KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Arūnas Čiukšys

LIETUVIŲ KALBOS ANAFORŲ SPRENDIMO ĮVERTINIMO GALIMYBIŲ SUDARYMAS

Baigiamasis magistro projektas

Vadovas
mag. V. Žitkus

Konsultantė
prof. dr. L. Nemuraitė

KAUNAS, 2016

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

LIETUVIŲ KALBOS ANAFORŲ SPRENDIMO ĮVERTINIMO GALIMYBIŲ SUDARYMAS

Baigiamasis magistro projektas

Informacinių sistemų inžinerijos studijų programa (kodas 621E15001)

Vadovas

mag. V. Žitkus

2016-05-

Recenzentas

doc. dr. T. Skersys

2016-05-

Projektą atliko

Arūnas Čiukšys

2016-05-

KAUNAS, 2016

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

(Fakultetas)

Arūnas Čiukšys

(Studento vardas, pavardė)

Informacinių sistemų inžinerijos studijų programa, 621E15001

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Lietuvių kalbos anaforų sprendimo įvertinimo galimybių sudarymas“

AKADEMINIO SĄŽININGUMO DEKLARACIJA

20 16 m. gegužės 24 d.

Kaunas

Patvirtinu, kad mano, **Arūno Čiukšio**, baigiamasis projektas tema „Lietuvių kalbos anaforų sprendimo įvertinimo galimybių sudarymas“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka) *(parašas)*

Čiukšys, Arūnas. Lietuvių kalbos anaforų sprendimo įvertinimo galimybių sudarymas. Magistro baigiamasis projektas / vadovas mag. Voldemaras Žitkus; Kauno technologijos universitetas, Informatikos fakultetas.

Mokslo kryptis ir sritis: Informatikos inžinerija, technologijos mokslai

Reikšminiai žodžiai: *anaforų sprendimas, anaforų sprendimo įvertinimas, įrankis ASAS, natūralios kalbos apdorojimas, lietuvių kalbos anaforų etalonas, semantika.*

Kaunas, 2016. 91 p.

SANTRAUKA

Nuolat augant internete publikuojamos informacijos kiekiui, prasminga paieška nestruktūrizuotuose tekstuose tampa vis svarbesnė. Semantinės, ne vien tik raktažodžiais grįstos, paieškos įgyvendinime svarbų vaidmenį atlieka natūralios kalbos apdorojimo sistemos. 2015 metais VDU ir KTU vykdyto projekto Semantika-LT metu buvo sukurta lietuvių kalbos sintaksinės ir semantinės analizės informacinė sistema (LKSSAIS), teikianti vartotojams NKA ir semantinės paieškos tekstuose paslaugas. Tačiau ši informacija leidžia realizuoti ribotą informacijos atpažinimą. Geresnių rezultatų būtų galima tikėtis, jei semantinei tekstų analizei būtų pateikta informacija apie anaforas. Pirmasis bandymas sukurti anaforų sprendimą lietuvių kalbai Semantika-LT projekte parodė, kad tokio sprendimo kūrimas turi būti laipsniškas, nuolat jį vertinant ir analizuojant kokybę. Norint atlikti tai efektyviai, reikia turėti vertinimą ir analizę automatizuojantį įrankį.

Kadangi lietuvių kalboje anaforų sprendimo įvertinimo įrankių nėra, buvo analizuojamos kitoms kalboms naudojamos priemonės, metodika ir jų pritaikymo galimybės dirbti lietuvių kalboje. Nustatyta, kad gali būti taikomi tie patys anaforų sprendimo įvertinimo ir analizės metodai, bet dėl kiekvienoje kalboje esančios skirtingos anaforų klasifikacijos, reikalingas įrankis pritaikytas dirbti su lietuvių kalba. Nuspręsta kurti atskirą nepriklausomą įrankį, kadangi nepritaikyto įrankio adaptavimas, galimai užimtų daugiau laiko nei naujo nepriklausomo įrankio sukūrimas.

Sudarius reikalavimų specifikaciją ir realizacijos projektą, buvo sukurtas protitipinis įrankis ASAS. Eksperimentas parodė, kad, sukūrus įrankį, buvo sudarytos galimybės analizuoti ir įvertinti lietuvių kalbos anaforų sprendimą. Eksperimento metu įrankiu ASAS buvo sudarytas lietuvių kalbos etaloninis tekstynas ir įvertintas realiai kuriamas lietuvių kalbos anaforų sprendimas.

Šiuo metu KTU Informacijos sistemų katedroje yra kuriami lietuvių kalbos automatiniai anaforų sprendimai (projekto Semantika-LT tąsa). Šiame darbe sukurtu prototipiniu įrankiu ASAS galės būti atliekamas jų įvertinimas ir analizė.

Čiukšys, Arūnas. *ENABLING ANAPHORA RESOLUTION EVALUATION AND BENCHMARKING FOR LITHUANIAN*: Master's thesis in Information Systems Engineering / supervisor M.S. Voldemaras Žitkus; The Faculty of Informatics, Kaunas University of Technology.

Research area and field: Informatics Engineering, Technology Science

Key words: anaphora resolution, evaluation of anaphora resolution, tool ASAS, natural language processing, gold corpus in Lithuanian language, semantic.

Kaunas, 2016. 91 p.

SUMMARY

As the amount of information published on the Internet is constantly growing, meaningful search in unstructured texts is becoming more important. Natural language processing systems play an important role in the implementation of a semantic, as opposed to mere keyword-based, search. In 2015, the Information System for Syntactic and Semantic Analysis of the Lithuanian Language (Lith. *Lietuvių kalbos sintaksinės ir semantinės analizės informacinė sistema, LKSSAIS*) was developed during the course of the project *Semantika-LT* implemented by Vytautas Magnus University and Kaunas University of Technology; this system provides users with NLP and semantic text search services. However, this information allows realising limited recognition of information. Better results could be expected if semantic text analysis included information on anaphoras. The first attempt to develop anaphora resolution for the Lithuanian language during *Semantika-LT* revealed that development of such resolution must be gradual by constantly assessing and analysing quality. In order to carry this out efficiently, it is essential to have an automated assessment and analysis tool.

Since the Lithuanian language does not include anaphora resolution assessment tools, tools used in other languages as well as methods and their application opportunities for the Lithuanian language were analysed in detail. It was established that the same anaphora resolution assessment and analysis methods may be used; however, due to the different classifications of anaphoras in different languages, the Lithuanian language requires a tool adapted to it. A decision was made to develop a separate independent tool because adaptation of an ill-adapted tool would possibly take more time than the development of a new independent tool.

Having developed a requirements specification and designed the implementation project, the prototype tool ASAS was developed. The experiment revealed that a developed tool generated opportunities to analyse and assess anaphora resolution in the Lithuanian language. During the experiment, the ASAS tool was used to create a Lithuanian gold corpus and evaluate the factually developed anaphora resolution in the Lithuanian language.

Currently, Lithuanian automated anaphora resolutions are being developed at Kaunas University of Technology Department of Information Systems (follow-up of the project *Semantika-LT*). The prototype ASAS tool developed throughout the course of this thesis could be used to carry out their assessment and analysis.

TURINYS

LENTELIŲ SĄRAŠAS

PAVEIKSLŲ SĄRAŠAS

TERMINŲ IR SANTRUMPŲ ŽODYNAS

Semantinė prasmė	Žodžiai ar frazės įgavę papildomą reikšmę, kurias suteikia to teksto kontekste esantys kiti žodžiai ar frazės.
Semantinė paieška	Tai tokia paieška, kai paieškos rezultatuose pateikiami ne tik tiesioginius raktinius žodžius atitinkantys rezultatai, bet ir pagal žodžių įgautas papildomas reikšmes.
Natūralios kalbos apdorojimas (NKA) (Angl. *Natural Language Processing*)	Tai kompiuterinė natūralios kalbos (kuri gali būti tiek ištarta, tiek rašytinė) analizė ir apdorojimas taikant įvairias technologijas, kurių tikslas - lingvistiniais metodais pritaikyti žmogaus kalbą įvairioms užduotims ar kompiuterinėms programoms.
Anafora **Anaforos pirmtakas** **Anaforos pasekėjas** **Anaforos objektas**	Anafora yra ryšys, kurios objekto reikšmė priklauso nuo kito žodžio tekste [1] – jos pirmtako (angl. *antecedent*) arba pasekėjo (angl. *postcedent*). Žodis, nuo kurios priklauso anaforos objekto reikšmė, vadinama anaforos pirmtaku jei ji yra prieš anaforos objektą arba anaforos pasekėju, jei ji yra po anaforos objekto. Pvz. • *Tomas nebuvo mokykloj. Jis sirgo.* „Tomas“ ir „Jis“ sudaro anaforinį ryšį. „*Tomas*“ yra anaforos objekto „*Jis*“ pirmtakas.
Anaforų sprendimas (Angl. *Anaphora Resolution*)	Tai anaforų identifikavimas ir susiejimas tekste. Anaforų sprendimas įprastai yra automatinis ir atliekamas sudarytos specialios kompiuterinės sistemos.
Anaforų etaloninis tekstynas (Angl. *Gold Standard Corpus*)	Tekstynas su tinkamai žmogaus sužymėtomis anaforomis.
Ontologijos	Tam tikros srities sąvokų visumos specifikavimas išreikštu pavidalu.

ĮVADAS

Nuolat augant internete publikuojamos informacijos kiekiui, prasminga paieška nestruktūrizuotuose tekstuose tampa vis svarbesnė. Prasmingos, ne vien tik raktažodžiais grįstos, paieškos įgyvendinime svarbų vaidmenį atlieka natūralios kalbos apdorojimo (NKA) sistemos, t. y. sistemos, kurios automatiškai išanalizuoja tekstą, nustato jo struktūrą, žodžių morfologines savybes, junginius, sakinių sintaksę, įvardytas esybes ir kitas teksto ir jo dalių savybes. Turint patikimai veikiančias NKA sistemas galima semantinės paieškos tekstuose plėtoti sprendimus, leidžiančius tiksliau atsakyti į vartotojų pateiktas užklausas, sumažinti rankinio paieškos darbo apimtis.

Informaciją apie NKA sprendimus vyraujančioms pasaulio kalboms galima rasti jau nuo 1970 m. Tuo tarpu NKA sprendimai lietuvių kalbai pradėti kurti palyginti neseniai ir ryškesnė pažanga stebima per pastaruosius keletą metų. 2015 metais VDU ir KTU vykdyto projekto Semantika-LT („Lietuvių kalbos sintaksinės – semantinės analizės sistema tekstynui, lietuviškam internetui ir viešojo sektoriaus taikymams“, vykdytas pagal Ekonomikos augimo veiksmų programos 3 prioriteto „Informacinė visuomenė visiems“ įgyvendinimo priemonę Nr. VP2-3.1-IVPK-12-K „Lietuvių kalba informacinėje visuomenėje“) metu buvo sukurta lietuvių kalbos sintaksinės ir semantinės analizės informacinė sistema (LKSSAIS), teikianti vartotojams NKA ir semantinės paieškos tekstuose paslaugas. LKSSAIS semantinė paieška grindžiama tekstų morfologine, sintaksine ir įvardytų esybių informacija, kurią pateikia atitinkami LKSSAIS NKA komponentai. Tačiau ši informacija leidžia realizuoti ribotą informacijos atpažinimą. Geresnių rezultatų būtų galima tikėtis, jei semantinei tekstų analizei būtų pateikta informacija apie anaforas. Anaforų sprendimas yra svarbi NKA sistemos dalis.

Anafora yra vienas iš koreferencijos ryšio tipų, kurį ji detalizuoja: išskiriamas anaforos pirmtakas ir objektas. Anaforų sprendimo [2] metu atpažįstami skirtingose teksto vietose esantys to paties subjekto paminėjimai, kurie susiejami anaforos ryšiu. Anaforos objektai kaip atskiri žodžiai dažnai neturi jokios prasmės (pvz., įvardinės anaforos objektai), bet, susiejus juos su anaforos pirmtaku, jie įgauna jo suteikiamą papildomą prasmę. Dėl to, apdorojant tekstą, yra galimybė iš jo išgauti daugiau informacijos. Pavyzdžiui: „Tomas šiandien nebuvo mokykloj. Jis sirgo.“ Žodžiai „Tomas“ ir „Jis“ sudaro anaforą (siejasi anaforos ryšiu). Be anaforų sprendimo negalėtume žinoti, kodėl šiandien Tomas nebuvo mokykloje arba kas sirgo. Be natūralios kalbos apdorojimo netenkama daug panašios informacijos.

Pirmasis bandymas sukurti anaforų sprendimą lietuvių kalbai Semantika-LT projekte parodė, kad tokio sprendimo kūrimas turi būti laipsniškas, nuolat jį vertinant ir analizuojant kokybę. Norint atlikti tai efektyviai, reikia turėti vertinimą ir analizę automatizuojantį įrankį.

Kadangi lietuvių kalboje anaforų sprendimo įvertinimo įrankių nebuvo, reikėjo analizuoti kitoms kalboms naudojamas priemones, metodiką ir jų pritaikymo dirbti lietuvių kalboje galimybes. Nustatyta, kad gali būti taikomi tie patys anaforų sprendimo įvertinimo ir analizės metodai. Anaforų sprendimo įvertinimas atliekamas palyginant anaforų sprendimo rezultatus su sudarytu etalonu ir apskaičiuojant išsamumą, tikslumą, F-vertę ir nuorodų dažnumą – kitų kalbų anaforų sprendimams įvertinti naudojamus dydžius. Kadangi jokiai kalbai nėra anaforų sprendimo, atpažįstančio anaforas be klaidų, etaloną gali sudaryti tik žmogus. Sudarytas etalonas turi būti kuo išsamesnis, turintis skirtingų žanrų tekstus, kadangi tas pats anaforų sprendimas gali skirtinguose tekstuose pasirodyti skirtingai. Norint anaforų sprendimus palyginti tarpusavyje, reikia lyginti jų rezultatus sprendžiant to pačio tekstyno anaforas.

Nors įvertinimo metodika gali būti taikoma ta pati, bet dėl kiekvienoje kalboje esančios skirtingos anaforų klasifikacijos reikalingas įrankis, pritaikytas dirbti su lietuvių kalbos tekstais. Įrankiui buvo iškelti ir kiti reikalavimai. Įrankis turėjo palaikyti JSON duomenų mainų formatą (kadangi projekte Semantika-LT realizuotos NKA priemonės palaiko JSON formatą, tikslinga pratęsti jo naudojimą). Taip pat buvo ir daugiau reikalavimų, kurie išsamiau aprašyti analizės skyriuje. Kadangi be papildomos adaptacijos tinkamo įrankio nebuvo rasta, o nepritaikyto įrankio adaptavimas galimai užimtų daugiau laiko nei naujo nepriklausomo įrankio sukūrimas, buvo nuspręsta kurti atskirą, naują nepriklausomą įrankį.

Šio darbo tikslas yra sudaryti lietuvių kalbos anaforų sprendimo įvertinimo galimybes sukuriant anaforų sprendimo įvertinimo ir analizės metodiką realizuojantį įrankį.

Norint pasiekti tikslą, turi būti atlikti šie uždaviniai:

1. Atlikti anaforų sprendimo ir jo įvertinimo galimybių analizę;
2. Sudaryti įvertinimo ir analizės metodiką realizuojančio įrankio reikalavimų specifikaciją, projektą;
3. Sudaryti įrankio realizacijos projektą;
4. Realizuoti ir ištestuoti įrankį;
5. Atlikti eksperimentą, kuris patvirtintų sudarytas anaforų sprendimo įvertinimo galimybes;
6. Apibendrinti tyrimo rezultatus.

Reikalavimų specifikacijos ir projektavimo etape buvo sudaryti įrankiui keliami funkciniai ir nefunkciniai reikalavimai.

Realizavimo etape, remiantis reikalavimu etape sudarytais modeliais bei specifikacijomis, buvo sudarytas realizacijos projektas.

Realizavus protitipinį įrankį ASAS, buvo sudarytas testavimo modelis, kuris padėjo aptikti ir ištaisyti įrankio klaidas bei patvirtinti įrankio atitikimą reikalavimams. Testavimo metu buvo testuojami tiek funkciniai, tiek nefunkciniai reikalavimai.

Atliktas eksperimentas parodė, kad, sukūrus įrankį, buvo sudarytos galimybės analizuoti ir įvertinti lietuvių kalbos anaforų sprendimą. Eksperimentą sudarė 2 dalys: pirmoje dalyje buvo sėkmingai įvertinti imitaciniai anaforų sprendimo rezultatai, o antroje dalyje buvo įvertintas realus KTU Informacinių sistemų katedroje kuriamas anaforų sprendimas. Antros dalies eksperimentui atlikti įrankiu ASAS buvo sudarytas lietuvių kalbos anaforų anotavimo etaloninis tekstynas, kurį sudarė 39 skirtingi tekstai (3 skirtingų kategorijų) iš Vytauto Didžiojo universiteto (VDU) ir Kauno technologijos universiteto (KTU) projekto „Lietuvių kalbos sintaksinės-semantinės analizės sistema tekstynui, lietuviškam internetui ir viešojo sektoriaus taikymams“ (VP2-3.1-IVPK-12-K-01-007) sudaryto ekonomikos ir teisės bei politikos tekstynų. Bendra sudaryto etaloninio tekstyno apimtis – 87914 simbolių. Etaloną sudarė KTU socialinių, humanitarinių mokslų ir menų fakulteto bakalauro studijų studentė.

Šiuo metu KTU Informacijos sistemų katedroje yra kuriami lietuvių kalbos automatiniai anaforų sprendimai (projekto Semantika-LT tąsa). Šiame darbe sukurtu prototipiniu įrankiu ASAS galės būti atliekamas jų įvertinimas ir analizė.

Buvo pristatyti 2 straipsniai, susiję su šiame darbe atliekamu tyrimu. 2015 metų tarpuniversitetinėje magistrantų ir doktorantų konferencijoje „IT 2015“ buvo pristatytas straipsnis „Ko-referencijų sprendimo įrankio sukūrimo lietuvių kalbai galimybių analizė“. Šio darbo rezultatai buvo pristatyti 2016 metų tarpuniversitetinėje magistrantų ir doktorantų konferencijoje „IVUS 2016“. Buvo pristatytas straipsnis „ASAS – lietuvių kalbos anaforų sprendimo analizės ir įvertinimo prototipas“. Straipsniai buvo išspausdinti konferencijų leidiniuose ir pateikiami šio darbo prieduose.

1. ANAFORŲ SPRENDIMO IR JO ĮVERTINIMO GALIMYBIŲ ANALIZĖ

1.1. Analizės tikslas

Nustatyti anaforų sprendimo įvertinimo ir analizės galimybes: taikomą metodiką, naudojamus įrankius. Nustatyti, kaip būtų galima metodus ir priemones pritaikyti darbui su lietuvių kalba.

Norint pasiekti tikslą, turi būti įgyvendinti šie uždaviniai:

1. Išnagrinėti anaforų algoritmus ir sprendimus;
2. Nustatyti anaforų sprendimo įvertinimo specifiką: taikomą įvertinimo metodiką ir kriterijus;
3. Išanalizuoti anaforų specifiką;
4. Išsiaiškinti ir palyginti įrankius, kuriais galėtų būti atliekamas anaforų sprendimų įvertinimas ir analizė.

1.2. Tyrimo objektas, sritis ir problema

Tyrimo problema. Šiuo metu kuriami lietuviško teksto automatiniai anaforų sprendimai [1], tačiau reikia juos įvertinti.

Tyrimo objektas. Lietuvių kalbos anaforų sprendimo įvertinimo procesas.

Tyrimo sritis. Natūralios kalbos apdorojimas: anaforos, anaforų sprendimas, anaforų sprendimo įvertinimas.

1.3. Lietuvių kalbos anaforų sprendimo įvertinimo proceso analizė

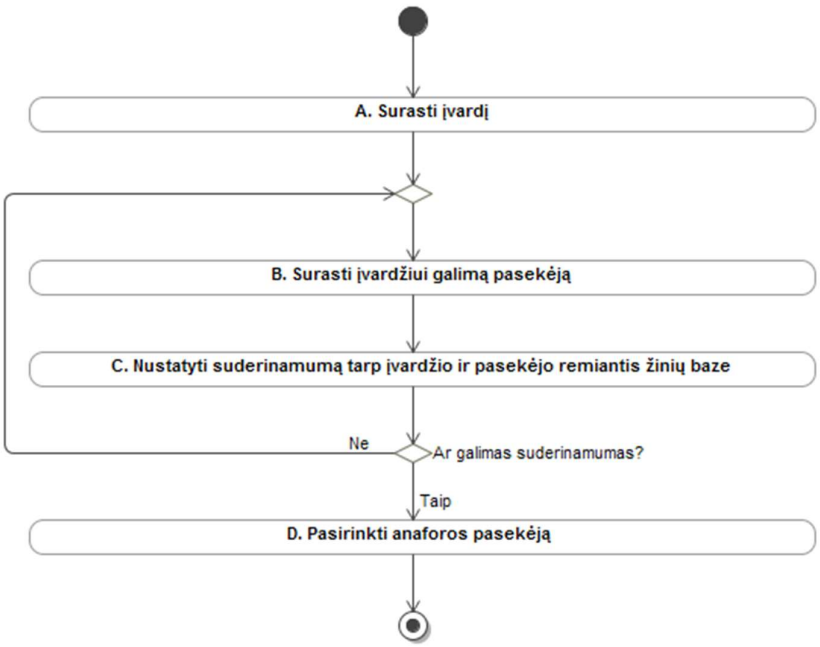
1.3.1. Anaforų algoritmai

Anaforų sprendimai visose kalbose kuriami remiantis algoritmais, skirstomais į dvi pagrindines kategorijas – paremtais žiniomis algoritmais (angl. *knowledge-rich*) ir neparemtais žiniomis algoritmais (angl. *knowledge-poor*). [3]

Paremti žiniomis algoritmai – tai algoritmai, kurių neatsiejama dalis yra žmogaus rankiniu būdu sudaryta žinių bazė. Daugelis pirmųjų algoritmų buvo šios kategorijos. Populiariausias yra Hobso algoritmas, kurio pagrindu yra vis dar kuriami nauji algoritmai.

- *Hobso (angl. Hobbs') algoritmas* [4]

Hobso algoritmo veikimas pavaizduotas 1 pav. Remiantis algoritmu, pirma tekstas turi būti iki galo išskaidomas sintaksiniu medžiu. Tuomet, medžio analizės metu, medyje surandamas įvardis. Paskui surandamas po įvardžio einantis jam galimas pasekėjas. Remiantis žinių baze yra tikrinamas jų suderinamumas pagal lytį, kiekį, asmenį ir kt. Nustačius pirmąjį suderinamumą, jis ir pasirenkamas. Kitų nebeieškoma. Jei ir būtų galimi keli kandidatai, atitinkantys visus tris suderinamumo punktus, algoritmas pasirinks pirmąjį rastą, nors teisingas variantas gali būti kitas. Šiai problemai spręsti Hobsas siūlo sudaryti išimtis (apribojimus) - aksiomas, paremtas realiu žmogaus pasaulio suvokimu. Tai yra sudėtinga, kadangi sukurti tokias išimtis reikia labai daug laiko ir jų gali būti labai daug. Skirtingiems teksto tipams gali reikėti sudaryti skirtingus apribojimus, kurie gali prieštarauti vienas kitam. Nepaisant šio algoritmo trūkumų, jis yra naudojamas kaip bazė, kuriant naujus algoritmus.



1 pav. Hobso algoritmas

- *Centravimo (angl. centering) teorija paremti algoritmai* [5]

Pagrindinė centravimo algoritmo prielaida – prieš tai buvusiame sakinyje daiktavardis yra naudojamas kaip įvardis sekančiame sakinyje. Vėliau jų suderinamumas tikrinamas žinių bazėje. Bet pagal šį algoritmą kuriami sprendimai atitinkamo įvardžio pirma ieško tame pačiame sakinyje ir tik paskui kituose. Nors šiuo atveju analizei naudojamas sintaksinio medžio principas (kaip ir Hobso algoritme), ne visada tekstas turi būti iki galo išskaidytas sintaksiniu medžiu.

- *Aktyvavimo (angl. activation) teorija paremti algoritmai*

Veikia taip pat kaip ir centravimo algoritmai, bet skiriasi tuo, kad analizuojamos ir netiesioginės reikšmės. Pvz.: *„Paukščiai jau pradėjo migruoti dėl vėlyvo rudens.“* Žodis *migracija* nebuvo tiesiogiai paminėtas sakinyje, bet jis laikomas galima prielaida ir parenkamas panašiai, kaip parenkamas anaforos pirmtakas pagal Hobso algoritmą.

- *Universalios tinklo kalbos (angl. Universal Networking Language) algoritmas*

Algoritmas remiasi panašia teorija kaip ir centravimo algoritmai. Šiuo atveju remiamasi ryšių tipais, pagal kuriuos galimi pirmtakai buvo nustatyti prieš tai buvusiame sakinyje. Semantinė informacija yra naudojama sudarant universalios tinklo kalbos diagramą, remiantis naudotais įvardžių tipais ir semantinėmis reikšmėmis. Šis algoritmas buvo pasiūlytas morfologiškai turtingai tamilų kalbai ir gali netikti kitoms kalboms.

- *Semantinio suderinamumo algoritmas*

Nustatomas semantinis suderinamumas tarp įvardžio ir galimo pirmtako. Pvz.: *„Tomas buvo labai pavargęs ir vairuodamas automobilį vos neužmigo.“* Nustatoma, kad Tomas yra asmuo, kad žmogus gali užmigti ir kad pavargęs žmogus yra linkęs užmigti. Todėl nustatoma, kad *„Tomas buvo pavargęs“* ir *„Tomas vos neužmigo“* yra semantiškai suderinami. Algoritmas taip pat nustato sintaksinį atstumą tarp įvardžio ir galimo pirmtako. Semantinis suderinamumas nustatomas remiantis transformuotomis OWL technologijomis.

Neparemti žiniomis algoritmai. Paremti iš esmės automatiniu būdu sudaroma žinių baze ir daug mažiau žmogaus rankiniu būdu sudaryta informacija.

- *Statistikos teorija paremti algoritmai*

Sudaromas tikimybinis modelis remiantis atstumu tarp įvardžio ir galimo pirmtako; vieta sintaksiniame medyje; lytimi; iškilumu; sąveikos tarp įvardžio pagrindinės dalies, pirmtako ir galimų pirmtakų kiekio (teikiama pirmenybė dažniau minimiems pirmtakams).

- *Automatinio apsimokymo teorija paremti algoritmai*

Tai yra sistemos, kurios atlieka įvairias natūralios kalbos apdorojimo užduotis ne tik anaforų sprendimo srityje. Kitų natūralios kalbos apdorojimo sudėtinių dalių trūkumai daro neigiamą poveikį anaforų sprendimui. Bandymai pagerinti vieną iš pirmųjų automatinio apsimokymo algoritmų, anaforų sprendimui išplečiant nuo 12 kryptinių savybių iki 53, sumažino kelis jo įvertinimo kriterijus - išsamumą ir tikslumą. Nors pagrindinio funkcionalumo lingvistiniai išplėtimai pagerino anaforų sprendimą.

- *Kiti metodai*

Statistikos teorija remiamasi ne kaip pagrindu, bet kaip papildomu įrankiu nuspręsti, kurį pirmtaką pasirinkti, kai randami keli galimi kandidatai.

Dažniausiai naudojamų algoritmų įvertinimas ir palyginimas yra pavaizduotas 1 lentelėje. Ne visi anaforų algoritmai nustato visas anaforas, t. y. sugeba atpažinti tik tam tikrus anaforų tipus, pavyzdžiui, įvardinių anaforų nustatymą.

1 lentelė. Algoritmų palyginimas [1]

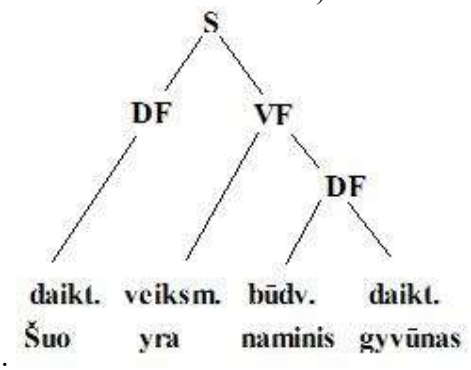
Algoritmas	Tipas	Spręsti anaforų išraiškų tipai	Tikslumas
Hobso	Sintaksinis	Pagrindiniai įvardžiai: jis, ji, jie (angl. *he, she, they, it*)	81.8-91.7% priklausomai nuo teksto tipo
BFP	Centravimo teorija	Įvardžiai (nepaminėti kurie)	49-90% priklausomai nuo teksto tipo
Kairė-dešinė centravimas	Modifikuota centravimo teorija	Įvardžiai (nepaminėti kurie)	72.1-81% priklausomai nuo teksto tipo
Statistiniai algoritmai	Tikimybinis (angl. *probabilistic*) modelis	Jis, ji (angl. *he, she, it*) ir jų (angl. *their*) keletas formų	82.9-84.2%
Automatinio apsimokymo	Automatinio apsimokymo	Daiktavardžių frazės (įskaitant įvardžius)	65.5-67.3%
Anaforų sprendimas naudojant universalią tinklo kalbą	Universali tinklo kalba	Įvardžiai	67%

1.3.2. Anaforų sprendimas

Algoritmai negali būti naudojami be pritaikymo kalbos specifikai, nors daugeliui kalbų, tarp jų ir anglų, kuriami anaforų algoritmai paremti tais pačiais anaforų algoritmais [6]. Ta pati sudėtis sudarė ir sprendimą artimiausiai lietuvių kalbai – latvių kalbai [7]. Šiuo metu daugelis kuriamų anaforų sprendimų grįsti žiniomis paremtais metodais, kuriems reikalingi pirminio apdorojimo uždaviniai. Dėl to anaforų sprendimas nepriklausomai nuo kalbos gali būti skaidomas į du etapus: pirminio apdorojimo etapą (angl. *preprocessing*) ir esminio apdorojimo etapą (angl. *processing*). Pirminio apdorojimo metu yra atliekami tam tikri morfologinio kalbos apdorojimo, sintaksinės analizės (angl. *parsing*) ir įvardytų esybių nustatymo (angl. *named-entity recognition*) uždaviniai [8]. Esminio apdorojimo metu atliekami tam tikri specifiniai anaforos ryšių identifikavimo veiksmai, kurie priklauso nuo algoritmo specifikos.

Sintaksinė analizė [9] yra struktūrinio aprašymo suteikimas simbolių eilutei, remiantis tam tikra gramatika. Sintaksinę analizę atlieka sintaksinis analizatorius (angl. *parser*) – šį uždavinį sprendžianti kompiuterinė programa. Sintaksinė analizė skirstoma į sudedamųjų dalių struktūros sudarymą (angl. *part-of-speech tagging*) ir priklausomybių struktūros sudarymą (angl. *relationships tagging*). Sintaksinę struktūrą įprasta vaizduoti medžio tipo diagramomis.

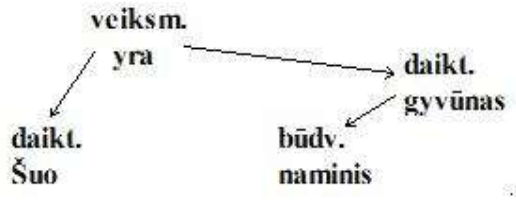
Sudedamųjų dalių struktūros [9] sudarymo metu kiekvienas žodis tekste yra priskiriamas jam priklausančiai kalbos daliai tiek remiantis apibrėžimu, tiek pagal kontekstą, kuriam žodis priklauso. Struktūros diagramos viršūnės atitinka tam tikro ilgio įėjimo eilutės segmentus, o briaunos nusako, kaip ilgesni segmentai sudaryti iš mažesnių. Viršutinis lygis atitinka visą sakinį. Pavyzdžiui, (S – sakinys, DF – daiktavardžio frazė, VF – veiksmažodžio frazė):



2 pav. Sudedamųjų dalių struktūra [9]

Skirtingame kontekste žodžiai gali sudaryti skirtingas kalbos dalis. Kalbos dalių žymėjimas seniau buvo vykdomas tik rankiniu būdu. Dabar kalbos dalių žymėjimas atliekamas realizuojant automatinius algoritmus, kurie yra skirstomi į paremtus taisyklėmis ir stochastinius (tikimybinius). Vienas iš paprasčiausių kalbos dalių sprendimų anglų kalbai yra Eriko Brilo sudedamųjų dalių struktūros sudarytojas (Eric Brill POS tagger [10]). Jis iš pradžių priskiria kiekvienam žodžiui kalbos dalį, remiantis tikimybe, kokia kalbos dalis dažniausiai tam žodžiui yra priskiriama. Vėliau, priskyrus žodžiams kalbos dalis, atliekama kita iteracija ir ieškoma klaidų analizuojant žodžius smulkesniame kontekste.

Priklausomybių struktūroje [9] (pavyzdys pateikiamas 3 pav.) mazgai yra elementarūs įėjimo eilutės segmentai, o ryšiai nusako sintaksinius ryšius tarp elementaraus segmento ir nuo jo priklausomo priklausomybių struktūros pomedžio.



1 pav. Priklausomybių struktūra [9]

Įvardytų esybių analizės metu teksto elementai suskirstomi į iš anksto apibrėžtas kategorijas – klasifikuojami į žmonių vardus, įmones, vietoves, laiko elementus, kiekius, pinigų vienetus ir kt. Pavyzdžiui:

[Paulius]$_{asmuo}$ [2015]$_{laikas}$ m. nusipirko 100 [UAB „Net septyni“]$_{įmonė}$ akcijų.

Geriausios sistemos, skirtos anglų kalbai, sugeba teisingai identifikuoti vaidmenis didesniu nei 90% tikslumu.

1.3.3. Anaforų sprendimo įvertinimo metodika ir charakteristika

XX a. 9-ojo dešimtmečio pabaigoje buvo organizuojamos angliško teksto automatinio natūralios kalbos apdorojimo sistemų konferencijos *MUC* (angl. *Message Understanding Conference*). Pagrindinė konferencijų paskirtis buvo įvertinti ir tarpusavyje palyginti teksto apdorojimo sistemas, joms atliekant tas pačias užduotis, t. y. apdorojant tuos pačius tekstus siekiant gauti tuos pačius rezultatus [11], [12]. Sprendimams vertinti buvo pasirinkta naudoti dydžius R (išsamumas), P (tikslumas) ir F-vertė.

Išsamumas R – (1) formulė, kilęs iš signalų teorijos. Parodo santykį tarp gautų teisingų rezultatų C ir visų tekste esančių rezultatų T kiekių.

$$R = C / T \quad (1)$$

Tikslumas P – (2) formulė. Parodo santykį tarp tekste gautų teisingų rezultatų C ir visų gautų rezultatų F kiekių.

$$R = C / F \quad (2)$$

Šioms charakteristikoms papildomai vertinti naudojama F-vertė (angl. *F-measure*) – (3) formulė. Tai yra harmoninė išraiška tarp tikslumo (P) ir išsamumo (R).

$$F = (2 \; x \; P \; x \; R) / (P + R) \quad (3)$$

MUC konferencijų metu pasirinkti įverčiai ir sudaryti tekstynai, iš dalies ar pilna apimtimi naudojami iki šiol. Jais įvertinamos įvairios anglų ir kitų kalbų automatinės teksto apdorojimo sistemos. Naudojami ne tik minėtieji įverčiai, bet ir nuorodų dažnumas RR – (4) formulė. Panašus į tikslumą matas, bet vietoje vardiklyje naudojamo sistemos gautų teisingų rezultatų kiekio imama visų tekste esančių rezultatų (T) ir sistemos blogų rezultatų (E) kiekių suma.

$$RR = C / (T + E) \quad (4)$$

Šiuo metu anglų kalboje vertinant įvairius sprendimus yra naudojami MUC konferencijų tekstų rinkiniai [13], [14]. Tokiu būdu sprendimai gali būti palyginami pavieniui (angl. *off-site*).

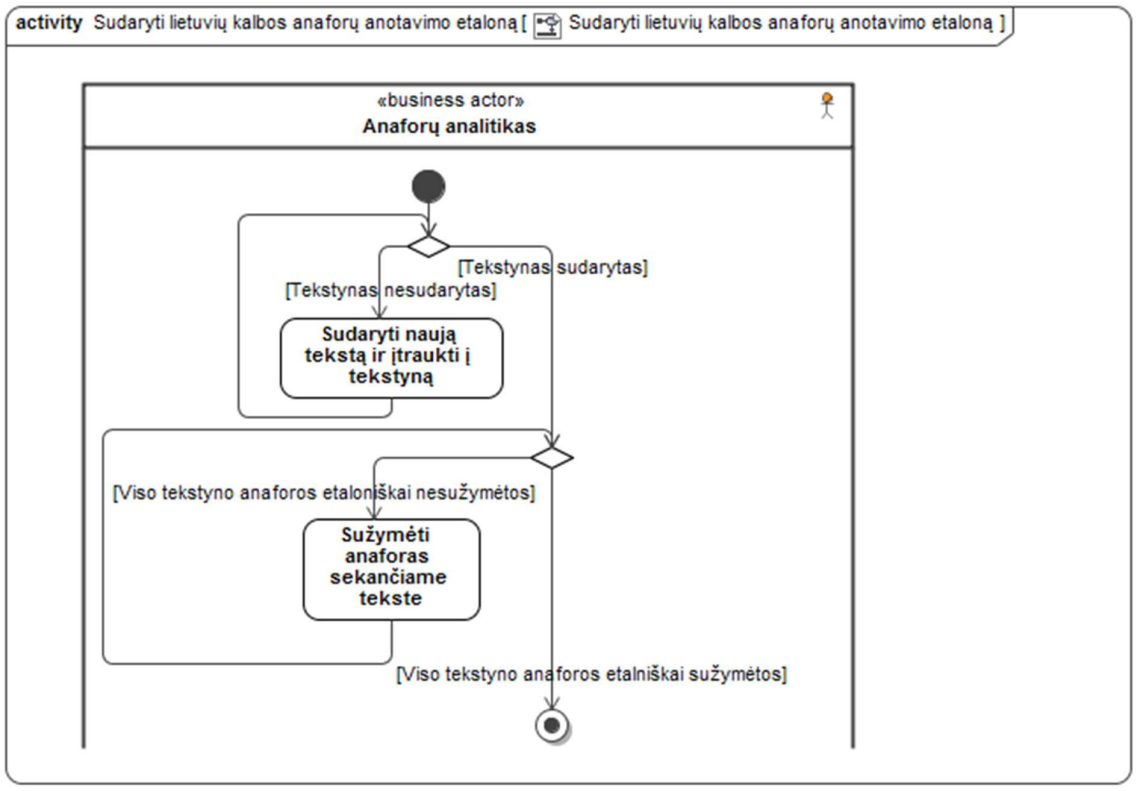
Automatinių sprendimų įvertinimas apdorojant skirtingą tekstą negalėjo būti lyginamas, nes skirtingi algoritmai vertinant skirtingo ilgio ar žanro tekstus gali kiekvienąkart pasirodyti skirtingai. Todėl MUC konferencijose automatiniams sprendimams įvertinti reikėjo pateikti tas pačias užduotis. Tuo tikslu buvo sukurtas tekstynas ir sudarytas tekstyno etalonas (angl. *gold standard corpus*). Norint išvengti netikslumų dėl minėtų problemų, tekstas turėjo būti ilgas ir turintis daug skirtingų žodžių. Trečiosios MUC konferencijos metu buvo naudojamas tekstų rinkinys, kurį sudarė 400 000 žodžių, iš kurių 18 000 buvo skirtingi.

Kadangi MUC konferencijose buvo įvertinami skirtingi natūralios kalbos apdorojimo sprendimai, jie turėjo įvertinimo sistemai rezultatus pateikti vieningu formatu. Tam buvo naudojamas XML formatas. Pagrindiniai formatai, kuriuos naudoja įvairūs sprendimai pastarąjį dešimtmetį [15], [16], [17], veikiantys natūralios kalbos apdorojimo srityje, yra XML ir JSON. XML formatas, lyginant su JSON, labiau apkrauna kompiuterio laikinąją atmintį, bet lengviau skaitomas žmogui ir galimas didesnis apdorojimo greitis. Ir atvirkščiai: naudojant JSON, reikia mažiau atminties, bet sunkiau žmogaus skaitomas, o programinis apdorojimas lėtesnis nei XML. Tiek pirmieji kuriami automatiniai anaforų sprendimai, tiek kiti natūralios kalbos apdorojimo sprendimai lietuvių kalboje (projektas Semantika-LT) duomenims pateikti naudoja JSON formatą. Siekiant išlaikyti kitose sistemose priimtų sprendimų tęstinumą ir taip užtikrinti naujų sprendimų integravimą, būtų tikslinga pasirinkti JSON formatą.

Dėl skirtumų tarp kalbų anaforų sprendimo sistemos negali būti daugiakalbės ir lietuvių kalbai angliški MUC tekstynai negali būti naudojami. Bet lietuvių kalbos tekstyno, kuriam sudarytas anaforų anotavimo etalonas, šiuo metu nėra. Tekstyno ir jo etalono sudarymo procesas yra labai sudėtingas: surinkti tinkamo ilgio ir sudėtingumo tekstus, turinčius daug skirtingų žodžių, sudaryti jų anotavimo etalonus, sužymint anaforas tekste rankiniu būdu, gali tik lietuvių kalbos specialistas (anaforų analitikas), o norint sprendimus palyginti tarpusavyje skirtingi kūrėjai turi naudoti tą patį tekstyną. Todėl kompiuterizuotą įrankį tikslinga naudoti ne tik tekstynui ir jo etalonui sudaryti, bet ir norint sudaryti galimybę skirtingiems anaforų sprendimo kūrėjams prieiti prie sudaryto tekstyno. Priėjimas prie vieningos sudarytos duomenų bazės įgalina anaforų sprendimo įvertinimą pavieniui. Be to, sutaupo resursų, nes nereikia sudarinėti tekstyno ir jo etalono kiekvienąkart iš naujo. Antra, net ir naudojant bendrai prieinamą tą patį tekstyną, norint gauti sprendimų įvertinimo dydžius tikslinga naudoti kompiuterizuotą įrankį – verčių apskaičiavimas rankiniu būdu būtų ypač imlus darbui ir laikui.

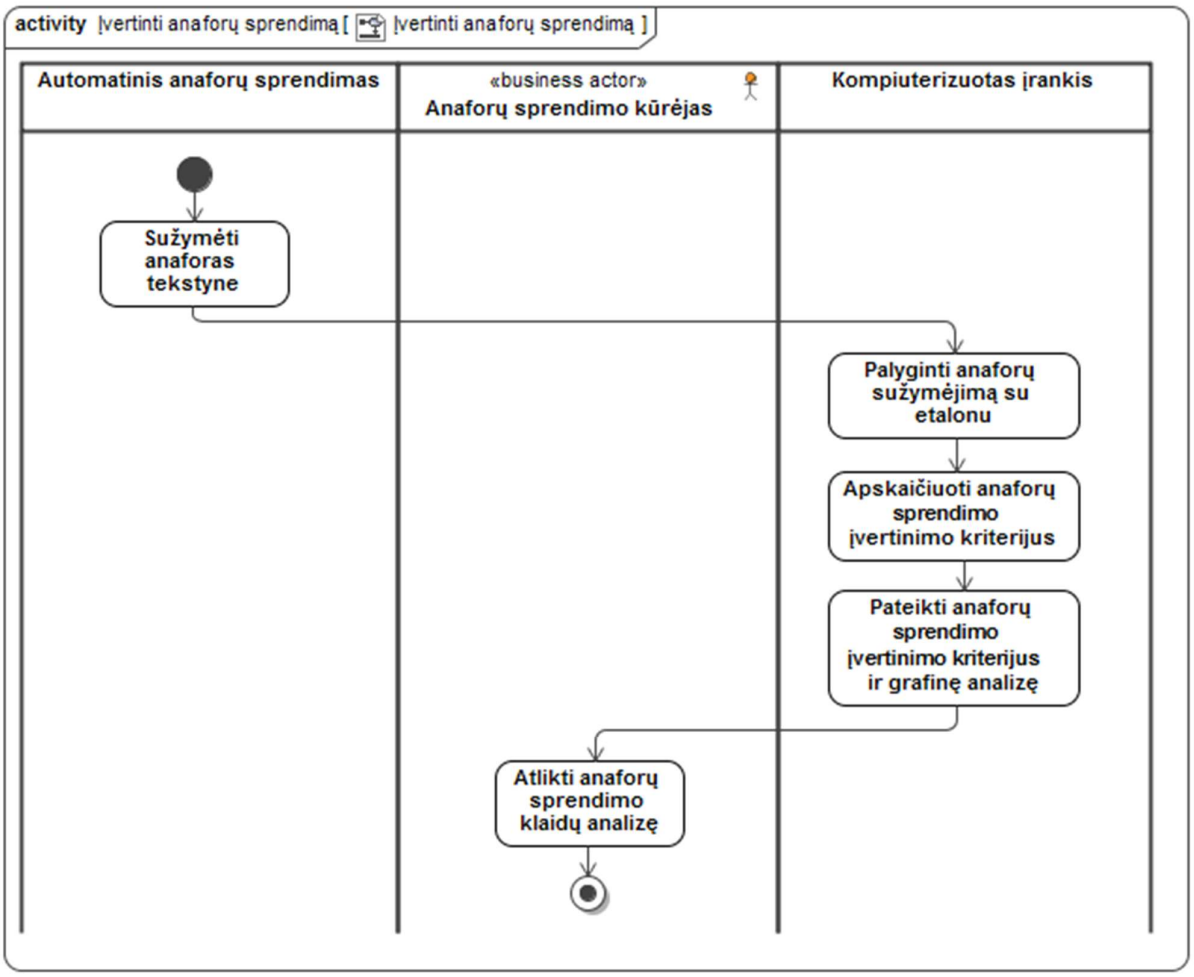
Taigi anaforų sprendimo įvertinimas susideda iš dviejų dalių.

Pirma dalis. Tekstyno ir etalono sudarymas (3 pav.). Anaforų analitikas sudaro įvairių sričių ir žanrų (pvz. interviu, moksliniai straipsniai, naujienų straipsniai) bei dydžių tekstų rinkinius ir teisingai sužymi anaforas rankiniu būdu. Kiekvienam anaforos ryšiui yra priskiriamas vienas iš tipų.



3 pav. Etalono sudarymo veiklos modelis

Antra dalis. Anaforų sprendimo įvertinimas (4 pav.). Automatinis anaforų sprendimas apdoroja turimą tekstyną ir sužymi anaforas. Sužymėjimai yra palyginami su etalonu ir yra išvedamos išsamumo, tikslumo, F-vertės ir nuorodų dažnumo įvertinimo charakteristikos. Kūrėjui, kuriančiam anaforų sprendimo sistemą, svarbu pamatyti konkrečius sistemos įverčius. Tokiu būdu kūrėjas gali žinoti savo progresą. Bet tam, kad kūrėjas galėtų tobulinti sistemą, reikia žinoti, kurias anaforas sistema atpažino teisingai, o kurių neatpažino ar atpažino klaidingai. Be grafinio rezultatų atvaizdavimo detali, rankiniu būdu atliekama rezultatų analizė būtų ypač imli darbui. Rezultatų analizės efektyvumui pagerinti tikslinga naudoti grafinį sprendimo rezultato su etalonu palyginimą, kurį galėtų pateikti kompiuterizuotas įrankis.



4 pav. Anaforų sprendimo įvertinimo veiklos modelis

1.3.4. Anaforų tipai

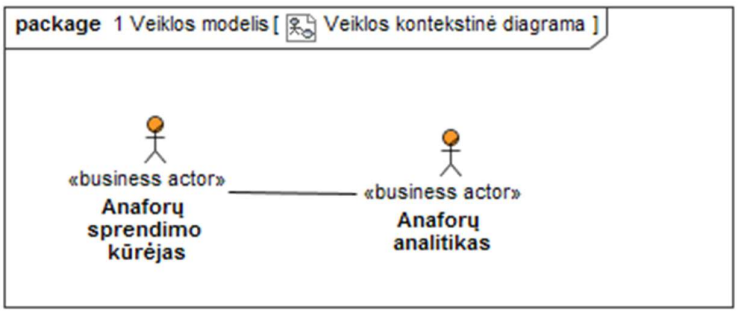
Anaforos yra skirstomos į tipus. Pagal anaforos objekto tipą ir dėl skirtumų tarp kalbų jos kiekvienoje kalboje klasifikuojamos skirtingai. 2014 m. buvo pasiūlyta pirmoji lietuvių kalbos anaforų taksonomija [18], kurioje buvo išskirti 3 anaforų tipai: morfologinio, leksinio semantinio ir dalykinės srities tipo. Tipai buvo suskirstyti į daugiau potipių. Pavyzdžiui, tikrinio daiktavardžio – *Tomas, Petras*; vietos prieveiksmio – *čia, ten*; asmeninio įvardžio – *aš, tu, jie, jos*. Daugelis kitoms kalboms esamų anaforų sprendimų nesugeba atpažinti visų anaforų tipų.

Pirmieji anaforų sprendimai, kuriami lietuvių kalbai, yra skirti įvardinėms anaforoms atpažinti. Svarbu, kad sprendimo įvertinimo įrankis sudarytų galimybę, kuriant etaloninį tekstyną, išskirti anaforų tipus ir įvertinti anaforų sprendimą sprendžiant tik tam tikro tipo anaforas. Be to, kol nėra nusistovėjusios ir bendrai pripažįstamos anaforų taksonomijos lietuvių kalbai, svarbu įrankyje turėti galimybę laisvai sukurti naujus tipus ar redaguoti esamus.

1.4. Lietuvių kalbos anaforų sprendimo įvertinimo proceso naudotojų analizė

Lietuvių kalbos anaforų sprendimo įvertinimo procese dalyvauja 2 naudotojų tipai (5 pav.):

1. Anaforų sprendimo kūrėjas, kuris sudaro lietuvių kalbos anaforų sprendimus;
2. Anaforų analitikas, kuris sudaro lietuvių kalbos tekstynus ir jų etalonus.



5 pav. Veiklos sąveikų modelis

1.5. Esamų anaforų anotavimo / įvertinimo įrankių analizė

Esamiems įrankiams palyginti pasirinkti 8 kriterijai: koreferencijų anotavimo galimybė, anaforų anotavimo galimybė, įvertinimo rezultatų pateikimas matais R, P, F, RR, bendros prieigos galimybė, duomenų mainų palaikymas *JSON* formatu, skirtingų anotacijų grafinio palyginimo galimybė, įrankio tobulinimo galimybė ir ar įmanoma išmokti naudotis įrankiu per 1 dieną. 2 lentelėje buvo palyginti daugiausia kriterijų tenkinantys 3 įrankiai.

1.5.1. GATE įrankių grupė

GATE [16], [19], [20] įrankiai yra kuriami ir palaikomi ne pelno siekiančios organizacijos, todėl visi grupės komponentai yra atvirojo kodo (sukurti *Java* programavimo kalba). *GATE* yra apdorojimo komponentų integruota kūrimo aplinka (angl. *integrated development environment*) bei informacijos išgavimo (angl. *Information Extraction*) sistema, pateikiama kartu su daug plėtinių, leidžiančių sistemą adaptuoti įvairiems specifiniams poreikiams. Duomenų mainams su išoriniais sprendimais *GATE* naudojamas *XML* formatas. Bet yra sukurtas specifinį *Twitter JSON* palaikantis įskiepis. Todėl standartiniam *JSON* formatui turėtų būti kuriamas papildomas įskiepis. Kalbos specifinės raidės palaikomos naudojant unikodą.

Pagrindinis *GATE* komponentas yra vartotojo kompiuteryje paleidžiama taikomoji programa *GATE Developer*, kurią naudojant galima anotuoti ir grafiškai analizuoti koreferencijų grandinėlę. Taip pat įrankis turi integruotą skirtingų anotacijų įvertinimo funkciją (*GATE AnnotationDiff*), kuri automatiškai apskaičiuoja tikslumą, išsamumą ir F-vertę. Bet be papildomų priedų *GATE Developer* negalima žymėti anaforų pirmtakų ir objektų. Jau sukurtų tokių priedų rasti nepavyko. Todėl jie turėtų būti sukurti papildomai. *GATE Teamware* naršyklėje veikianti (internetinė) programa ir *GATE Cloud* sudaro bendradarbiavimo galimybes kuriant tekstynus ir jų etalonus.

GATE yra plačiai taikoma sistema, skirta įvairiems NKA naudotojams:

1. programuotojams, kuriantiems programinę natūralios kalbos apdorojimo įrangą;
2. kalbos tyrėjams;
3. kalbos apdorojimo mokytojams ir dėstytojams.

Dėl plataus programos pritaikymo galimybių *GATE* išmokti naudotis yra sunku.

1.5.2. eHost

eHost [20] įrankis sukurtas Sveikatos apsaugos informacinių tyrimo konsorciumo (angl. *A Consortium for Healthcare Informatics Research*), į kurį įeina Jutos universitetas ir Solt Leik Sičio sveikatos apsaugos sistema. Kūrėjų įrankis, kurio pirminė paskirtis yra medicinos tekstų anotavimas, naudojamas įvairiuose projektuose nuo 2010 m.

Pavadinimas *eHost* yra termino „išplėstinis žmogaus įrankių rinkinys“ (angl. *extensible Human Oracle Suite of Tools*) santrumpa. Tai yra atvirojo kodo *Java* programavimo kalba sukurtas įrankis, veikiantis interneto naršyklėje vietiniame kompiuteryje. *eHost* įgalina konceptų, jų savybių ir ryšių tarp jų anotavimą.

Pagrindinės įrankio savybės: tekstų generavimas ir pradinis automatinis anotavimas remiantis žodynu; rankinis ryšių anotavimas; kreipiniai į žodynus naudojant adaptuojamą programavimo sąsają (angl. *API*); duomenų mainai *XML* formatu.

1.5.3. Anafora

Anafora [17] yra anotavimo įrankis, sukurtas Kolorado universitete. Tai yra atvirojo kodo įrankis, sukurtas *Python* programavimo kalba. Kuriant *Anaforą*, pagrindiniai kriterijai buvo naudojimosi paprastumas ir universalaus taikymo galimybė – vykdant tiek smulkius, tiek stambius projektus. Pabrėžiamas įrankio privalumas ir išskirtinumas: *Anafora* veikia debesyje (angl. *Cloud*) – paleidžiama interneto naršyklėje, o duomenys saugomi nutolusioje tarnybinėje stotyje. Todėl *Anafora* gali veikti bet kurioje operacinėje sistemoje. *Anafora* duomenų mainams naudoja *XML* formatą.

2 lentelė. Esamų įrankių palyginimas

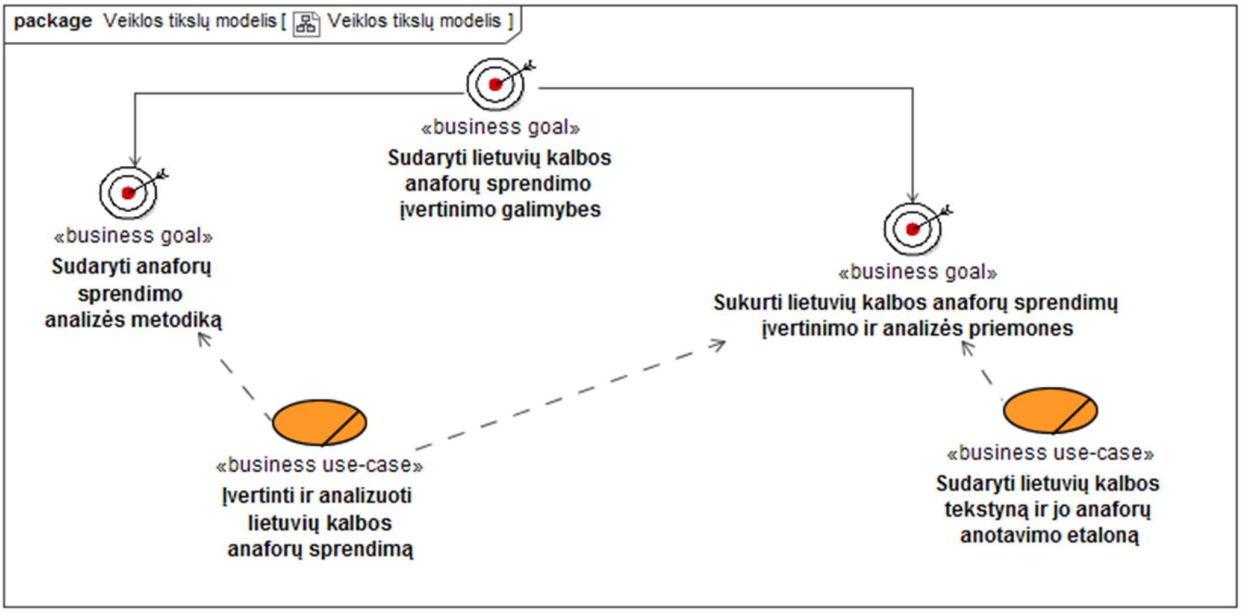
Kriterijus	***GATE***	***eHost***	***Anafora***
Koreferencijų anotavimo galimybė	+	+	+
Anaforų anotavimo galimybė	-	-	-
Įvertinimo rezultatų pateikimas matais R, P, F, RR	+	-	-
Bendros prieigos galimybė	+	-	+
Duomenų mainai *JSON* formatu	- /+ (*Twitter JSON*)	-	-
Skirtingų anotacijų grafinis palyginimas	+	-	-
Įrankio tobulinimo galimybė	+	+	+
Įmanoma išmokti naudotis įrankiu per 1 dieną	-	+	+

Iš nagrinėtų įrankių labiausiai pasirinktus palyginimo kriterijus atitiko *GATE*. Pagrindinė problema – *GATE* nėra pritaikytas detaliam anaforų žymėjimui, t. y. nesukūrus papildomų įskiepių, neįmanoma atskirai žymėti anaforos objektų ar pirmtakų. Galima žymėti tik koreferencijų grandinėles.

GATE yra sudėtingas įrankis, kurio pirminė paskirtis – sudėtingų kompleksinių sprendimų kūrimas. Todėl GATE patobulinimas, kad atitiktų visus išsikeltus kriterijus, galimai būtų sudėtingesnis sprendimas, nei sukurti naują nepriklausomą įrankį.

1.6. Siekiamas sprendimas

Tikslas. Sudaryti lietuvių kalbos anaforų sprendimo įvertinimo galimybes sukuriant anaforų sprendimo įvertinimo ir analizės metodiką realizuojantį įrankį. Grafiškai tikslas pavaizduotas 6 pav.



6 pav. Veiklos tikslų modelis

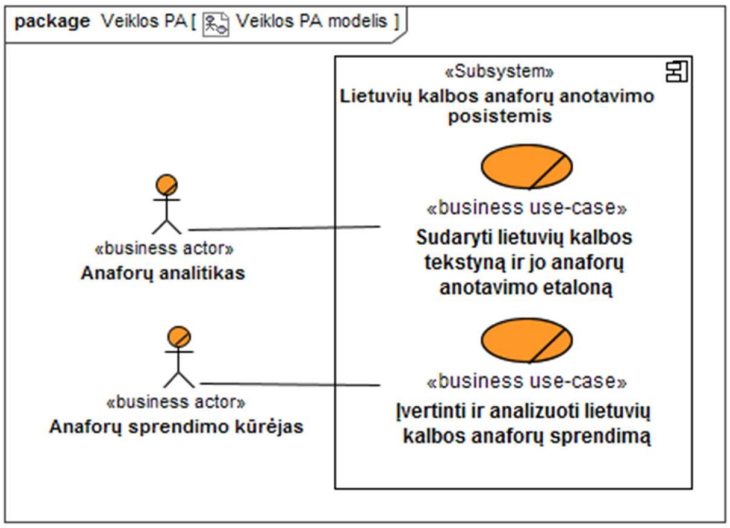
Šiam tikslui pasiekti, turi būti atlikti šie uždaviniai:

1. Atlikti anaforų sprendimo ir jo įvertinimo galimybių analizę;
2. Sudaryti įvertinimo ir analizės metodiką realizuojančio įrankio reikalavimų specifikaciją, projektą;
3. Sudaryti įrankio realizacijos projektą;
4. Realizuoti ir ištestuoti įrankį;
5. Atlikti eksperimentą, kuris patvirtintų sudarytas anaforų sprendimo įvertinimo galimybes;
6. Apibendrinti tyrimo rezultatus.

Siekiami privalumai. Sudaryta galimybė įvertinti anaforų sprendimą įrankiu, kuriuo ypač lengva išmokti naudotis ir kurio duomenų vientisumą užtikrintų taikoma debesų technologija.

Siekiama sukurti prototipinį įrankį ASAS, kuriuo būtų galima:

1. Sudaryti lietuvių kalbos tekstyną ir jo anaforų anotavimo etaloną;
2. Įvertinti ir analizuoti lietuvių kalbos anaforų sprendimą: eksportuoti JSON formatu tekstyno duomenis ir importuoti JSON formatu automatinio anaforų sprendimo rezultatus. Pagal sudarytą analizės metodiką pateikti grafinę klaidų analizę.



7 pav. Veiklos PA modelis

1.7. Analizės išvados

1. Kitoms kalboms naudojama anaforų sprendimo įvertinimo metodika gali būti naudojama ir lietuvių kalbos anaforų sprendimui vertinti;
2. Anaforų sprendimo įvertinimas atliekamas anaforų sprendimo rezultatus palyginant su etalonu ir matematiškai suskaičiuojant įvertinimo charakteristiką. Kadangi nėra anaforų sprendimo, sprendžiančio anaforas be klaidų net ir vyraujančioms pasaulio kalboms, toks etalonas turi būti sudaromas žmogaus, naudojantis kompiuterizuotu įrankiu;
3. Kadangi anaforų tipų klasifikacija kiekvienoje kalboje yra specifinė, įrankis, realizuojantis įvertinimo metodiką, negali būti universalus tarp kalbų ir turi būti adaptuotas lietuvių kalbai;
4. Kitos kalbos įrankio adaptavimas dirbti su lietuvių kalba galimai užimtų daugiau laiko nei naujo nepriklausomo įrankio sukūrimas, todėl geriau kurti atskirą nepriklausomą įrankį ir taip sudaryti lietuvių kalbos anaforų sprendimo įvertinimo galimybes.

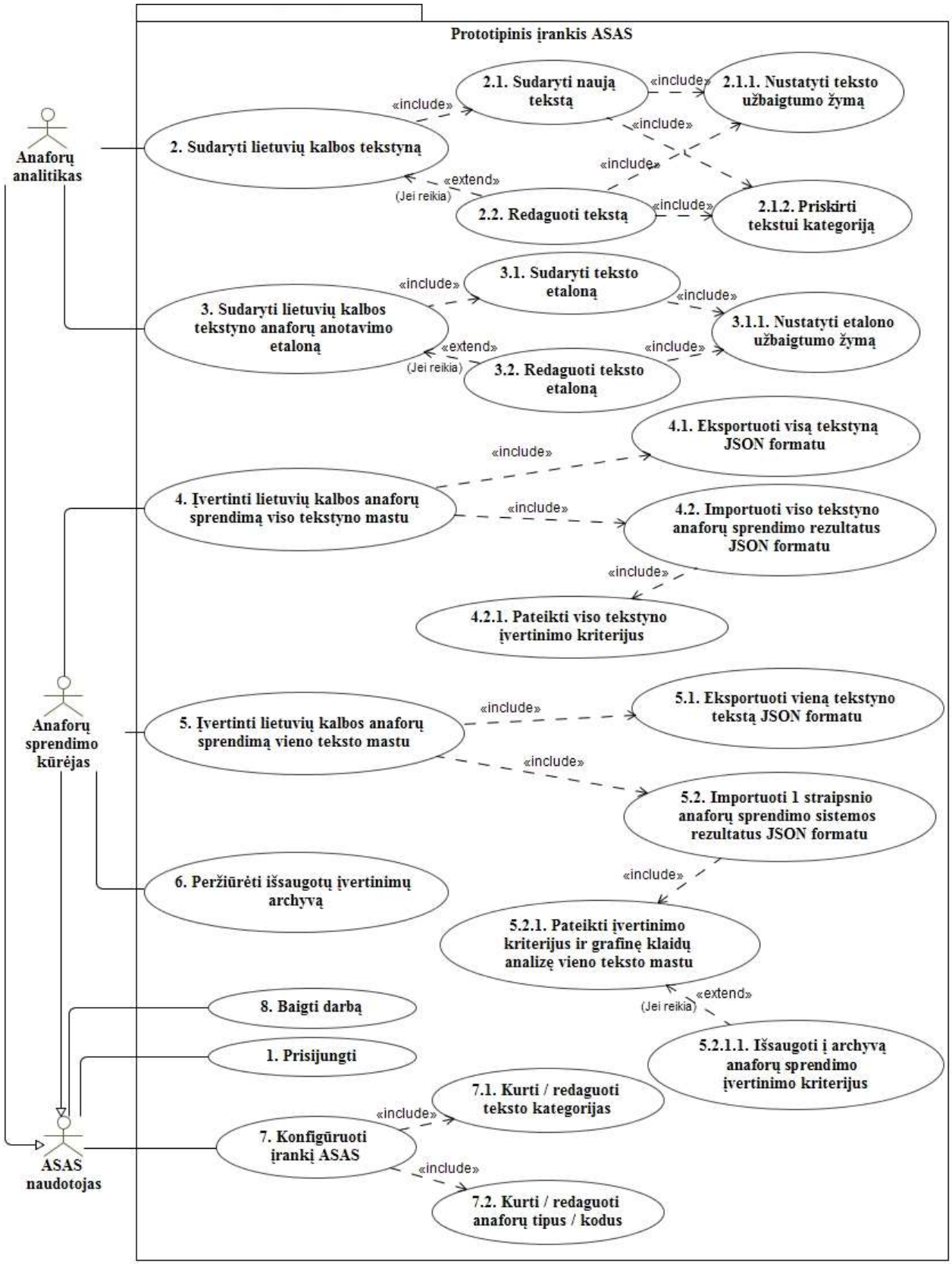
2. ĮRANKIO ASAS REIKALAVIMŲ SPECIFIKACIJA IR PROJEKTAS

Šiame skyriuje bus pateikti prototipinio įrankio ASAS funkciniai reikalavimai (panaudojimo atvejai bei jų specifikacijos ir veiksmų diagramos) ir nefunkciniai reikalavimai, dalykinės srities modelis ir įvertinimo bei grafinės analizės algoritmo modelis.

2.1. Reikalavimų specifikacija

2.1.1. Panaudojimo atvejai, jų specifikacijos ir veiklos diagramos

Pateikta kompiuterizuojamų panaudojimo atvejų diagrama (2 pav.). Joje pavaizduotos įrankio ASAS naudotojų rolės: anaforų analitikas ir anaforų sprendimo kūrėjas. Lentelėse pateiktos panaudojimo atvejų specifikacijos, o paveikslėliuose – scenarijų modeliai.

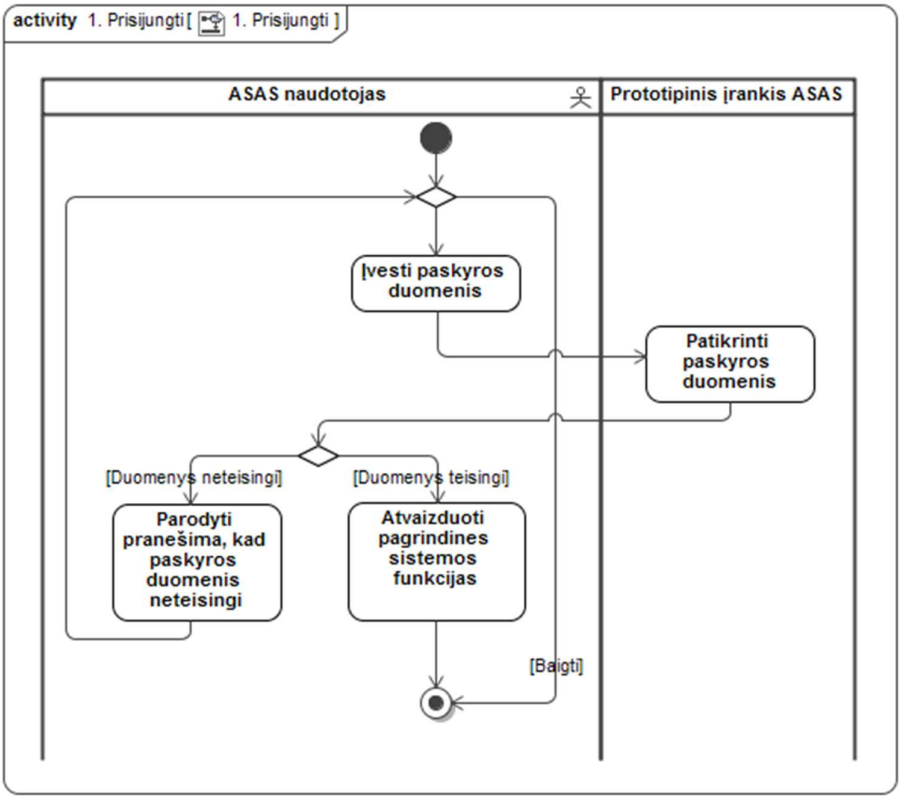


8 pav. Prototipinio įrankio ASAS panaudojimo atvejų modelis

3 lentelėje aprašyta panaudojimo atvejo „1. Prisijungti“ specifikacija, o 9 pav. pateiktas jo scenarijaus modelis.

3 lentelė. PA „1. Prisijungti“ specifikacija

Panaudojimo atvejis „1. Prisijungti“.		
Tikslas. Sistemos naudotojui prisijungti prie sistemos ir prieiti prie jos funkcijų.		
Išankstinė sąlyga.		Sistemos naudotojas paleidęs sistemą.
Aktorius.		ASAS naudotojas.
Sužadinimo sąlyga.		Naudotojas pasirenka sistemos paleidimo nuorodą.
Susiję panaudojimo atvejai	**Išplečia PA**	–
	Apima PA	–
	Specializuoja PA	–
Pagrindinis įvykių srautas		**Sistemos reakcija ir sprendimai**
Naudotojas įveda savo paskyros duomenis: naudotojo vardą ir slaptažodį.		Sistema patikrina prisijungimo duomenis. Jeigu prisijungimo duomenys teisingi, sistema atvaizduoja ekrane pagrindinių sistemos funkcijų langą. Jei prisijungimo duomenys neteisingi, sistema pateikia pranešimą, kad prieiga prie pagrindinių programos funkcijų negalima.

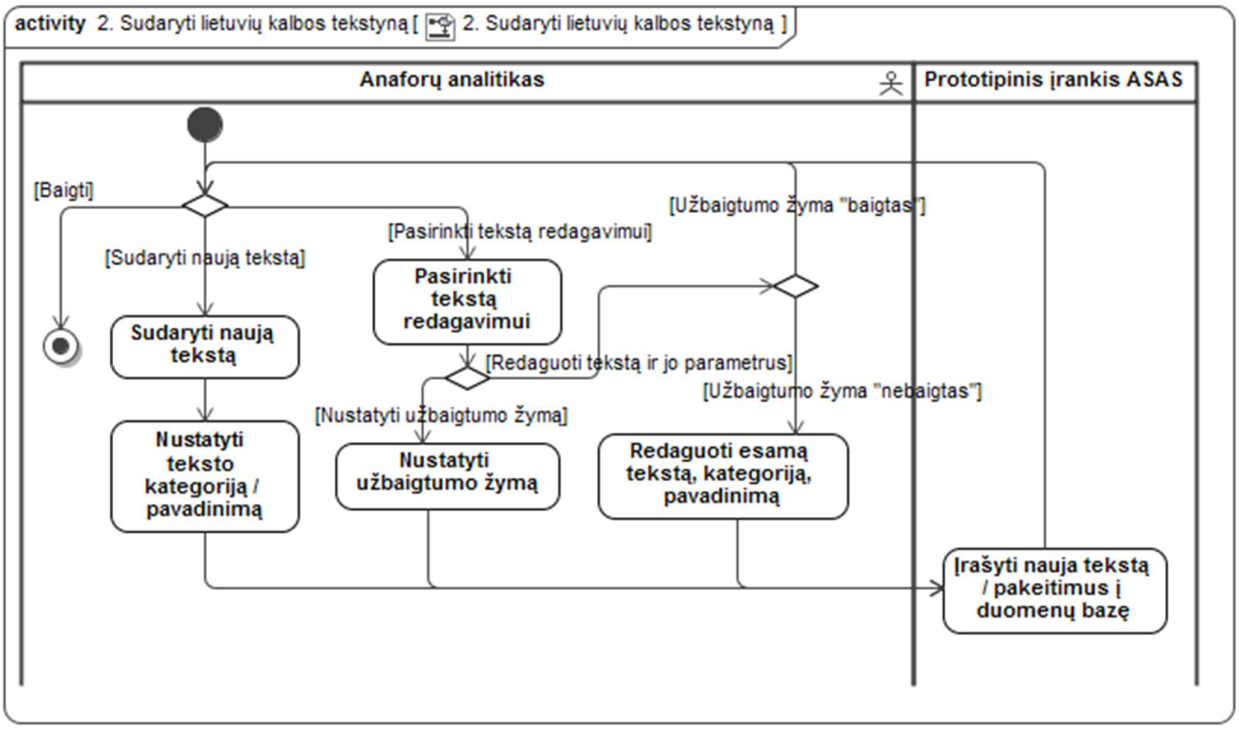


9 pav. PA „1. Prisijungti“ scenarijaus modelis

4 lentelėje aprašyta panaudojimo atvejo „2. Sudaryti lietuvių kalbos tekstyną“ specifikacija, o 10 pav. pateiktas jo scenarijaus modelis.

4 lentelė. PA „2. Sudaryti lietuvių kalbos tekstyną“ specifikacija

<table>
<tr><td colspan="3">Panaudojimo atvejis „2. Sudaryti lietuvių kalbos tekstyną“.</td></tr>
<tr><td colspan="3">Tikslas. Sudaryti lietuvių kalbos tekstyną, kuris bus naudojamas anaforų sprendimams vertinti, sudarius jo anaforų anotavimo etaloną.</td></tr>
<tr><td colspan="2">Išankstinė sąlyga.</td><td>Sistemos naudotojas prisijungęs prie sistemos.</td></tr>
<tr><td colspan="2">Aktorius.</td><td>Anaforų analitikas.</td></tr>
<tr><td colspan="2">Sužadinimo sąlyga.</td><td>Sistemos naudotojas pasirenka „Naujas tekstas“.</td></tr>
<tr><td rowspan="3">Susiję panaudojimo atvejai</td><td>Išplečia PA</td><td>–</td></tr>
<tr><td>Apima PA</td><td>–</td></tr>
<tr><td>Specializuoja PA</td><td>–</td></tr>
<tr><td colspan="2">Pagrindinis įvykių srautas</td><td>Sistemos reakcija ir sprendimai</td></tr>
<tr><td colspan="2">Panaudojimo atvejis „2.1. Sudaryti naują tekstą“.</td><td></td></tr>
<tr><td colspan="3">Alternatyvūs scenarijai</td></tr>
<tr><td colspan="2">Panaudojimo atvejis „2.2. Redaguoti tekstyno tekstą“.</td><td></td></tr>
</table>

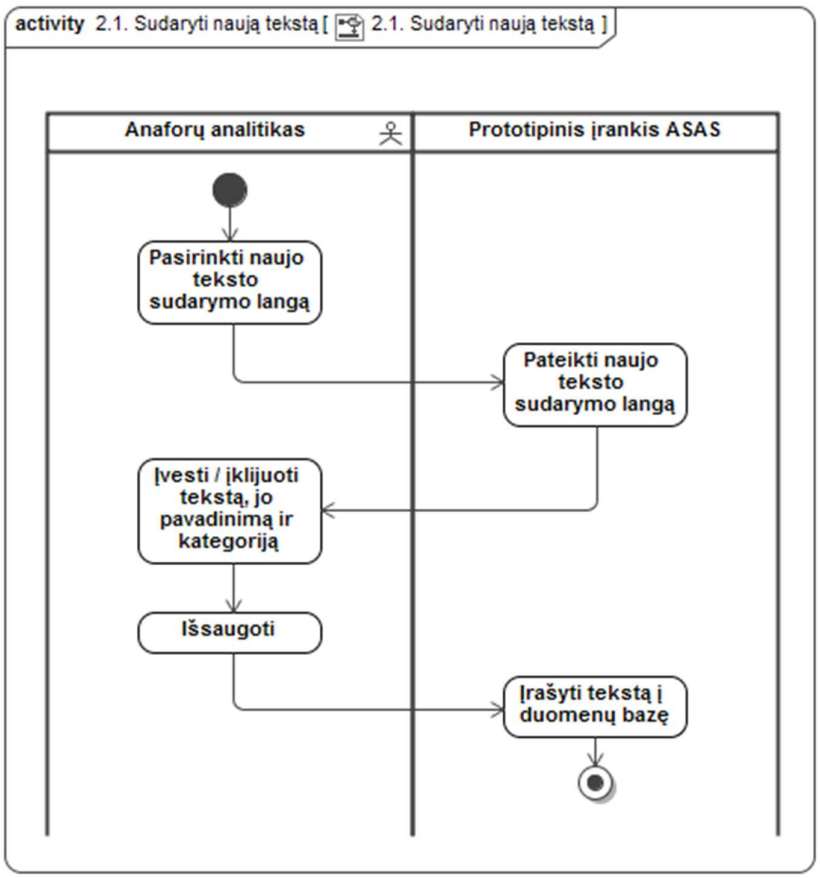


10 pav. PA „2. Sudaryti lietuvių kalbos tekstyną“ scenarijus

5 lentelėje aprašyta panaudojimo atvejo „2.1. Sudaryti naują tekstą“ specifikacija, o 11 pav. pateiktas jo scenarijaus modelis.

5 lentelė. PA „2.1. Sudaryti naują tekstą“ specifikacija

<table>
<tr><td colspan="3">Panaudojimo atvejis „2.1. Sudaryti naują tekstą“.</td></tr>
<tr><td colspan="3">Tikslas. Sudaryti naują tekstą (papildyti tekstyną nauju tekstu).</td></tr>
<tr><td colspan="2">Išankstinė sąlyga.</td><td>Sistemos naudotojas prisijungęs prie sistemos.</td></tr>
<tr><td colspan="2">Aktorius.</td><td>Anaforų analitikas.</td></tr>
<tr><td colspan="2">Sužadinimo sąlyga.</td><td>Naudotojas pasirenka sudaryti naują tekstą.</td></tr>
<tr><td rowspan="3">Susiję panaudojimo atvejai</td><td>Išplečia PA</td><td>„2. Sudaryti lietuvių kalbos tekstyną“</td></tr>
<tr><td>Apima PA</td><td>–</td></tr>
<tr><td>Specializuoja PA</td><td>–</td></tr>
<tr><td colspan="2">Pagrindinis įvykių srautas</td><td>Sistemos reakcija ir sprendimai</td></tr>
<tr><td colspan="2"></td><td>Sistema atvaizduoja ekrane naujo teksto sudarymo langą.</td></tr>
<tr><td colspan="2">Sistemos naudotojas įveda tekstą, įrašo pavadinimą, priskiria kategoriją ir pasirenka „išsaugoti“.</td><td>Tekstas yra įrašomas į duomenų bazę ir priskiriamas tekstynui.</td></tr>
</table>

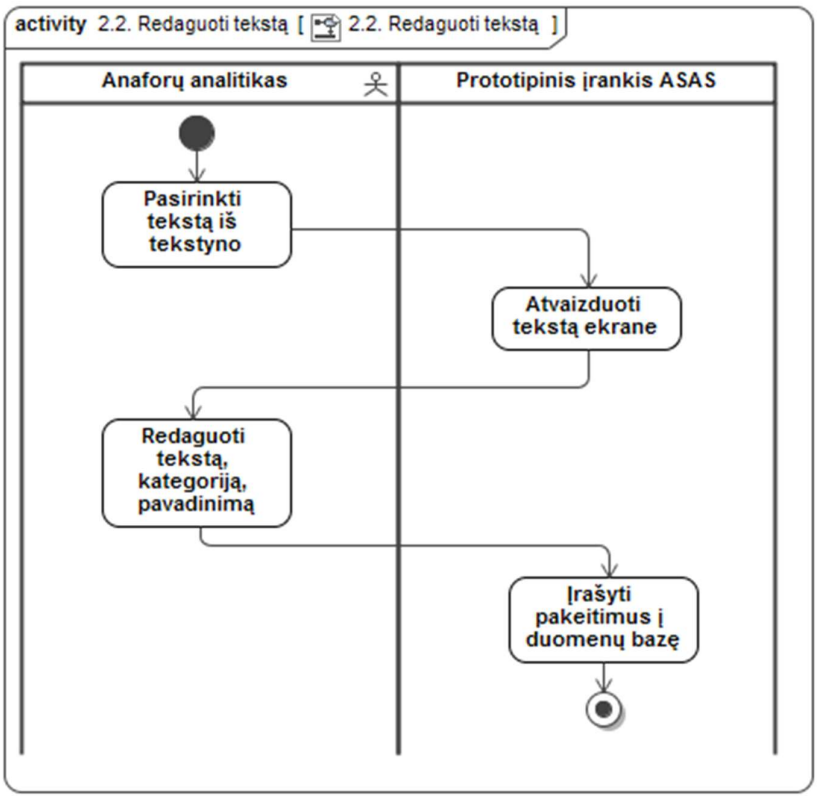


11 pav. PA „2.1. Sudaryti naują tekstą“ scenarijus

6 lentelėje aprašyta panaudojimo atvejo „2.2. Redaguoti tekstą“ specifikacija, o 12 pav. pateiktas jo scenarijaus modelis.

6 lentelė. PA „2.2. Redaguoti tekstą“ specifikacija

Panaudojimo atvejis „2.2. Redaguoti tekstą“.		
Tikslas. Redaguoti jau sudarytą tekstą tekstyne.		
Išankstinė sąlyga.		Sistemos naudotojas prisijungęs prie sistemos.
Aktorius.		Anaforų analitikas.
Sužadinimo sąlyga.		Naudotojas pasirenka „Įkelti tekstą“.
Susiję panaudojimo atvejai	**Išplečia PA**	„2. Sudaryti lietuvių kalbos tekstyną“
	Apima PA	–
	Specializuoja PA	–
Pagrindinis įvykių srautas		**Sistemos reakcija ir sprendimai**
		Sistema atvaizduoja ekrane tekstus, iš kurių susidaro tekstynas.
Sistemos naudotojas pasirenka norimą redaguoti tekstą su žyma „nebaigtas“.		Sistema atvaizduoja ekrane pasirinktą tekstą. Tekstas gali būti redaguojamas, keičiama jo kategorija ir / ar pavadinimas.
Sistemos naudotojas atlieka reikiamą korekciją ir pasirenka „išsaugoti pakeitimus“ (jei tekstas nėra su žyma „baigtas“).		Jei tekstas buvo redaguotas, įrašomi pakeitimai duomenų bazėje.

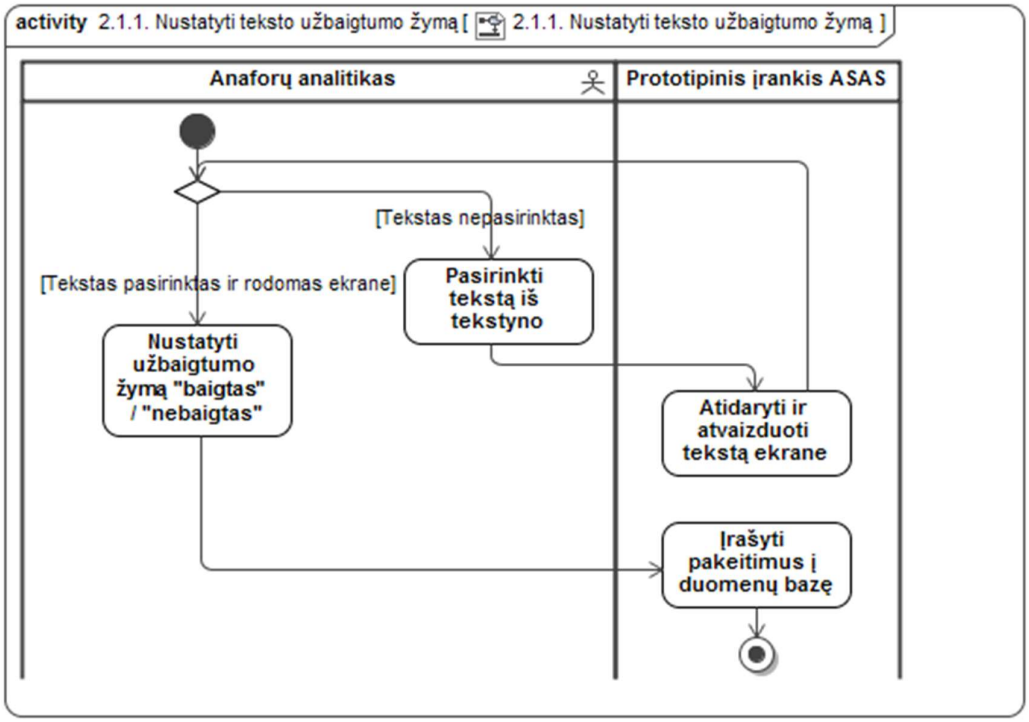


12 pav. PA „2.2. Redaguoti tekstą“ scenarijus

7 lentelėje aprašyta panaudojimo atvejo „2.1.1. Nustatyti teksto užbaigtumo žymą“ specifikacija, o 13 pav. pateiktas jo scenarijaus modelis.

7 lentelė. PA „2.1.1. Nustatyti teksto užbaigtumo žymą“ specifikacija

Panaudojimo atvejis „2.1.1. Nustatyti teksto užbaigtumo žymą“.		
Tikslas. Nustatyti tekstui žymą „baigtas“ arba „nebaigtas“.		
Išankstinė sąlyga.		Sistemos naudotojas prisijungęs prie sistemos. Atidarytas teksto redagavimo langas arba įvestas ir išsaugotas naujas tekstas.
Aktorius.		Anaforų analitikas.
Sužadinimo sąlyga.		Naudotojas pasirenka nustatyti žymą „baigtas“ arba „nebaigtas“.
Susiję panaudojimo atvejai	**Išplečia PA**	–
	Apima PA	„2.1. Sudaryti naują tekstą“ / „2.2. Redaguoti tekstą“
	Specializuoja PA	–
Pagrindinis įvykių srautas		**Sistemos reakcija ir sprendimai**
		Sistema nustato tekstui žymą „baigtas“ arba „nebaigtas“.

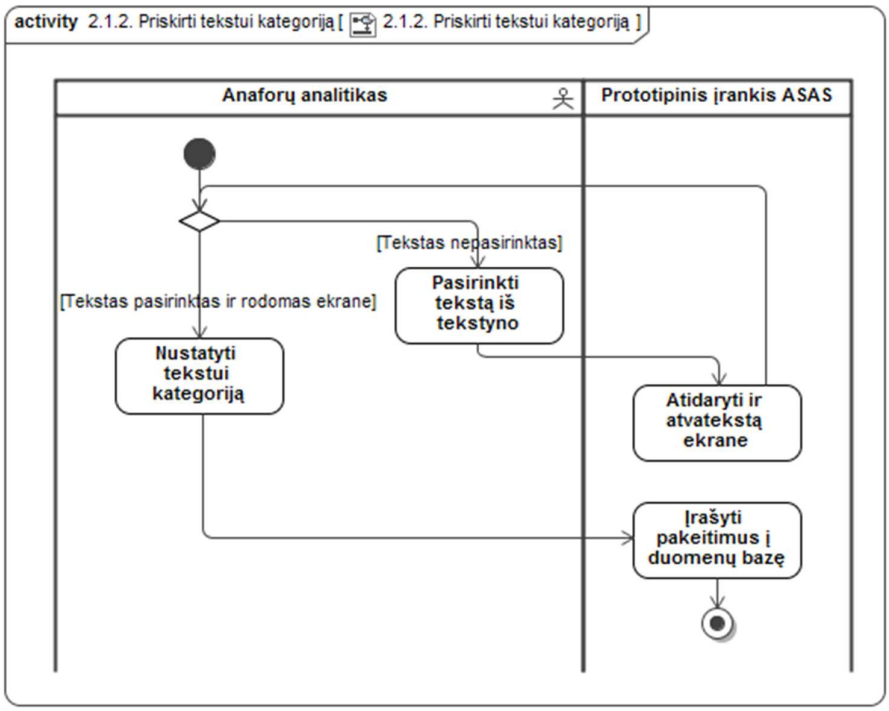


13 pav. PA „2.1.1. Nustatyti teksto užbaigtumo žymą“ scenarijus

8 lentelėje aprašyta panaudojimo atvejo „2.1.2. Priskirti tekstui kategoriją“ specifikacija, o 14 pav. pateiktas jo scenarijaus modelis.

8 lentelė. PA „2.1.2. Priskirti tekstui kategoriją“ specifikacija

Panaudojimo atvejis „2.1.2. Priskirti tekstui kategoriją“.		
Tikslas. Nustatyto tekstui kategoriją.		
Išankstinė sąlyga.		Sistemos naudotojas prisijungęs prie sistemos. Atidarytas teksto redagavimo langas arba įvestas ir išsaugotas naujas tekstas.
Aktorius.		Anaforų analitikas.
Sužadinimo sąlyga.		Naudotojas pasirenka teksto kategorijos išsiskleidžianti sąrašą, kuriame pateikiamos pasirinkimui galimos teksto kategorijos.
Susiję panaudojimo atvejai	**Išplečia PA**	–
	Apima PA	„2.1. Sudaryti naują tekstą“ / „2.2. Redaguoti tekstą“
	Specializuoja PA	–
Pagrindinis įvykių srautas		**Sistemos reakcija ir sprendimai**
Sistemos naudotojas pasirenka reikiamą kategoriją.		Sistema priskiria tekstui kategoriją.
Sistemos naudotojas pasirenka „išsaugoti“.		Sistema įrašo parinktą teksto kategoriją į duomenų bazę.

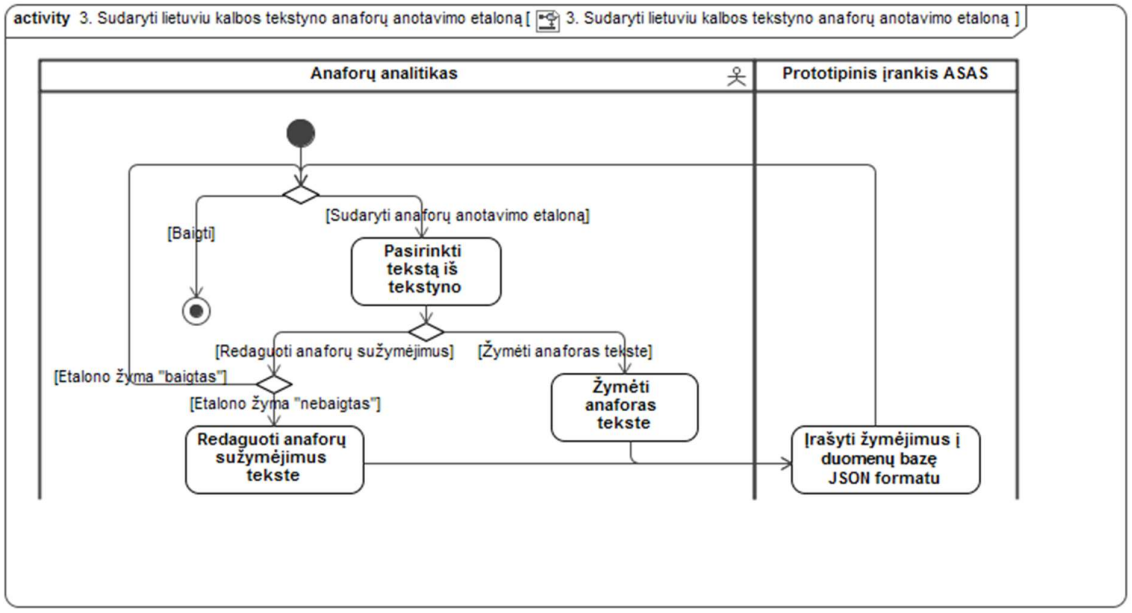


14 pav. PA „2.1.2. Priskirti tekstui kategoriją“ scenarijus

9 lentelėje aprašyta panaudojimo atvejo „3. Sudaryti lietuvių kalbos tekstyno anaforų anotavimo etaloną“ specifikacija, o 15 pav. pateiktas jo scenarijaus modelis.

9 lentelė. PA „3. Sudaryti lietuvių kalbos tekstyno anaforų anotavimo etaloną“ specifikacija

<table>
<tr><td colspan="3">Panaudojimo atvejis „3. Sudaryti lietuvių kalbos tekstyno anaforų anotavimo etaloną“.</td></tr>
<tr><td colspan="3">Tikslas. Sudaryti lietuvių kalbos tekstyno anaforų anotavimo etaloną, kuris bus naudojamas anaforų sprendimui vertinti.</td></tr>
<tr><td colspan="2">Išankstinė sąlyga.</td><td>Sistemos naudotojas prisijungęs prie sistemos. Ekrane įkeltas tekstas iš tekstyno su žyma „baigtas“.</td></tr>
<tr><td colspan="2">Aktorius.</td><td>Anaforų analitikas.</td></tr>
<tr><td colspan="2">Sužadinimo sąlyga.</td><td>Sistemos naudotojas pasirenka „Sukurti etaloną“.</td></tr>
<tr><td rowspan="3">Susiję panaudojimo atvejai</td><td>Išplečia PA</td><td>–</td></tr>
<tr><td>Apima PA</td><td>–</td></tr>
<tr><td>Specializuoja PA</td><td>–</td></tr>
<tr><td colspan="2">Pagrindinis įvykių srautas</td><td>Sistemos reakcija ir sprendimai</td></tr>
<tr><td colspan="2">„3.1. Sudaryti teksto etaloną“</td><td></td></tr>
<tr><td colspan="3">Alternatyvūs scenarijai</td></tr>
<tr><td colspan="2">Sistemos naudotojas pasirenka „Įkelti etaloną“.</td><td>Sistema atvaizduoja ekrane tekstą su sužymėtomis anaforomis.</td></tr>
<tr><td colspan="2">Sistemos naudotojas sukuria naujas anaforas arba ištrina senas ir pasirenka „Išsaugoti“.</td><td>Sistema įrašo anaforų sužymėjimą į duomenų bazę.</td></tr>
</table>

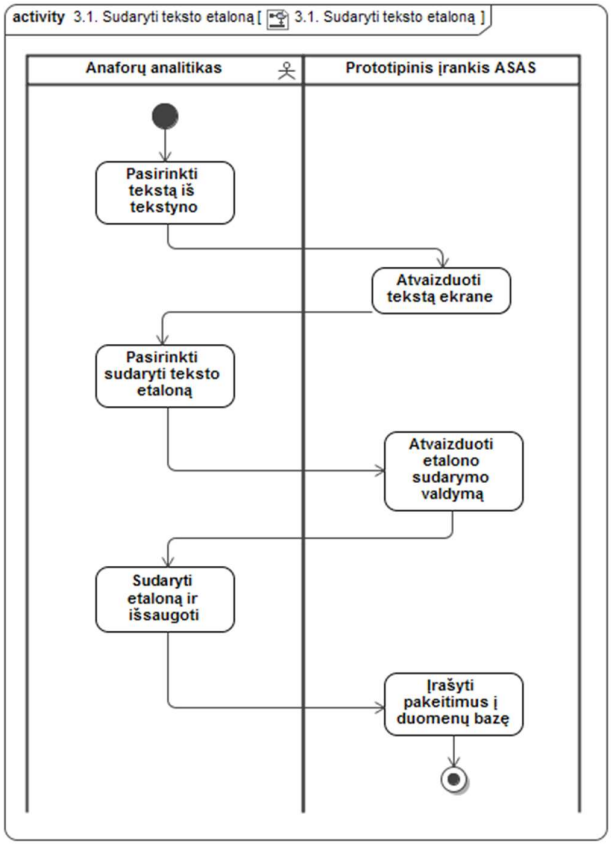


15 pav. PA „Sudaryti lietuvių kalbos tekstyno anaforų anotavimo etaloną“ scenarijus

10 lentelėje aprašyta panaudojimo atvejo „3.1. Sudaryti teksto etaloną“ specifikacija, o 16 pav. pateiktas jo scenarijaus modelis.

10 lentelė. PA „3.1. Sudaryti teksto etaloną“ specifikacija

Panaudojimo atvejis „3.1. Sudaryti teksto etaloną“.		
Tikslas. Sudaryti vieno teksto iš tekstyno etaloną.		
Išankstinė sąlyga.		Sistemos naudotojas prisijungęs prie sistemos. Ekrane įkeltas tekstas iš tekstyno su žyma „baigtas“.
Aktorius.		Anaforų analitikas.
Sužadinimo sąlyga.		Sistemos naudotojas pasirenka „Sukurti etaloną“.
Susiję panaudojimo atvejai	**Išplečia PA**	–
	Apima PA	„3. Sudaryti lietuvių kalbos tekstyno anaforų anotavimo etaloną“.
	Specializuoja PA	–
Pagrindinis įvykių srautas		**Sistemos reakcija ir sprendimai**
		Sistema atvaizduoja ekrane teksto anaforų anotavimo etalono sudarymo langą.
Sistemos naudotojas sužymi anaforas ir pasirenka „Išsaugoti etaloną“.		Etaloninis anaforų sužymėjimas yra įrašomas į duomenų bazę.

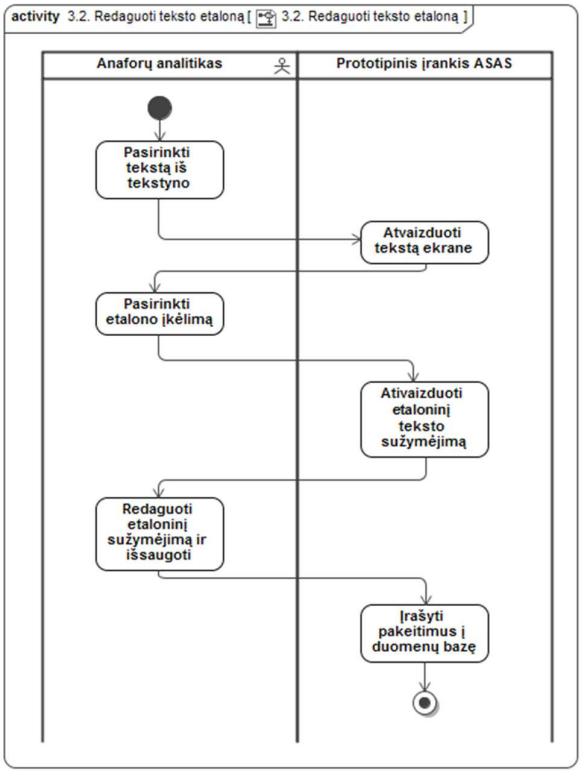


16 pav. PA „3.1. Sudaryti teksto etaloną“ scenarijus

11 lentelėje aprašyta panaudojimo atvejo „3.2. Redaguoti teksto etaloną“ specifikacija, o 17 pav. pateiktas jo scenarijaus modelis.

11 lentelė. PA „3.2. Redaguoti teksto etaloną“ specifikacija

<table>
<tr><td colspan="3">Panaudojimo atvejis „3.2. Redaguoti teksto etaloną“.</td></tr>
<tr><td colspan="3">Tikslas. Redaguoti vieno teksto iš tekstyno etaloną.</td></tr>
<tr><td colspan="2">Išankstinė sąlyga.</td><td>Sistemos naudotojas prisijungęs prie sistemos. Ekrane įkeltas tekstas iš tekstyno ir įkeltas jo etalonas su žyma „nebaigtas“.</td></tr>
<tr><td colspan="2">Aktorius.</td><td>Anaforų analitikas.</td></tr>
<tr><td colspan="2">Sužadinimo sąlyga.</td><td>Sistemos naudotojas pasirenka „Sukurti etaloną“.</td></tr>
<tr><td rowspan="3">Susiję panaudojimo atvejai</td><td>Išplečia PA</td><td>„3. Sudaryti lietuvių kalbos tekstyno anaforų anotavimo etaloną“.</td></tr>
<tr><td>Apima PA</td><td>–</td></tr>
<tr><td>Specializuoja PA</td><td>–</td></tr>
<tr><td colspan="2">Pagrindinis įvykių srautas</td><td>Sistemos reakcija ir sprendimai</td></tr>
<tr><td colspan="2"></td><td>Sistema atvaizduoja ekrane teksto anaforų anotavimo etalono sudarymo langą.</td></tr>
<tr><td colspan="2">Sistemos naudotojas sužymi anaforas ir parenka „Išsaugoti etaloną“.</td><td>Etaloninis anaforų sužymėjimas yra įrašomas į duomenų bazę.</td></tr>
</table>

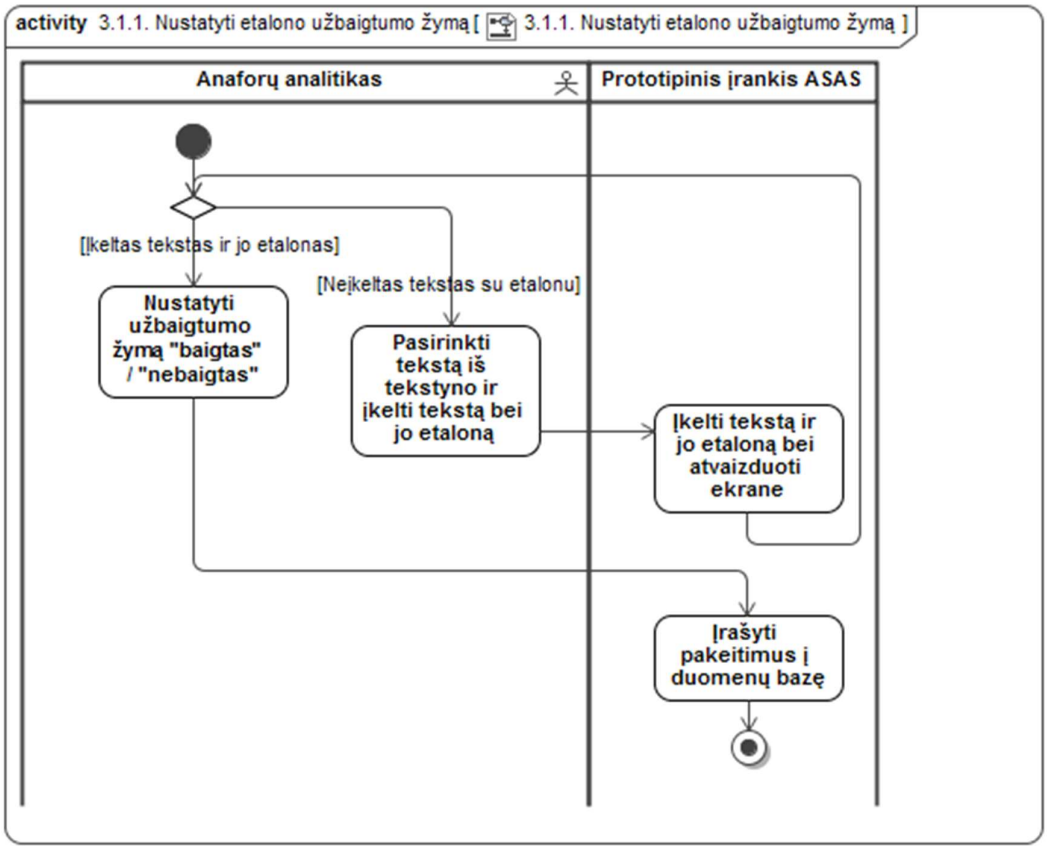


17 pav. PA „Redaguoti teksto etaloną“ scenarijus

12 lentelėje aprašyta panaudojimo atvejo „3.1.1. Nustatyti etalono užbaigtumo žymą“ specifikacija, o 18 pav. pateiktas jo scenarijaus modelis.

12 lentelė. PA „3.1.1. Nustatyti etalono užbaigtumo žymą“ specifikacija

Panaudojimo atvejis „3.1.1. Nustatyti etalono užbaigtumo žymą“.		
Tikslas. Nustatyti vieno teksto iš tekstyno etalonui žymą „baigtas“ arba „nebaigtas“.		
Išankstinė sąlyga.		Sistemos naudotojas prisijungęs prie sistemos. Ekrane įkeltas tekstas iš tekstyno ir įkeltas jo etalonas.
Aktorius.		Anaforų analitikas.
Sužadinimo sąlyga.		Naudotojas pasirenka nustatyti teksto etalonui žymą „baigtas“ arba „nebaigtas“.
Susiję panaudojimo atvejai	**Išplečia PA**	–
	Apima PA	„3.1. Sudaryti teksto etaloną“ / „3.2. Redaguoti teksto etaloną“.
	Specializuoja PA	–
Pagrindinis įvykių srautas		**Sistemos reakcija ir sprendimai**
		Sistema nustato teksto etalonui žymą „baigtas“ arba „nebaigtas“.

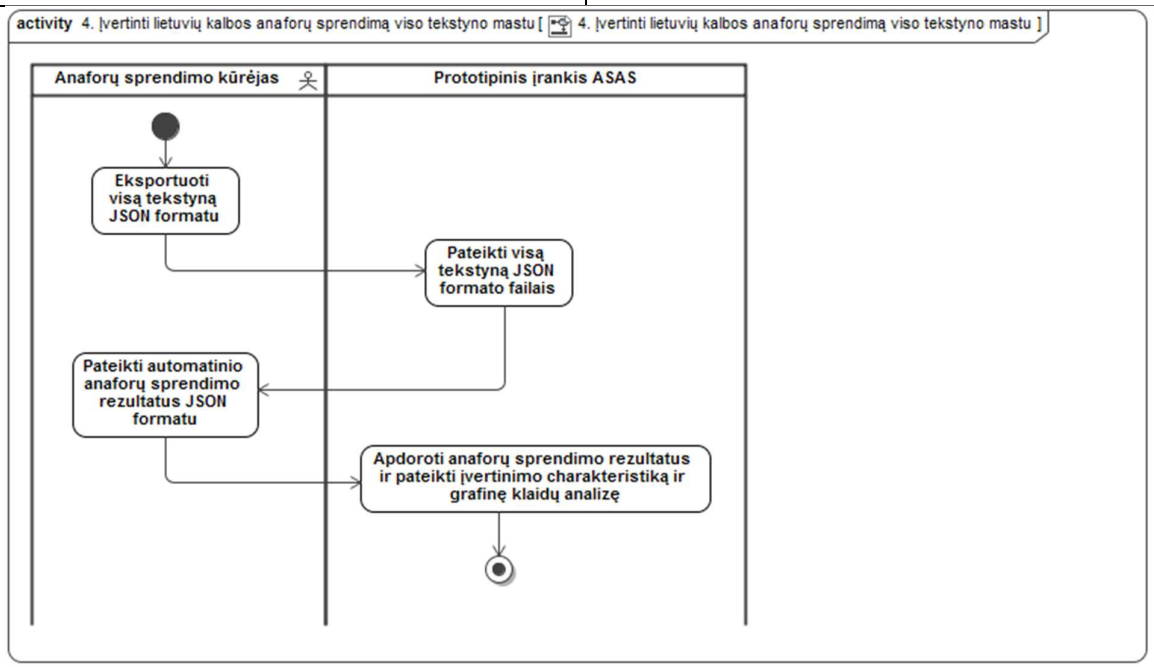


18 pav. PA „3.1.1. Nustatyti etalono užbaigtumo žymą“ scenarijus

13 lentelėje aprašyta panaudojimo atvejo „4. Įvertinti lietuvių kalbos anaforų sprendimą viso tekstyno mastu“ specifikacija, o 19 pav. pateiktas jo scenarijaus modelis.

13 lentelė. PA „4. Įvertinti lietuvių kalbos anaforų sprendimą viso tekstyno mastu“ specifikacija

Panaudojimo atvejis „4. Įvertinti lietuvių kalbos anaforų sprendimą viso tekstyno mastu“.		
Tikslas. Įvertinti automatinio anaforų sprendimo rezultatus ir gauti įvertinimo kriterijus viso tekstyno mastu.		
Išankstinė sąlyga.		Sistemos naudotojas prisijungęs prie sistemos ir pasirinkęs „Įkelti tekstą“ langą.
Aktorius.		Anaforų sprendimo kūrėjas.
Sužadinimo sąlyga.		Sistemos naudotojas pasirenka „Tekstyno vertinimas“.
Susiję panaudojimo atvejai	**Išplečia PA**	–
	Apima PA	–
	Specializuoja PA	–
Pagrindinis įvykių srautas		**Sistemos reakcija ir sprendimai**
		Atvaizduoja viso tekstyno vertinimo langą.
Pasirenka „Importuoti sprendimo anotaciją JSON“.		Atvaizduoja viso tekstyno sprendimo anotacijų JSON formato failų parinkimo langą.
Parenkami viso tekstyno sprendimo anotacijų JSON formato failai.		Atvaizduoja anaforų sprendimo įvertinimą viso tekstyno mastu. Atvaizduojami anaforų sprendimo įvertinimo kriterijai dydžiais R, P, F-vertė, RR ir tarpinės kriterijų apskaičiavimo reikšmės: T, F, C, E.
Alternatyvūs scenarijai		
Parenkami ne visi ar netinkami tekstyno sprendimo anotacijų JSON formato failai.		Parodomas klaidos pranešimas.

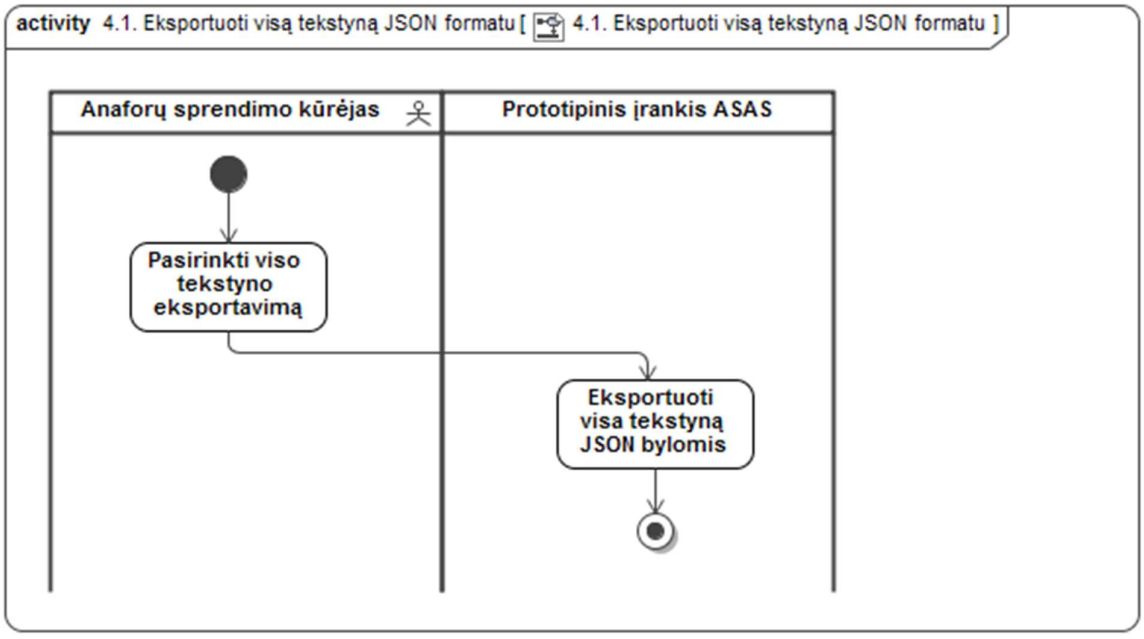


19 pav. PA „4. Įvertinti lietuvių kalbos anaforų sprendimą viso tekstyno mastu“ scenarijus

14 lentelėje aprašyta panaudojimo atvejo „4.1. Eksportuoti visą tekstyną JSON formatu“ specifikacija, o 20 pav. pateiktas jo scenarijaus modelis.

14 lentelė. PA „4.1. Eksportuoti visą tekstyną JSON formatu“ specifikacija

<table>
<tr><td colspan="3">Panaudojimo atvejis „4.1. Eksportuoti visą tekstyną JSON formatu“.</td></tr>
<tr><td colspan="3">Tikslas. Eksportuoti visą tekstyną JSON formatu, t. y. visų tekstų, sudarančių tekstyną, failus.</td></tr>
<tr><td colspan="2">Išankstinė sąlyga.</td><td>Sistemos naudotojas prisijungęs prie sistemos ir pasirinkęs „Įkelti tekstą“ langą.</td></tr>
<tr><td colspan="2">Aktorius.</td><td>Anaforų sprendimo kūrėjas.</td></tr>
<tr><td colspan="2">Sužadinimo sąlyga.</td><td>Sistemos naudotojas pasirenka „Eksportuoti tekstyną JSON“.</td></tr>
<tr><td rowspan="3">Susiję panaudojimo atvejai</td><td>Išplečia PA</td><td>–</td></tr>
<tr><td>Apima PA</td><td>„4. Įvertinti lietuvių kalbos anaforų sprendimą viso tekstyno mastu“.</td></tr>
<tr><td>Specializuoja PA</td><td>–</td></tr>
<tr><td colspan="2">Pagrindinis įvykių srautas</td><td>Sistemos reakcija ir sprendimai</td></tr>
<tr><td colspan="2"></td><td>Parodomas pranešimas, kiek bylų viso bus eksportuota, ir atidaromas langas parinkti kelią, kur įrašyti bylas.</td></tr>
<tr><td colspan="2">Sistemos naudotojas nurodo katalogą, kur įrašyti bylas.</td><td>Įrašomos bylos į nurodytą vietą.</td></tr>
</table>

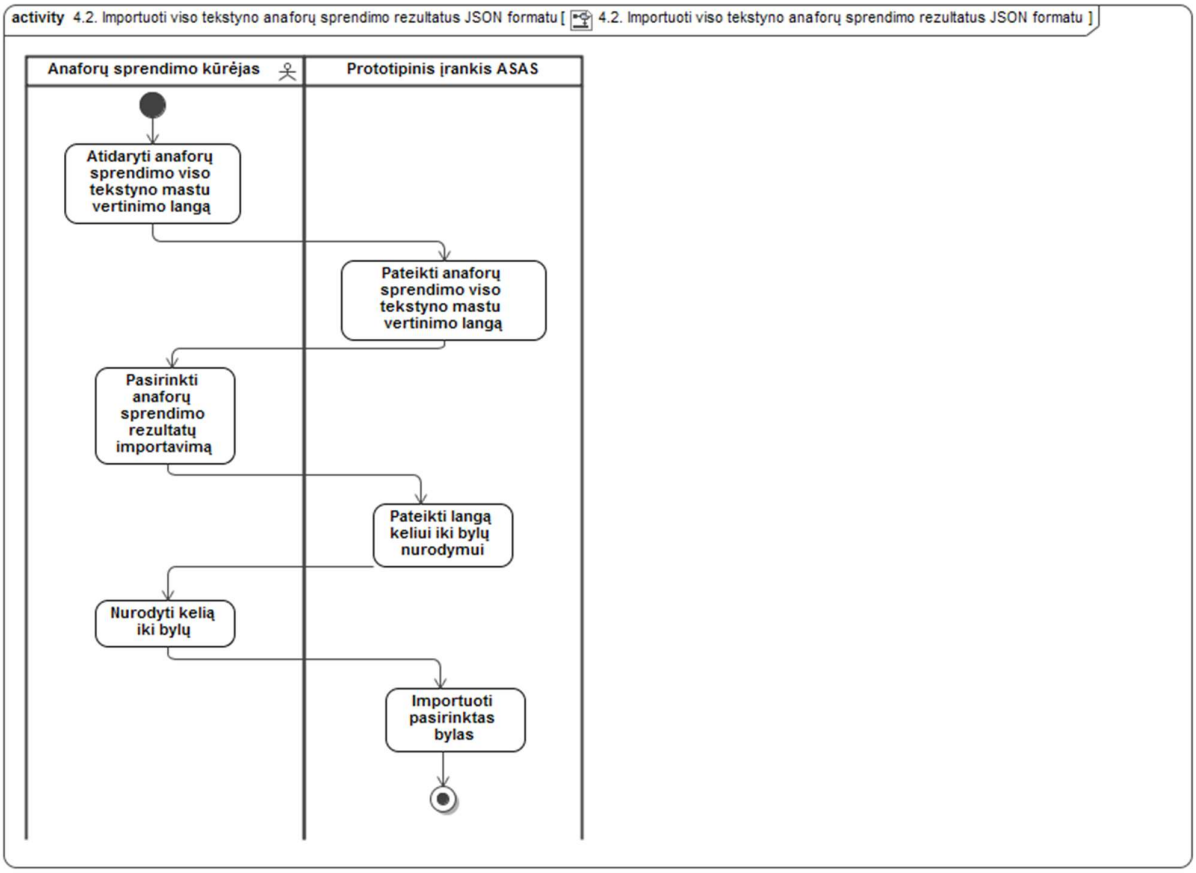


20 pav. PA „4.1. Eksportuoti visą tekstyną JSON formatu“ scenarijus

15 lentelėje aprašyta panaudojimo atvejo „4.2. Importuoti viso tekstyno anaforų sprendimo rezultatus JSON formatu“ specifikacija, o 21 pav. pateiktas jo scenarijaus modelis.

15 lentelė. PA „4.2. Importuoti viso tekstyno anaforų sprendimo rezultatus JSON formatu“ specifikacija

Panaudojimo atvejis „4.2. Importuoti viso tekstyno anaforų sprendimo rezultatus JSON formatu“.		
Tikslas. Importuoti viso tekstyno anaforų sprendimo rezultatus JSON formatu, t. y. visų tekstų, sudarančių tekstyną, rezultatų failus.		
Išankstinė sąlyga.		Sistemos naudotojas prisijungęs prie sistemos. Įjungtas viso tekstyno vertinimo langas.
Aktorius.		Anaforų sprendimo kūrėjas.
Sužadinimo sąlyga.		Sistemos naudotojas pasirenka „Importuoti sprendimo anotaciją JSON“.
Susiję panaudojimo atvejai	**Išplečia PA**	–
	Apima PA	„4. Įvertinti lietuvių kalbos anaforų sprendimą viso tekstyno mastu“.
	Specializuoja PA	–
Pagrindinis įvykių srautas		**Sistemos reakcija ir sprendimai**
		Atvaizduoja ekrane langą, kuriame reikia nurodyti kelią iki anaforų sprendimo bylų su rezultatais.
Sistemos naudotojas parenka JSON formato bylas (nurodo kelią iki jų).		Atvaizduojami anaforų sprendimo įvertinimo kriterijai dydžiais R, P, F-vertė, RR ir tarpinės kriterijų apskaičiavimo reikšmės: T, F, C, E.
Alternatyvūs scenarijai		
Parenkamos netinkamos JSON formato bylos, t. y. parenkama bent viena ne JSON formato byla arba kito teksto (pagal ID) anaforų anotacijos byla.		Parodomas klaidos pranešimas.

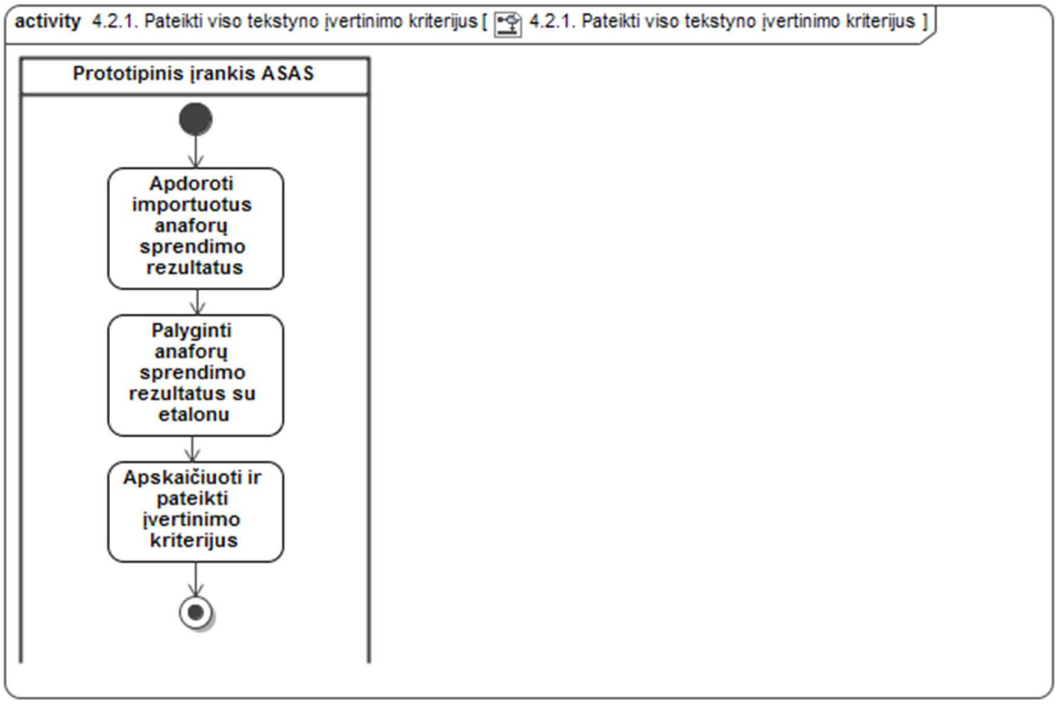


21 pav. PA „4.2. Importuoti viso tekstyno anaforų sprendimo rezultatus JSON formatu“ scenarijus

16 lentelėje aprašyta panaudojimo atvejo „4.2.1. Pateikti viso tekstyno įvertinimo kriterijus“ specifikacija, o 22 pav. pateiktas jo scenarijaus modelis.

16 lentelė. PA „4.2.1. Pateikti viso tekstyno įvertinimo kriterijus“ specifikacija

<table>
<tr><td colspan="3">Panaudojimo atvejis „4.2.1. Pateikti viso tekstyno įvertinimo kriterijus“.</td></tr>
<tr><td colspan="3">Tikslas. Gauti anaforų sprendimo rezultatų įvertinimo kriterijus.</td></tr>
<tr><td colspan="2">Išankstinė sąlyga.</td><td>Sistemos naudotojas prisijungęs prie sistemos. Įjungtas viso tekstyno vertinimo langas.</td></tr>
<tr><td colspan="2">Aktorius.</td><td>Anaforų sprendimo kūrėjas.</td></tr>
<tr><td colspan="2">Sužadinimo sąlyga.</td><td>Sistemos naudotojas importavo anaforų sprendimo JSON rezultatų bylas.</td></tr>
<tr><td rowspan="3">Susiję panaudojimo atvejai</td><td>Išplečia PA</td><td>–</td></tr>
<tr><td>Apima PA</td><td>„4.2. Importuoti viso tekstyno anaforų sprendimo rezultatus JSON formatu“.</td></tr>
<tr><td>Specializuoja PA</td><td>–</td></tr>
<tr><td colspan="2">Pagrindinis įvykių srautas</td><td>Sistemos reakcija ir sprendimai</td></tr>
<tr><td colspan="2"></td><td>Atvaizduojami anaforų sprendimo įvertinimo kriterijai dydžiais R, P, F-vertė, RR ir tarpinės kriterijų apskaičiavimo reikšmės: T, F, C, E.</td></tr>
<tr><td colspan="2">Alternatyvūs scenarijai</td><td></td></tr>
<tr><td colspan="2">Parenkama netinkama JSON formato byla, t. y. parenkamas ne JSON formato byla arba kito teksto (pagal ID) anaforų anotacijos byla.</td><td>Parodomas klaidos pranešimas.</td></tr>
</table>

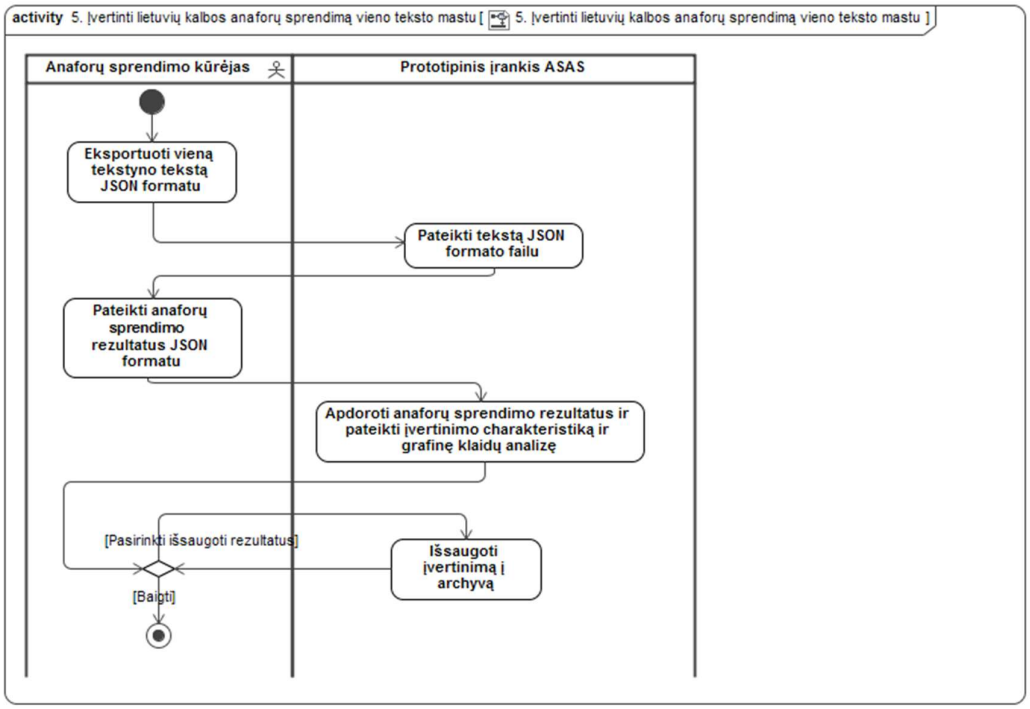


22 pav. PA „4.2.1. Pateikti viso tekstyno įvertinimo kriterijus“ scenarijus

17 lentelėje aprašyta panaudojimo atvejo „5. Įvertinti lietuvių kalbos anaforų sprendimą vieno teksto mastu“ specifikacija, o 23 pav. pateiktas jo scenarijaus modelis.

17 lentelė. PA „5. Įvertinti lietuvių kalbos anaforų sprendimą vieno teksto mastu“ specifikacija

<table>
<tr><td colspan="3">Panaudojimo atvejis „5. Įvertinti lietuvių kalbos anaforų sprendimą vieno teksto mastu“.</td></tr>
<tr><td colspan="3">Tikslas. Įvertinti automatinio anaforų sprendimo rezultatus ir gauti įvertinimo kriterijus bei grafinę klaidų analizę vieno teksto mastu.</td></tr>
<tr><td colspan="2">Išankstinė sąlyga.</td><td>Sistemos naudotojas prisijungęs prie sistemos ir įkeltas vienas tekstas bei įkeltas jo anaforų anotavimo etalonas.</td></tr>
<tr><td colspan="2">Aktorius.</td><td>Anaforų sprendimo kūrėjas.</td></tr>
<tr><td colspan="2">Sužadinimo sąlyga.</td><td>Sistemos naudotojas pasirenka „Įkelti sprendimo anotaciją JSON“.</td></tr>
<tr><td rowspan="3">Susiję panaudojimo atvejai</td><td>Išplečia PA</td><td>–</td></tr>
<tr><td>Apima PA</td><td>–</td></tr>
<tr><td>Specializuoja PA</td><td>–</td></tr>
<tr><td colspan="2">Pagrindinis įvykių srautas</td><td>Sistemos reakcija ir sprendimai</td></tr>
<tr><td colspan="2"></td><td>Atvaizduoja anaforų sprendimo rezultatų JSON bylos parinkimo langą.</td></tr>
<tr><td colspan="2">Nurodo anaforų sprendimo rezultatų JSON bylą.</td><td>Atvaizduojami anaforų sprendimo įvertinimo kriterijai dydžiais R, P, F-vertė, RR ir tarpinės kriterijų apskaičiavimo reikšmės: T, F, C, E ir grafinė klaidų analizė.</td></tr>
<tr><td colspan="3">Alternatyvūs scenarijai</td></tr>
<tr><td colspan="2">Parenkamas netinkamas failas.</td><td>Parodomas klaidos pranešimas.</td></tr>
</table>

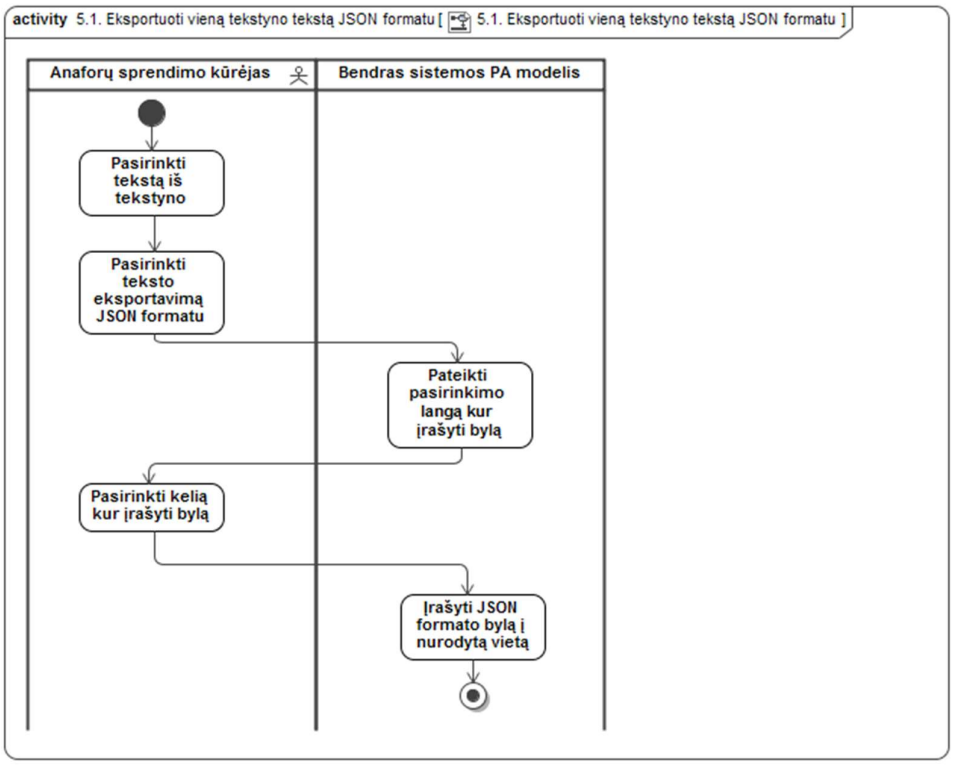


23 pav. PA „5. Įvertinti lietuvių kalbos anaforų sprendimą vieno teksto mastu“ scenarijus

18 lentelėje aprašyta panaudojimo atvejo „5.1. Eksportuoti vieną tekstyno tekstą JSON formatu“ specifikacija, o 24 pav. pateiktas jo scenarijaus modelis.

18 lentelė. PA „5.1. Eksportuoti vieną tekstyno tekstą JSON formatu“ specifikacija

<table>
<tr><th colspan="3">Panaudojimo atvejis „5.1. Eksportuoti vieną tekstyno tekstą JSON formatu“.</th></tr>
<tr><td colspan="3">Tikslas. Eksportuoti viena tekstyno tekstą JSON formatu.</td></tr>
<tr><td colspan="2">Išankstinė sąlyga.</td><td>Sistemos naudotojas prisijungęs prie sistemos ir įkeltas vienas tekstas bei įkeltas jo anaforų anotavimo etalonas.</td></tr>
<tr><td colspan="2">Aktorius.</td><td>Anaforų sprendimo kūrėjas.</td></tr>
<tr><td colspan="2">Sužadinimo sąlyga.</td><td>Sistemos naudotojas pasirenka „Eksportuoti straipsnį JSON“.</td></tr>
<tr><td rowspan="3">Susiję panaudojimo atvejai</td><td>Išplečia PA</td><td>–</td></tr>
<tr><td>Apima PA</td><td>„5. Įvertinti lietuvių kalbos anaforų sprendimą vieno teksto mastu“.</td></tr>
<tr><td>Specializuoja PA</td><td>–</td></tr>
<tr><td colspan="2">Pagrindinis įvykių srautas</td><td>Sistemos reakcija ir sprendimai</td></tr>
<tr><td colspan="2"></td><td>Atidaromas langas, kuriame reikia nurodyti, kur eksportuoti teksto JSON bylą.</td></tr>
<tr><td colspan="2">Sistemos naudotojas nurodo katalogą, kur įrašyti bylas.</td><td>Įrašomos bylos į nurodytą katalogą.</td></tr>
</table>

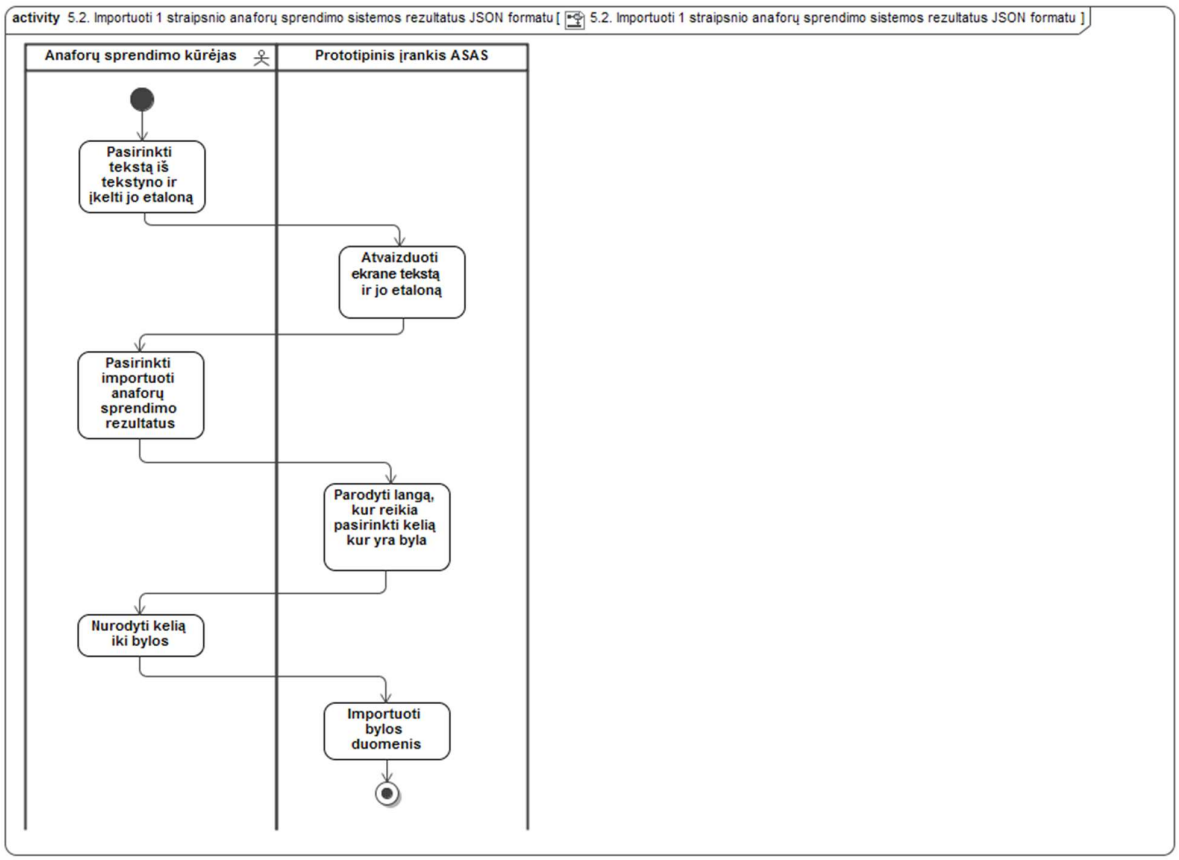


24 pav. PA „5.1. Eksportuoti vieną tekstyno tekstą JSON formatu“ scenarijus

19 lentelėje aprašyta panaudojimo atvejo „5.2. Importuoti 1 straipsnio anaforų sprendimo sistemos rezultatus JSON formatu“ specifikacija, o 25 pav. pateiktas jo scenarijaus modelis.

19 lentelė. PA „5.2. Importuoti 1 straipsnio anaforų sprendimo sistemos rezultatus JSON formatu“ specifikacija

<table>
<tr><th colspan="3">Panaudojimo atvejis „5.2. Importuoti 1 straipsnio anaforų sprendimo sistemos rezultatus JSON formatu“.</th></tr>
<tr><td colspan="3">Tikslas. Importuoti vieno teksto anaforų sprendimo rezultatus JSON formatu.</td></tr>
<tr><td colspan="2">Išankstinė sąlyga.</td><td>Sistemos naudotojas prisijungęs prie sistemos ir įkeltas vienas tekstas bei įkeltas jo anaforų anotavimo etalonas.</td></tr>
<tr><td colspan="2">Aktorius.</td><td>Anaforų sprendimo kūrėjas.</td></tr>
<tr><td colspan="2">Sužadinimo sąlyga.</td><td>Sistemos naudotojas pasirenka „Įkelti sprendimo anotaciją JSON“.</td></tr>
<tr><td rowspan="3">Susiję panaudojimo atvejai</td><td>Išplečia PA</td><td>–</td></tr>
<tr><td>Apima PA</td><td>„5. Įvertinti lietuvių kalbos anaforų sprendimą vieno teksto mastu“.</td></tr>
<tr><td>Specializuoja PA</td><td>–</td></tr>
<tr><td colspan="2">Pagrindinis įvykių srautas</td><td>Sistemos reakcija ir sprendimai</td></tr>
<tr><td colspan="2"></td><td>Atvaizduoja ekrane bylos kelio parinkimo langą. Reikia nurodyti anaforų sprendimo JSON rezultatų bylą.</td></tr>
<tr><td colspan="2">Sistemos naudotojas nurodo kelią iki bylos.</td><td>Atvaizduojami anaforų sprendimo įvertinimo kriterijai dydžiais R, P, F-vertė, RR ir tarpinės kriterijų apskaičiavimo reikšmės: T, F, C, E bei grafinė analizė.</td></tr>
<tr><td colspan="2">Alternatyvūs scenarijai</td><td></td></tr>
<tr><td colspan="2">Nurodomas kelias iki netinkamos bylos.</td><td>Parodomas klaidos pranešimas.</td></tr>
</table>

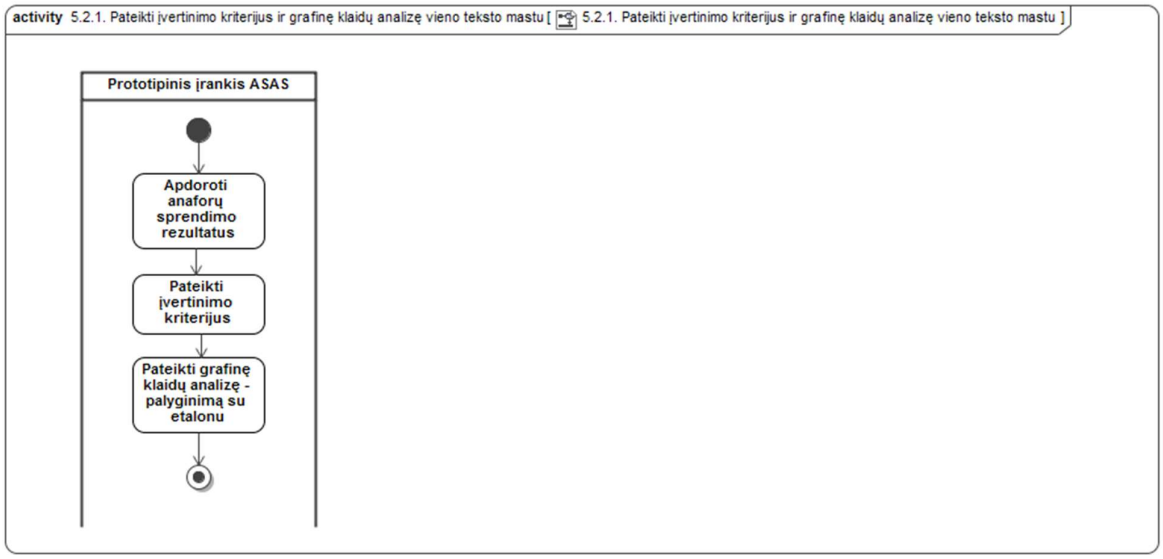


25 pav. PA „5.2. Importuoti 1 straipsnio anaforų sprendimo sistemos rezultatus JSON formatu“ scenarijus

20 lentelėje aprašyta panaudojimo atvejo „5.2.1. Pateikti įvertinimo kriterijus ir grafinę klaidų analizę vieno teksto mastu“ specifikacija, o 26 pav. pateiktas jo scenarijaus modelis.

20 lentelė. Panaudojimo atvejo „5.2.1. Pateikti įvertinimo kriterijus ir grafinę klaidų analizę vieno teksto mastu“ specifikacija

<table>
<tr><td colspan="3">Panaudojimo atvejis „5.2.1. Pateikti įvertinimo kriterijus ir grafinę klaidų analizę vieno teksto mastu“.</td></tr>
<tr><td colspan="3">Tikslas. Gauti anaforų sprendimo rezultatų įvertinimo kriterijus ir grafinę klaidų analizę vieno teksto mastu.</td></tr>
<tr><td colspan="2">Išankstinė sąlyga.</td><td>Sistemos naudotojas prisijungęs prie sistemos ir įkeltas vienas tekstas bei įkeltas jo anaforų anotavimo etalonas.</td></tr>
<tr><td colspan="2">Aktorius.</td><td>Anaforų sprendimo kūrėjas.</td></tr>
<tr><td colspan="2">Sužadinimo sąlyga.</td><td>Sistemos naudotojas importavo anaforų sprendimo JSON rezultatų bylą.</td></tr>
<tr><td rowspan="3">Susiję panaudojimo atvejai</td><td>Išplečia PA</td><td>–</td></tr>
<tr><td>Apima PA</td><td>„5.2. Importuoti 1 straipsnio anaforų sprendimo sistemos rezultatus JSON formatu“.</td></tr>
<tr><td>Specializuoja PA</td><td>–</td></tr>
<tr><td colspan="2">Pagrindinis įvykių srautas</td><td>Sistemos reakcija ir sprendimai</td></tr>
<tr><td colspan="2"></td><td>Atvaizduojami anaforų sprendimo įvertinimo kriterijai dydžiais R, P, F-vertė, RR ir tarpinės kriterijų apskaičiavimo reikšmės: T, F, C, E.</td></tr>
<tr><td colspan="2">Alternatyvūs scenarijai</td><td></td></tr>
<tr><td colspan="2">Parenkama netinkama JSON formato byla, t. y. parenkama ne JSON formato byla arba kito teksto (pagal ID) anaforų anotacijos byla.</td><td>Parodomas klaidos pranešimas.</td></tr>
</table>

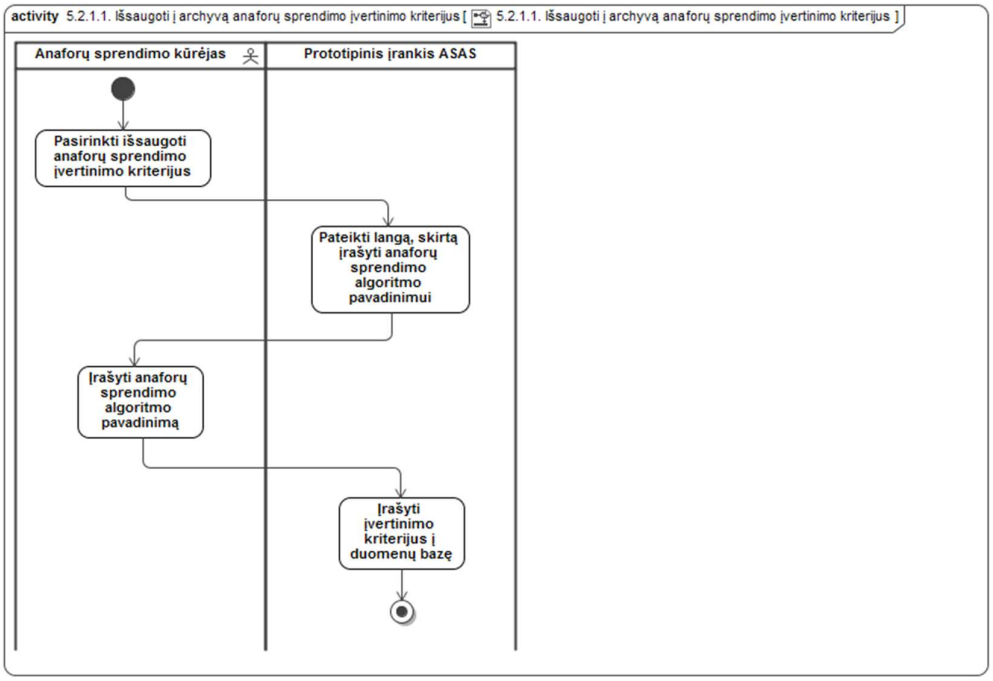


26 pav. PA „5.2.1. Pateikti įvertinimo kriterijus ir grafinę klaidų analizę vieno teksto mastu“ scenarijus

21 lentelėje aprašyta panaudojimo atvejo „5.2.1.1. Išsaugoti į archyvą anaforų sprendimo įvertinimo kriterijus“ specifikacija, o 27 pav. pateiktas jo scenarijaus modelis.

21 lentelė. PA „5.2.1.1. Išsaugoti į archyvą anaforų sprendimo įvertinimo kriterijus“ specifikacija

Panaudojimo atvejis „5.2.1.1. Išsaugoti į archyvą anaforų sprendimo įvertinimo kriterijus“.		
Tikslas. Gauti anaforų sprendimo rezultatų įvertinimo kriterijus ir grafinę klaidų analizę vieno teksto mastu.		
Išankstinė sąlyga.		Sistemos naudotojas prisijungęs prie sistemos. Ekrane atvaizduoti anaforų sprendimo įvertinimo kriterijai vieno teksto mastu.
Aktorius.		Anaforų sprendimo kūrėjas.
Sužadinimo sąlyga.		Sistemos naudotojas pasirenka „Išsaugoti įvertinimą“.
Susiję panaudojimo atvejai	**Išplečia PA**	„5.2.1. Pateikti įvertinimo kriterijus ir grafinę klaidų analizę vieno teksto mastu“.
	Apima PA	–
	Specializuoja PA	–
Pagrindinis įvykių srautas		**Sistemos reakcija ir sprendimai**
		Atvaizduojami langas įrašyti anaforų sprendimo pavadinimui.
Įrašomas anaforų sprendimo pavadinimas.		Įrašomi įvertinimo kriterijai į duomenų bazę.
Alternatyvūs scenarijai		
Nenurodomas anaforų sprendimo pavadinimas.		Parodomas klaidos pranešimas.

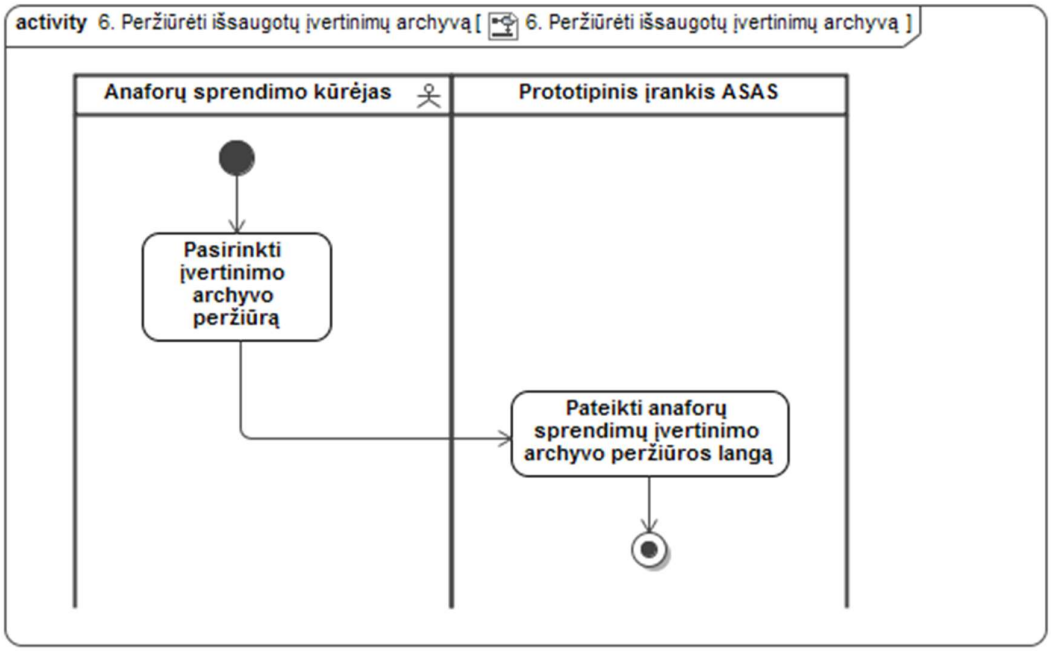


27 pav. PA „5.2.1.1. Išsaugoti į archyvą anaforų sprendimo įvertinimo kriterijus“ scenarijus

22 lentelėje aprašyta panaudojimo atvejo „6. Peržiūrėti išsaugotų įvertinimų archyvą“ specifikacija, o 28 pav. pateiktas jo scenarijaus modelis.

22 lentelė. PA „6. Peržiūrėti išsaugotų įvertinimų archyvą“ specifikacija

<table>
<tr><td colspan="3">Panaudojimo atvejis „6. Peržiūrėti išsaugotų įvertinimų archyvą“.</td></tr>
<tr><td colspan="3">Tikslas. Atvaizduoti ekrane anaforų sprendimų įvertinimų sąrašą.</td></tr>
<tr><td colspan="2">Išankstinė sąlyga.</td><td>Sistemos naudotojas prisijungęs prie sistemos.</td></tr>
<tr><td colspan="2">Aktorius.</td><td>Anaforų sprendimo kūrėjas.</td></tr>
<tr><td colspan="2">Sužadinimo sąlyga.</td><td>Sistemos naudotojas pasirenka „Įvertinimų archyvas“.</td></tr>
<tr><td rowspan="3">Susiję panaudojimo atvejai</td><td>Išplečia PA</td><td>–</td></tr>
<tr><td>Apima PA</td><td>–</td></tr>
<tr><td>Specializuoja PA</td><td>–</td></tr>
<tr><td colspan="2">Pagrindinis įvykių srautas</td><td>Sistemos reakcija ir sprendimai</td></tr>
<tr><td colspan="2"></td><td>Atvaizduoja ekrane anaforų sprendimų įvertinimų sąrašą.</td></tr>
</table>

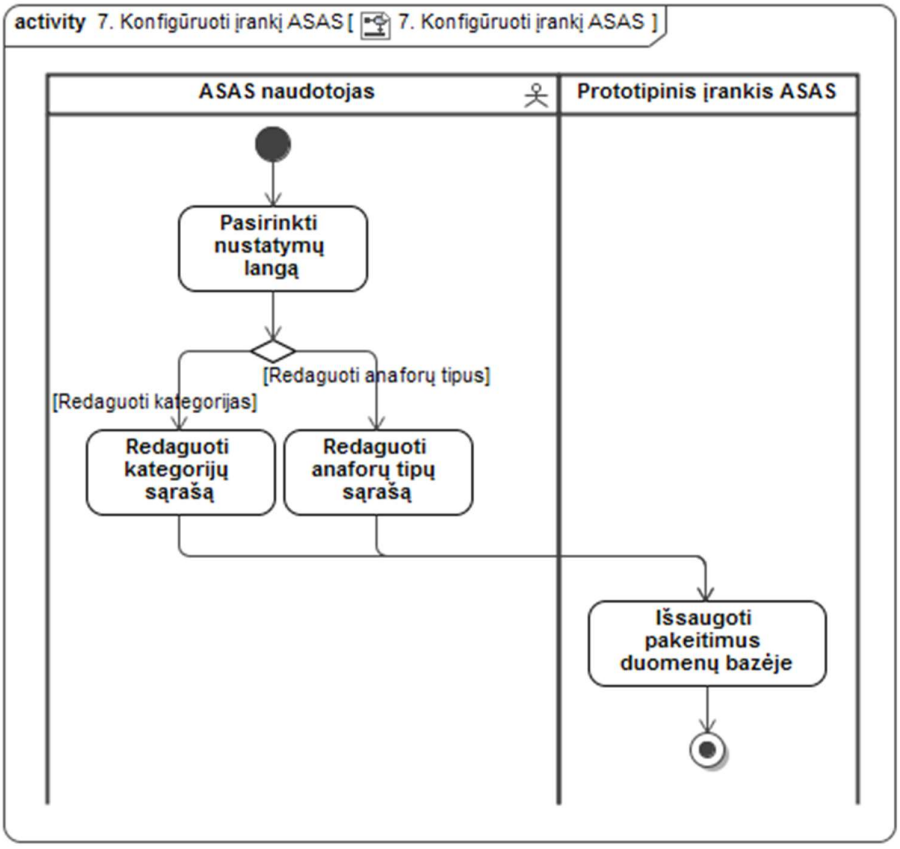


28 pav. PA „6. Peržiūrėti išsaugotų įvertinimų archyvą“ scenarijus

23 lentelėje aprašyta panaudojimo atvejo „7. Konfigūruoti įrankį ASAS“ specifikacija, o 29 pav. pateiktas jo scenarijaus modelis.

23 lentelė. PA „7. Konfigūruoti įrankį ASAS“ specifikacija

Panaudojimo atvejis „7. Konfigūruoti įrankį ASAS“.		
Tikslas. Konfigūruoti įranki ASAS: kurti / redaguoti teksto kategorijas, kurti / redaguoti anaforų tipus / kodus.		
Išankstinė sąlyga.		Sistemos naudotojas prisijungęs prie sistemos.
Aktorius.		ASAS naudotojas.
Sužadinimo sąlyga.		Sistemos naudotojas pasirenka „Nustatymai“.
Susiję panaudojimo atvejai	**Išplečia PA**	–
	Apima PA	–
	Specializuoja PA	–
Pagrindinis įvykių srautas		**Sistemos reakcija ir sprendimai**
		Atvaizduojamas ekrane nustatymų langas.

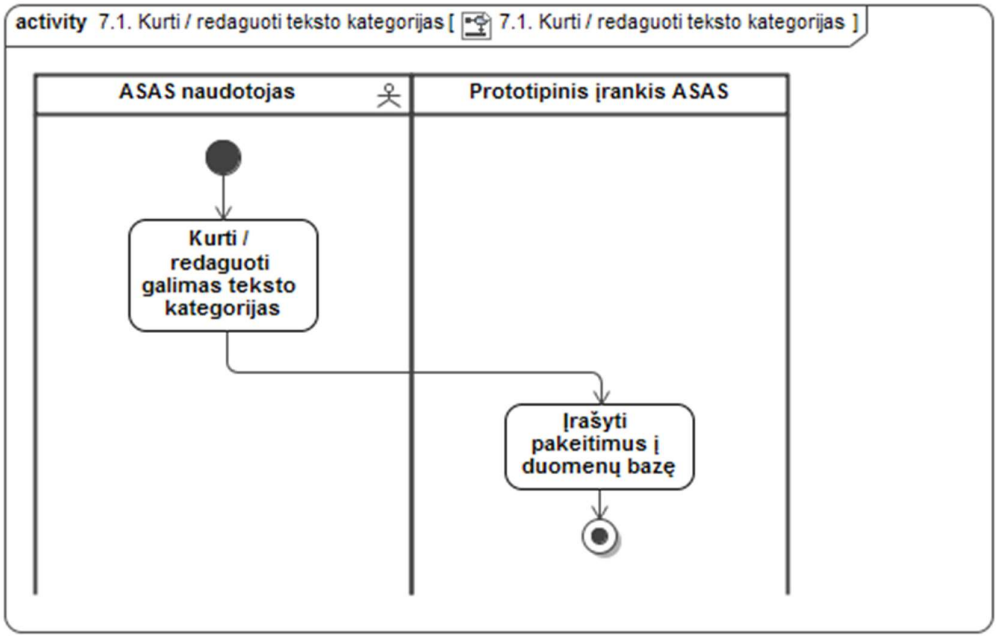


29 pav. PA „7. Konfigūruoti įrankį ASAS“ scenarijus

24 lentelėje aprašyta panaudojimo atvejo „7.1. Kurti / redaguoti teksto kategorijas“ specifikacija, o 30 pav. pateiktas jo scenarijaus modelis.

24 lentelė. PA „7.1. Kurti / redaguoti teksto kategorijas“ specifikacija

Panaudojimo atvejis „7.1. Kurti / redaguoti teksto kategorijas“.		
Tikslas. Kurti / redaguoti teksto kategorijas.		
Išankstinė sąlyga.		Sistemos naudotojas prisijungęs prie sistemos. Atidarytas nustatymų langas.
Aktorius.		ASAS naudotojas.
Sužadinimo sąlyga.		Sistemos naudotojas pasirenka redaguoti / ištrinti ar sukurti naują kategoriją.
Susiję panaudojimo atvejai	**Išplečia PA**	–
	Apima PA	„7. Konfigūruoti įrankį ASAS“.
	Specializuoja PA	–
Pagrindinis įvykių srautas		**Sistemos reakcija ir sprendimai**
		Atvaizduojamas teksto kategorijos pavadinimo įrašymo langas.
Įrašomas naujas kategorijos pavadinimas / redaguojamas senas kategorijos pavadinimas.		Sukuriama nauja kategorija / pakoreguojamas kategorijos pavadinimas.
Alternatyvūs scenarijai		
		Ištrinama kategorija arba parodomas klaidos pranešimas, jei kategorijos trinti negalima, kai jau yra priskirta bent vienam tekstui.

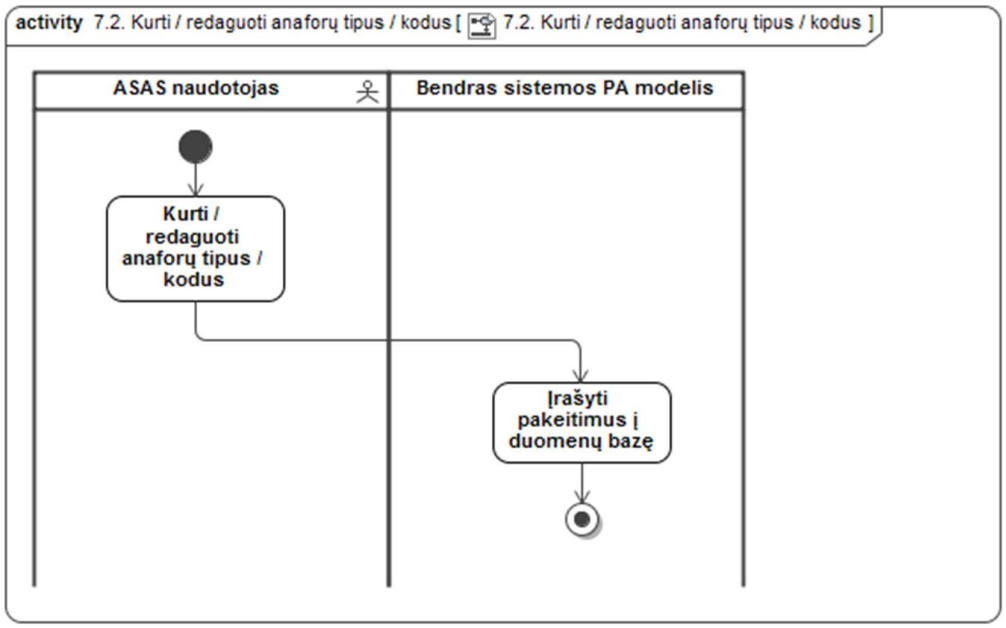


30 pav. PA „7.1. Kurti / redaguoti teksto kategorijas“ scenarijus

25 lentelėje aprašyta panaudojimo atvejo „7.2. Kurti / redaguoti anaforų tipus / kodus“ specifikacija, o 31 pav. pateiktas jo scenarijaus modelis.

25 lentelė. PA „7.2. Kurti / redaguoti anaforų tipus / kodus“ specifikacija

<table>
<tr><td colspan="3">Panaudojimo atvejis „7.2. Kurti / redaguoti anaforų tipus / kodus“.</td></tr>
<tr><td colspan="3">Tikslas. Kurti / redaguoti anaforų tipus / kodus.</td></tr>
<tr><td colspan="2">Išankstinė sąlyga.</td><td>Sistemos naudotojas prisijungęs prie sistemos. Atidarytas nustatymų langas.</td></tr>
<tr><td colspan="2">Aktorius.</td><td>ASAS naudotojas.</td></tr>
<tr><td colspan="2">Sužadinimo sąlyga.</td><td>Sistemos naudotojas pasirenka redaguoti / ištrinti ar sukurti naują tipą.</td></tr>
<tr><td rowspan="3">Susiję panaudojimo atvejai</td><td>Išplečia PA</td><td>–</td></tr>
<tr><td>Apima PA</td><td>„7. Konfigūruoti įrankį ASAS“.</td></tr>
<tr><td>Specializuoja PA</td><td>–</td></tr>
<tr><td colspan="2">Pagrindinis įvykių srautas</td><td>Sistemos reakcija ir sprendimai</td></tr>
<tr><td colspan="2"></td><td>Atidaromas anaforos tipo pavadinimo / kodo įrašymo langas.</td></tr>
<tr><td colspan="2">Įrašomas naujas anaforos tipo pavadinimas ir kodas / redaguojamas senas anaforos tipo pavadinimas ir kodas.</td><td>Sukuriamas naujas anaforos tipas / pakeičiamas anaforos tipo pavadinimas kodas.</td></tr>
<tr><td colspan="2">Alternatyvūs scenarijai</td><td></td></tr>
<tr><td colspan="2"></td><td>Ištrinamas anaforos tipas arba parodomas klaidos pranešimas, jei anaforos tipo trinti negalima.</td></tr>
</table>

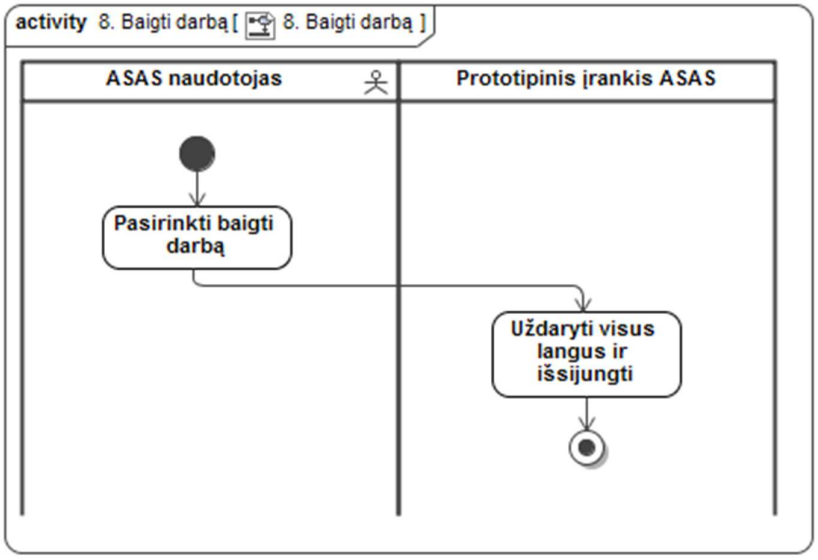


31 pav. PA „7.2. Kurti / redaguoti anaforų tipus / kodus“ scenarijus

26 lentelėje aprašyta panaudojimo atvejo „8. Baigti darbą“ specifikacija, o 32 pav. pateiktas jo scenarijaus modelis.

26 lentelė. PA „8. Baigti darbą“ specifikacija

Panaudojimo atvejis „8. Baigti darbą“.		
Tikslas. Išjungti įrankį.		
Išankstinė sąlyga.		Sistemos naudotojas prisijungęs prie sistemos.
Aktorius.		ASAS naudotojas.
Sužadinimo sąlyga.		Naudotojas pasirenka „Baigti darbą“
Susiję panaudojimo atvejai	**Išplečia PA**	–
	Apima PA	–
	Specializuoja PA	–
Pagrindinis įvykių srautas		**Sistemos reakcija ir sprendimai**
		Sistema išsijungia.



32 pav. PA „8. Baigti darbą“ scenarijus

2.1.2. Nefunkciniai reikalavimai

27 lentelėje pateiktas nefunkcinis reikalavimas sistemos stiliui.

27 lentelė. Nefunkcinis reikalavimas sistemos stiliui

Panaudojimo atvejai #	Visi.
Reikalavimo tipas #	10b
Reikalavimas:	Įrankis turi atrodyti solidžiai.
Pagrindimas:	Įrankiu turi pasitikėti jo naudotojai.
Tinkamumo kriterijus:	Patvirtinti turi daugiau nei 90% įrankio naudotojų.

28 lentelėje pateiktas nefunkcinis reikalavimas naudojimo paprastumui.

28 lentelė. Nefunkcinis reikalavimas naudojimo paprastumui

Panaudojimo atvejai #	3., 3.1, 3.2
Reikalavimo tipas #	9a
Reikalavimas:	Anaforų anotavimo procesas turi būti ypač patogus.
Pagrindimas:	Svarbu susitelkti į atliekamą darbą.
Tinkamumo kriterijus:	Patvirtinti turi daugiau nei 90% įrankio naudotojų.

29 lentelėje pateiktas nefunkcinis reikalavimas sistemos panaudojamumui - mokymuisi.

29 lentelė. Nefunkcinis reikalavimas sistemos panaudojamumui – mokymasis

Panaudojimo atvejai #	Visi.
Reikalavimo tipas #	11c
Reikalavimas:	Įrankiu turi būti ypač paprasta išmokti naudotis.
Pagrindimas:	Svarbu susitelkti į atliekamą darbą.
Tinkamumo kriterijus:	Daugiau nei 90% įrankio naudotojų turi sugebėti išmokti naudotis savarankiškai per 20 min.

30 lentelėje pateiktas nefunkcinis reikalavimas sistemos panaudojamumui – suprantamumas ir mandagumas.

30 lentelė. Nefunkcinis reikalavimas sistemos panaudojamumui – suprantamumas ir mandagumas

Panaudojimo atvejai #	Visi.
Reikalavimo tipas #	11d
Reikalavimas:	Įrankis turi naudoti simbolius ir žodžius, natūraliai suprantamus žmonėms, dirbantiems natūralios kalbos apdorojimo srityje.
Pagrindimas:	Įrankiu naudosis žmonės dirbantys natūralios kalbos apdorojimo srityje.
Tinkamumo kriterijus:	Daugiau nei 90% naujų įrankio naudotojų, turinčių patirtį natūralios kalbos apdorojimo srityje, neturi kilti klausimų apie žodžių ar simbolių, naudojamų įrankyje, reikšmes.

31 lentelėje pateiktas nefunkcinis reikalavimas sistemos vykdymo savybėms – užduočių vykdymo greitis.

31 lentelė. Nefunkcinis reikalavimas sistemos vykdymo savybėms – užduočių vykdymo greitis

Panaudojimo atvejai #	Visi.
Reikalavimo tipas #	12a
Reikalavimas:	Paleisti įrankį ir atlikti bet kurią funkciją turi būti įmanoma greitai.
Pagrindimas:	Įrankis turi nekelti naudotojų susierzinimo dėl greičio.
Tinkamumo kriterijus:	Įrankio paleidimas ir bet kurio lango atidarymas turi būti įmanomas per 5 sekundes.

32 lentelėje pateiktas nefunkcinis reikalavimas sistemos vykdymo savybėms – pirmas reikalavimas tikslumui.

32 lentelė. Nefunkcinis reikalavimas sistemos vykdymo savybėms – pirmas reikalavimas tikslumui

Panaudojimo atvejai #	Visi.
Reikalavimo tipas #	12a
Reikalavimas:	Įrankis turi veikti greitai.
Pagrindimas:	Įrankis turi nekelti naudotojų susierzinimo dėl greičio.
Tinkamumo kriterijus:	Bet kokie įrankio naudotojo (žmogaus) veiksmai su įrankiu: išsaugojimas, lango atidarymas, eksportavimas, importavimas turi būti atliekami per ne ilgiau nei 2 sekundes.

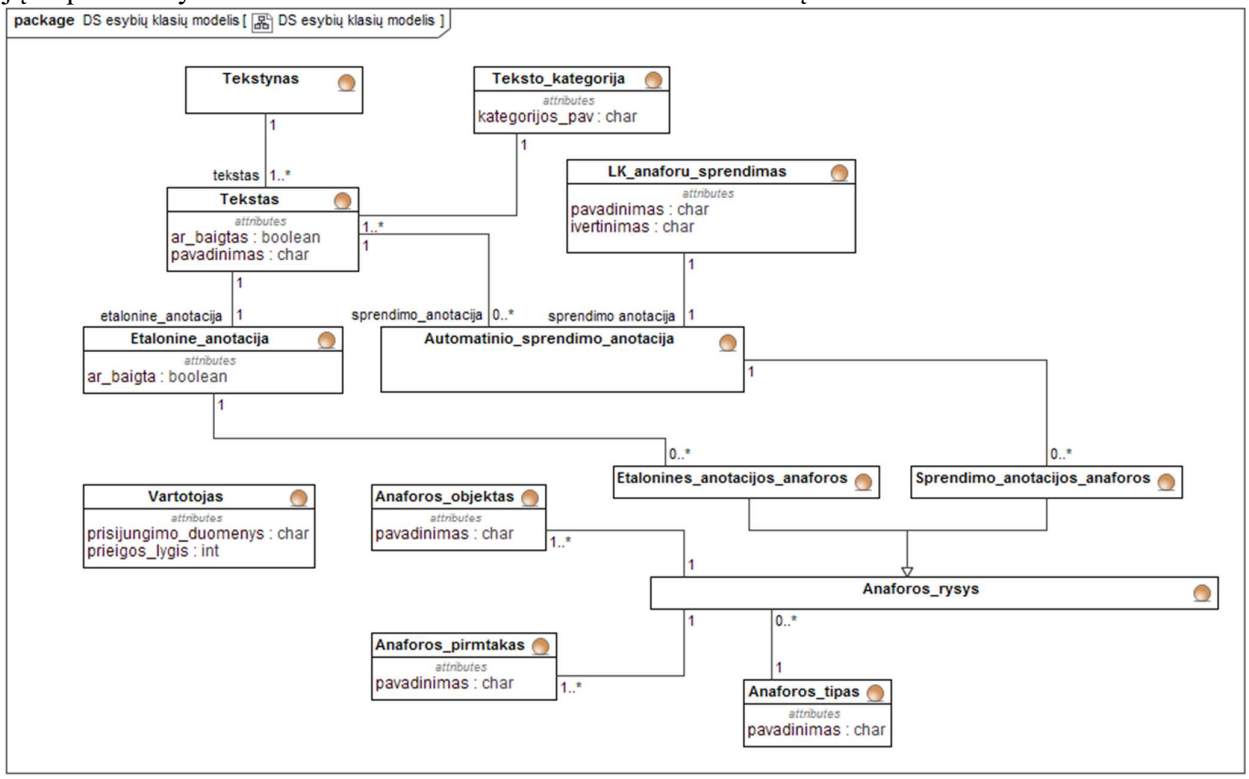
32 lentelėje pateiktas nefunkcinis reikalavimas sistemos vykdymo savybėms – antras reikalavimas tikslumui.

33 lentelė. Nefunkcinis reikalavimas vykdymo savybėms – antras reikalavimas tikslumui

Panaudojimo atvejai #	4., 4.2.1, 5.2.1, 6.
Reikalavimo tipas #	12c
Reikalavimas:	Visos naudojamos ir pateikiamos skaitinės išraiškos turi būti apvalinamos.
Pagrindimas:	Turi būti naudojamos tokios skaitinės reikšmės, kurias lengva skaityti / suprasti.
Tinkamumo kriterijus:	Visos naudojamos / pateikiamos anaforų sprendimo įvertinimo reikšmės ir tarpiniai rezultatai turi būti apvalinami vienos dešimt tūkstantosios tikslumu, o procentinės išraiškos – vienos šimtosios tikslumu.

2.2. Dalykinės srities modelis

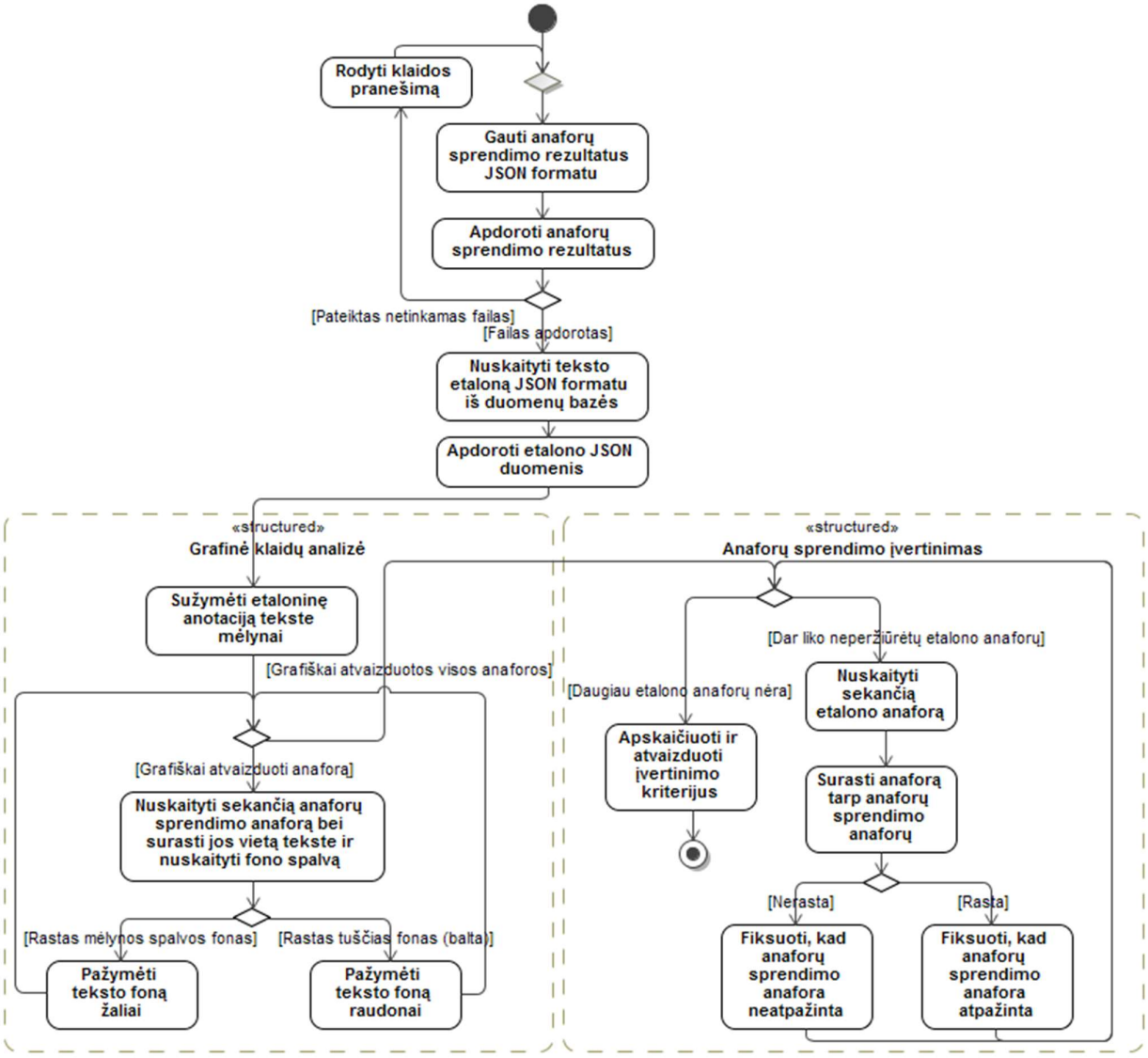
33 pav. pateiktas dalykinės srities modelis. Jame vaizduojamos kuriamo prototipo esybės bei jų tarpusavio ryšiai. Šiuo modeliu remiantis bus sukurtas duomenų bazės modelis.



33 pav. Dalykinės srities modelis

2.3. Anaforų sprendimo įvertinimo ir grafinės analizės algoritmas

34 pav. pateiktas įvertinimo ir analizės metodiką realizuojančio algoritmo modelis, skirtas anaforų sprendimo grafinei klaidų analizei ir įvertinimo kriterijų apskaičiavimui.



34 pav. Anaforų sprendimo įvertinimo ir grafinės analizės algoritmo modelis

2.4. Reikalavimų apibendrinimas

1. Šiame skyriuje suformuluoti protitipinio įrankio ASAS funkciniai ir nefunkciniai reikalavimai, kurių realizavimas turėtų sudaryti galimybes įvertinti lietuvių kalbos automatinius anaforų sprendimus.
2. Kadangi lietuvių kalba natūralios kalbos apdorojimo srityje laikoma neturinčia pakankamai resursų, todėl įrankis turi būti lankstus. Dėl to numatytos jo konfigūravimo galimybės.
3. Įrankiui keliami nefunkciniai reikalavimai turi užtikrinti, kad juo būtų lengvai, noriai ir efektyviai naudojamasi vykdant įvairius projektus.
4. Lietuvių kalbos tekstyno sudarymas numatytos galimybės įrankiu ir jo eksportavimas JSON formatu galbūt galėtų padėti vykdant kitus projektus (ne pagal tiesioginę paskirtį – anaforų sprendimų įvertinimas) natūralios kalbos apdorojimo srityje.

3. ĮRANKIO ASAS REALIZACIJOS PROJEKTAS

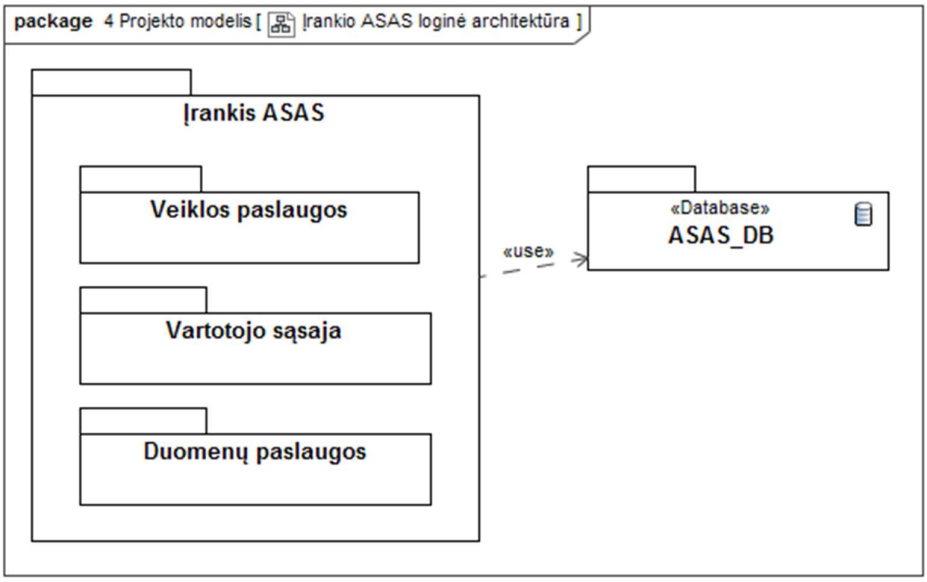
3.1. Projekto tikslas

Projekto tikslas – naudojant *CASE* priemonę suprojektuoti įrankio ASAS realizacijos modelį ir duomenų bazės schemą.

3.2. Architektūros projektas

3.2.1. Loginė architektūra

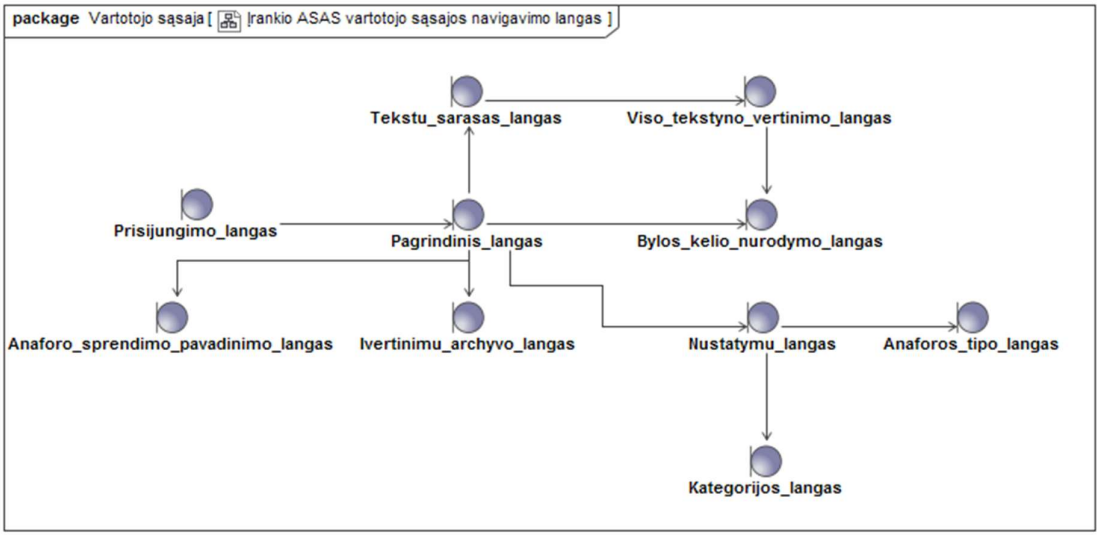
35 pav. pateikta sistemos loginė architektūra. Duomenų paslaugų posistemyje realizuota sąsaja su DB: vykdomos įvairios duomenų nuskaitymo ir įrašymo funkcijos.



35 pav. Įrankio ASAS loginės architektūros modelis

3.2.2. Vartotojo paslaugos

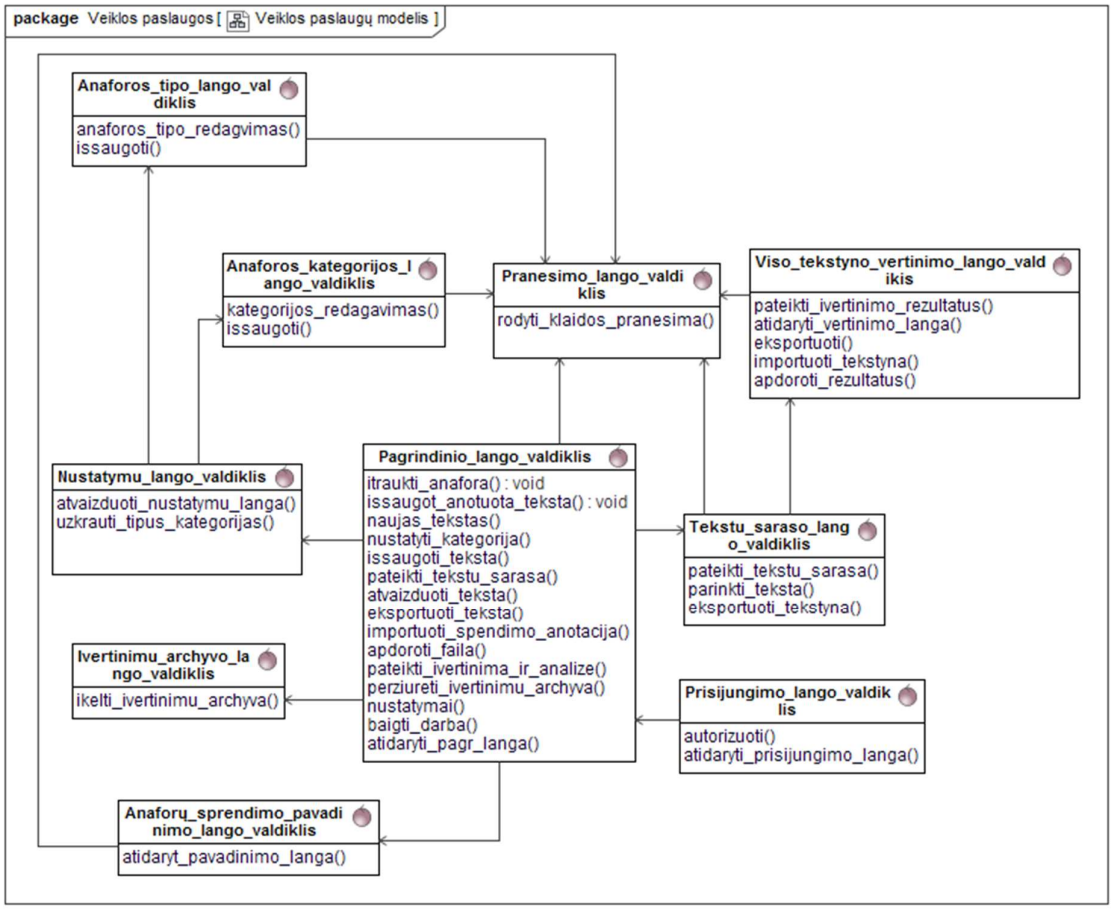
36 pav. pateiktas vartotojo sąsajos navigavimo planas. Paleidus įrankį ASAS, rodomas prisijungimo langas. Sėkmingai prisijungus, patenkama į pagrindinį langą iš kurio galima pareiti į visus kitus langus.



36 pav. Vartotojo sąsajos navigavimo planas

3.2.3. Veiklos paslaugos

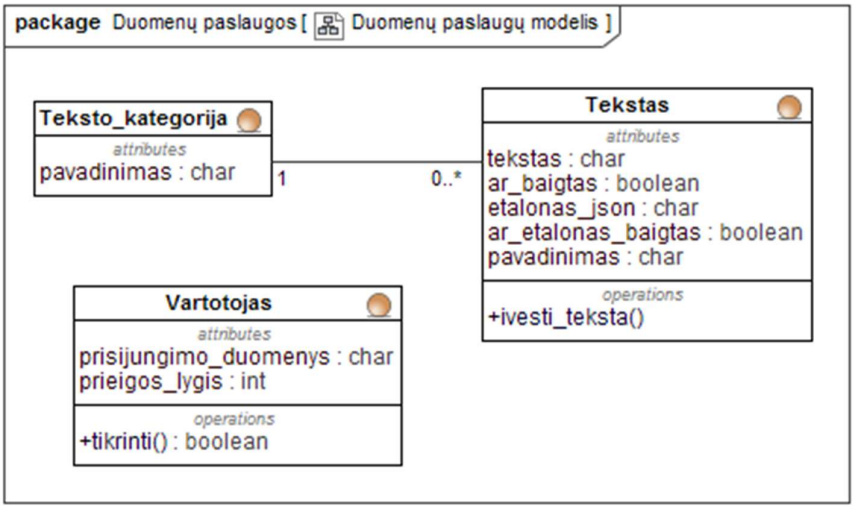
37 pav. pateiktas įrankio ASAS valdymo klasių modelis.



37 pav. Valdymo klasių modelis

3.2.4. Duomenų paslaugos

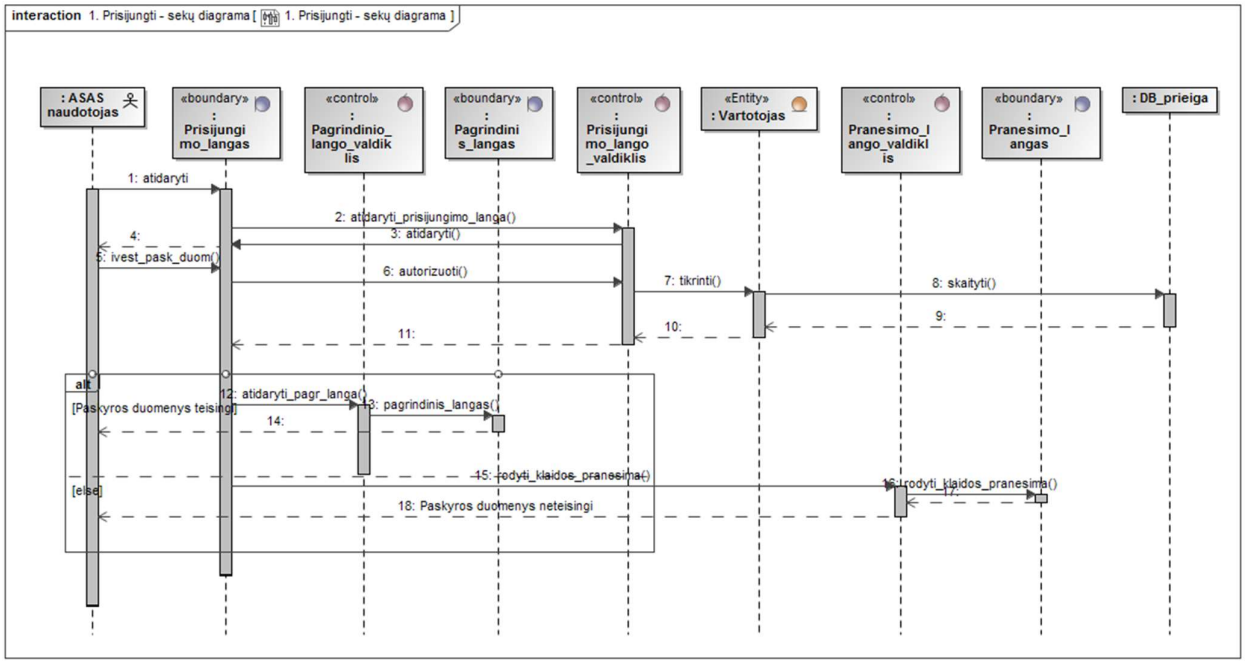
38 pav. pateiktas esybių klasių modelis ir jų tarpusavio sąveika.



38 pav. Esybių klasių modelis

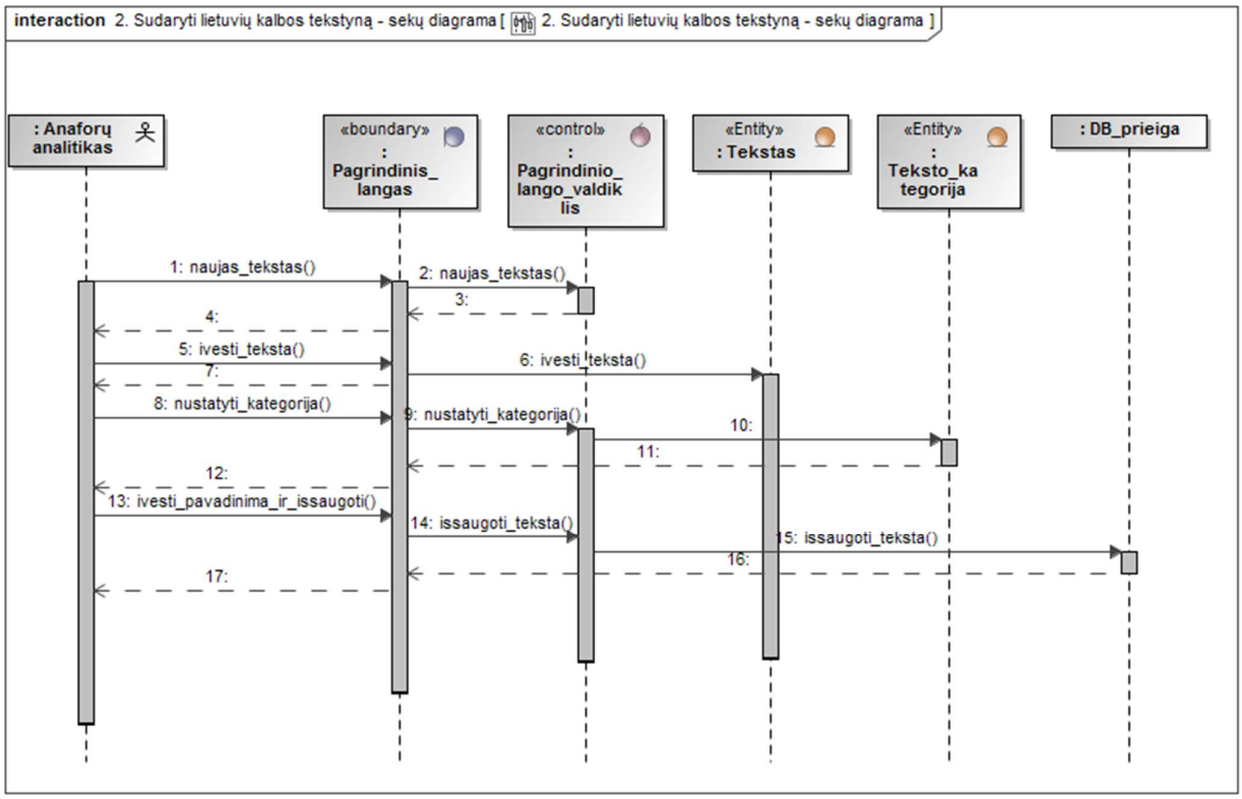
3.3. Elgsenos modelis

Elgsenos modelis pavaizduojamas panaudojimų atvejų realizacijos sekų diagramomis. 39 pav. pateikta panaudojimo atvejo „1. Prisijungti“ realizacijos sekų diagrama.



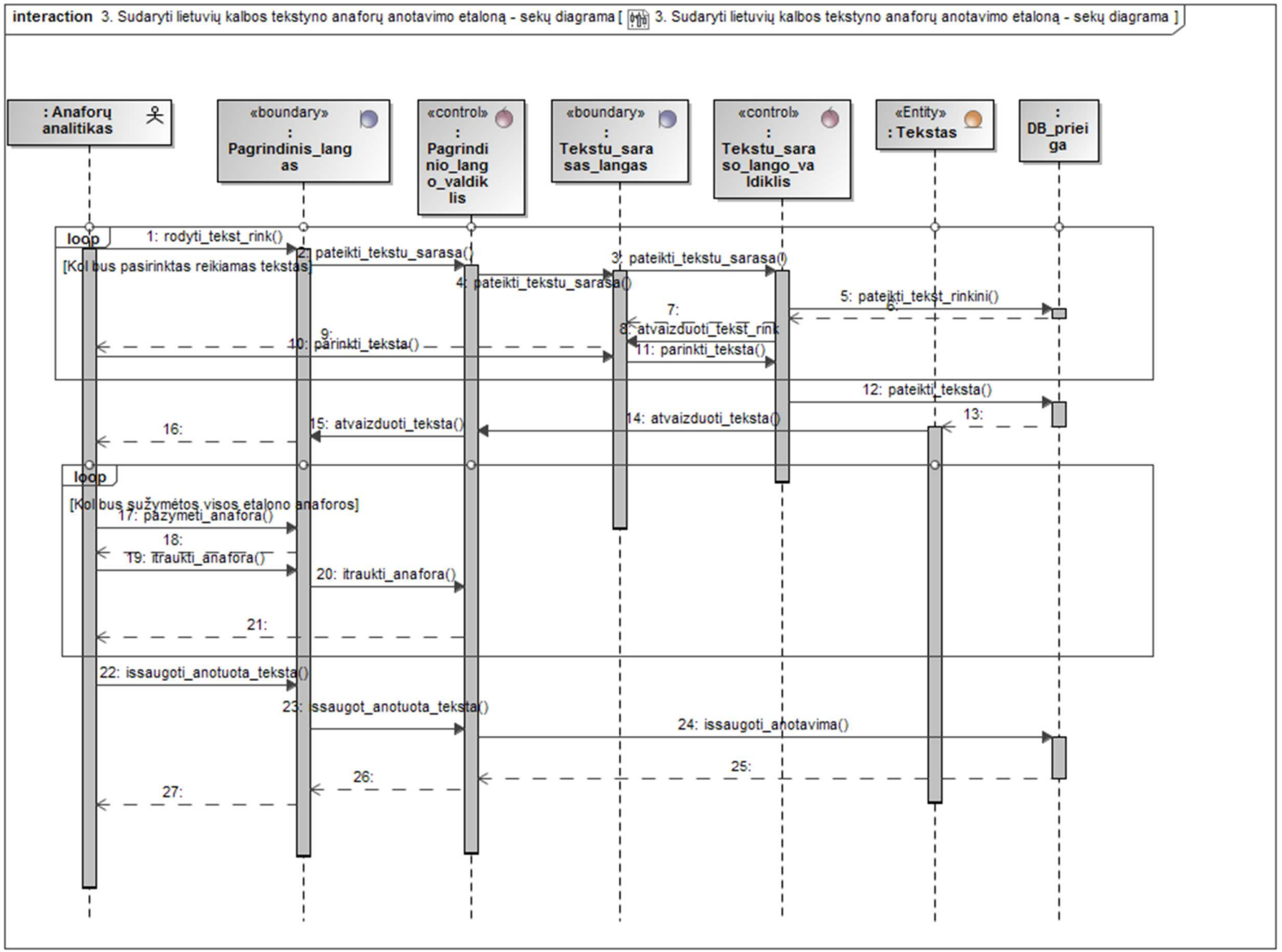
39 pav. Panaudojimo atvejo „1. Prisijungti“ realizacijos sekų diagrama

40 pav. pateikta panaudojimo atvejo „2. Sudaryti lietuvių kalbos tekstyną“ realizacijos sekų diagrama.



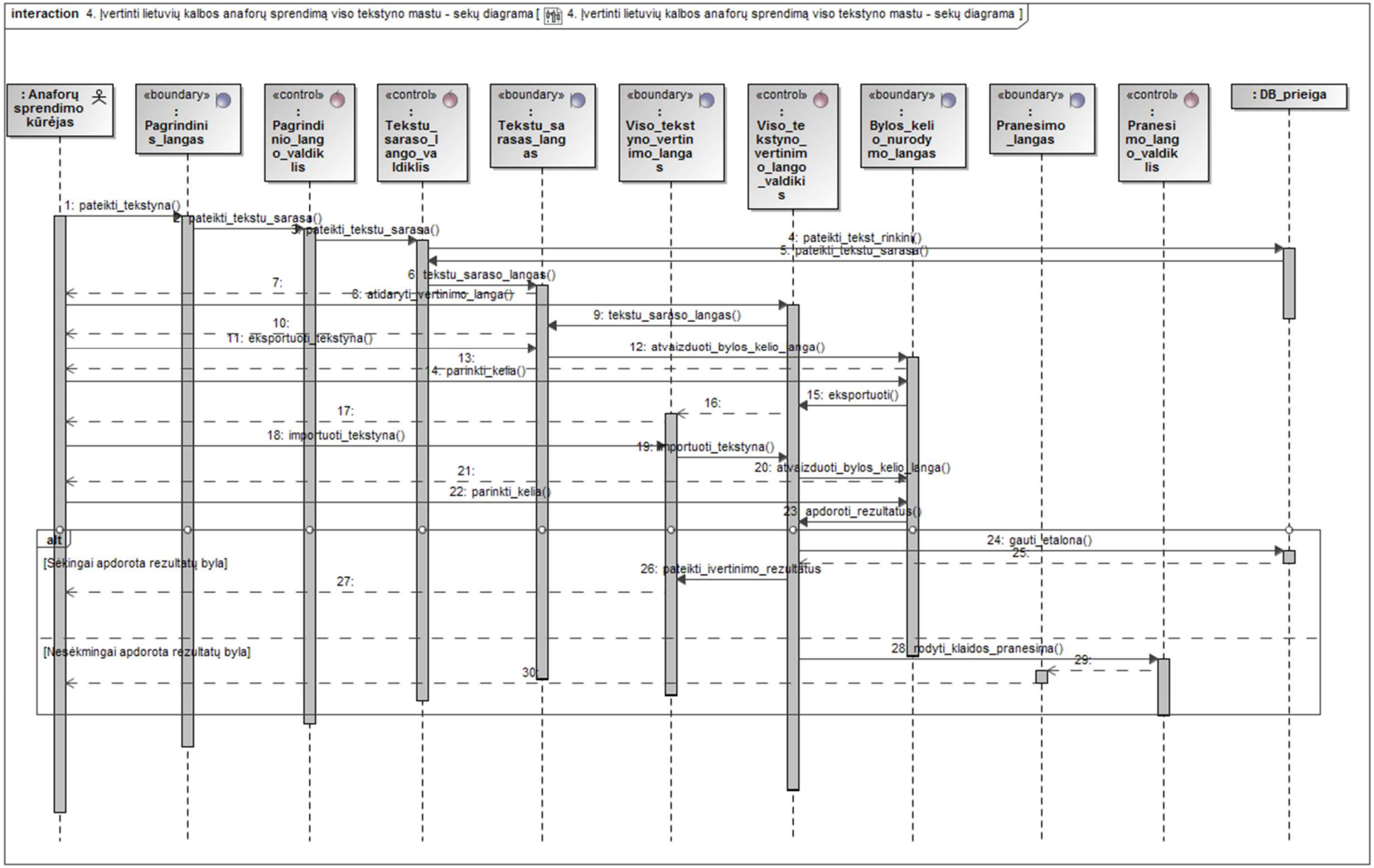
40 pav. 3.3.1. Panaudojimo atvejo „2. Sudaryti lietuvių kalbos tekstyną“ realizacijos sekų diagrama

41 pav. pateikta panaudojimo atvejo „3. Sudaryti lietuvių kalbos tekstyno anaforų anotavimo etaloną“ realizacijos sekų diagrama.



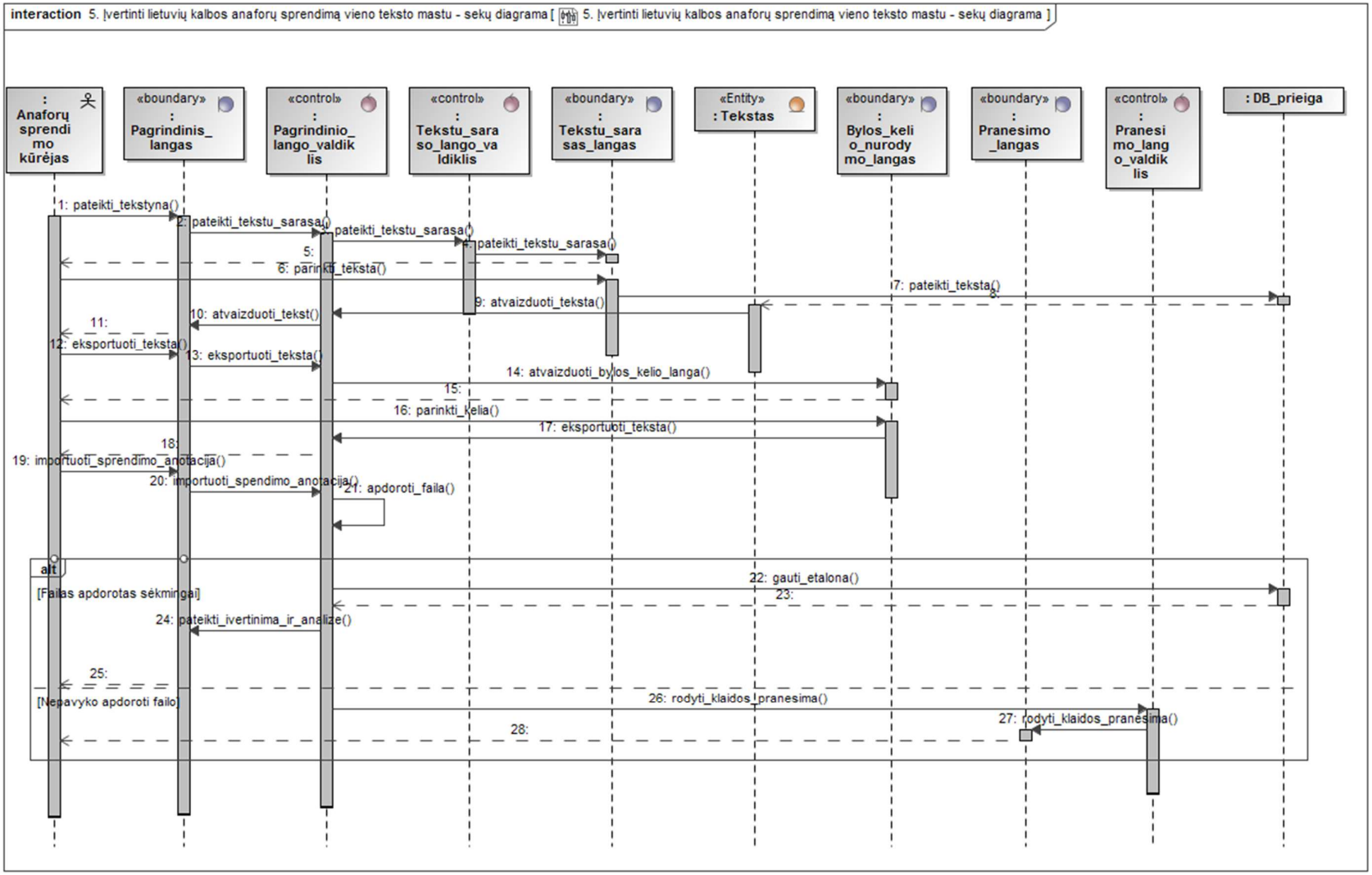
41 pav. 3.3.1. Panaudojimo atvejo „3. Sudaryti lietuvių kalbos tekstyno anaforų anotavimo etaloną“ realizacijos sekų diagrama

42 pav. pateikta panaudojimo atvejo „4. Įvertinti lietuvių kalbos anaforų sprendimą viso tekstyno mastu“ realizacijos sekų diagrama.



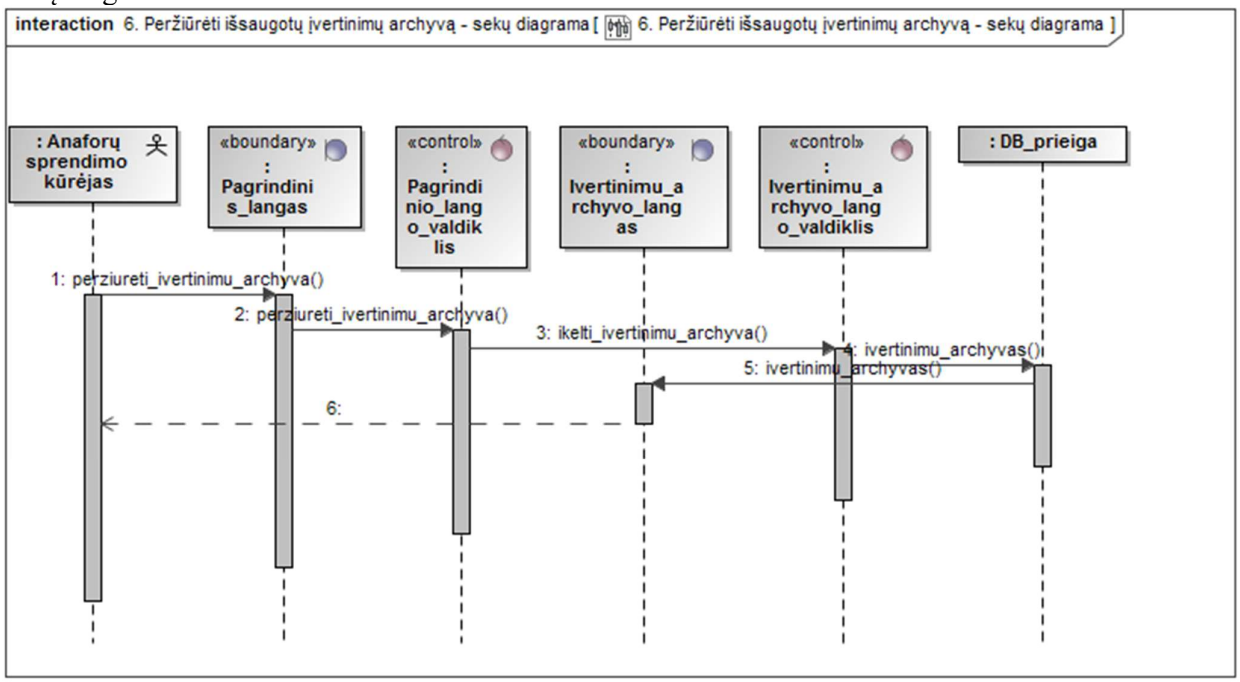
42 pav. 3.3.1. Panaudojimo atvejo „4. Įvertinti lietuvių kalbos anaforų sprendimą viso tekstyno mastu“ realizacijos sekų diagrama

43 pav. pateikta panaudojimo atvejo „5. Įvertinti lietuvių kalbos anaforų sprendimą vieno teksto mastu“ realizacijos sekų diagrama.



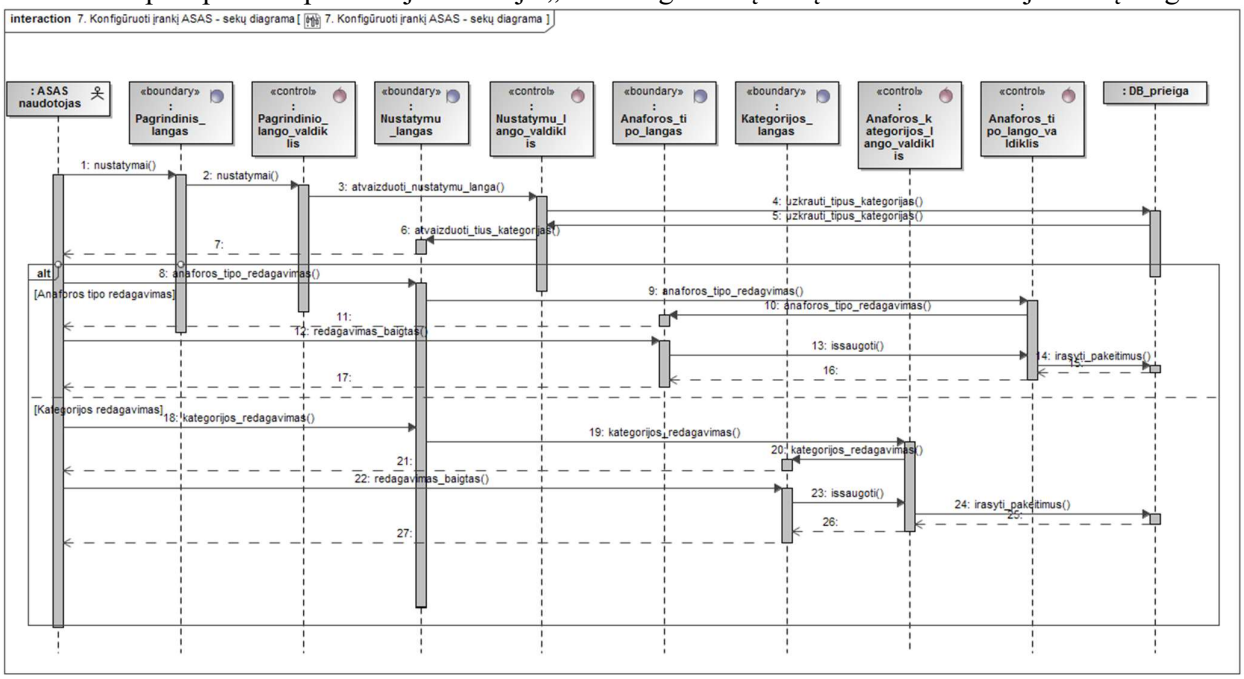
43 pav. 3.3.1. Panaudojimo atvejo „5. Įvertinti lietuvių kalbos anaforų sprendimą vieno teksto mastu“ realizacijos sekų diagrama

44 pav. pateikta panaudojimo atvejo „6. Peržiūrėti išsaugotų įvertinimų archyvą“ realizacijos sekų diagrama.



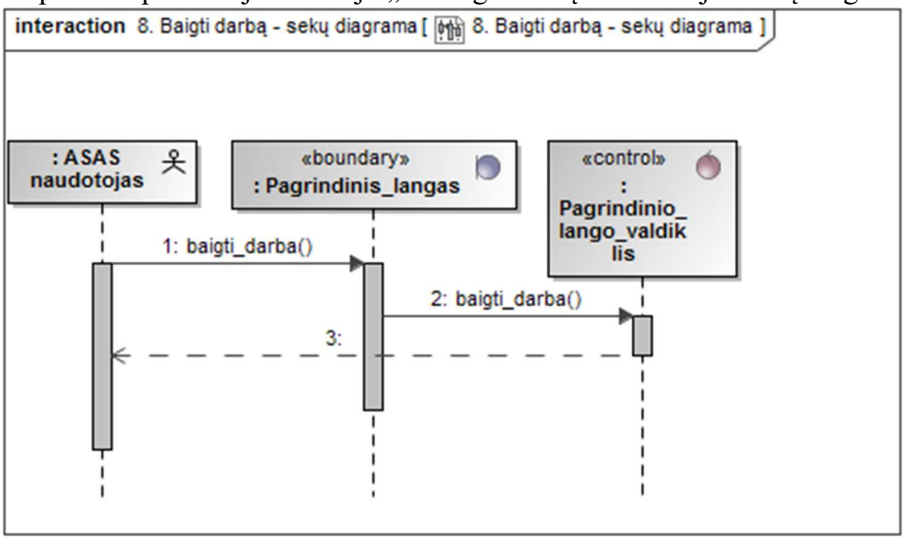
44 pav. 3.3.1. Panaudojimo atvejo „6. Peržiūrėti išsaugotų įvertinimų archyvą“ realizacijos sekų diagrama

45 pav. pateikta panaudojimo atvejo „7. Konfigūruoti įrankį ASAS“ realizacijos sekų diagrama.



45 pav. 3.3.1. Panaudojimo atvejo „7. Konfigūruoti įrankį ASAS“ realizacijos sekų diagrama

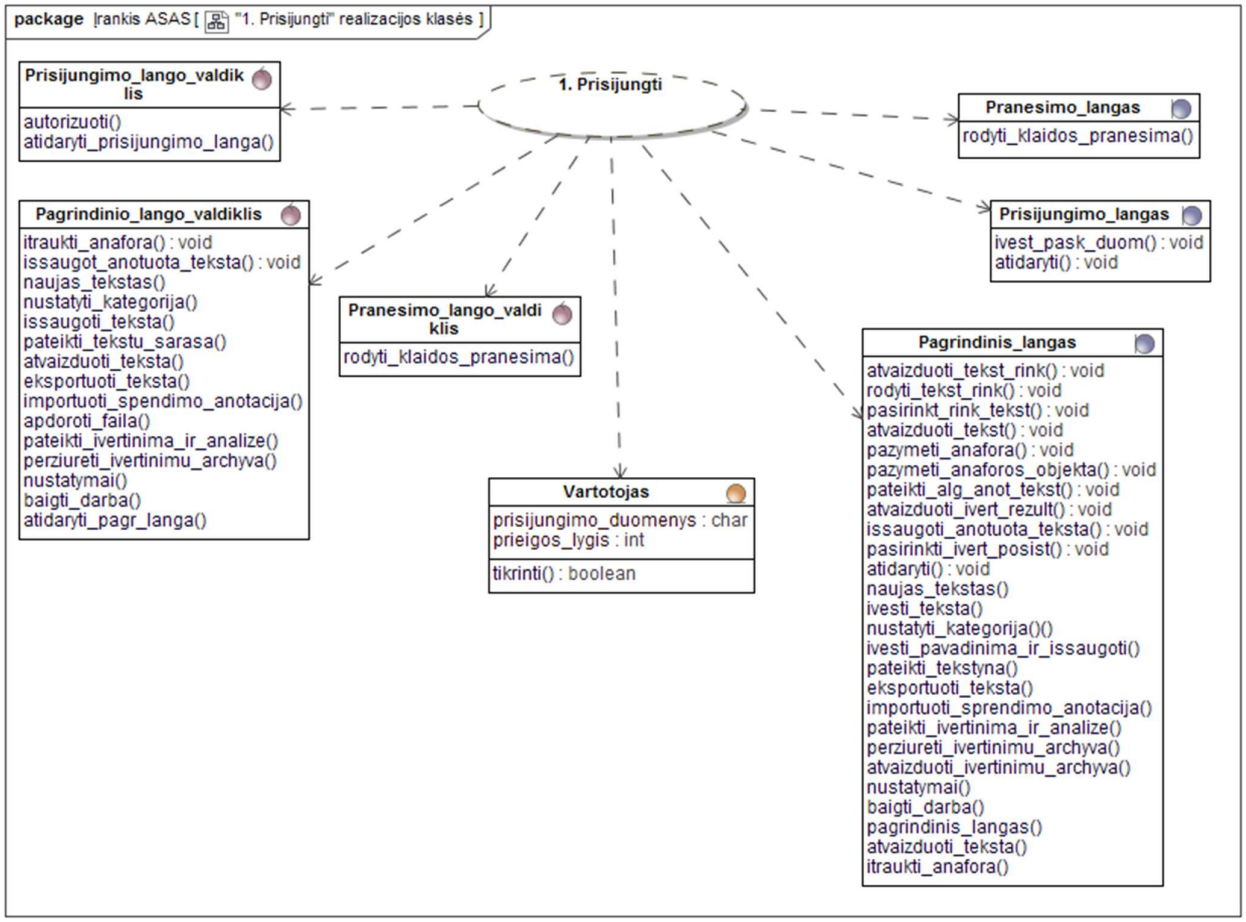
46 pav. pateikta panaudojimo atvejo „8. Baigti darbą“ realizacijos sekų diagrama.



46 pav. PA „8. Baigti darbą“ realizacijos sekų diagrama

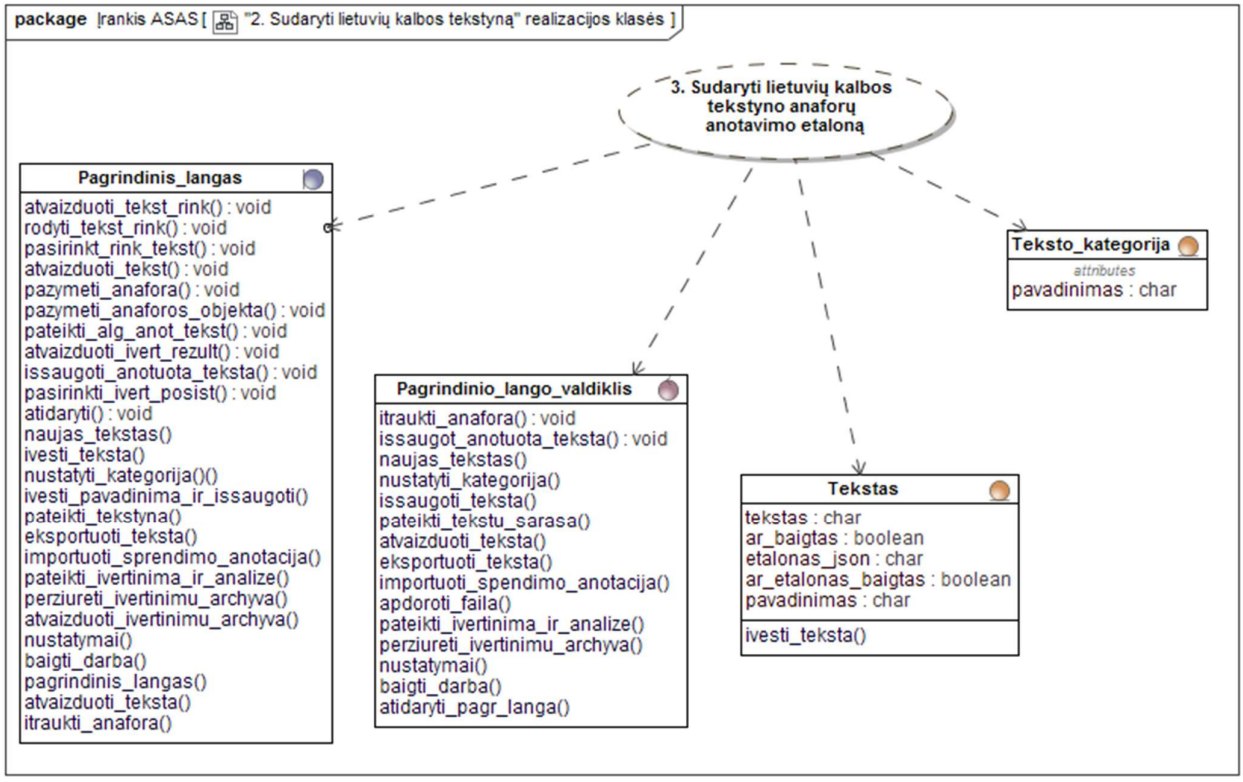
3.4. Detalus projektas

47 pav. pateikta panaudojimo atvejo „1. Prisijungti“ realizacija projekto klasėmis.



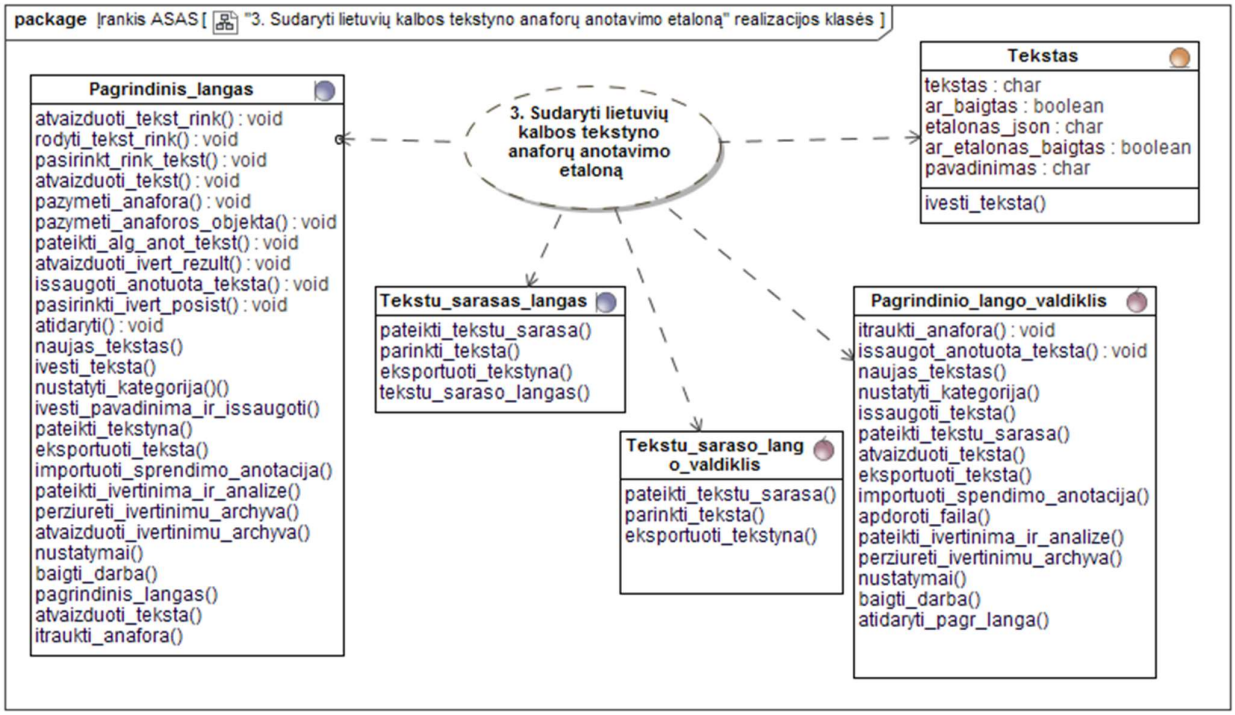
47 pav. PA „1. Prisijungti“ realizaciją projekto klasėmis

48 pav. pateikta panaudojimo atvejo „2. Sudaryti lietuvių kalbos tekstyną“ realizacija projekto klasėmis.



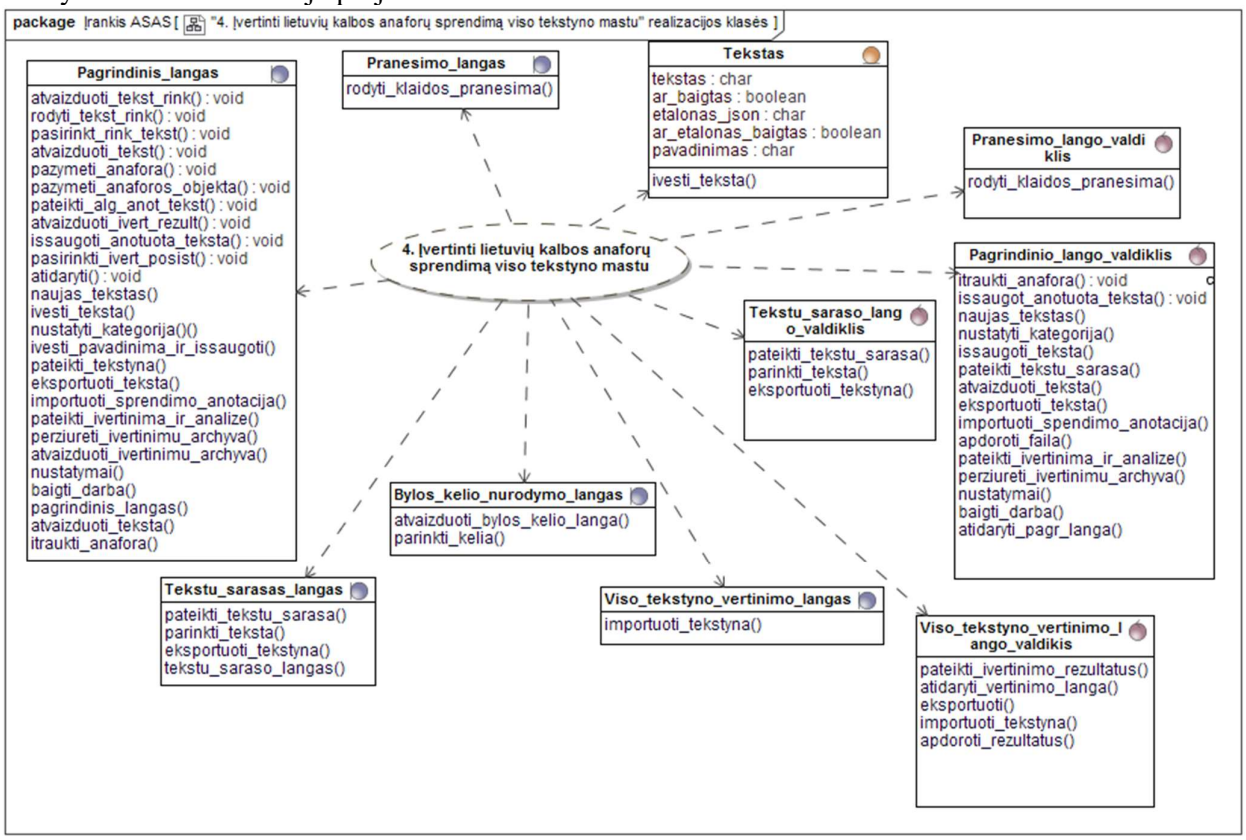
48 pav. PA „2. Sudaryti lietuvių kalbos tekstyną“ realizacija projekto klasėmis

49 pav. pateikta panaudojimo atvejo „3. Sudaryti lietuvių kalbos tekstyno anaforų anotavimo etaloną“ realizacija projekto klasėmis.



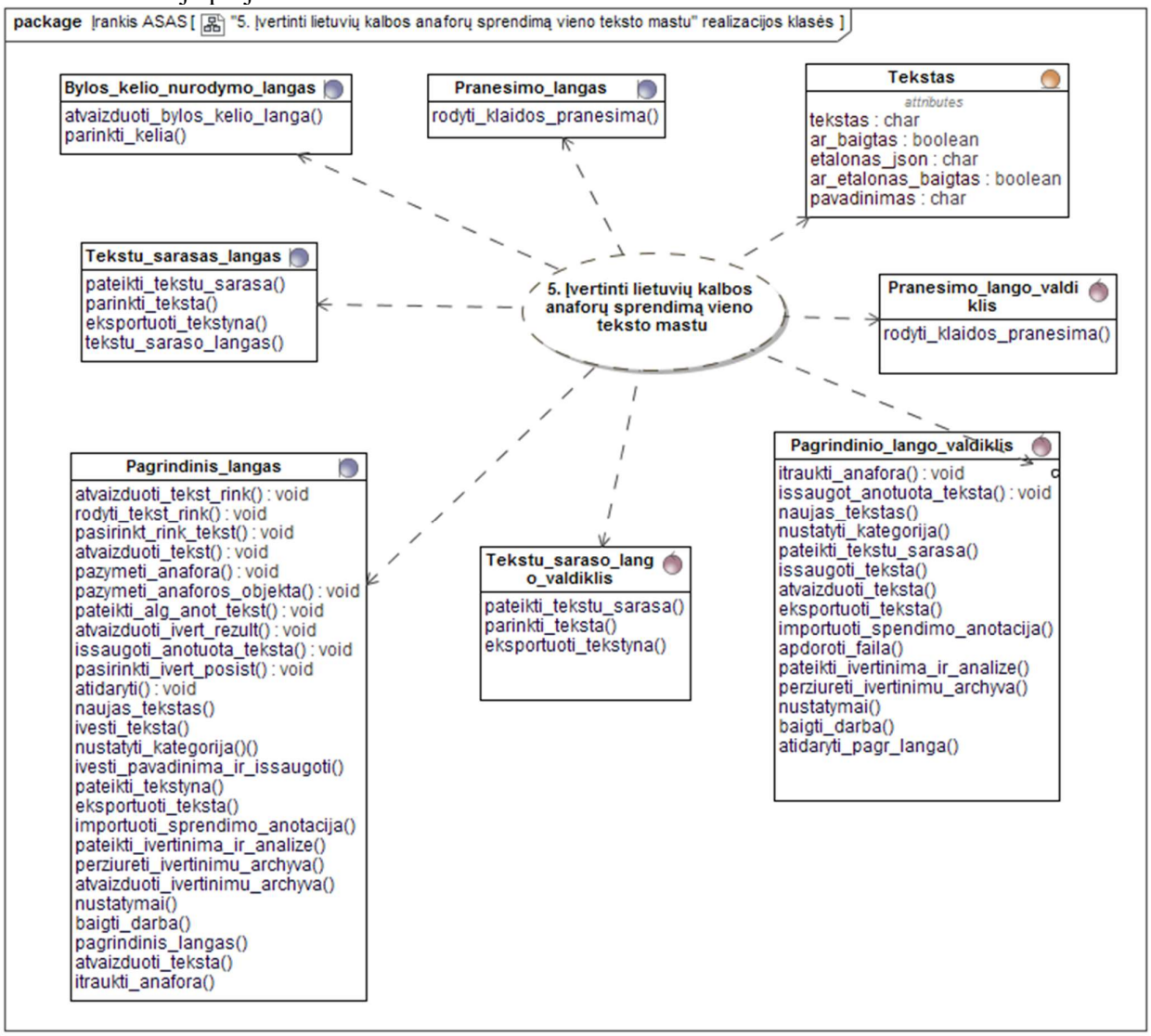
49 pav. PA „3. Sudaryti lietuvių kalbos tekstyno anaforų anotavimo etaloną“ realizacija projekto klasėmis

50 pav. pateikta panaudojimo atvejo „4. Įvertinti lietuvių kalbos anaforų sprendimą viso tekstyno mastu“ realizacija projekto klasėmis.



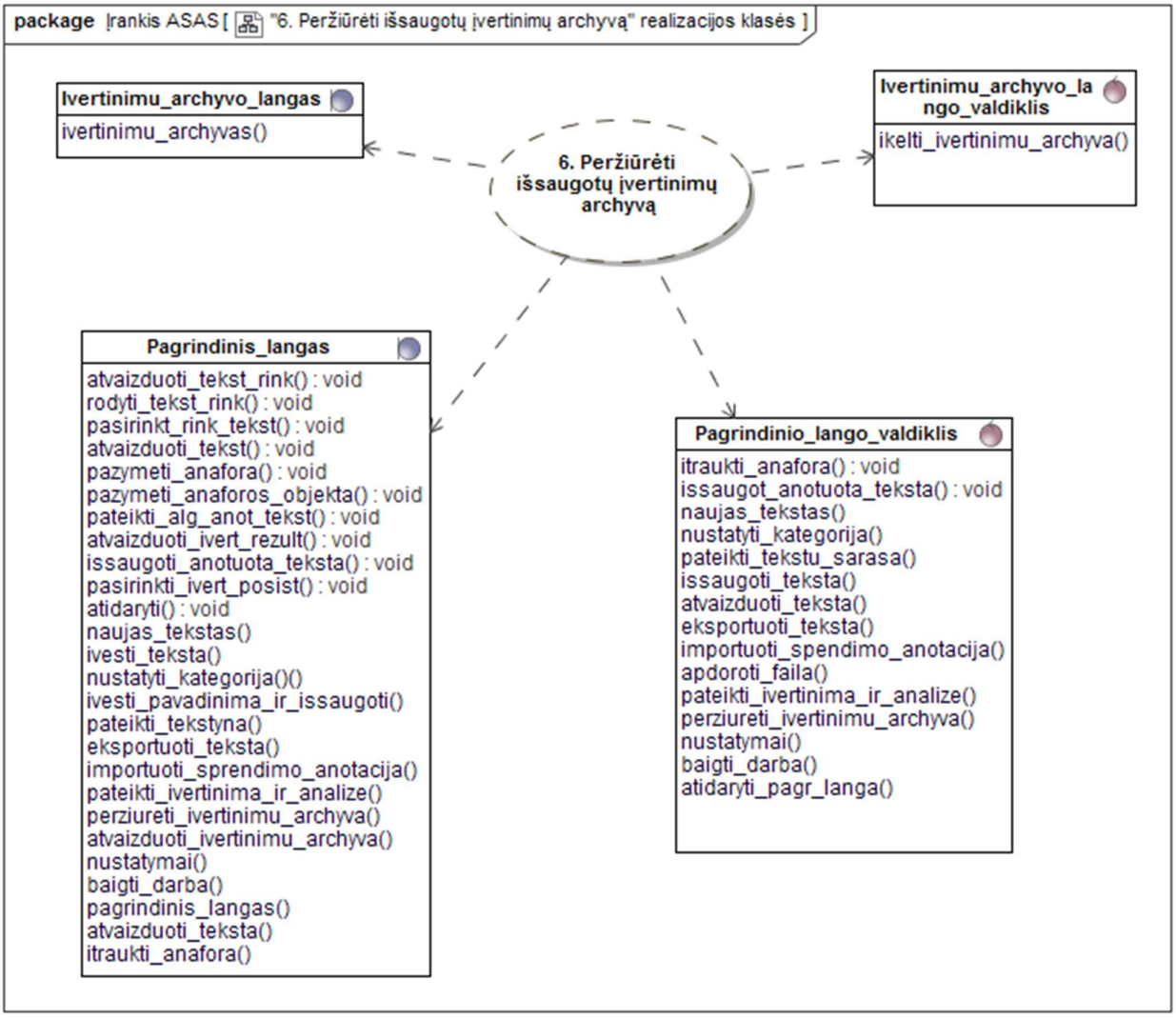
50 pav. PA „4. Įvertinti lietuvių kalbos anaforų sprendimą viso tekstyno mastu“ realizacija projekto klasėmis

51 pav. pateikta panaudojimo atvejo „5. Įvertinti lietuvių kalbos anaforų sprendimą vieno teksto mastu“ realizacija projekto klasėmis.



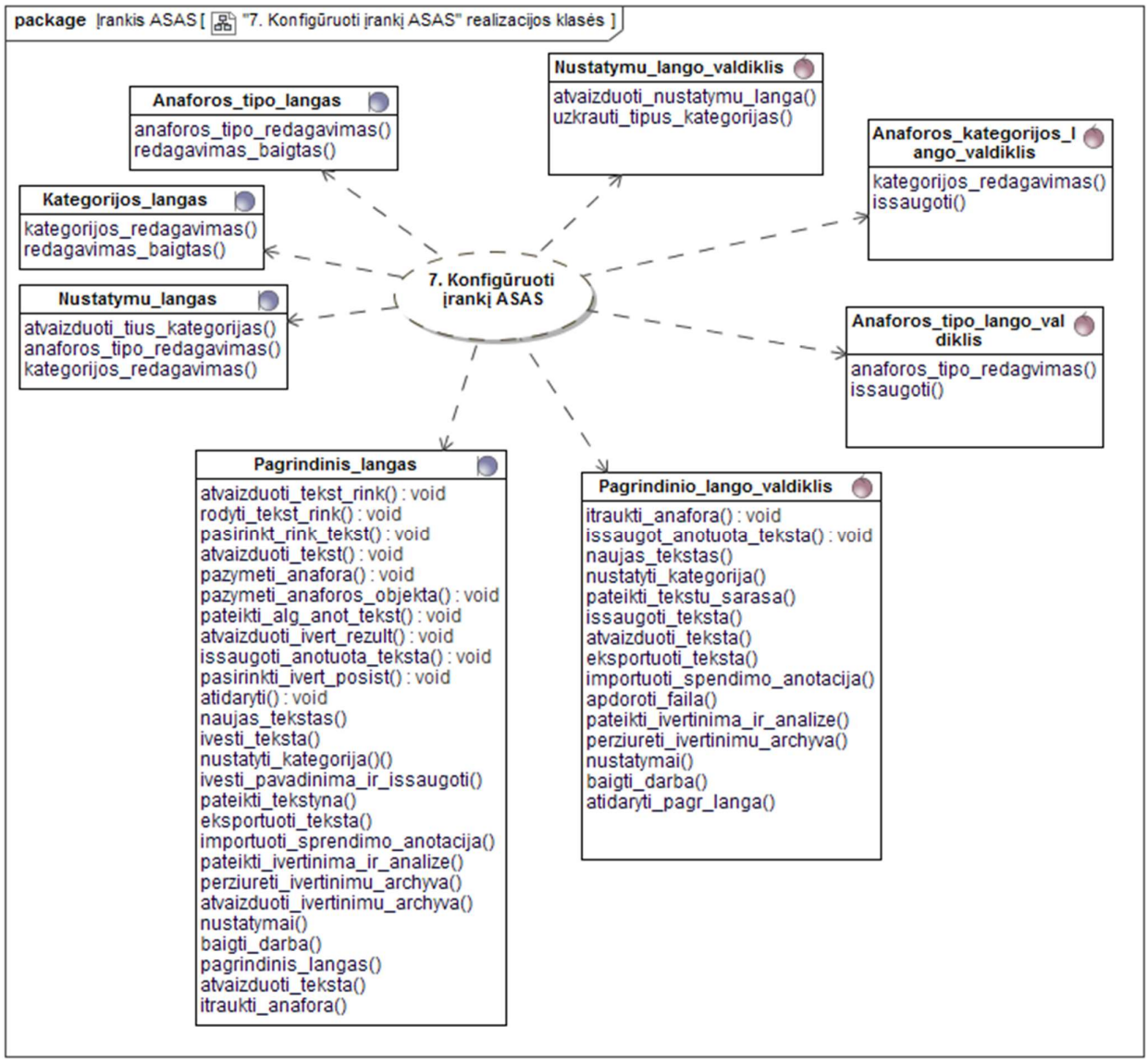
51 pav. PA „5. Įvertinti lietuvių kalbos anaforų sprendimą vieno teksto mastu“ realizacija projekto klasėmis

52 pav. pateikta panaudojimo atvejo „6. Peržiūrėti išsaugotų įvertinimų archyvą“ realizacija projekto klasėmis.



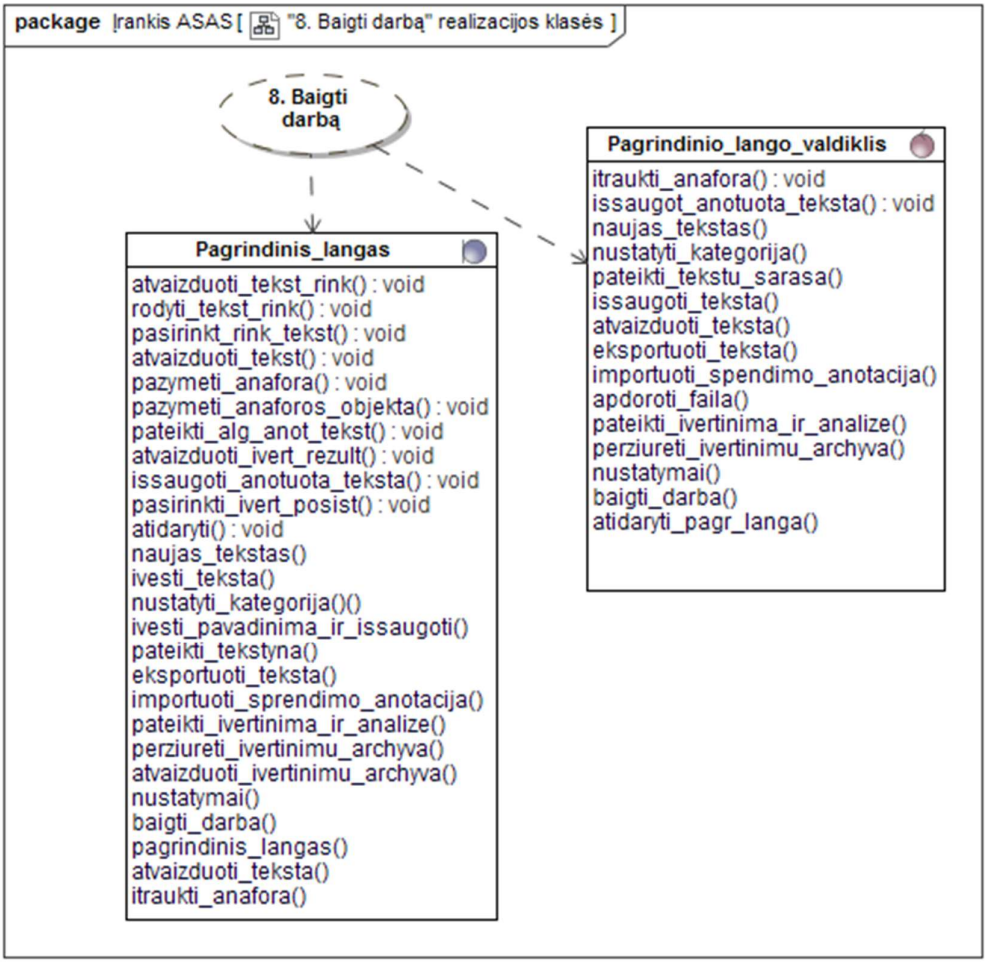
52 pav. PA „6. Peržiūrėti išsaugotų įvertinimų archyvą“ realizacija projekto klasėmis

53 pav. pateikta panaudojimo atvejo „7. Konfigūruoti įrankį ASAS“ realizacija projekto klasėmis.



53 pav. PA „7. Konfigūruoti įrankį ASAS“ realizacija projekto klasėmis

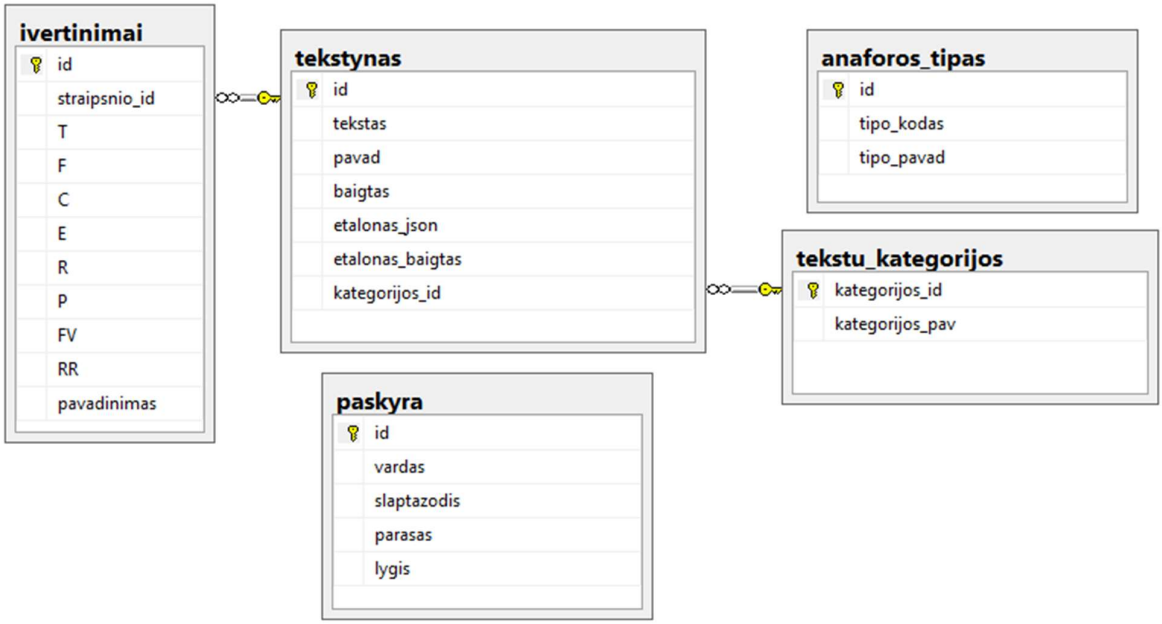
54 pav. pateikta panaudojimo atvejo „8. Baigti darbą“ realizacija projekto klasėmis.



54 pav. PA „8. Baigti darbą“ realizacija projekto klasėmis

3.5. Duomenų bazės schema ir duomenų aprašymas

Duomenų bazės modelis pateikiamas 55 pav. Jame pateikiamos projektuojamos duomenų bazės lentelės, jų elementai ir ryšiai tarp jų.



55 pav. Duomenų bazės modelis

Lentelėse (34 lentelė, 35 lentelė, 36 lentelė, 37 lentelė, 38 lentelė) pateikiama detali duomenų bazės lentelių elementų specifikacija.

34 lentelė. Duomenų lentelės „tekstynas“ duomenų aprašymas

Stulpelio pavadinimas	Duomenų tipas	Gali būti tuščias	Aprašymas
id	int (automatiškai generuojamas įrašo įterpimo metu), unikalus raktas	Ne	Skirta įrašo identifikavimui.
tekstas	nvarchar(max)	Ne	Skirta neanotuotam tekstui saugoti.
baigtas	bit	Ne	Skirta pažymėjimui, kad teksto sudarymas yra baigtas.
etalonas_json	nvarchar(max)	Taip	Skirta saugoti JSON formato etaloninę anotaciją tekstiniu pavidalu. JSON formate detaliai saugomos anaforos: objektai ir pirmtakai, jų vietos tekste bei anaforos ryšio tipas. Saugoma informacija, kurio teksto yra JSON anotacija. Tiek anaforų pirmtakai, tiek objektai yra saugomi dviem sveikais skaičiais, kurių pirmasis reiškia pradžios simbolį tekste, o antrasis – ilgį. Anaforos tipo kodo saugojimas šiuo metu neįgyvendintas, nes nėra anaforų kodų klasifikacijos. Kadangi kodui vieta numatyta, jo vietoje yra saugoma *null* reikšmė. Įrašo pavyzdys: *{"teksto_id":"2","anaforos":[{"pirmtakai":["28,7","35,9"],"objektai":["92,7","99,7"],"tipo_kodas":null}]}*
etalonas_baigtas	bit	Ne	Skirta pažymėti, kad etalono sudarymas yra baigtas.
kategorijos_id	int (išorinis raktas lentelei tekstynu_kategorijos)	Ne	Skirta identifikuoti kategoriją iš tekstynu_kategorijos lentelės.

35 lentelė. Duomenų lentelės „tekstynu_kategorijos“ duomenų aprašymas

Stulpelio pavadinimas	Duomenų tipas	Gali būti tuščias	Aprašymas
kategorijos_id	int (automatiškai generuojamas įrašo įterpimo metu), unikalus raktas	Ne	Skirta identifikuoti įrašui.
kategorijos_pav	nvarchar(max)	Ne	Kategorijos pavadinimas.

36 lentelė. Duomenų lentelės „paskyra“ duomenų aprašymas

Stulpelio pavadinimas	Duomenų tipas	Gali būti tuščias	Aprašymas
id	int (automatiškai generuojamas įrašo įterpimo metu), unikalus raktas	Ne	Skirta identifikuoti įrašui.
Vardas	nvarchar(50)	Ne	Vartotojo prisijungimo vardui.
slaptazodis	nvarchar(MAX)	Ne	Vartotojo prisijungimo slaptažodžiui.
Parasas	nvarchar(MAX)	Ne	Vartotojo tikras vardas ir pavardė.
Lygis	nvarchar(50)	Ne	Vartotojo priėjimo lygis.

37 lentelė. Duomenų lentelės „anaforos_tipas“ duomenų aprašymas

Stulpelio pavadinimas	Duomenų tipas	Gali būti tuščias	Aprašymas
Id	int (automatiškai generuojamas įrašo įterpimo metu), unikalus raktas	Ne	Skirta identifikuoti įrašui.
tipo_kodas	nvarchar(50)	Ne	Anaforos tipo kodas.
tipo_pavad	nvarchar(MAX)	Ne	Anaforos tipo pavadinimas.

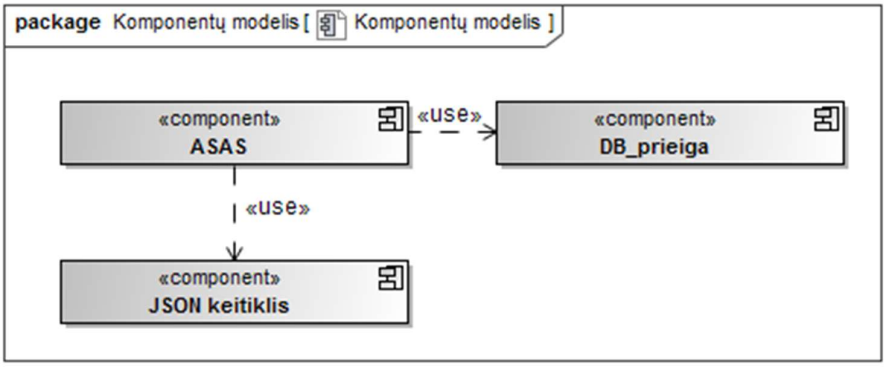
38 lentelė. Duomenų lentelės „ivertinimai“ duomenų aprašymas

Stulpelio pavadinimas	Duomenų tipas	Gali būti tuščias	Aprašymas
id	int (automatiškai generuojamas įrašo įterpimo metu), unikalus raktas	Ne	Skirta identifikuoti įrašui.
straipsnio_id	int	Ne	Straipsnio identifikacinis kodas.
T	nvarchar(50)	Ne	Teisingų anaforų tekste kiekis.
F	nvarchar(50)	Ne	Sprendimo rasta anaforų straipsnyje.
C	nvarchar(50)	Ne	Teisingai automatinio anaforų sprendimo atpažintų anaforų kiekis tekste (teisingų rezultatų kiekis).
E	nvarchar(50)	Ne	Automatinio anaforų sprendimo neatpažintų anaforų kiekis tekste.
R	nvarchar(50)	Ne	Automatinio anaforų sprendimo įvertinimo matas – išsamumas.
P	nvarchar(50)	Ne	Automatinio anaforų sprendimo įvertinimo matas – tikslumas.
FV	nvarchar(50)	Ne	Automatinio anaforų sprendimo įvertinimo matas – F-vertė.
RR	nvarchar(50)	Ne	Automatinio anaforų sprendimo įvertinimo matas – nuorodų dažnumas.
pavadinimas	nvarchar(MAX)	ne	Automatinio anaforų sprendimo pavadinimas.

3.6. Realizacijos modelis

3.6.1. Komponentų modelis

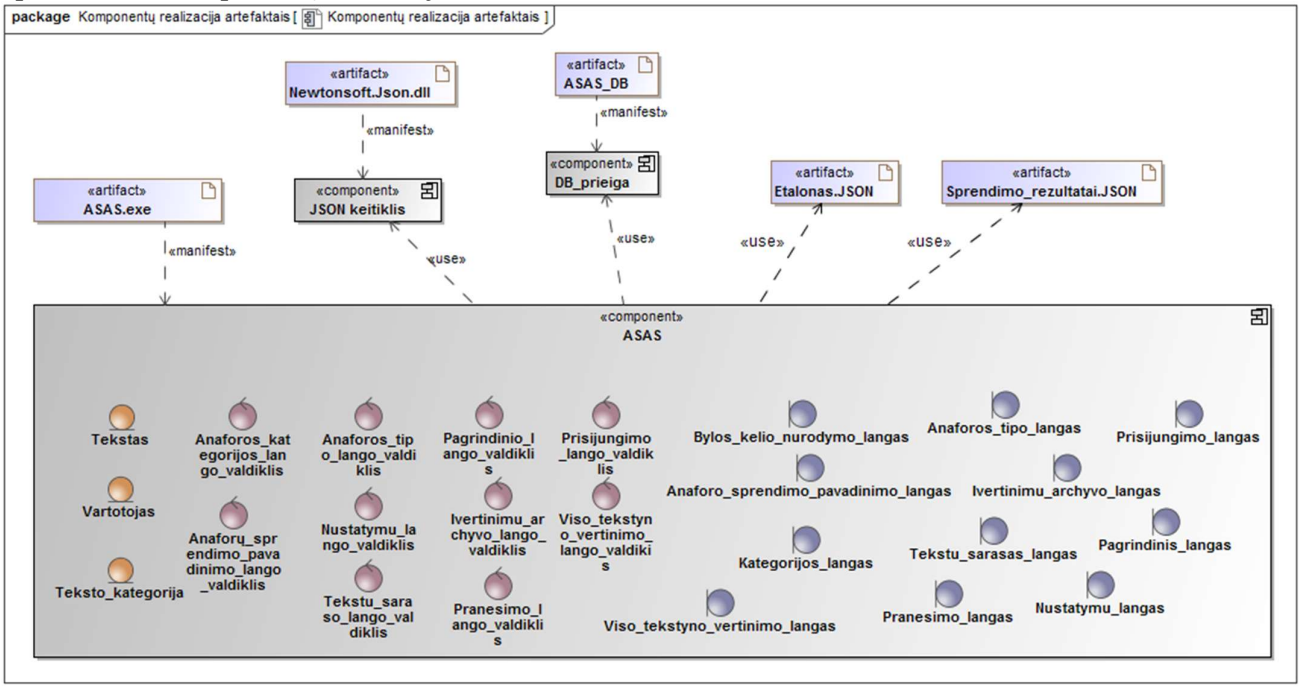
56 pav. pateikiamas įrankio ASAS komponentų modelis.



56 pav. Įrankio ASAS komponentų modelis

3.6.2. Komponentų realizacijos artefaktais modelis

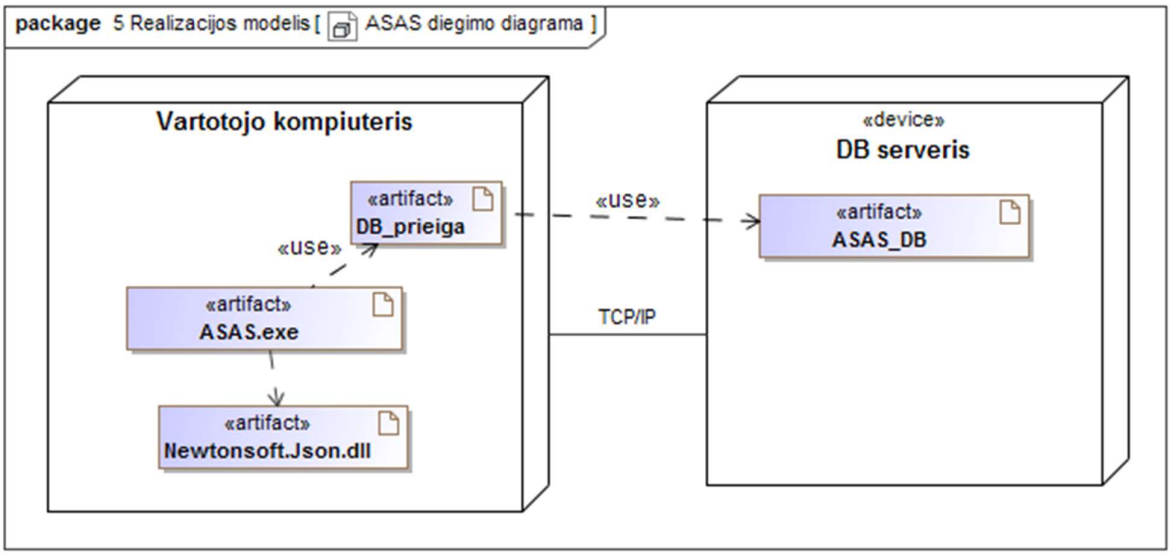
Komponentai realizuojami artefaktais. Artefaktai bus išdėstyti fiziniuose įrenginiuose. 57 pav. pateikiamas komponentų realizacijos artefaktais modelis.



57 pav. Komponentų realizacijos artefaktais modelis

3.6.3. Diegimo modelis

Įrankio ASAS diegimo modelis pateiktas 58 pav.



58 pav. Įrankio ASAS įdiegimo modelis

4. ĮRANKIO ASAS REALIZACIJA IR TESTAVIMAS

Šiame skyriuje aprašomas realizuoto prototipinio įrankio ASAS realizavimo priemonės, diegimo ir veikimo aprašai. Sudaromas testavimo modelis ir atliekamas funkcinių ir nefunkcinių reikalavimų testavimo modelis.

4.1. Realizavimo priemonės

Prototipas ASAS suprogramuotas programavimo kalba *C#* naudojant integruotą kūrimo aplinką „Microsoft Visual Studio Community 2015“. Įrankis internete prieinamas adresu http://asas.netseptyni.lt

ASAS yra išleistas „Microsoft OneClick Deployment“ technologija. Naudojama „Microsoft SQL Server“ duomenų bazė. Įrankio duomenų bazė ir diegimo prieiga yra tarnybinėje stotyje, prie kurios jungiasi klientinė dalis. Klientinė dalis įdiegiama vartotojo kompiuteryje iš diegimo prieigos. Todėl ASAS klientinė dalis gali veikti tik tuomet, kai yra ryšys su tarnybinės stoties ASAS dalimi. Įrankis ASAS, kiekvienąkart prieš pradėdamas darbą (t. y. kai paleidžiama kliento dalis vartotojo kompiuteryje), tikrina, ar nėra išleistas atnaujinimas. Jei yra – automatiškai atsinaujina.

Ši diegimo schema suteikia šiuos privalumus:

I. Serverio dalies apkrova: apkraunamas tik „SQL“ serveris, nes visa programos logika vykdoma vartotojų kompiuteriuose. Tai leidžia sutaupyti resursų techninei įrangai;

II. Jokie duomenys nėra saugomi vartotojų kompiuteriuose – debesų (angl. *Cloud*) technologija. Tai užtikrina prieinamų duomenų vientisumą. Be to, vartotojams nereikia rūpintis dėl atsarginių duomenų kopijų. Naujiems vartotojams prieiga prie duomenų gali būti suteikiama sukuriant paskyrą, įgalinančią naudotis įrankiu;

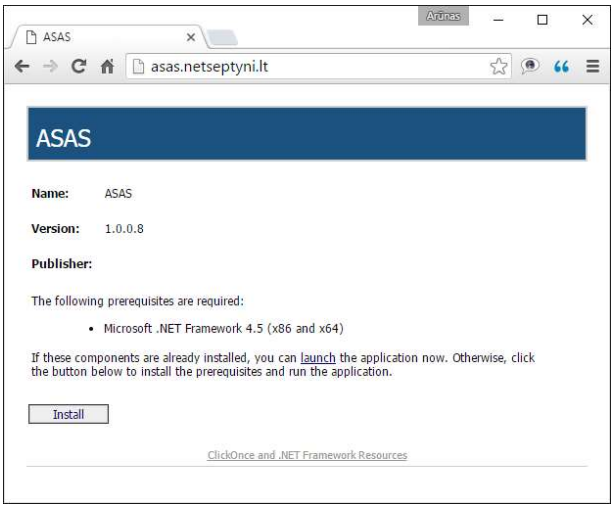
III. Paprastas įrankio diegimas atliekamas vienu paspaudimu.

Šios technologijos pagrindinis trūkumas yra tai, kad įrankis gali būti įdiegtas tik „Windows“ operacinėje sistemoje su įdiegtu ne žemesniu nei *.Net 4.5* karkasu.

Duomenų mainams *JSON* formatu naudojama biblioteka (įskiepis) „JSON.Net 8.0.2“.

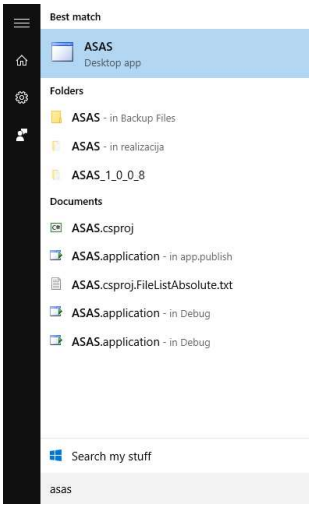
4.2. Diegimo aprašas

Įrankio ASAS įdiegimo byla prieinama naršyklėje, užkrovus internetinį puslapį (59 pav.) asas.netseptyni.lt, kuriame galima matyti naujausią ASAS leidimo versiją ir pasirinkti įdiegti ASAS ir / arba reikiamą .Net karkaso versiją (v4.5).



59 pav. Įrankio ASAS diegimo internetinis puslapis

Įdiegtas įrankis randamas „Windows“ taikomųjų programų sąraše (60 pav.). Lengviausiai surasti galima programų paieškoje įvedus: „ASAS“.

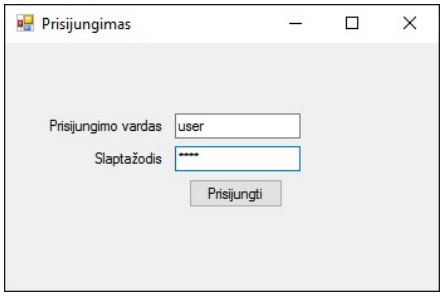


60 pav. ASAS paleidimas „Windows“

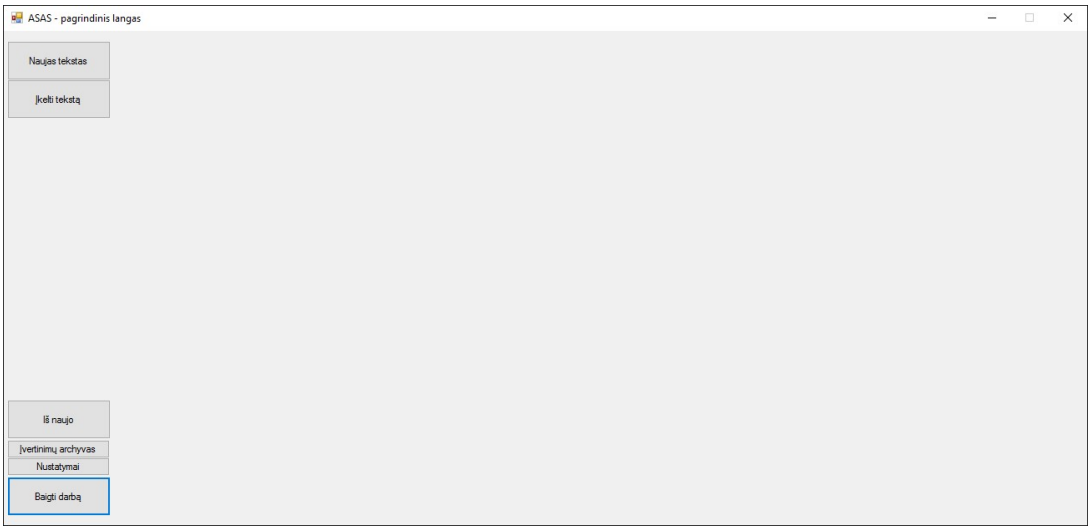
4.3. Veikimo aprašas

Įrankis ASAS, prieš pateikdamas grafinę sąsają vartotojui, pirma jungiasi prie diegimo tarnybinės stoties ir patikrina, ar nėra išleistų atnaujinimų.

Jei ASAS negali užmegzti ryšio su diegimo ir duomenų serveriu, ASAS paleidimas yra nutraukiamas ir parodomas klaidos pranešimas. Jei atnaujinimų yra, ASAS automatiškai atsinaujina naujausiu leidimu. Atsinaujinus ar neradus atnaujinimų, atidaromas autorizacijos langas, kuriame reikia įvesti vartotojo vardą ir slaptažodį. Vartotojas įveda savo prisijungimo vardą, slaptažodį, spaudžia „Prisijungti“ ir pradeda darbą.



61 pav. Prisijungimas prie įrankio ASAS

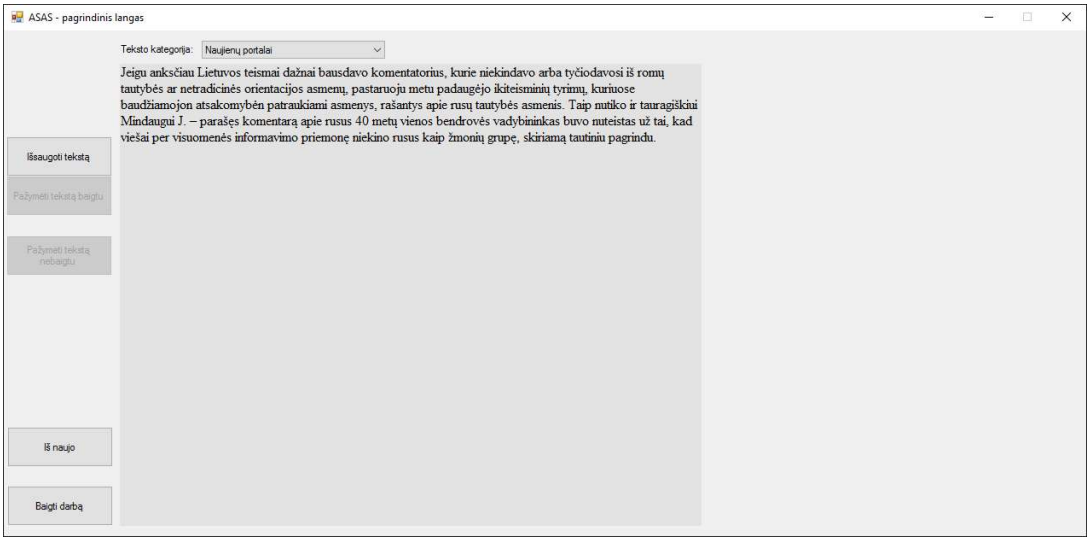


62 pav. Įrankio ASAS pagrindinis langas

Iš pagrindinio įrankio ASAS lango (62 pav.) yra prieinamos šios funkcijos:

- Sukurti naują tekstą ar įkelti ir redaguoti buvusį tekstą iš tekstyno;
- Redaguojant buvusį tekstą galima pakeisti jo kategoriją;
- Paspaudus mygtuką „Išsaugoti“, sukurtas tekstas yra pridedamas į tekstyną ir parodomas informacinis pranešimas. Jei redaguojamas senas tekstas – išsaugomi pakeitimai;
- Baigus teksto sudarymą, galima jį pažymėti „baigtu“. Teksto būseną vėl pakeitus į „nebaigtas“, bus pašalinta etaloninė anotacija, jei tokia buvo sudaryta. Todėl vartotojas įkėlęs buvusį tekstą, kuris yra „baigtas“, to teksto redaguoti negali;
- Tik pažymėjus tekstą „baigtu“ galima sudaryti to teksto etaloną;
- Teksto etalonas sudaromas pasirinkus „Sudaryti etaloną“;
- Jei teksto etalonas buvo pradėtas daryti, etaloną galima įkelti;
- Anaforos žymimos pasirinkus „Nauja anafora“ ir pele tekste pažymėjus jos objektus ir pirmtakus, nurodžius tipą ir paspaudus „Išsaugoti anaforą“;
- Etaloninei anotacijai nustačius žymą „Baigta“, galima eksportuoti tekstą, kurio turimas etalonas – JSON formato byla;
- Galima eksportuoti JSON etaloninę anotaciją bei importuoti anaforų sprendimo JSON anotaciją;
- Užkrovus sprendimo anotaciją, pateikiama įvertinimo charakteristika ir grafinė analizė;
- Nustatymuose galima sukurti ar redaguoti anaforų tipus ir tekstų kategorijas.

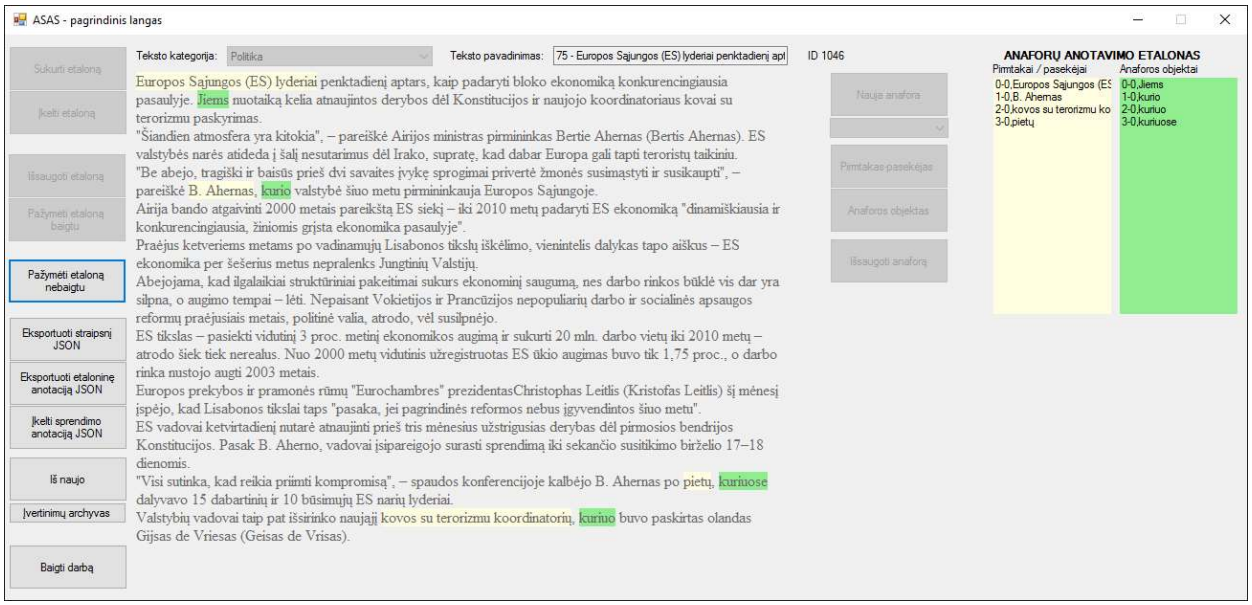
4.3.1. Tekstyno sudarymas



63 pav. Tekstyno sudarymas įrankiu ASAS

1. Vartotojas turi būti prisijungęs prie sistemos;
2. Pagrindiniame lange pasirenkama „Sukurti naują tekstą“ arba „Įkelti esamą tekstą iš tekstyno“;
3. Pasirinkus naujo teksto kūrimą, rodomas tuščias teksto įvedimo langas ir kategorijos priskyrimo meniu su neparinkta kategorija. Įkėlus jau esamą tekstą, lange rodomas įkeltas tekstas, o meniu – esama priskirta kategorija. Tol, kol tekstui nėra uždėta žyma „baigtas“, laikoma, kad tekstas yra nesukurtas ir dar gali būti redaguojamas: papildomas, taisomas, keičiama kategorija, bet negali būti pradedamas anaforų žymėjimas (anotavimo etalono sudarymas);
4. Tekstą papildyti galima dviem būdais: įvesti tekstą rankiniu būdu arba įklijuoti klavišų kombinacija *Ctrl + V*;
5. Sudarant naują tekstą arba toliau tęsiant jau sukurto teksto redagavimą, tekstas nėra išsaugomas tol, kol nėra pasirenkama „Išsaugoti“. Tekstą išsaugant pirmąkart (kuriant naują tekstą), privalo būti parinkta teksto kategorija. Pirmo išsaugojimo metu tekstas yra pridedamas į tekstyną. Vėliau teksto kategorija gali būti pakeista. Tekstyno sudarymo pavyzdys pateikiamas 63 pav.

4.3.2. Etalono sudarymas

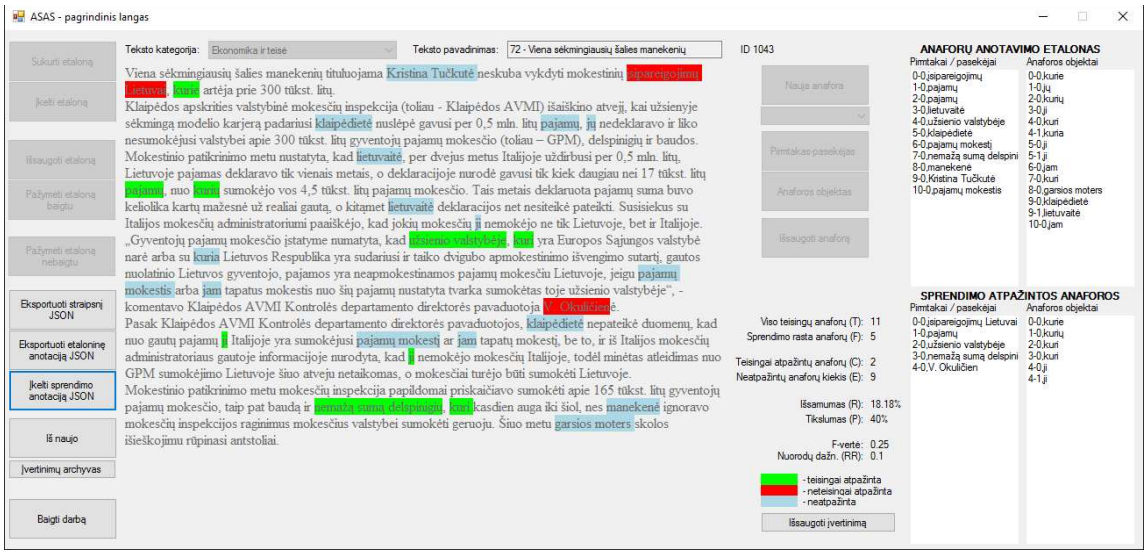


64 pav. Etalono sudarymas įrankiu ASAS

1. Vartotojas turi būti prisijungęs prie sistemos;
2. Pagrindiniame lange pasirenkama „Įkelti esamą tekstą iš tekstyno“;
3. Pagrindiniame lange atsiranda galimybė įkelti etaloną, jei tekstas turi žymą „baigtas“.
4. Įkėlus etaloną, sąrašuose rodomos tekste sužymėtos anaforos;
5. Kol etalonui nėra priskirta žyma „baigtas“, etalono sudarymas gali būti pratęstas – trinamos jau pažymėtos anaforos ar pažymimos naujos;
6. Sudarant etaloną, anaforų žymėjimas ar korekcijos nėra išsaugomos tol, kol neparenkama „Išsaugoti“.

4.3.3. Anaforų sprendimo įvertinimas

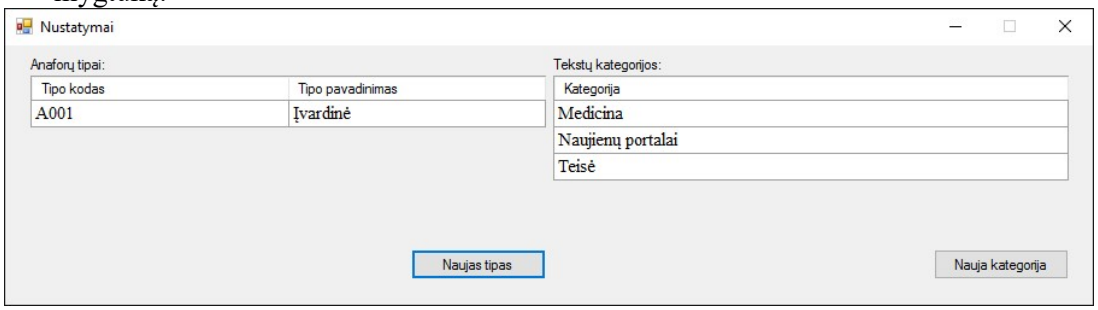
1. Vartotojas turi būti prisijungęs prie sistemos;
2. Pagrindiniame lange pasirenkama „Įkelti esamą tekstą iš tekstyno“;
3. Pagrindiniame lange atsiranda galimybė įkelti etaloną, jei tekstas turi žymą „baigtas“;
4. Jei etalonas turi žymą „baigtas“, galima eksportuoti neanotuotą tekstą tekstinės bylos pavidalu. Eksportuotas tekstas skirtas anaforų sprendimo sistemai anotuoti;
5. Anaforų sprendimo įvertinimas pateikiamas importavus anaforų sprendimo anotaciją. Laikoma, kad anafora sprendimo buvo atpažinta, jei buvo atpažinti visi pirmtakai.



65 pav. Anaforų sprendimo įvertinimas vieno teksto mastu įrankiu ASAS

4.3.4. Įrankio ASAS parametrų konfigūravimas

1. Vartotojas turi būti prisijungęs prie sistemos;
2. Pasirenkama „Nustatymai“;
3. Atsidariusiame lange (66 pav.) galima pridėti, ištrinti ar redaguoti esamus anaforų tipus, anaforų kodus ir tekstų kategorijas. Kontekstinis meniu iškviečiamas paspaudus dešinį pelės mygtuką.



66 pav. Įrankio ASAS parametrų konfigūravimas

4.4. Testavimo modelis, duomenys, rezultatai

Prototipinio įrankio ASAS testavimo metu buvo siekiama nustatyti jo klaidas bei atitikimą funkciniams ir nefunkciniams reikalavimams. Testavimo atvejus galime skirstyti į 2 dalis: funkciniai testavimo atvejai (funkciniams reikalavimams patikrinti) ir nefunkciniai testavimo atvejai (nefunkciniams reikalavimams patikrinti). Visiems testavimo atvejams buvo atliekami „juodos dėžės“ testai. Nefunkciniai testavimo atvejų testų rezultatai buvo gaunami apklausos / stebėjimų metodais. Testavimui buvo sudarytas testavimo modelis, kuris buvo taikomas visame testavimo etape. Testavimo rezultatai buvo fiksuojami lentelėje (39 lentelė). Testavimo metu užfiksavus nelauktą rezultatą buvo ištaisomos klaidos ir testas pakartojamas. Testai buvo kartojami, kol būdavo gautas laukiamas rezultatas.

Testavimo atvejai (funkciniai testai):

1. Sistemos naudotojo autentifikavimo testavimas.
2. Lietuvių kalbos tekstyno sudarymo testavimas;
3. Lietuvių kalbos tekstyno redagavimo testavimas;
4. Lietuvių kalbos anaforų anotavimo etalono sudarymo testavimas;
5. Lietuvių kalbos anaforų anotavimo etalono redagavimo testavimas;
6. Tekstyno eksportavimo JSON formatu testavimas;
7. Anaforų sprendimo įvertinimo viso tekstyno mastu testavimas;

8. Vieno tekstyno teksto eksportavimo JSON formatu testavimas;
9. Anaforų sprendimo įvertinimo vieno teksto mastu testavimas;
10. Įvertinimų archyvo peržiūros testavimas;
11. Konfigūravimo testavimas;
12. Įrankio išjungimo testavimas.

Testavimo atvejai (nefunkciniai testai):

13. Ar įrankis atrodo solidžiai;
14. Ar patogus anaforų anotavimo procesas;
15. Ar įrankiu ASAS paprasta išmokti naudotis;
16. Ar įrankis naudoja simbolius ir žodžius natūraliai suprantamus žmonėms dirbantiems NKA srityje;
17. Ar greitai galima paleisti įrankį ir atlikti kiekvieną funkciją;
18. Ar įrankis veikia greitai;
19. Ar apvalinamos pateikiamos reikšmės.

39 lentelė. Testavimo modelis

Testavimo atvejis / Testuojami PA	**Testavimo aprašymas**	**Laukiamas rezultatas**	**Testai ir jų rezultatai**
1 / 1.	Prisijungti prie sistemos 3 skirtingomis paskyromis.	Sėkmingas prisijungimas ir pagrindinio lango atidarymas.	1. Blogai įvestas slaptažodis. 2. Gautas laukiamas rezultatas.
2 / 2. 2.1 2.1.1 2.1.2	Sudaryti lietuvių kalbos tekstyną: sudaryti kelis naujus tekstus tiek įvedant, tiek įklijuojant. Tekstams priskirti kategoriją, pavadinimus. Nustatyti tekstų užbaigtumo žymas „baigtas“. Paleisti įrankį ASAS iš naujo ir patikrinti, ar tekstai yra ir tekstynas sudarytas.	Sudarytas tekstynas susidedantis iš kelių tekstų.	1. Tekstai neišsaugoti. 2. Gautas laukiamas rezultatas.
3 / 2. 2.2 2.1.1 2.1.2	Nustatyti tekstyno tekstų užbaigtumo žymas „nebaigtas“. Redaguoti tekstus. Nustatyti tekstyno tekstų užbaigtumo žymas „baigtas“. Paleisti įrankį ASAS iš naujo ir patikrinti, ar korekcijos išsaugotos.	Atlikta tekstų iš tekstyno korekcija: pakeistas tekstas, pakeista kategorija, pakeistas pavadinimas.	1. Nepavyko nustatyti užbaigtumo žymos „baigtas“. 2. Gautas laukiamas rezultatas.
4 / 3. 3.1 3.1.1	Sudaryti lietuvių kalbos tekstyno anaforų anotavimo etaloną. Sudaryti visų tekstyno tekstų etaloną jose sužymint anaforas. Nustatyti tekstų užbaigtumo žymas „baigtas“. Paleisti įrankį ASAS iš naujo ir patikrinti, ar etalonai išsaugoti.	Sudarytas lietuvių kalbos tekstyno anaforų anotavimo etalonas.	1. Gautas laukiamas rezultatas.
5 / 3. 3.2. 3.1.1.	Nustatyti visų tekstyno tekstų etalonų užbaigtumo žymas „nebaigtas“. Redaguoti etalonus. Nustatyti visų tekstyno tekstų etalono užbaigtumo žymas „baigtas“. Paleisti įrankį ASAS iš naujo ir patikrinti, ar korekcijos išsaugotos.	Atlikta visų tekstyno tekstų korekcija.	1. Gautas laukiamas rezultatas.

6 / 4.1.	Eksportuoti visą tekstyną JSON formatu. Peržiūrėti eksportuotas bylas, ar korektiškas turinys.	Eksportuotas tekstynas JSON formato bylomis (1 failas – 1 tekstas), kurių turinys – teksto ID, tekstas.	1. Eksportuotos tuščios JSON formatų bylos. 2. Gautas laukiamas rezultatas.
7 / 4. 4.2. 4.2.1.	Įvertinti imitacinius lietuvių kalbos anaforų sprendimo rezultatus. Importuoti viso tekstyno anaforų sprendimo imitacinius rezultatus JSON formatu. Patikrinti ar pateikti teisingi įvertinimo kriterijai.	Pateikti tekstyno įvertinimo kriterijai ir tarpiniai skaičiavimo rezultatai. Pateikti rezultatai: T, F, C, E, R, P, F-vertė, RR.	1. Pateikti blogi įvertinimo kriterijai ir tarpiniai skaičiavimo rezultatai. 2. Importuojant rezultatus - sisteminė klaida. 3. Gautas laukiamas rezultatas.
8 / 5.1.	Eksportuoti vieną tekstyno tekstą JSON formatu. Patikrinti eksportuotą bylą, ar korektiškas turinys.	Eksportuotas tekstas JSON formato byla, kurio turinys – teksto ID ir tekstas.	1. Gautas laukiamas rezultatas.
9 / 5. 5.2. 5.2.1. 5.2.2.1.	Įvertinti lietuvių kalbos anaforų sprendimą vieno teksto mastu. Importuoti 1 teksto imitacinius anaforų sprendimo sistemos rezultatus. Gauti įvertinimo kriterijus ir grafinę klaidų analizę. Patikrinti, ar pateikti teisingi įvertinimo kriterijai ir grafinė klaidų analizė. Išsaugoti į archyvą anaforų sprendimo įvertinimo kriterijus.	Pateikti tekstyno įvertinimo kriterijai ir tarpiniai skaičiavimo rezultatai. Pateiktos reikšmės: T, F, C, E, R, P, F-vertė, RR. Pateikta grafinė klaidų analizė žaliai pažymint teisingai aptiktas anaforas, raudonai – neteisingai, o mėlynai – neatpažintas anaforas.	1. Visas tekstas pažymėtas viena spalva – bloga grafinė analizė. 2. Gautas laukiamas rezultatas.
10 / 6.	Peržiūrėti išsaugotą įvertinimų archyvą. Patikrinti, ar atvaizduojama korektiška ir pilna informacija.	Pateiktas tekstyno tekstų įvertinimų sąrašas. Pateikti rezultatai: T, F, C, E, R, P, F-vertė, RR. Prie kiekvienų rezultatų pateiktas atitinkamas teksto ID ir anaforų sprendimo pavadinimas.	1. Gautas laukiamas rezultatas.
11 / 7. 7.1. 7.2.	Konfigūruoti įrankį ASAS. Sukurti naujas teksto kategorijas ir anaforų tipus / kodus. Tuomet jas pakoreguoti. Perkrauti ASAS ir patikrinti, ar gauti teisingi rezultatai.	Sukurtos naujos kategorijos, kurias galima pasirinkti redaguojant ar kuriant tekstą. Sukurti nauji anaforų tipai, kuriuos galima pasirinkti žymint anaforas.	1. Neišsaugotos nei kategorijų nei anaforų korekcijos. 2. Gautas laukiamas rezultatas.
12 / 8.	Paleisti kiekvieną įrankio ASAS langą ir iš jo baigti darbą.	Kiekvienąkart išjungtas įrankis ASAS.	1. Gautas laukiamas rezultatas.
13 / Visi	5 asmenų apklausa po 60 min naudojimosi įrankiu – ar įrankis atrodo solidžiai.	Turi patvirtinti daugiau nei 90% naudotojų.	1. Gautas laukiamas rezultatas.
14 / 3. 3.1 3.2	5 asmenų apklausa po 60 min naudojimosi įrankiu – ar patogus anaforų anotavimo procesas.	Turi patvirtinti daugiau nei 90% naudotojų.	1. Gautas laukiamas rezultatas.

15 / Visi	5 asmenų apklausa po 20 min naudojimosi įrankiu – ar pavyko išmokti naudotis.	Turi patvirtinti daugiau nei 90% naudotojų.	1. Gautas laukiamas rezultatas.
16 / Visi	5 asmenų apklausa po 60 min naudojimosi įrankiu – ar suprantami visi pateikiami terminai.	Turi patvirtinti daugiau nei 90% naudotojų.	1. Gautas laukiamas rezultatas.
17 / Visi	Ar greitai galima paleisti įrankį ir atlikti kiekvieną funkciją (panaudojamumas).	Turi būti įmanoma per ne ilgiau nei 5 sekundes.	1. Gautas laukiamas rezultatas.
18 / Visi	Ar įrankis veikia greitai.	Kiekvieno lango atidarymas / funkcijos atlikimas turi trukti ne ilgiau nei 2 sekundes.	1. Gautas laukiamas rezultatas.
19 / 4. 4.2.1 5.2.1 6.	Ar apvalinamos pateikiamos reikšmės.	Visos naudojamos / pateikiamos anaforų sprendimo įvertinimo reikšmės ir tarpiniai rezultatai turi būti apvalinami vienos dešimt tūkstantosios tikslumu, o procentinės išraiškos – vienos šimtosios tikslumu.	1. Gautas laukiamas rezultatas.

5. ANAFORŲ SPRENDIMO ĮVERTINIMO IR ANALIZĖS ĮRANKIU ASAS EKSPERIMENTINIS TYRIMAS

Šiame skyriuje aprašomas eksperimentinis tyrimas, kuriuo yra įvertinamas lietuvių kalbos anaforų sprendimas. Įvertinimas atliekamas naudojantis realizuotu prototipiniu įrankiu ASAS pagal sudarytą įvertinimo metodiką.

5.1. Eksperimento apibrėžimas

Eksperimentas skirstomas į 2 etapus.

Pirmame etape atliekamas pavyzdinės sistemos lygio tyrimas. Atliekamas eksperimentas su ypač mažos apimties imitaciniais duomenimis, kurio tikslas – įsitikinti, kad su imitaciniais mažais duomenų kiekiais ir prototipiniu įrankiu ASAS galima įvertinti lietuvių kalbos anaforų sprendimą pagal įvertinimo metodiką.

Antrame etape atliekamas pramoninio lygio realaus atvejo tyrimas. Eksperimento metu yra numatyta atlikti realaus lietuvių kalbos anaforų sprendimo įvertinimą [21]. Numatytasis lietuvių kalbos anaforų sprendimas yra KTU Informacinių sistemų katedroje kuriamas anaforų sprendimas. Eksperimentą, kurio tikslas – įsitikinti, kad realizuotas įrankis ASAS sudaro galimybę atlikti anaforų sprendimo įvertinimą realioje aplinkoje, sudarys 3 dalys.

5.2. Eksperimento planas

5.2.1. I etapas.

Prototipinio įrankio ASAS programuotojo bus sukurtas lietuviško teksto tekstynas iš 1 teksto, kurį sudarys 4 sakiniai. Sukurtam tekstynui bus sudaromas imitacinis anaforų anotavimo etaloninis tekstynas (nebūtinai teisingas, kadangi programuotojas nėra lietuvių kalbos specialistas). Bus sudaryta imitacinė anaforų sprendimo sistemos rezultatų JSON formato byla, kuri įrankyje ASAS bus importuota ir atliktas imitacinių rezultatų įvertinimas bei klaidų analizė. Šiame etape eksperimento subjektas – programuotojas, stebimas objektas – lietuvių kalbos anaforų sprendimo sistemos rezultatų įvertinimo procesas ir analizė su imitaciniais duomenimis, visas roles atliekant vienam ASAS naudotojui. Nepriklausomi kintamieji – prototipinis įrankis ASAS ir klaidų grupės (funkcinių reikalavimų, techninės ir programavimo klaidos). Priklausomas kintamasis – rezultato pasiekimo galimybė. Šiuo etapu siekiama patvirtinti, kad naudojantis prototipiniu įrankiu ASAS yra galimybė įvertinti lietuvių kalbos anaforų sprendimą ir atlikti klaidų analizę dirbant su imitaciniais duomenimis pavyzdinėje aplinkoje.

5.2.2. II etapas.

Antro eksperimento etapo metu jo dalyviai fiksuojamas klaidas pagal grupes (funkcinių reikalavimų, techninės ir programavimo klaidos) registruos specializuotoje klaidų registravimo sistemoje „MantisBT“, įdiegtoje ir pasiekiamoje internete adresu http://tasks.netseptyni.lt/. Kiekvienam eksperimento dalyviui bus sukurta paskyra. Antras etapas suskirstytas į 3 dalis.

1 dalis. Naudojantis ASAS bus sudaromas realus lietuvių kalbos tekstynas ir jo anaforų anotavimo etalonas (etaloninis tekstynas). Šią užduotį turi atlikti anaforų analitikas, kurio specializacija yra lietuvių kalba. Todėl šį darbą atliks KTU socialinių, humanitarinių mokslų ir menų fakulteto bakalauro studijų studentė Laura Klusaitė. Šioje dalyje eksperimento subjektas – anaforų analitikė, stebimas objektas – lietuvių kalbos tekstyno ir jo anaforų anotavimo etalono sudarymo procesas. Nepriklausomi kintamieji – prototipinis įrankis ASAS, anaforų analitikės patirtis ir fiksuojamos klaidų grupės (funkcinių reikalavimų, techninės ir programavimo klaidos). Priklausomas kintamasis – rezultato pasiekimo galimybė. Šia dalimi siekiama patvirtinti, kad naudojantis prototipiniu įrankiu ASAS yra galimybė sudaryti lietuvių kalbos tekstyną ir jo anaforų anotavimo etaloną realioje aplinkoje.

2 dalis. Naudojantis ASAS bus eksportuojamas pirmoje dalyje sudarytas lietuvių kalbos tekstynas JSON formatu, realios anaforų sprendimo sistemos automatiniam anaforų anotavimui. Anaforų sprendimų kūrėjui tekstynas bus pateiktas JSON formato bylomis. Šios dalies eksperimento subjektas – anaforų sprendimo kūrėjas, stebimas objektas – sudaryto lietuvių kalbos tekstyno gavimo

JSON formatu procesas. Nepriklausomi kintamieji – prototipinis įrankis ASAS ir fiksuojamos klaidų grupės (funkcinių reikalavimų, techninės ir programavimo klaidos). Priklausomas kintamasis – rezultato pasiekimo galimybė. Šia dalimi siekiama patvirtinti, kad naudojantis prototipiniu įrankiu ASAS yra galimybė gauti sudarytą lietuvių kalbos tekstyną ir jo anaforų anotavimo etaloną JSON formatu realioje aplinkoje.

3 dalis. Naudojantis įrankiu ASAS bus įvertinamas realus lietuvių kalbos anaforų sprendimas ir atliekama klaidų analizė. Realaus automatinės anaforų sprendimo sistemos anaforų anotavimo rezultatai bus pateikti tekstinėje byloje JSON formatu ir importuoti įrankyje ASAS jų įvertinimui ir analizei. Šioje dalyje eksperimento subjektas – anaforų sprendimo kūrėjas, stebimas objektas – lietuvių kalbos anaforų sprendimo įvertinimo ir analizės procesas. Priklausomas kintamasis – rezultato pasiekimo galimybė. Šia dalimi siekiama patvirtinti, kad naudojantis prototipiniu įrankiu ASAS yra galimybė įvertinti lietuvių kalbos anaforų sprendimą ir atlikti klaidų analizę realioje aplinkoje bei įsitikinti, kad dirbant su skirtingais tekstais gaunami skirtingi įvertinimo rezultatai.

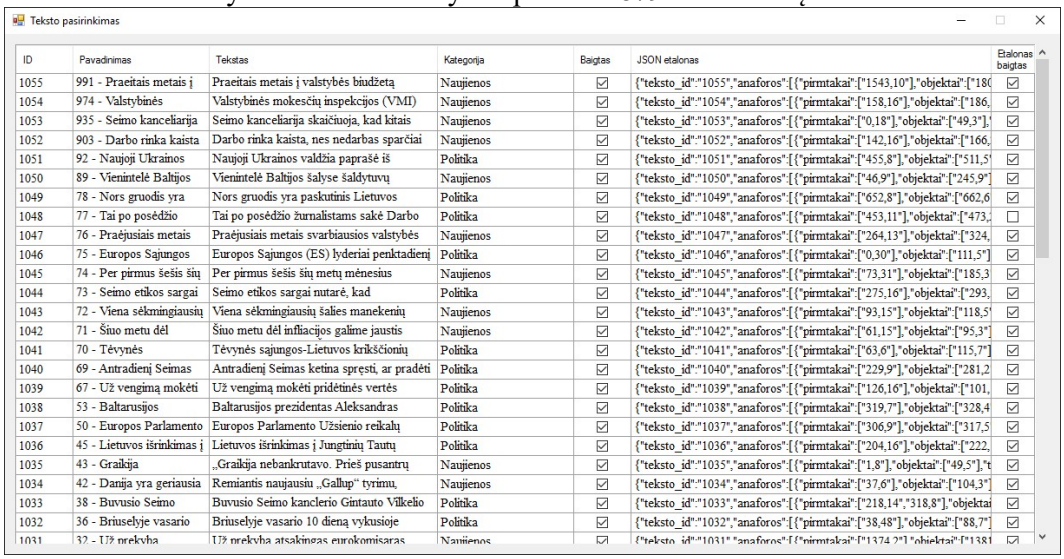
5.3. Eksperimento vykdymas

5.3.1. I etapas.

Pirmame eksperimento vykdymo etape buvo sudarytas imitacinis tekstynas iš 3 sakinių ir sudarytas jo imitacinis etalonas. Tekstynas buvo eksportuotas JSON formato failu. Taip pat buvo sudaryti imitaciniai automatinės anaforų sprendimo sistemos rezultatai JSON formato faile. Imitaciniai rezultatai buvo importuoti įrankyje ASAS. Buvo gautos imitacinės įvertinimo charakteristikos ir grafinė klaidų analizė.

5.3.2. II etapas.

1 dalis. Buvo sudarytas lietuvių kalbos tekstynas įrankiu ASAS. Tekstynas sudarytas iš 39 atrinktų 3 skirtingų kategorijų straipsnių iš Vytauto Didžiojo universiteto (VDU) ir Kauno technologijos universiteto (KTU) projekto „Lietuvių kalbos sintaksinės-semantinės analizės sistema tekstynui, lietuviškam internetui ir viešojo sektoriaus taikymams“ (VP2-3.1-IVPK-12-K-01-007) sudarytų ekonomikos ir teisės bei politikos tekstynų. Viso sudarytą tekstyną sudaro 21 straipsnis iš priskirtų ekonomikos ir teisės kategorijai ir 18 - politikos. Sudarius tekstyną buvo sužymėtos 3 tipų anaforos: veiksmažodinės, daiktavardinės, asmens. Tokiu būdų buvo sudarytas lietuviško teksto anaforų anotavimo etaloninis tekstynas. Bendra tekstyno apimtis – 87914 simbolių.



ID	Pavadinimas	Tekstas	Kategorija	Baigtas	JSON etalonas	Etalonas baigtas
1055	991 - Praeitais metais į	Praeitais metais į valstybės biudžetą	Naujienos	☑	{"teksto_id":"1055","anaforos":[{"pirmtakai":["1543,10"],"objektai":["180	☑
1054	974 - Valstybinės	Valstybinės mokesčių inspekcijos (VMI)	Naujienos	☑	{"teksto_id":"1054","anaforos":[{"pirmtakai":["158,16"],"objektai":["186,	☑
1053	935 - Seimo kanceliarija	Seimo kanceliarija skaičiuoja, kad kitais	Naujienos	☑	{"teksto_id":"1053","anaforos":[{"pirmtakai":["0,18"],"objektai":["49,3"],	☑
1052	903 - Darbo rinka kaista	Darbo rinka kaista, nes nedarbas sparčiai	Naujienos	☑	{"teksto_id":"1052","anaforos":[{"pirmtakai":["142,16"],"objektai":["166,	☑
1051	92 - Naujoji Ukrainos	Naujoji Ukrainos valdžia paprašė iš	Politika	☑	{"teksto_id":"1051","anaforos":[{"pirmtakai":["455,8"],"objektai":["511,5	☑
1050	89 - Vienintelė Baltijos	Vienintelė Baltijos šalyse šaldytuvų	Naujienos	☑	{"teksto_id":"1050","anaforos":[{"pirmtakai":["46,9"],"objektai":["245,9"	☑
1049	78 - Nors gruodis yra	Nors gruodis yra paskutinis Lietuvos	Politika	☑	{"teksto_id":"1049","anaforos":[{"pirmtakai":["652,8"],"objektai":["662,6	☑
1048	77 - Tai po posėdžio	Tai po posėdžio žurnalistams sakė Darbo	Politika	☑	{"teksto_id":"1048","anaforos":[{"pirmtakai":["453,11"],"objektai":["473,	☐
1047	76 - Praėjusiais metais	Praėjusiais metais svarbiausios valstybės	Naujienos	☑	{"teksto_id":"1047","anaforos":[{"pirmtakai":["264,13"],"objektai":["324,	☑
1046	75 - Europos Sąjungos	Europos Sąjungos (ES) lyderiai penktadienį	Politika	☑	{"teksto_id":"1046","anaforos":[{"pirmtakai":["0,30"],"objektai":["111,5"]	☑
1045	74 - Per pirmus šešis šių	Per pirmus šešis šių metų mėnesius	Naujienos	☑	{"teksto_id":"1045","anaforos":[{"pirmtakai":["73,31"],"objektai":["185,3	☑
1044	73 - Seimo etikos sargai	Seimo etikos sargai nutarė, kad	Politika	☑	{"teksto_id":"1044","anaforos":[{"pirmtakai":["275,16"],"objektai":["293,	☑
1043	72 - Viena sėkmingiausių	Viena sėkmingiausių šalies manekenių	Naujienos	☑	{"teksto_id":"1043","anaforos":[{"pirmtakai":["93,15"],"objektai":["118,5	☑
1042	71 - Šiuo metu dėl	Šiuo metu dėl infliacijos galime jaustis	Naujienos	☑	{"teksto_id":"1042","anaforos":[{"pirmtakai":["61,15"],"objektai":["95,3"	☑
1041	70 - Tėvynės	Tėvynės sąjungos-Lietuvos krikščionių	Politika	☑	{"teksto_id":"1041","anaforos":[{"pirmtakai":["63,6"],"objektai":["115,7"]	☑
1040	69 - Antradienį Seimas	Antradienį Seimas ketina spręsti, ar pradėti	Politika	☑	{"teksto_id":"1040","anaforos":[{"pirmtakai":["229,9"],"objektai":["281,2	☑
1039	67 - Už vengimą mokėti	Už vengimą mokėti pridėtinės vertės	Politika	☑	{"teksto_id":"1039","anaforos":[{"pirmtakai":["126,16"],"objektai":["101,	☑
1038	53 - Baltarusijos	Baltarusijos prezidentas Aleksandras	Politika	☑	{"teksto_id":"1038","anaforos":[{"pirmtakai":["319,7"],"objektai":["328,4	☑
1037	50 - Europos Parlamento	Europos Parlamento Užsienio reikalų	Politika	☑	{"teksto_id":"1037","anaforos":[{"pirmtakai":["306,9"],"objektai":["317,5	☑
1036	45 - Lietuvos išrinkimas į	Lietuvos išrinkimas į Jungtinių Tautų	Politika	☑	{"teksto_id":"1036","anaforos":[{"pirmtakai":["204,16"],"objektai":["222,	☑
1035	43 - Graikija	„Graikija nebankrutavo. Prieš pusantrų	Naujienos	☑	{"teksto_id":"1035","anaforos":[{"pirmtakai":["1,8"],"objektai":["49,5"],"t	☑
1034	42 - Danija yra geriausia	Remiantis naujausiu „Gallup“ tyrimu,	Naujienos	☑	{"teksto_id":"1034","anaforos":[{"pirmtakai":["37,6"],"objektai":["104,3"]	☑
1033	38 - Buvusio Seimo	Buvusio Seimo kanclerio Gintauto Vilkelio	Politika	☑	{"teksto_id":"1033","anaforos":[{"pirmtakai":["218,14","318,8"],"objektai	☑
1032	36 - Briuselyje vasario	Briuselyje vasario 10 dieną vykusioje	Politika	☑	{"teksto_id":"1032","anaforos":[{"pirmtakai":["38,48"],"objektai":["88,7"]	☑
1031	32 - Už prekyba	Už prekyba atsakingas eurokomisaras	Naujienos	☑	{"teksto_id":"1031","anaforos":[{"pirmtakai":["1374,2"],"objektai":["1381	☑

67 pav. Etaloninis tekstynas

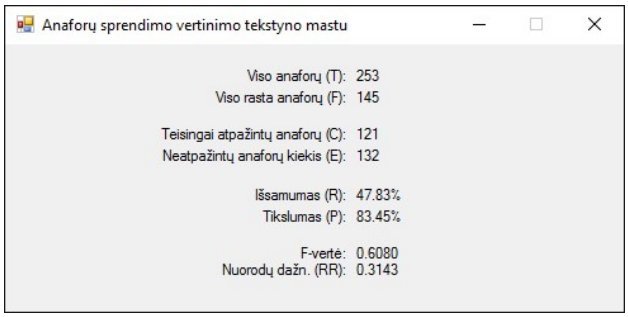
2 dalis. Buvo eksportuotas prototipiniu įrankiu ASAS sudarytas lietuviško teksto tekstynas JSON formatu tekstinio tipo bylomis. Viso buvo suformuotos 39 atskiros bylos kiekvienam tekstui.

3 dalis. Įrankyje ASAS buvo importuoti 39 straipsnių 39 etaloninės anotacijos JSON formatu tekstinio tipo bylomis (visas tekstynas). Įrankis ASAS palygino automatinio anaforų sprendimo sistemos lietuviško teksto anaforų anotaciją su etalonu ir pateikė įvertinimo charakteristiką 4 įvertinimo kriterijais: išsamumu (R), tikslumu (P), F-verte (F) ir nuorodų dažnumu (RR). Įvertinimo charakteristikos buvo pateiktos kiekvienam tekstui atskirai ir bendrai visam tekstynui. Taip pat buvo pateikta grafinė klaidų (neatitikimų) analizė (pavyzdys: 65 pav.). Kiekvieno teksto įvertinimai pateikti 68 pav., o įvertinimas viso tekstyno mastu – 69 pav.

Įvertinimų archyvas

Straipsnis	T	F	C	E	R	P	F-vertė	RR	Sprendimas
1055	2	1	0	2	0%	0%	0	0	VŽ ASS
1054	4	4	4	0	100%	100%	1	1	VŽ ASS
1053	6	5	5	1	83.33%	100%	0.9091	0.7143	VŽ ASS
1052	7	3	3	4	42.86%	100%	0.6	0.2727	VŽ ASS
1051	8	2	2	6	25%	100%	0.4	0.1429	VŽ ASS
1050	6	2	2	4	33.33%	100%	0.5	0.2	VŽ ASS
1049	6	2	2	4	33.33%	100%	0.5	0.2	VŽ ASS
1048	10	5	4	6	40%	80%	0.5333	0.25	VŽ ASS
1047	7	3	3	4	42.86%	100%	0.6	0.2727	VŽ ASS
1046	4	3	3	1	75%	100%	0.8571	0.6	VŽ ASS
1045	7	3	3	4	42.86%	100%	0.6	0.2727	VŽ ASS
1044	10	3	3	7	30%	100%	0.4615	0.1765	VŽ ASS
1043	11	5	2	9	18.18%	40%	0.25	0.1	VŽ ASS
1042	7	4	1	6	14.29%	25%	0.1818	0.0769	VŽ ASS
1041	8	3	3	5	37.5%	100%	0.5455	0.2308	VŽ ASS
1040	7	6	5	2	71.43%	83.33%	0.7692	0.5556	VŽ ASS
1039	7	7	7	0	100%	100%	1	1	VŽ ASS
1038	7	5	2	5	28.57%	40%	0.3333	0.1667	VŽ ASS
1037	9	8	6	3	66.67%	75%	0.7059	0.5	VŽ ASS
1036	10	4	3	7	30%	75%	0.4286	0.1765	VŽ ASS
1035	8	4	2	6	25%	50%	0.3333	0.1429	VŽ ASS
1034	7	3	3	4	42.86%	100%	0.6	0.2727	VŽ ASS
1033	7	6	3	4	42.86%	50%	0.4615	0.2727	VŽ ASS
1032	3	3	3	0	100%	100%	1	1	VŽ ASS
1031	3	2	2	1	66.67%	100%	0.8	0.5	VŽ ASS
1030	7	4	3	4	42.86%	75%	0.5455	0.2727	VŽ ASS
1029	7	6	6	1	85.71%	100%	0.9231	0.75	VŽ ASS
1028	4	0	0	4	0%	0%	0	0	VŽ ASS
1027	5	0	0	5	0%	0%	0	0	VŽ ASS
1026	7	4	4	3	57.14%	100%	0.7273	0.4	VŽ ASS
1025	3	0	0	3	0%	0%	0	0	VŽ ASS
1024	4	1	1	3	25%	100%	0.4	0.1429	VŽ ASS
1023	8	6	5	3	62.5%	83.33%	0.7143	0.4545	VŽ ASS
1022	7	3	3	4	42.86%	100%	0.6	0.2727	VŽ ASS
1021	4	4	4	0	100%	100%	1	1	VŽ ASS
1020	7	6	6	1	85.71%	100%	0.9231	0.75	VŽ ASS
1019	11	11	9	2	81.82%	81.82%	0.8182	0.6923	VŽ ASS
1018	3	0	0	3	0%	0%	0	0	VŽ ASS
1017	5	4	4	1	80%	100%	0.8889	0.6667	VŽ ASS

68 pav. Anaforų sprendimo įvertinimų archyvas



69 pav. Anaforų sprendimo įvertinimas viso tekstyno mastu

5.4. Eksperimento rezultatų analizė ir interpretavimas

5.4.1. I etapas.

Pirmo eksperimento vykdymo etape rastos klaidos buvo programuotojo ištaisomos, kol buvo pasiekta, kad prototipinis įrankis ASAS veiktų su imitaciniais duomenimis be klaidų. Eksperimento pirmas etapas patvirtino, kad prototipinis įrankis ASAS gali gerai veikti su mažos apimties imitaciniais duomenimis ir sudaro galimybę įvertinti lietuviško teksto anaforų sprendimą. Prototipinis įrankis ASAS ir eksperimentas su imitaciniais duomenimis buvo pristatytas 21-ojo tarpuniversitetinėje informacinių technologijų konferencijoje IVUS 2016 (straipsnis pateikiamas 2 priede).

5.4.2. II etapas.

Antrame eksperimento vykdymo etape buvo dirbama realioje aplinkoje. Surastos funkcinių reikalavimų (patobulinimas) ir techninės / programavimo klaidos (klaida) buvo fiksuojamos specializuotoje klaidų registravimo sistemoje „MantisBT“. Viso „MantisBT“ sistemoje buvo užregistruota 10 klaidų: 3 techninės / programavimo klaidos ir 7 patobulinimai. Visos užregistruotos klaidos buvo antro etapo 1 dalyje (lietuvių kalbos tekstyno ir jo anaforų anotavimo etalono sudarymas). Visos techninės / programavimo klaidos buvo ištaisytos bei atlikti 6 patobulinimai. 1 patobulinimo dėl neaktualumo nuspręsta neįgyvendinti. 70 pav. ir 71 pav. pateiktos „MantisBT“ klaidų ataskaitos.

pagal būseną	atvira	išspręsta	uždaryta	iš viso
užregistruota	1	-	-	1
išspręsta	-	9	-	9

pagal svarbą	atvira	išspręsta	uždaryta	iš viso
triviali	1	1	0	2
klaida tekste	0	1	0	1
maža	0	6	0	6
programos lūžimas	0	1	0	1

pagal kategoriją	atvira	išspręsta	uždaryta	iš viso
Klaida	0	3	0	3
Patobulinimas	1	6	0	7

70 pav. „MantisBT“ klaidų ataskaita pagal grupes

Pranešėjas pagal sprendimus	atvira	išspręsta	vėl atidaryta	neįmanoma atkartoti	neišsprendžiama	dublikatas	nieko nereikėjo keisti	suspenduota	nebus sprendžiama	% Klaidinga
Laura.Klusaite	0	9	0	0	0	0	0	0	1	10%

Sprendėjas pagal sprendimus	atvira	išspręsta	vėl atidaryta	neįmanoma atkartoti	neišsprendžiama	dublikatas	nieko nereikėjo keisti	suspenduota	nebus sprendžiama	% Išspręsta
Arunas.Ciuksys	0	9	0	0	0	0	0	0	1	100%

71 pav. „MantisBT“ klaidų ataskaita pagal pranešėją ir sprendėją

1 dalyje buvo sėkmingai sudarytas realus lietuvių kalbos tekstynas ir jo anaforų anotavimo etalonas.

2 dalyje tekstynas eksportuotas JSON formatu tekstinio tipo bylomis.

3 dalyje – atliktas anaforų sprendimo įvertinimas ir pateikta grafinė rezultatų analizė. Visuose etapuose įvertinimas buvo atliktas be klaidų: tiek vertinant kiekvieną straipsnį pavieniui (viso 39), tiek

vertinant visą tekstyną bendrai. Eksperimentas patvirtino, kad prototipinis įrankis ASAS sudaro galimybę įvertinti lietuvių kalbos anaforų sprendimo sistemą realioje aplinkoje su didesniais nei imitaciniai duomenimis, bendradarbiaujant skirtingiems aktoriams, dirbantiems skirtingose vietose, ir atlikti klaidų analizę. Buvo sudaryta galimybė įvertinti lietuvių kalbos automatinius anaforų sprendimus ir patvirtinta, kad tas pats sprendimas vertinant skirtingus tekstus pasirodo visiškai skirtingai.

6. REZULTATŲ APIBENDRINIMAS IR IŠVADOS

Šiuo tyrimu buvo sudarytos lietuvių kalbos anaforų sprendimo įvertinimo ir analizės galimybės sukuriant prototipinį įrankį ASAS. Įrankis ASAS realizuoja anaforų sprendimo įvertinimo metodiką bei apskaičiuoja ir pateikia įvertinimo kriterijus. Sukurtasis įrankis šiuo metu, kiek žinoma, yra vienintelis tam skirtas lietuviškas įrankis. Šiuo metu KTU Informacijos sistemų katedroje yra kuriami automatiniai anaforų sprendimai (projekto Semantika-LT tąsa), kurių analizė galės būti atliekama naudojant įrankį ASAS.

Išvados:

1. Buvo atlikta anaforų sprendimo ir jo įvertinimo galimybių analizė, kuri parodė, kad reikalingas įrankis lietuvių kalbos anaforų sprendimo įvertinimui ir analizei;
2. Reikiamam įrankiui ir juo įgyvendinamai metodikai buvo sudaryta reikalavimų specifikacija ir projektas;
3. Pagal įrankio reikalavimų specifikaciją buvo sudarytas realizacijos projektas bei realizuotas prototipinis įrankis ASAS. Taip buvo sudarytos lietuvių kalbos anaforų sprendimo įvertinimo galimybės;
4. Įrankiu ASAS buvo sudarytas lietuviško teksto anaforų anotavimo etalonas, kuris gali būti naudojamas ateityje lietuvių kalbos anaforų sprendimo įvertinimui;
5. Atlikus eksperimentą buvo įsitikinta, kad naudojantis įrankiu ASAS galima įvertinti anaforų sprendimą ir atlikti jo klaidų analizę realioje aplinkoje.

7. LITERATŪRA

[1] V. Žitkus ir L. Nemuraitė, „Taxonomy of anaphoric expressions as a starting point for anaphora resolution in Lithuanian corpus.," įtraukta *19-oji tarpuniversitetinė magistrantų ir doktorantų konferencija "Informacinė visuomenė ir universitetinės studijos"*, Kaunas, 2014.

[2] R. Mitkov, „Anaphora Resolution," London, UK, 2002.

[3] M. A. Kabadjov, A Comprehensive Evaluation of Anaphora Resolution and Discourse-new Classification, Colchester, 2007.

[4] S. Teufel, University of Cambridge, 02 03 2011. [Tinkle]. Available: https://www.cl.cam.ac.uk/teaching/1011/L104/lec12-2x2.pdf. [Kreiptasi 21 01 2015].

[5] A. Joshi, R. Prasad ir E. Miltsakaki, „Anaphora resolution: A centering approach.," įtraukta *Encyclopedia of language and linguistics*, Chicago, 2005.

[6] V. Stoyanov, C. Cardie, N. Gilbert, E. Riloff, D. Buttler ir D. Hysom, „Coreference Resolution with Reconcile," įtraukta *Proceedings of the ACL 2010 Conference Short Papers*, 2010.

[7] A. Znotins ir P. Paikens, „Coreference Resolution for Latvian," įtraukta *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 2014.

[8] E. Tjong Kim Sang ir F. De Meulder, „Introduction to the CoNLL-2003 shared task: language-independent named entity recognition," įtraukta *CONLL '03 Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, 2003.

[9] A. Čiukšys ir L. Nemuraitė, „Ko-referencijų sprendimo įrankio sukūrimo lietuvių kalbai galimybių analizė," įtraukta *IT 2015 20-oji tarpuniversitetinė magistrantų ir doktorantų konferencija*, Kaunas, 2015.

[10] E. Brill, „A simple rule-based part of speech tagger," įtraukta *ANLC '92 Proceedings of the third conference on Applied natural language processing*, 1992.

[11] N. Chincor, D. Lewis ir L. Hirschman, „Evaluating Message Understanding Systems: An Analysis Of The Third Message Understanding Conference (MUC-3)," *Computational Linguistics Journal,* 1993.

[12] R. Grishman ir B. Sundheim, „Message Understanding Conference-6: a brief history," *Computational linguistics,* t. 1, 1996.

[13] H. Lee, A. Chang, Y. Peirsman, N. Chambers, M. Surdeanu ir D. Jurafsky, „Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules," *Computational Linguistics,* 1 t. iš 239-4, pp. 886-916, 2013.

[14] B. Xian, F. Zahari ir D. Lukose, „Benchmarking ARS: anaphora resolution system.," įtraukta *11th International Conference on Knowledge Management and Knowledge Technologies*, 2011.

[15] C. Muller ir M. Strube, „MMAX: A Tool for the Annotation of Multi-modal Corpora," įtraukta *Proceedings of the 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, 2001.

[16] The University of Sheffield, Department of Computer Science, „GATE," [Tinkle]. Available: https://gate.ac.uk/. [Kreiptasi 01 02 2016].

[17] W. Chen ir W. Styler, „Anafora: A Web-based General Purpose Annotation Tool," įtraukta *Proceedings of the NAACL HLT 2013 Demonstration Session*, Atlanta, Georgia, USA, 2013.

[18] V. Žitkus ir L. Nemuraitė, „Taxonomy of anaphoric expressions as a starting point for anaphora resolution in Lithuanian corpus," įtraukta *19-oji tarpuniversitetinė magistrantų ir doktorantų konferencija: IVUS*, Kaunas, 2014.

[19] H. Cunningham, D. Maynard, K. Bontcheva ir V. Tablan, „GATE: an Architecture for Development of Robust HLT Applications.," įtraukta *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, 2002.

[20] C. Leng, B. South ir S. Shuying, „eHOST: The Extensible Human Oracle Suite of Tools | BLUlab,“ [Tinkle]. Available: http://blulab.chpc.utah.edu/content/ehost-extensible-human-oracle-suite-tools/. [Kreiptasi 02 2016].

[21] V. Žitkus ir L. Nemuraitė, „First Steps in Automatic Anaphora Resolution in Lithuanian Language Based on Morphological Annotations and Named Entity Recognition,“ įtraukta *ICIST*, Druskininkai, 2015.

[22] The University of Sheffield, Department of Computer Science, 2014. [Tinkle]. Available: https://gate.ac.uk/sale/tao/tao.pdf.

8. PRIEDAI

8.1. Straipsnis tarpuniversitetinėje konferencijoje „IT 2015“ – „Ko-referencijų sprendimo įrankio sukūrimo lietuvių kalbai galimybių analizė“

KO-REFERENCIJŲ SPRENDIMO ĮRANKIO SUKŪRIMO LIETUVIŲ KALBAI GALIMYBIŲ ANALIZĖ

Arūnas Čiukšys[1], **Lina Nemuraitė**[2]

Kauno technologijos universitetas, informacinių sistemų katedra, Studentų g. 50-308b, Kaunas, Lietuva, [1]*arunas.ciuksys@ktu.edu,* [2]*lina.nemuraite@ktu.lt*

Santrauka (abstract). Šiame straipsnyje trumpai aprašyta ko-referencijų sprendimo sudėtis ir įvertinimo procesas, apžvelgti egzistuojantys ko-referencijos sprendimo realizavimo ir įvertinimo įrankiai. Aprašyti lietuviški ištekliai ir pateiktos išvados, kuriomis remiantis būtų galima atlikti tolimesnę analizę ir sukurti pirmąjį lietuvišką įrankį.

Raktiniai žodžiai: kalbos apdorojimas, ko-referencijų sprendimas, Gate, LVCoref, Semantika LT

1 Įvadas

Ko-referencijų sprendimas yra natūralios kalbos apdorojimo (NKA) sistemos lingvistinio modulio fundamentali dalis. NKA sistemos paskirtis yra automatiškai apdoroti ir analizuoti labai didelius informacijos srautus, kuriuos apdoroti žmogui priimtinu laiku būtų neįmanoma arba tai reikalautų labai daug išteklių. Viena iš pavyzdinių natūralios kalbos apdorojimo užduočių yra įvairių informacinių pranešimų analizė terorizmo prevencijos tikslais. Nors tokių sistemų nauda yra neabejotina, sukurti patikimai veikiančias sistemas yra labai daug išteklių reikalaujantis ir sudėtingas uždavinys. Vyraujančių pasaulio kalbų natūralios kalbos analizės tyrimai vykdomi jau nuo 1970 m., bet lietuvių kalbai jie pradėti tik pastaraisiais metais.

Ko-referencijų sprendimas (angl. *coreference resolution*) prasideda nuo morfologinio natūralios kalbos apdorojimo. Ko-referencijos sprendimo metu atpažįstami skirtingose teksto vietose esantys to paties subjekto paminėjimai, kurie susiejami ko-referencijos ryšiu. Kiekvienas subjektas turi atskirą ko-referencijos ryšį [1]. Pavyzdžiui (paminėjimai parodyti laužtiniuose skliaustuose, o atskirų subjektų ryšiai sunumeruoti):

[Lietuvis$_1$] [Paulius Simonavičius$_1$] yra [jaunas mokslininkas$_1$] ir [profesorius universitete$_1$]. [Profesorius$_1$], kartu su [profesoriumi$_2$] [Vytautu Meškausku$_2$], atliko svarbų lietuvių kalbos lingvistikos tyrimą. [Jis$_1$], kartu su [savo$_1$] [kolega$_2$] [Meškausku$_2$], skaitys pranešimą Italijoje.

Lietuvių kalbai šiuo metu yra kuriami automatiniai anaforų (tam tikro tipo ko-referencijos ryšių) nustatymo algoritmai [13], tačiau juos reikia realizuoti ir įvertinti. Projekte „Semantika LT“ ko-referencijų nustatymas leistų padidinti semantinio anotavimo galimybes ir išgauti žymiai daugiau semantinės informacijos. Šiam tyrimui labai padėtų lietuvių kalbos ko-referencijų specifikos, kitų kalbų algoritmų tinkamumo, jų kombinavimo būdų ir kokybės įvertinimo galimybių analizė bei tam skirtų priemonių sukūrimas. Tam reikalingas įrankis – kompiuterizuota sistema, leidžianti realizuoti ir įvertinti automatinius algoritmus.

Nors lietuvių kalbai šiuo metu nėra ko-referencijų sprendimo įrankių ir NKA srityje ji laikoma neturinti pakankamai išteklių (angl. *under-resourced*), tam tikri ištekliai galimi. Kitoms pasaulio kalboms yra sukurta daug natūralios kalbos apdorojimo kompiuterinių sistemų, tarp jų ir ko-referencijų sprendimo ir analizės įrankių, bet dėl kalbų skirtumų jie nėra universalūs. Todėl tikslinga ištirti galimybes sukurti kompiuterizuotą ko-referencijų sprendimo sistemą lietuvių kalbai.

Šiame straipsnyje bus apžvelgta ko-referencijų sprendimo sudėtis, ko-referencijų sprendimo įvertinimo procesas, ko-referencijų sprendimo įrankiai ir ištekliai lietuviškam įrankiui kurti. Pabaigoje pateikiamos išvados ir numatomi ateities darbai.

2 Ko-referencijų sprendimo sudėtis

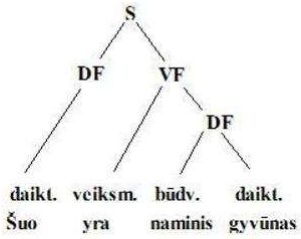
Įvairioms kalboms kuriami ko-referencijų sprendimo algoritmai remiasi tais pačiais metodais, kurie yra skirstomi į žiniomis paremtus (angl. *knowledge-rich approach*) ir žiniomis neparemtus metodus (angl. *knowledge-poor approach*) [7]. Šiuo metu daugelis kuriamų ko-referencijų sprendimų taiko žiniomis paremtus metodus, kurie reikalauja pirminio apdorojimo. Ko-referencijų sprendimai, sukurti žiniomis paremtų metodų pagrindu, gali būti skaidomi į du etapus: pirminio apdorojimo (angl. *preprocessing*) ir esminio apdorojimo (angl. *processing*). Pirminio apdorojimo etapo metu yra atliekami tam tikri morfologinio kalbos apdorojimo, sintaksinės analizės (angl. *parsing*) ir įvardytų esybių nustatymo (angl. *named-entity recognition*) uždaviniai. Esminio apdorojimo metu atliekami tam tikri ko-referencijos ryšių identifikavimo veiksmai, kurie priklauso nuo algoritmo specifikos [12, 13, 9].

2.1 Sintaksinė analizė

Sintaksinė analizė yra struktūrinio aprašymo suteikimas simbolių eilutei, remiantis tam tikra gramatika. Sintaksinę analizę atlieka sintaksinis analizatorius (angl. *parser*) – šį uždavinį sprendžianti kompiuterinė

programa. Norint realizuoti sintaksinį analizatorių, reikia formalizmo gramatikai parašyti; šiuo formalizmu parašytos duotajai kalbai skirtos gramatikos; algoritmo. Sintaksinė analizė lietuvių kalboje skirstoma į sudedamųjų dalių struktūros sudarymą (angl. *part-of-speech tagging)* ir priklausomybių struktūros sudarymą (angl. *relationships tagging*). Sintaksinę struktūrą įprasta vaizduoti medžio tipo diagramomis [6].

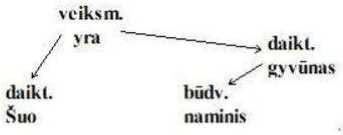
Sudedamųjų dalių struktūros sudarymo metu kiekvienas žodis tekste yra priskiriamas jam priklausančiai kalbos daliai, tiek remiantis apibrėžimu, tiek pagal kontekstą, kuriam žodis priklauso. Struktūros diagramos viršūnės atitinka tam tikro ilgio įėjimo eilutės segmentus, o briaunos nusako, kaip ilgesni segmentai sudaryti iš mažesnių. Viršutinis lygis atitinka visą sakinį [6]. Pavyzdžiui (S – sakinys, DF – daiktavardžio frazė, VF – veiksmažodžio frazė):



1 pav. Sudedamųjų dalių struktūra

Dėl to, kad skirtingame kontekste žodžiai gali sudaryti skirtingas kalbos dalis, kalbos dalių žymėjimas seniau buvo vykdomas tik rankiniu būdu. Dabar kalbos dalių žymėjimas atliekamas realizuojant automatinius algoritmus. Vienas iš kalbos dalių žymėtojų anglų kalbai yra Eriko Brilo sudedamųjų dalių struktūros sudarytojas (angl. *Eric Brill POS tagger*), kuris iš pradžių kiekvienam žodžiui priskiria kalbos dalį, remdamasis tikimybe, kokia kalbos dalis tam žodžiui dažniausiai priskiriama. Vėliau, priskyrus žodžiams kalbos dalis, atliekama kita iteracija ir ieškoma klaidų, analizuojant žodžius smulkesniame kontekste [1].

Priklausomybių struktūroje mazgai yra elementarūs įėjimo eilutės segmentai, o ryšiai nusako sintaksinius ryšius tarp elementaraus segmento ir nuo jo priklausomo priklausomybių struktūros pomedžio [6].



2 pav. Priklausomybių struktūra

2.2 Įvardytų esybių analizė

Įvardytų esybių analizės metu teksto elementai suskirstomi į iš anksto apibrėžtas kategorijas – klasifikuojami į žmonių vardus, įmones, vietoves, laiko elementus, kiekius, pinigų vienetus ir kt. Nors galimi ir universalūs, kelioms kalboms tinkami sprendimai, aukšti rezultatai pasiekiami tik pritaikius algoritmą tam tikrai kalbai, pasitelkus tos kalbos išteklius, sudarytus rankiniu būdu. Geriausios sistemos, pritaikytos natūralios kalbos srityje pažengusiai anglų kalbai, sugeba prilygti žmogui ir teisingai identifikuoti vaidmenis didesniu nei 90% tikslumu [4, 10, 12]. Pavyzdžiui:

[Paulius]$_{asmuo}$ [2015]$_{laikas}$ m. nusipirko 100 [UAB „Net septyni“]$_{įmonė}$ akcijų.

3 Ko-referencijų sprendimo įvertinimas

Nors ko-referencijų sprendimo algoritmai susideda iš dviejų dalių – pirminio ir esminio apdorojimo, kūrimo procesas tuo neapsiriboja. Tyrimams reikalingas ne tik automatinio algoritmo realizavimas, bet ir algoritmo rezultatų įvertinimas (angl. *evaluation*) ir palyginimas su etalonu (angl. *benchmarking*). Teksto apdorojimo sistemų tyrimo konferencijų *MUC* (angl. *Message Understanding Conference*) metu buvo sudaryti metodai, kuriais remiantis, įvertinamos ir palyginamos egzistuojančios anglų kalbos tekstų apdorojimo sistemos. Padarytos išvados, kad [2, 4]:

1. Norint įvertinti kalbos apdorojimo algoritmą, būtina sukurti tekstyną, kurio apdorojimo rezultatai būtų laikomi standartiniais kalbos apdorojimo sistemos įverčiais, nes apdorojant skirtingus tekstus automatiniai algoritmai gauna skirtingus įvertinimo rezultatus;

2. Tekstų rinkinys turi būti platus ir sudarytas iš pakankamo skaičiaus skirtingų tam tikros kalbos žodžių;

3. Norint tekstų apdorojimo sistemas palyginti tarpusavyje, jų rezultatai turi būti pateikiami tam tikru šablonu, t. y., jos turi turėti užduoto šablono sudarymo modulį (angl. *template generation*);

4. Turi būti nustatytos charakteristikos – matai, kuriais remiantis būtų galima tekstų apdorojimo sistemas įvertinti ir palyginti tiek tarpusavyje, tiek pavieniui (angl. *off-site*).

MUC metu buvo sukurti išsamūs tekstų rinkiniai. Pavyzdžiui, *MUC-3* tekstų rinkinį sudaro 400 000 žodžių, iš kurių 18 000 skirtingi. Algoritmams įvertinti pasirinktos išsamumo (angl. *recall*) ir tikslumo (angl. *precision*) charakteristikos. Tiek sukurti tekstų rinkiniai, tiek įvertinimo charakteristikos iki šiol naudojamos ko-referencijų sprendimams įvertinti ir palyginti [2, 11]. Ko-referencijų sprendimą galima įvertinti ir gauti minėtus įvertinimo matus tik palyginant automatinio algoritmo rezultatus su etalonu (angl. *gold standard corpus*). Etaloną sudaro žmogus, kuris teisingai sužymi tekste ko-referencijos ryšius. Kadangi rankiniu būdu algoritmų palyginimas ir įvertinimas būtų labai daug laiko reikalaujantis procesas, tam reikalingas įrankis, kuriuo būtų galima tiek sukurti etaloninį tekstyną, tiek įvertinti ir palyginti algoritmus, apdorojant skirtingus tekstus.

4 Ko-referencijų sprendimo įrankiai

Ko-referencijų sprendimo įrankiai reikalingi tiek automatiniams ko-referencijų sprendimų algoritmams realizuoti, tiek ko-referencijų sprendimų analizei ir įvertinimui. Nors kitoms kalboms atlikti minėtas funkcijas yra sukurta daug įvairių kompiuterinių sistemų, šiame straipsnyje analizuojami tik tie įrankiai, kurių galimybės pritaikyti lietuvių kalbai yra geriausios.

Gate yra apdorojimo komponentų integruota kūrimo aplinka (angl. *integrated development environment*) bei plačiai naudojama informacijos išgavimo (angl. *information extraction*) kompiuterinė sistema pateikiama kartu su daug plėtinių, iš kurių pagrindiniai yra morfologinio teksto apdorojimo įrankiai. *Gate* yra karkasas ir grafinio kūrimo aplinka, leidžianti kurti, vystyti ir naudoti kalbos apdorojimo komponentus. Pagrindinis *Gate* programinės įrangos grupės produktas yra *Gate Developer*. Kiti grupės produktai skirti tiek bendram darbui, tiek įvairių žinių bazių kūrimui. *Gate Developer* įrankiu yra sukurta ko-referencijų sprendimo sistemų įvairioms pasaulio kalboms, nors pagrindinė sistemos kalba yra anglų. Pagrindiniai *Gate* naudotojai veikia tiek mokslo, tiek verslo srityje – natūralios kalbos apdorojimo programinės įrangos kūrėjai, kalbos tyrėjai. *Gate Developer* sukurtas taip, kad būtų galima plėsti jo galimybes, kuriant plėtinius. Plėtiniai gali būti skirti bet kokios kalbos apdorojimui, nes kalbos specifinės raidės palaikomos naudojant unikodą (angl. *unicode*). *Gate* turi integruotą rezultatų įvertinimo įrankį *Gate AnnotationDiff*, kuris pateikia rezultatų įvertinimo charakteristikas, tarp jų tikslumą ir išsamumą [3].

LVCoref yra 2014 m. sukurtas pirmasis ko-referencijų sprendimo įrankis latvių kalbai. Įrankis kurtas kaip didesnės kalbos apdorojimo sistemos dalis ir išleistas atvirojo kodo licencijos sąlygomis. *LVCoref* sudaro standartiniai pirminio ir esminio apdorojimo komponentai [12]. Nors ši sistema nėra numatyta adaptuoti kitoms kalboms, tokia galimybė atsirastų koreguojant programinį kodą.

5 Ko-referencijų sprendimo ištekliai

Kuriant įrankį lietuvių kalbai, galima naudotis žinių bazėmis, pavyzdžiui, esamais lietuvių kalbos žodynais – vienakalbiais, dvikalbiais, ortografiniais, frazeologiniais, terminologiniais, biografiniais, tematiniais, enciklopediniais; tekstynais – dažniniais žodžių sąrašais, korkordansais, mokslo kalbos tekstynais; VDU dabartinės lietuvių kalbos tekstynu, turinčiu 140 mln. žodžių, surūšiuotų į dalis pagal tipą, sritį, žanrą, temą ir turinčių registruotą bibliografinę informaciją – autorių, pavadinimą, leidimo metus, leidyklą. Jame galima atlikti žodžių ar jų dalių paiešką, sužinoti žodžių dažnį kiekvienoje tekstyno dalyje, dažniausius žodžių junginius, gauti pasirinkto pločio (50, 100, 150 ar 300 simbolių) korkordansus. Be to, yra išleisti kompiuteriniai įrankiai *TildeNER* ir *LeMo* [6].

TildeNER yra pirmas ir šiuo metu vienintelis įrankis, skirtas lietuvių kalbos įvardytoms esybėms atpažinti, sukurtas remiantis teorijomis, skirtomis nuo kalbos nepriklausomiems įvardytų esybių atpažinimo įrankiams kurti. Modernūs įvardytų esybių atpažinimo įrankiai remiasi mašininio mokymosi technologijomis sukurta žinių baze, bet dėl išteklių trūkumo *TildeNER* remiasi universaliais principais. Nors įrankio rezultatai nėra labai aukšti, jie pakankami naudoti – tikslumas yra apie 40–70% priklausomai nuo to, kokių kategorijų elementus reikia atpažinti. Pasak kūrėjų, įrankis gali būti greitai gerinamas gavus pakankamą finansavimą, nors jo tobulinimas ir taip nėra apleistas. Įrankis yra atvirojo kodo ir gali būti laisvai modifikuojamas, tobulinamas ir naudojamas [8].

LeMo programa pagrinde skirta lietuviškam tekstui lemuoti ir turi lietuviškų žodžių gramatinės analizės ir sintezės galimybes. Programa sukurta kaip autoriaus hobis ir yra nemokama, bet nėra atviro kodo [6].

6 Analizės išvados ir ateities darbai

Atlikus ko-referencijų sprendimo įrankio sukūrimo lietuvių kalbai galimybių analizę padarytos išvados, kad:

1. Nors bandoma tuos pačius ko-referencijų sprendimo algoritmus pritaikyti artimoms kalboms [5] ir skirtingų kalbų ko-referencijų sprendimo algoritmai remiasi tais pačiais metodais, dėl kalbų skirtumų sprendimas negali būti universalus. Ko-referencijų sprendimo pirminio apdorojimo sprendimai sudaromi, remiantis kalbos gramatikos formalizmu; esminis apdorojimas taip pat realizuojamas pagal kalbos specifiką;

2. Ko-referencijų sprendimo ir įvertinimo įrankis turi leisti tiek realizuoti ko-referencijų sprendimą, tiek jį įvertinti, todėl turi apimti šias funkcijas:

- Etalonų sudarymą, rankiniu būdu anotuojant ko-referencijas tekste;
- Pirminio ir esminio etapų algoritmų realizavimą;
- Rezultatų generavimą pagal šabloną;
- Algoritmų įvertinimą.

3. Įrankis *Gate* su jo išplėtimo galimybėmis buvo adaptuotas įvairių pasaulio kalbų ko-referencijoms spręsti, todėl galima daryti prielaidą, kad įrankį galima adaptuoti ir lietuvių kalbai. Be to, galima naudoti tą patį algoritmų įvertinimo modulį, kadangi algoritmams įvertinti naudojamos tos pačios įvertinimo charakteristikos;

KTU informacijos sistemų katedroje kuriamas semantinio anotatoriaus prototipas (vykstančio „Semantika LT“ projekto dalis), kuris šiuo metu jau sugeba identifikuoti tam tikrą ontologijos individų tipų, ryšių ir įvykių aibę. Šiam projektui yra kuriami automatiniai anaforų (vieno iš ko-referencijos ryšio tipų) nustatymo sprendimai, bet reikia juos įvertinti.

Remiantis atlikta analize ir prieinamais bei kuriamais ištekliais, tolimesniais darbais bus siekiama detaliau ištirti įrankio, kuriuo būtų galima realizuoti kuriamus anaforų algoritmus ir juos įvertinti, sukūrimo galimybes ir jas realizuoti.

Literatūros sąrašas

[1] **Brill E.** A simple rule-based part of speech tagger. *ANLC '92 Proceedings of the third conference on Applied natural language processing*. 1992.

[2] **Chinchor N., Lewis D. D., Hirschman L.** Evaluating Message Understanding Systems: An Analysis Of The Third Message Understanding Conference (MUC-3) *Computational Linguistics Journal.* 1993.

[3] **Cunningham H., Maynard D., Bontcheva K., Tablan V.** GATE: an Architecture for Development of Robust HLT Applications. *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*. 2002.

[4] **Grishman R., Sundheim B.** Message Understanding Conference-6: a brief history. *COLING '96 Proceedings of the 16th conference on Computational linguistics - Volume 1*. 1996.

[5] **Holen G. I.** Automatic Anaphora Resolution for Norwegian (ARN) *Department of Linguistics and Scandinavian Studies, Master.* 2006.

[6] **Kasparaitis P.** Kompiuterinės lingvistikos įvadas. *Dr. Pijaus Kasparaičio paskaitų medžiaga.* 2012.

[7] **Mitkov R., Lappin S., Boguraev B.** Introduction to the special issue on computational anaphora resolution. *Computational Linguistic*. 2001.

[8] **Pinnis M.** Latvian and Lithuanian Named Entity Recognition with TildeNER *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. 2012.

[9] **Stoyanov V., Cardie C., Gilbert N., Riloff E., Buttler D., Hysom D.** Coreference Resolution with Reconcile. *Proceedings of the ACL 2010 Conference Short Papers.* 2010.

[10] **Tjong Kim Sang E., De Meulder F.** Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. *CONLL '03 Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*. 2003.

[11] **Xian B. C. M., Zahari F., Lukose D.** Benchmarking ARS: anaphora resolution system. *11th International Conference on Knowledge Management and Knowledge Technologies*. 2011.

[12] **Znotins A., Paikens P.** Coreference Resolution for Latvian. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14).* 2014.

[13] **Žitkus V., Nemuraitė L.** Taxonomy of anaphoric expressions as a starting point for anaphora resolution in Lithuanian corpus. *19-oji tarpuniversitetinė magistrantų ir doktorantų konferencija: IVUS.* 2014.

Making coreference resolution tool for Lithuanian: an analysis of the possibilities

In this paper, coreference resolution composition, evaluation and existing tools are briefly overviewed. Lithuanian resources are described and analysis conclusions are made. According to this research, further detailed analysis and the first Lithuanian tool could be created.

8.2. Straipsnis tarpuniversitetinėje konferencijoje „IVUS 2016“ – „ASAS – lietuvių kalbos anaforų sprendimo analizės ir įvertinimo prototipas“.

ASAS – lietuvių kalbos anaforų sprendimo analizės ir įvertinimo prototipas

Arūnas Čiukšys
Kauno technologijos universitetas
Informacijos sistemų katedra
Kaunas, Lietuva
arunas.ciuksys@ktu.edu

Rita Butkienė
Kauno technologijos universitetas
Informacijos sistemų katedra
Kaunas, Lietuva
rita.butkiene@ktu.lt

Santrauka. **Kasdien augant skaitmeninės informacijos kiekiui, vis aktualesnė tampa semantinė paieška. Semantinė paieška padeda iš nestruktūrizuotų tekstų išgauti daugiau informacijos, nei atliekant paiešką pagal raktinius žodžius. Vienas iš kokybiškesnę semantinę paiešką leidžiančių įgyvendinti komponentų yra anaforų sprendimas. Anaforų sprendimo sistemos lietuvių kalbai šiuo metu yra kuriamos. Tačiau įrankių, leidžiančių efektyviai įvertinti sukurtus ar kuriamus anaforų sprendimus, nėra. Todėl buvo nuspręsta sukurti kompiuterizuotą įrankio prototipą ASAS, kuris ir pristatytas šiame straipsnyje. ASAS suteikia galimybę įvertinti anaforų sprendimus, sukuriant anaforų anotavimo tekstyną ir jo etaloną, automatiškai palyginant anaforų sprendimo sistemos sužymėtas anaforų anotacijas tekstyne su etalonu.**

Reikšminiai žodžiai – ASAS, anaforos, lietuvių kalbos anaforų sprendimo analizė ir įvertinimas, semantika, natūralios kalbos apdorojimas, skaitmeninė informacija, koreferencijos.

I. ĮVADAS

Nuolat augant internete publikuojamos informacijos kiekiui, prasminga paieška nestruktūrizuotuose tekstuose tampa vis svarbesnė. Prasmingos, ne vien tik raktažodžiais grįstos paieškos įgyvendinime svarbų vaidmenį atlieka natūralios kalbos apdorojimo (NKA) sistemos, t. y. sistemos, kurios automatiškai išanalizuoja tekstą, nustato jo struktūrą, žodžių morfologines savybes, junginius, sakinių sintaksę, įvardytas esybes ir kitas teksto ir jo dalių savybes. Turint patikimai veikiančias NKA sistemas galima semantinės paieškos tekstuose plėtoti sprendimus, leidžiančius tiksliau atsakyti į vartotojų pateiktas užklausas, sumažinti rankinio paieškos darbo apimtis.

Informacijos apie NKA sprendimus vyraujančioms pasaulio kalboms galima rasti jau nuo 1970 m. Tuo tarpu NKA sprendimai lietuvių kalbai pradėti kurti palyginti neseniai ir ryškesnė pažanga stebima per pastaruosius keletą metų. 2015 metais VDU ir KTU vykdyto projekto Semantika-LT („Lietuvių kalbos sintaksinės – semantinės analizės sistema tekstynui, lietuviškam internetui ir viešojo sektoriaus taikymams", vykdytas pagal Ekonomikos augimo veiksmų programos 3 prioriteto „Informacinė visuomenė visiems“ įgyvendinimo priemonę Nr. VP2-3.1-IVPK-12-K „Lietuvių kalba informacinėje visuomenėje“) metu buvo sukurta lietuvių kalbos sintaksinės semantinės analizės informacinė sistema (LKSSAIS), teikianti vartotojams NKA ir semantinės paieškos tekstuose paslaugas. LKSSAIS semantinė paieška grindžiama tekstų morfologine, sintaksine ir įvardytų esybių informacija, kurią pateikia atitinkami LKSSAIS NKA komponentai. Tačiau ši informacija leidžia realizuoti tik ribotą informacijos atpažinimą. Geresnių rezultatų būtų galima tikėtis, jei semantinei tekstų analizei būtų pateikta informacija apie anaforas. Anaforų sprendimas yra svarbi NKA sistemos dalis.

Anafora yra vienas iš koreferencijos ryšio tipų, kurį ji detalizuoja: išskiriamas anaforos pirmtakas ir objektas. Anaforų sprendimo [1] metu atpažįstami skirtingose teksto vietose esantys to paties subjekto paminėjimai, kurie susiejami anaforos ryšiu. Anaforos objektai, kaip atskiri žodžiai, dažnai neturi jokios prasmės (pvz., įvardinės anaforos objektai), bet susiejus juos su anaforos pirmtaku, jie įgauna prasmę, kurią suteikia anaforos pirmtakas. Dėl to, apdorojant tekstą, yra galimybė iš jo išgauti daugiau informacijos. Pavyzdžiui: „Tomas šiandien nebuvo mokykloj. Jis sirgo.“ Žodžiai „Tomas“ ir „Jis“ sudaro anaforą (siejasi anaforos ryšiu). Be anaforų sprendimo negalėtume žinoti, kodėl šiandien Tomas nebuvo mokykloj arba, kas sirgo. Be natūralios kalbos apdorojimo netenkama daug panašios informacijos.

Anaforų sprendimo sistemos yra labai sudėtingos [2]. Pirmasis bandymas sukurti anaforų sprendimą lietuvių kalbai Semantika-LT projekte parodė, kad tokio sprendimo kūrimas turi būti laipsniškas, nuolat tikrinant ir vertinant sukurto algoritmo kokybę. Norint atlikti tai efektyviai, reikia turėti metodiką ir analizę bei vertinimą automatizuojančius įrankius.

Antrame skyriuje aprašomi kriterijai, kuriais turėtų pasižymėti anaforų sprendimams vertinti naudojami įrankiai, ir jų pagrindimas. Trečiame skyriuje pristatomas sukurtas įrankio prototipas ASAS.

II. MOTYVACIJA

Kuriant lietuvių kalbos anaforų sprendimo sistemas svarbu sudaryti sąlygas įvertinti, ar priimti sprendimai dėl anaforų atpažinimo algoritmų duoda gerus rezultatus. Efektyvesniam vertinimui praverstų kompiuterizuotas įrankis, kuris įgalintų sudaryti anaforų anotavimo etaloninį tekstyną ir jį palyginti su anaforų sprendimo sistemos rezultatais, bei išvestų įvertinimo matus.

A. Sprendimo įvertinimas, etalono sudarymas ir bendra prieiga

XX a. 9-tojo dešimtmečio pabaigoje buvo organizuojamos angliško teksto automatinio natūralios kalbos apdorojimo sistemų konferencijos *MUC* (angl. *Message Understanding*

Conference). Pagrindinė konferencijų paskirtis buvo įvertinti ir tarpusavyje palyginti teksto apdorojimo sistemas, joms atliekant tas pačias užduotis, t. y. apdorojant tuos pačius tekstus siekiant gauti tuos pačius rezultatus [3] [4]. Sprendimams vertinti buvo pasirinkta naudoti dydžius R (išsamumas), P (tikslumas) ir F-vertė.

Išsamumas R (1), kilęs iš signalų teorijos, parodo santykį tarp gautų teisingų rezultatų C ir visų tekste esančių rezultatų T kiekių.

$$R = C / T \quad (1)$$

Tikslumas P (2) parodo santykį tarp tekste gautų teisingų rezultatų C ir visų gautų rezultatų F kiekių.

$$R = C / F \quad (2)$$

Šioms charakteristikoms papildomai vertinti naudojama F-vertė (angl. *F-measure*) (3). Tai yra harmoninė išraiška tarp tikslumo (P) ir išsamumo (R).

$$F = (2\ x\ P\ x\ R)/\ (P + R) \quad (3)$$

Automatinių sprendimų įvertinimas apdorojant skirtingą tekstą negalėjo būti lyginamas, nes skirtingi algoritmai vertinant skirtingo ilgio ar žanro tekstus gali kiekvienąkart pasirodyti skirtingai. Todėl *MUC* konferencijose automatiniams sprendimams įvertinti reikėjo pateikti tas pačias užduotis. Tuo tikslu buvo sukurtas tekstynas ir sudarytas tekstyno etalonas (angl. *gold standard corpus*). Norint išvengti netikslumų dėl minėtų problemų, tekstas turėjo būti ilgas ir turintis daug skirtingų žodžių. Trečiosios *MUC* konferencijos metu buvo naudojamas tekstų rinkinys, kurį sudarė 400 000 žodžių, iš kurių 18 000 buvo skirtingi.

Konferencijų metu pasirinkti įverčiai ir sudaryti tekstynai, iš dalies ar pilna apimti naudojami iki šiol. Jais įvertinamos įvairios anglų ir kitų kalbų automatinės teksto apdorojimo sistemos. Naudojami ne tik minėtieji įverčiai, bet ir nuorodų dažnumas RR (4) – panašus į tikslumą matas, bet vietoje vardiklyje naudojamo sistemos gautų teisingų rezultatų kiekio imama visų tekste esančių rezultatų (T) ir sistemos blogų rezultatų (E) kiekių suma.

$$RR = C/\ (T + E) \quad (4)$$

Šiuo metu anglų kalboje vertinant įvairius sprendimus yra naudojami MUC konferencijų tekstų rinkiniai [5] [6]. Tokiu būdu sprendimai gali būti palyginami pavieniui (angl. *off-site*).

Dėl skirtumų tarp kalbų anaforų sprendimo sistemos negali būti daugiakalbės ir lietuvių kalbai angliški MUC tekstynai negali būti naudojami. Bet lietuvių kalbos tekstyno, kuriam sudarytas anaforų anotavimo etalonas, šiuo metu nėra.

Tekstyno ir jo etalono sudarymo procesas yra labai sudėtingas procesas: surinkti tinkamo ilgio ir sudėtingumo tekstus, turinčius daug skirtingų žodžių, sudaryti jų anotavimo etalonus, sužymint anaforas tekste rankiniu būdu, gali tik lietuvių kalbos specialistas, o norint sprendimus palyginti tarpusavyje skirtingi kūrėjai turi naudoti tą patį tekstyną. Todėl kompiuterizuotą įrankį tikslingą naudoti ne tik tekstynui ir jo etalonui sudaryti, bet ir galimybei skirtingiems anaforų sprendimo sistemų kūrėjams prieiti prie sudaryto tekstyno. Priėjimas prie vieningos sudarytos duomenų bazės įgalina anaforų sprendimo įvertinimą pavieniui. Be to, sutaupo resursų, nes nereikia sudarinėti tekstyno ir jo etalono. Antra, net ir naudojant bendrai prieinamus tuos pačius tekstynus, norint gauti sprendimų įvertinimo dydžius tikslinga naudoti kompiuterizuotą įrankį – verčių apskaičiavimas rankiniu būdu būtų ypač imlus darbui ir laikui.

B. Skirtingi anaforų tipai

Anaforos yra skirstomos į tipus. Pagal anaforos objekto tipą ir dėl skirtumų tarp kalbų, jos kiekvienoje kalboje klasifikuojamos skirtingai. 2014 m. buvo pasiūlyta pirmoji lietuvių kalbos anaforų taksonomija [7], kurioje buvo išskirti 3 anaforų tipai: morfologinio, leksinio semantinio ir dalykinės srities tipo. Tipai buvo suskirstyti į daugiau potipių. Pavyzdžiui, tikrinio daiktavardžio – *Tomas, Petras*; vietos prieveiksmio – *čia, ten*; asmeninio įvardžio – *aš, tu, jie, jos*. Daugelis kitoms kalboms esamų anaforų sprendimų nesugeba atpažinti visų anaforų tipų.

Pirmieji anaforų sprendimai, kuriami lietuvių kalbai, yra skirti įvardinėms anaforoms atpažinti. Svarbu, kad sprendimo įvertinimo įrankis suteiktų galimybę, kuriant etaloninį tekstyną, išskirti anaforų tipus ir įvertinti algoritmą sprendžiant tik tam tikro tipo anaforas. Be to, kol nėra nusistovėjusios ir bendrai pripažįstamos anaforų taksonomijos lietuvių kalbai, svarbu įrankyje turėti galimybę laisvai sukurti naujus tipus ar redaguoti esamus.

C. Sprendimo rezultatų analizė – grafinis palyginimas

Kūrėjui, kuriančiam anaforų sprendimo sistemą, svarbu pamatyti konkrečius sistemos įverčius. Tokiu būdu kūrėjas gali žinoti savo progresą. Bet tam, kad kūrėjas galėtų tobulinti sistemą, reikia žinoti, kurias anaforas sistema atpažino gerai, o kurių neatpažino ar atpažino klaidingai. Be grafinio rezultatų atvaizdavimo detali, rankiniu būdu atliekama rezultatų analizė būtų ypač imli darbui. Rezultatų analizės efektyvumui pagerinti tikslinga naudoti grafinį sprendimo rezultato su etalonu palyginimą, kurį galėtų pateikti kompiuterizuotas įrankis.

D. Duomenų mainai

Pagrindiniai formatai, kuriuos naudoja įvairūs sprendimai [8] [9] [10], veikiantys natūralios kalbos apdorojimo srityje, yra *XML* ir *JSON*. Naudojant *XML* formatą, lyginant su *JSON*, labiau yra apkraunama kompiuterio laikinoji atmintis, bet lengviau skaitomas žmogui ir galimas didesnis apdorojimo greitis. Ir atvirkščiai: naudojant *JSON*, reikia mažiau atminties, bet sunkiai žmogaus skaitomas, o programinis apdorojimas lėtesnis nei XML. Tiek pirmieji kuriami automatiniai anaforų sprendimai, tiek kiti natūralios kalbos apdorojimo sprendimai lietuvių kalboje (projektas Semantika-LT) duomenims pateikti naudoja *JSON* formatą. Siekiant išlaikyti kitose sistemose priimtų sprendimų tęstinumą ir taip užtikrinti naujų sprendimų integravimą, būtų tikslinga pasirinkti JSON formatą. Be to, esant reikalui, yra įmanoma sukurti programas, kurios įgalintų duomenų konversiją tarp formatų.

E. Kiti svarbus aspektai

Lietuvių kalba NKA srityje laikytina neturinti pakankamai išteklių (angl. *under-resourced*), todėl nėra nusistovėjusių poreikių įvertinant lietuvių kalbos anaforų sprendimus. Be to, daugelis įrankių po sukūrimo dažnai yra netobulinami ir

nepalaikomi. Todėl atvirojo kodo formatas leistų adaptuoti įrankį pagal besikeičiančius vartotojų poreikius ir pratęsti įrankio palaikymą ir tobulinimą susiklosčius situacijai, kai autorius nusprendžia nepratęsti įrankio palaikymo ir tobulinimo.

Svarbi įrankio savybe turėtų būti greitas išmokstamumas juo naudotis. Greit išmokus, pavyzdžiui, per dieną, naudotojas galėtų įrankiu dirbti veiksmingai ir našiai.

F. Esami anaforų anotavimo / įvertinimo sprendimai

Esamiems įrankiams palyginti pasirinkti 8 kriterijai: koreferencijų anotavimo galimybė, anaforų anotavimo galimybė, įvertinimo rezultatų pateikimas matais R, P, F, RR, bendros prieigos galimybė, duomenų mainų palaikymas *JSON* formatu, skirtingų anotacijų grafinio palyginimo galimybė, įrankio tobulinimo galimybė ir ar įmanoma išmokti naudotis įrankiu per 1 dieną. Palyginimui *„Lent. 1“* buvo pasirinkti 3 įrankiai, kurie tenkina daugiausiai pasirinktų kriterijų:

1) GATE įrankių grupė

GATE [9] [11] įrankiai yra kuriami ir palaikomi ne pelno siekiančios organizacijos, todėl visi grupės komponentai yra atvirojo kodo (sukurti *Java* programavimo kalba). *GATE* yra apdorojimo komponentų integruota kūrimo aplinka (angl. *integrated development environment*) bei informacijos išgavimo (angl. *Information Extraction*) sistema, pateikiama kartu su daug plėtinių, leidžiančiu sistemą adaptuoti įvairiems specifiniams poreikiams. Duomenų mainams su išoriniais sprendimais *GATE* naudojamas *XML* formatas. Bet yra sukurtas specifinį *Twitter JSON* palaikantis įskiepis. Standartiniam *JSON* formatui turėtų būti kuriamas papildomas įskiepis. Kalbos specifinės raidės palaikomos naudojant unikodą.

Pagrindinis *GATE* komponentas yra vartotojo kompiuteryje paleidžiama taikomoji programa *GATE Developer*, kurią naudojant galima anotuoti ir grafiškai analizuoti koreferencijų grandinėlę. Taip pat įrankis turi integruotą skirtingų anotacijų įvertinimo funkciją (*GATE AnnotationDiff*), kuri automatiškai apskaičiuoja tikslumą, išsamumą, F-vertę. Bet be papildomų priedų *GATE Developer* negalima žymėti anaforų pirmtakų ir objektų. Jau sukurtų tokių priedų rasti nepavyko. Todėl jie turėtų būti sukurti papildomai. *GATE Teamware* naršyklėje veikianti (internetinė) programa ir *GATE Cloud* sudaro bendradarbiavimo galimybes kuriant tekstynus ir jų etalonus.

GATE yra plačiai taikoma sistema, skirta įvairiems NKA naudotojams:

- programuotojams, kuriantiems programinę natūralios kalbos apdorojimo įrangą;
- kalbos tyrėjams;
- kalbos apdorojimo mokytojams ir dėstytojams.

Dėl didelio programos pritaikymo galimybių *GATE* išmokti naudotis yra sunku.

2) eHost

eHost [12] įrankis sukurtas Sveikatos apsaugos informacinių tyrimo konsorciumo (angl. *A Consortium for Healthcare Informatics Research*), į kurį įeina Jutos universitetas ir Solt Leik Sičio sveikatos apsaugos sistema. Kūrėjų įrankis, kurio pirminė paskirtis yra medicinos tekstų anotavimas, naudojamas įvairiuose projektuose nuo 2010 m.

Pavadinimas *eHost* yra termino „išplėstinis žmogaus įrankių rinkinys“ (angl. *extensible Human Oracle Suite of Tools*) santrumpa. Tai yra atvirojo kodo *Java* programavimo kalba sukurtas sistemos prototipas, veikiantis per naršyklę vietiniame kompiuteryje. *eHost* įgalina konceptų, jų savybių ir ryšių tarp jų anotavimą.

Pagrindinės įrankio savybės: tekstų generavimas ir pirminis anotavimas iš žodynų; rankinis ryšių anotavimas; kreipiniai į žodynus naudojant adaptuojamą programavimo sąsają (angl. *API*); duomenų mainai *XML* formatu.

3) Anafora

Anafora [10] yra anotavimo įrankis, sukurtas Kolorado universitete. Tai yra atvirojo kodo įrankis, sukurtas *Python* programavimo kalba. Kuriant *Anaforą*, pagrindiniai kriterijai buvo lengvas išmokstamumas ir pritaikomumas tiek mažiems, tiek dideliems projektams ir įvairių galimų teksto ryšių anotacijų palaikomumas.

Kūrėjų įvardijamas įrankio privalumas ir išskirtinumas – *Anafora* veikia nutolusiame serveryje (debesyje) ir paleidžiama per interneto naršyklę. Todėl *Anafora* gali veikti bet kurioje operacinėje sistemoje be papildomo diegimo ar vietinio duomenų saugojimo. Dirbant su programa darbiniai duomenys yra nuolat automatiškai išsaugomi debesyje. *Anafora* duomenų mainams naudoja *XML* formatą.

1 LENTELĖ. ESAMŲ ĮRANKIŲ PALYGINIMAS

Kriterijus	***GATE***	***eHost***	***Anafora***
Koreferencijų anotavimo galimybė	+	+	+
Anaforų anotavimo galimybė	-	-	-
Įvertinimo rezultatų pateikimas matais R, P, F, RR	+	-	-
Bendros prieigos galimybė	+	-	+
Duomenų mainai *JSON* formatu	- /+ (*Twitter JSON*)	-	-
Skirtingų anotacijų grafinis palyginimas	+	-	-
Įrankio tobulinimo galimybė	+	+	+

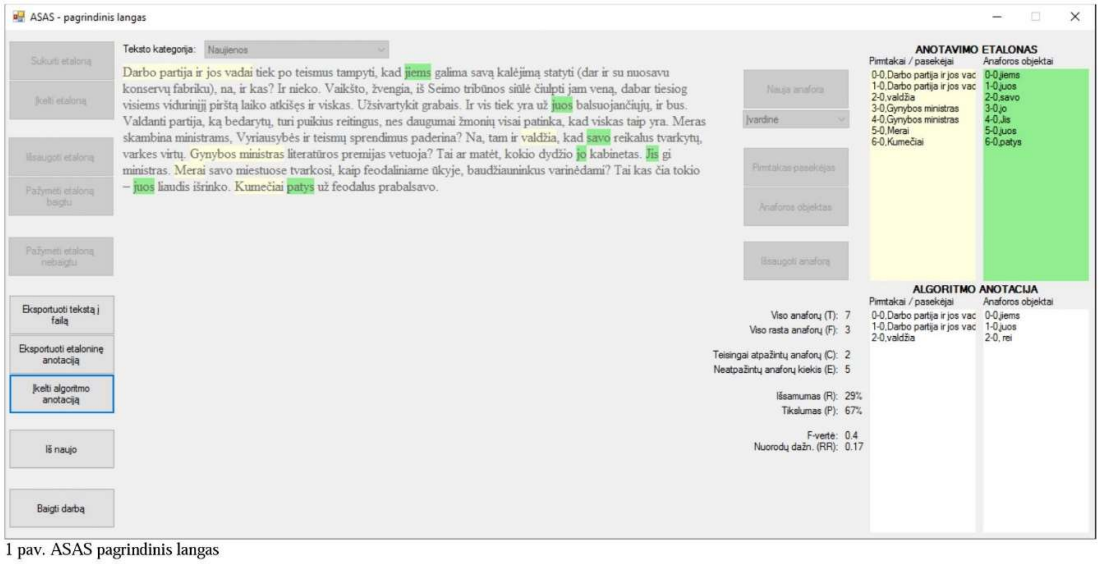
Įmanoma išmokti naudotis įrankiu per 1 dieną	-	+	+

Iš nagrinėtų įrankių artimiausias pasirinktiems kriterijams yra *GATE*, bet visų keliamų reikalavimų neatitinka. *GATE* yra sudėtinga išmokti naudotis (to padaryti per 1 dieną nepavyko), ir jis yra daugiau skirtas sudėtingiems kompleksiniams sprendimams kurti, o ne sprendimams analizuoti ir įvertinti. Pagrindinė problema yra ta, kad *GATE* nėra pritaikytas detaliam anaforų analizavimui t. y. nesukūrus papildomų įskiepių neįmanoma atskirai analizuoti anaforos objektų ar pirmtakų – galima tik žymėti koreferencijų grandinėles neišskiriant anaforos pirmtakų ir objektų. Todėl yra tikslinga sukurti nepriklausomą įrankį, atitinkantį visus reikiamus kriterijus.

III. SPRENDIMAS

Sukurtasis įrankio prototipas ASAS *„1 pav.“* išpildo antrajame skyriuje aprašytus kriterijus, išskyrus skirtingų anotacijų grafinį palyginimą. Įrankis šiuo metu yra tobulinamas ir dar šiais metais (2016 m.) ši funkcija atsiras. Šiame skyriuje pateiktos ASAS realizacijos ir veikimo detalės.

ASAS veikia vartotojo kompiuteryje su *Windows* operacine sistema ir suteikia jos naudotojui galimybę sudaryti anaforų anotavimo etalonus ir juos palyginti su anaforų sprendimo sistemos rezultatais, išvesdamas įvertinimo matus.



1 pav. ASAS pagrindinis langas

A. *ASAS funkcijos*

1) Prisijungimas – prisijungimas įvedant paskyros duomenis: vartotojo vardą ir slaptažodį;

2) Tekstyno sudarymas su galimybėmis:

a) nustatyti užbaigtumo žymą tekstyno elementui (tekstui),

b) klasifikuoti atskirus tekstyno tekstus pagal žanrą priskiriant jam tipą;

3) Etaloninio tekstyno sudarymas, apimantis anaforos objektų ir tipų žymėjimą tekste, parenkant anaforos tipą, su galimybėmis:

a) nustatyti užbaigtumo žymą,

b) atšaukti užbaigtumo žymą;

4) Anaforų sprendimo įvertinimas, apimantis:

a) teksto, kuriam sudarytas anaforų etalonas, eksportavimą tekstinio failo pavidalu,

b) teksto etaloninės anaforų anotacijos eksportavimą *JSON* formatu tekstinio failo pavidalu,

c) anaforų sprendimo sistemos rezultatų, pateiktų tekstiniame faile *JSON* formatu, importavimą,

d) įvertinimo rezultatų gavimą – grafinį atvaizdavimą ir įverčių R, P, F, RR pateikimą;

5) ASAS parametrų konfigūravimas, apimantis:

a) naujų anaforų tipų ir jų kodų sukūrimą ir senų tipų (kodų) redagavimą,

b) naujų teksto tipų sukūrimą ir senų tipų redagavimą,

c) užbaigtumo žymos atšaukimą.

B. *Sprendimo įgyvendinimas*

Prototipas ASAS suprogramuotas programavimo kalbą *C#*. Įrankis internete prieinamas adresu http://asas.netseptyni.lt

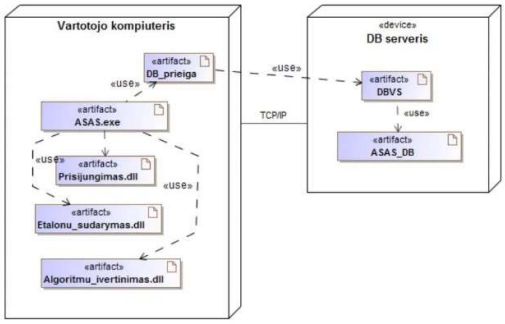
ASAS yra išleistas *Microsoft OneClick Deployment* technologija. Naudojama *Microsoft SQL Server* duomenų bazė. Įrankio duomenų bazė ir diegimo prieiga veikia tarnybinėje stotyje, prie kurios jungiamasi su kliento dalimi, kuri yra

įdiegiama vartotojo kompiuteryje iš diegimo prieigos *„ 2 pav.“*. Todėl ASAS kliento dalis gali veikti tik tuomet, kai yra ryšys su tarnybinės stoties ASAS dalimi. ASAS, kiekvienąkart prieš pradėdamas darbą (t. y. kai paleidžiama kliento dalis vartotojo kompiuteryje), tikrina ar nėra išleistas atnaujinimas. Jei yra – automatiškai atsinaujina.

Ši diegimo schema suteikia daug privalumų:

- Serverio dalies apkrova: apkraunamas tik *SQL* serveris, nes visa programos logika vykdoma vartotojų kompiuteriuose. Tai leidžia sutaupyti resursų techninei įrangai;
- Jokie duomenys nėra saugomi vartotojų kompiuteriuose (debesų technologija), todėl jie yra vientisi ir visiems prieinami vienodi. Vartotojams nereikia rūpintis dėl atsarginių duomenų kopijų. Naujiems vartotojams prieiga prie duomenų gali būti suteikiama paprasčiausiai sukuriant jiems paskyrą, įgalinančią naudotis įrankiu;
- Ypač paprastas įrankio diegimas, atliekamas vienu diegimo failo paspaudimu jį paleidžiant;

Šios technologijos pagrindinis trūkumas yra tai, kad įrankis gali būti įdiegtas tik *Windows* operacinėje sistemoje su įdiegtu ne žemesniu nei 4.5 versijos *.Net* karkasu.



2 pav. ASAS diegimo diagrama

C. *Duomenų modelis*

Duomenų mainams ir saugojimui naudojama biblioteka *JSON.Net 8.0.2*. Anotacijos yra įrašomos *JSON* formatu duomenų bazėje *string* tipo įraše. Pats tekstas saugomas atskiru *string* įrašu. Sukūrus kiekvieną tekstą, jam yra priskiriamas unikalus identifikacinis numeris, kuris *JSON* turi būti pateikiamas priekyje. ASAS šį numerį naudoja teksto identifikavimui importuojant anotaciją. Tiek anaforų pirmtakai, tiek objektai yra saugomi dviem sveikais skaičiais, kurių pirmasis nusako pradžios simbolį tekste, o antrasis ilgį. Anaforos tipo kodo išsaugojimas šiuo metu neįgyvendintas, nes nėra anaforų kodų klasifikacijos. Kadangi kodui vieta numatyta iš anksto, jo vietoje yra saugoma *null* reikšmė. *„3 pav.“* pateiktas teksto ir jo anotacijos grafinės vizualizacijos ASAS programoje ir anotacijos *JSON* formatu tekstiniame faile, pavyzdys.

Tomas nėjo į mokykla. Jis sirgo. Mokytoja apie tai informuota nebuvo, todėl ji sunerimo. Vaikai visada privalo informuoti el. paštu.

Tomas nėjo į mokykla. Jis sirgo. Mokytoja apie tai informuota nebuvo, todėl ji sunerimo. Vaikai visada privalo informuoti el. paštu.

{"teksto_id":"1014","anaforos":[{"pirmtakai":["0,5"],"objektai":["22,3"],"tipo_kodas":null},{"pirmtakai":["33,8"],"objektai":["76,2"],"tipo_kodas":null}]}

3 pav. Pavyzdinis tekstas ir jo grafinė bei JSON anotacijos.

D. *ASAS veikimo aprašymas ir pagrindinių funkcijų atlikimo scenarijus*

ASAS prieš pateikdamas grafinę sąsają vartotojui, pirma automatiškai jungiasi prie diegimo tarnybinės stoties ir patikrina, ar nėra išleistų atnaujinimų.

Jei ASAS negali užmegzti ryšio su diegimo ir duomenų serveriu, ASAS paleidimas yra nutraukiamas ir parodomas klaidos pranešimas. Užmezgus ryšį ir radus atnaujinimų – ASAS automatiškai atsinaujina naujausiu leidimu. Atsinaujinus ar neradus atnaujinimų, atidaromas autorizacijos langas, kuriame reikia įvesti vartotojo vardą ir slaptažodį. Vartotojas autorizuojasi savo paskyra ir pradeda darbą. Paskyros tipas gali būti dviejų tipų: tyrėjas arba administratorius. Nuo to priklauso prieinamų funkcijų sąrašas. Tyrėjo role dirbantis vartotojas gali atlikti 1, 2, 3, 4 funkcijas. Administratorius – visas 5 funkcijas, t. y. gali konfigūruoti įrankio ASAS parametrus.

I. Funkcijos „2. Tekstyno sudarymas“ scenarijus

- Vartotojas turi būti prisijungęs tyrėjo arba administratoriaus role;
- Pagrindiniame lange pasirenkama „Sukurti naują tekstą“ arba „Įkelti esamą tekstą iš tekstyno“;
- Pasirinkus naujo teksto kūrimą, rodomas tuščias teksto įvedimo langas ir kategorijos priskyrimo meniu su neparinkta kategorija. Įkėlus jau esamą tekstą, lange rodomas įkeltas tekstas, o meniu – esama priskirta kategorija. Tol, kol tekstui nėra uždėta žyma „baigtas“, laikoma, kad tekstas yra nesukurtas ir dar gali būti redaguojamas: papildomas, taisomas, keičiama kategorija, bet negali būti pradedamas anaforų žymėjimas (anotavimo etalono sudarymas);
- Tekstas papildytas gali būti dviem būdais: įvedant tekstą rankiniu būdu arba įklijuojant klavišų kombinacija *Ctrl + V*;
- Sudarant naują tekstą arba, toliau tęsiant jau sukurto teksto redagavimą, tekstas nėra išsaugomas tol, kol nėra pasirenkama išsaugoti. Tekstą išsaugant pirmąkart (kuriant naują tekstą), privalo būti nustatyta teksto kategorija. Pirmo išsaugojimo metu tekstas yra pridedamas į tekstyną. Tęstinio redagavimo metu, teksto kategorija gali būti pakeista.

II. Funkcijos „3. Etaloninio tekstyno sudarymas“ scenarijus

- Vartotojas turi būti prisijungęs tyrėjo arba administratoriaus role;
- Pagrindiniame lange pasirenkama „Įkelti esamą tekstą iš tekstyno“;

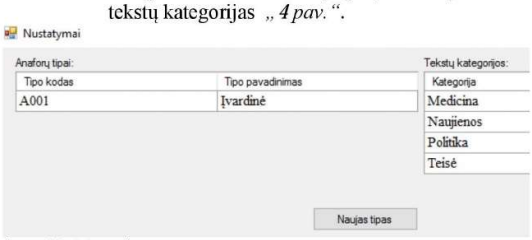
- Tuomet pagrindiniame lange atsiranda galimybė įkelti etaloną, jei tekstas turi žymą „baigtas“.
- Užkrovus etaloną, sąrašuose rodomos tekste sužymėtos anaforos. Anaforos taip pat yra pateikiamos ir grafiškai skirtingomis spalvomis atvaizduojant pirmtakus ir objektus;
- Tol, kol etalonui nėra priskirta žyma „baigtas“, etalono sudarymas gali būti pratęstas: trinamos jau pažymėtos anaforos ar pažymimos naujos;
- Sudarant etaloną, anaforų žymėjimas ar korekcijos nėra išsaugomos tol, kol neparenkama išsaugoti.

III. Funkcijos „4. Anaforų sprendimo įvertinimas“ scenarijus

- Vartotojas turi būti prisijungęs tyrėjo arba administratoriaus role;
- Pagrindiniame lange pasirenkama užkrauti esamą tekstą iš tekstyno;
- Tuomet pagrindiniame lange atsiranda galimybė užkrauti etaloną, jei tekstas turi žymą „baigtas“;
- Jei etalonas turi žymą „baigtas“, galima eksportuoti neanotuotą tekstą tekstinio failo pavidalu. Eksportuotas tekstas skirtas anaforų sprendimo sistemai anotuoti arba eksportuoti jį kaip etaloninę anotaciją;
- Anaforų sprendimo įvertinimas atliekamas importuojant anaforų sprendimo anotaciją. Importavus automatiškai yra išvedami įvertinimo kriterijai. Laikoma, kad anafora sprendimo buvo atpažinta, jei buvo atpažinti visi pirmtakai ir visi objektai nesuklystant nei vienu simboliu.

IV. Funkcijos „5. ASAS parametrų konfigūravimas“ scenarijus

- Vartotojas turi būti prisijungęs tyrėjo arba administratoriaus role;
- Pasirenkamas nustatymų meniu;
- Nustatymų menių galima pridėti, ištrinti ar redaguoti esamus anaforų tipus, anaforų kodus ir tekstų kategorijas *„ 4 pav. “*.



4 pav. Nustatymų langas

IŠVADOS IR ATEITIES DARBAI

Šiame straipsnyje buvo pristatytas lietuvių kalbos anaforų sprendimo analizės ir įvertinimo prototipas ASAS. ASAS šiuo metu, kiek žinoma, yra vienintelis tam skirtas lietuviškas įrankis.

Atlikta esamų įrankių analizė parodė, kad nei vienas iš jų neatitiko visų iškeltų reikalavimų, todėl buvo nuspręsta sukurti naują nepriklausomą prototipą. Sukurtas prototipas atitinka 7 iš 8 jam iškeltų pagrindinių reikalavimų, bet šiuo metu yra tobulinamas ir šiais metais (2016 m.) bus užbaigtas.

Kadangi šiuo metu (2016 m. vasario mėn.) lietuvių kalbos anaforų anotavimo tekstynai ir jų etalonai, naudojantis ASAS, yra tik sudarinėjami, nebuvo atliktas prototipo eksperimentas su didesniais duomenų kiekiais. Dėl to nėra aiškus ASAS veikimo stabilumas su didelėmis duomenų apimtimis. Su mažais duomenų kiekiais ASAS veikė be klaidų ir gavo teisingus imitacinio anaforų sprendimo įvertinimo matus.

Šiuo metu KTU Informacijos sistemų katedroje yra kuriami automatiniai anaforų sprendimai (projekto Semantika-LT tąsa), kurių analizė galės būti atliekama naudojant ASĄ.

Tolimesniais darbais bus siekiama sudaryti automatinio anaforų sprendimo įvertinimo metodiką ir tobulinti jau sukurtą įvertinimo ir analizės prototipą.

LITERATŪRA

[1] R. Mitkov, „Anaphora Resolution“ London, UK, 2002.

[2] A. Čiukšys ir L. Nemuraitė, „Ko-referencijų sprendimo įrankio sukūrimo lietuvių kalbai galimybių analizė,“ įtraukta *20-oji tarpuniversitetinė magistrantų ir doktorantų konferencija: "Informacinės technologijos 2015"*, Kaunas, 2015.

[3] N. Chincor, D. Lewis ir L. Hirschman, „Evaluating Message Understanding Systems: An Analysis Of The Third Message Understanding Conference (MUC-3)“ *Computational Linguistics Journal*, 1993.

[4] R. Grishman ir B. Sundheim, „Message Understanding Conference-6: a brief history,“ *Computational linguistics*, t. 1, 1996.

[5] H. Lee, A. Chang, Y. Peirsman, N. Chambers, M. Surdeanu ir D. Jurafsky, „Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules,“ *Computational Linguistics*, 1 t. iš 239-4, pp. 886-916, 2013.

[6] B. Xian, F. Zahari ir D. Lukose, „Benchmarking ARS: anaphora resolution system.,“ įtraukta *11th International Conference on Knowledge Management and Knowledge Technologies*, 2011.

[7] V. Žitkus ir L. Nemuraitė, „Taxonomy of anaphoric expressions as a starting point for anaphora resolution in Lithuanian corpus,“ įtraukta *19-oji tarpuniversitetinė magistrantų ir doktorantų konferencija: IVUS*, Kaunas, 2014.

[8] C. Muller ir M. Strube, „MMAX: A Tool for the Annotation of Multi-modal Corpora,“ įtraukta *Proceedings of the 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, 2001.

[9] The University of Sheffield, Department of Computer Science, „GATE,“ [Tinkle]. Available: https://gate.ac.uk/. [Kreiptasi 02 2016].

[10] W. Chen ir W. Styler, „Anafora: A Web-based General Purpose Annotation Tool,“ įtraukta *Proceedings of the NAACL HLT 2013 Demonstration Session*, Atlanta, Georgia, USA, 2013.

[11] H. Cunningham, D. Maynard, K. Bontcheva ir V. Tablan, „GATE: an Architecture for Development of Robust HLT Applications.,“ įtraukta *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, 2002.

[12] C. Leng, B. South ir S. Shuying, „eHOST: The Extensible Human Oracle Suite of Tools | BLULab,“ [Tinkle]. Prieinama: http://blulab.chpc.utah.edu/content/ehost-extensible-human-oracle-suite-tools/. [Kreiptasi 02 2016].