KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACINIŲ SISTEMŲ INŽINERIJOS STUDIJŲ PROGRAMA

AURIMAS GUDAS

AUTOMATIZUOTAS LIETUVIŠKO TEKSTO SEMANTINIS ANOTAVIMAS

Magistro darbas

Darbo vadovas
doc. dr. Rita Butkienė

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACINIŲ SISTEMŲ INŽINERIJOS STUDIJŲ PROGRAMA

AURIMAS GUDAS

AUTOMATIZUOTAS LIETUVIŠKO TEKSTO SEMANTINIS ANOTAVIMAS

Magistro darbas

Darbo vadovas:
doc. dr. Rita Butkienė
2013-05-22

Recenzentas:
prof. dr. Gintaras Palubeckis
2013-05-22

Atliko:
IFM-1/4 gr. studentas
Aurimas Gudas
2013-05-22

KAUNAS, 2013

AUTORIŲ GARANTINIS RAŠTAS
DĖL PATEIKIAMO KŪRINIO

2013 - 05 - 22 d.

Kaunas

Autoriai,_____

(vardas, pavardė)

_____,

patvirtina, kad Kauno technologijos universitetui pateiktas baigiamasis bakalauro (magistro) darbas (toliau vadinama -Kūrinys)_____

(kūrinio pavadinimas)

_____

pagal Lietuvos Respublikos autorių ir gretutinių teisių įstatymą yra originalus ir užtikrina, kad

1) jį sukūrė ir parašė Kūrinyje įvardyti autoriai;
2) Kūrinys nėra ir nebus įteiktas kitoms institucijoms (universitetams) (tiek lietuvių, tiek užsienio kalba);
3) Kūrinyje nėra teiginių, neatitinkančių tikrovės, ar medžiagos, kuri galėtų pažeisti kito fizinio ar juridinio asmens intelektinės nuosavybės teises, leidėjų bei finansuotojų reikalavimus ir sąlygas;
4) visi Kūrinyje naudojami šaltiniai yra cituojami (su nuoroda į pirminį šaltinį ir autorių);
5) neprieštarauja dėl Kūrinio platinimo visomis oficialiomis sklaidos priemonėmis.
6) atlygins Kauno technologijos universitetui ir tretiesiems asmenims žalą ir nuostolius, atsiradusius dėl pažeidimų, susijusių su aukščiau išvardintų Autorių garantijų nesilaikymu;
7) Autoriai už šiame rašte pateiktos informacijos teisingumą atsako Lietuvos Respublikos įstatymų nustatyta tvarka.

Autoriai

_____

_____ (vardas, pavardė) (parašas)

_____

_____ (vardas, pavardė) (parašas)

_____

_____ (vardas, pavardė) (parašas)

_____

_____ (vardas, pavardė) (parašas)

SANTRAUKA

Pagrindinis šio darbo tikslas - išanalizuoti lietuvių kalbos semantinio anotavimo procesą ir sukurti metodologiją, leisiančią įgyvendinti semantinio anotavimo proceso automatizavimą. Kompiuteris, kitaip nei žmogus, nesupranta ar tekstas yra rišlus ir prasmingas, ar neturi jokio rišlumo ir prasmės. Tai ypač atsiliepia verčiant tekstus, lyginant tos pačios kalbos tekstus vieną su kitu ir t.t. Semantinis anotavimo procesas leidžia išspręsti šią problemą sukurdamas metodiką, kuri leidžia aprašyti žodžių sąsajas sakiniuose, tačiau norint tekstą semantiškai suanotuoti rankiniu būdu, reikia turėti specifinių žinių ir tai reikalauja didelių laiko sąnaudų. Norint išvengti šių problemų semantinį procesą būtina automatizuoti.

Šiame darbe buvo išanalizuotas lietuvių kalbai tinkantis semantinio anotavimo procesas, sukurta metodologija, leidžianti įgyvendinti semantinio anotavimo proceso automatizavimą. Metodologijos pagrindu JAVA programavimo kalba įgyvendintas automatizuoto semantinio anotavimo proceso realizavimas, atliktas eksperimentas ir pateiktos išvados.

SUMMARY

Automated Semantic Anotation of the Lithuanian Text

Major aim of this work – analyze Lithuanian language semantic annotation process and develop methodology which let implement automate semantic annotation process. The computer, unlike the person, does not understand did the text is coherent and meaningful, or have no coherence and meaning. This is particularly vulnerable to the translation of the text, compared with the same language texts with one another, etc. Semantic annotation process allows to solve the problem of creating a methodic that enables to describe links between words in sentences, but in order to semantically annotate the text manually, you need to have specific knowledge and it requires time-consuming. To avoid these problems it is necessary to automate the process of semantic.

In this work was analyzed the Lithuanian language suitability for semantic annotation process, developed methodology to implement semantic annotation of process automation. Methodology based Java programming language implementation of automated semantic annotation process realization of the experiment was conducted and the following conclusions.

Turinys

ĮVADAS

Darbas parengtas pagal Informacinių sistemų inžinerijos magistrantūros studijų programos reikalavimus.

Internete publikuotame lietuviškame tekste, kaip ir kiekvienos kitos kalbos tekste, susidaro dviprasmybių. Žmonėms jas susieti į vieną rišlų tekstą nėra sunkų, tačiau kompiuteris to padaryti negali, nes jis nesupranta, kaip žodžiai susieti vieni su kitais. Tai ypač atsiliepia verčiant tekstus, lyginant tos pačios kalbos tekstus vieną su kitu ir t.t. Kompiuteris, atlikdamas šias operacijas, daro aiškias logines kalbos klaidas, nors pats to nesupranta. Šiuo metu galimas sprendimo būdas yra semantinė anotacija. Ji nurodo, kokie žodžiai gali būti susieti vieni su kitais ir padeda įveikti žodinės kalbos dviprasmybes. Natūralios kalbos dviprasmybės - tai reiškinys, kai žodis gali turėti keletą visiškai skirtingų reikšmių, pvz., Kaunas gali būti ir miestas ir žmogaus vardas.

Darbo tikslai - išanalizuoti semantinio anotavimo procesą, tinkantį lietuvių kalbai, remiantis išanalizuotu procesu sukurti metodologiją, galinčią automatizuoti semantinio anotavimo procesą.

Darbo uždavinys - remiantis metodologijos pagrindu sukurti programą ir jos veikimą patikrinus eksperimentu, pateikti rezultatus.

Darbo svarba - tausojant žmogiškuosius išteklius atlikti lietuviško teksto semantinį anotavimą kompiuteriu.

Dokumento struktūra: Teksto semantinio anotavimo proceso ir įrankių analizė.

Sprendimų reikalavimų analizė ir projektavimas.

1. TEKSTO SEMANTINIO ANOTAVIMO PROCESO IR ĮRANKIŲ ANALIZĖ

1.1.Analizės tikslas

Semantinis teksto anotavimas rankiniu būdu - ilgas procesas ir norint tekstą suanotuoti semantiškai reikia specifinių žinių. Semantinio proceso automatizavimas reikalingas tam, jog išspręstų specifinių žinių ir laiko sąnaudų problemas. Automatizavus semantinį anotavimą programos vartotojui teliks nurodyti kokį tekstą jis nori semantiškai suanotuoti, o už jį tai padarys programa.

Analizės tikslas - išanalizuoti semantinio anotavimo procesą tinkantį lietuvių kalbai ir įrankius, kurių pagalba būtų įgyvendintas semantinio anotavimo proceso automatizavimas. Norint išanalizuoti semantinio anotavimo įrankius ir kitus galimus tyrimo problemos sprendimo variantus, bus taikomas dalykinės srities tyrimo metodas. Analizės metu bus nagrinėjami įvairūs semantinio anotavimo įrankiai ir kiti problemos sprendimo variantai siekiant išsiaiškinti kuris įrankis geriausiai tinka automatiškai anotuoti lietuvišką tekstą semantiniu būdu. Siekiant gauti kuo tikslesnius rezultatus analizės metu įrankiai bus lyginami vienas su kitu. Pagrindinis kriterijus pagal kurį bus lyginami įvairūs problemos sprendimo variantai - įrankio arba kito problemos sprendimo būdo pritaikomumas lietuvių kalbos semantiniam anotavimui.

Tyrimo objektas - teksto semantinio anotavimo procesas.

Tyrimo sritis - teksto semantinio anotavimo proceso automatizavimas, esami semantinio anotavimo įrankiai, formatai, standartai ir kitos priemonės.

Tyrimo problema - kompiuteris, savo nuožiūra, negali suprasti ar tekstas yra rišlus ir prasmingas ar neturintis jokios prasmės ir jokio rišlumo. Tai labai smarkiai atsiliepia tekstų vertimams ir t.t. Norint, išspręsti šią problemą reikia sukurti priemonę, kuria naudojantis kompiuteris galėtų atskirti prasmingą tekstą nuo teksto neturinčio jokios prasmės. Sprendimo realizacija ir testavimas. Eksperimentinis sprendimo tyrimas. Rezultatų apibendrinimas ir išvados. Literatūra.

1.2.Semantinio anotavimo tyrimo objekto analizė

Anotavimo paaiškinimas

Anotacija arba žymėjimas yra atributų, nuorodų ir t.t. pridėjimas prie dokumento arba prie pasirinktos teksto vietos. Pridėtoji informacija teikia papildomą informaciją (metaduomenis) apie dokumento gabalus. Egzistuoja trys pagrindiniai teksto anotavimo tipai semantinis, morfologinis ir sintaksinis.

Semantinio anotavimo paaiškinimas

Semantinis anotavimas - teksto žymėjimas remiantis metodika, kurios pagalba galima išspręsti natūralioje kalboje pasitaikančias dviprasmybes.

Semantinio anotavimo proceso esmė - teksto dviprasmybių panaikinimas.

Natūralios kalbos dviprasmybės - tai reiškinys, kai žodis gali turėti keletą visiškai skirtingų reikšmių, pvz., Kaunas gali būti ir miestas ir žmogaus vardas.

Semantinis anotavimo procesas susieja anotavimui skirto teksto žodžius, frazes su ontologijoje esančios apibrėžtos srities informacija. Semantinio anotavimo procesas, vadovaudamasis lietuvių kalbos gramatika, suskirsto tekstą į smulkesnes dalis pvz. frazes ar žodžius ir priskiria jiems ontologijoje esančius duomenis (tiksli semantinio anotavimo proceso schema pateikta 4 pav.). Susiejąs žodžius ir frazes su ontologijos informacija semantiškai anotuotas tekstas yra išsaugojamas faile kuris yra paruoštas naudojimui.

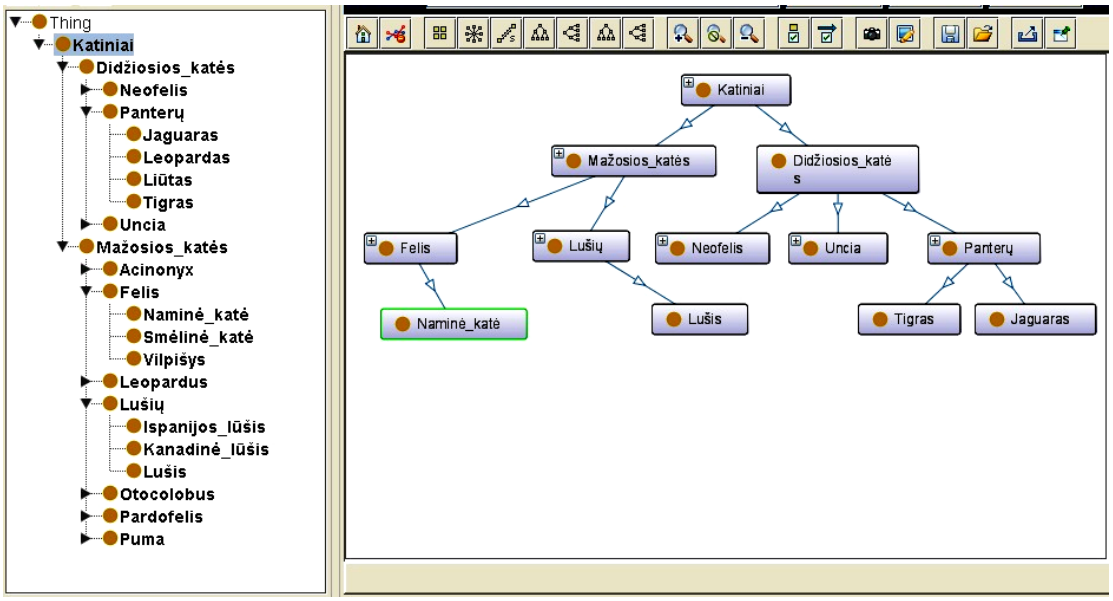
Ontologijos paaiškinimas

Bendrinis ontologijos apibrėžimas būtų toks: "ontologija - tam tikros srities sąvokų visumos specifikavimas išreikštu pavidalu" (merriam-webster žodynas).

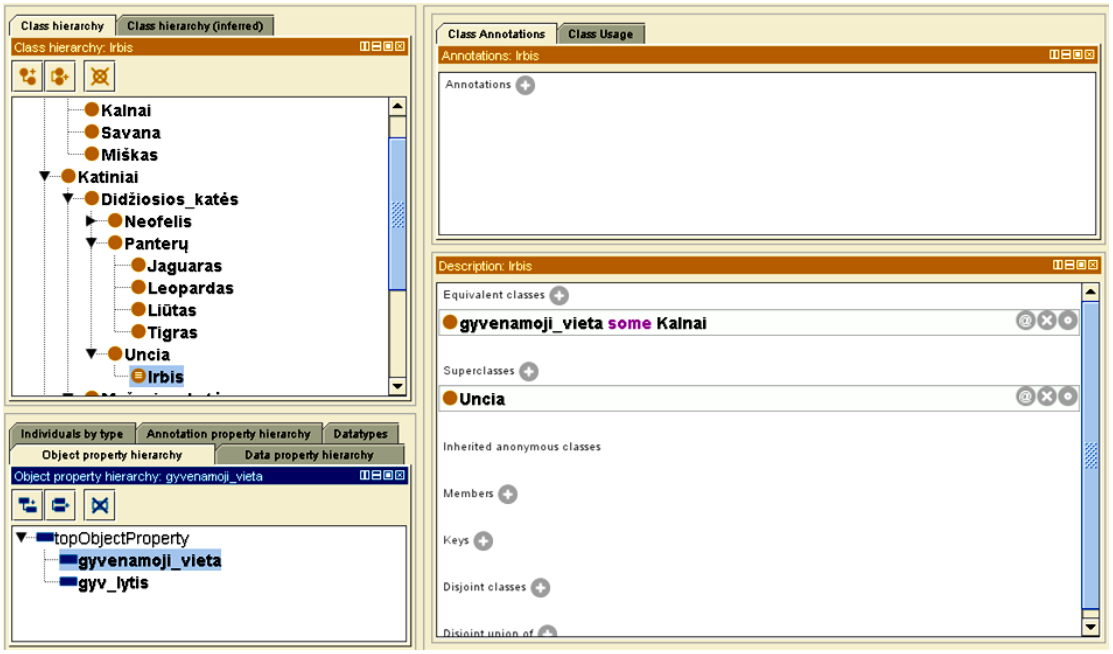
Ontologijos vaidmuo semantiniame anotavime

Ontologijos vaidmuo semantiniame teksto anotavime yra identifikuoti kokios ontologijos klasę ar jos egzempliorių išreiškia nagrinėjamas teksto žodis, su kokiais žodžiais jis gali būti susietas ir su kokiais negali. Pavyzdžiui liūtas yra didžiųjų kačių pošeimio atstovas. Anotavimo procesas susieja žodį liūtas su ontologija, kuri savo ruožtu identifikuoja liūtą kaip katinių šeimos atstovą, priklausantį didžiųjų kačių pošeimei.

Norimos srities žodį arba žodžių grupę galima susieti ne tik pagal aukštesnio lygio klasę, bet ir pagal kitas klases, su kuriomis nagrinėjamas žodis ar žodžių aibė įgautų didesnę prasmę. Pavyzdžiui katinių šeimos atstovų rūšis būtų tikslinga apjungti su tos rūšies gyvenamąja vieta, nes toks apjungimas padidintų žodžių identifikacijos tikslumą. Su kitomis klasėmis žodis arba žodžių aibė gali būti susieta naudojantis objektų savybių ryšiais. Katinių šeimos atstovo Irbio (objekto) susietumas su jo gyvenamąja vieta (subjektu) naudojant kardinalmo ryšį parodytas 2 paveikslėlyje.



1 pav. Katinių šeimos atstovų ontologijos objektų grafinis vaizdas



2pav. Katinių šeimos atstovo Irbio taisyklių aprašymas

Ontologijos kūrimo ir semantinio anotavimo procesas vykimas:

1. Naudojant ontologijos kūrimo įrankį sukuriama nagrinėjamos žodžių srities klases. Klasės sugrupuotos medžio principu, pradedant stambiausiomis,

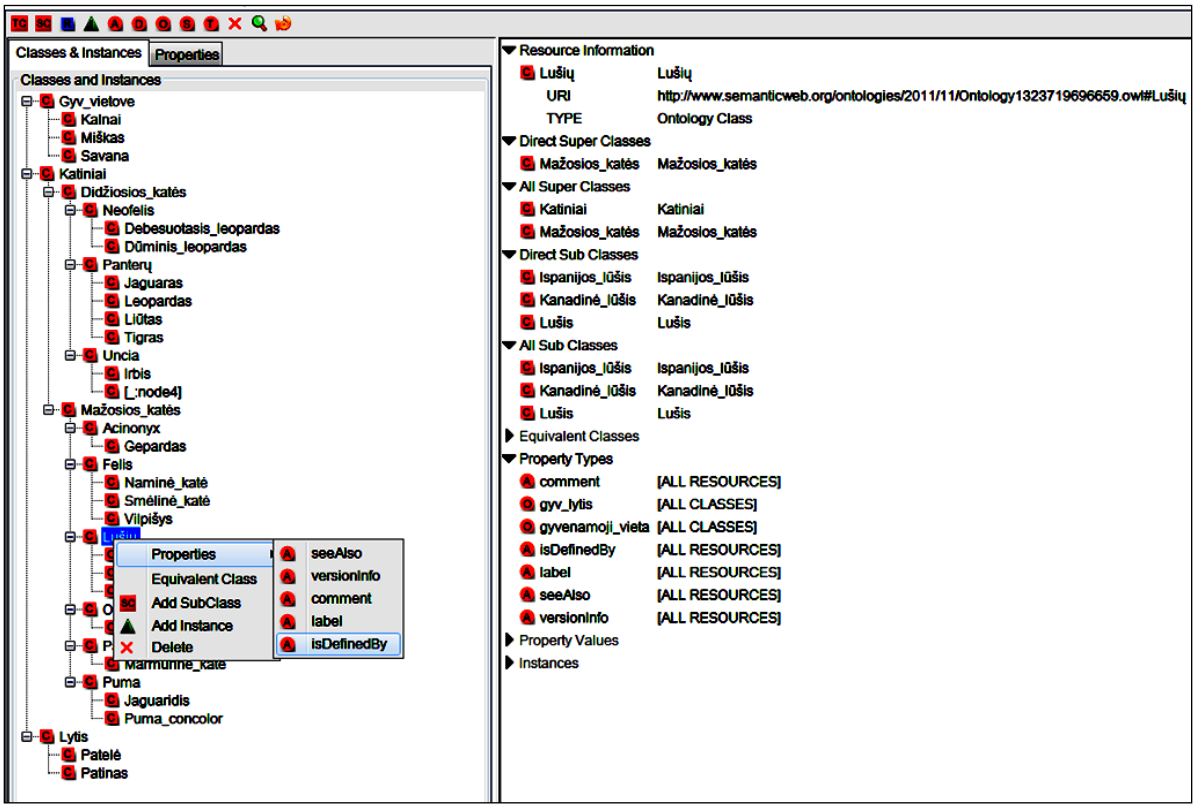
labiausiai apibendrintomis klasėmis ir baigiant smulkiausiomis, tikslinėmis klasėmis. Šuo atveju pasirinkta sritis - visa informacija susijusi su katinių šeimos atstovais.

2. Sukuriamos tikslinančios klasės. Šuo atvejų tai galėtu būti gyvenamosios vietos, amžiaus ir panašios klasės.
3. Sukuriamos klasių taisyklės.
4. Sukurta ontologija išsaugojama RDF formatu.
5. Į serverį, pritaikytą RDF formato palaikymui importuojama ontologija.
6. Anotavimo įrankis susiejamas su ontologiją palaikančiu serveriu.
7. Pasirenkamas tekstas, kuri norima anotuoti (duomenų atpažinimo procesas atpažįsta visus internetinius straipsnius HTML ir XML formatais)
8. Importuojant tekstą į semantinio anotavimo įrankį nurodomas tekstinio failo kodavimas. Tekstams, parašytiems lietuvių kalba, reikia nurodyti UTF-8 formatą.
9. Tekstas turi būti sulemojamas t.y. visoms žodžių formoms turi būti nurodomos jų pagrindinės formos (veiksmažodžio bendratis, daiktavardžio kilmininkas ir t.t.)
10. Paleidžiamas anotavimo procesas (procesas pateiktas 4 paveikslėlyje)
11. Sukuriama direktorija saugoti anotuotus dokumentus.
12. Pažymimos anotuojamos dokumento vietos
13. Sukuriamas semantinių reikšmių failas kiekvienam anotuotam dokumentui. Per semantinės reikšmės atskleidimo failą kiekvienas procesorius palaikantis ontologijas gali suprasti tikslinę žodžio reikšmę.

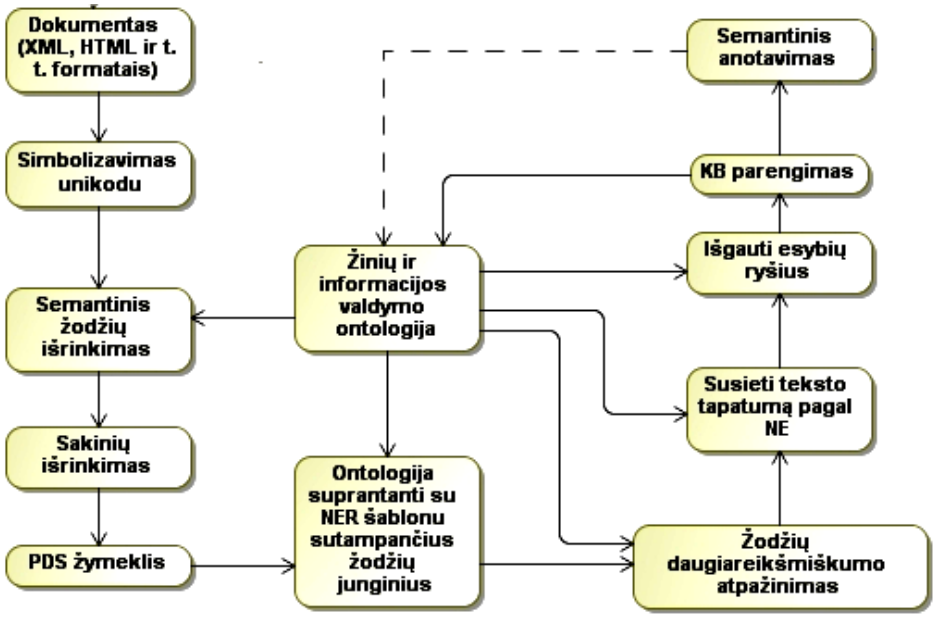
Pastabos:

Jeigu pasirenkama daug tekstų, kuriuos norima anotuoti, juos galima grupuoti saugyklose.

Sudarinėjant ontologijos taisykles gali neišeiti apsvarstyti visų galimų atvejų ir anotuojant dokumentą rezultatai gali netenkinti keliamų reikalavimų. Tačiau anotavimo įrankiai leidžia redaguoti sudarytą ontologiją ir jos taisykles. Tokiu būdu galima patikslinti ontologiją. Pavyzdys 3 paveikslėlyje.



3 pav. Ontologijos redagavimas naudojantis anotavimo programa



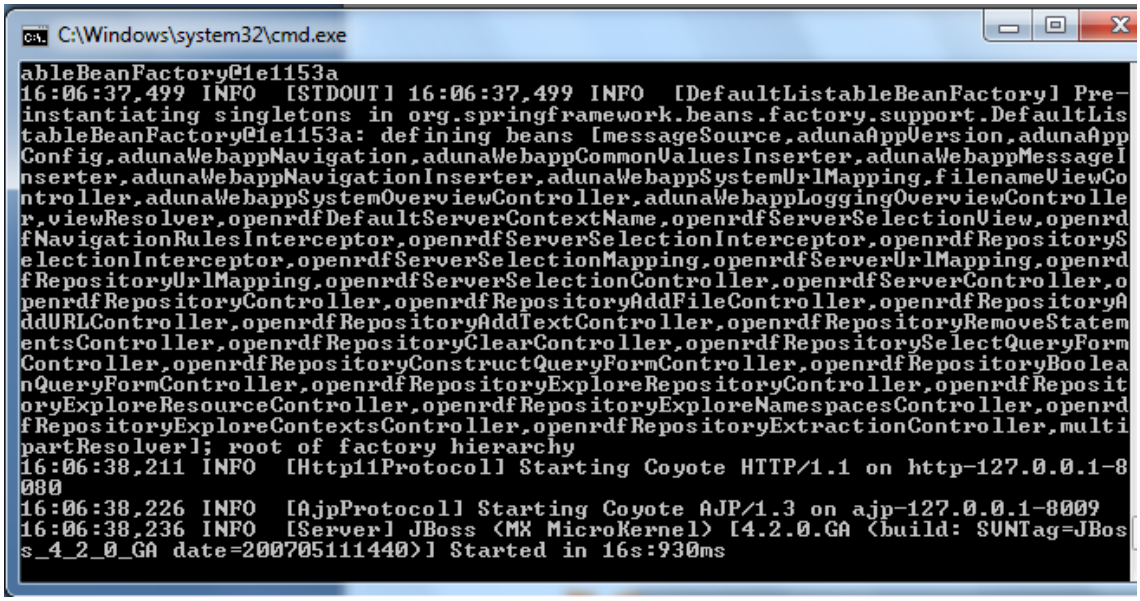
4 pav. Semantinio teksto anotavimo procesas

1.3. Ontologijos palaikymui skirtų įrankių sukonfigūravimas

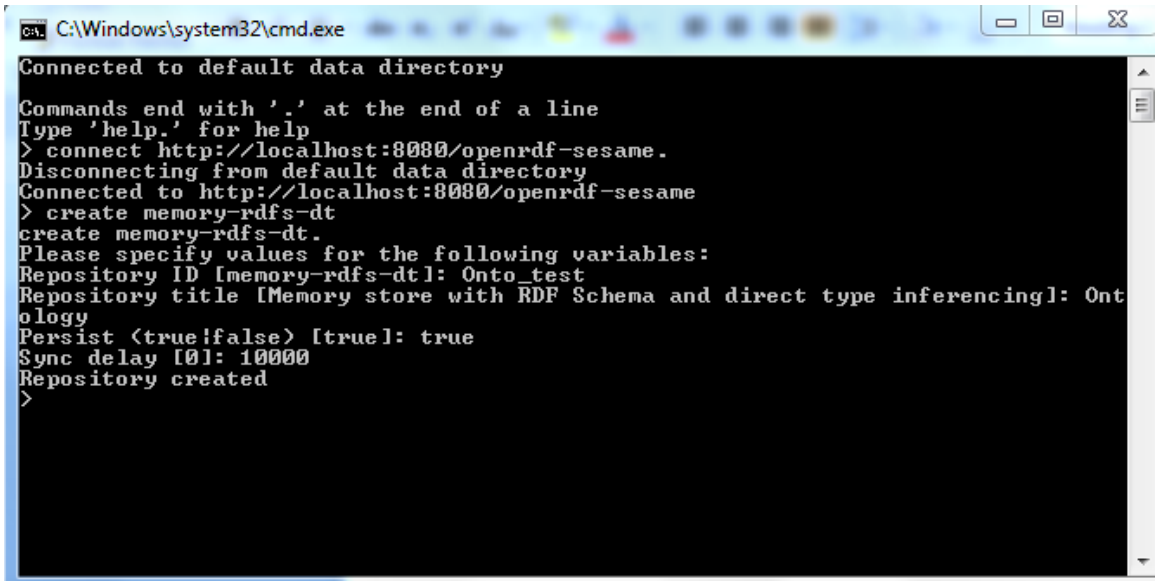
Norint, jog programa galėtų naudoti ontologiją reikia sukonfigūruoti įrankius, skirtus ontologijos palaikymui.

Iš pradžių ontologiją reikia susikurti naudojant ontologijos kūrimo programą pvz. „Protege“ ar kt., tuomet galime apjungti ją su anotavimo įrankiu. Norint tai padaryti reikia serverio, į kuri būtų galima patalpinti ontologiją. Tai padaryti galima pasileidžiant „Sesame“ serverį (jis yra ne kas kita, kaip atviro kodo Java framework‘as). Šis serveris skirtas operuoti su RDF/XML tipo failais, taip pat jame galima rašyti užklausas SeRQL arba SPARQL formatais norint išgauti duomenis tam tikrais skerspjūviais. „Protege“ leidžia kurti tokio formato failus. Tačiau vien „Sesame“ serverio nepakanka. Šį serverį reikia importuoti į serverį, į kurį integruotas „Apache“ ir yra skirtas kitų serverių arba servletų arba aplikacijų palaikymui. Importavimas vyksta įkelianti openrdf-sesame.war ir openrdf-workbench.war failus į atitinkamą vietą (priklauso koks aplikacijų serveris naudojamas). Populiariausi aplikacijų serveriai yra „Tomcat“ arba „J-Boss“.

Norint, jog „Sesame“ serveris ir jį palaikantis aplikacijų serveris būtų paleisti reikia paleisti jų vykdomuosius .bat failus. „Sesame“ serveriui šis failas randasi {Sesame serverio būvimo vieta}\bin\start-console.bat (jei serveris naujesnis .bat failas vadinis „console.bat“). Kai „Sesame“ serveris paleistas reikia paleisti aplikacijų serverį. „J-Boss“ serverio .bat failo buvimo vieta „{J-Boss serverio būvimo vieta}\{j-boss versija}\bin\run.bat“ o „Tomcat“ - „{Tomcat serverio būvimo vieta}\{tomcat}\bin\startup.bat“. Paliesti abudu serveriai matomi 5-ame ir 6-ame paveikslėliuose. Reikia pastebėti 6-ame paveikslėlyje „Sesame“ serveris ne tik paleistas, bet ir prijungtas prie „J-Boss“ serverio per nurodytą portą (nesvarbu ar naudojamas „J-Boss“ ar „Sesame“ serveris, yra išskiriamas 8080 portas, per kurį vyksta prisijungimas pie „Sesame“ serverio) ir sukurtas repositorius į kurį bus galima kelti ontologiją. Naujesnėse „Sesame“ serverio versijose reopsitorius galima sukurti nesinaudojant komandine eilute.

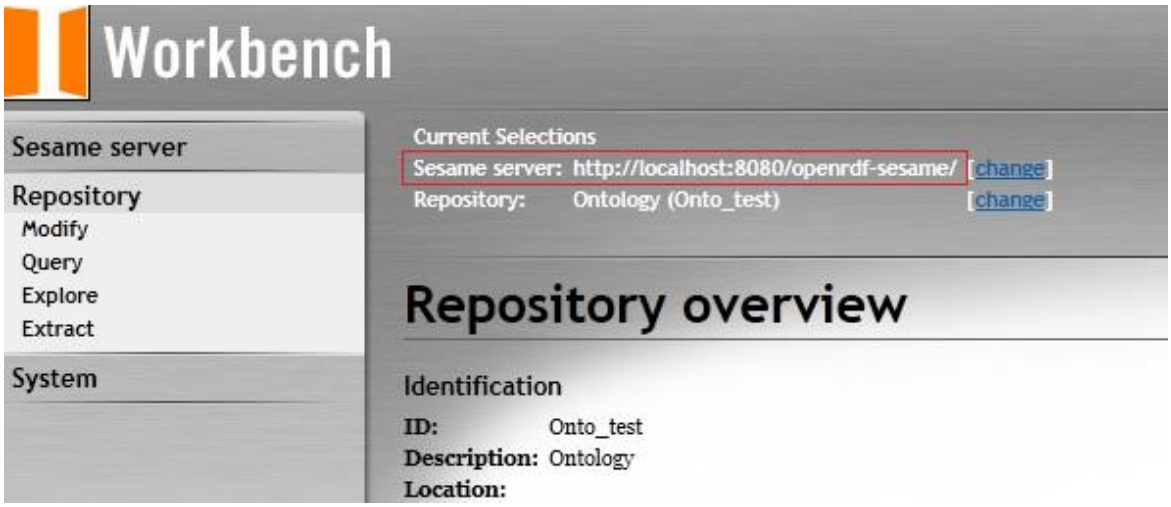


5 pav. „J-Boss“ serverio paleidimas



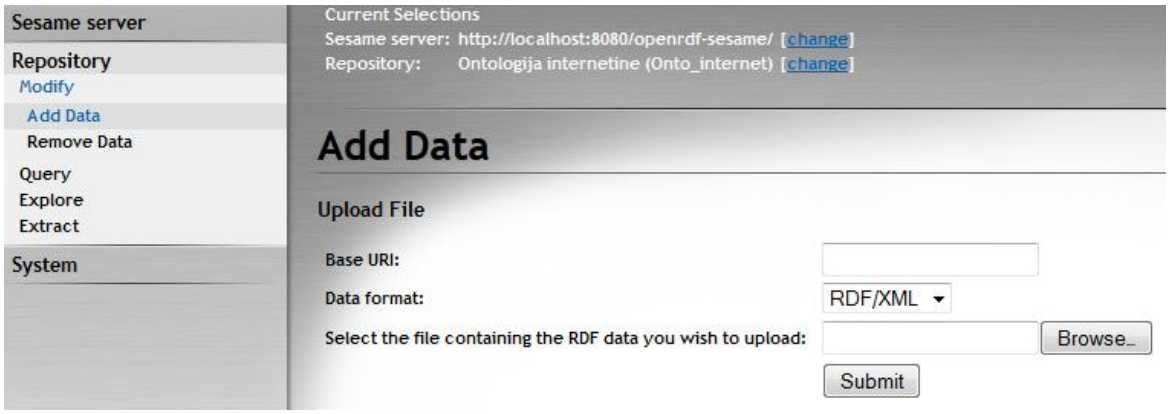
6 pav. „Sesame“ serverio paleidimas

Sėkmingai paleidus šiuos serverius gaunamas vaizdas, kurį matome 7-ame paveikslėlyje. Nurodytas „Sesame“ serverio adresas ir sukurtas repositorius pavadinimu „Onto_test“.

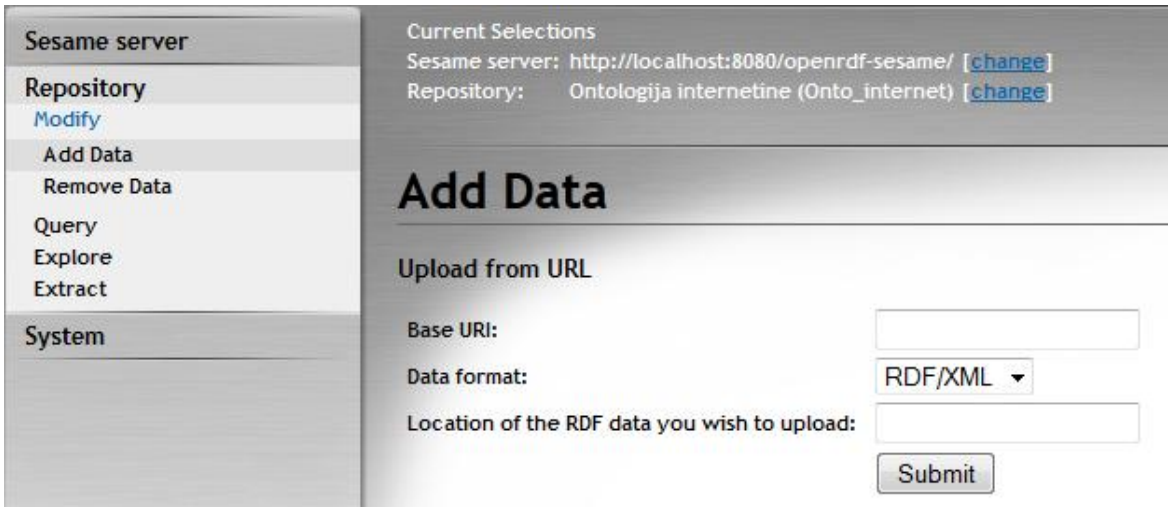


7 pav. „Sesame“ serverio adresas

Kai pastarieji veiksmai atlikti reikia įkleti ontologiją. Ontologijos įkėlimas pademonstruotas 8-ame ir 9-ame paveikslėliuose. Skirtumas tik toks, kad 8-ame paveikslėlyje ontologija importuota iš standžiojo disko, o 9-ame paveikslėlyje ontologija importuota iš internetinio puslapio.



8 pav. Ontologijos importavimas į „Sesame“ serverį iš standžiojo disko



9 pav. Ontologijos importavimas į „Sesame“ serverį iš URL

1.4. Vartotojų analizė

1.4.1. Vartotojų aibė, tipai ir savybės

Pasirinktos kalbos srities (šuo atveju informacija apie katinių šeimos atstovus) žodžių susiejimas vienas su kitu dalykinės srities specialistams neturėtų būti sudėtingas, kadangi tai yra kalba, su kurią jie dirba kiekvieną dieną. Programuotojams pasirinktos žodžių srities skirstymas į siauresnes sritis t.y. į poklases ir kintamųjų aprašymas, kardinalumų nustatymas (pagal kalbos specialistų duotus žodžių sąryšius) neturėtų kelti problemų.

Vartotojo tipas	**Savybės**
Lietuvių kalbos specialistai	Dalykinės srities specialistai, išmanantys sritį. Nemokantys programuoti, bet turintys žinių, kokiu būdu geriausia susiet žodžius
Programuotojai	Žinantys kaip operuojama su klasėmis, kaip apsirašo kintamieji, kaip nustatinėti kardinalumus ir t.t.

1 lent. Vartotojo tipų ir savybių aprašymas

1.4.2. Vartotojų tikslai ir problemos

Lietuvių kalbos specialistams svarbu, kad užduotys jiems būtų pateikiamos kiek įmanoma artimesnės jų įprastinėms darbo užduotims ir kad jose nebūtų nieko, kas susiję su programavimu.

1.5. Esamų sprendimų analizė

Vieni iš populiariausių semantinio anotavimo įrankių yra "Gate" , "MnM", buvo pasirinkti kaip geriausi semantinio anotavimo įrankiai.

Teiginys	Sistema	
	Gate	MnM
Automatinis puslapių anotavimas	+	+
Pusiau automatinis puslapių anotavimas	+	+
Rankinis puslapių anotavimas	+	+
Anotavimui skirtu tekstų įkėlimas iš failo	+	+
Prisijungimas prie anotavimui skirtų tekstų per serverį	+	+
Ontologijos importavimas iš failo	+	+
Prisijungimas prie ontologijos per serverį	+	+
Failų formatų palaikymas	+	-
Importuotos ontologijos koregavimas	+	+
Populiarumas	+	-
Tekstinės informacijos apie įrankį gausa	+	+
Vaizdinės informacijos apie įrankį gausa	+	-
Galimybė peržvelgti kalbų ir vykdomuosius resursus	+	-
Aiški darbo aplinka	9	7

2 lent. "Gate" ir "MnM" anotavimo įrankių palyginimas

Simbolis	Paaiškinimas
+	Sistema tenkina teiginį.
-	Sistema netenkina teiginio.

0-10	Įvertinimas nuo 0 iki 10, kaip sistema atitinka teiginį. 0 – neatitinka; 10 – pilnai atitinka;

3 lent. Įrankių palyginimo lenteles paaiškinimai

Kitas variantas - sukurti Web servisą, kuris suanotuotų vartotojo siunčiamą tekstą turimos ontologijos pagrindu.

Web serviso privalumai:

Pagrindinis Web serviso privalumas - nepriklausomumas. Semantinio anotavimo Web servisą galėtų naudoti įvairaus tipo programos. Programose reikėtų minimalaus kodo įterpimo, jog Web servisą galima būtų panaudoti tiek "MnM" ir "Gate", tiek kituose įrankiuose nepriklausomai nuo jų technologijos. Web servisai su programomis bendrauja per struktūrizuotus protokolus, kuriems neaktualu, kokios technologijos pagrindu sukurta programa. Sukūrus Web servisą tikėtina, kad:

- Bus žymiai didesnės jo panaudojimo galimybės;
- Bus galima Web servisą panaudoti visose atviro kodo programose;
- Neapkraunamas kliento kompiuteris. Visus veiksmus atliks serveris, kuriame bus Web servisas;

Web serviso trūkumai:

Pagrindinis Web serviso trūkumas - bet kokiai programai, kuri ketina naudoti semantinio anotavimo Web servisą, reikės minimalaus kodo įterpimo, kurį turės atlikti vartotojas.

Pastabos:

Kadangi lietuviška abėcėlė skirias nuo angliškos, importuojant planuojama anotuoti tekstą į "Gate" įrankį, reikia nurodyti jo kodavimo formatą UTF-8.

Atsižvelgus į sprendimų analizės išvadas, pasirinkta kurti semantinio anotavimo Web servisą.

1.6. Siekiamas sprendimas

Šiame darbe siekiama sukurti lietuvių kalbos semantinio anotavimo sprendimą, kuris turėtų kuo didesnias pritaikymo galimybes. Reikalinga apsvarstyti 2 atvejus:

1. Web serviso kūrimas
2. Atviro kodo anotavimo programos modernizavimas taip, jog šis atpažintų lietuvių kalbos taisykles ir galėtų semantiškai anotuoti lietuvišką tekstą.

1.7. Darbo tikslas ir uždaviniai

Darbo tikslas - sudaryti sąlygas efektyviau semantiškai anotuoti lietuviškus tekstus ir tuo prisidėti prie lietuvių kalbos įsitvirtinimo pasauliniame semantiniame tinkle.

Darbo uždaviniai:

1. Išanalizuoti semantinio teksto anotavimo įrankius.
2. Nustatyti semantinių anotavimo įrankių tinkamumą lietuviško teksto semantiniam anotavimui.
3. Išanalizuoti semantinio anotavimo proceso tinkamumą lietuvių kalbai.
4. Sukurti metodologiją, kurios pagrindu bus realizuotas semantinio anotavimo proceso automatizavimas.
5. Remiantis metodologija pasirinkti tinkamas, o jei trūksta, sukurti naujas priemones ir pademonstruoti jų veikimą.
6. Įvertinti darbo rezultatus.

1.8. Rizikos faktorių analizė

Kadangi semantinio anotavimo proceso automatizavimas sudėtingas reiškinys, gali kilti sunkumų kuriant metodologiją.

Automatizuotas semantinio anotavimo proceso algoritmas gali pilnai neišnaudoti ontologijos.

1.9. Rezultato kokybės kriterijai

Rezultato kokybes kriterijai:

- Automatizuotas semantinio anotavimo procesas sugebėti operuoti su visais ontologijoje esančiais objektais, subjektais, predikatais ir reikiamais jų atributais, tokiais kaip rdfs:label
- Automatizuotas semantinio anotavimo procesas turi atpažinti subjektą, objektą, predikatą ne tik kaip atskirą žodį, bet ir kaip keletą žodžių, einančių vienas šalia kito.
- Norint užtikrinti greitaveiką semantinio anotavimo procesas pagal žodžio linksnius turi atpažinti norimo anotuoti teksto objektus ir subjektus.

1.10. Panašių sistemų (Lietuvos ir tarptautiniu mastu) analizė

Norint semantiškai anotuoti lietuvišką tekstą reikia, kad įrankis sugebėtų suprasti lietuvišką tekstą naudodamasis lietuvių kalbos taisyklėmis. Tokio įrankio Lietuvoje nėra sukurta. Užsienyje taip pat nėra sukurta, nes kitos šalys rūpinasi savo šalies anotavimo įrankiais.

1.11. Analizės išvados

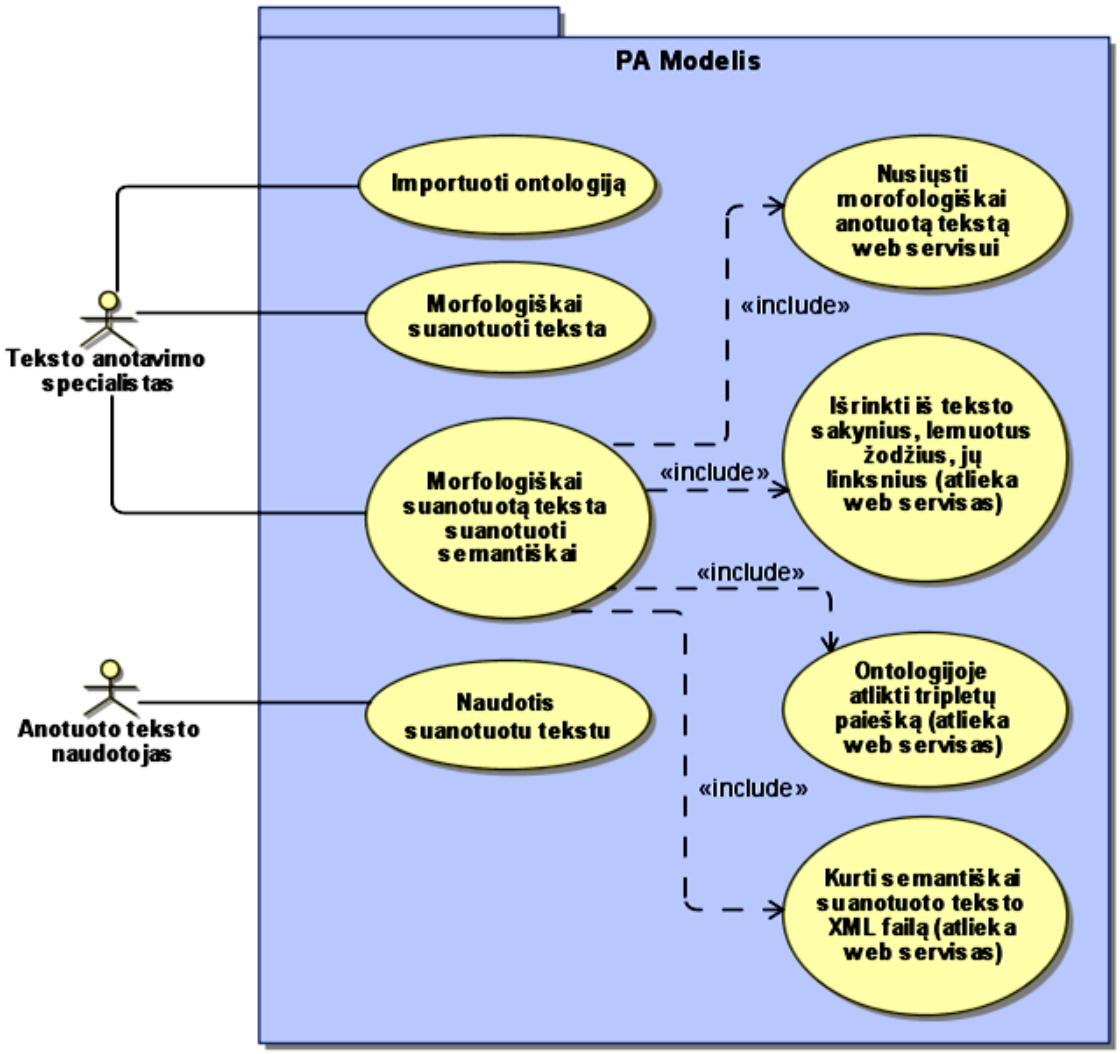
1. Išanalizavus ir palyginus semantinius anotavimo įrankius ir apsvarsčius visus galimus variantus buvo nuspręsta kuri web servisą, kuris atliktų semantinį teksto anotavimą ontologijos pagrindu.

2. SPRENDIMO REIKALAVIMŲ SPECIFIKACIJA IR PROJEKTAS

2.1. Reikalavimų specifikacija, funkciniai reikalavimai

Panaudojimo atveju modelis

Panaudojimo atvejų modelyje (10 pav.) pateikti visi semantinio anotavimo proceso panaudojimo atvejai, taip pat ontologijos, norimo anotuoti teksto ir lietuvių kalbos gramatikos panaudojimo atvejai.



10 pav. semantinio anotavimo proceso PA modelis

Kompiuterizuojamų panaudojimo atvejų paaiškinimai:

PA numeris	1.
Vartotojai, kurie vykdo PA	Teksto anotavimo specialistas
PA aprašas	
PA tiekimo kriterijus	
PA scenarijus	1. Sesame serveryje pasirinkti Add (Modify skiltyje) 2. Pasirinktu ontologiją RDF data file 3. Spausti mygtuką "Upload"

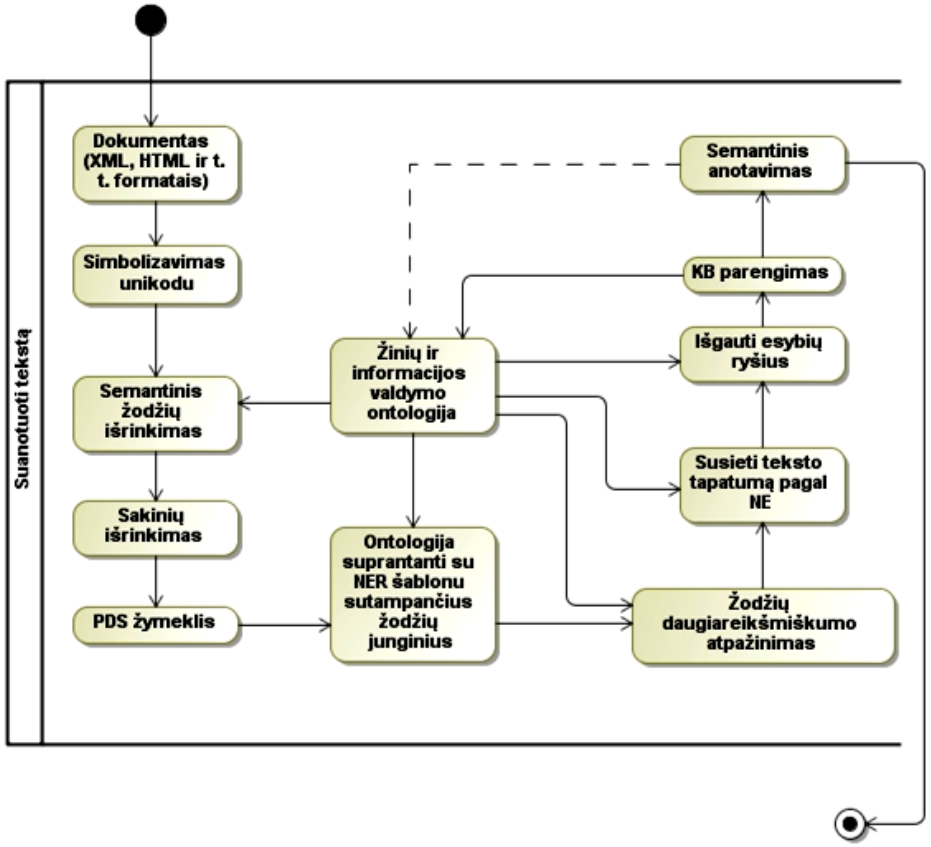
PA numeris	2.
Vartotojai, kurie vykdo PA	Teksto anotavimo specialistas
PA aprašas	

PA tiekimo kriterijus	Paspaudus mygtuką "Anotuoti tekstą" paleidžiamas semantinio anotavimo procesas, kartu ir jam priklausanti semantinių frazių išrinkimo funkcija
PA scenarijus	1. Paspausti teksto anotavimo mygtuką

PA numeris	3.
Vartotojai, kurie vykdo PA	Teksto anotavimo specialistas
PA aprašas	
PA tiekimo kriterijus	Kliento aplikacijoje paspausti mygtuką, kuris paleidžia teksto semantinio anotavimo procesą. Šis procesas vyksta web servise su kuriuo komunikuoja kliento aplikacija
PA scenarijus	1. Paspausti mygtuką "Anotuoti tekstą"

Veiklos diagrama

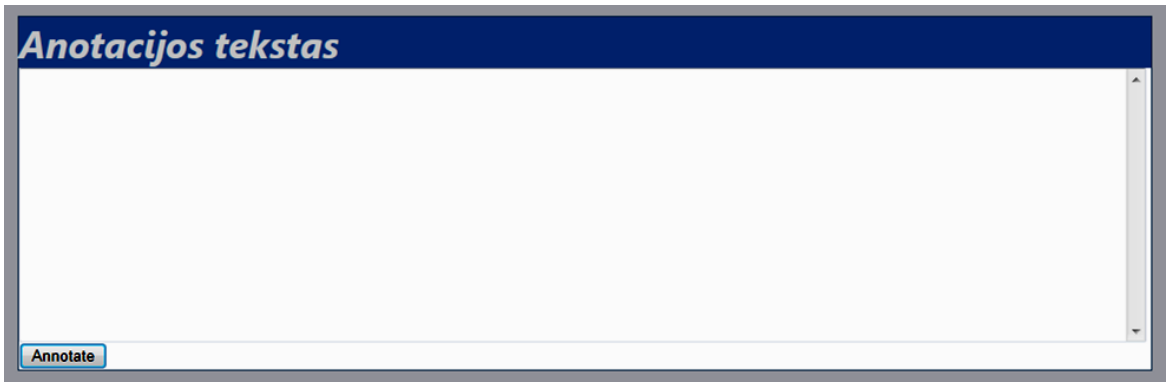
11 paveikslėlyje pavaizduota viso semantinio anotavimo proceso veiklos diagrama nuo norimo anotuoti teksto importavimo į programą iki semantiškai anotuoto teksto.



11 pav. veiklos diagrama

Vartotojo interfeiso modelis

12 paveikslėlyje pateiktas galimas vartotojo sąsajos modelis. Šiame paveikslėlyje pavaizduotas sąsajos modelis yra pilnai veikiantis.



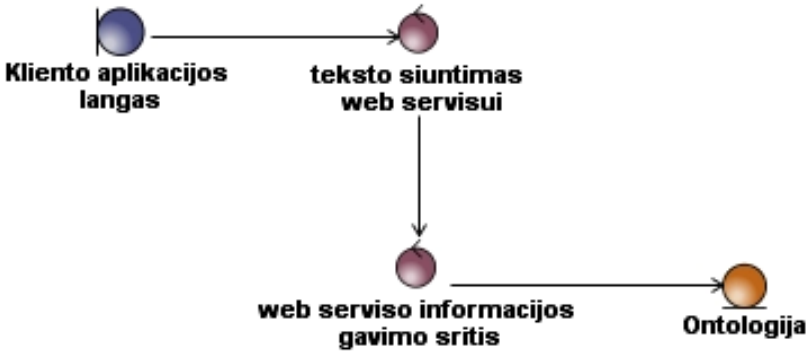
12 pav. galimas vartotojo interfeiso modelis

Robustiškumo diagramos

Robustiškumo diagramos parodo, kokias ribines, valdymo ir esybių klases reikia realizuoti ir kaip jos turi būti realizuotos norint išgauti atitinkančią reikalavimus programą.

Teksto nusiuntimo Web servisui robustiškumo diagrama

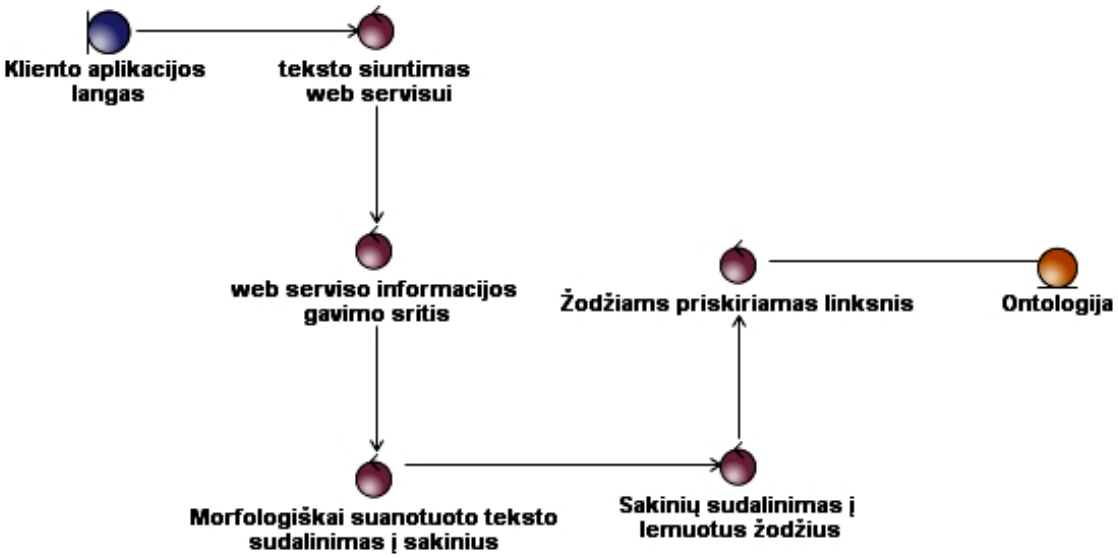
13 paveikslėlyje pavaizduota teksto nusiuntimo Web servisui robustiškumo diagrama.



13 pav. teksto nusiuntimo web servisui robustiškumo diagrama

Teksto suskaidymo robustiškumo diagrama

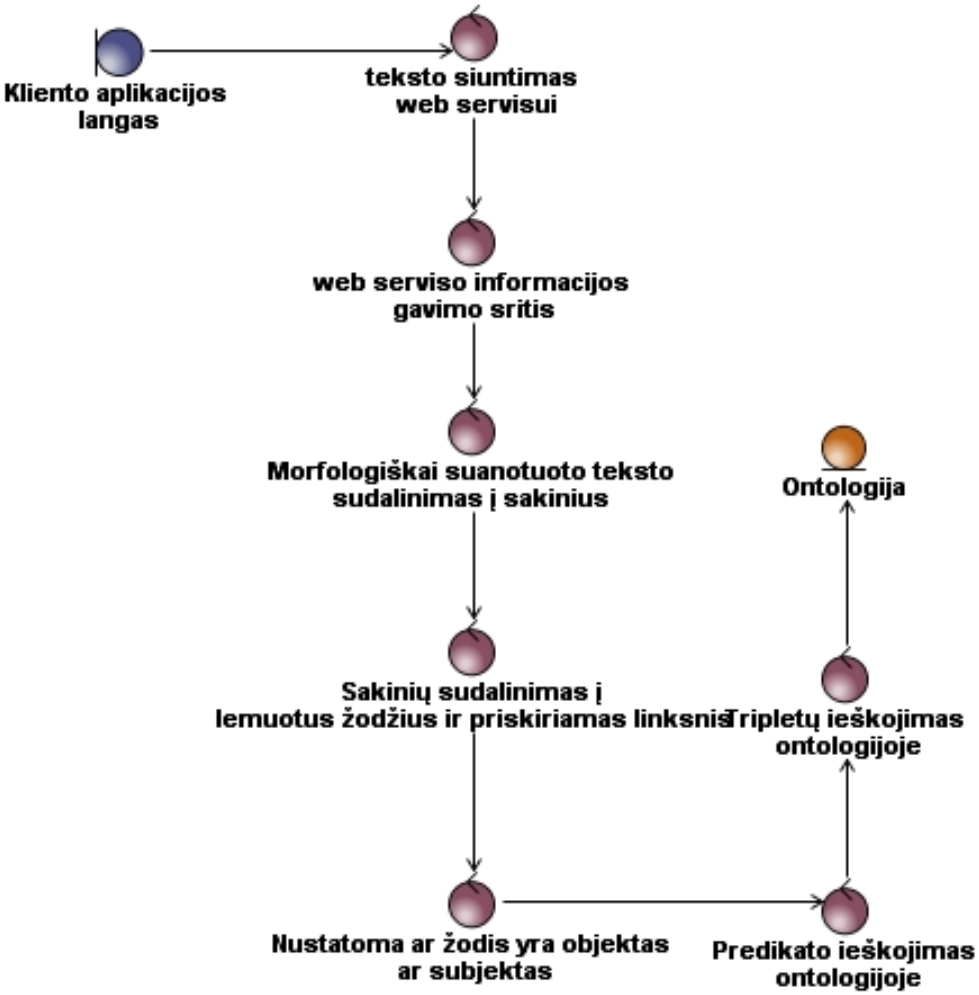
14 paveikslėlyje pavaizduota teksto suskaidymo robustiškumo diagrama.



14 pav. Kalbos taisyklių robustiškumo diagrama

Žodžių paieškos ontologijoje robustiškumo diagrama

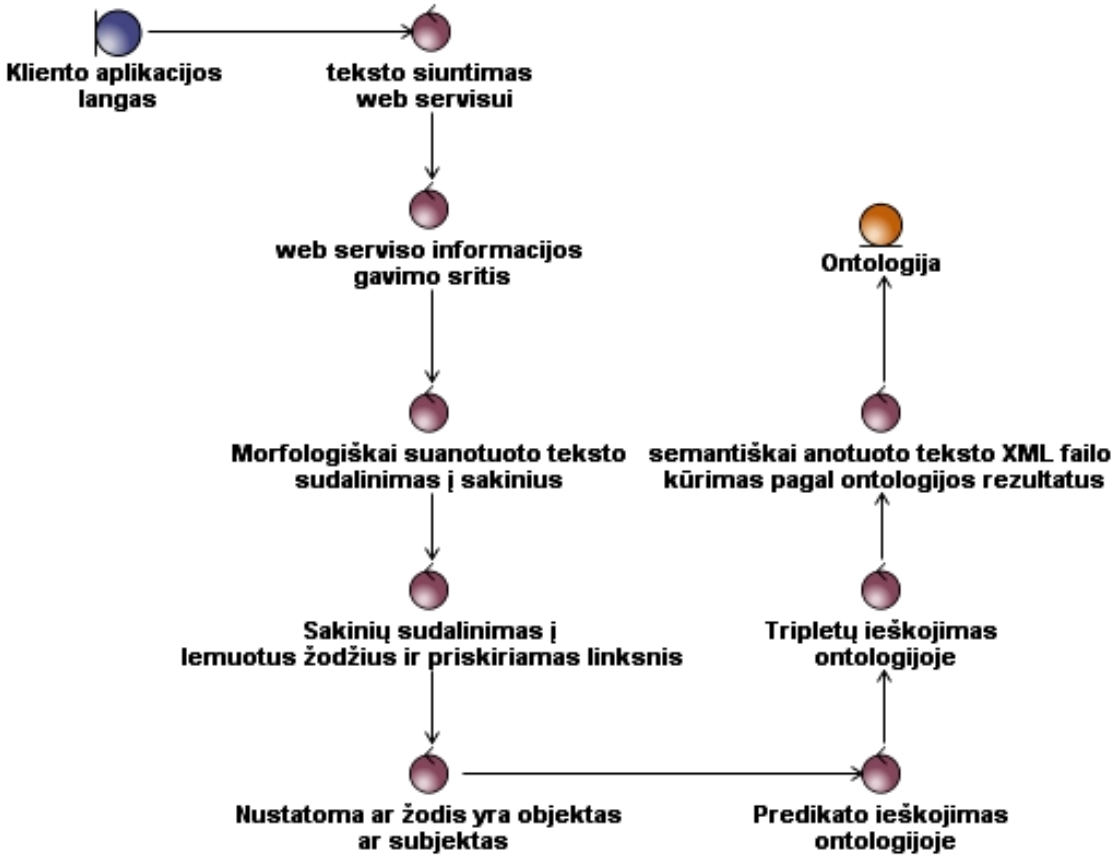
15 paveikslėlyje pavaizduota žodžių paieškos ontologijoje robustiškumo diagrama.



15 pav. Žodžių paieškos ontologijoje robustiškumo diagrama

Semantiškai suanotuoto teksto XML failo kūrimo robustiškumo diagrama

16 paveikslėlyje pavaizduota semantiškai suanotuoto teksto XML failo kūrimo robustiškumo diagrama.

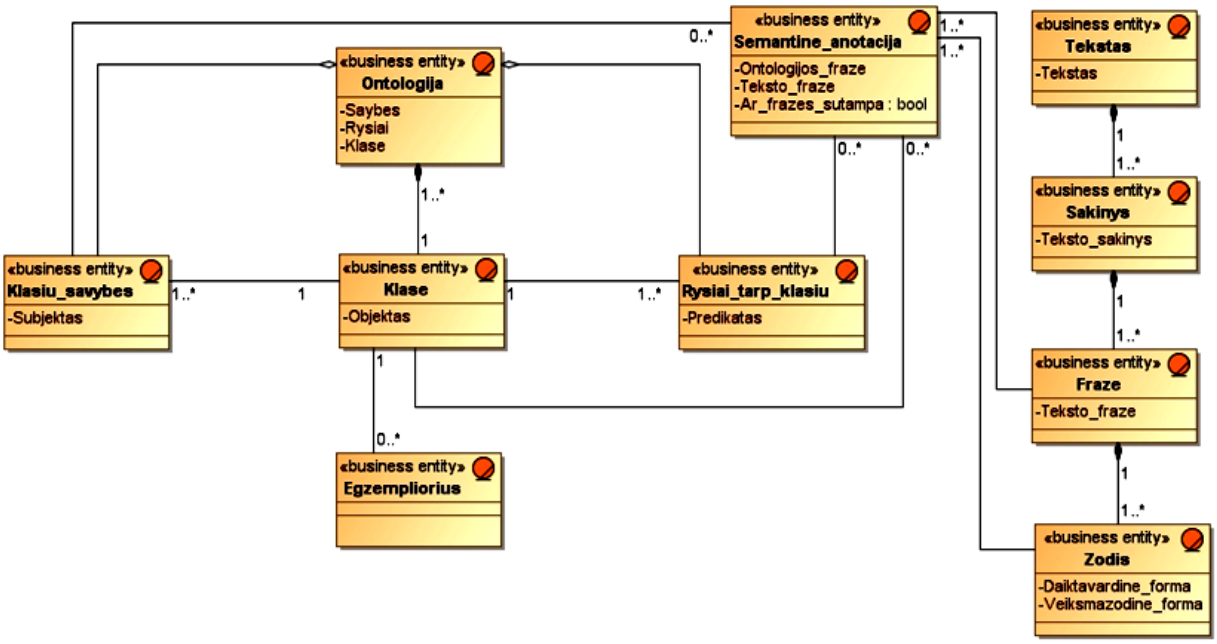


16 pav. Semantiškai suanotuoto teksto XML failo kūrimo robustiškumo diagrama

2.2. Dalykinės srities modelis

UML esybių klasių diagrama

17 paveikslėlyje pavaizduota esybių klasių diagrama. Ji parodo kaip siejasi ontologija, semantinės anotacija ir norimas anotuoti tekstas. Iš esybių klasių diagramos galime matyti, kad semantinis anotavimo procesas analizuoja tekstą išskirstydamas jį į frazes ir sakinius. Žinodamas kiekvieno sakinio objektą, subjektą, predikatą ir taip išvengdamas dviprasmybių, frazę lygina su ontologijos klasėmis. Jei frazė atitinka ontologijos taisykles semantinio anotavimo procesas nagrinėjamai frazei priskiria su ta fraze susijusius ontologijos metaduomenis.



17 pav. UML esybių klasių diagrama

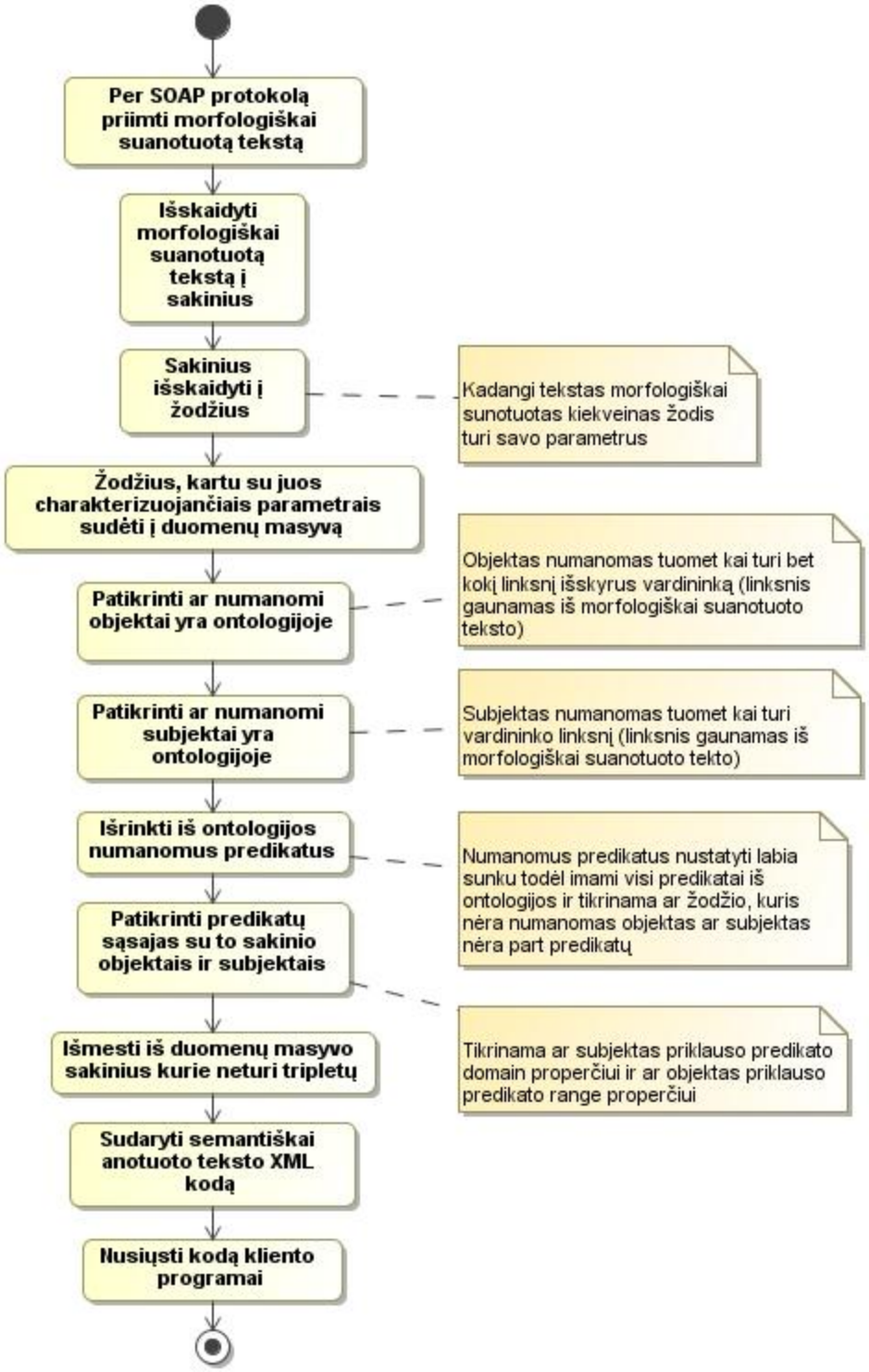
2.3. Reikalavimų analizės apibendrinimas:

1. Atlikus sistemos reikalavimų analizę pavyko aiškiau išsiaiškinti norimo anotuoti tekto, ontologijos ir semantinio anotavimo proceso sąsają.
2. Aiškiau suprasta kokius semantinio anotavimo procesus reikės modernizuoti.

2.4. Sistemos architektūros projektas

Sistemos loginė architektūra

18 paveikslėlyje pavaizduota sistemos loginė architektūra. Sistemos loginėje architektūroje atvaizduotos sistemos posistemes ir jų tarpusavio sąryšį. Joje aiškiai galima matyti web serviso elgseną gavus morfologiškai suanotuotą tekstą.



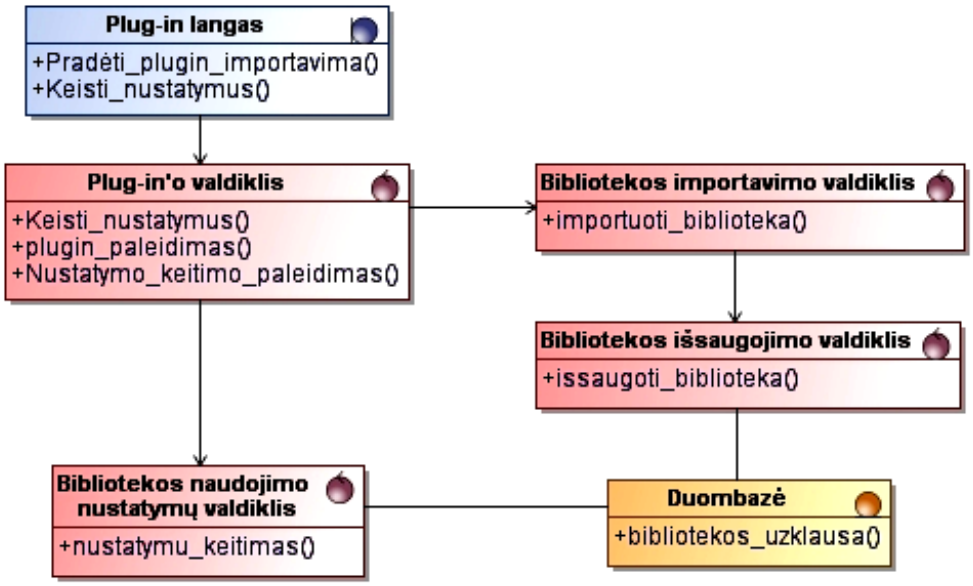
18 pav. Sistemos loginė architektūra

Bendras klasių modelis

Bendrame klasių modelis parodo kokios operacijos vykdomos ribinėse, valdymo ir esybių klasėse.

Bendras kalbos klasių modelis

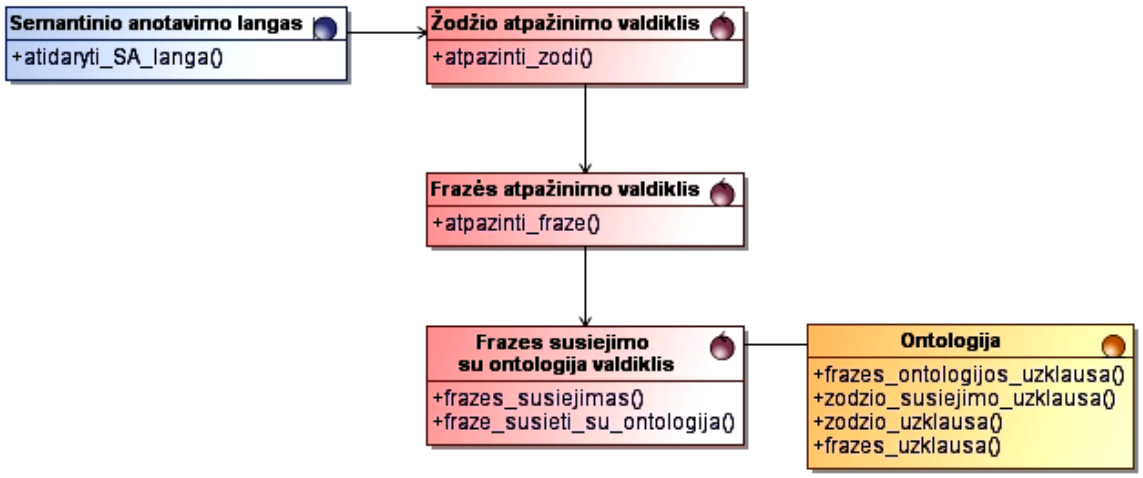
19 paveikslėlyje pavaizduotas bendras kalbos klasių modelis.



19 pav. Bendras kalbos klasių modelis

Bendras frazių išrinkimo klasių modelis

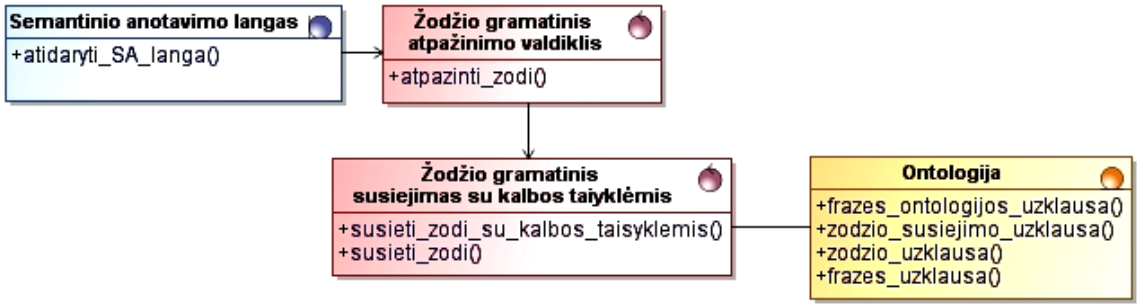
20 paveikslėlyje pavaizduotas bendras frazių išrinkimo klasių modelis.



20 pav. Bendras frazių išrinkimo klasių modelis

Bendras POS (part-of-speech) klasių modelis

21 paveikslėlyje pavaizduotas bendras POS (part-of-speech) klasių modelis.



22 pav. Bendras POS klasių modelis

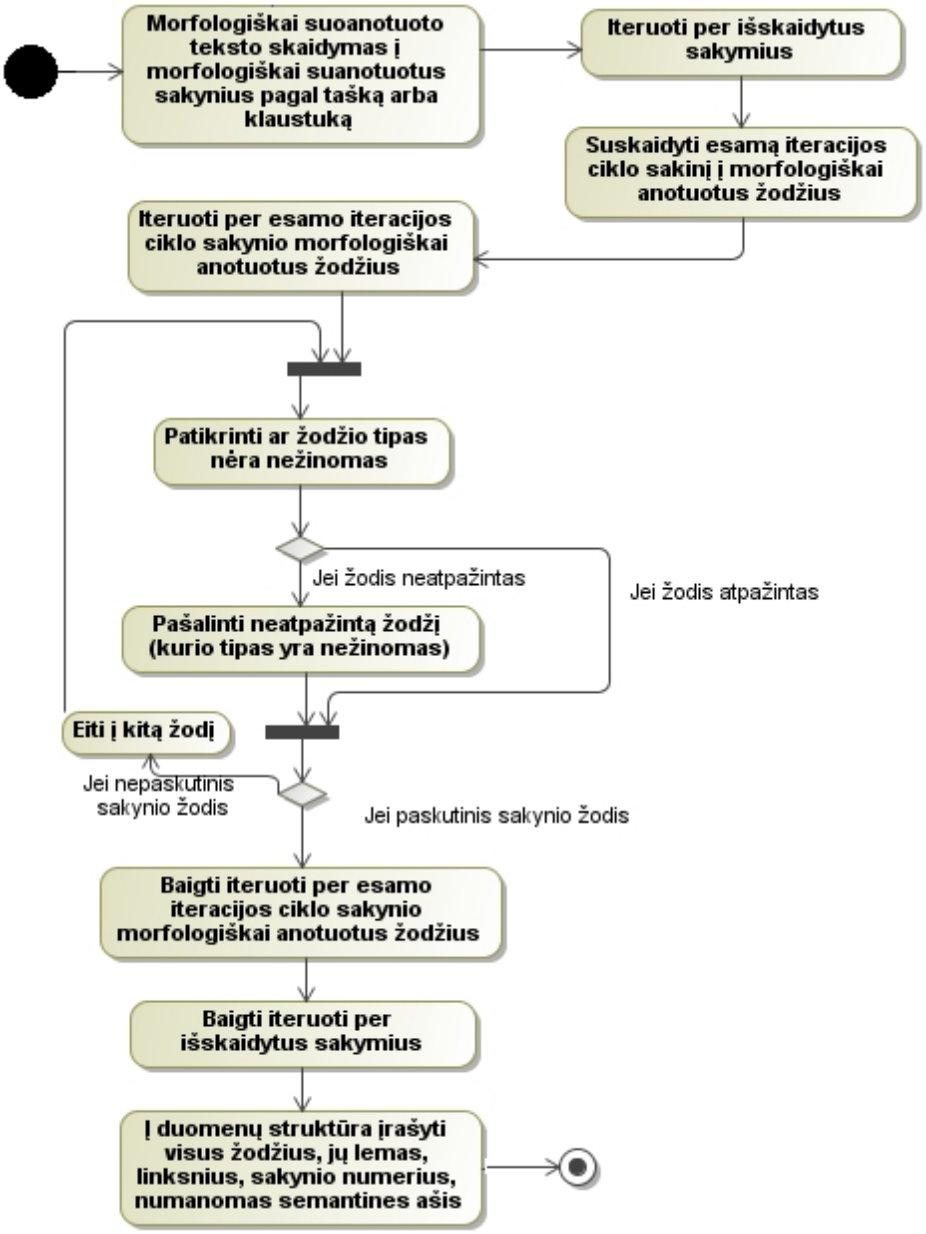
2.5. Sistemos metodologija

2.5 skyriuje pateikiamos veiklos diagramos kurios atspindi kokia metodologija remiasi sistemos veikimo principas. Diagramos pateiktos išskaidytos į dalis nuo metodologijos veikimo pradžios iki pabaigos ir yra sistemos logines architektūros (18 pav.) detalesnis vaizdas. Prie kiekvienos diagramos pateikiamas prierašas kokią sistemos loginės architektūros diagramoje esančią klases(-ę) jos atspindi.

Teksto skaidymo veiklos diagrama

23 Paveikslėlyje pavaizduota teksto skaidymo veiklos diagrama parodo metodu morfologiškai suanotuotas tekstas išskaidomas į sakinius, sakiniai į žodžius ir sudedami į

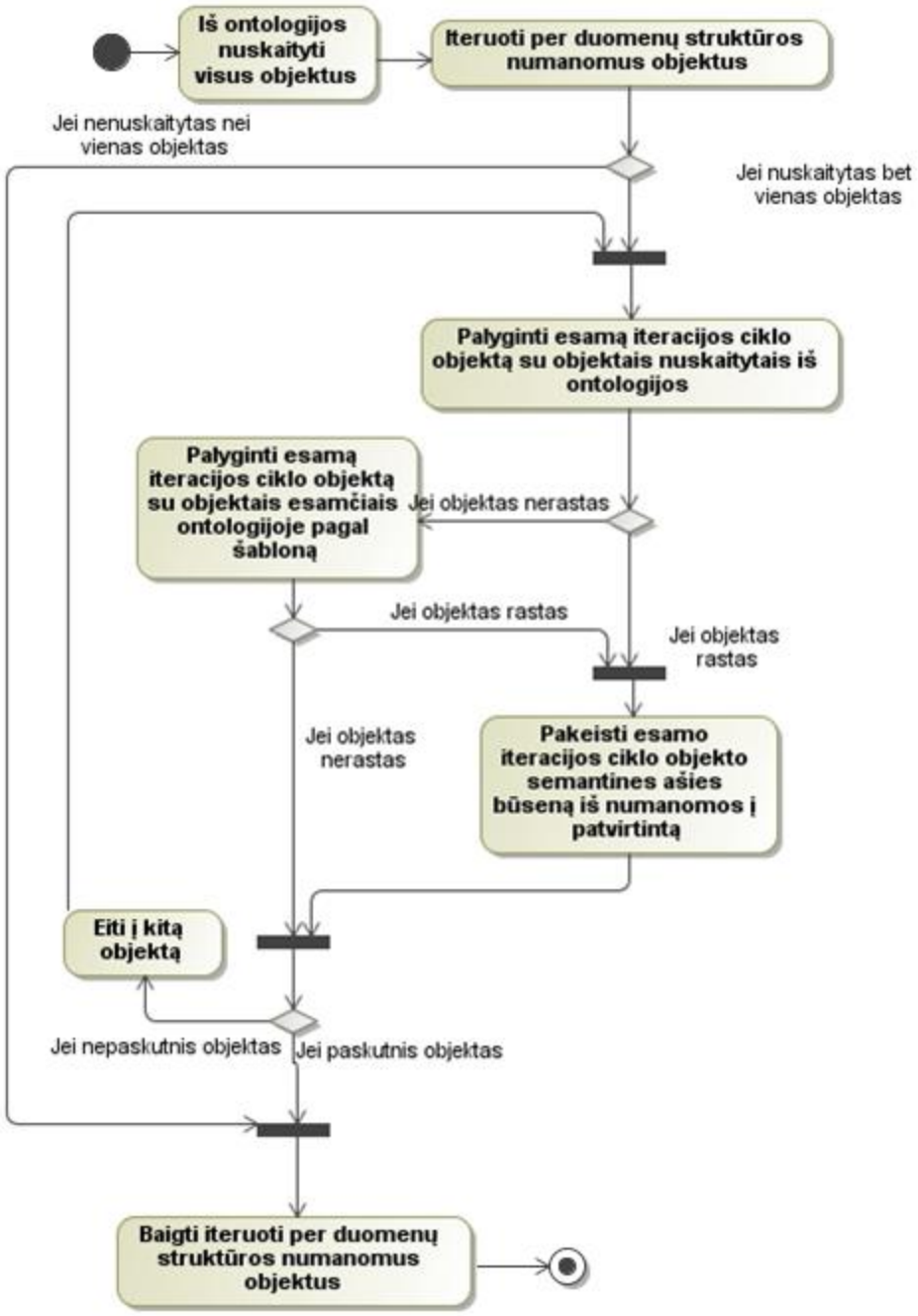
duomenų struktūrą. Duomenų struktūroje taip pat saugoma sakinio semantine ašis, linksnis, lemma. Visos šie žodį charakterizuojantys parametrai gaunami iš morfologiškai suanotuot teksto išskyrus sakinio numerį. Būtent dėl šios priežasties iš pradžių morfologiškai suanotuotas tekstas yra skaidomas į sakinius ir žinant sakinio numerį - į žodžius. Ši diagrama apima šias sistemos loginės architektūros diagramoje esančias klases: "*Išskaidyti morfologiškai suanotuotą tekstą į sakinius*", "*Sakinius išskaidyti į žodžius*" ir "*Žodžius, kartu su juos charakterizuojančiais parametrais sudėti į duomenų masyvą*".



23 pav. Teksto skaidymo veiklos diagrama

Objektų aptikimo ontologijoje veiklos diagrama

24 Paveikslėlyje pavaizduota objektų aptikimo ontologijoje veiklos diagrama parodo kaip morfologiškai anotuotame tekste atpažįstami objektai. Morfologiškai suanotuoto teksto išsaugojimo duomenų struktūroje metu tam tikri žodžiai atpažįstami kaip numanomi objektai. Tai padaroma patikrinant linksnį. Jei linksnis yra bet koks kitas, išskyrus vardininką, žodžio semantine ašis yra pažymima kaip numanomas objektas. Tuomet nuskaitomi visi ontologijos objektai ir žodis palyginamas su žodžiais nuskaitytais iš ontologijos. Tai padaroma dvejais būdais, žodžius lyginant tiesiogiai ir pagal šabloną. Iš diagramos matyti jog objektu gali būti pažymėta ir žodžių grupė. Jei bent vienu atveju žodžiai sutampa tuomet semantiškai anotuoto žodžio semantinė ašis yra pažymima kaip objektas. Ši diagrama apima "*Patikrinti ar numanomi objektai yra ontologijoje*" sistemos loginės architektūros diagramoje esančią klasę.

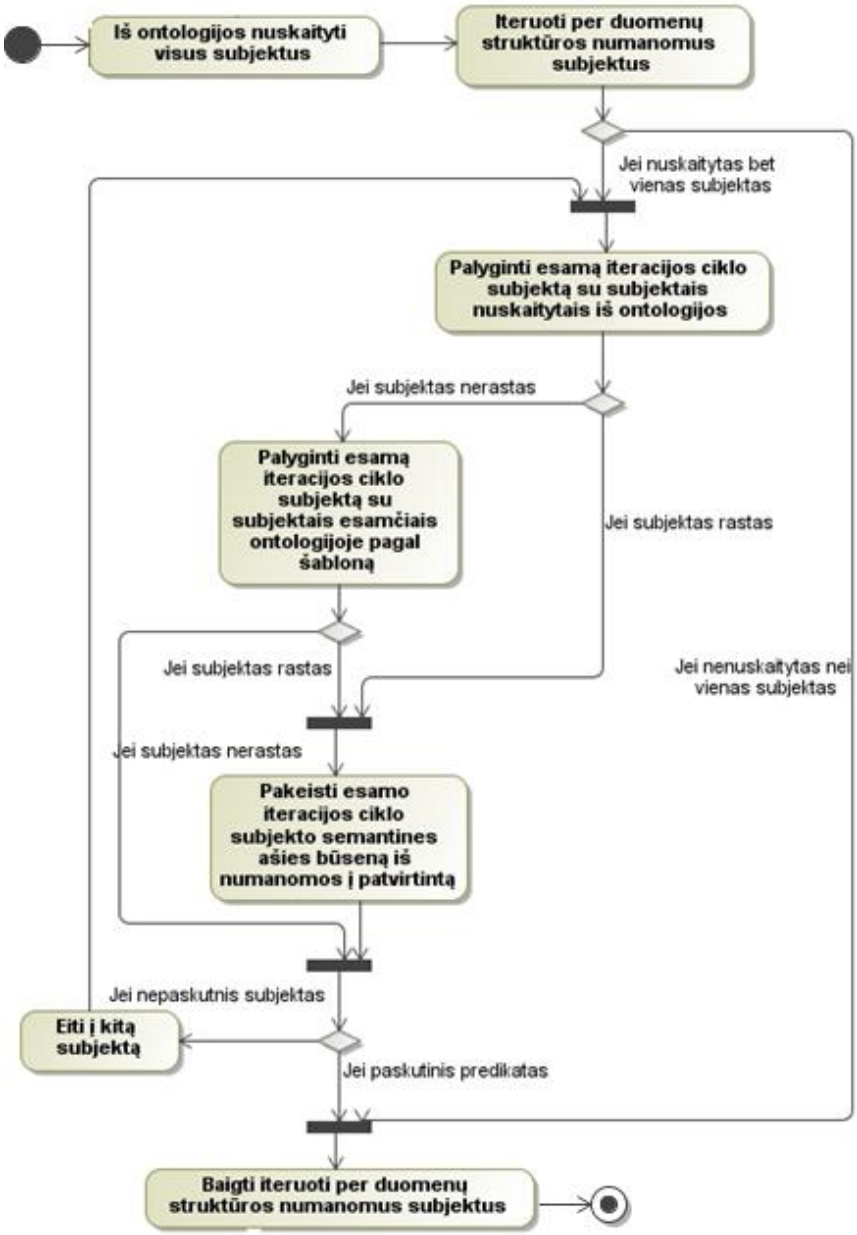


24 pav. Objektų aptikimo ontologijoje veiklos diagrama

Subjektų aptikimo ontologijoje veiklos diagrama

25 Paveikslėlyje pavaizduota subjektų aptikimo ontologijoje veiklos diagrama parodo kaip morfologiškai anotuotame tekste atpažįstami subjektai. Morfologiškai suanotuoto teksto išsaugojimo duomenų struktūroje metu tam tikri žodžiai atpažįstami

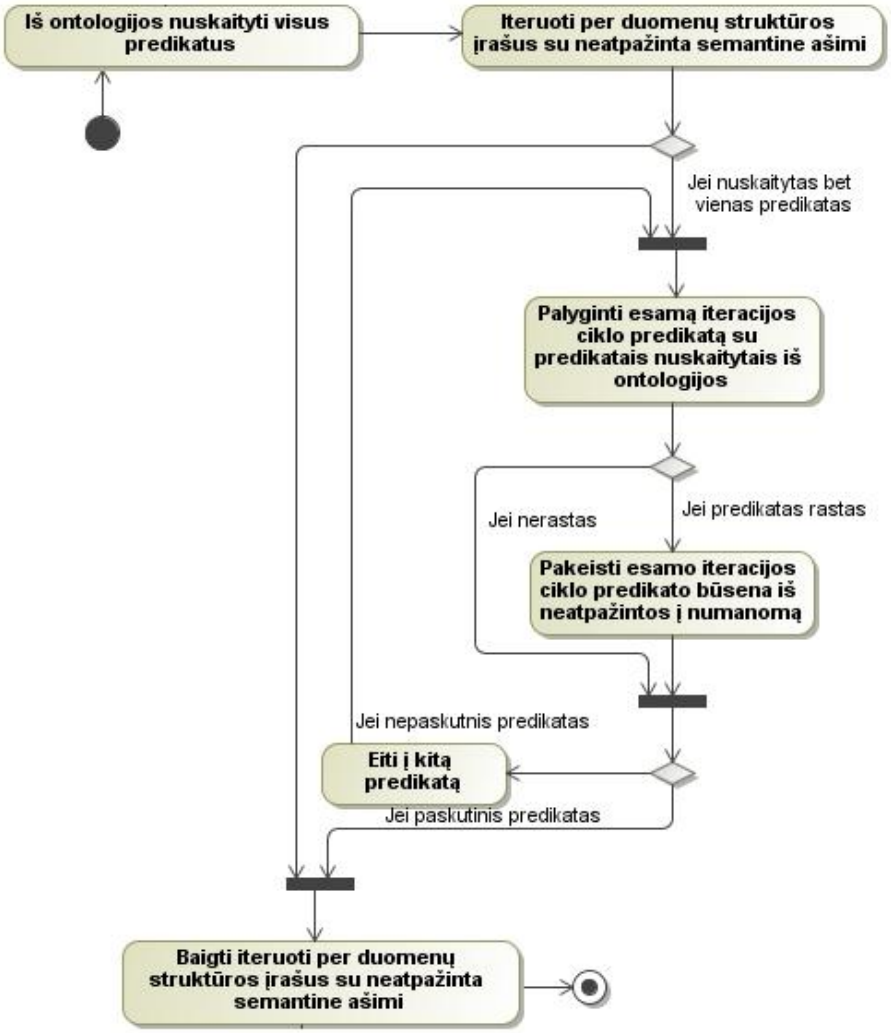
kaip numanomi subjektai. Tai padaroma patikrinant linksnį. Jei linksnis yra vardininkas žodžio semantine ašis yra pažymima kaip numanomas subjektas. Tuomet nuskaitomi visi ontologijos subjektai ir žodis palyginamas su žodžiais nuskaitytais iš ontologijos. Tai padaroma dvejais būdais, žodžius lyginant tiesiogiai ir pagal šabloną. Jei bent vienu atveju žodžiai sutampa tuomet semantiškai anotuoto žodžio semantinė ašis yra pažymima kaip subjektas. Iš diagramos matyti jog subjektu gali būti pažymėta ir žodžių grupė. 19 paveikslėlyje esanti diagrama apima "*Patikrinti ar numanomi subjektai yra ontologijoje*" sistemos loginės architektūros diagramoje esančią klasę.



25 pav. Subjektų aptikimo ontologijoje veiklos diagrama

Numanomų predikatų aptikimo ontologijoje veiklos diagrama

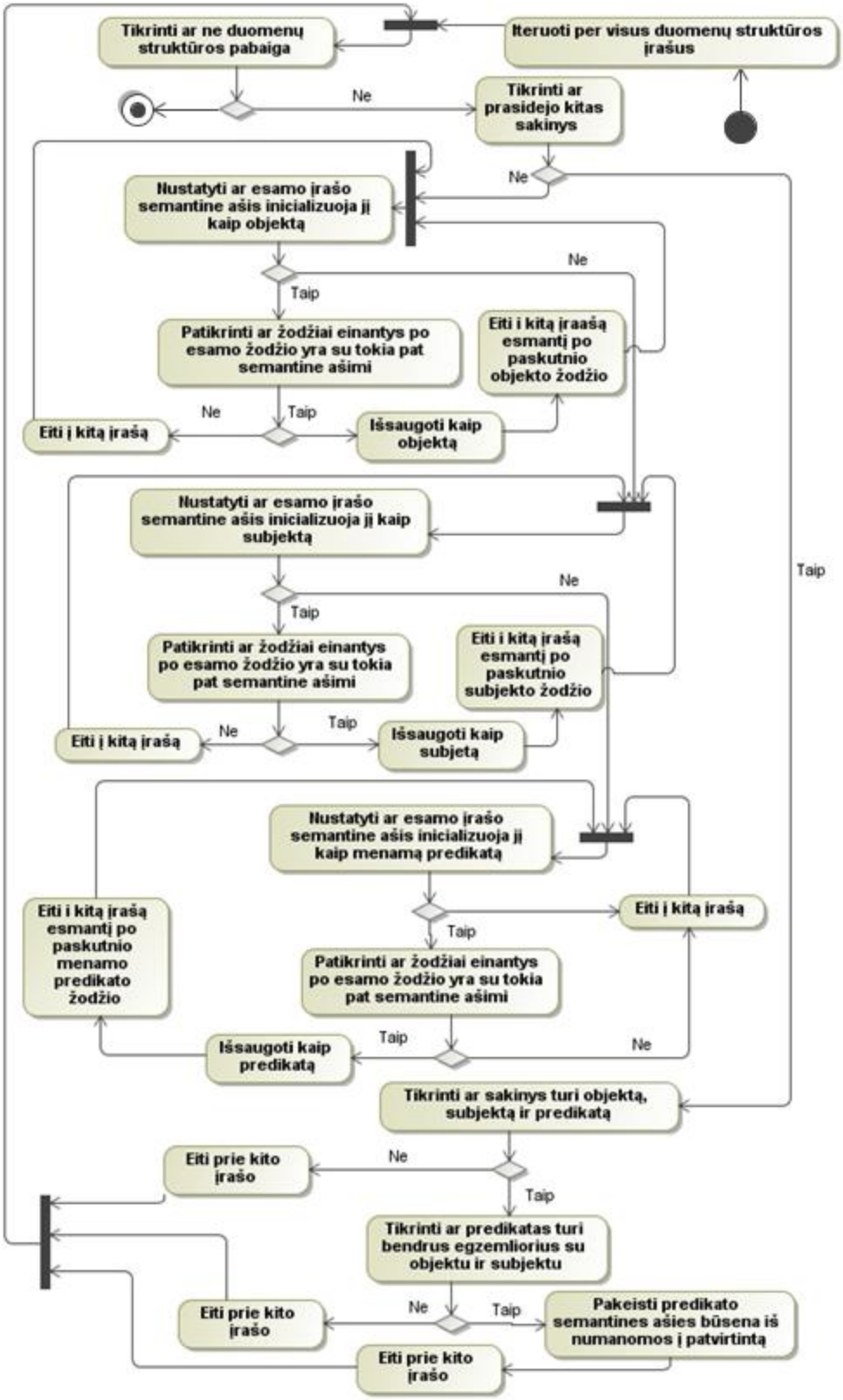
26 Paveikslėlyje pavaizduota numanomų predikatų aptikimo ontologijoje veiklos diagrama parodo kaip morfologiškai anotuotame tekste atpažįstami numanomi predikatai. Iš ontologijos nuskaitomi visi predikatai. Tuomet kiekvienas žodis, kurio semantinė ašis yra nežinoma, lyginamas su predikatais ontologijoje. Kai žodis atitinka ontologijos predikatą tuomet duomenų struktūros žodis pažymimas kaip numanomas predikatas. Iš diagramos matyti jog numanomu predikatu gali būti pažymėta ir žodžių grupė. 20 paveikslėlyje esanti diagrama apima "*Išrinkti iš ontologijos numanomus predikatus*" sistemos loginės architektūros diagramoje esančią klasę.



26 pav. Predikatų aptikimo ontologijoje veiklos diagrama

Tripleto aptikimo veiklos diagrama

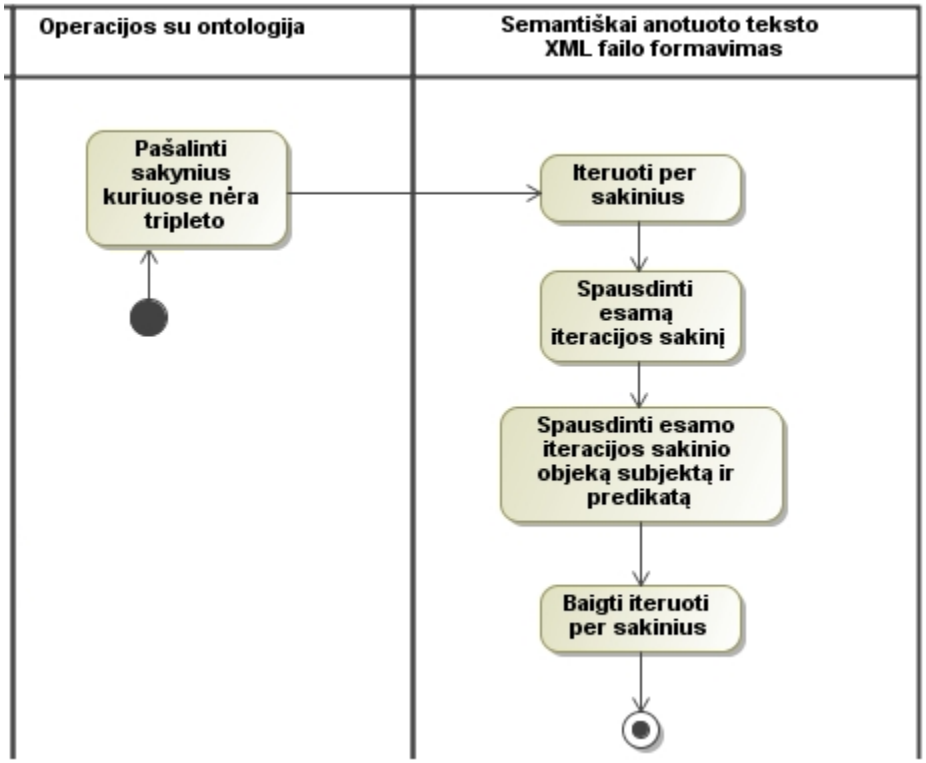
27 Paveikslėlyje pavaizduotas diagrama, kuri atspindi metodą, nusakantį kokiu būdu aptinkami tripletai. Tipletas sudaromas tik tuomet kai predikatas turi bendrų egzempliorių su objektu ir subjektu. Jeigu bendrų egzempliorių nėra tuomet reiškia, kad to sakinio objektas, subjektas ir predikatas nesudaro tripleto. Tripleto aptikimas sakinyje vyksta taip: einama per sakinio žodžius ir tikrinama ar to žodžio semantinė ašis nėra pažymėta kaip subjektas objektas ar numanomas predikatas. Jei yra - žodis išsaugojamas ir tikrinama ar tolimesnių žodžių semantinė ašis nėra tokia pati kaip surastojo žodžio. Jei semantinė ašis tokia pati tuomet išsaugojama visa žodžių grupė. Tokiu būdu patikrinus viso sakinio žodžius surandami subjektas, objektas ir numanomas predikatas. Tuomet tikrinama ar predikatas turi bendru egzempliorių objekto ir subjekto atžvilgiu. Jei bendri egzemplioriai egzistuoja sudaromas tripletas. Taip patikrinami visi sakiniai. 21 paveikslėlyje esanti diagrama apima "*Patikrinti predikatų sąsajas su to sakinio objektais ir subjektais*" sistemos loginės architektūros diagramoje esančią klasę.



27 pav. Tripleto aptikimo veiklos diagrama

Nereikalingų sakinių pašalinimo ir semantiškai anotuoto teksto formavimo diagrama

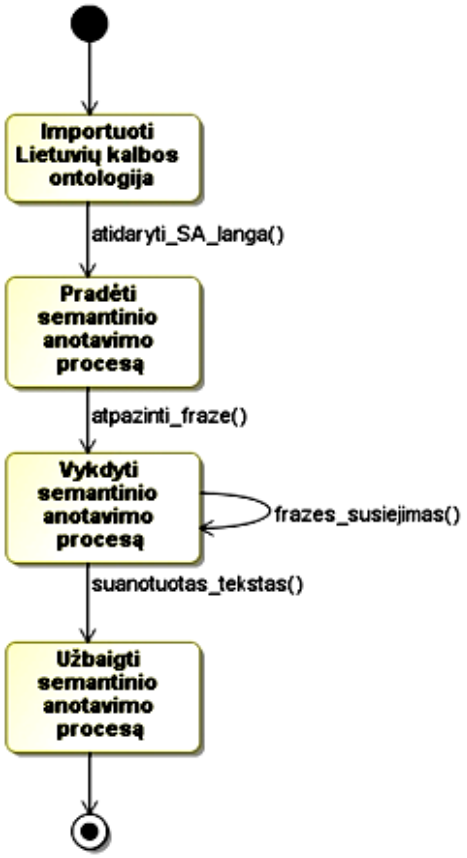
28 paveikslėlyje pateikiama diagrama parodanti kaip vyksta semantiškai anotuoto teksto išsaugojimas XML failo pavidalu. Diagrama apima šias sistemos loginės architektūros diagramoje esančias klases: "*Išmesti iš duomenų masyvo sakinius kurie neturi tripletų*" ir " *Sudaryti semantiškai anotuoto teksto XML kodą*".



28 pav. Nereikalingų sakinių pašalinimo ir semantiškai anotuoto teksto formavimo diagrama

2.6. Detalus projektas

29 paveikslėlyje pavaizduota objektų būsenų kitimų diagrama.



29 pav. Objektų būsenų kitimų diagrama

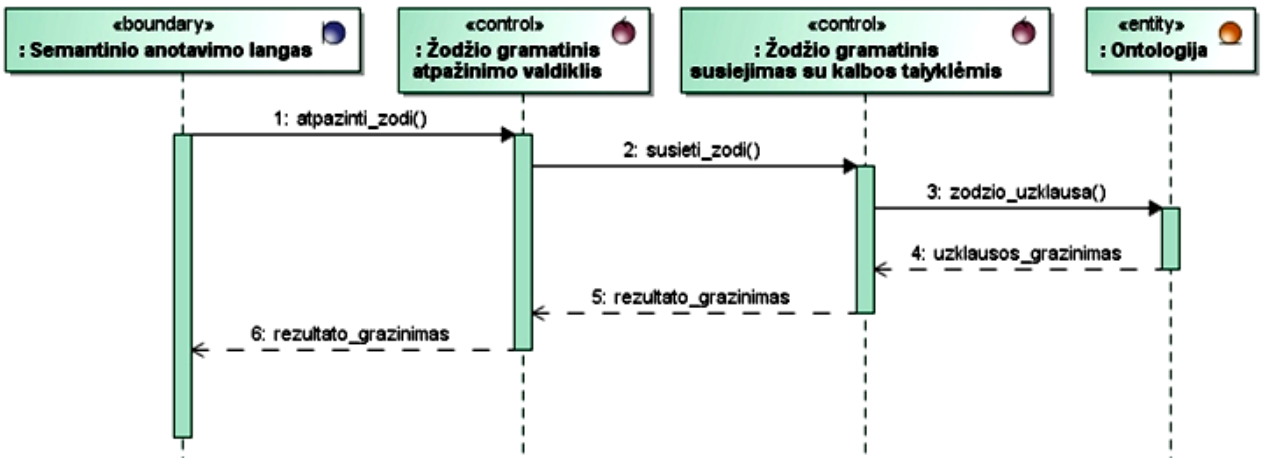
2.7. Sistemos elgsenos modelis

Sekų modeliai

Sudarytos pagrindinių panaudojimo atvejų sekos diagramos, kurios rodo bendravimą tarp klasių, realizuojančių panaudojimo atvejus.

POS (part-of-speech) sekų modelis

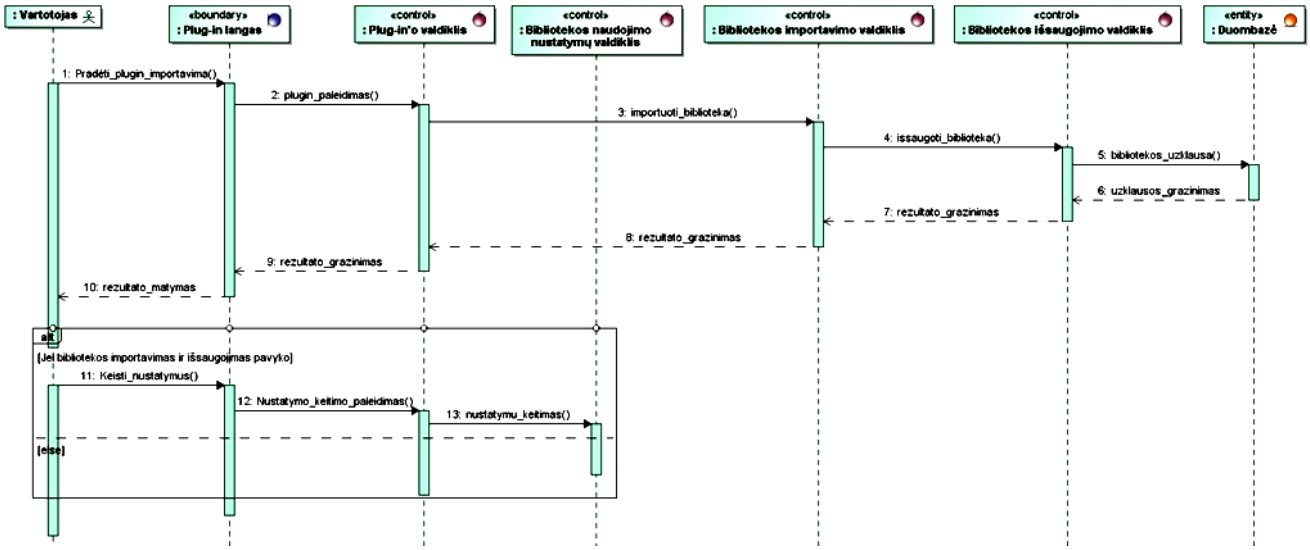
30 paveikslėlyje pavaizduotas POS (part-of-speech) sekų modelis.



30 pav. POS sekų modelis

Kalbos taisyklių sekų modelis

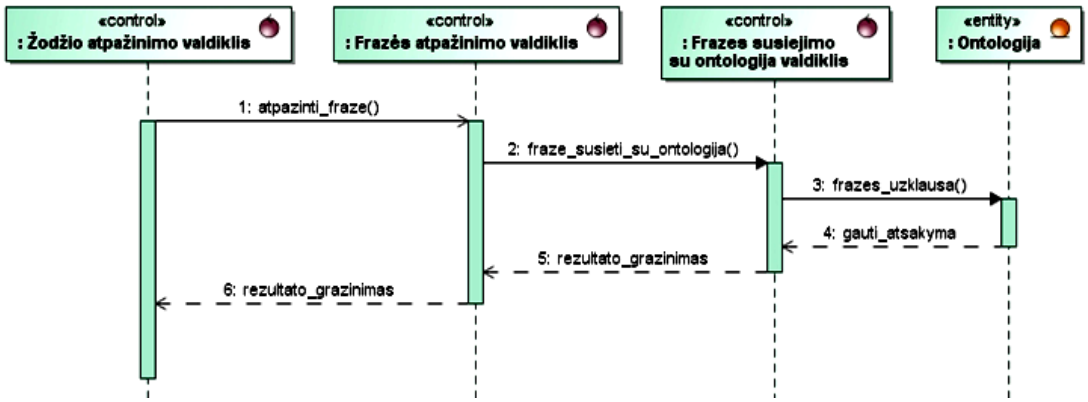
31 paveikslėlyje pavaizduotas kalbos taisyklių sekų modelis.



31 pav. Kalbos taisyklių sekų modelis

Frazių išrinkimo sekų modelis

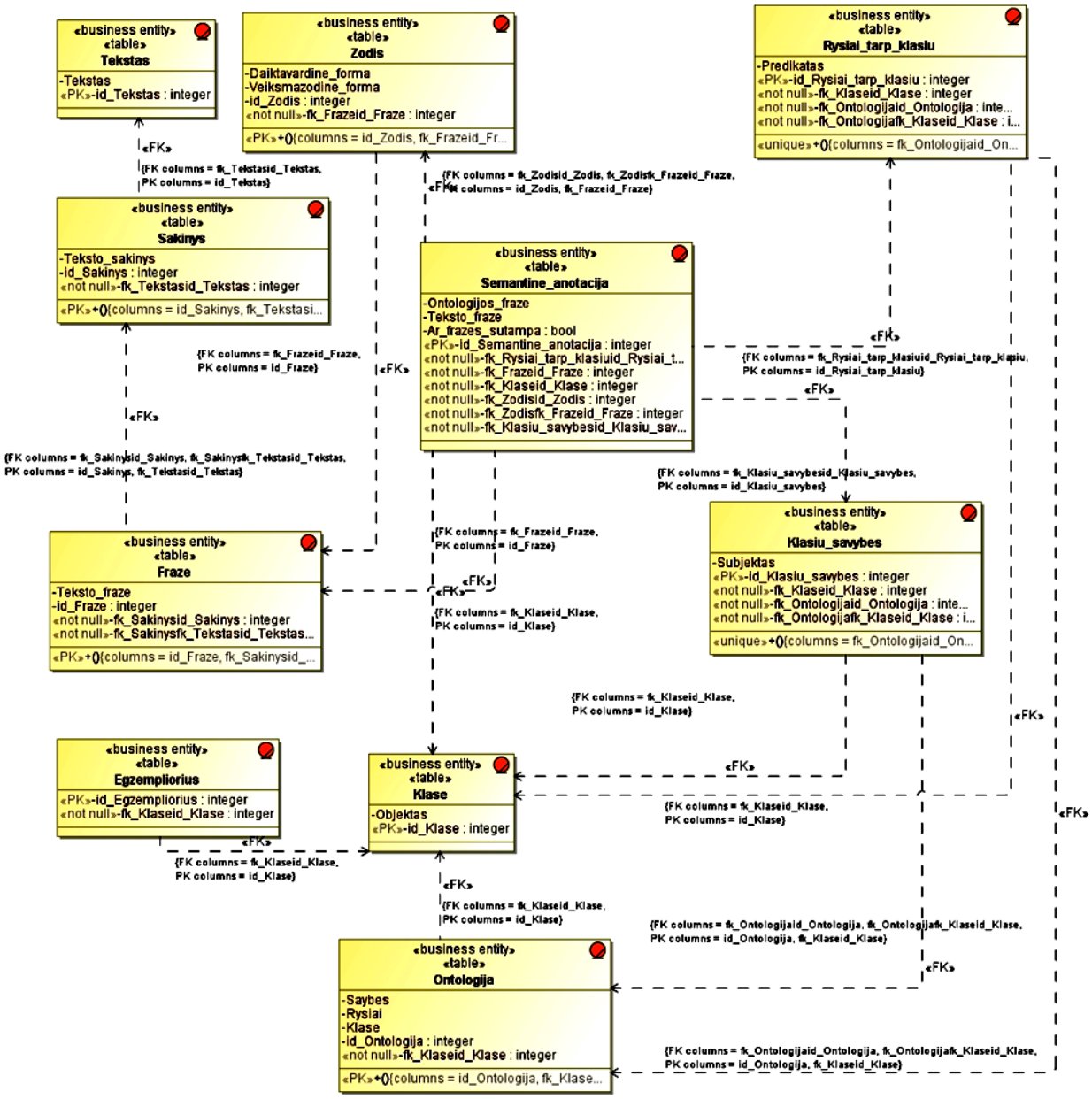
32 paveikslėlyje pavaizduotas frazių išrinkimo sekų modelis.



32 pav. Frazių išrinkimo sekų modelis

2.8. Duomenų bazės schema

33 paveikslėlyje pateikta sistemos duomenų bazės schema. Šioje schemoje galima aiškiai matyti semantinio anotavimo, ontologijos ir frazių sąryšį.



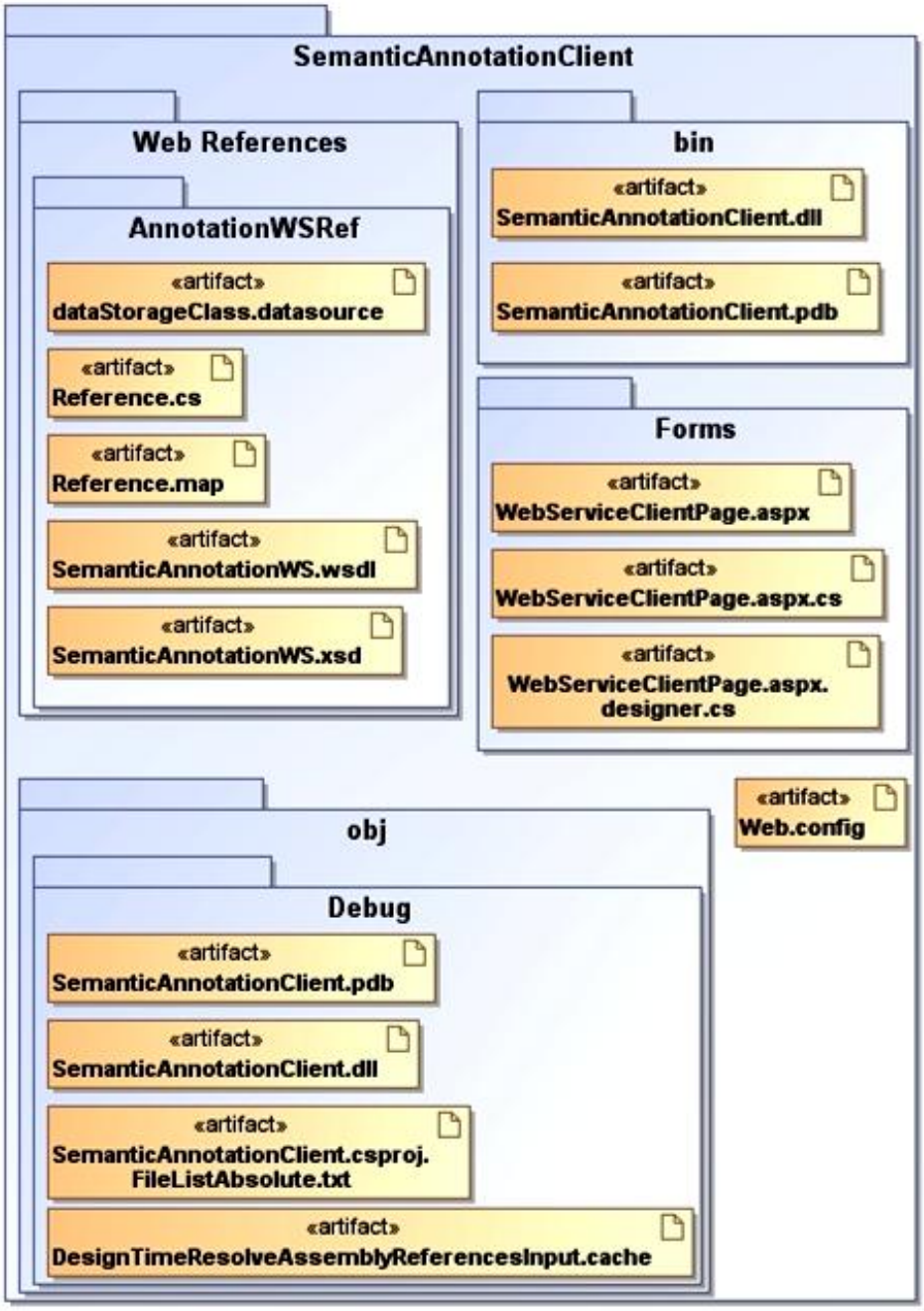
33 pav. Sistemos duomenų bazės schema

2.9. Realizacijos modelis

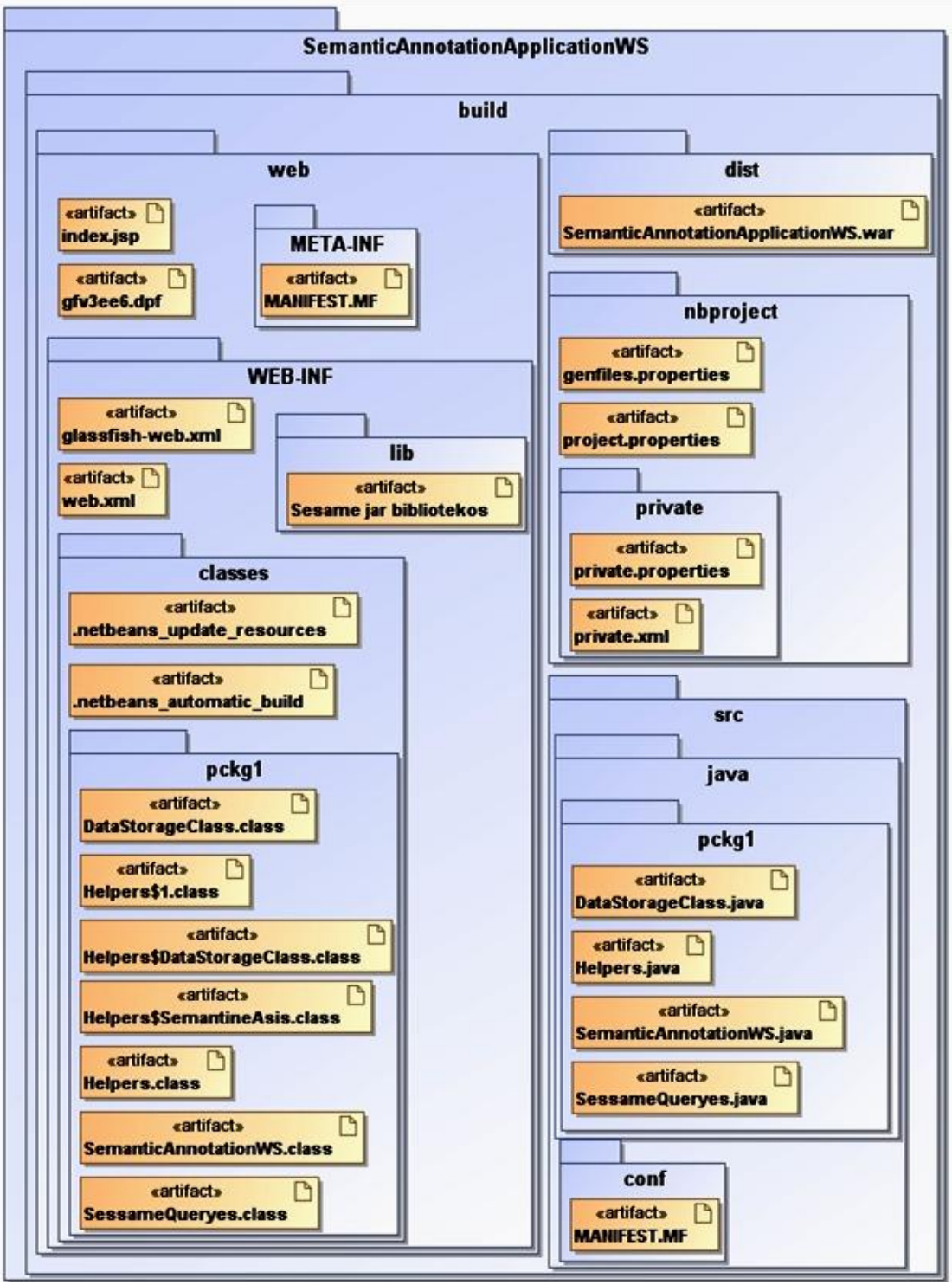
34 paveikslėlyje pavaizduota komponentų realizavimo artefaktais diagrama.

Šioje diagramoje pavaizduota kliento programa, kuri SOAP protokolu komunikuoja su Web servisu. Norėdami, jog kliento programa galėtų komunikuoti su web servisu reikia, kad programa turėtų Web nuorodą (angl. web referense). Kliento programos

komponentų diagramoje pavaizduota kokie failai reikalingi web nuorodai kartu su failais priklausančiais programai.



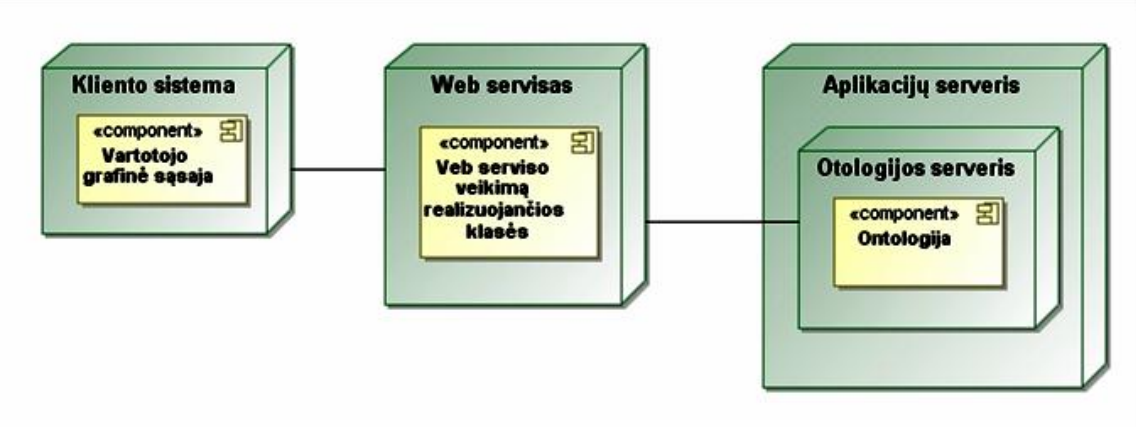
34 pav. Kliento programos komponentų realizavimas artefaktais



35 pav. Semantinio anotavimo Web serviso komponentų realizavimas artefaktais

3. SPRENDIMO REALIZACIJA IR TESTAVIMAS

Semantinio anotavimo sistema realizuota web serviso pagrindu. Web servisas savo ruoštu url pagalba komunikuoja su ontologija, kuri yra patalpinta "Sesame" serveryje. "Sesame" serveris patalpintas aplikacijų serveryje (tai gali būti arba "jboss" arba "tomcat" serveris). 36 pav. Pavaizduota diagrama atspindint



36 pav. Sistemos sudedamųjų dalių diagrama

3.1. Web serviso realizavimas

Web serviso veikimas pagrįstas metodologija kuri remiasi trimis etapais:

- Morfologiškai suanotuoto teksto išskaidymu ir sudėjimu į duomenų struktūras
- Teksto semantinis anotavimas
- Semantiškai anotuoto teksto XML formavimas

Taip pat web servisas turi komunikavimą su kitomis programomis. Jis vykdomas SOAP protokolu.

3.1.1. Morfologiškai teksto išskaidymu ir sudėjimu į duomenų struktūras

Morfologiškai suanotuotas tekstas iš pradžių skaidomas į sakinius. Žinant sakinio numerį tekstas išskaidomas į žodžius išsaugant jo sakinio numerį. Morfologiškai suanotuotame žodyje pašalinamos nereikalingos teksto vietos. Tuomet morfologiškai suanotuoti žodžiai, jų lemos, linksniai sudedami į

duomenų masyvus. Pastarieji parametrai gaunami tiesiogiai iš morfologiškai suanotuoto teksto, tačiau yra išskaičiuojami parametrai, tokie kaip žodžio semantine ašis ir sakinio numeris. Morfologiškai suanotuoto žodžio pavyzdys:

<word="magistras" lemma="magistras" type="dkt., vyr. g., vns., V."/>

Programos kodas realizuojantis morfologiškai suanotuoto žodžio parametrų išrinkimą ir sudėjimą į duomenų struktūra. Komentarai aprašo kodo veikimą.

```
public DataStorageClass getWords(String lemmaString, int index){
    DataStorageClass dsc = new DataStorageClass();
    String[] linksniai = new String[] {"V.", "K.", "N.", "G.",
    "Įn.", "Vt."};
    String[] sentence = lemmaString.split("\"");
    //lemma masyve eina antras
    String[] temp = sentence[1].split("\\(");
    //tikrina ar žodis turi bendratį
    if(temp.length > 1){
        dsc.setLemma(temp[0]);
    }
    else{
        dsc.setLemma(sentence[1]);
    }
    dsc.setSakinioNr(index);
    //nustato ar subjektas ar objektas
    for(int i = 0; i < linksniai.length; i++){
        if(sentence[3].contains(linksniai[i])){
            dsc.setLinksnis(linksniai[i]);
            if(linksniai[i].matches("V.")){
                dsc.setSemantineAsis('s');
            }
            else{
                dsc.setSemantineAsis('o');
            }
        }
    }
    if(dsc.getSemantineAsis() == '\0')
    {
            //jei nei subjektas nei objektas tuomet n = nežinomas
            dsc.setSemantineAsis('n');
    }
    return dsc;
}
```

Atlikus operacijas su morfologiškai anotuoto teksto išskaidymu pereinama prie pagrindinio etapo - teksto semantinio anotavimo.

3.1.2. Teksto semantinis anotavimas

Teksto semantinio anotavimo etape pateikti tik verčiausi dėmesio semantinio anotavimo sprendimai.

Teksto semantinis anotavimas prasideda nuo objektų ir subjektų suradimo. Pagal linksnį nustatomi galimi objektai ir subjektai. Tuomet galimi objektai ir subjektai lyginami su iš ontologijos nuskaitytais objektais ir subjektais. Jei tarp ontologijos objektų ir subjektų reikiamų objekto ir subjekto nėra tuomet morfologiškai anotuoto teksto objektai ir subjektai žymimi kaip neatpažinti

Programos kodas realizuojantis subjektų suradimą ontologijoje. Komentarai aprašo kodo veikimą.

```
public boolean getSubjects(){
//užklausa, nuskaitanti subjektus ir jų label'ius
        String queryString = "SELECT * WHERE { ?name rdf:type owl:Class;
rdfs:label ?label.}";
        if(subjects == null){
            subjects = new ArrayList<String>();
        }
        if(getResults(queryString, subjects)){
            return true;
        }
        else{
            return false;
        }
    }

private boolean getResults(String queryString, List<String> list){
        try {
            RepositoryConnection con = myRepository.getConnection();
            try {
                TupleQuery tupleQuery = con.prepareTupleQuery
                (QueryLanguage.SPARQL, queryString);
                TupleQueryResult result = tupleQuery.evaluate();
                try {
                   List<String> bindingNames = result.getBindingNames();
                    while (result.hasNext()){
                        BindingSet bindingSet = result.next();
                        Value firstValue = bindingSet.getValue
                        (bindingNames.get(0));
                        String allValue = firstValue.stringValue();
                        //iš nuskaitytos duomenų struktūros atfiltruoja
                        //label'į
                        String label = allValue.split("#")[1];
                        list.add(label);
                    }
                }
                finally {
```

```
                    result.close();
                    if(!list.isEmpty()){
                        return true;
                    }
                    else{ return false; }
                }
            }
            finally {
                //con.close();
            }
        }
        catch (OpenRDFException e) {
        // handle exception
        }
        return false;
    }
```

Programos kodas realizuojantis subjektų semantines ašies pakeitimą ontologijos subjektų atžvilgiu. Komentarai aprašo kodo veikimą.

```
private List<DataStorageClass> FindSubjects(List<DataStorageClass> data)
    {
        //patikrina ar iš ontologijos nuskaitytas bent vienas subjektas
        if(sQueryes.getSubjects()){
            for(int i = 0; i < data.size(); i++){
                DataStorageClass dStorage = data.get(i);
            //patikrina ar semantine ašis nuskao subjektą
                if(dStorage.getSemantineAsis() == 's'){

if(!sQueryes.checkIfSubjectExist(dStorage.getLemma())){
    //jei subjektas ontologijoje nerastas tipas   pakeičiamas į nežinomą
                    dStorage.setSemantineAsis('n');
                }
                }
            }
        }
        return data;
    }
```

Technologiškai pats sudėtingiausias dalykas sukurtas web servise - tripletų atpažinimo algoritmas. Algoritmas atpažįsta ar sakinyje yra objektas, subjektas ir predikatas. Tuomet pagal predikatą patikrina ar objektas, subjektas ir predikato "domain" ir "range" atributai turi sąsajas su objektų ir subjektų egzemplioriais. Jei turi - objektas, subjektas ir predikatas sudaro tripletą.

Programos kodas realizuojantis tripletų suradimą ontologijoje. Komentarai aprašo kodo veikimą.

```
    private List<DataStorageClass> FindTripleteList
(List<DataStorageClass> data){
```

```
        int predicateId = 0;
        int nextSakynioNr = 0;
        String object = "";
        String subject = "";
        String predicate = "";
        for(int i = 0; i < data.size(); i++){
            DataStorageClass dsThis = data.get(i);
            DataStorageClass dsNext = null;
            if(i+1 < data.size()){
                dsNext = data.get(i+1);
                nextSakynioNr = dsNext.getSakinioNr();
            }
            if(dsNext == null){
                nextSakynioNr = dsThis.getSakinioNr();
            }
            if(dsThis.getSakinioNr() == nextSakynioNr){
                char c = dsThis.getSemantineAsis();
                if(c == 'o') {
                    object = dsThis.getLemma();
                }
                else if(c == 's'){
                    subject = dsThis.getLemma();
                }
                else if(c == 'm'){
                    predicate = dsThis.getLemma();
                    predicateId = i;
                }
            }
            else{
                //jei sakiniu nr nesutampa reiskias esamas sakinys baiges
                if(!object.isEmpty() && !subject.isEmpty() &&
    !predicate.isEmpty()){
                    //sQueryes.getTriplete(object, subject, predicate);
                    if(sQueryes.checkIfTripleteExist(object, subject,
    predicate)){
                        DataStorageClass dsTriplete = data.get(predicateId);
                        dsTriplete.setSemantineAsis('p');
                        data.set(predicateId, dsTriplete);
                    }
                }
            }
            //jei paskutinis zmodis masyve reiskia daugaiu sakiniu nera
             //todėl patikrinam paskutini sakini
            if(i == data.size()-1){
                if(!object.isEmpty() && !subject.isEmpty() &&
    !predicate.isEmpty()){
                    //sQueryes.getTriplete(object, subject, predicate);
                    if(sQueryes.checkIfTripleteExist(object, subject,
    predicate)){
                        DataStorageClass dsTriplete = data.get(predicateId);
                        dsTriplete.setSemantineAsis('p');
                        data.set(predicateId, dsTriplete);
                    }
                }
            }
        }
        return data;
    }
```

3.1.3. Semantiškai anotuoto teksto XML formavimas

Semantiškai anotuoto teksto XML formavimas susideda iš dviejų dalių: semantiškai anotuoto teksto formavimo algoritmo, kuris duomenis sudeda į duomenų struktūras skirtas jų spausdinimui ir spausdinimo algoritmo, kuris įrašo duomenis į string tipo objektą su visais reikiamais XML atributais. Semantiškai anotuoto teksto formavimo algoritmas yra panašus į tripletų atpažinimo algoritmą. Spausdinimo algoritmas atspausdina sakinius, su objektais, subjektais ir predikatais.

Spausdinimo algoritmo suformuoto XML teksto pavyzdys.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
      <corpus id="sample">
        <body>
         <s id="sent_0">
           <terminals>
                <t id=" a_0000 " word=atletas/>
                <t id=" a_1000 " word=dalyvauti/>
                <t id=" a_2000 " word=varžybos/>
           </terminals>
           <nonterminals>
                <edge label="obj" idref=" a_2000 " />
                <edge label="sub" idref=" a_0000 " />
                <edge label="prd" idref=" a_1000 " />
           </nonterminals>
         </s>
        </body>
</corpus>
```

3.2. Ontologijos realizavimas

Web servisas naudojo jau realizuotą ontologiją. Tačiau kai kurias jos vietas reikėjo modifikuoti norint gauti tinkama rezultatą. Viena iš vietų - pridėti <rdfs:label> atributą. Šis atributas pridedamas tam, kad Web servisas žodžių nelygintų pagal <rdf:about> atributą, kuris nėra tikslus, nes nusako žodžius atvaizduoja be tarpų ir be lietuviškų raidžių, kas savo ruožtu reikalautų papildomo žodžių atpažinimo variklio.

```
        <owl:Class rdf:about="&spo;BarjerinisBegimas">
            <rdfs:subClassOf rdf:resource="&spo;Begimas"/>
        <rdfs:label>barjerinis bėgimas</rdfs:label>
        </owl:Class>
```

4. EKSPERIMENTINIS SPRENDIMO TYRIMAS

Eksperimentas atliekamas suanotuojant tekstą rankiniu būdu, tuomet atsižvelgti kiek iš rankiniu būdu suanotuotų žodžių yra ontologijoje ir palyginti su rezultatais, kurie gauti suanotavus programa. Reikia atsižvelgti į tai, jog programa semantiškai anotuoja sakinį tik tuomet jei sakinys turi objektą, subjektas ir predikatą kuriuos galima surasti ontologijoje ir jeigu predikatas turi bendrų egzempliorių su objektu ir subjektu.

Eksperimentui naudojamas tekstas:

"Rusijos futbolo aukščiausioje arba „Premjer“ lygoje pirmadienį įvyko vienerios ir paskutinės priešpaskutinio 29-o turo rungtynės. Machačkalos „Anži“ klubas savo aikštėje susitiko su Maskvos „Lokomotiv“ komanda ir nugalėjo varžovus rezultatu 2:1. Ši pergalė Degestano ekipai likus vienam turui iki čempionato pabaigos, leido užsitikrinti bronzos medalius. „Anži“ klubas Rusijos čempionato medalius iškovojo pirmą kartą istorijoje. Virgilijus Alekna dalyvavo 2006 metų Europos lengvosios atletikos čempionate."

Tekstas suanotuotas rankiniu būdu:

Rusijos futbolo aukščiausioje arba „Premjer“ lygoje pirmadienį įvyko vienerios ir paskutinės priešpaskutinio 29-o turo rungtynės.

Subjektas: 29-o turo rungtynės

Objektas: Rusijos futbolo aukščiausioje arba „Premjer“ lygoje

Predikato pavadinimas: įvyko

Machačkalos „Anži“ klubas savo aikštėje susitiko su Maskvos „Lokomotiv“ komanda ir nugalėjo varžovus rezultatu 2:1.

Subjektas: Machačkalos „Anži“ klubas

Objektas: Maskvos „Lokomotiv“ komanda

Predikato pavadinimas: susitiko, nugalėjo

Ši pergalė Degestano ekipai likus vienam turui iki čempionato pabaigos, leido užsitikrinti bronzos medalius.

Subjektas: pergalė

Objektas: bronzos medalius

Predikato pavadinimas: leido užsitikrinti

„Anži“ klubas Rusijos čempionato medalius iškovojo pirmą kartą istorijoje.

Subjektas: „Anži“ klubas

Objektas: Rusijos čempionato medalius

Predikato pavadinimas: iškovojo

Virgilijus Alekna dalyvavo 2006 metų Europos lengvosios atletikos čempionate.

Subjektas: Virgilijus Alekna

Objektas: 2006 metų Europos lengvosios atletikos čempionate

Predikato pavadinimas: dalyvavo

Programos semantiškai suanotuotas tekstas:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <corpus id="sample">
    <body>
     <s id="sent_0">
       <terminals>
            <t id=" a_0000 " word= Rusijos />
            <t id=" a_1000 " word= futbolo />
            <t id=" a_2000 " word= aukščiausioje />
            <t id=" a_3000 " word= arba />
            <t id=" a_4000 " word= Premjer />
            <t id=" a_5000 " word= lygoje />
            <t id=" a_6000 " word= pirmadienį />
            <t id=" a_7000 " word= įvyko />
            <t id=" a_8000 " word= vienerios />
            <t id=" a_9000 " word= ir />
            <t id=" a_10000 " word= 29/>
            <t id=" a_11000 " word= o/>
            <t id=" a_12000 " word= turo />
            <t id=" a_13000 " word= rungtynės />
            </terminals>
       <nonterminals>
            <edge label="obj" idref=" a_0000 " />
            <edge label="obj" idref=" a_1000 " />
            <edge label="obj" idref=" a_2000 " />
            <edge label="sub" idref=" a_13000 " />
            <edge label="prd" idref=" a_7000 " />
       </nonterminals>
     </s>
    </body>
```

```
  <body>
   <s id="sent_1">
      <terminals>
            <t id=" a_0000 " word= Anži />
            <t id=" a_1000 " word= klubas />
            <t id=" a_2000 " word= Rusijos />
            <t id=" a_3000 " word= čempionato />
            <t id=" a_4000 " word= medaliais />
            <t id=" a_5000 " word= iškovojo />
            <t id=" a_6000 " word= pirmą />
            <t id=" a_7000 " word= kartą />
            <t id=" a_8000 " word= istorijoje />
      </terminals>
      <nonterminals>
            <edge label="obj" idref=" a_2000 " />
            <edge label="obj" idref=" a_3000 " />
            <edge label="obj" idref=" a_4000 " />
            <edge label="sub" idref=" a_0000 " />
            <edge label="sub" idref=" a_1000 " />
            <edge label="prd" idref=" a_5000 " />
      </nonterminals>
   </s>
  </body>
  <body>
   <s id="sent_1">
      <terminals>
            <t id=" a_0000 " word= Virgilijus />
            <t id=" a_1000 " word= Alekna />
            <t id=" a_2000 " word= dalyvavo />
            <t id=" a_3000 " word= 2006 />
            <t id=" a_4000 " word= metų />
            <t id=" a_5000 " word= Europos />
            <t id=" a_6000 " word= lengvosios />
            <t id=" a_7000 " word= atletikos />
            <t id=" a_8000 " word= čempionate />
      </terminals>
      <nonterminals>
            <edge label="obj" idref=" a_4000 " />
            <edge label="obj" idref=" a_5000 " />
            <edge label="obj" idref=" a_6000 " />
            <edge label="obj" idref=" a_7000 " />
            <edge label="obj" idref=" a_8000 " />
            <edge label="sub" idref=" a_0000 " />
            <edge label="sub" idref=" a_1000 " />
            <edge label="prd" idref=" a_2000 " />
      </nonterminals>
   </s>
  </body>
</corpus>
```

Programa suanotavo tris sakinius iš penkių. Ontologijoje reikiamos informacijos galėjo pateikti keturiems sakiniams. Nesuanotuotas buvo antrasis sakinys *"Machačkalos „Anži“ klubas savo aikštėje susitiko su Maskvos „Lokomotiv“ komanda ir nugalėjo varžovus rezultatu 2:1"*. Šiuo atveju programa suanotavo 75% galimų atvejų. Realus programos pajėgumas viršija 50%, norint padidinti suanotuotų sakinių reikėtų programos tipletų paieškos algoritmą modernizuoti taip, kad šis sugebėtų atpažinti sudėtinius sakinius.

Rankinio anotavimo ir kompiuterinio anotavimo vizualus palyginimas

Rankiniu būdu semantiškai suanotuotas tekstas:

Rusijos futbolo aukščiausioje arba „Premjer“ lygoje pirmadienį įvyko vienerios ir paskutinės priešpaskutinio 29-o turo rungtynės. Machačkalos „Anži“ klubas savo aikštėje susitiko su Maskvos „Lokomotiv“ komanda ir nugalėjo varžovus rezultatu 2:1. Ši pergalė Degestano ekipai likus vienam turui iki čempionato pabaigos, leido užsitikrinti bronzos medalius. „Anži“ klubas Rusijos čempionato medaliais pasipuošė pirmą kartą istorijoje. Virgilijus Alekna dalyvavo 2006 metų Europos lengvosios atletikos čempionate.

Programos semantiškai suanotuotas tekstas:

Rusijos futbolo aukščiausioje arba „Premjer“ lygoje pirmadienį įvyko vienerios ir paskutinės priešpaskutinio 29-o turo rungtynės. Machačkalos „Anži“ klubas savo aikštėje susitiko su Maskvos „Lokomotiv“ komanda ir nugalėjo varžovus rezultatu 2:1. Ši pergalė Degestano ekipai likus vienam turui iki čempionato pabaigos, leido užsitikrinti bronzos medalius. „Anži“ klubas Rusijos čempionato medaliais pasipuošė pirmą kartą istorijoje. Virgilijus Alekna dalyvavo 2006 metų Europos lengvosios atletikos čempionate.

Reikėtų atkreipti dėmesį, jog ontologija netūrėjo pakankamai informacijos, jog galėtų leisti suanotuoti šį sakinį: *"Ši pergalė Degestano ekipai likus vienam turui iki čempionato pabaigos, leido užsitikrinti bronzos medalius."*

5. REZULTATŲ APIBENDRINIMAS IR IŠVADOS

Apibendrinimas

Išbandžius pagal metodologiją sukurtą semantinio anotavimo programą buvo nustatyta, jog sunkiausia anotuoti sudurtinius sakinius kurių subjektuose ir objektuose yra tikrinių daiktavardžių, įvairių pavadinimų ir kurie turi keletą predikatų, jeigu žodžiai priklausantys subjektui ar objektu eina ne vienas paskui kitą. Jei subjektai ir objektai

susideda kelių žodžių, kurie neina vienas paskui kita, labai sunku nustatyti ar šie žodžiai priklauso vienam subjektui ar tai yra atskiri subjektai. Tas pats galioja predikatams.

Išvados

Atliekant automatizuoto semantinio anotavimo proceso metodologijos kūrimą ir realizavimą JAVA programavimo kalba buvo išsiaiškinta, jog semantinio anotavimo tikslumas labiau priklauso nuo tripletų paieškos algoritmo, mažiau - nuo objektų ir subjektų atpažinimo algoritmo.

Kuo tikslesnė ontologija, tuo geresni semantinio anotavimo rezultatai, nekeičiant semantinio anotavimo algoritmo.

Kuo paprastesnė sakinio konstrukcija tuo didesnė tikimybė, jog sakinys bus suanotuotas tinkamai.

6. Literatūra

1. **Wikipedia, Text annotation** http://en.wikipedia.org/wiki/Text_annotation
2. **Wikipedia, Semantics** http://en.wikipedia.org/wiki/Semantics
3. **Kompiuterinė lingvistika, Morfologinis dabartinės lietuvių kalbos tekstyno anotavimas** http://donelaitis.vdu.lt/publikacijos/morf_anotavimas_2007.pdf
4. **Wikipedia, Text mining** http://en.wikipedia.org/wiki/Text_mining
5. **Wikipedia, Ontologija (informatika)** http://lt.wikipedia.org/wiki/Ontologija_(informatika)
6. **Annotation tools** http://annotation.semanticweb.org/tools/
7. **What is the Corpus?** http://www.ruscorpora.ru/en/corpora-intro.html
8. **What are semantic annotation?** http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.7985&rep=rep1&type=pdf
9. **Gate user guide** http://gate.ac.uk/sale/tao/splitch3.html#x6-340003
10. **MmM user manual** http://projects.kmi.open.ac.uk/akt/MnM/MnM_User_Manual.html