. Aprašytame pavyzdyje, tikrinta ieškomosios klasės atžvilgiu. Jei lyginama paslaugos klasių atžvilgiu, pakinta atmetimo sąlygos:

Atmetimo požymis paslaugos atžvilgiu (A – paslaugos klasė, B – ieškomoji klasė):

- $A = B$, klasės semantiškai vienodos
- $A > B$, tai yra A klasė priklauso B klasė – semantiškai vienodos

Biparte metodas

Biparte metodas paremtas įvairių lyginamųjų klasių rinkinių sudarymu naudojantis grafų atvaizdavimu. Grafo viršūnės – klasės, lankai – panašumo laipsnio sąlyga. Toliau sudaromas grafas išskaidomas į mažesnius grafus atskiriant pagal panašumo sąlygas. Vykdomas grafų reikšmių įvertinimas.

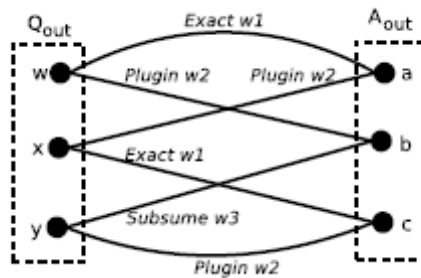
Remsimės šaltinio [23] pavydžiu, jog turime tokias aibes su nariais: ieškomų klasių - $Q_{out} \{w, x, y\}$ ir paslaugos klasių - $A_{out} \{a,b,c\}$. Nustatome klasių lyginimo sąlygas ir galimas reikšmes(8 pav.).

Degree of Match	Weight of edge
Exact	w_1
Plugin	w_2
Subsumes	w_3

9 pav. Lyginimo reikšmės

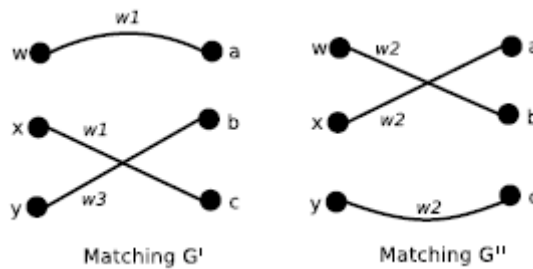
Exact – kai klasės lygios, *plugin* – ieškomoji klasė yra vaiko klasė paslaugos klasės atžvilgiu, *subsumes* – ieškomoji klasė yra tėvinė paslaugos klasei. w reikšmės pasiskirsto reikšmių dydžiu - $w_1 < w_2 < w_3$ t.y. atitikmuo turi turėti mažiausią apskaičiuotą reikšmę, kita sąlyga didesnė ir t.t.

Iš aprašytų aibių Q_{out} ir A_{out} sudaromas grafas (10 pav.). Sudaryti visi įmanomi rinkiniai ir įvertintos rinkinių panašumų reikšmės.



10 pav. Parametrų rinkiniai

Toliau iš šio grafo išskiriame du papildomus grafus: G' – grafai, kurie tenkina sąlygas *exact* ir *subsume* ir G'' – visi likę lankai (11 pav.).



11 pav. Tikslių(exact) ir panašių(subsume) grafai

Iš kiekvieno grafų G' ir G'' lankų reikšmių surandame didžiausias reikšmes ir jas palyginime. Šiuo atveju, tai $G' - w_3$ ir $G'' - w_2$. Įvertinus $w_2 < w_3$, teigiama, kad bendrasis koeficientas yra w_2 .

Šiuo metodu bandoma tiksliau įvertinti įvestų parametrų įvertinimą

2.7.6. Semantinių panašumų metodų palyginimas

Apžvelgus semantinių panašumų nustatymo metodus galima išskirti palyginamas charakteristikas (lentelė). Metodo vykdymo laiko charakteristikos palyginimas aprašytos šiame straipsnyje [20].

2 lentelė

Semantinių panašumų metodų palyginimas

Metodas	Metodo vykdymo laikas	Tikslumas	Realizacijos sudėtingumas
Greedy	Lėtas	Didelis	Paprastas
False	Greitas	Mažas	Vidutiniškas
Biparte	Vidutinis	Didelis	Sudėtingas

Pagal metodo vykdymo laiką lėčiausias yra **Greedy** metodas, nes jo veikimo principas pagrįstas visų reikšmių mechaniskas apskaičiavimas, bet panašumo apskaičiavimo tikslumas didelis. Greičiausias **False** metodas – dėl lyginamųjų variantų rinkinių atmetimo. Realizacijos požiūriu sudėtingiausia suprojektuoti **Biparte** metodu algoritmą.

Magistro darbo tyrimui geriausiai tinka realizuoti algoritmą paremtą **Greedy** metodu, kurį paprasta suprojektuoti, didelis panašumo apskaičiavimo tikslumas. Kadangi algoritmo tyrime akcentuojame algoritmo veikimo laiką – šis metodas tinkamas.

2.8. Standartinio registro modelio realizacijos duomenų bazėje analizė

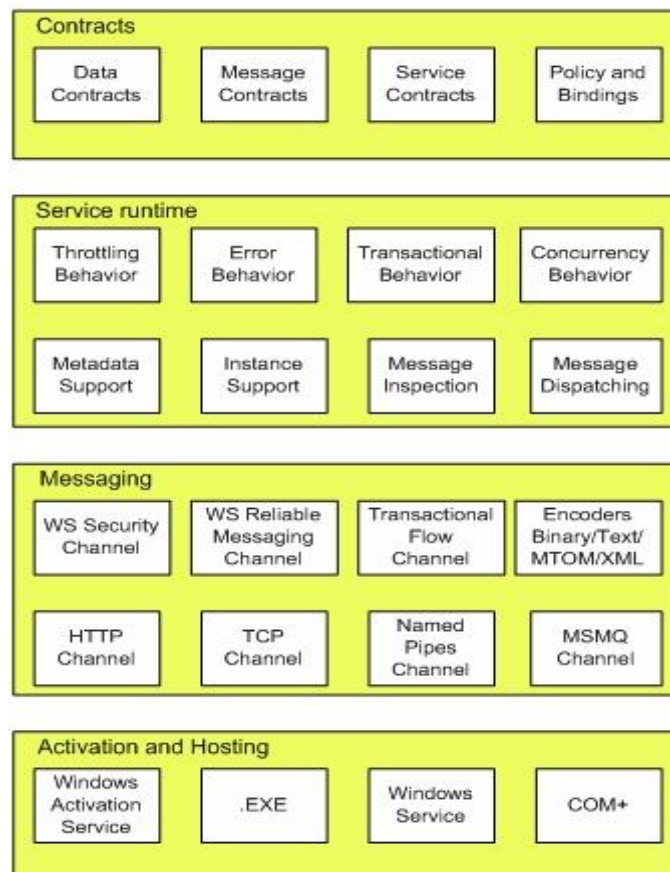
Šiame skyriuje nagrinėjame registro realizacijos duomenų bazėje būdus. Žinant, kad XML struktūros yra analogiškos struktūros duomenų bazei, kurioje galime sudaryti duomenų struktūras (lenteles), ryšius. Kaip pavyzdį panagrinėsime Microsoft registro modelis realizaciją duomenų bazėje [13].

Šis paketas pateiktas, kaip naujo UDDI registro modelio realizacija. Kaip žinome Microsoft kartu su IBM pirmieji suprojektavo UDDI registro modelį, ir jį pritaikė. Tik vėliau UDDI registrai tapo nebe efektyvus ir jų buvo atsisakyta. Pradėta projektuoti naujus modelius. Šis registro modelis integruojamas su .NET technologija. Pagrindiniai aspektai, kurie atsispindi suprojektuotame registro modelyje, tai naujos registro valdymo funkcijos ir pačių

registrų aprašų saugojimas duomenų bazėje. Skirtingai negu kiti projektai, kurie registrus saugo XML failuose. Tai pat architektūrinė dalis aprašoma kartu su papildomai registro duomenimis kaip vienas modelis.

Registro struktūra

Pateikiamo registro modelis, skiriasi nauja registro aprašo struktūra. Šis modelis aprašo ne vien tik standartišką registrą, bet įtraukia papildomai architektūrinės dalis, kaip serviso iškvietimas, parametrų perdavimas, kaip anksčiau ši dalis buvo aprašoma WSDL failuose. Tam atlikti panaudotas Windows Communication Foundation (toliau WCF) paketas. WCF suskirstytas į lygmenis. Kiekvienas lygmuo turi savo specifika, panaudojimą. Jie pateikiami 12 paveikslėlyje.

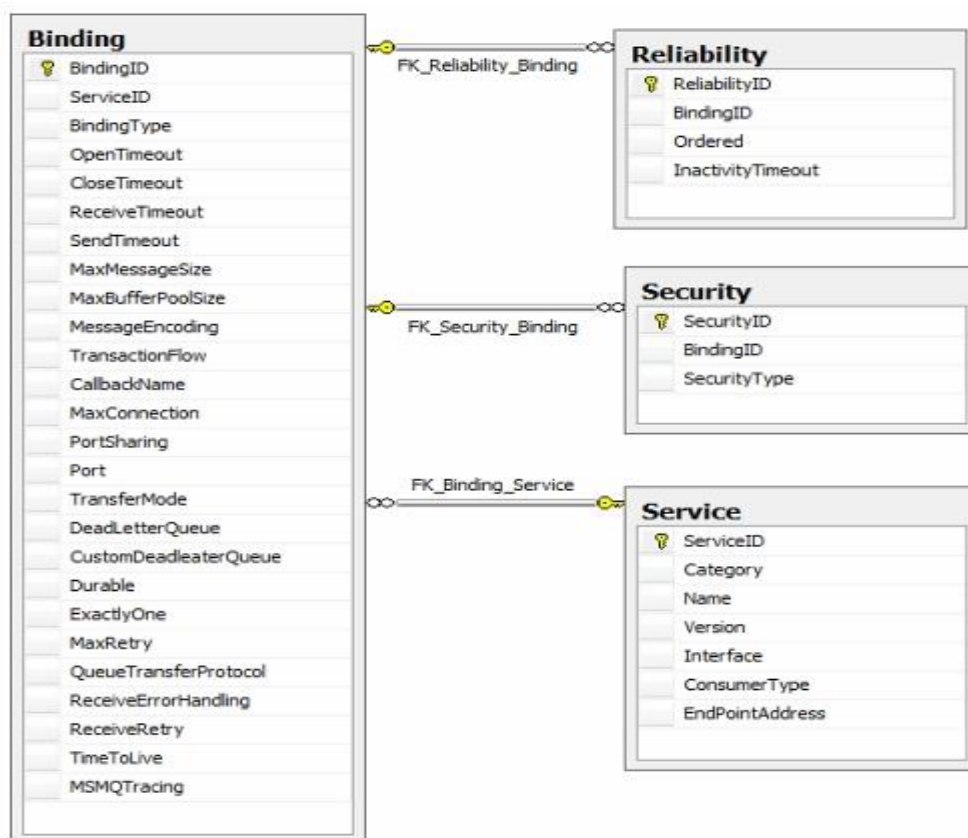


12 pav. WFC lygmenys

- *Contracts* – tai aukčiausias lygmuo, jis skirtas servisų aprašymui, verslo aprašymui. Paslaugų metodų iškvietimo aprašymas, kokiais protokolais dirbama ir t.t. Ši dalis primena WSDL ir UDDI registro mišinį;

- *Service runtime* - tai serverio procesų lygmuo, kurie aktyviai atlieka ir vykdo užklausų operacijas, vykdo servisų iškvietimo, valdymą. Kviečiančių funkcijų interpretavimas;
- *Messaging* – tai pranešimų, serverio modulių, duomenų mainų specifikavimas. Tai fizininė architektūros dalis. Protokolų nustatymas: HTTP, TCP ir t.t.;
- *Activation and Hosting* – žemiausias lygmuo. Architektūros organizavimas, serverio aplikacijų veikimas.

Kaip matome iš šios registro architektūros, išdėstymo. Semantinei paieškai atlikti naudojamas aukščiausias lygmuo.



12 pav. Registro struktūra duomenų bazėje [4]

12 paveikslėlyje pateikta registro modelio realizacija duomenų bazėje. Pateiktoje struktūroje matome, kad serviso aprašymas yra *Service* lentelėje. Joje pateikiama serviso informacija, kaip pavadinimas, kategorija kuriai priskiriama ir kita. Kiekvienas servisas yra unikalus sistemoje ir kiekvieną atskirai sieja atitinkama, serviso iškvietimo realizacijos lentelė *Binding*. Joje aprašomi visi serviso naudojami parametrai, gyvavimo laikas, kokie protokolai naudojami ir t.t. Taip siejamo lentelės *Security* ir *Reliability*. Kurios aprašo serviso naudojimo saugumo lygmenį. Ši dalis analogiškai WSDL faile buvo aprašoma tai pat.

Tokia registro realizacija patogi, nes patogesnis registro įrašų valdymas. Patogesnė paieška, paprasčiausiai pakankama realizuotos funkcijos su SQL sintaksės užklausa. Semantikos naudojimui pakanka sukurti lentelės įrašus, kurios saugotu siejamos ontologijos klases.

2.9. Esamų semantinių registro modelių analizė

Tokių semantinių registro modelių nedaug. Visu pirma, patys UDDI registrai nebenaudojami praktiškai dėl semantikos aprašymo trūkumo. Išsiplėtus internetinių technologijų sričiai, nebeužteko registro aprašų. Įtraukus naujus elementus į registro aprašus, tekdavo kurti naujas funkcijas, kurios sugebėtų suprasti ir atpažinti papildomai įtrauktus elementus. Šiuo atveju įvairios įmonės projektuoja, savus registrus, arba naudojami naujai sukurtais registro modeliais.

Nauji registrai būna išplėsti panaudojus UDDI struktūrą. Įtraukti nauji elementai, suprojektuojamos naujos API funkcijos darbui su naujais registrais. Tai pat pakito registro organizavimo, laikymo architektūra, vieni registrai saugomi XML sintaksės failais, kiti aprašomi tiesiog, kaip įrašai duomenų bazėje. Tokiuose registruose jau sudaromos galimybės projektuoti semantines paieškas, į naujus registro aprašo elementus įtraukiant kategorijas ir kitas reikalingas žymes.

Esamiems dabartiniams UDDI standarto registrams pritaikyti semantikas skirti specializuoti paketai. Anksčiau nagrinėtame skyriuje mes apžvelgėme semantikos išplėtimą UDDI registre ir WSDL. Tad iš esamų projektų verta paminėti dabar naudojamus modelius:

- OWL-S [11]
- SAWSDL [18]

2.9.1. OWL-S

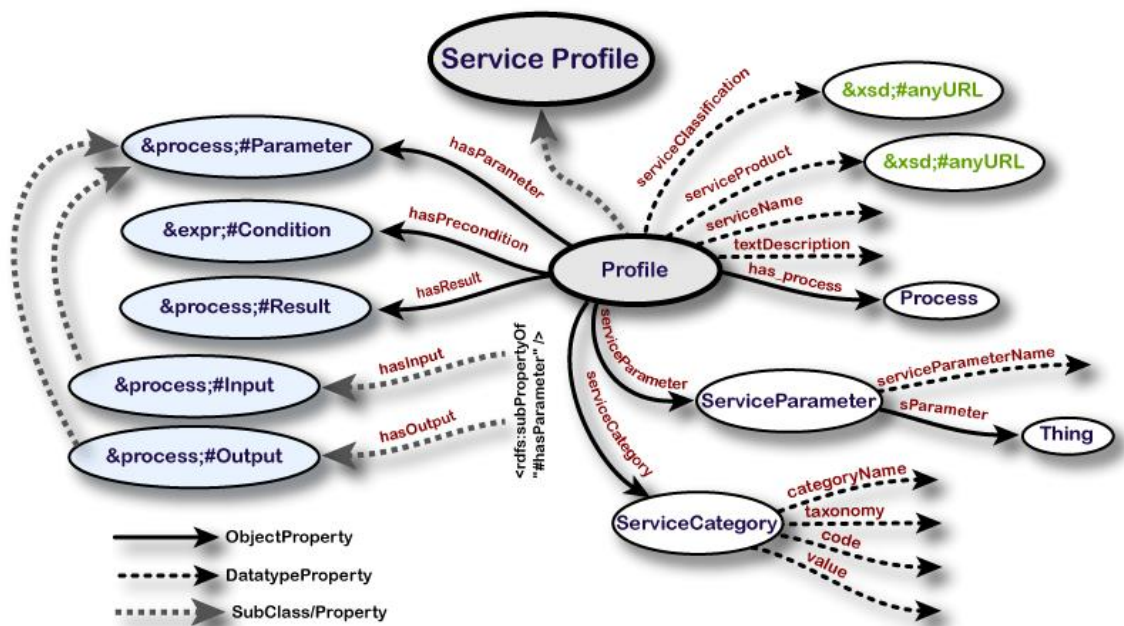
OWL-S – tai UDDI priedas orientuotas į ontologijos panaudojimą t.y. semantiką aprašant interneto paslaugas. Tai failas, parašytas XML sintakse, kuriame aprašomi OWL objektai. OWL-S yra integruojamas kartu su WSDL failu, kuris atsakingas už SOAP paketų mainus, servisų iškvietimą[12]. Galiausiai atliekamas UDDI registro susiejimas su OWL-S aprašais. Registrų semantikos siejimas nagrinėtas buvo ankstesniame 2.6 skyriuje.

OWL-S aprašo struktūra skirstoma į tris dalis:

- *service profile* – aprašo serviso informacija: pavadinimas, ką jis daro, internetinių paslaugų teikėjo informacija, kontaktai ir t.t.

- *process model* - aprašoma kaip klientas bendrauja su servisu: kokie išskviečiami metodai, įvedami parametrai.
- *service grounding* – žemiausias lygis, aprašomas serviso veikimo protokolai, pranešimų perdavimo formatai ir t.t.

Visos išvardintos dalys jau yra semantiškai aprašomos. Svarbiausioji dalis tai *service profile*. 6 paveikslėlyje pateiktos klasės ir jų savybių ryšiai esantys *service profile* dalyje. Ji atsakinga už informacijos pateikimą apie patį servisą. Ši dalis sviri, nes joje atsispindi struktūros reikalingos semantiškai paieškai [11].



13 pav. Klasių ir savybių ryšiai [5]

2.9.2. SAWSDL

Dažnai projektuojant semantinį registrą remiamasi WSDL aprašo išplėtimu. Tai prie papildomų elementų pridėdant naujus elementus, kurie tikėtų naudoti kaip registrus. Kaip žinome WSDL failas skirtas aprašyti tik funkcinis parametrus sistemos architektūros lygmenyje. Aprašomos serviso funkcijos, įvedimo išvedimo parametrai, duomenų mainų protokolai. Bet jis nepateikia jokių duomenų apie pačia paslaugą, tai pat nesuteikia jokių semantinių paieškos galimybių. WSDL semantinio aprašo išplėtimui buvo sukurtas SAWSDL paketas [18].

SAWSDL (angl. The Semantic Annotations for WSDL and XML Schema) – projektas realizuojantis esamų WSDL dokumentų aprašo semantinį išplėtimą. Pati SAWSDL aprašo dokumentacija nėra ontologinė, jis labai naudojamas kaip tarpinis aprašas. Jame

galima nurodyti įvairias ontologines struktūras, kurios aprašo WSDL faile paslaugas, operacijas ir parametrus. Semantika papildytas WSDL failas, neturi jokios įtakos vėliau įtraukiant, registruojant paslaugas paprastame registre. Tokiu būdu nereikia papildyti semantika standartinį veikiančią UDDI registrą.

Paieškos algoritmas analizuoja UDDI registrą ir atitinkamuose elementuose suranda WSDL aprašą. Jame aptinka operacijas, paslaugas aprašytas OWL ar RDF ontologijos forma. Tuo naudojantis galima atlikti semantinį panašumo nustatymą.

SAWSDL naudojimo pavyzdys

```
<wsdl:types>
  <xs:schema targetNamespace="http://www.w3.org/2002/ws/sawSDL/spec/wsdl/order#"
    elementFormDefault="qualified">
    <xs:element name="OrderRequest"

sawSDL:modelReference="http://www.w3.org/2002/ws/sawSDL/spec/ontology/purchaseorder#
OrderRequest"

sawSDL:loweringSchemaMapping="http://www.w3.org/2002/ws/sawSDL/spec/mapping/RDFOnt2R
equest.xml">
    <xs:complexType> ...
```

Operacijos aprašymas:

```
<wsdl:operation name="order" pattern="http://www.w3.org/ns/wsdl/in-out"
sawSDL:modelReference="http://www.w3.org/2002/ws/sawSDL/spec/ontology/purch
aseorder#RequestPurchaseOrder">
  <wsdl:input element="OrderRequest" />
  <wsdl:output element="OrderResponse" />
```

2.9.3. Semantinio registro modelių palyginimas

Sudėtinga lyginti modelius, nes jie skiriasi specifikacija ir architektūrine realizacija. Galimas bendras aprašymas tam tikrose kategorijose, aprašant plusus ir minusus. Palyginimui pateikiami 2.8 ir 2.9 skyriuje aprašyti modeliai:

- Microsoft registro modelis .NET paketui
- OWL-S paketas UDDI išplėtimui
- Registro modeliai išplečiant WSDL galimybes - SAWSDL

Palyginimas pagal:

Architektūrinį sprendimą. Microsoft registro modelis nuo likusių skiriasi savo realizavimu. Pats registras integruotas WCF modulį, bendras paslaugos aprašymas įterptas į bendrą architektūros veikimo aprašą. Tai pat registro elementai įrašomi kaip įrašai duomenų bazėje. Skirtingai nei likusiose, kurios registrų aprašymai pateikti XML sintakse aprašytais failais.

Registro aprašo realizacija. Visi jie naudoja UDDI registro aprašymą, kaip pamatinį variantą. Tik SAWSDL, kuris išplečiami pagal WSDL. Microsoft registrai saugomi duomenų bazėje. Patogus registru registravimas, redagavimas ir šalinimas, nes operacijų principai, tokie pat kaip ir naudojant paprastas SQL užklausas. Likusių aprašai, pateikiami XML sintakse. Tokių registru redagavimas, ar kitokios operacijos sudėtingesnės. Reikia kurti specializuotas funkcijas, kurios palengvintų tokių registru paiešką. Registrai pateikti kaip XML aprašai, lengviau integruojami į įvairias operacines platformas ir programavimo įrankius. Pvz.: WSDL aprašą gali suprasti Java, PHP, C++ ir visos kitos galimos kalbos, kurios turi XML sintaksės palaikymo funkcijų.

Registro struktūra. OWL-S ir WSDL išplečiamieji registro modeliai panašūs, mažai skiriasi elementų struktūra. .NET platformos struktūra, iš dalies panaši, kaip ir OWL-S ar WSDL, bet labiau specifinė. Pirmosios dvi naudoja įvairias aprašymo schemas, metodo iškvietimas – atskira struktūra, parametrai – dar kitoje. O .NET viena standartinę struktūra, kurioje viskas surašoma į vieną didelę struktūrą.

Semantika. Čia didesnes galimybes teikia OWL-S. Ji labiau orientuota į semantinę paiešką, jos aprašai labiau detalizuojami naudojantis didesnio OWL lygio logikas. Išplėstinis registras yra lankstus, jį galima lengvai paruošti semantikai, bet sudėtingesnė realizacija. Microsoft suteikia galimybes semantinei paieškai, bet nėra stipriau paruošta reiklesnei semantinei paieškai.

API. XML pagrindo modeliai naudoja specifines funkcijas darbui su registrais. Tai priklauso nuo realizuoto modelio galimybių. OWL-S registru Protégé [12], sukuria specialias funkcijas darbui su registru. Išplėstiniam WSDL modelių registrams, tai pat kuriamos funkcijos, kurios priklauso nuo registro modelio XML aprašo subtilybių. Paprastesnė Microsoft registro modelio realizacija. Jos funkcijų bibliotekos, paremtos duomenų bazės SQL paieškomis.

2.10. Analizės išvados

1. Išanalizavus interneto paslaugų architektūrą ir paslaugų paieškos poreikius, prieita išvados, kad tikslinga realizuoti semantinės paieškos priemones.

2. Išanalizavus semantinių registru kūrimo principus ir technologijas, nuspręsta pasirinkti paprasčiausią variantą – UDDI registro modelį su OWL išplėtimu. Semantinė paieška bus vykdoma pagal ieškomos operacijos įėjimų ir išėjimų parametrų semantinį sutapimą ar panašumą.

3. Kadangi paslaugų skaičius nuolat didėja, o didelius duomenų kiekius geriausia saugoti duomenų bazėse, nuspręsta realizuoti semantinį registrą reliacinėje duomenų bazėje.

4. Apžvelgus semantinės paieškos algoritmus, panašumų nustatymo metodus, semantinės paieškos algoritmui realizuoti buvo pasirinktas Greedy panašumų skaičiavimo metodas.

5. Registro valdymui ir semantinei paslaugų paieškai duomenų bazėje reikia suprojektuoti programines priemones registro sukūrimui, redagavimui ir semantikos priskyrimui, bei paieškos programą registre esančių paslaugų semantinės paieškos algoritmui tirti.

3. SEMANTINIO REGISTRO REIKALAVIMŲ SPECIFIKACIJA IR ANALIZĖ

3.1. Siekiamo registro modelio ir jo algoritmo tyrimo sprendimų apibrėžimas

Atlikus 2-oje dalyje semantinio registro, algoritmų analizę galime apibrėžti projektuojamas priemones šio magistrinio darbo tyrimui.

Visų pirma reikalingas semantinis registro modelis ir jo realizacija. Iš atliktos registru modelių analizės nuspręsta pasirinkti, standartinį UDDI registro modelį. Šį modelį supaprastinti ir išplėsti papildant semantiką aprašančiomis struktūromis. Suprastintas modelis plačiau apžvelgtas 3.4 skyriuje.

Toliau suprojektuotą modelį realizuoti duomenų bazėje. Duomenų bazės realizacijai pasirenkama MySQL duomenų bazė, dėl nesudėtingo jos valdymo ir naudojimo.

Analizės dalyje žinome, jog registru sistemoje dalyvauja du dalyviai:

- Vartotojas - atliekantis paiešką interneto paslaugų paiešką registruose;
- Administratorius – atliekantis registru valdymo funkcijas, paslaugų įtraukimo, šalinimo, aprašymo funkcijas.

Reikalinga realizuoti programas skirtą vartotojui ir administratoriui, kadangi atliekamas tyrimas tik ištirti semantinį modelį ir jame veikiančią semantinės paieškos algoritmą, tai visų galimų funkcijų nerealizuojama.

Vartotojo paieškos programa - užtenka turėti tokias funkcijas:

- Prisijungimo prie registro bazės realizuotoje duomenų bazėje
- Duomenų bazių valdymo funkcijas:(užklausų, rezultatų formavimas SQL sakiniiais)
- Tiriamojo semantinės paieškos algoritmo realizacija
- Atliktų paieškų rezultatų pateikimas programos sąsajoje
- Paieškos parametrų įvedimas

Administratoriaus programos galimos funkcijos:

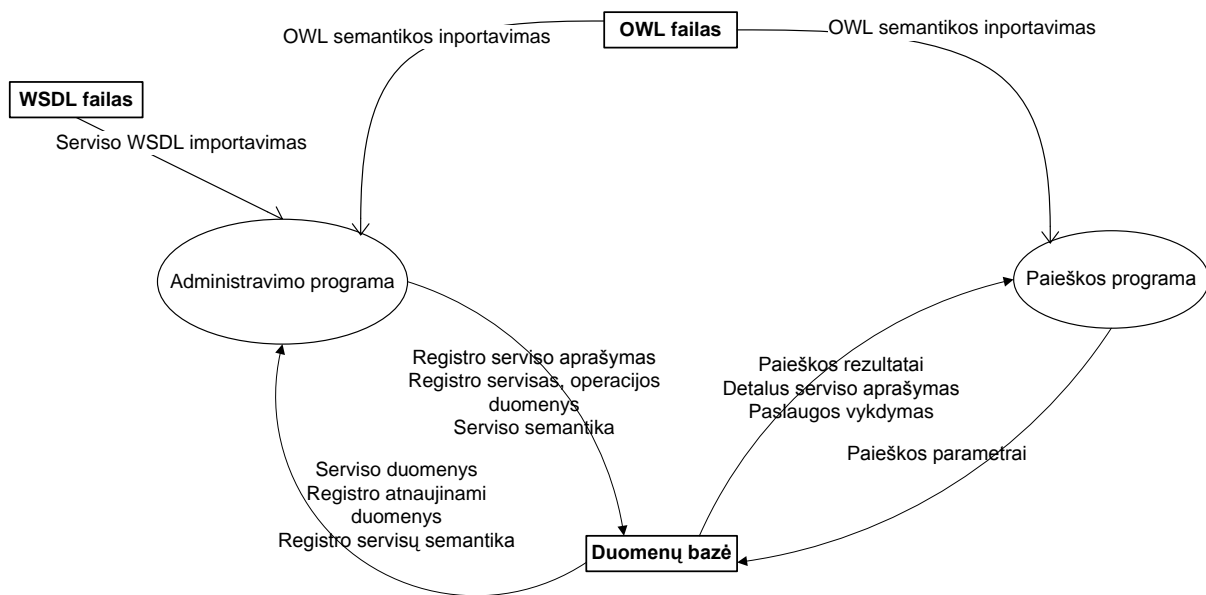
- Prisijungimas prie administruojamos registro realizuoto duomenų bazėje
- Registro valdymo funkcijas: sukurti registrą, įtraukti interneto paslaugas, aprašyti jas;
- Atlikti semantinius priskyrimus operacijos ir jų parametrams;
- Organizacijos aprašymo pildymas

- Registro bazės automatinį sukūrimą pasirinktoje duomenų bazėje t.y. sukurtų duomenų bazėje lenteles, ryšius duomenim saugoti. Tai reikalinga atliekant programų testus, eksperimentus, kai kas kart prireikia naujos registrų bazės.

Programų realizacija padaryta Java SE programavimo paketu - sukurtos dvi taikomosios programos. Paslaugų paieškos programa (vartotojui), registrų administravimo programa (administratoriui). Java kalba parinkta, nes tai nekomercinis paketas, turi daug papildomų funkcijų bibliotekų darbu su duomenų bazėmis ir SQL. Taip pat paketus skirtus dirbti WSDL ir OWL failais. J4WSDL – atlikti WSDL dokumento XML sintaksės struktūrų analizę, išrinkti duomenis. OpenOWL API - OWL paketas turi funkcijas ontologijų klasių lyginimo, paieškos ontologijos struktūrose.

3.2. Reikalavimų specifikacija

3.2.1. Veiklos kontekstas



14 pav. Veiklos konteksto schema

Veiklos įvykių sąrašas bei juose naudojami informacijos srautai pateikiami lentelėje 1.

3 lentelė

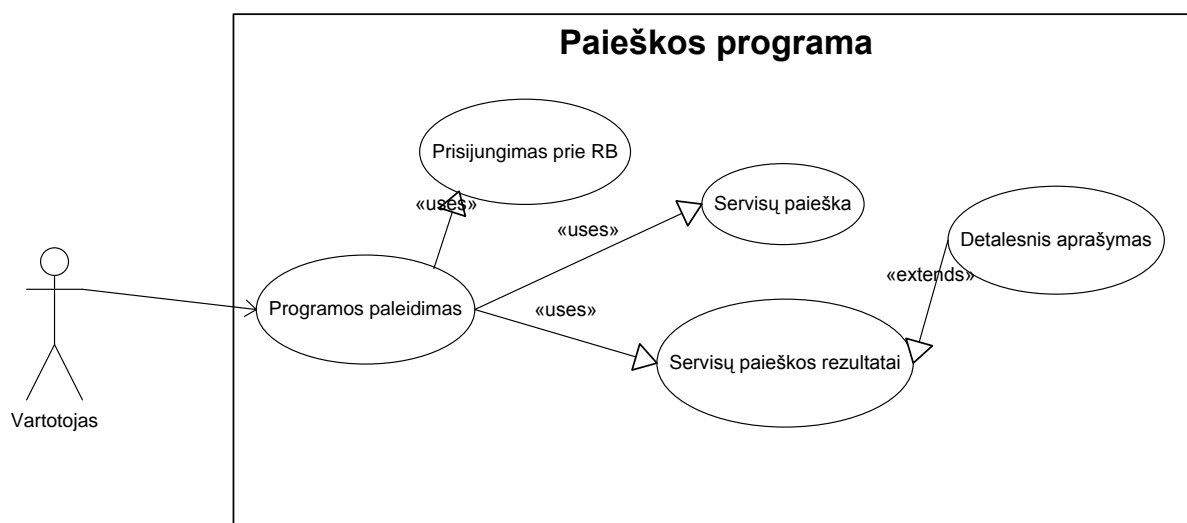
Veiklos įvykių sąrašas

Eil.nr.	Įvykio pavadinimas	Įeinantys/išeinantys informacijos srautai
1	Serviso WSDL importavimas	WSDL failas (in)

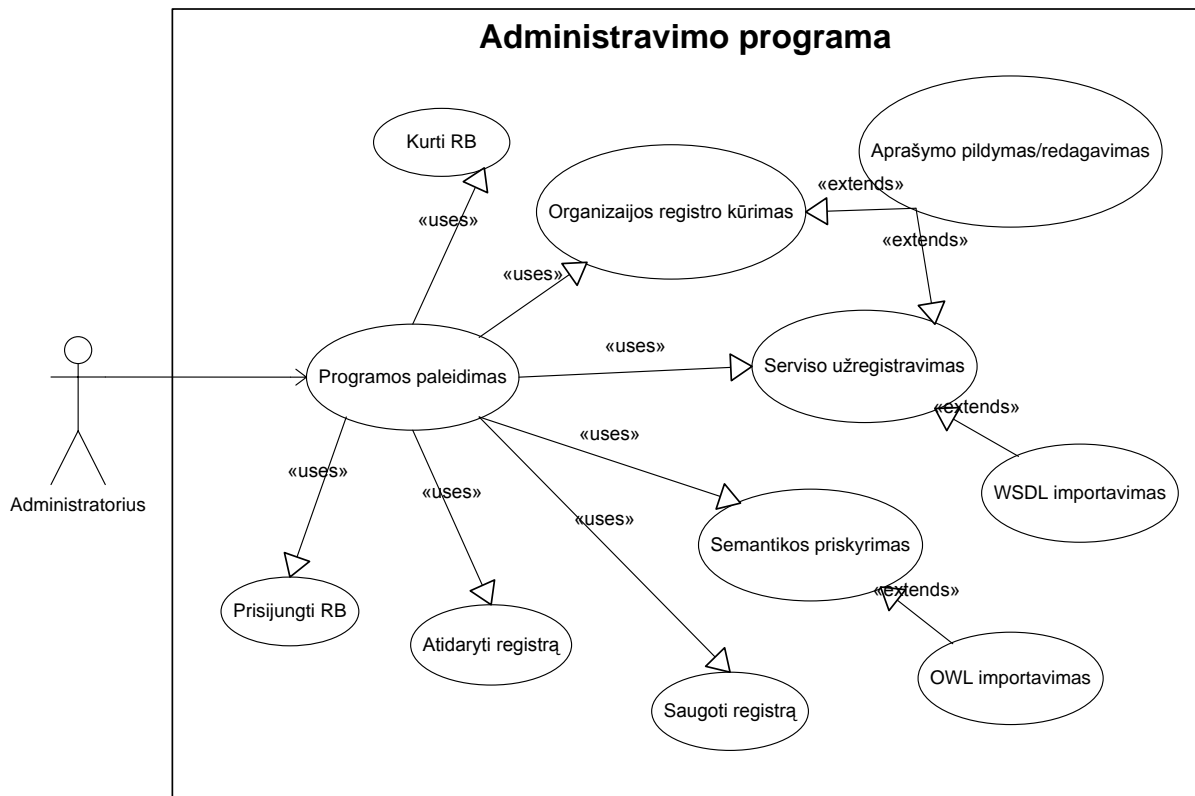
2	OWL semantikos importavimas	OWL failas (in)
3	Administravimas: registro serviso aprašymas	Tekstinė informacija (in)
4	Administravimas: serviso operacijų aprašymas	Tekstinė informacija (in)
5	Administravimas: ontologijos objektų(klasių) priskirimas serviso struktūros elementams: operacijoms, įėjimų ir išėjimų parametrms	Ontologijos objektai (klasės) (in) Operacijos(in) Įėjimų, išėjimų parametrai (in)
6	Administravimas: naujo registro (šablono) kūrimas	Registras(in)
7	Administravimas: registro saugojimas	Registro struktūra(in)
8	Administravimas: registro struktūros kūrimas reliacinėje DB	Duomenų bazė(in) Registro struktūra (out)
9	Paieška: semantinės paieškos vykdymas	Paieškos parametrai(in)
10	Paieška: paieškos rezultatų vaizdavimas	Paieškos rezultatai(out)

3.2.2. Panaudojimo atvejai

Paveikslėliuose 15 ir 16 pateikiami paieškos programos ir administravimo programos panaudojimo atvejų diagramos.



15 pav. Paieškos programos panaudojimo atvejų programa



16 pav. Administravimo programos panaudojimo atvejų diagrama

Pateikiami visų panaudojimo atvejų aprašymai.

1. PANAUDOJIMO ATVEJIS: Prisijungimas prie RB (registro bazės)

Vartotojas/Aktorius:	Vartotojas
Aprašas:	Procesas, kurio metu iškviečiamas prisijungimo prie RB dialogo langas. Vartotojas suveda prisijungimo duomenis, atliekama autorizacija. Po sėkmingo prisijungimo leidžiama toliau naudotis paieškos programa.
Prieš sąlyga:	Vartotojas pirmą kartą paleidžia paieškos programą
Sužadinimo sąlyga:	Suvedami duomenys ir atliekama autorizacija
Po-sąlyga:	Leidžiama dirbti toliau su paieškos programa

2. PANAUDOJIMO ATVEJIS: Internetinių paslaugų (servisų) paieška

Vartotojas/Aktorius:	Vartotojas
Aprašas:	Apima procesą, kurio metu vartotojas suveda paieškos parametrus. Toliau atliekama internetinių paslaugų operacijų paieška pagal įvestus duomenis
Prieš sąlyga:	Įvesti pradiniai paieškos parametrai
Sužadinimo sąlyga:	Vartotojas vykdo paieškos funkciją
Po-sąlyga:	Suformuojamas ir pateikiamas rastų paslaugų operacijų sąrašas su nustatytais semantiniais panašumų koeficientais.

3. PANAUDOJIMO ATVEJIS: Internetinių paslaugų (servisų) paieška

Vartotojas/Aktorius:	Vartotojas
Aprašas:	Apima procesą, kurio metu vartotojas suveda paieškos parametrus. Toliau atliekama internetinių paslaugų operacijų paieška pagal įvestus duomenis
Prieš sąlyga:	Įvesti pradiniai paieškos parametrai
Sužadinimo sąlyga:	Vartotojas vykdo paieškos funkciją
Po-sąlyga:	Suformuojamas rastų paslaugų operacijų sąrašas su nustatytais semantiniais panašumų koeficientais.

4. PANAUDOJIMO ATVEJIS: Internetinių paslaugų (servisų) paieškos rezultatai

Vartotojas/Aktorius:	Vartotojas
Aprašas:	Šios veiklos metu paieškos programoje pateikiami atliktos paieškos rezultatai.
Prieš sąlyga:	Turi būti atlikta semantinė paieška
Sužadinimo sąlyga:	Semantinės paieškos funkcijos pabaiga
Po-sąlyga:	Vartotojo paieškos programoje pateikiamas vaizdžiai rezultatų sąrašas

5. PANAUDOJIMO ATVEJIS: Detalesnis aprašymas

Vartotojas/Aktorius:	Vartotojas
Aprašas:	Papildomas procesas, kuris pateikia detalesnį paslaugos aprašymą iš rastų paieškos rezultatų
Prieš sąlyga:	Turi būti atlikta semantinė paieška ir pateikti rezultatai
Sužadinimo sąlyga:	Vartotojas pasirenka iš rezultatų sąrašo operaciją ir atlieka funkciją „Detalesnė informacija“
Po-sąlyga:	Pateikiama papildoma informaciją apie operacijos paslaugą

6. PANAUDOJIMO ATVEJIS: Kurti RB (registro bazę)

Vartotojas/Aktorius:	Administratorius
Aprašas:	Procesas, kurio metu administratorius gali sukurti naują registrų bazės struktūrą pasirinktoje reliacinėje duomenų bazėje. Duomenų bazėje sukuriama duomenų lentelės, ryšiai. Šie objektai atitinkamai realizuoja registro modelio struktūrą. Toliau sukurtą RB galima naudoti, vykdant kitus vėliau aprašytus panaudojimo atvejus.
Prieš sąlyga:	Administratorius įveda duomenų bazės prisijungimo duomenis, nurodo bazę, kurioje bus kuriamos lentelės.
Sužadinimo sąlyga:	Vykdoma funkcija „Kurti“
Po-sąlyga:	Pateikiamas pranešimas apie RB sukūrimo sėkmingumą.

7. PANAUDOJIMO ATVEJIS: Prisijungti prie RB (registro bazės)

Vartotojas/Aktorius:	Administratorius
Aprašas:	Procesas, kurio metu administratorius gali prisijungti prie sukurtos RB bazės ir toliau vykdytis administravimo veiksmus. Šis panaudojimo atvejis analogiškas 1 panaudojimo atvejui.
Prieš sąlyga:	Administratorius pateikia prisijungimo duomenis prie RB. RB turi būti jau sukurta.
Sužadinimo sąlyga:	Vykdoma funkcija „Prisijungti“.
Po-sąlyga:	Pateikiamas pranešimas apie prisijungimo sėkmingumą, aktyvuojamos papildomos administravimo funkcijos

8. PANAUDOJIMO ATVEJIS: Organizacijos registro kūrimas

Vartotojas/Aktorius:	Administratorius
Aprašas:	Procesas, kurio metu administratorius turi užregistruoti verslo organizaciją (įmonę, korporaciją), kuri teikia savo internetines paslaugas. Aprašomi įvairūs kontaktiniai duomenys, organizacijos trumpesnis ir detalesnis aprašymas ir t.t.
Prieš sąlyga:	Administratorius turi būti prisijungęs prie RB.
Sužadinimo sąlyga:	Vykdoma funkcija „Kurti registrą“.
Po-sąlyga:	Suformuojamas registro šablono karkasas, kurios „verslo organizacija“ dalyje galima įvesti aprašymo duomenis.

9. PANAUDOJIMO ATVEJIS: Interneto paslaugos(serviso) užregistravimas

Vartotojas/Aktorius:	Administratorius
Aprašas:	Procesas, kurio metu sukurtame registre, galima įtraukti organizacijos publikuojamus servisu. Įtraukiant servisą, nurodomas jo WSDL failo šaltinis, toliau administracinė programa išanalizuoja duomenis. Aptiktas operacijas, jų parametrus išsaugo ir pateikia vaizdžiai administratoriaus programoje.
Prieš sąlyga:	Administratorius turi būti prisijungęs prie RB ir sukūręs/atidaręs registrą. Pateikiamas serviso WSDL šaltinio adresas.
Sužadinimo sąlyga:	Vykdoma funkcija „Įtraukti servisą“.
Po-sąlyga:	Aptiktas operacijas, jų parametrus išsaugo ir pateikia vaizdžiai administratoriaus programoje

10. PANAUDOJIMO ATVEJIS: Aprašymo pildymas/redagavimas

Vartotojas/Aktorius:	Administratorius
Aprašas:	Procesas, kurio metu administratorius gali atlikti registro tekstinių aprašymų apie organizaciją ar serviso dalis, papildymą/redagavimą.
Prieš sąlyga:	Administratorius turi būti prisijungęs prie RB ir sukūręs/atidaręs registrą.
Sužadinimo sąlyga:	Vykdoma funkcija redaguoti.
Po-sąlyga:	Atlikti tekstiniai aprašymų pakeitimai

11. PANAUDOJIMO ATVEJIS: WSDL importavimas

Vartotojas/Aktorius:	Administratorius
Aprašas:	Procesas, kurio metu administratorius nurodo internetinės paslaugos(serviso) WSDL failo interneto adresą. Toliau atliekamas šio failo struktūros analizė, kur randama informacija apie servisą, jo turimas operacijas. Kiekvienos operacijos įėjimų ir išėjimų parametrus: skaičių, parametro tipą.
Prieš sąlyga:	Administratorius turi būti prisijungęs prie RB ir sukūręs/atidaręs registrą.
Sužadinimo sąlyga:	Kai vykdoma serviso užregistravimo panaudojimo atvejis
Po-sąlyga:	Rastų objektų: operacijos, parametrų įtraukimas į registrą

12. PANAUDOJIMO ATVEJIS: OWL importavimas

Vartotojas/Aktorius:	Administratorius
Aprašas:	Procesas, kurio metu administratorius nurodo ontologiją, kurioje aprašyti semantiniai objektai(klasės), kurios vėliau bus siejamos su paslaugos operacijom, parametrais.
Prieš sąlyga:	Administratorius turi būti prisijungęs prie RB ir sukūręs/atidaręs registrą.
Sužadinimo sąlyga:	Kai vykdomas semantikos priskyrimo panaudojimo atvejis
Po-sąlyga:	Importuoja ir paruošia objektus(klases) priskyrimo atvejui

13. PANAUDOJIMO ATVEJIS: Semantikos priskyrimas

Vartotojas/Aktorius:	Administratorius
Aprašas:	Procesas, kurio metu administratorius atlieka semantinių objektų(klasių) susiejimą su atitinkamomis operacijos, įėjimų ir išėjimų parametrais.
Prieš sąlyga:	Administratorius turi būti prisijungęs prie RB, sukūręs/atidaręs registrą ir turi įtrauktus servigus.
Sužadinimo sąlyga:	Atliekami priskyrimo veiksmai
Po-sąlyga:	Išsaugomi registre susieti objektai

14. PANAUDOJIMO ATVEJIS: Atidaryti registrą

Vartotojas/Aktorius:	Administratorius
Aprašas:	Procesas, kurio metu administratorius gali atidaryti esamus registrus iš RB (Registru bazės) ir redaguoti, pakeisti norimas registro dalis: verslo organizaciją, įtrauktas paslaugas ar semantinius susiejimus
Prieš sąlyga:	Administratorius turi būti prisijungęs prie RB.
Sužadinimo sąlyga:	Atliekama atidarymo funkcija
Po-sąlyga:	Atveriamas registro karkasas su užpildytais esamais duomenimis

15. PANAUDOJIMO ATVEJIS: Saugoti registrą

Vartotojas/Aktorius:	Administratorius
Aprašas:	Procesas, kurio metu administratorius atlieka registro saugojimą t.y. registro įrašymą į RB (Registro bazę). Atitinkamų duomenų bazės lentelių užpildymus
Prieš sąlyga:	Administratorius turi būti prisijungęs prie RB, sukūręs/atidaręs registrą ir atlikęs veiksmus su juo.
Sužadinimo sąlyga:	Vykdoma „saugojimo“ funkcija
Po-sąlyga:	Registras išsaugomas RB duomenų bazėje.

3.2.3. Funkciniai reikalavimai

<u>Reikalavimas #:</u>	1	<u>Reikalavimo tipas:</u>	1	<u>Ivykis/panaudojimo atvejis #:</u>	1/1
<u>Aprašymas:</u>	Vartotojas turi įvykdyti prisijungimo autorizaciją prie RB (Registro bazės).				
<u>Pagrindimas:</u>	Nepavykus autorizacijos, negalimas tolimesnis programos naudojimas.				
<u>Šaltinis:</u>	Užsakovas				
<u>Tikimo kriterijus:</u>	Teisingai atlikta autorizacija				
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5		
<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	1-5		
<u>Papildoma medžiaga:</u>					
<u>Istorija:</u>	Užregistruotas 2009 balandžio 21 d.				

<u>Reikalavimas #:</u>	2	<u>Reikalavimo tipas:</u>	2	<u>Ivykis/panaudojimo atvejis #:</u>	1/2
<u>Aprašymas:</u>	Vartotojas turi suvesti paieškos programoje paieškos parametrus: ontologijos objektus(klases). Parametrams įvesti skiriamos trys dalys: funkcijos(operacijos) ontologinis objektams, įėjimų objektams, išėjimų objektams.				
<u>Pagrindimas:</u>	Paieškos vykdymui, turi būti užtikrinta, kad bent viena iš trejų įvedimo dalių yra užpildyta.				
<u>Šaltinis:</u>	Užsakovas				
<u>Tikimo kriterijus:</u>	Paieškai pateikiamas bent vienas ieškomasis parametras.				
<u>Užsakovo tenkinimas:</u>	4	<u>Užsakovo netenkinimas:</u>	4		
<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	Nėra		
<u>Papildoma medžiaga:</u>					
<u>Istorija:</u>	Užregistruotas 2009 balandžio 21 d.				

<u>Reikalavimas #:</u>	3	<u>Reikalavimo tipas:</u>	2	<u>Ivykis/panaudojimo atvejis #:</u>	2/2
------------------------	---	---------------------------	---	--------------------------------------	-----

<u>Aprašymas:</u>	Vartotojas turi suvesti paieškos programoje paieškos parametrus: ontologijos objektus(klases). Parametrams įvesti skiriamos trys dalys: funkcijos(operacijos) ontologinis objektams, įėjimų objektams, išėjimų objektams.		
<u>Pagrindimas:</u>	Kadangi parametrai – tai semantinės klasės ir jie dažnai pateikiami iš didelių ontologijų, vartotojas gali nežinoti norimo parametro klasės pavadinimą. Reikalinga sukurti papildoma funkciją, kuri iškvietu galimų klasių pasirinkimų variantus		
<u>Šaltinis:</u>	Užsakovas		
<u>Tikimo kriterijus:</u>	Sudaromos galimybės iškviesti ontologijos klasių pasirinkimo galimybę.		
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5
<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	Nėra
<u>Papildoma medžiaga:</u>			
<u>Istorija:</u>	Užregistruotas 2009 balandžio 21 d.		

<u>Reikalavimas #:</u>	4	<u>Reikalavimo tipas:</u>	2	<u>Ivykis/panaudojimo atvejis #:</u>	3/2
<u>Aprašymas:</u>	Vartotojas turi suvesti paieškos programoje paieškos parametrus: ontologijos objektus(klases). Parametrams įvesti skiriamos trys dalys: funkcijos(operacijos) ontologinis objektams, įėjimų objektams, išėjimų objektams.				
<u>Pagrindimas:</u>	Reikalinga stebėti vartotojo įvedus parametrus: parametrų klasės pavadinimai negali būti didelių ilgių, negali būti specifinių simbolių ir t.t.				
<u>Šaltinis:</u>	Užsakovas				
<u>Tikimo kriterijus:</u>	Atliekamas įvestų parametrų duomenų patikrinimas prieš paiešką				
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5		
<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	Nėra		
<u>Papildoma medžiaga:</u>					
<u>Istorija:</u>	Užregistruotas 2009 balandžio 21 d.				

<u>Reikalavimas #:</u>	5	<u>Reikalavimo tipas:</u>	3	<u>Ivykis/panaudojimo atvejis #:</u>	1/4
<u>Aprašymas:</u>	Po atliktos paieškos rezultatų pateikimas turi atitikti vartotojo pasirinkimą. Rezultatų sąrašą rikiuoti pagal operacijų pavadinimas, servisų pavadinimą, koeficientus.				
<u>Pagrindimas:</u>	Reikalinga suteikti galimybę vartotojui pasirinkti rezultatų sąrašo išdėstymą pagal norimus raktus.				
<u>Šaltinis:</u>	Užsakovas				

<u>Tikimo kriterijus:</u>	Galimybė pasirinkti sąrašo formavimą		
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5
<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	Nėra
<u>Papildoma medžiaga:</u>			
<u>Istorija:</u>	Užregistruotas 2009 balandžio 21 d.		

<u>Reikalavimas #:</u>	6	<u>Reikalavimo tipas:</u>	4	<u>Ivykis/panaudojimo atvejis #:</u>	1/6
<u>Aprašymas:</u>	Kuriant registrų bazę (RB) reliacinės duomenų bazės pagrindu, administratorius suveda prisijungimo duomenis, nurodo bazės pavadinimą, kurioje formuoti registro struktūros lentelės. Problema iškyla, jei administratorius kuriama RB jau yra naudojamoje bazėje t.y. joje jau yra prikurti lentelių ir taip toliau.				
<u>Pagrindimas:</u>	Turi būti suteikta galimybė projektuoti sekančias RB vienoje duomenų bazėje. Tai galima suteikiant papildomą žymę formuojant lentelės: pridėdant papildoma administratoriaus įvestą priesagą prie lentelių.				
<u>Šaltinis:</u>	Užsakovas				
<u>Tikimo kriterijus:</u>	Galimybė įvesti papildomus atributus, kurie būtų automatiškai priskiriami kuriant lentelės.				
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5		
<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	Nėra		
<u>Papildoma medžiaga:</u>					
<u>Istorija:</u>	Užregistruotas 2009 balandžio 21 d.				

<u>Reikalavimas #:</u>	7	<u>Reikalavimo tipas:</u>	5	<u>Ivykis/panaudojimo atvejis #:</u>	1/8
<u>Aprašymas:</u>	Kuriant naują registrą reikalinga administratoriui pateikti registro karkaso šabloną t.y. pateikti skyriai su tuščiais laukeliais, kuriuos vėliau turi užpildyti. Organizacijos skyriuje administratorius suveda įvairaus tipo duomenis: tekstinius aprašymus, kontaktinę informaciją (telefono nr. ,el. paštas, organizacijos būstinės adresas)				
<u>Pagrindimas:</u>	Siekiant, kad informacinis organizacijos aprašymas būtų tikslus ir aiškus. Reikalinga patikrinti įvairių duomenų korektiškumą, pvz.: el. paštas privalo turėti @ simbolį arba telefono nr. turi susidaryti iš tam tikro skaičiaus kiekio.				
<u>Šaltinis:</u>	Užsakovas				
<u>Tikimo kriterijus:</u>	Atliekama informacinių duomenų patikra				
<u>Užsakovo tenkinimas:</u>	4	<u>Užsakovo netenkinimas:</u>	5		
<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	Nėra		
<u>Papildoma medžiaga:</u>					

Istorija: Užregistruotas 2009 balandžio 21 d.

Reikalavimas #: 8 Reikalavimo tipas: 6 Ivykis/panaudojimo atvejis #: 1/11

Aprašymas: Importuojant interneto paslaugos WSDL failą iš administratoriaus šaltinio, būtinas jo administravimo programos palaikymo nustatymas.

Pagrindimas: Yra įvairių variantų WSDL failų struktūros, tam kad tiksliai ir kokybiškai galima būtų išanalizuoti reikalinga nustatyti WSDL tipą. Tai administratoriaus programa gali žinoti, koku būdu vykdyti WSDL failo skaitymą.

Šaltinis: Užsakovas

Tikimo kriterijus: Atliekamas WSDL struktūros tipo nustatymas

Užsakovo tenkinimas: 5 Užsakovo netenkinimas: 5

Priklausomybės: Nėra Konfliktai: Nėra

Papildoma medžiaga:

Istorija: Užregistruotas 2009 balandžio 21 d.

Reikalavimas #: 9 Reikalavimo tipas: 6 Ivykis/panaudojimo atvejis #: 1/12

Aprašymas: Importuojant OWL ontologiją, būtinas jo patikrinimas, ar teisinga OWL struktūra. Jei struktūra neteisinga, programa negali kokybiškai ir tiksliai dirbti, atlikti funkcijas su semantika, ontologijos klasėmis.

Pagrindimas: Jei struktūra neteisinga, programa negali kokybiškai ir tiksliai dirbti: atlikti funkcijas su semantika, ontologijos klasėmis.

Šaltinis: Užsakovas

Tikimo kriterijus: Atliekamas OWL struktūros patikrinimas

Užsakovo tenkinimas: 5 Užsakovo netenkinimas: 5

Priklausomybės: Nėra Konfliktai: Nėra

Papildoma medžiaga:

Istorija: Užregistruotas 2009 balandžio 21 d.

Reikalavimas #: 10 Reikalavimo tipas: 7 Ivykis/panaudojimo atvejis #: 1/13

Aprašymas: Atliekant semantinių priskyrimo veiksmus registre turi būti užregistruota bent viena internetinė paslauga(servisas). Tai pat priskiriant semantines klases reikia atlikti visus priskyrimus su visomis serviso operacijomis, parametrais.

Pagrindimas: Reikalingas patikrinimas, ar visi serviso elementai susisieti su pasirinktomis semantinėmis klasėmis

<u>Šaltinis:</u>	Užsakovas		
<u>Tikimo kriterijus:</u>	Tikrinimas, ar pilnai susietos visos struktūros su semantiniiais objektais(klasėmis).		
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5
<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	Nėra
<u>Papildoma medžiaga:</u>			
<u>Istorija:</u>	Užregistruotas 2009 balandžio 21 d.		

<u>Reikalavimas #:</u>	11	<u>Reikalavimo tipas:</u>	8	<u>Ivykis/panaudojimo atvejis #:</u>	1/15
<u>Aprašymas:</u>	Atlikus visas registro dalis vykdomas saugojimo veiksmas. Jo metu visose registro struktūros dalyse esanti informacija surašoma į registrų bazę, kuri realizuota reliacinės duomenų bazės pagrindu. Vienu vyksta saugojimas visų dalių.				
<u>Pagrindimas:</u>	Kadangi vyksta visų dalių saugojimas, reikalinga patikrinti prieš saugojimą, ar bent organizacinė dalis aprašyta. Tai pat semantikos priskyrimo dalis negali būti atliekama, jei nėra užpildyta servisų dalis t.y. nėra jokių įtrauktų paslaugų.				
<u>Šaltinis:</u>	Užsakovas				
<u>Tikimo kriterijus:</u>	Atliekamas tikrinimas, ar bent organizacijos dalis užpildyta.				
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5		
<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	Nėra		
<u>Papildoma medžiaga:</u>					
<u>Istorija:</u>	Užregistruotas 2009 balandžio 21 d.				

3.3. Nefunkciniai reikalavimai

3.3.1. Reikalavimai saugumui

<u>Reikalavimas #:</u>	12	<u>Reikalavimo tipas:</u>	9	<u>Ivykis/panaudojimo atvejis #:</u>	1/1-5
<u>Aprašymas:</u>	Vartotojo paieškos programai nekeliama aukšti saugumo reikalavimai.				
<u>Pagrindimas:</u>	Vartotojas atlieka tik prisijungimą prie RB bazės, ir kelias vykdo kelias pagrindines funkcijas su duomenų baze. Šios funkcijos skirtos paimti duomenis iš bazės. Tada paprastam veikimui pakanka ir paprastos vartotojo autorizacijos.				
<u>Šaltinis:</u>	Užsakovas				
<u>Tikimo kriterijus:</u>	Pakankamas saugumas – standartinė vartotojo autorizacija				
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5		

<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	Nėra
<u>Papildoma medžiaga:</u>			
<u>Istorija:</u>	Užregistruotas 2009 balandžio 21 d.		

<u>Reikalavimas #:</u>	13	<u>Reikalavimo tipas:</u>	9	<u>Ivykis/panaudojimo atvejis #:</u>	1/6-15
<u>Aprašymas:</u>	Administravimo programai keliami didesni saugumo reikalavimai negu paieškos programai				
<u>Pagrindimas:</u>	Administratorius skirtingai negu vartotojas dirba su visomis duomenų bazės funkcijomis t.y. atliekant registrų įrašymus (saugojimą į lenteles), ar modifikavimą – atnaujinimą. Iš to seka, kad registro administravimo dalis turi labiau būti užtikrinta saugumo požiūriu. Administratoriaus prisijungimo prie RB autorizacija turi būti aukštesnio saugumo lygio.				
<u>Šaltinis:</u>	Užsakovas				
<u>Tikimo kriterijus:</u>	Reikalingas aukštesnio lygio saugumo lygis.				
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5		
<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	Nėra		
<u>Papildoma medžiaga:</u>					
<u>Istorija:</u>	Užregistruotas 2009 balandžio 21 d.				

3.3.2. Reikalavimai sąsajai

<u>Reikalavimas #:</u>	14	<u>Reikalavimo tipas:</u>	10	<u>Ivykis/panaudojimo atvejis #:</u>	1/1-5
<u>Aprašymas:</u>	Paieškos programa vykdo semantinę paiešką paslaugų registro bazėje. Paieškos parametrai yra ontologijos klasės. Paieškai pateikiami trejų tipų parametrų įvedimai: operacijos, įėjimai, išėjimai. Pačioje programoje formuojamos programos rezultatų pateikimas.				
<u>Pagrindimas:</u>	Dėl patogumo vartotojui sudaroma sąsaja, kurios viršutinėje dalyje paieškos įvedimo laukai, žemiau pateikiamas rezultatų sąrašas. Sąrašas pateikiamas lentelės pavidalu, kur stulpeliuose pateikti paslaugos duomenys.				
<u>Šaltinis:</u>	Užsakovas				
<u>Tikimo kriterijus:</u>	Paruoša bendra paieškos laukų ir rezultatų išvedimo forma				
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5		
<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	Nėra		
<u>Papildoma medžiaga:</u>					
<u>Istorija:</u>	Užregistruotas 2009 balandžio 21 d.				

<u>Reikalavimas #:</u>	15	<u>Reikalavimo tipas:</u>	10	<u>Ivykis/panaudojimo atvejis #:</u>	1/6-15
------------------------	----	---------------------------	----	--------------------------------------	--------

<u>Aprašymas:</u>	Registras kūrimas sudarytas iš trijų dalių: informacijos apie paslaugos teikėją organizacijos aprašymas, paslaugų(servisų) įtraukimas į registrą ir semantikos priskirimas.		
<u>Pagrindimas:</u>	Minėtas tris dalis galima susieti bendroje sąsajoje. Kai kuriamas registras, šios dalys turi būti suformuotos ir pateiktos administratoriui, kaip šablonas.		
<u>Šaltinis:</u>	Užsakovas		
<u>Tikimo kriterijus:</u>	Paruošiamas registro šablonas kuriant naują registrą.		
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5
<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	Nėra
<u>Papildoma medžiaga:</u>			
<u>Istorija:</u>	Užregistruotas 2009 balandžio 21 d.		

<u>Reikalavimas #:</u>	16	<u>Reikalavimo tipas:</u>	10	<u>Įvykis/panaudojimo atvejis #:</u>	1/6-15
<u>Aprašymas:</u>	Semantikos priskyrimo dalyje reikalingas patogus paslaugos(serviso) struktūros išdėstymas ir OWL ontologijos klasių pateikimas. To sekoje administratorius vaizdžiai matydamas formos objektus gali atlikti priskyrimo veiksmus.				
<u>Pagrindimas:</u>	Struktūros išdėstymas patogiausiais yra atliekamas medžio principu. Pagrindiniai objektai atstoja medžio šakas, toliau išeinantys objektai – pošakiai.				
<u>Šaltinis:</u>	Užsakovas				
<u>Tikimo kriterijus:</u>	Paslaugos struktūros ir ontologijos klasių vaizdavimas medžio principu.				
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5		
<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	Nėra		
<u>Papildoma medžiaga:</u>					
<u>Istorija:</u>	Užregistruotas 2009 balandžio 21 d.				

3.3.3. Reikalavimai panaudojamumui

<u>Reikalavimas #:</u>	17	<u>Reikalavimo tipas:</u>	11	<u>Įvykis/panaudojimo atvejis #:</u>	1/1-5
<u>Aprašymas:</u>	Dirbant su paieškos programa, vartotojams turi būti keliami minimalūs kompiuterinių žinių reikalavimai				
<u>Pagrindimas:</u>	Vartotojai turi mokėti dirbti su operacine sistema. Jei taikomoji programa su grafine sąsaja – reikalingos minimalios žinios (menu supratimas, programos langų uždarymas atidarymas). Jei sąsaja paremta interneto naršykle – minimalios žinios dirbant su interneto puslapiais.				
<u>Šaltinis:</u>	Užsakovas				
<u>Tikimo kriterijus:</u>	Nereikalaujama sudėtingų žinių.				

<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5
<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	Nėra
<u>Papildoma medžiaga:</u>			
<u>Istorija:</u>	Užregistruotas 2009 balandžio 21 d.		

<u>Reikalavimas #:</u>	18	<u>Reikalavimo tipas:</u>	11	<u>Ivykis/panaudojimo atvejis #:</u>	1/6-15
<u>Aprašymas:</u>	Dirbant su administravimo programa, administratoriams turi būti keliami didesni negu vidutiniai kompiuterinių žinių reikalavimai				
<u>Pagrindimas:</u>	Administratorius turi išmanyti darbą su serverio sistemomis, duomenų bazėmis. Turėti žinių apie interneto paslaugų technologijas ir jų naudojimą.				
<u>Šaltinis:</u>	Užsakovas				
<u>Tikimo kriterijus:</u>	Reikalaujama sudėtingų žinių.				
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5		
<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	Nėra		
<u>Papildoma medžiaga:</u>					
<u>Istorija:</u>	Užregistruotas 2009 balandžio 21 d.				

3.3.4. Reikalavimai vykdymo charakteristikoms

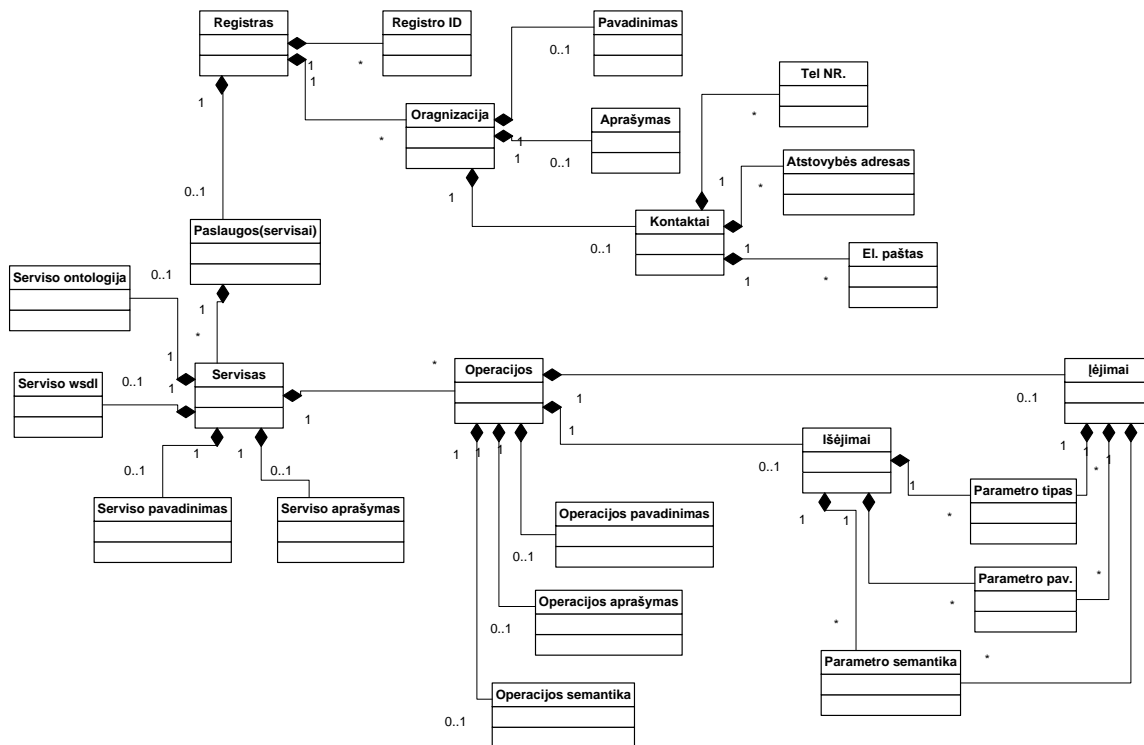
<u>Reikalavimas #:</u>	19	<u>Reikalavimo tipas:</u>	12	<u>Ivykis/panaudojimo atvejis #:</u>	1/1-5
<u>Aprašymas:</u>	Semantinės paieškos programa turi išlaikyti minimalius reikalavimus interneto ryšiui.				
<u>Pagrindimas:</u>	Reikalinga tam, kad darbas su sistema būtų efektyvus				
<u>Šaltinis:</u>	Užsakovas				
<u>Tikimo kriterijus:</u>	Programos darbas su internetu turi užtrukti kuo mažiau laiko.				
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5		
<u>Priklausomybės:</u>	Nėra	<u>Konfliktai:</u>	Nėra		
<u>Papildoma medžiaga:</u>					
<u>Istorija:</u>	Užregistruotas 2009 balandžio 21 d.				

3.4. Semantinio registro duomenų modelis

Semantinio registro duomenų modelis formuojamas remiantis UDDI registro duomenų modeliu. Toks modelis yra didelės apimties ir dauguma duomenų struktūrų yra nereikalingos.

Atsisakyta UDDI registro techninės dalys: *binding*, *template* ir.t.t. Svarbiausios šio modelio dalys yra organizacijos aprašymas, interneto paslaugų operacijos ir parametrai. Pastaroji dalis siejama su ontologijos klasėmis. Apsibrėžus svarbiausias dalis galima supaprastinti UDDI modelį ir pertvarkius pritaikyti realizacijai duomenų bazėje.

Suprastintas semantinis duomenų modelis pateiktas paveikslėlyje 17.



17 pav. Semantinio registro duomenų modelis

Iš 17 paveikslėlio matome registro duomenų modelio išsidėstymą. Pagrindinė klasė tai *Registras*. Šiai klasei priklauso organizacijos informacijos poklases: *Pavadinimas*, *Aprašymas*, *Kontaktai*. Minėtos klasės gali ir nebūti, realiame naudojime registro egzistavimui užtenka ir registro identifikacijos numerio sukūrimo.

Toliau iš *Registro* klasė privalo turėti *Paslaugos(servisai)* klasę. Šie duomenis išsiskirsto atitinkamai: *paslaugos pavadinimas*, *aprašymas* ir *operacijos*, *serviso wsdl*, *serviso ontologija*. *Serviso wsdl* klasė nurodo paslaugos WSDL failą, *serviso ontologija* – nurodo ontologiją, kurioje yra klasės susietos su paslaugoje esančiomis operacijomis ir parametrais.

Operacijos turi penkis duomenų tipus: *operacijos aprašymą*, *operacijos pavadinimas*, *iėjimai* ir *išėjimai*. Penktasis – *operacijos semantika*, nusako ontologijos klasę ar kitą semantišką objektą. Vykdamt administracinėje programoje semantikos priskyrimus, minėtoje

dalyje išsaugomas ontologijos klasės pavadinimas ar kitas žymuo taip priskiriamas operacijai atitinkamas semantikos požymis iš tam tikros srities ontologijos.

Skaidant operacijas gaunami įėjimų ir išėjimų duomenų struktūros. Parametro pavadinimas – nusako operacijoje esančio parametro kintamąjį. Parametro tipas – minėto kintamojo duomenų tipą: *string*, *integer* ir t.t.

Analogiškai tiek įėjimams ir išėjimams kaip ir operacijai – semantikos nusakymui esanti duomenų klasė *Parametro semantika*.

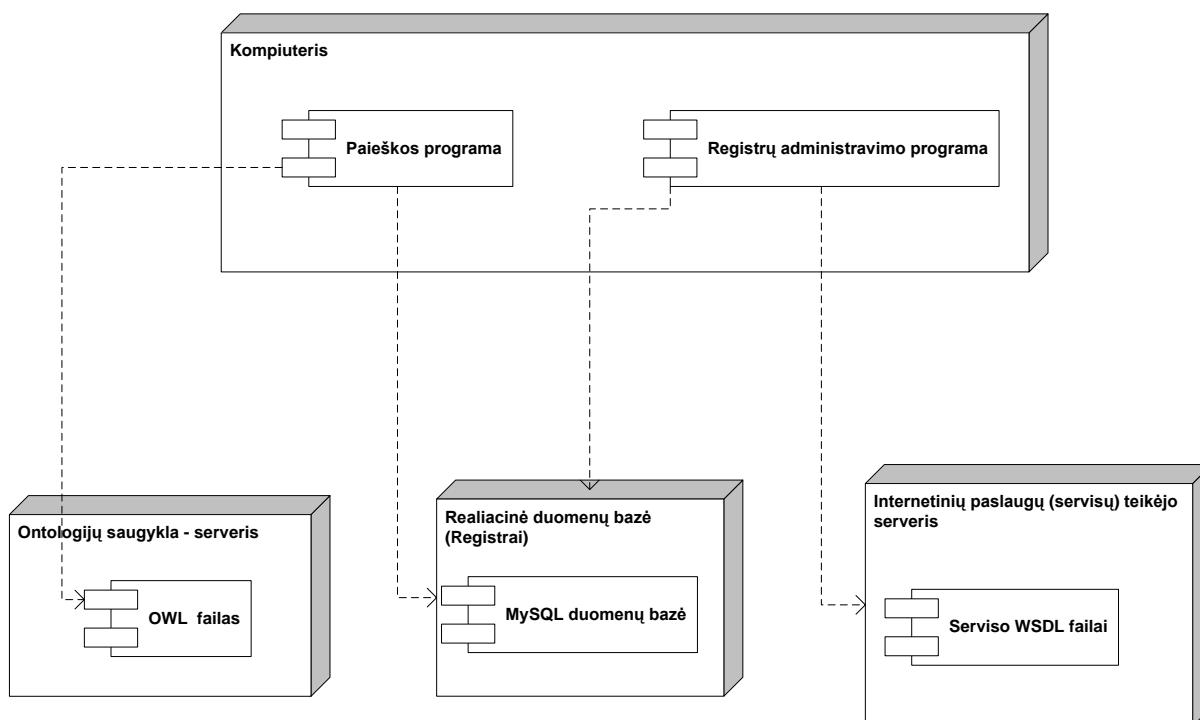
Pasinaudojus šiuo modeliu realizuojame semantinio registro struktūrą Duomenų bazėje.

3.5. Reikalavimų analizės išvados

1. Magistro darbo tyrimui aprašytos realizuojamos priemonių specifikacijos.
2. Specifikuotas supaprastintas UDDI pagrindu semantinis registro modelis, kuris realizuojamas MySQL duomenų bazėje.
3. Darbui su registro baze kuriamos dvi programos: vartotojo paieškos programa semantiniame registre ir administratoriaus registrų valdymo programa.
4. Programos buvo specifikuotos, nustatyti panaudojimo atvejai, iškelti ir išanalizuoti reikalavimai.
5. Aprašytų specifikacijų pakanka sukurti priemonės semantinės paieškos algoritmo tyrimui registre, kuris realizuotas duomenų bazėje

4. SEMANTINIO REGISTRO PROGRAMŲ ARCHITEKTŪRA

4.1. Bendra komponentų išsidėstymo schema



18 pav. Bendra komponentų išsidėstymo schema

Paveikslėlyje 18 parodyta bendra projekto išsidėstymo schema. Paieškos interneto paslaugų programa veikia vartotojo kompiuteryje, administratoriaus kompiuteryje – registų administravimo programa. Programos bendrai naudojami realiacine duomenų baze, kurioje imituojama registro bazė.

Paieškos programa atlieka ontologijos importavimą iš ontologijos saugyklos. Kadangi ontologijos saugyklos tipas nėra apibrėžtas (gali būti ftp, http serveris), laikome kad OWL failas pasiekiamas HTTP protokolu iš mums žinomo internetinio puslapio serverio.

Administravimo programa papildomai bendrauja su internetinių paslaugų serveriu, kurioje talpinami veikiančios paslaugos(servisai). Kiekviena paslauga turi atitinkamą savo WSDL failą, kuris reikalingas norint į registrą įtraukti paslaugas. Todėl reikalinga importuoti WSDL failą iš žinomo internetinių paslaugų teikėjo serverio.

Toliau sekančiuose skyriuose pateikiama detalesnė kiekvieno komponento architektūra.

4.2. Projekto loginė architektūra

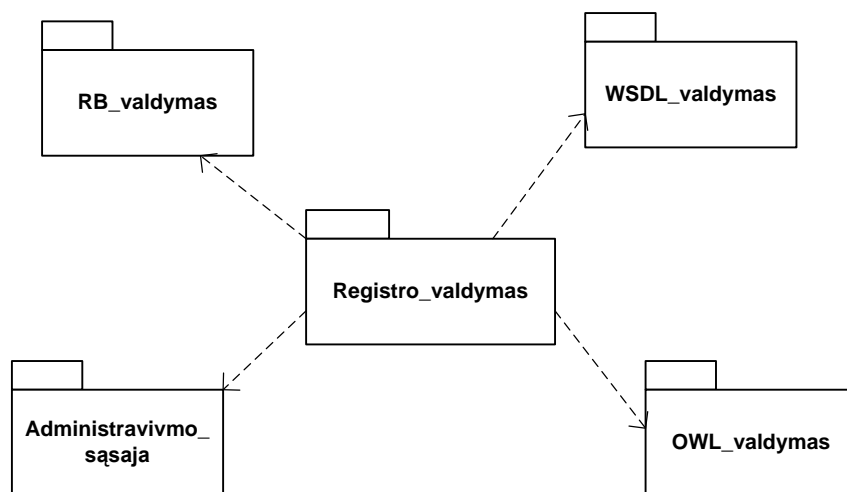
Šiame poskyryje pateikiama projekto loginė architektūra, išskaidyta į programinius paketus. Kadangi projektuojamos dvi programos - pateikiamos atskiros jų loginės architektūros:

- Internetinių paslaugų paieškos programos
- Registrų administravimo programos

4.2.1. Registrų administravimo programos loginė architektūra

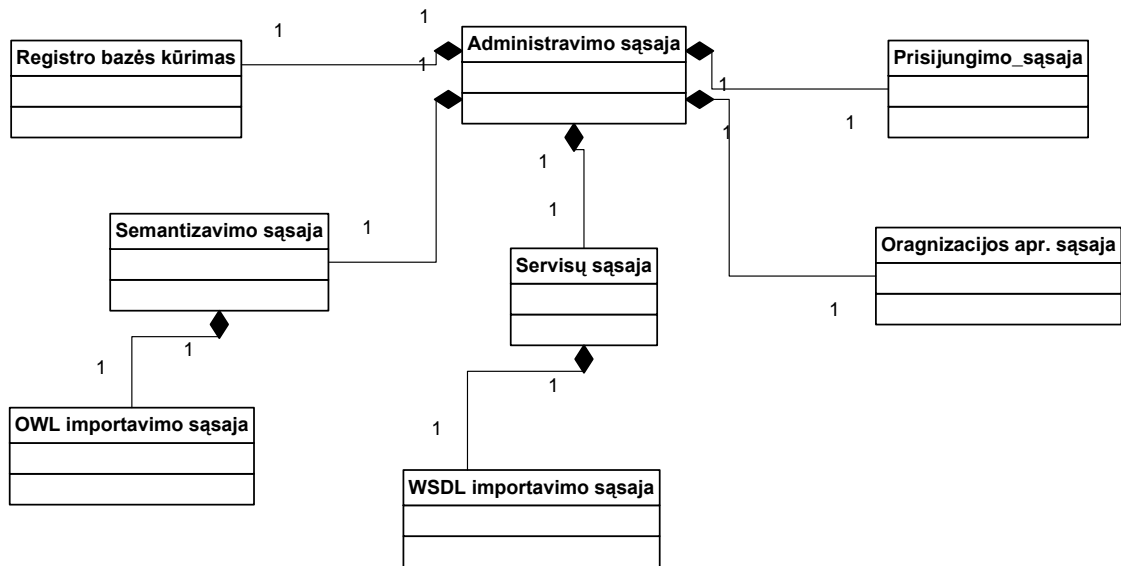
Registrų administravimo programa išskaidoma į penkis paketus (19 pav.):

- *RB_valdymas* – klasės kurios, atlieka operacijas su duomenų baze. Užklausų formavimas, jų vykdymas, informacijos pateikimą kitom paketų klasėm;
- *WSDL_valdymas* – klasės, kurios atlieka WSDL failo importavimą ir analizę;
- *OWL_valdymas* – klasės, kurios atlieka OWL failo importavimą, jos struktūros analizę. Atlieka OWL semantinių objektų paieškos, palyginimo funkcijas;
- *Administravimo_sąsaja* – klasės, kuriomis formuojama grafinė sąsaja tarp pačios programos ir vartotojo;
- *Registro_valdymas* – klasės, kurios atlieka manipuliacijas su registru. Tai pagrindinės klasės, kurios valdo prieš tai minėtus klasių paketus.



19 pav. Administravimo programos loginis klasių vaizdas

Administravimo_sąsaja



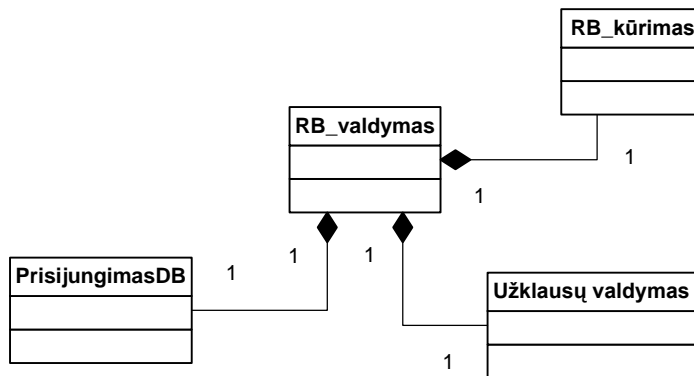
20 pav. *Administravimo sąsaja* loginis klasių vaizdas

Administravimo_sąsaja paketą sudaro (20 pav.):

- *Registro bazės kūrimas* - formuoja registro bazės dialogo langą. Kuriame administratorius gali suvesti būsimos registru bazės duomenis. Administratorius pateikia prisijungimo duomenis prie duomenų bazės, kurioje nori kurti registru bazę. Po sėkmingų veiksmų, vartotojui pranešama apie bazės sukūrimą;
- *Prisijungimo sąsaja* – formuoja prisijungimo dialogo langą prie egzistuojančios registru bazės;
- *Organizacijos aprašymo sąsaja* – įgalina administratoriui vesti duomenis apie registre užregistruotą verslo organizacija, kuri teikia internetines paslaugas. Organizacijos aprašymą sudaro: pavadinimo, interneto adreso, detalesnio organizacijos veiklos aprašymo, kontaktinių duomenų įvedimą;
- *Servisų sąsaja* – pateikia administratoriui iš importuoto WSDL failo rastas serviso ar jau registre įtrauktų servisų operacijas ir jų įėjimus, išėjimus. Operacijos irgi aprašomos, kokia funkciją jos atlieka. O įėjimai ir išėjimai vėliau naudojami semantikos priskirime;
- *Semantizavimo sąsaja* - formuoja sąsaja, semantinių klasių iš importuoto OWL failo struktūros susiejimą su atitinkamais serviso objektais (operacijos, įėjimai, išėjimai). Sąsajoje serviso struktūra pateikiama medžio išdėstymo principu;

- *OWL importavimo sąsaja* - pateikia dialogo langą, kur vartotojas nurodo OWL failo interneto adresą, kuri reikia importuoti;
- *WSDL importavimo sąsaja* – pateikia dialogo langą, kur vartotojas nurodo WSDL failo internetinį adresą, kuri reikia importuoti.

Rb_valdymas



21 pav. *Rb_valdymas* loginis klasių vaizdas

Rb_valdymas paketą sudaro (21 pav.):

- *PrisijungimasDB* – klasė atliekanti prisijungimo funkcijas prie registrų bazės, gražinanti informaciją apie sėkmingą/nesėkmingą prisijungimą;
- *Užklausių valdymas* - klasė atliekanti užklausių formavimus, jų vykdymą ir rezultatų gražinimą atitinkamos kitoms klasėms;
- *RB_kūrimas* - šios klasės funkcijos sukuria, pasirinktoje duomenų bazėje pagal administratoriaus pateiktus duomenis, registrų bazę.

WSDL_valdymas



22 pav. *WSDL_valdymas* loginis klasių vaizdas

WSDL_valdymas paketą sudaro (22 pav.):

- *WSDL_valdymas* – klasė, kuri atlieką importuoto WSDL failo analizę. Jos metu išrenka faile aprašytas serviso operacijas, jų įėjimų ir išėjimų informaciją;

- *WSDL_importavimas* – atlieka WSDL importavimą iš administratoriaus nurodyto šaltinio.

OWL_valdymas

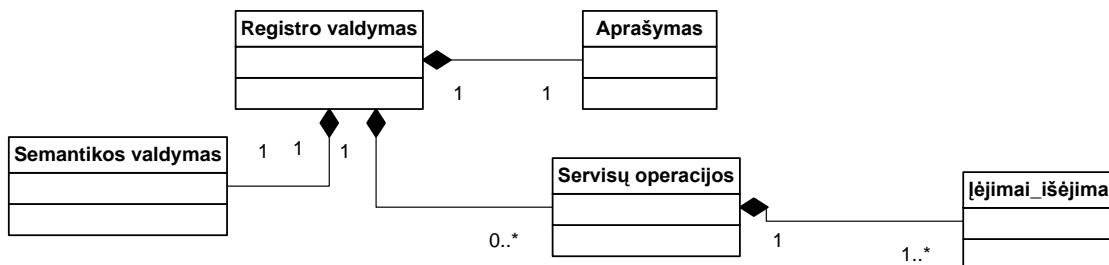


23 pav. *OWL_valdymas* loginis klasių vaizdas

OWL_valdymas paketą sudaro(23 pav.):

- *OWL_valdymas* – klasė, kuri atlieką importuoto OWL failo analizę. Turi funkcijas reikalingas dirbti su OWL struktūra: klasių išskirimas, palyginimas, ryšių radimas tarp klasių;
- *OWL_importavimas* – atlieka OWL importavimą iš administratoriaus nurodyto šaltinio.

Registro valdymas



24 pav. *Registro_valdymas* loginis klasių vaizdas

Registro valdymas paketą sudaro(24 pav.):

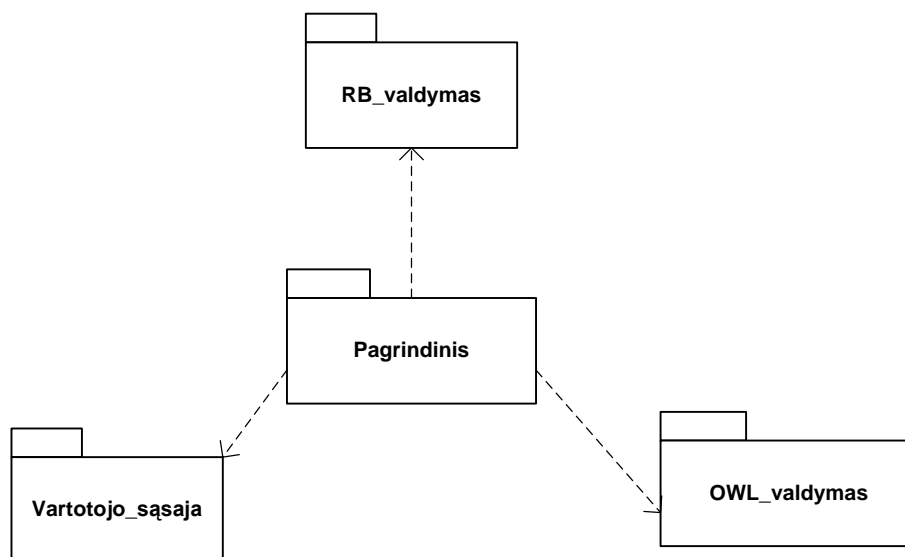
- *Servisų operacijos* – klasė, kuri saugo serviso struktūrą (operacijos, įėjimai, išėjimai), kitus parametrus iš importuoto WSDL failo;
- *Įėjimai_išėjimai* – smulkesnė klasė manipuliuojanti kiekvienos serviso operacijos įėjimų ir išėjimų parametrais;
- *Aprašymas* – šios klasės funkcijos atlieka operacijų, jų parametų tekstinį aprašymų, saugojimą susiejimą;

- *Semantikos_valdymas* – svarbiausia klasė, kuri operuoja semantinių klasių priskirimą(susiejimą) atitinkamiems serviso objektams – operacijoms, parametrų.

4.2.2. Paieškos programos loginė architektūra

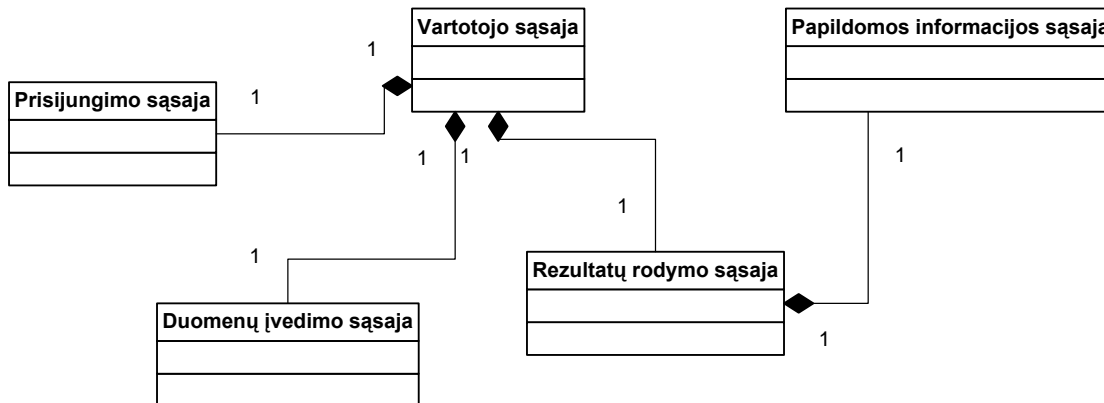
Paieškos programa išskaidoma į keturis paketus(25 pav.):

- *RB_valdymas* – tai panašus paketas, kaip ir anksčiau aprašyta *RB_valdymo* paketas administravimo loginėje architektūroje, kuri atlieka pagrindines funkcijas su duomenų baze: užklausos, jų vykdymas ir t.t.;
- *Vartotojo_sąsaja* – paketo klasės, formuoja vartotojui paieškos atlikimo sąsają;
- *OWL_valdymas* – paketo klasės, panašios administravimo loginėje architektūroje. Atliekančios manipuliacijas su OWL failo importavimu ir analize.
- *Pagrindinis* – pagrindinės klasės, kurios valdo prieš tai minėtų paketų klases.



25 pav. Bendras paieškos programos loginis klasių vaizdas

Vartotojo_sąsaja

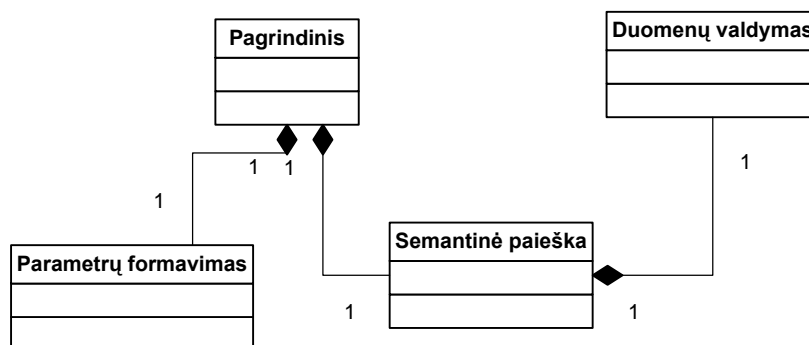


26 pav. Vartotojo_sąsaja loginis klasių vaizdas

Vartotojo_sąsaja paketą sudaro(26 pav.):

- *Prisijungimo sąsaja* – klasė, kuri formuoja prisijungimo dialogo langą prie egzistuojančios registrų bazės;
- *Rezultatų rodymo sąsaja* – atliktos paieškos rezultatų pateikimas vartotojui sąrašo pavidalu;
- *Duomenų įvedimo sąsaja* – formuojami ieškomų parametrų laukai, paieškos pradėjimo funkcijos;
- *Papildomos informacijos sąsaja* – klasė, kuri papildomai suformuoja dialogo langą papildomos informacijos išvedimui apie serviso operaciją.

Pagrindinis

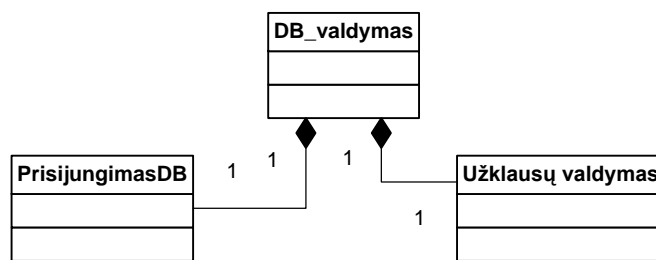


27 pav. Pagrindinis loginis klasių vaizdas

Pagrindinis paketą sudaro(27 pav.):

- *Parametru formavimas* – klasė, kuri formuojanti įvestus parametrus; atskiria ontologijos klasės žodžius; sudaro aibes
- *Duomenų valdymas* – klasė vykdanči iš paslaugos gautų semantikų, operacijų ir parametru valdymą;
- *Semantinė paieška* – klasė, kurioje realizuotas paieškos algoritmas.

Rb_valdymas



28 pav. *RB_valdymas* loginis klasių vaizdas

Rb_valdymas paketą sudaro:

- *PrisijungimasDB* – klasė atliekanti prisijungimo funkcijas prie registrų bazės, gražinanti informaciją apie sėkmingą/nesėkmingą prisijungimą;
- *Užklausų valdymas* - klasė atliekanti užklausų formavimus, jų vykdymą ir rezultatų gražinimą atitinkamos kitoms klasėms;

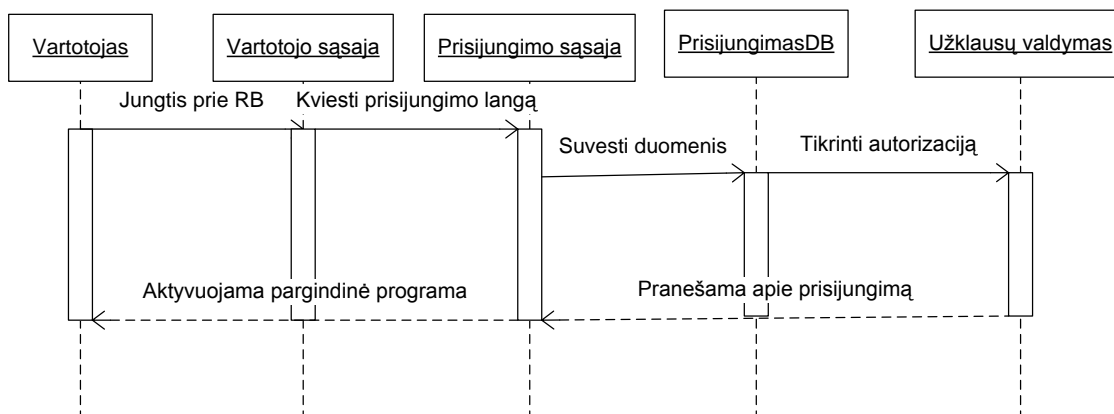
4.3. Programų dinaminiai procesų vaizdai

Šiame poskyryje pateikiama projekte esančių procesų dinaminiai vaizdai: veiklos , sekų diagramomis. Kadangi projektuojamos dvi programos - pateikiami atskiri jų procesų vaizdai:

- Internetinių paslaugų paieškos programos
- Registrų administravimo programos

4.3.1. Paieškos programos dinaminiai vaizdai

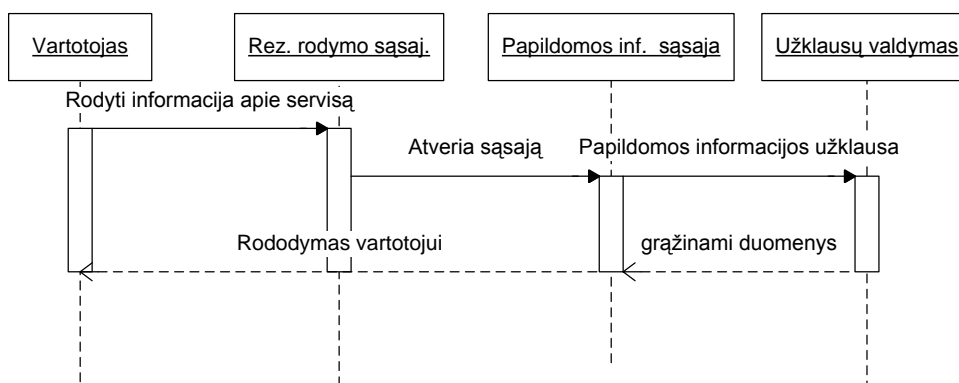
Prisijungimo prie registrų bazės sekos diagrama.



29 pav. Prisijungimo prie registrų bazės sekos diagrama

Diagramoje (29 pav.) pavaizduotas standartinė prisijungimo eiga, kai vartotojas turi atlikti autorizaciją programos pradžioje, jog galėtų toliau dirbti . Vartotojas suveda prisijungimo duomenis prisijungimo programos lange, vėliau vyksta prisijungimas prie nurodytos registrų bazės. Tikrinama, ar vartotojas gali prisijungti. Jei pavyksta – jam leidžiama dirbti toliau su visa programa.

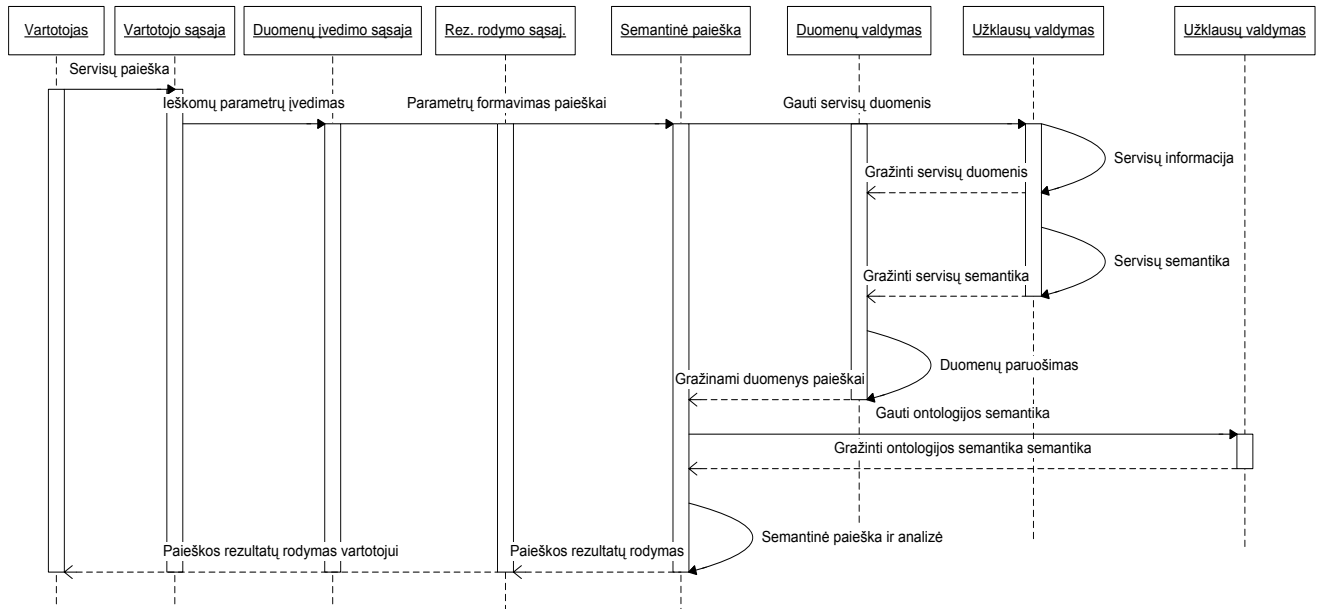
Papildomos informacijos apie pasirinktą operaciją rodymo sekos diagrama.



30 pav. Papildomos informacijos rodymo sekos diagrama

Paveikslėlyje 30 pateikiama, kad vartotojas po atliktos paieškos mato sąrašą rastų operacijų. Jis pasirenka vieną iš jų, toliau jam atveriamas dialogo langas, kuris parodo papildomos informacijos apie operaciją, servisą.

Semantinės paieškos sekos diagrama.



31 pav. Semantinės paieškos sekos diagrama

Paminėsime, kad šioje diagramoje (31 pav.) nepilnai aprašyta „semantinė paieška ir analizė“ eiga. Ši eiga detaliau aprašyta 4.5 skyriuje.

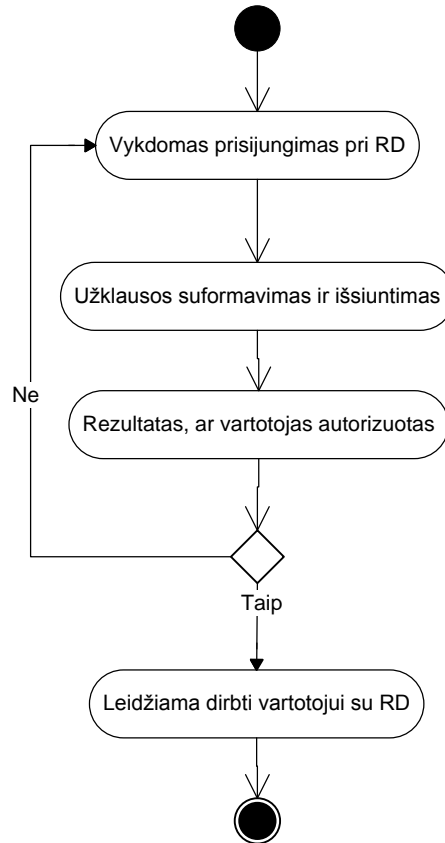
Iš diagramos matome, kad vartotojas visu pirma įveda paieškos parametrus: funkcijos parametras, įėjimų ir išėjimų parametrus. Toliau šie duomenys pateikiami semantinės paieškos klasei. Gaunama registro bazėje esančių servisų informacija: operacijos, įėjimų, išėjimų skaičius, jų tipus. Tuo pačiu gaunamos susietos semantinės klasės su atitinkamais minėtais paslaugos objektais.

Pagal paslaugos tipą surandamas OWL ontologijos failas ir jis importuojamas. Paruošiama ontologijos struktūra naudojimui paieškoje.

Galiausia vykdomas vartotojo įvestų paieškos parametrų semantinis panašumo lyginimas su iš registro bazės gautos paslaugos semantinėmis klasėmis. Palyginus - randamas panašumo koeficientas, kuris nurodo kiek vartotojo įvesti parametrai atitinka paslaugoje esančią operaciją (operacijas). Panašumų nustatinėjimas vykdomas su visais registruose esančiais servisais. Rezultatai kaupiami ir pateikiami vartotojui.

4.3.2. Internetinių paslaugų paieškos procesai

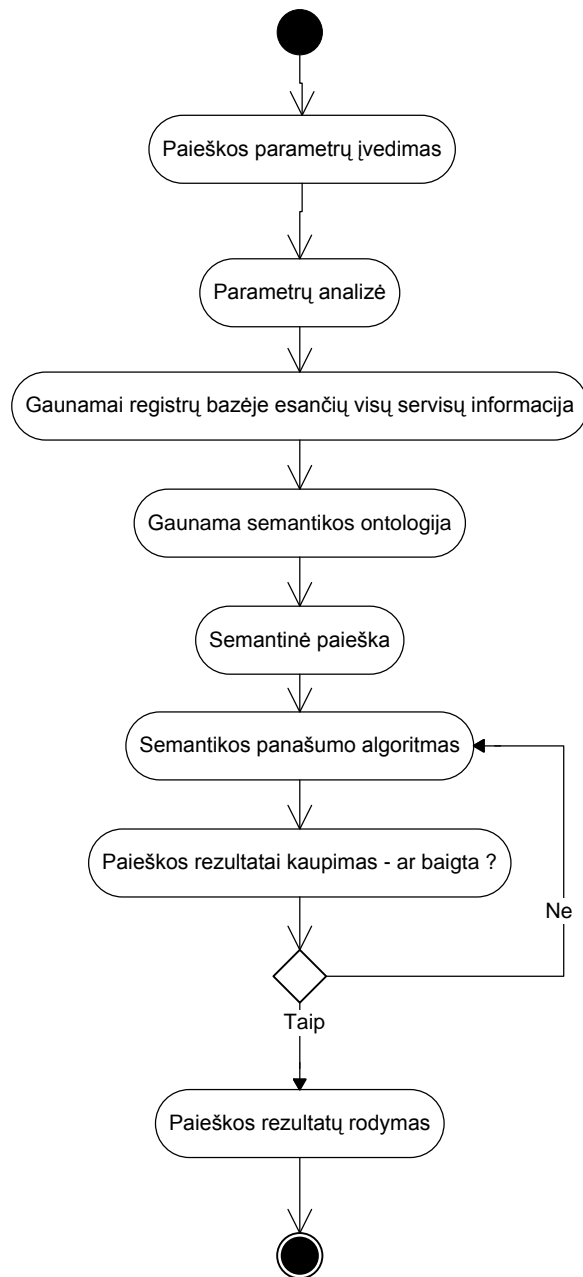
Prisijungimo prie registro bazės veiklos diagrama.



32 pav. Prisijungimo prie registro bazės veiklos diagrama

32 paveikslėlyje pateikta prisijungimo prie registro bazės veiklos diagrama. Vartotojas suveda prisijungimo duomenis atitinkame programos lange, toliau vyksta prisijungimas prie nurodytos registrų bazės. Tikrinama, ar vartotojas gali prisijungti. Jei pavyksta – jam leidžiama dirbti toliau su programa, jei ne kartojamas prisijungimas.

Apibendrintas paslaugų paieškos procesas.

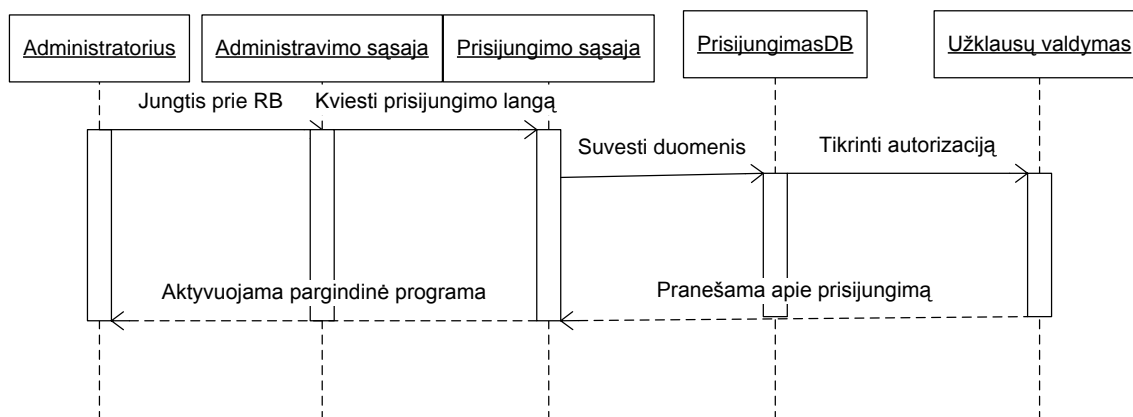


33 pav. Apibendrintas paieškos proceso vaizdas

33 paveikslėlyje pavaizduotas apibendrintas paieškos proceso vaizdas. Vartotojas įveda paieškos parametrus. Parametrai formuojami ir paruošiami paieškai. Toliau pagal įvestą paslaugos funkcijų kategoriją importuojama srities ontologija. Atliekama semantinė paieška. Paieška baigia darbą, patiekiami rezultatai vartotojui.

4.3.3. Registrų administravimo programos dinaminiai vaizdai

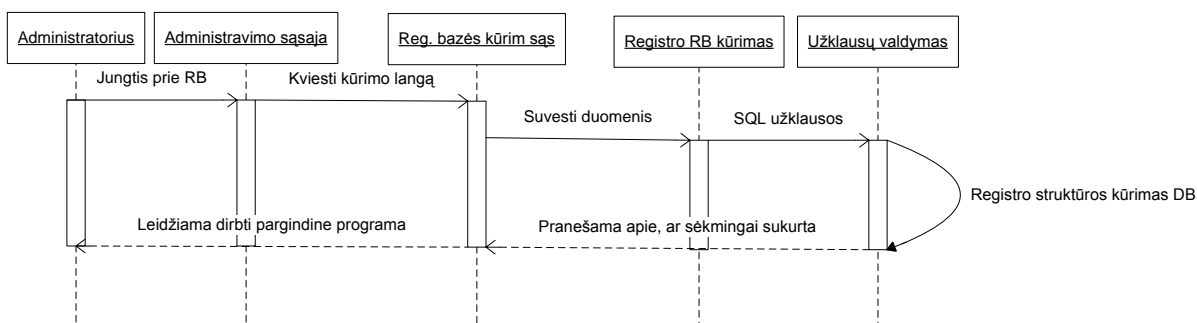
Prisijungimo prie administruojamos registro bazės sekos diagrama.



34 pav. Prisijungimo prie administruojamos registro bazės

34 paveikslėlyje pateikta administratoriaus prisijungimo eiga, panaši kaip vartotojo programos prisijungimas. Reikia atlikti autorizaciją programos pradžioje, kad galima būtų su ja dirbti. Administratorius suveda prisijungimo duomenis atitinkame programos lange, vyksta prisijungimas prie nurodytos registrų bazės. Tikrinama, ar administratorius gali prisijungti. Jei pavyksta – jam leidžiama dirbti toliau su visa programa.

Registro bazės sukūrimas sekos diagrama.

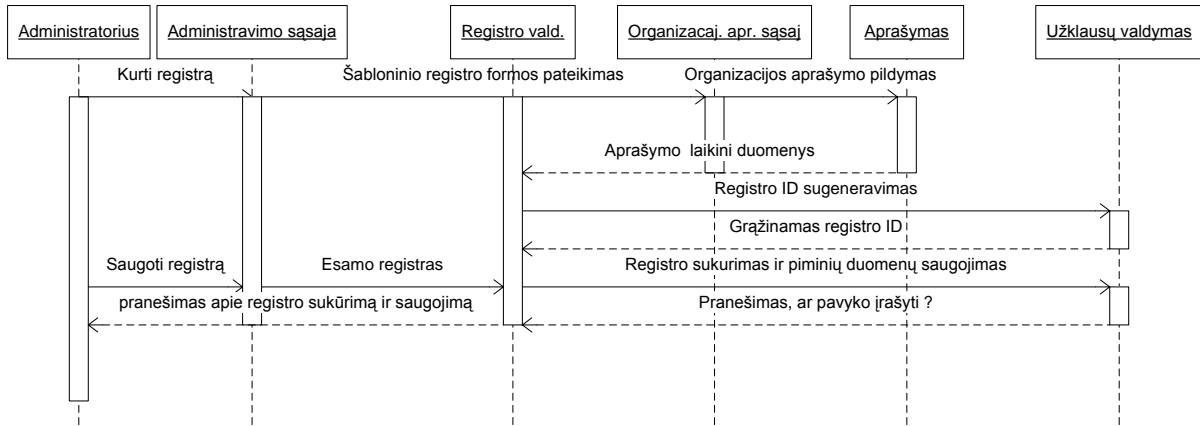


35 pav. Registro bazės sukūrimo sekos diagrama

35 paveikslėlyje pavaizduotas registro bazės kūrimo seka. Registro bazės kūrimo metu administratorius nurodo duomenų bazės serverį, prisijungimo duomenis, bazę kurioje kurti registro struktūrą. Programa gavusi duomenis ir sėkmingai prisijungus prie duomenų bazės,

sukuria lentelės struktūrą, ryšius nurodytoje bazėje. Galiausiai programa administratoriui praneša apie registro bazės sukūrimą.

Pirminio registro sukūrimas su organizacijos aprašu sekos diagrama.



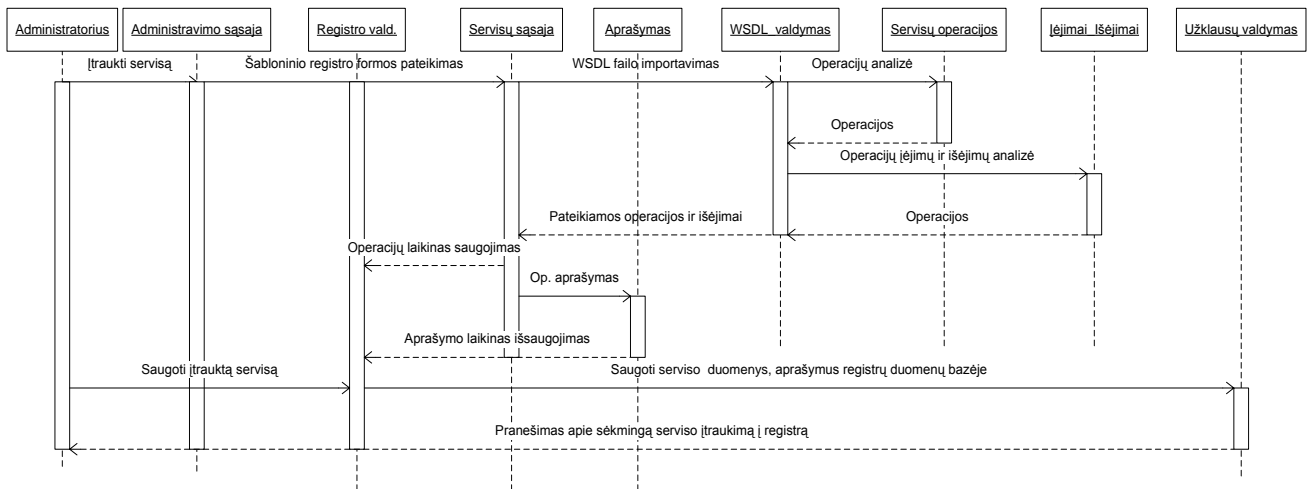
36 pav. Pirminio registro sukūrimas su organizacijos aprašu sekos diagrama

Pirminis registro kūrimas – tai registro sukūrimas, kai registras yra tuščias. Šiuo atveju tuščio registro negalima sukurti, reikalinga į registrą įtraukti verslo organizaciją, kuriai šis registras ir priklausys. Pirminio registro išsaugojimui būtina sąlyga, kad būtų bent verslo organizacijos aprašymas.

Administratorius užpildo pateiktoje organizacijos aprašymo sąsajoje laukelius, kaip pavadinimas, interneto adresas, detalesnį organizacijos aprašymą, kontaktus.

Po visko administratorius gali atlikti saugojimą – registro įrašymą į duomenų bazės lenteles.

Paslaugos įtraukimas į registrą sekos diagrama.



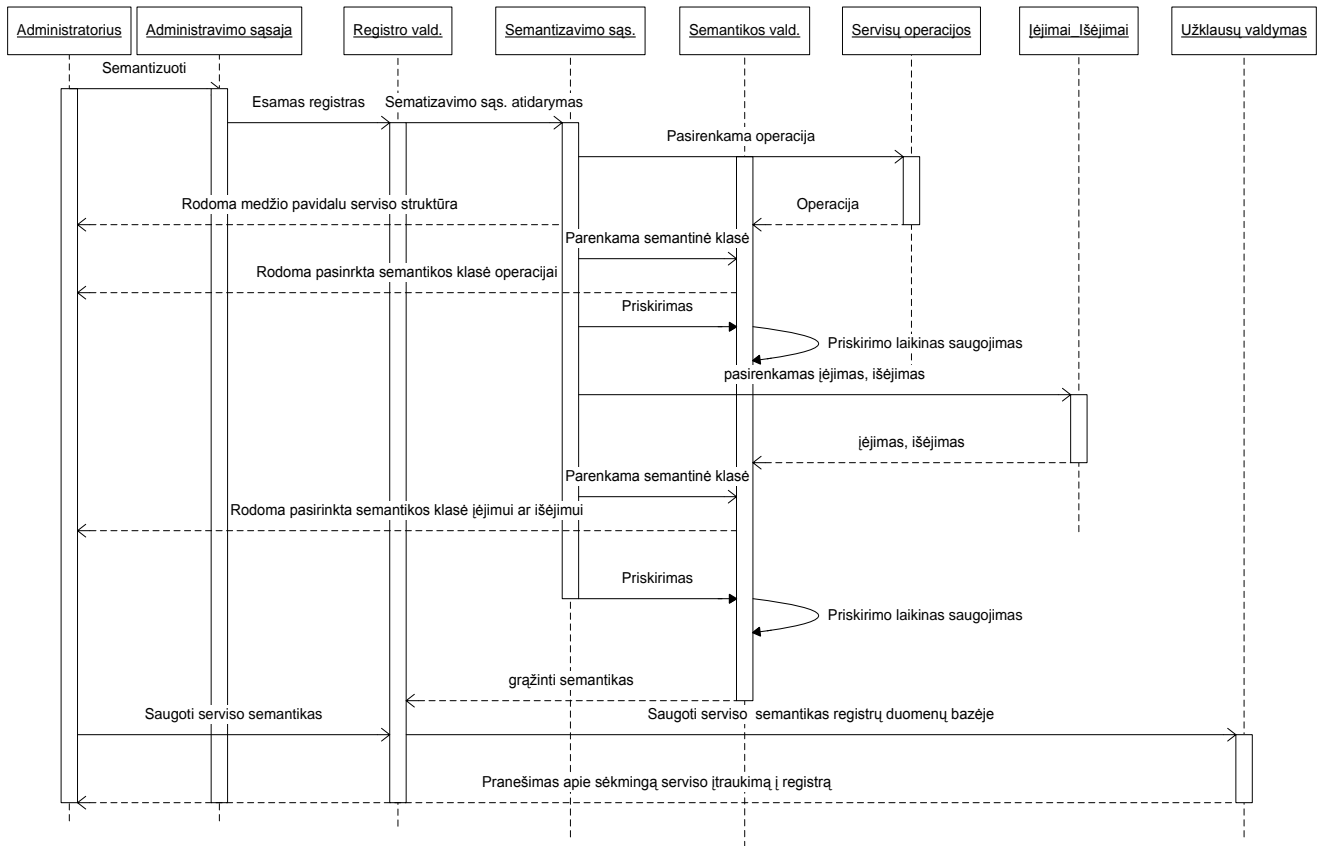
37 pav. Paslaugos įtraukimo į registrą sekos diagrama

Administratorius norėdamas įtraukti registrą, atveria serviso sąsajos langą. Visų pirma jam norint įkelti servisą, reikia pateikti veikiančios paslaugos WSDL failo šaltinį. Jis iškviečia WSDL importavimo langą, nurodo internetinį adresą. WSDL valdymo klasės išanalizuoja importuoto WSDL struktūrą išgaudama duomenis apie operacijas, jų įėjimų ir išėjimų parametrus.

Gauti duomenys formuojami ir rodomi administratoriui servisų sąsajoje. Joje jis gali tekstu aprašyti paslaugą, ką ji atlieka, ir esančias operacijas ir jų parametrus.

Po visko galima saugoti servisą. Serviso duomenys išsiunčiami ir įrašomi į atitinkamas registro struktūrinės lentelės.

Registre esamų serviso operacijų ir parametrų semantikos priskyrimo sekos diagrama.

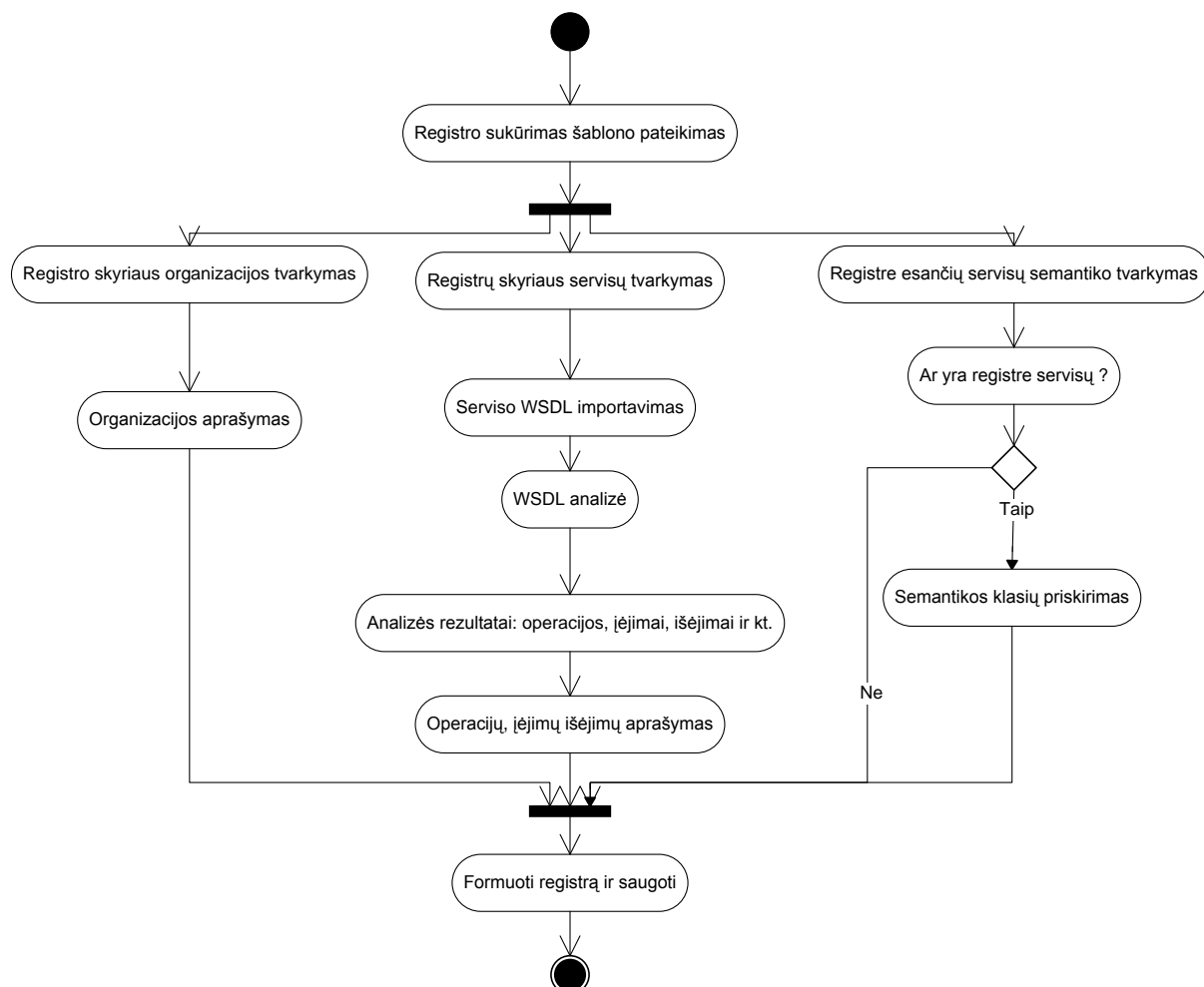


38 pav. Registre esamų paslaugos operacijų ir parametrų semantikos priskyrimo sekos diagrama

Administratorius atvėręs semantikos sąsają, gali pradėti atlikti serviso operacijų ir parametrų susiejimą su atitinkamomis semantinėmis klasėmis. Administratorius nurodo objektą, parenka jo manymu tinkamą semantinę klasę ir galiausiai priskiria. Atlikęs visus priskirimus su serveryje esančiais objektais, galima išsaugoti. Saugojant semantiniai aprašai įrašomi į atitinkamos registro lentelės stulpelius.

4.3.4. Registru administravimo programos veiklos diagramos

Bendra registro kūrimo, aprašymo, semantikos priskyrimo registro bazės veiklos diagrama (39 pav.).

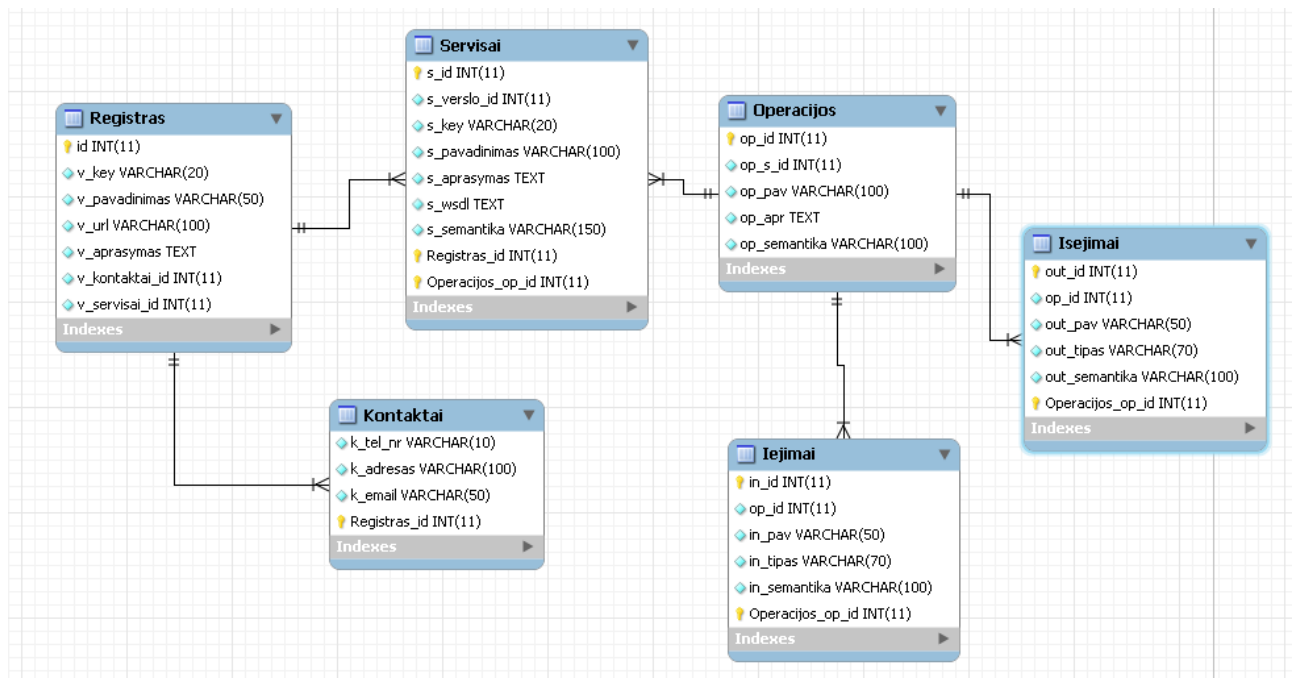


39 pav. Bendra registro kūrimo, aprašymo ir semantikos priskyrimo registro bazės veikla

Sukūrus pirminį registro šabloną, galima lygiagrečiai atlikti veiksmus su registru. Galima iš pradžių aprašyti organizaciją, kam priklauso servisas ar servisi. Po to įtraukti į norimus servisi, juos aprašyti. Galiausiai po serviso įtraukimo, galima atlikti jo objektų semantinių klasių priskyrimą. Atliekamas registro formavimas ir saugojimas.

4.4. Registro duomenų bazė

Remiantis 3.4 skyriuje specifikuotu registro duomenų modeliu realizuota duomenų bazės schema imituojantis semantinį registrą. Duomenų bazės struktūra pateikta 40 paveikslėlyje.



40 pav. Realizuoto registro modelio duomenų bazės struktūra

Realizuojant duomenų bazės struktūrą dėl paprastumo specifikuoto registro modelio tam tikros klasės buvo apjungtos. Toliau pateikiama detalesnė struktūros sandara.

Registracijos lentelė

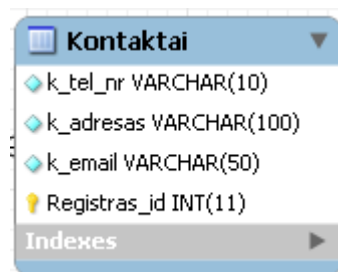


41 pav. Registro lentelė

Pagrindinė duomenų bazės lentelė(pav. 41) aprašanti verslo organizacijos struktūrą: pavadinimas, organizacijos aprašymas ir t.t.

- *id* – pirminis lentelės raktas.
- *v_key* – papildomas unikalus raktas vaizduojantis registro id simbolių eilutę.
- *v_pavadinimas* – organizacijos sutrumpintas pavadinimas.
- *v_url* – organizacijos interneto svetainės adresas.
- *v_aprasymas* – detalesnis aprašymas apie verslo organizaciją.
- *v_kontaktai_id* – saugomos kontaktų lentelės raktai.
- *v_servisai_id* – saugomos servisų lentelės raktai.

Kontaktai lentelė

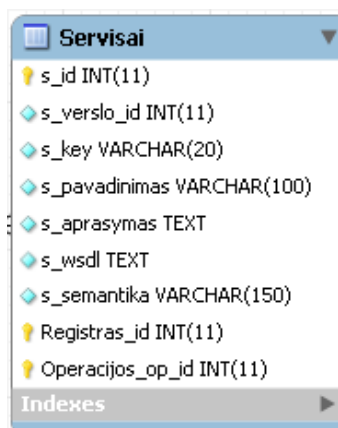


42 pav. Kontaktai lentelė

Kontaktai lentelė – saugomi kontaktinės informacijos įrašai(42 pav.).

- *Registras_id* – lentelės pirminis raktas
- *k_tel_nr* – telefono numeris eilutės simbolių formate
- *k_adresas* – organizacijos buveinės adresas
- *k_email* – elektroninio pašto adresas

Servisai lentelė

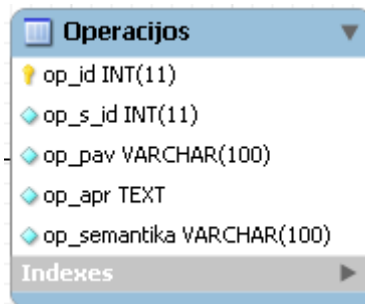


43 pav. Servisai lentelė

Servisai lentelė - šioje lentelėje saugomi įrašai apie organizacijos interneto paslaugą. Paslaugų aprašymai, paslaugos charakteristikos ir kiti papildomi duomenys(pav. 43).

- *s_id* – lentelės pirminis raktas.
- *s_verslo_id* – įrašas saugantis registro lentelės identifikacinį numerį
- *s_key* – serviso(interneto paslaugos) unikalus simbolinis raktas
- *s_pavadinimas* – paslaugos pavadinimas
- *s_wsdl* – įrašas saugantis įkeltos paslaugos WSDL šaltinio internetinį adresą.
- *s_aprasymas* – šiame įrašė saugomas detalesnis paslaugos aprašymas
- *s_semantika* – įrašė saugomas susietas serviso ontologijos OWL pavadinimas

Operacijos lentelė

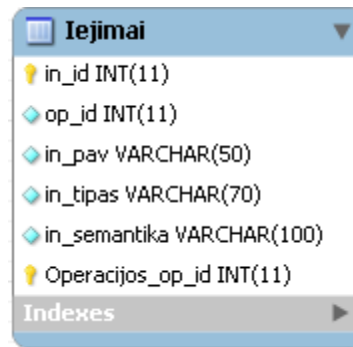


44 pav. Operacijos lentelė

Operacijos lentelė - šioje lentelėje saugomi įrašai apie organizacijos interneto paslaugą(44 pav.).

- *op_id* – lentelės pirminis raktas.
- *op_s_id* – įrašas saugantis serviso lentelės identifikacinį numerį
- *op_pav* – operacijos pavadinimas
- *op_apr* – detalesnis operacijos aprašymas
- *op_semantika* – įrašė saugomas susieto su operacijos su ontologija, klasės pavadinimas

Iėjimai lentelė



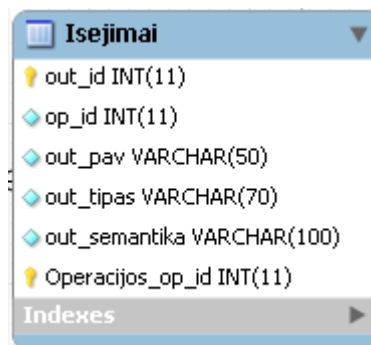
Column Name	Data Type	Key Type
in_id	INT(11)	Primary
op_id	INT(11)	Foreign
in_pav	VARCHAR(50)	None
in_tipas	VARCHAR(70)	None
in_semantika	VARCHAR(100)	None
Operacijos_op_id	INT(11)	Foreign

45 pav. Iėjimai lentelė

Iėjimai lentelė– įrašai apie įėjimų parametrus: kintamojo pavadinimas, tipus ir siejamas semantikas(45 pav.).

- *in_id* – lentelės pirminis raktas.
- *op_id* – įrašas saugantis serviso lentelės identifikacinį numerį siejantis atitinkamą operaciją
- *in_pav* – įėjimo pavadinimas
- *in_tipas* – įrašas saugantis įėjimo parametro tipą
- *in_semantika* – įrašė saugomas susieto įėjimo parametro su ontologija, klasės pavadinimas

Išėjimai lentelė



Column Name	Data Type	Key Type
out_id	INT(11)	Primary
op_id	INT(11)	Foreign
out_pav	VARCHAR(50)	None
out_tipas	VARCHAR(70)	None
out_semantika	VARCHAR(100)	None
Operacijos_op_id	INT(11)	Foreign

46 pav. Išėjimai lentelė

Išėjimai lentelė– įrašai apie išėjimų parametrus, analogiška įėjimų lentelei(46 pav.)

- *out_id* – lentelės pirminis raktas.
- *op_id* – įrašas saugantis serviso lentelės identifikacinį numerį siejantis atitinkamą operaciją

- *out_pav* – įėjimo pavadinimas
- *out_tipas* – įrašas saugantis įėjimo parametro tipą
- *out_semantika* – įrašė saugomas susieto išėjimo parametro su ontologija, klasės pavadinimas

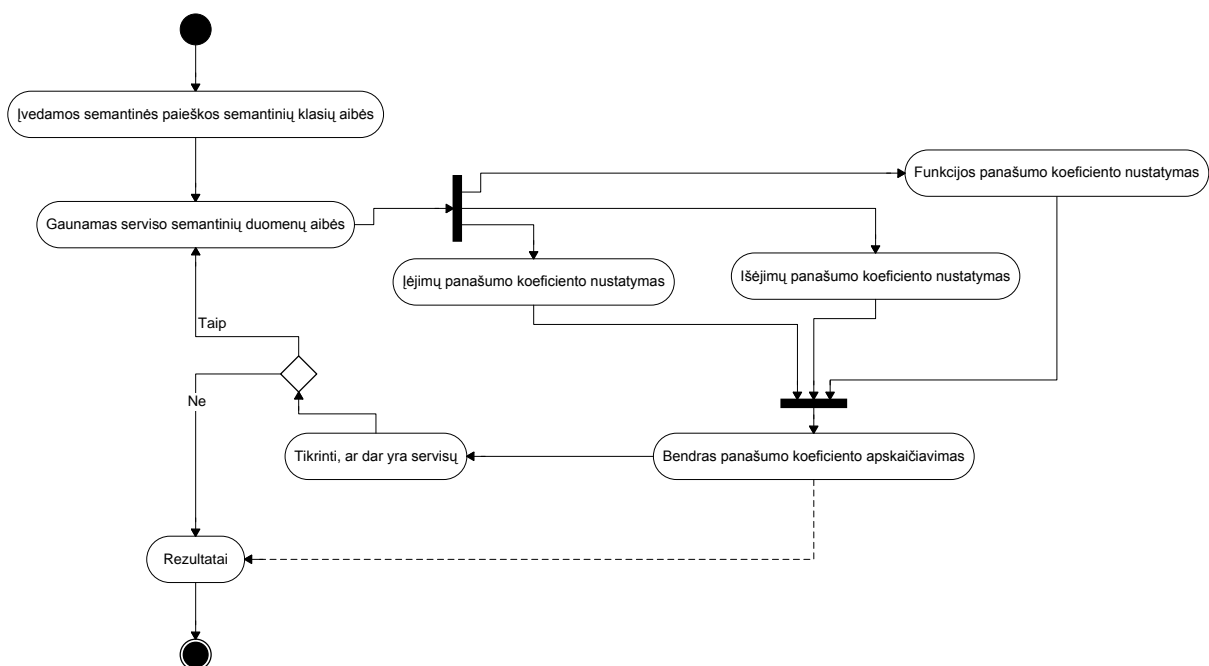
4.5. Semantinės paieškos algoritmas

Šiame skyriuje aprašomas detaliau realizuoto semantinės paieškos algoritmas. Ankstesniuose šio darbo analizės ir specifikacijos dalyse buvo apibrėžtas algoritmas, kokie paieškos duomenys įvedimi, koks laukiamas rezultatas.

Semantinės paieškos algoritmas veikimo proceso atžvilgiu išskiriamas į dvi dalis: **bendras algoritmo ciklas** ir **panašumo koeficiento nustatymo ciklas** iš įvestų paieškos ir interneto paslaugos semantinių palyginimo.

Bendras algoritmo ciklas

Bendras algoritmo ciklo procesas pateiktas veiklos schemeje (pav. 47). Ši proceso tikslas – nustatyti panašumą t.y. panašumo koeficientą tarp vartotojo įvestų paieškos semantinių klasių ir semantiniame registro bazėje esamų paslaugų susietomis semantinėmis klasėmis.



47 pav. Bendras algoritmo ciklas

Iš paveikslėlio 47 matome, proceso pradžioje įvedamos semantinės paieškos parametrų klasės. Klasės atitinkamai išskaidomos į tris aibes: funkcijų(operacijų), įėjimų ir išėjimų. Toliau pradedamas pagrindinis ciklas, kurio metu kreipiamasi į registro bazę, gaunama pirma pasitaikiusi interneto paslauga. Gaunamos paslaugą aprašančios semantinės klasės, kurios išskaidomos į paieškos aibes: funkcijų(operacijų), įėjimų ir išėjimų. Atliekamas atitinkamos kategorijos gautų aibių panašumų nustatymas. Atitinkamai nustačius panašumų koeficientus atliekamas bendras panašumo koeficiento skaičiavimas. Bendras koeficientas nurodo, koks panašumas tarp įvestų parametrų ir serviso turimų operacijų(funkcijų), parametrų t.y. ar operacija turi reikiamą funkcionalumą. Ar įvedimo ir išvedimo parametrai semantiškai panašūs į ieškomuosius.

Bendras koeficientas apskaičiuojamas įvertinus kiekvienos dalies parametrų svorį. Šio algoritmo realizacijoje mes taikome logiškai išvestus svorius:

- Funkcijų(operacijų) panašumas įtakoja – 0,4
- Įėjimų panašumas įtakoja – 0,3
- Išėjimų panašumas įtakoja – 0,3

Bendro koeficiento skaičiavimo formulė:

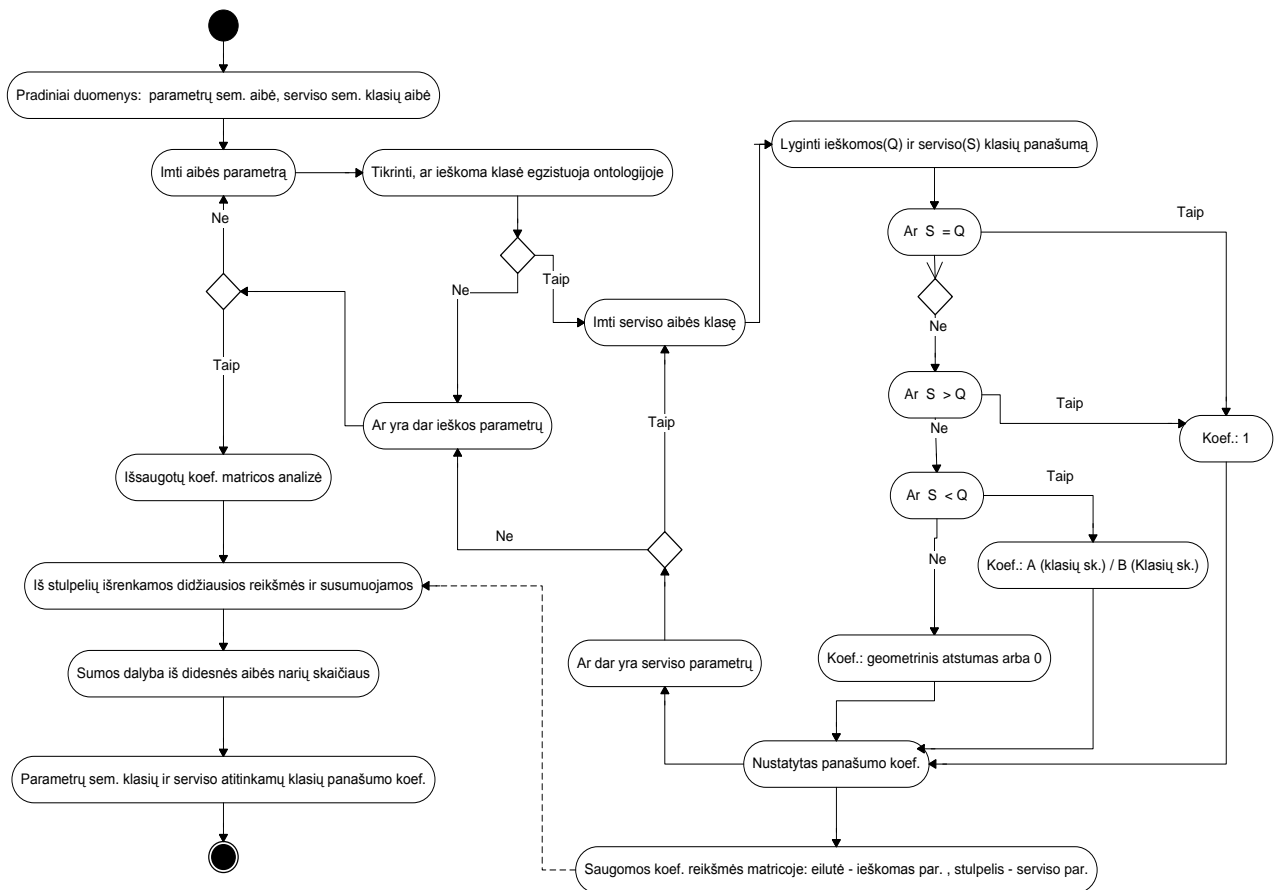
$$Bkoef = Fkoef * 0,4 + Fin * 0,3 + Fout * 0,3$$

Bkoef – bendras koeficientas; *Fkoef* – funkcijos panašumo koeficientas; *Fin* – įėjimų panašumo koeficientas; *Fout* – išėjimų panašumo koeficientas.

Galiausiai atlikus bendro panašumo koeficientų apskaičiavimus su visomis paslaugomis registru bazėje formuojamas rezultatų išvedimas.

Panašumo koeficiento nustatymo ciklas

Panašumo koeficiento nustatymo algoritmo ciklo procesas pateiktas veiklos scheme (pav. 48). Ši proceso tikslas – nustatyti panašumą t.y. panašumo koeficientą tarp atitinkamos kategorijos aibių: funkcijos(operacijų), įėjimų ir išėjimų.



48 pav. Panašumo koeficiento nustatymo ciklas

Proceso pradžioje pateikiamo lyginamos semantinių klasių aibės. Imamas pirmas paieškos aibės klasės narys. Klasė patikrinama, ar ji priklauso atitinkamai paslaugų ontologijai. Jei jos nėra, nėra tikslo atlikti tolimesnius veiksmus, tada pereinama prie kito aibės klasės nario.

Nustačius, jog klasės priklauso ontologijai – ji lyginama su serviso(interneto paslaugos) semantinių klasių aibės nariais. Ieškomoji klasė(Q) lyginama su serviso(S) klase.

Jei $S=Q$, tai ieškoma klasė semantiškai vienoda serviso klasei t.y. jos nusako tą patį objektą. Tuomet panašumo koeficientas lygus 1.

Jei $S > Q$, tai laikoma, jog viena iš serviso tėvo klasės išeinančių vaiko klasių semantiškai vienoda su ieškomąja t.y. išvestoji vaiko klasė turi tokias pačias savybes, kaip ir nurodytoji tėvinė klasė. Tuomet taikomas panašumo koeficientas kaip ir pirmoje sąlygoje – 1.

Jei $S < Q$, tai ieškoma klasė ontologijoje yra tėvinė serviso klasės atžvilgiu. Laikoma, jog serviso klasė dalinai turi tas pačias savybes, kaip ir ieškomoji. Semantišku požiūriu jos nėra vienodos. Panašumo koeficientas apskaičiuojamas remiantis, koku santykiu prarandamas semantinis panašumas. Skaičiavimo formulė:

$$koef = \frac{SN}{QN}$$

koef – panašumo koeficientas; *SN* – iš serviso išvedamų visų vaiko klasių skaičius ontologijos atžvilgiu; *QN* – iš ieškomosios klasės išvedamų visų vaiko klasių skaičius tos pačios kaip ir serviso ontologijos atžvilgiu.

Jei netenkinamos pirmosios sąlygos, bet abi klasės priklauso tai pačiai ontologijai, laikoma jog klasės ontologijos atžvilgiu labai skiriasi semantiškai, jos beveik nevienodos. Tuomet panašumo koeficientas nustatomas pasirinktinai. Galima laikyti – 0, arba jei reikia vis tiek nurodyti, jog klasės priklauso tai pačiai sričiai t.y. ontologijai, bet semantiškai labai skiriasi remiamasi geometrinio atstumo apskaičiavimu ontologijos klasių hierarchijoje. Apskaičiavimo formulė:

$$koef = \sqrt{\frac{SN \cap QN}{SN \cup QN} * \frac{SN \cap QN}{QN}}$$

koef – panašumo koeficientas; *SN* – iš serviso išvedamų visų vaiko klasių skaičius ontologijos atžvilgiu; *QN* – iš ieškomosios klasės išvedamų visų vaiko klasių skaičius tos pačios kaip ir serviso ontologijos atžvilgiu.

Atlikus koeficiento apskaičiavimą išsaugomos reikšmės matricoje. Toliau su tuo pačia einamąja paieškos klase skaičiuojamas koeficientas su kita serviso klase. Po to seka analogiškai skaičiavimai su kita paieškos semantine klase ir lyginama su tais pačiais serviso klasės duomenis.

Po visų klasių rinkinių panašumų nustatymo, gaunama matrica. Gautos matricos pavyzdys pateikta 4 lentelėje.

4 lentelė

Koeficientų matricos pavyzdys			
	S1	S2	S3
Q1	1,0	0	1
Q2	0,5	1	1
Q3	0,09	0,19	0

Šioje lentelėje surašyti panašumų koeficientai gauti lyginant paieškos(Q) ir serviso(S) aibių klasių narius. Iš matricos stulpelių reikšmių išrenkame didžiausio koeficiento reikšmę. Iš gautas didžiausias reikšmes susumuojame ir padalijame iš didesnės aibės skaičiaus, šiuo atveju iš S aibės narių kiekio t.y. 3 . Galiausiai gauta reikšmė yra ieškomas bendras atitinkamos kategorijos panašumo koeficientas.

5. TYRIMUI REALIZUOTOS PROGRAMINĖS PRIEMONĖS

Realizuotos programinės priemonės:

- Semantinės paieškos registrų bazėje programos realizacija.
- Administracinė registrų valdymo programos realizacija.

Registrų administracinė programos realizuotos funkcijos:

- Registro kūrimas
- Verslo organizacijų kontaktų įtraukimas
- Interneto paslaugų įtraukimas į registrą
- Paslaugų operacijų ir jų parametrų analizė pagal WSDL failą
- Semantikos priskirimas
- Registro atnaujinimas

Servisų paieškos registruose programa tenkina šiuos tikslus:

- Paieškos atlikimas nurodant semantinės paieškos parametrus ontologijos klasėmis. Paieška atliekama ieškant semantinių panašumų su minėtomis parametrų grupėmis.
- Paieškos rezultatų pateikimas nuo panašiausio operacijos turinčio serviso iki mažiausiai. Panašumas įvertinamas panašumo koeficientais.

5.1. Registrų administravimo programa

Bendra administravimo programos sąsaja pateikta paveikslėlyje (49 pav.)

49 pav. Bendra administravimo programos sąsaja

5.1.1. Naujo registro sukūrimas

Prieš pradėdant darbą reikia prisijungti prie registrų duomenų bazės. Sugeneruojamas šablonas t.y. leidžiama valdyti tris skyrius:

- *Verslo organizacija*
- *Verslo organizacijos servisai*
- *Owl/Semantika*

50 pav. Verslo organizacija aprašas

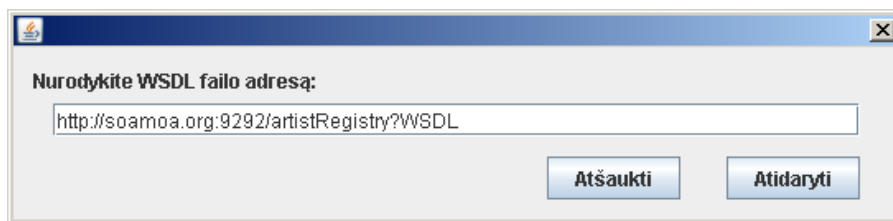
Verslo organizacija

Šis skyrius pavaizduotas 50 pav. Jame aprašoma organizacijos kontaktiniai duomenys ir bendra informacija:

- Organizacijos pavadinimas
 - Organizacijos interneto svetainės adresas
 - Organizacijos aprašymas
- Kontaktinė informacija:
- Verslo organizacijai priklausančio kontaktinio asmens telefono numeris
 - Organizacijos būstinė
 - Kontaktinės elektroninis paštas

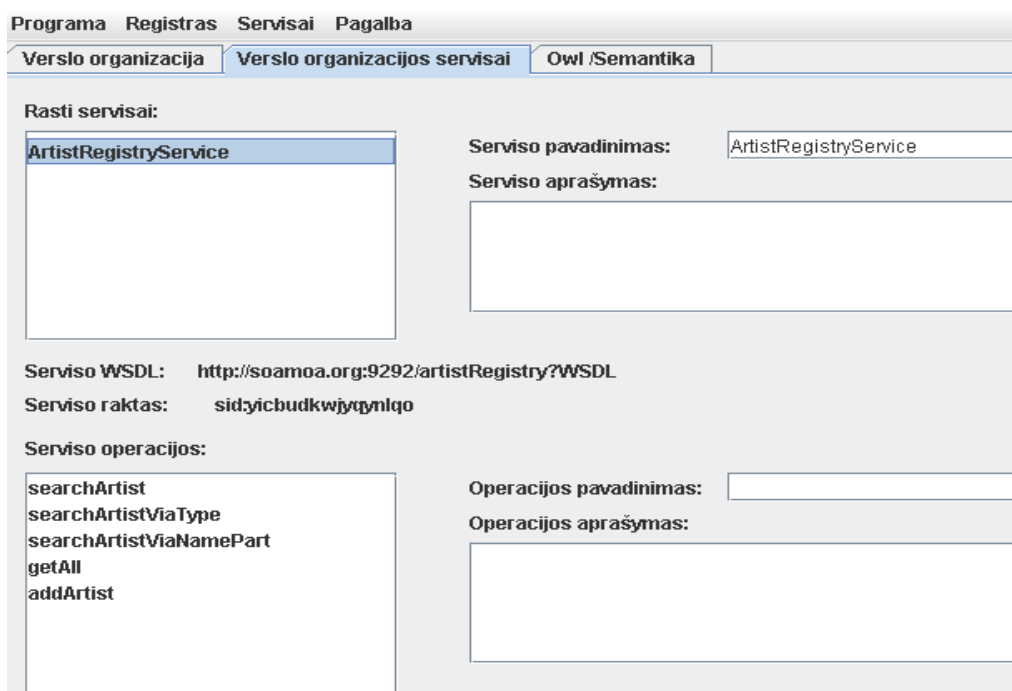
5.1.2. Interneto paslaugų įtraukimas į registrą

Interneto paslaugos įtraukimas: *Servisai* – *Įtraukti naują WSDL*. Dažniausiai vienas WSDL failas aprašo vieną paslaugą(servisą), tai patogu tokiu būdu įtraukti ir papildomus servisus. Įtraukdami WSDL aprašą turime pateikti pilnutinį serviso WSDL failo vietą, pavyzdžiui(51 pav.): <http://soamoa.org:9292/artistRegistry?WSDL>



51 pav. WSDL failo importavimas

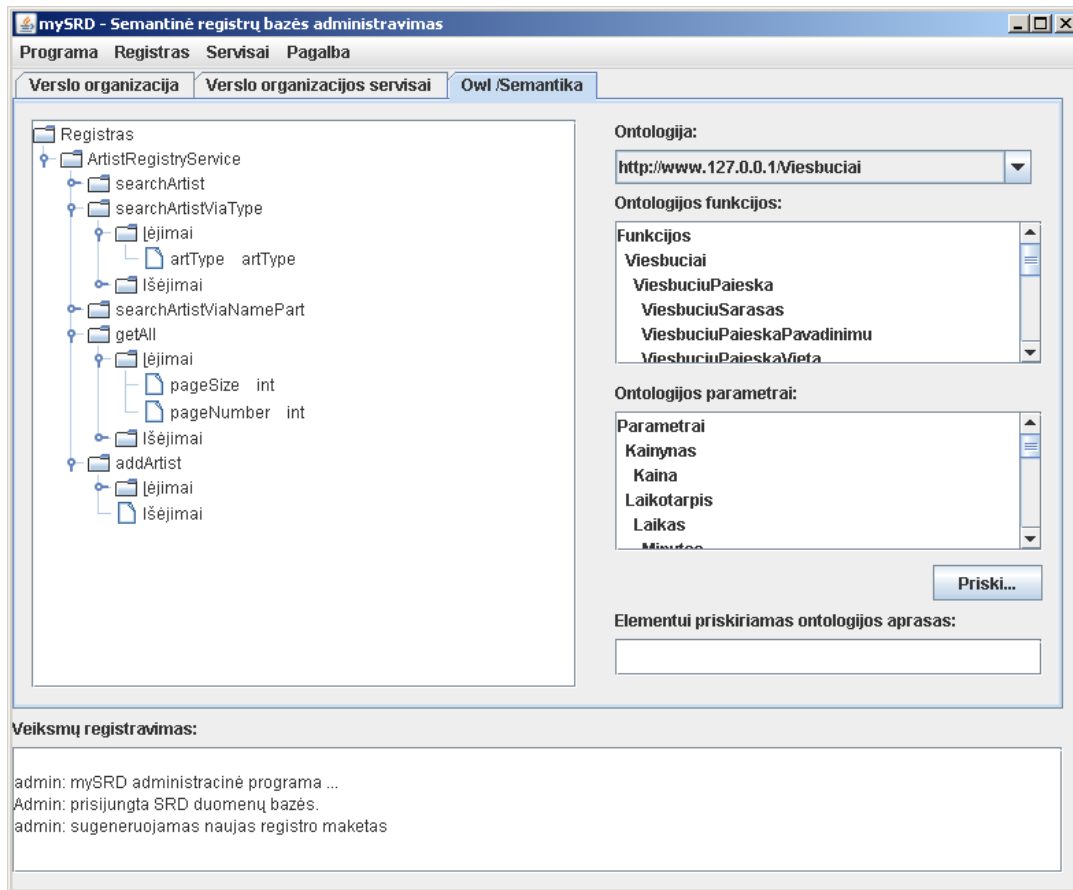
Atidarius automatiškai importuojamas WSDL failas, atliekama struktūros analizė ir rastos paslaugos, operacijos pateikiami vartotojui(52 pav.).



52 pav. Įkeltos paslaugos operacijos

5.1.3. Semantikos priskyrimas

Skyrius Owl/Semantika – leidžia atlikti priskyrimus iš importuoto ontologijos failo, kuris paruoštas šiam modeliui. OWL faile yra aprašytos operacijų, įėjimų ir išėjimų semantinės klasės. Pačiai interneto paslaugai priskiriama bendroji ontologija. Ontologija aprėpia tam tikrą sritį. Paveikslėlyje 53 pateikiama „Owl/Semantikos“ skyriaus sąsaja.



53 pav. Semantikos priskyrimo sąsaja

Kairėje pusėje pateikiamas registro struktūros medis su įtraukta interneto paslauga *ArtistRegistryService* ir išeinančiomis operacijomis. Operacijos turis šakas vaizduojančias įėjimus ir išėjimus. Pavyzdžiui: operacija *searchArtistViaType*, jis turi įėjimo parametrus *artType* ir tipas *artType*. Dešinėje pusėje matome įkeltą ontologijos struktūrą.

Atitinkamo serviso elemento semantinio įrašo priskyrimas visais atvejais panašus:

- Pasirenkamas paslaugos elementas t.y. šaka.
- Iš dešinės pusės pasirenkama reikiama klasės ontologija
- Spaudžiamas mygtukas priskirti; semantinis įrašas priskiriamas prie reikiamo elemento

5.2. Registrų paieškos programa

5.2.1. Paieškos programos sąsaja

Prisijungus prie registrų bazės, paieškos programa automatiškai įkrauna ontologijas – taip paruošdama programą semantinei paieškai. Paieškos programos sąsaja pateikta paveikslėlyje 54.

Servasas	Operacija	Įėjimų sk.	Išėjimų sk.	Panašumo koeficientas

54 pav. Semantinės paieškos sąsaja

Paieškos programos sąsają sudaro dvi dalys:

- Paieškos parametrų vedimo dalis – kur pasirenkama ieškomų servisų kategorija, pasirenkamos arba įrašomos norimos ieškoti paslaugos funkcijos(operacijos) ir jos įvedimų ir išvedimų klasės..
- Rezultatų rodymo dalis – pateikiami rezultatai apie rastus servisu su ieškota funkcija

5.2.2. Semantinė paieška

Semantinė paieška formuojama trijų dalių:

- Kategorijos pasirinkimas
- Ieškomos funkcijos pasirinkimas
- Ieškomų funkcijų įėjimų ir išėjimų parametrų pasirinkimas

Pasirinkite kategoriją:

Pasirinkite ieškomą serviso funkciją:

Pasirinkite serviso parametrus:

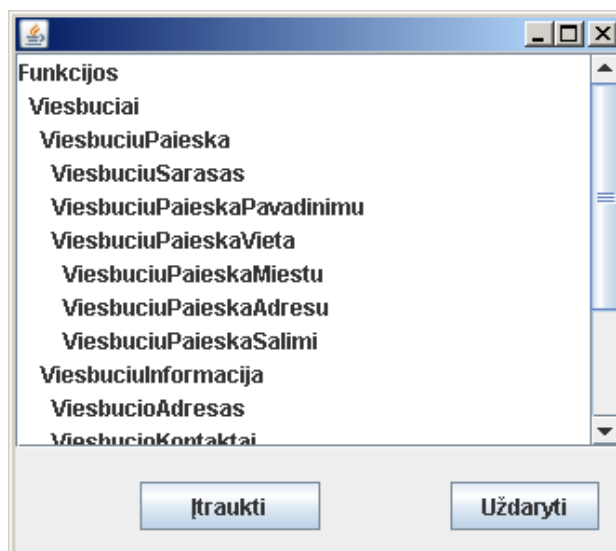
Įvedimo:

Išvedimo:

55 pav. Parametrų įvedimas

Kategorijos pasirinkimas tolygus ontologijos pasirinkimui pateiktame paveikslėlyje matome viešbučių ontologiją. Taip nurodoma: ieškok servisų, kurie priklauso tai kategorijai.

Toliau įvedama norima funkcija ar operacija. Palengvinant norimos operacijos ontologijos klasės įvedimą pagelbsti vedlys, kuris iškviečiamas dešinėje pusėje esančiu mygtuku *Įtraukti*. Paveikslėlyje 56 patiekiamas struktūrinė ontologijos klasės pasirinkimo sąsaja.



56 pav. Funkcijų ontologinių klasių pasirinkimas

Įkeliami semantiniai funkcijų/veiksmų žodžiai, kurie nusako tam tikrą ieškomą veiksmą. Pasirenkamas žodis ir spaudžiame mygtuką *Įtraukti*. Pasirinktas žodis automatiškai įtraukiamas į funkcijų eilutę.

Atlikus parametrų įvedimą galima pradėti atlikti paiešką. Paieškos pradžios sąsaja pateikta paveikslėlyje 57.

57 pav. Paieškos pradžia su įvestais parametrais

Semantinės paieškos rezultatai

Rezultatai pateikiami lentelės pavidalu(58 pav.).

Servisas	Operacija	Įėjimų sk.	Išėjimų sk.	Panašumo koeficientas
HelloService	sayHello	1	1	1
ArtistRegistryService	searchArtistViaType	1	1	1
ArtistRegistryService	searchArtistViaNamePart	1	1	1
ArtistRegistryService	getAll	2	2	0,85
ArtistRegistryService	searchArtist	5	5	0,76
ArtistRegistryService	addArtist	5	5	0,46

58 pav. Rastų operacijų rezultatai

5.3. Testavimas

Testavimo tikslas – patikrinti, ar magistro darbo tyrimui realizuotos programinės priemonės veikia be klaidų, kad neįtakotų eksperimento vykdymo.

Testuojama:

1. Registro sukūrimas: organizacijos aprašymas, paslaugų įkėlimas ir semantikos priskyrimas
2. Patikrinti, ar teisingai atliekamas paslaugų operacijų radimas pagal įvestus semantinius požymius

1 testavimas

1. Sukuriamas fiktyvus WSDL failo aprašas.
2. Paruošiama tuščia registrų bazė.
3. Sukuriamas registras ir užpildomas verslo organizacijos aprašymas(59 pav.)

59 pav. Įkeliama paslauga

4. Importuojamas fiktyvus WSDL failo aprašas(60 pav.)

60 pav. WSDL failo importavimas

5. Aprašome struktūrą(61 pav.)

Rasti servisi:

WorlHotelSearch

Serviso pavadinimas: WorlHotelSearch

Serviso aprašymas: Paslaugos aprašymas

Serviso WSDL: http://tornado.ktu.lt/~kpis/WSDL/duom4.wsdl

Serviso raktas: sid:wqjpquyngtmtkpzy

Serviso operacijos:

GautiTeikiamasPaslaugas
leskotiViesbucio
GautiViesbAdresa

Operacijos pavadinimas: GautiViesbAdresa

Operacijos aprašymas: Randamas viešbutis pagal adresą

61 pav. Operacijos aprašymas

6. Atliekamas semantikos priskyrimas(62 pav.)

Registras

- WorlHotelSearch
 - GautiTeikiamasPaslaugas
 - Išėjimai
 - Viesbutis string
 - Išėjimai
 - Paslaugos string
 - leskotiViesbucio
 - Išėjimai
 - viesbucio_pav string
 - Išėjimai
 - valstybe string
 - viesbutis string
 - GautiViesbAdresa
 - Išėjimai
 - Valstybe string
 - Viesbutis string
 - Išėjimai
 - Adresas string
 - Kontaktine_inf string

Ontologija: http://www.127.0.0.1/Viesbuciai

Ontologijos funkcijos:

- PaieskaKataloguose
- ViesbuciuPaieskaVieta
- PaieskaAdresu
- PaieskaMiestu**
- PaieskaSalimi
- ViesbuciuPaslaugos

Ontologijos parametrai:

- Vieta
- Salis
- Adresas
- Miestas
- Gatve
- Paslaugos**

Pris...

Elementui priskiriamas ontologijos aprašas: Paslaugos

62 pav. Operacijų ir jų parametų semantikos priskyrimas

7. Išsaugome registrą

8. Atliekame paiešką – rezultate turime matyti tris operacijas(63 pav.)

9. Peržiūrime *IeskotiViesbucio* detalesnę informaciją (64 pav.)

Pasirinkite kategoriją: ▼

Pasirinkite ieškomą serviso funkciją:

Pasirinkite serviso parametrus:

Išvedimo:

Išvedimo:

Servisas	Operacija	Iėjimų sk.	Išėjimų sk.	Panašumo koeficient...
WorlHotelSearch	IeskotiViesbucio	1	1	0,73
WorlHotelSearch	GautiViesbAdresa	2	2	0,7
WorlHotelSearch	GautiTeikiamasPasla...	1	1	0,06

63 pav. Paieškos rezultatai

Servisas: WorlHotelSearch

Serviso raktas: sid:wqjpquyngtmtkpzy

Serviso aprašymas:

Operacija: IeskotiViesbucio

Operacijos aprašymas:

Iėjimai/Išėjimai:

Pavadinimas	Tipas	Semantika
viesbucio_pav	string	Viesbutis

Pavadinimas	Tipas	Semantika
valstybe	string	Salis
viesbutis	string	Viesbutis

Verslo organizacija: Testuojamas pavadinimas

Serviso iškviatimo WSDL: <http://tornado.ktu.lt/~kpis/WSDL/duom4.wsdl>

64 pav. Operacijos detalesnės informacijos vaizdavimas

2 testavimas

1. Naudojamos pirmojo testo sukurtoomis sąlygomis t.y. registrų bazėje yra 3 operacijos. Paveikslėlyje 64 parodytas *IeskotiViesbucio* operacijos detalus aprašymas. Jame galima matyti, priskirtas semantines klases. Naudojantis priede esančiu 69 paveikslėlyje pavaizduota viešbučių ontologija parinksime testuojamus semantinės paieškos parametrų klases ir stebėsime operacijos *IeskotiViesbucio* panašumo koeficiento kitimą.
2. Įvedami parametrai: funkcijos – *ViesbuciuPaslaugos*, įėjimas – *Viesbutis*, išėjimas – *Data*. Rezultatai pateikti paveikslėlyje (65 pav.). Matome, jog panašumo koeficientas sumažėjo, tai įtakojo funkcijos klasių mažas panašumas. Pagal ontologiją *ViesbuciuPaslaugos* ir *ViesbuciuPaieska* dalinai panašios, jos turi bendrą tėvinę klasę. Pagal realizuotą semantinės paieškos algoritmą, su tokia dalinė sąlyga ir klasės skiriasi per viena šaka - funkcijos panašumas sumažėja pusiau.

Servisas	Operacija	Įėjimų sk.	Išėjimų sk.	Panašumo koeficient...
WorlHotelSearch	GautiViesbAdresa	2	2	0,37
WorlHotelSearch	IeskotiViesbucio	1	1	0,36
WorlHotelSearch	GautiTeikiamasPasla...	1	1	0,06

65 pav. Rinkinio *ViesbuciuPaslaugos*, *Viesbutis*, *Data* testas

3. Įvedami parametrai: funkcijos – *ViesbuciuPaieska*, įėjimas – *Viesbutis*, išėjimas – *Vieta*, *Viesbutis*. Rezultatai pateikti paveikslėlyje (x pav.). Sudaromos sąlygos, jog stebima operacija gautu idealų panašumo koeficientą t.y. 1

Servisas	Operacija	Įėjimų sk.	Išėjimų sk.	Panašumo koeficient...
WorlHotelSearch	IeskotiViesbucio	1	1	1
WorlHotelSearch	GautiViesbAdresa	2	2	0,75
WorlHotelSearch	GautiTeikiamasPasla...	1	1	0

66 pav. Idealaus varianto testas

Atlikti testai parodė, jog programos veikimas tinkamas eksperimento vykdymui.

6. EKSPERIMENTINIS SEMANTINĖS PAIEŠKOS ALGORITMO TYRIMAS

6.1. Eksperimentinio tyrimo tikslas ir uždaviniai

Eksperimentų tikslas – ištirti semantinės paieškos algoritmo veikimo charakteristikas semantiniame registre, kuris realizuotas duomenų bazėje. Semantinės paieškos algoritmas realizuotas remiantis Greedy metodu. Tiriamas algoritmas detaliau aprašytas 4.5.2 skyriuje.

Tyrimo uždavinys: ištirti semantinės paieškos algoritmo greitaveiką: kiek laiko užtrunka paieška visoje registro bazėje, kai jame esančių internetinių paslaugų skaičius didėja;

Eksperimente naudojamų duomenų apribojimai. Kadangi eksperimentai atliekami su semantiškais duomenimis, tyrimo rezultatai gaunami daugiareikšmiai. Dėl to eksperimentuojant pradiniai duomenų sudaromi iš vienos srities ontologijos. Atitinkamam uždaviniui sudaromi logiškai parinkti duomenys.

6.2. Semantinės paieškos algoritmo greitaveikos tyrimas

Ištirti semantinės paieškos algoritmo greitaveiką: kiek laiko užtrunka paieška visoje registro bazėje, kai jame esančių internetinių paslaugų operacijų skaičius didėja.

Keliama hipotezė, kurią norima paneigti: didėjant registro bazėje užregistruotų paslaugų skaičiui paieškos algoritmo laikas didėja eksponentiškai. Taip gali įvykti, nes paslaugoje gali būti daugiau negu viena operacija, kiekviena operacija turi skirtingą skaičių įėjimų ir išėjimų. Taigi paieškos parametrų galimų kombinacijų skaičius gali būti didelis.

Alternatyvi hipotezė: paieškos algoritmo laikas kinta tolygiai didėjant. Šis atvejis būtų priimtinas, norint taikyti algoritmą.

Nepriklausomi kintamieji :

- Paieškos algoritmas;
- Naudojama statinė ontologijos klasės struktūra iš priedo (69 pav.);
- Statiniai paieškos parametrų rinkiniai.

Priklausomi kintamieji:

- Paieškos laikas;
- Internetinių paslaugų skaičius registro bazėje, jis didinamas į bazę įtraukiant fiktyvias paslaugas ir susiejamos su varijuojančiomis ontologijos klasėmis;

Eksperimento vykdymas

Sudaromi statiniai paieškos parametrų rinkiniai naudojantis pasirinkta statine ontologijos struktūra pateikta priede 10.1 skyriuje (69 pav.). Ontologijos sritis – viešbučiai. Rinkiniai pateikti lentelėje.

5 lentelė

Paieškos parametrų variantų rinkiniai

Varianto rinkinio nr.	Funkcijos parametras	Iėjimo parametrai	Išėjimo parametrai
1	Viesbuciai	Viesbutis	Kaina
2	ViesbuciuPaieska	Viesbutis	Adresas, Kaina
3	KambarioRegistravimas	Viesbutis, Laikas	Vieta, Kaina

1 variantų rinkiniu bandomė atlikti paslaugos paiešką, kai ieškome operacijos, kuri būtų iš viešbučių kategorijos, operacijos įėjimas - viešbučio pavadinimas ir operacijos rezultatas, išvedama kaina.

2 variantų rinkiniu ieškome operacijų, kurie atlieka viešbučių paiešką, pagal viešbučio pavadinimą ir rezultatas grąžinamas adresas ir/arba kaina.

3 variantų rinkiniu: kambario registravimo tipo operacijų, kur pateikti reikia viešbučio pavadinimą, registracijos laiką, o rezultatas viešbučio vieta(adresas) ir kaina.

Kiekvienas variantas gali nežymiai įtakoti paieškos laiką, jei registro bazėje esančios susietos ontologijos klasės labiau skiriasi semantiniu panašumu.

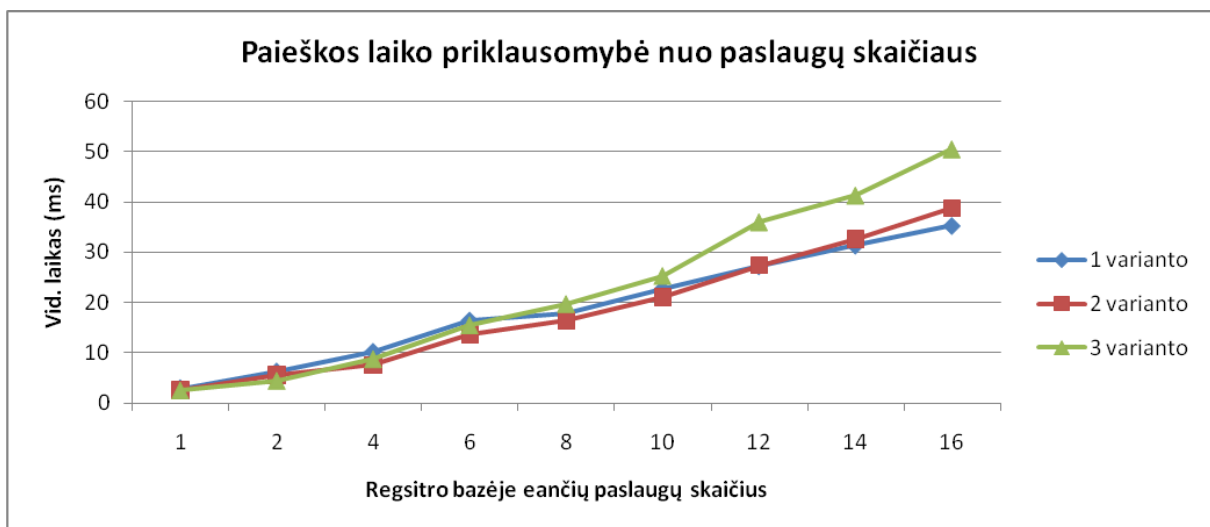
Eksperimento vykdymas:

1. Papildoma esamoji registrų bazė naujomis paslaugomis
2. Atliekami paieškos laiko skaičiavimai su trimis variantais
3. Laiko skaičiavimas su kiekvienu rinkiniu atliekamas 100 kartų, gautos laiko vertės susumuojamos ir išvedamas vidutinis laikas. Tai reikalinga dėl eksperimento vykdymo aplinkos, kuomet laiko reikšmės priklauso nuo kompiuterinės sistemos apkrovimo.
4. Išsaugomi atlikti skaičiavimai
5. Kartojami žingsniai 2 – 4

Eksperimento rezultatai pateikiami priede 7 lentelėje.

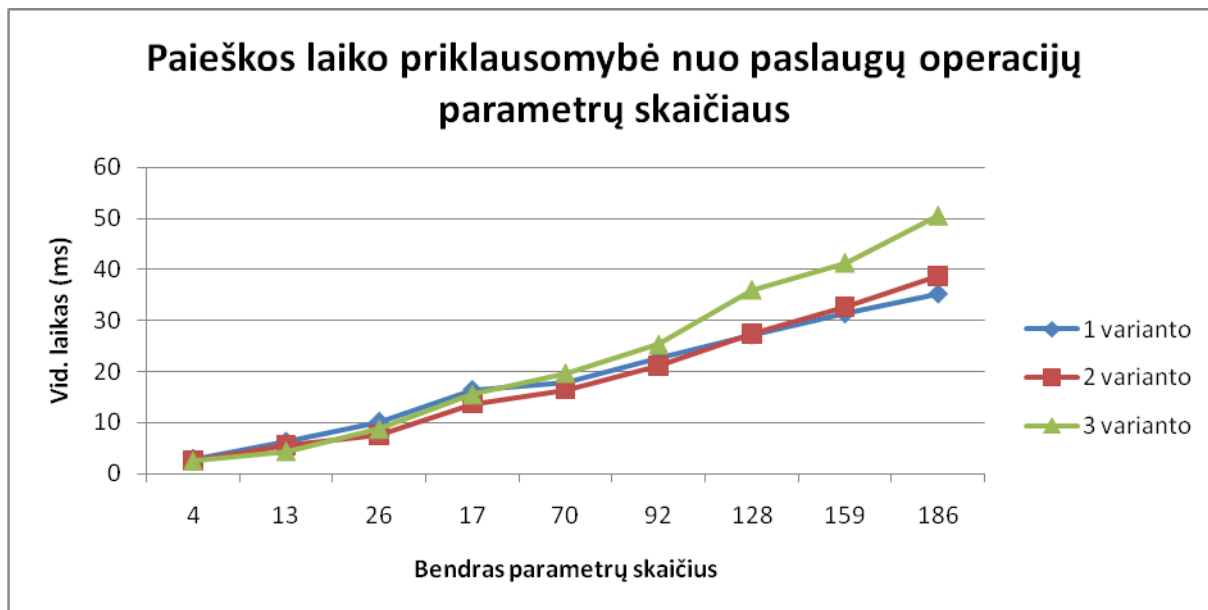
Eksperimento rezultatų analizė

Atlikus eksperimentą galima nustatyti, kad paieškos laikas nuo paslaugų skaičiaus nedidėja eksponentiškai – pirminė hipotezė paneigiama. Gauti rezultatai labiau patvirtina alternatyviają hipotezę, jog semantinės paieškos laikas tolygiai didėjant priklauso nuo registro bazėje esančių paslaugų skaičiaus. Pateikiamas laiko priklausomybės nuo paslaugų skaičiaus variantų grafikai(67 pav.).



67 pav. Paieškos laiko priklausomybė nuo paslaugų skaičiaus

Trečio varianto grafikas didėjant paslaugų skaičiui paieška labiau užtrunka, negu pirmo ar antro varianto. Žymesnis skirtumas atsiranda kai paslaugų skaičius didesnis negu 10. Matomai tai įtakoja trečio varianto didesnis parametru skaičius. Paieškos vykdymo metu algoritmas atlieka visus paieškos parametro klasių lyginimus su paslaugos operacijų ir parametru klasėmis. Kuo daugiau lyginimų, tuo didesnis panašumo skaičiavimo laikas. Tiksliau pamatyti galima paanalizavus bendrus paslaugos operacijų įėjimų ir išėjimų skaičius. Tai yra susumuoti kiekvieno atvejo paslaugų parametrus ir atvaizduoti grafiku (68 pav.).



68 pav. Paieškos laiko priklausomybė nuo bendro parametrų skaičiaus

6.3. Eksperimento išvados

Atlikus eksperimentą, patvirtinta hipotezė, kad didėjant paslaugų ir jų parametrų skaičiui paslaugų paieškos laikas didėja tolygiai ir kartu paneigta nulinė hipotezė, kad semantinės paieškos algoritmo veikimo laikas didėja eksponentiškai, kas reikštų, kad paieškos algoritmas praktiniam taikymui nėra tinkamas.

7. IŠVADOS

1. Išanalizavus semantinių registrų modelius ir architektūrą, buvo padaryta prielaida, kad semantinį registrą tikslinga realizuoti paprastesniu modeliu duomenų bazėje.

2. Išanalizavus semantinės paieškos algoritmus, buvo nutarta realizuoti semantinę paiešką skaičiuojant paslaugų parametrų panašumą Greedy metodu.

3. Realizavus ir ištestavus semantinį registro modelį duomenų bazėje, buvo įsitikinta, kad jame galima saugoti semantinius paslaugų aprašus.

4. Sukurtos programinės priemonės darbui su semantiniu registru leido atlikti eksperimentinius paslaugų paieškos algoritmų tyrimus.

5. Semantinės paieškos algoritmo tyrimo eksperimento rezultatai parodė, kad didėjant paslaugų ir jų parametrų skaičiui paieškos laikas didėja tolygiai ir neišauga eksponentiškai, o tai reiškia, kad sukurtą algoritmą galima taikyti praktikoje kuriant semantinį registrą realiam naudojimui.

8. LITERATŪRA

- [1] BOOTH, D.; HAAS, H.; MCCABE, F. Web Services Architecture. ,
<http://www.w3.org/TR/ws-arch/> 2009 03 08
- [2] Qi Yu , Xumin Liu, Athman Bouguettaya, Brahim Medjahed. Deploying and managing
Web services: issues, solutions and directions. Springer-Verlag, 2006m. p. 541-542
- [3] UDDI - A Foundation for Web Services,
<http://www.idealliance.org/papers/xml2001/papers/html/03-02-03.html#d28e60352>
2009 11 03
- [4] UDDI Tutorial, http://www.tutorialspoint.com/uddi/uddi_elements.htm 2009 03 08
- [5] Yi Sun, Shaoyi He, Jack Y. Leu Syndicating. Web Services: A QoS and user-driven
approach. ELSEVIER, 2006m. , 243 – 245 p.
- [6] Web service http://en.wikipedia.org/wiki/Web_service 2009 03 08
- [7] Jiangang Ma and Yanchun Zhang . Finding web services on the WEB. School of
Computer Science & Mathematics, Victoria University, Australia. p. 1-2
- [8] WSDL tutorials <http://www.w3schools.com/WSDL/default.asp> 2009 03 08
- [9] SOAP protocol [http://en.wikipedia.org/wiki/SOAP_\(protocol\)](http://en.wikipedia.org/wiki/SOAP_(protocol)) 2009 02 02
- [10] Understanding WSDL in a UDDI registry
<http://www.ibm.com/developerworks/webservices/library/ws-wsdl/> 2009 02 02
- [11] OWL-S: Semantic Markup for Web Services
<http://www.w3.org/Submission/OWL-S/> 2009 02 02
- [12] Protégé OWL tutorial
<http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf> 2009 02 03
- [13] Boris Lublinsky. Implementing a Service Registry for .NET Web Services. CentraSite,
2008
- [14] Andrea D'Ambrogio. A Model-driven WSDL Extension for Describing the QoS
of Web Services. University of Roma TorVergata
- [15] Kim Christensen, Thorbjørn Højgaard Olesen, Lone Leth Thomsen. Matching
semantically described web service using ontologies, Denmark, p. 267 - 272
- [16] OWL Web Ontology <http://www.w3.org/TR/owl-features/> , 2009 03 08
- [17] Resource Description Framework <http://www.w3.org/RDF/> 2009 03 08
- [18] Philippe Larvet, Benoit Chirstope, Alain Pastor. Semantization of legacy web services:
From WSDL to SAWSDL, France , p. 130 - 134

- [19] David Martin. Semantic Web Services, Knowledge Media Institute, Open University, p. 8-15
- [20] Umesh Bellur, Harin Vadodaria and Amit Gupta. Semantic Matchmaking Algorithms, Department of Computer Science and Engineering, Indian Institute of Technology, Bombay, p. 481- 501
- [21] Okkyung Choi, Sangyong Han, Ajith Abraham. Semantic Matchmaking Services Model for the intelligent Web Services, Department of Computer Science & Engineering, Korea,p. 5-10
- [22] Umesh Bellur, Roshan Kulkarni. Improved Matchmaking Algorithm for Semantic Web Services Based on Bipartite Graph Matching, Kanwal Rekhi School of Information Technology, IIT Bombay, p. 1 - 9

9. TERMINŲ IR SANTRUMPŲ ŽODYNĖLIS

- **Web Service (WS)** - Pasaulio tinklo paslauga, tinklo paslauga, interneto paslauga, „web servisas“, servisas
- **UDDI (Universal Description, Discovery, and Integration)** - Universali aprašymo, radimo ir integravimo kalba
- **WSDL (Web Service Definition Language)** – tinklo paslaugų apibrėžimo kalba WSDL
- **XML (Extensible Markup Language)** - išplečiamoji žymėjimo kalba
- **WCF (Windows Communication Foundation)** – Microsoft projektavimo paketas, skirtas interneto servisų valdymui.
- **JAVA** - Bendros paskirties, aukšto lygio, objektiškai orientuota, nuo platformos nepriklausoma programavimo kalba sukurta Sun Microsystems firmoje.
- **UML** - Unifikuota modeliavimo kalba, naudojama objektiškai orientuotame projektavime (angl. Unified Modeling Language)
- **PĮ** - Programinė įranga
- **PA** - Panaudojimo atvejis
- **OWL** - Žinių atvaizdavimo kalba ontologijoms kurti, patvirtinta World Wide Web konsorciumo
- **SQL** - Struktūrizuota užklausų kalba (angl. Structured Query Language) - populiariausia iš šiuo metu naudojamų kalbų, skirtų aprašyti duomenis ir manipuliuoti jais reliacinių duomenų bazių valdymo sistemose
- **API (Application Programming Interface)** - Tai yra sąsaja, kuria viena programa sąveikauja su kita programa

10. PRIEDAI

10.1. Testavime ir eksperimente naudotos viešbučių ontologijos schema



69 pav. Viešbučių ontologijos schema

10.2. Eksperimento duomenys

6 lentelė

Paieškos parametrų variantų rinkiniai

	Funkcijos par.	Įėjimo par.	Išėjimo par.
1	viesbuciai	viesbutis	kaina
2	viesbuciu paieška	viebustis	adresas, kaina
3	kambarioRegistravimas	viesbutis, laikas	Vieta kaina

Eksperimento rezultatai

Band sk.	Var. Nr.	Paslaugų sk.	Op. sk.	Par. sk.	Vid. Laikas (ms)
1	1	1	2	4	2.92
4	1	2	5	13	6.25
7	1	4	10	26	10.19
10	1	6	20	17	16.42
13	1	8	25	70	17.84
16	1	10	33	92	22.71
19	1	12	41	128	27.21
22	1	14	51	159	31.36
25	1	16	60	186	35.28
2	2	1	2	4	2.59
5	2	2	5	13	5.53
8	2	4	10	26	7.59
11	2	6	20	17	13.69
14	2	8	25	70	16.45
17	2	10	33	92	21.09
20	2	12	41	128	27.42
23	2	14	51	159	32.66
26	2	16	60	186	38.77
3	3	1	2	4	2.5
6	3	2	5	13	4.42
9	3	4	10	26	8.75
12	3	6	20	17	15.49
15	3	8	25	70	19.71
18	3	10	33	92	25.31
21	3	12	41	128	36.01
24	3	14	51	159	41.32
27	3	16	60	186	50.54