



KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA

# **REIKALAVIMŲ SPECIFIKACIJOS ANALIZĖS PRIEMONIŲ TYRIMAS IR KŪRIMAS**

**Magistro darbas**

Atliko: stud. V.Pukėnas IFM – 1/4 gr.

Vadovė: doc. dr. R.Butkienė

Data: 2007 01 18

Kaunas, 2007

# Turinys

1.	Summary .....	3
2.	Įvadas .....	4
3.	Analizės dalis.....	5
3.1	Tyrimo sritis, objektas ir problema .....	5
3.2	Reikalavimų specifikavimas IS kūrimo proceso kontekste .....	6
3.3	Reikalavimų specifikacijų kokybė .....	9
3.4	Formalizmai reikalavimų specifikacijoms atvaizduoti.....	13
3.5	CASE įrankiuose realizuotos priemonės specifikacijoms (modeliams) analizuoti .....	17
3.6	Analizės išvados.....	22
4.	Projekto dalis .....	23
4.1	Reikalavimų modelis.....	23
4.1.1	Kontekstinė diagrama.....	24
4.1.2	Sistemos panaudojimo atvejų diagrama .....	26
4.1.3	Panaudojimo atvejų specifikacijos .....	27
4.1.4	Dalykinės srities klasių diagrama.....	39
4.1.5	Vartotojo interfeiso modelis .....	41
4.1.6	Nefunkciniai sistemos reikalavimai .....	44
4.2	Sistemos projektas .....	46
4.2.1	Panaudojimo atvejų realizacijos modelis .....	46
4.2.2	Sistemos architektūra – statinės struktūros modelis .....	46
4.2.3	Sistemos elgsenos modelis .....	47
4.2.4	Duomenų bazės modelis.....	52
4.2.5	Realizacijos modelis.....	60
4.2.6	Reikalavimai sistemos funkcionavimo palaikymui.....	62
5.	Eksperimentinis tyrimas .....	65
5.1	Eksperimentinio diegimo aprašymas.....	65
5.2	Sistemos naudojimo instrukcija.....	67
5.2.1	Prisijungimas.....	67
5.2.2	Matricos elementų parinkimas .....	72
5.2.3	Matricos kūrimas.....	73
5.2.4	Papildomos savybės .....	75
5.3	Sistemos veikimo ir savybių analizė .....	80
5.4	Kokybės kriterijų įvertinimas.....	85
5.5	Sistemos taikymo rekomendacijos .....	88
5.6	Tolimesnio sistemos tobulinimo, plėtojimo galimybės.....	89
6.	Išvados.....	90
7.	Literatūra .....	92
8.	Priedai.....	94
8.1	Panaudojimo atvejų realizuojamos klasės .....	95
8.2	Statinės architektūros modelio klasių diagramos .....	100
8.3	Ribinių klasių, valdiklių ir esybių modelis.....	102
8.4	Sistemos elgsenos modelio papildomos diagramos .....	106
8.5	Sistemos papildymas nauju saugyklos tipu .....	109
8.6	Straipsnis .....	114
8.6.1	Reikalavimų specifikavimas IS kūrimo proceso kontekste .....	114
8.6.2	Reikalavimų specifikacijų kokybė .....	115
8.6.3	Formalizmai reikalavimų specifikacijoms atvaizduoti.....	116
8.6.4	CASE įrankių papildomos priemonės specifikacijų (modelių) analizavimui.....	118
8.6.5	Baigiamieji darbai .....	118

# 1. Summary

## **Research and Development of Tools For Analysis of Requirements Specification**

In the early stage of building a new informational system requirement engineering involves all the processes related to the requirement system and research of problems emerged during this process. In this stage the requirements specification of corresponding information system are built. Many different notations serve for this purpose, selection of which is dependant on available resources and project goals. Requirements specification must be of good quality. It is evaluated by characteristics of quality. Specifications could be analyzed and improved by using various tools including CASE tools. The specification is developed and its quality is improved by making analysis of requirements. Well done specification is basis for creating high leveled functionality system.

## 2. Įvadas

Bet kurią reikalavimų specifikaciją gali analizuoti žmogus. Tačiau esant dideliame sistemos modeliui tai gali būti ypač ilgas ir sudėtingas darbas, kurio rezultatai priklausys nuo specifikaciją analizuojančio darbuotojo kompetencijos ir patirties. Tam, kad palengvinti analitiko darbą, kuriamos įvairios specifikacijų analizės kompiuterizuotos priemonės, kurios palaiko formalią arba pusiau formalią specifikavimo kalbą ir metodus. Tuo būdu efektyviai įmanoma kelti specifikacijų kokybę. Sukurti natūralia kalba užrašytų specifikacijų analizės priemonę būtų ypač sudėtinga. O realizuoti universalų įrankį tiesiog neįmanoma, nes kompiuterizuojamos IS reikalavimų specifikacijos priklauso nuo konkrečių metodų. Skirtingam metodui turi būti skirtos atskiros priemonės.

## 3. Analizės dalis

### 3.1 Tyrimo sritis, objektas ir problema

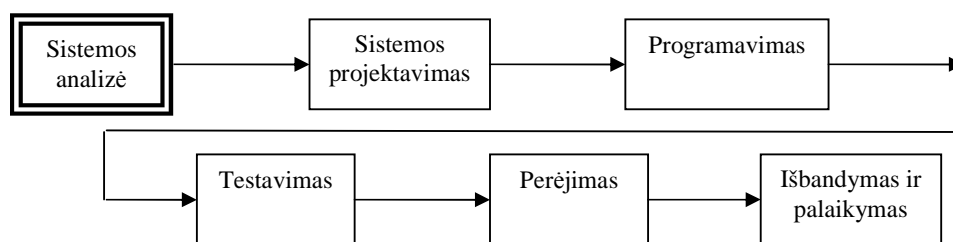
*Tyrimo sritis* – CASE įrankiai, skirti reikalavimų inžinerijai ir turintys papildomo funkcionalumo priemones skirtas analizuoti sudarytas reikalavimų specifikacijas.

*Tyrimo objektas* – Informacijos sistemų katedroje kuriamas CASE įrankis reikalavimams specifikuoti.

Ankstyvojoje kompiuterizuojamos informacinės sistemos kūrimo stadijoje reikalavimų inžinerija apima visus darbus susijusius su reikalavimų sistemai nustatymu ir šio proceso metu išskylančių problemų tyrinėjimu. Šiame etape kuriama atitinkamos informacinės sistemos reikalavimų specifikacija. Jai sudaryti yra pasiūlyta skirtingų notacijų, kurių naudojimas atitinkamu atveju priklauso nuo projekto tikslų ir prieinamų išteklių. Reikalavimų specifikacija turi būti kokybiška. O tai įvertinama atitinkamomis kokybės charakteristikomis. Specifikacijoms analizuoti gali būti panaudotos įvairios priemonės, tame tarpe ir CASE įrankiai. Atliekant reikalavimų analizę, specifikacija plėtojama ir keliama jos kokybė. Kokybiška specifikacija yra pagrindas sukurti aukšto funkcionalumo lygio sistemą.

### 3.2 Reikalavimų specifikavimas IS kūrimo proceso kontekste

Informacinės sistemos (IS) kūrimo darbai prasideda nuo sistemingo reikalavimų išsiaiškinimo, jų analizavimo ir užrašymo. Pradinėje IS kūrimo pakopoje – sistemos analizėje (1 pav. IS gyvavimo ciklo schemeje ji vaizduojama proceso pradžioje) – sutelkiamas dėmesys darbui su reikalavimų pateikėjais, kad išgautumėme visą reikalingą informaciją, apie kompiuterizuojamą sritį. Tinkamai to neatlikus, nebus galima įgyvendinti sistemos su aukšto lygio funkcionalumu, ji tiksliai neatitiks užsakovo poreikių, reikės papildomų pastangų, laiko ir išlaidų klaidų taisymui. Remiantis sukurta reikalavimų specifikacija bus vykdomi kiti IS gyvavimo ciklo etapai. Daug paprasčiau ir pigiau ištaisyti klaidas, aptiktas sistemos kūrimo proceso pradžioje, negu bandyti spręsti neatitikimus diegimo ir palaikymo stadijų metu. [1, 2]



1 pav. Apibendrinta IS gyvavimo ciklo schema.

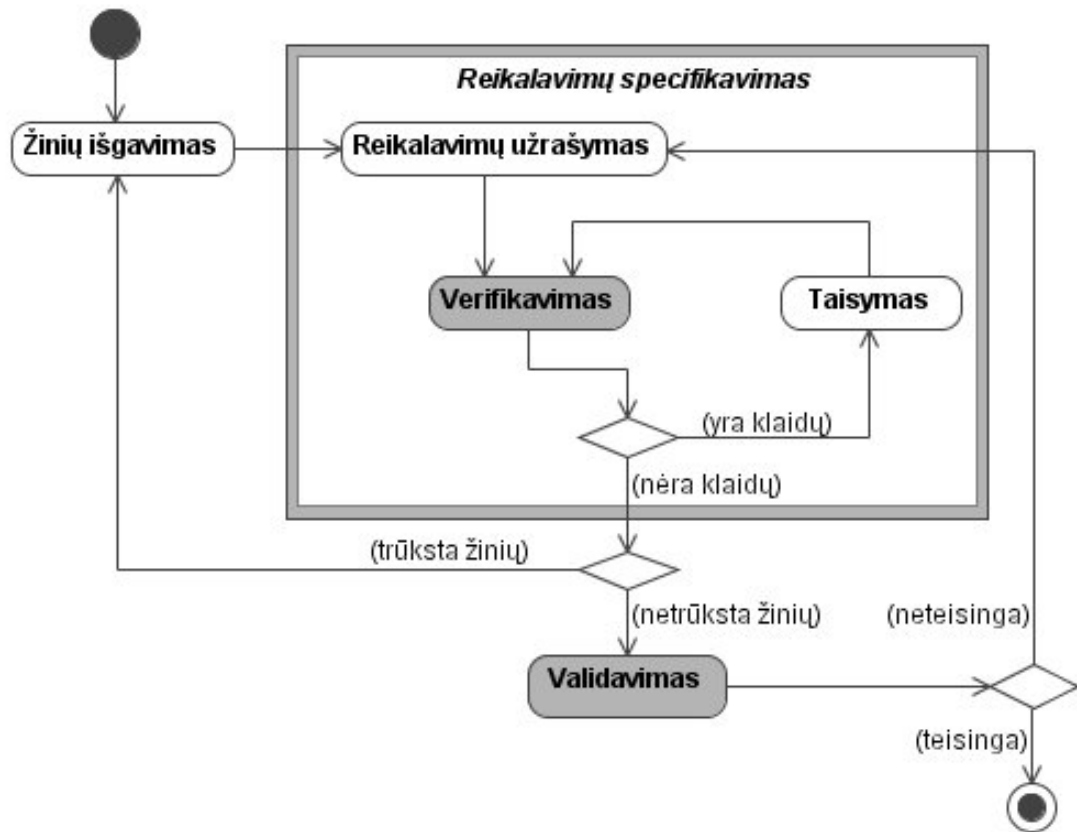
*Reikalavimų inžinerija* – terminas, naudojamas ankstyvojoje sistemos kūrimo stadijoje, kuri apima problemų, susijusių su kompiuterizuojama sritimi, tyrinėjimą, reikalavimus sistemai, vartotojus. Reikalavimų inžinerija pirmiausia prasideda nuo reikalavimų išgavimo. Šios užduoties tikslas yra surinkti ir atvaizduoti srities, kurią turės palaikyti kompiuterizuota informacinė sistema, reikalavimus. Jų užrašymas paprastai vadinamas reikalavimų specifikavimu. Šiam tikslui pasiūlytos skirtingos notacijos - nuo nestruktūrizuotų ir neformalių tekstų iki aukšto lygio formalių matematinių notacijų. Išgauti reikalavimai analizuojami, kad išsiaiškinti kompiuterizuojamą sritį ir išsiaiškinti IS kūrimo problemas, patikrinti reikalavimų specifikacijos kokybę ir ją kelti. Pačią specifikaciją galima apibūdinti kaip architektūrinį, nuo diegimo nepriklausomą būsimosios sistemos planą, kuriame atvaizduojamas užsakovo poreikis įvairioms kuriamos IS savybėms. *IS projektavimo inžinerija* vadinamos tolesnės kūrimo fazės, kuriose naudojama ši specifikacija, skirta funkcionuojančiai sistemai projektuoti ir įdiegti, bei, taip pat, sistemai patikrinti. Jau realizavus sistemą, specifikacija taps vieninteliu šaltiniu, kuris gali būti panaudotas patikrinti ar įdiegta sistema tiksliai atitinka tai, kas buvo užsakyta. Nepriekaištingai sudaryta specifikacija – tai pagrindas gerai funkcionuojančiai informacinei sistemai įgyvendinti su kuo mažesnėmis palaikymo ir pritaikymo išlaidomis. Pasikeitus aplinkai ir, dėl tos pat priežasties,

reikalavimams informacinei sistemai, reikalavimų specifikacija tampa aprašu, skirtu permodeliuoti pokyčius bei aptarti kaip jie turėtų būti suderinti su IS pokyčiais. [ 1, 2, 3, 4]

Reikalavimų inžinerijos tikslas yra sukurti kokybišką specifikaciją. Todėl reikalavimų analizės procese numatyti verifikavimo ir validavimo subprocesai (2 pav.). Jų tikslai yra patikrinti: ar formali ir konceptuali specifikacija yra tinkama ir ar ji tinkamai išreiškia funkcinis reikalavimus, suformuluotus vartotojo. Verifikuojant ir validuojant reikalavimų specifikaciją, gali būti pagerinta reikalavimų specifikacijos kokybė ir tuo būdu sumažintos išlaidos bei rizika tolesniuose IS kūrimo etapuose.

Verifikavimas apima reikalavimų specifikacijos savybių, pateiktų kitame skyrelyje, tikrinimą. Atliekant verifikavimą sprendžiamos sintaksinio ir semantinio lygio problemos. Sintaksinio lygio problema yra įrodyti, kad specifikacija yra pilna ir neprieštaringa. Sprendžiant semantinio lygio problemas reikia įvertinti reikšmes, pateiktas specifikacijoje, bei ieškoti nesuderinamumų. Kuomet dar tik renkami reikalavimai, kiekvienas jų verifikuojamas kokybės kriterijų, tokių, kaip teisingumas, patikrinamumas, keičiamumas ir kt. atžvilgiu. Kitos savybės, tokios, kaip pilnumas ir atsekamumas, įvertinamos tik išgavus pilnus reikalavimus. [3, 5]

Validavimas - tai procesas, kuris užtikrina, kad reikalavimai tiksliai atitiktų užsakovo poreikius. Pagrindinis tikslas – nustatyti nesutapimus tarp to, ko reikalauja klientas, ir to, kas suformuluota reikalavimuose. [6]



2 pav. Reikalavimų specifikacijų sudarymo ir tikrinimo procesas.



### 3.3 Reikalavimų specifikacijų kokybė

Kokybiška specifikacija svarbi tuo, kad būtent ja naudojantis atliekama nuodugni reikalavimų analizė. Remiantis neišsamiai ir netiksliai specifikuotais reikalavimais sudėtinga sukurti tinkamą IS, o bet kokios sukurtos IS kokybė priklauso nuo surinktų ir išanalizuotų pradinių duomenų apie ją. Tiesa, nereiktų tikėtis sudaryti specifikaciją, kurioje kiekvienas reikalavimas pilnai atitinka kokybės charakteristikas. Nepaisant to, kaip išsamiai jie renkami, analizuojami, tikrinami ir tobulinami, jie niekuomet nebus idealūs. Tačiau, jeigu bus laikomasi kokybės charakteristikų, apibūdintų žemiau, sukurta reikalavimų specifikacija bus kokybiškesnė ir, svarbiausia, galutinis produktas labiau atitiks užsakovo poreikius. [7]

Tikėtina, kad, kol nebus pradėta eksploatuoti sistema, nebus rastos jokios klaidos. Suradus trūkumus vėlyvame IS kūrimo etape neišvengiama žymių projekto nuostolių ir papildomų laiko sąnaudų klaidų ištaisymui.

Specifikacijos kokybės charakteristikos gerinamos atliekant reikalavimų specifikacijų analizę. Būtent, norint sukurti tinkamą sistemą, kokybės atžvilgiu reikalavimų specifikacijos turi atitikti minėtas charakteristikas. Įvairūs CASE įrankių papildomi funkcionalumai (aprašomi skyriuje „CASE įrankiuose realizuotos priemonės specifikacijoms (modeliams) analizuoti“), skirti specifikacijų analizei, leidžia gerinti specifikacijų kokybę. Šiame tiriamajame darbe tai aktualu, kadangi nustačius pagal kokius kokybės kriterijus vertinama reikalavimų specifikacija ir patyrinėjus jau egzistuojančius specifikacijos analizės įrankius, galima pradėti kurti savo įrankį.

Žemiau pateikta 1. lentelė vaizduoja informacinės sistemos reikalavimų specifikacijos esmines charakteristikas. Pateiktos būdingos savybės glaudžiai siejasi su reikalavimų specifikacijų analize, leidžiančia šias savybes gerinti. Įvairiuose šaltiniuose [1, 8, 9, 10, 11, 12] paminėta ir kitokių savybių, kurias turėtų atitikti reikalavimų specifikacija, tačiau čia apibendrintos svarbiausios iš jų.

*1. lentelė - Tinkamų reikalavimų specifikacijų charakteristikos*

<b>Reikalavimų specifikavimo kokybės vertinimo kriterijai</b>	<b>Reikšmės</b>
Pilnumas, užbaigtumas (angl.	Pilna reikalavimų specifikacija turi tiksliai apibrėžti visas galimas sąlygas, sistemos galimybes, realaus pasaulio situacijas, su kuriomis bus susidurta, ir galimas reakcijas į jas. Joje turėtų būti įtraukti visi

completeness)	reikalavimai ir aspektai, susiję su funkcionalumu, charakteristikomis, projekto apribojimais ir t.t. Taip pat turėtų būti sukaupti reikalingi atsakymai į nenumatytus įvykius. Nereikia įtraukti aplinkybių, su kuriomis nebus susidurta.
Neprieštarīgumas, darna (anlg. consistency)	Darni specifikacija tai tokia, kurioje nėra prieštaravimų tarp atskirų reikalavimų formuluočių.  Profesorius Pericles Loucopoulos išskiria: <ul style="list-style-type: none"> <li>• <i>Vidinis neprieštarīgumas</i> – jokių prieštaravimų modelyje;</li> <li>• <i>Išorinis neprieštarīgumas</i> – tai, kas galioja probleminei sričiai, turi būti nustatyta ir reikalavimų modelyje.</li> </ul> Išorinio neprieštarīgumo kriterijus turi du aspektus: <ol style="list-style-type: none"> <li>1. Informacija specifikacijoje turi būti korektiška.</li> <li>2. Informacija specifikacijoje turi būti svarbi.</li> </ol>
Tikslumas (anlg. accuracy)	Reikalavimų specifikacija turi tiksliai apibrėžti sistemos gebėjimus realaus pasaulio aplinkoje, taip pat kaip ji su juo siejasi ir sąveikauja.
Minimalumas (anlg. minimality)	Modelyje turi būti saugoma tik tai, kas neabejotinai reikalinga.
Nepertekliškumas (anlg. redundancy)	Modelyje tie patys reikalavimai neturi kartotis.
Keičiamumas (anlg. modifiability)	Vartotojai paprastai keičia savo reikalavimus kuriamai sistemai, todėl modifikuojama specifikacija turi būti tokia, kad būtų galima suderinti atitinkamas struktūras ir charakteristikas. Atkreipkite dėmesį, jog elementų dubliavimas padidina aiškumą, tačiau atlikus pakeitimus atsiranda keblumų.  Loginė bei hierarchinė reikalavimų specifikacijos struktūra neturėtų apsunkinti bet kokią reikalingą modifikavimą.
Klasifikavimas (anlg. ranking)	Atskiri reikalavimų specifikacijos reikalavimai turi būti hierarchiškai klasifikuojami remiantis pastovumu, saugumu, suvoktu įgyvendinimo sudėtingumu arba pagal kitokį kriterijų, kuris

	<p>pagelbsti sudarant specifikaciją. Kuo didesnė ir sudėtingesnė problema, tuo sunkesnė užduotis sudaryti specifikaciją, padedančią suprasti.</p>
<p>Testuojamumas (angl. testing)</p>	<p>Reikalavimų specifikacija turi būti sudaryta taip, kad leistų patikrinti ar sistema atitinka reikalavimus.</p>
<p>Trasuojamumas (angl. traceability)</p>	<p>Reikalavimų trasavimo procesas užtikrina, kad reikalavimai, apibrėžti specifikacijoje, yra įvykdomi. Reikia įvertinti kiekvieną reikalavimą kaip atskirą formuluotę, kuri gali būti trasuojama, tuomet projekto komanda gali įsitikinti, kad sistemai nustatyti reikalavimai išpildomi. Kiekvienas reikalavimas yra patikrinamas, kad būtų galima priskirti atitinkamą užduotį, skirtą išpildyti šį reikalavimą. Tuo būdu galima tiksliau įvertinti planavimą, išlaidas ir resursus.</p>
<p>Nedviprasmiškumas (angl. unambiguoty)</p>	<p>Specifikacija turi sudaryti tik vienareikšmiškai suprantamus reikalavimus. Jie negali turėti kelių prasmių. Daugiausia su tuo susiduriama naudojant natūralią kalbą. Kad išvengtų to, tinkamai turėtų būti naudojamos formalios ar pusiau formalios kalbos. Taip pat, dviprasmiškumo rizika mažinama naudojant struktūrizuoto modeliavimo būdus, modelių tobulinimą įtraukiant vartotojų.</p>
<p>Pagrįstumas (angl. validatability)</p>	<p>Reikalavimai turi atitikti jų pateikėjų poreikius ir norus. Jie turi būti pagrįsti ir atsakyti į klausimus:</p> <ul style="list-style-type: none"> <li>• Ar įmanoma užtikrinti, kad kiekvienas reikalavimas yra tai, ko reikia užsakovui?</li> <li>• Ar įmanoma užtikrinti, kad kiekvienas reikalavimas yra tai, ko reikia vartotojui?</li> <li>• Ar įmanoma užtikrinti, kad kiekvienas reikalavimas yra tai, ko reikia rinkos atstovui?</li> </ul> <p>Pagrįsta specifikacija tai tokia, kurioje visi projekto dalyviai (vadybininkai, projektuotojai, užsakovai) gali ją suprasti, analizuoti ir pritarti. Tai viena iš pagrindinių priežasčių kodėl reikalavimų specifikacijos rašomos natūralia kalba.</p>

<p>Patikrinamumas (angl. verifiability)</p>	<p>Reikalavimai visuomet turi pradinius informacijos šaltinius, todėl svarbu, kad jie juos ir atitiktų. Taip pat, reikia, kad reikalavimai atitiktų naudojamus standartus, rekomendacijas bei šablonus. Tam, kad reikalavimų specifikacija būtų patikrinama viename abstrakcijos lygyje, ji turi būti besiderinanti su abstrakcija kitame lygyje. Todėl reikalavimai turi būti patikrinami (įrodomi):</p> <p>Ar galima patikrinti kiekvieną reikalavimą su jo šaltiniu?</p> <p>Ar galima patikrinti kiekvieną reikalavimą su jo susietu standartu, rekomendacijomis ir/arba šablonu?</p>
<p>Tinkamumas naudoti (angl. usability)</p>	<p>Taip pat, kaip ir taikomosios programos, reikalavimai turi daugelį vartotojų (pvz.: vadovų, klientų, rinkos, vartotojų atstovai, kūrėjai, projektuotojai, testuotojai, palaikymo personalas), kurie naudoja juos įvairiais tikslais. Todėl atitinkami reikalavimai turi būti praktiškai daugybei tarpininkų:</p> <ul style="list-style-type: none"> <li>• Ar visi reikalavimai suprantami ir patogūs užsakovų ir vartotojų tarpininkams, kurie turi jais naudotis srities valdymui ir įvertinimui.</li> <li>• Ar visi reikalavimai suprantami ir patogūs vadovams, kurie turi naudotis jais tiek srities, tiek išlaidų, planavimų valdymui, eigos metrikoms.</li> <li>• Ar visi architektūros požiūriu svarbūs reikalavimai suprantami ir naudingi kūrėjams, kurie remiantis jais kurs struktūrą?</li> </ul> <p>Ar visi reikalavimai suprantami ir patogūs projektuotojams ir programuotojams, kurie juos turi įgyvendinti? [9, 10, 11, 12]</p>

### 3.4 Formalizmai reikalavimų specifikacijoms atvaizduoti

Specifikacijų analizavimas priklauso nuo jų sudarymo notacijų – naudojantis analizės įrankiu gauti analizės išvadas automatiškai galima tik tuomet, jeigu analizuojama specifikacija yra formali. Analizuojant pusiau formalias specifikacijas, rezultatai analitikui pateikiami patogesne forma, o išvados ir specifikacijos kokybės kėlimo užduotys priklauso nuo jo tolesnio darbo.

Kompiuterizuotos informacinės sistemos reikalavimų specifikacijoms sudaryti naudojami įvairūs formalizmai. Formali specifikacija ir neformali specifikacija – du priešingi reikalavimų specifikavimo būdai. Be to, taip pat egzistuoja tam tikrų tarpinių atvaizdavimų, kurie pasirodė naudingi kaip bendradarbiavimo priemonės tarp sistemos užsakovo ir jos kūrėjo. Pusiau formalūs žymėjimai taip pat naudojami aprašyti projektą. Jie neturi standartinės notacijos. Formali specifikacija tai yra tam tikro savybių rinkinio, kurį turi atitikti sistema, išraiška, pateikta tam tikroje formalioje kalboje ir tam tikrame abstrakcijos lygyje. 2. lentelė vaizduoja formalumų skirtingas kategorijas bei atitinkamus pavyzdžius.

2. lentelė - Notacijų klasifikavimas.

Neformalios	Pusiau formalios	Formalios
Šie metodai neturi išbaigtų taisyklių rinkinių, kuriais būtų apribojami kuriami modeliai. Būdingi atvejai - natūrali kalba (parašytas tekstas) ir nestructūrizuoti vaizdai.	Šie metodai turi apibrėžta sintaksę. Būdingi atvejai - grafiniai būdai turintys tikslias taisykles, kurios tiksliai apibūdina sąlygas pagal kurias leidžiamas konstravimas, tekstiniai bei grafiniai aprašymai, esant ribotoms patikrinimo galimybėms.	Šie metodai turi griežtą apibrėžtą sintaksę ir semantiką. Sudaromas esminis teorinis modelis, pagal kurį gali būti patikrintas aprašymas, išreikštas matematine notacija. Būdingi atvejai - specifikavimo kalbos, grindžiamos logika.
Pavyzdžiai:		
Specifikacijos natūralia kalba	Duomenų/valdymo, srautų diagramos; esybių ryšių diagramos; panaudojimo atvejų diagramos	Petri tinklai; VDM; Z

Terminas “formalus” dažniausiai yra susijęs su “tikslumu”. Specifikacija yra formali, jei ji išreikšta kalboje, specialiai sukurtoje programinės įrangos specifikavimui ir turinčioje tris komponentus:

- taisyklės, nustatančios sakinių gramatinį žymėjimą (sintaksė);
- tam tikra elementų aibė, skirta sakinių interpretavimui atitinkamos srities ribose (semantika);
- taisyklės, naudingai informacijai iš specifikacijos išvesti; jas turi atitikti specifikuojamas objektas. Šis komponentas sudaro pagrindą automatizuotai specifikacijų analizei.

Formalūs metodai propaguojami kaip vienas iš būdų charakterizuoti reikalavimus esant matematiniam tikslumui ir griežtumui. Jie nusako būdus kurti sistemos modeliams tiksliai apibrėžiant jų struktūrą ir elgseną. Naudojant įrankius, pagrįstus formaliais specifikavimo metodais, galimas automatizuotas savybių patikrinimas sudėtinguose modeliuose. [12, 13, 14]

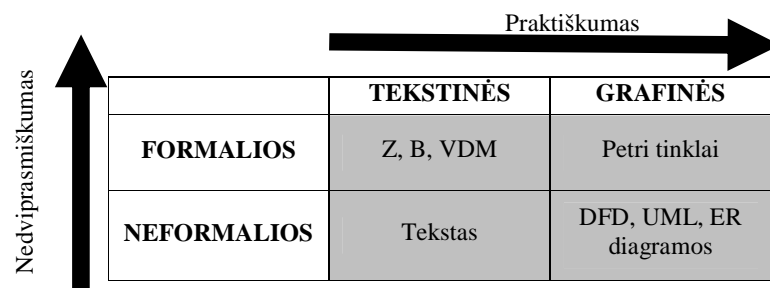
Formalių kalbų pranašumas tas, jog jos yra nedviprasmiškos ir tikslios, ir jas galima palaikyti kompiuteriais bei matematinėmis analizėmis. Tačiau visa tai turi savo kainą - personalas turi išmokti atitinkamą kalbą (kuri daugumai žmonių atrodo sudėtinga) ir formalios specifikacijos sudarymui reikia daugiau laiko. Kadangi grindžiami matematinėmis struktūromis ir notacijomis, priešingai nei natūrali kalba, formalūs metodai yra ne tokie intuityvūs nespecializuotai publikai. [7]

Natūrali kalba yra vienas iš būdų, skirtų padėti vartotojui sudaryti taikomosios srities reikalavimų specifikaciją, kuri kuriama jo pateikiamų teiginių atžvilgiu. Vaizdavimas natūralia kalba yra pradinė reikalavimų formuluotės nustatymo forma. Ji yra patogi, nes leidžia ne techninio pobūdžio vartotojams suprasti sistemos reikalavimus bei ja galima nusakyti daugybę smulkių kompiuterizuojamos srities detalių. Reikalavimų inžinerijoje natūrali kalba yra dažniausiai naudojamas vaizdavimo būdas reikalavimų sudarymui. Ji yra universali, lanksti ir plačiai taikoma, bet, deja, iš prigimties dviprasmiška bei neapibrėžta. Ją naudojant sudėtinga perprasti visos sistemos reikšmę, kadangi žvelgiama tik į tekstinę specifikaciją. O iš anksto nusakytos semantikos trūkumas padidina tikimybę klaidų, atsirandančių dėl suvokimo ir būdingų dviprasmybių. Taip pat egzistuoja kitos problemos naudojant natūralią kalbą. Dėl šios priežasties programinės įrangos kūrėjai projektuoja ir diegia sistemas, kurios neveikia taip, kaip užsakė vartotojas, kadangi projektuotojo reikalavimų specifikacijos supratimas gali skirtis nuo užsakovo. Kaip alternatyva – grafinės

notacijos - paprastai aiškesnės, tačiau jas sudėtingiau praplėsti. Trečiame paveiksle pavaizduotos tekstinės bei grafinės charakteristikos yra ortogonalios specifikacijų formalumo atžvilgiu. [15]

Neformalią specifikaciją sudaryti dažniausiai nesudėtinga, tačiau tokiu būdu gauta ji būna prastos kokybės ir sudėtinga atlikti reikalavimų analizę. Todėl neužtikrinama, jog specifikacijos bus išbaigtos, nedviprasmiškos ir jos pilnai atitiks funkcinių ir nefunkcinių reikalavimų charakteristikas. Tačiau daugelis specialistų plačiai naudoja neformalias specifikacijas, kadangi formalūs metodai laikomi sudėtingais ir brangiais. Formalioms specifikacijoms sudaryti reikia daugiau laiko ir specialistų pagalbos, bet jos palaiko analizę ir sumažina testavimų pastangas.

Grafinė notacija yra vienas iš galimų natūralios kalbos panaudojimo alternatyvų, skirta reikalavimų specifikavimui. Ji yra plačiai paplitusi ir priimtina, tam tikru mastu naudojama reikalavimų inžinerijoje. Grafinis žymėjimas apima elementus, skirtus aprašyti sistemos struktūrą ir elgseną. Daugeliu atvejų informacija suvokiama lengviau, kai ji pateikiama grafiškai. Egzistuojančių išsamių iliustruotų vaizdų dėka vartotojai, neapmokyti formalių metodų, tinkamai supranta specifikaciją ir sumažėja jų supratimo atotrūkis nuo išmanančių griežtą formalizmą. Grafinė specifikacija tai tokia, kurios elementai yra vizualiniai, ne tekstiniai. Tačiau, nepaisant kalbinės atitikties, grafinės specifikacijos nėra taip paprastai susijusios su jų tekstinėmis kopijomis.



3 pav. – Reikalavimų specifikavimo notacijų klasifikavimas.

Manoma, jog natūralios kalbos, grafinės notacijos ir formalių metodų derinys yra būtinas reikalavimų inžinerijos problemų sprendimui:

- Reikalavimų specifikacija turi būti prieinama formatu tinkamu ne techninio pobūdžio vartotojams;
- Formalūs metodai turi būti panaudoti reikalavimams modeliuoti ir valdyti, kad padėti projektuotojams iš anksto rasti nesuderinamumus.
- Vartotojui palankios ir formalios reikalavimų versijos turi būti sinchronizuojamos automatiškai;

- Turėtų būti panaudotas automatizavimas, skirtas palengvinti formaliems modeliams generuoti iš natūralios kalbos reikalavimų, tuo būdu supaprastinant reikalavimų inžinerijos užduotį kuriant modelius nuo pradžių. [4, 7, 12]

Sudėtinga paruošti konkrečią, formalią ir visiems projekto dalyviams suprantamą reikalavimų specifikaciją. Sistemos vartotojas turi patikrinti visus reikalavimus, nes jis yra jų pateikėjas. Tačiau tai įmanoma tik jei reikalavimai pavaizduoti jam suvokiama forma. Reikalavimų specifikacijos kūrimui eilinis vartotojas gali pasitelkti paprasčiausią ir jam neabejotinai suprantamą formą – natūralią kalbą. Tačiau tokiu būdu sudarytas specifikacijas sudėtinga analizuoti. Formalūs metodai netaikomi ankstyvosiose reikalavimų specifikavimo stadijose, kai reikalavimai yra migloti. Formalizmai, naudojami tradiciniuose metoduose reikalavimams specifikuoti, yra nepriimtini vartotojui, jeigu jis neturi specialių žinių jiems suprasti. Tačiau svarbu nepamiršti, jog būtina kelti reikalavimų specifikacijos kokybę. Formalių specifikacijų (sudarytų formaliais metodais, formaliomis kalbomis) privalumas tas, kad mes galime patikrinti jų teisingumą, atitikimą kokybės charakteristikoms. O svarbiausia, jog galime jas analizuoti ir, tuo būdu, gerinti kokybę. Tokiais atvejais talkintų CASE įrankiai. Tačiau egzistuoja tam tikri apribojimai. Ypač sudėtinga būtų sukurti CASE įrankį, kuris leistų atlikti reikalavimų specifikacijų, parašytų natūralia kalba, analizę. Kadangi natūrali kalba nėra formalizuota ir neturi apibrėžtų griežtų taisyklių reikalavimams specifikuoti, šiuo būdu sudarytas specifikacijas gali analizuoti tik žmogus, pvz.: analitikas. Specifikacijos kokybės gerinimas priklausys nuo jo kompetencijos ir patirties. Visiškai priešinga situacija naudojant formalias kalbas ir metodus, kurie turi griežtas taisykles reikalavimams specifikuoti. Tokioms specifikacijoms analizuoti nėra ypač sudėtinga sukurti kompiuterizuotą priemonę, kadangi iš anksto žinomos reikalavimų užrašymo taisyklės. Naudojantis šiais įrankiais galima tiksliai rasti klaidas, padarytas specifikacijoje ir tuo būdu efektyviai kelti jos kokybę. Taip pat naudojantis CASE įrankiais įmanoma atlikti ir specifikacijų, užrašytų pusiau formaliomis kalbomis, analizę. Tačiau šiuo atveju, skirtingai nuo specifikacijų, sudarytų formaliais metodais ir kalbomis, atlikta analizė nebus tokia išsami ir tiksli. Taip yra dėl to, kad šiuo būdu sudarytose specifikacijose klaidų ir neatitikimų paiešką galime kompiuterizuoti tik iš dalies. Pusiau formalios kalbos turi tam tikras taisykles reikalavimams specifikuoti, tačiau jos nėra griežtos. Sukurta analizės priemonė pateiktų ne konkrečius rezultatus, o tik tam tikrą specifikacijos sritį, kur tikėtini netikslumai ir klaidos.



### **3.5 CASE įrankiuose realizuotos priemonės specifikacijoms (modeliams) analizuoti**

Reikalavimų specifikavimas yra sudėtingas darbas, kur svarbų vaidmenį atlieka automatizuoti specifikavimo įrankiai. Bet kurią reikalavimų specifikaciją gali analizuoti žmogus. Tačiau esant dideliame sistemos modeliui tai gali būti ypač ilgas ir sudėtingas darbas, kurio rezultatai priklausys nuo specifikaciją analizuojančio darbuotojo kompetencijos ir patirties. Tam, kad palengvintume analitiko darbą, kuriama įvairi specializuota programinė įranga, kuri įgalina efektyviau analizuoti ir tokiu būdu pakelti specifikacijų kokybę. Paprastai ši PĮ geba analizuoti tik tam tikra formalia arba pusiau formalia kalba užrašytas specifikacijas. Sukurti natūralia kalba užrašytų specifikacijų analizės priemonę būtų ypač sudėtinga. O realizuoti universalų įrankį tiesiog neįmanoma, nes taisyklės, pagal kurias sudaromos specifikacijos ir kurių laikymąsi/pažeidimą tikrina įrankiai analizės metu, bus skirtingos skirtingiems specifikavimo metodams. Atskiram metodui turi būti skirtos atskiros priemonės.

Reikalavimų informacinei sistemai specifikacijos susideda iš diagramų ir modelių. Žvelgiant smulkiau, juos sudaro įvairūs elementai. Problema iškyla tada, kai informacinių sistemų analitikui (gali būti ir kitam vartotojui) reikalinga žinoti tam tikrų elementų tarpusavio ryšius bei priklausomybes. Esant šiems modeliams ir diagramoms nedidelėms, elementų tarpusavio sąryšius nesunkiai galima atsekti tiesiog žvelgiant į grafinę specifikaciją. Tačiau tai tampa ypač sudėtinga ir painu, kai diagramas ir modelius sudaro labai daug elementų ir jie tarpusavyje susiję daugybe ryšių. Problema tampa opesne, jeigu reikalinga atrasti sąryšius tarp besisiejiančių elementų, esančių skirtinguose modeliuose. Sistemų analitikui belieka rankiniu būdu tyrinėti specifikacijas ir ieškoti jų dominančių ryšių. Tačiau tai gana neefektyvus ir laiko sąnaudų reikalaujantis darbas. Kuriama sistema – sprendimas kompiuterizuoti tokio pobūdžio darbą.

Daugelis CASE įrankių pasižymi papildomais funkcionalumais, skirtais analizuoti sudarytus modelius. Tai labai aktualu, kai šie modeliai dideli ir sudėtingi. Galima patikrinti jų savybes, elementų priklausomybes, struktūros ir elgsenos santykį, atitikimą reikalavimams. Analizuojant modelius gerinamos sistemos savybės, geriau išsiginama į probleminę sritį, mažinamos papildomos išlaidos, taupomas kūrimo laikas, keliama reikalavimų specifikacijos kokybė. Palyginimui nubraižyta 3. lentelė, kurioje pateikti kai kurie CASE įrankiai, turintys papildomas priemones kompiuterizuojamos sistemos modelių analizavimui.

3. lentelė – CASE įrankiai, turintys papildomų funkcionalumų reikalavimų specifikacijų analizavimui.

Gamintojas	Pavadinimas	Paskirtis	WWW adresas	Vartotojai pasaulyje	Metodas
Oracle	Oracle Designer - Matrix Diagrammer	Matricių pavidalu kuriami ir vaizduojami ryšiai tarp saugykloje saugomų skirtingų tipų elementų. Sankirtos elementas parodo kaip vienas objektas priklauso nuo kito. [17]	<a href="http://www.oracle.com">http://www.oracle.com</a>	MasterCard International; Cisco Systems; Canon; LG.Philips LCD; ViewSonic; Alcatel; SDI Corp.; Toshiba Australia;	CDM (Oracle's Custom Development Method)
IBM-Rational	Rational RequisitePro - Use case Attribute Matrix View	Naudojant šį įrankį, pateikiamas visų arba tik atrinktų panaudojimo atvejų bei jų atitinkamų atributų poaibis. Tai naudinga sudarant panaudojimo atvejus, nustatant kuris panaudojimo atvejis pavestas kuriam projektuotojui, koks jų įgyvendinimo sudėtingumas, ar kokia tvarka jie turėtų būti realizuoti. [18]	<a href="http://www.rational.com">http://www.rational.com</a>	Finnair; The Royal Bank of Canada; Siemens Medical Solutions; UK Post Office; Automatic Data Processing; Danske Bank; LG Capital; DATEV;	RUP (Rational Unified Process)
	Rational RequisitePro - Traceability Matrix	Atsekamumo matricioje vaizduojami ryšiai tarp panaudojimo atvejų ir reikalavimų sistemai. Nustatant santykius tarp tikslų ir kuriamos sistemos savybių, galima tiksliai nustatyti kokių tikslų nebūtų įmanoma išpildyti, jeigu nuspręstume nerealizuoti tam tikros savybės. Naudojant atsekamumo matricią, galima atsakyti į klausimą: “Kokie panaudojimo atvejai siejasi su atitinkamais sistemos reikalavimais?” Ir taip pat, jei srities poreikiai pasikeičia, tai kokie panaudojimo atvejai įtakojami?			
Proforma Corporation	ProVision Workbench - Completeness Checker	Automatizuotas pilnumo tikrinimo įrankis leidžia analizuoti visą modelį bei atlikti tokias užduotis, kaip trūkstamų aprašymų, savybių ar praleistų ryšių patikrinimą. Tikslas – užtikrinti, kad modelio objektai ir jų ryšiai būtų išsamūs ir tikslūs.	<a href="http://www.proformacorp.com">http://www.proformacorp.com</a>	US Air Force; Hewlett Packard; Autodesk; CompuCredit; Seagate; PeopleSoft; Motorola; Warner Bros;	ProGuide

Tęsinys (3. lentelė).

Gamintojas	Pavadinimas	Paskirtis	WWW adresas	Vartotojai pasaulyje	Metodas
Proforma Corporation	ProVision Workbench-Model Comparison	Įrankis skirtas nustatyti dviejų modelių panašumams ir skirtumams. [19]	<a href="http://www.proformacorp.com">http://www.proformacorp.com</a>	US Air Force; Hewlett Packard; Autodesk; CompuCredit; Seagate; PeopleSoft; Motorola; Warner Bros;	ProGuide
No Magic, Inc.	MagicDraw UML - Compare Projects	Tai vizualus modelių skirtumų pavaizdavimas. Galima nustatyti pakeitimus tarp dviejų skirtingų modelio versijų. Atvaizduojami tokie pakeitimai, kaip nauji elementai, ištrinti elementai, elementai su pakeistais duomenimis, vidiniai elementų pakitimai.	<a href="http://www.magicdraw.com">http://www.magicdraw.com</a>	IBM; Sun Microsystems; BEA Systems; Xerox; Siemens AG; British Columbia University; Daimler Chrysler; Ford;	Nuo konkretaus metodo nepriklauso
	MagicDraw -UML Metrics	Metrika, tai skaitinė reikšmė, skirta įvertinti modelį, kuri apskaičiuojama ir gali būti: <ul style="list-style-type: none"> <li>• UML modelio metrikos</li> <li>• Sistemos metrikos</li> <li>• Reikalavimų metrikos.</li> </ul>			
	MagicDraw - Usages / Dependent Elements	Šis papildomas funkcionalumas padeda rasti informaciją apie tai, kaip modelio elementai siejasi su kitais elementais, kokie ryšiai tarp diagramoje panaudotų ir kitų priklausomų elementų. Tai labai naudinga analizuojant santykius tarp elementų ar ieškant diagramų kur šie elementai atvaizduoti. [20]			
Sparx Systems	Enterprise Architect Professional - Element Relationship Matrix	Sąryšių matrica leidžia tyrinėti ryšius tarp modelio elementų. Naudojantis ja taip pat galima kurti, redaguoti ir ištrinti ryšius tarp elementų. Ši matrica, tai patogus ryšių vizualizavimo būdas. [21]	<a href="http://www.sparxsystems.com">http://www.sparxsystems.com</a>		E-ValueFlow™ Designer

3. lentelėje pateikti įrankiai skirti analizuoti pusiau formalias reikalavimų specifikacijas. Tai reiškia, kad specifikacijų automatiškai išanalizuoti neįmanoma ir galutinėms išvadoms gauti papildomai reikalingas analitiko darbas. Šiomis kompiuterizuotomis priemonėmis automatizuotai ir specialistui priimtinesne forma pateikiami tam tikri rezultatai. Todėl specifikacijų kokybė nepriklauso vien tik nuo analizės įrankio. Remiantis analizės priemonėmis gautais rezultatais specifikacijos kokybė gali būti keliami tolimesniu analitiko darbu, grindžiamu jo žiniomis ir patirtimi.

3. lentelėje aprašytuose įrankiuose dažniausiai naudojama analizės forma yra dvimatė matrica. Keletas priežasčių ją naudoti:

- Tai patogus ir nesudėtingas būdas analizei atlikti. Dažnai matricos pagalba ieškoma priklausomybių tarp dviejų tipų objektų, pateikiamų eilutėse ir stulpeliuose. Tuo tarpu sankirtų celės identifikuoja tarpusavio ryšius. Svarbiausia, jog šiuo būdu paprasčiau galima identifikuoti specifikacijų kokybės charakteristikas, pavyzdžiui: pilnumas, minimalumas, neprieštaringumas, nepertekliškumas.
- Modelio reprezentatyvumas ir kompaktiškas sistemos vizualizavimas. Priešingai negu paprastoje tekstinėje formoje, labiau priimtina analizuoti reikalavimų specifikaciją, kuomet jos elementai sistemingai ir pagal tam tikras savybes ir/ar taisykles pateikti matricoje (taip pat ir kitose analizės formose). Taip pat naudojamos tam tikros papildomos matricos modifikacijos: subcelės, apibendrinančios celės ir kt.

Tam, kad aukščiau aprašytus analizės įrankius įvertintume pagal įvairius kriterijus ir palygintume tarpusavyje, nubraižyta 4. lentelė. Tyrinėjant analizės priemones, stengtasis nustatyti bendriausius kriterijus. Jų paaiškinimai:

1. Rezultatų pateikimo forma – koku būdu pateikiami analizės rezultatai.
2. Integravimas su CASE įrankiu – analizės priemonė yra atskira programa ar neatsiejamai integruota į CASE įrankį.
3. Galimybė redaguoti ir/ar kurti objektus pateiktus rezultatų formoje – kai kurie įrankiai tiesiogiai leidžia redaguoti ir/arba kurti, ištrinti objektus, pateikiamus analizės formoje.
4. Priklausomumas nuo metodo – analizės įrankis priklausantis nuo IS kūrimo metodo ar ne.
5. Priklausomumas nuo saugyklos tipo – vieni analizės įrankiai leidžia analizuoti specifikacijas, saugomas tik vieno tipo saugykloje, o kitais galima atlikti kelių

tipų saugyklų analizę.

6. Priklausomumas nuo saugyklos struktūros – žymi, ar įrankio funkcionalumas priklauso nuo specifikacijų saugyklos struktūros.
7. Įrankiui reikalinga lokali jo DB – vieni įrankiai funkcionalumui palaikyti naudoja lokalią duomenų bazę, kiti duomenis saugoja kitu būdu. Pavyzdžiui, „Enterprise Architect“ priemonė naudoja „.eap“ failus. Tuo tarpu „MagicDraw“ paremtas XML‘u.
8. Analizės rezultatų išsaugojimas – nusakoma, ar analizės rezultatus galima įrankiu tik peržiūrėti, ar taip pat išsisaugoti į failą.
9. Analizės rezultatų filtravimas ir/arba rūšiavimas – esant daug išvestų rezultatų, patogu būtų juos surūšiuoti ir/arba atlikti filtravimą.
10. Analizės rezultatų vaizdavimo nustatymai: šriftas, jo dydis ir/arba kt. – kai kurie įrankiai leidžia rezultatų lange pakeisti vizualinį jų pateikimą.
11. Keli rezultatų pateikimo režimai – vartotojo patogumui ir rezultatų aiškumui įrankyje gali būti realizuoti keli skirtingi tų pačių rezultatų pateikimo būdai.
12. Pradinių elementų nurodymas prieš skaičiuojant/išvedant analizės rezultatus – tam, kad įrankis pateiktų analizės rezultatus, pradiniai duomenys gali būti nurodomi pirmiausiai. Tačiau toks veiksmų eiliškumas yra ne visuose įrankiuose.
13. Nutolusios saugyklos analizavimas – šiuo kriterijumi apibrėžiama, ar galima įrankiu analizuoti specifikacijas saugoma kitame kompiuteryje (serveryje).

4. lentelė – CASE įrankių palyginimas

Gamintojas	Oracle Designer	Rational RequisitePro		ProVision Workbench		MagicDraw UML			Enterprise Architect
Įrankis	Matrix Diagrammer	Use case Attribute Matrix View	Traceability Matrix	Completeness Checker	Model Comparison	Compare Projects	Metrics	Usages / Dependent Elements	Element Relationship Matrix
Kriterijus									
1. Rezultatų pateikimo forma	matrica	matrica	matrica	lentelė	lentelė	lentelė	lentelė	lentelė	matrica
2. Integravimas su CASE įrankiu	√	√	√	√	√	√	√	√	√
3. Galimybė redaguoti ir/ar kurti objektus pateiktus rezultatų formoje	√	-	-	-	√	√	-	√	√
4. Priklausomumas nuo metodo	√	√	√	√	√	-	-	-	√
5. Priklausomumas nuo saugyklos tipo	√	√	√	√	√	√	√	√	-
6. Priklausomumas nuo saugyklos struktūros	√	√	√	√	√	√	√	√	√
7. Įrankiui reikalinga lokali jo DB	√	√	√	-	-	-	-	-	-
8. Analizės rezultatų išsaugojimas	√	√	√	-	-	-	√	-	√
9. Analizės rezultatų filtravimas ir/arba rūšiavimas	√	√	√	√	√	√	√	√	-
10. Analizės rezultatų vaizdavimo nustatymai: šriftas, jo dydis ir/arba kt.	√	-	-	-	-	-	-	-	-
11. Keli rezultatų pateikimo režimai	√	-	-	-	-	-	-	-	-
12. Pradinių elementų nurodymas prieš skaičiuojant/išvedant analizės rezultatus	√	√	√	√	√	√	√	√	-
13. Nutolusios saugyklos analizavimas	√	-	-	-	-	-	-	-	√

### 3.6 Analizės išvados

Atlikus tiriamojo darbo analizę, nustatyta, jog neegzistuoja universalus reikalavimų specifikacijų analizės įrankis, nepriklausantis nuo specifikavimo metodų, specifikacijų sudarymo formalizmo ir saugyklų tipų. Tačiau jų visų pagrindinis tikslas – padėti sistemų analitikams kelti reikalavimų specifikacijų kokybę. Išanalizavus esamus įrankius pusiau formalioms reikalavimų specifikacijoms analizuoti, nustatyta:

- Naudojant analizės įrankius reikalavimų specifikacijoms analizuoti, konkrečios išvados nepateikiamos. Tad sistemų analitikui lieka dar daug darbo, priklausančio nuo jo žinių ir patirties.
- Objektų, saugomų specifikacijoje, tarpusavio ryšiams atvaizduoti tinkamiausia analizės forma yra matrica. Išnaudojama jos savybė – galimybė gauti objektų, pateiktų matricos eilutėse ir stulpeliuose, sankirtos celes.
- Visi egzistuojantys specifikacijų analizės įrankiai priklauso nuo saugyklos struktūros. Kuriant savo įrankį bus į tai atsižvelgta, nes įmanoma jį praplėsti, kad nepriklaustų nuo metodo ir juo būtų galima analizuoti bet kokias specifikacijas, veikiančias „MS SQL Server“ pagrindu. T.y. priešingai negu jau egzistuojantys įrankiai, kuriamą sistemą siekiama realizuoti kaip galima universalesnę.
- Vieninteliu ištyrinėtu analizės įrankiu „Element Relationship Matrix“ galima analizuoti specifikacijas, patalpintas keliose skirtingo tipo saugyklose. Kuriant savo įrankį siekiama, jog juo būtų galima analizuoti specifikacijas, saugomas ne tik „MS SQL Server“ tipo saugyklose.

## 4. Projekto dalis

Kuriamos sistemos modeliavimui pasirinktas RUP (Rational Unified Process) procesas bei „MagicDraw UML“ modeliavimo priemonė. Sekančiuose skyreliuose (3.2 ir 3.3) pateikti detalūs kuriamos sistemos UML modeliai, kiekvienas jų aprašytas ir pakomentuotas atskirai. Modeliavimo procesas apima visus etapus nuo sistemos konteksto išskyrimo iki realizacijos modelio sukūrimo. Diagramos, nusakančios smulkesnes sistemos struktūras ir elgsenos detales, pateiktos priede.

### 4.1 Reikalavimų modelis

Pasiūlyta reikalavimų specifikacijų analizės forma yra matrica. Būtent bus panaudota jos savybė sukirsti eilutes ir stulpelius. Eilučių ir stulpelių sankirtos žymės dviejų objektų tarpusavio priklausomybę. Vartotojas, norėdamas atrasti sąsajas tarp konkrečių reikalavimų specifikacijų elementų, tiesiog sistemai juos nurodo. T.y. pažymi tuos elementus, kuriuos nori matyti kaip eilučių aibę; tuos, kuriuos nori matyti kaip stulpelių aibę; tuos, kuriuos nori matyti sankirtose. Tam, kad sistema suformuotų reikalavimų specifikacijų elementų priklausomybių matricą, šiuos tris elementus nurodyti yra būtina. Be to, priklausomai nuo vartotojo poreikių, sistemai galima nurodyti daugiau detalių, kad po to suformuota matrica būtų išsamesnė ir teiktų daugiau informacijos. T.y. neskaitant eilučių, stulpelių ir sankirtų aibių, jei tik egzistuoja atitinkami ryšiai, vartotojas gali nurodyti pasirinktos eilutės bei pasirinkto stulpelio apibendrinančias aibes.

Reiktų paminėti tai, jog Informacijos sistemų katedroje sukurto CASE įrankio saugykloje, priklausančioje nuo ODRES metodo, yra sukurtos lentelės kiekvienam modelių elementui atskirai. Pavyzdžiui, lentelėje „Esybe“ saugomos visos nagrinėjamo organizacijos veiklos konteksto esybės; lentelėje „Rsys“ saugomi visi nagrinėjamo organizacijos veiklos konteksto ryšiai tarp esybių. Saugyklos lentelių stulpeliuose saugomi tų elementų atributai – pavadinimas, kodas ir kt. Nustatant parametrus formuojamai matricai gauti, t.y. vartotojui nurodžius eilutę, stulpelį, sankirtą ar apibendrinančius elementus, jis turėtų taip pat pažymėti kiekvieno jų atributus, kuriuos norėtų matyti matricoje. Pavyzdžiui, nurodoma eilutė „Esybe“ ir pažymimas jos atributas „n\_vardas“. Suformuotoje matricoje konkrečioje eilutėje vartotojas matys pažymėtos eilutės vardą (reikšmę). Toks pat principas ir su stulpeliais, sankirtomis bei apibendrinančiais elementais. Reiktų paminėti tai, jog vartotojas turi galimybę pasirinkti, kad eilučių ir stulpelių sankirtos būtų vaizduojamos ne kertančio elemento reikšme (pažymėtu

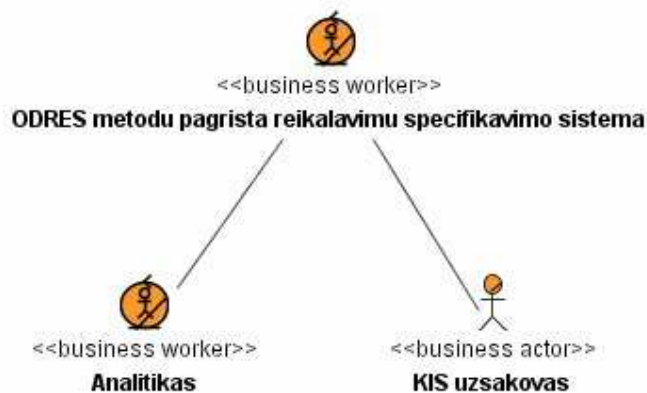


sankirtos atributu), o, tarkim, tik varnele.

Kuriamu analizės įrankiu bus galima analizuoti tas reikalavimų specifikacijas, kurių saugyklos realizuotos tam tikros DBVS duomenų bazėje. Jis nebus taikomas, jeigu saugykla bus paremta, pavyzdžiui, XML pagrindu.

#### 4.1.1 Kontekstinė diagrama

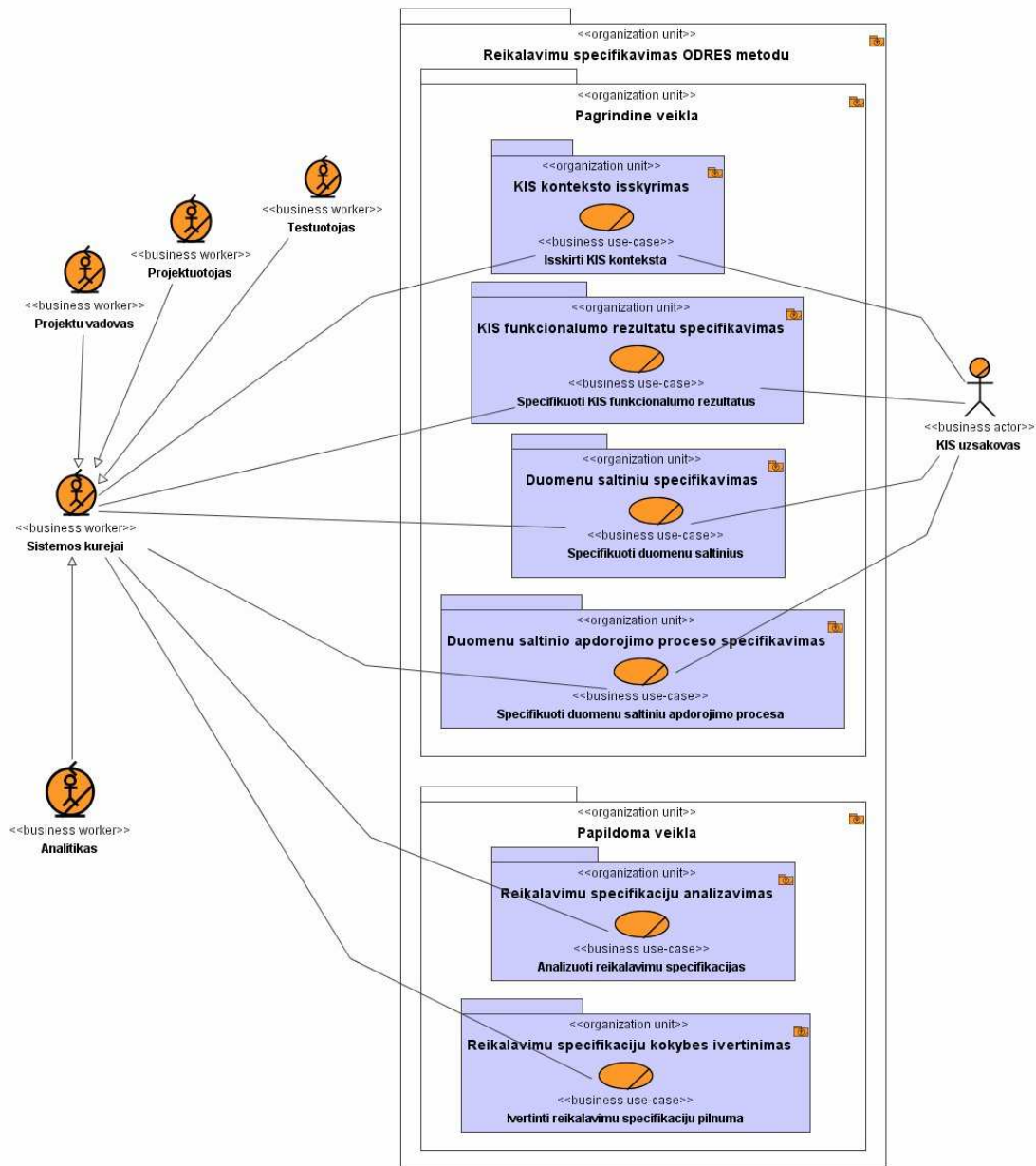
Abstrakčiausiame lygyje sistemos veiklą vaizduoja kontekstinė diagrama (4 pav.). Reikalavimų specifikacijos sudaromos ODRES (Output Driven Requirements Specification) metodu. Į šį procesą įtraukiamas ir kompiuterizuotos informacinės sistemos (KIS) užsakovas, su kurio pagalba tikslinamos sistemos detalės pokalbio metu. Diagramoje jam suteiktas išorinio veiklos darbuotojo stereotipas („business actor“). Specifikacijas, sudarytas ODRES metodu, analizuoja analitikas, kuriam suteiktas vidinio veiklos darbuotojo stereotipas.



4 pav. Veiklos kontekstinė diagrama

Veiklos panaudojimo atvejų modelyje (5 pav.) vaizduojamas aukščiausio lygio organizacinis vienetas „Reikalavimų specifikavimas ODRES metodu“, kuris skaidomas į pagrindinę ir papildomą veiklą. Pagrindinė veikla apima reikalavimų specifikavimą, susidedantį iš 4 smulkesnių etapų – KIS konteksto išskyrimo, funkcionalumo rezultatų specifikavimo, duomenų šaltinių specifikavimo, duomenų šaltinių apdorojimo proceso specifikavimo. Remiantis ODRES metodu, į kiekvieną pagrindinės veiklos stadiją įtraukiamas ir kompiuterizuotos informacinės sistemos užsakovas.

Papildoma veikla apima sudarytų specifikacijų analizę bei kokybės vertinimą. Šio organizacinio vieneto veikloje kompiuterizuotos informacinės sistemos užsakovas nedalyvauja.

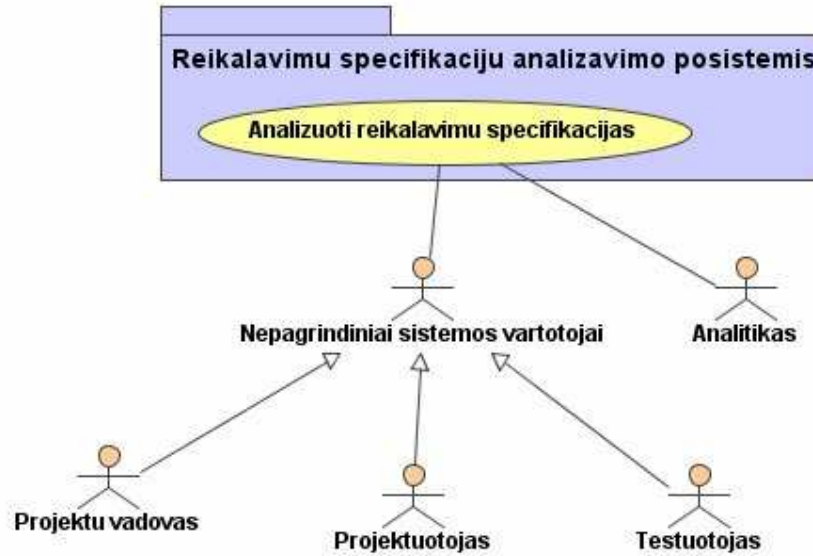


5 pav. Veiklos panaudojimo atvejų diagrama

Veiklos panaudojimo atvejų modelyje vidiniais darbuotojais vaizduojami sistemų kūrėjai: projekto vadovas, projektuotojas, testuotojas bei analitikas, kuris, beje, laikomas pagrindiniu vartotoju. Tačiau jie visi gali naudotis sistema vienodomis teisėmis, kiekvienam atskirai nėra apribojimų.

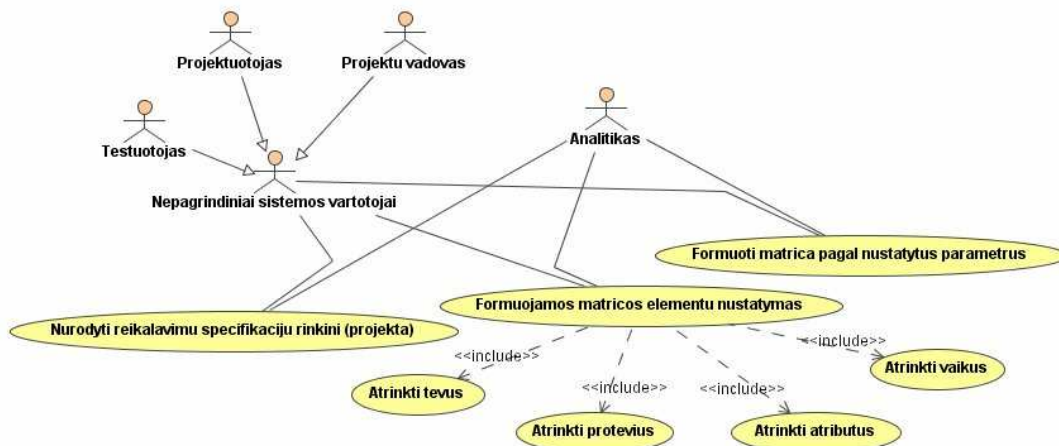
#### 4.1.2 Sistemos panaudojimo atvejų diagrama

Iš veiklos panaudojimo atvejų diagramos (5 pav.) išskirta dalykinė sritis (6 pav.) - „Reikalavimų specifikacijų analizavimo“ posistemis. Būtent, bus kuriama sistema analizuoti reikalavimų specifikacijas, sudarytas ODRES metodu.



6 pav. Išskirtas reikalavimų specifikacijų analizavimo (kompiuterizuojamos srities) posistemis.

Dalykiniai sričiai sukurti pradiniai panaudojimo atvejai (7 pav.), kurie tolesniuose etapuose bus kompiuterizuojami. 4.1.3 skyrelyje kiekvienam panaudojimo atvejui sukurtos tikslios specifikacijų lentelės, sekų bei veiklos diagramos.

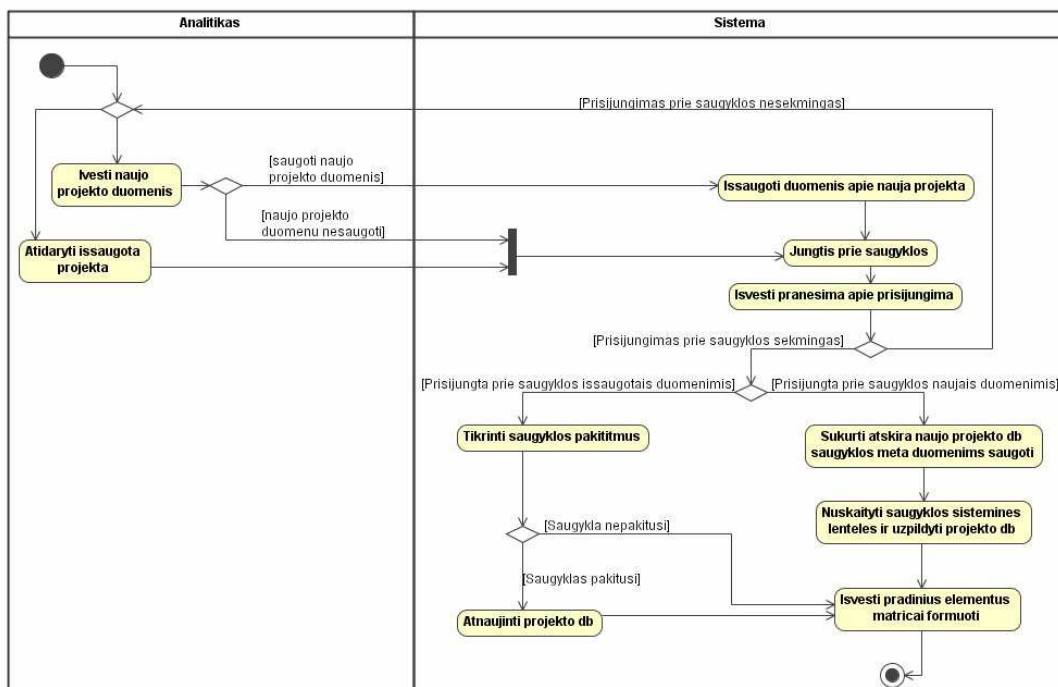


7 pav. Reikalavimų specifikacijų analizavimo posistemisio (kompiuterizuojamos srities) panaudojimo atvejai

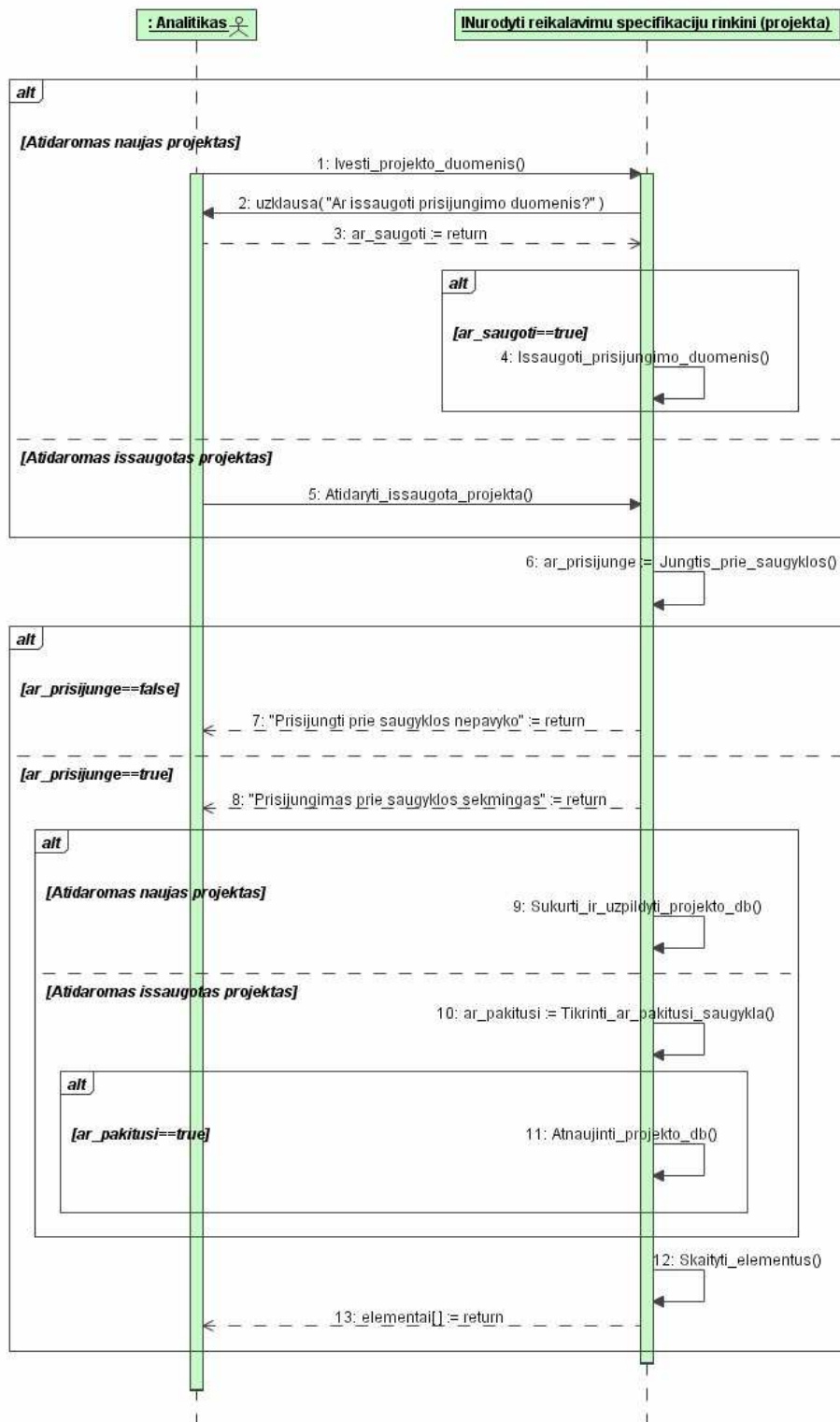
### 4.1.3 Panaudojimo atvejų specifikacijos

Panaudojimo atvejui „Nurodyti reikalavimų specifikacijų rinkinį (projektą)“ nubraižyta veiklos diagrama (8 pav.), sekų diagrama (9 pav.), bei sudaryta detali specifikacija 5. lentelė.

Sistemos darbas prasideda vartotojui nurodžius reikalavimų specifikacijų rinkinį, kuris saugomas saugykloje ir kurį norima analizuoti. Tai padaryti galima dviem būdais – suvesti prisijungimo prie saugyklos duomenis ranka arba pasirinkti jau išsaugotus duomenis. Kiekvienu atveju sistema elgiasi skirtingai. Galiausiai sėkmingai prisijungus prie saugyklos vartotojui išvedami pradiniai duomenys.



8 pav. Veiklos diagrama panaudojimo atvejui „Nurodyti reikalavimų specifikacijų rinkinį (projektą)“



9 pav. Sekų diagrama panaudojimo atvežiui „Nurodyti reikalavimų specifikacijų rinkinį (projektą)“

5. lentelė– Panaudojimo atvejo „Nurodyti reikalavimų specifikacijų rinkinį (projektą)“  
specifikacija

<b>Sužadinantis įvykis</b>	Vartotojas nurodo duomenis sistemai prisijungti prie reikalavimų specifikacijų saugyklos
<b>Prieš sąlyga</b>	Vartotojas turi žinoti prisijungimo duomenis. Arba jie turi būti išsaugoti sistemoje.
<b>Ryšiai su kitais PA</b>	-
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas įveda naujo prisijungimo prie saugyklos duomenis bei nurodo ar juos išsaugoti.	1.1. Jei vartotojas nurodė, kad duomenys būtų išsaugoti, sistema išsaugo juos savo duomenų bazėje. 1.2. Sistema jungiasi prie reikalavimų specifikacijų saugyklos. 1.3. Sistema prisijungia prie saugyklos ir išveda pranešimą apie sėkmingą prisijungimą. 1.4. Sistema sukuria duomenų bazę naujam projektui, ją užpildo iš saugyklos sisteminių lentelių nuskaitytais duomenimis. <b>Po sąlyga:</b> sistemos duomenų bazė papildyta naujais duomenimis.
	2. Sistema vartotojui išveda pradinius duomenis nustatymams atlikti.
<b>Alternatyvūs scenarijai</b>	
1a. Vartotojas nurodo išsaugotą projektą.	1.1a. Sistema vykdo žingsnius Nr. 1.2. ir Nr. 1.3.
	1.3a. Prisijungti prie saugyklos nepavyksta, sistema išveda pranešimą ir baigia darbą.
	1.4a. Sistema tikrina ar pakitusi saugykla nuo paskutinio prisijungimo prie jos. Jeigu saugykla pakitusi, tuomet sistema iš naujo skaito reikalavimų saugyklą ir atnaujina sistemos duomenų bazę.
<b>Po sąlyga</b>	Vartotojas gali nurodyti elementus matricai formuoti.
<b>Veiklos taisyklės</b>	Jungiantis prie saugyklos naujais duomenimis, vartotojas nurodo serverio vardą, duomenų bazės vardą, prisijungimo vardą, slaptažodį. Taip pat nurodo ar saugoti naujus duomenis. Neprivalomas laukas yra „Aprašymas“. Jungiantis išsaugotais duomenimis, vartotojas gali juos redaguoti.
<b>Specialūs nefunkciniai reikalavimai</b>	Reikalavimų specifikacijų saugykla, prie kurios sistema jungiasi (ir į kurią vėliau kreipiasi), turi veikti „MS SQL Server 2000“ arba „MS SQL Server 2005“ pagrindu.

Šiame darbe naudojamos specialios sąvokos – vaikas, tėvas, protėvis. Jos pasirinktos klasifikuoti reikalavimų specifikacijų objektams vienas kito atžvilgiu. Kiekviena iš šių sąvokų reiškia tam tikro pobūdžio elementų aibę ir parodo jų santykį su kito pobūdžio elementų aibe. Konkrečiai kalbant, kuriamoje sistemoje matricos eilutės ir stulpeliai vadinami tėvais, jų

sankirtos – vaikais, o eilutes bei stulpelius apibendrinančios aibės – protėviais. Tiesa, skirtingoms aibėms gali priklausyti tie patys elementai.

Panaudojimo atvejis „Formuojamos matricos elementų nustatymas“ apima keturis kitus: „Atrinkti vaikus“, „Atrinkti tėvus“, „Atrinkti protėvius“, „Atrinkti atributus“. Sekų diagramoje (10 pav.) tam atvaizduoti panaudotos „ref“ nuorodos į atitinkamų panaudojimo atvejų sekų diagramas, kurios šiame skyrelyje specifikuotos toliau. Nuorodų blokuose naudojami perduodami argumentai:

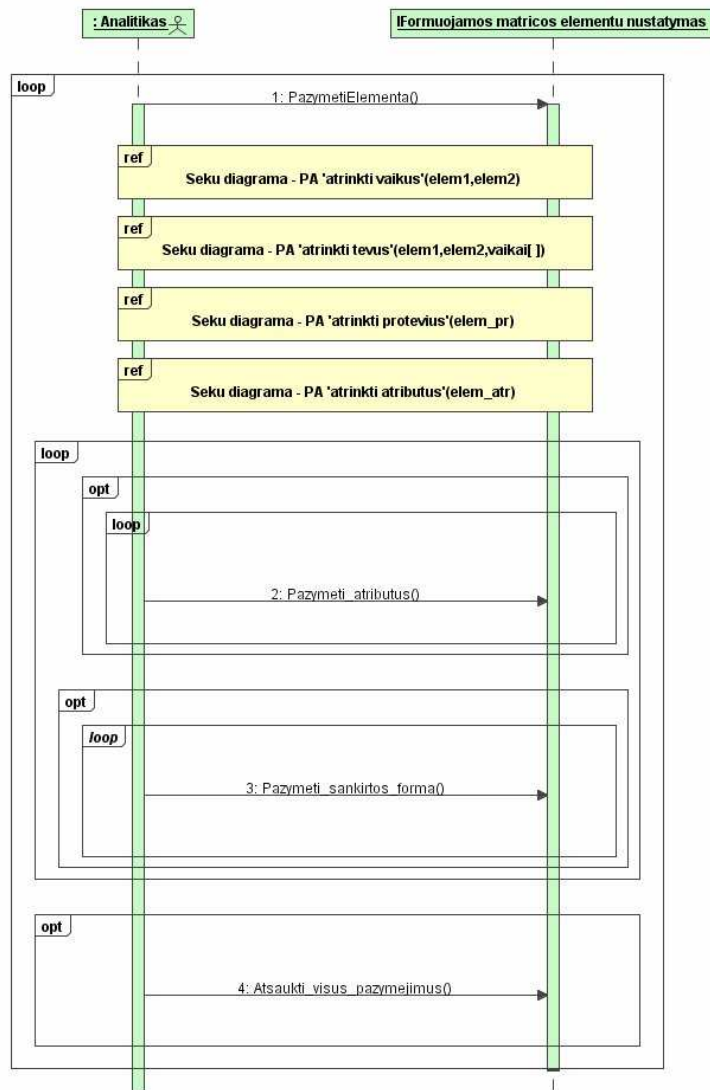
1. Pirmajame bloke nurodytas argumentas „elem1“ žymi vieną iš tėvų, o „elem2“ – žymi kitą tėvą.
2. Antrajame nuorodos bloke: "elem1" ir "elem2" žymi vieną iš tėvų arba vaiką. T.y. "elem1" rodo elementą (vaiką arba vieną iš tėvų), kuris pažymimas. O "elem2" rodo elementą (vaiką arba vieną iš tėvų), kuris jau buvo pažymėtas anksčiau. Masyvas "vaikai[]" skirtas atvejui, kai pažymimas tik tėvas, ir perduodamas, kad atrinkti kiekvieno vaiko antrus tėvus.
3. Trečiajame nuorodos bloke argumentas „elem\_pr“ žymi elementą (vieną iš tėvų), kuriam bus atrenkami protėviai. Ši dalis vykdoma tik tada, kai pasirenkamas tėvas. Tokiu atveju "elem\_pr" įgauna pažymėto tėvo reikšmę.
4. Ketvirtajame nuorodos bloke: argumentas „elem\_atr“ žymi elementą (vaiką, vieną iš tėvų arba vieną iš protėvių), kuriam sistema atrinks atributus.

Pažymėjus elementą „įeinama“ į kiekvieną nuorodos bloką ir atliekami pagal sąlygas leistini veiksmai. Vartotojui pradėjus darbą su sistema, leidžiama pasirinkti tik vaiką arba vieną iš tėvų. Pažymėjus vieną iš šių elementų yra atrenkami su jais susiję kiti elementai, kuriuos vartotojas vėl gali rinktis. 10 paveiksle esančioje diagramoje panaudoti „loop“ ciklai, norint pavaizduoti, kaip vartotojas gali kartoti elementų pažymėjimą; pažymėjus vieną elementą gali iš naujo rinktis kitą; gali atšaukti visus pažymėjimus ir veiksmus kartoti. Nėra griežtos elementų pažymėjimo tvarkos. Vartotojas gali pasirinkti pirmiau tiek vaiką, tiek vieną iš tėvų. Priešingai yra su protėviais, kuriuos pažymėjus atrenkami tik jų atributai.

Pažymėjus elementą atrenkami ir išvedami jo atributai, tačiau nėra būtina juos iškart rinktis. Svarbiausia, jog pažymėjus elementą, būtų pažymėtas nors vienas jo atributas prieš formuojant matricą. Sankirtos formą taip pat galima nurodyti bet kada. Tiesa, nepažymėjus sankirtos formos, pagal nutylėjimą eilučių ir stulpelių sankirtos bus vaizduojamos varnele.

6. lentelėje pateikta panaudojimo atvejo „Formuojamos matricos elementų

nustatymas“ tekstinė specifikacija bei 11 paveiksle detali veiklos diagrama.

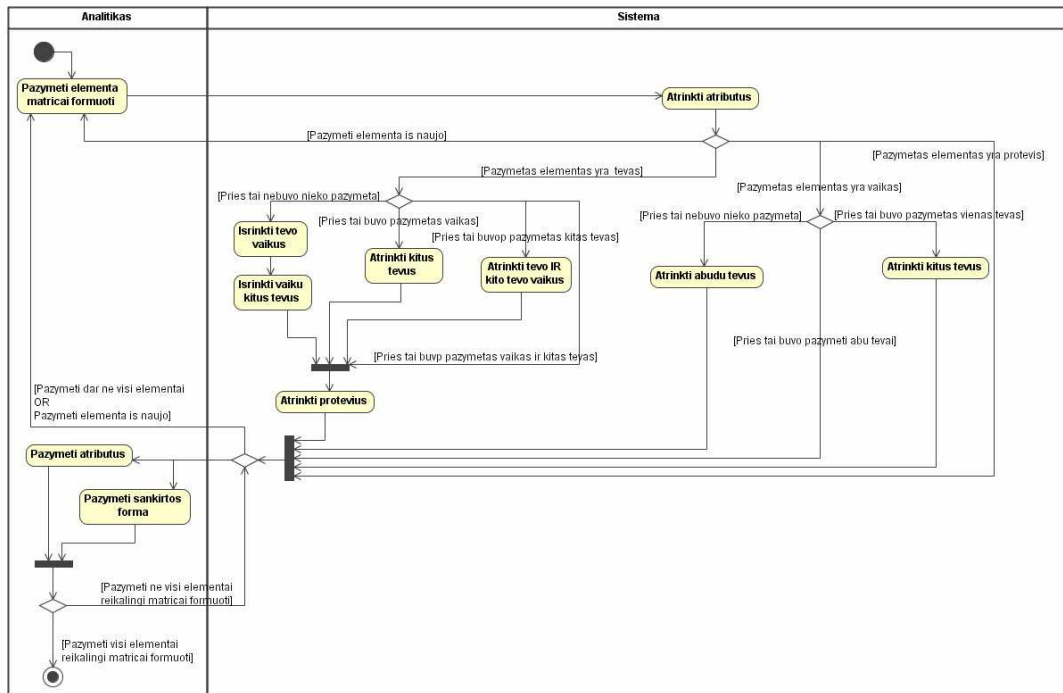


10 pav. Sekų diagrama panaudojimo atvejui „Formuojamos matricos elementų nustatymas“



6. lentelė– Panaudojimo atvejo „Formuojamos matricos elementų nustatymas“  
specifikacija

<b>Sužadinantis įvykis</b>	Vartotojas nurodo elementus matricai formuoti.
<b>Prieš sąlyga</b>	Užpildyta sistemos duomenų bazė; išvesti abu tėvai ir vaikai.
<b>Ryšiai su kitais PA</b>	Apibendrinantys PA: „Atrinkti protėvius“, „Atrinkti tėvus“, „Atrinkti vaikus“, „Atrinkti atributus“
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas pažymi vaiką.	1.1a. Jei nieko nepažymėta, sistema atrenka pažymėto vaiko atributus ir abu tėvus. 1.1b. Jei pažymėtas vienas iš tėvų, sistema atrenka vaiko atributus ir kitus tėvus. 1.1c. Jei pažymėti abudu tėvai, sistema atrenka vaiko atributus.
2. Vartotojas pažymi atributus.	
3. Vartotojas pažymi sankirtos formą.	
<b>Alternatyvūs scenarijai</b>	
1a. Vartotojas pažymi tėvą.	1a.1a. Jei nieko nepažymėta, sistema atrenka pažymėtam tėvui atributus, protėvius, visus vaikus, visiems vaikams antrus tėvus. 1a.1b. Jei pažymėtas vaikas, sistema atrenka atributus, protėvius ir antrus tėvus. 1a.1c. Jei pažymėtas kitas tėvas, sistema atrenka atributus, protėvius ir dviejų tėvų vaikus.
1b. Vartotojas pažymi protėvį.	1b.1. Sistema atrenka pažymėtam protėviui atributus.
2a. Vartotojas pažymi atributus iš naujo, arba pažymi kitus atributus, arba atributų kol kas nežymi.	
3a. Vartotojas pakeičia sankirtos forma, arba kol kas nežymi sankirtos formos.	
<b>Po sąlyga</b>	Nurodyti visi reikalingi elementai matricos formavimui. Sistema pasirodusi nuskaityti elementų reikšmes iš specifikacijų saugyklos.
<b>Veiklos taisyklės</b>	Iš pradžių sistema leidžia pažymėti vieną iš tėvų ar vaiką. Tik parinkus tėvą atitinkamai atrenkami jo protėviai. Pažymėjus protėvį atrenkami tik jo atributai, t.y. tėvai ir vaikai protėvio atžvilgiu neatrinkinėjami. Vartotojas gali keisti pažymėtus elementus kiek tik nori. Norint gauti matrica, mažiausiai turi būti pažymėti abu tėvai, vaikas ir kiekvieno jų atributai.



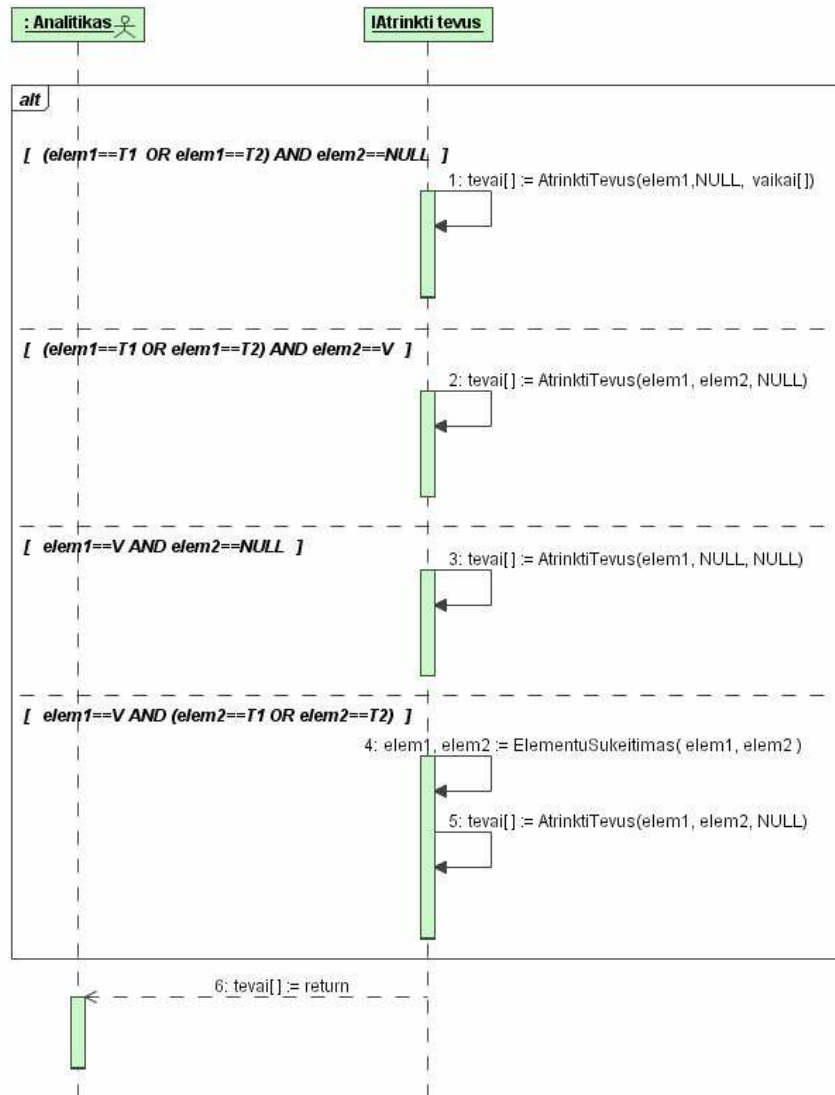
11 pav. Veiklos diagrama panaudojimo atvejui „Formuojamos matricos elementu nustatymas“

12 pav., 13 pav., 14 pav., 15 pav. pateiktos sekų diagramos (kartu ir jų detalios tekstinės specifikacijos) tiems panaudojimo atvejams, kurie išplečia PA „Formuojamos matricos elementu nustatymas“.

Panaudojimo atvejui „Atrinkti tėvus“ nubraižyta sekų diagrama esanti 12 paveiksle. Joje panaudotas „alt“ fragmentas. Kiekvienu atveju sistema atrenka tėvus, tačiau rezultatas priklauso nuo sąlygos, t.y. nuo to, kokius elementus vartotojas pažymi. „alt“ fragmentu pavaizduotos keturios situacijos, kai:

1. vienam iš anksčiau pažymėtų tėvų jau atrinkti jo vaikai, o sistema šiems visiems vaikams atrenka jų galimus kitus tėvus. Argumentas „elem1“ žymi prie tai pažymėtą vieną tėvą, o masyvas „vaikai[]“ saugo tam tėvui atrinktus visus jo vaikus.
2. pažymėtam tėvui ir pažymėtam vaikui sistema atrenka kitus tėvus. Argumentas „elem1“ žymi prie tai pažymėtą vieną tėvą, o „elem2“ – pažymėtą vaiką. Šis punktas apima ir tą atvejį kai pirma pažymimas vaikas, o po to tėvas.
3. pažymėtam vaikui sistema atrenka abudu tėvus. Šiuo atveju atrinktų abiejų tėvų aibės bus vienodos. Čia argumentas „elem1“ žymi pažymėtą vaiką.
4. paskutinė situacija panaši į antrąją. Tik šiuo atveju reikia atlikti papildomą operaciją,

sukeičiančią parametrus „elem1“ su „elem2“ vietomis, o tolesni veiksmai tokie patys.

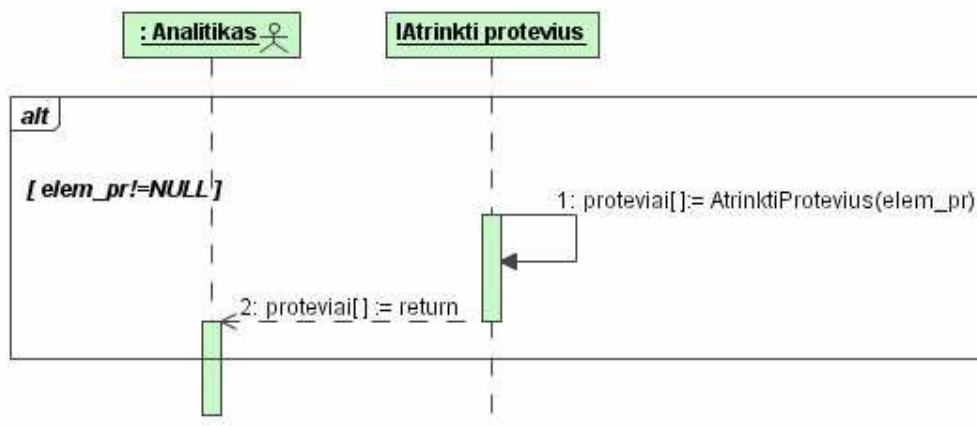


12 pav. - Sekų diagrama panaudojimo atvejui „Atrinkti tėvus“

7. lentelė – Panaudojimo atvejo „Atrinkti tėvus“ specifikacija

<b>Sužadinantis įvykis</b>	Vartotojas pažymi tėvą ar vaiką.
<b>Prieš sąlyga</b>	Išvesti pradiniai elementai.
<b>Ryšiai su kitais PA</b>	Išplečia PA „Formuojamos matricos elementų nustatymas“
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
	1. Sistema pažymėtam tėvui ir jo vaikams atrenka kitus tėvus.
<b>Alternatyvūs scenarijai</b>	
	1a. Pažymėtam vienam iš tėvų ir pažymėtam vaikui atrenka kitus tėvus.
	1b. Pažymėtam vaikui sistema atrenka abudu tėvus.
	1c. Pažymėtam vaikui ir pažymėtam vienam iš tėvų sistema atrenka kitus tėvus.
<b>Po sąlyga</b>	Vartotojo pažymėtam elementui atrinkti ir išvesti susiję tėvai.

Sistema atrenka protėvius tik tuomet, kai pažymimas tėvas. 13 paveiksle panaudotas argumentas „elem\_pr“ žymio pažymėtą tėvą, kuriam sistema turi atrinkti protėvius.

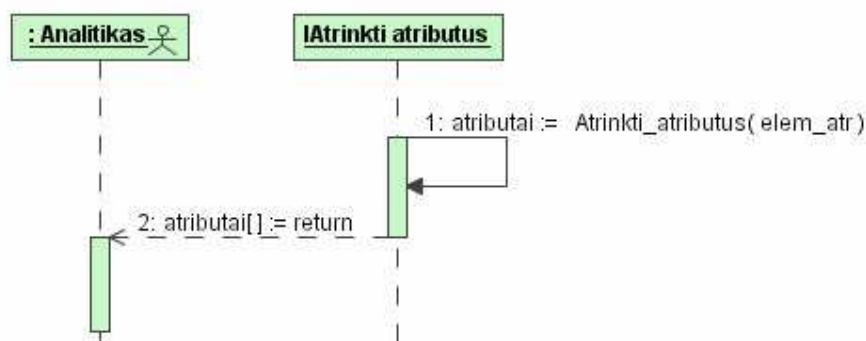


13 pav. Sekų diagrama panaudojimo atvejui „Atrinkti protėvius“

8. lentelė– Panaudojimo atvejo „Atrinkti protėvius“ specifikacija

<b>Sužadinantis įvykis</b>	Vartotojas pažymi tėvą.
<b>Prieš sąlyga</b>	Išvesti pradiniai elementai.
<b>Ryšiai su kitais PA</b>	Išplečia PA „Formuojamos matricos elementų nustatymas“
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
	Sistema pažymėtam tėvui atrenka jo protėvius.
<b>Po sąlyga</b>	Vartotojo pažymėtam tėvui atrinkti ir išvesti protėviai.

Atributus sistema atrenka kiekvienam elementui atskirai. 14 paveiksle esančioje sekų diagramoje argumentas „elem\_art“ žymi elementą, kuriam sistema turi atrinkti atributus.



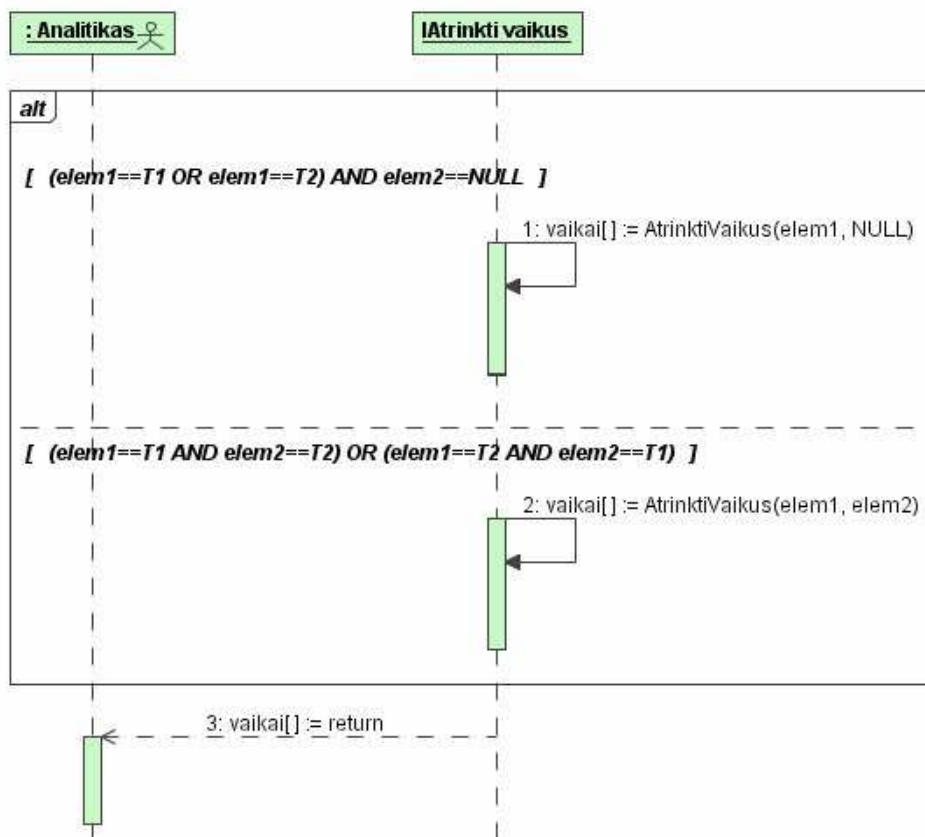
14 pav. Sekų diagrama panaudojimo atvejui „Atrinkti atributus“

#### 9. lentelė– Panaudojimo atvejo „Atrinkti atributus“ specifikacija

<b>Sužadinantis įvykis</b>	Vartotojas pažymi tėvą, vaiką arba protėvį.
<b>Prieš sąlyga</b>	Išvesti pradiniai elementai.
<b>Ryšiai su kitais PA</b>	Išplečia PA „Formuojamos matricos elementų nustatymas“
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
	Sistema pažymėtam elementui (tėvui, vaikui ar protėviui) atrenka jo atributus.
<b>Po sąlyga</b>	Vartotojo pažymėtam elementui atrinkti ir išvesti atributai.

Panaudojimo atvejui „Atrinkti vaikus“ sekų diagramoje (15 pav.) panaudotas „alt“ fragmentas. Sistema elgiasi skirtingai esant dviem situacijoms, kai:

1. atrenka vaikus pažymėtam vienam iš tėvų - argumentas „elem1“ žymi tą vieną tėvą.
2. atrinka vaikus, priklausančius dviem tėvams. Argumentai „elem1“ ir „elem2“ žymi du skirtingus tėvus.



15 pav. Sekų diagrama panaudojimo atvejui „Atrinkti vaikus“

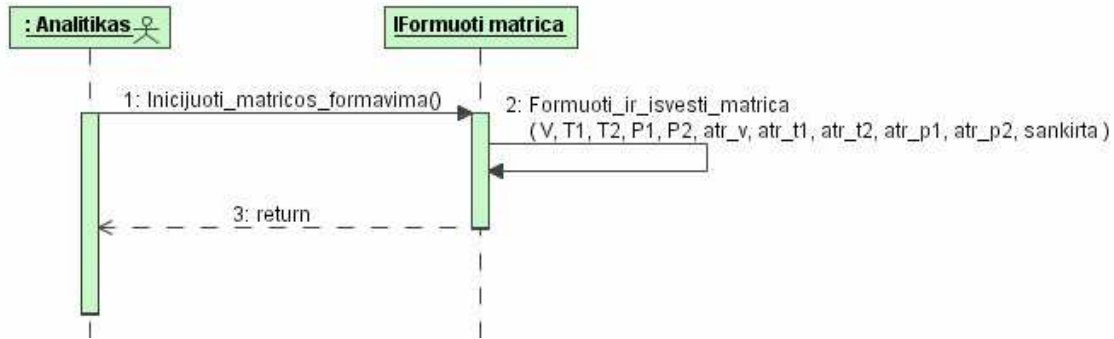
10. lentelė– Panaudojimo atvejo „Atrinkti vaikus“ specifikacija

<b>Sužadinantis įvykis</b>	Vartotojas pažymi tėvą.
<b>Prieš sąlyga</b>	Išvesti pradiniai elementai.
<b>Ryšiai su kitais PA</b>	Išplečia PA „Formuojamos matricos elementų nustatymas“
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
	1a. Sistema pažymėtam tėvui atrinka jo vaikus.
	1b. Sistema pažymėtiems dviem tėvams atrinka bendrus vaikus.
<b>Po sąlyga</b>	Atrinkti ir išvesti vaikai.

Paskutinio panaudojimo atvejo „Formuoti matricą pagal nustatytus parametrus“ (16 pav.) sekų diagramoje pavaizduota operacija „Formuoti\_ir\_įvesti\_matrica“ turi sarašą argumentų, saugančių vartotojo nurodytas reikšmes matricai sudaryti:

- V – matricos eilučių ir stulpelių sankirtų aibė (vaikai);
- T1 – eilučių aibė (vieni tėvai);
- T2 – stulpelių aibė (kiti tėvai);
- P1 – eilutes apibendrinanti aibė (vieni protėviai);

- P2 – stulpelius apibendrinanti aibė (kiti protėviai);
- atr\_v, atr\_t1, atr\_t2, atr\_p1, atr\_p2 – žymi kiekvieno elemento atskirai atributų aibes;
- sankirta – boolean tipo kintamasis, žymintis sankirtą vaizduoti varnele ar sankirtų aibe.



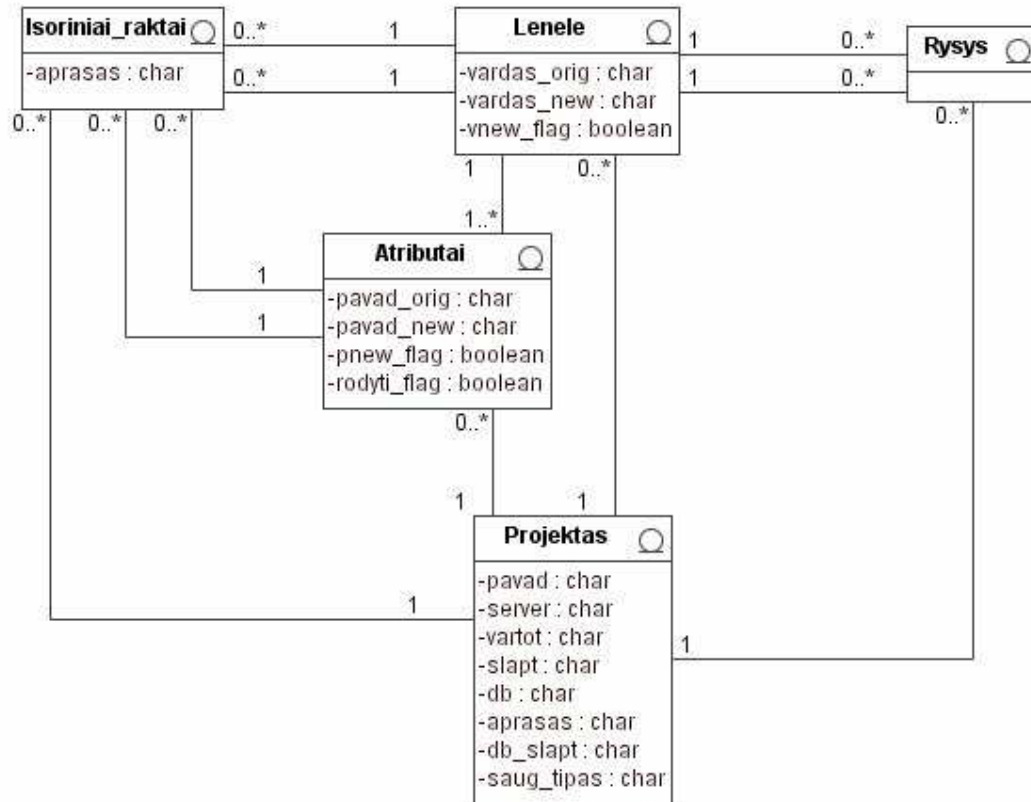
16 pav. Sekų diagrama panaudojimo atvejui „Formuoti matricą pagal nustatytus parametrus“

11. lentelė– Panaudojimo atvejo „Formuoti matricą pagal nustatytus parametrus“  
specifikacija

<b>Sužadinantis įvykis</b>	Vartotojas inicijuoja sistemą matricos suformavimui.
<b>Prieš sąlyga</b>	Nurodyti visi reikalingi elementai matricos formavimui.
<b>Ryšiai su kitais PA</b>	-
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas inicijuoja matricos formavimą.	1.1. Sistema kreipiasi į reikalavimų specifikacijų saugyklą, kad išrinkti duomenis, kurie bus vaizduojami suformuotoje matricoje.
<b>Po sąlyga</b>	Išvesta matrica, suformuota pagal vartotojo nurodytus parametrus.

#### 4.1.4 Dalykinės srities klasių diagrama

Šiame sistemos kūrimo etape klasių modelis apibrėžia dalykinės srities konceptus (veiklos objektus) ir jų tarpusavio ryšius. 17 paveiksle esanti diagrama projektavimo etape bus transformuojama į duomenų bazės schemą.



17 pav. Dalykinės srities esybių modelis

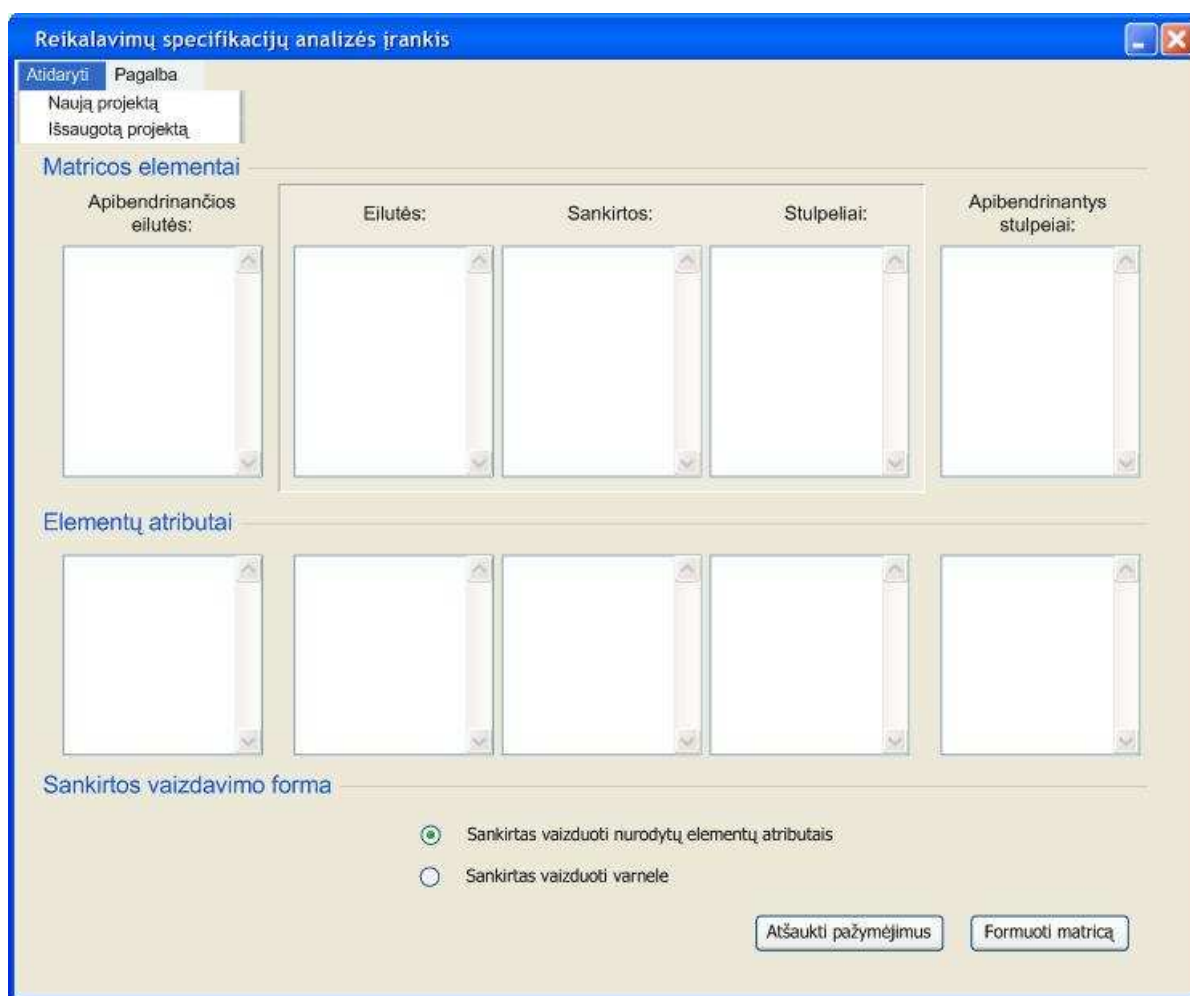
Tam, kad išbandytumėme kuriamą reikalavimų specifikacijų analizės įrankį, naudojamos reikalavimų specifikacijos, sukurtos ODRS metodu. 18 paveiksle pateikta diagrama vaizduoja nuo šio metodo priklausančių konceptų tarpusavio ryšius, kuriais ir paremta specifikacijų analizė.





#### 4.1.5 Vartotojo interfeiso modelis

Vartotojo interfeisą sudaro tik keletas langų. Pagrindinis jų pateiktas 19 paveiksle. Jis suskirstytas į tris sritis – matricos elementai, elementų atributai ir sankirtos vaizdavimo forma. Pirmi penki sąrašo laukai skirti reikalavimų specifikacijų (sudaromos matricos) elementams išvesti. Čia vartotojas juo renkasi. Laukai „Eilutės“, „Sankirtos“ ir „Stulpeliai“ yra pradiniai laukai, į kuriuos sistema pirmiausiai išveda duomenis ir kuriuose pažymėti elementus būtina. Sąrašo laukų pavadinimai parinkti būtent pagal matricos struktūrą, juose išvedamos atrinktos reikšmės. Kiti penki sąrašo laukai skirti konkrečių elementų atributams išvesti bei juos pažymėti. Todėl jie patalpinti lygiagrečiai po kiekvienu matricos elementų lauku.



19 pav. Pagrindinis programos langas

Sankirtos vaizdavimo formos srityje galimas vienas iš dviejų pasirinkimų – eilučių ir stulpelių sankirtas vaizduoti sankirtos elementu, t.y. konkrečia reikšme – pažymėtu atributu, arba paprasčiausiai varnele.

Pagrindiniame meniu galima pasirinkti atidaryti naują projektą arba išsaugotą.

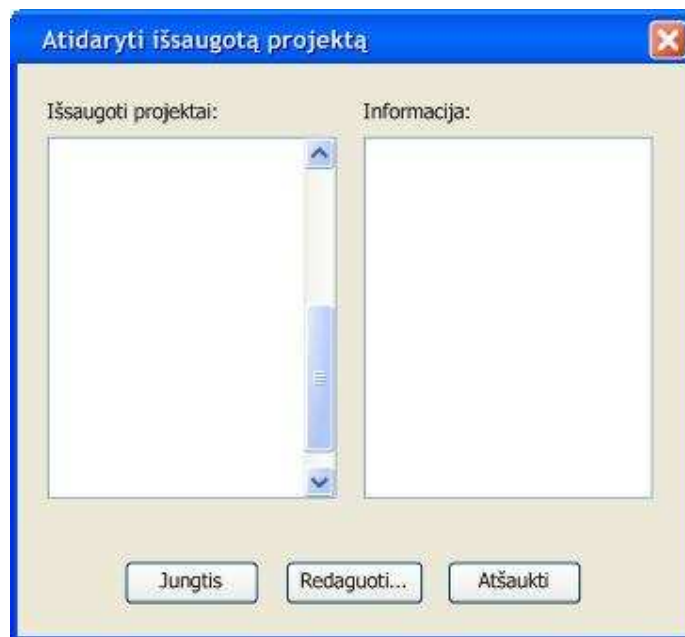
Pasirinkus naują, pateikiamas 20 paveiksle esantis langas su laukeliais, kuriuos reikia užpildyti norint, kad sistema prisijungtų prie konkrečios reikalavimų specifikacijų saugyklos. Pagal nutylėjimą žemiau esanti varnelė būna pažymėta. Ją atšaukus, sistema prie reikalavimų specifikacijų saugyklos jungsis, tačiau prisijungimo duomenų neišsaugos.



The image shows a Windows-style dialog box titled "Atidaryti naują projektą". It has a blue title bar with a close button (X) in the top right corner. The main area is light beige and contains several input fields and a text area. The fields are labeled: "Serverio vardas:", "Duomenų bazės vardas:", "Prisijungimo vardas:", and "Slaptažodis:". Below these is a larger text area labeled "Apibūdinimas: (neprivalomas)". Underneath the text area is a checked checkbox with the text "Išsaugoti prisijungimo duomenis". At the bottom of the dialog are two buttons: "Jungtis" and "Atšaukti".

20 pav. Naujo projekto (reikalavimų specifikacijų rinkinio) atidarymo langas

Norint atidaryti projektą, kurio duomenys sistemoje išsaugoti anksčiau, pasirenkamas išsaugoto projekto atidarymo meniu mygtukas ir pateikiamas 21 paveiksle esantis langas. Jį sudaro du laukai. Pirmajame pateikiamas sąrašas išsaugotų projektų (reikalavimų specifikacijų saugyklų). Kurį nors pažymėjus, greta esančiame lauke išvedama informacija apie jį. Mygtuku „Redaguoti...“ galima taisyti išsaugotą informaciją.



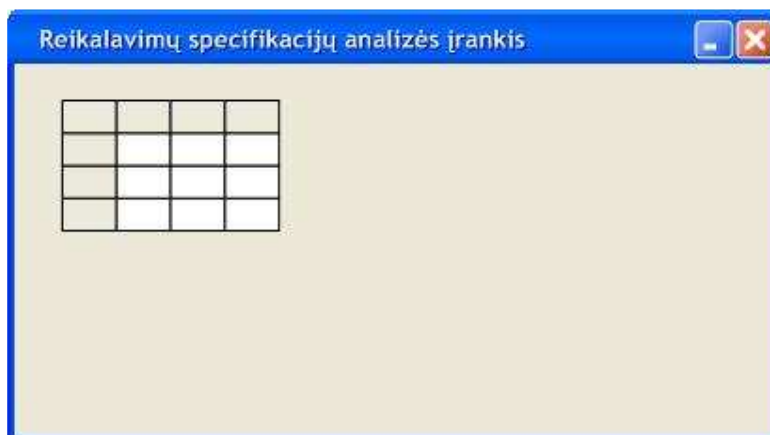
21 pav. Išsaugoto projekto (reikalavimų specifikacijų rinkinio) atidarymo langas

Apie sėkmingą ar nesėkmingą prisijungimą prie saugyklos sistema išveda pranešimus:



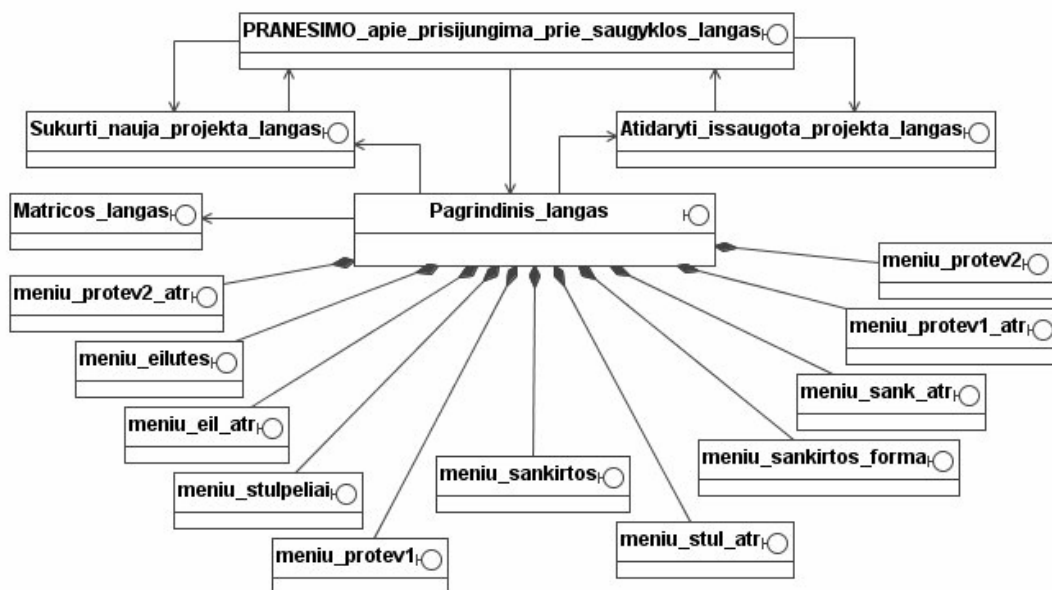
22 pav. Pranešimų langai

Svarbus dar vienas langas, pateiktas 23 paveiksle. Jame išvedama suformuota matrica. Žemiau pateiktas paveikslas nėra labai informatyvus, nes matrica užpildoma ir tinkamai suformuojama tik vartotojui pažymėjus reikalingus elementus.



23 pav. Matricos langas

Pavaizduoti perėjimams iš vieno lango į kitą sukurtas vartotojos interfeiso navigavimo planas (24 pav.). Kompozicijos ryšiais pavaizduota, jog sąrašo elementai yra pagrindinio lango dalys. Rodyklėmis pažymėti galimi perėjimai iš vieno lango į kitą.



24 pav. Sistemos vartotojų interfeiso navigavimo planas

#### 4.1.6 Nefunkciniai sistemos reikalavimai

- Lokaliai kompiuteryje turi būti suinstaliuota „MS SQL Server“ DBVS.
- Prieš analizuojant „MS Access“ tipo saugyklą, būtina nustatyti sisteminių lentelių „MSysObjects“ ir „MSysRelationships“ skaitymo teises.
- Jungiantis prie specifikacijų saugyklos (tiek lokaliai, tiek nutolusiu būdu), kompiuteryje turi būti suinstaliuoti atitinkamos tvarkyklės. Vartotojui leidžiama pakeisti prisijungimo būdą.
- Norint eksportuoti nubraižytą analizės matricą į .xls failą, reikalinga turėti suinstaliuotą „MS Excel“ programą.
- Vartotojui pateikiama suformuota matrica turėtų būti kelių formatų (pavyzdžiai pavaizduoti 25 pav., 26 pav. ir 27 pav.), priklausomai ar vartotojas pažymėjo apibendrinančią (-čias) aibę:

	Stulpelis1	Stulpelis2	Stulpelis3
Eilutė1			
Eilutė2			
Eilutė3			

25 pav. Matricos formatas, kai nepažymėta nei viena apibendrinanti aibė

		Stulpelis1	Stulpelis2	Stulpelis3
Apib.Eil.1	Eilutė1			
	Eilutė2			
Apib.Eil.2	Eilutė3			
	Eilutė4			

26 pav. Matricos formatas, kai pažymėta tik eilutės apibendrinanti aibė (analogiškai ir pažymėjus tik stulpelio aibę)

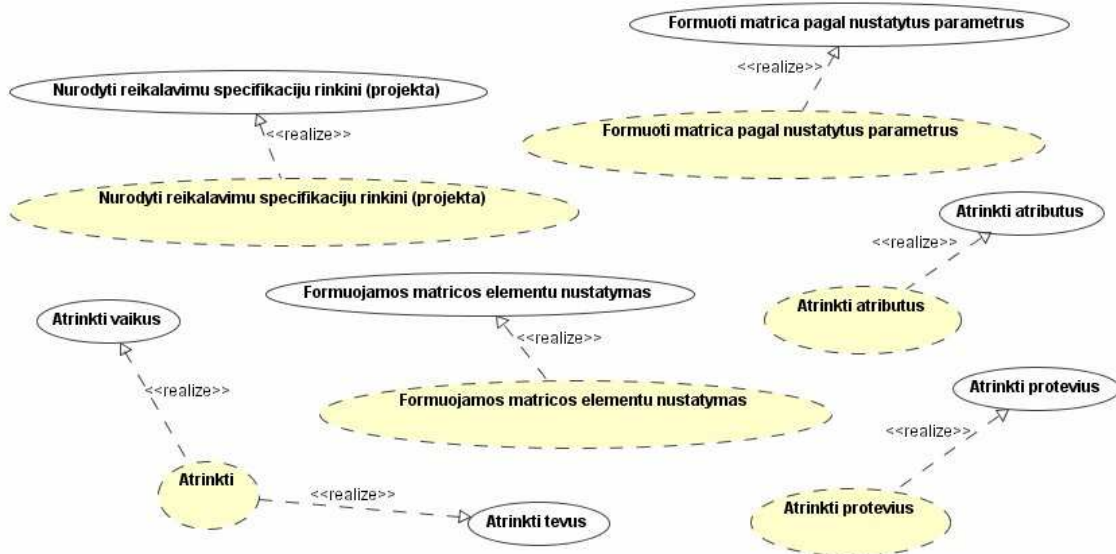
		Apib.Stul1		Apib.Stul2	
		Stulpelis1	Stulpelis2	Stulpelis3	Stulpelis4
Apib.Eil.1	Eilutė1				
	Eilutė2				
Apib.Eil.2	Eilutė3				
	Eilutė4				

27 pav. Matricos formatas, kai pažymėtos eilutės ir stulpelio apibendrinančios aibės

## 4.2 Sistemos projektas

### 4.2.1 Panaudojimo atvejų realizacijos modelis

Kiekvienam panaudojimo atvejui (7 pav.) sukurtos jų realizacijos. 28 pav. matyti, jog panaudojimo atvejus „Atrinkti vaikus“ ir „Atrinkti tėvus“ realizuoja viena realizacija „Atrinkti“.

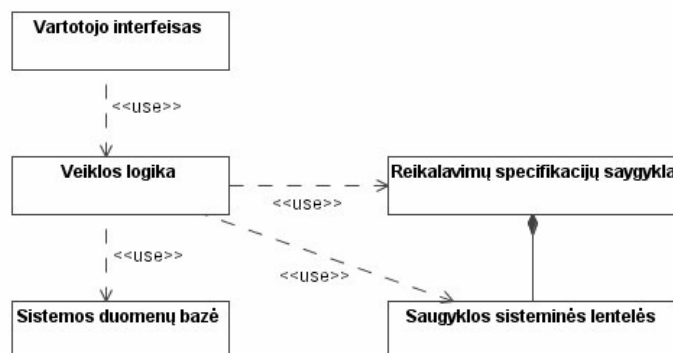


28 pav. Bendras panaudojimo atvejų realizacijos modelis.

Priedo 8.1 skyrelyje pateikta atskirai kokias klases realizuoja kiekviena panaudojimo atvejo realizacija.

### 4.2.2 Sistemos architektūra – statinės struktūros modelis

Kuriama sistema nėra didelė, todėl jos loginės architektūros modelis remiasi klasikine trijų lygių architektūra (29 pav.). Čia duomenų paslaugos išskaidytos į kelis atskirus posistemius.



29 pav. Sistemos loginės architektūra.

Analizės modelio ribinių klasių, valdiklių ir esybės diagramos pateiktos priede 8.3 skyriuje. Analizės etapo klasės transformuotos į projekto etapo klases, joms sukurtos operacijos ir atributai. Kiekvienam loginės architektūros lygiui atskirai priskirtos jiems priklausančios klasės ir jų tarpusavio ryšiai. Priede 8.2 skyrelyje pateiktos visos klasių diagramos kiekvienam statinės struktūros posistemiiui:

- Vartotojo interfeiso klasės - 72 pav.
- Veiklos paslaugų klasės - 73 pav.
- Duomenų paslaugų klasės - 74 pav.

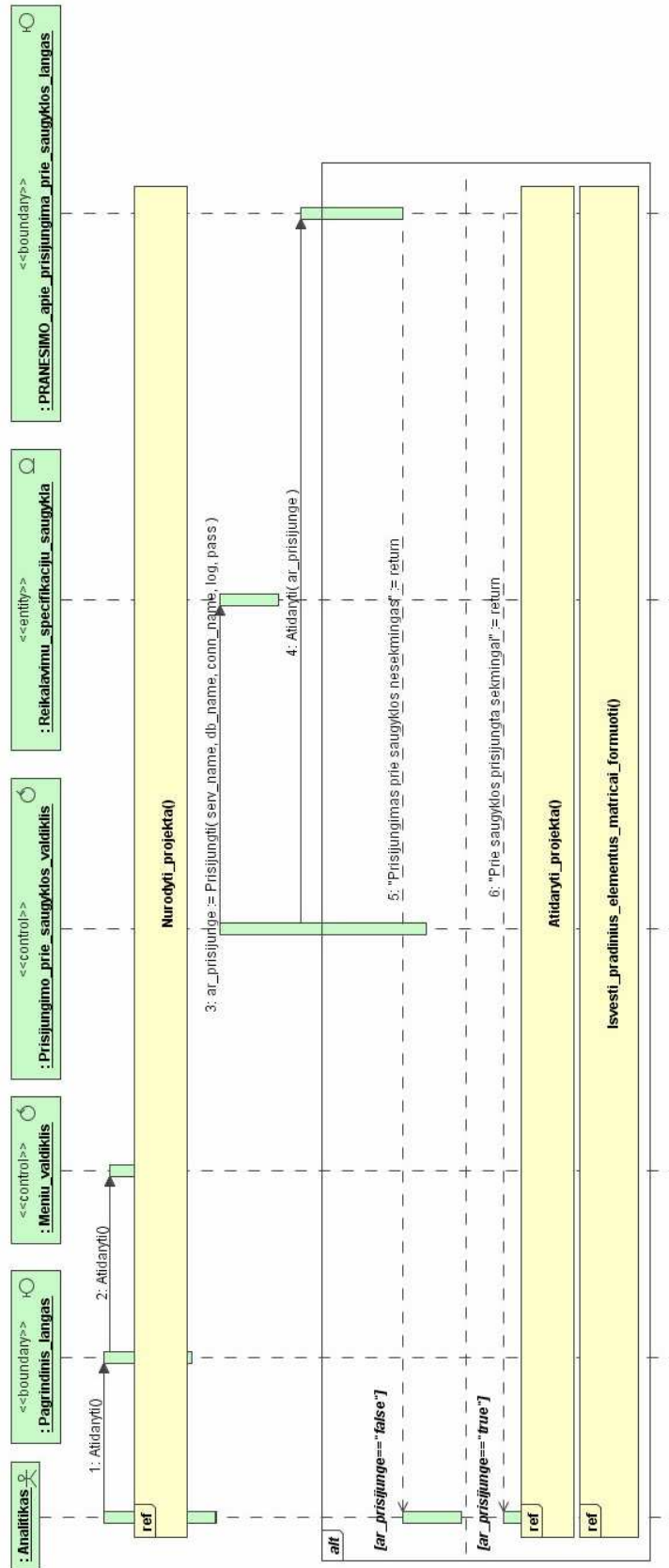
#### **4.2.3 Sistemos elgsenos modelis**

Kiekvienai panaudojimo atvejų realizacijai sukurtos sekų diagramos:

- Nurodyti reikalavimų specifikacijų rinkinį - 30 pav.
- Formuojamos matricos elementų nustatymas – 31 pav.
- Atrinkti - 32 pav.
- Atrinkti atributus - 33 pav.
- Atrinkti protėvius - 34 pav.
- Formuoti matricą pagal nustatytus parametrus - 35 pav.

Jose atspindimos pagrindinės sąveikos tarp sistemos architektūrinių elementų.



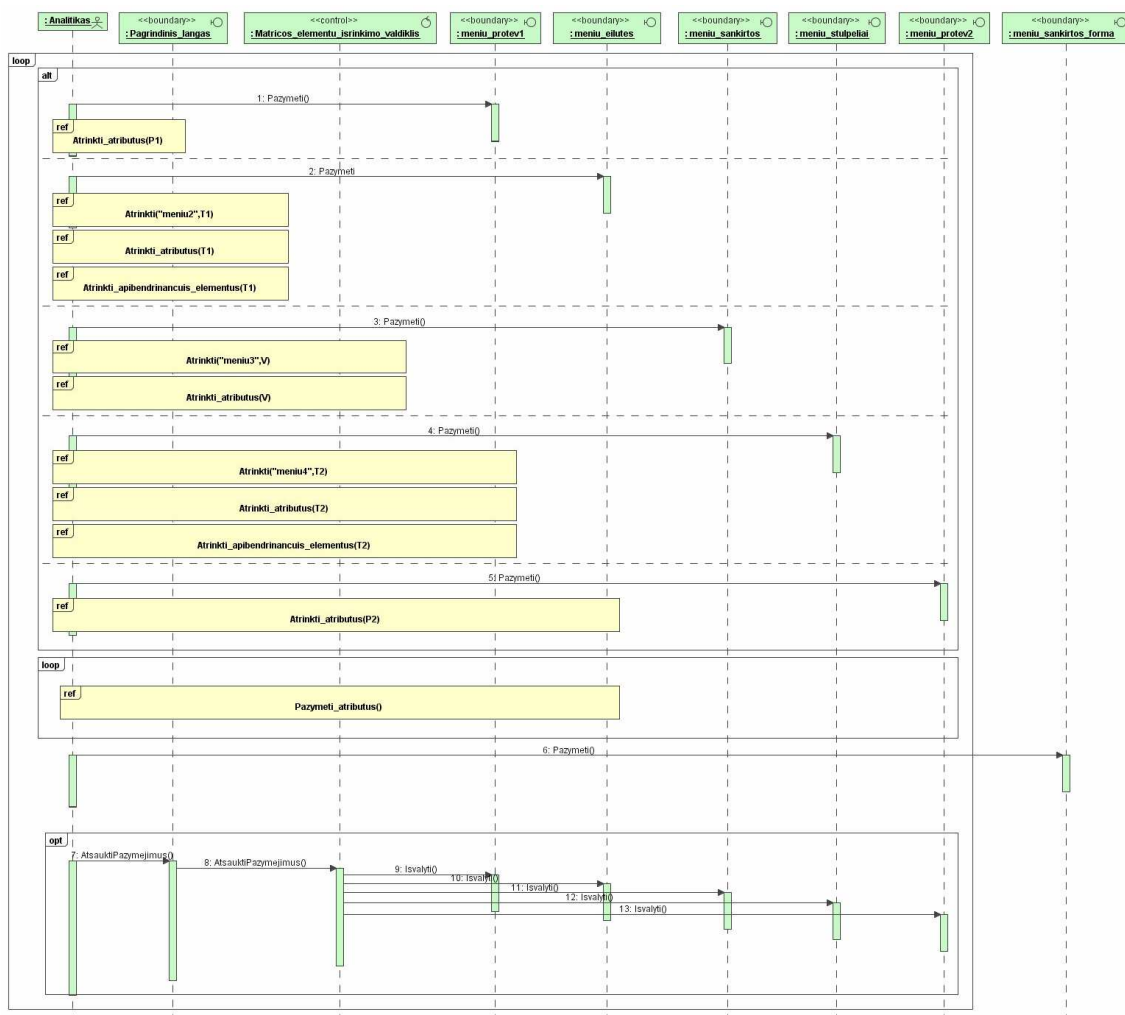


30 pav. Sekų diagrama panaudojimo atvejo „Nurodyti reikalavimų specifikacijų rinkinį (projektą)“ realizacijai

30 paveiksle esančioje panaudojimo atvejo „Nurodyti reikalavimų specifikacijų rinkinį“ sekų diagramoje panaudoti trys „ref“ blokų nuorodos, nukreipiančios į kitas sekų diagramas, kurios pateiktos priede:

- „Nurodyti projektą“ – 80 pav.
- „Atidaryti projektą“ – 81 pav.
- „Išvesti pradinį elementus matricai formuoti“ - 82 pav.

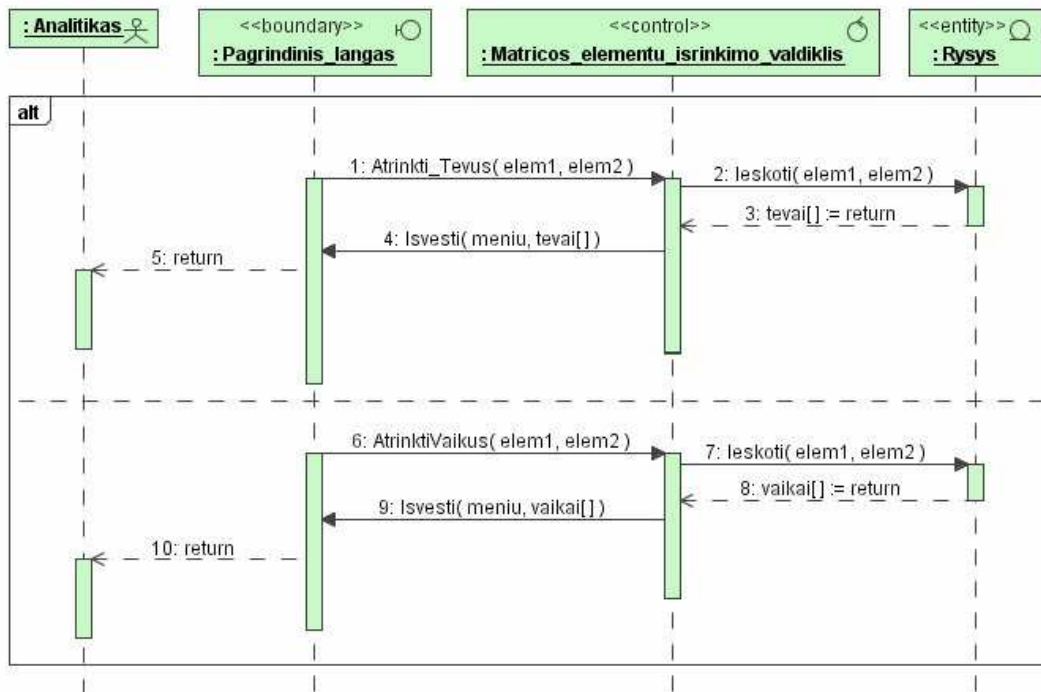
31 paveiksle esančioje panaudojimo atvejo „Formuojamos matricos elementų nustatymas“ sekų diagramoje taip pat naudojami „ref“ nuorodų blokai į kitas sekų diagramas, pateiktas 32 pav., 33 pav. ir 34 pav. Dar viena sekų diagrama į kurią nukreipia „ref“ nuorodą, pateikta priede 83 paveiksle.



31 pav. Sekų diagrama panaudojimo atvejo „Formuojamos matricos elementų nustatymas“ realizacijai

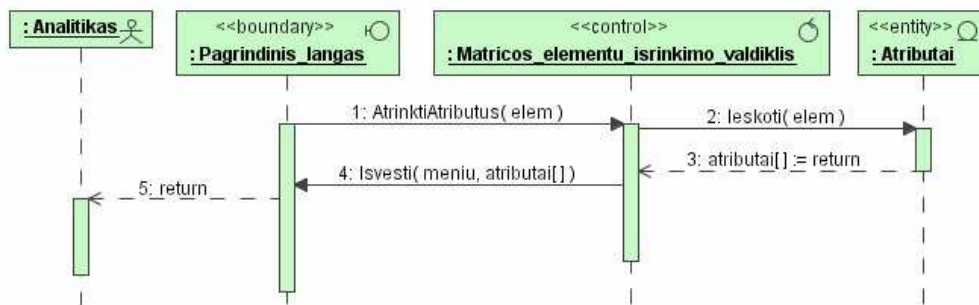
32 paveiksle esančioje sekų diagramoje pavaizduotas pranešimas „Atrinkti\_Tevus“, kuriame naudojami du parametrai – „elem1“ ir „elem2“. Jie nurodo kokiems elementams atrenkami tėvai. Čia vienas iš parametru gali turėti NULL reikšmę.

Pranešime „Išvesti“ nurodoma į kurį sąrašo lauką išvesti rezultatus bei nurodomi patys rezultatai, kuriuos reikia išvesti.

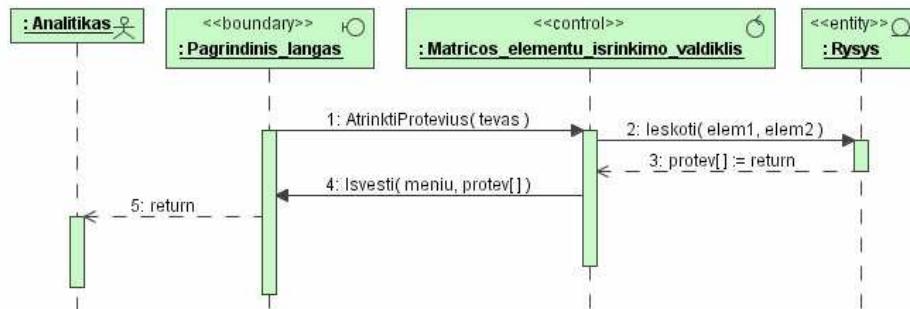


32 pav. Sekų diagrama (REF nuoroda) panaudojimo atvejo „Atrinkti“ realizacijai

33 paveiksle esančioje sekų diagramoje tarp klasių naudojamas pranešimas „AtrinktiAtributus“. Atributų atrinkimas atliekamas bet kuriems vartotojo nurodytiems matricos elementams, nurodomiems kaip parametras šioje operacijoje.

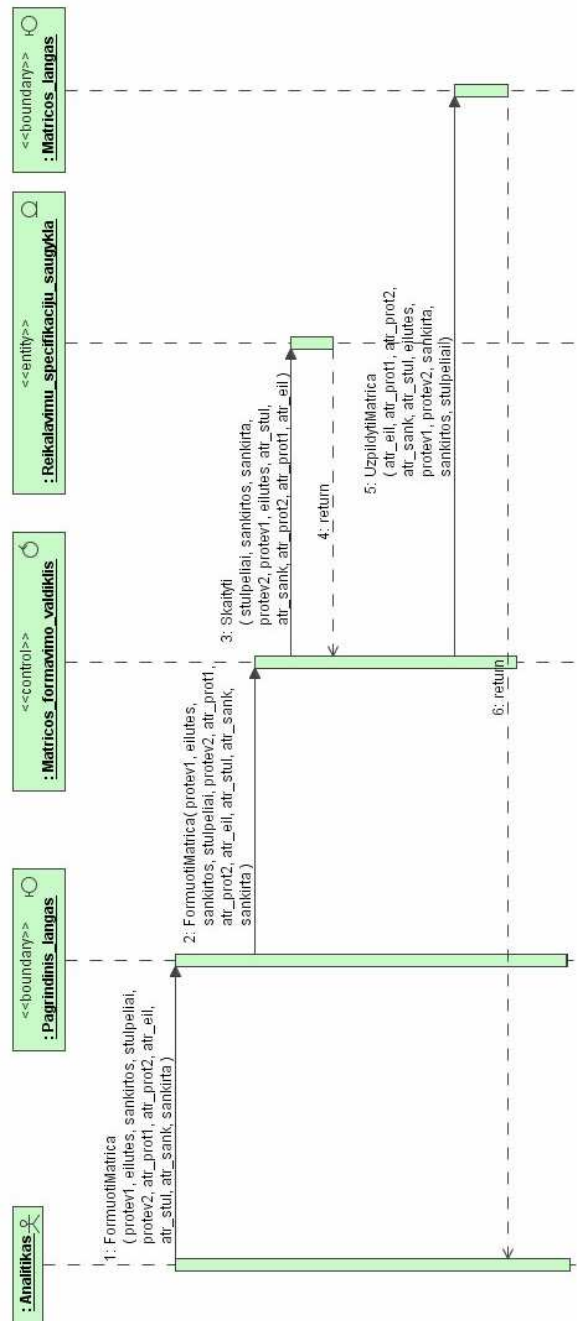


33 pav. Sekų diagrama (REF nuoroda) panaudojimo atvejo „Atrinkti atributus“ realizacijai



34 pav. Sekų diagrama (REF nuoroda) panaudojimo atvejo „Atrinkti protevius“ realizacijai

35 paveiksle esanti sekų diagrama specifikuoja panaudojimo atvejį „Formuoti matricą pagal nustatytus parametrus“. Pranešime „FormuotiMatrica“ perduodami argumentai žymi vartotojo nustatytus elementus, pagal kuriuos bus nuskaitoma reikalavimų specifikacijų saugykla ir užpildyta analizės matrica.



35 pav. Sekų diagrama panaudojimo atvejo „Formuoti matricą pagal nustatytus parametrus“ realizacijai

#### 4.2.4 Duomenų bazės modelis

Duomenų bazės schema gauta transformavus esybes iš dalykinės srities (reikalavimų) modelio. Sistemos funkcionalumui palaikyti, reikalinga duomenų bazė, kurios schema pateikta 36 paveiksle.

Lentelėje „Projektas“, kuri talpinama į duomenų bazę „REPOSITORIES“, saugomi

duomenys, skirti prisijungti prie reikalavimų specifikacijų saugyklų. Jos stulpeliai:

- „pavad,, – projekto pavadinimas, sukurtas vartotojo;
- „server“ – serverio, kuriame saugoma reikalavimų specifikacijų saugykla, vardas arba IP adresas;
- „vartot“ – vartotojo vardas jungiantis prie serverio, kuriame saugoma RS saugykla;
- „slapt“ – prisijungimo slaptažodis;
- „db“ – duomenų bazės (RS saugyklos) vardas. Access atveju tai būtų MDB failo direktorija;
- „aprasas“ – vartotojo sukurtas aprašas reikalavimų specifikacijų projektui;
- „db\_slapt“ - .MDB failo slaptažodis, kurio gali būti reikalaujama „MS Access“ saugyklos atveju.
- „saug\_tipas“ – saugyklos tipas.

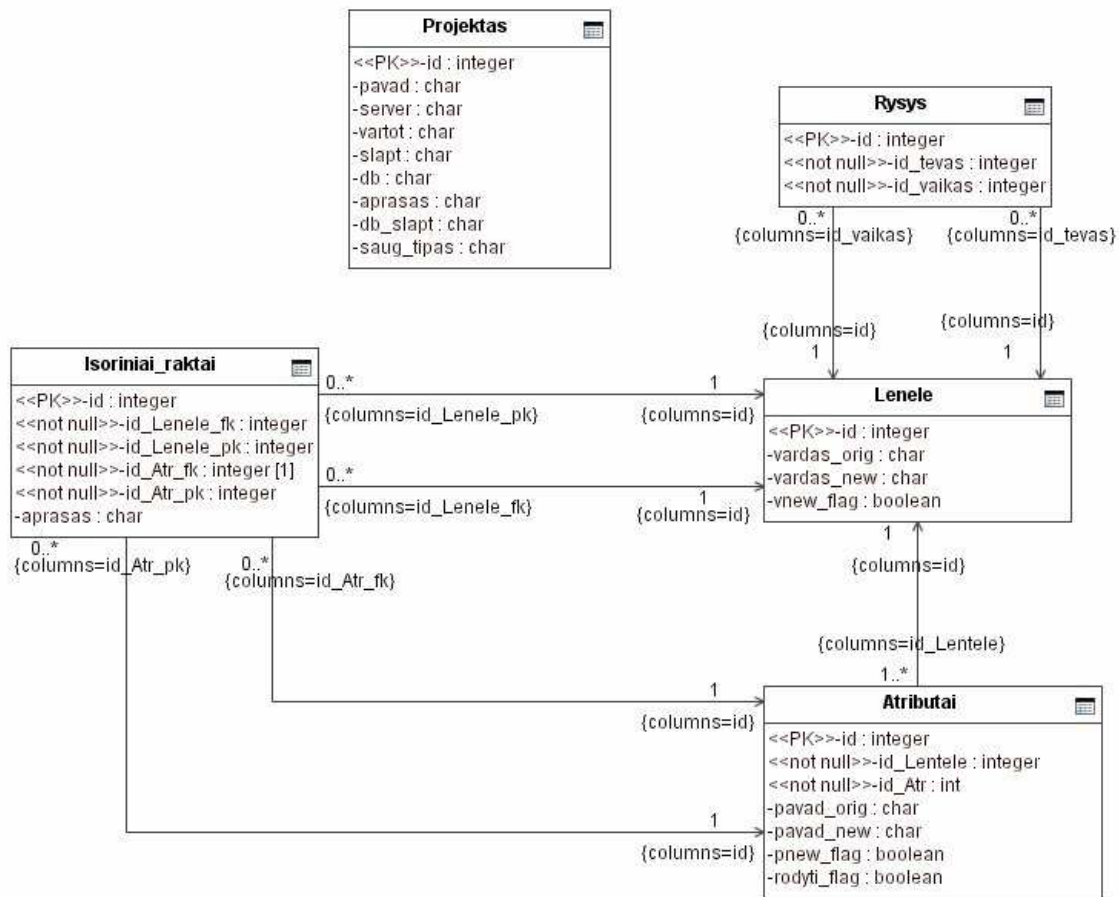
Tam, kad prisijungtume prie saugyklos, nebūtina suvesti visų duomenų, nes kai kurie iš jų nėra privalomi ar nėra reikalingi. Taip pat jie gali priklausyti nuo saugyklos tipo. Tai iliustruoja 12. lentelė.

12. lentelė - Prisijungimo duomenų būtinumas.

Stulpelis	MS SQL Server	MS Access
id	Privalomas	Privalomas
pavad	Privalomas	Privalomas
server	Privalomas	Nėra
vartot	Privalomas	Neprivalomas
slapt	Privalomas	Neprivalomas
db	Privalomas	Privalomas
aprasas	Neprivalomas	Neprivalomas
db_slapt	Nėra	Neprivalomas
saug_tipas	Privalomas	Privalomas

Kiekvieną kartą išsaugant naujus prisijungimo duomenis, sukuriama atskira projekto duomenų bazė su lentelėmis „Lentele“, „Atributai“, „Rysys“ ir „Isoriniai\_raktai“, kuriose saugomi konkrečios saugyklos meta duomenys:

- „Lentele“ – saugyklos lentelės.
- „Atributai“ – lentelių atributai.
- „Rysys“ ir „Isoriniai\_raktai“ – ryšių tarp lentelių duomenys.



36 pav. Sistemos duomenų bazės schema

Naujo projekto duomenų bazei suteikiamas pavadinimas, kuris prasideda „REPOSIT\_“ ir papildomai pridodamas id numeris. Galimas atvejis, kuomet sukuriama laikina projekto duomenų bazė (kai vartotojas atsisako išsaugoti naujo projekto duomenis). Tuomet naujai duomenų bazei suteikiamas pavadinimas „REPOSIT\_TEMP“.

Kiekvienos lentelės, esančios projekto duomenų bazėje, atributų aprašai:

1. Lentelė „Rysys“:
  - „id\_tevas“ – RS saugykloje esančios lentelės, kuriai egzistuoja kita lentelė, turinti FK įrašą į šią lentelę, id.
  - „id\_vaikas“ - RS saugykloje esančios lentelės, kuri turi FK įrašą į kitą lentelę, id.
2. Lentelė „Lentele“:
  - „vardas\_orig“ – RS saugyklos lentelės originalus vardas.
  - „vardas\_new“ – vartotojo sukurtas naujas vardas lentelei.
  - „vnew\_flag“ – reikšmė „true“ žymi, jog RS saugyklos objektą vaizduoti nauju vardu. Priešingu atveju rodomas originalus vardas.
3. Lentelė „Atributai“:

- „id\_Lentele“ – FK laukas, kuriame saugomas id lentelės, kuriai priklauso šis atributas.
- „id\_Atr“ – atributo eilės numeris konkrečioje lentelėje.
- „pavad\_orig“ – originalus atributo pavadinimas.
- „pavad\_new“ – vartotojo sukurtas naujas pavadinimas.
- „pnew\_flag“ – „true“ reikšmė žymi, jog atributas bus vaizduojamas nauju pavadinimu.
- „rodyti\_flag“ – „false“ reikšmė žymi, jog atributas nebus vaizduojamas apskritai.

#### 4. Lentelė „Isoriniai\_raktai“:

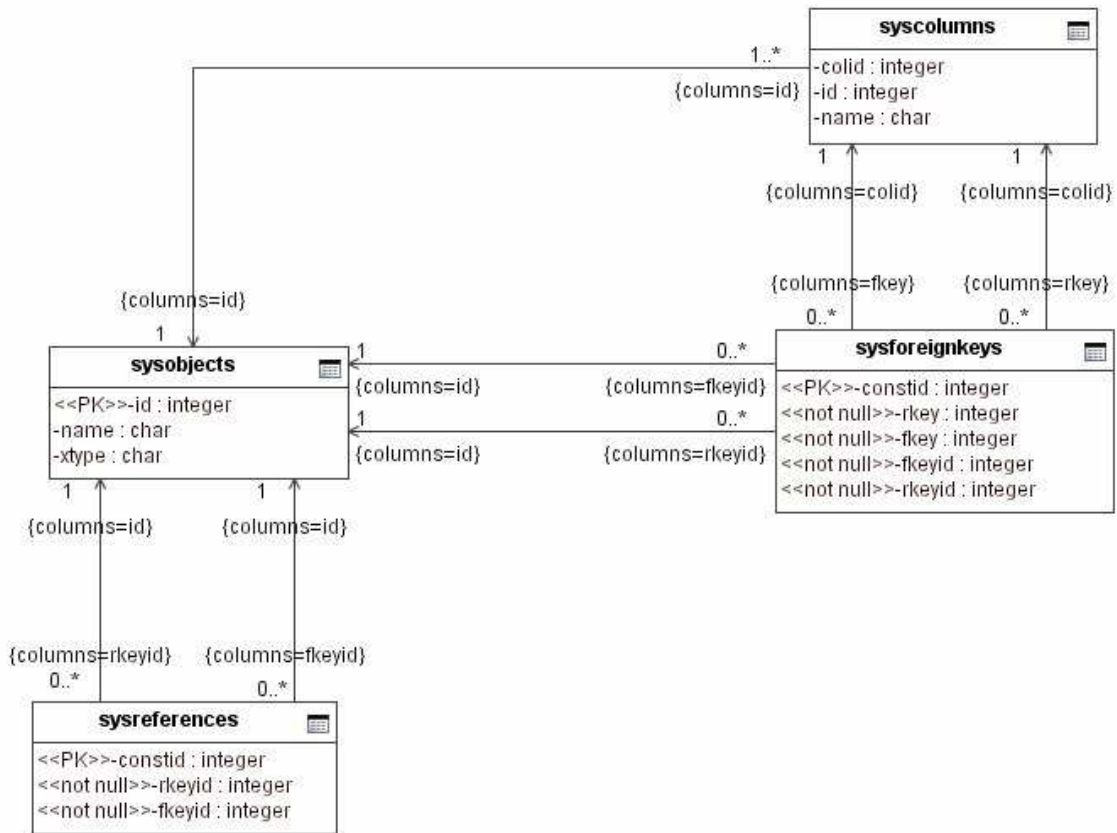
- „aprasas“ – vartotojo sukurtas du objektus siejančio ryšio aprašas.
- „id\_Lentele\_fk“ – lentelė, turinti FK raktą, kuris sieja ją su kita lentele, turiančia PK raktą.
- „id\_Lentele\_pk“ – lentelė, turinti PK raktą ir jai egzistuoja kita susieta lentelė, turinti FK raktą.
- „id\_Atr\_fk“ – saugoma atitinkamos lentelės FK reikšmė.
- „id\_Atr\_pk“ – saugoma atitinkamos lentelės PK reikšmė.

37 pav. ir 38 pav. nubraižytos schemos vaizduoja DBVS sisteminės lentelės, iš kurių išrenkami saugyklos metaduomenys. Jomis tik pasinaudojau ir pats nekūriau. Užduotis prieš programuojant įrankį – išsiaiškinti kokiose sisteminėse lentelėse kokie metaduomenys saugomi ir pasirinkti reikalingus sistemai. Pateikti ne išsamūs duomenų bazių vaizdai, o tik tos lentelės ir atitinkami jų atributai, kuriuos naudoja kuriamas įrankis. Pirmoji schema skirta „MS SQL Server“ saugyklos tipui:

1. „sysobjects“ – saugo objektus, sukurtus duomenų bazėje.
  - Atributas „xtype“ nusako koks saugomo objekto tipas. Kuriamą sistemą atsirenka tik tuos objektus, kurių atributo „xtype“ reikšmė yra „U“ (angl. „user table“).
  - Stulpelyje „name“ saugomas objekto vardas.
2. „syscolumns“ – saugo duomenų bazės lentelių stulpelius.
  - „colid“ skirtas saugoti stulpelio eilės numeriui atitinkamoje lentelėje.
  - „name“ – stulpelio pavadinimas.
3. „sysreferences“ – saugoma informacija apie ryšius tarp duomenų bazėje esančių lentelių.
4. „sysforeignkeys“ – saugomi susietų lentelių bei jų atributų (PK ir FK laukų)

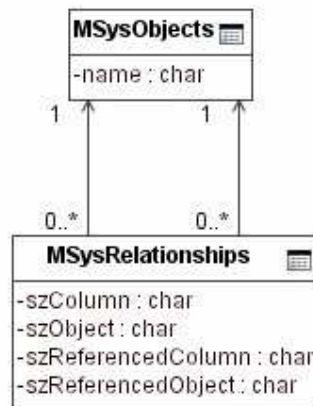


informacija.



37 pav. Sisteminių lentelių schema (MS SQL Server saugykla)

„MS Access“ saugyklos atveju, įrankis naudoja šias sisteminės duomenų bazės lenteles:



38 pav. Sisteminių lentelių schema (MS Access saugykla)

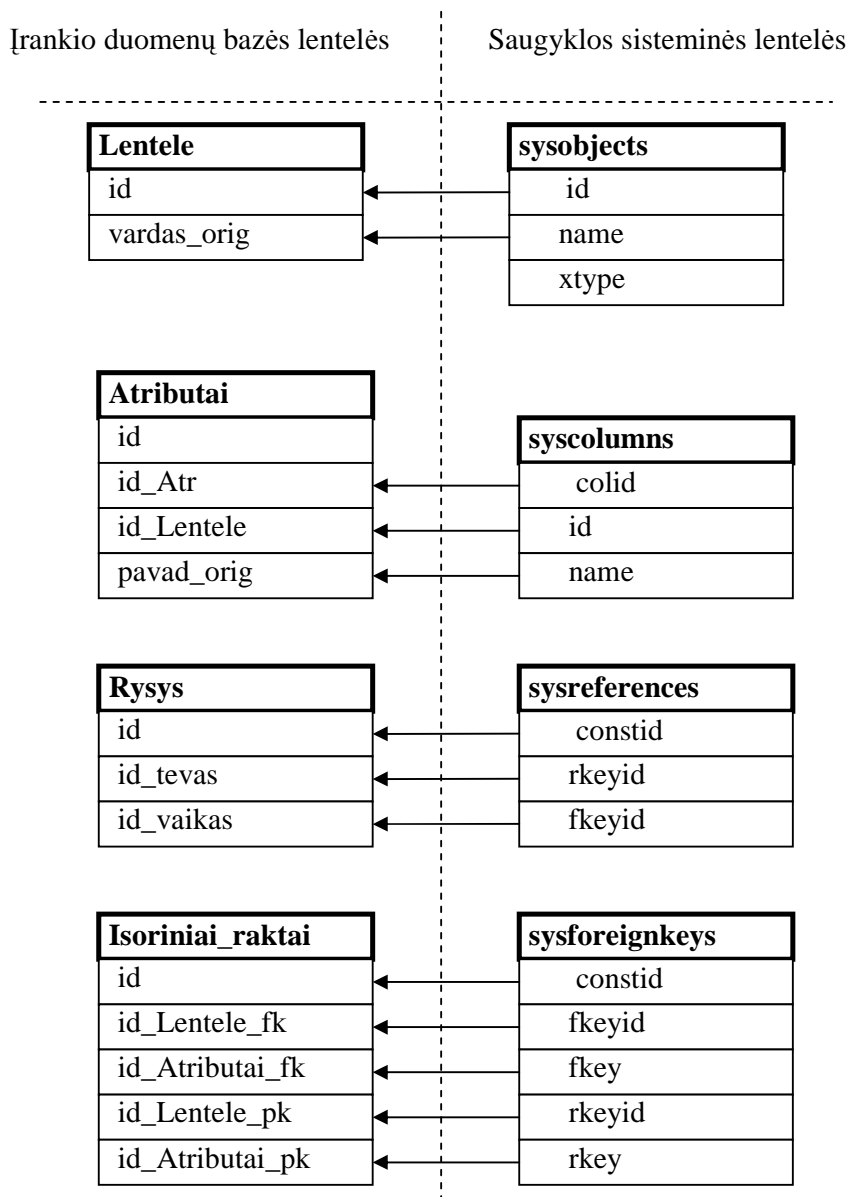
Čia:

1. „MSysObjects“ – saugomi duomenų bazėje sukurti objektai.
  - „name“ – objekto vardas.

2. „MSysRelationships“ – saugomi duomenys apie ryšius tarp duomenų bazės lentelių.
  - „szObject“ – saugomas lentelės, kuri susiejama išoriniu raktu su kita (pagrindine) lentele, vardas. „szColumn“ - šios lentelės stulpelis, kuriame saugomas atitinkamas atributas, .
  - „szReferencedObject“ – saugomas lentelės (pagrindinės), su kuria susieta kita lentelė, vardas. „szReferencedColumn“ – šios lentelės atributas, su kuriuo susietas kitos lentelės atributas „szColumn“.

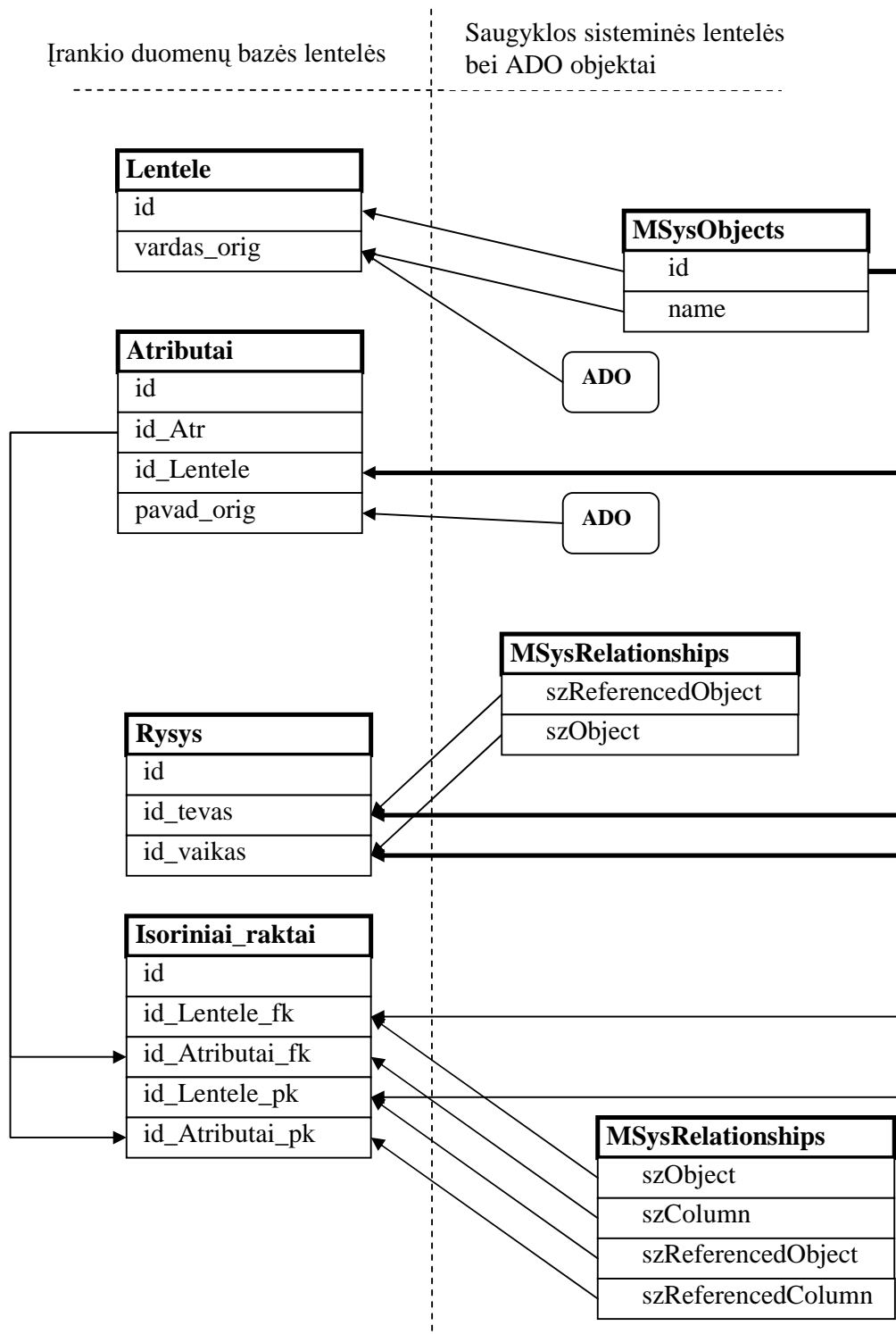
Natūralu, jok skirtingų DBVS sistemines lentelėse saugomi neidentiški metaduomenys bei pačios meta lentelės skirtingos. O „MS Access“ saugyklos tipo atveju jose randami net ne visi sistemos veikimui reikalingi metaduomenys. Tokiu atveju programuojant pasinaudota ADO objektų galimybėmis.

39 paveikslas vaizduoja kokius metaduomenimis įrankis išrenka iš „MS SQL Server“ tipo saugyklos ir į kokias lokalias duomenų bazės lenteles juos nukopijuoja. Svarbus sisteminių lentelių nuskaitymo eiliškumas – pradedama nuo lentelės „sysobjects“ ir metaduomenys, kurių „xtype“ reikšmė lygi „U“, kopijuojami į duomenų bazės lentelę „Lentele“. Synchroniškai skaitoma ir lentelė „syscolumns“. Jos duomenimis užpildoma lentelė „Atributai“, į kurios stulpelį „id“ įrašomos sugeneruotos unikalios „int“ tipo reikšmės. Galiausiai metaduomenimis užpildomos lentelės „Rsys“ ir „Isoriniai\_raktai“.



39 pav. Įrankio duomenų bazės užpildymas „MS SQL Server“ tipo saugyklos metaduomenimis

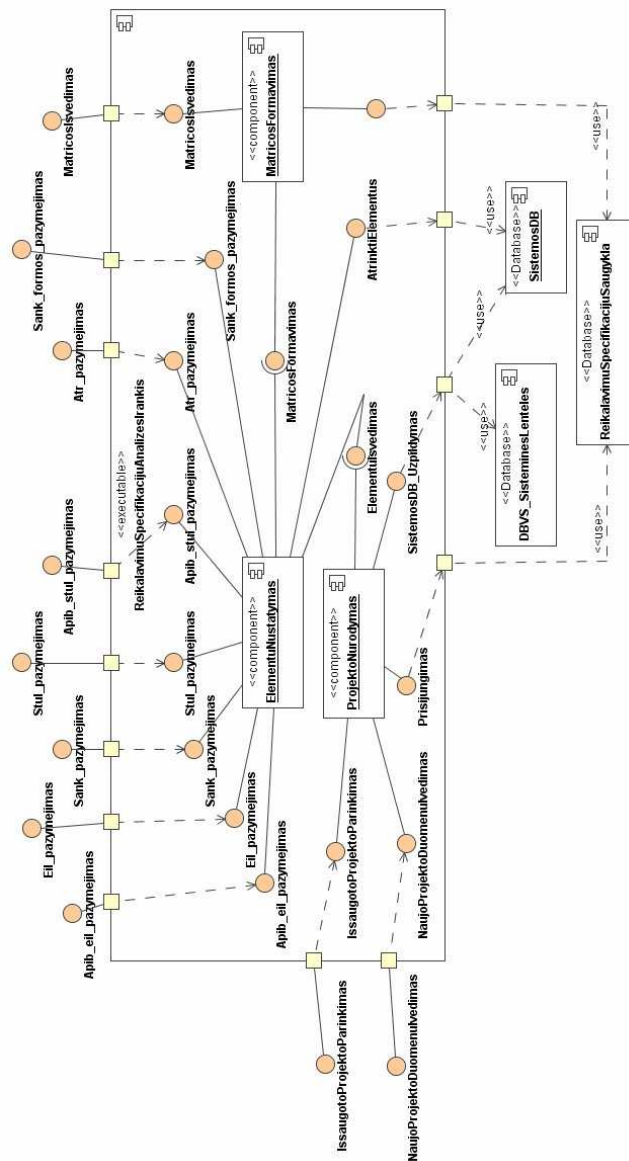
Analizuojant reikalavimų specifikacijas paremtas „MS Access“ pagrindu, įrankis išrenka saugyklos metaduomenis tik iš dviejų aukščiau aprašytų sisteminių lentelių: „MSysObjects“ ir „MSysRelationships“. Tam, kad tinkamai užpildyti įrankio duomenų bazę, papildomai naudojami ADO objektai, kurių pagalba išgaunami trūkstami saugyklos metaduomenys. Tai iliustruota žemiau pateiktame 40 paveiksle. Kaip matyti, kai kurioms įrankio duomenų bazės lentelėms užpildyti reikalinga naudoti kombinuotus būdus.



40 pav. Įrankio duomenų bazės užpildymas „MS Access“ tipo saugyklos metaduomenimis

## 4.2.5 Realizacijos modelis

Komponentų diagramoje (41 pav.) išskirtas pagrindinis komponentas „ReikalavimuSpecifikacijuAnalizesIrankis“ (su stereotipu „executable“), savyje papildomai talpinantis tris kitus komponentus (su stereotipais „component“), kuriais išskaidomas sistemos darbas į tris pagrindinius posistemius, t.y. komponentų struktūra sudaryta pagal sistemos funkcijas. Komponentas „ProjektoNurodymas“ realizuoja klases, naudojamas nurodant reikalavimų specifikacijų rinkinį bei prisijungiant prie saugyklos. Komponentas „ElementuNustatymas“ apima tas klases, kurios naudojamos vartotojui nurodant elementus matricos formavimui. Trečiasis komponentas „MatricosFormavimas“ susideda iš klasių, dalyvaujančių matricos formavime ir rezultatų pateikime vartotojui.



41 pav. Sistemos programinių komponentų architektūra

Komponentas „ProjektoNurodymas“ apima klases:

- Vartotojo sąsajos:
  - Pagrindinis\_langas
  - Sukurti\_nauja\_projekta\_langas
  - meniu\_sankirtos, meniu\_stulpeliai, meniu\_eilutes
  - Atidaryti\_issaugota\_projekta\_langas
- PRANESIMO\_apie\_prisijungima\_prie\_saugyklos\_langas
- Veiklos logikos:
  - Prisijungimo\_prie\_saugyklos\_valdiklis
  - Projekto\_valdiklis
  - Meniu\_valdiklis
  - Matricos\_pradiniu\_elementu\_isvedimo\_valdiklis

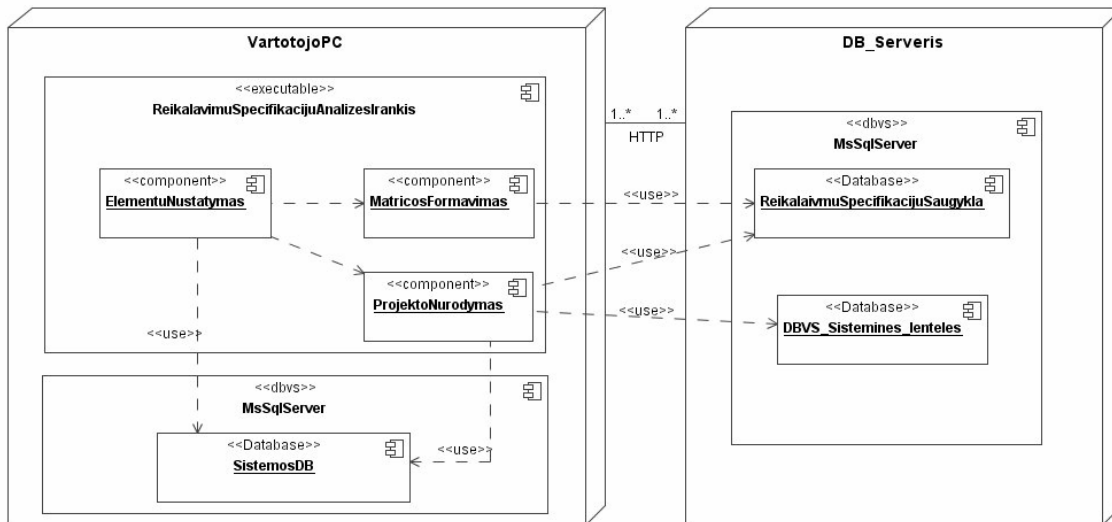
Komponentas „ElementuNustatymas“ apima klases:

- Vartotojo sąsajos:
  - Pagrindinis\_langas
- Veiklos logikos:
  - Matricos\_elementu\_isrinkimo\_valdiklis

Komponentas „MatricosFormavimas“ apima klases:

- Vartotojo sąsajos:
  - Pagrindinis\_langas
  - Matricos\_langas
- Veiklos logikos:
  - Matricos\_formavimo\_valdiklis

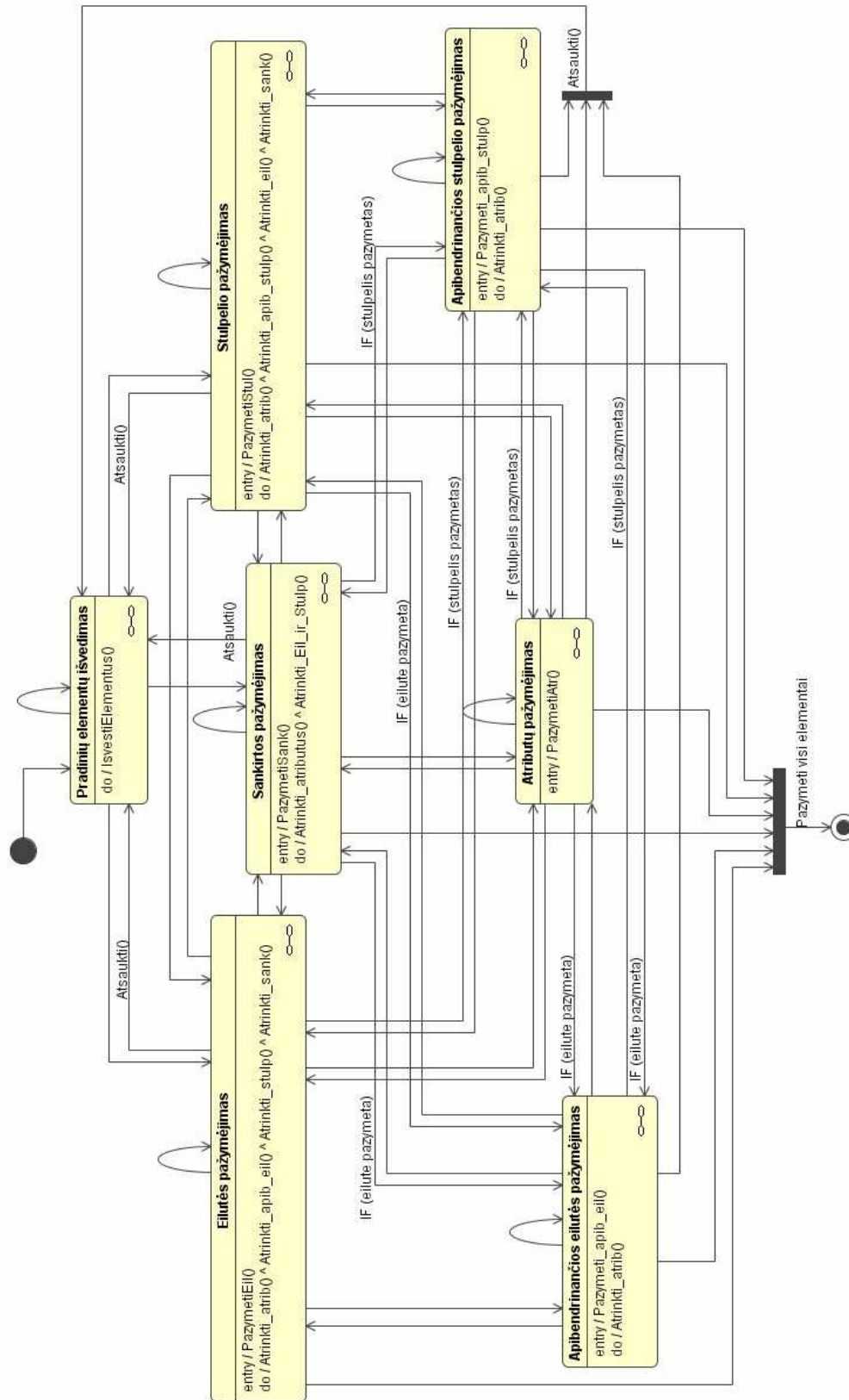
42 paveiksle pateiktas sistemos diegimo modelis, vaizduojantis fizinį techninių įrenginių (vartotojo kompiuterio ir duomenų bazių serverio) išdėstymą bei vykdomųjų programų komponentų paskirstymą juose.



42 pav. Sistemos diegimo modelis

#### 4.2.6 Reikalavimai sistemos funkcionavimo palaikymui

Vienintelis reikalavimas duomenų bazei – ji turi veikti „MS SQL Server 2000“ arba „MS SQL Server 2005“ pagrindu. Taip pat vartotojo kompiuteryje, kuriame suinstaliuojamas analizės įrankis, taip pat turi būti suinstaliuota minėta DBVS. Internetu prie duomenų bazių serverio programa jungiasi „http“ protokolu.



43 pav. Būsenų diagrama matricos elementams nustatyti

Vartotojas pats išsirenka elementus, kuriuos nori matyti matricoje ir kurioje jos vietoje



– eilutėse, stulpeliuose ar sankirtose. Visų pažymėtų elementų priklausomybių ryšiai ir bus atvaizduoti suformuotoje matricoje.

*Matricos formavimas.* Vos tik nurodžius visus reikiamus, spaudžiamas mygtukas „Formuoti matricą“. Sistema atidaro naują langą, kuriame išveda suformuotą elementų priklausomybių matricą, pagal vartotojo nurodytus parametrus.

## 5. Eksperimentinis tyrimas

Atlikus analitinį tyrimą, sukurta sistemos realizacija, pagrįsta parašytu projektu. Toliau atskirai kiekviename skyrelyje apibendrinami eksperimentinio tyrimo rezultatai, detalizuojami pasiekimai baigiamojo darbo metu ir išvados, kurių prieita išbandant jau sukurta programą.

### 5.1 Eksperimentinio diegimo aprašymas

Informacijos sistemų katedroje sukurta ODRES saugykla yra pagrindinė, kuriai buvo numatyta realizuoti šį analizės įrankį. Programos veikimas išbandytas jungiantis prie šios saugyklos tiek nutolusiu būdu (naudojant kompiuterio IP adresą), tiek lokaliai. Abiem atvejais gauti rezultatai nesiskiria ir yra teisingi. Tai įmanoma patikrinti žvelgiant į saugyklos struktūros schemą ir testavimui naudojant konkrečius duomenimis. Tiesa, sistemos veikimas sulėtėja analizuojant nutolusių reikalavimų specifikacijų saugyklą.

Kadangi sistemos funkcionavimui reikalinga lokali duomenų bazė, būtina turėti kompiuteryje suinstaliuotą „MS SQL Server“ duomenų bazių valdymo sistemą. Neegzistuoja reikalavimo kurią versiją naudoti – sistema veikia tiek su „2000“, tiek su „2005“. Tiesa, ODRES reikalavimų specifikacijų saugykla sukurta „MS SQL Server 2000“ pagrindu.

Sistemos išbandymui ir pavyzdinės matricos atvaizdavimui pasinaudosime ODRES metodu. Tarkime, parinksime tokius objektus iš KIS funkcionalumo rezultatų ir duomenų šaltinių perduodamų duomenų srautų modelių:

- Eilučių aibei nurodysime duomenų šaltinius „Duom\_salt“. Šioje saugyklos lentelėje saugoma informacija apie organizacijos objektus saugančius duomenis, reikalingus funkcijoms įvykdyti.
- Lentelėje „Rezultatai“ saugoma informacija apie KIS funkcionalumo metu formuojamus rezultatus arba žodžiu perduodamus informacijos srautus. Šį objektą nurodysime stulpelių aibei.
- Sankirtoms parenkamas „Srautas\_R“. Šioje lentelėje saugoma informacija apie informacijos srautą tarp duomenų šaltinio ir rezultato.[28]

T.y. sudarant kompiuterizuojamos IS reikalavimų specifikaciją ODRES metodu, KIS funkcionalumo rezultatų (išvedamos informacijos) modelyje specifikuojami rezultatai. Po to kuriamas duomenų šaltinių perduodamų duomenų srautų modelis, kuriame identifikuojami duomenų šaltiniai ir specifikuojami jų perduodami duomenų srautai.

Pavyzdinei matricai sukurti saugykloje saugomi tik reliatyvūs duomenys, kuriais

pasinaudosime tik pavyzdžiui sukurti.

Rezultatai	Rezultatai1	Rezultatai2	Rezultatai3
Duom_salt			
Duom_salt1	Srautas_R1	Srautas_R2	
Duom_salt2		Srautas_R3	
Duom_salt3			Srautas_R4
Duom_salt4			
Duom_salt5	Srautas_R5		

44 pav. Nubraižytos matricos pavyzdys.

Taigi, žvelgiant į tokią matricą, galime teigti, kad konkretus specifikuotas kompiuterizuojamos IS rezultatas „Rezultatai1“ susijęs su duomenų šaltiniu „Duom\_salt1“. Be to, jie susiję per informacijos srautą „Srautas\_R1“. Sistemų analitikas naudodamasis tokia matrica gali ieškoti jį dominančių ryšių tarp objektų, pavyzdžiui, ar kiekvienas rezultatas pagrįstas nors vienu duomenų šaltiniu? Šiame pavyzdyje matyti, kad duomenų šaltinis „Duom\_salt4“ neatvaizduojamas į nei vieną rezultatą. Tokiu atveju kyla klausimas – galbūt šis duomenų šaltinis reikalavimų specifikacijoje apskritai nereikalingas.

Saugyklos, veikiančios „MS Access“ pagrindu analizuojamos lokaliai kompiuteryje bei suteikiama galimybė tai atlikti tinkle. Skirtingai nei kiti saugyklų tipai, šiuo atveju .MDB failo nereikia importuoti į duomenų bazių valdymo sistemą (DBVS). Pakanka jį saugoti kietajame diske.

Norint naudotis programa, pakanka nukopijuoti į pasirinktą direktoriją .EXE failą bei kartu su juo egzistuojantį katalogą „files“, kuriame saugomi papildomi sistemos failai.

## 5.2 Sistemos naudojimo instrukcija

Paleidus programą, atidaromas pagrindiniame sistemos langas (45 pav.), kuriame atliekami dauguma veiksmų analizės matricai sukurti. Meniu mygtukų paskirtys ir darbo su programa veiksmų eiliškumas nuosekliai išdėstytas kituose poskyriuose.

ODRES matricu braiziklis

Veiksmai Prisijungimai Nustatymai Pagalba

MATRICOS ELEMENTU PARINKIMAS

APIB.EILUTES EILUTES SANKIRTOS STULPELIAI APIB.STULPELIAI

ATRIBUTU PARINKIMAS

PRISIJUNGIMAI

Irankio duomenų bazė 'REPOSITORIES':

Projektas duomenų bazė:

DB pavadinimas: .....

Projekto pavadinimas: .....

Reikalavimų specifikacijų saugykla:

.....

MATRICOS ELEMENTU PARINKIMAI  Skaičiuoti rezultatus

	Objektas:	Viso:
Apib. eilute:	-----	--
Eilute:	-----	--
Sankirta:	-----	--
Stulpelis:	-----	--
Apib. stulpelis:	-----	--

SANKIRTŲ VAIZDAVIMAS

Matricos eilučių ir stulpelių sankirtas vaizduoti:

sankirtos objekto pažymėtų atributų reikšmėmis

simboliu arba simbolių eilute:

Prisijunkite prie irankio duomenų bazės

45 pav. Sistemos pagrindinis langas.

### 5.2.1 Prisijungimas

Pradiniai veiksmai prieš atliekant reikalavimų specifikacijų analizę - prisijungimas prie įrankio duomenų bazės bei pačios reikalavimų specifikacijų saugyklos. Tam skirtas meniu mygtukas „Prisijungimai“, kuris turi du submenu mygtukus:

1. „Prisijungti prie įrankio DB...“ – prisijungimui prie lokaliai įrankio duomenų bazės.
2. „Prisijungti prie RS saugyklos...“ – prisijungimui prie lokaliai arba nutolusios reikalavimų specifikacijų saugyklos.

The image shows a Windows-style dialog box with a title bar that reads "Prisijungimas prie įrankio DB". The dialog has a light beige background and a dark border. It contains the following elements from top to bottom:
 

- A label "Serveris:" followed by a text input field.
- A label "localhost:" followed by a small square checkbox.
- A label "Vartotojo vardas:" followed by a text input field.
- A label "Slaptažodis:" followed by a text input field.
- At the bottom, there are two buttons: "Jungtis" on the left and "Atšaukti" on the right.

46 pav. Prisijungimo prie įrankio duomenų bazės langas

Jungiantis prie įrankio lokalsios duomenų bazės (46 pav.), reikalinga įvesti:

- Serverio vardą (arba pažymėti „varnelę“, reiškiančią, kad bus jungiamasi prie egzistuojančio lokalaus serverio);
- Vartotojo vardą;
- Prisijungimo slaptažodį.

Sėkmingai prisijungus prie įrankio duomenų bazės, būtina prisijungti prie norimos analizuoti reikalavimų specifikacijų saugyklos. Čia prisijungimo duomenys skiriasi, priklausomai nuo saugyklos tipo. „MS SQL Server saugyklos atveju įvedami privalomi duomenys (20 pav47 pav.):

- Serverio, kuriame saugoma reikalavimų specifikacijų saugykla, vardas;
- Vartotojo vardas;
- Prisijungimo slaptažodis;
- Duomenų bazės vardas;
- Jeigu pažymėta „varnelė“ „Išsaugoti projektą“, tuomet būtina įvesti ir naują projekto pavadinimą.

47 pav. Prisijungimo prie MS SQL Server tipo saugyklos langas

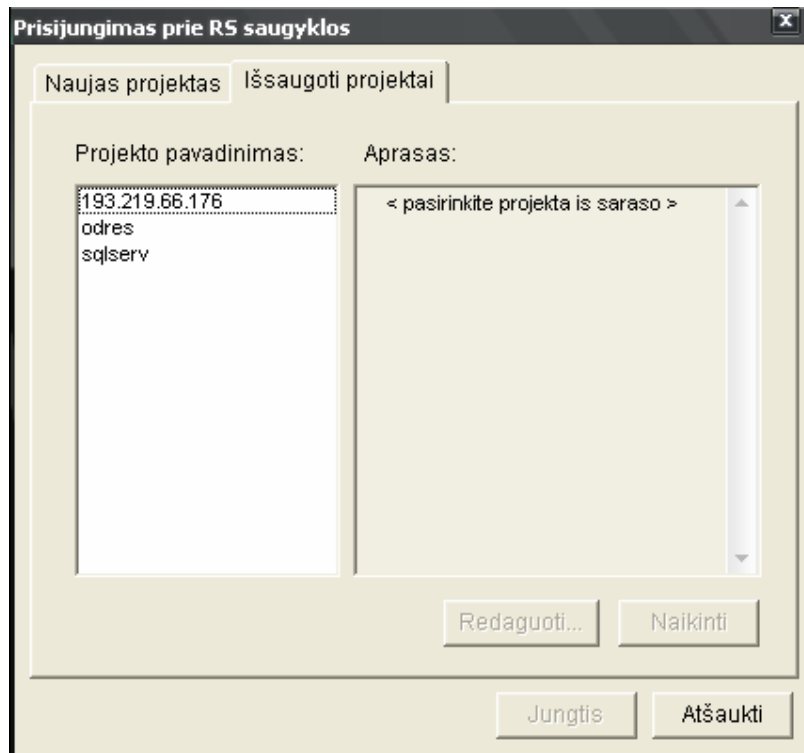
„MS Access“ saugyklos tipo atveju (48 pav.) būtina nurodyti .MDB failą bei naują projekto pavadinimą, jeigu atitinkamai pažymėtas langelis „Išsaugoti projektą“. Visi kiti laukai – neprivalomi, tačiau gali būti reikalaujama šių duomenų:

- Vartotojo vardas;
- Slaptažodis;
- .MDB failo slaptažodis.

48 pav. Prisijungimo prie MS Access tipo saugyklos langas

Prieš jungiantis prie „MS Access“ tipo saugyklos, būtina nustatyti sisteminių lentelių „MSysRelationships“ bei „MSysObjects“ , esančių „MS Access“ saugykloje, skaitomumo teises. Tai galima atlikti per meniu „Tools → Security → User and Group Permissions“. Priešingu atveju analizės įrankis negalės nuskaityti meta duomenų.

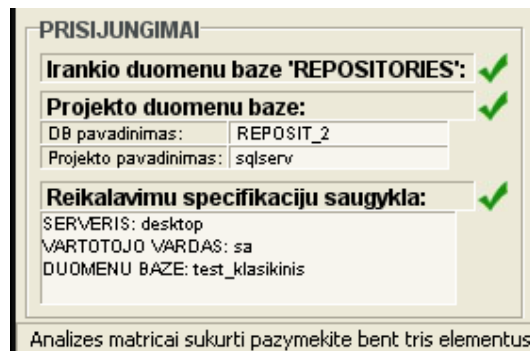
Kartą suvedus prisijungimo duomenis ir juos išsaugojus įrankio duomenų bazėje, vėliau galima jungtis prie atitinkamos reikalavimų specifikacijų saugyklos iš naujo nesuvedinėjant prisijungimo duomenų. Lange „Prisijungimas prie RS saugyklos“ paspaudus šaknelę „Išsaugoti projektai“ atveriamas lango skiltis (49 pav21 pav.), kurioje pateikiami visi išsaugoti projektai. Dešiniajame sąrašė matomi projektų pavadinimai. Kairiuoju pelės klavišu du kartus paspaudus ant pasirinktojo, kairiajame sąrašė pateikiama išsaugota informacija apie jį. Paspaudus „Redaguoti“ galima pakeisti bet kurio išsaugoto projekto prisijungimo duomenis ar aprašą. Mygtuku „Naikinti“ ištrinamas projektas iš įrankio duomenų bazės.



49 pav. Išsaugoto projekto (reikalavimų specifikacijų rinkinio) atidarymo langas

Pagrindinio lango apatiniame kairiajame kampe informacinėje skiltyje „Prisijungimai“ (50 pav.50 pav.) pateikiama informacija apie aktyvius/neaktyvius prisijungimus:

- „Įrankio duomenų bazė ‚REPOSITORIES‘“. Ikona indikuoja, ar prisijungta prie lokalios įrankio duomenų bazės.
- „Projekto duomenų bazė“. Papildomai išvedami duomenų bazės bei projekto, prie kurio prisijungta, pavadinimas.
- „Reikalavimų specifikacijų saugykla“. Išvedami tokie duomenys, kaip serverio vardas, vartotojo vardas ir saugyklos (duomenų bazės) pavadinimas.



50 pav. Informacinė skiltis „Prisijungimai“



Šioje skiltyje naudojamos dviejų tipų ikonos, pateiktos 51 pav. Raudonas perbrauktas apskritimas žymi, jog prie atitinkamos duomenų bazės programa neprisijungusi. Žalia „varnelė“ – reiškia esamą aktyvų prisijungimą.



51 pav. Naudojamos ikonos

Pagrindinio sistemos lango apačioje esanti juosta (52 pav.) skirta išvesti įvairiai informacijai apie tai, kokius veiksmus atlieka programa.

Prisijungta prie įrankio duomenų bazės. Jūnkites prie R5 saugyklos

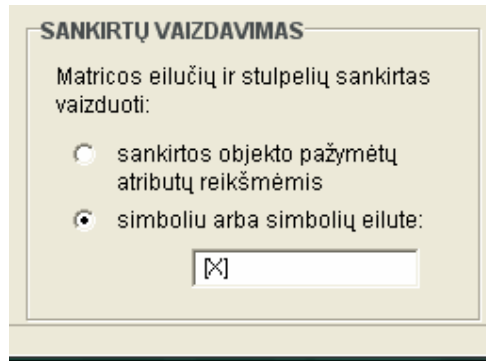
52 pav. Informacinė juosta

### 5.2.2 Matricos elementų parinkimas

Sėkmingai prisijungus prie lokalsios įrankio duomenų bazės ir reikalavimų specifikacijų saugyklos, programos pagrindiniame lange (19 pav. 45 pav.) esančiuose sąrašų laukuose išvedami pradiniai duomenys. Galima pradėti žymėti pradinis elementus analizės matricai kurti. Parinkus bet kurį matricos elementą, būtina nurodyti bent vieną jo atributą. Analizės matricai sukurti būtina parinkti bent eilutės, sankirtos ir stulpelio objektus. Jie laikomi pagrindiniais. Apibendrinančios eilutės ir apibendrinančio stulpelio elementai laikomi papildomais ir neprivalomais.

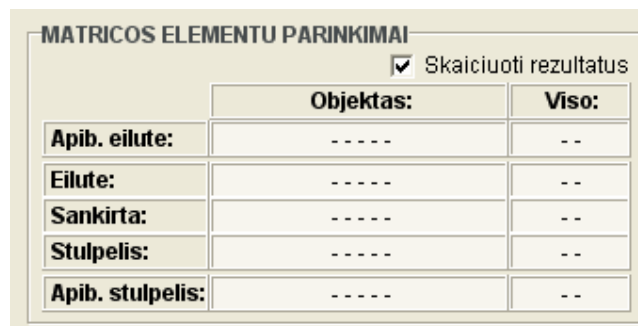
Matricos elementų parinkimą galima atlikti pradėdant nuo bet kurio pagrindinio objekto – eilutės, stulpelio ar sankirtos. Atžvilgiu bet kurio pažymėto elemento atrenkami ir į likusius sąrašų laukus išvedami atitinkamai susiję elementai. Po kiekvienu matricos elementų lauku žemiau išdėstyti atributų laukai. Bet kuriam pažymėtam objektui atrenkami ir išvedami ir jo atributai.

Pagrindinio lango apatiniame dešiniajame kampe egzistuoja skiltis „Sankirtų vaizdavimas“ (53 pav.). Pažymėjus pirmąjį „radio“ tipo mygtuką, matricos eilučių ir stulpelių sankirtose elementų ryšiai bus žymimi juos siejančio elemento reikšme. Tai reiškia, jog šis vaizdavimo būdas vaizduos ne tik tai, jog du objektai susiję, tačiau konkrečiai parodo kokiu elementu. Prieš kuriant matricą, nurodžius antrąjį „radio“ tipo mygtuką, eilutės ir stulpelio susijusių dviejų elementų ryšys sankirtos celėje bus vaizduojamas vartotojo nurodytu simboliu.



53 pav. Informacinė skiltis „Sankirtų vaizdavimas“

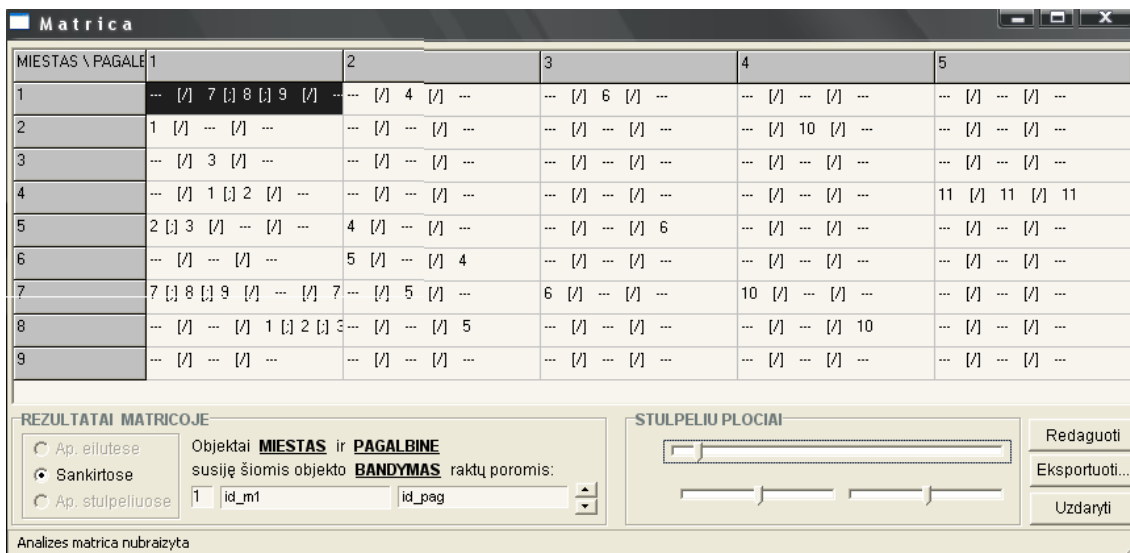
Papildomai pagrindinio lango informacinėje skiltyje „Matricos elementų parinkimai“ (54 pav.) rodoma kiek tam tikras pažymėtas matricos elementas saugykloje saugo duomenų. Esant nors vienai reikšmei „0“, nebus tikslinga kurti analizės matricą. Vartotojui nusprendus, jog šioje skiltyje pateikiama informacija jam nenaudinga, jis gali tai išjungti paspausdamas „Skaičiuoti rezultatus“.



54 pav. Informacinė skiltis „Matricos elementų parinkimai“

### 5.2.3 Matricos kūrimas

Pažymėjus bent tris pagrindinius objektus – eilutę, stulpelį ir sankirtą - galima kurti analizės matricą. Tam spaudžiamas meniu mygtukas „Veiksmai → Kurti analizės matricą“. Atsidaro naujas langas (23 pav.) ir jame išvedami rezultatai. Pagrindiniame lauke pateikiama pagal vartotojo atliktus pažymėjimus nubraižyta matrica. Jos stulpelių pločius galima reguliuoti slankiojant slinktis skiltyje „STULPELIŲ PLOČIAI“. Dešinėje – papildomi mygtukai.



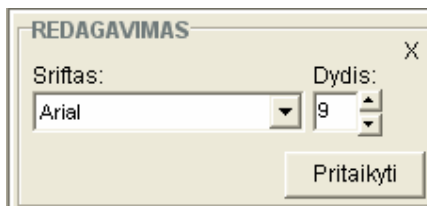
55 pav. Suformuotos matricos langas

Skiltis „REZULTATAI MATRICOJE“ yra informacinė. Ji skirta patikslinti kokius rezultatus išvedami matricos celėse:

- Apibendrinančiose eilutėse
- Sankirtose
- Apibendrinančiuose stulpeliuose

Pirmajame laukelyje rodomas išorinių raktų porų eilės numeris, kadangi matricoje gali būti sukirsti objektai esant keliems ryšiams. Antrajame ir trečiajame laukuose išvedami atitinkamai pažymėtų objektų išoriniai raktai. Užvedus pelės rodyklę ant kurio nors, bus parodomas paaiškinimas, jeigu tik vartotojas prieš tai jį susikūrė.

Paspaudus mygtuką „Redaguoti“ atidaromas panelis (56 pav.), kuriame galima nurodyti šriftą ir jo dydį, taikomą matricoje esantiems rezultatams.



56 pav. Šrifto ir šrifto dydžio keitimo panelis

Labai patogiu rezultatus išsaugoti .XLS faile. Tam spaudžiama „Eksportuoti...“ ir atsidaro naujas panelis (57 pav. 57 pav.). Čia egzistuoja dvi skiltys:

1. „Rezultatų filtravimas“ – galima atlikti matricoje pateiktų rezultatų filtravimą:
  - Visus rezultatus saugoti vienoje matricoje, t.y. taip, kaip matoma ekrane.

- Visus rezultatus saugoti atskirose matricose.
  - Galima pasirinkti kurias matricas saugoti.
2. „Specialūs simboliai“ – vartotojui suteikiama galimybė pakeisti specialius simbolius, naudojamus matricoje.

Paspaudus mygtuką „Saugoti“ lieka tik nurodyti failo vardą ir direktoriją, kurioje programa išsaugos rezultatų failą.

57 pav. Matricos eksportavimo panelis

#### 5.2.4 Papildomos savybės

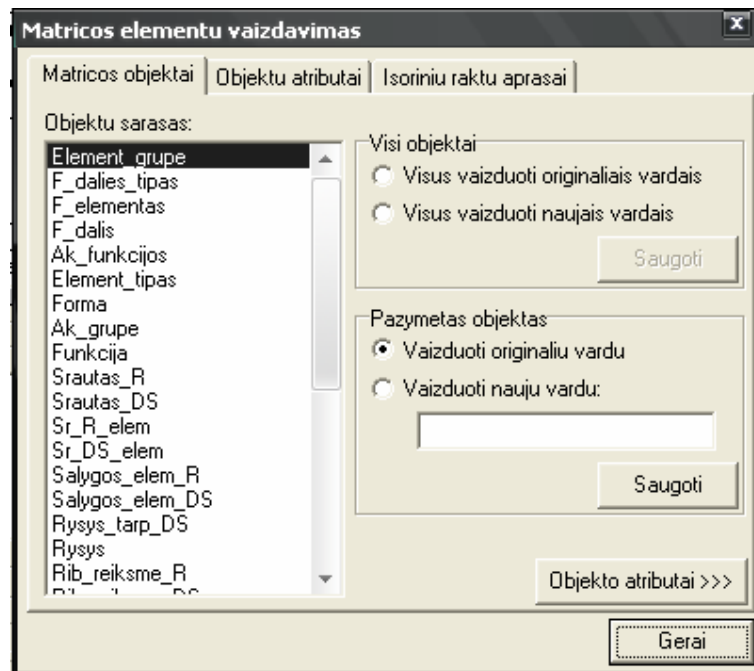
Sistemoje vartotojui leidžiama atlikti tam tikrus nustatymus. Tam skirtas meniu mygtukas „Nustatymai“, kurį paspaudus išskleidžiami kiti papildomi mygtukai:

- „Sumažintas langas“.
- „Pradinių elementų vaizdavimas“.
- „Specialūs simboliai“.
- „Prisijungimai“.

Paspaudus „Sumažintas langas“, pagrindinis sistemos langas bus rodomas be apatinių informacinių skilčių – „Prisijungimai“, „Matricos elementų parinkimai“, „Sankirtų vaizdavimas“.

Paspaudus mygtuką „Matricos elementų vaizdavimas“ vartotojui suteikiama galimybė atlikti įvairių matricos elementų vaizdavimo nustatymus. Egzistuoja trys skiltys:

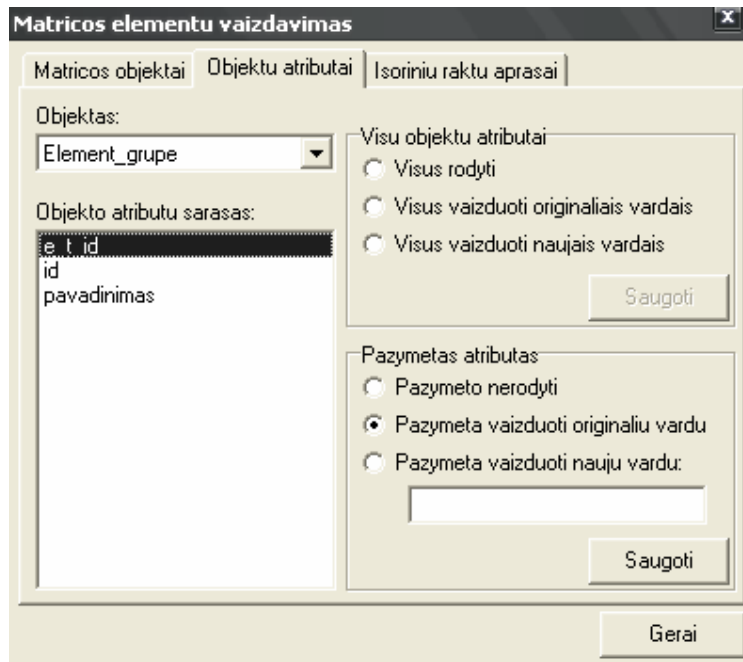
1. 58 paveiksle pavaizduota lango skiltis, skirta matricos objektų rodomumui nustatyti. Kairėje pusėje pateikiamas objektų, kurių vaizdavimą ketinama nustatinėti, sąrašas. Dešinėje lango dalyje mygtukų pagalba keičiami objektų pavadinimai arba nurodoma juos apskritai sistemoje nevaizduoti.



58 pav. Matricos objektų vaizdavimo nustatymas

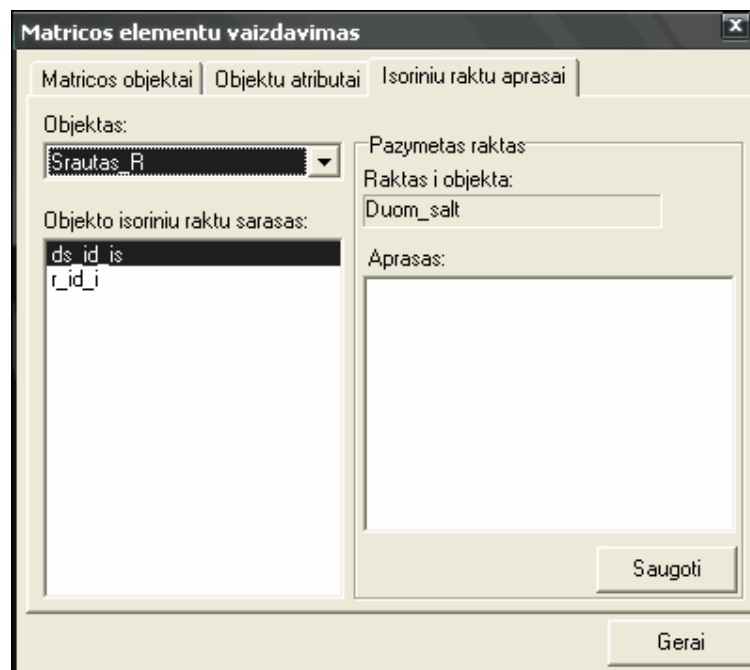
Pažymėjus kurį nors objektą ir paspaudus mygtuką „Objekto atributai“, bus atveriamas sekanti skiltis (59 pav.) su joje tiesiogiai bus galima nustatinėti jau pažymėto objekto atributus.

2. 59 pav. – skiltis, matricos objektų (elementų) atributų rodomumui nustatyti. Į iškrentantį sąrašą išvedami visi RS saugyklos objektai. O po juo esančiame sąrašo lauke – konkretaus objekto atributai. T.y. pasirinkus kurį nors objektą, išvedami visi jo atributai. O jų vaizdavimas nustatinėjamas analogiškai kaip ir pačių objektų.



59 pav. Objektų atributų vaizdavimo nustatymas

- 60 pav. pavaizduotoje lango skiltyje vartotojas gali apsirašyti bet kuriuos konkretaus objekto išorinius raktus. Iškretnančiame sąrašė išvardijami objektai, turintys išorinius raktus. O patys raktai pateikiami žemiau esančiame lauke, pažymėjus konkretų objektą.



60 pav. Išorinių raktų aprašų sukūrimas

Laukelyje „Raktas į objektą“ išvedamas to objekto vardas, į kurį nurodo pažymėtas raktas. Aprašas įvedamas į laukelį „Aprašas“.

Vartotojui suteikiama galimybė pakeisti specialius simbolius. Tai atliekama 61 paveiksle pateiktame lange, kuris atidaromas meniu mygtuku „Specialūs simboliai“. Pakeitimai tiesiog įvedami į norimą laukelį „Naujas:“ ir užfiksuojami mygtuko „Išsaugoti“ paspaudimu. Norint atkurti specialius simbolius pagal nutylėjimą, tam skirtas mygtukas „Numatyti“.

**Specialūs simboliai**

Galite pakeisti specialius simbolius, naudojamus braizant analizes matrica. Kiekvieno aprašymas toliau atskirai

1. Naudojamas atskirti keletui skirtingu rezultatu vienoje celeje, kur kiekvienas is ju gautas analizuojant skirtingus rysius tarp dvieju objektu.  
Esamas: [ / ] Naujas: [ / ]
2. Naudojamas atskirti keletui skirtingu rezultatu vienoje celeje, kur kiekvienas is ju atskirai gautas analizuojant ta pati rysis tarp dvieju objektu.  
Esamas: [ : ] Naujas: [ : ]
3. Atskiriamos skirtingos to pacio objekto atributu (pazymetu pagrindiniame lange) reiksmes, isvestos kaip vienas rezultatas.  
Esamas: [ -- ] Naujas: [ -- ]
4. Simbolis, indikuojantis, jog tarp dvieju atitinkamu analizuojamu objektu rysys neegzistuoja.  
Esamas: [ ... ] Naujas: [ ... ]
5. Simbolis indikuojantis, jog neegzistuoja objekto atributas. Taip gali buti, jei sis atributas yra isorinis raktas (FK) i kita objekta arba jei siam atributui neuzdrausta NULL reiksme.  
Esamas: [ \* ] Naujas: [ \* ]

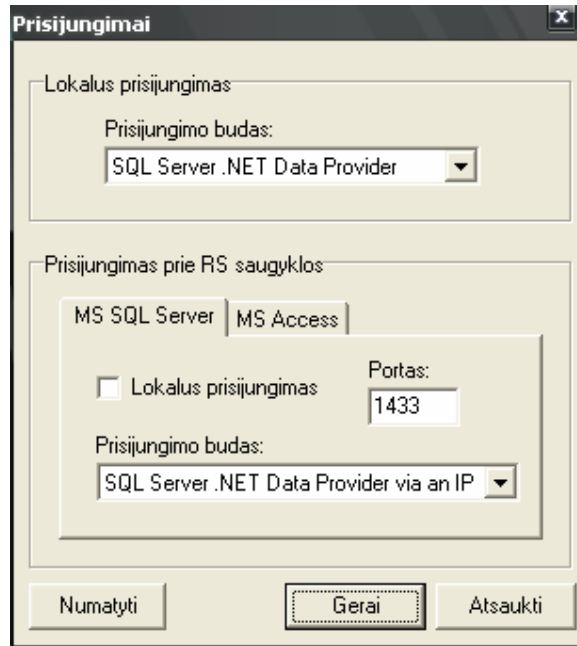
Numatyti Išsaugoti Uždaryti

61 pav. Specialių matricos simbolių keitimo langas

Vartotojui suteikiama galimybė nustatyti programos prisijungimo prie įrankio duomenų bazės bei saugyklos būdus (62 pav.):

- Prisijungimui prie įrankio duomenų bazės skirta skiltis „Lokalus prisijungimas“.

- Prisijungimo prie RS saugyklos – žemiau esanti skiltis. Atskirai kiekvienam saugyklos tipui skirti du paneliai: „MS SQL Server“ ir „MS Access“. Pirmojo tipo saugyklai papildomai galima nurodyti portą. Ketinant jungtis ir analizuoti lokaliai saugomą specifikaciją, rekomenduojama pažymėti „varnelę“ „Lokalus prisijungimas“.



62 pav. Programos prisijungimų būdų nustatymo langas

Paspaudus mygtuką „Numatyti“, bus atlikti nustatymai pagal nutylėjimą. Mygtuko „Gerai“ paspaudimu išsaugomi vartotojo atlikti nauji nustatymai.



### 5.3 Sistemos veikimo ir savybių analizė

Norint atlikti konkrečios reikalavimų specifikacijų saugyklos analizę naudojant sukurta sistema, egzistuoja pagrindinis nefunkcinis reikalavimas – saugykla turi veikti „MS SQL Server“ pagrindu. Kuriant programą didžiausias dėmesys buvo skirtas ODRES reikalavimų specifikacijų saugyklai. Ja remiantis buvo realizuojamas sistemos funkcionalumas. Tačiau nuspręsta neapsiriboti vien tik šia saugykla ir jos struktūra. Buvo nuspręsta sukurti kaip galima universalesnį analizės įrankį, t.y., kad sistema veiktų su kitos skirtingos struktūros saugykla nei ODRES. Programa išbandyta su daugeliu skirtingų saugyklų pavyzdžių tam, kad garantuotume programos funkcionalumą. Siekiant tikslo kuo labiau praplėsti sistemos taikymą, eksperimentų metu buvo ieškota specialių atvejų ir išimčių, susijusių su saugyklos struktūra.

Sukurti įrankį, skirtą analizuoti ne tik ODRES reikalavimų specifikacijų saugyklą, buvo viena iš idėjų kaip praplėsti jo funkcionalumą ir universalumą. Kuriant šią analizės priemonę, iškeltas papildomas reikalavimas – sistemą papildyti, jog ji būtų nepriklausoma nuo saugyklos tipo. T.y. programos funkcionalumui papildomai apibrėžtas naujas universalumo reikalavimas. Nepaisant „MS SQL Server“ saugyklos tipo, įrankis taip pat realizuotas analizuoti saugyklą, veikiančią „MS Access“ pagrindu. Sudėtinga sukurti analizės įrankį, kuris veiktų su bet kokio tipo RS saugykla. Tam paliktos praplėtimo galimybės.

Nors sistema nerealizuota analizuoti reikalavimų specifikacijų kitokio tipo saugyklą, tai neabejotinai galima atlikti importavus ją kaip lokalią duomenų bazę, veikiančią „MS SQL Server“ pagrindu.

Elementai matricai braižyti skirstomi:

1. Pagrindiniai – eilutės, stulpeliai, sankirtos. Norint sukurti analizės matricą, šiuos elementus būtina pažymėti. Juos renkantis pelės paspaudimu, nėra svarbu kokia eilės tvarka tai bus daroma. Atlikti elementų pažymėjimus galima bet kuria tvarka, pradedant tiek nuo eilutės, stulpelio ar sankirtos.
2. Papildomi – apibendrinančios eilutės, apibendrinantys stulpeliai. Tai elementai, kurių nebūtina pažymėti ir nepaisant to matricą bus galima sukurti. Juos galima pasirinkti pelės paspaudimu tik tuomet, kai atitinkamai jau pasirinktas eilutės ar stulpelio elementas.

Pažymėjus bet kurį objektą, būtina pasirinkti bent vieną jo atributą, kurio reikšmėmis bus vaizduojamas atitinkamas elementas matricoje.

Pasirinkus objektą iš sąrašo lauko, šį veiksmą galima pakartoti iš naujo pele pažymint kitą

elementą. Kiekviena kartą tai darant, elementams suteikiamas prioritetas, t.y. atžvilgiu elemento, turinčio aukštesnį prioritetą, atrenkami visi likę elementai. Bet kada galima atšaukti visus pažymėjimus ir šiuos veiksmus kartoti iš naujo.

Papildomai galima atlikti keletą programos nustatymų - vartotojui leidžiama pasirinkti kokių būdu jungtis prie reikalavimų specifikacijų saugyklos kiekvieno tipo atveju. Tai naudinga, jei tam tikru atveju kompiuteryje nėra suinstaliuota reikalingų tvarkyklių ir pan.:

1. MS SQL Server lokaliai prisijungimui:
  - SQL Server .NET Data Provider.
  - OLE DB Provider for SQL Server.
  - ODBC Driver for SQL Server.
2. MS SQL Server nutolusiam prisijungimui:
  - SQL Server .NET Data Provider via an IP Address.
  - OLE DB Provider for SQL Server via an IP Address.
  - ODBC Driver for SQL Server via an IP Address.
3. MS Access prisijungimui:
  - ODBC Driver for Access
  - OLE DB Provider for Jet

Sistemoje naudojami specialūs simboliai tam tikrų situacijų atvejais. Juos vartotojas gali pakeisti pagal savo poreikius. Dėl suprantamo ir neklaidinančio rezultatų pateikimo analizės matricioje, egzistuoja specialių simbolių reikšmės pagal nutylėjimą:

1. Simbolis „[/]“ naudojamas atskirti keliems rezultatams, gautiems, analizuojant kelis skirtingus ryšius tarp dviejų objektų, susietų keliais išoriniais raktais. Toliau tam, kad situaciją iliustruoti pavyzdžiu, naudojama ODRES reikalavimų specifikacijų saugykla:

13. lentelė - Specialaus simbolio „[/]“ panaudojimas sistemoje.

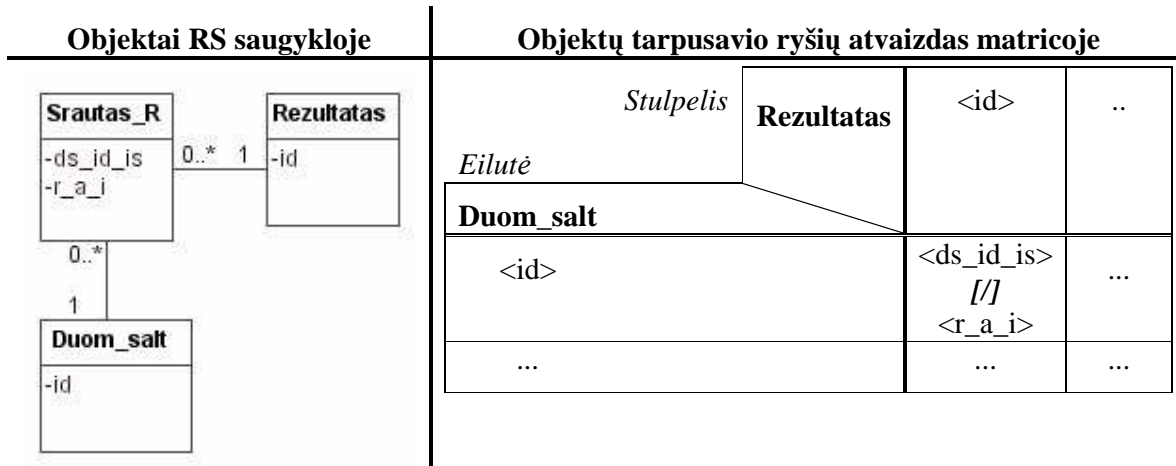
Objektai RS saugykloje		Objektų tarpusavio ryšių atvaizdas matricioje											
<b>Srautas_DS</b> -ds_id_is -ds_id_i	0..* ——— 0..*	1 ——— 1	<b>Duom_salt</b> -id										
			<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; border: none;"><i>Ap.eilutė</i></th> <th style="text-align: center; border: none;"><i>Eilutė</i></th> </tr> <tr> <th style="text-align: center; border: none;"><b>Srautas_DS</b></th> <th style="text-align: center; border: none;"><b>Duom_salt</b></th> </tr> </thead> <tbody> <tr> <td style="border: none; padding: 5px;"><code>&lt;ds_id_is&gt; [/]</code></td> <td style="border: none; padding: 5px;"><code>&lt;id&gt;</code></td> </tr> <tr> <td style="border: none; padding: 5px;"><code>&lt;ds_id_i&gt;</code></td> <td style="border: none; padding: 5px;"></td> </tr> <tr> <td style="border: none; padding: 5px; text-align: center;">...</td> <td style="border: none; padding: 5px; text-align: center;">...</td> </tr> </tbody> </table>	<i>Ap.eilutė</i>	<i>Eilutė</i>	<b>Srautas_DS</b>	<b>Duom_salt</b>	<code>&lt;ds_id_is&gt; [/]</code>	<code>&lt;id&gt;</code>	<code>&lt;ds_id_i&gt;</code>		...	...
<i>Ap.eilutė</i>	<i>Eilutė</i>												
<b>Srautas_DS</b>	<b>Duom_salt</b>												
<code>&lt;ds_id_is&gt; [/]</code>	<code>&lt;id&gt;</code>												
<code>&lt;ds_id_i&gt;</code>													
...	...												

Kadangi objektas „Srautas\_DS“ turi du išorinius raktus į objektą „Duom\_salt“, tokiu

atveju matricioje šiems dvigubo ryšio rezultatams atskirti naudojamas simbolis „[/]“. Šiame ir sekančiuos specialių simbolių pavyzdžiuose skaitoma, jog matricioje elementai vaizduojami atributais, pavaizduotais kairėje lentelės pusėje esančiame paveiksle.

Dar vienas pavyzdys, iliustruojantis simbolio „[/]“ panaudojimą:

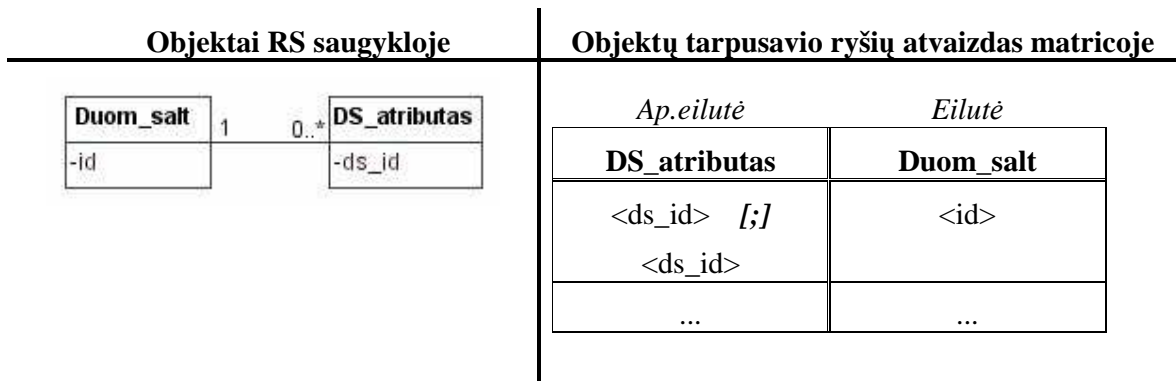
14. lentelė- Specialaus simbolio „[/]“ panaudojimas sistemoje.



Čia „Duom\_salt“ parinktas eilutės elementu matricioje; „Rezultatas“ – kaip stulpelio elementas; „Srautas\_R“ – kaip eilutės ir stulpelio sankirtos elementas.

2. Simbolis „[;]“ naudojamas panašiose situacijose, kaip ir prieš tai aprašytas. Tik šiuo atveju juo atskiriami keli skirtingi rezultatai, gauti esant tam pačiam ryšiui tarp dviejų objektų.

15. lentelė- Specialaus simbolio „[;]“ panaudojimas sistemoje.



Čia sukurtos matricos eilutėje vaizduojamos elementų „Duome\_salt“ reikšmės, o stulpelyje – „DS\_atributas“ reikšmės. Analizuojant vieną vienintelį ryšį tarp šių dviejų objektų, gali būti rasto kelios reikšmės, kurioms atskirti ir naudojamas simbolis „[;]“.

3. Pažymėjus bet kuri elementą programos pagrindiniame lange – apibendrinančią eilutę, eilutę, sankirtą, stulpelį ar apibendrinantį stulpelį - būtina pažymėti ir jo atributą, kurio reikšme atitinkamas elementas bus vaizduojamas. Simboliu „\_,\_“ atskiriamos

reikšmės, gautos kiekvieno atributo atveju atskirai. Tarkime, kaip matricos elementą parenkame „Srautas\_DS“ (matricos sankirtai) ir pažymime du jo atributus – „kodas“ ir pavadinimas“. Tuomet nubraižytoje analizės matricoje šis elementas bus vaizduojamas dviem minėtomis reikšmėmis, atskirtomis simboliu „\_“.

16. lentelė- Specialaus simbolio „\_“ panaudojimas sistemoje.

Objektai RS saugykloje	Objektų tarpusavio ryšių atvaizdas matricoje								
<table border="1"> <tr><td><b>Srautas_DS</b></td></tr> <tr><td>-id</td></tr> <tr><td>-kodas</td></tr> <tr><td>-pavadinimas</td></tr> </table>	<b>Srautas_DS</b>	-id	-kodas	-pavadinimas	<p style="text-align: center;"><i>Sankirta</i></p> <table border="1"> <tr><td><b>Srautas_DS</b></td></tr> <tr><td>&lt;kodas&gt; [_,_]</td></tr> <tr><td>&lt;pavadinimas&gt;</td></tr> <tr><td>...</td></tr> </table>	<b>Srautas_DS</b>	<kodas> [_,_]	<pavadinimas>	...
<b>Srautas_DS</b>									
-id									
-kodas									
-pavadinimas									
<b>Srautas_DS</b>									
<kodas> [_,_]									
<pavadinimas>									
...									

Tarkime, „Srautas\_DS“ atveju, reikalavimų specifikacijų saugykloje neegzistuočių kurio nors atributo reikšmė (įrašas), tokiu atveju vietoj jo bus naudojamas specialus simbolis „\*“.

4. Jeigu tarp dviejų objektų neegzistuoja joks juos siejantis ryšys, tuomet analizės matricos celėje ši situacija vaizduojama simboliu „---“.

Reiktų paminėti tam tikras su saugyklos struktūra susijusias situacijas, kurioms esant analizės įrankis yra tinkamas atlikti analizei:

1. Tarp dviejų lentelių egzistuoja du ar daugiau ryšių.
2. Viena ir ta pati lentelė turi ryšį į save.
3. Lentelė turi turėti ne daugiau kaip vieną PK lauką.

Reikalavimų specifikacijos objektų pavadinimai gali būti visiškai neinformatyvūs sistemos vartotojui, nes dažniausiai tam naudojami sutrumpinimai ir pan. Todėl suteikiama galimybė susikurti naujus pavadinimus, kurie bus vaizduojami kaip matricos elementai. Taip pat tie patys objektai gali turėti įvairių atributų, kurie neturi prasmės sistemos vartotojui. T.y. egzistuoja PK ir FK laukai bei kt. Tokiu atveju galima atlikti nustatymus, jog apskritai nerodyti šiuos atributus pagrindinio lango laukuose, kuriuose vartotojas renkasi elementus. Be to, kaip ir pačių objektų, taip pat ir atributų pavadinimus galima pakeisti labiau priimtinais. Tokio pobūdžio nustatymai atliekami meniu skiltyje „Nustatymai → Pradinių elementų vaizdavimas“.

Būtina atsižvelgti į tai, jog vartotojo kompiuteris egzistuoja LAN tinkle. Dėl tos priežasties nebus įmanoma prisijungti prie nutolusios reikalavimų specifikacijų saugyklos,

jeigu sistemų administratorius neatliks reikalingų tinklo ugniasienių (angl. „firewall“) nustatymų.

Kadangi vartotojui suteikiama galimybė eksportuoti rezultatų matricą į .XLS failą, būtina kompiuteryje turėti suinstaliuotą „MS Excel“. Eksportuotus rezultatus galima redaguoti bei atsispausdinti.

„MS SQL Server“ saugyklos tipo atveju jungiantis prie nutolusio kompiuterio, reikia žinoti portą, kuris, pagal nutylėjimą, yra 1411. Tam tikrais saugumo atvejais šis portas gali būti uždraustas. Todėl vartotojui suteikiama galimybė jį pakeisti ir jungti naudojant naują portą.

Prieš jungiantis prie RS saugyklos, veikiančios „MS Access“ pagrindu, būtina nustatyti sisteminių lentelių „MSysObjects“ ir „MSysRelationships“ skaitymo teises. Priešingu atveju sistema negalės nuskaityti sisteminių duomenų. Taip pat, jei .MDB failas turi slaptažodį, vartotojas jį gali įvesti jungiantis arba panaikinti naudojant DBVS.

## 5.4 Kokybės kriterijų įvertinimas

3.5 skyriuje pagal tam tikrus kriterijus lyginamos pasaulyje egzistuojančios analogiškos sistemos. Išskyrus „MagicDraw“ pakete esančias, visos kitos reikalavimų specifikacijų analizavimo priemonės yra priklausančios nuo konkretaus metodo. Eksperimentiniam tyrimui atlikti sukurtas įrankis skirtas analizuoti ODRES metodu specifikuotus reikalavimus. Tačiau jį kuriant realizuotas tam tikras universalumo lygis būti nepriklausomu nuo reikalavimų specifikavimo metodo. Tai reiškia, jog sukurta analizės priemonė galima analizuoti bet kokią reikalavimų specifikaciją, veikiančią „MS SQL Server“ bei „MS Access“ pagrindu.

Nuspręsta, jog tinkamiausia analizės forma yra matrica, kuri taip pat naudojama ir kitose pasaulyje gerai žinomose CASE priemonėse: „Oracle Designer – Matrix Diagrammer“, Rational RequisitePro - Use case Attribute Matrix View“, „Enterprise Architect Professional - Element Relationship Matrix“ ir kt. Pagal įrankių funkcionalumą ir taikymą analizės matricai nubraižyti, eksperimentinio tyrimo metu sukurta sistema daugiausia panašumų turi su „Oracle“ produktu „Matrix Diagrammer“. Jiems abiemis būdingos savybės:

- Abu skirti analizuoti pusiau formalias specifikacijas.
- Eilučių ir stulpelių parinkimas norint nubraižyti matricą.
- Pažymėtų eilučių ir stulpelių elementų atributų nurodymas.
- Eilučių ir stulpelių sankirtų parinkimas.
- Keletas sankirtų vaizdavimo būdų.
- Matricos braižymas.

Be abejo, egzistuoja skirtumų tarp įrankių, kurie kaip analizės formą naudoja matricą. „Oracle“ sukurtame „Matrix Diagrammer“ priemonėje renkant elementus matricai sukurti, pažymimi du objektai – eilučių ir stulpelių aibės. ODRES metodu realizuotame analizės įrankyje vartotojui leidžiama pasirinkti ne tik eilučių bei stulpelių aibes, bet taip pat jis privalo nurodyti ir jų sankirtos aibę. Papildomai, bet nebūtinai, galima pažymėti eilučių apibendrinančių eilučių aibę bei stulpelių apibendrinančių stulpelių aibę. Be to, mano sukurtoje priemonėje vartotojas pasirenka eilučių ir stulpelių sankirtos vaizdavimo formą. T.y. sankirtų celėse pateikiamus duomenis – tiesiog kryžiuokai (arba bet koks kitas simbolis), identifikuojantis ryšį tarp atitinkamų eilutės ir stulpelio objektų, arba konkrečia sankirtos objekto reikšme. Tuo tarpu „Matrix Diagrammer“ įrankyje ryšys tarp dviejų objektų matricoje žymimas tik „varnele“.

Sukūrus analizės įrankį, jį galima palyginti su pasaulyje jau egzistuojančiomis kitomis

priemonėmis. Analitinės dalies 3.5 skyriuje nustatyti įvairūs kriterijai ir jais įvertinti kiti analizės įrankiai. Dabar remiantis tais pačiais kriterijais galiu įvertinti ir savo sistemą. Tuo tikslu sukurta 17. lentelė, kurios paskutiniame stulpelyje įterpti mano kurto įrankio įvertinimai.

Tinkamiausia rezultatų pateikimo forma mano sukurtoje sistemoje – matrica. Įrankis veikia kaip atskira programa, jis nėra integruotas į ODRES metodu paremtą CASE įrankį. Sukurtoje analizės matricoje, kur išvedami rezultatai, specifikacijos objektų redagavimas, sukūrimas ar naikinimas neleidžiamas. Tam, kad išbandytume analizės įrankį, jis sukurtas ODRES metodu sudarytoms specifikacijoms analizuoti, tačiau jis nėra priklausomas nuo šio metodo. Taip pat siekta, jį jis nebūtų priklausomas nuo saugyklos tipo ir struktūros. Tuo tikslu jis realizuotas analizuoti specifikacijas, paremtas „MS SQL Server“ bei „MS Access“ pagrindu. Be to, palikta galimybė jį praplėsti bet kokiam kitam tipui. Įrankis naudoja savo lokalią „MS SQL Server“ duomenų bazę, kurioje saugo projektų duomenis. Prieš kuriant analizės matricą, būtina nurodyti pradinis elementus. Vartotojui suteikiama galimybė rezultatus matricoje filtruoti, išsisaugoti į failą ir, taip pat, įrankio rezultatų lange keisti šriftą bei šrifto dydį. Taip pat jis pasirenka vieną iš dviejų eilučių ir stulpelių sankirtų žymėjimo režimą. Yra galimybė analizuoti specifikacijas ne tik lokaliai, tačiau ir nutolusiame kompiuteryje.

17. lentelė - Analizės įrankių palyginimų lentelė

Gamintojas	Oracle Designer	Rational RequisitePro		ProVision Workbench		MagicDraw UML			Enterprise Architect	-
Įrankis	Matrix Diagrammer	Use case Attribute Matrix View	Traceability Matrix	Completeness Checker	Model Comparison	Compare Projects	Metrics	Usages / Dependent Elements	Element Relationship Matrix	Sukurta sistema
Kriterijus										
1. Rezultatų pateikimo forma	matrica	matrica	matrica	lentelė	lentelė	lentelė	lentelė	lentelė	matrica	matrica
2. Integravimas su CASE įrankiu	√	√	√	√	√	√	√	√	√	-
3. Galimybė redaguoti ir/ar kurti objektus pateiktus rezultatų formoje	√	-	-	-	√	√	-	√	√	-
4. Priklausomumas nuo metodo	√	√	√	√	√	-	-	-	√	-
5. Priklausomumas nuo saugyklos tipo	√	√	√	√	√	√	√	√	-	-
6. Priklausomumas nuo saugyklos struktūros	√	√	√	√	√	√	√	√	√	-
7. Įrankiui reikalinga lokali jo DB	√	√	√	-	-	-	-	-	-	√
8. Analizės rezultatų išsaugojimas	√	√	√	-	-	-	√	-	√	√
9. Analizės rezultatų filtravimas ir/arba rūšiavimas	√	√	√	√	√	√	√	√	-	√
10. Analizės rezultatų vaizdavimo nustatymai: šriftas, jo dydis ir tt.	√	-	-	-	-	-	-	-	-	√
11. Keli rezultatų pateikimo režimai	√	-	-	-	-	-	-	-	-	√
12. Pradinių elementų nurodymas prieš skaičiuojant/išvedant analizės rezultatus	√	√	√	√	√	√	√	√	-	√
13. Nutolusios saugyklos analizavimas	√	-	-	-	-	-	-	-	√	√



## 5.5 Sistemos taikymo rekomendacijos

ODRES reikalavimų specifikacijų saugykla yra pagrindinė, kuriai buvo kuriamas analizės įrankis. Be to, ji veikia „MS SQL Server“ pagrindu, kaip ir pati sistemos duomenų bazė. Kadangi saugykla saugoma Informacijos sistemų katedros serveryje, gana patogiu būdu importuoti jos kopiją į tą patį kompiuterį, kuriame įdiegta programa. Darbas lokaliai spartesnis ir nereikia atlikti papildomų tinklo konfigūravimų.

Projektavimo metu buvo iškeltas papildomas reikalavimas sistemai – nepriklausomumas nuo reikalavimų specifikacijų saugyklos tipo. Tuo tikslu sistema papildyta analizuoti specifikacijas, veikiančias ir „MS Access“ pagrindu. Šio saugyklos tipo atveju nereikia nieko importuoti į duomenų bazių valdymo sistemą. Pakanka į kompiuterį, kuriame įdiegta programa, įsirašyti .MDB failą.

Nubraižius analizės matricą, rekomenduojama ją eksportuoti į .XLS failą. Tuomet atsidarius šį failą su „MS Excel“, galima atlikti bet kokius redagavimus ir atsispausdinti.

## **5.6 Tolimesnio sistemos tobulinimo, plėtojimo galimybės**

Pagrindinis sistemos praplėtimo tikslas – būti nepriklausomu nuo reikalavimų specifikacijų saugyklos tipo. Jam pasiekti palikta galimybė įrankį papildyti. Prisijungimas prie RS saugyklos ir sisteminių duomenų nuskaitymas yra svarbiausios sistemos dalys, kurias reiktų sukurti, norint realizuoti papildymą. Gali būti taip, jog kai kurios DBVS, pvz. „MS Access“, neturi visų sisteminių duomenų, reikalingų analizės įrankiui. Todėl galima pasinaudoti ADO objektais, kurie naudojami sistemoje.

## 6. Išvados

Kuriamai sistemai modeliuoti panaudotas RUP (Rational Unified Process) projektavimo procesas bei vizualinio modeliavimo CASE įrankis „MagicDraw“. Darbe išskirti du pagrindiniai modeliai:

1. Reikalavimų modelis. Jame pateikiamas sistemos kontekstas ir sukurti panaudojimo atvejai. Jiems specifikuoti panaudotos sekų ir veiklos diagramos bei tekstinės lentelės, kuriose apibrėžtos „prieš“ ir „po“ sąlygos, sužadinantys įvykiai, ryšiai su kitais panaudojimo atvejais, pagrindiniai ir alternatyvūs įvykių srautai. Sukurta dalykinės srities klasių diagrama. Vartotojo interfeiso modelyje pateikti pagrindiniai sistemos langai ir navigavimo planas. Galiausiai nustatyti reikalavimai sistemos funkcionalumui.
2. Projekto modelis. Kiekvienam panaudojimo atvejui nubraižyta panaudojimo atvejų realizacijos diagrama. Prieduose pateiktos kiekvieną realizaciją realizuojančios klasės. Sukurti sistemos statinės struktūros bei elgsenos modeliai. Prieduose papildomai pateiktos ir nuodugniai aprašytos ribinių klasių, valdiklių ir esybių modeliai. Sukurtas sistemos duomenų bazės schemas ir realizacijos (komponentų ir diegimų) modeliai. Taip pat nustatyti reikalavimai sistemos funkcionavimo palaikymui.

Analizės įrankis išbandytas analizuoti ODRES metodu sudarytoms specifikacijoms, saugomoms „MS SQL Server“ tipo saugykloje. Be to, sistema praplėsta analizuoti saugyklas, veikiančias „MS Access“ pagrindu. Įrankis nėra priklausomas nuo ODRES metodo saugyklos struktūros, todėl juo galima analizuoti bet kokias kitas specifikacijas, saugomas minėtų tipų saugyklose. Taip pat paliktos praplėtimo galimybės, kad būtų galima atlikti analizę su bet kokio kito tipo saugyklomis.

Kaip analizės forma panaudota matrica, kurios sankirtos celės identifikuoja ryšius tarp objektų, pateiktų eilutėse ir stulpeliuose. Analizės įrankiui realizuoti panaudota „Borland C++ Builder“ programavimo aplinka. Duomenų bazė sukurta „MS SQL Server“ pagrindu.

Užbaigus reikalavimų specifikacijų analizės įrankį, nustatyta:

- Bet kurią reikalavimų specifikaciją gali analizuoti žmogus. Tačiau esant dideliame sistemos modeliui tai gali būti ypač ilgas ir sudėtingas darbas, kurio rezultatai priklausys nuo specifikaciją analizuojančio darbuotojo kompetencijos ir patirties. Pasiūlyta sistema – sprendimas spręsti tokio pobūdžio problemas

- Ieškant ryšių tarp ODRES metodu sudarytos specifikacijos objektų tikslingiausia įrankį naudoti esant dideliems modeliams, kuomet sudėtinga atrasti ryšius tarp objektų rankiniu būdu.
- Kuo analizės matrica paprastesnė ir mažiau perkrauta duomenimis, tuo ji patogesne analizuoti.

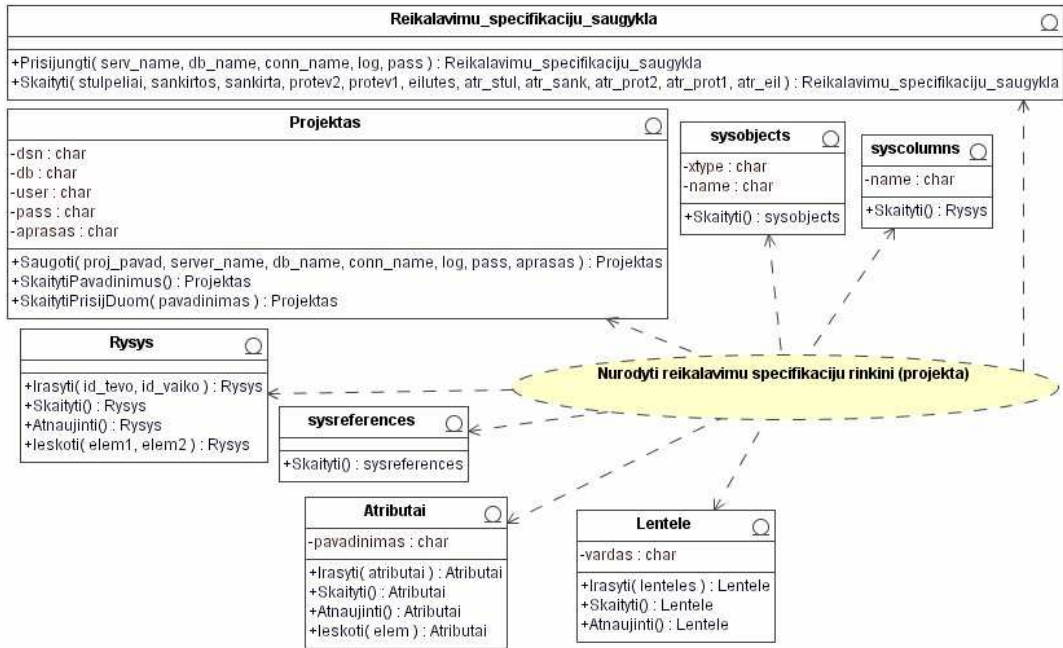
## 7. Literatūra

1. Smitha S., Laia L. ir Khedria R. Requirements analysis for engineering computation. Computing and Software Department, McMaster University, 2004: 5 p.
2. Butkienė R., Butleris R. The Approach for User Requirements Specification. Department of Information Systems, Kaunas University of Technology, 2001: pp. 1-3
3. Bubenko J.A. Wangler B. Research directions in conceptual specification development. SYSLAB, Dept. of Computer and Systems Science and SISU – the Swedish Institute for System Development, 1991: pp. 1-5.
4. Tse T.H., Pong L. An Examination of Requirements Specification Languages. Department of Computer Science, The University of Hong Kong: 2003, pp. 1-4.
5. Bahill A.T., Henderson S.J.: Requirements Development, Verification, and Validation Exhibited in Famous Failures, 2005
6. Lobo L.O. Analysis and Evaluation of Methods for Activities in the Expanded Requirements Generation Model (x-RGM). Blacksburg, Virginia, 2004: 24 p.
7. Karl E. Wieggers. Automating Requirements Management. Process Impact.
8. Saaman E.H. Another formal Specification language. IPA Dissertation Series, 2000: 3 p.
9. Donn Le Vie. Writing Software Requirements Specifications. The Official TECHWR-L Web Site. Prieiga per internetą [žiūrėta 2006 01 20]:  
<http://www.techwr-l.com/techwhirl/magazine/writing/softwarerequirementspecs.html>
10. Requirements specification, chapter 10. Extract from: STARTS Purchasers' Handbook Second Edition, 1989: 18 p..
11. Bubenko J.A. Extending the Scope of Information Modelling. Department of Computer and Systems Science, Royal Institute of Technology and Swedish Institute for Systems Development, 1993: pp. 2-6, 19.
12. Smith M. Project Management Planning - Top-Level Requirements Specification, 1997.
13. Rossel P., Contreras R., Cecilia B.M. Graphic specification of abstract data types. Revista Facultad de Ingeniería, Chile Vol.12 N°1, 2004: pp. 15-16.
14. Lamsweerde A. Formal Specification: a Roadmap. Département d'Ingénierie Informatique, Université catholique de Louvain, B-1348 Louvain-la-Neuve (Belgium), 2000.
15. Fernandes R., Cowie A.J. Capturing Informal Requirements as Formal Models. AWRE - Australian Workshop on Requirements Engineering, 2004.
16. Hallock P. Natural Language Modeling. InConcept (Information Conceptual Modeling, Inc.). Prieiga per internetą [žiūrėta 2006 01 21]: <http://www.inconcept.com/JCM/February2001/hallocktext.html>
17. Oracle Corporation oficiali svetainė [interaktyvi]. Prieiga per internetą [žiūrėta 2006 04 01]: <http://www.oracle.com>
18. IBM Rational Software oficiali svetainė [interaktyvi]. Prieiga per internetą [žiūrėta 2006 04 01]: <http://www.rational.com>
19. Proforma Corporation oficiali svetainė [interaktyvi]. Prieiga per internetą [žiūrėta 2006 04 01]: <http://www.proformacorp.com>
20. No Magic, Inc. oficiali svetainė [interaktyvi]. Prieiga per internetą [žiūrėta 2006 04 01]: <http://www.magicdraw.com>
21. Sparx Systems oficiali svetainė [interaktyvi]. Prieiga per internetą [žiūrėta 2006 04 01]: <http://www.sparxsystems.com>
22. Asmeninė Mariaus Gedmino svetainė. Prieiga per internetą [žiūrėta 2006 05 16]:  
<http://mg.b4net.lt/>
23. SQL Server Code,Tips and Tricks, Performance Tuning - [interaktyvi]. Prieiga per internetą [žiūrėta 2006 05 29]:  
<http://sqlservercode.blogspot.com>
24. Prentice Hall oficiali svetainė [interaktyvi]. Prieiga per internetą [žiūrėta 2006 05 29]:  
<http://vig.prenhall.com>

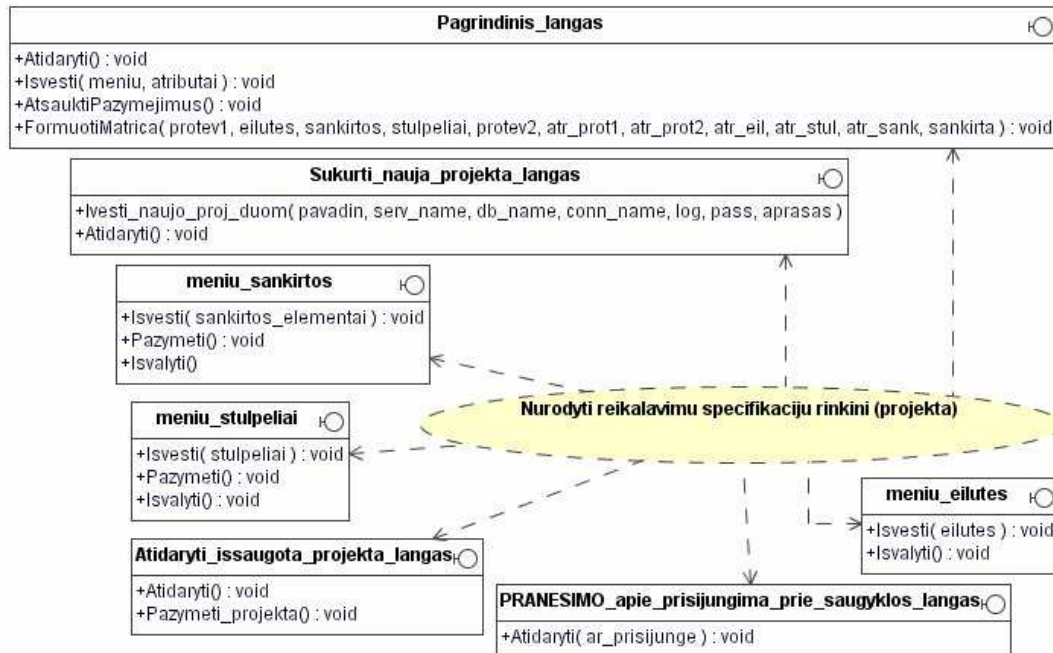
25. Brainbench oficiali svetainė [interaktyvi]. Prieiga per internetą [žiūrėta 2006 05 29]:  
<http://www.brainbench.com>
26. Agile Modeling (AM) oficiali svetainė [interaktyvi]. Prieiga per internetą [žiūrėta 2006 05 29]:  
<http://www.agilemodeling.com>
27. Sparx Systems oficiali svetainė [interaktyvi]. Prieiga per internetą [žiūrėta 2006 05 29]:  
<http://www.sparxsystems.com>
28. Butkienė R. Informacijos šaltų specifikacijos metaduomenų bazės objektų aprašymas.

## **8. Priedai**

## 8.1 Panaudojimo atvejų realizuojamos klasės

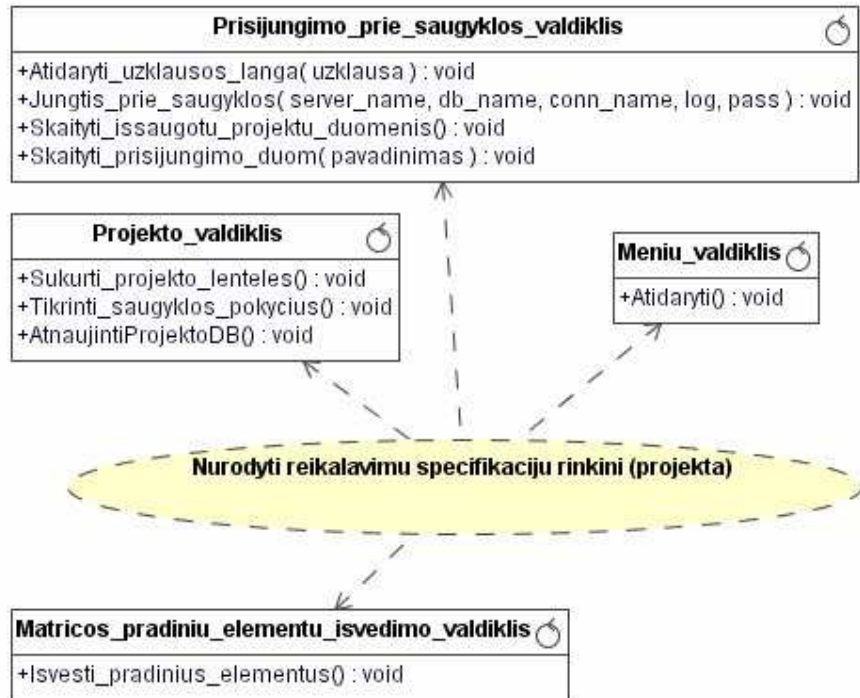


63 pav. Panaudojimo atvejo "Nurodyti reikalavimų specifikacijų rinkinį (projektą)" realizuojamos esybių klasės

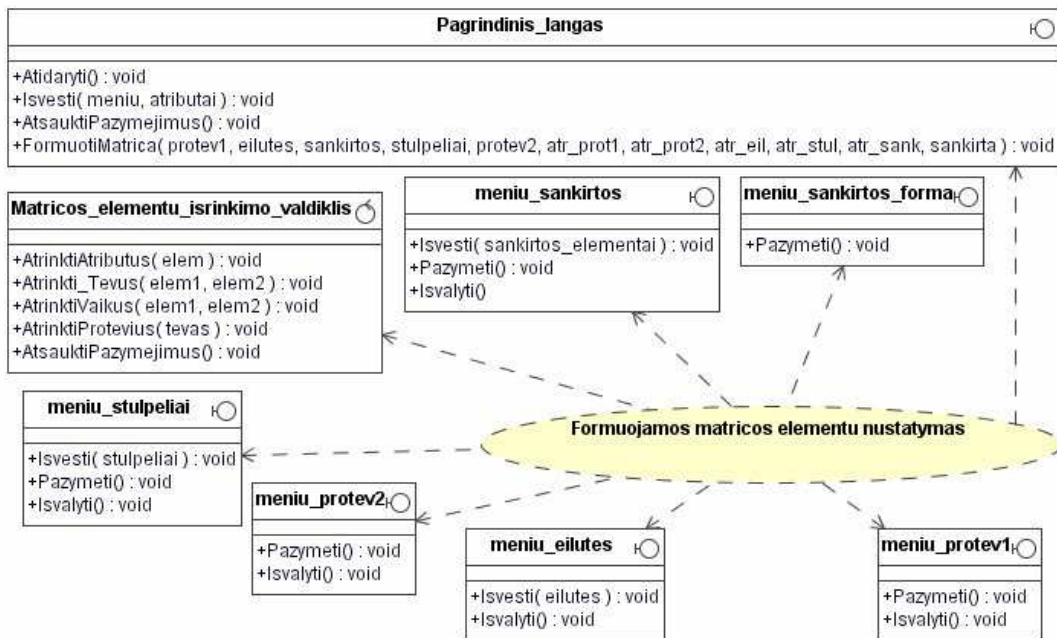


64 pav. Panaudojimo atvejo "Nurodyti reikalavimų specifikacijų rinkinį (projektą)" realizuojamos ribinės klasės

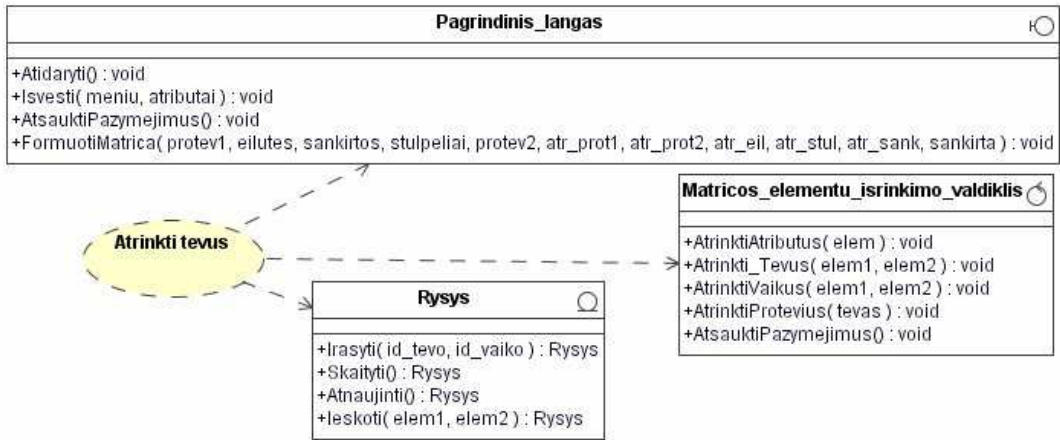




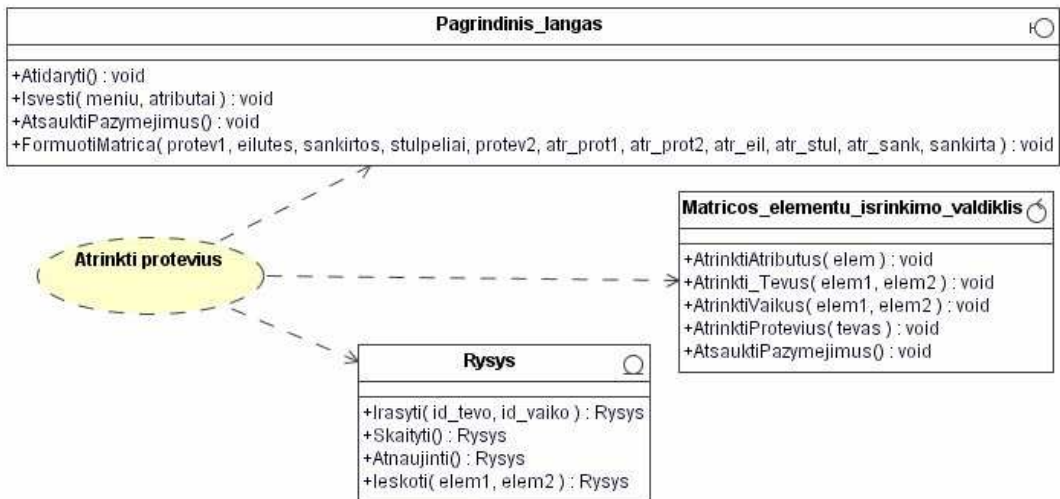
65 pav. Panaudojimo atvejo "Nurodyti reikalavimu specifikacijų rinkinį (projektą)" realizuojamos valdiklių klasės



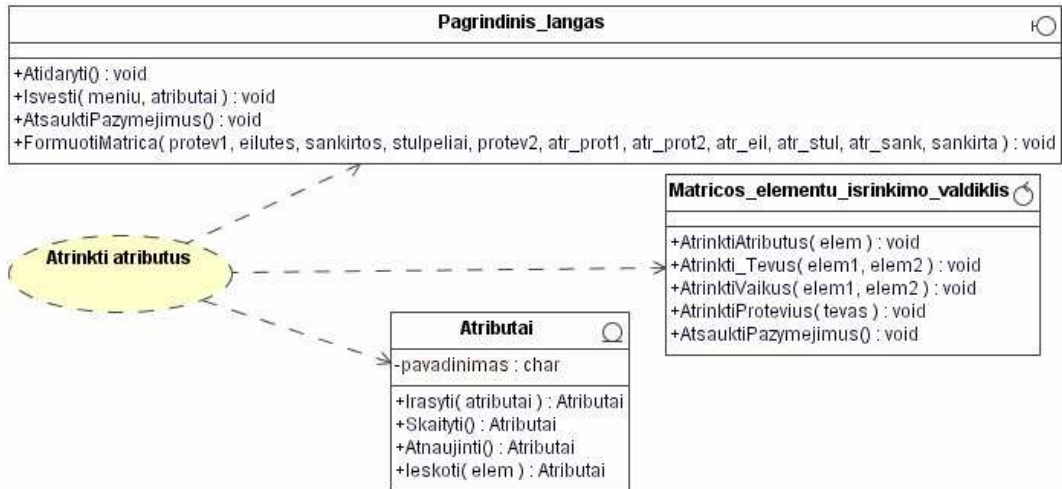
66 pav. Panaudojimo atvejo "Formuojamos matricos elementų nustatymas" realizacijos diagrama



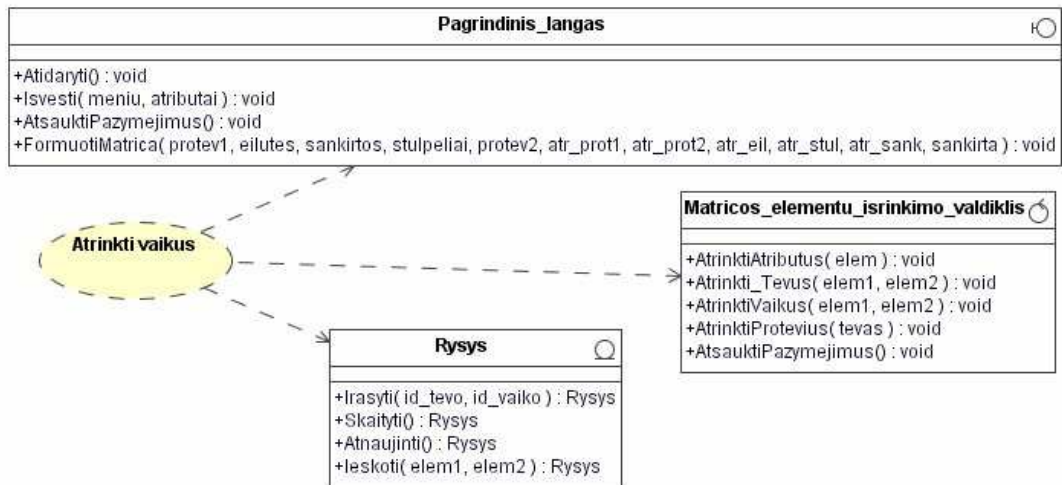
67 pav. Panaudojimo atvejo "Atrinkti tėvus" realizacijos diagrama



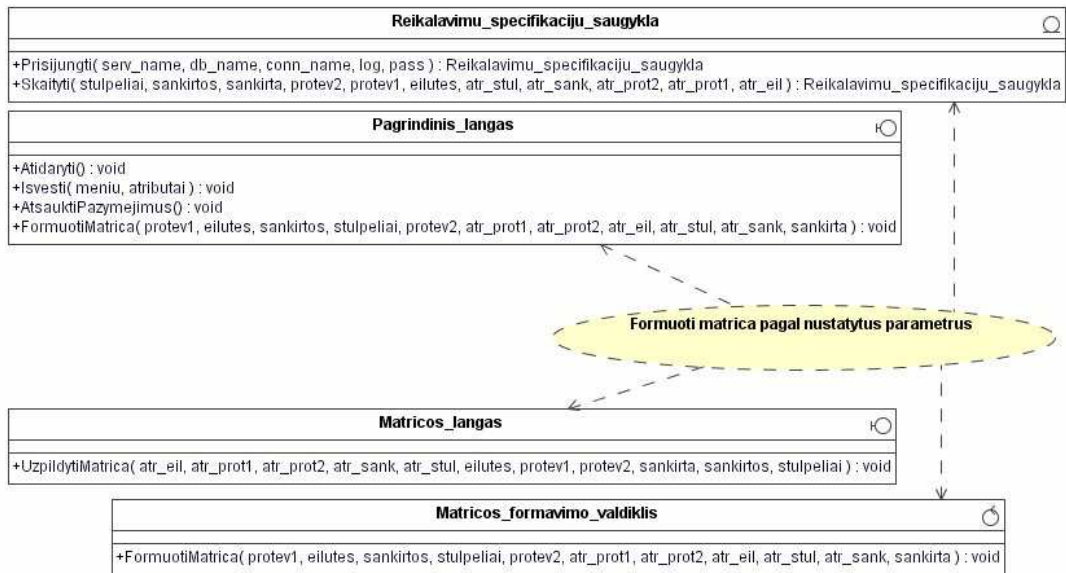
68 pav. Panaudojimo atvejo "Atrinkti protevius" realizacijos diagrama



69 pav. Panaudojimo atvejo "Atrinkti atributus" realizacijos diagrama

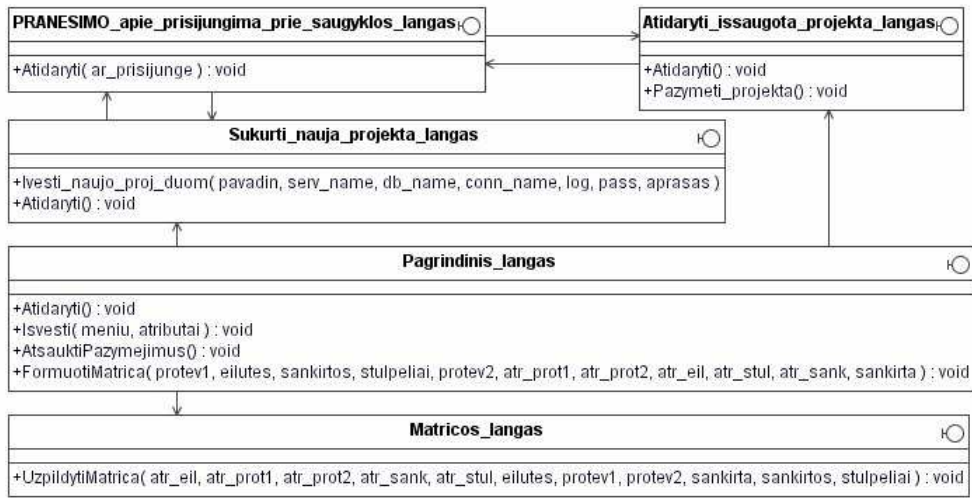


70 pav. Panaudojimo atvejo "Atrinkti vaikus" realizacijos diagrama

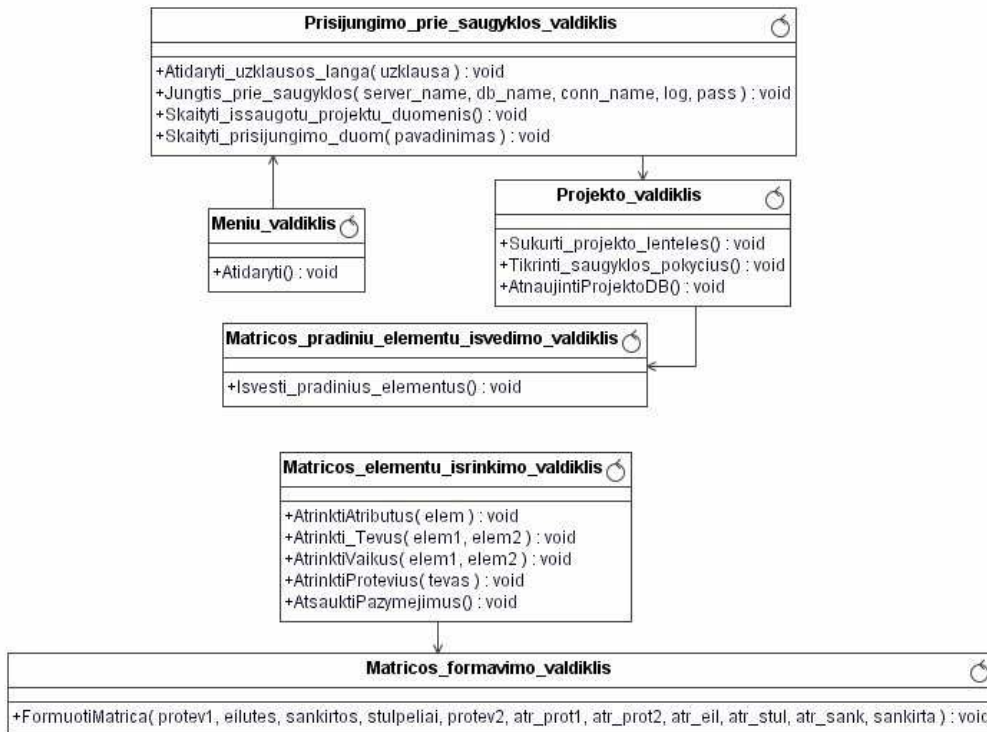


71 pav. Panaudojimo atvejo "Formuoti matrica pagal nustatytus parametrus" realizacijos diagrama

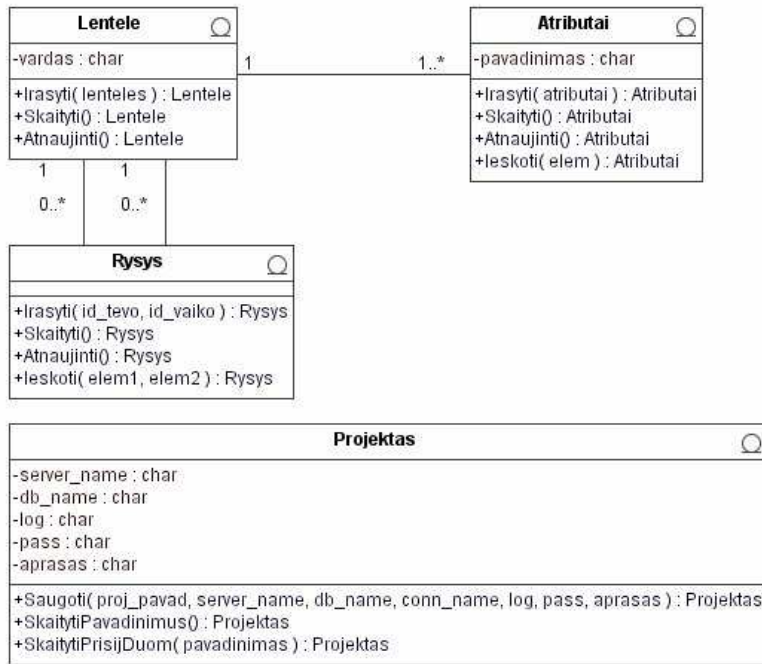
## 8.2 Statinės architektūros modelio klasių diagramos



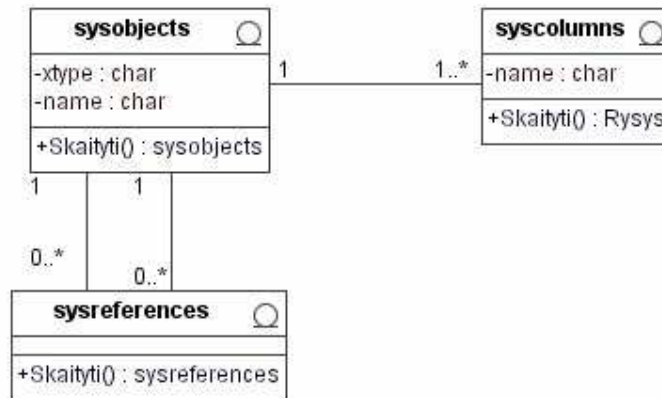
72 pav. Sistemos vartotojo paslaugas (interfeisas)



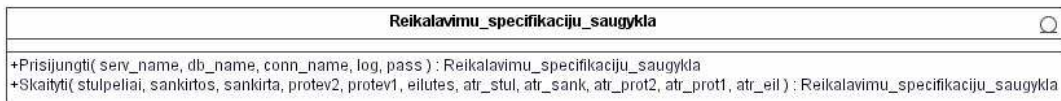
73 pav. Sistemos veiklos paslaugas (logika)



74 pav. Sistemos duomenų paslaugos – sistemos duomenų bazė



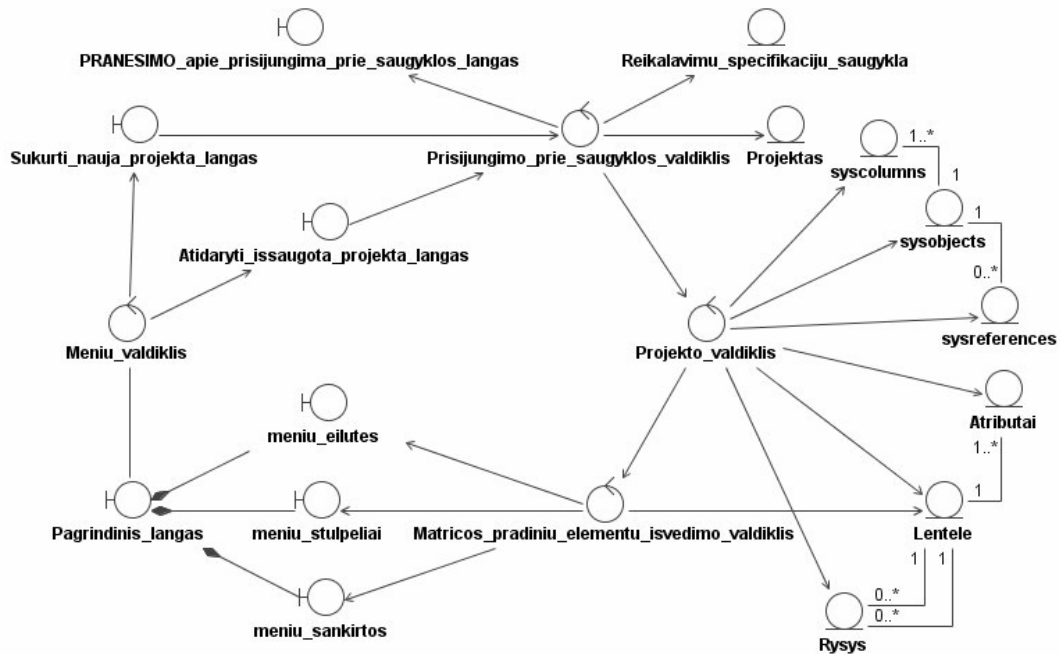
75 pav. Sistemos duomenų paslaugos – saugyklos DBVS sisteminės lentelės



76 pav. Sistemos duomenų paslaugos – reikalavimų specifikacijų saugykla

### 8.3 Ribinių klasių, valdiklių ir esybių modelis

Sudarytos analizės klasių diagramos rodo, kokios klasės realizuoja kokius panaudojimo atvejus. Analizuojant panaudojimo atvejų veiksmų žingsnius, kiekvienam iš jų sukurti vartotojo interfeisai ir valdikliai.



77 pav. Ribinių klasių, valdiklių ir esybių diagrama panaudojimo atvejui „Nurodyti reikalavimų specifikacijų rinkinį (projektą)“

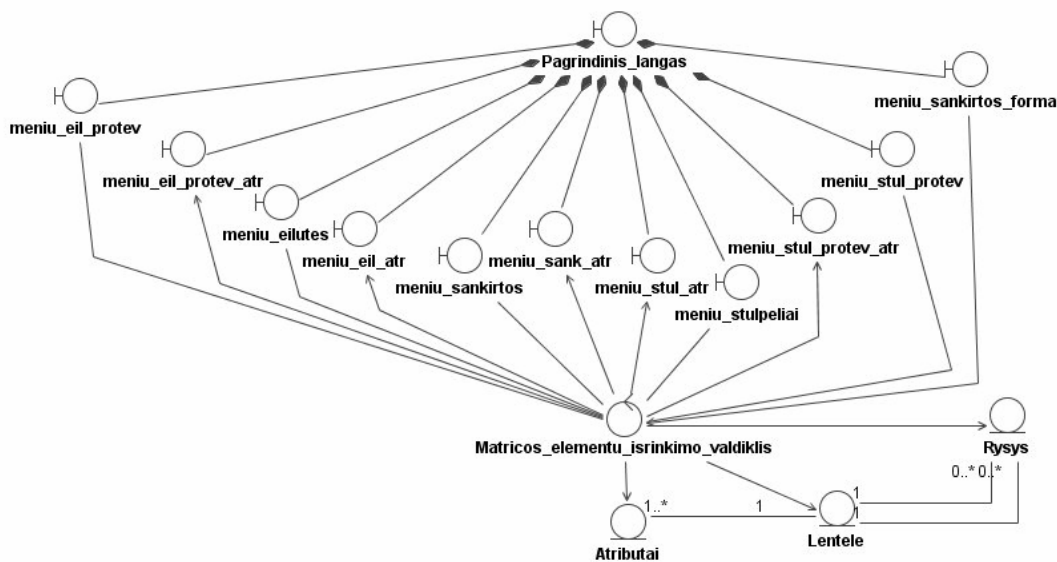
77 paveiksle esanti ribinių klasių, valdiklių ir esybių diagrama, vaizduoja klases, kurios realizuoja panaudojimo atvejį „Nurodyti reikalavimų specifikacijų rinkinį (projektą)“. Prieš nustatant elementus matricai formuoti, pirmiausia reikia nurodyti norimą saugyklą, kurioje saugomos sudarytos specifikacijos. Tai galima padaryti dviem būdais. Pirmasis – žinant prisijungimo prie saugyklos duomenis, juos suvesti rankiniu būdu. Antrasis būdas – pasirinkti iš sąrašo anksčiau išsaugoto projekto duomenis ir juos nurodyti, kad sistema prisijungtų prie reikalavimų specifikacijų saugyklos. 77 paveiksle vaizduojamų analizės klasių logika būtų tokia:

- Pagrindiniame lange vartotojas per meniu turi pasirinkti prisijungimo prie saugyklos būdą: 1. suvesti prisijungimo duomenis ranka; 2. iš sąrašo pasirinkti išsaugoto projekto duomenis. Kiekvienu atveju atidaromas skirtingas langas.
- Pirmuoju atveju prisijungimo valdiklis nuskaito vartotojo suvestus prisijungimo duomenis. Taip pat patikrina ar pažymėta varnelė, nurodanti kad išsaugoti suvestus duomenis. Jeigu pažymėta, lentelėje „Projektas“ juos išsaugo. Antruoju atveju – iš



lentelės „Projektas“ nuskaito reikiamus duomenis ir jungiasi prie saugyklos. Apie prisijungimą valdiklis vartotoją informuoja atidarydamas pranešimo langą.

- Prisijungimo valdiklis kreipiasi į projekto valdiklį. Jeigu prisijungimo duomenys buvo suvedinėjami ranka, tuomet jis tiesiog iš reikalavimų specifikacijų saugyklos DBVS sisteminių lentelių nuskaito duomenis apie saugykloje saugomas lenteles, lentelių atributus ir ryšius, ir jais užpildo sistemos duomenų bazę. Jeigu buvo pasirinkti jau išsaugoti prisijungimo duomenys, tuomet projekto valdiklis tikrina duomenų, esančių sistemos duomenų bazėje, nesutapimus su duomenimis, saugomais saugyklos DBVS sistemines lentelėse. Jeigu yra nesutapimų, tuomet duomenis iš saugyklos perrašo į sistemos DB.
- Projekto valdiklis kreipiasi į matricos pradinių elementų valdiklį. Šis kreipiasi į sistemos duomenų bazę ir į tris pagrindinio lango sąrašo elementus – eilučių, stulpelių ir sankirtų meniu – išveda pradinius elementus matricos formavimo nustatymui.



78 pav. Ribinių klasių, valdiklių ir esybių diagrama panaudojimo atvejui „Formuojamos matricos elementų nustatymas“

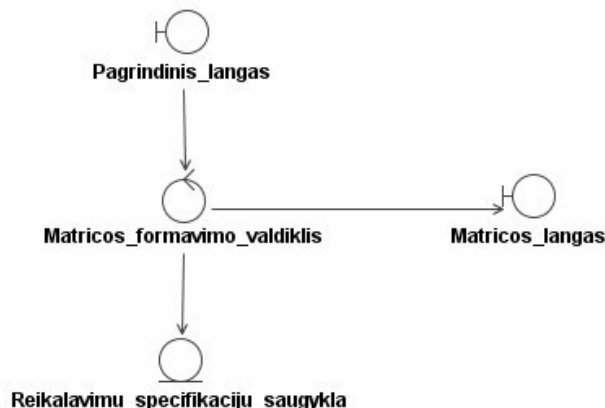
78 paveiksle esanti ribinių klasių, valdiklių ir esybių diagrama, vaizduoja klases, kurios realizuoja panaudojimo atvejį „Formuojamos matricos elementų nustatymas“. Ši diagrama taip pat apima praplečiančius panaudojimo atvejus: „Atrinkti tėvus“, „Atrinkti protėvius“, „Atrinkti vaikus“, „Atrinkti atributus“.

Prieš sistemai suformuojant matricą, vartotojas turi nurodyti eilučių, stulpelių, eilučių ir stulpelių sankirtų elementus, taip pat apibendrinančių eilučių bei stulpelių elementus, kiekvieno nurodyto elemento atributus ir sankirtos formą. Pagrindinis langas susideda iš



matricos elementų sąrašų meniu, kuriuose vartotojas ir atlieka pažymėjimus. 78 paveikslo analizės logika tokia:

- Sistemai prisijungus prie reikalavimų specifikacijų saugyklos, vartotojas pasirinktinai gali pradėti žymėjimus eilučių, stulpelių ir sankirtų sąrašuose.
- Matricos elementų išrinkimo valdiklis elgiasi skirtingai, priklausomai nuo to, kokį elementą vartotojas pažymi. Pažymėjus eilutę, stulpelį, sankirtą ar bet kurią iš apibendrinančių elementų, valdiklis pirmiausia atrenka atributus ir atitinkamame atributų sąrašo meniu juos išveda. Atributams išrinkti valdiklis kreipiasi į esybę „Atributai“.
- Vartotojui pirmiausiai pažymėjus, tarkim, eilutę, matricos elementų išrinkimo valdiklis kreipiasi į esybes „Ryšys“ bei „Lentelė“ ir pagrindiniame lange atitinkamo sąrašo meniu išveda su pažymėta eilute besisiejančius stulpelius bei sankirtas. Toliau vartotojui pažymėjus, tarkim, sankirtą, valdiklis atrenka jai atributus, juos išveda. Tada atrenka jau anksčiau nurodytai eilutei ir pažymėtai sankirtai priklausančius stulpelius ir juos išveda. Panašiu principu valdiklis elgiasi žymėjimus pradėdant ir nuo kitų elementų.
- Tiesa, vartotojui pažymint, tarkim, eilutę, valdiklis kreipiasi į esybes „Ryšys“ bei „Lentelė“ ir atrenka su šia eilute besisiejančias apibendrinančias eilutes. Panašiu principu valdiklis veikia ir pažymėjus stulpelį.
- Atšaukiant padarytus pažymėjimus ar elementus nurodant iš naujo, valdiklis į tai reaguoja, kreipiasi į esybes „Ryšys“ bei „Lentelė“ ir išveda reikiamus elementus į sąrašų meniu.



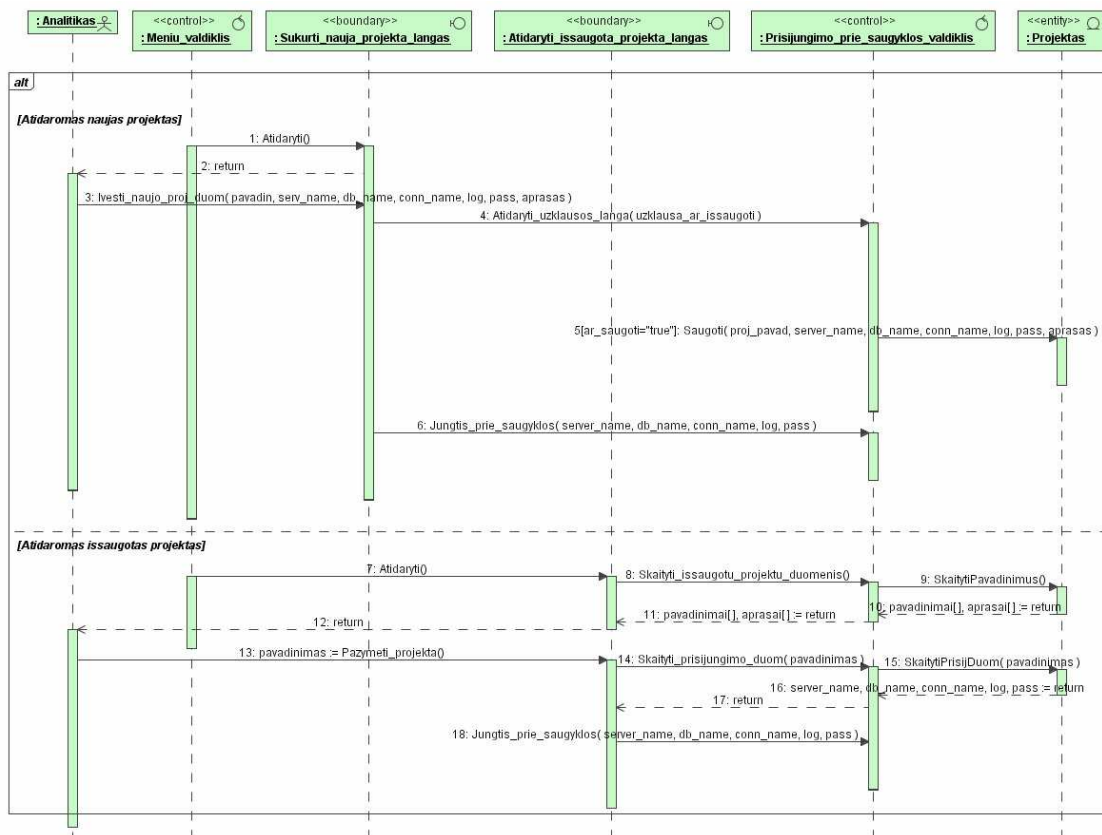
79 pav. Ribinių klasių, valdiklių ir esybių diagrama panaudojimo atvejui „Formuoti matricą pagal nustatytus parametrus“

Nurodžius visus reikiamus elementus, sistema pasiruošusi formuoti matricą. 79 paveikslas realizuoja paskutinįjį panaudojimo atvejį „Formuoti matricą pagal nustatytus

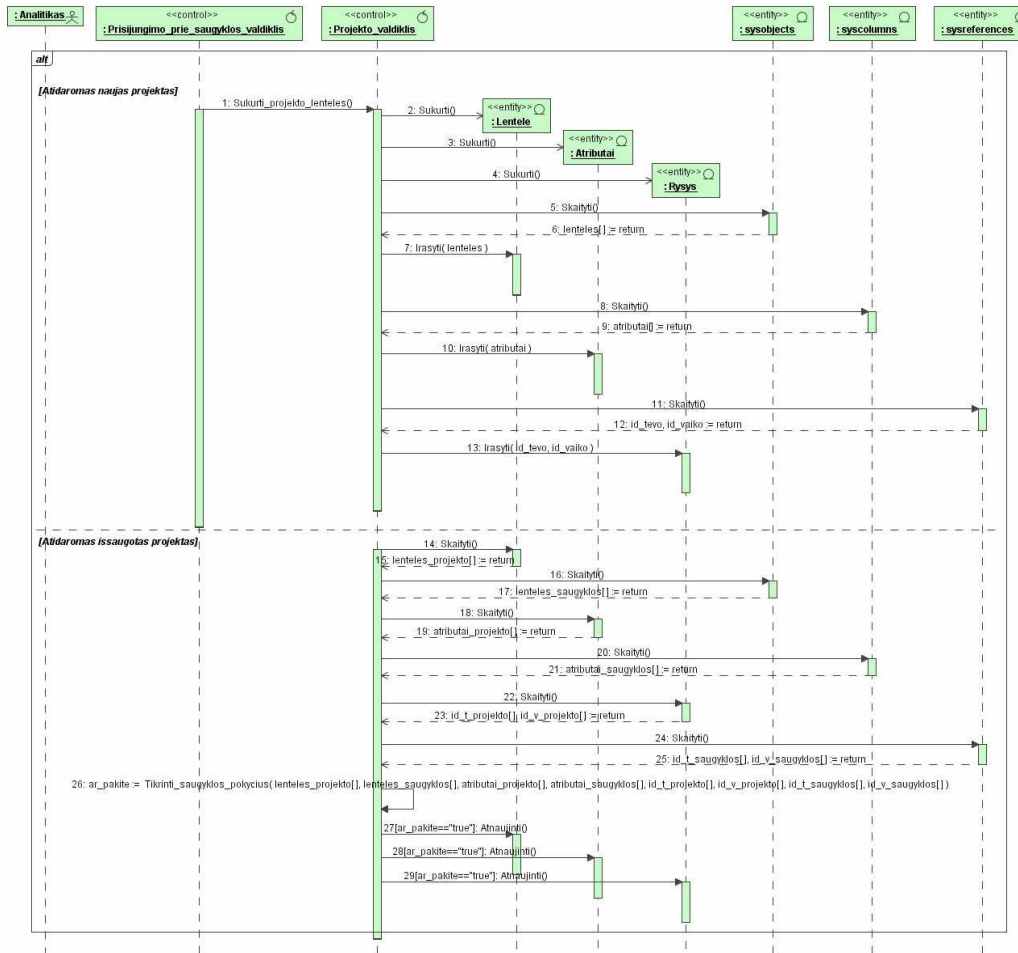
parametrus. Pagrindiniame lange vartotojas paspaudžia matricos formavimo mygtuką. Pagrindinis langas kreipiasi į matricos formavimo valdiklį. Pastarasis gavęs visus reikalingus parametrus matricai formuoti, kreipiasi į reikalavimų specifikacijų saugyklą, ištraukia nurodytas reikšmes ir jomis užpildo matricą, kurią pateikia matricos lange.

## 8.4 Sistemos elgsenos modelio papildomos diagramos

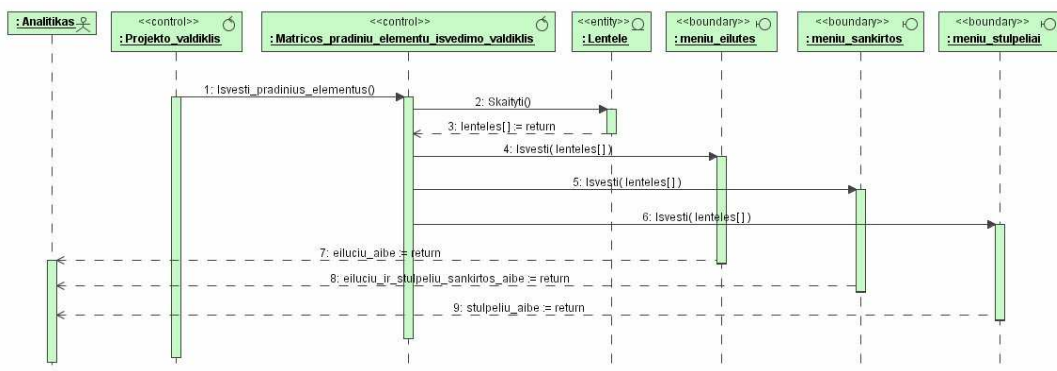
Sekų diagramos įeinančios į panaudojimo atvejo „Nurodyti reikalavimų specifikacijų rinkinį (projektą)“ sekų diagramą:



80 pav. Sekų diagrama (REF nuoroda) „Nurodyti projektą“

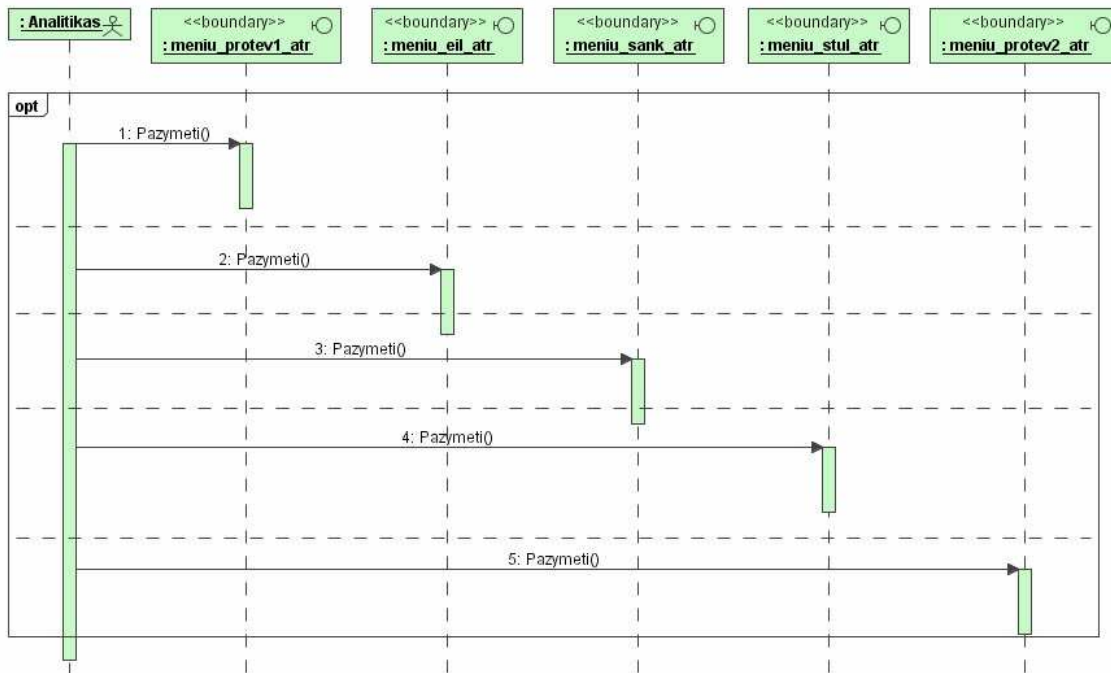


81 pav. Sekų diagrama (REF nuoroda) „Atidaryti projektą“



82 pav. Sekų diagrama (REF nuoroda) „Išvesti pradinius elementus matricai formuoti“

Sekų diagrama, susieta su panaudojimo atveju „Formuojamos matricos elementu nustatymas“ realizacija:



83 pav. Sekų diagrama (REF nuoroda) „Pažymėti atributus“

## 8.5 Sistemos papildymas nauju saugyklos tipu

Šiame skyriuje pateikti paaiškinimai programos kodo redagavimui/papildymui, norint jos funkcionalumą praplėsti nauju saugyklos tipu.

Pirmas žingsnis norint papildyta programą, kad ji analizuotų naujo tipo saugyklą, reikia šeštoje formoje į „RadioGroup“ elementą įterpti naują „RadioButton“ tipo mygtuką. Jam automatiškai suteikiamas indeksas 2. Indeksas 0 skirtas „MS SQL Server‘ui“, o indeksas 1 – „MS Access‘ui“.

Žemiau esančioje lentelėje išvardinti veiksmai norint papildyti programos funkcionalumą, kad ji analizuotų dar vieno tipo RS saugyklą. Tokių papildymų galima atlikti kiek tik norime. Tuomet veiksmai būtų analogiški pateiktiems žemiau. Lentelėje esančiuose paaiškinimuose, kur pateikiama kodo ištrauka, tereikia nutrinti komentarus ir redaguoti kodą. Plečiant programos funkcionalumą kitomis tipo RS saugyklomis, analogiški veiksmai atliekami su tuo pačiu pateiktu kodu. Lentelės kairiajame stulpelyje naudojami laužtiniuose skliaustuose pateikiami numeriai, kurie nurodo vietą atitinkamame „cpp“ faile, kur reikalinga redaguoti kodą.

18. lentelė - „Unit6.cpp“ failo redagavimo nuorodos.

<p><b>[#1]</b></p>	<p>Metodas <code>edit_uzpildymas()</code>, atitinkamai pagal nustatytą saugyklos tipą, nustato mygtuko "Jungtis" „enable/disable“ reikšmę, atsižvelgiant į tai, jog užpildyti/neužpildyti privalomi „Edit“ tipo laukai. Skirtingam saugyklos tipui privalomi laukai gali skirtis.</p> <p>Pirmiausia ties [#1] žyme reikalinga nutrinti komentarus. Kaip jau minėta, įterpus į „RadioGroup“ objektą naujo RS saugyklos tipo „radiobutt“ tipo mygtuką, jam suteikiamas indeksas 2.</p> <pre> case 2:     if((CheckBox1-&gt;Checked==true) &amp;&amp; (Edit1-&gt;Text!="") &amp;&amp; &lt;kiti privalomi Editxx laukai&gt;))         Button1-&gt;Enabled=true;     else if ((CheckBox1-&gt;Checked==false) &amp;&amp; &lt;kiti privalomi Editxx laukai&gt;))         Button1-&gt;Enabled=true;     else         Button1-&gt;Enabled=false;     break; </pre> <p>Vietoj „&lt;kiti privalomi Editxx laukai&gt;“ išvardijami visi „Edit“ tipo laukai, kuriuos būtina užpildyti nustačius naują saugyklos tipą. Pvz. jei tai „Edit2“ ir „Edit3“, tai vietoj „&lt;kiti privalomi Editxx laukai&gt;“ reiks įrašyti:</p> <pre> (Edit2-&gt;Text!="") &amp;&amp; (Edit3-&gt;Text!="") </pre>
<p><b>[#2]</b></p>	<p>Esant skirtingam saugyklos tipui, vartotojo gali būti pareikalauti įvesti skirtingus</p>

	<p>prisijungimo duomenis. Todėl turi būti pakoreguota sąsaja. Pvz. „SQL Serverio“ atveju yra reikalingas slaptažodis. Tuo tarpu Access saugyklos tipas nėra būtinas.</p> <p>Ties [#2] žyme ištrinami komentarų simboliai. Toliau reikės sukurti metodą „nustatyti_iterfeisa_&lt;RS saugyklos tipas&gt;()“, kuris tokiu būdu sukonfigūruoja „Edit“ tipo laukus.</p>
<b>[#2a]</b>	<p>Ties žyme [#2a] turi būti sukurta metodas „nustatyti_iterfeisa_&lt;naujas RS saugyklos tipas&gt;()“. Jis analogiškas kitiems dviems: „nustatyti_iterfeisa_mssqlserver()“ ir „nustatyti_iterfeisa_msaccess()“.</p> <p>Kuriant šį metodą rekomenduojama naudotis aprašymu 19. lentelėje. Joje parodyti kokios sąsajos objektai redaguojami ir kokios turi būti objektų reikšmės kiekvienos saugyklos atveju. Papildant programos funkcionalumą nauju saugyklos tipu, ši lentelė papildytų nauju stulpeliu analogišku „MS SQL Server“ ar „MS Access“.</p> <p>Prireikus naujos saugyklos atveju redaguoti kokią nors objekto reikšmę, tačiau šiam objektui neesant 19. lentelėje, pastaroji lentelė papildoma nauja eilute, skirta papildomam objektui. Tuomet teks aprašyti šio objekto reikšmę visuose metoduose: „nustatyti_iterfeisa_mssqlserver()“, „nustatyti_iterfeisa_msaccess()“ ir „nustatyti_iterfeisa_&lt;naujas RS saugyklos tipas&gt;()“.</p> <p>Nepamiršti deklaruoti naujai sukurtą metodą „nustatyti_iterfeisa_&lt;naujas RS saugyklos tipas&gt;()“ faile „Unit6.h“</p>
<b>[#3]</b>	<p>Vienas iš reikalavimų yra tai, kad programa neleistų sukurti dviejų projektų vienai ir tai pačiai RS saugyklos tipui. Todėl reikia tikrinti ne tik tai, ar nėra jau išsaugoto projekto tokiu pačiu vardu, bet ir kitus būtinus laukus. Pvz. SQL Server atveju įvedant prisijungimo duomenis būtina patikrinti, kad saugykloje nebūtų išsaugotas projektas, turintis tą patį serverio vardą, vartotojo vardą ir duomenų bazę. T.y., kad būtų uždrausta kurti projektą, kuris jau išsaugotas įrankio duomenų bazėje. Access saugyklos tipo atveju reikia patikrinti, kad nebūtų projekto turinčio tą patį duomenų šaltinį. Tam kiekvienos saugyklos tipo atveju kuriamas naujas metodas:</p> <p>„SQL Server“ – „tikrinti_ar_egzist_mssqlserver()“. Tikrina, ar nėra išsaugoto projekto su tokiu pačiu serverio vardu, vartotojo vardu ir duomenų bazės vardu.</p> <p>„Access“ – „tikrinti_ar_egzist_access()“. Tikrina, ar nėra išsaugoto projekto su tokiu pačiu duomenų šaltiniu.</p> <p>Analogišką metodą „tikrinti_ar_egzist_xx()“ reikia sukurti naujam RS saugyklos tipui „xx“. Pirmiausiai nutrinami komentarų simboliai – šioje vietoje metodas bus iškviečiamas.</p>
<b>[#3a]</b>	<p>Metodo „tikrinti_ar_egzist_&lt;naujas RS saugyklos tipas&gt;(&lt;sarasas butinu patikrinti Edit reiksmiu&gt;)“ kodas. Taip pat nepamiršti šio metodo deklaruoti „Unit6.h“ faile.</p>

19. lentelė - Sąsajos objektų savybės

Objekto savybė  Objektas	MS SQL Server: nustatyti_iterfeisa_mssqlserver()			MS Access nustatyti_iterfeisa_msaccess()		
	Visible	PasswordChar	Caption	Visible	PasswordChar	Caption
Label2			Serveris:			Vartotojo vardas:
Label3			Vartotojo vardas:			Slaptažodis:
Label4			Slaptažodis:			Duomenų šaltinis:
Label5			Duomenų bazė			MDB slaptažodis:
Label10	TRUE			TRUE		
Label11	TRUE			FALSE		
Label12	TRUE			FALSE		
Label13	TRUE			TRUE		
Label14	TRUE			FALSE		
Edit3		NULL		"*"		
Edit4		"*"		NULL		
Edit5		NULL		"*"		
SpeedButton1	FALSE			TRUE		

Pažymėjimai:

- Tuščia celė – objekto savybė egzistuojama, bet programiškai ji nenustatinėjama. Jos reikšmė pagal nutylėjimą nustatyta „Object Inspector“ iuje“;
- Perbraukta celė - objekto savybė neegzistuoja.

Bet kokiam saugyklos tipui gali būti atliekami tam tikri nustatymai. Bendriausiu nustatymu laikomas prisijungimo prie saugyklos būdas. Jam nurodyti skirta „Form8“. Taip pat gali būti kitų nustatymų prisijungiant prie RS saugyklos, kaip pvz. porto numeris ar kt. Realizuojant programoje naują saugyklos tipą ir norint vartotojui leisti atlikti panašius nustatymus, turi būti atlikti papildymai „Form8“. Bendriausiu konfigūravimu visų tipų saugykloms laikomas prisijungimo būdo nustatymas. Taigi, norint leisti vartotojui nurodyti bent tai, objekte „PageControl1“ įterpiamas naujas „Tab“ mygtukas, o į jį - „ComboBox4“ objektas. Tuomet kataloge sukuriamas failas „saug\_<saugyklos tipas>.conf“ skirtas saugoti naujo saugyklos tipo nustatymams.

Toliau programos kode atliekami pakeitimai, nurodyti 20. lentelėje, jeigu norime vartotojui leisti keisti vien prisijungimo prie saugyklos būdą. Taip pat šie pakeitimai atliekami tik tada, kai norima realizuoti kelis galimus prisijungimo būdus. Priešingu atveju nebelieka prasmės.



20. lentelė - failo „Unit8.cpp“ redagavimas

[#1]	Nutrinami komentarų simboliai. Kodo paaiškinimas greta.
[#2]	Redaguojamas komentarais pažymėtas kodas analogiškai pagal prieš tai esantį kodą. Čia – „<prisijungimo budas nr.2>“ bus rodomas naujai sukurtame „ComboBox4“ objekte. Paskutinį įterpti tą prisijungimo būdą, kurį norėtų vartotojas naudoti pagal nutylėjimą.
[#3]	Redaguojamas komentarais pažymėtas kodas. Čia „xx“ - prisijungimo būdo, naudojamo pagal nutylėjimą, indeksas.
[#4]	Redaguojamas komentarais pažymėtas kodas. Čia kviečiamas metodas „nustatymu_issaugojimas_universalus()“, kuriam perduodami du argumentai: <ul style="list-style-type: none"> <li>- „ComboBox4“, tai naujai sukurtas objektas objekte „TabSheet3”.</li> <li>- Antras argumentas, tai failo, kuriame bus saugomi naujo saugyklos tipo nustatymai, failas.</li> </ul>

21. lentelė Failo „Unit6.cpp“ redagavimas:

[#1]	Ištrinami komentarų simboliai. Rašomas kodas SQL užklausiai suformuoti. Metodas lentelėje „Projektas“ išsaugo vartotojo naujai įvestus duomenis. Čia naudojami kintamieji: „edi1“, „edi2“ ir t.t. – žymi „Edit“ laukuose įvestas simbolių eilutes; „mem1” – kintamasis skirtas saugoti tekstui, įvestam į „Memo1”. 22. lentelė vaizduoja iš kokių interfeiso objektų užpildoma lentelė „Projektas“ kiekvienos saugyklos tipo atžvilgiu.
[#2]	Esant skirtingam saugyklos tipui, vartotojo gali būti pareikalauti įvesti skirtingus prisijungimo duomenis. Todėl esant skirtingam saugyklos tipui turi būti pakoreguota sąsaja. Pvz. „SQL Server“ saugyklos atveju yra reikalingas slaptažodis. Tuo tarpu „Access“ saugyklai jis nėra būtinas. Tad nutrinami komentarų simboliai ir redaguojamas kodas. Ties [#2a] sukuriamas kodas, kuris nustatys sąsają pagal naują saugyklos tipą.
[#3]	Pagal nefunkcinius reikalavimus, įrankio duomenų bazėje negali būti išsaugoti keli projektai, skirti vienai ir tai pačiai saugyklai. Pvz.: „SQL Server“ atveju negali būti sukurti keli projektai, saugantys duomenis, skirtus prisijungti prie vieno ir to paties serverio, vartotojos vardo ir duomenų bazės. „Access“ atveju negali būti sukurti keli projektai analizuoti vienam ir tam pačiam „MDB“ failui. Panašūs reikalavimai gali būti keliami naujam saugyklos tipui. Todėl ties [#3a] kuriamas naujas metodas „tikrinti_ar_egzist_<naujas RS saugyklos tipas>( <sarasas butinu patikrinti Edit reiksmiu>“ analogiškas metodams „tikrinti_ar_egzist_mssqlserver(ed2, ed3, ed5)“ ir „tikrinti_ar_egzist_access(ed4)“.
[#4]	Metodas „prisijungimo_eilute“ grąžina suformuotą prisijungimo eilutę kiekvieno saugyklos tipo atveju. Nutrynus komentarų simbolius redaguoti kodą.
[#5]	Metodas išveda informaciją lauke „Form1->Statictext5” apie reikalavimų specifikacijų saugyklą, prie kurios prisijungta. Ištrinami komentarų simboliai ir redaguojamas kodas – formuojamas „AnsiString“ tipo kintamasis „caption_text“. Skirtingo saugyklos tipo (esant skirtingai „rdio_itind“ reikšmei, kuri žymi kuris tipas parinktas) gali būti išvedama skirtinga informacija. Perduodami argumentai – vartotojo įvesto reikšmės į „Edit“ tipo laukus.
[#6]	Metodas „pazymeto_projekto_info()“. Čia „XX“ žymi naują projekto tipą, kurio reikšmė

	saugoma lentelėje „Projektas“. Užklauso pagalba iš lentelės nuskaitoma „Memo2“ lauke pažymėto projekto visa informacija. Tuomet reikia sukurti kodą informacijos pranešimui „pazymeto_proj_info“, apibūdinančiam pažymėtą projektą, suformuoti.
[#7a]	Sukuriamas kintamasis naujo saugyklos tipo reikšmei saugoti. Čia "XX" žymi reikšmę, kuri saugoma lentelėje „Projektas“.
[#7b]	Redaguojamas kodas – turi būti gražinama „int“ tipo reikšmė, atitinkanti naują saugyklos tipą.
[#7c]	Redaguojamas kodas. Metodui jungtis_prie_rs_saugyklos perduodami argumentai, kurių reikšmės randamos prieš tai „SQL“ užklauso pagalba.

22. lentelė - įvedimo laukų paskirtys

Lentelės „Projektas“ stulpeliai	pavad	server	vardot	slapt	db	aprasas	db_slapt
SQL Server	Edit1	Edit2	Edit3	Edit4	Edit5	Memo1	-
Access	Edit1	-	Edit2	Edit3	Edit4	Memo1	Edit5

23. lentelė - failo „Unit7.cpp“ redagavimas:

[#1]	„XX“ žymi naujo saugyklos tipo reikšmė saugoma lentelėje „Projektas“. Papildomai reikalinga sukurti metodą „nustatyti_interfeisa_xx()“, kuris nustato objektų „Label“ bei kitas reikšmes „Form7“ lange. Nurodyti kokius SQL užklausa nuskaitytus duomenis išvesti į atitinkamus „Edit“ laukus.
[#2]	„IF“ sakinytis papildomas „Edit“ laukų vardais, kurias būtina užpildyti pasirinkus naujo saugyklos tipo projektą.
[#3]	Reikalinga sukurti naują metodą „ar_egzistuoja_xx(...)“, kuris patikrintų, ar atlikus kokio nors projekto duomenų redagavimus, lentelėje „Projektas“ nėra kito projekto, turinčio tokį pat vardą ar skirtą tai pačiai saugyklai analizuoti.
[#4]	Sukuriamas naujas metodas „issaugoti_pakeitimus_xx()“, kuris SQL užklausa atnaujins redaguojamo projekto duomenis, saugomus lentelėje „Projektas“.

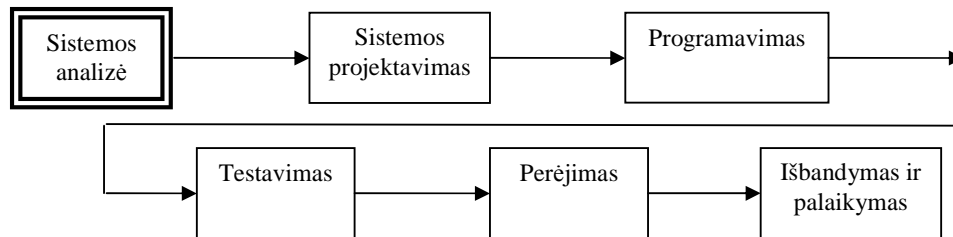
## 8.6 Straipsnis

### CASE ĮRANKIAI REIKALAVIMŲ SPECIFIKACIJOMS ANALIZUOTI

Ankstyvojoje kompiuterizuojamos informacinės sistemos kūrimo stadijoje reikalavimų inžinerija apima visus darbus susijusius su reikalavimų sistemai nustatymu ir šio proceso metu išylančių problemų tyrinėjimu. Šiame etape kuriama atitinkamos informacinės sistemos reikalavimų specifikacija. Jai sudaryti yra pasiūlyta skirtingų notacijų, kurių naudojimas atitinkamu atveju priklauso nuo projekto tikslų ir prieinamų išteklių. Reikalavimų specifikacija turi būti kokybiška. O tai įvertinama atitinkamomis kokybės charakteristikomis. Specifikacijoms analizuoti gali būti panaudotos įvairios priemonės, tame tarpe ir CASE įrankiai. Atliekant reikalavimų analizę, specifikacija plėtojama ir keliama jos kokybė. Kokybiška specifikacija yra pagrindas sukurti aukšto funkcionalumo lygio sistemą.

#### 8.6.1 Reikalavimų specifikavimas IS kūrimo proceso kontekste

Informacinės sistemos (IS) kūrimo darbai prasideda nuo sistemingo reikalavimų išsiaiškinimo, jų analizavimo ir užrašymo. Pradinėje IS kūrimo pakopoje – sistemos analizėje (1.pav. IS gyvavimo ciklo schemeje ji vaizduojama proceso pradžioje) – sutelkiamas dėmesys darbui su reikalavimų pateikėjais, kad išgauti visą informaciją, reikalingą suprasti kompiuterizuojamą sritį. Daug paprasčiau ir pigiau ištaisyti klaidas, aptiktas sistemos kūrimo proceso pradžioje, negu bandyti spręsti neatitikimus diegimo ir palaikymo stadijų metu. [1, 2]



1. pav. Apibendrinta IS gyvavimo ciklo schema.

*Reikalavimų inžinerija* – terminas, naudojamas ankstyvojoje sistemos kūrimo stadijoje, kuri apima problemų, susijusių su kompiuterizuojama sritimi, tyrinėjimą, reikalavimus sistemai, vartotojus. Reikalavimų inžinerija pirmiausia prasideda nuo reikalavimų išgavimo. Šios užduoties tikslas yra surinkti ir atvaizduoti inžineriją, kurią turės palaikyti kompiuterizuota informacinė sistema, reikalavimus. Jų užrašymas, paprastai, vadinamas reikalavimų specifikavimu. Šiam tikslui pasiūlytos skirtingos notacijos - nuo nestruktūrizuotų ir neformalių tekstų iki aukšto lygio formalių matematinių notacijų. Išgauti reikalavimai analizuojami, kad išsiaiškinti kompiuterizuojamos srities kūrimo problemas, patikrinti reikalavimų specifikacijos atitikimą kokybės kriterijams ir ją tobulinti. Pačią specifikaciją galima apibūdinti kaip architektūrinį, nuo diegimo nepriklausomą būsimosios sistemos planą, kuriame atvaizduojamas užsakovo poreikis įvairioms kuriamos IS savybėms. *IS projektavimo inžinerija* vadinamos tolesnės kūrimo fazės, kuriose naudojama ši specifikacija, skirta funkcionuojančios sistemos projektavimui ir diegimui, bei, taip pat, sistemos patikrinimui. Jau realizavus sistemą, specifikacija taps vieninteliu šaltiniu, kuris gali būti panaudotas patikrinti ar įdiegta sistema tiksliai atitinka tai, kas buvo užsakyta. Nepriekaištingai sudaryta specifikacija tampa svarbiu pagrindu įgyvendinti aukšto lygio funkcionalumą informacinėje sistemoje, esant kuo mažesnėms palaikymo ir pritaikymo išlaidoms. Pasikeitus aplinkai ir, dėl tos pat priežasties, reikalavimams informacinei sistemai, reikalavimų specifikacija tampa aprašymu, skirtu permodeliuoti pokyčius bei aptarti kaip jie turėtų būti suderinti su IS pokyčiais.[1, 2, 3, 4]

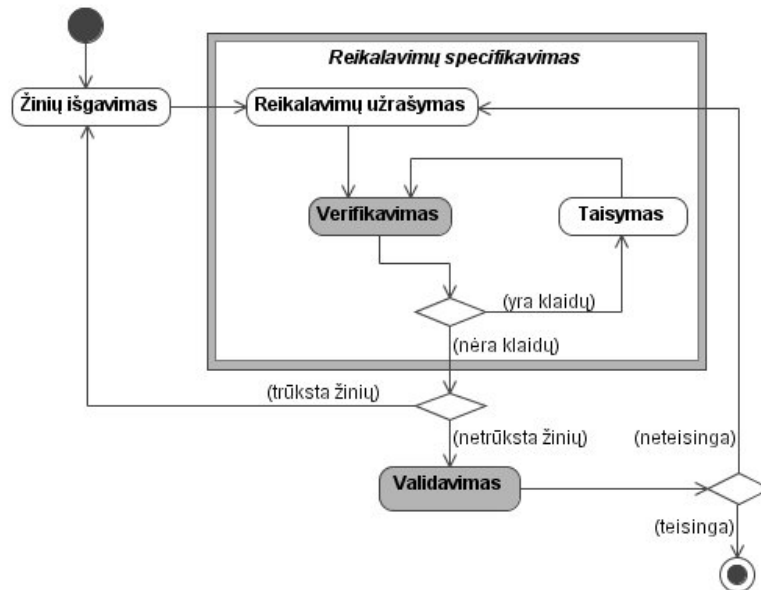
Kadangi reikalavimų inžinerijos tikslas yra sukurti kokybišką specifikaciją, derėtų išskirti reikalavimų analizės proceso verifikavimo ir validavimo subprocesus (2.pav.). Jų tikslai yra patikrinti ar formali ir konceptuali specifikacija yra tinkama, ir ar ji tinkamai išreiškia funkcinius reikalavimus, suformuluotus vartotojo. Atliekant verifikavimą ir validavimą ankstyvuosiuose informacinės sistemos kūrimo žingsniuose, gali būti pagerinta reikalavimų specifikacijos kokybė ir sumažintos išlaidos bei rizika tolesniuose etapuose.

Verifikavimas apima reikalavimų specifikacijos savybių, tikrinimą. Problemos, atliekant verifikavimą, sprendžiamos sintaksiniam ir semantiniame lygyje. Sintaksiniam lygmenyje problema yra įrodyti, kad specifikacija yra pilna ir neprieštaraujanti. Semantiniame lygmenyje reikia įvertinti reikšmes, pateiktas specifikacijoje, bei ieškoti nesuderinamumų. Kuomet dar tik renkami reikalavimai, kiekvienas jų verifikuojami kokybės kriterijų, tokių, kaip teisingumas, patikrinamumas, keičiamumas ir kt. atžvilgiu. Kitos savybės, tokios, kaip pilnumas ir atsekamumas, įvertinamos tik išgavus pilnus reikalavimus.[3, 5]

Validavimas - tai procesas, kuris užtikrina, kad reikalavimai tiksliai atitiktų užsakovo poreikius.

Pagrindinis tikslas – nustatyti nesutapimus tarp to, ko reikalauja klientas, ir to, kas suformuluota reikalavimuose.[6]

Kokybiška reikalavimų specifikacija svarbi tuo, jog būtent ja naudojantis atliekama nuodugni reikalavimų analizė. Tikėtina, jog, kol nebus pradėta eksploatuoti sistema, nebus rastos jokios klaidos. O suradus trūkumus vėlyvame etape neišvengiama žymių projekto nuostolių ir papildomų laiko sąnaudų klaidų ištaisymui.



2. pav. Reikalavimų specifikacijų sudarymo ir tikrinimo procesas.

## 8.6.2 Reikalavimų specifikacijų kokybė

Kokybės atžvilgiu, reikalavimų specifikacijos nusakomos tam tikromis charakteristikomis, apibūdintomis žemiau. Šios charakteristikos gerinamos atliekant reikalavimų specifikacijų analizę. Įvairiuose šaltiniuose (1., 7., 8., 9., 10., 11.) paminėta ir kitokių savybių, kurias turėtų atitikti reikalavimų specifikacija, tačiau čia apibendrintos svarbiausios iš jų :

- **Pilnumas (angl. completeness).** Pilna reikalavimų specifikacija turi tiksliai apibrėžti visas galimas sąlygas, sistemos galimybes, realaus pasaulio situacijas, su kuriomis bus susidurta, ir galimas reakcijas į jas.
- **Neprieštaringumas, darna (angl. consistency).** Darni specifikacija tai tokia, kurioje nėra prieštaravimų tarp atskirų reikalavimų formuluočių.
- **Tikslumas (angl. accuracy).** Reikalavimų specifikacija turi tiksliai apibrėžti sistemos gebėjimus realaus pasaulio aplinkoje, taip pat kaip ji su juo siejasi ir sąveikauja.
- **Minimalumas (angl. minimality).** Modelyje turi būti saugoma tik tai, kas neabejotinai reikalinga.
- **Nepertekliškumas (angl. redundancy).** Modelyje tie patys reikalavimai neturi kartotis.
- **Keičiamumas (angl. modifiability).** Vartotojai kartais keičia savo reikalavimus kuriamai sistemai, todėl modifikuojama specifikacija turi būti tokia, kad būtų galima suderinti atitinkamas struktūras ir charakteristikas.
- **Klasifikavimas (angl. ranking).** Atskiri reikalavimų specifikacijos reikalavimai turi būti hierarchiškai klasifikuojami remiantis pastovumu, saugumu, suvoktu įgyvendinimo sudėtingumu arba pagal kitokią kriterijų, kuris pagelbsti sudarant specifikaciją.
- **Testuojamumas (angl. testing).** Reikalavimų specifikacija turi būti sudaryta taip, kad leistų patikrinti ar sistema atitinka reikalavimus.
- **Trasuojamumas (angl. traceability).** Reikalavimų trasavimo procesas užtikrina, kad reikalavimai, apibrėžti specifikacijoje, yra įvykdomi. Reikia įvertinti kiekvieną reikalavimą kaip atskirą formuluotę, kuri gali būti trasuojama, tuomet projekto komanda gali įsitikinti, kad sistemai nustatyti reikalavimai išpildomi.
- **Nedviprasmiškumas (angl. unambiguity).** Specifikacija turi sudaryti tik vienareikšmiškai suprantamus reikalavimus. Jie negali turėti kelių prasmų. Daugiausia su tuo susiduriama naudojant

natūralią kalbą. Kad išvengti to, tinkamai turėtų būti naudojamos formalios ar pusiau formalios kalbos.

- **Pagrįstumas (angl. *validability*).** Reikalavimai turi atitikti jų pateikėjų poreikius ir norus. Pagrįsta specifikacija tai tokia, kurioje visi projekto dalyviai (vadybininkai, projektuotojai, užsakovai) gali ją suprasti, analizuoti ir pritarti. Tai viena iš pagrindinių priežasčių kodėl reikalavimų specifikacijos rašomos natūralia kalba.
- **Patikrinamumas (angl. *verifiability*).** Reikalavimai visuomet turi pradinius informacijos šaltinius, todėl svarbu, kad jie juos ir atitiktų. Taip pat, reikia, kad reikalavimai atitiktų naudojamus standartus, rekomendacijas bei šablonus. ,
- **Tinkamumas naudoti (angl. *usability*).** Iš daugelio vartotojų surenkami reikalavimai, kurie turi būti praktiški: ar visi reikalavimai suprantami? Ar specifikacijos sudarymo būdas suprantamas užsakovui ir vartotojui, kurie turės naudotis pateiktais reikalavimais, kad valdyti ir įvertinti tam tikrą sritį? Ar visi reikalavimai suprantami ir patogūs vadovams, kurie turi naudotis jais tiek srities, tiek išlaidų, planavimų valdymui, eigos metrikoms? Ar visi architektūros požiūriu svarbūs reikalavimai suprantami ir naudingi kūrėjams, kurie remiantis jais kurs struktūrą? Ar visi reikalavimai suprantami ir patogūs projektuotojams ir programuotojams, kurie juos turės įgyvendinti?

### 8.6.3 Formalizmai reikalavimų specifikacijoms atvaizduoti

Kompiuterizuotos informacinės sistemos reikalavimų specifikacijoms sudaryti naudojami įvairūs formalizmai. Formali specifikacija ir neformali specifikacija – du priešingi reikalavimų specifikavimo būdai. Be to, taip pat egzistuoja tam tikrų tarpinių atvaizdavimų, kurie pasirodė naudingi kaip bendradarbiavimo priemonės tarp sistemos užsakovo ir jos kūrėjo. Pusiau formalūs žymėjimai taip pat naudojami aprašyti projektą. Jie neturi standartinės notacijos. Formali specifikacija tai yra tam tikro savybių rinkinio, kurį turi atitikti sistema, išraiška, pateikta tam tikroje formalioje kalboje ir tam tikrame abstrakcijos lygyje. Lentelė Nr.1 vaizduoja formalumų skirtingas kategorijas bei atitinkamus pavyzdžius.

Lentelė Nr. 1. Notacijų klasifikavimas.

Neformalios	Pusiau formalios	Formalios
Šie metodai neturi išbaigtų taisyklių rinkinių, kuriais būtų apribojami kuriami modeliai. Būdingi atvejai - natūrali kalba (parašytas tekstas) ir nestruktūrizuoti vaizdai.	Šie metodai turi apibrėžta sintaksę. Būdingi atvejai - grafiniai būdai turintys tikslias taisykles, kurios tiksliai apibūdina sąlygas pagal kurias leidžiamas konstravimas, tekstiniai bei grafiniai aprašymai, esant ribotoms patikrinimo galimybėms.	Šie metodai turi griežtą apibrėžtą sintaksę ir semantiką. Sudaromas esminis teorinis modelis, pagal kurį gali būti patikrintas aprašymas, išreikštas matematine notacija. Būdingi atvejai - specifikavimo kalbos, grindžiamos logika.
Pavyzdžiai:		
Specifikacijos natūralia kalba	Duomenų/valdymo, srautų diagramos; esybių ryšių diagramos; panaudojimo atvejų diagramos	Petri tinklai; VDM; Z

Terminas “formalus” dažniausiai yra susijęs su “tikslumu”. Specifikacija yra formali, jei ji išreikšta kalboje, specialiai sukurtoje programinės įrangos specifikavimui ir turinčioje tris komponentus:

- taisyklės, nustatančios sakinių gramatinį žymėjimą (sintaksė);
- tam tikra elementų aibė, skirta sakinių interpretavimui atitinkamos srities ribose (semantika);
- taisyklės, naudingai informacijai iš specifikacijos išvesti; jas turi atitikti specifikuojamas objektas. Šis komponentas sudaro pagrindą automatizuotai specifikacijų analizei.

Formalūs metodai propaguojami kaip vienas iš būdų charakterizuoti reikalavimus esant matematiniam tikslumui ir griežtumui. Jie nusako būdus kurti sistemos modeliams tiksliai apibrėžiant jų struktūrą ir elgseną. Naudojant įrankius, pagrįstus formaliais specifikavimo metodais, galimas automatizuotas savybių patikrinimas sudėtinguose modeliuose. [12, 13, 14]

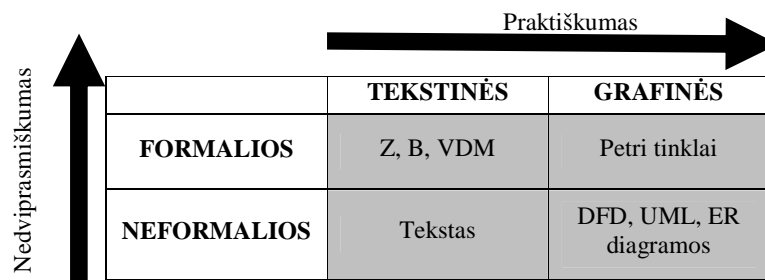
Formalių kalbų pranašumas tas, jog jos yra nedviprasmiškos ir tikslios, ir jas galima palaikyti kompiuteriais bei matematinėmis analizėmis. Tačiau visa tai turi savo vertę - personalas turi išmokti atitinkamą kalbą (kuri daugumai žmonių atrodo sudėtinga) ir formalios specifikacijos sudarymui reikia daugiau laiko. Kadangi grindžiami matematinėmis struktūromis ir notacijomis, priešingai nei natūrali kalba, formalūs metodai yra ne tokie intuityvūs nespecializuotai publikai. [7]

Natūrali kalba yra vienas iš būdų, skirtų padėti vartotojui sudaryti taikomosios srities reikalavimų specifikaciją, kuri kuriama jo pateikiamų teiginių atžvilgiu. Vaizdavimas natūralia kalba yra pradinė reikalavimų formuluotės nustatymo forma. Ji yra patogi, nes leidžia ne techninio pobūdžio vartotojams suprasti sistemos reikalavimus bei ja galima nusakyti daugybę smulkių kompiuterizuojamos srities detalių. Reikalavimų

inžinerijoje natūrali kalba yra dažniausiai naudojamas vaizdavimo būdas reikalavimų sudarymui. Ji yra universali, lanksti ir plačiai taikoma, bet, deja, iš prigimties dviprasmiška bei neapibrėžta. Ją naudojant sudėtinga perprasti visos sistemos reikšmę, kadangi žvelgiama tik į tekstinę specifikaciją. O iš anksto nusakytos semantikos trūkumas padidina tikimybę klaidų, atsirandančių dėl suvokimo ir būdingų dviprasmybių. Taip pat egzistuoja kitos problemos naudojant natūralią kalbą. Dėl šios priežasties programinės įrangos kūrėjai projektuoja ir diegia sistemas, kurios neveikia taip, kaip užsakė vartotojas, kadangi projektuotojo reikalavimų specifikacijos supratimas gali skirtis nuo užsakovo. Kaip alternatyva – grafinės notacijos - paprastai aiškesnės, tačiau jas sudėtingiau praplėsti. Trečiame paveiksle pavaizduotos tekstinės bei grafinės charakteristikos yra ortogonalios specifikacijų formalumo atžvilgiu. [15]

Neformalią specifikaciją sudaryti dažniausiai nesudėtinga, tačiau tokiu būdu gauta ji būna prastos kokybės ir sudėtinga atlikti reikalavimų analizę. Todėl neužtikrinama, jog specifikacijos bus išbaigtos, nedviprasmiškos ir jos pilnai atitiks funkcinių ir nefunkcinių reikalavimų charakteristikas. Tačiau daugelis žmonių plačiai naudoja neformalias specifikacijas, kadangi formalūs metodai laikomi sudėtingais ir brangiais. Formalioms specifikacijoms sudaryti reikia daugiau laiko ir specialistų pagalbos, bet jos palaiko analizę ir sumažina testavimų pastangas.

Grafinė notacija yra vienas iš galimų natūralios kalbos panaudojimo alternatyvų, skirta reikalavimų specifikavimui. Ji yra plačiai paplitusi ir priimtina, tam tikru mastu naudojama reikalavimų inžinerijoje. Grafinis žymėjimas apima elementus skirtus aprašyti struktūrinę ir elgsenos informaciją. Daugeliu atvejų informacija suvokiama lengviau, kai ji pateikiama grafiškai. Egzistuojančių išsamių iliustruotų vaizdų dėka vartotojai, neapmokyti formalių metodų, tinkamai supranta specifikaciją ir sumažėja jų supratimo atotrūkis nuo išmanančių griežtą formalizmą. Grafinė specifikacija tai tokia, kurios elementai yra vizualiniai, ne tekstiniai. Tačiau, nepaisant kalbinės atitikties, grafinės specifikacijos nėra taip paprastai susijusios su jų tekstinėmis kopijomis.



3 pav. – Reikalavimų specifikavimo notacijų klasifikavimas.

Manoma, jog natūralios kalbos, grafinės notacijos ir formalių metodų derinys yra būtinas reikalavimų inžinerijos problemų sprendimui:

- Reikalavimų specifikacija turi būti prieinama formatu tinkamu ne techninio pobūdžio vartotojams;
- Formalūs metodai turi būti panaudoti reikalavimų modeliavimui ir valdymui, kad padėti projektuotojams iš anksto rasti nesuderinamumus.
- Vartotojui palankios ir formalios reikalavimų versijos turi būti sinchronizuojamos automatiškai;
- Turėtų būti panaudotas automatizavimas, skirtas palengvinti formalių modelių generavimą iš natūralios kalbos reikalavimų, tuo būdu supaprastinant reikalavimų inžinerijos užduotį kuriant modelius nuo pradžių. [4, 7, 12]

Sudėtinga paruošti konkrečią, formalią ir visiems projekto dalyviams suprantamą reikalavimų specifikaciją. Sistemos vartotojas turi patikrinti visus reikalavimus, nes jis yra jų pateikėjas. Tačiau tai įmanoma tik jei reikalavimai pavaizduoti jam suvokiama forma. Reikalavimų specifikacijos kūrimui eilinis vartotojas gali pasitelkti paprasčiausią ir jam neabejotinai suprantamą formą – natūralią kalbą. Tačiau tokiu būdu sudarytas specifikacijas sudėtinga analizuoti. Formalūs metodai netaikomi ankstyvosiose reikalavimų specifikavimo stadijose, kai reikalavimai yra migloti. Formalizmai, naudojami tradiciniuose metoduose reikalavimų specifikavimui, yra nepriimtini vartotojui, jeigu jis neturi specialių žinių jiems suprasti. Tačiau svarbu nepamiršti, jog būtina kelti reikalavimų specifikacijos kokybę. Formalių specifikacijų (sudarytų formaliais metodais, formaliomis kalbomis) privalumas tas, kad mes galime patikrinti jų teisingumą, atitikimą kokybės charakteristikoms. O svarbiausia, jog galime jas analizuoti ir, tuo būdu, gerinti kokybę. Tokiais atvejais talkintų CASE įrankiai. Tačiau egzistuoja tam tikri apribojimai. Ypač sudėtinga būtų sukurti CASE įrankį, kuris leistų atlikti reikalavimų specifikacijų, parašytų natūralia kalba, analizę. Kadangi natūrali kalba nėra formalizuota ir neturi apibrėžtų griežtų taisyklių reikalavimų specifikavimui, šiuo būdu sudarytas specifikacijas gali analizuoti tik žmogus, pvz.: analitikas. Specifikacijos kokybės gerinimas priklausys nuo jo kompetencijos ir patirties. Visiškai priešinga situacija naudojant formalias kalbas ir metodus, kurie turi griežtas taisykles reikalavimų specifikavimui. Tokioms specifikacijoms analizuoti nėra ypač sudėtinga sukurti kompiuterizuotą priemonę, kadangi iš anksto žinomos reikalavimų užrašymo taisyklės. Naudojantis šiais įrankiais galima tiksliai rasti klaidas, padarytas specifikacijoje ir tuo būdu efektyviai kelti jos kokybę. Taip pat naudojantis CASE įrankiais

įmanoma atlikti ir specifikacijų, užrašytų pusiau formaliomis kalbomis, analizę. Tačiau šiuo atveju, skirtingai nuo specifikacijų, sudarytų formaliais metodais ir kalbomis, atlikta analizė nebus tokia išsami ir tiksli. Taip yra dėl to, kad šiuo būdu sudarytose specifikacijose klaidų ir neatitikimų paiešką galime kompiuterizuoti tik iš dalies. Pusiau formalią kalbą turi tam tikras taisykles reikalavimams specifikuoti, tačiau jos nėra griežtos. Sukurta analizės priemonė pateiktų ne konkrečius rezultatus, o tik tam tikrą specifikacijos sritį, kur tikėtini netikslumai ir klaidos.

#### 8.6.4 CASE įrankių papildomos priemonės specifikacijų (modelių) analizavimui

Daugelis CASE įrankių pasižymi papildomais funkcionalumais, skirtais analizuoti sudarytus modelius. Tai labai aktualu, jei šie modeliai dideli ir sudėtingi. Tuomet galima nustatyti jų savybes, elementų priklausomybes, struktūros ir elgsenos santykių, atitikimą reikalavimams. Analizuojant modelius didinamas kuriamos sistemos našumas, geriau išsiginama į probleminę sritį, mažinamos papildomos išlaidos, taupomas kūrimo laikas, keliama reikalavimų specifikacijos kokybė. Palyginimui toliau pateikti ir detalizuoti kai kurie CASE įrankiai, turintys papildomas priemones kompiuterizuojamos sistemos modelių analizavimui:

**Oracle Designer - Matrix Diagrammer.** Matricių pavidalu kuriami ir vaizduojami ryšiai tarp saugykloje saugomų skirtingų tipų elementų. Sankirtos elementas parodo kaip vienas objektas priklauso nuo kito. [16]

**Rational RequisitePro - Use case Attribute Matrix View.** Naudojant šį įrankį, pateikiamas visų arba tik atrinktų panaudojimo atvejų bei jų atitinkamų atributų poaibis. Tai naudinga sudarant panaudojimo atvejus, nustatant kuris panaudojimo atvejis pavestas kuriam projektuotojui, koks jų realizavimo sudėtingumas, ar kokia tvarka jie turėtų būti realizuoti.

**Rational RequisitePro - Traceability Matrix.** Atsekamumo matricioje vaizduojami ryšiai tarp panaudojimo atvejų ir reikalavimų sistemai. Nustatant santykius tarp tikslų ir kuriamos sistemos savybių, galima tiksliai nustatyti kokių tikslų nebūtų įmanoma išpildyti, jeigu nuspręstume nerealizuoti tam tikros savybės. Naudojant atsekamumo matricią, galima atsakyti į klausimą: „Kokie panaudojimo atvejai siejami su atitinkamais sistemos reikalavimais?“ Ir taip pat, jei reikalavimai sričiai pasikeičia: „Kokie panaudojimo atvejai įtakojami?“ [17]

**ProVition Workbench - Completeness Checker.** Automatizuotas pilnumo tikrinimo įrankis leidžia analizuoti visą modelį bei aptikrinti trūkstamus aprašymus, praleistas savybes ar praleistus ryšius. Tikslas – užtikrinti, kad modelio objektai ir jų ryšiai būtų išsamūs ir tikslūs. [18]

**MagicDraw - UML Metrics.** Metrika, tai skaitinė reikšmė, skirta įvertinti modelį ir apskaičiuojama atžvilgiu trijų skirtingų pozicijų:

- UML modelio metrikos
- Sistemos metrikos
- Reikalavimų metrikos. [19]

**Enterprise Architect Professional - Element Relationship Matrix.** Sąryšių matrica leidžia tyrinėti ryšius tarp modelio elementų. Naudojantis ja galima kurti, redaguoti ir ištrinti ryšius tarp elementų. Ši matrica, tai patogus ryšių vizualizavimo būdas. [20]

#### 8.6.5 Baigiamieji darbai

Bet kurią reikalavimų specifikaciją gali analizuoti žmogus. Tačiau esant dideliame sistemos modeliui tai gali būti ypač ilgas ir sudėtingas darbas, kurio rezultatai priklausys nuo specifikaciją analizuojančio darbuotojo kompetencijos ir patirties. Tam, kad palengvinti analitiko darbą, kuriamos įvairios specifikacijų analizės kompiuterizuotos priemonės, kurios palaiko formalią arba pusiau formalią specifikavimo kalbą ir metodus. Tuo būdu efektyviai įmanoma kelti specifikacijų kokybę. Sukurti natūralia kalba užrašytų specifikacijų analizės priemonę būtų ypač sudėtinga. O realizuoti universalų įrankį tiesiog neįmanoma, nes kompiuterizuojamos IS reikalavimų specifikacijos priklauso nuo konkrečių metodų. Skirtingam metodui turi būti skirtos atskiros priemonės.

Tolesniuose magistrinio darbo etapuose ruošiuosi suprojektuoti ir realizuoti papildomo funkcionalumo priemonę reikalavimų specifikacijų analizavimui. Mano siūlomas analizės įrankis priklausys nuo ODRES (Output Driven Requirements Specification) metodo, kuris nėra griežtai formalizuotas. O išbandymui savo kuriamą priemonę susiesiu su KTU Informacijos sistemų katedroje sukurtu CASE įrankiu. Jie abu palaikys tą patį metodą bei reikalavimų specifikacijų saugyklą. Mano kuriamo įrankio naudojimas palengvins analitiko darbą, atliekant ODRES metodu specifikuotų reikalavimų analizę. Reikalavimų analizavimui ruošiuosi panaudoti matricią, kurioje bus galima sukurti įvairius modelių elementus, tuo būdu nesunkiai bus atrandamos

priklausomybės tarp jų.

#### LITERATŪRA (REFERENCES)

29. Smitha S., Laia L. ir Khedria R. Requirements analysis for engineering computation. Computing and Software Department, McMaster University, 2004: 5 p.
30. Butkienė R., Butleris R. The Approach for User Requirements Specification. Department of Information Systems, Kaunas University of Technology, 2001: pp. 1-3
31. Bubenko J.A. Wangler B. Research directions in conceptual specification development. SYSLAB, Dept. of Computer and Systems Science and SISU – the Swedish Institute for System Development, 1991: pp. 1-5.
32. Tse T.H., Pong L. An Examination of Requirements Specification Languages. Department of Computer Science, The University of Hong Kong: 2003, pp. 1-4.
33. Bahill A.T., Henderson S.J.: Requirements Development, Verification, and Validation Exhibited in Famous Failures, 2005
34. Lobo L.O. Analysis and Evaluation of Methods for Activities in the Expanded Requirements Generation Model (x-RGM). Blacksburg, Virginia, 2004: 24 p.
35. Saaman E.H. Another formal Specification language. IPA Dissertation Series, 2000: 3 p.
36. Donn Le Vie. Writing Software Requirements Specifications. The Official TECHWR-L Web Site. Prieiga per internetą [žiūrėta 2006 01 20]:  
<http://www.techwr-l.com/techwhirl/magazine/writing/softwarerequirementspecs.html>
37. Requirements specification, chapter 10. Extract from: STARTS Purchasers' Handbook Second Edition, 1989: 18 p..
38. Bubenko J.A. Extending the Scope of Information Modelling. Department of Computer and Systems Science, Royal Institute of Technology and Swedish Institute for Systems Development, 1993: pp. 2-6, 19.
39. Smith M. Project Management Planning - Top-Level Requirements Specification, 1997.
40. Rossel P., Contreras R., Cecilia B.M. Graphic specification of abstract data types. Revista Facultad de Ingeniería, Chile Vol.12 N°1, 2004: pp. 15-16.
41. Lamsweerde A. Formal Specification: a Roadmap. Département d'Ingénierie Informatique, Université catholique de Louvain, B-1348 Louvain-la-Neuve (Belgium), 2000.
42. Fernandes R., Cowie A.J. Capturing Informal Requirements as Formal Models. AWRE - Australian Workshop on Requirements Engineering, 2004.
43. Hallock P. Natural Language Modeling. InConcept (Information Conceptual Modeling, Inc.). Prieiga per internetą [žiūrėta 2006 01 21]: <http://www.inconcept.com/JCM/February2001/hallocktext.html>
44. Oracle Corporation oficiali svetainė [interaktyvi]. Prieiga per internetą [žiūrėta 2006 04 01]: <http://www.oracle.com>
45. IBM Rational Software oficiali svetainė [interaktyvi]. Prieiga per internetą [žiūrėta 2006 04 01]: <http://www.rational.com>
46. Proforma Corporation oficiali svetainė [interaktyvi]. Prieiga per internetą [žiūrėta 2006 04 01]: <http://www.proformacorp.com>
47. No Magic, Inc. oficiali svetainė [interaktyvi]. Prieiga per internetą [žiūrėta 2006 04 01]: <http://www.magicdraw.com>
48. Sparx Systems oficiali svetainė [interaktyvi]. Prieiga per internetą [žiūrėta 2006 04 01]: <http://www.sparxsystems.com>

#### CASE TOOLS FOR ANALYZING OF REQUIREMENTS SPECIFICATIONS

In the early stage of building a new informational system requirement engineering involves all the processes related to the requirement system and research of problems emerged during this process. In this stage the requirements specification of corresponding information system are built. Many different notations serve for this purpose, selection of which is dependant on available resources and project goals. Requirements specification must be of good quality. It is evaluated by characteristics of quality. Specifications could be analyzed and improved by using various tools including CASE tools. The specification is developed and its quality is improved by making analysis of requirements. Well done specification is basis for creating high leveled functionality system.