



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Raimundas Dundulis
**SAP išplėtimas adaptuojamų taikomųjų
programų kūrimui**

Magistro darbas

Darbo vadovas
doc. dr. Lina Nemuraite

Kaunas, 2006



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Raimundas Dundulis
**SAP išplėtimas adaptuojamų taikomųjų
programų kūrimui**

Magistro darbas

Kalbos konsultantė

Lietuvių k. katedros lekt.
J. Mikelionienė

2006-05

Recenzentas

doc. dr. S. Maciulevičius

2006-05

Vadovas

doc. dr. Lina Nemuraitė
2006-05

Atliko

IFM-0/2 gr. stud.
Raimundas Dundulis
2006-05

Kaunas, 2006

SAP EXPANSION FOR ADAPTIVE APPLICATIONS DEVELOPMENT

Summary

The research work is focused on analysis of possibilities of business rules formalization, storage in database tables, user friendliness improvement, and rules management.

Business rules have been taken into investigation not long ago. Possibilities of using business rules in business software are still being defined. Formalization and usage of business rules in IT systems is very important, because rules are very often in any business, and any rule has a strong tendency to change over time. The changes of rules are caused by dynamic laws, competitors' decisions, changing environment in the market, and many other factors. Business rules influence and guide the behavior of employees (influence on business), influence the information that is recorded in IT systems as well as restrict the modification of this information (influence on IT).

The practical task of the work is to design and to develop a business rules management system for usage in SAP environment. The system should optimize the work of coders during the development and maintenance of additional functions for SAP R/3 ERP system.

Turinys

1. ĮVADAS.....	3
2. VEIKLOS TAISYKLIŲ VALDYMO SPRENDIMŲ ANALIZĖ.....	6
2.1. VEIKLOS TAISYKLĖS	6
2.1.1. Veiklos taisyklių surinkimas	6
2.1.2. Praktikoje naudojamos programos ir išskylančios kliūtys.....	7
2.1.3. Svarbiausios veiklos taisyklių savybės	7
2.1.4. Formali veiklos taisyklių specifikacija	8
2.2. VEIKLOS TAISYKLIŲ VALDYMO SISTEMOS	8
2.2.1. Rule Markup Language (RuleML).....	10
2.2.2. Rinkoje egzistuojančių VTVS apžvalga	11
2.2.3. VTVS kiekybinis ir kokybinis palyginimas.....	13
2.3. PROBLEMOS ANALIZĖ	14
2.4. PRIIMTI SPRENDIMAI.....	15
2.5. ANALIZĖS IŠVADOS.....	17
3. VEIKLOS TAISYKLIŲ HIERARCHIJOS MODELIS ERP SISTEMAI.....	18
3.1. SISTEMOS PASKIRTIS.....	18
3.2. VEIKLOS SUDĖTIS	18
3.3. SISTEMOS SUDĖTIS.....	20
3.4. SISTEMOS PATEIKIMO ŽINGSNIAI.....	21
3.5. ARCHITEKTŪROS SPECIFIKACIJA	21
3.5.1. Architektūros pateikimas.....	21
3.5.2. Architektūros tikslai ir apribojimai.....	22
3.5.3. Panaudojimo atvejų vaizdas.....	23
3.5.4. Sistemos statinis vaizdas	28
3.5.5. Sistemos dinaminis vaizdas	29
3.5.6. Išdėstymo vaizdas.....	39
3.5.7. Duomenų vaizdas	41
4. EKSPERIMENTINIS SUKURTOS VTVS TYRIMAS	45
4.1. VT HIERARCHIJOS PRIVALUMAI	45
4.2. VT IR VT HIERARCHIJOS SAUGOJIMAS UNIVERSALIOJE DB	46
4.3. VT HIERARCHIJOS PANAUDOJIMAS OPTIMIZUOJANT VTVS VEIKIMĄ.....	47
4.4. VEIKLOS TAISYKLIŲ ĮVEDIMO TOBULINIMAS	48
5. IŠVADOS	50
6. LITERATŪRA.....	51
7. SANTRUMPŲ IR TERMINŲ ŽODYNAS.....	53
8. PRIEDAI.....	54
8.1. REIKALAVIMŲ SPECIFIKACIJA	55

1. ĮVADAS

Tyrimo sritis

Šio darbo tyrimo sritis yra veiklos taisyklių modeliai ir jų valdymo sistemos. Darbe nagrinėjamos veiklos taisyklių sandaros formalizavimo, užrašymo duomenų bazės lentelėse, suprantamumo vartotojui gerinimo, bei manipuliavimo taisyklėmis galimybės.

Veiklos taisyklė – tai išraiška, kuri apibrėžia arba apriboja tam tikrus veiklos aspektus. Veiklos taisyklės skirtos užtikrinti tam tikrą struktūrinę veiklos būseną ar kontroliuoti bei įtakoti veiklos elgseną.

Veiklos taisyklių valdymo sistema yra skirta centralizuoti organizacijos taisyklių atradimą, saugojimą, modifikavimą ir vykdymą. Veiklos taisyklių valdymo sistemos naudojimas gali leisti padidinti automatizavimo laipsnį, sutrumpinti reakcijos į pokyčius laiką ir sumažinti palaikymo kaštus.

Darbo tikslai

Šiame darbe siekiama ištirti veiklos taisyklių sandarą ir tarpusavio ryšius, išanalizuoti egzistuojančius veiklos taisyklių surinkimo, klasifikavimo ir saugojimo būdus.

Taip pat siekiama surasti, sumodeliuoti ir sukurti optimalų sprendimą veiklos taisyklėms valdyti *SAP R/3* sistemoje. Sukurtoji veiklos taisyklių valdymo sistema turi palengvinti programuotojų darbą *SAP R/3* sistemos išplėtimo, modifikavimo ir palaikymo metu. Būtina užtikrinti, kad sukurtoji sistema palaikys Lietuvos Muitinės *SAP R/3* sistemoje naudojamas veiklos taisykles.

Problemos aktualumas

Nuolat vykstantys ir dažnai skubios reakcijos reikalaujantys pokyčiai tapo įprastu dalyku šiuolaikinėje verslo aplinkoje. Šie pokyčiai įtakoja ne tik patį verslą, bet ir jį palaikančias informacines sistemas. Siekiant neatsilikti nuo gyvenimo ir išlikti konkurencingiems, tiek verslo procesai, tiek informacinės sistemos turi būti nuolat modifikuojami, atnaujinami, adaptuojami prie naujos aplinkos. Kuriant verslo informacines sistemas visuomet buvo išskiriami trys pagrindiniai elementai: duomenys, procesai ir taisyklės. Pirmuosius du jau įprasta integruoti remiantis objektiniu programavimu, tuo tarpu taisyklės vis dar dažnai neišskiriamos, ir yra išbarstytos po visą programų kodą.

Veiklos taisyklės kinta gan dažnai. Šie pokyčiai negali būti efektyviai atspindimi informacinėse sistemose, jeigu veiklos logika yra tiesiogiai užprogramuota išeties tekstuose.

Atlikus eksperimentinį sukurtos veikos taisyklių valdymo sistemos tyrimą nustatyta, kad sistema palengvina vartotojams veiklos taisyklių įvedimą bei modifikavimą, sumažina klaidų tikimybę ir leidžia greičiau pritaikyti turimą programinę įrangą prie veiklos pokyčių. Taip pat numatytos sistemos plėtimo galimybės.

Naudota programinė įranga

Sistemai projektuoti buvo naudota *Rational Rose Enterprise*, bei *MS Visio* programinė įranga. Visi programavimo darbai buvo atlikti *ABAP* kalba *SAP* aplinkoje. *SAP* naudojama duomenų bazė – *MS SQL 2000*.

2. VEIKLOS TAISYKLIŲ VALDYMO SPRENDIMŲ ANALIZĖ

2.1. Veiklos taisyklės

Veiklos taisyklės arba **veiklos taisyklių rinkiniai** aprašo veiksmus, apibrėžimus ir apribojimus taikomus organizacijoje siekiant jos užsibrėžtų tikslų. Pavyzdžiui veiklos taisyklė gali teigti, kad nuolatinių klientų kredito likutis netikrinamas, arba gali suskirstyti skolininkus pagal mokumą, arba aprašyti pageidaujamus tiekėjus ir tiekimo grafikus. Šios taisyklės gali būti naudojamos siekiant geriau įvykdyti kompanijos tikslus, padėti vadovybei bendrauti su darbuotojais, būti pagrindu bendraujant su trečiomis šalimis, nurodyti kokybės kriterijus, būti pagrindu automatizuojant gamybą, padėti suprasti kompanijos veiklą, joje naudojamus standartus.

Svarbu suprasti skirtumą tarp strategijos/taktikos ir veiklos taisyklių. Veikos taisyklės yra tai, kuo strategija ir taktika yra paremtos. Taisyklės tiesiog pasako organizacijai, ką reikia daryti, tuo tarpu strategija ir taktika sukuria veiklos taisykles tokias, kad jų laikantis būtų sėkmingai pasiekti organizacijos tikslai. Kaip analogiją ryšiui tarp veikos taisyklių ir strategijos galima pateikti ryšį tarp kelio vedančio iš taško A į tašką B ir žemėlapiu, kuriame tas kelias atvaizduotas.

Veikos taisyklės gali apimti įvairiausias sritis:

- rinkodaros strategija;
- kainodara;
- ryšių s klientais valdymas;
- personalo valdymas;
- kontrolė;
- darbų sekų valdymas [19].

2.1.1. Veiklos taisyklių surinkimas

Organizacija gali nuspręsti iš anksto aprašyti savo veiklos principus taisyklių duomenų bazėje. Pavyzdžiui galima nusišamdyti konsultantus, kad šie atliktų organizacijos procesų ir struktūros analizę ir dokumentuotų visus tuo metu naudojamus standartus ir metodus.

Tačiau dažniausiai veiklos taisyklės yra identifikuojamos įvairių projektų metu, formalių reikalavimų surinkimo etape. Tokiu atveju veiklos taisyklių surinkimas yra atsitiktinis. Tokie

projektai kaip, naujo produkto paleidimas į gamybą, gali suformuoti visai naują veiklos taisyklių rinkinį. Pavyzdžiui šio naujo produkto atsiradimas gali priversti vadovybę peržiūrėti kompanijos tikslus. Taigi naudojant atsitiktinį taisyklių surinkimo būdą atsiranda rizika, kad aprašytasis veiklos taisyklių rinkinys bus neišsamus ar netgi turės konfliktuojančių taisyklių tarp skirtingų padalinių.

2.1.2. Praktikoje naudojamos programos ir iškylančios kliūtys

Nepaisant daugelio programinės įrangos gamintojų, konsultantų ir tyrimų institutų siūlančių įvairius veiklos taisyklių sprendimus, veiklos taisyklės dažniausiai renkamos tik tuomet, kai to pareikalauja įstatymai arba bandant žengti pirmąjį žingsnį link procesų automatizavimo ar tiesiog kaip trumpalaikė pagalba inžinierių komandai. Pagrindinė tokios situacijos priežastis yra tai, kad veiklos taisyklių rinkinio priežiūrai dažniausiai reikalingi papildoma darbo jėga, taigi ir papildomos išlaidos. Šios išlaidos yra tuo didesnės, kuo dinamiškesnė ir mažiau nusistovėjusi yra veikla ir jos aplinka, pavyzdžiui naujose kompanijose. Kitas faktorius mažinantis veiklos taisyklių naudojamumą yra darbuotojų motyvacijos stoka išsiskirti su didžiausią vertę organizacijai turinčiu turtu – veiklos taisyklių žinojimu. Trečiasis faktorius yra technologijos, tinkamos veikos taisyklėms palaikyti. Klasikinis sprendimas yra veikos taisyklių valdymo sistemos, tačiau jos vis dar išlieka daugiau universitetų tyrimų objektai nei realiai naudojamos sistemos.

Programinė įranga aromatizuoja veiklos taisykles remiantis veiklos logika. Kartais sąvokos veiklos taisyklės ir veiklos logika yra supainiojamos ir naudojamos kaip tą patį reiškiančios, tačiau iš tikro pirmoji nusako veiklos procesus plačiąja reikšme, o antroji – kompiuterizuotus veikos procesus. Identifikuoti veiklos taisykles naudinga netgi tuomet, kai neketinama automatizuoti jokio veiklos proceso.

2.1.3. Svarbiausios veiklos taisyklių savybės

- Deklaratyvumas. Veiklos taisyklė – tai tam tikro fakto apie organizaciją deklaravimas. Tai yra bandymas aprašyti organizacijos veiklos procesus, o ne juos nustatyti. Todėl sakoma, kad veiklos taisyklės yra atrandamos, o ne sukuriamos.
- Vienareikšmiškumas. Veiklos taisyklės gali būti tikrai arba visiškai įvykdytos, arba visiškai neįvykdytos, negali būti tarpinių variantų. Tarkime turime taisyklę, kuri teigia, kad lėktuvo keleivis gali pasikeisti savo dviejų krypčių ekonominės klasės bilietą į verslo klasę, jeigu yra laisvų bilietų ir, jeigu keleivis sumoka kainų skirtumą. Reiškia

bilieta galima arba keisti arba ne, t.y. negalima vienai kryptčiai klase pakeisti, o kitai ne.

- Aiški, nepriklausoma konstrukcija. Reikia atskirti dalykus, kurie aprašo veiklą ir veiklos strategiją ir taktiką. Veiklos taisyklės turi būti paprastos ir kategoriškos konstrukcijos, jose negali būti sudėtingų, ciklinių priklausomybių.
- Natūrali kalba. Pageidautina, kad šalia formalaus veiklos taisyklės aprašymo būtų ir jos apibūdinimas natūralia kalba. Taip ji bus suprantama platesniam ratui žmonių.
- Orientacija į pagrindinę veiklą. Veiklos taisyklės turi būti orientuotos į darbuotojus dirbančius pagrindinės organizacijos veiklos srityje, ne į pagrindinę veiklą palaikančias technines tarnybas.
- Pagrindinės veiklos nuosavybė. Už veiklos taisykles turi būti atsakingi pagrindinės veiklos darbuotojai. Taigi, taisykles surinkti, įvesti, modifikuoti turėtų pagrindinės veiklos darbuotojai, o ne techninės tarnybos.

2.1.4. Formali veiklos taisyklių specifikacija

Jeigu situacija to reikalauja, veiklos taisyklės gali būti aprašytos labai formalia kalba. Pavyzdžiui to gali reikėti aprašant techninę veiklą, tokią kaip konstruktorių organizacijos veikos procesus. Kitas pavyzdys, kur formalus specifikavimas neišvengiamas – kuomet siekiama aprašyti patį taisyklių surinkimo procesą. Tam naudojamos formalios kalbos: UML, *Z notation*, *Business Rules Markup Language*, BPEL ir BPMN.

Idealu būtų, jei būtų galima automatiškai sudaryti formalią specifikaciją iš natūralios kalbos taisyklių aprašų.

2.2. Veiklos taisyklių valdymo sistemos

Veiklos taisyklių galima rasti kiekviename procese kiekvienoje organizacijoje. Veiklos taisyklių valdymo sistema (VTVS) yra skirta centralizuoti organizacijos taisyklių atradimą, saugojimą, modifikavimą ir vykdymą. VTVS naudojimas gali leisti padidinti automatizavimo laipsnį, sutrumpinti reakcijos į pokyčius laiką ir sumažinti palaikymo kaštus. Saugant veiklos taisykles nepriklausomai nuo susijusių duomenų ir atskirai nuo programų naudojamų palaikyti veiklos procesus galima jas naudoti pakartotinai, palengvina procesų valdymą ir padidina realiai vykstančių procesų atitikimą numatytiems. Be to tai leidžia keisti taisykles nekeičiant programų kodo.

Organizacijos gan dažnai turi į servisą orientuotą programų architektūrą. Tačiau tokiu atveju yra ypatingai sunku pasiekti, kad visas programų ūkis darniai veiktų vykdant priimtus veiklos sprendimus. Dar sunkiai tokioje architektūroje yra padaryti pakeitimus, pasikeitus veiklos sprendimams.

Veiklos taisyklių valdymo sistemos yra labai efektyvus mechanizmas įgyvendinant veiklos sprendimų logiką į servisą orientuotoje programų architektūroje. VTVS veikimo esmė yra ta, kad visos organizacijoje naudojamos programos naudoja veiklos taisykles, saugomas vienoje saugykloje ir pilnai aprašančias veiklos procesus. VTVS gali būti ne tik į servisą orientuotos architektūros programų jungiamoji grandis, bet ir pagalbiniis įrankis sprendimų priėmimo procese.

Veiklos taisyklių valdymo pagrindinis tikslas yra sprendimų priėmimo proceso ir technikos (programų įvestis / išvestis, valdymo kodas, išėjties tekstai) atskyrimas. Naudojantis vienintele VTVS galima atlikti visus su veiklos taisyklėmis susijusius veiksmus: taisyklių saugojimas, valdymas ir sprendimų įdiegimas į programinę įrangą.

Išskiriami keturi svarbiausi pranašumai įgyjami naudojant veiklos taisyklių valdymo sistemą bet kurioje organizacijoje:

- net ir naudojant įvairias operacines sistemas ir programas, išvengiama pakartotinio sprendimų logikos diegimo kiekvienoje iš jų;
- galima keisti sprendimų logiką, nekeičiant nei operacinių sistemų nei programinės įrangos kodo, tokiu būdu ženkliai sumažinant testavimo kaštus po diegimo;
- galima aprašyti sudėtingą sprendimų logiką. To negalima padaryti naudojant standartinius sprendimus: lenteles ar medžius;
- atsiranda galimybė automatizuoti daug procesų, kuriuos anksčiau teko daryti rankiniu būdu.

Daug programų naudojamų įvairiose veiklos srityse naudoja VTVS sprendimų logikai valdyti. Organizacijos naudoja VTVS įvairiems tikslams:

- paraiškoms valdyti;
- apmokėjimų patvirtinimams valdyti;
- duomenims tikrinti;
- draudimui patvirtinti ir polisams įkainoti;
- problemų sprendimo procedūroms;

- produktų sudėties pakeitimams patvirtinti;
- atitikimui įstatymams valdyti [15].

Veiklos taisyklių valdymo sistema paprastai susideda iš šių komponentų:

Taisyklių arba žinių duomenų bazė. Taisyklės aprašo turimas žinias reakcinėmis taisyklėmis. Kitų tipų taisyklės naudojamos retai, kadangi daugelis VTVS palaiko tik reakcines taisykles. Taigi, kitų tipų taisykles tenka išreikšti reakcinėmis taisyklėmis.

Taisyklių įvedimo sąsaja. Programos komponentas skirtas taisyklėms įvesti į duomenų bazę.

Darbinė atmintis. Darbinėje atmintyje saugomi objektai, kuriems bus taikoma taisyklė. Taisyklės vykdymo metu darbinės atminties turinys gali keistis dėl vykdomų veiksmų.

Sulyginimo variklis. Sulyginimo procesas prasideda taisyklės atitinkančios darbinę atmintį išrinkimu iš duomenų bazės. Paprastai vienam darbinės atminties turiniui taikoma daugiau nei viena taisyklė, todėl sulyginimo variklis sudaro išrinktų taisyklių taikymo grafiką. Šio grafiko sudarymas yra konfliktų valdymo uždavinys, kuriam spręsti yra sudaryta daug algoritmų, iš kurių žinomiausias yra *RETE* [12].

2.2.1. Rule Markup Language (RuleML)

Veiklos taisyklių panaudojimas informacinėse sistemose nuolat auga. Pasak Gartner, 2007-aisiais metais augimas prognozuojamas iki 80 %. Veiklos taisyklės naudojamos tokiose įvairiose srityse kaip finansai, sveikatos apsauga, e. komercija, tinklo paslaugos. Nepaisant to, yra didelis poreikis standartizuoti taisyklių užrašymą tarp įvairių platformų [14].

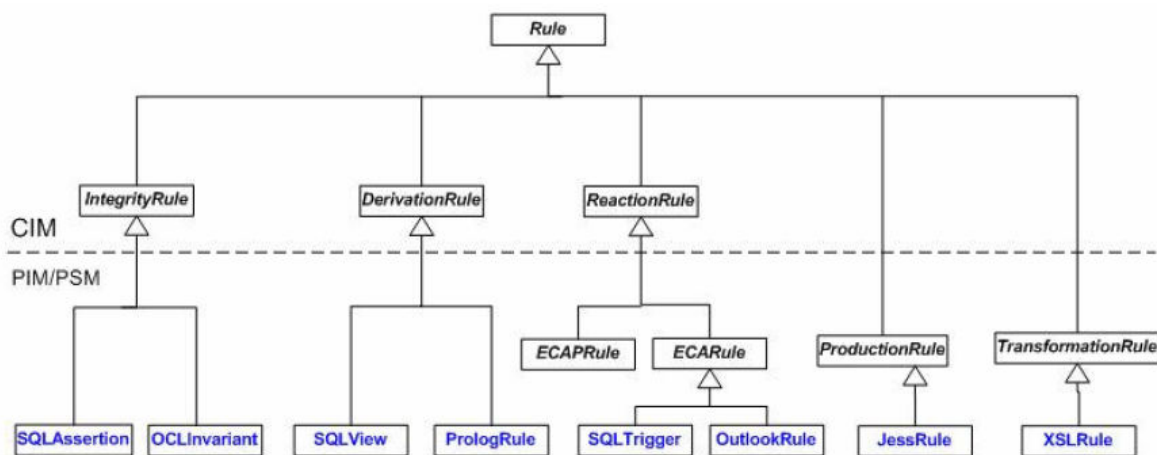
Tarptautinės *Rule Markup* iniciatyvos tikslas yra išplėtoti *RuleML* iki kanoninės tinklo kalbos veiklos taisyklėms užrašyti naudojant XML kalbą.

RuleML yra paremta daugeliu žinomų sistemų, tarp kurių:

- BRML (*Business Rules Markup Language*, IBM);
- RFML (*Relational-Functional Markup Language*, DFKI);
- XRML (*Extensible Rule Markup Language*, KAIST) [17].

Yra sukurta abstrakti *RuleML* kalbos sintaksė. Jai užrašyti pasirinktas MOF (*Meta-Object Facility*) modelis, kuriame nenaudojami jokie specialūs simboliai. Abstrakti sintaksė

apibrėžia pagrindinius taisyklių paskirties aspektus ir jų semantines kategorijas.



1 pav. Veiklos taisyklių kategorijos

Apibrėžimo (*Integrity*), išvedimo (*derivation*) ir reakcinės (*reaction*) kategorijos laikomos svarbiomis tiek teoriniu (taisyklių netaikant kompiuterizavimui), tiek praktiniu požiūriu (pritaikius kompiuterizavimui). Tuo tarpu struktūrinės (*production*) ir transformacijų (*transformation*) taisyklės laikomos svarbiomis tik praktiniu požiūriu.

- **Apibrėžimo taisyklės** naudojamos konstantoms apibrėžti.
- **Reakcinės taisyklės** turi tokią struktūrą: sąlyga – veiksmas.
- **Išvedimo taisyklės** naudojamos išgauti (išrinkimo, išvadų darymo būdais) informaciją iš turimos informacijos.
- **Transformacijų taisyklės** naudojamos apibrėžti skaičiavimo formules, turinio sudėties transformacijas ir pan.
- **Struktūrinių taisyklių** paskirtis yra apibrėžti taisyklių tarpusavio priklausomybes ir tokiu būdu sudaryti taisyklių hierarchiją [11].

2.2.2. Rinkoje egzistuojančių VTVS apžvalga

Šiuo metu rinkoje yra nemažai įrankių, skirtų veiklos taisyklėms valdyti. Tokie įrankiai paprastai yra brangūs, dažniausiai juos naudoja bankai, draudimo kompanijos bei kitos organizacijos, kurioms tokių įrankių naudojimas – neišvengiamas dalykas.

2.2.2.1. Blaze Advisor

Remiantis Gartner 2003 metų ataskaita, *FairIsaac* kompanijos produktas *Blaze Advisor* yra rinkos lyderis veiklos taisyklių srityje.

Blaze Advisor pasižymi šiomis savybėmis:

- XML pagrindu sukurta daugiavartotojiška saugykla (*repository*), įgalinanti programuotojų grupes dirbti koordinuotai, naudoti bendrus taisyklių aprašus;
- veiklos taisyklėms užrašyti naudojama panaši į anglų, struktūrizuota taisyklių kalba (SRL). Ji leidžia programuotojams lengvai ir greitai užrašyti taisykles. Taip pat, nesudėtinga sukurti sąsajas vartotojams, kur jie patys nesunkiai gali įrašyti naujas ar koreguoti egzistuojančias taisykles;
- taisyklių skaičius neribojamas;
- galimybė dirbti su nepilnais duomenimis (tai dažnai pasitaiko realiame pasaulyje);
- saugoma taisyklių versijavimo istorija;
- įtraukti taisyklių šablonai, palengvinantys naujų taisyklių užrašymą [5].

2.2.2.2. JRules

JRules – kompanijos ILOG veiklos taisyklių valdymo sistema, skirta JAVA aplikacijoms. Tai viena sparčiausių rinkoje siūlomų sistemų. Ji specialiai pritaikyta objektiniam programavimui.

JRules pasižymi šiomis savybėmis:

- naudoja pačios kompanijos ILOG sukurtą taisyklių užrašymo kalbą;
- taisyklės ir jų metaduomenys saugomi vienoje saugykloje (*repository*);
- turi įrankius taisyklėms kurti ir redaguoti [6].

2.2.2.3. Rules4J

Rules4J – XML pagrindu sukurta Java taisyklių valdymo sistema. Ji buvo sukurta kaip palyginti pigi ir paprasta naudoti alternatyva kitoms galingoms taisyklių valdymo sistemoms. Ji pritaikyta veikti J2EE ir J2SE architektūrose.

Pagrindinės *Rules4J* savybės:

- taisyklės užrašomos XML failų aibėje – nereikia mokytis specialios kalbos;
- turi galingą taisyklių redagavimo įrankį;
- suderinama su *JSR-94* [13].

2.2.3. VTVS kiekybinis ir kokybinis palyginimas

1 lentelė. VTVS palyginimas

Savybė	<i>JRules</i>	<i>Blaze Advisor</i>	<i>Rules4J</i>
Lanksti taisyklių aprašymo kalba	Taip	Taip	Taip
Sprendimų lentelių palaikymas	Taip	Taip	Taip
Sprendimų medžių palaikymas	Taip	Taip	Ne
Klaidų paieška	Taip	Taip	Taip
Taisyklių optimizavimas	Taip	Taip	Ne
Java API palaikymas	Taip	Taip	Taip
Žiniatinklio taisyklių palaikymas	Taip	Taip	Taip
Versijavimo palaikymas	Taip	Taip	Ne

Išanalizavus trečiame skyriuje aptartas veiklos taisyklių valdymo sistemų realizacijas, nesunku pastebėti jų skirtumus ir panašumus:

- naudojamos skirtingos, pačių kompanijų sukurtos veiklos taisyklių užrašymo kalbos;
- taisyklės įrašomos bendroje saugykloje;
- pateikiamos priemonės, palengvinančios taisyklių kūrimą ir redagavimą.

Gartner, įvertinusi sistemas pagal žemiau pateiktus kriterijus, pripažino *FairIsaak* kompanijos produktą *Blaze Advisor* rinkos lyderiu.

Gartner vertinimo kriterijai:

- vizijos išbaigtumas:
 - technologija – produkto kokybė;
 - rinka – užimama vieta rinkoje;
 - viešieji ryšiai – prisistatymas rinkoje per reklamą, interneto svetaines, pristatymus;

- išorinės investicijos – išorinių investicijų kiekis ir tvirtumas;
- galimybės veikti:
 - produktas – produkto sudėtis, kaina, diegimo ir naudojimo paprastumas;
 - priežiūra ir paslaugos – paslaugų kokybė po diegimo;
 - valdymas – vadovų komandos patirtis ir stiprumas [7].

2.3. Problemos analizė

Lietuvos muitinės ERP sistemoje naudojama labai daug įvairių tipų dokumentų, kurie visi turi tuos pačius laukus ir yra saugomi vienoje duomenų bazės lentelėje. Vienas tipas nuo kito atskiriamas pagal keletą laukų kombinacijos reikšmes. Taip pat, egzistuoja taisyklės, siejančios vienus dokumentus su kitais. Kuriant ir palaikant Lietuvos muitinės informacinę sistemą, daugybėje vietų neišvengiamai tenka programuoti vieno ar kito tipo dokumentų išrinkimą bei vieno dokumentų informacijos susiejimą su kitų dokumentų informacija. Ši procesą dar labiau apsunkina tai, kad keičiantis įstatymams atsiranda nauji dokumentų tipai, naikinami ar modifikuojami egzistuojantys.

Šiuo metu veiklos taisyklės jau yra pradėtos naudoti. Taisyklės yra saugomos reliacinės duomenų bazės lentelėse. Šiuo metu nėra jokios vartotojo sąsajos, palengvinančios taisyklių įvedimą. Vienintelis būdas įvesti naują taisyklę – užrašyti ją tiesiai lentelių laukuose. Toks darbas yra nelengvas ir reikalaujantis daug laiko sąnaudų net puikiai išmanančiam šių lentelių struktūrą programuotojui. Jau užrašytų taisyklių paieška ir modifikavimas taip pat reikalauja daug kruopštaus darbo ir laiko.

Užsakovas pageidauja, kad būtų sukurta grafinė aplinka veiklos taisyklėms valdyti, supaprastinti taisyklių įvedimą, paiešką ir modifikavimą. Reikalaujama, kad sukurtoji grafinė aplinka būtų lengvai išplečiama. Egzistuojančios duomenų bazės struktūros reikalaujama nekeisti.

Tolimesnis taisyklių panaudojimas programuojant vykdomas pasinaudojant funkcijų rinkiniu, kuris iš aprašytų taisyklių suformuoja reikiamą rezultatą. Funkcijų rinkinį užsakovas prižiūrės pats. Reikalaujama jame nedaryti jokių pakeitimų.

2.4. Priimti sprendimai

Kurti VTVS ERP sistemoje

Nagrinėtos veiklos taisyklių valdymo sistemos yra pakankamai universalios, jas galima pritaikyti dirbti su daugeliu žinomų verslo valdymo programų. Daugelyje apžvalgų, konferencijų pranešimų bei pačių VTVS gamintojų reklaminėje medžiagoje ypač akcentuojamas išorinių VTVS naudingumas, kai kompanijoje daugelis skirtingų programų sistemų į bendrą visumą jungiamos remiantis į servisus orientuota architektūra (SOA). Taip yra todėl, kad SOA atveju, neturint veiklos taisyklių valdymo sistemos, atsiradus vienam proceso pasikeitimui, gali tekti rankiniu būdu modifikuoti kiekvieną programą. SOA atveju iš tikrųjų vienintelis sprendimas yra išorinė VTVS, nes norint sukurti vidinę, ją tektų kurti kiekvienoje naudojamose programose. Dar viena gera išorinių VTVS savybė – jos turi galimybę veikti serveryje ir turi gerą prisijungimo teisių valdymą.

Kai organizacijoje naudojama tik viena programinė įranga, pavyzdžiui, visas veiklas apimanti ERP, diegti išorinę VTVS gali neapsimokėti. Tokiu atveju jau reiktų palyginti naujos VTVS kūrimo ERP ribose privalumus su išorinės VTVS trūkumais. Visų pirma, išorinė VTVS visuomet veiks lėčiau nei vidinė. Antra, ERP sistemos programuotojai turi papildomai išmokti dirbti su išorine VTVS. Trečia, reikia pasverti išorinės VTVS diegimo ir priežiūros kaštus ir vidinės VTVS kūrimo kaštus.

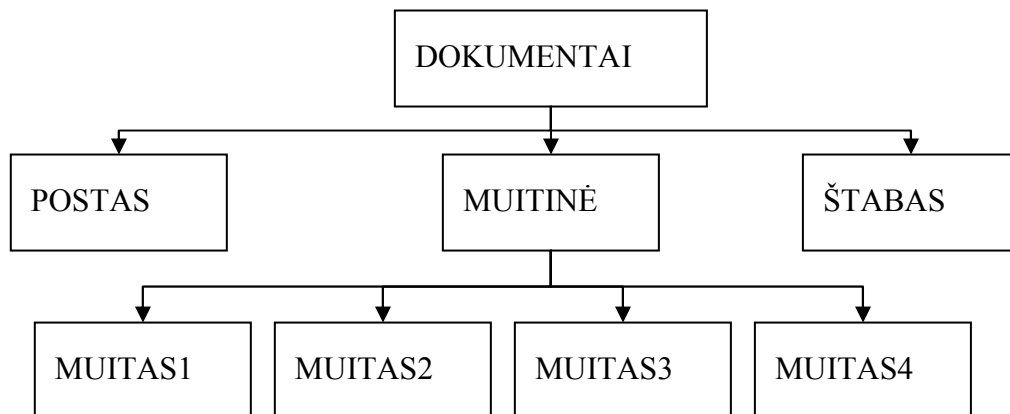
Mūsų atveju buvo nuspręsta kurti vidinę veiklos taisyklių valdymo sistemą SAP R/3 sistemoje. Toks sprendimas priimtas dėl šių priežasčių:

- jau yra sukurta duomenų bazė veiklos taisyklėms saugoti bei funkcijos joms naudoti;
- nauja programinė įranga sukeltų daug rūpesčių: reikalinga papildoma technika (serveriai), papildomos sistemos išmanymas;
- būtų sudėtinga naudoti išorines VTVS iš ABAP kalbos. Teiktų pakeisti daug programos teksto ir keisti programavimo metodiką, norint sukurti sąsają su išorine VTVS;
- išorinės VTVS taisyklių įvedimas nėra pakankamai patogus. Tuo tarpu kuriamojoje VTVS įvedimą bus galima prisitaikyti konkrečiai savo reikmėms;

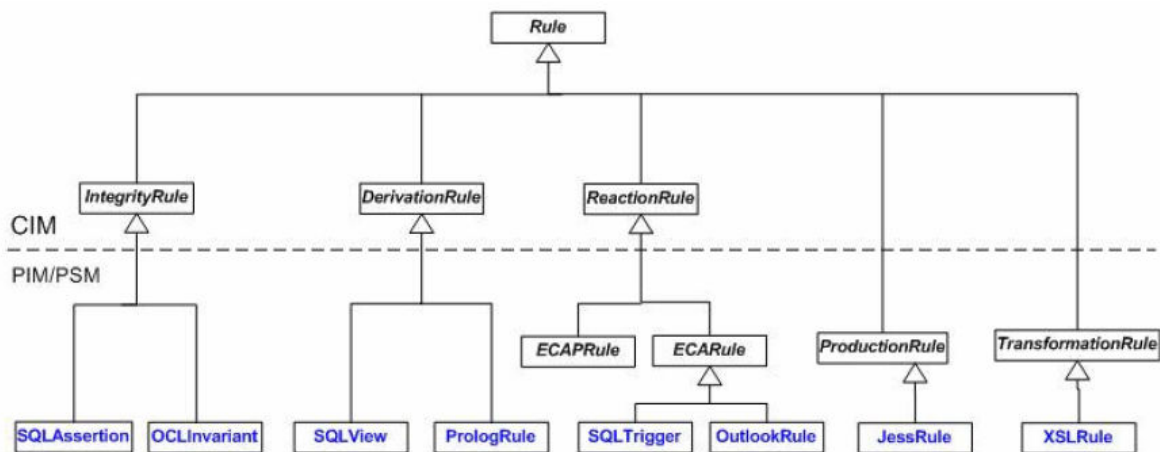
Kurti VT hierarchijų valdymo sistemą

Remiantis atlikta literatūros ir egzistuojančių sprendimų analize, priimtas sprendimas sudaryti veiklos taisyklių hierarchiją. Sudarant hierarchiją, ypatingas dėmesys turi būti skirtas veiklos taisyklių valdymo sistemos universalumui išlaikyti, t.y. siekti, kad pati sistema neapribotų hierarchijos sudarymo kriterijų ir netgi tam pačiam taisyklių rinkiniui leistų sudaryti neribotą skaičių skirtingų hierarchinių medžių. Skirtingų hierarchijų kriterijų pavyzdžiai gali būti:

- pagal dokumentų tipus (2 pav.). Smulčiausią šaką sudaro tik su konkrečiu dokumentu susiję taisyklės. Stambesnėmis šakomis dokumentai jungiami į dokumentų tipus;
- pagal taisyklių semantines kategorijas (remiantis *RuleML*) (3 pav.). Smulčiausiai šakai priklauso taisyklės, atliekančios tą pačią funkciją SQL sakinyje. Stambesnės šakos – anksčiau minėtos *RuleML* modelyje:
 - apibrėžimo;
 - išvedimo;
 - reakcinės;
 - struktūrinės;
 - transformacijų.



2 pav. VT hierarchija pagal dokumentų tipus



3 pav. VT hierarchija pagal taisyklių semantines kategorijas

VT hierarchija turi būti saugoma toje pačioje duomenų bazėje kaip ir veiklos taisyklės. Su hierarchija susiję įrašai turi neiškreipti ten saugomų taisyklių ir nesuardyti bendros idėjos. Hierarchijos įrašai turi būti lengvai atskiriami nuo kitų taisyklių, tačiau ir patys hierarchijos įrašai turi turėti galimybę būti traktuojami kaip veiklos taisyklės pagal tą patį principą, kaip ir tikrosios taisyklės.

2.5. Analizės išvados

- Atlikus literatūros analizę, identifikuotos 6 svarbiausios veiklos taisyklių savybės, kurias turi būti siekiama kiek įmanoma labiau išpildyti, kuriant bet kokią VTVS.
- Aprašyti pagrindiniai veiklos taisyklių valdymo sistemų komponentai.
- Įvertintos trys populiariausias veikos taisyklių valdymo sistemos, aprašyti jų privalumai ir trūkumai..
- Remiantis atlikta problemos analize bei rinkoje esančių produktų įvertinimu, buvo priimtas sprendimas kurti VTVS SAP sistemos viduje.
- Buvo nuspręsta patobulinti egzistuojantį veiklos taisyklių modeli, sukuriant VT hierarchijų valdymo sistemą.

3. VEIKLOS TAISYKLIŲ HIERARCHIJOS MODELIS ERP SISTEMAI

3.1. Sistemos paskirtis

Sistemos tikslas – sukurti grafinę vartotojo sąsają taisyklėms įvesti ir atvaizduoti. Grafinė sąsaja turi:

1. užtikrinti įvedamų taisyklių sintaksinį teisingumą;
2. palengvinti taisyklių paiešką;
3. atvaizduoti taisyklių tarpusavio sąsają.

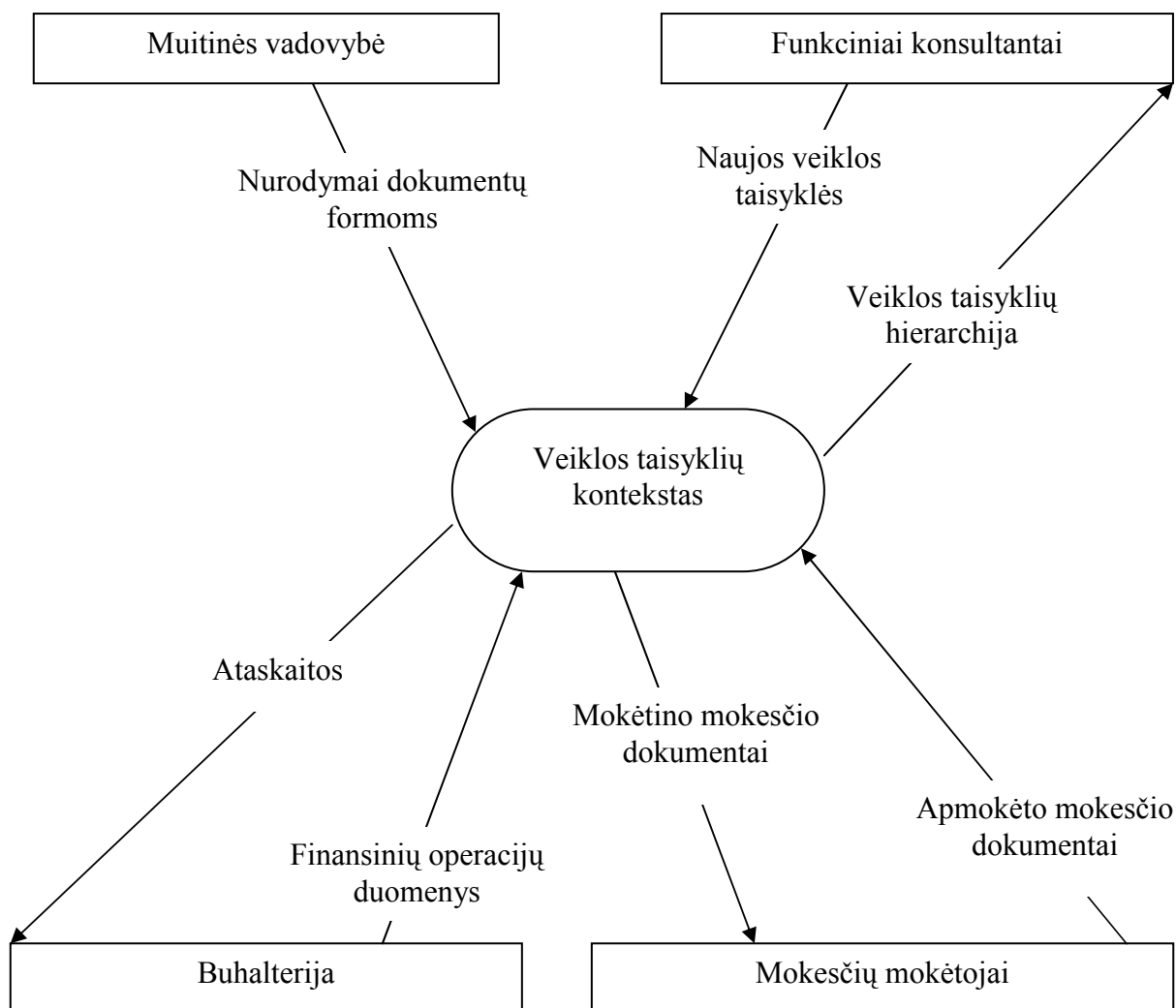
Produkto užsakovas – UAB „ERP“. Visos jau sukurto veiklos taisyklių karkaso autorinės teisės priklauso UAB „ERP“. Naujosios VT karkaso grafinės sąsajos autorinės teisės taip pat priklausys šiai įmonei.

Informacinės sistemos vartotojas:

- vartotojo sprendžiami uždaviniai (atliekamos funkcijos):
 1. veiklos taisyklių aprašymas;
 2. veiklos taisyklių klasifikavimas;
 3. karkaso funkcijų naudojimas informacinės sistemos moduluose;
- patirtis dalykinėje srityje – vartotojas, turintis programuotojo išsilavinimą ir patirties darbe su *SAP* sistema;
- patirtis informacinėse technologijose – informacinių sistemų programuotojo darbo patirtis;
- papildomos vartotojo charakteristikos:
 1. veiklos taisyklių išmanymas;
 2. anglų kalbos žinios.

3.2. Veiklos sudėtis

Veiklos konteksto diagrama pateikta 4 pav.



4 pav. Veiklos konteksto diagrama

Veiklos įvykių sąrašas pateikiamas 2 lentelėje.

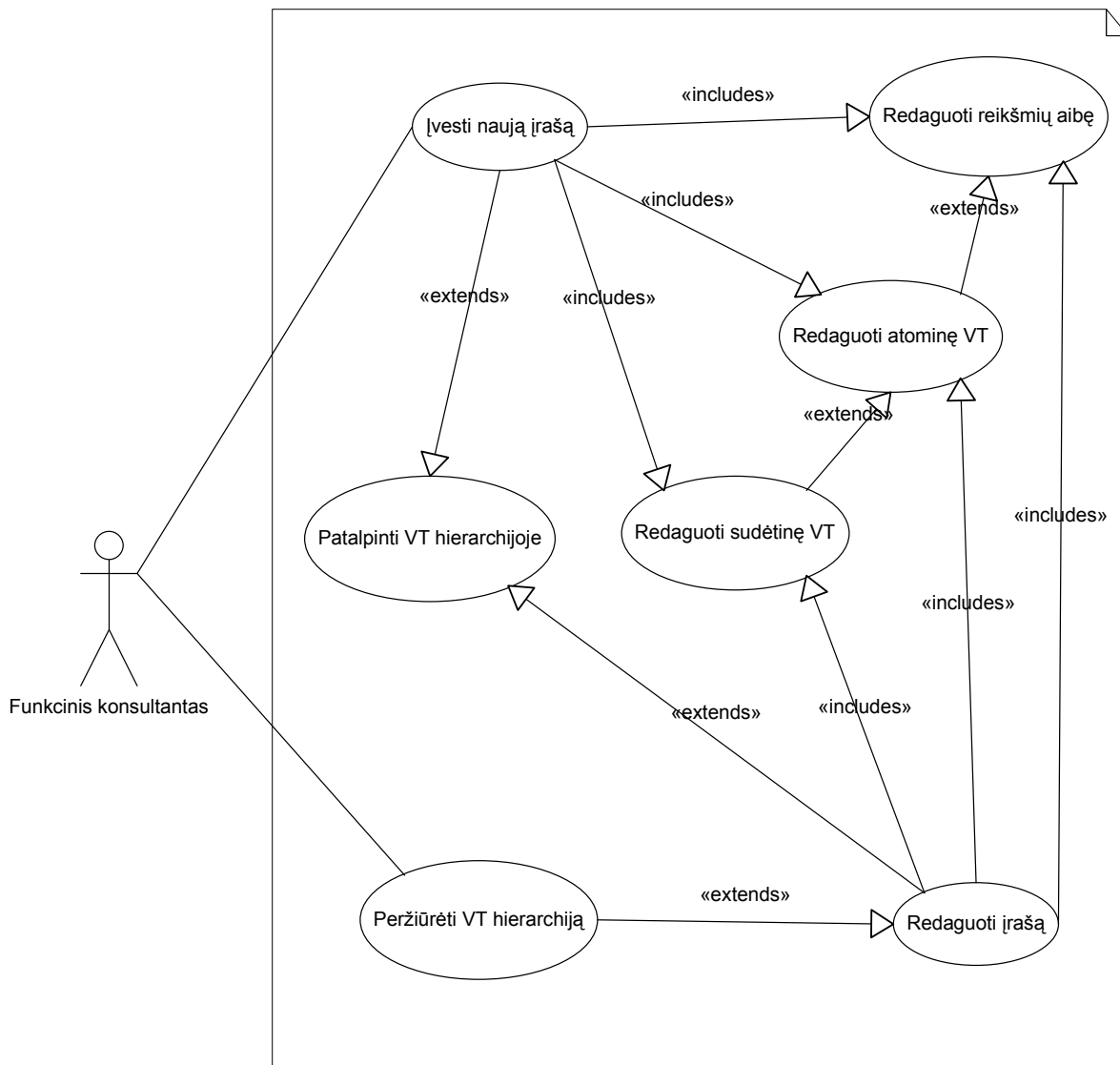
2 lentelė. Veiklos įvykių sąrašas

Eil. Nr.	Įvykio pavadinimas	Įeinantys/išeinantys informacijos srautai
1	Muitinės vadovybė duoda nurodymus dokumentų formoms	Nurodymai dokumentų formoms (in)
2	Funkciniai konsultantai įveda naujas veiklos taisykles	Naujos veiklos taisyklės (in)
3	Funkciniai konsultantai revizuoja veiklos taisyklių hierarchiją	Veiklos taisyklių hierarchija (out)
4	Mokesčių mokėtojai gauna užklausas mokėtiniems mokesčiams	Mokėtino mokesčio dokumentai (out)
5	Mokesčių mokėtojai pristato apmokėtų mokesčių dokumentus	Apmokėto mokesčio dokumentai (in)
6	Buhalterija perduoda finansinių operacijų duomenis	Finansinių operacijų duomenys (in)

6	Buhalterija gauna ataskaitas apie įvykdytas mokesčių užklausas	Finansinių operacijų duomenys (in)
---	--	------------------------------------

3.3. Sistemos sudėtis

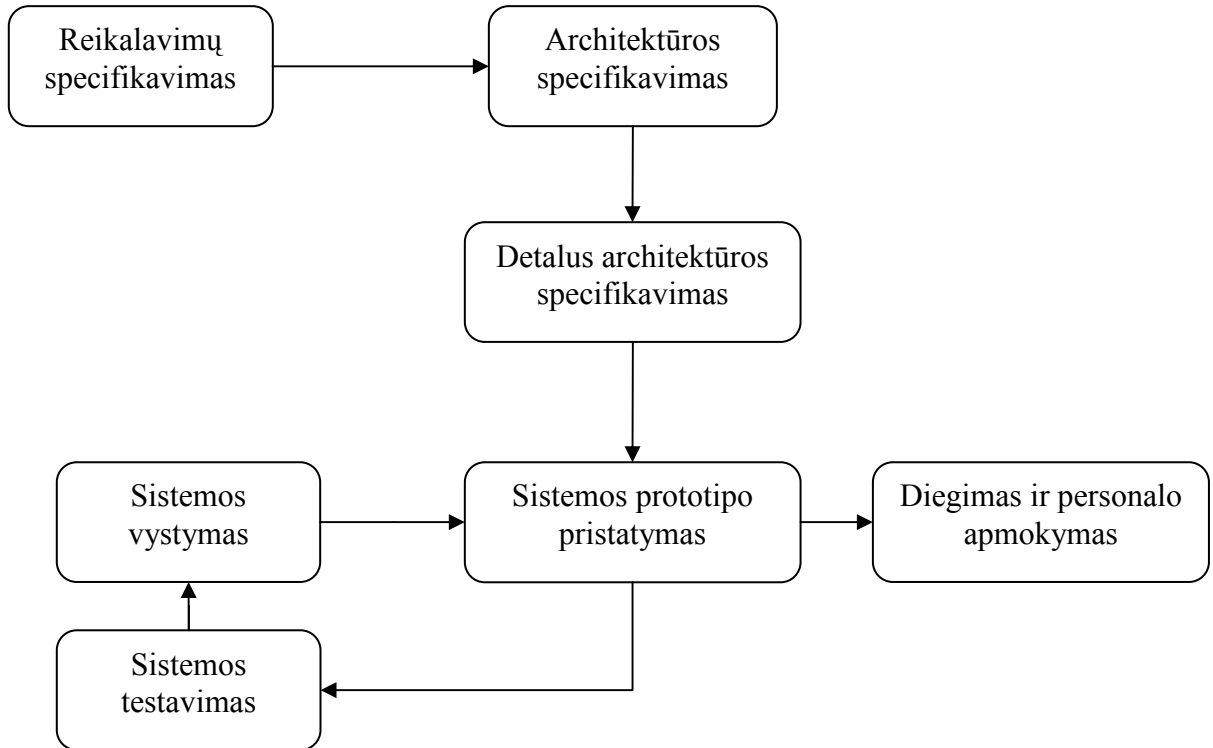
Ribas tarp sistemos ir vartotojo nusako panaudojimo atvejų diagrama. Kuriamojo sistemos modulio panaudojimo atvejų diagrama pateikiama 5 pav.



5 pav. Panaudojimo atvejų diagrama

3.4. *Sistemos pateikimo žingsniai*

Pagrindiniai sistemos kūrimo ir pateikimo žingsniai pateikiami 6 pav.



6 pav. Sistemos kūrimo žingsniai

3.5. *Architektūros specifikacija*

3.5.1. *Architektūros pateikimas*

Kuriamoje sistemoje panaudota paskirstytos kliento-serverio sistemos architektūra. Tai sąlygoja sistemos kūrimo aplinka – SAP WEB *Application Server*.

Sistemos architektūra detalizuojama naudojant įvairius vaizdus:

- **panaudojimo atvejų vaizdas** – pateikiami sistemos panaudojimo atvejai ir jų specifikacijos. Naudojami modeliavimo elementai: panaudojimo atvejų diagrama, panaudojimo atvejų specifikacijos;
- **sistemos statinis vaizdas** – pateikiamas sistemos išskaidymas į paketus. Aprašomi pagrindiniai sistemos komponentai. Naudojami modeliavimo elementai: komponentų diagrama, klasių diagrama;

- **sistemos dinaminis vaizdas** – detalizuojami panaudojimo atvejai. Naudojami modeliavimo elementai: sekų, bendradarbiavimo, būsenų ir veiklos diagramos;
- **išdėstymo vaizdas** – aprašoma techninės įrangos, kurioje sistema bus išdėstyta ir veiks, konfigūracija bei sistemos komponentai, išdėstyti atskiruose techninės įrangos mazguose. Naudojami modeliavimo elementai: komponentų diagrama, sistemos išdėstymo diagrama;
- **duomenų vaizdas** – pateikiamas duomenų bazės modelis ir lentelių laukų aprašymai.

3.5.2. Architektūros tikslai ir apribojimai

Bus kuriama ne atskira sistema, o jau egzistuojančios ir funkcionuojančios sistemos posistemis – veiklos taisyklių valdymo karkaso grafinė sąsaja. Taigi, pagrindiniai sistemos architektūros principai jau yra apibrėžti. Reikalaujama nemodifikuoti esamos architektūros.

Grafinė sąsaja bus kuriama naudojant *ABAP* kalbą. Ji naudos jau egzistuojančią duomenų bazę.

Reikalaujama projektuoti grafinę sąsają taip, kad būtų užtikrintas maksimalus jos plečiamumas, todėl, projektuojant sistemą, reikia kreipti ypatingą dėmesį kiekvieno komponento universalumui ir nepriklausomumui nuo kitų. Sukurta sistema neturi riboti esamos duomenų bazės plečiamumo.

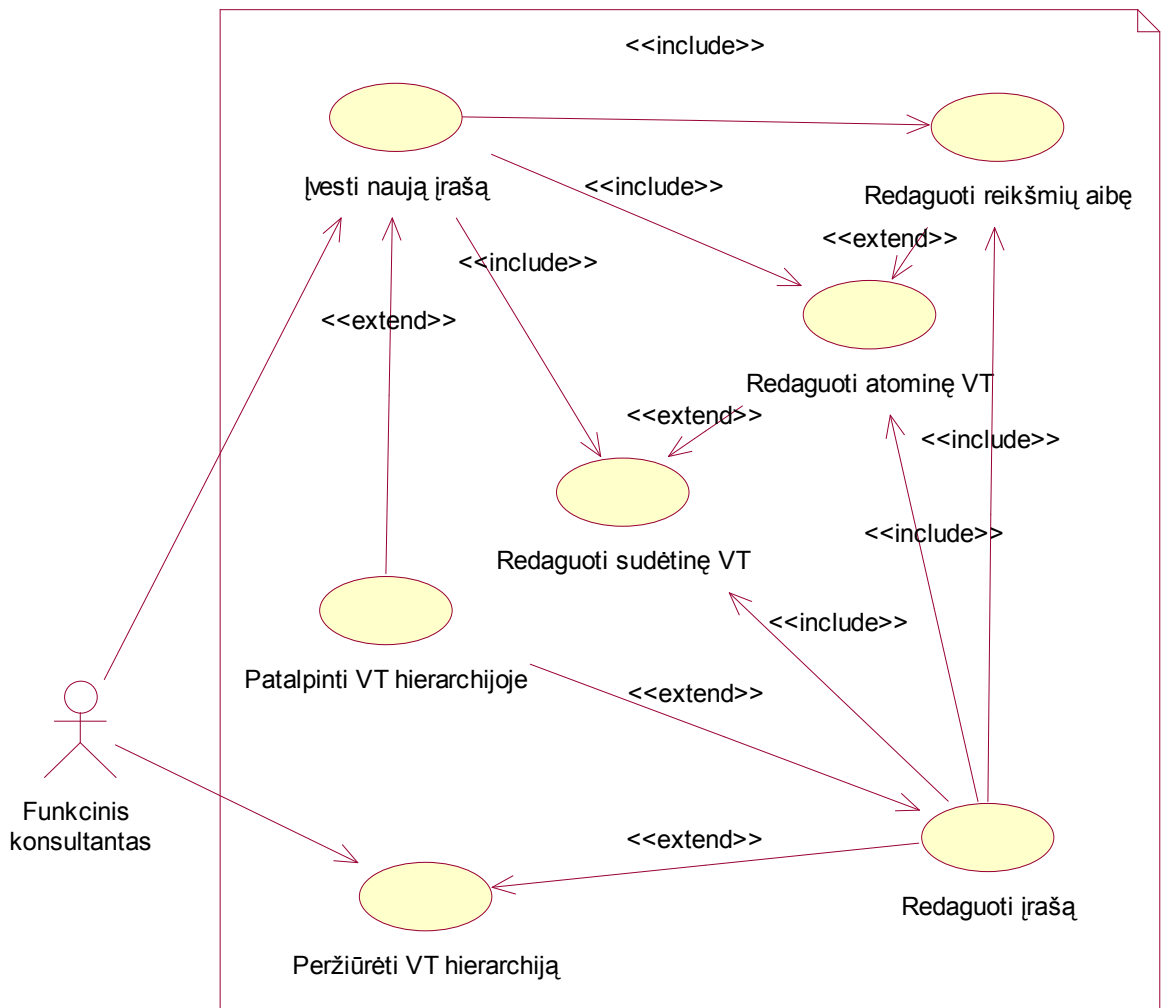
Nereikalaujama, kad grafinė sąsaja užtikrintų saugumą, kadangi saugumą užtikrina *SAP WEB Application Server*.

Sistema veikia *SAP WEB Application Server* aplinkoje, duomenų bazė saugoma *Microsoft SQL Server* duomenų bazių valdymo sistemoje. Sistemos projektavimas bus atliekamas naudojant *Rational Rose Enterprise Edition 2002* paketą.

3.5.3. Panaudojimo atvejų vaizdas

3.5.3.1. Panaudojimo atvejų diagrama

Kuriamojo sistemos modulio panaudojimo atvejų diagrama pateikiama 7 pav.



7 pav. Panaudojimo atvejų diagrama

3.5.3.2. Panaudojimo atvejų specifikacijos

3 lentelė. Panaudojimo atvejis „Įvesti naują įrašą“

1. Panaudojimo atvejis	Įvesti naują įrašą
Tikslas	Įvesti naują veiklos taisyklę arba reikšmių aibę
Aktoriai	Funkcinis konsultantas
Ryšiai su kitais PA	Apima panaudojimo atvejus: „redaguoti atominę taisyklę“, „redaguoti sudėtinę taisyklę“ ir „redaguoti reikšmių aibę“; gali būti išplėstas iki panaudojimo atvejo „patalpinti VT hierarchijoje“.
Nefunkciniai reikalavimai	2
Prieš-sąlygos	Nėra.
Sužadinimo sąlyga	Vartotojas iš meniu pasirenka naujo įrašo įvedimo komandą.
Po-sąlygos	Naujas įrašas įrašytas.
Pagrindinis scenarijus	Sistemos reakcija ir sprendimai
1. Pasirenkamas naujos VT įvedimas	1.1. Atidaroma naujo įrašo įvedimo forma.
2. Pasirenkamas panaudojimo atvejis: „redaguoti atominę taisyklę“, „redaguoti sudėtinę taisyklę“, „redaguoti reikšmių aibę“)	2.1. Sukuriamas naujas įrašas atitinkamoje lentelėje. 2.2. Pereinama į pasirinktą panaudojimo atvejį.
3. Jei norima, pereinama į panaudojimo atvejį „patalpinti VT hierarchijoje“	
4. Paspaudžiamas formos uždarymo mygtukas.	4.1. Forma uždaroma.
Alternatyvūs scenarijai	Vartotojas uždaro formą nepasirinkęs jokios komandos.

4 lentelė. Panaudojimo atvejis „Peržiūrėti VT hierarchiją“

2. Panaudojimo atvejis	Peržiūrėti VT hierarchija
Tikslas	Peržiūrėti veiklos taisyklių hierarchiją.
Aktoriai	Funkcinis konsultantas
Ryšiai su kitais PA	Gali būti išplėstas iki panaudojimo atvejo „redaguoti įrašą“.
Nefunkciniai reikalavimai	1, 2, 3
Prieš-sąlygos	Sistemoje yra bent viena taisyklė.
Sužadinimo sąlyga	Vartotojas iš meniu pasirenka veiklos taisyklių peržiūros komandą.
Po-sąlygos	Nėra.
Pagrindinis scenarijus	Sistemos reakcija ir sprendimai
1. Pasirenkama VT hierarchijos peržiūra	1.1. Atidaroma forma vaizduojanti aukščiausią VT hierarchijos lygmenį.
2. Išskleidžiamas norima hierarchijos šaka.	
3. Jei norima, pereinama į panaudojimo atvejį „redaguoti įrašą“	
4. Paspaudžiamas formos uždarymo mygtukas.	4.1. Forma uždaroma.

Alternatyvūs scenarijai	Nėra
3. Panaudojimo atvejis	Redaguoti įrašą
Tikslas	Redaguoti veiklos taisyklę arba reikšmių aibę.
Aktoriai	Funkcinis konsultantas
Ryšiai su kitais PA	Apima panaudojimo atvejus: „redaguoti atominę taisyklę“, „redaguoti sudėtinę taisyklę“ ir „redaguoti reikšmių aibę“; gali būti išplėstas iki panaudojimo atvejo „patalpinti VT hierarchijoje“.
Nefunkciniai reikalavimai	2
Prieš-sąlygos	Sistemoje yra bent viena taisyklė.
Sužadinimo sąlyga	Vartotojas VT hierarchijos formoje pasirenka taisyklės arba reikšmių aibės redagavimo komandą..
Po-sąlygos	Egzistuojantis įrašas pakoreguotas ir išsaugotas.
Pagrindinis scenarijus	Sistemos reakcija ir sprendimai
1. Pasirenkamas įrašo koregavimas	1.1. Atidaroma įrašo redagavimo forma.
2. Jei norima įrašą redaguoti, pasirenkama redagavimo komanda.	2.1. Priklausomai nuo atidaryto įrašo pereinama į vieną iš panaudojimo atvejų: „redaguoti atominę taisyklę“, „redaguoti sudėtinę taisyklę“, „redaguoti reikšmių aibę“.
3. Jei norima, pereinama į panaudojimo atvejį „patalpinti VT hierarchijoje“	
4. Paspaudžiamas formos uždarymo mygtukas.	4.1. Forma uždaroma.
Alternatyvūs scenarijai	Vartotojas uždaro formą nepasirinkęs jokios komandos.

5 lentelė. Panaudojimo atvejis „Redaguoti sudėtinę taisyklę“

4. Panaudojimo atvejis	Redaguoti sudėtinę taisyklę
Tikslas	Redaguoti sudėtinę veiklos taisyklę.
Aktoriai	Funkcinis konsultantas
Ryšiai su kitais PA	Gali būti išplėstas iki panaudojimo atvejo „redaguoti atominę taisyklę“.
Nefunkciniai reikalavimai	2
Prieš-sąlygos	Sistemoje yra bent viena taisyklė.
Sužadinimo sąlyga	Vartotojas iš naujo įrašo įvedimo formos arba iš įrašo redagavimo pasirinko sudėtinės taisyklės redagavimą.
Po-sąlygos	Sudėtinė taisyklė poredaguota ir įrašyta lentelėje.
Pagrindinis scenarijus	Sistemos reakcija ir sprendimai
1. Pasirenkamas sudėtinės taisyklės redagavimas	1.1. Atidaroma sudėtinių taisyklių lentelė ir pagalbinių įrankių juosta.

2. Redaguojami taisyklės identifikatoriai	
3. Jei norima, pereinama į panaudojimo atvejį „redaguoti atominę taisyklę“	
4. Redaguojamas pirmos taisyklės identifikatorius	
5. Redaguojamas loginis operatorius	
6. Redaguojamas antros taisyklės identifikatorius	
7. Patvirtinami įvesti duomenys	7.1. VT duomenys išsaugomi, sudėtinųjų taisyklių lentelė ir įrankių juosta uždaromi.
Alternatyvūs scenarijai	Vartotojas uždaro formą dar nesuvekęs visų duomenų.

6 lentelė. Panaudojimo atvejis „Redaguoti atominę taisyklę“

5. Panaudojimo atvejis	Redaguoti atominę taisyklę
Tikslas	Redaguoti atominę veiklos taisyklę.
Aktoriai	Funkcinis konsultantas
Ryšiai su kitais PA	Gali būti išplėstas iki panaudojimo atvejo „redaguoti reikšmių aibę“.
Nefunkciniai reikalavimai	2
Prieš-sąlygos	Sistemoje yra bent viena taisyklė.
Sužadavimo sąlyga	Vartotojas iš naujo įrašo įvedimo formos arba iš įrašo redagavimo formos, arba iš sudėtinės taisyklės redagavimo įrankių juostos pasirinko atominės taisyklės redagavimą.
Po-sąlygos	Atominė taisyklė poredaguota ir įrašyta lentelėje.
Pagrindinis scenarijus	Sistemos reakcija ir sprendimai
1. Pasirenkamas atominės taisyklės redagavimas	1.1. Atidaroma atominių taisyklių lentelė ir pagalbinių įrankių juosta.
2. Redaguojami taisyklės identifikatoriai	
3. Redaguojamas kintamasis	
4. Redaguojamas loginis operatorius	
5. Jei norima, pereinama į panaudojimo atvejį „įvesti reikšmių aibę“	
6. Patvirtinami įvesti duomenys	6.1. VT duomenys išsaugomi, atominių taisyklių lentelė ir įrankių juosta i
Alternatyvūs scenarijai	Vartotojas uždaro formą dar nesuvekęs visų duomenų.

7 lentelė. Panaudojimo atvejis „Redaguoti reikšmių aibę“

6. Panaudojimo atvejis	Redaguoti reikšmių aibę
Tikslas	Redaguoti reikšmių aibę.
Aktoriai	Funkcinis konsultantas
Ryšiai su kitais PA	Nėra
Nefunkciniai reikalavimai	2
Prieš-sąlygos	Sistemoje yra bent viena reikšmių aibė.
Sužadinimo sąlyga	Vartotojas iš naujo įrašo įvedimo formos arba iš įrašo redagavimo formos, arba iš atominės taisyklės redagavimo įrankių juostos pasirinko reikšmių aibės redagavimą.
Po-sąlygos	Reikšmių aibė poredaguota ir įrašyta lentelėje.
Pagrindinis scenarijus	Sistemos reakcija ir sprendimai
1. Pasirenkamas reikšmių aibės redagavimas	1.1. Atidaroma reikšmių aibių lentelė.
2. Redaguojami reikšmių aibės identifikatoriai	
3. Redaguojamas reikšmių aibės ženklas	
4. Redaguojamos reikšmių aibės ribos	
5. Patvirtinami įvesti duomenys	5.1. Reikšmių aibės duomenys išsaugomi, reikšmių aibės lentelė ir įrankių juosta uždaromi.
Alternatyvūs scenarijai	Vartotojas uždaro formą dar nesuvedęs visų duomenų.

8 lentelė. Panaudojimo atvejis „Patalpinti VT hierarchijoje“

7. Panaudojimo atvejis	Patalpinti VT hierarchijoje
Tikslas	Patalpinti veiklos taisyklę hierarchijoje.
Aktoriai	Funkcinis konsultantas
Ryšiai su kitais PA	Nėra
Nefunkciniai reikalavimai	2
Prieš-sąlygos	Sistemoje yra bent viena taisyklė.
Sužadinimo sąlyga	Vartotojas iš naujo įrašo įvedimo formos arba įrašo redagavimo formos pasirinko komandą „patalpinti VT hierarchijoje“.
Po-sąlygos	Veiklos taisyklės vieta hierarchijoje pakeista.
Pagrindinis scenarijus	Sistemos reakcija ir sprendimai
1. Pasirenkamas veiklos taisyklės vietos hierarchijoje keitimas	1.1. Atidaroma veiklos taisyklių hierarchijos forma.
2. Grafiškai pele nurodoma naujoji VT vieta hierarchijoje.	
3. Naujoji vieta patvirtinama atsakymu į klausimą „Taip“.	3.1. Naujoji VT vieta hierarchijoje išsaugoma. 3.2. Veiklos taisyklių hierarchijos forma automatiškai uždaroma.
Alternatyvūs scenarijai	Vartotojas nepatvirtina naujosios VT vietos hierarchijoje.

3.5.4. Sistemos statinis vaizdas

3.5.4.1. Apžvalga

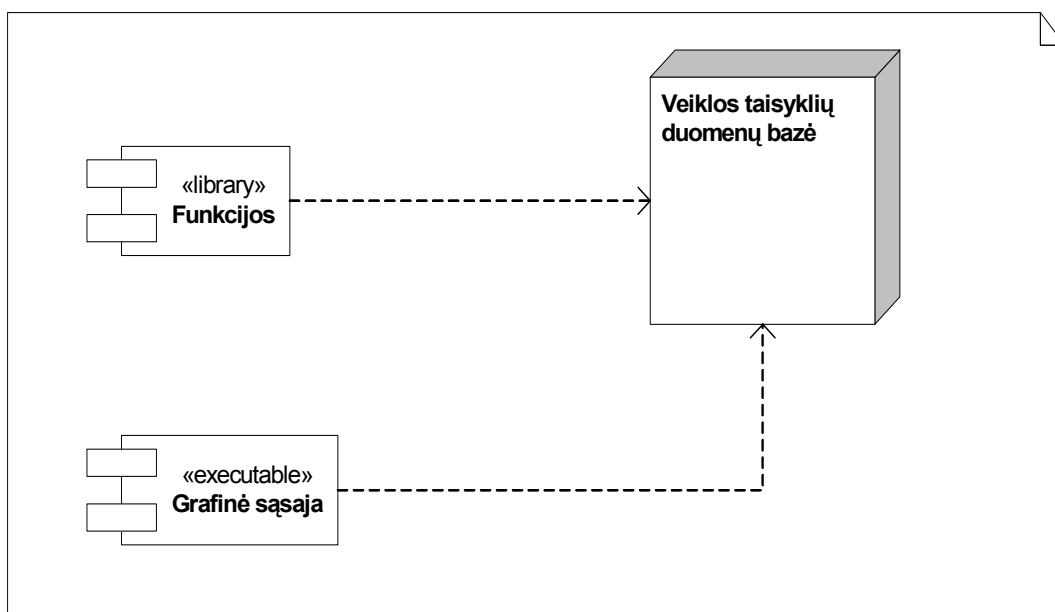
Projekto tikslas yra išlaisvinti sistemos vartotojus nuo rutinos valdant sudėtingas veiklos taisykles:

- užtikrinti įvedamų taisyklių sintaksinį teisingumą;
- palengvinti taisyklių paiešką;
- pagerinti taisyklių skaitomumą.

Veiklos taisyklių valdymo karkasą sudaro trys dalys:

- VT duomenų bazė;
- VT panaudojimui skirtų funkcijų biblioteka;
- grafinė sąsaja.

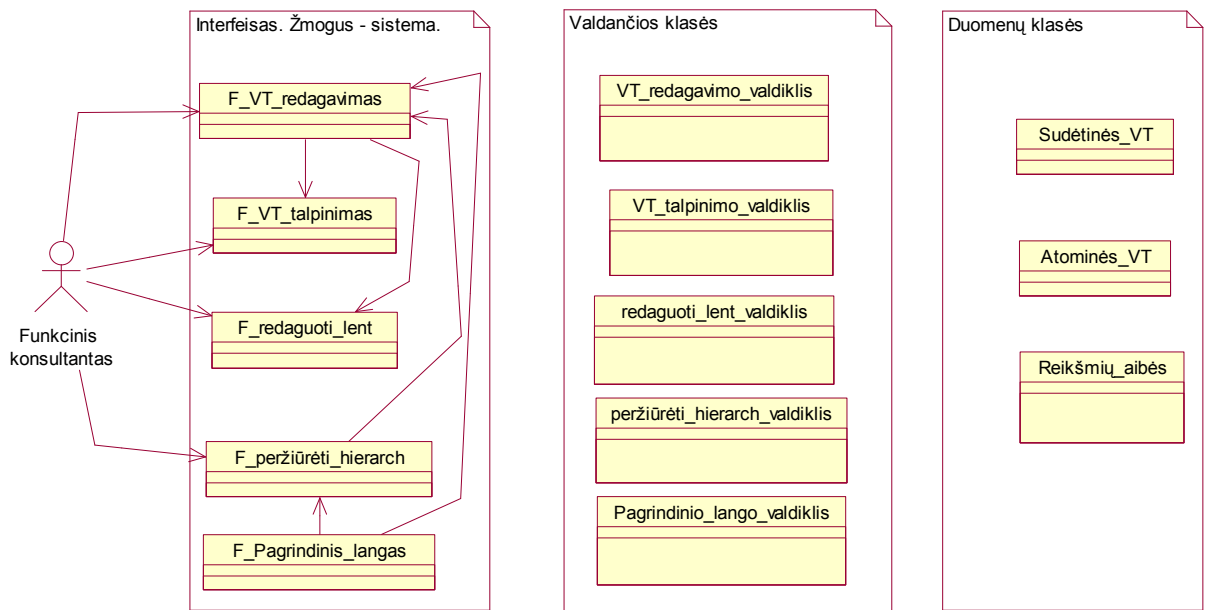
Sistemos komponentų diagrama pateikiama 8 pav.



8 pav. Sistemos komponentų diagrama

3.5.4.2. Paketų detalizavimas

Funkcijų biblioteka šiame projekte nėra nagrinėjama, todėl šiame dokumente pateikiama tik naujai kuriamos grafinės sąsajos ir susietos veiklos taisyklių duomenų bazės architektūros specifikacija. Šio posistemio klasių diagrama pateikiama 9 pav.



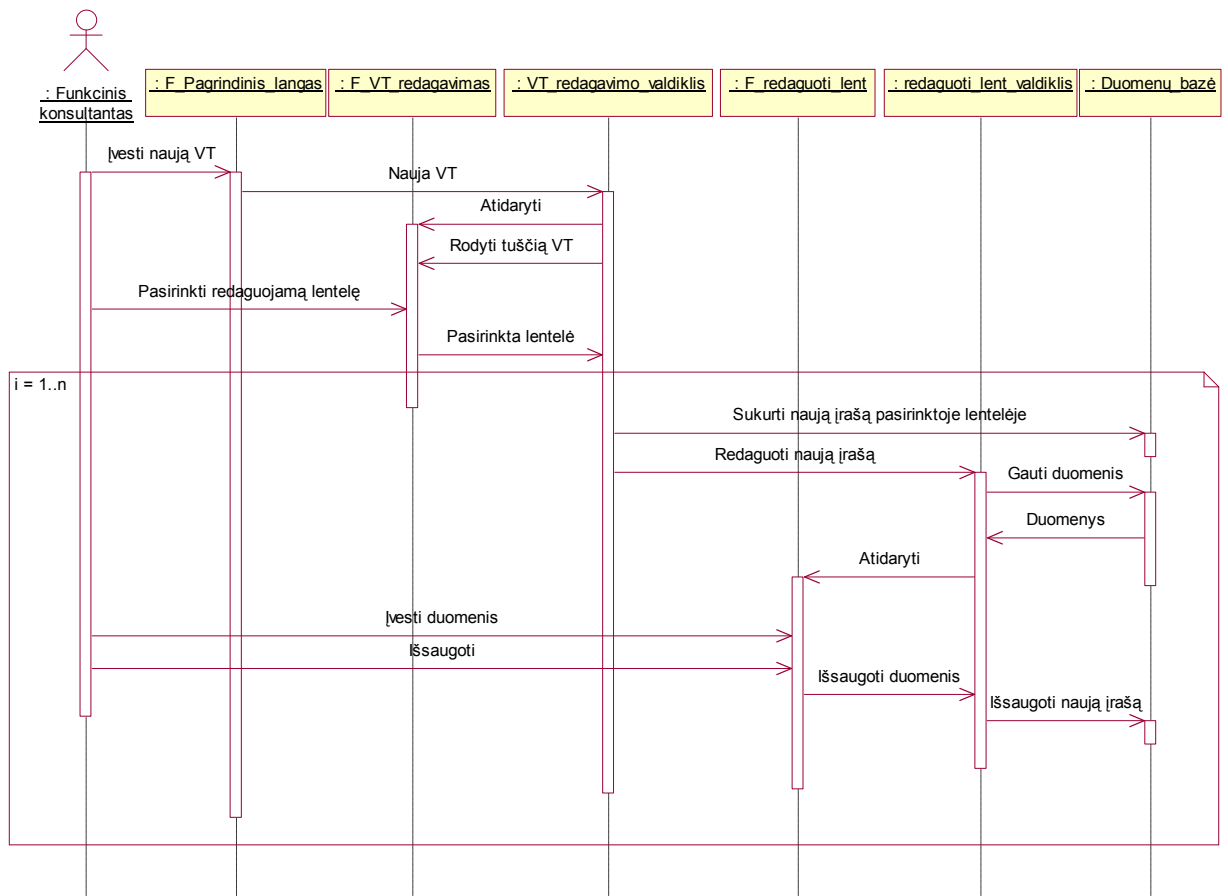
9 pav. grafinės sąsajos posistemio klasių diagrama

3.5.5. Sistemos dinaminis vaizdas

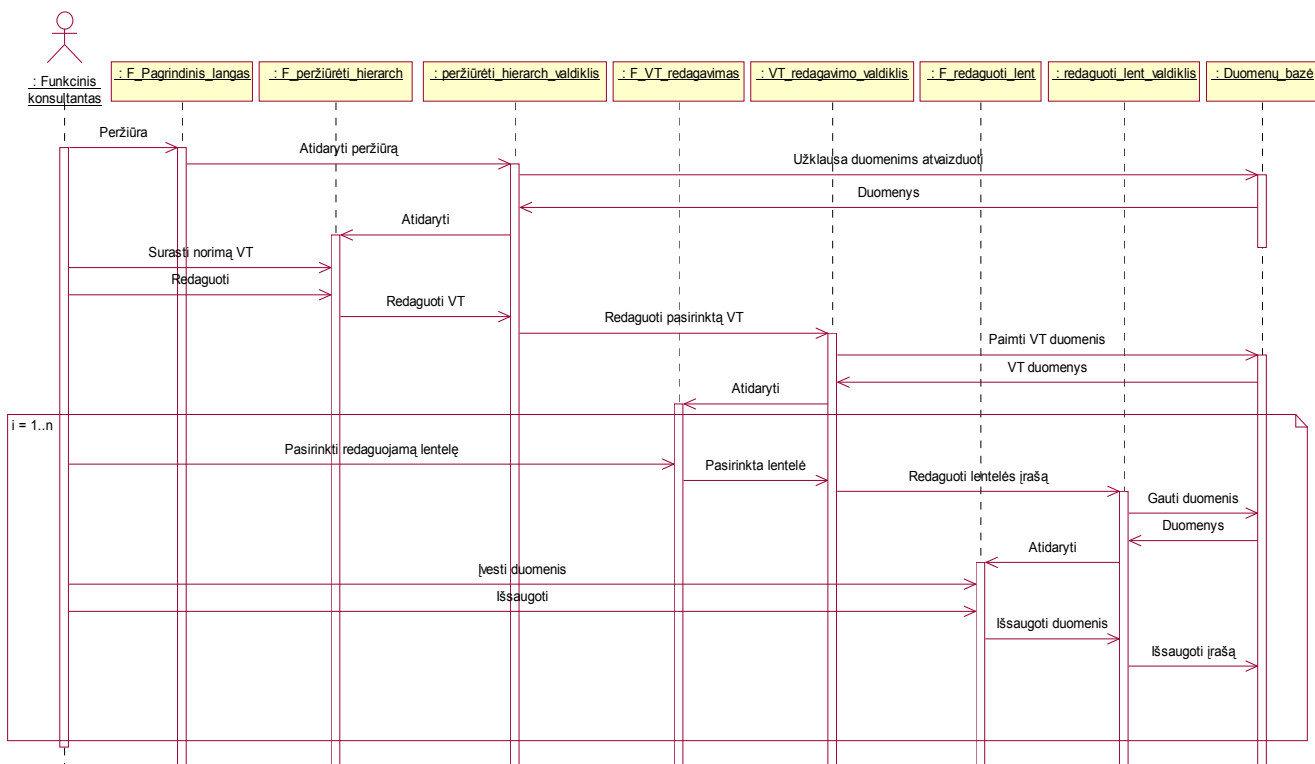
3.5.5.1. Sekų diagramos

Darbų sekų diagramos aprašo sistemos atliekamų darbų eigos bei objektų tarpusavio sąveikos tęstinumą laiko atžvilgiu. Šios diagramos nusako, kada ir kokie veiksmai turi būti atliekami.

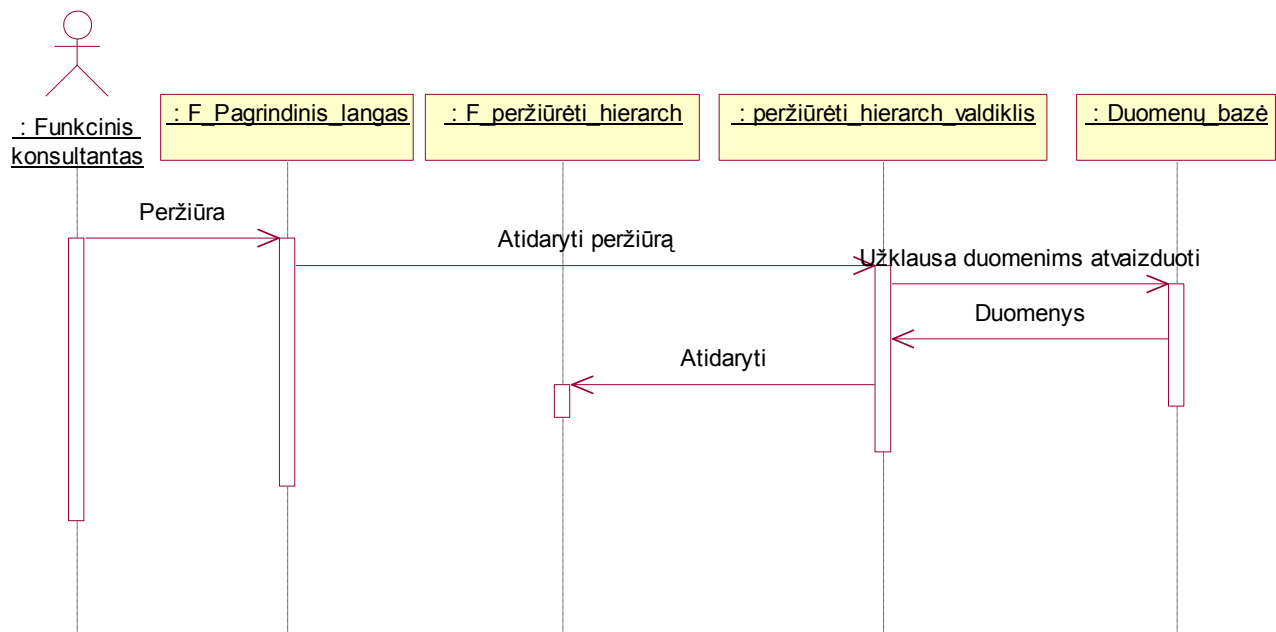
Sekų diagramos pateikiamos 10 – 16 pav.



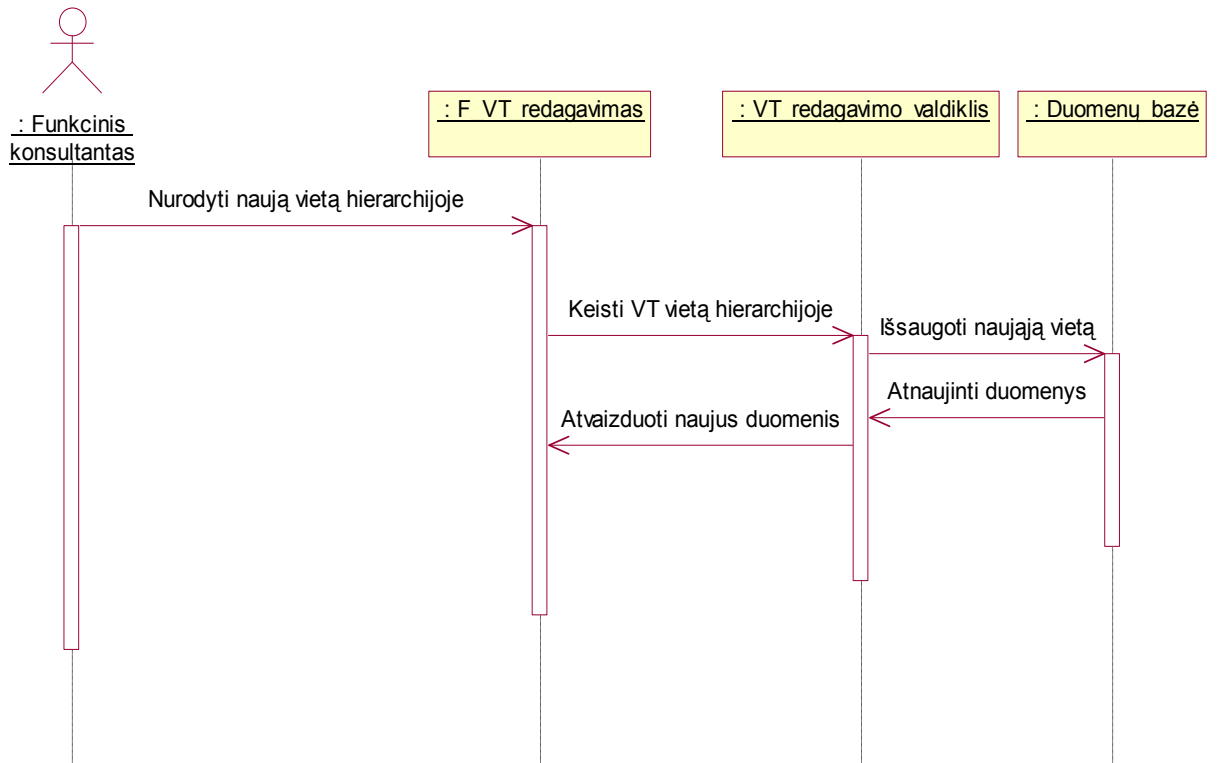
10 pav. Naujo įrašo įvedimo sekų diagrama



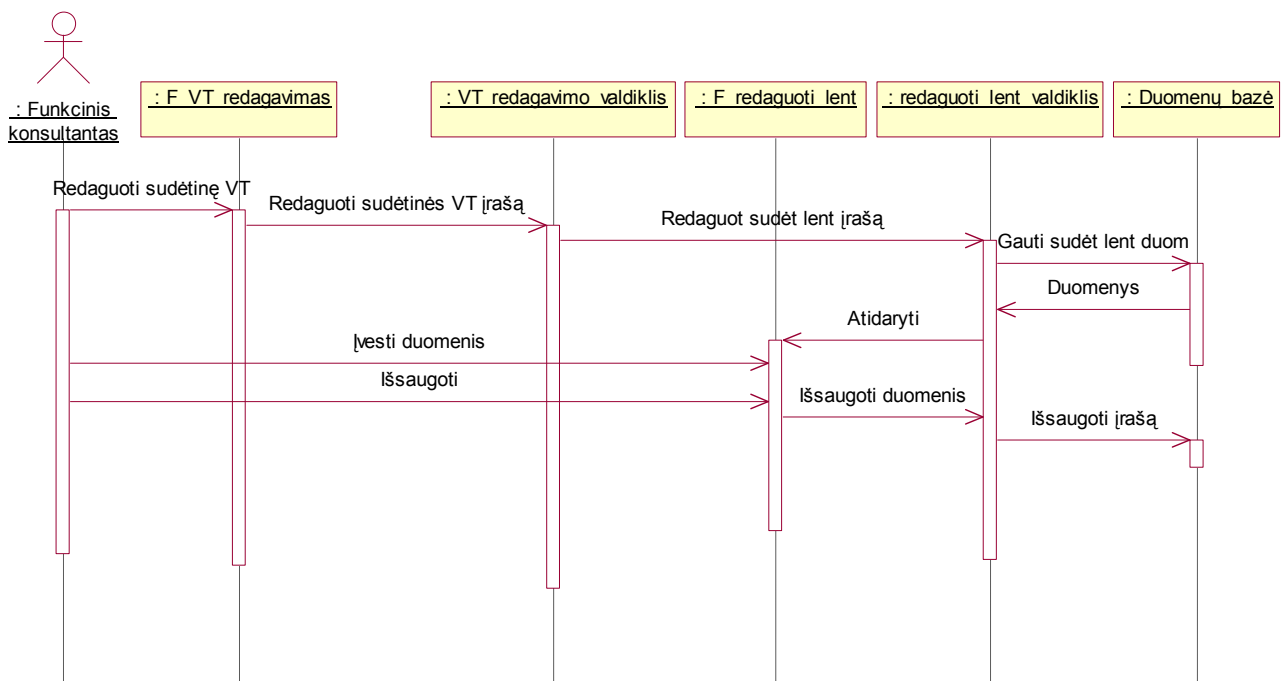
11 pav. Įrašo redagavimo sekų diagrama



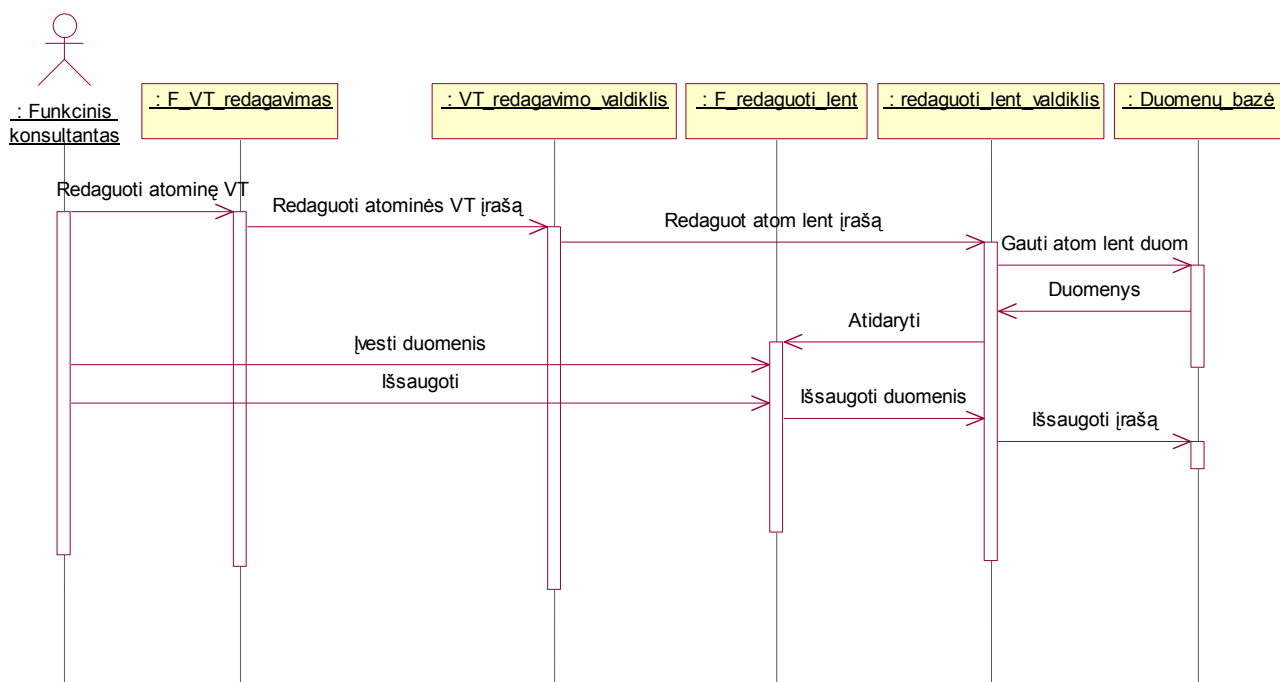
12 pav. VT hierarchijos peržiūros sekų diagrama



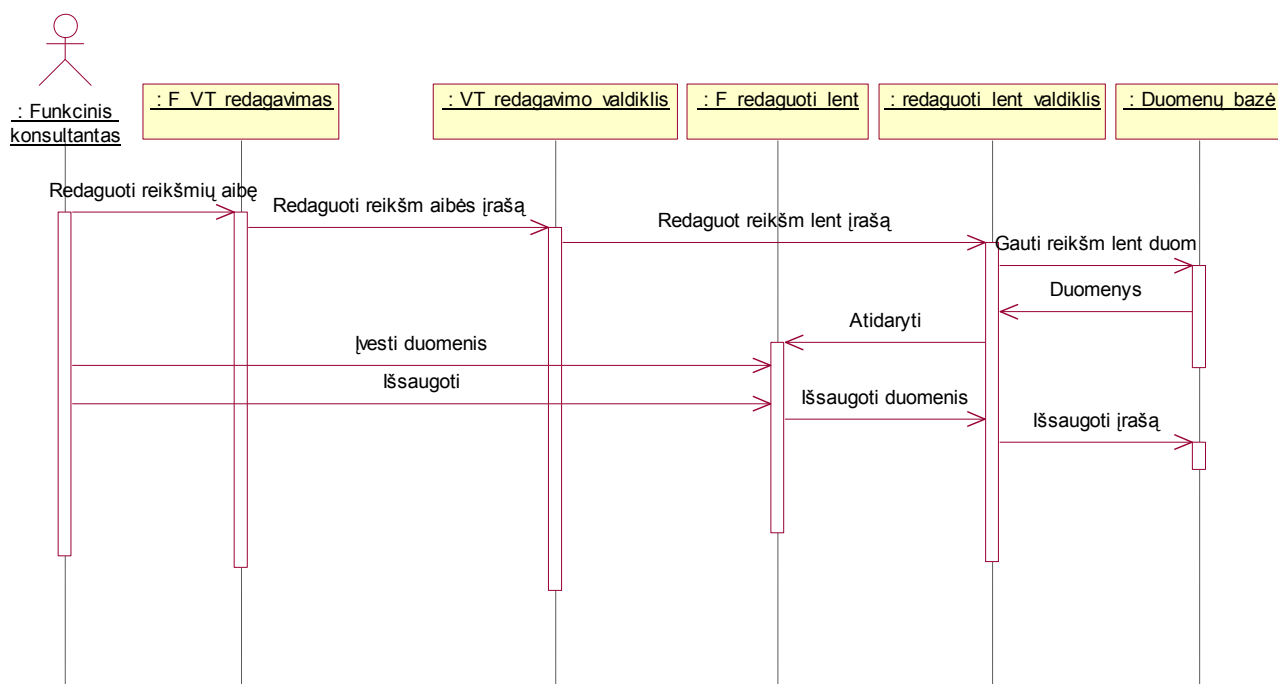
13 pav. VT patalpinimo hierarchijoje sekų diagrama



14 pav. Sudėtinės VT redagavimo sekų diagrama



15 pav. Atominės VT redagavimo sekų diagrama

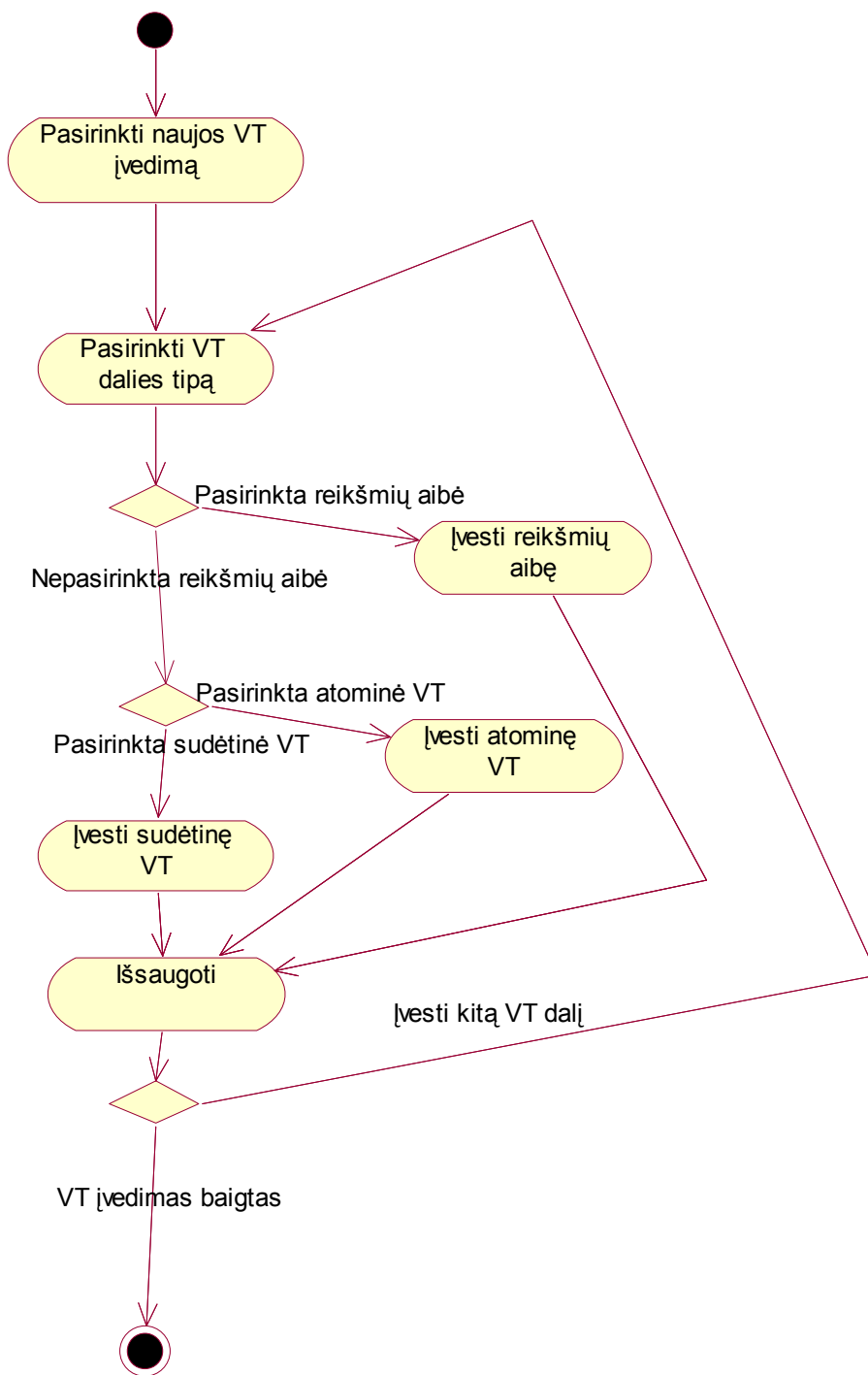


16 pav. Reikšmių aibės redagavimo sekų diagrama

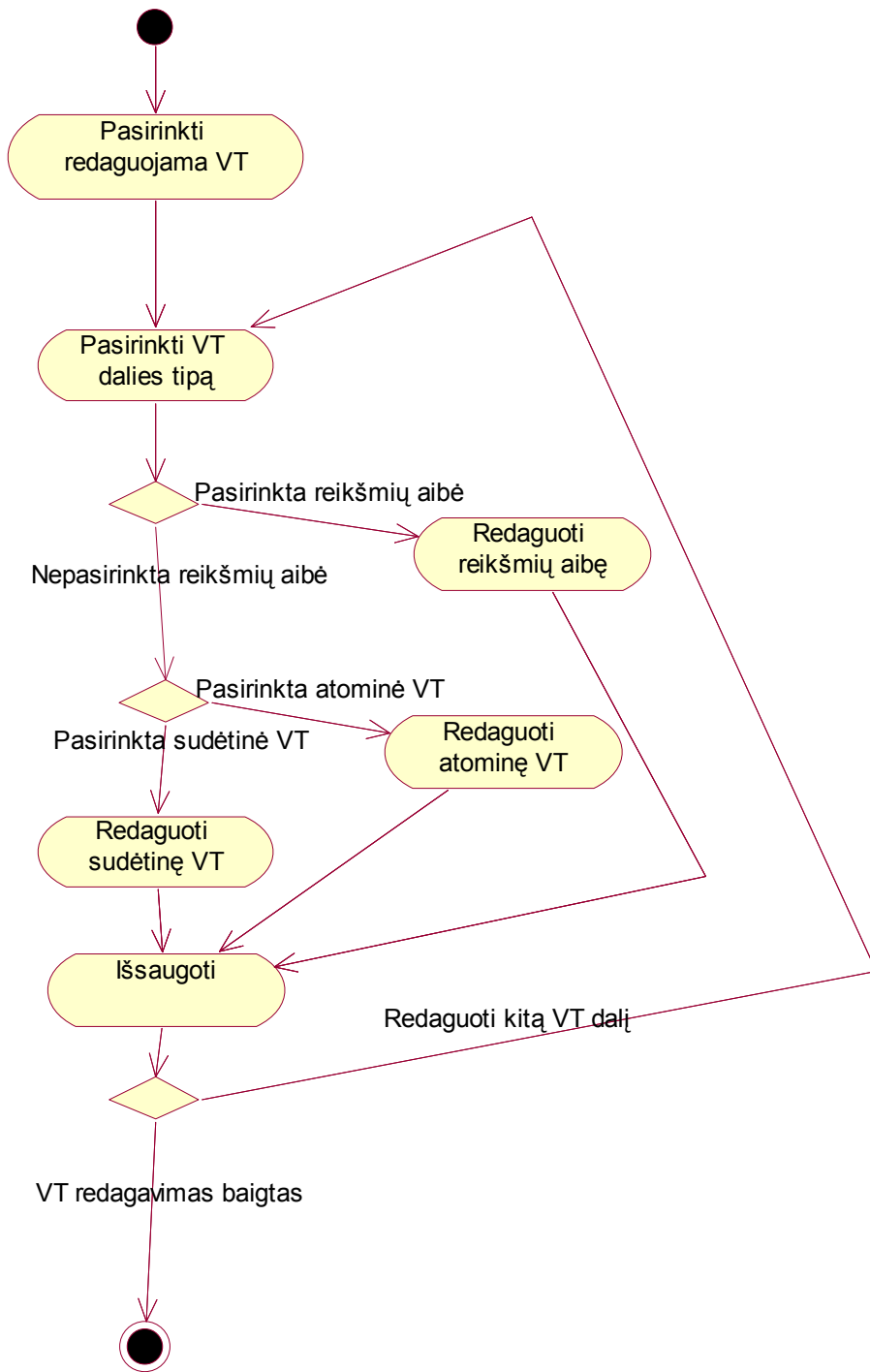
3.5.5.2. Veiklos diagramos

Veiklos diagramos vaizduoja elgseną sistemos viduje. Veiklos diagramos sudarytos skaidant veiklą pagal panaudojimo atvejus.

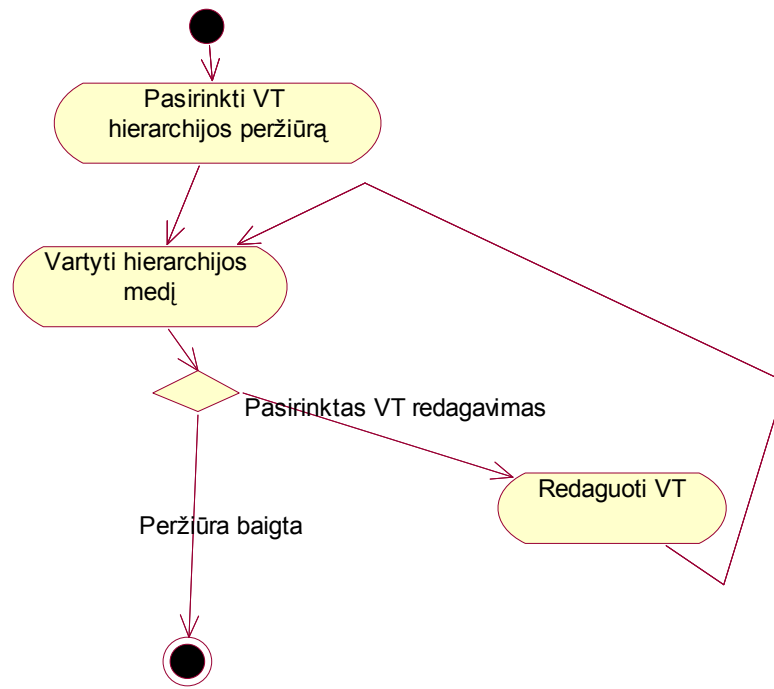
Veiklos diagramos pateiktos 17 – 23 pav.



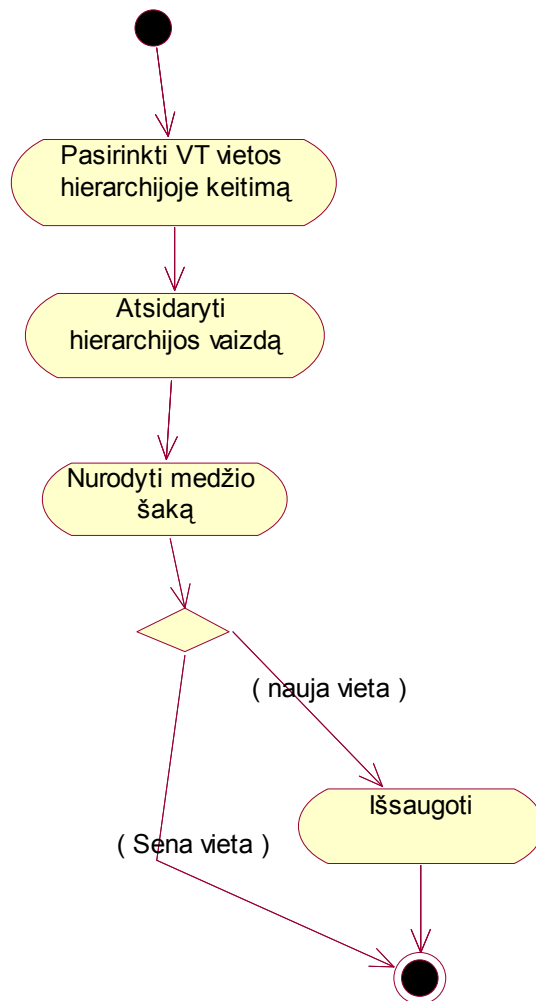
17 pav. Naujos VT įvedimo veiklos diagrama



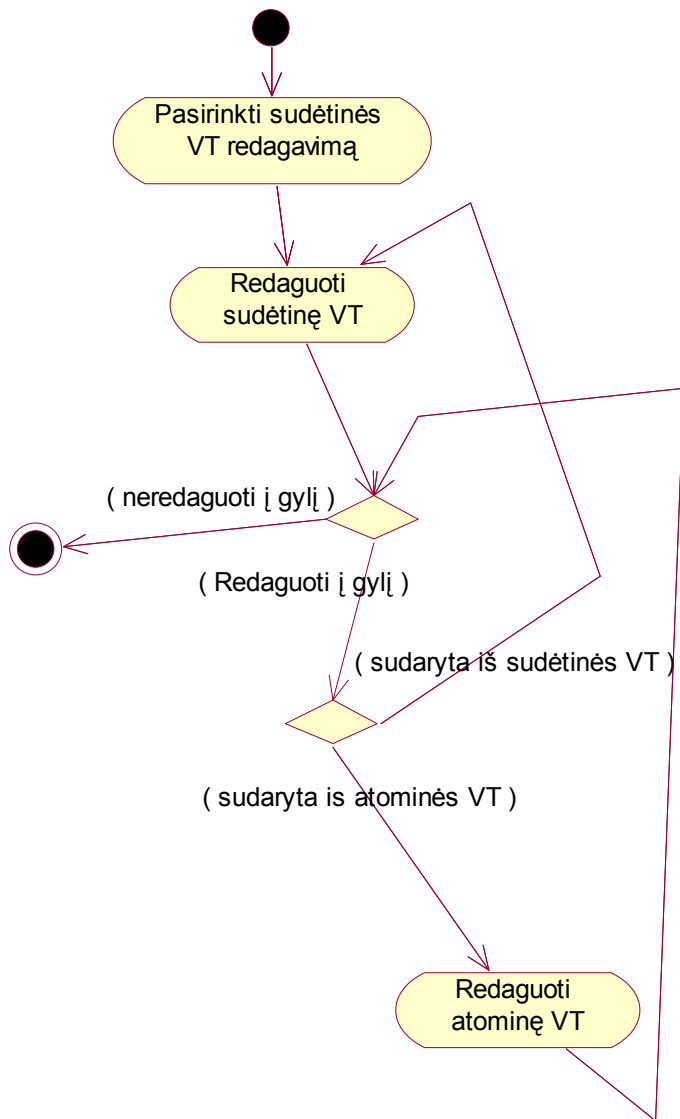
18 pav. VT redagavimo veiklos diagrama



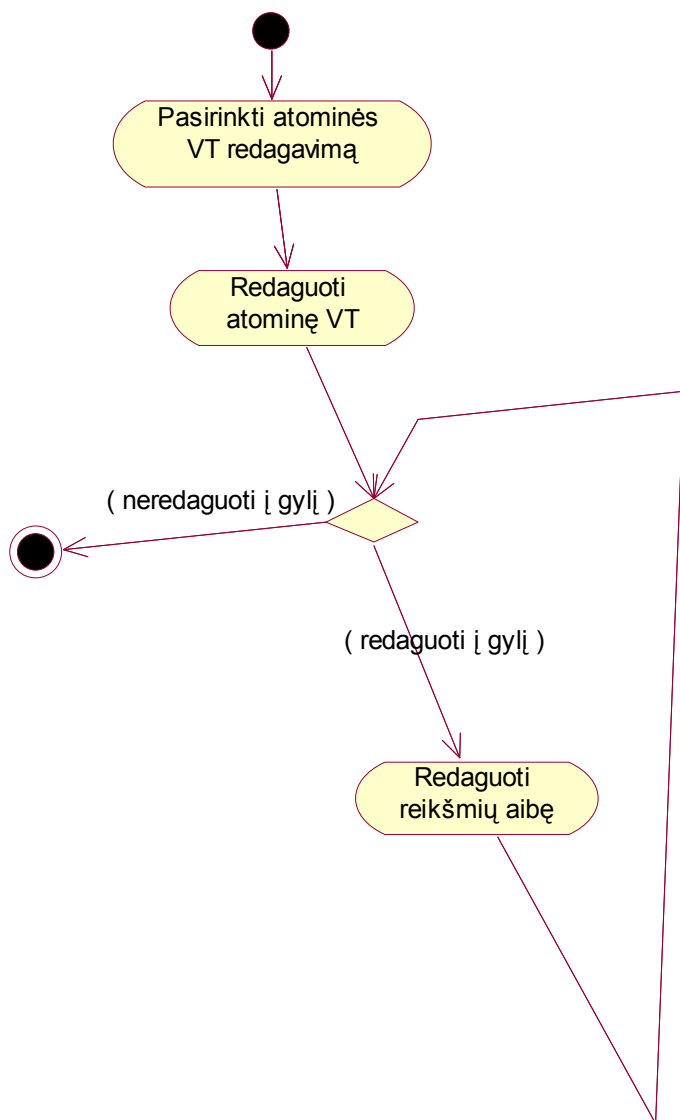
19 pav. VT hierarchijos peržiūros veiklos diagrama



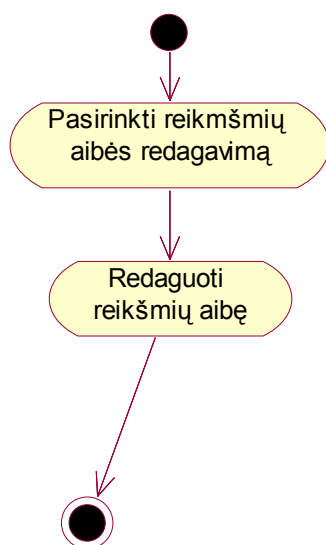
20 pav. VT patalpinimo hierarchijoje veiklos diagrama



21 pav. Sudėtinės VT redagavimo veiklos diagrama



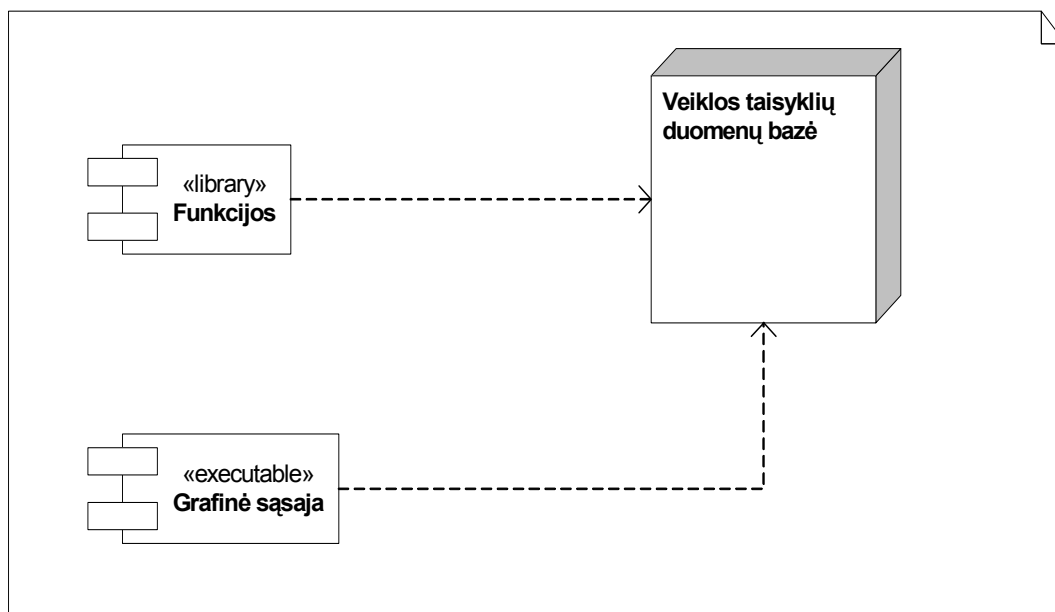
22 pav. Atominės VT redagavimo veiklos diagrama



23 pav. Reikšmių aibės redagavimo veiklos diagrama

3.5.6. Išdėstymo vaizdas

Kuriama grafinė sąsaja bus integruota į Lietuvos muitinės informacinėje sistemoje veikiančią veiklos taisyklių valdymo karkasą. Visus duomenis grafinė sąsaja ims iš jau egzistuojančios duomenų bazės, tuo tarpu su funkcijų biblioteka grafinė sąsaja visiškai nebus susieta. Sistemos komponentų diagrama pateikiama 24 pav.



24 pav. Sistemos komponentų diagrama

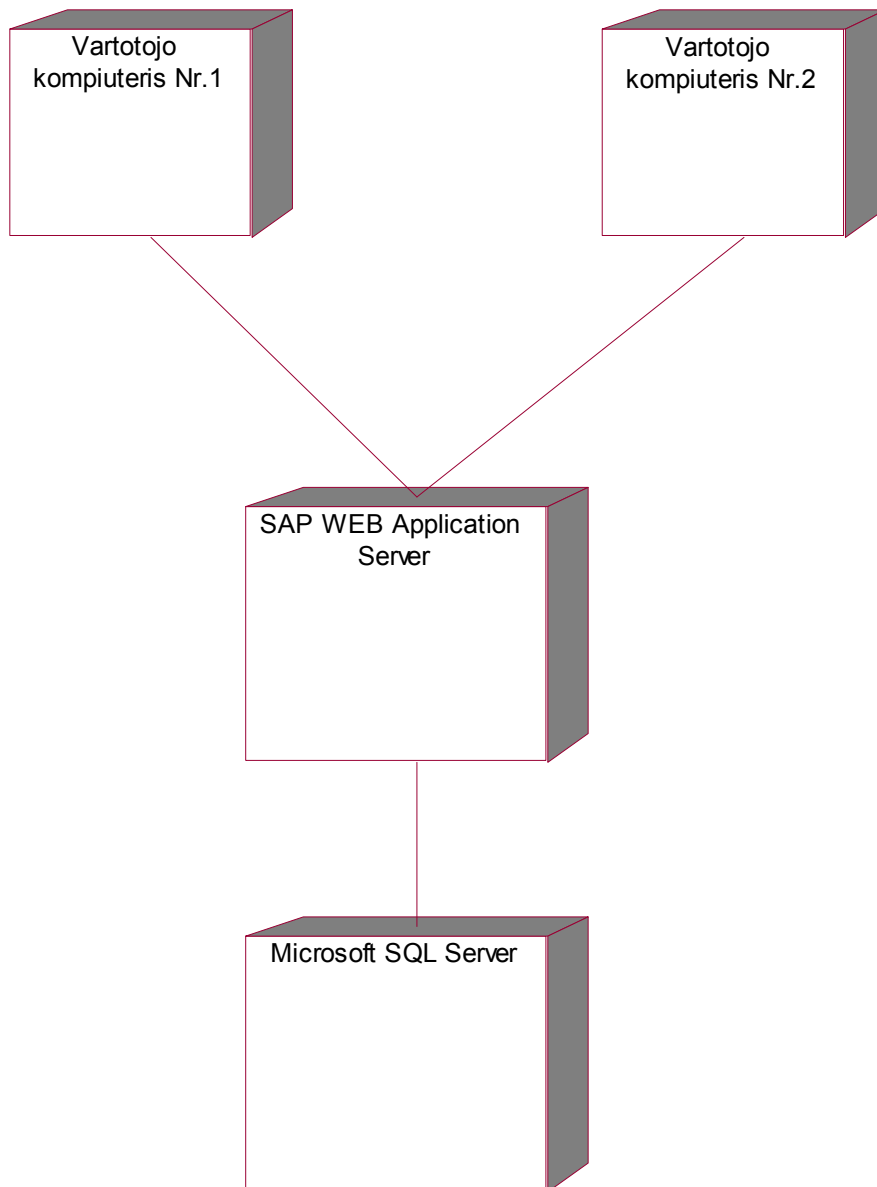
Pagrindinė techninė įranga reikalinga siekiant užtikrinti sistemos funkcionalumą:

- *Windows Server 2000/XP* operacinė sistema,
- *Microsoft SQL Server*,
- *SAP WEB Application Server*.

Visi sistemos komponentai bus įdiegti serverio kompiuteryje. Vartotojai prie sistemos jungsis naudodami kliento kompiuteryje įdiegtą *SAP* sistemą.

Nagrinėjamas veiklos taisyklių valdymo karkasas veikia kliento–serverio sistemos architektūroje. Kuriama grafinė sąsaja veiks *SAP WEB Application Server* serveryje.

Sistemos išdėstymo vaizdas pateikiamas 25 pav.



25 pav. Sistemos išdėstymo vaizdas

3.5.7. Duomenų vaizdas

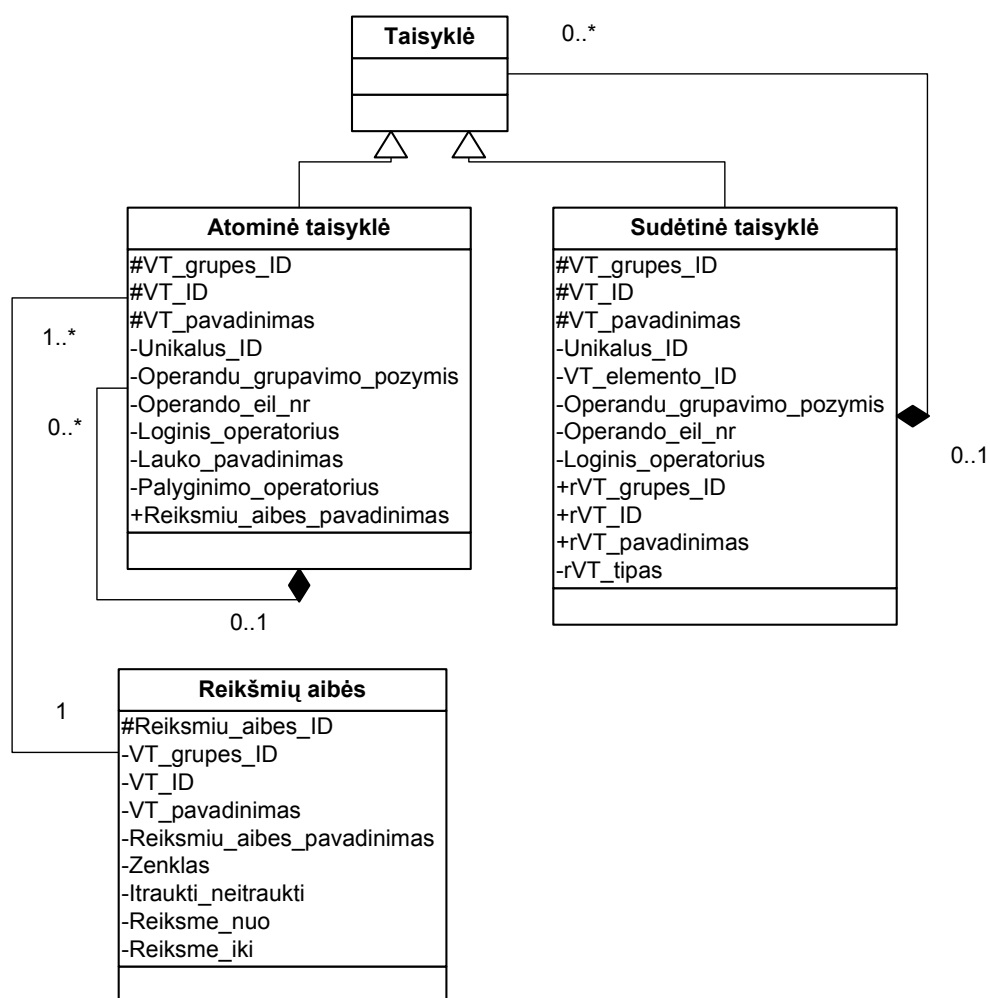
Duomenų bazė jau yra sukurta. Reikalaujama jos nemodifikuoti. Duotoji duomenų bazė sudaryta iš trijų lentelių nesujungtų reliaciniais ryšiais:

- ZUTL_CBUSRULS – Sudėtinės taisyklės
- ZUTL_BUSRULS – Atominės taisyklės
- ZUTL_VALRNGS – Reikšmių aibės

Atomine taisyklę gali sudaryti vienas arba daugiau atominių taisyklių lentelės įrašų ir po vieną reikšmių aibių lentelės įrašą kiekvienam atominių taisyklių lentelės įrašui.

Sudėtinė taisyklė gali būti sudaryta iš atominių arba kitų sudėtinių taisyklių.

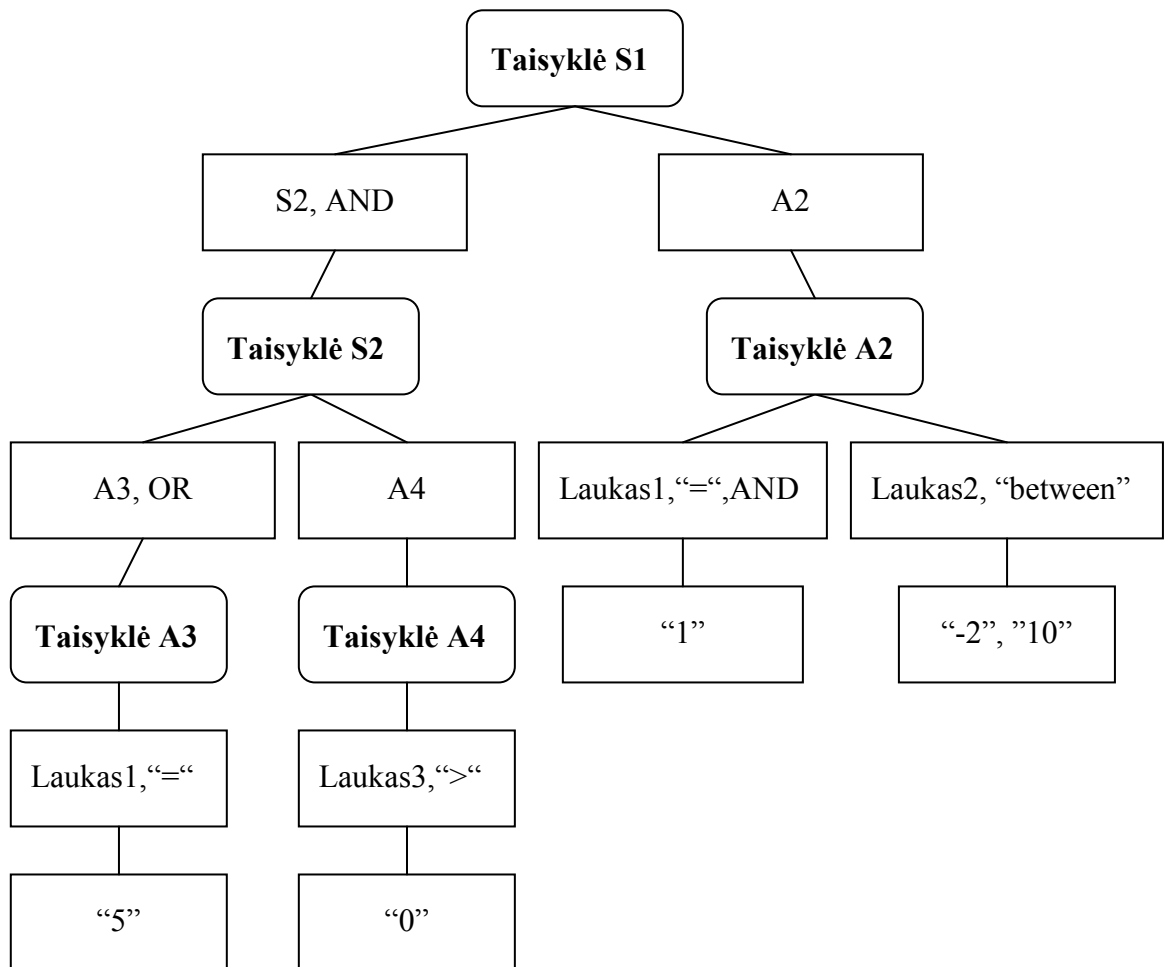
Duomenų bazės loginių ryšių vaizdas pateikiamas klasių diagramoje 26 pav.



26 pav. Duomenų bazės klasių diagrama

27 pav. pateikiamas pavyzdys, kaip duomenų bazės lentelėse atrodytų sudėtinė taisyklė S1. Taisyklė S1 susideda iš kitos sudėtinės taisyklės S2 bei atominės taisyklės A2. Taisyklė S2 susideda iš atominių taisyklių A3 ir A4.

$S1 = (laukas1 = 5 \text{ OR } laukas3 > 0) \text{ AND } (laukas1 = 1 \text{ AND } (laukas2 > -2 \text{ AND } laukas2 < 10))$



27 pav. Hierarchinė veiklos taisyklės vidinė struktūra

Kuriamojo veiklos taisyklių valdymo karkaso fizinės duomenų bazės modelis pateikiamas 28 pav.

ZUTL_BUSRULS
MANDT RLITMID RLGRPID RLID RLNAME PRECD SEQNO LOGOP FLDNM CMPOP RNGNM

ZUTL_VALRNGS
MANDT RNGID RLGRPID RLID RLNAME RNGNM SGN OPT LOW HIGH

ZUTL_CBUSRULS
MANDT CRLITMID CRLGRPID CRLID CRLNAME PRECD SEQNO LOGOP RLGRPID RLID RLNAME RLTYPE

28 pav. Duomenų bazės modelis

29 – 31 pav. pateikiami lentelių laukų paaiškinimai.

Attributes Delivery and Maintenance Fields Entry help/check Currency/Quantity Fields							
Lauko pav.	R...	Initi...	Duomenų elemen.	Data ty...	Ilgis	Deš. ...	Trumpas aprašas
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0	Klientas
CRLITMID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_RITID	NUMC	20	0	Veiklos taisyklės elemento identifikatorius
CRLGRPID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_RLGRPID	CHAR	10	0	Veiklos taisyklių grupės identifikatorius
CRLID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_RULID	CHAR	6	0	Veiklos taisyklės identifikatorius
CRLNAME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_RULMM	CHAR	20	0	Veiklos taisyklės pavadinimas
PRECD	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_PRECD	DEC	3	0	Operandų grupavimo požymis (skliaustelių sk.)
SEQNO	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_SEQNO	DEC	3	0	Operando eilės numeris
LOGOP	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_LOGOP	CHAR	3	0	Loginis operatorius
RLGRPID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_RLGRPID	CHAR	10	0	Veiklos taisyklių grupės identifikatorius
RLID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_RULID	CHAR	6	0	Veiklos taisyklės identifikatorius
RLNAME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_RULMM	CHAR	20	0	Veiklos taisyklės pavadinimas
RLTYPE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_RULTP	CHAR	1	0	Veiklos taisyklės tipas

29 pav. ZUTL_CBUSRULS – sudėtinės veiklos taisyklės

Attributes Delivery and Maintenance Fields Entry help/check Currency/Quantity Fields							
Lauko pav.	R...	Initi...	Duomenų elemen.	Data ty...	Ilgis	Deš. ...	Trumpas aprašas
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0	Klientas
RLITMID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_RITID	NUMC	20	0	Veiklos taisyklės elemento identifikatorius
RLGRPID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_RLGRPID	CHAR	10	0	Veiklos taisyklių grupės identifikatorius
RLID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_RULID	CHAR	6	0	Veiklos taisyklės identifikatorius
RLNAME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_RULMM	CHAR	20	0	Veiklos taisyklės pavadinimas
PRECD	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_PRECD	DEC	3	0	Operandų grupavimo požymis (skliaustelių sk.)
SEQNO	<input type="checkbox"/>	<input type="checkbox"/>	ZUTL_SEQNO	DEC	3	0	Operando eilės numeris
LOGOP	<input type="checkbox"/>	<input type="checkbox"/>	ZUTL_LOGOP	CHAR	3	0	Loginis operatorius
FLDNM	<input type="checkbox"/>	<input type="checkbox"/>	FIELDNAME	CHAR	30	0	Lauko pavadinimas
CMPOP	<input type="checkbox"/>	<input type="checkbox"/>	ZUTL_CMPPOP	CHAR	10	0	Palyginimo operatorius
RNGNM	<input type="checkbox"/>	<input type="checkbox"/>	ZUTL_RNGNM	CHAR	10	0	Reikšmių aibės pavadinimas

30 pav. ZUTL_BUSRULS – atominės veiklos taisyklės

Attributes Delivery and Maintenance Fields Entry help/check Currency/Quantity Fields							
Lauko pav.	R...	Initi...	Duomenų elemen.	Data ty...	Ilgis	Deš. ...	Trumpas aprašas
<u>MANDT</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0	Klientas
<u>RNGID</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_RNGID	NUMC	20	0	Reikšmių aibės identifikatorius
<u>RLGRPID</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_RLGRPID	CHAR	10	0	Veiklos taisyklių grupės identifikatorius
<u>RLID</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_RLID	CHAR	6	0	Veiklos taisyklės identifikatorius
<u>RLNAME</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_RLNM	CHAR	20	0	Veiklos taisyklės pavadinimas
<u>RNGNM</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_RNGNM	CHAR	10	0	Reikšmių aibės pavadinimas
<u>SGN</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	DDSIGN	CHAR	1	0	Type of SIGN component in row type of a Ranges type
<u>OPT</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	DDOPTION	CHAR	2	0	Type of OPTION component in row type of a Ranges type
<u>LOW</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZUTL_RNGVL	CHAR	32	0	Aibės elemento reikšmė
<u>HIGH</u>	<input type="checkbox"/>	<input type="checkbox"/>	ZUTL_RNGVL	CHAR	32	0	Aibės elemento reikšmė

31 pav. ZUTL_VALRNGS – reikšmių aibės

4. EKSPERIMENTINIS SUKURTOS VTVS TYRIMAS

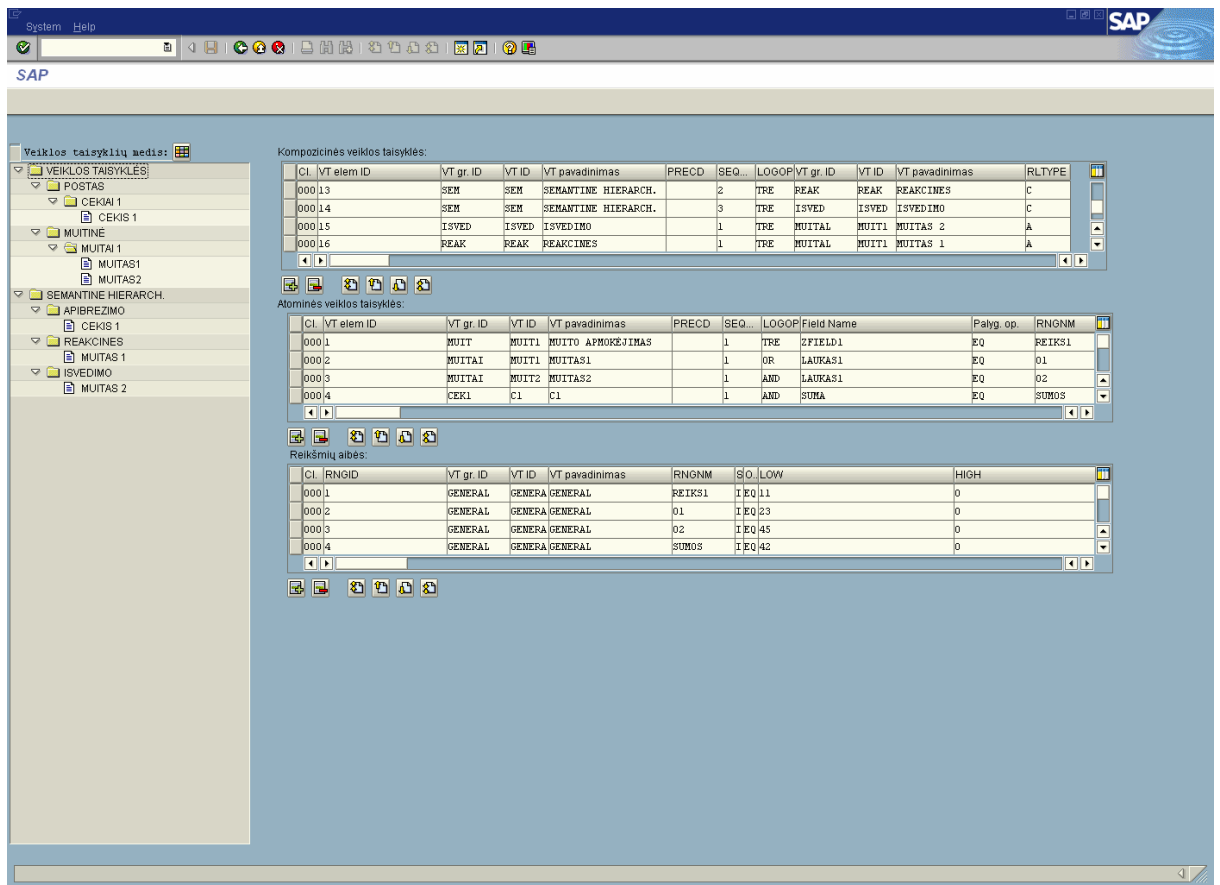
Sukurtoji VTVS ženkliai palengvina programuotojų darbą, sumažina klaidų tikimybę, paspartina pakeitimų įkėlimą į taikomąsias programas.

4.1. VT hierarchijos privalumai

Projektuojant VTVS, remiantis atlikta literatūros ir egzistuojančių sprendimų analize, buvo priimtas sprendimas sudaryti veiklos taisyklių hierarchiją. Sudarant hierarchiją, ypatingas dėmesys buvo skirtas veiklos taisyklių karkaso universalumui išlaikyti, t.y. siekta, kad pats karkasas neapribotų hierarchijos sudarymo kriterijų ir netgi tam pačiam taisyklių rinkiniui leistų sudaryti neribotą skaičių skirtingų hierarchinių medžių. Skirtingų hierarchijų kriterijų pavyzdžiai gali būti:

- pagal dokumentų tipus. Smulčiausią šaką sudaro tik su konkrečiu dokumentu susiję taisyklės. Stambesnėmis šakomis dokumentai apjungiami į dokumentų tipus;
- pagal taisyklių semantines kategorijas (remiantis RuleML). Smulčiausiai šakai priklauso taisyklės atliekančios tą pačią funkciją SQL sakinyje. Stambesnės šakos – anksčiau minėtos RuleML modelyje:
 - apibrėžimo;
 - išvedimo;
 - reakcinės;
 - struktūrinės;
 - transformacijų.

Abi hierarchijos bei fizinis taisyklių vaizdas gautas naudojantis sukurtąja VTVS parodyti 32 pav.



32 pav. VT hierarchijos pagal dokumentų tipus ir pagal semantines kategorijas

Kiekvieną taisyklių hierarchiją, arba kiekvieną pjūvį, galima panaudoti skirtingu atžvilgiu nagrinėjant bazėje esančias taisykles. Pavyzdžiui:

- pasinaudojant VT hierarchija pagal dokumentų tipus galima taip supaprastinti veiklos taisyklių įvedimą ir modifikavimą, kad galima visiškai perduoti VT rinkinio valdymą verslo vieneto darbuotojams, t.y. sutaupyti daug VT priežiūros kaštų visiškai atsisakant IT specialistų paslaugų;
- VT hierarchiją pagal taisyklių semantines kategorijas galima naudoti optimizuojant VTVS veikimą vykdymo metu.

Galimybė sukurti naują hierarchiją pagal bet koki kriterijų, neįtakojant egzistuojančių hierarchinių ryšių ir neperprogramuojant pačios veiklos taisyklių valdymo sistemos, sukuria neribotas galimybes toliau plėtoti VTVS netrukdam egzistuojančios sistemos veikimui.

4.2. VT ir VT hierarchijos saugojimas universalioje DB

Kuriant hierarchiją buvo atsižvelgta į nefunkcinį reikalavimą nemodifikuoti egzistuojančios duomenų bazės struktūros. Šį reikalavimą įvykdyti leido hierarchijos užrašymas lygiai

tokiomis pat taisyklėmis kaip ir nagrinėjamosios veiklos taisyklės, t.y. veiklos taisyklės gali aprašyti pačios save, panašiai kaip XML kalboje galima išplėsti pačią XML kalbą.

33 pav. parodytos vienoje lentelėje kartu esančios pagrindinės taisyklės, aprašančios sąlyginius sakinius ir taisyklės nustatančios hierarchiją. Eilutės 8 ir 9 yra „where“ tipo taisyklės, tuo tarpu visos kitos eilutės yra medžio tipo taisyklės, sudarančios hierarchijas. Taisyklės tipas nustatomas pagal stulpelio LOGOP reikšmę:

1. „TRE“ – hierarchinė;
2. kita reikšmė – kitas tipas

MANDT	CRLITMID	CRLGRPID	CRLID	CRLNAME	PRECD	SEQNO	LOGOP	RLGRPID	RLID	RLNAME	RLTYPE
000	00000000000000000001	MUITINE	MUITA1	MUITAI 1	0	1	TRE	MUITAI	MUIT1	MUITAS1	A
000	00000000000000000002	MUITINE	MUITA1	MUITAI 1	0	2	TRE	MUITAI	MUIT2	MUITAS2	A
000	00000000000000000003	POSTAS	CEK11	CEKIAI 1	0	1	TRE	CEKIAI	CEK1	CEKIS 1	C
000	00000000000000000004	POSTAS	POSTAS	POSTAS	0	1	TRE	POSTAS	CEK11	CEKIAI 1	C
000	00000000000000000005	MUITINE	MUITIN	MUITINE	0	1	TRE	MUITINE	MUITA1	MUITAI 1	C
000	00000000000000000006	VT	VT	VEIKLOS TAISYKLĖS	0	1	TRE	POSTAS	POSTAS	POSTAS	C
000	00000000000000000008	CEKIAI	CEK1	CEKIS 1	0	1	AND	CEK1	C1	C1	A
000	00000000000000000009	CEKIAI	CEK1	CEKIS 1	0	2		CEK1	C2	C2	A
000	00000000000000000010	VT	VT	VEIKLOS TAISYKLĖS	0	0	TRE	MUITINE	MUITIN	MUITINE	C
000	00000000000000000011	SEM	SEM	SEMANTINE HIERARCH.	0	1	TRE	APIB	APIB	APIBREZIMO	C
000	00000000000000000012	APIB	APIB	APIBREZIMO	0	1	TRE	CEKIAL	CEK1	CEKIS 1	A
000	00000000000000000013	SEM	SEM	SEMANTINE HIERARCH.	0	2	TRE	REAK	REAK	REAKCINES	C
000	00000000000000000014	SEM	SEM	SEMANTINE HIERARCH.	0	3	TRE	ISVED	ISVED	ISVEDIMO	C
000	00000000000000000015	ISVED	ISVED	ISVEDIMO	0	1	TRE	MUITAL	MUIT1	MUITAS 2	A
000	00000000000000000016	REAK	REAK	REAKCINES	0	1	TRE	MUITAL	MUIT1	MUITAS 1	A

33 pav. ZUTL_CBUSRULS lentelė.

4.3. VT hierarchijos panaudojimas optimizuojant VTVS veikimą

Programų vykdymo metu, jos gan dažnai kreipiasi į veiklos taisyklių valdymo sistemą su užklausomis į vieną ar kitą taisyklę. Susidarius dideliame veiklos taisyklių rinkiniui, dažnos užklausos į duomenų bazę gali lėtinti programų veikimą. Todėl yra tikslinga ieškoti būdų išvengti taisyklių paieškos duomenų bazėje kiekvieną kartą gavus užklausą iš VTVS naudojančios programos.

Paprasčiausias sprendimas būtų naudoti laikinas tokios pat struktūros lenteles kaip ir pagrindinės. Tokiu būdu veiklos taisyklių lentelės užkraunamos į operatyviają atmintį ir jų valdymas patikimas DBVS. Tai efektyvus ir paprastas sprendimas, tačiau užkraunamų į atmintį lentelių dydį riboja turima operatyvioji atmintis. Jeigu lentelių dydis kelis kartus viršija turimą atmintį, rezultatas bus beveik toks pat kaip neoptimizuojant veiklos taisyklių paieškos visai: vis tiek bus reikalingas dažnas kreipimasis į kietąjį diską.

Kitas galimas sprendimas – dažniausiai naudojamas veiklos taisyklės kaupti atmintyje objektų pavidalu. Objektų panaudojimas atrodo itin patraukliai todėl, kad būtų galima panaudoti taisyklių suskirstymą į hierarchiją pagal semantines kategorijas. Programos, naudojančios veiklos taisyklės, reikiamos taisyklės visuomet ieško pagal semantinę kategoriją, kadangi programos tekste kviečiant taisyklę, visuomet žinoma ar reikalinga „order by“ tipo taisyklė, ar „where“ tipo. Jeigu VT duomenų bazė yra iš anksto suskirstyta pagal kategorijas, ženkliai susiaurėja paieškos ratas.

Nagrinėjant kiekvienos naujos įvedamos taisyklės kontekstą, galima atpažinti, kuriai semantinei kategorijai taisyklė turėtų priklausyti. Reiškia, galima sukurtą VTVS papildyti taip, kad hierarchija pagal semantines kategorijas būtų sudarinėjama automatiškai, neapkraunant vartotojo papildomu darbu.

Taisyklių kaupimui atmintyje galima pasinaudoti atvirojo kodo programa „*Condition*“. Panašiai kaip ir tiriamajame darbe, programoje taisyklės skaidomos į atomines ir jungiamos tarpusavyje.

Condition

„*Condition*“ – nemokama atvirojo kodo programa, pasak autoriaus, skirta *WHERE* sąlygoms generuoti. Jos autorius – programuotojas mėgėjas „*The Code Project*“ svetainėje pasivadinęs *WilsonProgramming* slapyvardžiu.

„*Condition*“ – tai savarankiškai *Windows* terpėje veikianti programa, kaip rezultatą gražinanti tekstinę eilutę – *WHERE* sąlygą. Visai nesudėtinga šią programą prisitaikyti savo reikmėms: ji parašyta *C#* kalba, išeities tekstai – vieši.

Sąlygos sakiniui formuoti naudojama klasė *Condition*. Klasės objektas saugo informaciją apie atominę taisyklę: dvi logines išraiškas ir jas jungiantį loginį operatorių. Vietoje loginių išraiškų galima patalpinti kitą objektą, ir, tokiu būdu, jungiant objektus suformuoti sudėtingą, tačiau palyginti lengvai modifikuojamą, sudėtinę taisyklę. [20]

4.4. Veiklos taisyklių įvedimo tobulinimas

Naujų taisyklių įvedimas

Dar vienas būdas išplėsti sukurtą VTVS, ir bemaž būtų labiausiai pageidautinas – patobulinti taisyklių įvedimo sistemą taip, kad jas galėtų įvesti verslo vieneto darbuotojas.

Ši problema vis dar išlieka pati opiausia. Ji nėra išspręsta ir kitose komercinėse veiklos taisyklių valdymo sistemose. Universalaus būdo bet kokioms veiklos taisyklėms įvesti, ne VT

specialistui suprantama forma, turbūt nėra visai. Taigi veiklos taisyklių įvedimo forma turi priklausyti nuo apdorojamo taisyklių rinkinio.

Mūsų atveju didžiausius keblumus sudaro tai, kad verslo vieneto žmogus vieną taisyklę suvokia kaip vieno dokumento aprašymą, t.y. jį apibūdinančių laukų išvardijimą. Tačiau iš tikrųjų apie vieną dokumentą turi būti sukurtos keletas taisyklių: „order by“ ir „where“ tipų. Galimas netgi toks atvejis, kad, pavyzdžiui „order by“ tipo taisyklių vienam dokumentui reikės dviejų. Tuo tarpu verslo vieneto darbuotojas nėra programuotojas ir nesupranta tokių sąvokų.

Yra galimas dalinis sprendimas šiam konkrečiam atvejui. Konsultuojantis su programuotojais reikėtų nustatyti, kokio tipo taisyklės pasitaiko dažniausiai, ir kurti tokią vartotojo sąsają, kuri vartotojui nurodžius dokumentą apibūdinančius laukus ir jų reikšmes, sukurtų dažniausiai naudojamas taisykles. Tuomet programuotojams nebereikėtų vesti į sistemą kiekvienos taisyklės, o tik tas kurios neatitinka nustatytų standartinių taisyklių.

Bet kuriuo atveju, mano paminėtos hipotezės reikalauja detalaus tyrimo. Tačiau išlieka rizika, kad atlikus tyrimą paaiškės, kad aprašyta grafinė sąsaja neatpirks jai kurti skirtų investicijų.

Egzistuojančių taisyklių modifikavimas

Laimei, naujų taisyklių įvedimas nėra toks dažnas įvykis kaip egzistuojančių taisyklių modifikavimas. Tobulinant grafinę sąsają taisyklių modifikavimui, galima pasinaudoti sukurta veiklos taisyklių hierarchija pagal dokumentų tipus.

Hierarchijos pagalba galima lengvai atsidaryti visas taisykles susijusias su tam tikru dokumentu. Joms atvaizduoti sukūrus formą, šią užduotį galima patikėti verslo vieneto darbuotojui.

5. IŠVADOS

1. Norint patobulinti esamą *ERP* taisyklių valdymo karkasą, buvo atlikta veiklos taisyklių modeliavimo kalbų ir taisyklių valdymo sistemų analizė.
2. Pasiūlytas sprendimas vaizduoti taisykles ir jų rinkinius medžio struktūromis.
3. Pasiūlyta tą patį taisyklių rinkinį klasifikuoti pagal skirtingus kriterijus skirtingiems tikslams. Tai atveria dideles galimybes toliau plėtoti sistemą, netrukdam jai lygiagrečiai veikti.
4. Sukurta veiklos taisyklių valdymo sistema yra pirmoji tokia *SAP* aplinkoje.
5. Ji realizuota kaip nepriklausomas komponentas, todėl galėtų būti pritaikyta bet kuriai *SAP* aplinkoje veikiančiai sistemai kurti ar modifikuoti.
6. Sukurta sistema palengvina vartotojams veiklos taisyklių įvedimą bei modifikavimą, sumažina klaidų tikimybę ir leidžia greičiau pritaikyti turimą programinę įrangą prie veiklos pokyčių.

6. LITERATŪRA

- Amit - Active Middleware Technology [žiūrėta 2005 m. sausio 10 d.]. Prieiga per internetą: <<http://www.haifa.il.ibm.com/projects/software/amit/tools.html>>
- AnswerBag [žiūrėta 2004 m. gruodžio 18 d.]. Prieiga per internetą: <http://www.answerbag.com/q_view.php/10743 >
- Business rules community. Frequently Asked Questions [žiūrėta 2004 m. gruodžio 13 d.]. Prieiga per internetą: < <http://www.brcommunity.com/faqs.php> >
- DFKI - About us [žiūrėta 2005 m. sausio 10 d.]. Prieiga per internetą: <<http://www.dfki.de/web/about/index.en.html>>
- FairIsaac. Blaze Advisor [žiūrėta 2004 m. gruodžio 14 d.]. Prieiga per Internetą: <<http://www.fairisaac.com/Fairisaac/Solutions/Enterprise+Decision+Management/Business+rules/Blaze+Advisor/>>
- ILOG JRules [žiūrėta 2004 m. gruodžio 15 d.]. Prieiga per Internetą: <<http://www.ilog.com/products/jrules/>>
- J.Sinur. Gartner. The Business Rule Engine 2003 Magic Quadrant [žiūrėta 2004 m. gruodžio 16 d.]. Prieiga per internetą: <<http://www.gartner.com/reprints/fairisaac/114166.html>>
- javarules.org. Rule Languages [žiūrėta 2004 m. gruodžio 16 d.]. Prieiga per internetą: <<http://www.javarules.org/modules.php?op=modload&name=News&file=article&sid=6>>
- KAIST. About KAIST - Missions And Origins [žiūrėta 2005 m. sausio 10 d.]. Prieiga per internetą: < <http://www.kaist.edu/> >
- Meta Object Facility Specification. [žiūrėta 2005 m. sausio 20 d.]. Prieiga per internetą: <<http://www.omg.org/docs/formal/02-04-03.pdf>>
- MOF-RuleML: The Abstract Syntax of RuleML as a MOF Model. Wagner, Gerd; Tabet, Said; Boley, Harold [žiūrėta 2005 m. sausio 20 d.]. Prieiga per internetą: <<http://www.omg.org/docs/br/03-10-02.pdf>>
- Rosenberg F., Dustdar S. Design and Implementation of a Service-Oriented Business Rules Broker. Proceedings of the 2005 Seventh IEEE International Conference on E-Commerce Technology Workshops (CECW'05). Viena, 2005, p. 2-3.

- Rules4J [žiūrėta 2004 m. gruodžio 15 d.]. Prieiga per Internetą:
<<http://www.rules4j.com/>>
- Said Tabet. University of Massachusetts Boston Computer Science Colloquium. Business Rules Markup, Modelling, and Interchange: A Model Driven approach using RuleML. Inference Web Inc. and the RuleML Initiative [žiūrėta 2004 m. gruodžio 14 d.]. Prieiga per internetą:
<http://www.cs.umb.edu/News_Events/Lectures_Seminars/Colloquiums/talk04_11_17_1.htm>
- Taylor J. Achieving decision consistency across SOA-based Enterprise. A FairIsaak white paper.2005, p. 1-2.
- The Business Rules Group. Defining Business Rules [žiūrėta 2004 m. gruodžio 17 d.]. Prieiga per internetą: < <http://www.businessrulesgroup.org/brgdefn.htm> >11
- The Rule Markup Initiative [žiūrėta 2004 m. gruodžio 14 d.]. Prieiga per internetą:
<<http://www.ruleml.org/>>
- W3C. Extensible Markup Language (XML) 1.0 (Third Edition). W3C Recommendation 04 February 2004 [žiūrėta 2004 m. gruodžio 18 d.]. Prieiga per internetą:
<<http://www.w3.org/TR/REC-xml/>>
- Wikipedia. Business rule definition. [žiūrėta 2006 m. balandžio 20 d.]. Prieiga per internetą: < http://en.wikipedia.org/wiki/Business_rule >
- WilsonProgramming. The Code Project. Creating on-the-fly SQL "WHERE" clauses. [žiūrėta 2004 m. gruodžio 2 d.]. Prieiga per internetą:
<<http://www.codeproject.com/info/error404.asp?404>; http://www.codeproject.com/cs/database/Database_Conditions.asp>

7. SANTRUMPŲ IR TERMINŲ ŽODYNAS

VT - Veiklos taisyklė (*business rule*) – teiginys, kuris nurodo arba apibrėžia tam tikrą verslo aspektą. [16].

VTVS – veiklos taisyklių valdymo sistema.

SAP („*Systeme, Anwendungen, Produkte in der Datenverarbeitung*“) – Vokietijos programinės įrangos firma. Taip pat vadinama ir aplinka kurioje veikia šios kompanijos sukurta veiklos valdymo sistema *SAP R/3* [2]

XML (*Extensible Markup Language*) - SGML kalbos poaibis. Tikslas – suteikti galimybę pateikti SGML tinkle panašiai kaip yra pateikiama HTML. [18]

SRL (*Structured Rule Language*) – Struktūrizuota taisyklių kalba, panaši į natūralią anglų kalbą. [1]

DFKI (*Deutsches Forschungszentrum für Künstliche Intelligenz*) – vienas didžiausių dirbtinio intelekto pagrindu veikiančios programinės įrangos tyrimo institutų pasaulyje. [4]

KAIST (*Korea Advanced Institute of Science and Technology*) – žymus tarptautinio lygio technikos mokslų institutas Korėjoje. [9]

MOF (*Meta-Object Facility*) – konsorciumo OMG sukurtas įrankis – abstrakti kalba ir karkasas skirti nuo technologijų nepriklausomiems metamodeliams apibrėžti, konstruoti ir valdyti. [10]

8. PRIEDAI

8.1. Reikalavimų specifikacija

1. PROJEKTO VAROVAI

1.1. *Sistemos paskirtis*

1.1.1. Projekto kūrimo pagrindas (pagrindimas)

Kompanijos dažnai susiduria su įvairiomis veiklos taisyklėmis. Nuolat kintančias taisykles sąlygoja valstybinės institucijos, verslo vienetai, klientai, konkurentai ir bendros rinkos sąlygos. Veiklos taisyklės įtakoja ir valdo žmonių elgesį (verslo pažiūriu), taip pat įtakoja informaciją, kuri yra įrašoma į informacines sistemas bei apriboja šios informacijos modifikavimą (informacinių pažiūriu). Verslo pažiūriu gerinti veiklos taisyklių valdymą būtina tam, kad padidinti organizacijos žinojimą apie tai kaip, kodėl, kada, kur ir kas aktyvuoja taisykles bei, kad pagerinti šių procesų valdymą. Geresnis veiklos taisyklių išmanymas ir jų valdymas gali padėti atsakyti į šiuos klausimus:

- Nuostatų laikymąsi jūsų organizacijoje;
- Kokia rizika iškyla organizacijai, darbuotojams paliekant kompaniją;
- Organizacijos strategijos lankstumą;
- Galimybę staiga pakeisti organizacijos politiką ar įdiegti naujas nuostatas.

Lietuvos muitinėje naudojama labai daug įvairių tipų dokumentų. Vienas tipas nuo kito atskiriamas pagal keleto laukų kombinacijos reikšmes. Taip pat, egzistuoja taisyklės siejančios vienus dokumentus su kitais. Kuriant ir palaikant Lietuvos muitinės informacinę sistemą daugybėje vietų neišvengiamai tenka atrinkti vieno ar kito tipo dokumentus bei susieti vienu dokumentu informaciją su kitų dokumentų informacija. Šį procesą dar labiau apsunkina tai, kad keičiantis įstatymams atsiranda nauji dokumentų tipai, naikinami ar modifikuojami egzistuojantys.

Turimas veiklos taisyklių valdymo karkasas yra pakankamai universalus, tačiau labai sudėtingas ir reikalauja visiško karkaso architektūros išmanymo bei atima daug laiko taisyklių modifikavimui. Taisyklės vedamos tiesiai į lenteles, nėra įrankio taisyklių sintaksinio teisingumo tikrinimui, todėl yra labai didelė įvedimo klaidų tikimybė.

1.1.2. Sistemos tikslai (paskirtis)

Sistemos tikslas – sukurti grafinę vartotojo sąsają taisyklių įvedimui ir atvaizdavimui. Grafinė sąsaja turi:

- užtikrinti įvedamų taisyklių sintaksinį teisingumą;
- palengvinti taisyklių paiešką;

- atvaizduoti taisyklių tarpusavio sąsają.

1.2. Užsakovai, pirkėjai ir kiti sistema suinteresuoti asmenys

1.2.1. Užsakovas

Produkto užsakovas – UAB „ERP“. Visos jau sukurtos veiklos taisyklių karkaso autorinės teisės priklauso UAB „ERP“. Naujosios VT karkaso grafinės sąsajos autorinės teisės taip pat priklausys šiai įmonei.

1.2.2. Pirkėjai

Sukurtas produktas skirtas vidiniam UAB „ERP“ naudojimui. Jis bus naudojamas kaip sudėtinė Lietuvos Muitinės informacinės sistemos dalis.

1.2.3. Kiti sistema suinteresuoti asmenys

Magistrinio darbo vadovas.

1.3. Vartotojai

Informacinės sistemos projektuotojas

- Vartotojo sprendžiami uždaviniai (atliekamos funkcijos):
 - Veiklos taisyklių aprašymas;
 - Veiklos taisyklių klasifikavimas.
 - Karkaso funkcijų naudojimas informacinės sistemos moduluose.
- Patirtis dalykinėje srityje – vartotojas turintis programuotojo išsilavinimą, turintis patirties darbe su SAP sistema.
- Patirtis informacinėse technologijose – informacinių sistemų programuotojo darbo patirtis.
- Papildomos vartotojo charakteristikos:
 - veiklos taisyklių išmanymas;
 - anglų kalbos žinios.

2. PROJEKTO APRIBOJIMAI

2.1. Įpareigojantys apribojimai

2.1.1. Apribojimai sprendimui

Karkaso grafinė sąsaja turi būti kuriama naudojant ABAP programavimo kalbą, SAP aplinkoje. Grafinė sąsaja turi būti pilnai suderinama su egzistuojančiomis karkaso dalimis.

2.1.2. Diegimo aplinka

Kuriama grafinė aplinka bus diegiama SAP tinklo aplikacijų tarnybinėje stotyje (Web Application Server) Lietuvos muitinės informacinėje sistemoje.

2.1.3. Bendradarbiaujančios sistemos

Kuriama grafinė sąsaja neturės tiesioginių ryšių su kitais posistemiais. Ji bus veiklos taisyklių karkaso sudedamoji dalis.

2.1.4. Komerciniai specializuoti programų paketai

Nėra

2.1.5. Numatoma darbo vietos aplinka

Darbo su sistema vietos bus įrengtos įprastoje biuro aplinkoje.

2.1.6. Sistemos kūrimo terminai

Sistemos kūrimo terminai sudaryti remiantis modulio „Programų sistemų architektūros analizė“ darbų tvarkos grafiku. Numatomi sistemos kūrimo ir tarpinių atsiskaitymų terminai pateikiami 4.1 lentelėje.

4.1 lentelė

Eil. Nr.	Atsiskaitymas	Data
1	Reikalavimų specifikacija	2005.03.10
2	Architektūros specifikacija	2005.04.15
3	Detali architektūros specifikacija	2005.05.20
4	Sistemos įdiegimas	2005.12.20
5	Pilnos sistemos dokumentacijos paruošimas	2005.01.05

2.1.7. Sistemos kūrimo biudžetas

Nėra

2.2. Terminų žodynas

Karkasas (angl. Framework) – pakartotiniam panaudojimui skirtų programinių modulių rinkinys.

Veiklos taisyklė – išraiška, kuri apibrėžia arba apriboja tam tikrus veiklos aspektus, pvz., tam, kad užtikrinti tam tikrą struktūrinę veiklos būseną ar kontroliuoti bei įtakoti veiklos elgseną.

Atominės taisyklės – atominę taisyklę sudaro vienas kintamasis ir viena reikšmių aibė.

Sudėtinės taisyklės – sudėtinė taisyklė gali būti sudaryta iš keleto atominių taisyklių ar netgi iš keleto kitų sudėtinių taisyklių.

2.3. Svarbūs faktai ir prielaidos

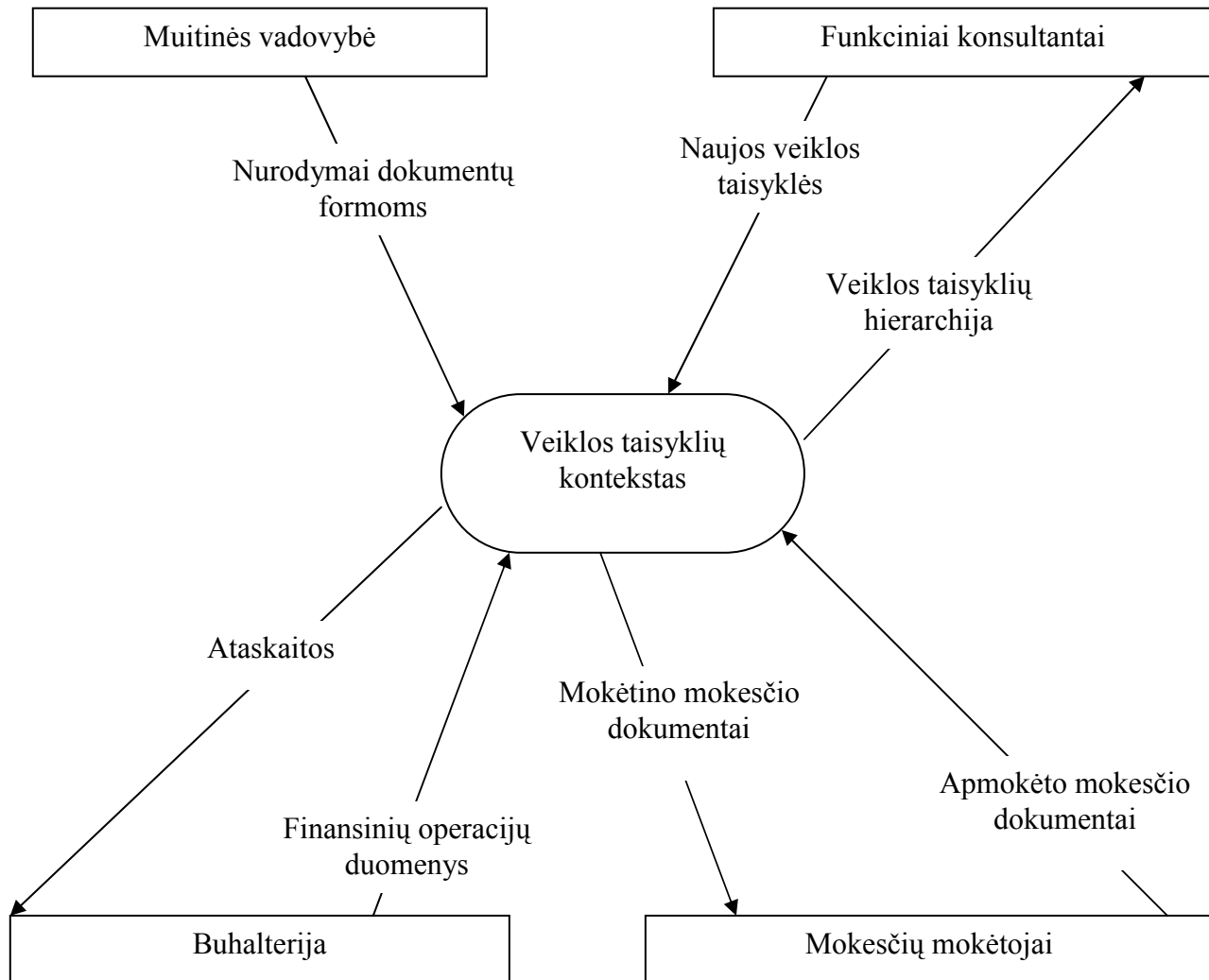
Nėra.

3. FUNKCINIAI REIKALAVIMAI

3.1. Veiklos sudėtis

3.1.1. Veiklos kontekstas

Veiklos konteksto diagrama pateikta 7.1 paveikslėlyje.



7.1 pav. Veiklos konteksto diagrama

3.1.2. Veiklos padalinimas

Veiklos įvykių sąrašas pateikiamas 7.1 lentelėje.

7.1 lentelė

Eil. Nr.	Įvykio pavadinimas	Įeinantys/išeinantys informacijos srautai
1	Muitinės vadovybė duoda nurodymus dokumentų formoms	Nurodymai dokumentų formoms (in)
2	Funkciniai konsultantai įveda naujas veiklos taisykles	Naujos veiklos taisyklės (in)
3	Funkciniai konsultantai revizuoja veiklos taisyklių hierarchiją	Veiklos taisyklių hierarchija (out)
4	Mokesčių mokėtojai gauna užklausas mokėtiniams mokesčiams	Mokėtino mokesčio dokumentai (out)
5	Mokesčių mokėtojai pristato apmokėtų mokesčių dokumentus	Apmokėto mokesčio dokumentai (in)
6	Buhalterija perduoda finansinių operacijų duomenis	Finansinių operacijų duomenys (in)
6	Buhalterija gauna ataskaitas apie įvykdytas mokesčių užklausas	Finansinių operacijų duomenys (in)

3.2. Sistemos sudėtis

3.2.1. Sistemos ribos

Ribas tarp sistemos ir vartotojo nusako panaudojimo atvejų diagrama. Kuriamojo sistemos modulio panaudojimo atvejų diagrama pateikiama 8.1 paveiksle.



8.1 pav. Panaudojimo atvejų diagrama

3.2.2. Panaudojimo atvejų sąrašas

1. Panaudojimo atvejis	Ivesti naują įrašą
Tikslas	Įvesti naują veiklos taisyklę arba reikšmių aibę
Aktoriai	Funkcinis konsultantas
Ryšiai su kitais PA	Apima panaudojimo atvejus: „redaguoti atominę taisyklę“, „redaguoti sudėtinę taisyklę“ ir „redaguoti reikšmių aibę“; gali būti išplėstas iki panaudojimo atvejo „patalpinti VT hierarchijoje“.
Nefunkciniai reikalavimai	2
Prieš-sąlygos	Nėra.
Sužadinimo sąlyga	Vartotojas iš meniu pasirenka naujo įrašo įvedimo komandą.
Po-sąlygos	Naujas įrašas įrašytas.
Pagrindinis scenarijus	Sistemos reakcija ir sprendimai
1. Pasirenkamas naujos VT įvedimas	1.1. Atidaroma naujo įrašo įvedimo forma.
2. Pasirenkamas panaudojimo atvejis: „redaguoti atominę taisyklę“, „redaguoti sudėtinę taisyklę“, „redaguoti reikšmių aibę“)	2.1. Sukuriamas naujas įrašas atitinkamoje lentelėje. 2.2. Pereinama į pasirinktą panaudojimo atvejį.
3. Jei norima, pereinama į panaudojimo atvejį „patalpinti VT hierarchijoje“	
4. Paspaudžiamas formos uždarymo mygtukas.	4.1. Forma uždaroma.
Alternatyvūs scenarijai	Vartotojas uždaro formą nepasirinkęs jokios komandos.

2. Panaudojimo atvejis	Peržiūrėti VT hierarchiją
Tikslas	Peržiūrėti veiklos taisyklių hierarchiją.
Aktoriai	Funkcinis konsultantas
Ryšiai su kitais PA	Gali būti išplėstas iki panaudojimo atvejo „redaguoti įrašą“.
Nefunkciniai reikalavimai	1, 2, 3
Prieš-sąlygos	Sistemoje yra bent viena taisyklė.
Sužadinimo sąlyga	Vartotojas iš meniu pasirenka veiklos taisyklių peržiūros komandą.
Po-sąlygos	Nėra.
Pagrindinis scenarijus	Sistemos reakcija ir sprendimai
1. Pasirenkama VT hierarchijos peržiūra	1.1. Atidaroma forma vaizduojanti aukščiausią VT hierarchijos lygmenį.
2. Išskleidžiamas norima hierarchijos šaka.	
3. Jei norima, pereinama į panaudojimo atvejį „redaguoti įrašą“	
4. Paspaudžiamas formos uždarymo mygtukas.	4.1. Forma uždaroma.

Alternatyvūs scenarijai	Nėra
--------------------------------	------

3. Panaudojimo atvejis	Redaguoti įrašą
Tikslas	Redaguoti veiklos taisyklę arba reikšmių aibę.
Aktoriai	Funkcinis konsultantas
Ryšiai su kitais PA	Apima panaudojimo atvejus: „redaguoti atominę taisyklę“, „redaguoti sudėtinę taisyklę“ ir „redaguoti reikšmių aibę“; gali būti išplėstas iki panaudojimo atvejo „patalpinti VT hierarchijoje“.
Nefunkciniai reikalavimai	2
Prieš-sąlygos	Sistemoje yra bent viena taisyklė.
Sužadinimo sąlyga	Vartotojas VT hierarchijos formoje pasirenka taisyklės arba reikšmių aibės redagavimo komandą..
Po-sąlygos	Egzistuojantis įrašas pakoreguotas ir išsaugotas.
Pagrindinis scenarijus	Sistemos reakcija ir sprendimai
1. Pasirenkamas įrašo koregavimas	1.1. Atidaroma įrašo redagavimo forma.
2. Jei norima įrašą redaguoti, pasirenkama redagavimo komanda.	2.1. Priklausomai nuo atidaryto įrašo pereinama į vieną iš panaudojimo atvejų: „redaguoti atominę taisyklę“, „redaguoti sudėtinę taisyklę“, „redaguoti reikšmių aibę“.
3. Jei norima, pereinama į panaudojimo atvejį „patalpinti VT hierarchijoje“	
4. Paspaudžiamas formos uždarymo mygtukas.	4.1. Forma uždaroma.
Alternatyvūs scenarijai	Vartotojas uždaro formą nepasirinkęs jokios komandos.

4. Panaudojimo atvejis	Redaguoti sudėtinę taisyklę
Tikslas	Redaguoti sudėtinę veiklos taisyklę.
Aktoriai	Funkcinis konsultantas
Ryšiai su kitais PA	Gali būti išplėstas iki panaudojimo atvejo „redaguoti atominę taisyklę“.
Nefunkciniai reikalavimai	2
Prieš-sąlygos	Sistemoje yra bent viena taisyklė.
Sužadinimo sąlyga	Vartotojas iš naujo įrašo įvedimo formos arba iš įrašo redagavimo pasirinko sudėtinės taisyklės redagavimą.
Po-sąlygos	Sudėtinė taisyklė poredaguota ir įrašyta lentelėje.
Pagrindinis scenarijus	Sistemos reakcija ir sprendimai
1. Pasirenkamas sudėtinės taisyklės redagavimas	1.1. Atidaroma sudėtinių taisyklių lentelė ir pagalbinių įrankių juosta.

2. Redaguojami taisyklės identifikatoriai	
3. Jei norima, pereinama į panaudojimo atvejį „redaguoti atominę taisyklę“	
4. Redaguojamas pirmos taisyklės identifikatorius	
5. Redaguojamas loginis operatorius	
6. Redaguojamas antros taisyklės identifikatorius	
7. Patvirtinami įvesti duomenys	7.1. VT duomenys išsaugomi, sudėtinių taisyklių lentelė ir įrankių juosta uždaromi.
Alternatyvūs scenarijai	Vartotojas uždaro formą dar nesuvedęs visų duomenų.

5. Panaudojimo atvejis	Redaguoti atominę taisyklę
Tikslas	Redaguoti atominę veiklos taisyklę.
Aktoriai	Funkcinis konsultantas
Ryšiai su kitais PA	Gali būti išplėstas iki panaudojimo atvejo „redaguoti reikšmių aibę“.
Nefunkciniai reikalavimai	2
Prieš-sąlygos	Sistemoje yra bent viena taisyklė.
Sužadinimo sąlyga	Vartotojas iš naujo įrašo įvedimo formos arba iš įrašo redagavimo formos, arba iš sudėtinės taisyklės redagavimo įrankių juostos pasirinko atominės taisyklės redagavimą.
Po-sąlygos	Atominė taisyklė poredaguota ir įrašyta lentelėje.
Pagrindinis scenarijus	Sistemos reakcija ir sprendimai
1. Pasirenkamas atominės taisyklės redagavimas	1.1. Atidaroma atominių taisyklių lentelė ir pagalbinių įrankių juosta.
2. Redaguojami taisyklės identifikatoriai	
3. Redaguojamas kintamasis	
4. Redaguojamas loginis operatorius	
5. Jei norima, pereinama į panaudojimo atvejį „įvesti reikšmių aibę“	
6. Patvirtinami įvesti duomenys	6.1. VT duomenys išsaugomi, atominių taisyklių lentelė ir įrankių juosta i
Alternatyvūs scenarijai	Vartotojas uždaro formą dar nesuvedęs visų duomenų.

6. Panaudojimo atvejis	Redaguoti reikšmių aibę
Tikslas	Redaguoti reikšmių aibę.
Aktoriai	Funkcinis konsultantas
Ryšiai su kitais PA	Nėra
Nefunkciniai reikalavimai	2
Prieš-sąlygos	Sistemoje yra bent viena reikšmių aibė.

Sužadavimo sąlyga	Vartotojas iš naujo įrašo įvedimo formos arba iš įrašo redagavimo formos, arba iš atominės taisyklės redagavimo įrankių juostos pasirinko reikšmių aibės redagavimą.
Po-sąlygos	Reikšmių aibė pareduota ir įrašyta lentelėje.
Pagrindinis scenarijus	Sistemos reakcija ir sprendimai
1. Pasirenkamas reikšmių aibės redagavimas	1.1. Atidaroma reikšmių aibių lentelė.
2. Redaguojami reikšmių aibės identifikatoriai	
3. Redaguojamas reikšmių aibės ženklas	
4. Redaguojamos reikšmių aibės ribos	
5. Patvirtinami įvesti duomenys	5.1. Reikšmių aibės duomenys išsaugomi, reikšmių aibės lentelė ir įrankių juosta uždaromi.
Alternatyvūs scenarijai	Vartotojas uždaro formą dar nesuvedęs visų duomenų.

7. Panaudojimo atvejis	Patalpinti VT hierarchijoje
Tikslas	Patalpinti veiklos taisyklę hierarchijoje.
Aktoriai	Funkcinis konsultantas
Ryšiai su kitais PA	Nėra
Nefunkciniai reikalavimai	2
Prieš-sąlygos	Sistemoje yra bent viena taisyklė.
Sužadavimo sąlyga	Vartotojas iš naujo įrašo įvedimo formos arba įrašo redagavimo formos pasirinko komandą „patalpinti VT hierarchijoje“.
Po-sąlygos	Veiklos taisyklės vieta hierarchijoje pakeista.
Pagrindinis scenarijus	Sistemos reakcija ir sprendimai
1. Pasirenkamas veiklos taisyklės vietos hierarchijoje keitimas	1.1. Atidaroma veiklos taisyklių hierarchijos forma.
2. Grafiškai pele nurodoma naujoji VT vieta hierarchijoje.	
3. Naujoji vieta patvirtinama atsakymu į klausimą „Taip“.	3.1. Naujoji VT vieta hierarchijoje išsaugoma. 3.2. Veiklos taisyklių hierarchijos forma automatiškai uždaroma.
Alternatyvūs scenarijai	Vartotojas nepatvirtina naujosios VT vietos hierarchijoje.

3.3. Funkciniai reikalavimai ir reikalavimai duomenims

3.3.1. Funkciniai reikalavimai

Reikalavimo #: 1	Reikalavimo tipas: Funkcinis	Panaudojimo atvejis #: 1
Aprašymas: Turi būti galimybė įvesti 3 tipų naujas taisykles: where, order by, if.		
Pagrindimas: Karkasas šiuo metu palaiko 3 tipų taisykles: where, order by, if..		
Šaltinis: Užsakovas		
Tinkamumo kriterijus: Yra galimybė įvesti 3 tipų naujas taisykles: where, order by, if.		
Užsakovo patenkinimas: 5		Užsakovo nepatenkinimas: 5
Priklausomybės: Nėra		Konfliktai: Nėra
Papildoma medžiaga: Nėra		
Istorija: Užregistruota 2004.11.11		

Reikalavimo #: 2	Reikalavimo tipas: Funkcinis	Panaudojimo atvejis #: 2
Aprašymas: Grafinis taisyklių hierarchijos vaizdas.		
Pagrindimas: Dėl veiklos taisyklių sudėtingumo, sistemos derinimo metu būtinas maksimalus galimas aiškumas, norint išvengti klaidų.		
Šaltinis: Užsakovas		
Tinkamumo kriterijus: Veiklos taisyklių hierarchija pateikta kaip grafinis vaizdas.		
Užsakovo patenkinimas: 5		Užsakovo nepatenkinimas: 4
Priklausomybės: Nėra		Konfliktai: Nėra
Papildoma medžiaga: Nėra		
Istorija: Užregistruota 2005.04.26		

Reikalavimo #: 3	Reikalavimo tipas: Funkcinis	Panaudojimo atvejis #: 2
Aprašymas: Turi būti galimybė kiekvieną taisyklę atsidaryti lentelėje ir redaguoti.		
Pagrindimas: Funkcinis konsultantas pageidauja turėti galimybę pasiekti taisykles lentelėse tiesiai iš hierarchinio vaizdo kad sutaupyti daug laiko.		
Šaltinis: Užsakovas		
Tinkamumo kriterijus: Kiekvieną taisyklę galima atsidaryti lentelėje tiesiai iš hierarchinio vaizdo.		
Užsakovo patenkinimas: 5		Užsakovo nepatenkinimas: 5
Priklausomybės: Nėra		Konfliktai: Nėra
Papildoma medžiaga: Nėra		

Istorija: Užregistruota 2005.04.26

Reikalavimo #: 4	Reikalavimo tipas: Funkcinis	Panaudojimo atvejis #: 3
Aprašymas: Redagavimui atidarytoje lentelėje turi būti atfiltruoti tik su pasirinkta taisykle susiję įrašai.		
Pagrindimas: Funkcinis konsultantas pageidauja redagavimui atidarytoje lentelėje matyti atfiltruotus tik su pasirinkta taisykle susijusius įrašus.		
Šaltinis: Užsakovas		
Tinkamumo kriterijus: Redagavimui atidarytoje lentelėje matomi atfiltruoti tik su pasirinkta taisykle susiję įrašai.		
Užsakovo patenkinimas: 5	Užsakovo nepatenkinimas: 5	
Priklausomybės: Nėra	Konfliktai: Nėra	
Papildoma medžiaga: Nėra		
Istorija: Užregistruota 2005.04.26		

Reikalavimo #: 5	Reikalavimo tipas: Funkcinis	Panaudojimo atvejis #: 2
Aprašymas: Turi būti galimybė peržiūrėti taisyklės vidinės hierarchijos grafinį vaizdą.		
Pagrindimas: Galimybė matyti vidinę taisyklės hierarchiją žymiai padidina taisyklės suprantamumą ir sumažina klaidų tikimybę.		
Šaltinis: Užsakovas		
Tinkamumo kriterijus: Yra galimybė peržiūrėti grafiškai taisyklės vidinės hierarchijos vaizdą.		
Užsakovo patenkinimas: 5	Užsakovo nepatenkinimas: 5	
Priklausomybės: Nėra	Konfliktai: Nėra	
Papildoma medžiaga: Nėra		
Istorija: Užregistruota 2005.04.26		

Reikalavimo #: 6	Reikalavimo tipas: Funkcinis	Panaudojimo atvejis #: 2
Aprašymas: Taisyklių susietų hierarchiniais ryšiais hierarchija ir vidinė taisyklių hierarchija turi būti pateikiamos atskiruose languose.		
Pagrindimas: Aiškios atskirtis tarp taisyklių susietų hierarchiniais ryšiais ir vidinės taisyklės hierarchijos nebuvimas žymiai padidintų klaidų tikimybę.		
Šaltinis: Užsakovas		
Tinkamumo kriterijus: Taisyklių susietų hierarchiniais ryšiais hierarchija ir vidinė taisyklių hierarchija pateikiamos atskiruose languose.		

Užsakovo patenkinimas: 5	Užsakovo nepatenkinimas: 5
Priklausomybės: Nėra	Konfliktai: Nėra
Papildoma medžiaga: Nėra	
Istorija: Užregistruota 2005.04.26	

3.3.2. Reikalavimai duomenims

Turi būti naudojama egzistuojant karkaso duomenų struktūra. Karkase naudojamos trys nesusietos lentelės. Jų aprašymai pateikiami paveikslėlyje 9.1.

ZUTL_BUSRULS	ZUTL_VALRNGS	ZUTL_CBUSRULS
MANDT RLITMID RLGRPID RLID RLNAME PRECD SEQNO LOGOP FLDNM CMPOP RNGNM	MANDT RNGID RLGRPID RLID RLNAME RNGNM SGN OPT LOW HIGH	MANDT CRLITMID CRLGRPID CRLID CRLNAME PRECD SEQNO LOGOP RLGRPID RLID RLNAME RLTYPE

9.1 pav. Duomenų bazės struktūra

4. NEFUNKCINIAI REIKALAVIMAI

4.1. Reikalavimai sistemos išvaizdai

Reikalavimo #: 1	Reikalavimo tipas: Išvaizda	Panaudojimo atvejis #: 2
Aprašymas: Veiklos taisyklių hierarchijos pateikimas turi būti panašus į Explorer programoje naudojamą katalogų medį.		
Pagrindimas: Kadangi vartotojai yra pratę prie Explorer tipo medžio hierarchinio vaizdo, nebus gaišamas laikas adaptacijai..		
Šaltinis: Užsakovas		
Tinkamumo kriterijus: Veiklos taisyklių hierarchija atvaizduojama panašiai į Explorer medį..		
Užsakovo patenkinimas: 5	Užsakovo nepatenkinimas: 4	
Priklausomybės: Nėra	Konfliktai: Nėra	
Papildoma medžiaga: Nėra		
Istorija: Užregistruota 2005.05.05		

4.2. Reikalavimai panaudojamumui

Reikalavimo #: 2	Reikalavimo tipas: Panaudojamumo	Panaudojimo atvejis #: 1, 2, 3, 4, 5, 6, 7
Aprašymas: Sistema turi būti nesudėtinga naudotis funkciniais konsultantams.		
Pagrindimas: Prieš naudojant sistemą neturi reikėti dalyvauti papildomuose mokymuose.		
Šaltinis: Užsakovas		
Tinkamumo kriterijus: Funkcinis konsultantas turi sugebėti įsisavinti sistemos naudojimą visų panaudojimo atvejų spektre per 1 darbo dieną nuo darbo su sistema pradžios.		
Užsakovo patenkinimas: 3	Užsakovo nepatenkinimas: 3	
Priklausomybės: Nėra	Konfliktai: Nėra	
Papildoma medžiaga: Nėra		
Istorija: Užregistruota 2005.05.05		

4.3. Reikalavimai vykdymo charakteristikoms

Reikalavimo #: 3	Reikalavimo tipas: Vykdymo	Panaudojimo atvejis #: 2
Aprašymas: Sistema turi atvaizduoti hierarchiją be klaidų.		
Pagrindimas: Hierarchija neturi klaidinti funkcinį konsultantą.		
Šaltinis: Užsakovas		
Tinkamumo kriterijus: Privalomas 100% tikslumas atvaizduojant VT hierarchiją..		
Užsakovo patenkinimas: 5	Užsakovo nepatenkinimas: 5	
Priklausomybės: Nėra	Konfliktai: Nėra	
Papildoma medžiaga: Nėra		

Istorija: Užregistruota 2004.11.11

4.4. Reikalavimai veikimo sąlygoms

Reikalavimo #: 4	Reikalavimo tipas: Veikimo	Panaudojimo atvejis #:
Aprašymas: Kuriamas modulis turi būti rašomas SAP sistemos palaikoma kalba ABAP.		
Pagrindimas: Reikalaujama kad modulis veiktų SAP Web Application Server aplinkoje..		
Šaltinis: Užsakovas		
Tinkamumo kriterijus: Kuriamas modulis parašytas ABAP kalba.		
Užsakovo patenkinimas: 5	Užsakovo nepatenkinimas: 5	
Priklausomybės: Nėra	Konfliktai: Nėra	
Papildoma medžiaga: Nėra		
Istorija: Užregistruota 2005.02.01		

4.5. Reikalavimai sistemos priežiūrai

Reikalavimo #: 5	Reikalavimo tipas: Priežiūros	Panaudojimo atvejis #:
Aprašymas: Turi būti galimybė nesunkiai atnaujinti sukurtą modulį.		
Pagrindimas: Net ir po įdiegimo, sistema bus vystoma.		
Šaltinis: Užsakovas		
Tinkamumo kriterijus: Sistemos išėities tekstuose turi būti naudojami išsamūs komentarai.		
Užsakovo patenkinimas: 3	Užsakovo nepatenkinimas: 3	
Priklausomybės: Nėra	Konfliktai: Nėra	
Papildoma medžiaga: Nėra		
Istorija: Užregistruota 2005.02.02		

4.6. Reikalavimai saugumui

Reikalavimo #: 6	Reikalavimo tipas: Saugumo	Panaudojimo atvejis #:
Aprašymas: Sistemos duomenys turi būti apsaugoti nuo neteisėtos prieigos.		
Pagrindimas: Sistemoje saugomomis veiktos taisyklėmis remiasi didelė dalis visos Lietuvos muitinės informacinės sistemos, todėl jų saugumas yra ypač svarbus.		
Šaltinis: Užsakovas		
Tinkamumo kriterijus: Sistemos duomenys turi būti prieinami tik vartotojams, kuriems yra leidžiama dirbti su sistema.		
Užsakovo patenkinimas: 3	Užsakovo nepatenkinimas: 5	
Priklausomybės: Nėra	Konfliktai: Nėra	
Papildoma medžiaga: Nėra		
Istorija: Užregistruota 2004.11.11		

4.7. Kultūriniai – politiniai reikalavimai

Nėra

4.8. Teisiniai reikalavimai

Nėra

5. PROJEKTO IŠEIGA

5.1. Atviri klausimai

- Esamos duomenų struktūros modifikavimo galimybės – turimas karkasas jau eksploatuojamas.
- Yra tikimybė, kad kuriant grafinę sąsają atsiras naujų veiklos taisyklių tipų.

5.2. Egzistuojantys sprendimai

Nėra. Grafinė aplinka kuriama konkrečiam veiklos taisyklių valdymo karkasui.

5.3. Naujos problemos

5.3.1. Problemos diegimo aplinkai

Vartotojai turės sugaišti laiko diegiant naująją grafinę sąsają, bei mokantis su ja dirbti.

5.3.2. Įtaka jau instaliuotoms sistemoms

Nesitikima.

5.3.3. Neigiamas vartotojų nusiteikimas

Nesitikima.

5.3.4. Kliudantys diegimo aplinkos apribojimai

Nėra.

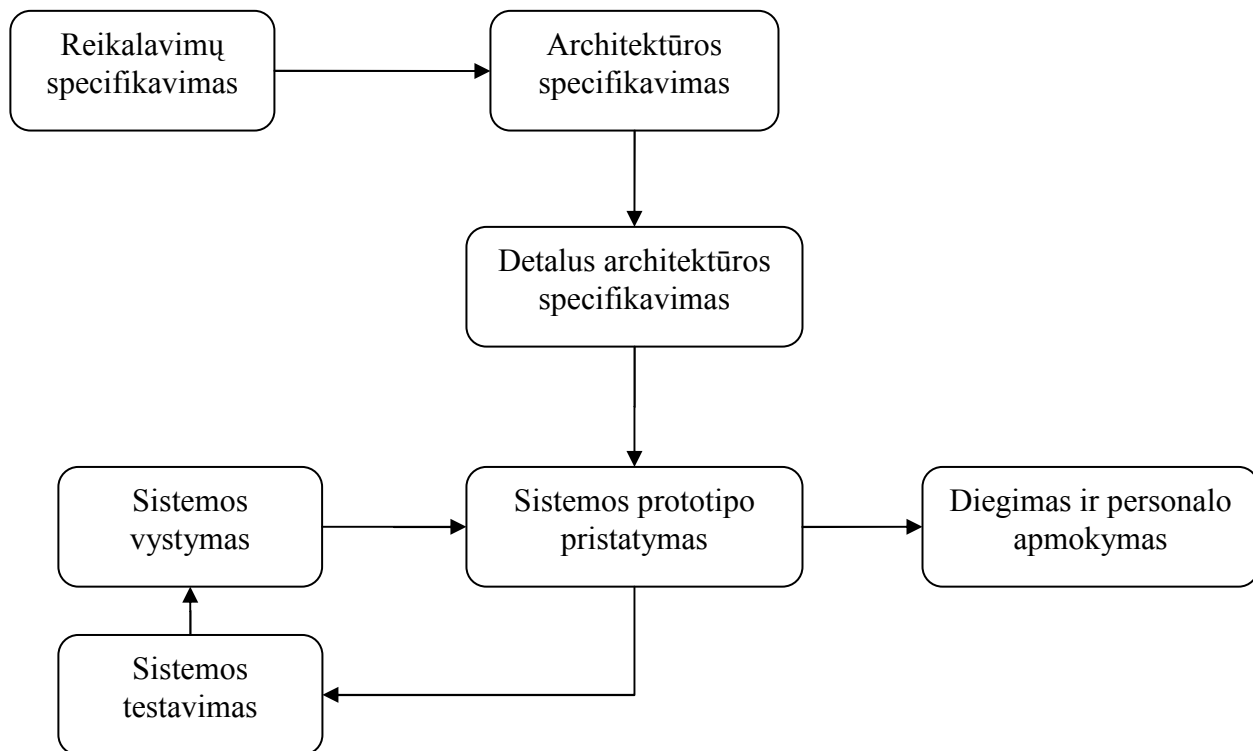
5.3.5. Galimos naujos sistemos sukeltos problemos.

Nesitikima.

5.4. Uždaviniai

5.4.1. Sistemos pateikimo žingsniai

Pagrindiniai sistemos kūrimo ir pateikimo žingsniai pateikiami 21.1 paveikslėlyje.



21.1 pav. Sistemos kūrimo žingsniai

5.5. Pritaikymas

5.5.1. Reikalavimai esamų duomenų perkėlimui

Duomenų perkėlimas nebus reikalingas.

5.5.2. Reikalingas duomenų transformavimas perkeliant į naują sistemą

Nereikalingas.

5.6. Rizikos

5.6.1. Galimos sistemos kūrimo rizikos

Galimos sistemos kūrimo rizikos bei jų įvertinimai pateikiami lentelėje 23.1.

23.1 lentelė

Nr.	Rizika	Rizikos kategorija	Aprašymas	Tikimybė	Įtaka
1	Įvertinimo	Projekto ir produkto	Per trumpas projekto realizavimo laikas	Vidutinė	Leistina
2	Įvertinimo	Projekto ir produkto	Blogai įvertintas PĮ dydis	Vidutinė	Leistina
3	Organizacinė	Projekto	Finansinės problemos priverčia mažinti projekto biudžetą	Žema	Rimta
4	Produkto konkurencija	Verslo	Konkuruojantis produktas pasirodo rinkoje, kai sistema dar nebaigta	Vidutinė	Rimta
5	Reikalavimų	Projekto ir produkto	Reikalavimų pakeitimai iššaukia didelius architektūrinius pasikeitimus	Vidutinė	Rimta
6	Reikalavimų	Projekto ir produkto	Didesnis reikalavimų pakeitimų kiekis nei laukta	Aukšta	Leistina
7	Specifikacijų	Projekto ir produkto	Esminių reikalavimų specifikacijos vėluoja	Vidutinė	Leistina
8	Technologijos pasikeitimas	Verslo	Pagrindinės technologijos, kurių pagrindu veikia sistema, pakeistos naujomis.	L. žema	Katastrofiška
9	Technologinė	Produkto	Programa veikia lėčiau nei laukta	Vidutinė	Leistina
10	Žmonių	Projekto	Patyręs personalas paliks projektą dar jį nebaigus	L. žema	Rimta
11	Žmonių	Projekto	Pasikeis projekto organizacinė struktūra	L. žema	Leistina

5.6.2. Rizikų valdymo planas

Rizikos valdymo planas pateikiamas 23.2 lentelėje.

23.2 lentelė

Nr.	Rizikos aprašymas	Rizikos valdymo planas
1	Per trumpas projekto realizavimo laikas	Kruopščiai sudaryti projekto tvarkaraštį. Projekto metu stebėti, kad kritiniai projekto darbai nevėluotų ir nepadidintų projekto trukmės. Įvykus uždelsimams, pakartotinai suplanuoti likusias veiklas siekiant išvengti projekto vėlavimo.
2	Blogai įvertintas PĮ dydis	Kruopščiai sudaryti reikalavimų ir architektūros specifikacijas. Išsiaiškinti kaip įmanoma daugiau reikalavimų prieš prasidedant sistemos realizavimo fazei.
3	Finansinės problemos priverčia mažinti projekto biudžetą	Pasistengti kuo anksčiau gauti reikiamas projektui lėšas.
4	Konkuruojantis produktas pasirodo rinkoje, kai sistema dar nebaigta	Rinkos stebėjimas. Potencialių sistemos pirkėjų paieška dar nesibaigus realizavimo fazei. Aktyvi reklamos kampanija.
5	Reikalavimų pakeitimai iššaukia didelius architektūrinius pasikeitimus	Išsiaiškinti kaip įmanoma daugiau reikalavimų prieš prasidedant sistemos realizavimo fazei.
6	Didesnis reikalavimų pakeitimų kiekis nei laukta	Išsiaiškinti kaip įmanoma daugiau reikalavimų prieš prasidedant sistemos realizavimo fazei.
7	Esminių reikalavimų specifikacijos vėluoja	Išsiaiškinti kaip įmanoma daugiau reikalavimų prieš prasidedant sistemos realizavimo fazei.
8	Pagrindinės technologijos, kurių pagrindu veikia sistema, pakeistos naujomis.	Pasirinkti „stabilias“ technologijas ir įrankius. Gerai išsiaiškinti jų perspektyvas artimiausiu metu.
9	Programa veikia lėčiau nei laukta	Realizavimo fazėje stengti taupyti kompiuterio resursus. Rinktis „greičiau veikiančius“ sprendimus.
10	Patyręs personalas paliks projektą dar jį nebaigus	Užtikrinti stabilią personalo materialinę bazę. Personalo motyvacija bei skatinimas.
11	Pasikeis projekto organizacinė struktūra	Detalus dokumentavimas. Vengimas situacijų kai darbą daro ir yra už jį atsakingas tik vienas žmogus.

5.7.

5.8. *Kaina*

Nėra.

5.9. *Vartotojo dokumentacija ir apmokymas*

Būsimos vartotojo dokumentacijos sąrašas:

- Sistemos diegimo instrukcija;
- Sistemos vartotojo vadovas;

Dokumentacija bus rengiama bei koreguojama lygiagrečiai viso sistemos kūrimo proceso metu.

Sistemos vartotojų apmokymai bus vykdomi sistemos diegimo bei testavimo metu.

5.10. *Perspektyviniai reikalavimai*

Nėra.

5.11. *Idėjos ir sprendimai*

Néra.