

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Aurimas Laurikaitis

**Veiklos taisyklių metodologija ir jos taikymas kuriant
programinę įrangą**

Informatikos mokslo magistro baigiamasis darbas

Darbo vadovas
doc. dr. R. Butleris

Kaunas, 2004

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

TVIRTINU

Katedros vedėjas
doc. dr. R. Butleris

2004 05

**Veiklos taisyklių metodologija ir jos taikymas kuriant
programinę įrangą**

Informatikos mokslo magistro baigiamasis darbas

Kalbos konsultantė
Lietuvių kalbos katedros lektorė
dr. J. Mikelionienė
2004 05

Vadovas
doc. dr. R. Butleris
2004 05

Recenzentė
doc. dr. R. Misevičienė
2004 05

Atliko
IFM-8/2 gr. stud.
A. Laurikaitis
2004 05

Kaunas, 2004

Turinys

1.	ĮVADAS.....	6
2.	VEIKLOS TAISYKLIŲ KONCEPCIJA.....	8
2.1	Veiklos taisyklių koncepcijos prasmė.....	8
2.2	Veiklos taisyklių klasifikavimas	9
2.3	Veiklos taisyklių struktūrizavimo modeliai	9
2.4	Veiklos taisyklių modelių palyginimas.....	14
3.	VEIKLOS TAISYKLIŲ STRUKTŪRIZUOTAS MODELIS	16
3.1	Veiklos taisyklių struktūrizuotas pavidalas.....	16
3.2	Veiklos taisyklių saugykla	30
4.	VEIKLOS TAISYKLIŲ SISTEMOS ARCHITEKTŪRINIAI SPRENDIMAI.....	38
4.1	Įvadas	38
4.2	Panaudojimo atvejų modelis	38
4.3	Loginis modelis.....	42
4.4	Paskirstymo ir komponentų modelis.....	65
4.5	Duomenų modelis	66
4.6	Sistemos realizacija.....	67
5.	VEIKLOS TAISYKLIŲ IŠSAUGOJIMO EKSPERIMENTAS	69
5.1	Įvadas	69
5.2	Veiklos taisyklių įvedimas į sistemą.....	69
5.3	Veiklos taisyklių išsaugojimas saugykloje.....	75
5.4	Veiklos taisyklių atkūrimas.....	76
5.5	Eksperimento rezultatai.....	83
6.	IŠVADOS.....	85
7.	LITERATŪRA.....	87
8.	TERMINŲ IR SANTRUMPŲ ŽODYNAS.....	89
9.	PRIEDAI.....	91
	1 priedas Veiklos taisyklių modelis pagal GUIDE.....	91
	2 priedas Straipsnis „Veiklos taisyklių šablonai struktūriniams teiginiams specifikuoti“	92
	3 priedas Veiklos taisyklių sistemos pagrindinis langas.....	99
	4 priedas Veiklos taisyklės išsaugojimas sistemoje.....	100
	5 priedas Eksperimento metu į sistemą įvestos veiklos taisyklės.....	105

SUMMARY

Business Rules Approach and its Appliance for Software Development

This work analyses Business Rule methodology and its benefits against traditional software development concepts. Business Rule methodology is observed, several methods of Business Rule classification are presented: Ross, MDC, GUIDE. The work assesses strengths and weaknesses of each model. GUIDE model is chosen for further development of Business Rules methodology. Suggested improvements of this model are presented.

This work proposes improvements of GUIDE model, which includes Business Rule templates and repository schema. The benefits of using templates are analyzed and templates for structural and action assertion Business Rules are presented. The repository schema for structural and action assertion rules, which are specified using templates, is provided.

The project of system, which uses Business Rule concepts and its suggested improvements, is provided. This system records and stores Business Rules in Business Rules repository. The system experiment, where proposed improvements are tested, is presented.

SANTRAUKA

Šiame darbe nagrinėjama veiklos taisyklių koncepcija, jos pranašumai prieš tradicinės programinės įrangos metodologijas. Pateikiama veiklos taisyklių koncepcijos literatūros apžvalga, pristatomi įvairūs veiklos taisyklių klasifikavimo ir struktūrizavimo modeliai: Roso, MDC, GUIDE. Šie modeliai yra palyginami, nustatomi jų privalumai ir trūkumai. Tolimesniam veiklos taisyklių koncepcijos vystymui pasirenkamas GUIDE modelis, apibrėžiami siūlomi šio modelio papildymai.

Darbe siūlomi GUIDE modelio patobulinimai, apimantys veiklos taisyklių šablonus ir veiklos taisyklių saugyklą. Nustatomas šablonų panaudojimo tikslingumas bei pasiūlomi šablonai struktūrinių ir veiksmo teiginių veiklos taisyklėms struktūrizuoti užrašyti. Pateikiami reikalavimai veiklos taisyklių saugyklai bei siūlomas specifikuotas veiklos taisyklių saugyklos modelis.

Pristatoma pagal veiklos taisyklių koncepciją ir jos siūlomus patobulinimus suprojektuota sistema. Ši sistema realizuoja veiklos taisyklių įvedimą bei išsaugojimą veiklos taisyklių saugykloje. Pristatomas eksperimentas, kuriame pateikiamas tyrimas, kaip siūlomi GUIDE metodo patobulinimai veikia realizuotoje sistemoje.

1. ĮVADAS

Sistemų analitikų naudojami metodai jau seniai leidžia apibrėžti organizacijos veiklą įvairiais pjūviais [5]. Ši metodika leidžia nustatyti verslo naudojamus duomenų struktūras bei atliekamas funkcijas, tačiau ilgą laiką nebuvo atsižvelgiama į labai svarbų aspektą – taisyklių rinkinį, apibrėžiantį kaip organizacija veikia – t. y. taisykles, kurios apriboja, leidžia arba reguliuoja organizacijos veiksmus. Plėtojantis informacinėms technologijoms ir užsakovų poreikiams tokia programų kūrimo koncepcija ėmė nebetenkinti nei užsakovų, nei programinės įrangos kūrėjų, todėl pradėta ieškoti būdų kaip išreikšti nestruktūrizuotas taisykles. **Veiklos taisyklių projektas** užpildė šią programų kūrimo koncepcijos spragą.

GUIDE veiklos taisyklių projektas įgyvendintas 1997 metais. Šio projekto autoriai apibrėžė ir struktūrizavo taisykles, kurios išreiškia verslo struktūrą ir daro įtaką jo veiklos procesui. GUIDE projektas įgyvendino šiuos tikslus [3]:

- Nustatė ir aprašė veiklos taisyklių ir susijusias sąvokas, leidžiančias nustatyti, kas yra veiklos taisyklė.
- Apibrėžė koncepcinį veiklos taisyklių modelį, kuris leidžia išreikšti (informacinių technologijų profesionalams reikšmingomis sąvokomis), kas yra veiklos taisyklės ir kaip jos pritaikomos realiose informacinėse sistemose.

GUIDE modelis apibūdina natūralia kalba išreikštas veiklos taisykles, tačiau nestruktūrizuota kalba neįmanoma tiksliai ir nedviprasmiškai užrašyti taisyklių. Modelio autoriai pabrėžia formalios užrašymo formos sukūrimo būtinybę, tačiau plačiau šio aspekto nenagrinėja. Norint GUIDE modelį pritaikyti praktinėms reikmėms, reikia nustatyti šablonus [11], apimančius įvairius veiklos taisyklių tipus.

Turint šablonų pagrindus sudarytą veiklos taisyklių specifikaciją, veiklos taisykles galima kaupiti duomenų saugykloje, kur jomis galima lengvai manipuluoti [8, 15]. Kiekviena veiklos taisyklė suformuojama iš saugykloje išsaugotų jos sudėtinių dalių ir tipo raktinių žodžių. Veiklos taisyklių saugyklos GUIDE modelis taip pat neapibrėžia.

Darbo tikslas:

- Išanalizuoti veiklos taisyklių koncepciją, jos aktualumą bei klasifikaciją.
- Apžvelgti įvairias literatūroje pateiktas veiklos taisyklių metodologijas ir palyginti jas su GUIDE koncepcija.
- Nustatyti įvairių veiklos taisyklių, klasifikuotų pagal GUIDE, tipų šablonus.
- Suprojektuoti veiklos taisyklių saugyklą, leidžiančią išsaugoti veiklos taisykles, įvestas pagal nustatytus šablonus.

- Suprojektuoti ir realizuoti programinę Įrangą, kuri Įgyvendintų GUIDE veiklos taisyklių koncepciją ir jos pateiktus papildymus.
- Atlikti veiklos taisyklių išsaugojimo eksperimentą Įrodantį, kad GUIDE veiklos taisyklių koncepcija bei jos papildymai veikia praktiškai.

Informatikos mokslo magistro baigiamasis darbas pristatomas dalimis: veiklos taisyklių koncepcijos, struktūrizuoto modelio, sistemos architektūrinių sprendimų ir išsaugojimo eksperimento.

Veiklos taisyklių koncepcijos dalyje apžvelgiama veiklos taisyklių metodologija, klasifikacija, struktūrizavimo modeliai, jų privalumai bei trūkumai. Analitinėje dalyje suformuluojamas uždavinys, pateikiama galimų sprendimų apžvalga, priimto sprendimo pagrindimas.

Veiklos taisyklių struktūrizuoto modelio dalyje pateikiamas analitinėje dalyje suformuluoto uždavinio koncepcinis sprendimas.

Veiklos taisyklių sistemos architektūrinių sprendimų dalyje pateikiama sukurtos programinės Įrangos techninė – projektinė dokumentacija. Šioje dalyje pateikiamas pasirinkto sprendimo realizacijos kelias.

Veiklos taisyklių išsaugojimo eksperimento dalyje yra pateikiamas sukurtos ir Įdiegtos programinės Įrangos eksperimentinis tyrimas. Eksperimentinėje dalyje Įrodoma, kad struktūrizuoto modelio dalyje pasiūlyti metodai veikia praktiškai.

2. VEIKLOS TAISYKLIŲ KONCEPCIJA

2.1 Veiklos taisyklių koncepcijos prasmė

Veiklos taisyklė yra teiginys, kuris apibrėžia arba apriboja tam tikrą organizacijos veiklos aspektą. Veiklos taisyklės apibrėžia ir reguliuoja organizacijos struktūrą, veiksmus bei strategiją [5]. Jos užfiksuotos organizacijos veiklos krypties ir procedūrų vadovuose, užsakovų sutartyse, tiekėjų susitarimuose ir vadybos strategijose. Kiekvieno darbuotojo žinios – taip pat VT. VT egzistuoja įvairiose taikomose programose – nuo paprasčiausio internetinio puslapio iki gausių veiklos taisyklių, susistemintų į teisinės kalbos B2B (*business to business* – verslas verslui) prekybos partnerių sutartis. Finansų ir draudimo kompanijos privalo reguliuoti staigius vidaus ir išorės veiklos politikos krypčių pasikeitimus. Telekomunikacijų kompanijos svaiginančiu greičiu pristato naujas paslaugų ir kainodaros schemas. E. verslo, savitarnos ir asmens identifikavimas internete reikalauja realaus laiko užsakymų sekimo, pardavimų istorijos ir pirkėjų pasirinkimų. Kiekviena iš šių verslo sričių labai priklauso nuo informacinių technologijų skyrių sugebėjimo taikomose programose įdiegti verslo politikos, nuostatų ir strategijų taisykles.

Verslo naudojamos veiklos taisyklės pasižymi dinaminėmis savybėmis, todėl tradicinės taikomosios programos nesugeba pakankamai efektyviai palaikyti specifinių poreikių. Verslo logika, kuri yra įdiegta į taikomųjų programų objektus, duomenų bazės struktūras ar saugomas procedūras, yra sunkiai palaikoma ir jos pasikeitimai brangiai kainuoja. Įvykus pakitimams verslo srityje, kad jie būtų atspindėti, programų kodas taip pat turi būti keičiamas. Šių laikų greitai besikeičiančioje verslo ir e. verslo aplinkoje, konkurencija privertė organizacijas prisitaikyti prie besikeičiančios rinkos per neįtikėtina trumpą laiką. Gana dažnai turima programinė įranga nepajėgi pakankamai greitai, kad organizacija išliktų konkurencinga, atspindėti šių pasikeitimų. Šiuo atveju vienintelis sprendimas – veiklos taisyklių koncepcijos panaudojimas taikomose programose.

Veiklos taisyklių atskyrimas nuo taikomosios programos kodo ir jų įdiegimas, panaudojant atskirą veiklos taisyklių sluoksnį, padaro taikomąsias programas lengviau pritaikomas ir palaikomas. Kadangi VT yra išskiriamos iš programos kodo, jos gali būti keičiamos nepriklausomai, neperkompilijuojant programos. Jei veiklos taisyklės yra atvaizduojamos natūralia verslo kalba, tai verslo klientai ir analitikai gali lengvai įrašyti naujas ir modifikuoti senas taisykles.

Verslas naudoja įvairias veiklos taisykles: vienos apibrėžia organizacijos struktūrą, kitos susieja organizacijos struktūrinius vienetus į visumą, dar kitos apriboja organizacijos veiksmus ir strategiją. Atsižvelgiant į VT apibrėžiamą ar apribojamą aspektą veiklos taisyklės gali būti skirstomos į rūšis.

2.2 Veiklos taisyklių klasifikavimas

Veiklos taisyklių teiginiai pagal apibrėžiamą ar apribojamą aspektą yra skirstomi [1, 3, 13] į keturias rūšis:

- *Veiklos sąvokų apib. režimai*

Elementariausias veiklos taisyklės elementas – apibrėžimas, kuris naudojamas nustatyti VT prasmę. Sąvokos apibrėžimas yra veiklos taisyklė aprašanti, kaip žmonės supranta ir išreiškia dalykus. Sąvokos išreiškiamos specialiaisiais terminais arba esybėmis esybių / ryšių diagramoje.

- *Faktai susiejantys sąvokas tarpusavyje*

Organizacijos prigimtis ar jos veikianti struktūra gali būti apibrėžta faktais, kurie susieja sąvokas tarpusavyje. Apibrėžimas, kad **klientas gali pateikti užsakymą**, yra veiklos taisyklė. Faktai išreiškiami natūralios kalbos sakiniais arba grafinio modelio sąryšiais, atributais ir apibendrinimais.

- *Apribojimai*

Nustato dinامينius organizacijos aspektus. Kiekvienos Įmonės elgesys yra apribotas tam tikru atžvilgiu, o tai susiję su apribojimais, kai duomenų saugyklos Įrašai gali ar negali būti atnaujinti. Neleisti atnaujinti Įrašo daugeliu atveju yra tas pats, kas neleisti Įvykti veiksmui.

- *Išvestys*

Nustato kaip tam tikros žinios gali būti pertvarkytos į kitas žinias. Išvestys nustato matematinės arba euristinės veiklos taisykles, kurių nereikia tiksliai išsaugoti, nes esant būtinybei jos gali būti išvedamos iš realiai egzistuojančių ar kitų išvestų VT [14]. Pvz.

Užsakymo suma suskaičiuojama produkto kiekį padauginus produkto kainos.

Šiame skyriuje pateikta VT klasifikacija yra apibendrinta, konkretūs VT struktūrizavimo modeliai detaliau klasifikuoja VT.

2.3 Veiklos taisyklių struktūrizavimo modeliai

Nuo pat VT koncepcijos atsiradimo pradžios buvo siūlomi įvairūs jų klasifikavimo ir struktūrizavimo modeliai [7]. Galima pastebėti, jog dauguma tiek ir informacinių sistemų projektavimo Įrankius kuriančių bendrovių (*USoft Corporation*, *Vision Software*, *Versata Inc.* ir kt.), tiek ir atskirų tyrinėtojų (*Barbara von Halle*, *James Odell*, *Margaret Thorpe* ir kt.) pasiūlytų metodų pasižymi nepastovumu bei susiaurinta pritaikymo sritimi. Nepaisant to, šiuo metu galima išskirti tris VT klasifikavimo modelius, pretenduojančius tapti bendru standartu. Tai grupės mokslininkų pasiūlytas GUIDE projektas [3], *Meta Data Coalition* kartu su *Microsoft* sukurtas modelis (toliau MDC modelis) [2, 4] bei Ronaldo Roso metodas [13]. Galima daryti prielaidą, kad šių trijų modelių

analizė pakankamai gerai atspindi dabartinę padėtį VT struktūrizavimo koncepcijos tyrimo srityje. Žemiau pateikiami trumpi kiekvieno iš paminėtų modelių aprašymai.

2.3.1 GUIDE projektas

GUIDE veiklos taisyklių projektas pradėtas įgyvendinti 1993 m. lapkričio mėnesį [3]. Šio projekto tikslas – apibrėžti ir klasifikuoti taisykles, išreiškiančias verslo struktūrą ir darančias įtaką jo veiklos procesui. GUIDE teorinė analizė yra tvirtas pagrindas realioms veiklos taisyklių koncepcijos pagrindu veikiančioms sistemoms [7]. GUIDE projekte nagrinėjamas ne tik veiklos taisyklių klasifikavimas, bet ir jų struktūra bei išskyrimo principai. VT suprantama kaip teiginys, apibrėžiantis arba apribojantis tam tikrus veiklos aspektus.

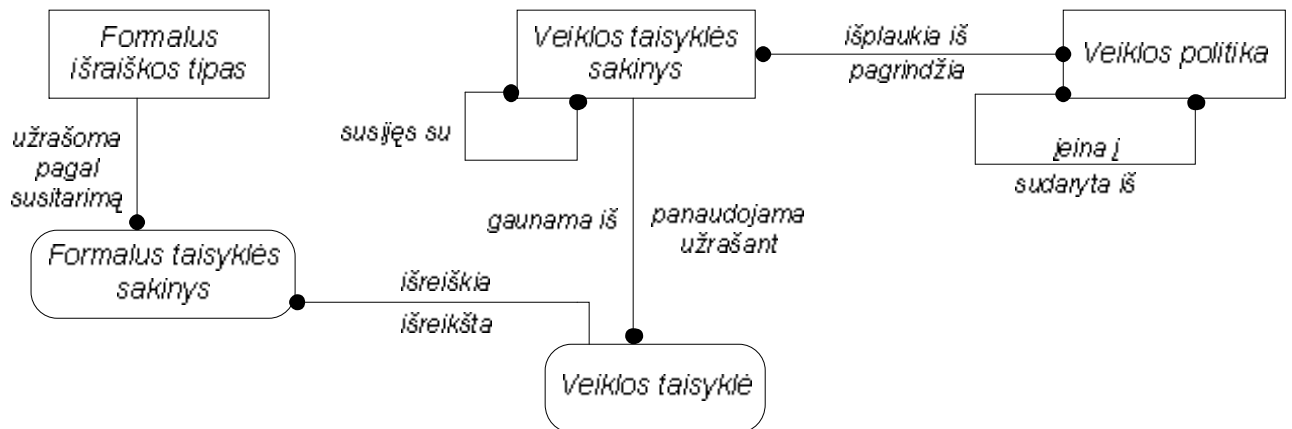
GUIDE nurodo veiklos taisyklių kilmę (2.2.1 pav., pilnas veiklos taisyklių modelis pagal GUIDE yra pateikiamas 1 priede). Bendriausias GUIDE koncepcijos objektas yra *veiklos politika* – bendrą organizacijos veiklos aspektą apibūdinanti charakteristika. Pvz. **Įmonė nuomoja klientams automobilius nepažeisdama Įstatymų bei tinkamomis eksploatuoti sąlygomis**. Kiekviena *veiklos politika* gali susidėti iš detalesnių *veiklos politikų*, taip pat *veiklos politika* gali būti vienos ar daugiau bendresnių *veiklos politikų* sudėtinė dalis.

Veiklos politika pagrindžia vieną ar daugiau *veiklos taisyklių sakinių*, kurie taip pat gali išplaukti iš vienos arba daugiau *veiklos politikų*. *Veiklos taisyklės sakiny* – deklaratyvus verslo struktūros arba apribojimo teiginys. Pvz. **Esant bent vienai iš žemiau paminėtų sąlygų, turėtų būti atliekama automobilio techninė apžiūra arba remontas: 1) nuo paskutinio techninio aptarnavimo pravažiuota 5000 kilometrų, 2) blogai veikia stabdžiai, 3) triukšmingas dujų išmetimas arba skleidžiami dūmai, 4) pažeistas kėbulas, apšvietimas arba stiklas, 5) pastebimas žymus skysčių nutekėjimas**. Kiekvienas *veiklos taisyklės sakiny* gali būti susijęs su vienu arba daugiau *veiklos taisyklių sakinių*.

Veiklos taisyklės sakiny, savo ruožtu, pagrindžia vieną ar daugiau nedalomų *veiklos taisyklių*, kurios taip pat gali išplaukti iš vieno arba daugiau *veiklos taisyklės sakinių*. Kaip ir *veiklos taisyklės sakiny*, *veiklos taisyklė* apibrėžia arba apriboja tam tikrus veiklos aspektus, tačiau priešingai nei *veiklos taisyklės sakiny* VT nebegali būti susmulkinta į dar detalesnes *veiklos taisykles*. Jeigu *veiklos taisyklė* būtų skaidoma į dalis, tuomet būtų prarandama svarbi veiklos informacija. Pvz. **Automobiliui, kuris pravažiavo 5000 kilometrų nuo paskutinės techninės aptarnavimo, turėtų būti atlikta techninė apžiūra**.

Svarbu pažymėti, kad organizacijos naudoja *veiklos taisykles*, nepaisant to, kokia forma jos yra išreikštos. Organizacijos jau seniai veikė pagal nustatytas VT, nors niekas dar nė nesvajėjo jų struktūrizuoti. *Veiklos taisyklės* reguliuoja organizacijos veiklą nepriklausomai nuo analitiko pastangų struktūrizuoti ir aprašyti jas.

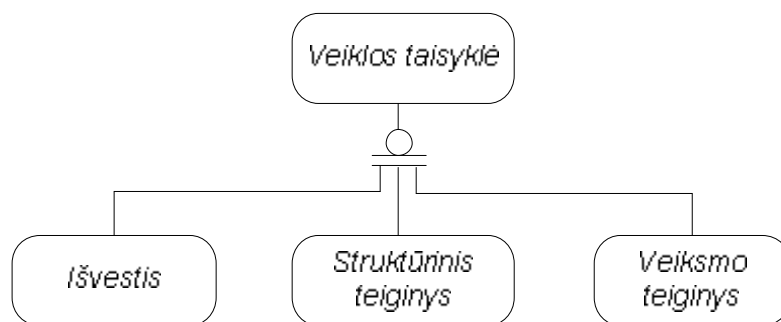
Veiklos taisyklė yra išreiškiama vienu ar daugiau *formalių taisyklės sakinių*, tačiau kiekvienas *formalus taisyklės sakinys* išreiškia tik vieną nedalomą veiklos taisyklę. *Formalus taisyklės sakinys* aprašo veiklos taisyklę apibrėžta formalia gramatika. *Formalus taisyklės sakinys* privalo būti užrašytas pagal *formalaus išraiškos tipo* užrašymo taisyklės. *Formalaus išraiškos tipų* pvz.: **struktūrizuota natūrali kalba, IDEF1X, Roso notacija** ir t. t. *Formalaus taisyklės sakinio*, užrašyto struktūrizuota natūralia kalba, pvz. **Jeigu Auto.km-dabartinis-periodas > 5000, tai iškviešti Techninė-apžiūra (Auto.id)**.



2.2.2 pav. Veiklos taisyklių kilmė pagal GUIDE

GUIDE modelyje yra išskiriama keletas VT tipų [8] (2.2.2 pav.):

- *Struktūrinis teiginys* – sąvokos apibrėžimas arba fakto formuluotė, kuri išreiškia tam tikrą organizacijos struktūros aspektą.
- *Veiksmo teiginys* – apribojimą arba sąlygą nusakantis sakinys, kuris apriboja arba valdo organizacijos veiksmus.
- *Išvestis* – sakinys, apibūdinantis žinias, kurios yra išvedamos iš kitų veiklos žinių.



2.2.2 pav. Veiklos taisyklių tipai pagal GUIDE

Kiekvienas iš trijų taisyklių tipų yra detalizuojamas į aibę potipių. Svarbiausieji pateikti 1 lentelėje.

2.2.1 lentelė. Veiklos taisyklės tipų detalizavimas

<i>Struktūrinis teiginys</i>	<i>Veiksmo teiginys</i>	<i>Išvestis</i>
<ul style="list-style-type: none"> • Sąvoka <i>tipai:</i> Bendroji Veiklos <ul style="list-style-type: none"> • Faktas <i>kilmės:</i> Išvestinis Pirminis <i>tipai:</i> Atributas Apibendrinimas Dalyvis Asociacija Agregatas Vaidmuo	<i>klasės:</i> Sąlyga Apribojimas Leidimas <i>tipai:</i> Įgalintuvas Laikmatis Vykdymo <i>pobūdis:</i> Veiksmą valdantis Veiksmą įtakojantis	<ul style="list-style-type: none"> • Matematinis skaičiavimas • Loginė išvada

2.3.2 MDC modelis

Meta Data Coalition (MDC) kartu su *Microsoft* sudarytas VT modelis yra naujausias iš visų trijų nagrinėjamų modelių [2, 6]. Vis dėlto, VT klasifikavimo požiūriu šis modelis nusileidžia kitiems projektams. GUIDE projekto autoriai parengė esminių skirtumų analizę, įrodančių GUIDE modelio pranašumą [4]. Tačiau MDC modelis pasižymi ypač detaliu taisyklių saugojimo modelio aprašu. Trumpas MDC modelio aprašas pateikiamas žemiau.

Veiklos taisyklė MDC modelyje apibrėžia, kaip pereiti nuo vienos būsenos prie kitos arba kaip uždrausti tokį perėjimą. Taigi tai daugiau deklaratyvus teiginys negu procedūrinis aprašas.

Veiklos teiginys (*business rambling*) palaiko VT izoliavimo procesą. Tai yra pradinis VT išvedimo taškas. Veiklos teiginys gali susidėti iš daugiau nei vienos VT. Taip pat jis gali būti nepastovus, prieštarauti kitiems teiginiams ar netgi būti klaidingas. Resursas – tai veiklos teiginio šaltinis (apibrėžtas organizaciniame modelyje [2]).

Struktūrizuotas VT atvaizdas gali priklausyti šiems pagrindiniams tipams:

- *Sąvoka* – Įtraukia sąvokos apibrėžimą į veiklos terminų žodyną. Tokia VT naudojama veiklos žodžiams apibrėžti.
- *Faktas* – dokumentuoja ryšius tarp elementų. Pavyzdžiui, ryšiai tarp esybių, ryšiai tarp esybių ir atributų („priklauso“ ir pan.).
- *Veiksmas* – VT, valdanti veiksmų iškvietimą. Ji nurodo sąlygas, kurias patenkinus, veiksmas turi būti vykdomas. Tai gali būti išankstinės sąlygos (*pre-conditions*), vėlesnės sąlygos (*post-conditions*) ir paleidimo (*trigger*) tipo sąlygos.

- *Draudimas* – aprašo VT išvedimą ar išvestį iš kitų taisyklių arba matematinius skaičiavimus. Šios VT dar vadinamos *išvestomis taisyklėmis*, nes jos saugo informaciją, kuri yra dinamiškai išvedama.

Veiklos taisyklės gali būti grupuojamos i *VT aibes*. Jos gali atspindėti nuoseklumą, specifinę veiklos sritį, realizavimo subtilybes, organizacines arba projekto struktūras.

Pažymėtina, jog MDC modelis buvo kuriamas kaip realaus programinio projekto *Open Information Model (OIM)* dalis. Tai lėmė ypatingą dėmesį techninėms realizacijos detalėms, kurios kituose dviejuose modeliuose nėra griežtai apibrėžtos. Tokiu būdu, šis modelis iš esmės pasiūlo VT saugyklos prototipą. Yra numatytas taisyklių saugojimas parametrais pagrįstu principu. Šio principo esmė yra ta, jos kiekvienas taisyklės elementas yra saugojamas atskirose lentelėse. Tokioje struktūroje yra numatyti ne tik kiekvieno VT modelio elemento atributai, bet ir jų ryšys su pagrindinio duomenų modelio elementais. Bazinis duomenų modelis šiuo atveju yra tiesiog būtinas, nes taisyklės yra išreiškiamos būtent šio modelio terminais.

2.3.3 Roso metodas

Metodas yra pasiūlytas Ronaldo Roso [13]. Roso veiklos taisyklių modelis, kaip ir GUIDE projektas, buvo kuriamas kaip bendro pobūdžio VT studija.

Pagal Roso metodą taisyklės gali būti *atominės* ir *išvestinės*. Yra išskiriami 32 atominių (nedalomų) taisyklių tipai, kurie grupuojami į 7 šeimas. Išvestinė taisyklė – tai taisyklė, kuri išreiškia kitų taisyklių aprašais. Išvestinės taisyklės nėra atominės ir gali būti sudarytos iš keleto atominių taisyklių arba kitų išvestinių taisyklių. Yra išskiriami 58 išvestinių taisyklių tipai, kurie pagal panaudojimo sritį yra grupuojami į 12 šeimų. Visos veiklos taisyklių šeimos yra pateiktos 2.2.2 lentelėje. Būtina pažymėti, kad ypač daug dėmesio R. Rosas skiria taisyklių tipų aprašymui, todėl šis modelis klasifikavimo požiūriu yra gana tikslus, tačiau tuo pačiu gana griežtas.

2.2.2 lentelė. Veiklos taisyklių tipų šeimos pagal Roso metodą

<i>Atominių taisyklių šeimos</i>	<i>Išvestinių taisyklių šeimos</i>	
I. Egzempliorių patvirtintojai	A. Egzempliorių testavimo	G. Sąlyginio laiko įvertinimo
II. Tipų patvirtintojai	B. Pozicijos patikrinimo	H. Atnaujinimų įvertinimo
III. Pozicijos patvirtintojai	C. Atributų modifikavimo	I. Veiklos koordinavimo
IV. Funkciniai patvirtintojai	D. Sekos valdymo	J. Egzempliorių įgalinimo
V. Matematiniai įvertintojai	E. Sekos specifikuojimo	K. Egzempliorių kopijavimo
VI. Projekcijos valdikliai	F. Kompozicijos struktūrų testavimo	L. Egzempliorių iššaukimo

Be detalios taisyklių klasifikavimo sistemos, R. Roso metode yra pasiūlyta ir unikali VT modeliavimo, užrašant jas grafine forma, metodika.

Taisyklės teiginio pateikimas formalios taisyklės pavidalu susideda iš 5 pagrindinių etapų. Analitikas turi atsakyti į šiuos klausimus:

1. Ar taisyklė yra *apribojimas*, ar *sąlyga*? (kitais tariant, ar taisyklė reikalauja vykdymo, ar yra išreikšta tik kaip patikrinimas)
2. Kas yra taisyklės *bazė*? (t. y. kuriam tipui duomenų modelyje reikėtų „priskirti“ nagrinėjamą „taisyklę“)
3. Kas yra taisyklės *korespondentas* (-ai)? (t. y. kokie tipai duomenų modelyje būtini norint patikrinti taisyklę)
4. Koks taisyklės tipas? (t. y. kokius testus pritaiko taisyklė)
5. Koks ryšys taro *bazės* ir *korespondento*? (t. y. kaip tipai, reikalingi taisyklei patvirtinti, susiję tarpusavyje)

Radus atsakymus į šiuos klausimus, taisyklė yra užrašoma grafiniu pavidalu naudojant R. Roso pasiūlytą grafinę notaciją. Akivaizdu, kad norint formaliai užrašyti taisykles, reikia naudotis dalykinės srities duomenų modeliu. Koks būtent modelis turėtų būti naudojamas, R. Rosas nenurodo.

2.4 Veiklos taisyklių modelių palyginimas

Pasirenkant veiklos taisyklių struktūrizavimo modelį, reikia atkreipti dėmesį į šiuos aspektus:

1. Modelio apibrėžtumas. Kiekvienas VT tipas, klasė ar kitoks poaibis turi būti griežtai apibrėžtas. Analitikui neturi kilti abejonių dėl užregistruotos taisyklės priklausymo vienai ar kitai grupei.
2. Saugojimo modelis. Galutinis VT struktūrizavimo proceso tikslas yra VT saugykla. Žinant, kad kiekviena taisyklių klasifikacija pasižymi unikaliomis savybėmis ir specifine struktūrizavimo metodika, tikslus atitinkamo saugyklos modelio apibrėžimas yra labai svarbus.
3. Modelio aiškumas ir paprastumas. VT koncepcija pagrįstų taikomųjų programų galutiniai vartotojai yra įvairių sričių specialistai. Jie nėra IT ekspertai, neturi pakankamai laiko ir žinių gilintis į sudėtingus klasifikavimo modelius, todėl jiems yra reikalingas paprastas ir efektyvus VT klasifikacijos modelis.

Atlikus palyginamąją modelių analizę prieita tokių išvadų:

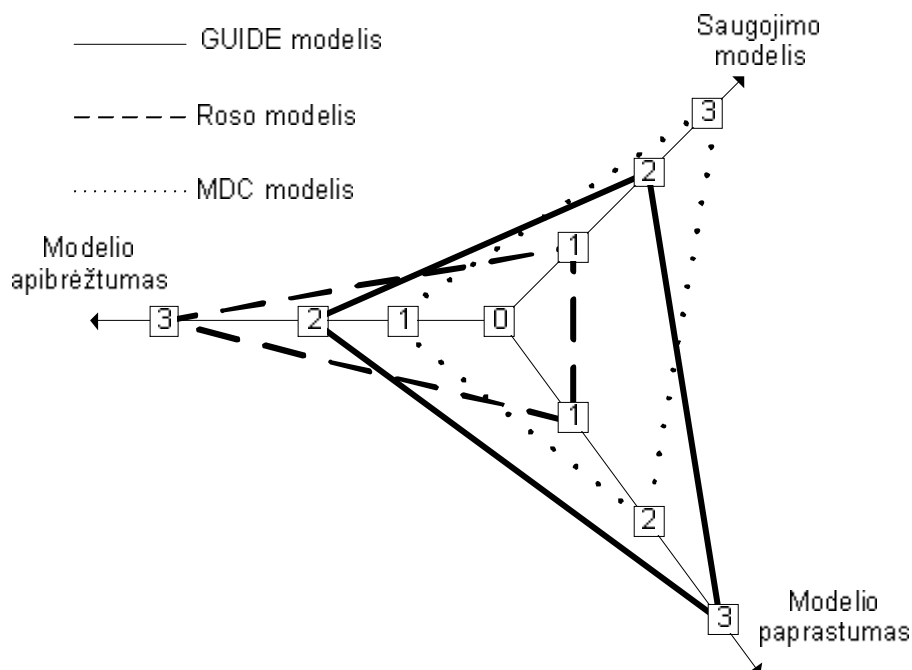
1. Detaliausias VT klasifikavimo aspektu yra Roso metodas.
2. Labiausiai išvystytas VT saugojimo problemos sprendimas pasiūlytas MDC modelyje.
3. Lengviausiai įsisavinama yra GUIDE modelio klasifikacija.

Roso pasiūlytoji klasifikacija yra ypač detali ir šiuo atžvilgiu lenkia savo konkurentus, ypač MDC modelį. Tačiau Roso metodas yra sudėtingas ir sunkiai suprantamas, ypač IT išsilavinimo

neturintiems specialistams. MDC modelis, nors ir išsiskiria gana detaliu VT saugojimo metodikos aprašymu, tačiau yra palyginti silpnas klasifikavimo atžvilgiu.

GUIDE modelis turi labai stipriai išvystytą teorinę bazę. Modelis yra ypač lengvai įsisavinamas ir šiuo atžvilgiu gerokai lenkia savo konkurentus. Jo klasifikacija yra gana detali. Patobulinus GUIDE modelį ir kitais aspektais jis nenusileistų Roso ir MDC metodams.

Grafinė modelių palyginimo interpretacija pateikta 2.4.1 pav. Koordinačių ašys vaizduoja aspektus, pagal kuriuos vertinami modeliai. Kiekvienam modeliui priskiriama konkreti vertinimo reikšmė taškais nuo 1 (pagal nagrinėjamą kriterijų yra blogiausias) iki 3 (geriausias). Modelio apibrėžtumo aspektu geriausias Roso modelis (3 t.), antras GUIDE modelis (2 t.), blogiausias MDC modelis (1 t.). Saugojimo modelio aspektu labiausiai išvystytas MDC modelis (3 t.), antras GUIDE modelis (2 t.), mažiausiai išvystytas Roso modelis (1 t.). Aiškumo ir paprastumo aspektu pranašiausias yra GUIDE modelis (3 t.), antras MDC modelis (2 t.), sunkiausiai suprantamas Roso modelis (1 t.). Koordinačių ašyje atidėjus kiekvieno modelio vertinimo reikšmę pagal kiekvieną kriterijų ir sujungus linijomis taškus, matyti, kad išsiskiria GUIDE modelis. Jis pranašesnis už kitus modelius pagal pasirinktus kriterijus.



2.4.1 pav. Veiklos taisyklių modelių palyginimas

Apibendrinant atliktą palyginamąją analizę galima teigti, kad nė vienas iš nagrinėjamų modelių netenkina visų trijų užsibrėžtų reikalavimų, tačiau daugiausiai privalumų turi GUIDE modelis. Todėl IS projektavimo proceso, pagrįsto VT koncepcija, palaikymui bus panaudotas GUIDE modelis. Svarbiausi žingsniai pritaikant GUIDE modelį praktinėms reikmėms:

- Taisyklių teiginių struktūrizuoto pavidalo nustatymas.
- Veiklos taisyklių saugyklos sudarymas.

3. VEIKLOS TAISYKLIŲ STRUKTŪRIZUOTAS MODELIS

3.1 Veiklos taisyklių struktūrizuotas pavidalas

3.1.1 Šablonų panaudojimas ir jų privalumai

Struktūrizuotoms veiklos taisyklėms užrašyti tikslinga panaudoti šablonus. Šablonas – pakartotinai panaudojamas modelis [8, 10], nustatantis gairių, kaip sukurti tam tikrą produktą ar atlikti veiklą, rinkinį. VT fiksavimui naudojamų šablonų privalumai:

- Šablonai yra pirmasis žingsnis struktūrizuojant VT.

VT idėja yra gerai išnagrinėta ir pristatyta, tačiau taisyklių struktūrizavimas ir fiksavimas vis dar tyrimams ir tobulinimui atvira sritis. Atslūgus pirmajam susižavėjimui, sistemų analitikai ir užsakovai ima suprasti, kad reikalingas nuoseklus priėjimas užrašyti išgautas taisykles. Šablonų panaudojimas suteikia VT fiksavimo procesui darnos ir parodo kelią link technologinių sprendimų.

- Šablonai sumažina užfiksuotų taisyklių informacijos atvaizdavimo prieštarumą.

Geriausiai pastebima nauda, susijusi su šablonais, yra prieštarungumo sumažinimas. Visai komandai naudojantis tuo pačiu formatu VT fiksuoti, jos nariai gali pasidalinti taisyklėmis su bendradarbiais, nesirūpindami dėl taisyklių fiksavimo struktūros.

- Šablonai pašalina dviprasmiškumus ir įgalina atlikti VT patvirtinimą (*validation*).

Vienas iš pagrindinių VT metodo privalumų yra patvirtinimo paprastumas. Priešingai duomenų ar proceso modelio diagramoms, VT sąrašas gali būti lengvai naudojamas be papildomų paaiškinimų ar dokumentacijos. Šablonai dar labiau išplečia galimybes, suteikdami neprieštarinę naudojamų taisyklių aprašo formą. Be to šablonai, orientuoti į atskirą taisyklę, leidžia patvirtinti kiekvieną iš jų atskirai.

- Sistemos kūrėjams šablonas tarnauja kaip pirminė sistemos specifikavimo struktūra.

Šablonų neprieštarungumo savybę sistemų kūrėjai gali panaudoti kaip pirminę sistemos specifikavimo struktūrą. Kadangi kiekviena veiklos taisyklėse naudojama sąvoka yra esybė, atributas ar atributo reikšmė, šablono forma pateiktos taisyklės leidžia žymiai lengviau surasti sprendimus kuriant sistemą.

Literatūroje yra pasiūlyta įvairių taisyklių formą struktūrizuojančių šablonų [10], tačiau siūlomi šablonai nėra pakankamai gerai struktūrizuoti, skirtingiems veiklos taisyklių tipams pateikiami šablonai blogai susieti tarpusavyje. Tokių šablonų pagrindu įvestas taisyklės labai sunku panaudoti išsaugant jas veiklos taisyklių saugykloje. Be to literatūroje nagrinėjami šablonai nepalaiko GUIDE modelio. Todėl nuspręsta kurti naujus šablonus, kurie:

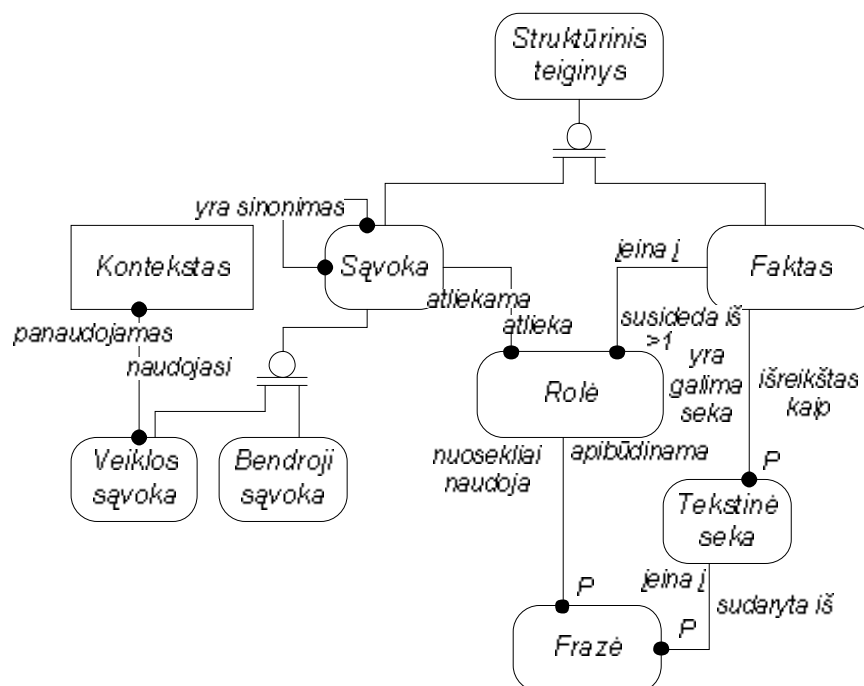
- pasižymėtų aukštu struktūrizavimo lygiu;
- būtų susiję tarpusavyje;
- leistų įvesti GUIDE modelio struktūrinių ir veiksmo teiginių taisykles.

Norint struktūrizuoti VT, šablonai turi būti aprašyti GUIDE modelyje išskirtiems VT tipams. Pirmiausia pristatomi šablonai paprasčiausiam veiklos taisyklių tipui – *struktūriniam teiginiams*, po to *veiksmo teiginiams*, nes *struktūriniai teiginiai* yra *veiksmo teiginių* sudedamoji dalis. Sistemoje, kurioje realizuoti *veiksmo* ir *struktūriniai teiginiai* galima išsaugoti bet kokį struktūrinį ar dinaminį organizacijos veiklos aspektą. *Išvesties* šablonai nepristatomi, nes išvestis apibrėžia kaip reikia išvesti vienas veiklos taisykles iš kitų. *Išvestis* sumažina veiklos taisyklių skaičių, tačiau papildomo organizacijos veiklos aspekto nenustato.

3.1.2 Struktūrinių teiginių šablonai

Struktūrinis teiginys nustato veiklai svarbius aspektus, kurie arba egzistuoja kaip koncepcinės esybės, arba siejasi su kitais reikšmingais dalykais. *Struktūrinis teiginys* – tipas, kuris sudaro visi kitų VT tipų pagrindą. Toliau bus detalizuojami šio tipo taisyklių šablonai.

Struktūriniai teiginiai skirstomi į *Sąvokas* ir *Faktus* (3.1.1 pav.).



3.1.1 pav. Sąvokos ir faktai

Sąvoka – tai žodis arba frazė, turinti konkrečią reikšmę veikloje. *Sąvoka* yra elementari VT dalis, kuri negali būti išskaidyta į dar mažesnes dalis. *Sąvoka* užrašoma kaip simbolių seka. Simbolių seka negali sutapti su taisyklių tipų šablonų raktiniais žodžiais.

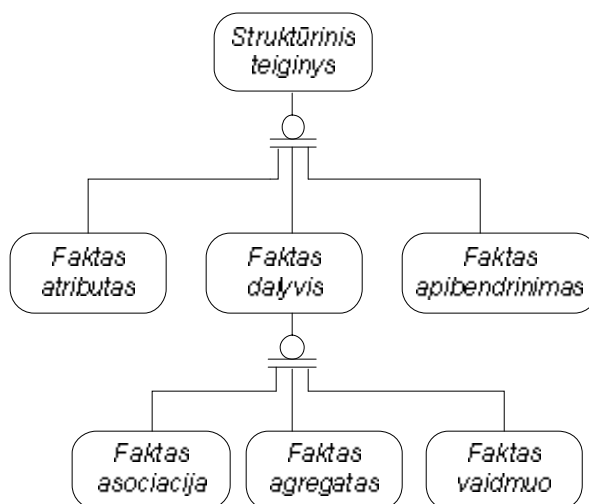
Pvz. *Sąvoka Gedimino pilis*. *Sąvoką* sudaro atitinkama simbolių seka, kuri logiškai nebegali būti išskaidyta į atskiras dalis, nes būtų prarandama informacija.

Faktas nusako ryšį tarp dviejų ar daugiau *sąvokų*. *Faktas* apima dvi arba daugiau *sąvokų*, o *sąvoka* gali būti panaudojama viename ir daugiau *faktų*.

Pvz. *Faktas Automobilis yra transporto priemonė* apima dvi *sąvokas*: **Automobilis** ir **Transporto priemonė**. Taip pat *sąvoka* gali būti panaudota keliuose *faktuose*. Pvz. *sąvoka Automobilis* gali būti panaudota ir kitame *fakte* **Spalva yra automobilio savybė**.

Faktai yra skirstomi į tipus (3.1.2 pav.). Kiekvieno tipo *faktai* užrašomi pagal nustatytą šabloną. Šabloną sudaro griežtai nustatytu eiliškumu pateikiami *raktiniai žodžiai* bei *sąvokos*. *Faktas* atkuriamas į šabloną nustatytu eiliškumu įstačius *raktinius žodžius* bei *fakto sąvokas*. Šablono sudėtinės dalys:

- *Raktinis žodis* – kiekvienam VT tipui nustatyta ir užfiksuota alfabetinių simbolių seka, kuri atskiria vienas *fakto dalis* nuo kitų. Vienas tipas gali turėti kelis *raktinius žodžius*.
- *Sąvoka* – kintanti *fakto dalis*. Kad būtų paprasčiau apibūdinti įvairių tipų *faktų šablonus*, *sąvokos fakte* skirstomos:
 - *Pagrindas* – *sąvoka*, panaudojama *fakte*. Vienas *fakto* tipas turi tik vieną *pagrindą*, o *pagrindą* sudaro tik viena *sąvoka*.
 - *Pirmoji kompozicija* – *sąvoka*, panaudojama *fakte*. Vienas *fakto* tipas turi tik vieną *pirmąją kompoziciją*, tačiau *kompoziciją* gali sudaryti kelios *sąvokos*.
 - *Antroji kompozicija* – *sąvoka*, panaudojama *fakte*. Vienas *fakto* tipas turi tik vieną *antrąją kompoziciją*, o *kompoziciją* sudaro tik viena *sąvoka*. *Antroji kompozicija* galima tik tuo atveju, kai *faktas* turi *pirmąją kompoziciją*, tačiau semantine prasme šios dvi *sąvokos* nesusijusios.
 - *Veiksmažodis* – *sąvoka*, panaudojama *fakte*. Vienas *fakto* tipas turi tik vieną *veiksmažodį*, o *pagrindą* sudaro tik viena *sąvoka*.



3.1.2 pav. Faktų tipai

Fakto tipai ir jų šablonai:

- *Atributas* yra nurodymas, kad tam tikra *sąvoka* apibūdina tam tikrą kitos *sąvokos* aspektą, t. y. *sąvoka yra* kitos *sąvokos* atributas.

Atributo šablonas:

<pirmoji kompozicija> yra <pagrindas> savybė.

Atributo šablono sudėtinės dalys apibūdintos 3.1.1 lentelėje.

3.1.1 lentelė. Atributo šablono sudėtinės dalys

Šablono sudėtinė dalis	Aprašymas
pirmoji kompozicija	<i>Sąvokos</i> , kurios apibūdina tam tikrus pagrinde nurodytos <i>sąvokos</i> aspektus
yra	Pirmasis šablono <i>raktinis žodis</i>
pagrindas	<i>Sąvoka</i> , kurios tam tikrą aspektą apibūdina pirmoje kompozicijoje nurodytos <i>sąvokos</i>
savybė	Antrasis šablono <i>raktinis žodis</i>

Pvz. **Spalva yra automobilio savybė**. Čia **spalva** – *pirmoji kompozicija*, **yra** – pirmasis *raktinis žodis*, **automobilis** – *pagrindas*, **savybė** – antrasis *raktinis žodis*.

- *Apibendrinimas* yra nurodymas, kad tam tikra *sąvoka* (supertipas) yra vienos ar daugiau kitų *sąvokų* potipių apibendrinimas. Šio tipo fakte viena *sąvoka* aprašo kitos *sąvokos* variantų poaibį.

Apibendrinimo šablonas:

<pirmoji kompozicija> yra <pagrindas>.

Apibendrinimo šablono sudėtinės dalys apibūdintos 3.1.2 lentelėje.

3.1.2 lentelė. Apibendrinimo šablono sudėtinės dalys

Šablono sudėtinė dalis	Aprašymas
pirmoji kompozicija	<i>Sąvokos</i> , kurios aprašo pagrinde nurodytos <i>sąvokos</i> variantų poaibį
yra	Šablono <i>raktinis žodis</i>
pagrindas	<i>Sąvoka</i> , kuri apibendrina pirmojoje kompozicijoje nurodytas <i>sąvokas</i>

Pvz. **Automobilis yra transporto priemonė**. Čia **automobilis** – *pirmoji kompozicija*, **yra** – *raktinis žodis*, **transporto priemonė** – *pagrindas*.

- *Dalyvis* yra tam tikra *sąvoka*, reiškianti ryšį tarp kitų *sąvokų*. *Faktas* – *dalyvis* dar skirstomas:
 - *Agregatas* – *faktas* - *dalyvis*, kuris nusako “Įeina į/susideda iš” sąryšį. Fakto *sąvokos* apibrėžia komponento tipus, sudarančius visumą.

Agregato šablonas:

<pagrindas> susideda iš <pirmoji kompozicija>.

Agregato šablono sudėtinės dalys apibūdintos 3.1.3 lentelėje.

3.1.3 lentelė. Agregato šablono sudėtinės dalys

Šablono sudėtinė dalis	Aprašymas
pagrindas	<i>Sąvoka</i> , kurios sudėtinės dalys nurodytos pirmoje kompozicijoje
susideda	Pirmasis šablono <i>raktinis žodis</i>
iš	Antrasis šablono <i>raktinis žodis</i>
pirmoji kompozicija	<i>Sąvokos</i> , iš kurių susideda pagrinde nurodyta <i>sąvoka</i>

Pvz. **Kompiuteris susideda iš sisteminio bloko, vaizduoklio, pelės, klaviatūros**. Čia **kompiuteris** – *pagrindas*, **susideda** – pirmasis *raktinis žodis*, **iš** – antrasis *raktinis žodis*, **sisteminis blokas, vaizduoklis, pelė, klaviatūra** – *pirmoji kompozicija*, sudaryta iš keturių *sąvokų*.

- *Vaidmuo* – *faktas* - *dalyvis*, aprašantis, kaip viena *sąvoka* gali tapti veikėju (kita *sąvoka*), sąveikaudama su savo aplinka.

Vaidmens šablonas:

<pagrindas> gali būti <antroji kompozicija>, <veiksmažodis> <pirmoji kompozicija>.

Vaidmens šablono sudėtinės dalys apibūdintos 3.1.4 lentelėje.

3.1.4 lentelė. Vaidmens šablono sudėtinės dalys

Šablono sudėtinė dalis	Aprašymas
pagrindas	<i>Sąvoka</i> , kuri gali tapti antroje kompozicijoje nurodyta <i>sąvoka</i> , sąveikaudama su aplinka, kuri nurodyta pirmoje kompozicijoje
gali	Pirmasis šablono <i>raktinis žodis</i>
būti	Antrasis šablono <i>raktinis žodis</i>
antroji kompozicija	<i>Sąvoka</i> , kuria sąveikaudama su aplinka gali tapti pagrinde nurodyta <i>sąvoka</i>
veiksmažodis	<i>Sąvoka</i> , kuri apibrėžia pagrindo <i>sąvokos</i> sąveikos esmę su pirmosios kompozicijos <i>sąvokos</i> apibrėžta aplinka
pirmoji kompozicija	<i>Sąvoka</i> , kuri nurodo aplinką, su kuria sąveikauja <i>sąvoka</i>

Pvz. **Asmuo gali būti bedarbis, netekęs darbo**. Čia **asmuo** – *pagrindas*, **gali** – pirmasis *raktinis žodis*, **būti** – antrasis *raktinis žodis*, **bedarbis** – *antroji kompozicija*, **netekęs** – *veiksmažodis*, **darbo** – *pirmoji kompozicija*.

- *Asociacija – faktas - dalyvis*, tiesiog atspindintis bet kokį kitą ryšio tipą, kuris yra ne vaidmuo ir ne agregatas. Šio tipo faktą apibūdina *veiksmažodžio sąvoka*, kuri apibrėžia ryšio esmę.

Asociacijos šablonas:

<pagrindas> gali <veiksmažodis> <pirmoji kompozicija>.

Asociacijos šablono sudėtinės dalys apibūdintos 3.1.5 lentelėje.

3.1.5 lentelė. Asociacijos šablono sudėtinės dalys

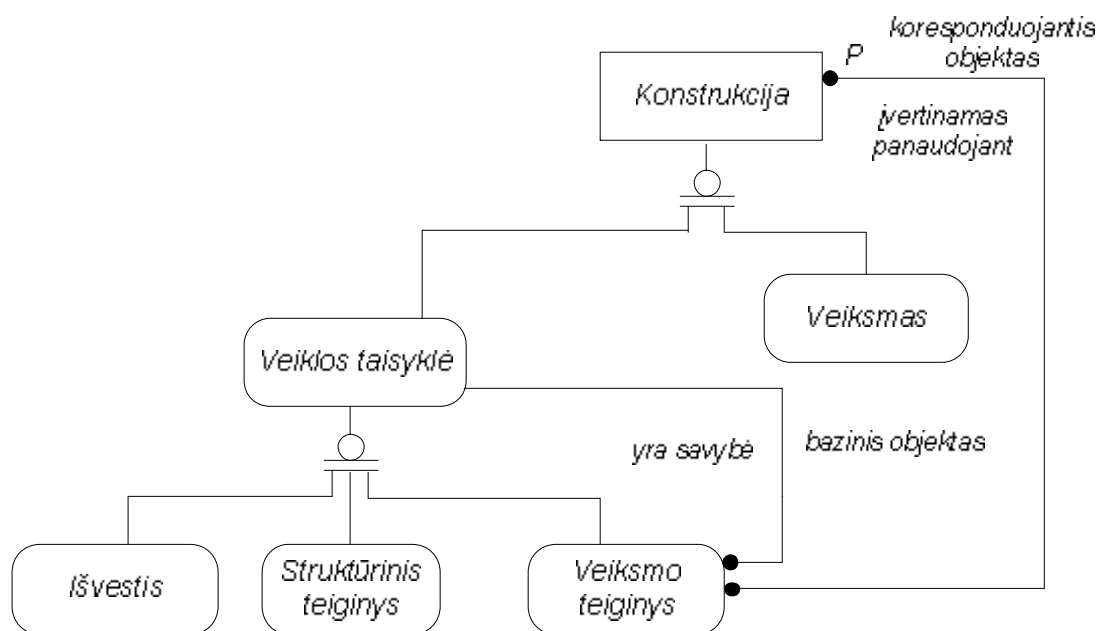
Šablono sudėtinė dalis	Aprašymas
pagrindas	<i>Sąvoka</i> , kuri susijusi su pirmoje kompozicijoje nurodyta <i>sąvoka</i>
gali	Pirmasis šablono <i>raktinis žodis</i>
veiksmažodis	<i>Sąvoka</i> , apibrėžianti ryšio tarp pagrindo ir pirmosios kompozicijos <i>sąvokų</i> esmę
pirmoji kompozicija	<i>Sąvoka</i> , kuri susijusi su pagrinde nurodyta <i>sąvoka</i>

Pvz. **Asmuo gali nusipirkti automobilį**. Čia **asmuo** – pagrindas, **gali** – raktinis žodis, **nusipirkti** – veiksmožodis, **automobilis** – pirmoji kompozicija.

Struktūriniai teiginiai ir jų dalys yra naudojami sudarant veiksmo teiginius, kurie aprašomi žemiau.

3.1.3 Veiksmo teiginių šablonai

Veiksmo teiginys – sakiny, kuris aprašo tam tikrą veiklos dinaminį aspektą. *Veiksmo teiginys* nustato apribojimus, kurie gaunami atlikus veiksmus. *Struktūriniai teiginiai* aprašo galimybes, o *veiksmo teiginiai* įveda apribojimus. *Veiksmo teiginių* modelis parodytas 3.1.3 pav..



3.1.3 pav. Veiksmo teiginiai

Veiksmo teiginys susideda iš bazinio ir koresponduojančio objektų. Bazinis objektas gali būti bet kuri veiklos taisyklė. Iš esmės, *veiklos teiginys* turi būti tam tikros vienos veiklos taisyklės savybė. Frazėje „jeigu ..., tai ...“ bazinis objektas yra frazė po žodelio „jeigu“. Bazinis objektas dažniausiai būna *struktūrinis teiginys*.

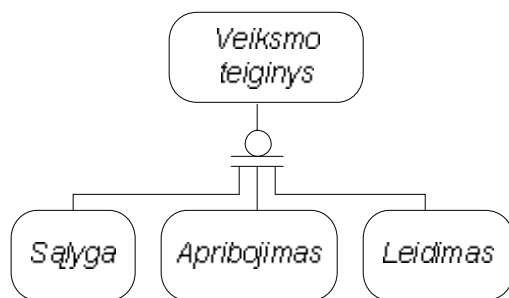
Koresponduojantis objektas gali būti kita veiklos taisyklė arba apibrėžtas veiksmas. Šias dvi galimybes apibendrina sąvoka *konstrukcija*. *Veiksmo teiginys* gali būti įvertintas panaudojant *konstrukciją*. Frazėje „jeigu ..., tai ...“ konstrukcijos vieta po žodelio „tai“. Pvz. **Studentas privalo atlikti laboratorinius darbus**. Sąvoka **Studentas** yra bazinis objektas, o faktas **Studentas gali atlikti laboratorinius darbus** yra koresponduojantis objektas.

Veiksmas – procesas, kuris įvykdo operaciją su tam tikrais objektais ir gali pakeisti tų objektų būsenas. Pats *veiksmas* negali būti apribotas, tačiau objektai su kuriais jis atlieka operacijas, gali būti apriboti. *Veiksmo* paleidimas arba įvykdymas yra kontroliuojamas veiklos taisyklių. *Veiksmas* gali būti atliktas vieną kartą, jei sąlygos yra patenkinamos.

Veiksmas užrašomas kaip simbolių seka. Simbolių seka negali sutapti su taisyklių tipų šablonų raktiniais žodžiais. *Veiksmas* yra elementari VT dalis, kuri negali būti išskaidyta į dar mažesnes dalis.

Pvz. *Veiksmas Atlikti remontą*. *Veiksmą* sudaro atitinkama simbolių seka, kuri logiškai negali būti išskaidyta į atskiras dalis, nes būtų prarandama informacija arba apie pačią veiksmo operaciją, arba apie jos objektą.

Veiksmo teiginiai yra skirstomi į klases (3.1.4 pav.). Kiekvienos klasės *veiksmo teiginiai* užrašomi pagal nustatytą šabloną. Šabloną sudaro griežtai nustatytu eiliškumu pateikiami *raktiniai žodžiai* bei *baziniai ir koresponduojantys objektai*. *Veiksmo teiginys* atkuriamas į šabloną nustatytu eiliškumu įstačius *raktinius žodžius* bei *bazinius ir koresponduojančius objektus*.



3.1.4 pav. Veiksmo teiginių klasės

Veiksmo teiginių klasės ir jų šablonai:

- *Sąlyga* yra teiginys, kad jei kažkas pasitvirtina (yra tiesa) – pritaikoma kita veiklos taisyklė arba atliekamas veiksmas. Pvz. *Sąlyga* gali klausti: „Ar klientas parodė vairuotojo teises?“, „Ar klientas skolingas?“.

Sąlygos šablonas:

jeigu <bazinis objektas>, tai <koresponduojantis objektas>.

Sąlygos šablono sudėtinės dalys apibūdintos 3.1.6 lentelėje.

3.1.6 lentelė. Sąlygos šablono sudėtinės dalys

Šablono sudėtinė dalis	Aprašymas
jeigu	Pirmasis šablono raktinis žodis
bazinis objektas	Teiginys, kuriam pasitvirtinus yra pritaikoma koresponduojančio objekto nurodyta veiklos taisyklė. Bazinio objekto suformavimo taisyklės ir jo sudėtinės dalys aptariamos žemiau
tai	Antrasis šablono raktinis žodis
koresponduojantis objektas	Veiklos taisyklė, kuri yra pritaikoma patenkinus bazinio objekto nurodytą sąlygą. Koresponduojančio objekto suformavimo taisyklės ir jo sudėtinės dalys aptariamos žemiau

Pvz. **Jeigu asmuo turi vairuotojo teises, tai išnuomoti automobilį**. Čia **jeigu** – pirmasis raktinis žodis, **asmuo turi vairuotojo teises** – bazinis objektas, **tai** – antrasis raktinis žodis, **išnuomoti automobilį** – koresponduojantis objektas.

Sąlygos baziniu objektu gali būti tokių tipų struktūriniai teiginiai: *apibendrinimas* ir *asociacija*. Nuo šių tipų priklauso bazinio objekto šablonai.

Jeigu bazinis objektas yra apibendrinimo struktūrinis teiginys, tuomet bazinio objekto šablonas:

<pagrindo sąvoka> yra <pirmosios kompozicijos sąvoka>

Testuojant bazinio objekto nurodytą sąlygą yra tikrinama, ar konkretus pagrinde nurodytos sąvokos egzempliorius yra iš to paties variantų poaibio, kaip ir viena konkreti pirmosios kompozicijos sąvoka. Jeigu konkretus egzempliorius priklauso pirmosios kompozicijos sąvokos variantų poaibiui, tuomet sąlyga tenkinama. Sąlygos, kurios bazinis objektas yra *apibendrinimas*, šablono sudėtinės dalys apibūdintos 3.1.7 lentelėje.

3.1.7 lentelė. Sąlygos bazinio objekto (apibendrinimo) šablono sudėtinės dalys

Šablono sudėtinė dalis	Aprašymas
pagrindo sąvoka	Sąvoka, kuri yra pirmojoje kompozicijoje nurodytos sąvokos apibendrinimas
yra	Pirmasis bazinio objekto šablono raktinis žodis
pirmosios kompozicijos sąvoka	Sąvoka, kuri yra pagrinde nurodytos sąvokos variantų poaibis. Baziniame objekte gali būti panaudota tik viena iš apibendrinime nurodytų pirmosios kompozicijos sąvokų

Pvz. Tarkime, kad turime apibendrinimą **Raudona, žalia, mėlyna yra automobilio spalvos**. Čia **automobilio spalva** – pagrindas (Įvairių spalvų apibendrinimas), **yra** – raktinis žodis, **raudona, žalia, mėlyna** – pirmoji kompozicija (automobilio spalvų variantų poaibis). Sudarant bazinio objekto sąlygą, apibendrinimo pagrindas yra bazinio objekto pagrindo sąvoka, o viena iš apibendrinimo pirmosios kompozicijos sąvokų – pirmosios kompozicijos sąvoka. Tarkime, kad turime sudarytą sąlygą **Automobilio spalva yra žalia**. Čia **automobilio spalva** – pagrindo sąvoka, **yra** – pirmasis raktinis žodis, **žalia** – pirmosios kompozicijos sąvoka. Ši sąlyga yra klausimas: „Ar automobilio spalva žalia?“. Jei konkretaus automobilio spalva žalia, tuomet sąlyga tenkinama.

Jeigu bazinis objektas yra asociacijos struktūrinis teiginys, tuomet bazinio objekto šablonas:

<pagrindo sąvoka> <veiksmazodžio sąvoka> <pirmosios kompozicijos sąvoka>

Testuojant bazinio objekto nurodytą sąlygą yra tikrinama, ar konkretus pagrinde nurodytos sąvokos egzempliorius yra susijęs realiu veiksmazodžio nurodytu ryšiu su viena konkrečia pirmoje kompozicijoje apibrėžta sąvoka. Jeigu konkretus egzempliorius veiksmazodžio nurodytu ryšiu yra susijęs su pirmąja kompozicija, tuomet sąlyga tenkinama. Sąlygos, kurios bazinis objektas yra asociacija, šablono sudėtinės dalys apibūdintos 3.1.8 lentelėje.

3.1.8 lentelė. Sąlygos bazinio objekto (asociacija) šablono sudėtinės dalys

Šablono sudėtinė dalis	Aprašymas
pagrindo sąvoka	Sąvoka, kuri yra susijusi su pirmoje kompozicijoje nurodyta sąvoka
veiksmazodžio sąvoka	Sąvoka, apibrėžianti ryšio tarp pagrindo ir pirmosios kompozicijos sąvokų esmę
pirmosios kompozicijos sąvoka	Sąvoka, kuri yra susijusi su pagrinde nurodyta sąvoka. Baziniame objekte gali būti panaudota tik viena iš asociacijoje nurodytų pirmosios kompozicijos sąvokų

Pvz. Tarkime, kad turime asociaciją **Asmuo gali turėti vairuotojo teises**. Čia **asmuo** – pagrindas, **gali** – pirmasis raktinis žodis, **turėti** – veiksmazodis, **vairuotojo teisės** – pirmoji kompozicija. Sudarant bazinio objekto sąlygą, asociacijos pagrindas yra bazinio objekto pagrindo sąvoka, o viena iš asociacijos pirmosios kompozicijos sąvokų – pirmosios kompozicijos sąvoka, asociacijos veiksmazodis – veiksmazodžio sąvoka. Tarkime, kad turime sudarytą sąlygą **Asmuo turi vairuotojo teises**. Čia **asmuo** – pagrindo sąvoka, **turi** – veiksmazodžio sąvoka, **vairuotojo**

teisės – pirmosios ko mpozicijos sąvoka . Ši sąlyga yra klausimas: „Ar asmuo turi vairuotojo teises?“. Jei konkretaus asmuo turi vairuotojo teises, tuomet sąlyga tenkinama.

Sąlygos koresponduojančiu objektu gali būti *veiksmo teiginys* – *apribojimas* (apribojimai aptariami žemiau) arba *veiksmas* . Koresponduojantis objektas neturi atskiro šablono, todėl yra jį apibrėžiančio *apribojimo* ar *veiksmo* kopija.

Pvz. Tarkime, kad turime nustatyti *sąlygą*, kad jei vairuotojas pažeidė eismo taisykles, tai jis turi sumokėti baudą. Sąlygos bazinis objektas yra *asociacija* **Asmuo pažeidė eismo taisykles**, o koresponduojantis objektas – *apribojimas* **Asmuo privalo sumokėti baudą**. Tuomet visa *Sąlyga*: **Jeigu asmuo pažeidė eismo taisykles, tai asmuo privalo sumokėti baudą**.

Kitas pvz. Tarkime, kad turime nustatyti *sąlygą*, kad jei automobilio padangos yra susidėvėjusios, jas reikia pakeisti. Sąlygos bazinis objektas yra *apibendrinimas* **Automobilio padangos yra susidėvėjusios**, o koresponduojantis objektas – *veiksmas* **Pakeisti padangas**. Tuomet visa *Sąlyga*: **Jeigu automobilio padangos yra susidėvėjusios, tai pakeisti padangas**.

- *Apribojimas* yra teiginys, kuris privalo būti teisingas. Apribojimas įsigalioja nedelsiant, kadangi ji uždraudžia bet kuriuos veiksmus, kurių rezultate teiginys gali tapti neteisingas. Jeigu *sąlyga* gali patikrinti konkrečią reikšmę, ir pagal patikrinimo rezultatus nurodyti tam tikrą veiksmą, tai apribojimas deklaruoja teiginį ir neleidžia atlikti veiksmų, kurių rezultate teiginys taptų neteisingas.

Apribojimo šablonas:

<bazinis objektas> privalo <koresponduojantis objektas>.

Apribojimo šablono sudėtinės dalys apibūdintos 3.1.9 lentelėje.

3.1.9 lentelė. Apribojimo šablono sudėtinės dalys

Šablono sudėtinė dalis	Aprašymas
bazinis objektas	Teiginys, kurio veiksmų aibę apriboja koresponduojantis objektas. Bazinio objekto suformavimo taisyklės ir jo sudėtinės dalys aptariamos žemiau
privalo	Šablono raktinis žodis
koresponduojantis objektas	Teiginys, apriboja galimybes atlikti veiksmus su baziniu objektu. Koresponduojančio objekto suformavimo taisyklės ir jo sudėtinės dalys aptariamos žemiau

Pvz. **Asmuo privalo pateikti asmens dokumentą**. Čia **asmuo** – *bazinis objektas*, **privalo** – *raktinis žodis*, **pateikti asmens dokumentą** – *koresponduojantis objektas*.

Apribojimo bazinis objektas yra *Sąvoka*. Bazinis objektas neturi atskiro šablono, jį sudaro konkreti *sąvoka*. Bazinį objektą sudaranti *sąvoka* privalo būti koresponduojančio objekto pagrindas. Pvz. **Studentas privalo išlaikyti egzaminą**. Čia **studentas** – *bazinis objektas*.

Apribojimo koresponduojančiu objektu gali būti tokių tipų struktūriniai teiginiai: *atributas*, *agregatas* ir *asociacija*. Nuo šių tipų priklauso koresponduojančio objekto šablonai.

Jeigu koresponduojantis objektas yra *asociacija*, tuomet koresponduojančio objekto šablonas:

<veiksmažodžio sąvoka> <pirmosios kompozicijos sąvoka>

Apribojimo, kurio koresponduojantis objektas yra *asociacija*, šablono sudėtinės dalys apibūdintos 3.1.10 lentelėje.

3.1.10 lentelė. *Apribojimo* koresponduojančio objekto (*asociacijos*) šablono sudėtinės dalys

Šablono sudėtinė dalis	Aprašymas
veiksmažodžio sąvoka	<i>Sąvoka</i> , apibrėžianti apribojamo ryšio esmę
pirmosios kompozicijos sąvokos	<i>Sąvokos</i> , kurios veiksmažodžio <i>sąvokos</i> nurodytu ryšiu privalo būti susijusios su asociacijos pagrinde nurodyta <i>sąvoka</i> (pagrindo <i>sąvoka</i> būtinai turi būti nurodyta apribojimo bazėje)

Pvz. Tarkime, kad turime asociaciją **Ūkininkas gali auginti galvijus**. Čia **ūkininkas** – *pagrindas*, **gali** – *raktinis žodis*, **auginti** – *veiksmažodis*, **galvijai** – *pirmoji kompozicija*. Sudarant apribojimą, asociacijos pagrindas yra bazinis objektas, o asociacijos pirmoji kompozicija ir veiksmažodis sudaro koresponduojančio objekto pirmosios kompozicijos sąvokas ir veiksmažodžio sąvoką. Tada apribojimas: **Ūkininkas privalo auginti galvijus**. Čia **ūkininkas** – *bazinis objektas*, **privalo** – *raktinis žodis*, **auginti** – koresponduojančio objekto *veiksmažodžio sąvoka*, **galvijai** – koresponduojančio objekto *pirmosios kompozicijos sąvoka*. Šis apribojimas nustato, kad bazinio objekto sąvoka **ūkininkas** būtinai turi **auginti galvijus**.

Jeigu koresponduojantis objektas yra *atributas* arba *agregatas*, tuomet koresponduojančio objekto šablonas:

turėti <pirmosios kompozicijos sąvoka>

Apribojimo , kurio koresponduojantis objektas yra *atributas* arba *agregatas* , šablono sudėtinės dalys apibūdintos 3.1.11 lentelėje.

3.1.11 lentelė. Apribojimo koresponduojančio objekto (atributo arba agregato) šablono sudėtinės dalys

Šablono sudėtinė dalis	Aprašymas
turėti	raktinis žodis
pirmosios kompozicijos sąvokos	1) <i>Sąvokos</i> , kurios privalo apibūdinti tam tikrus <i>atributo</i> pagrindo <i>sąvokos</i> aspektus (pagrindo <i>sąvoka</i> būtinai turi būti nurodyta apribojimo bazėje). 2) <i>Sąvokos</i> , kurios privalo būti <i>agregato</i> pagrindo <i>sąvokos</i> sudėtinės dalys (pagrindo <i>sąvoka</i> būtinai turi būti nurodyta apribojimo bazėje).

Pvz. Tarkime, kad turime atributą **Durų skaičius yra automobilio savybė**. Čia **durų skaičius** – *pirmoji kompozicija* , **yra** – *raktinis žodis* , **automobilis** – *pagrindas* . Sudarant apribojimą, atributo pagrindas yra bazinis objektas, o atributo pirmoji kompozicija ir koresponduojančio objekto raktinis žodis sudaro koresponduojantį objektą. Tada apribojimas: **Automobilis privalo turėti durų skaičių** (Įvedant duomenis į duomenų saugyklą). Čia **automobilis** – *bazinis objektas* , **privalo** – *raktinis žodis* , **turėti** – koresponduojančio objekto *raktinis žodis* , **durų skaičius** – koresponduojančio objekto *pirmosios kompozicijos sąvoką* . Šis *apribojimas* nustato, kad bazinio objekto sąvokai **automobilis** būtinai turi būti nurodytas **durų skaičius**.

- *Leidimas* apibrėžia konkrečią prerogatyvą ar privilegiją tam tikro veiksmo atžvilgiu. Leidimas – teiginys, kurį galima atvaizduoti predikatu „Tiktai X gali atlikti Y“, kur X yra aktorius, o Y yra veiksmas, kuris gali būti įvykdytas ar atliktas. Leidimai yra suteikiami tik toms sąvokoms, kurios gali atlikti nepriklausomą veiklą, pvz. žmonės, padaliniai, kompiuteriai ir t. t.

Leidimo šablonas:

tiktai <bazinis objektas> gali <koresponduojantis objektas>.

Leidimo šablono sudėtinės dalys apibūdintos 3.1.12 lentelėje.

3.1.12 lentelė. Leidimo šablono sudėtinės dalys

Šablono sudėtinė dalis	Aprašymas
tiktai	Pirmasis šablono raktinis žodis
bazinis objektas	Šavoka, kuri turi privilegiją koresponduojančio objekto konstrukcijos atžvilgiu. Bazinio objekto suformavimo taisyklės ir jo sudėtinės dalys aptariamos žemiau
gali	Antrasis šablono raktinis žodis
koresponduojantis objektas	Veiksmas, kurio atžvilgiu bazinis objektas turi privilegiją. Koresponduojančio objekto suformavimo taisyklės ir jo sudėtinės dalys aptariamos žemiau

Pvz. **Tiktai direktorius gali peržiūrėti Įmonės finansinius rodiklius**. Čia **tiktai** – pirmasis raktinis žodis, **direktorius** – bazinis objektas, **gali** – antrasis raktinis žodis, **peržiūrėti Įmonės finansinius rodiklius** – koresponduojantis objektas.

Leidimo bazinis objektas yra *Šavoka*. Bazinis objektas neturi atskiro šablono, jį sudaro konkreti *šavoka*. Bazinio objekto *šavoka* privalo sugebėti atlikti nepriklausomą veiklą. Pvz. **Tiktai pažangus studentas gali laikyti egzaminą**. Čia **pažangus studentas** – bazinis objektas.

Leidimo koresponduojančiu objektu gali būti tik *veiksmas*. Koresponduojantis objektas neturi atskiro šablono, jis yra *veiksmo* kopija.

Pvz. Tarkime, kad automobilio važiuoklės remontą gali atlikti tik pakankamą kvalifikaciją turintis mechanikas. Leidimo bazinis objektas yra *šavoka* **Kvalifikuotas mechanikas**, o koresponduojantis objektas – *veiksmas* **Remontuoti automobilio važiuoklę**. Tuomet visas *Leidimas*: **Tiktai kvalifikuotas mechanikas gali remontuoti automobilio važiuoklę**.

3.1.4 Šablonų panaudojimas veiklos taisyklių saugyklai

Apibendrinant šablonų panaudojimą veiklos taisyklių formos struktūrizavimui galima teigti, kad jie geriausiai tinka struktūrizuotam veiklos taisyklių užrašymui. Apibrėžti įvairių *struktūrinių* ir *veiksmo teiginių* tipų šablonai leidžia išsaugoti įvairius organizacijos struktūrinius ir dinامينius aspektus. Šablonas išskaido veiklos taisyklės į logiškai susijusias jos sudedamąsias dalis, kurias galima išsaugoti duomenų saugykloje. Žemiau nagrinėjama *struktūrinių* ir *veiksmo teiginių* VT saugyklos sudarymo galimybė ir pateikiamas jos specifikuotas modelis.

3.2 Veiklos taisyklių saugykla

3.2.1 Reikalavimai saugyklai

Turint šablonų pagrindu sudarytą VT specifikaciją, VT galima kaupti duomenų saugykloje, kur jomis galima lengvai manipuluoti [8, 15]. Kiekviena VT suformuojama iš saugykloje išsaugotų jos sudėtinių dalių ir tipo raktinių žodžių.

Kuriamam VT saugyklos modeliui yra keliami šie reikalavimai [6]:

1. Informacijos išsaugojimo lygis turi būti toks, kad būtų galima į duomenų bazę įvesti duomenis apie bet kokio *struktūrinio teiginio* potipio taisyklę, struktūrizuoto pagal GUIDE metodą.
2. Saugyklos duomenų bazėje saugomos informacijos turi pakakti, norint atkurti visą pradinę VT struktūrą.
3. Saugoma tik ta informacija apie taisyklę, kuri atspindi VT struktūroje. Informacija apie kiekvieno iš VT tipų struktūrą nesaugoma.

Taigi saugykloje turi būti saugoma tik VT struktūroje atspindinti informacija. Priimama prielaida, jog informacija apie kiekvieno tipo struktūrą bus užkoduota tiesiogiai VT atvaizdavimo sistemoje.

Veiklos taisyklių saugykla privalo išsaugoti visų šablonais apibrėžtų *struktūrinių* ir *veiksmo teiginių* tipų veiklos taisykles. Žemiau pateikiamas tokios saugyklos modelis.

3.2.2 Struktūrinių teiginių saugyklos modelis

Struktūriniai teiginiai skirstomi į *Sąvokas* ir *Faktus*, kuriems išsaugoti reikalingos atskiros lentelės. *Sąvokai* saugoti naudojama lentelė **Sąvoka** (3.2.1 pav.), kurios atributai: **s_id** – unikalus identifikacinis *sąvokos* kodas; **frazė** – *sąvoką* nusakantis žodis arba frazė.

Sąvoka	
PK	s_id
	frazė

3.2.1 pav. Lentelė Sąvoka

Sąvokos lentelė naudojama ir *veiksmožodžio* saugojimui. Lentelėje **Sąvoka** (3.2.2 pav.) yra pridodamas atributas **s_poz** – požymis, rodantis ar įrašas yra *sąvoka*, ar *veiksmožodis*.

Sąvoka	
PK	s_id
	fraze s_pož

3.2.2 pav. Modifikuota lentelė Sąvoka

Pvz. Turime veiklos taisyklę **Asmuo gali nusipirkti automobilį**. Saugykloje reikia išsaugoti sąvokas **asmuo**, **automobilis** bei veiksmažodį **nusipirkti**. *Sąvokos* lentelėje **sąvoka** turės pažymėtą požymį **s_pož**, tuo tarpu *veiksmažodžio* požymis **s_pož** nebus pažymėtas.

Faktui saugoti yra naudojama lentelė **Faktas** (3.2.3 pav.), kurios atributas: **f_id** – unikalus identifikacinis *fakto* kodas.

Faktas	
PK	f_id

3.2.3 pav. Lentelė Faktas

Faktą sudaro *sąvokų* *rolės*, kurios aprašo *fakte* dalyvaujančių *sąvokų* atliekamą semantinį vaidmenį. Kiekviena *sąvoka* gali atlikti vieną ar daugiau *sąvokų* *rolių*, o kiekviena *sąvokų* *rolė* turi panaudoti vieną *sąvoką* viename *fakte*.

Pvz. *Faktas* **Automobilis yra transporto priemonė** turi dvi *sąvokų* *roles*, kurios aprašo *sąvokų* **Automobilis** ir **Transporto priemonė** semantinį vaidmenį *fakte*. *Sąvokų* *rolėms* saugoti yra naudojama lentelė **Rolė** (3.2.4 pav.), kurios atributai: **f_id** – unikalus identifikacinis *fakto* kodas, **s_id** – unikalus identifikacinis *sąvokos* kodas.

Role	
FK1	s_id
FK2	f_id

3.2.4 pav. Lentelė Rolė

Faktą dažnai galima išreikšti keliais būdais. Netgi dvi *sąvokos* apimantis *faktas* gali būti nusakomas mažiausiai dviem būdais, priklausomai nuo *sąvokų* *rolių* vietos *fakte*. Pvz. *faktą* **Automobilis yra transporto priemonė**, sukeitus *sąvokas* vietomis, galima išreikšti taip: **Transporto priemonė yra automobilis**. Šiuos abu *faktus* sudaro tos pačios *sąvokos*, tačiau jų semantinė prasmė yra visiškai skirtinga. Veiklos taisyklių modelyje tai atvaizduojama *tekstine seka*. Kiekviena *tekstinė seka* sudaroma iš *frazių*, kuriose nuosekliai panaudojamos *sąvokų* *rolės*. Kad išsaugotume *tekstines*

sekas ir *frazes* reikalingos dvi naujos lentelės. Tačiau modifikavus lentelę **Rolė** (3.2.5 pav.) ir pridėjus du papildomus laukus **r_tipas** – rolės tipas, nusakantis *sąvokos* vietą *fakte*, **r_nr** – *sąvokos* eilės numeris tarp to paties tipo *sąvokų*, galima supaprastinti VT saugyklos struktūrą.

Role	
FK1	s_id
FK2	f_id
	r_tipas
	r_nr

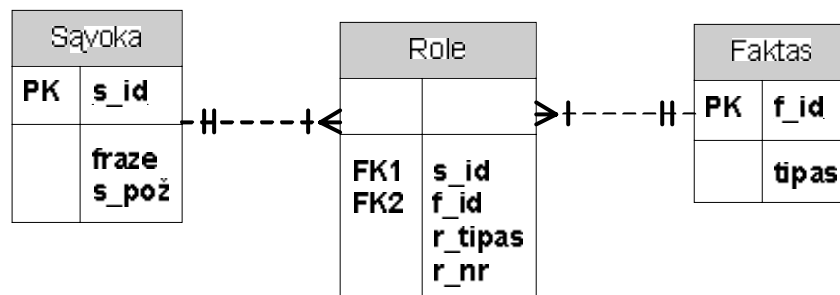
3.2.5 pav. Modifikuota lentelė Rolė

Faktai yra skirstomi į tipus. Kad šie tipai būtų atspindėti, lentelėje *Faktas* (3.2.6 pav.) reikia pridėti lauką **tipas**.

Faktas	
PK	f_id
	tipas

3.2.6 pav. Modifikuota lentelė Faktas

Visas saugyklos modelis, išsaugantis *struktūrinių teiginių* tipo VT, pagal GUIDE veiklos taisyklių metodologiją yra pavaizduotos 3.2.7 pav.



3.2.7 pav. Saugyklos modelis struktūriniams teiginiams išsaugoti

Struktūrinių teiginių saugyklos modelis naudojamas sudarant *veiksmo teiginių* saugyklos modelį, kuris aprašomas žemiau.

3.2.3 Veiksmo teiginių saugyklos modelis

Veiksmo teiginiams saugoti naudojama lentelė **Veiksmo teiginys** (3.2.8 pav.), kurios atributai: **v_id** – unikalus identifikacinis *veiksmo teiginio* kodas, **klasė** – *veiksmo teiginio* klasė.

Veiksmo teiginys	
PK	v_id
	klase

3.2.8 pav. Lentelė Veiksmo teiginys

Saugant *veiksmo teiginius* taip pat būtina išsaugoti ir veiksmus, kuriems sukuriama atskira lentelė **Veiksmas** (3.2.9 pav.). Šios lentelės atributai: **vk_id** – unikalus identifikacinis *veiksmo* kodas; **frazė** – *veiksmą* nusakanti frazė.

Veiksmas	
PK	vk_id
	frazė

3.2.9 pav. Lentelė Veiksmas

Pvz. Turime veiklos taisyklę **Jeigu klientas yra skolingas, tai pranešti policijai**. Saugykloje reikia išsaugoti *veiksmą pranešti policijai*.

Kiekvienas *veiksmo teiginys* turi bazinį ir koresponduojantį objektus, kurie gali būti įvairių rūšių veiklos taisyklės: *sąvokos, faktai, veiksmo teiginiai, veiksmai* . Todėl yra sukuriama lentelė **Objektas** (3.2.10 pav.), kurios atributai: **s_id** – unikalus identifikacinis *sąlygos* kodas, **f_id** – unikalus identifikacinis *fakto* kodas, **v_id** – unikalus identifikacinis *veiksmo teiginio* kodas, **vk_id** – unikalus identifikacinis *veiksmo* kodas, **vt_id** – unikalus identifikacinis *veiksmo teiginio* , kuriam priklauso objektas, kodas.

Objektas	
FK1	s_id
FK2	f_id
FK3	v_id
FK4	vk_id
FK5	tv_id

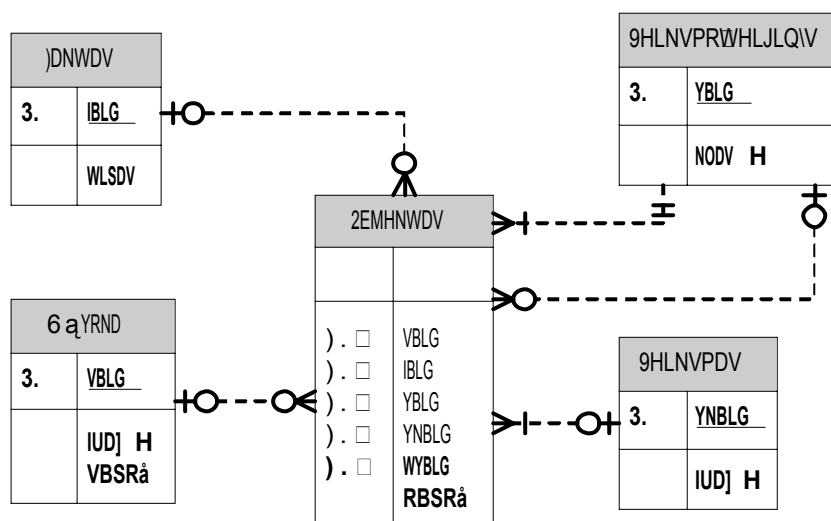
3.2.10 pav. Lentelė Objektas

Veiksmo teiginys turi bazinį ir koresponduojantį objektus, todėl lentelėje **Objektas** (3.2.11 pav.) reikia pridėti papildomą atributą: **o_poz** – požymis, rodantis ar objektas yra *bazinis*, ar *koresponduojantis* .

Objektas	
FK1	s_id
FK2	f_id
FK3	v_id
FK4	vk_id
FK5	tv_id
	o_poz

3.2.11 pav. Modifikuota lentelė Objektas

Saugyklos modelis fragmentas, išsaugantis *veiksmo teiginių* tipo VT, pagal GUIDE veiklos taisyklių metodologiją yra pavaizduotos 3.2.12 pav.



3.2.12 pav. Saugyklos modelis veiksmo teiginiams išsaugoti

Veiksmo teiginys išsaugomas:

- 1) lentelėje **Veiksmo teiginys** įrašomas identifikacinis numeris **v_id** ir klasė **klasė**;
- 2) lentelėje **Objektas** įrašomas bazinis objektas: nurodomas veiksmo teiginys, kuriam priklauso bazinis objektas, **tv_id**, požymis **o_pož** nustatomas į bazinį objektą, nurodomas identifikacinis kodas tos veiklos taisyklės, kuri yra nagrinėjamo veiksmo teiginio bazinis objektas (**s_id** arba **f_id**);
- 3) lentelėje **Objektas** įrašomas koresponduojantis objektas: nurodomas veiksmo teiginys, kuriam priklauso koresponduojantis objektas, **tv_id**, požymis **o_pož** nustatomas į koresponduojantį objektą, nurodomas identifikacinis kodas tos veiklos taisyklės, kuri yra nagrinėjamo veiksmo teiginio koresponduojantis objektas (**f_id**, **v_id** arba **f_id**).

Sudarytas *veiksmo teiginių* saugyklos modelis yra tik dalis bendresnio veiklos taisyklių saugyklos modelio, kuris yra pristatomas žemiau.

3.2.4 Bendras veiklos taisyklių saugyklos modelis

Sudaryta veiklos taisyklių saugykla yra skirta saugoti tik *struktūrinių* ir *veiksmo teiginių* tipo VT. Saugyklos modelį galima nesunkiai praplėsti įtraukiant naujas lenteles bei atributus, kurios leistų išsaugoti įvairią taisyklę identifikuojančią informaciją, kuri nėra tiesiogiai susijusi su pačios taisyklės semantinės reikšmės išsaugojimu. Realios sistemos saugykloje turėtų būti saugomi papildomi duomenys [6, 9], tokie kaip informacija apie VT sukūrimą, modifikavimą ir pan.

Informacijai apie veiklos taisyklių šaltinius (asmenis, dokumentas ir pan.) saugoti naudojama lentelė **Šaltinis** (3.2.13 pav.), kurios atributai: **slt_id** – unikalus identifikacinis *šaltinio* kodas, **pavadinimas** – *šaltinio* pavadinimas, **aprašas** – *šaltinio* aprašas.

Šaltinis	
PK	slt_id
	pavadinimas aprašas

3.2.13 pav. Lentelė Šaltinis

Veiklos taisyklių šaltinius galima priskirti kiekvienam faktui. Ši informacija saugoma lentelėje **Fakto šaltinis** (3.2.14 pav.), kurios atributai: **slt_id** – unikalus identifikacinis *šaltinio* kodas, **f_id** – unikalus identifikacinis *fakto* kodas.

Fakto šaltinis	
FK1	slt_id
FK2	f_id

3.2.14 pav. Lentelė Fakto šaltinis

Veiklos taisyklių šaltinius galima priskirti kiekvienam veiksmo teiginiui. Ši informacija saugoma lentelėje **Veiksmo teiginio šaltinis** (3.2.15 pav.), kurios atributai: **slt_id** – unikalus identifikacinis *šaltinio* kodas, **v_id** – unikalus identifikacinis *veiksmo teiginio* kodas.

Veiksmo teiginio šaltinis	
FK1	slt_id
FK2	v_id

3.2.14 pav. Lentelė Veiksmo teiginio šaltinis

Faktams galima priskirti papildomą informaciją. Kad ši informacija atsispindėtų lentelė **Faktas** papildoma (3.2.15 pav.) atributais: **sukur_data** – *fakto* sukūrimo data, **sukur_asmuo** –

faktą sukūręs asmuo, **modif_data** – *fakto* modifikavimo data, **modif_asmuo** – *faktą* modifikavęs asmuo, **aprašas** – papildoma tekstinė informacija apie *faktą*.

Faktas	
PK	f_id
	tipas sukur_data sukur_asmuo modif_data modif_asmuo aprašas

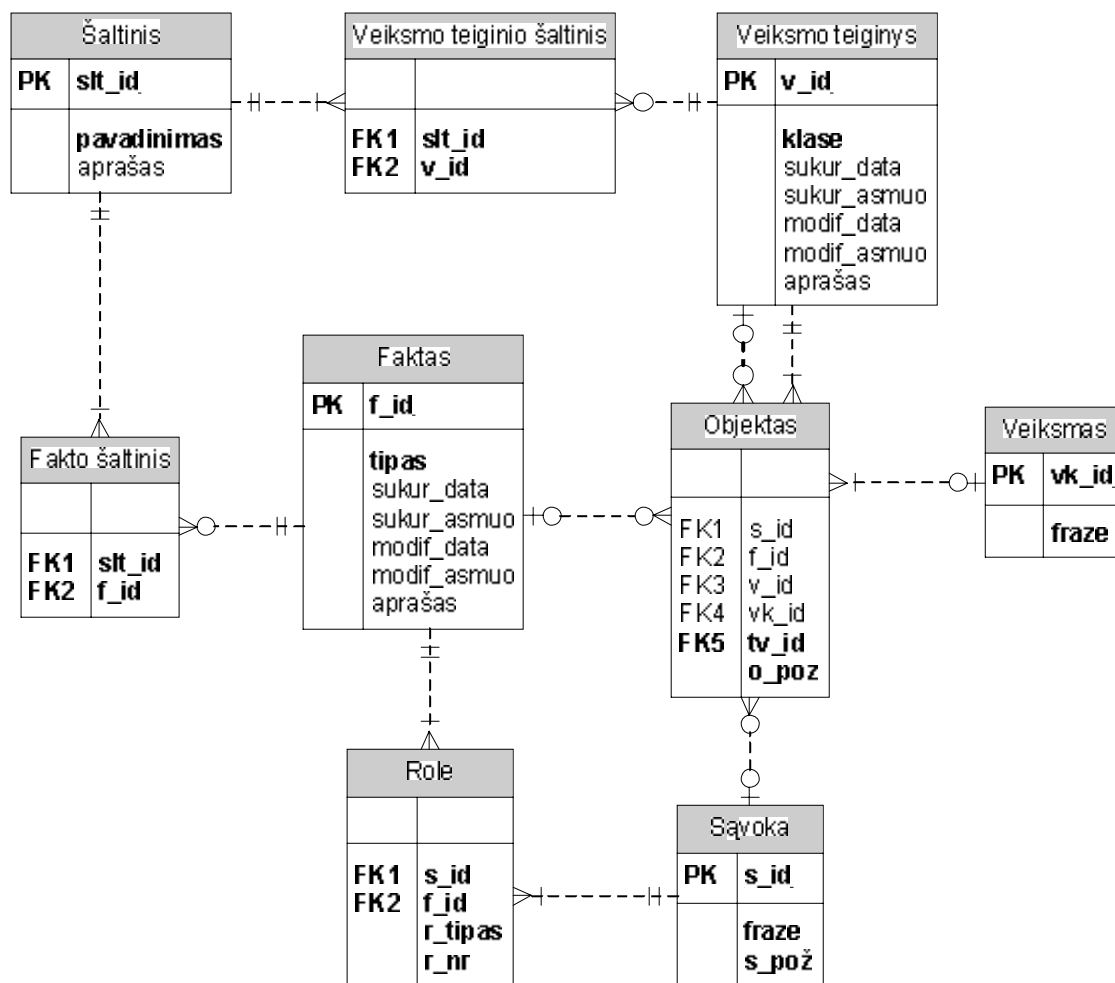
3.2.15 pav. Lentelės Faktas papildymas

Veiksmo teiginiams, kaip ir *faktams*, galima priskirti papildomą informaciją. Kad ši informacija atsispindėtų lentelė **Veiksmo teiginys** papildoma (3.2.16 pav.) atributais: **sukur_data** – *veiksmo teiginio* sukūrimo data, **sukur_asmuo** – *veiksmo teiginį* sukūręs asmuo, **modif_data** – *veiksmo teiginio* modifikavimo data, **modif_asmuo** – *veiksmo teiginį* modifikavęs asmuo, **aprašas** – papildoma tekstinė informacija apie *veiksmo teiginį*.

Veiksmo teiginys	
PK	v_id
	klase sukur_data sukur_asmuo modif_data modif_asmuo aprašas

3.2.16 pav. Lentelės Veiksmo teiginys papildymas

Visas veiklos taisyklių saugyklos modelis, kuris leidžia išsaugoti *struktūrinius ir veiksmo teiginius* bei papildomą informaciją apie juos yra pateiktas 3.2.17 pav.



3.2.17 pav. Veiklos taisyklių saugyklos modelis

Sudarytas saugyklos modelis atitinka VT saugyklos modeliui keliamus reikalavimus:

1. Informacijos išsaugojimo lygis yra pakankamas, kad būtų galima į duomenų bazę įvesti duomenis apie bet kokio *struktūrinio teiginio* potipio taisyklę bei bet kokio *veiksmo teiginio* poklasio taisyklę, struktūrizuotą pagal GUIDE metodą.
2. Saugyklos duomenų bazėje saugomos informacijos pakanka, norint atkurti visą pradinę VT struktūrą.
3. Saugoma tik ta informacija apie taisyklę, kuri atsispindi VT struktūroje. Informacija apie kiekvieno iš VT tipų struktūrą nesaugoma.

Taip pat saugykloje galima saugomi papildomus duomenis, tokius kaip informacija apie VT sukūrimą, modifikavimą ir pan.

4. VEIKLOS TAISYKLIŲ SISTEMOS ARCHITEKTŪRINIAI SPRENDIMAI

4.1 Įvadas

Sistema projektuojama atsižvelgiant į GUIDE veiklos taisyklių metodologiją bei pasiūlytus jos papildymus. Pagrindiniai sistemos tikslai:

- Realizuoti *struktūrinių* ir *veiksmo teiginių* veiklos taisyklių įvedimą pagal koncepcinėje dalyje pateiktą struktūrizuotų veiklos taisyklių įvedimo metodiką. Sistema turi patikrinti įvestą veiklos taisyklę ir nustatyti jos tipą. Po to veiklos taisyklė turi būti skaidoma į logiškai susijusias dalis, kurias galima saugoti atskirai.
- Realizuoti veiklos taisyklių saugyklą pagal koncepcinėje dalyje pateiktą struktūrą. Realizuota saugykla turi tenkinti veiklos taisyklių saugyklos apibrėžtus reikalavimus, iš kurių pagrindinis: galima atkurti kiekvienos į veiklos taisyklių saugyklą įvestos veiklos taisyklės pradinę formą.

Realizuota sistema bus tikrinama atliekant veiklos taisyklių išsaugojimo eksperimentą, kuris turi parodyti, kad sistema sukurta pagal veiklos taisyklių koncepciją bei jos papildymus gali įgyvendinti užsibrėžtus tikslus.

Sistemos architektūra pristatoma įvairiais modeliais: panaudojimo atvejų modelis, loginis modelis, paskirstymo modelis ir komponentų modelis. Šie modeliai yra pateikti Rose modeliu, aprašytu UML, taikant RUP projektavimo metodą.

Panaudojimo atvejų modelyje pateikta bendra sistemos panaudojimo atvejų diagrama, taip pat kiekvienam panaudojimo atvejui pateikiamos veiklos diagramos, kurios aprašo panaudojimo atvejo žingsnius.

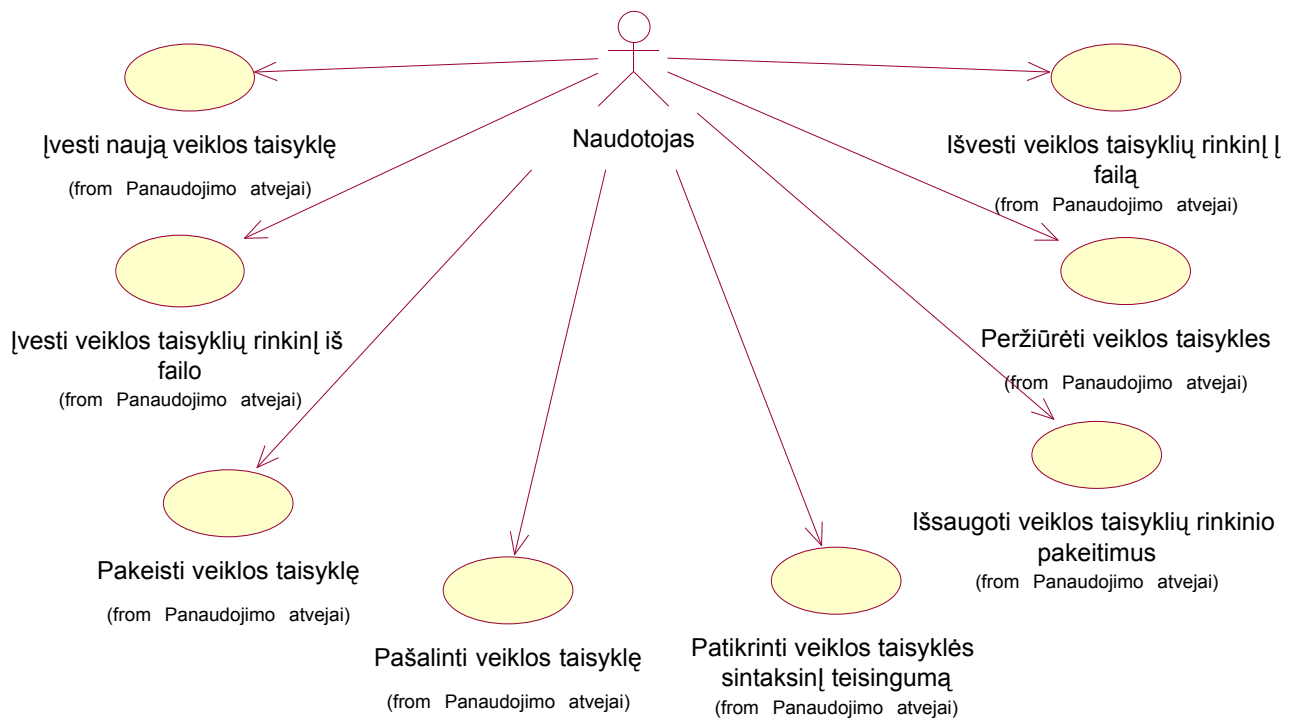
Loginiame modelyje pateiktos analizės klasių diagramos, kurios gaunamos analizuojant panaudojimo atvejus, sistemos klasių diagramas (sistemos struktūrą), kiekvieno paketo įvairių pjūvių klasių diagramas (trasų, bendros projekto, sąsajos navigavimo plano, detalios), panaudojimo atveju bendradarbiavimo diagramas, panaudojimo atvejų realizacijos ir sekų diagramas, sistemos būsenų diagramą.

Taip pat pristatyti sistemos paskirstymo ir komponentų modeliai.

4.2 Panaudojimo atvejų modelis

Veiklos taisyklių metodologijos taikymo IS projektavimo metu programinės įrangos panaudojimo atvejai yra pateikti 4.2.1 pav. Panaudojimo atvejai yra detalizuojami žemiau.

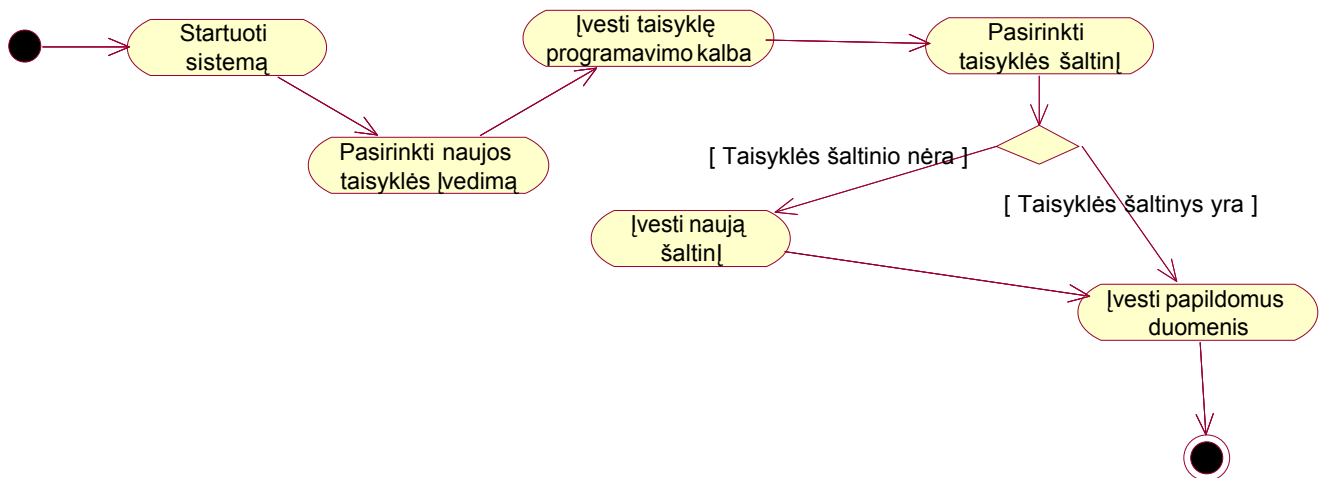
4.2.1 Panaudojimo atvejai



4.2.1 pav. Sistemos panaudojimo atvejų diagrama

[vesti naują veiklos taisyklę

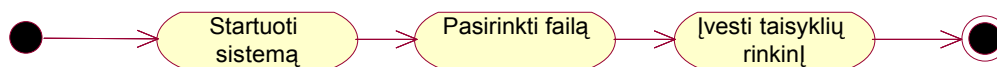
Šis panaudojimo atvejis (4.2.2 pav.) leidžia įvesti į atmintį veiklos taisyklę iš klaviatūros, pasinaudojant specialia veiklos taisyklių programavimo kalba. Taip pat kiekvienai taisyklei parenkamas taisyklės kilmės šaltinis bei papildomi duomenys (aprašas ir pan.). Panaudojimo atvejo aktorius – *naudotojas*.



4.2.2 pav. Naujos veiklos taisyklės įvedimo veiklos diagrama

Įvesti veiklos taisyklių rinkinį iš failo

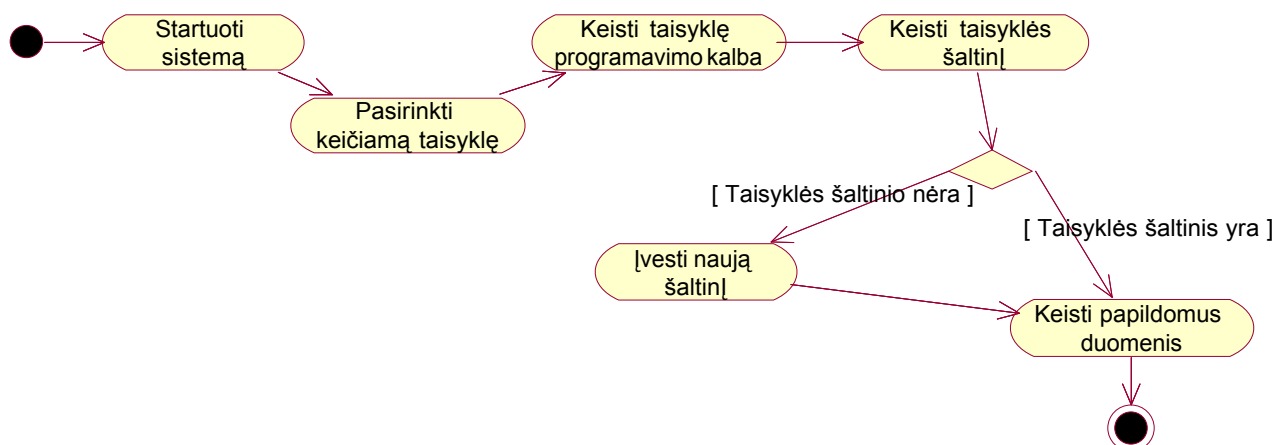
Šis panaudojimo atvejis (4.2.3 pav.) leidžia įvesti į atmintį veiklos taisyklių rinkinį iš pasirinkto failo. Panaudojimo atvejo aktorius – *naudotojas*.



4.2.3 pav. Veiklos taisyklių rinkinio įvedimo iš failo veiklos diagrama

Pakeisti veiklos taisyklę

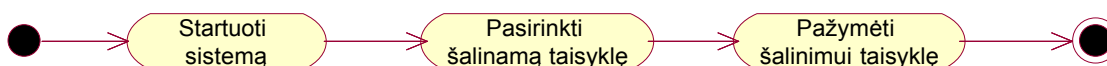
Šis panaudojimo atvejis (4.2.4 pav.) leidžia redaguoti į atmintį įvestą veiklos taisyklę, pasinaudojant specialia veiklos taisyklių programavimo kalba. Taip pat kiekvienai taisyklei gali būti pakeičiamas taisyklės kilmės šaltinis bei papildomi duomenys (aprašas ir pan.). Panaudojimo atvejo aktorius – *naudotojas*.



4.2.4 pav. Veiklos taisyklės pakeitimo veiklos diagrama

Pašalinti veiklos taisyklę

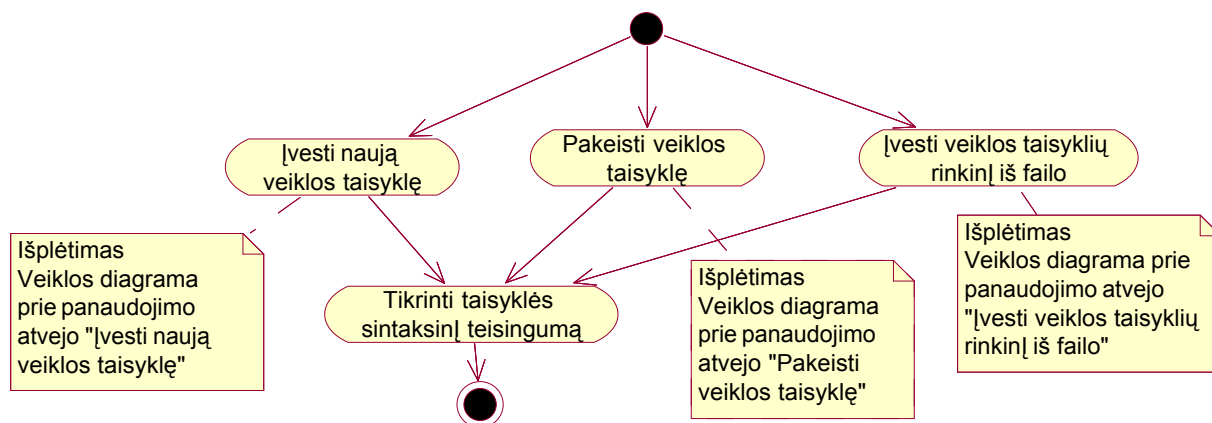
Šis panaudojimo atvejis (4.2.5 pav.) leidžia pažymėti numatomą šalinti veiklos taisyklę iš veiklos taisyklių saugyklos. Panaudojimo atvejo aktorius – *naudotojas*.



4.2.5 pav. Veiklos taisyklės pašalinimo veiklos diagrama

Patikrinti veiklos taisyklės teisingumą

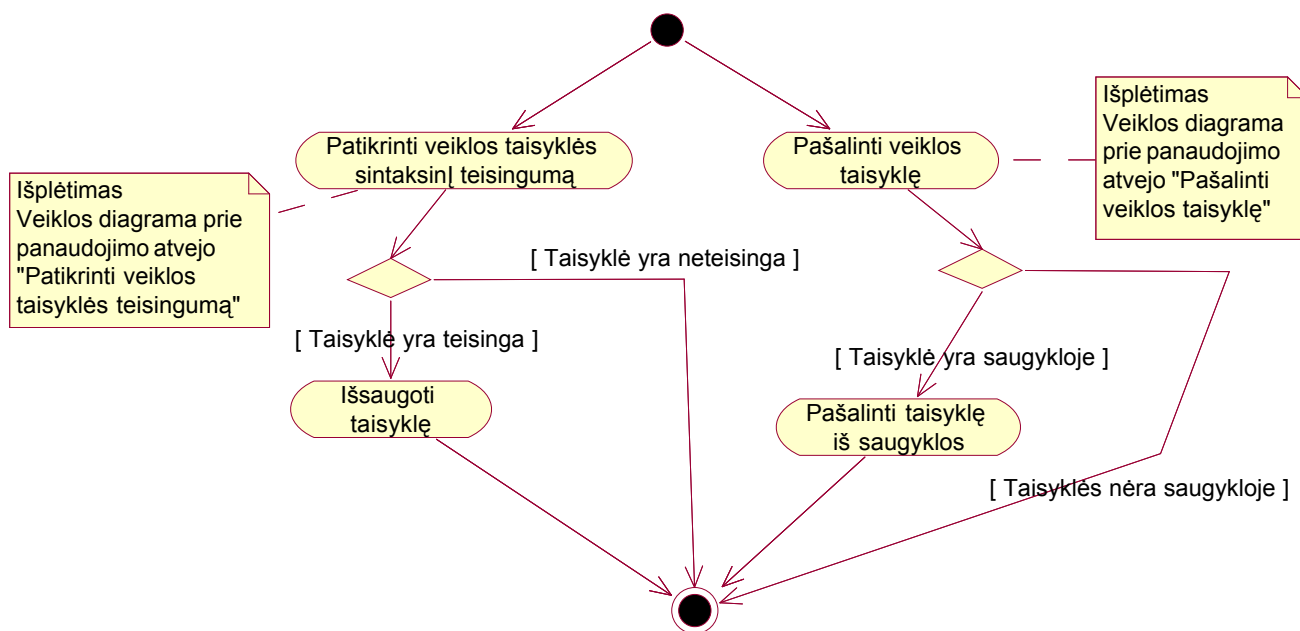
Šis panaudojimo atvejis (4.2.6 pav.) leidžia patikrinti teisingumą naujai įvestai veiklos taisyklei, redaguotai veiklos taisyklei arba veiklos taisyklėms įvedamoms iš failo. Panaudojimo atvejo aktorius – *naudotojas*.



4.2.6 pav. Veiklos taisyklės teisingumo įvertinimo veiklos diagrama

Išsaugoti veiklos taisyklių rinkinio pakeitimus

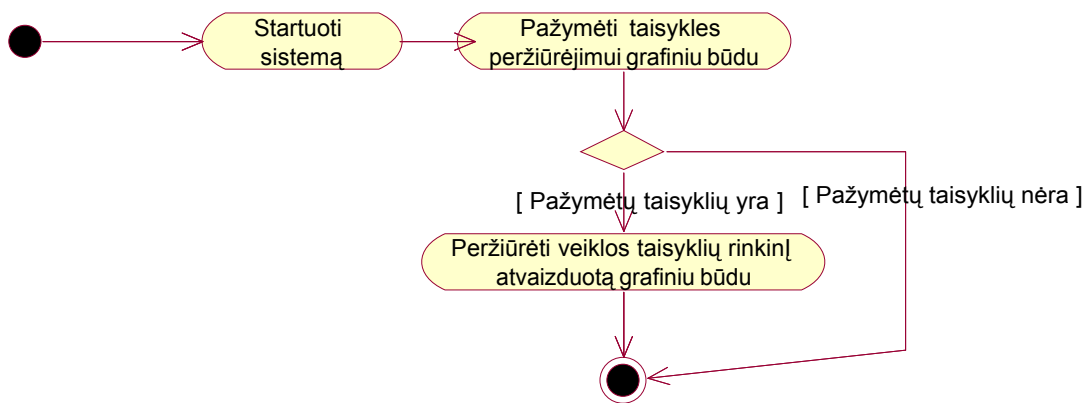
Šis panaudojimo atvejis (4.2.7 pav.) leidžia įrašyti naujas veiklos taisykles į saugyklą, pakeisti redaguotas bei pašalinti pažymėtas šalinimui. Panaudojimo atvejo aktorius – *naudotojas*.



4.2.7 pav. Veiklos taisyklių rinkinio išsaugojimo veiklos diagrama

Peržiūrėti veiklos taisykles

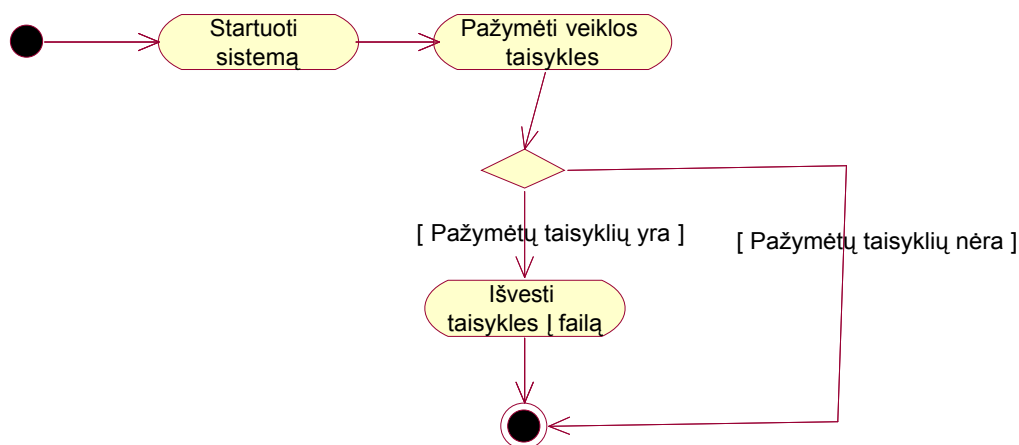
Šis panaudojimo atvejis (4.2.8 pav.) leidžia peržiūrėti pažymėtas veiklos taisykles, atvaizduotas grafiniu būdu. Panaudojimo atvejo aktorius – *naudotojas*.



4.2.8 pav. Veiklos taisyklių peržiūrėjimo veiklos diagrama

Išvesti veiklos taisyklių rinkinį į failą

Šis panaudojimo atvejis (4.2.9 pav.) leidžia išvesti į pasirinktą failą pažymėtas veiklos taisykles. Panaudojimo atvejo aktorius – *naudotojas*.



4.2.9 pav. Veiklos taisyklių rinkinio į failą išvedimo veiklos diagrama

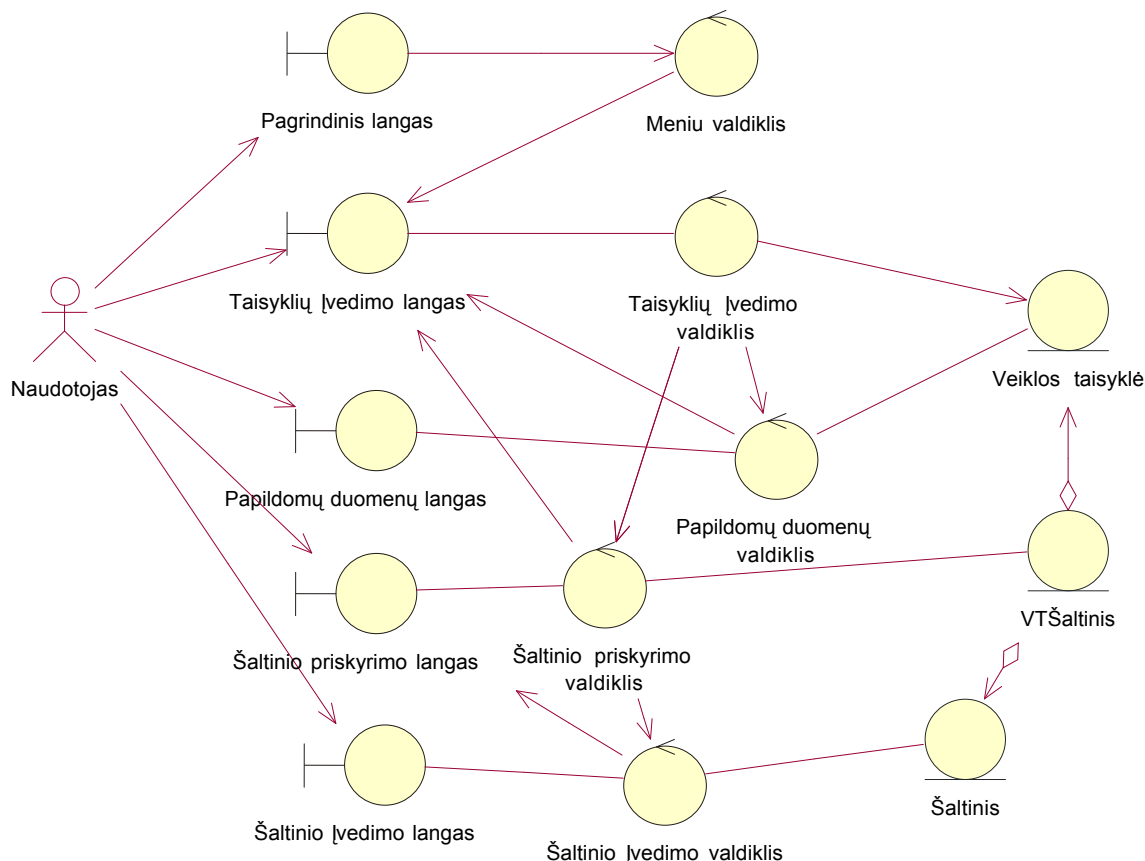
4.3 Loginis modelis

4.3.1 Analizės klasių diagramos

Naujos taisyklės įvedimas

Naudotojas pirmiausia prisijungia (4.3.1 pav.) prie pagrindinio lango (ribinė klasė). Pagrindiniame lange naudotojas pasirenka meniu punktą, atitinkantį panaudojimo atvejį „Įvesti naują veiklos taisyklę“. Tam tikslui sistema turi atidaryti taisyklių įvedimo langą (ribinė klasė). Šį langą atidaro meniu valdiklis. Taisyklių įvedimo valdiklis kontroliuoja veiklos taisyklių įvedimą į sistemą. Įvedęs veiklos taisyklę naudotojas grįžta į taisyklės įvedimo langą. Naudotojas pasirenka šaltinio langą. Šaltinio valdiklis leidžia priskirti jau įvestus taisyklės kilmės šaltinius. Tačiau esant reikalui

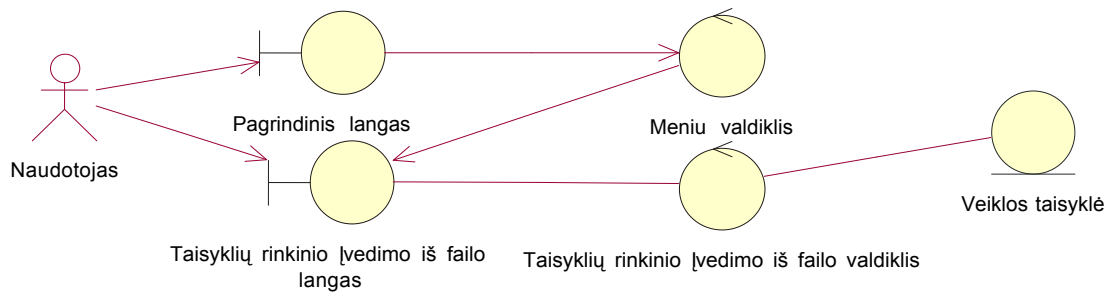
galima iškviešti šaltinio įvedimo langą, kuriame galima įvesti naują taisyklės kilmės šaltinį. Įvedęs taisyklės kilmės šaltinius naudotojas grįžta į taisyklės įvedimo langą. Naudotojas pasirenka papildomų duomenų įvedimą. Sistema atidaro papildomų duomenų langą, kur naudotojas įveda papildomus duomenis.



4.3.1 pav. Naujos taisyklės įvedimo analizės klasių diagrama

Veiklos taisyklių rinkinio įvedimas iš failo

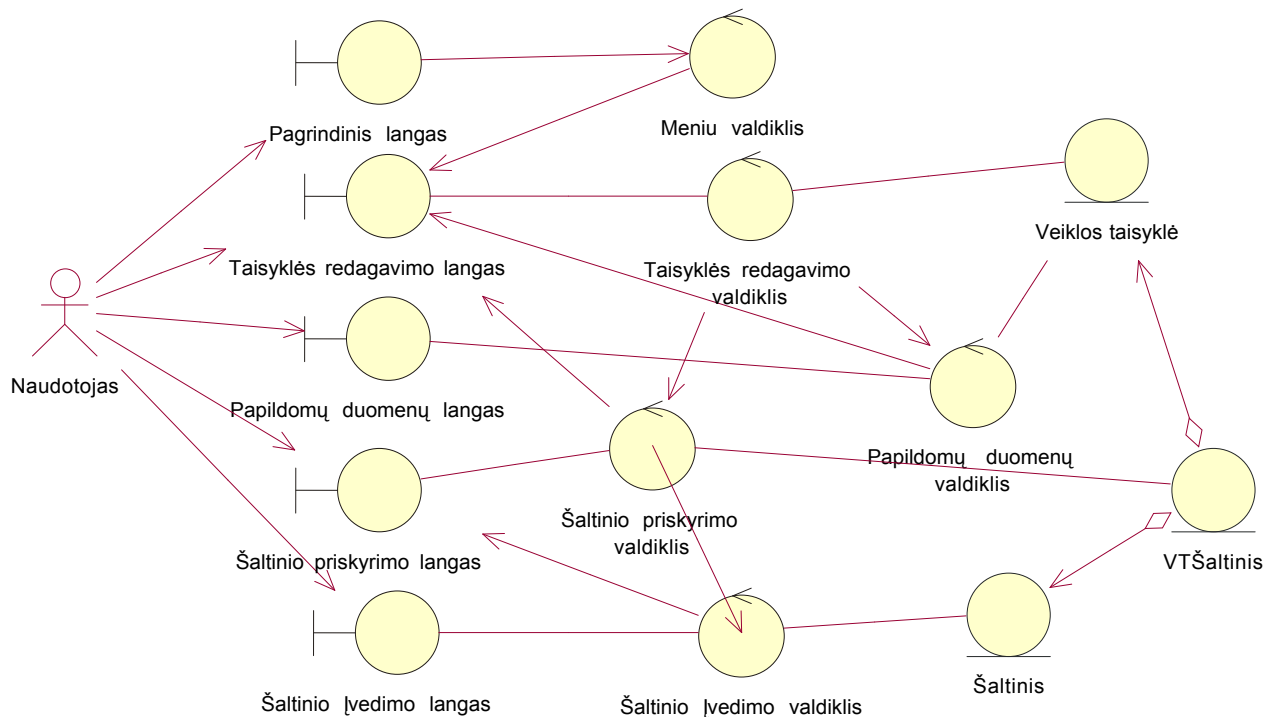
Naudotojas pirmiausia prisijungia (4.3.2 pav.) prie pagrindinio lango (ribinė klasė). Pagrindiniame lange naudotojas pasirenka meniu punktą, atitinkantį panaudojimo atvejį „Įvesti veiklos taisyklių rinkinį iš failo“. Tam tikslui sistema turi atidaryti taisyklių rinkinio įvedimo iš failo langą (ribinė klasė). Šį langą atidaro meniu valdiklis. Naudotojas pasirenka failą, iš kurio įvesti veiklos taisyklės, kurias įveda taisyklių rinkinio įvedimo iš failo valdiklis. Taisyklės yra išvedamos į taisyklių rinkinio įvedimo iš failo langą.



4.3.2 pav. Veiklos taisyklių rinkinio įvedimo iš failo analizės klasių diagrama

Veiklos taisyklės pakeitimas

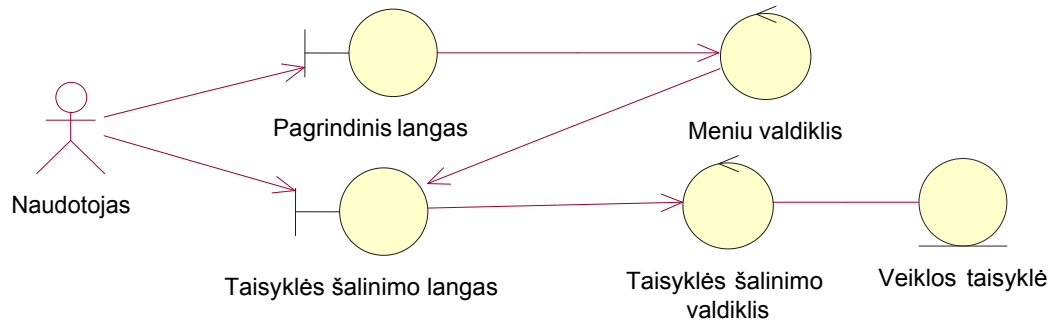
Naudotojas pirmiausia prisijungia (4.3.3 pav.) prie pagrindinio lango (ribinė klasė). Pagrindiniame lange naudotojas pasirenka meniu punktą, atitinkantį panaudojimo atvejį „Pakeisti veiklos taisyklę“. Tam tikslui sistema turi atidaryti taisyklės redagavimo langą (ribinė klasė). Šį langą atidaro meniu valdiklis. Taisyklės redagavimo valdiklis kontroliuoja veiklos taisyklių keitimą. Pakeitęs veiklos taisyklę naudotojas grįžta į taisyklės redagavimo langą. Naudotojas pasirenka šaltinio langą. Šaltinio valdiklis leidžia priskirti jau įvestus taisyklės kilmės šaltinius. Tačiau esant reikalui galima iškviešti šaltinio įvedimo langą, kuriame galima įvesti naują taisyklės kilmės šaltinį. Pakeitęs taisyklės kilmės šaltinius naudotojas grįžta į taisyklės redagavimo langą. Naudotojas pasirenka papildomų duomenų keitimą. Sistema atidaro papildomų duomenų langą, kur naudotojas pakeičia papildomus duomenis.



4.3.3 pav. Veiklos taisyklės pakeitimo analizės klasių diagrama

Veiklos taisyklės pašalinimas

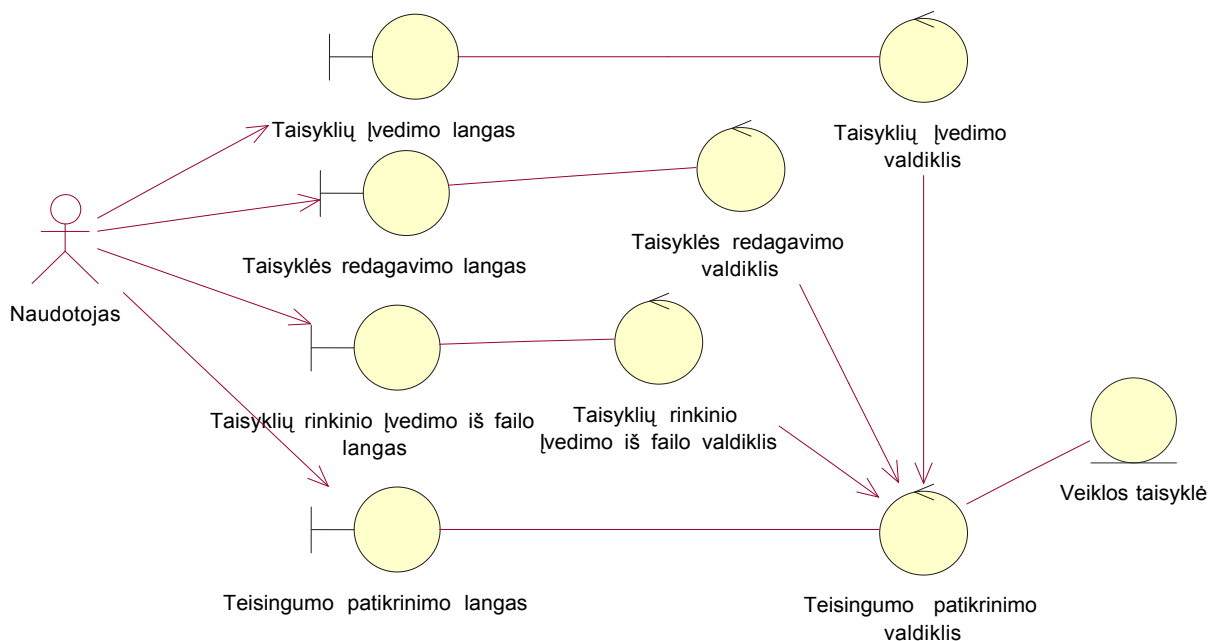
Naudotojas pirmiausia prisijungia (4.3.4 pav.) prie pagrindinio lango (ribinė klasė). Pagrindiniame lange naudotojas pažymi šalinamą taisyklę, tada pasirenka meniu punktą, atitinkantį panaudojimo atvejį „Pašalinti veiklos taisyklę“. Tam tikslui sistema turi atidaryti taisyklės šalinimo langą (ribinė klasė). Šį langą atidaro meniu valdiklis. Naudotojas patvirtina savo apsisprendimą pašalinti taisyklę, ir veiklos taisyklė pažymima pašalinimui.



4.3.4 pav. Veiklos taisyklės pašalinimo analizės klasių diagrama

Veiklos taisyklės teisingumo patikrinimas

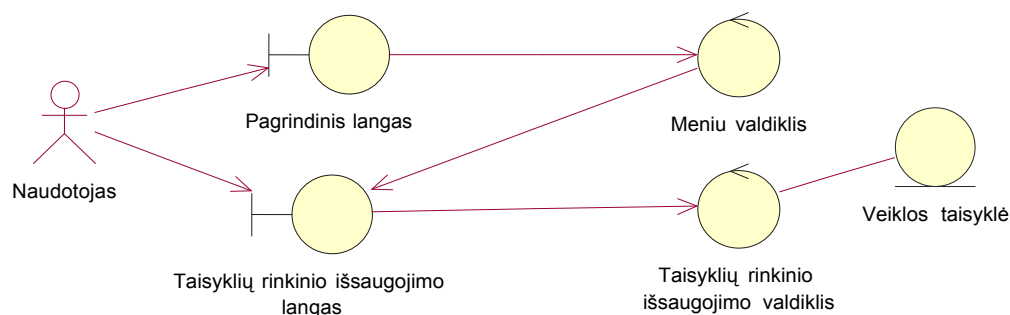
Naudotojas pirmiausia įveda naują taisyklę (4.3.5 pav.), pakeičia jau esamą sistemoje taisyklę arba įveda taisyklių rinkinį iš failo. Jis yra prisijungęs prie taisyklių įvedimo, taisyklės redagavimo ar taisyklių rinkinio įvedimo iš failo lango (ribinės klasės). Naudotojas pasirenka meniu punktą atitinkantį panaudojimo atvejį „Patikrinti veiklos taisyklės teisingumą“. Sistema patikrina atidarytą taisyklę ir išveda patikrinimo rezultatą į teisingumo patikrinimo langą.



4.3.5 pav. Veiklos taisyklės teisingumo patikrinimo analizės klasių diagrama

Veiklos taisyklių rinkinio išsaugojimas

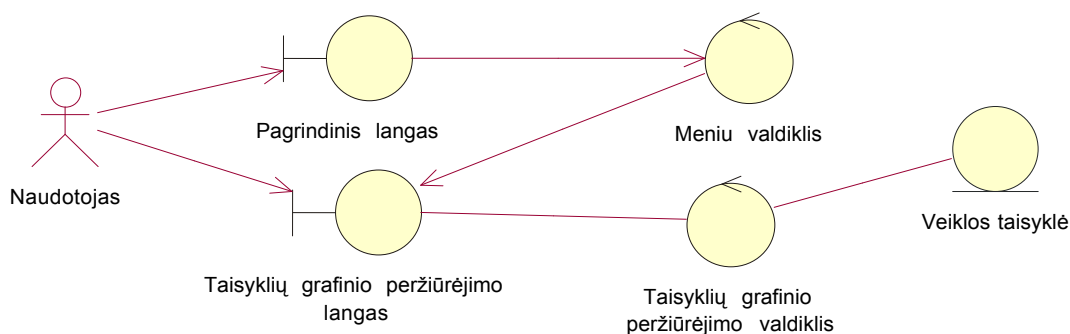
Naudotojas pirmiausia prisijungia (4.3.6 pav.) prie pagrindinio lango (ribinė klasė). Pagrindiniame lange naudotojas pasirenka meniu punktą, atitinkantį panaudojimo atvejį „Išsaugoti veiklos taisyklių rinkinio pasikeitimus“. Tam tikslui sistema turi atidaryti taisyklių rinkinio išsaugojimo langą (ribinė klasė). Šį langą atidaro meniu valdiklis. Naudotojas patvirtina savo pasirinkimą. Tuomet taisyklių rinkinio išsaugojimo valdiklis į saugyklą įrašo naujas taisykles, pakeičia redaguotas ir pašalina iš saugyklos pažymėtas šalinimui taisykles



4.3.6 pav. Veiklos taisyklių rinkinio išsaugojimo analizės klasių diagrama

Veiklos taisyklių peržiūrėjimas

Naudotojas pirmiausia prisijungia (4.3.7 pav.) prie pagrindinio lango (ribinė klasė). Pagrindiniame lange naudotojas pažymi peržiūrimas taisykles, tada pasirenka meniu punktą, atitinkantį panaudojimo atvejį „Peržiūrėti veiklos taisykles“. Tam tikslui sistema turi atidaryti taisyklės grafinio peržiūrėjimo langą (ribinė klasė). Šį langą atidaro meniu valdiklis. Taisyklių grafinio peržiūrėjimo valdiklis išveda peržiūrimas taisykles į taisyklių grafinio peržiūrėjimo langą.

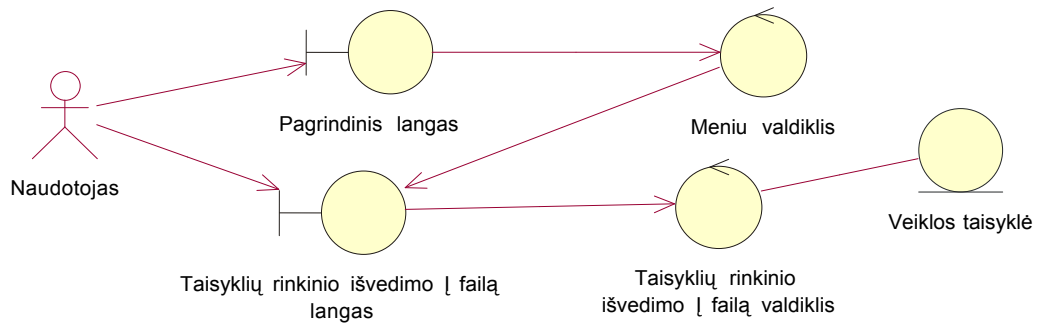


4.3.7 pav. Veiklos taisyklių peržiūrėjimo analizės klasių diagrama

Veiklos taisyklių rinkinio išvedimas į failą

Naudotojas pirmiausia prisijungia (4.3.8 pav.) prie pagrindinio lango (ribinė klasė). Pagrindiniame lange naudotojas pasirenka meniu punktą, atitinkantį panaudojimo atvejį „Išvesti veiklos taisyklių rinkinį į failą“. Tam tikslui sistema turi atidaryti taisyklių rinkinio išvedimo iš failą langą (ribinė klasė). Šį langą atidaro meniu valdiklis. Naudotojas pasirenka failą, į kurį išvesti veiklos

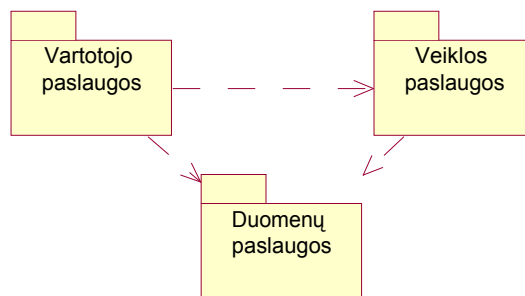
taisykles, kurias išveda taisyklių rinkinio išvedimo į failą valdiklis. Taisyklės yra išvedamos į naudotojo pasirinktą failą.



4.3.8 pav. Veiklos taisyklių rinkinio išvedimo į failą analizės klasių diagrama

4.3.2 Projekto modelio paketai

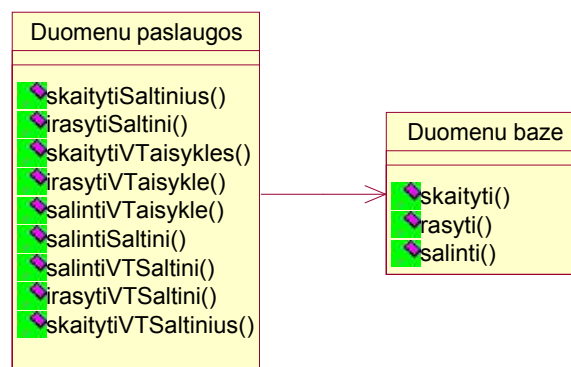
Parentant sistemos architektūrą, taikomi tipiniai architektūros modeliai (šablonai). Pasirenkamas tipinis trijų lygių architektūros modelis (4.3.9 pav.). Remiantis šiuo modeliu, sistema sudarys vartotojo, veiklos ir duomenų paslaugas, sugrupuotos į atitinkamus paketus.



4.3.9 pav. Trijų lygių sistemos klasių diagrama

Duomenų paslaugos

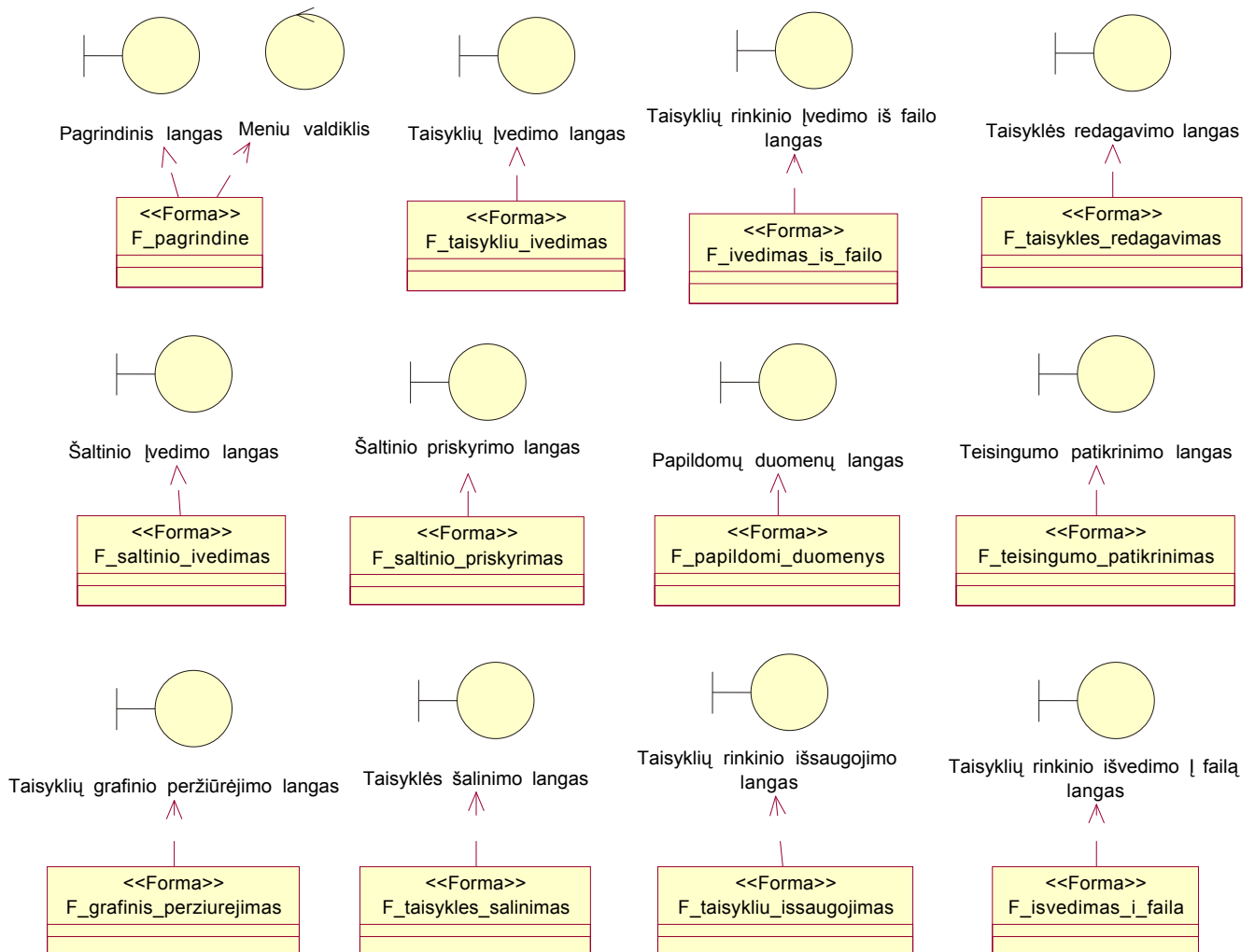
Sąveikos su duomenų baze operacijas paprastai realizuoja vienas ar keli moduliai, kurie atlieka prisijungimo, užklausų formavimo, duomenų skaitymo - rašymo operacijas. Projektuojamoje sistemoje visas šias operacijas realizuos viena tam skirta klasė “DB paslaugos” (4.3.10 pav.).



4.3.10 pav. Duomenų paslaugų klasių diagrama

Vartotojo paslaugos

Vartotojo paslaugos atitiks ribines klases, kurios buvo identifikuotos, remiantis panaudojimo atvejų specifikacijų analize bei vartotojo interfeiso eskizais (menu valdiklis, nors tai ir atskiras objekto tipas, sujungiamas kartu su pagrindiniu langas). Jos pateikiamos 4.3.11 pav.



4.3.11 pav. Vartotojo paslaugų klasių trasų diagrama

F_pagrindine – pagrindinė forma, iš kurios iškviečiamos kitos formos, realizuojančios tam tikrus panaudojimo atvejus.

F_taisykliu_ivedimas – forma, kuri realizuoja sąsają naujai taisyklei įvesti, taip pat papildomiems duomenims ir taisyklės kilmės šaltiniui įvesti.

F_ivedimas_is_failo – forma, kuri realizuoja sąsają taisyklių rinkiniui įvesti iš failo.

F_taisykles_redagavimas – forma, kuri realizuoja sąsają taisyklei keisti, taip pat papildomiems duomenims ir taisyklės kilmės šaltiniui keisti.

F_saltinio_ivedimas – forma, kuri realizuoja sąsają naujam taisyklės kilmės šaltiniui įvesti.

F_saltinio_priskyrimas – forma, kuri realizuoja sąsają taisyklės kilmės šaltiniui priskirti taisyklei.

F_papildomi_duomenys – forma, kuri realizuoja sąsają taisyklės papildomiems duomenims įvesti ir pakeisti.

F_teisingumo_patikrinimas – forma, kuri realizuoja sąsają taisyklių teisingumui patikrinti.

F_grafinis_perziurejimas – forma, kuri realizuoja sąsają taisyklių rinkiniui peržiūrėti grafiniu būdu.

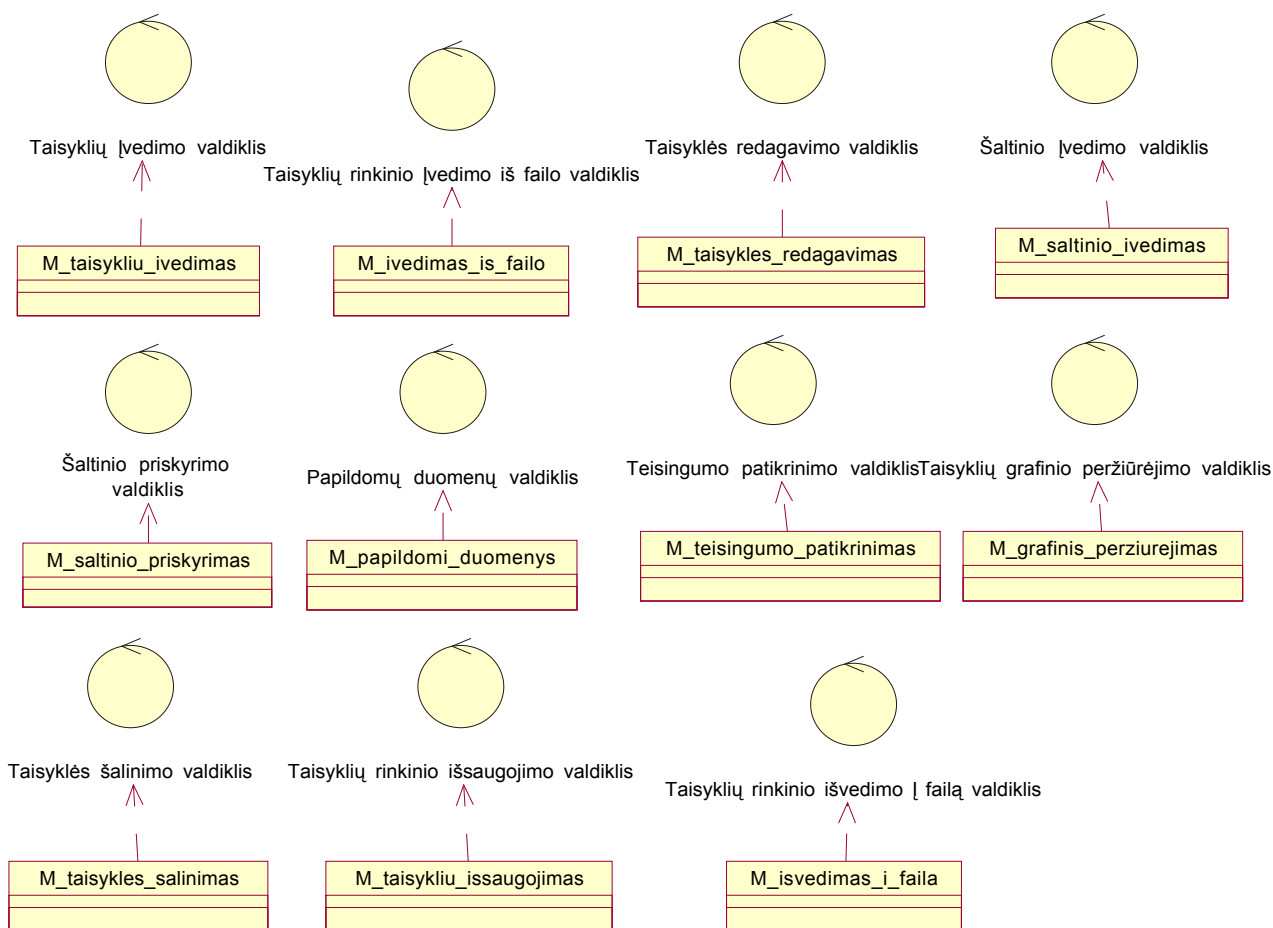
F_taisykles_salinimas – forma, kuri realizuoja sąsają taisyklei pažymėti šalintina.

F_taisykliu_issaugojimas – forma, kuri realizuoja sąsają naujoms taisyklėms įrašyti, senoms pakeisti saugykloje bei pažymėtų šalinimui taisyklėms pašalinti iš saugyklos.

F_isvedimas_i_faila – forma, kuri realizuoja sąsają taisyklių rinkiniui išvesti į failą.

Veiklos paslaugos

Programos projektas yra tarpinis sluoksnis tarp duomenų bazės ir vartotojo interfeiso. Veiklos paslaugos išvedamos iš analizės etapo valdiklių ir esybių klasių. Jos pateikiamos 4.3.12 pav.



4.3.12 pav. Veiklos paslaugų klasių trasų diagrama

M_taisykliu_ivedimas – valdiklis, kuris realizuoja naujos taisyklės įvedimą, taip pat papildomų duomenų ir taisyklės kilmės šaltinio įvedimą.

M_ivedimas_is_failo – valdiklis, kuris realizuoja taisyklių rinkinio įvedimą iš failo.

M_taisykles_redagavimas – valdiklis, kuris realizuoja taisyklės keitimą, taip pat papildomų duomenų ir taisyklės kilmės šaltinio keitimą.

M_saltinio_ivedimas – valdiklis, kuris realizuoja naujo taisyklės kilmės šaltinio įvedimą.

M_saltinio_priskyrimas – valdiklis, kuris realizuoja taisyklės kilmės šaltinio priskyrimą taisyklei.

M_papildomi_duomenys – valdiklis, kuris realizuoja taisyklės papildomų duomenų įvedimą ir pakeitimą.

M_teisingumo_patikrinimas – valdiklis, kuris realizuoja taisyklių teisingumo patikrinimą.

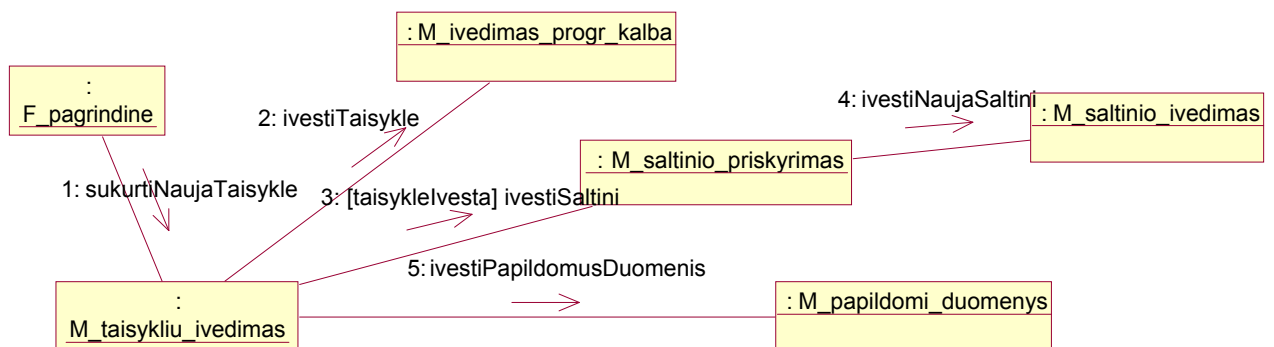
M_grafinis_perziurejimas – valdiklis, kuris realizuoja taisyklių rinkinio peržiūrėjimą grafiniu būdu.

M_taisykles_salinimas – valdiklis, kuris realizuoja taisyklės pažymėjimą šalintina.

M_taisykliu_issaugojimas – valdiklis, kuris realizuoja naujų taisyklių įrašymą, senų pakeitimą saugykloje bei pažymėtų šalinimui taisyklių pašalinimą iš saugyklos.

M_isvedimas_i_faila – valdiklis, kuris realizuoja taisyklių rinkinio išvedimą į failą.

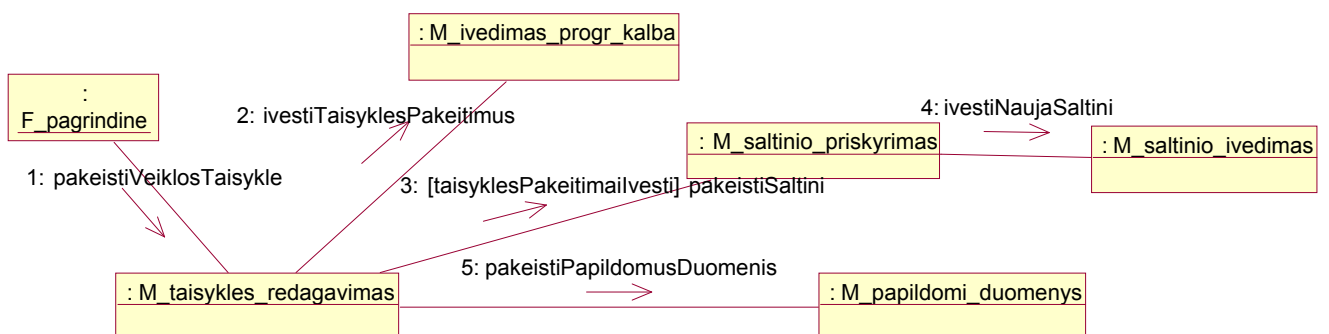
Panaudojimo atvejus realizuojančių formų ir valdiklių bendradarbiavimo diagramos pateikiamos 4.3.13 – 4.3.20 pav.



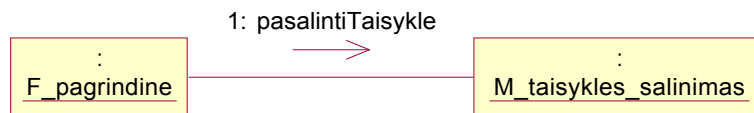
4.3.13 pav. Naujos taisyklės įvedimo bendradarbiavimo diagrama



4.3.14 pav. Veiklos taisyklių rinkinio įvedimo iš failo bendradarbiavimo diagrama



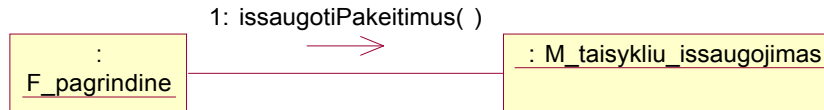
4.3.15 pav. Veiklos taisyklės pakeitimo bendradarbiavimo diagrama



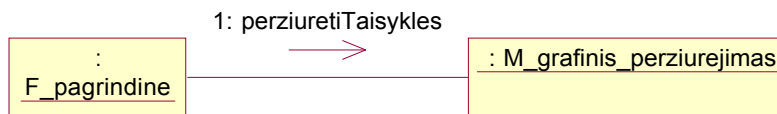
4.3.16 pav. Veiklos taisyklių pašalinimo bendradarbiavimo diagrama



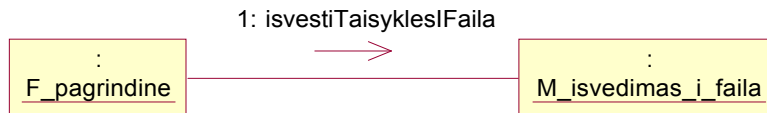
4.3.17 pav. Veiklos taisyklės teisingumo patikrinimo bendradarbiavimo diagrama



4.3.18 pav. Veiklos taisyklių rinkinio pasikeitimų išsaugojimo bendradarbiavimo diagrama



4.3.19 pav. Veiklos taisyklių peržiūrėjimo bendradarbiavimo diagrama

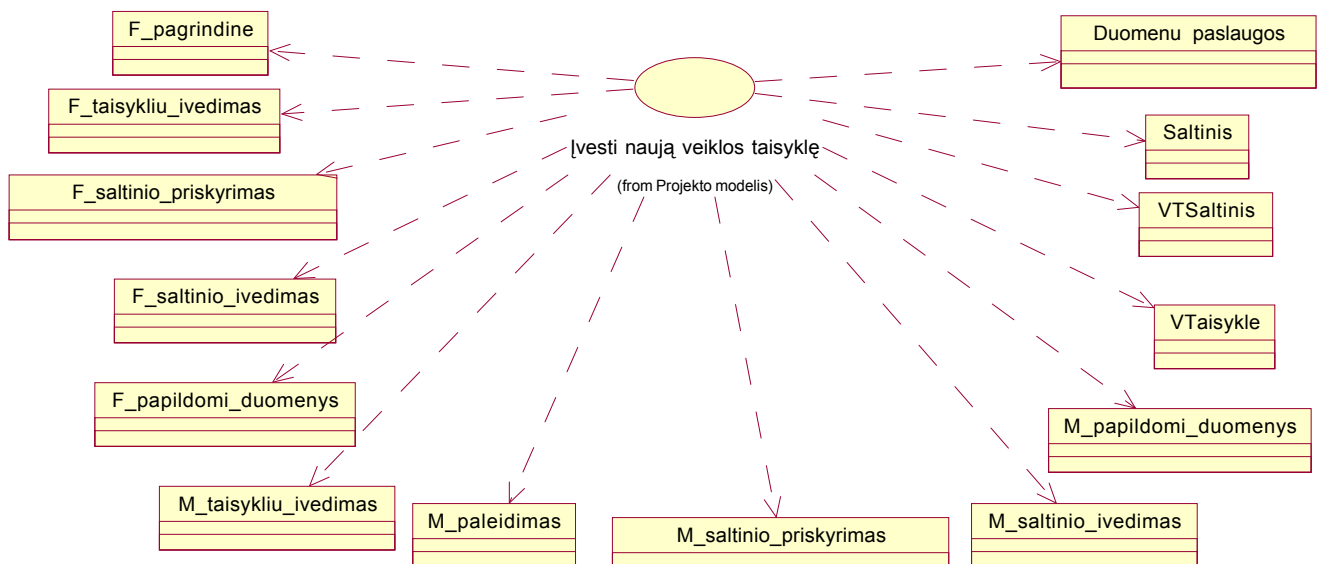


4.3.20 pav. Veiklos taisyklių rinkinio išvedimo į failą bendradarbiavimo diagrama

4.3.3 Panaudojimo atvejų realizacijos

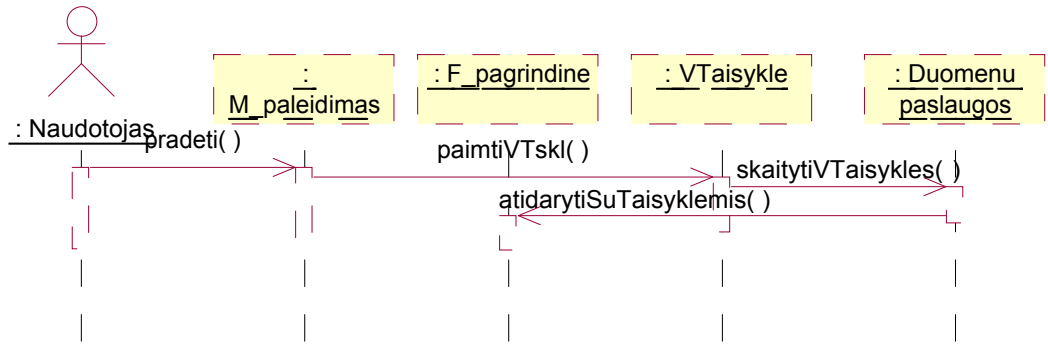
Įvesti naują veiklos taisyklę

Panaudojimo atvejo “Įvesti naują veiklos taisyklę” realizacijos diagrama pateikta 4.3.21 pav.

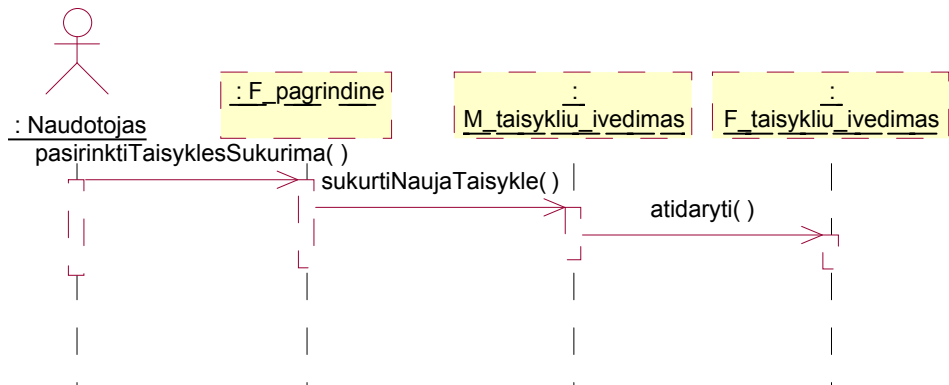


4.3.21 pav. Naujos taisyklės įvedimo panaudojimo atvejo realizacijos diagrama

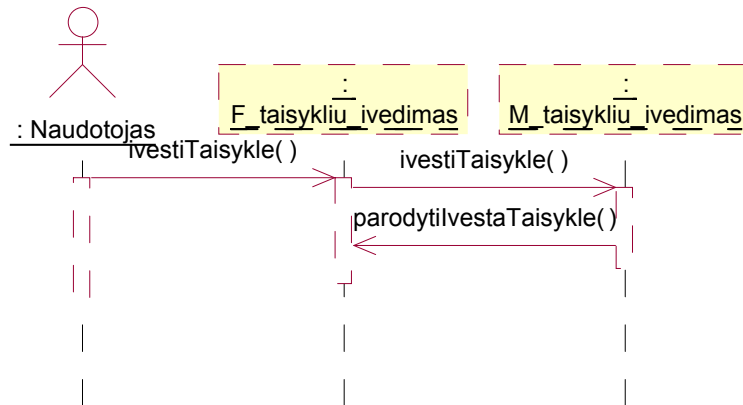
Panaudojimo atvejo “Įvesti naują veiklos taisyklę” sekų diagramos pateiktos 4.3.22 – 4.3.27 pav.



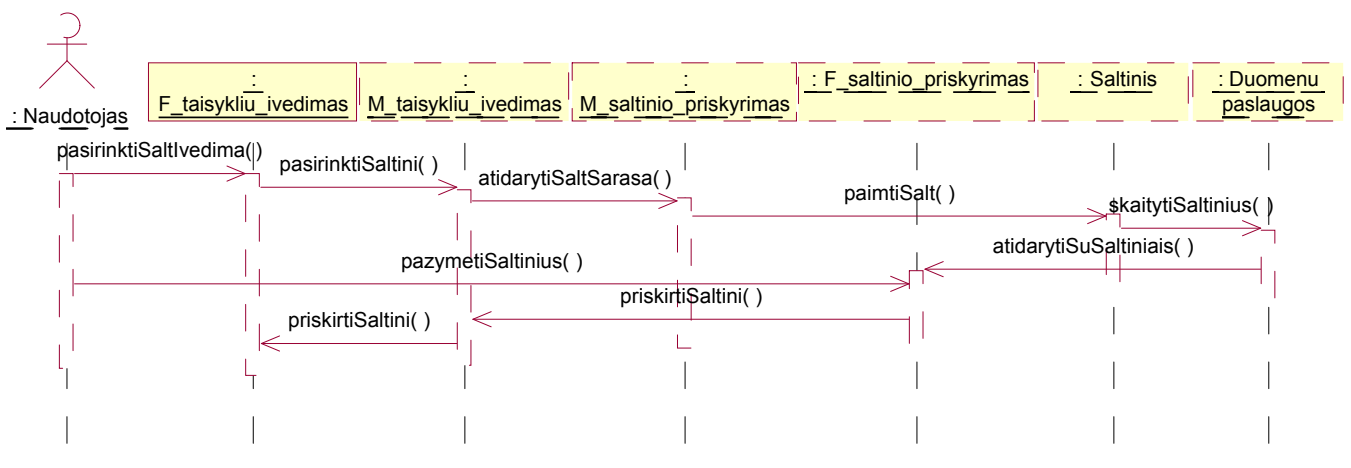
4.3.22 pav. Startuoti sistemą



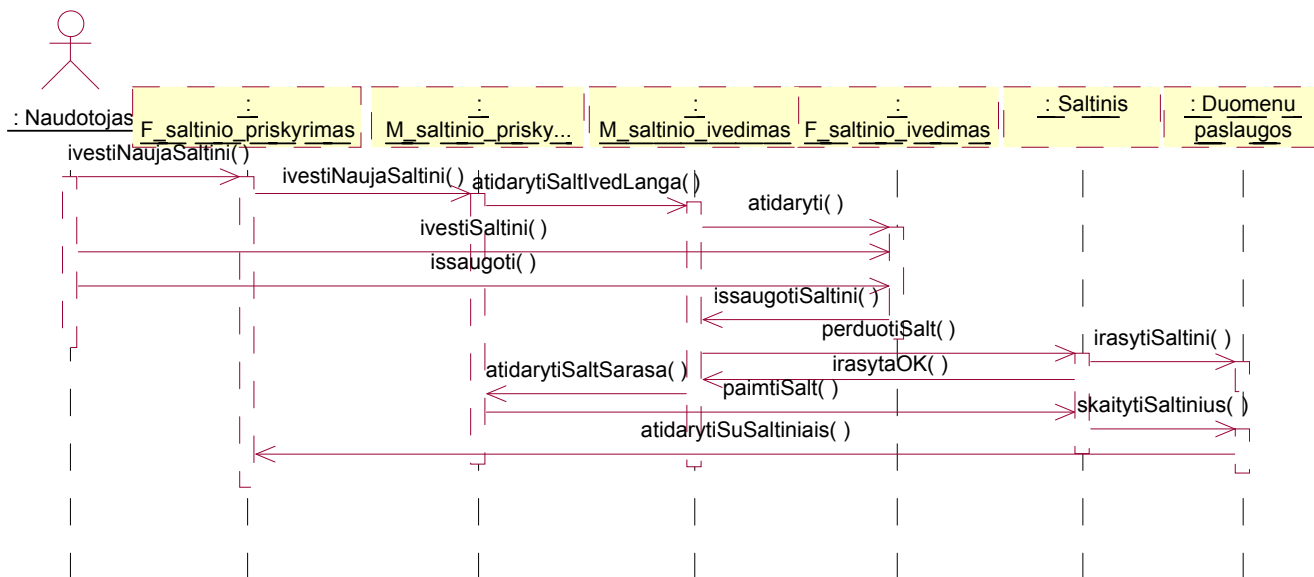
4.3.23 pav. Pasirinkti naujos taisyklės įvedimą



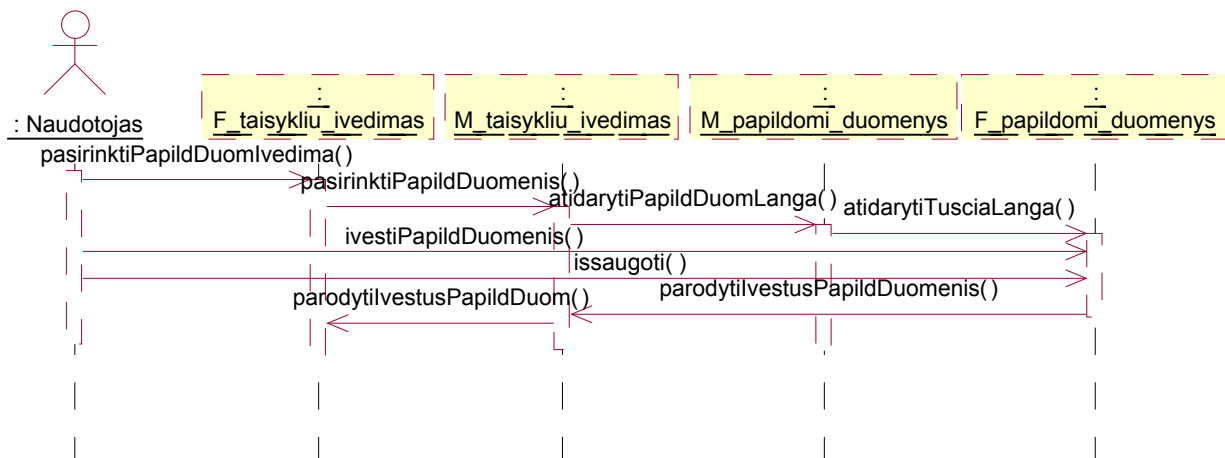
4.3.24 pav. Įvesti taisyklę programavimo kalba



4.3.25 pav. Pasirinkti taisyklės šaltinį



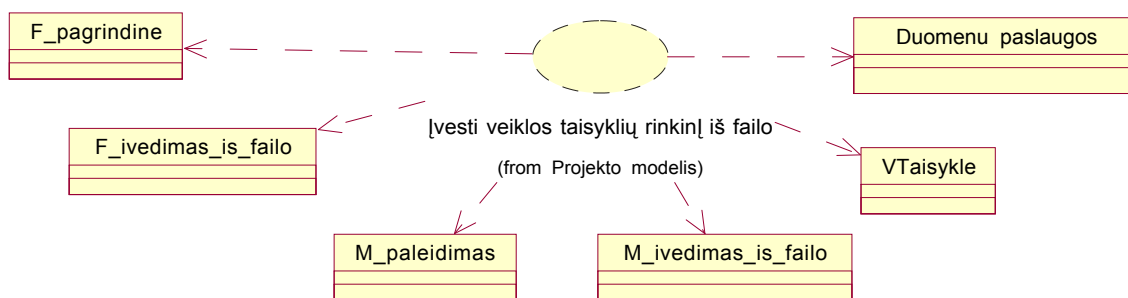
4.3.26 pav. Įvesti naują taisyklės šaltinį



4.3.27 pav. Įvesti papildomus duomenis

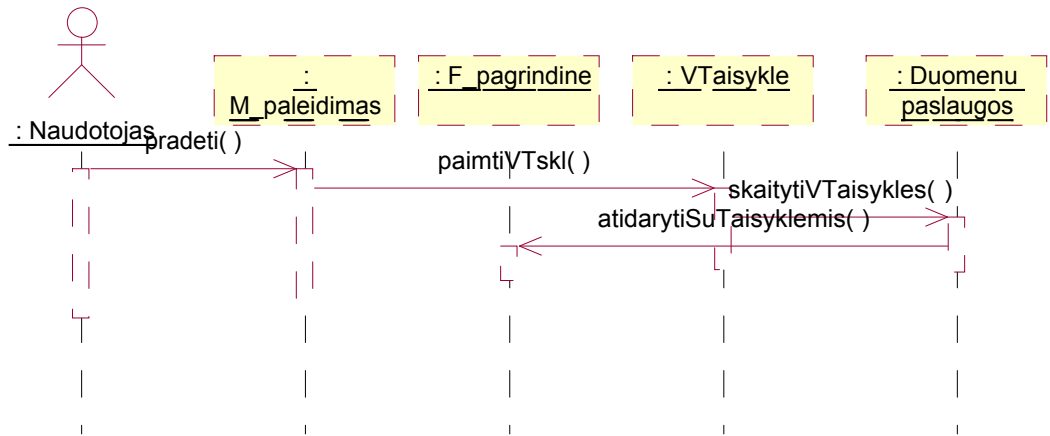
Įvesti veiklos taisyklių rinkinį iš failo

Panaudojimo atvejo “Įvesti veiklos taisyklių rinkinį iš failo” realizacijos diagrama pateikta 4.3.28 pav.

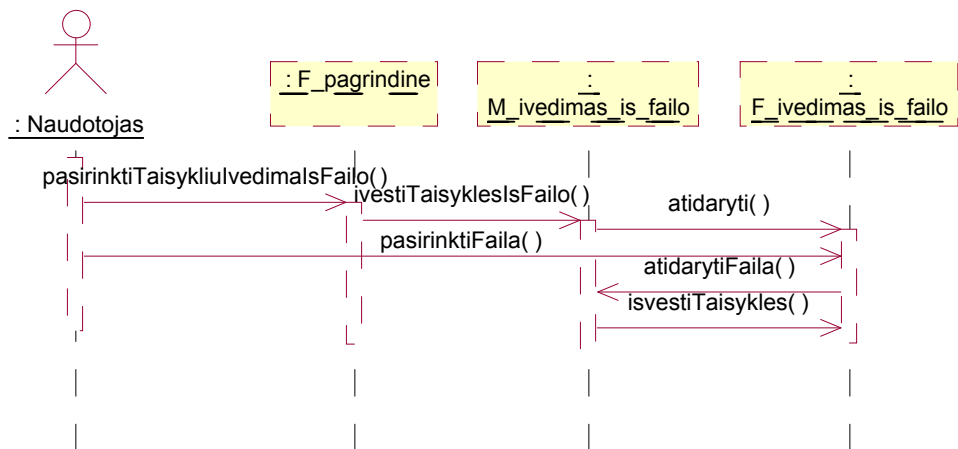


4.3.28 pav. Veiklos taisyklių rinkinio įvedimo iš failo panaudojimo atvejo realizacijos diagrama

Panaudojimo atvejo “Įvesti veiklos taisyklių rinkinį iš failo” sekų diagramos pateiktos 4.3.29 – 4.3.30 pav.



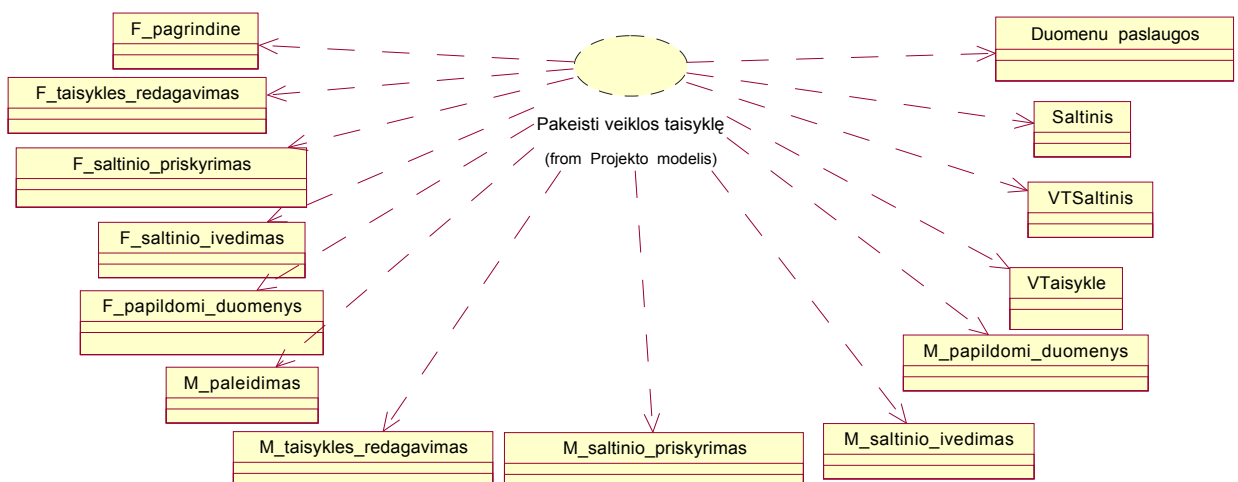
4.3.29 pav. Startuoti sistemą



4.3.30 pav. Įvesti taisyklių rinkinį

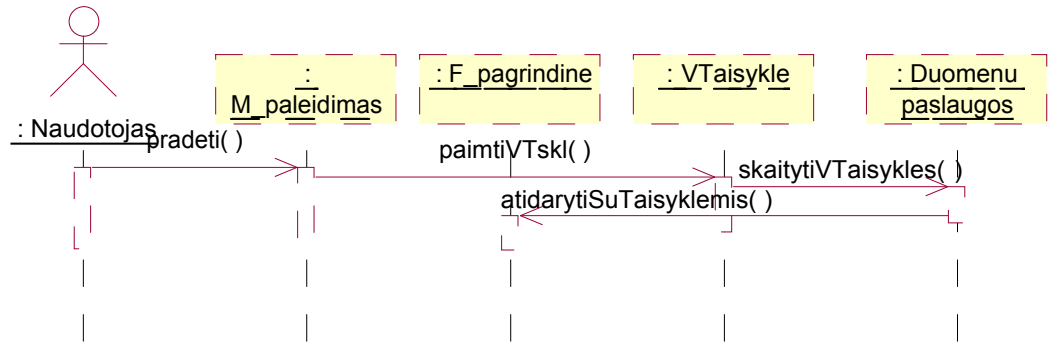
Pakeisti veiklos taisyklę

Panaudojimo atvejo “Pakeisti veiklos taisyklę” realizacijos diagrama pateikta 4.3.31 pav.

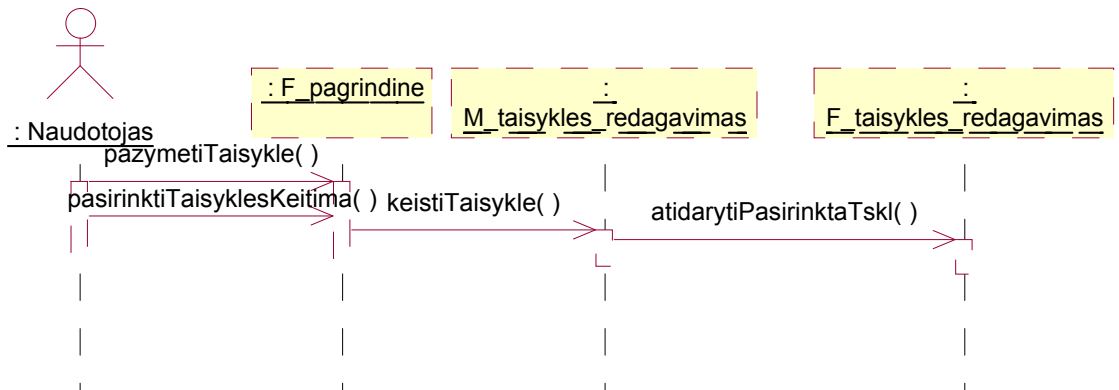


4.3.31 pav. Veiklos taisyklės pakeitimo panaudojimo atvejo realizacijos diagrama

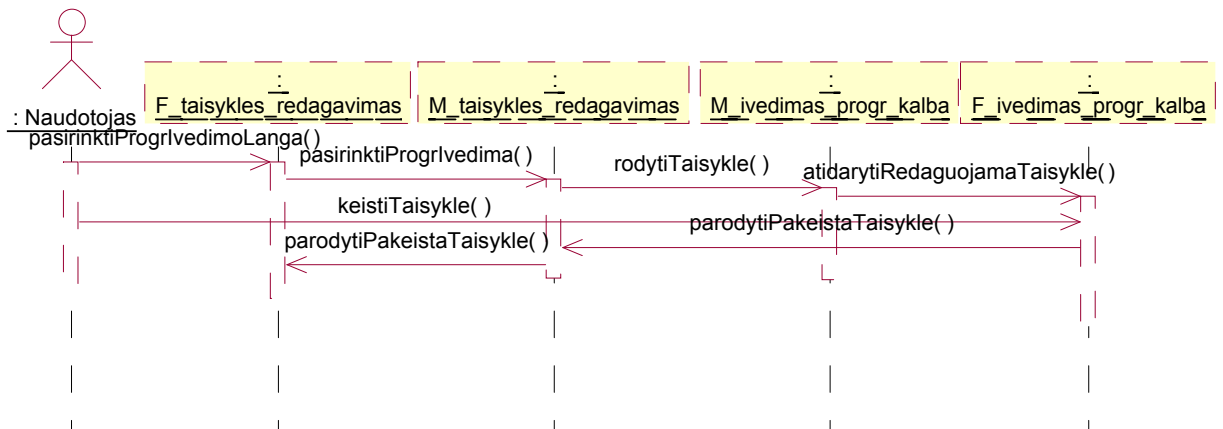
Panaudojimo atvejo “Pakeisti veiklos taisyklę” sekų diagramos pateiktos 4.3.32 – 4.3.37 pav.



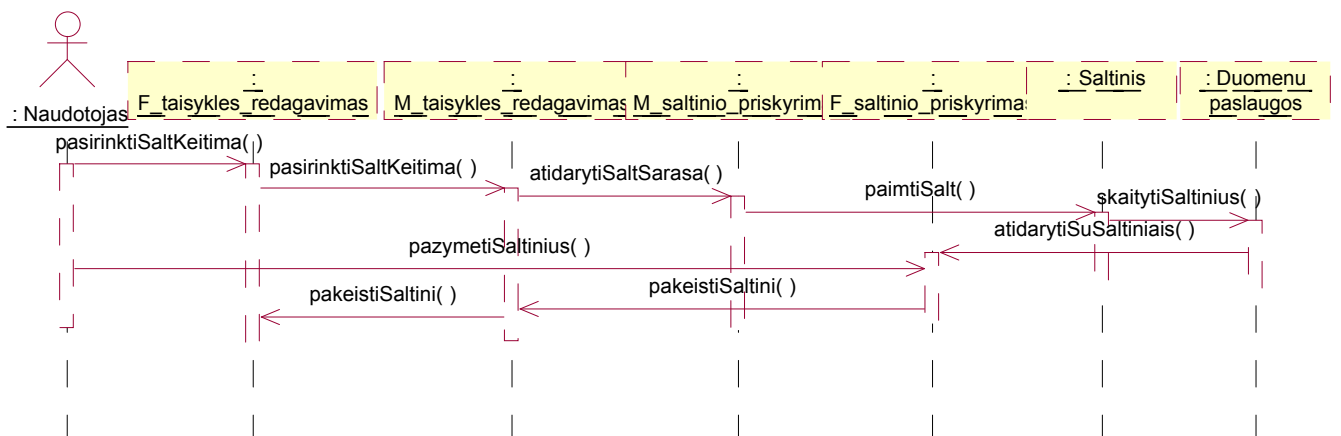
4.3.32 pav. Startuoti sistemą



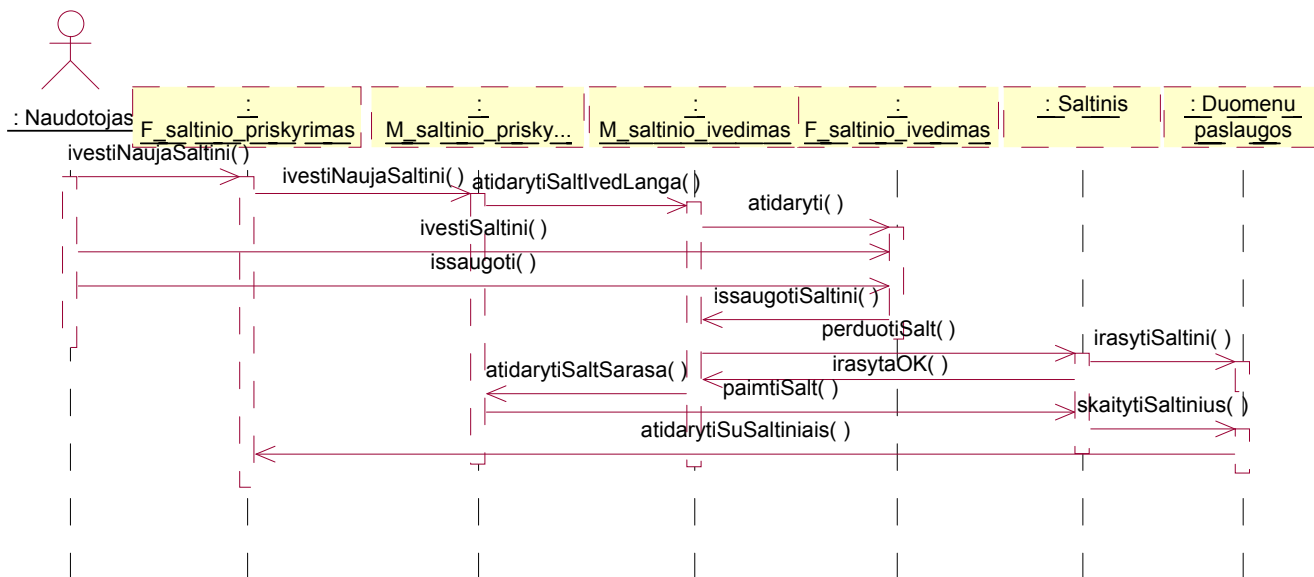
4.3.33 pav. Pasirinkti taisyklės keitimą



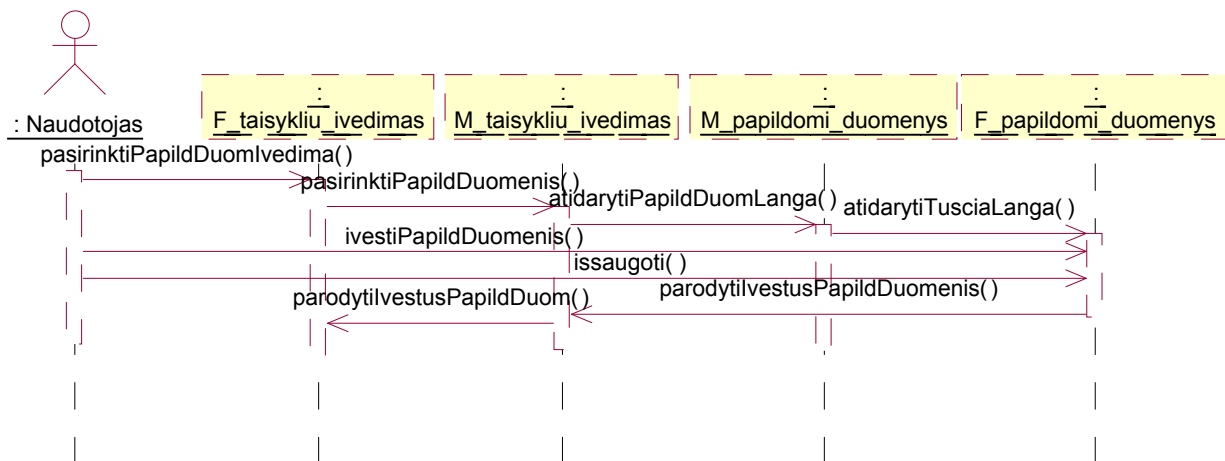
4.3.34 pav. Keisti taisyklę programavimo kalba



4.3.35 pav. Keisti taisyklės šaltinį



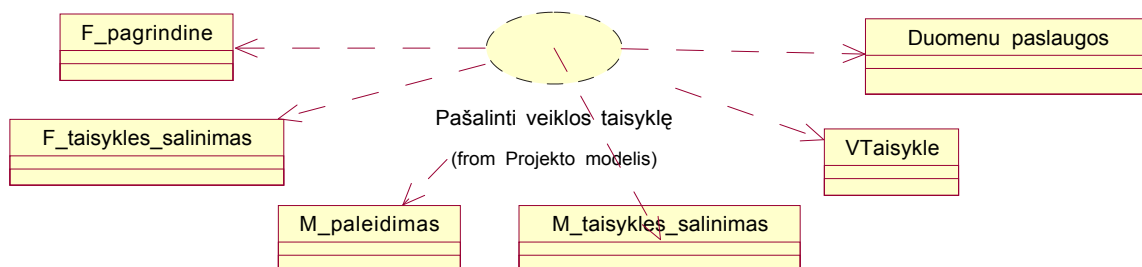
4.3.36 pav. Įvesti naują taisyklės šaltinį



4.3.37 pav. Keisti papildomus duomenis

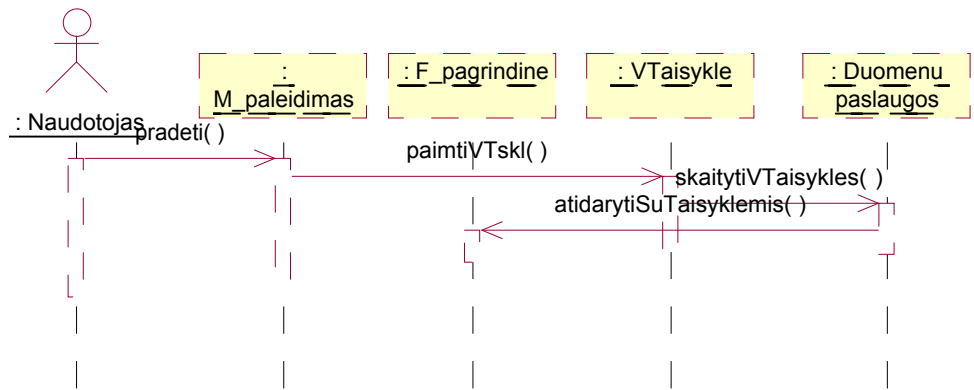
Pašalinti veiklos taisyklę

Panaudojimo atvejo “Pašalinti veiklos taisyklę” realizacijos diagrama pateikta 4.3.38 pav.

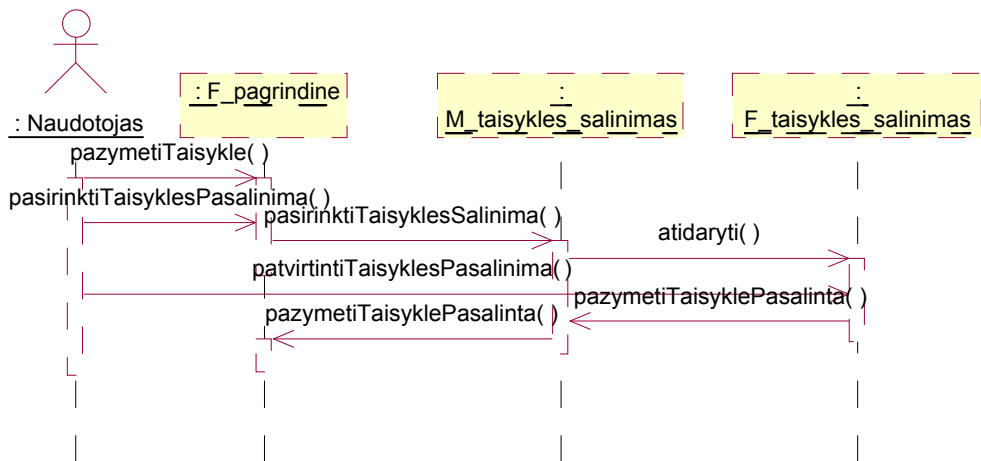


4.3.38 pav. Veiklos taisyklės pašalinimo panaudojimo atvejo realizacijos diagrama

Panaudojimo atvejo “Pašalinti veiklos taisyklę” sekų diagramos pateiktos 4.3.39 – 4.3.40 pav.



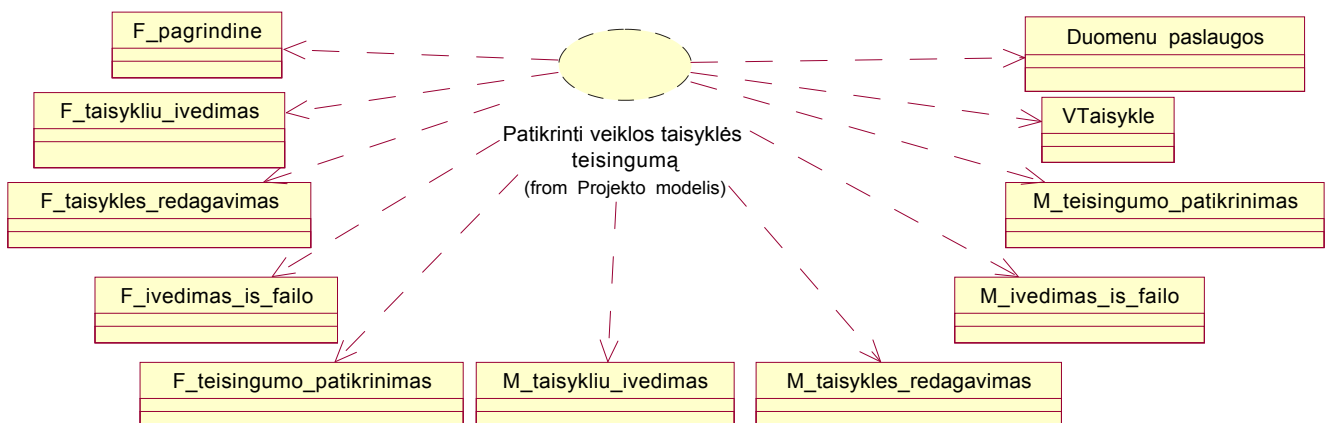
4.3.39 pav. Startuoti sistemą



4.3.40 pav. Pašalinti veiklos taisyklę

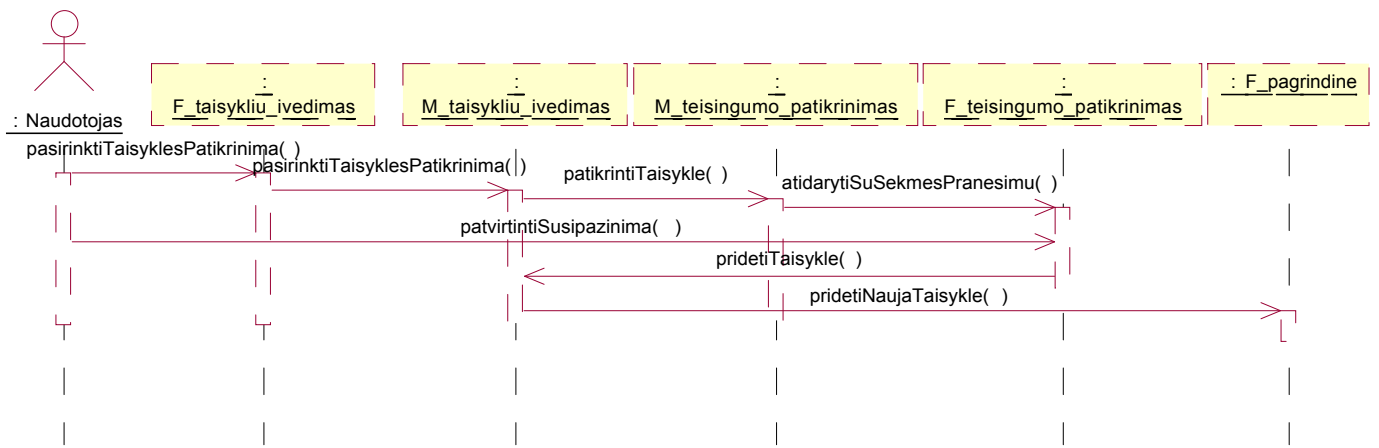
Patikrinti veiklos taisyklės teisingumą

Panaudojimo atvejo “Patikrinti veiklos taisyklės teisingumą” realizacijos diagrama pateikta 4.3.41 pav.

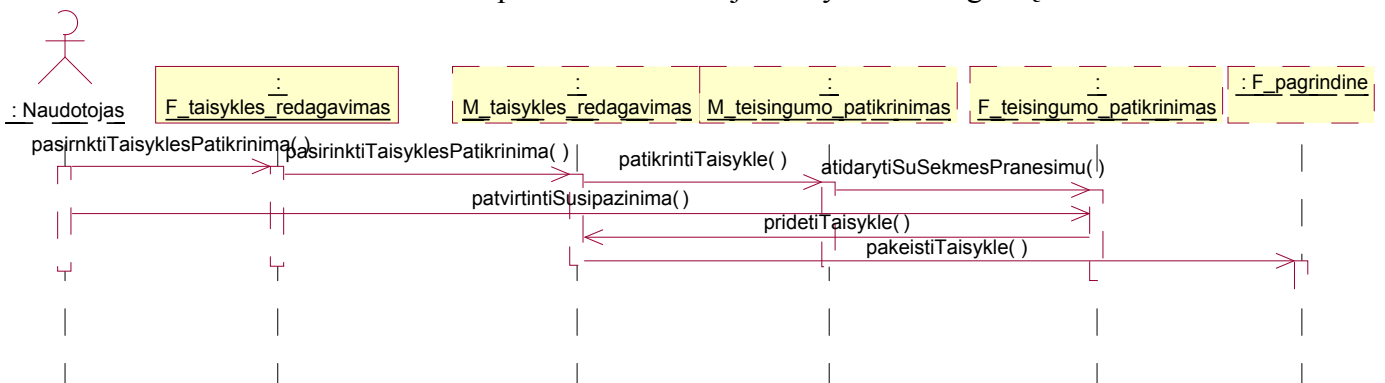


4.3.41 pav. Veiklos taisyklės teisingumo patikrinimo panaudojimo atvejo realizacijos diagrama

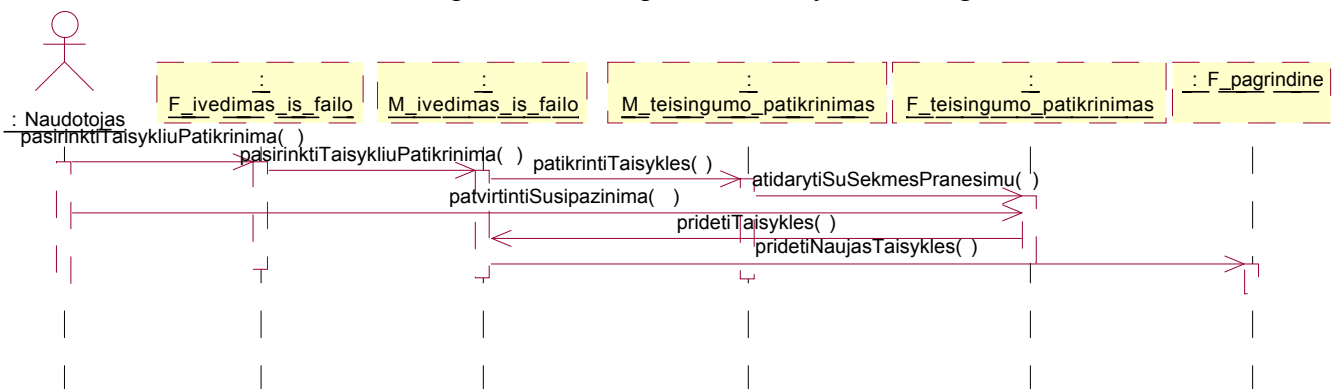
Panaudojimo atvejo “Patikrinti veiklos taisyklės teisingumą” sekų diagramos pateiktos 4.3.42 – 4.3.44 pav.



4.3.42 pav. Patikrinti naujos taisyklės teisingumą



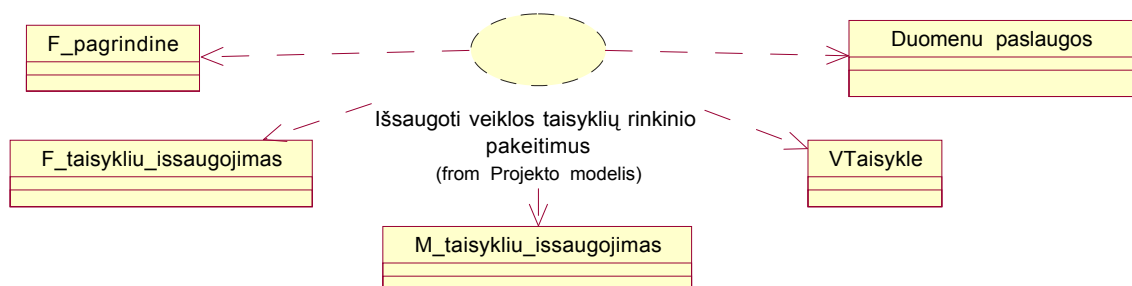
4.3.43 pav. Patikrinti pakeistos taisyklės teisingumą



4.3.44 pav. Patikrinti iš failo įvestos taisyklės teisingumą

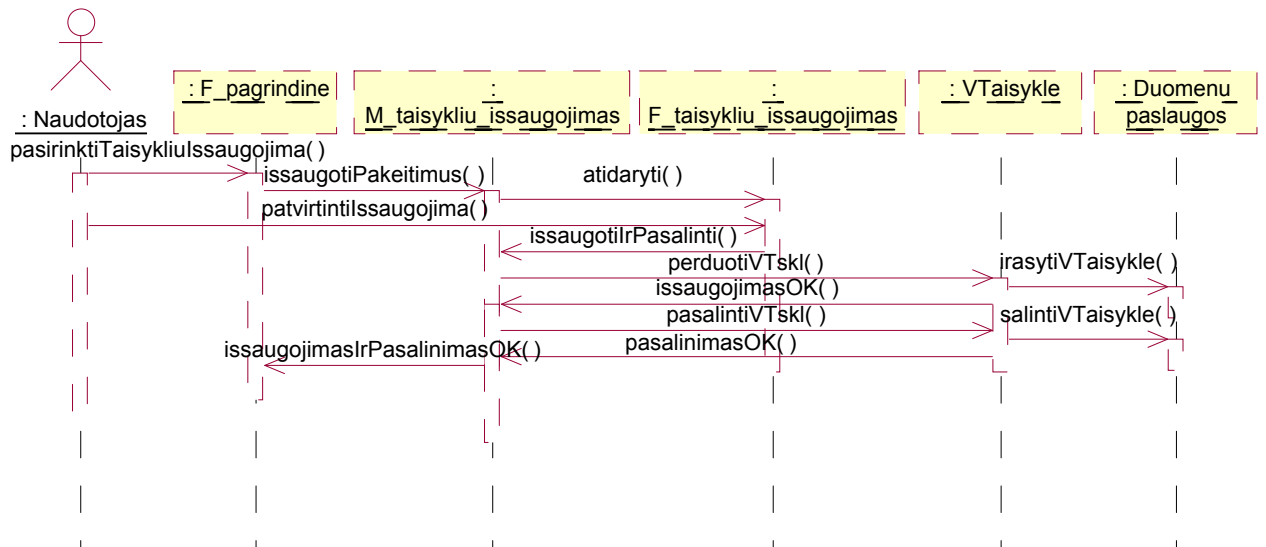
Išsaugoti veiklos taisyklių rinkinio pakeitimus

Panaudojimo atvejo “Išsaugoti veiklos taisyklių rinkinio pakeitimus” realizacijos diagrama pateikta 4.3.45 pav.



4.3.45 pav. Veiklos taisyklių rinkinio pakeitimų išsaugojimo panaudojimo atvejo realizacijos diagrama

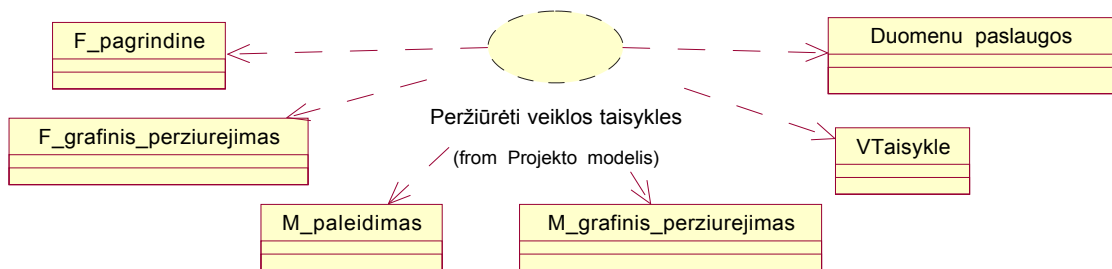
Panaudojimo atvejo “Išsaugoti veiklos taisyklių rinkinio pakeitimus ” seų diagrama pateikta 4.3.46 pav.



4.3.46 pav. Išsaugoti ir pašalinti taisykles

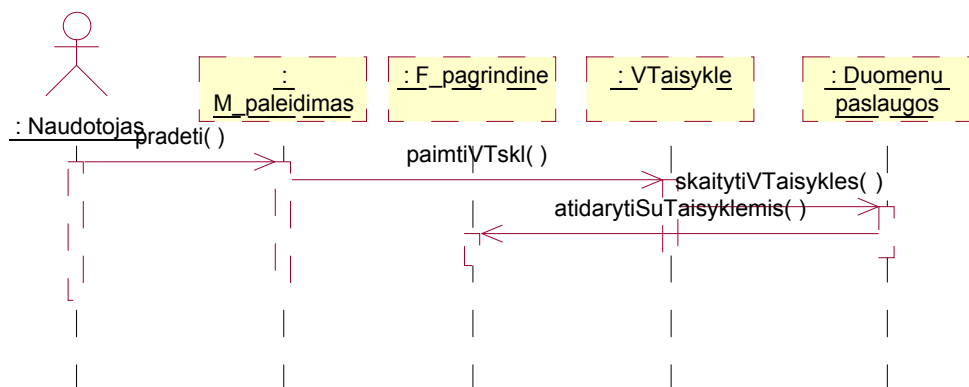
Peržiūrėti veiklos taisykles

Panaudojimo atvejo “Peržiūrėti veiklos taisykles” realizacijos diagrama pateikta 4.3.47 pav.

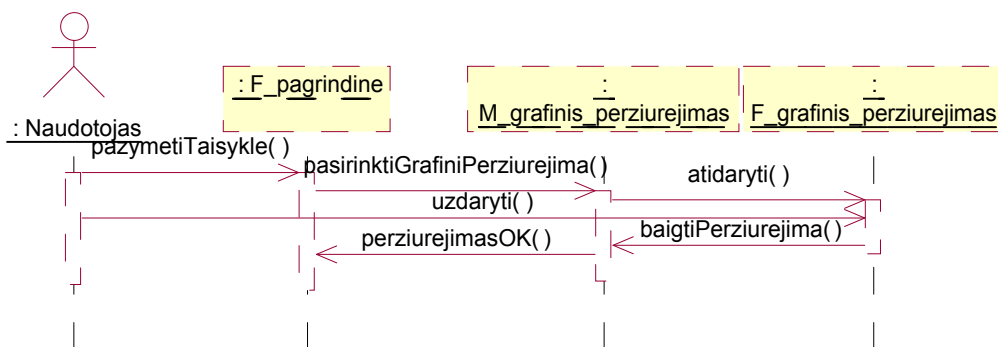


4.3.47 pav. Veiklos taisyklių peržiūrėjimo panaudojimo atvejo realizacijos diagrama

Panaudojimo atvejo “ Peržiūrėti veiklos taisykles ” seų diagramos pateiktos 4.3.48 – 4.3.49 pav.



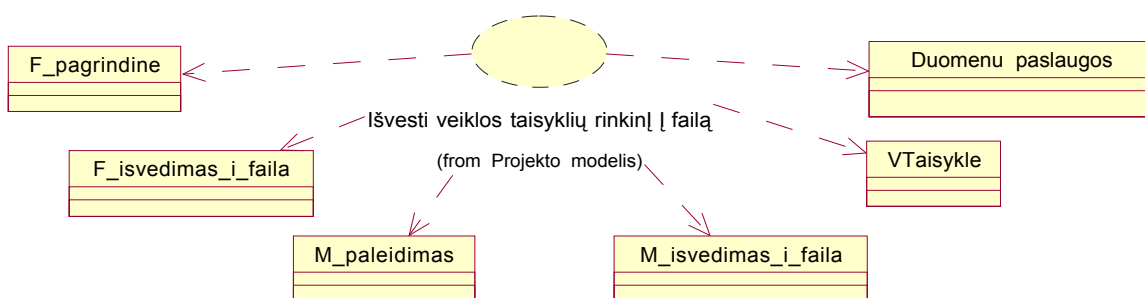
4.3.48 pav. Startuoti sistemą



4.3.49 pav. Peržiūrėti veiklos taisykles

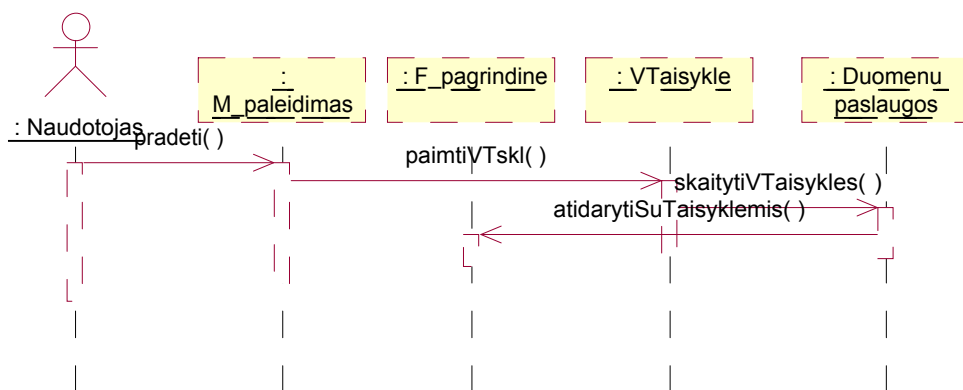
Išvesti veiklos taisyklių rinkinį į failą

Panaudojimo atvejo “Išvesti veiklos taisyklių rinkinį į failą” realizacijos diagrama pateikta 4.3.50 pav.

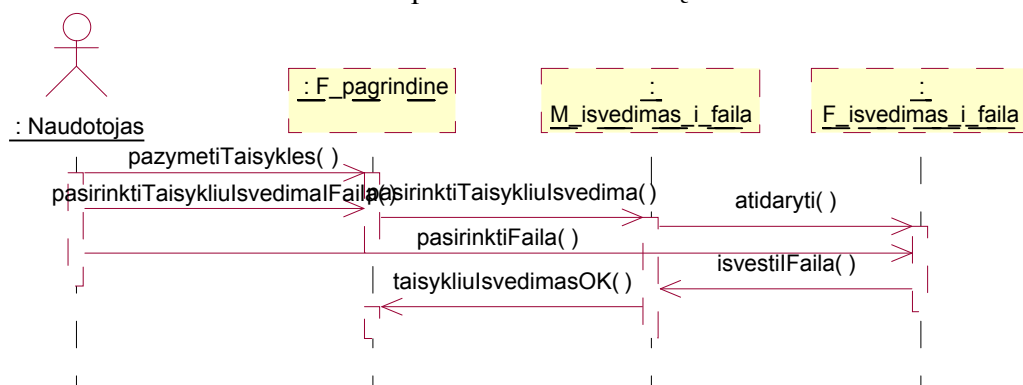


4.3.50 pav. Veiklos taisyklių rinkinio išvedimo į failą panaudojimo atvejo realizacijos diagrama

Panaudojimo atvejo “Išvesti veiklos taisyklių rinkinį į failą” sekų diagramos pateiktos 4.3.51 – 4.3.52 pav.



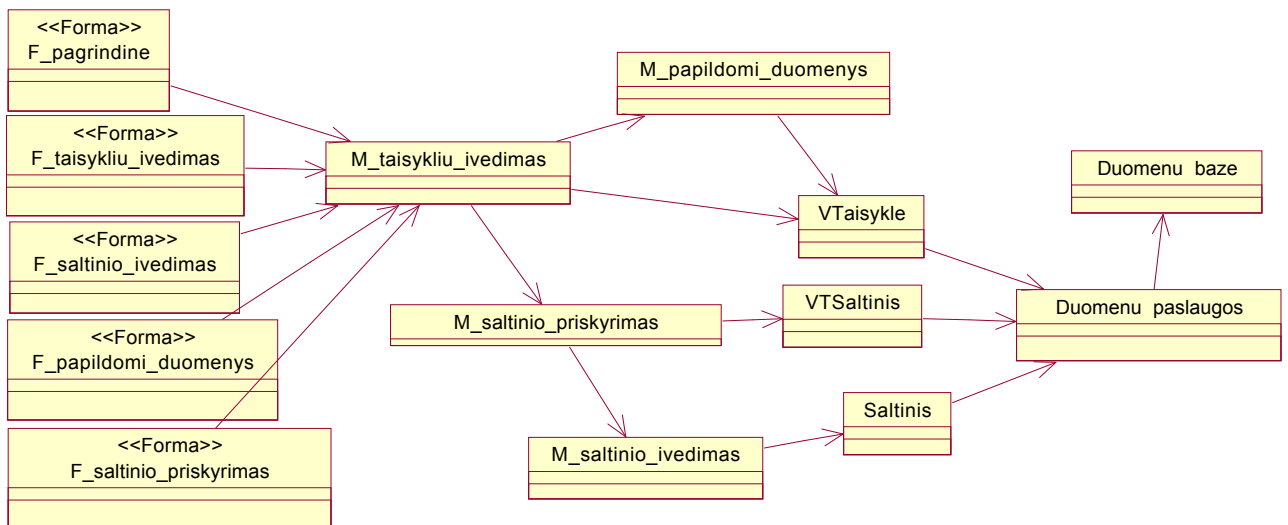
4.3.51 pav. Startuoti sistemą



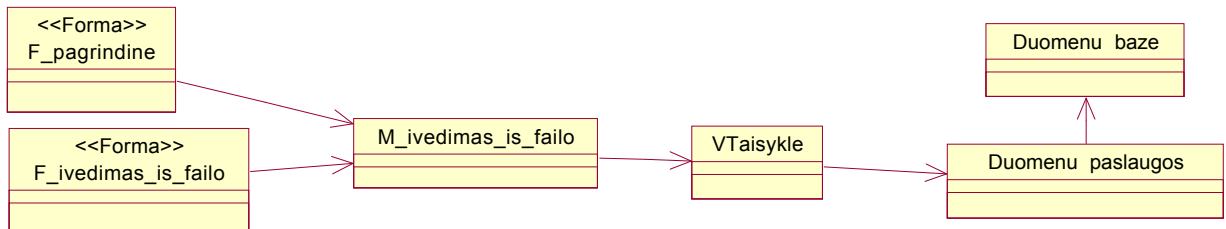
4.3.52 pav. Išvesti taisyklių rinkinį

4.3.4 Projekto modelio klasių diagramos

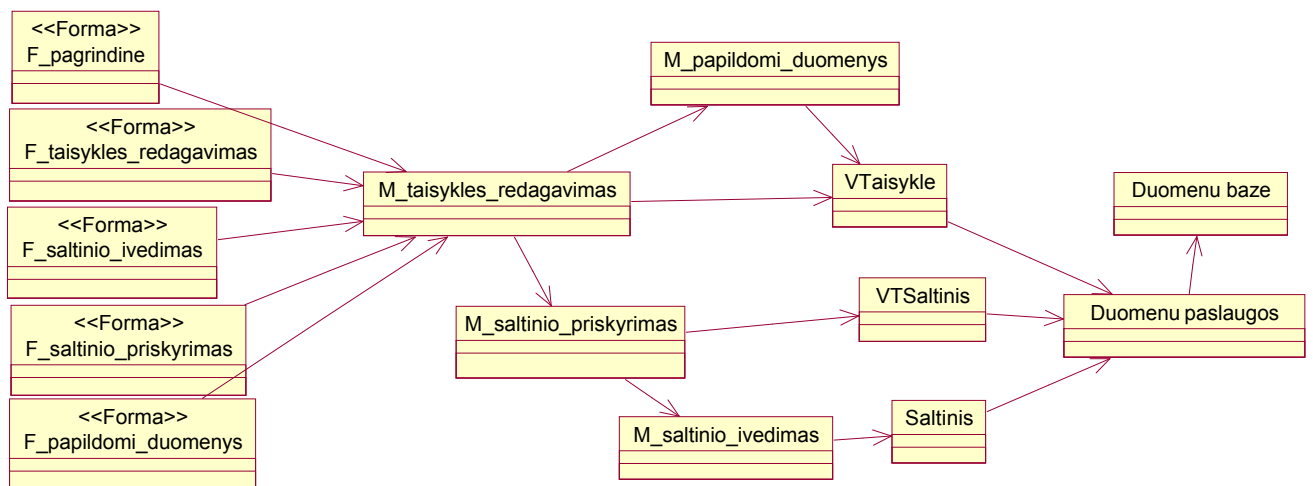
Bendros projekto klasių diagramos pateiktos 4.3.53 – 4.3.60 pav.



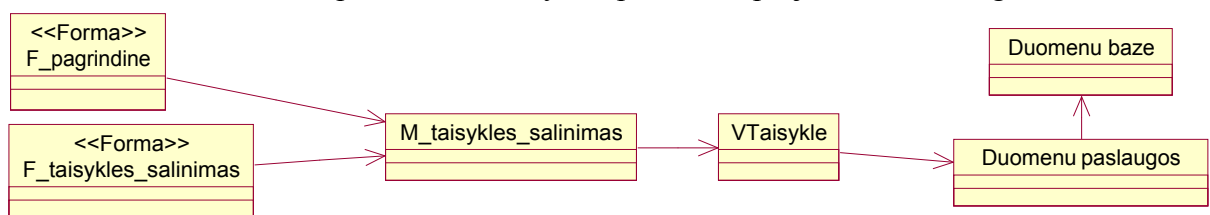
4.3.53 pav. Bendra naujos taisyklės įvedimo projekto klasių diagrama



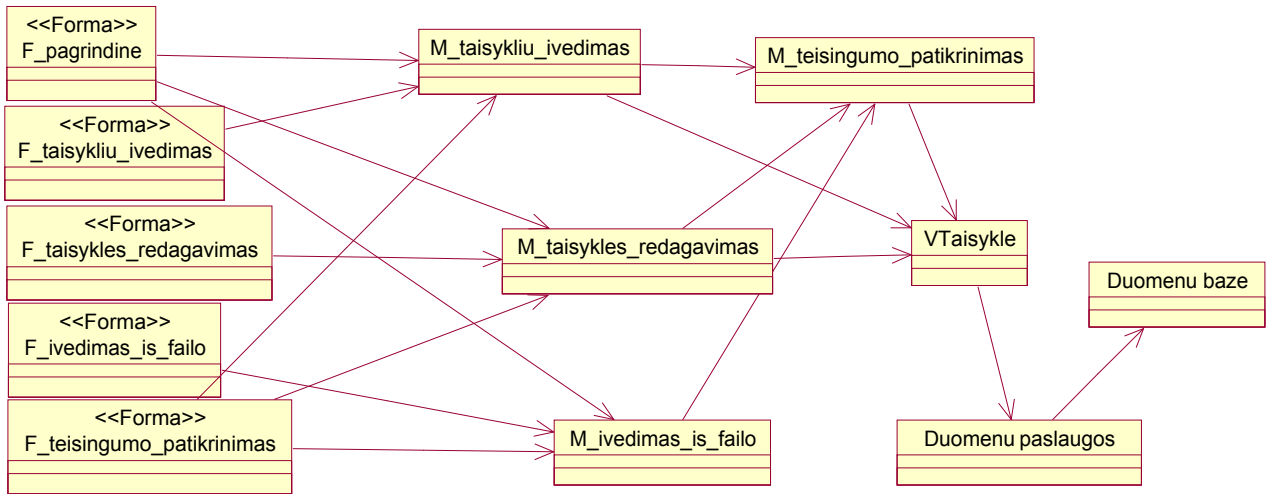
4.3.54 pav. Bendra taisyklių įvedimo iš failo projekto klasių diagrama



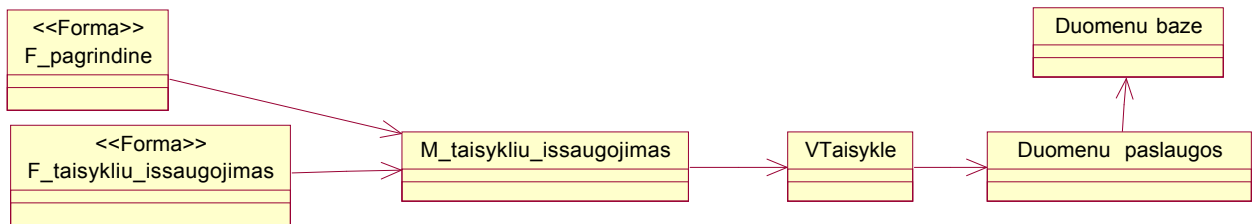
4.3.55 pav. Bendra taisyklių pakeitimo projekto klasių diagrama



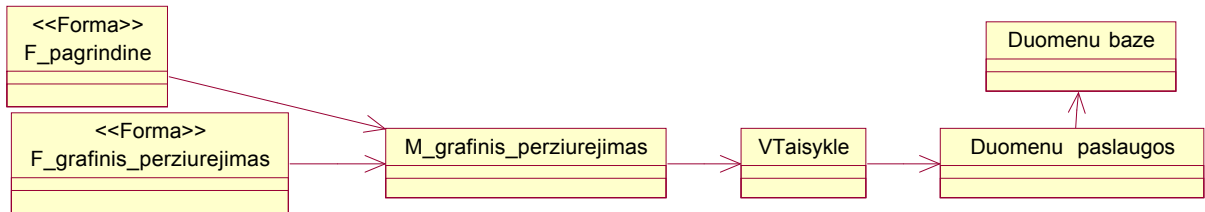
4.3.56 pav. Bendra taisyklės pašalinimo projekto klasių diagrama



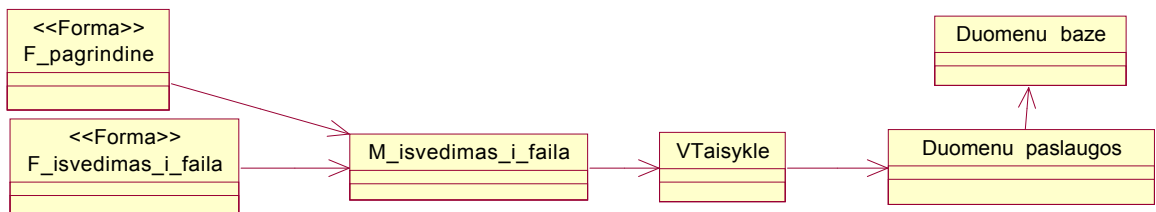
4.3.57 pav. Bendra taisyklių teisingumo tikrinimo projekto klasių diagrama



4.3.58 pav. Bendra taisyklių išsaugojimo projekto klasių diagrama

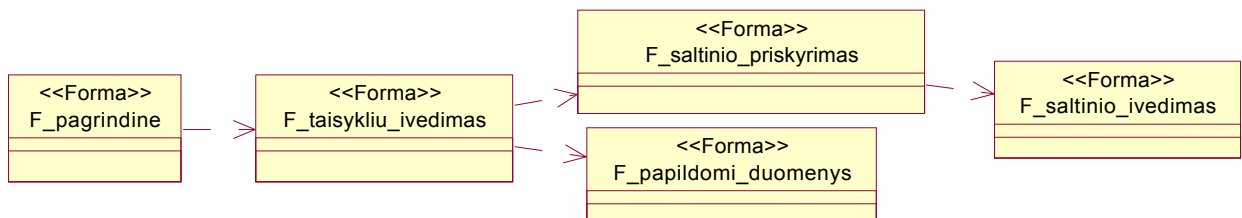


4.3.59 pav. Bendra taisyklių peržiūrėjimo projekto klasių diagrama

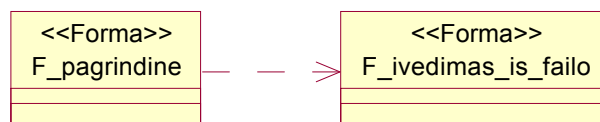


4.3.60 pav. Bendra taisyklių išvedimo į failą projekto klasių diagrama

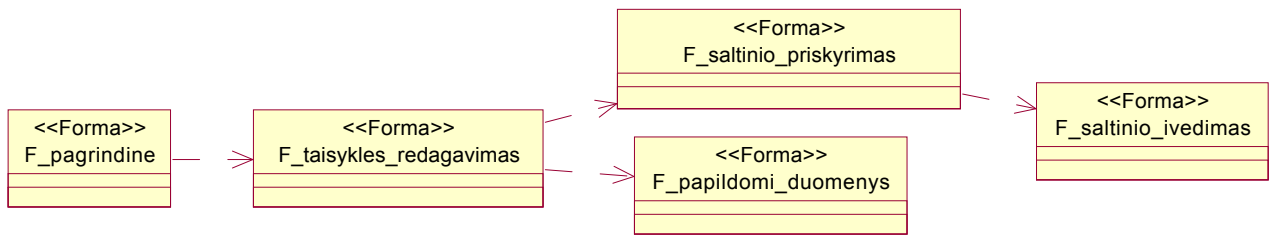
Sąsajos navigavimo planas pateiktas 4.3.61 – 4.3.68 pav.



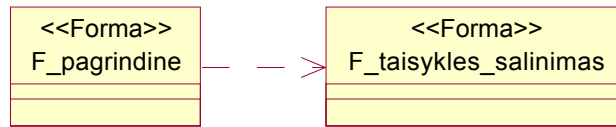
4.3.61 pav. Naujos taisyklės įvedimo sąsajos navigavimo planas



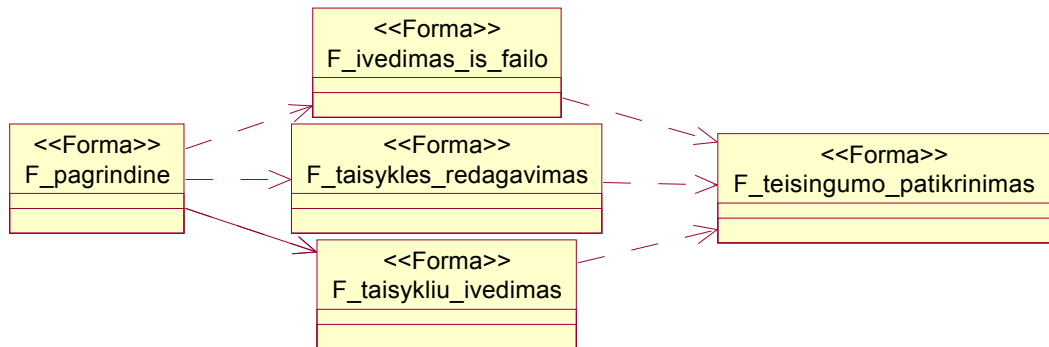
4.3.62 pav. Taisyklių įvedimo iš failo sąsajos navigavimo planas



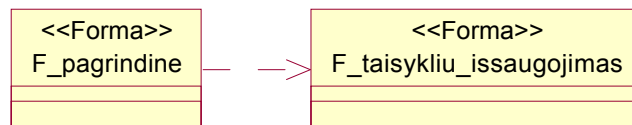
4.3.63 pav. Taisyklių pakeitimo sąsajos navigavimo planas



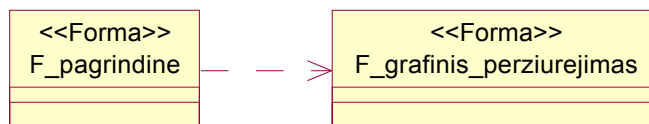
4.3.64 pav. Taisyklės pašalinimo sąsajos navigavimo planas



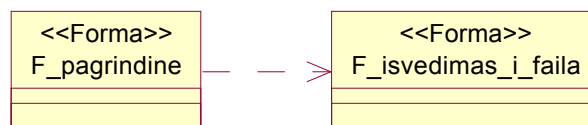
4.3.65 pav. Taisyklių teisingumo tikrinimo sąsajos navigavimo planas



4.3.66 pav. Taisyklių išsaugojimo sąsajos navigavimo planas



4.3.67 pav. Taisyklių peržiūrėjimo sąsajos navigavimo planas



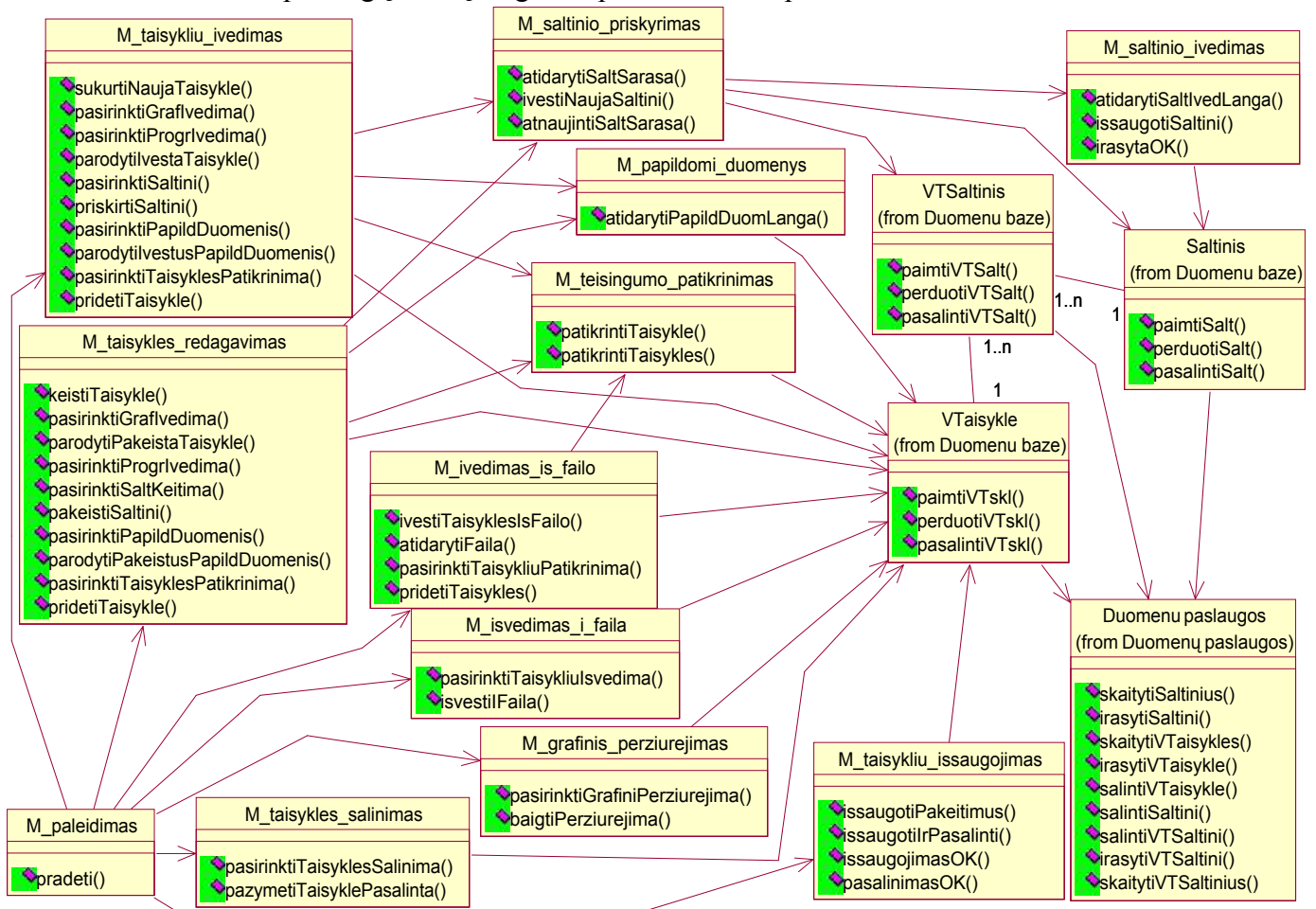
4.3.68 pav. Taisyklių išvedimo į failą sąsajos navigavimo planas

Detali vartotojo paslaugų klasių diagrama pateikta 4.3.69 pav.



4.3.69 pav. Detali vartotojo paslaugų klasių diagrama

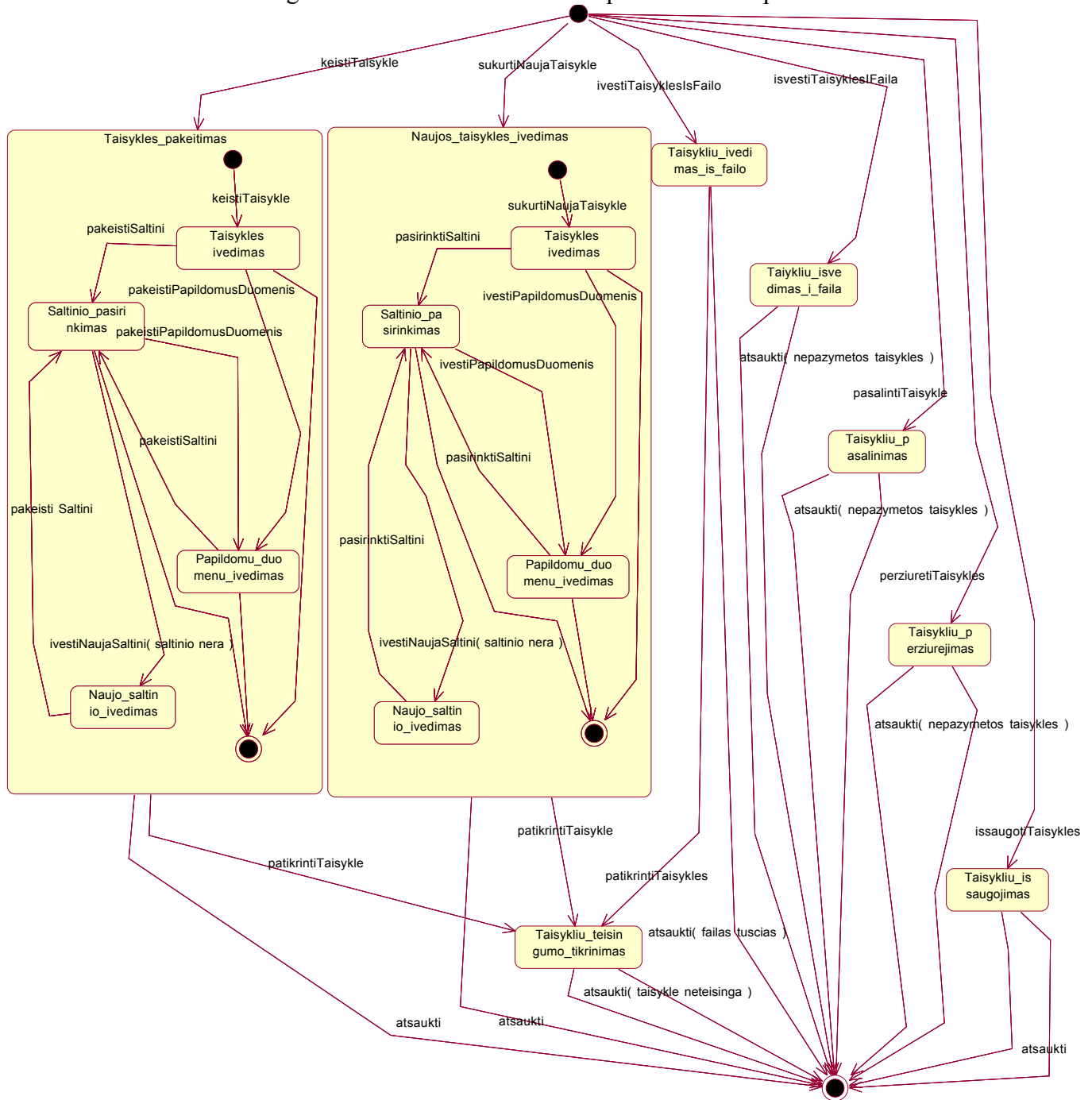
Detali veiklos paslaugų klasių diagrama pateikta 4.3.70 pav.



4.3.70 pav. Detali veiklos paslaugų klasių diagrama

4.3.5 Sistemos būsenos

Sistemos būsenos diagrama su vidinėmis būsenomis pateikta 4.3.71 pav.



4.3.71 pav. Sistemos būsenų diagrama su vidinėmis būsenomis

4.4 Paskirstymo ir komponentų modelis

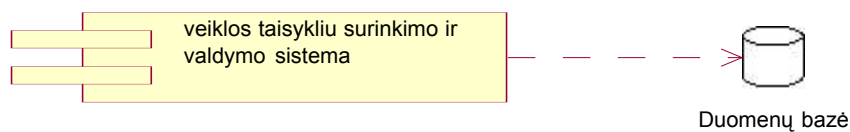
Paskirstymo diagrama pateikta 4.3.72 pav. Naudotojo taikomoji programa ir duomenų bazė yra tame pačiame asmeniniame kompiuteryje.



Taikomoji P|
Duomenų Bazė

4.4.1 pav. Paskirstymo diagrama

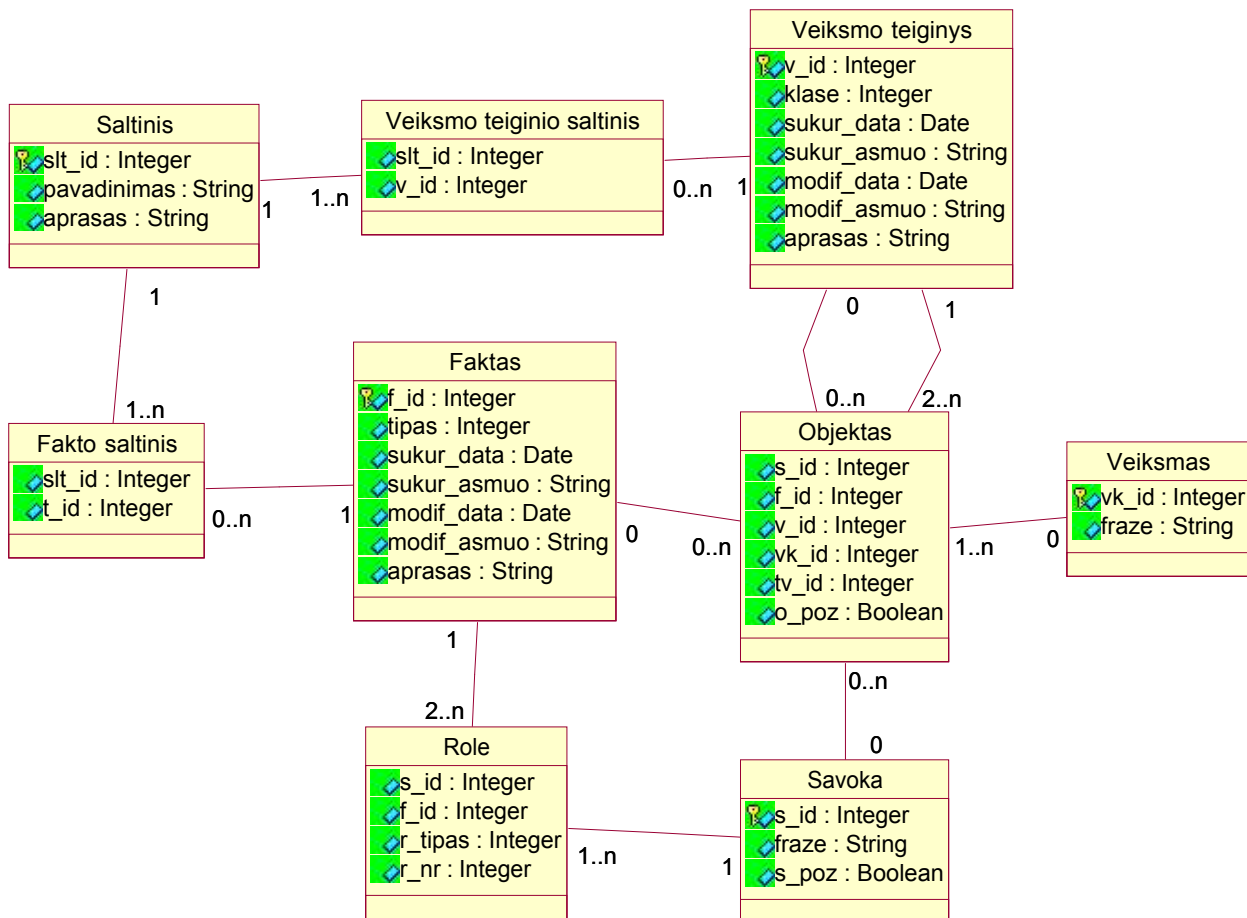
Veiklos taisyklių kaupimo sistemą sudaro du komponentai (4.4.2 pav.): veiklos taisyklių surinkimo ir valdymo sistema bei duomenų bazė.



4.4.2 pav. Komponentų diagrama

4.5 Duomenų modelis

Dalykinės srities klasių diagrama pateikta 4.5.1 pav.



4.6.1 pav. Dalykinės srities klasių diagrama

Veiklos taisyklių saugyklą sudaro šios lentelės:

- Savoka – lentelė, kurioje saugomos elementarios veiklos taisyklių dalys: sąvokos ir veiksmožodžiai. Lentelės atributai: **s_id** – unikalus identifikacinis sąvokos kodas, **fraze** – sąvoką nusakantis žodis arba frazė, **s_poz** – požymis, rodantis ar Įrašas yra sąvoka, ar veiksmožodis.
- Faktas – lentelė, kurioje saugomi faktai. Lentelės atributai: **f_id** – unikalus identifikacinis fakto kodas, **tipas** – fakto tipas, **sukur_data** – fakto sukūrimo data, **sukur_asmuo** – faktą sukūrusį asmuo, **modif_data** – fakto modifikavimo data, **modif_asmuo** – faktą modifikavęs asmuo.
- Role – lentelė, kurioje saugomos sakinių sudarančių sąvokų rolės, aprašančios sakinyje

dalyvaujančių sąvokų atliekamą semantinį vaidmenį. Lentelės atributai: **s_id** – unikalus identifikacinis sąvokos kodas, **f_id** – unikalus identifikacinis fakto kodas, **r_tipas** – rolės tipas, nusakantis sąvokos vietą fakte, **r_nr** – sąvokos eilės numeris tarp to paties tipo sąvokų.

- Veiksmo teiginys – lentelė, kurioje saugomi veiksmo teiginiai. Lentelės atributai: **v_id** – unikalus identifikacinis veiksmo teiginio kodas, **klase** – veiksmo teiginio klasė, **sukur_data** – veiksmo teiginio sukūrimo data, **sukur_asmuo** – veiksmo teiginį sukūrus asmuo, **modif_data** – veiksmo teiginio modifikavimo data, **modif_asmuo** – veiksmo teiginį modifikavęs asmuo.
- Veiksmas – lentelė, kurioje saugomi veiksmas. Lentelės atributai: **vk_id** – unikalus identifikacinis veiksmo kodas, **fraze** – veiksmą nusakanti frazė.
- Objektas – lentelė, kurioje saugomi veiksmo teiginį sudarantys objektai. Lentelės atributai: **s_id** – unikalus identifikacinis sąvokos kodas, **f_id** – unikalus identifikacinis fakto kodas, **v_id** – unikalus identifikacinis veiksmo teiginio kodas, **vk_id** – unikalus identifikacinis veiksmo kodas, **vt_id** – unikalus identifikacinis veiksmo teiginio, kuriam priklauso objektas, kodas, **o_poz** – požymis, rodantis objekto tipą, t. y. ar objektas yra bazinis, ar koresponduojantis.
- Šaltinis – lentelė, kurioje saugomi duomenys apie informacijos šaltinius (asmenis arba dokumentus), su kuriais susiejamos veiklos taisyklės. Lentelės atributai: **slt_id** – unikalus identifikacinis šaltinio kodas, **pavadinimas** – informacijos šaltinio pavadinimas, **aprasas** – informacijos šaltinio aprašas.
- Fakto šaltinis – lentelė, kurioje saugomi duomenys apie faktų informacijos šaltinius. Lentelės atributai: **slt_id** – unikalus identifikacinis šaltinio kodas, **f_id** – unikalus identifikacinis fakto kodas.
- Veiksmo teiginio šaltinis – lentelė, kurioje saugomi duomenys apie veiksmo teiginių informacijos šaltinius. Lentelės atributai: **slt_id** – unikalus identifikacinis šaltinio kodas, **v_id** – unikalus identifikacinis veiksmo teiginio kodas.

4.6 Sistemos realizacija

Realioji sistema buvo kuriama pagal GUIDE veiklos taisyklių metodologiją ir siūlomus jos patobulinimus. Sistema realizuota Java programavimo kalba. Ši kalba buvo pasirinkta, nes tai yra aukšto lygio, neutralios platformos, paprasta, pernešama (*portable*) ir objektiškai orientuota kalba. Leidžiant daugumą programinių kalbų sukurtas programos, jas reikia arba perkompiliuoti, arba interpretuoti iš naujo. Java šiuo atžvilgiu yra netradicinė, nes programa yra tuo pačiu metu yra ir kompiliuojama, ir interpretuojama. Kompiliatorius verčia programą į tarpinę kalbą, Javos baitų kodus (*Java bytecodes*) – nuo platformos nepriklausomas kodo instrukcijas. Kiekvieną kartą leidžiant programą interpretatorius vykdo Javo baitų kodo instrukcijas. Dėl to Javos programų nebereikia perkompiliuoti, net kai pernešama į kitą platformą.

Sistemos duomenų saugykla nebuvo numatyta jokių ypatingų reikalavimų, todėl pasirinkta Microsoft Access duomenų saugykla. Tai yra viena plačiausiai paplitusių duomenų saugyklų platinama kartu su MS Office paketu.

Sistemos realizavimo metu neiškilo didelių techninių problemų. Java programoje nekorektiškai veikia langų padalinimo funkcija (splitpane). Ši funkcija neleidžia nustatyti skiriamosios juostos pageidaujamoje vietoje, nors pagal specifikaciją turėtų. Java kūrėjai žino apie šią problemą, tačiau artimiausiu metu nėra numatę ją pašalinti. Tai nėra esminė problema ir vartotojui dėl to nepatogumų neturėtų iškilti.

Sistemos pagrindinės savybės:

1. Sistemos sąsajoje naudojama lietuvių kalba. Meniu punktai, pranešimai, žymės, pagalba pateikiami lietuvių kalba. Įvedant veiklos taisykles galima naudoti lietuvių kalbos simbolius – jie nėra iškraipomi.
2. Sistemos sąsają galima lengvai pritaikyti kitai kalbai. Meniu punktai, pranešimai, pagalba Įrašyti savybių (*properties*) faile, kurio turinį galima išversti į bet kokią kalbą. Pakeitus sąsajos savybių failą sistemos nereikia perkompiluoti – kitą kartą leidžiant programą matosi sąsajos pakeitimai.
3. Sistemoje realizuoti veiklos taisyklių struktūrizuoto pavidalo dalyje pateikti šablonai. Šablonai yra išsaugoti savybių (*properties*) faile, todėl juos galima modifikuoti: keisti taisyklių pavadinimus, šablonų dalis vietomis, išversti raktinius žodžius į kitą kalbą. Pakeitus šablonų savybių failą sistemos nereikia perkompiluoti – kitą kartą leidžiant programą veikia šablonų pakeitimai.
4. Sistema leidžia manipuluoti veiklos taisyklėmis: Įvesti, pakeisti, pašalinti, eksportuoti, importuoti, peržiūrėti taisyklės sudedamąsias dalis, Įvesti papildomus duomenis bei informacijos šaltinius. Sistemoje yra realizuotas veiklos taisyklių filtravimas pagal šaltinį, tipą arba taisyklės naudojamą sąvoką.
5. Sistemoje realizuota sąsaja su duomenų saugykla, kurioje pagal aprašytą saugojimo metodiką yra Įrašomos veiklos taisyklės.
6. Sistema užtikrina duomenų vientisumą. Sistema neišsaugo pakeitimų duomenų saugykloje, kol nėra perduoti visi duomenys. Programos „nulūžimas“ nėra pavojingas – duomenys veiklos taisyklių saugykloje neiškraipomi.
7. Sistema veikia Windows 98SE / ME / 2000 / XP operacinių sistemų aplinkoje.

Sistemos pagrindinis langas pateiktas 3 priede.

Naujos taisyklės Įvedimas į sistemą pateiktas 4 priede.

5. VEIKLOS TAISYKLIŲ IŠSAUGOJIMO EKSPERIMENTAS

5.1 Įvadas

Sistema sukurta atsižvelgiant į GUIDE veiklos taisyklių metodologiją bei siūlomus jos papildymus. Realizuota sistema tikrinama atliekant veiklos taisyklių išsaugojimo ir atkūrimo eksperimentą, kuris testuoja:

- Realizuotų *struktūrinių* ir *veiksmo teiginių* veiklos taisyklių įvedimą pagal koncepcinėje dalyje pateiktą struktūrizuotą veiklos taisyklių įvedimo metodiką. Tikrinama: ar sistema leidžia įvesti veiklos taisyklę; ar nustatyto jos tipą; ar veiklos taisyklė skaidoma į logiškai susijusias dalis, kurias galima saugoti atskirai.
- Realizuotų veiklos taisyklių saugyklą pagal koncepcinėje dalyje pateiktą struktūrą. Tikrinama: ar sistema išsaugo duomenis saugykloje pagal nustatytą struktūrą; ar sistema teisingai atkuria veiklos taisykles iš duomenų saugyklos.

Testuojant remiamasi automobilių nuomos punkto *veiklos politikos* fragmentu. Šis fragmentas yra paimtas iš GUIDE veiklos taisyklių projekte pateikto priedo [3], kad kiek įmanoma geriau atspindėtų GUIDE metodologiją. Pavyzdiniai *veiklos taisyklių sakiniai* sudaromi atsižvelgiant į pateiktą automobilių nuomos punkto *Veiklos politikos* fragmentą, tačiau nėra identiški GUIDE pateiktiems. Šio eksperimento metu, bandoma įrodyti, kad koncepcinėje dalyje pasiūlyti metodai veikia realioje sistemoje.

Be pagrindinių VT identifikuojančių ir aprašančių lentelių bei atributų, duomenų saugykloje saugomos papildomos informaciją apie taisykles teikiančios lentelės ir atributai (informacijos šaltiniai, sukūrimo data, sukūręs asmuo ir pan.). Siekiant supaprastinti eksperimentą ir padaryti jį aiškesniu, nėra modeliuojamas papildomos informacijos įvedimas į duomenų saugyklą.

5.2 Veiklos taisyklių įvedimas į sistemą

Prieš įvedant veiklos taisykles į sistemą, reikia nustatyti *veiklos politiką* sudarančius *veiklos taisyklių sakinius*, kurie po to skaidomi į nedalomas VT. Po to kiekviena VT klasifikuojama, nustatomos VT sudėtinės dalys (veiksmažodžiai, sąvokos – struktūriniais teiginiais; sąvokos, faktai, veiksmo teiginiai, veiksmai – veiksmo teiginiais; raktiniai žodžiai yra ignoruojami, nes jie tik nusako VT tipą, tačiau nesaugomi lentelėse), išskiriami įrašai, kurie bus įterpiami į duomenų saugyklą. Taisyklės įvedimas į sistemą pateiktas 4 priede.

Veiklos taisyklės sakiniai iš automobilių nuomos punkto *Veiklos politikos* :

1. Automobiliai, priskirti tam tikrai kategorijai, gali būti išnuomojami asmenims.

Šis VT sakinytis nėra nedaloma VT, todėl gali būti išskaidytas į dvi dalis:

- **Automobilio kategorija susideda iš automobilių.**

Vieta klasifikācijā: Struktūrinis teiginys / Faktas / Dalyvis / Agregātas.

Fakto daļys:

- a) pamirindas – *sāvoka* **automobilio kategorija**;
- b) pirmoji kompozicija – *sāvoka* **automobilis**.

Išskirti VT Iraiši pateikti 5.2.1 – 5.2.3 lentelē.

5.2.1 lentelē. Fakto Iraišas

f_id	tipas
1	<i>dalyvis – agregātas</i>

5.2.2 lentelē. Sāvokos Iraišas

s_id	fraze	s_poz
1	<i>automobilio kategorija</i>	<i>sāvoka</i>
2	<i>automobilis</i>	<i>sāvoka</i>

5.2.3 lentelē. Rolēs Iraišas

f_id	s_id	r_tipas	s_nr
1	1	<i>pagr</i>	0
1	2	<i>I komp</i>	0

- **Asmuo gali išsinuomoti automobilį**

Vieta klasifikācijā: Struktūrinis teiginys / Faktas / Dalyvis / Asociācija.

Fakto daļys:

- a) pamirindas – *sāvoka* **asmuo**;
- b) veiksmāžodis – *sāvoka* **išsinuomoti**;
- c) pirmoji kompozicija – *sāvoka* **automobilis**.

Išskirti VT Iraiši pateikti 5.2.4 – 5.2.6 lentelē.

5.2.4 lentelē. Fakto Iraišas

f_id	tipas
2	<i>dalyvis – asociācija</i>

5.2.5 lentelē. Sāvokos Iraišas

s_id	fraze	s_poz
3	<i>asmuo</i>	<i>sāvoka</i>
4	<i>išsinuomoti</i>	<i>veiksmāžodi s</i>

5.2.6 lentelė. Rolės Įrašas

f_id	s_id	r_tipas	s_nr
2	3	<i>pagr</i>	0
2	4	<i>veiksm</i>	0
2	2	<i>I komp</i>	0

2. Didesnė dalis užsakymų yra pateikiami iš anksto; nuomos periodas ir automobilio kategorija yra pateikiama užsakymo metu.

Šis VT sakinys nėra nedaloma VT, todėl gali būti išskaidytas į tris dalis:

- **Asmuo gali pateikti išankstinį užsakymą.**

Vieta klasifikacijoje: Struktūrinis teiginys / Faktas / Dalyvis / Asociacija.

Fakto dalys:

- pagrindas – *sąvoka* **asmuo**;
- veiksmažodis – *sąvoka* **pateikti**;
- pirmoji kompozicija – *sąvoka* **išankstinis užsakymas**.

Išskirti VT Įrašai pateikti 5.2.7 – 5.2.9 lentelėse.

5.2.7 lentelė. Fakto Įrašas

f_id	tipas
3	<i>dalyvis – asociacija</i>

5.2.8 lentelė. Sąvokos Įrašas

s_id	fraze	s_poz
5	<i>pateikti</i>	<i>veiksmažodis</i>
6	<i>išankstinis užsakymas</i>	<i>sąvoka</i>

5.2.9 lentelė. Rolės Įrašas

f_id	s_id	r_tipas	s_nr
3	2	<i>pagr</i>	0
3	5	<i>veiksm</i>	0
3	6	<i>I komp</i>	0

- **Išankstinis užsakymas yra užsakymas.**

Vieta klasifikacijoje: Struktūrinis teiginys / Faktas / Apibendrinimas.

Fakto dalys:

- pirmoji kompozicija – *sąvoka* **išankstinis užsakymas**;
- pagrindas – *sąvoka* **užsakymas**.

Išskirti VT Įrašai pateikti 5.2.10 – 5.2.12 lentelėse.

5.2.10 lentelė. Fakto Įrašas

f_id	tipas
4	<i>apibendrinimas</i>

5.2.11 lentelė. Sąvokos Įrašas

s_id	fraze	s_poz
7	<i>užsakymas</i>	<i>sąvoka</i>

5.2.12 lentelė. Rolės Įrašas

f_id	s_id	r_tipas	s_nr
4	6	<i>I komp</i>	0
4	7	<i>pagr</i>	0

- **Nuomos periodas, automobilio kategorija yra užsakymo savybė.**

Vieta klasifikacijoje: Struktūrinis teiginys / Faktas / Atributas.

Fakto dalys:

- pirmosios kompozicijos pirmoji *sąvoka* **nuomos periodas**;
- pirmosios kompozicijos antroji *sąvoka* **automobilio kategorija**;
- pagrindas – *sąvoka* **užsakymas**.

Išskirti VT Įrašai pateikti 5.2.13 – 5.2.15 lentelėse.

5.2.13 lentelė. Fakto Įrašas

f_id	tipas
5	<i>atributas</i>

5.2.14 lentelė. Sąvokos Įrašas

s_id	fraze	s_poz
8	<i>nuomos periodas</i>	<i>sąvoka</i>

5.2.15 lentelė. Rolės Įrašas

f_id	s_id	r_tipas	s_nr
5	1	<i>I komp</i>	1
5	8	<i>I komp</i>	0
5	7	<i>pagr</i>	0

3. Automobilis gali būti išnuomotas pateikus skubų užsakymą.

Šis VT sakiny s nėra nedaloma VT, todėl gali būti išskaidytas į dvi dalis:

- **Asmuo gali pateikti skubų užsakymą.**

Vieta klasifikacijoje: Struktūrinis teiginys / Faktas / Dalyvis / Asociacija.

Fakto dalys:

- a) pagrindas – *sąvoka* **asmuo**;
- b) veiksmožodis – *sąvoka* **pateikti**;
- c) pirmoji kompozicija – *sąvoka* **skubus užsakymas**.

Išskirti VT Įrašai pateikti 5.2.16 – 5.2.18 lentelėse.

5.2.16 lentelė. Fakto Įrašas

f_id	tipas
6	<i>dalyvis – asociacija</i>

5.2.17 lentelė. Sąvokos Įrašas

s_id	fraze	s_poz
9	<i>skubus užsakymas</i>	<i>sąvoka</i>

5.2.18 lentelė. Rolės Įrašas

f_id	s_id	r_tipas	s_nr
6	3	<i>pagr</i>	0
6	5	<i>veiksm</i>	0
6	9	<i>I komp</i>	0

- **Skubus užsakymas yra užsakymas.**

Vieta klasifikacijoje: Struktūrinis teiginys / Faktas / Apibendrinimas.

Fakto dalys:

- a) pirmoji kompozicija – *sąvoka* **skubus užsakymas**;
- b) pagrindas – *sąvoka* **užsakymas**.

Išskirti VT Įrašai pateikti 5.2.19 – 5.2.20 lentelėse.

5.2.19 lentelė. Fakto Įrašas

f_id	tipas
7	<i>apibendrinimas</i>

5.2.20 lentelė. Rolės Įrašas

f_id	s_id	r_tipas	s_nr
7	9	<i>I komp</i>	0
7	7	<i>pagr</i>	0

4. Užsakyme turi būti numatytas nuomos periodas ir automobilio kategorija.

Šis VT sakinys yra nedaloma VT, jos struktūrizuota forma:

- **Užsakymas privalo turėti nuomos periodą, automobilio kategoriją.**

Vieta klasifikacijoje: Veiksmo teiginys / Apribojimas.

Fakto dalys:

- a) bazinis objektas – *sqvoka* **užsakymas**;
- b) koresponduojantis objektas – *faktas* **Nuomos periodas, automobilio kategorija yra užsakymo savybė.**

Išskirti VT įrašai pateikti 5.2.21 – 5.2.22 lentelėse.

5.2.21 lentelė. Veiksmo teiginio įrašas

v_id	klasė
1	<i>apribojimas</i>

5.2.22 lentelė. Objekto įrašas

f_id	s_id	v_id	vk_id	tv_id	o_pož
-1	7	-1	-1	1	<i>baz. obj.</i>
5	-1	-1	-1	1	<i>koresp. obj.</i>

5. Jeigu pateiktas užsakymas yra skubus, tada reikia nedelsiant pateikti automobilį.

Šis VT sakiny yra nedaloma VT, jos struktūrizuota forma:

- **Jeigu užsakymas yra skubus užsakymas, tai pateikti automobilį.**

Vieta klasifikacijoje: Veiksmo teiginys / Sąlyga.

Fakto dalys:

- a) bazinis objektas – *faktas* **Skubus užsakymas yra užsakymas**;
- b) koresponduojantis objektas – *veiksmas* **pateikti automobilį.**

Išskirti VT įrašai pateikti 5.2.23 – 5.2.25 lentelėse.

5.2.23 lentelė. Veiksmo teiginio įrašas

v_id	klasė
2	<i>sąlyga</i>

5.2.24 lentelė. Veiksmo įrašas

vk_id	frazė
1	<i>pateikti automobilį</i>

5.2.25 lentelė. Objekto įrašas

f_id	s_id	v_id	vk_id	tv_id	o_pož
7	-1	-1	-1	2	<i>baz. obj.</i>
-1	-1	-1	1	2	<i>koresp. obj.</i>

Sudaryti VT įrašai yra įvesti į sistemą ir paruošti išsaugojimui duomenų saugykloje.

5.3 Veiklos taisyklių išsaugojimas saugykloje

Skyriuje 5.2, kur aprašomas veiklos taisyklių suformavimas, buvo sudaryti įrašai, kurie įterpiami į duomenų saugyklą. Duomenų bazės lentelės pateiktos 5.3.1 – 5.3.6 lentelėse.

5.3.1 lentelė. Lentelė Faktas

f_id	tipas
1	dalyvis – agregatas
2	dalyvis – asociacija
3	dalyvis – asociacija
4	apibendrinimas
5	atributas
6	dalyvis – asociacija
7	apibendrinimas

5.3.2 lentelė. Lentelė Sąvoka

s_id	fraze	s_poz
1	automobilio kategorija	sąvoka
2	automobilis	sąvoka
3	asmuo	sąvoka
4	išsinuomoti	veiksmazodis
5	pateikti	veiksmazodis
6	išankstinis užsakymas	sąvoka
7	užsakymas	sąvoka
8	nuomos periodas	sąvoka
9	skubus užsakymas	sąvoka

5.3.3 lentelė. Lentelė Rolė

f_id	s_id	r_tipas	r_nr
1	1	pagr.	0
1	2	I komp.	0
2	3	pagr.	0
2	4	veiksm..	0
2	2	I komp.	0
3	3	pagr.	0
3	5	veiksm..	0
3	6	I komp.	0
4	6	I komp.	0
4	7	pagr	0
5	1	I komp.	1
5	8	I komp	0
5	7	pagr	0
6	3	pagr.	0
6	5	veiksm..	0
6	9	I komp.	0
7	9	I komp.	0
7	7	pagr	0

5.3.4 lentelė. Lentelė Veiksmo teiginys

v_id	klasė
1	<i>apribojimas</i>
2	<i>sąlyga</i>

5.3.5 lentelė. Lentelė Veiksmas

vk_id	frazė
1	<i>pateikti automobilį</i>

5.3.6 lentelė. Lentelė Objektas

f_id	s_id	v_id	vk_id	tv_id	o_pož
-1	7	-1	-1	1	<i>baz. obj.</i>
5	-1	-1	-1	1	<i>koresp. obj.</i>
7	-1	-1	-1	2	<i>baz. obj.</i>
-1	-1	-1	1	2	<i>koresp. obj.</i>

5.4 Veiklos taisyklių atkūrimas

Veiklos politika atkurama iš saugykloje saugomų Įrašų. *Struktūriniai teiginiai* atkuriami iš lentelės **Faktas**, imant Įrašus iš eilės. Pagal fakto tipą nustatomas *fakto* šablonas, į kurio kintamą dalį Įstatomi *veiksmožodžiai* ir *sąvokos*. Kurias *sąvokas* Įstatyti nustatoma iš lentelės **RoLė**: pagal lauką f_id nustatomas naudojamos *sąvokos* identifikacinis numeris s_id. Tuomet iš lentelės **Sąvoka** pagal lauką s_id paimamas laukas fraze, kuris Įstatomas į lentelės **RoLė** r_tipas lauke nurodytą vietą. Jei r_tipas yra *pirmoji kompozicija*, ir tas pats faktas turi kelias *pirmąsias kompozicijas*, tai šios kintamos dalies *sąvokos* Įrašomos iš eilės pagal lauke r_nr nurodytą eilės numerį, atskiriant vieną nuo kitos kableliu.

Veiksmo teiginiai atkuriami iš lentelės **Veiksmo teiginys**, imant Įrašus iš eilės. Pagal veiksmo teiginio klasę nustatomas *veiksmo teiginio* šablonas, į kurio kintamą dalį Įstatomi baziniai ir koresponduojantys objektai. Kuriuos objektus Įstatyti nustatoma iš lentelės **Objektas**: pagal lauką tv_id nustatomas sąveikaujantis objektas (sąvoka, faktas, kitas veiksmo teiginys arba veiksmas) ir jo tipas (bazinis ar koresponduojantis objektas). Tuomet pagal nustatytą objekto šabloną objektas Įterpiamas į numatytą vietą.

Faktų atkūrimas:

1. Fakto, kurio f_id = 1, Įrašas pateiktas 5.4.1 lentelėje.

5.4.1 lentelė. Fakto Įrašas

f_id	tipas
1	<i>dalyvis – agregatas</i>

Fakto tipas – *dalyvis - agregatas*, jo šablonas: *<pagrindas> susideda iš <pirmoji kompozicija>*.

Lentelėje **RoLė**, nustatomi sąvokos identifikaciniai numeriai, kur f_id = 1 (s_id = 1 ir s_id = 2). Jie pateikiami 5.4.2 lentelėje.

5.4.2 lentelė. Rolės Įrašai

f_id	s_id	r_tipas	r_nr
1	1	pagr.	0
1	2	I komp.	0

Iš lentelės **Sąvoka**, pagal lauką s_id paimami reikalingi lauko fraze Įrašai, t. y. **automobilio kategorija** ir **automobilis**. Jie pateikiami 5.4.3 lentelėje.

5.4.3 lentelė. Sąvokos Įrašai

s_id	fraze	s_poz
1	automobilio kategorija	sąvoka
2	automobilis	sąvoka

Frazės, atsižvelgiant į sąvokos rolės tipą (r_tipas) Įstatomos Į šabloną: **automobilio kategorija** Į pagrindo vietą, o **automobilis** Į pirmosios kompozicijos vietą. Tokiu būdu atkurta VT:

Automobilio kategorija susideda iš automobilio.

2. Fakto, kurio f_id = 2, Įrašas pateiktas 5.4.4 lentelėje.

5.4.4 lentelė. Fakto Įrašas

f_id	tipas
2	dalyvis – asociacija

Fakto tipas – *dalyvis – asociacija*, jo šablonas: <pagrindas> gali <veiksmažodis> <pirmoji kompozicija>.

Lentelėje **Rolė**, nustatomi sąvokos identifikaciniai numeriai, kur f_id = 2 (s_id = 3, s_id = 4 ir s_id = 2). Jie pateikiami 5.4.5 lentelėje.

5.4.5 lentelė. Rolės Įrašai

f_id	s_id	r_tipas	r_nr
2	3	pagr.	0
2	4	veiksm..	0
2	2	I komp.	0

Iš lentelės **Sąvoka**, pagal lauką s_id paimami reikalingi lauko fraze Įrašai, t. y. **asmuo**, **išsiniuomoti**, **automobilis**. Jie pateikiami 5.4.6 lentelėje.

5.4.6 lentelė. Sąvokos Įrašai

s_id	fraze	s_poz
2	automobilis	sąvoka
3	asmuo	sąvoka
4	išsiniuomoti	veiksmažodis

Frazės, atsižvelgiant į sąvokos rolės tipą (r_tipas) Įstatomos Į šabloną: **asmuo** Į pagrindo vietą, **išsiniuomoti** Į veiksmažodžio vietą, o **automobilis** Į pirmosios kompozicijos vietą. Tokiu būdu atkurta VT:

Asmuo gali išsiniuomoti automobilį.

3. Fakto, kurio f_id = 3, Įrašas pateiktas 5.4.7 lentelėje.

5.4.7 lentelė. Fakto Įrašas

f_id	tipas
3	<i>dalyvis – asociacija</i>

Fakto tipas – *dalyvis - asociacija*, jo šablonas: <*pagrindas*> *gali* <*veiksmazodis*> <*pirmoji kompozicija*>.

Lentelėje **Rolė**, nustatomi sąvokos identifikaciniai numeriai, kur f_id = 3 (s_id = 3, s_id = 5 ir s_id = 6). Jie pateikiami 5.4.8 lentelėje.

5.4.8 lentelė. Rolės Įrašai

f_id	s_id	r_tipas	r_nr
3	3	<i>pagr.</i>	0
3	5	<i>veiksm..</i>	0
3	6	<i>I komp.</i>	0

Iš lentelės **Sąvoka**, pagal lauką s_id paimami reikalingi lauko frazė Įrašai, t. y. **asmuo**, **pateikti**, **išankstinis užsakymas**. Jie pateikiami 5.4.9 lentelėje.

5.4.9 lentelė. Sąvokos Įrašai

s_id	fraze	s_poz
3	<i>asmuo</i>	<i>sąvoka</i>
5	<i>pateikti</i>	<i>veiksmazo dis</i>
6	<i>išankstinis užsakymas</i>	<i>sąvoka</i>

Frazės, atsižvelgiant į sąvokos rolės tipą (r_tipas) Įstatomos Į šabloną: **asmuo** Į pagrindo vietą, **pateikti** Į veiksmazodžio vietą, o **išankstinis užsakymas** Į pirmosios kompozicijos vietą. Tokiu būdu atkurta VT:

Asmuo gali pateikti išankstinį užsakymą.

4. Fakto, kurio f_id = 4, Įrašas pateiktas 5.4.10 lentelėje.

5.4.10 lentelė. Fakto Įrašas

f_id	tipas
4	<i>apibendrinimas</i>

Fakto tipas – *apibendrinimas*, jo šablonas: <*pirmoji kompozicija*> *yra* <*pagrindas*>.

Lentelėje **Rolė**, nustatomi sąvokos identifikaciniai numeriai, kur f_id = 4 (s_id = 6 ir s_id = 7). Jie pateikiami 5.4.11 lentelėje.

5.4.11 lentelė. Rolės Įrašai

f_id	s_id	r_tipas	r_nr
4	6	<i>I komp.</i>	0
4	7	<i>pagr</i>	0

Iš lentelės **Sąvoka**, pagal lauką s_id paimami reikalingi lauko frazė Įrašai, t. y. **išankstinis užsakymas** ir **užsakymas**. Jie pateikiami 5.4.12 lentelėje.

5.4.12 lentelė. Sąvokos Įrašai

s_id	fraze	s_poz
6	<i>išankstinis užsakymas</i>	<i>sąvoka</i>
7	<i>užsakymas</i>	<i>sąvoka</i>

Frazės, atsižvelgiant į sąvokos rolės tipą (r_tipas) įstatomos į šabloną: **išankstinis užsakymas** į pirmosios kompozicijos vietą, o **užsakymas** į pagrindo vietą. Tokiu būdu atkurta VT:

Išankstinis užsakymas yra užsakymas.

5. Fakto, kurio f_id = 5, Įrašas pateiktas 5.4.13 lentelėje.

5.4.13 lentelė. Fakto Įrašas

f_id	tipas
5	<i>atributas</i>

Fakto tipas – *atributas*, jo šablonas: *<pirmoji kompozicija> yra <pagrindas> savybė.*

Lentelėje **Rolė**, nustatomi sąvokos identifikaciniai numeriai, kur f_id = 5 (s_id = 1, s_id = 8 ir s_id = 7). Jie pateikiami 5.4.14 lentelėje.

5.4.14 lentelė. Rolės Įrašai

f_id	s_id	r_tipas	r_nr
5	1	<i>I komp.</i>	1
5	8	<i>I komp</i>	0
5	7	<i>pagr</i>	0

Iš lentelės **Sąvoka**, pagal lauką s_id paimami reikalingi lauko fraze Įrašai, t. y. **automobilio kategorija**, **užsakymas** ir **nuomos periodas**. Jie pateikiami 5.4.15 lentelėje.

5.4.15 lentelė. Sąvokos Įrašai

s_id	fraze	s_poz
1	<i>automobilio kategorija</i>	<i>sąvoka</i>
7	<i>užsakymas</i>	<i>sąvoka</i>
8	<i>nuomos periodas</i>	<i>sąvoka</i>

Frazės, atsižvelgiant į sąvokos rolės tipą (r_tipas) įstatomos į šabloną: **nuomos periodas** į pirmosios kompozicijos pradžią, **automobilio kategorija** į pirmąją kompoziciją, už **nuomos periodo** sąvokos, atskiriant nuo jos kableliu, o **užsakymas** į pagrindo vietą. Tokiu būdu atkurta VT:

Nuomos periodas, automobilio kategorija yra užsakymo savybė.

6. Fakto, kurio f_id = 6, Įrašas pateiktas 5.4.16 lentelėje.

5.4.16 lentelė. Fakto Įrašas

f_id	tipas
6	<i>dalyvis – asociacija</i>

Fakto tipas – *dalyvis - asociacija*, jo šablonas: *<pagrindas> gali <veiksmažodis> <pirmoji kompozicija>.*

Lentelėje **Rolė**, nustatomi sąvokos identifikaciniai numeriai, kur $f_id = 6$ ($s_id = 3$, $s_id = 5$ ir $s_id = 9$). Jie pateikiami 5.4.17 lentelėje.

5.4.17 lentelė. Rolės Įrašai

f_id	s_id	r_tipas	r_nr
6	3	<i>pagr.</i>	0
6	5	<i>veiksm..</i>	0
6	9	<i>I komp.</i>	0

Iš lentelės **Sąvoka**, pagal lauką s_id paimami reikalingi lauko frazė Įrašai, t. y. **asmuo**, **pateikti**, **skubus užsakymas**. Jie pateikiami 5.4.18 lentelėje.

5.4.18 lentelė. Sąvokos Įrašai

s_id	frazė	s_poz
3	<i>asmuo</i>	<i>sąvoka</i>
5	<i>pateikti</i>	<i>veiksmazodis</i>
9	<i>skubus užsakymas</i>	<i>sąvoka</i>

Frazės, atsižvelgiant į sąvokos rolės tipą (r_tipas) įstatomos į šabloną: **asmuo** į pagrindo vietą, **pateikti** į veiksmazodžio vietą, o **skubus užsakymas** į pirmosios kompozicijos vietą. Tokiu būdu atkurta VT:

Asmuo gali pateikti skubų užsakymą.

7. Fakto, kurio $f_id = 7$, Įrašas pateiktas 5.4.19 lentelėje.

5.4.19 lentelė. Fakto Įrašas

f_id	tipas
7	<i>apibendrinimas</i>

Fakto tipas – *apibendrinimas*, jo šablonas: *<pirmoji kompozicija> yra <pagrindas>*.

Lentelėje **Rolė**, nustatomi sąvokos identifikaciniai numeriai, kur $f_id = 4$ ($s_id = 7$ ir $s_id = 9$). Jie pateikiami 5.4.20 lentelėje.

5.4.20 lentelė. Rolės Įrašai

f_id	s_id	r_tipas	r_nr
7	9	<i>I komp.</i>	0
7	7	<i>pagr</i>	0

Iš lentelės **Sąvoka**, pagal lauką s_id paimami reikalingi lauko frazė Įrašai, t. y. **užsakymas** ir **skubus užsakymas**. Jie pateikiami 5.4.21 lentelėje.

5.4.21 lentelė. Sąvokos Įrašai

s_id	frazė	s_poz
7	<i>užsakymas</i>	<i>sąvoka</i>
9	<i>skubus užsakymas</i>	<i>sąvoka</i>

Frazės, atsižvelgiant į sąvokos rolės tipą (r_tipas) įstatomos į šabloną: **skubus užsakymas** į pirmosios kompozicijos vietą, o **užsakymas** į pagrindo vietą. Tokiu būdu atkurta VT:

Skubus užsakymas yra užsakymas.

Veiksmo teiginių atkūrimas:

1. Veiksmo teiginio, kurio $v_id = 1$, įrašas pateiktas 5.4.22 lentelėje.

5.4.22 lentelė. Fakto įrašas

v_id	klasė
1	<i>apribojimas</i>

Veiksmo teiginio klasė – *apribojimas*, jo šablonas: *<bazinis objektas> privalo <koresponduojantis objektas>*.

Lentelėje **Objektas**, nustatomi objektų identifikaciniai numeriai, kur $tv_id = 1$ ($s_id = 7$ ir $f_id = 5$). Jie pateikiami 5.4.23 lentelėje.

5.4.23 lentelė. Objekto įrašai

f_id	s_id	v_id	vk_id	tv_id	$o_pož$
-1	7	-1	-1	1	<i>baz. obj.</i>
5	-1	-1	-1	1	<i>koresp. obj.</i>

Apribojimo bazinis objektas yra sąvoka, jis neturi atskiro šablono. Bazinio objekto vietoje iš lentelės **Sąvoka**, pagal lauką s_id paimamas reikalingas lauko frazė įrašas, t. y. **užsakymas**. Jis pateikiamas 5.4.24 lentelėje.

5.4.24 lentelė. Sąvokos įrašas

s_id	frazė	s_poz
7	<i>užsakymas</i>	<i>sąvoka</i>

Apribojimo koresponduojantis objektas yra faktas, todėl iš lentelės **Faktas**, pagal lauką f_id paimamas reikalingas fakto įrašas. Jis pateikiamas 5.4.25 lentelėje.

5.4.25 lentelė. Fakto įrašas

f_id	tipas
5	<i>atributas</i>

Koresponduojančio objekto fakto tipas – atributas, todėl koresponduojančio objekto šablonas: turėti *<pirmosios kompozicijos sąvoka>*.

Lentelėje **Rolė**, nustatomi pirmosios kompozicijos sąvokos identifikacinis numeris, kur $f_id = 5$ ($s_id = 1$ ir $s_id = 8$). Jie pateikiami 5.4.26 lentelėje.

5.4.26 lentelė. Rolės įrašai

f_id	s_id	r_tipas	r_nr
5	1	<i>1 komp.</i>	1
5	8	<i>1 komp.</i>	0

Iš lentelės **Sąvoka**, pagal lauką s_id paimami reikalingi lauko frazė įrašai, t. y. **nuomos periodas** ir **automobilio kategorija**. Jie pateikiami 5.4.27 lentelėje.

5.4.27 lentelė. Sąvokos įrašai

s_id	frazė	s_poz
1	<i>automobilio kategorija</i>	<i>sąvoka</i>
8	<i>nuomos periodas</i>	<i>sąvoka</i>

Frazės, atsižvelgiant į sąvokos rolės numerį (r_nr) įstatomos į koresponduojančio objekto šabloną: **turėti nuomos periodą, automobilio kategoriją**.

Objektai, įstatomi į veiksmo teiginio šabloną: **užsakymas** į bazinio objekto vietą, o **turėti nuomos periodą, automobilio kategoriją** į koresponduojančio objekto vietą. Tokiu būdu atkurta VT:

Užsakymas privalo turėti nuomos periodą, automobilio kategoriją.

2. Veiksmo teiginio, kurio v_id = 2, įrašas pateiktas 5.4.28 lentelėje.

5.4.28 lentelė. Fakto įrašas

v_id	klasė
2	sąlyga

Veiksmo teiginio klasė – *sąlyga*, jo šablonas: jeigu <bazinis objektas>, tai <koresponduojantis objektas>.

Lentelėje **Objektas**, nustatomi objektų identifikaciniai numeriai, kur tv_id = 2 (f_id = 7 ir vk_id = 1). Jie pateikiami 5.4.29 lentelėje.

5.4.29 lentelė. Objekto įrašai

f_id	s_id	v_id	vk_id	tv_id	o_pož
7	-1	-1	-1	2	baz. obj.
-1	-1	-1	1	2	koresp. obj.

Apribojimo bazinis objektas yra faktas, todėl iš lentelės **Faktas**, pagal lauką f_id paimamas reikalingas fakto įrašas. Jis pateikiamas 5.4.30 lentelėje.

5.4.30 lentelė. Fakto įrašas

f_id	tipas
7	apibendrinimas

Bazinio objekto fakto tipas – apibendrinimas, todėl bazinio objekto šablonas: <pagrindo sąvoka> yra <pirmosios kompozicijos sąvoka>.

Lentelėje **Rolė**, nustatomi sąvokos identifikaciniai numeriai, kur f_id = 7 (s_id = 9 ir s_id = 7). Jie pateikiami 5.4.31 lentelėje.

5.4.31 lentelė. Rolės įrašai

f_id	s_id	r_tipas	r_nr
7	9	1 komp.	0
7	7	pagr.	0

Iš lentelės **Sąvoka**, pagal lauką s_id paimami reikalingi lauko frazė įrašai, t. y. **nuomos periodas** ir **automobilio kategorija**. Jie pateikiami 5.4.32 lentelėje.

5.4.32 lentelė. Sąvokos įrašai

s_id	fraze	s_pož
7	užsakymas	sąvoka
9	skubus užsakymas	sąvoka

Frazės, atsižvelgiant į sąvokos rolės tipą (r_tipas) Įstatomos į bazinio objekto šabloną: **užsakymas yra skubus užsakymas**.

Apribojimo koresponduojantis objektas yra veiksmas, jis neturi atskiro šablono. Koresponduojančio objekto vietoje iš lentelės **Veiksmas**, pagal lauką vk_id paimamas reikalingas lauko fraze įrašas, t. y. **pateikti automobilį**. Jis pateikiamas 5.4.33 lentelėje.

5.4.33 lentelė. Veiksmo įrašas

vk_id	fraze
1	<i>pateikti automobilį</i>

Objektai, Įstatomi į veiksmo teiginio šabloną: **užsakymas yra skubus užsakymas** į bazinio objekto vietą, o **pateikti automobilį** į koresponduojančio objekto vietą. Tokiu būdu atkurta VT: **Jeigu užsakymas yra skubus užsakymas, tai pateikti automobilį**.

Sistemos langas su eksperimento metu Įvestomis veiklos taisyklėmis pateikiamas 5 priede.

5.5 Eksperimento rezultatai

Atliktas veiklos taisyklių išsaugojimo ir atkūrimo eksperimentas, kuris parodė kad:

- Sistema realizuoja *struktūrinių* ir *veiksmo teiginių* veiklos taisyklių Įvedimą pagal koncepcinėje dalyje pateiktą struktūrizuotą veiklos taisyklių Įvedimo metodiką. Sistema leidžia Įvesti veiklos taisyklę, nustatyto jos tipą, suskaido VT į logiškai susijusias dalis, kurias galima saugoti atskirai.
- Veiklos taisyklių saugykla realizuota pagal koncepcinėje dalyje pateiktą struktūrą. Sistema išsaugo duomenis saugykloje pagal nustatytą struktūrą, po to teisingai atkuria veiklos taisykles iš duomenų saugyklos.

Sistemos testavimo metu nustatyta, kad šablonai atitinka jiems išskeltus kriterijus:

- pasižymi pakankamai aukštu struktūrizavimo lygiu;
- yra glaudžiai susiję tarpusavyje;
- leidžia Įvesti struktūrinius bei veiksmo teiginius pagal GUIDE modelį.

Sistemos testavimo metu nustatyta, kad saugyklos modelis atitinka VT saugyklos modeliui keliamus reikalavimus:

1. Informacijos išsaugojimo lygis yra pakankamas, kad būtų galima į duomenų bazę Įvesti duomenis apie bet kokio *struktūrinio teiginio* potipio taisyklę bei bet kokio *veiksmo teiginio* poklasio taisyklę, struktūrizuotą pagal GUIDE metodą.

2. Saugyklos duomenų bazėje saugomos informacijos pakanka, norint atkurti visą pradinę VT struktūrą.

3. Saugoma tik ta informacija apie taisyklę, kuri atspindi VT struktūroje. Informacija apie kiekvieno iš VT tipų struktūrą nesaugoma.

Apibendrinant eksperimento rezultatus galima teigti, kad pasiūlyti GUIDE modelio patobulinimai papildė modelį:

1. Modelio apibrėžtumu. Struktūriniai ir veiksmo teiginiai yra griežtai apibrėžti. Analitikui neturi kilti abejonų dėl užregistruotos taisyklės priklausymo vienai ar kitai grupei.
2. Saugojimo modeliu. GUIDE klasifikacija pasižymi unikaliomis savybėmis ir specifine struktūrizavimo metodika, todėl tikslus atitinkamo saugyklos modelio apibrėžimas yra labai svarbus.

6. IŠVADOS

1. Tradicinė programinė įranga nepajėgi pakankamai greitai atspindėti sparčiai besikeičiančios aplinkos, todėl veiklos taisyklių koncepcijos siūlomi sprendimai yra labai aktualūs. Šios koncepcijos esmė – veiklos taisyklių atskyrimas nuo taikomosios programos kodo ir jų įdiegimas, panaudojant atskirą veiklos taisyklių sluoksnį, padarantį taikomąsias programas lengviau pritaikomas ir palaikomas.
2. Veiklos taisyklių koncepcijos literatūros analizė įrodo, kad egzistuoja įvairūs veiklos taisyklių klasifikavimo ir struktūrizavimo modeliai. Šiuose modeliuose didžiausias dėmesys skiriamas taisyklių klasifikacijai, o jų užrašymo formos ir saugojimo galimybės nagrinėjamos mažiau.
3. Literatūroje yra žinomi tokie veiklos taisyklių koncepcijos metodai: Roso, MDC, GUIDE. Pagrindinis Roso metodo privalumas: detali modelio klasifikacija, trūkumai: labai sudėtinga metodologija, nėra aprašyta veiklos taisyklių saugykla. Pagrindinis MDC privalumas: veiklos taisyklių saugojimo metodikos aprašymas, trūkumas: silpna veiklos taisyklių klasifikacija. Pagrindiniai GUIDE privalumai: paprastas, bet detalus klasifikavimas, stipriai išvystyta teorinė bazė, trūkumai: nestruktūrizuota veiklos taisyklių užrašymo forma, nėra aprašyta veiklos taisyklių saugykla.
4. Literatūroje pateiktų veiklos taisyklių koncepcijos metodų atlikta palyginamoji analizė įrodo, kad daugiausiai privalumų turi GUIDE modelis. Jis lenkia kitus modelius paprasta klasifikacija ir stipriai išvystyta teorine baze, nors nuo Roso metodo atsilieka klasifikacijos detalumu, o nuo MDC veiklos taisyklių saugojimo metodikos išvystymu.
5. GUIDE modelį reikia papildyti, kad jį būtų galima pritaikyti praktinėms reikmėms. Šiame darbe buvo pasiūlyti GUIDE modelio patobulinimai, apimantys:
 - Veiklos taisyklių šablonus. Šablonai nustato struktūrizuotą veiklos taisyklės užrašymo formą.
 - Veiklos taisyklių saugyklą. Saugyklos modelis specifikuoja veiklos taisyklės saugojimo metodiką.
6. Šablonus tikslinga panaudoti veiklos taisyklėms užrašyti, nes jie leidžia tiksliai ir nedviprasmiškai užrašyti įvairių tipų taisykles. Turint šablonų pagrindu sudarytą veiklos taisyklių specifikaciją galima lengvai nustatyti veiklos taisyklių tipą ir jas kaupti duomenų saugykloje.

7. Pasiūlyta veiklos taisyklių saugykla saugo šablonais apibrėžtų veiklos taisyklių tipų veiklos taisykles. Saugyklos modelis nustato kaip išsaugoti, o po to atkurti kiekvienos Į veiklos taisyklių saugyklą Įvestos veiklos taisyklės pradinę formą.
8. Siūlomi GUIDE metodo patobulinimai realizuoti kuriant veiklos taisyklių koncepcija pagrįstą sistemą. Sistema realizuoja veiklos taisyklių Įvedimą bei jų išsaugojimą veiklos taisyklių saugykloje pagal patobulintą GUIDE metodiką.
9. Veiklos taisyklių Įvedimo, pagal koncepcinėje dalyje pateiktą struktūrizuotų veiklos taisyklių Įvedimo metodiką, tyrimas Įrodo, kad veiklos taisyklių šablonai yra pakankamai apibrėžti. Analitikui neturi kilti abejonių dėl užregistruotos taisyklės priklausymo vienai ar kitai grupei.
10. Veiklos taisyklių išsaugojimo duomenų saugykloje tyrimas Įrodo, kad saugojimo modelis išsaugo duomenis saugykloje pagal nustatytą struktūrą, po to teisingai atkuria veiklos taisykles iš duomenų saugyklos.
11. Veiklos taisyklės išsaugojimo eksperimentas Įrodo, kad pasiūlytus koncepcinius GUIDE modelio papildymus galima naudoti kuriant realias veiklos taisyklių koncepcija pagrįstas sistemas.
12. Sukurti šablonai ir specifikuota veiklos taisyklių saugykla pristatyti tarpuniversitetinėje magistrantų ir doktorantų konferencijoje „Informacinė visuomenė ir universitetinės studijos 2004“ bei atspausdintas straipsnis „Veiklos taisyklių šablonai struktūriniais teiginiais specifikuoti“ (leidinio viršelio ir straipsnio kopijos pateikiamos 2 priede).

7. LITERATŪRA

- [1] **Baisley D. E.** Categories and Roles in Business Vocabulary//*The Business Rules Community* [interaktyvus]. 2004, [žiūrėta 2004-04-01]. Prieiga per Internetą: <http://www.brcommunity.com/p-b177.php>
- [2] Business Engineering Model. Business Rules. Review draft. *Meta Data Coalition* [interaktyvus]. 1999, [žiūrėta 2003-02-25]. Prieiga per internetą: <http://www.mdcinfo.com/OIM/models/BRM.html>
- [3] **Hay D., Healy K.A.** Defining Business Rules ~ What Are They Really?//*The Business Rules Group* [interaktyvus]. 2000, [žiūrėta 2003-02-18]. Prieiga per internetą: http://www.businessrulesgroup.org/first_paper/BRG-whatBR_3ed.pdf
- [4] **Healy K.A., Eulenberg M.** Response to MDC/Microsoft Business Rules Metadata//*The Business Rules Group* [interaktyvus]. 1999, [žiūrėta 2004-04-08]. Prieiga per internetą: <http://www.businessrulesgroup.org/brg-mdc/BRG-MDC.pdf>
- [5] ILOG JRules. Technical White Paper//ILOG [interaktyvus]. 2002, [žiūrėta 2003-04-06]. Prieiga per internetą: <http://www.ilog.com/products/rules/whitepapers/>
- [6] **Kapočius K., Butleris R.** Struktūrizuotų veiklos taisyklių architektūra//*Informacijos mokslai*. ISSN 1392-0561. 2001, Nr. 17, p. 46 – 56.
- [7] **Kapočius K., Butleris R.** Veiklos taisyklių struktūrizavimo projektuojant informacijos sistemas modelių analizė//*Informacinės technologijos 2001*. ISBN 9955-09-098-7. 2001, p. 298 – 304.
- [8] **Laurikaitis A.** Veiklos taisyklių šablonai struktūriniais teiginiais specifikuoti//*Informacinė visuomenė ir universitetinės studijos*. 2004, p 227 – 232.
- [9] **Motiejūnas L., Butleris R., Kapočius K.** Veiklos taisyklių specifikuojimo ir manipuliavimo modelis//*Informacinės technologijos verslui - 2002*. ISBN 9955-09-195-9. 2002, p 24 – 30.
- [10] **Reeder J.** Templates for Capturing Business Rules//*The Business Rules Community* [interaktyvus]. 2000, [žiūrėta 2004-03-09]. Prieiga per Internetą: <http://www.brcommunity.com/cgi-bin/x.pl/print/p-b056.html>
- [11] **Ross R. G.** Current Thoughts on Expressing Business Rules//*The Business Rules Community* [interaktyvus]. 2000, [žiūrėta 2004-04-12]. Prieiga per Internetą: <http://www.brcommunity.com/cgi-bin/x.pl/print/p-a237.html>
- [12] **Ross R. G.** The BRS Rule Classification Scheme//*The Business Rules Community* [interaktyvus]. 2004, [žiūrėta 2004-04-01]. Prieiga per Internetą: <http://www.brcommunity.com/>
- [13] **Ross R. G.** The Business Rule Book: Classifying, Defining and Modeling Rules//*Business Rules Solutions*. 1997, second edition.

- [14] **Taveter K., Wagner G.** Agent - Oriented Enterprise Modeling Based on Business Rules//*20th Int. Conf. on Conceptual Modeling (ER2001)* . 2001.
- [15] **Vaštakaitė R.** Veiklos taisyklių specifikavimo modelis šablonų pagrindu//*Informacinės technologijos ir mokslų integracija* . ISBN 9955-09-427-3. 2003, p. 129 – 132.

8. TERMINŲ IR SANTRUMPŲ ŽODYNAS

Agregatas – faktas - dalyvis, kuris nusako “įeina į/susideda iš” sąryšį.

Apribojimas (constraint) – 1) teiginys, kuris apibrėžia arba nustato tam tikrą organizacijos veiklos aspektą (kitaip *Veiksmo teiginys*).

2) teiginys, kuris privalo būti teisingas.

Asociacija – faktas - dalyvis, tiesiog atspindintis bet kokį kitą ryšio tipą (ne vaidmuo ir ne agregatas).

Bendroji sąvoka – kasdieninės kalbos žodis, turintis visuotinai priimtą prasmę.

Faktas – ryšys tarp dviejų ir daugiau sąvokų.

Faktas – apibendrinimas yra nurodymas, kad konkreti sąvoka (supertipas) yra vienos ar daugiau kitų sąvokų (potipių) apibendrinimas.

Faktas – atributas yra nurodymas, kad konkreti sąvoka yra kitos sąvokos atributas.

Faktas – dalyvis yra tam tikra sąvoka reiškianti ryšį tarp kitų sąvokų.

F_id – fakto identifikacinis kodas.

Formalus išraiškos tipas (formal expression type) – speciali formali gramatika, naudojama užrašyti formalius taisyklės sakinius.

Formalus taisyklės sakiny (formal rule statement) – sakiny, išreiškiantis vieną nedalomą veiklos gramatiką.

Įgalintuvas – veiksmo teiginys, kuris leidžia arba įgalina koresponduojančio objekto egzistavimą, jeigu egzistuoja bazinis objektas.

IS – informacinė sistema.

Išvestis (derivation) – sakiny, išreiškiantis žinias, kurios savo ruožtu išvedamos iš kitų veiklos žinių.

Išvestinis faktas – tai teiginys, sudarytas iš kitų teiginių.

IT – informacinės technologijos.

Laikmatis – tai veiksmo teiginys, kuris patikrina, leidžia arba uždraudžia, sukuria arba pašalina, atėjus apibrėžtam laiko momentui.

Leidimas – tai teiginys, kurį galima atvaizduoti predikatu “(Tiktai) x gali atlikti y”, kur x paprastai yra aktorius, o y yra veiksmas, kuris gali būti įvykdytas ar atliktas.

Loginė išvada – išvestis, kuri gauna išvestinį faktą panaudojant loginę indukciją arba dedukciją.

Matematinis skaičiavimas – išvestis, kuri gauna išvestinį faktą panaudojant apibrėžtą matematinį algoritmą.

Pirminis faktas – svarbus teiginys.

RUP – Rational Unified Process, IS projektavimo metodas.

Sąlyga – tai teiginys, kad, jei kažkas yra tiesa (pasitvirtina), tai pritaikoma kita veiklos taisyklė.

Sąvoka – tai žodis arba frazė, turinti konkrečią reikšmę veikloje.

Sąvoka – *reikšmė* yra konkretus daikto arba objekto egzempliorius ar reikšmė.

Sąvoka – *tipas* yra įvardinta egzempliorių ar reikšmių aibės abstrakcija.

S_id – sąvokos identifikacinis kodas.

S_poz – sąvokos požymis, nurodantis ar sąvoka yra veiksmažodis, ar sąvoka.

Struktūrinis teiginys (structural assertion) – apibrėžta mintis arba fakto konstatavimas, išreiškiantis tam tikrą organizacijos struktūros aspektą.

UML – Unified Modeling Language, vaizdinio modeliavimo kalba, 1997m. OMG (Object Management Group) patvirtinta standartu

Vaidmuo yra faktas – dalyvis, aprašantis, kaip viena sąvoka tampa veikėju (kita sąvoka), sąveikaudama su savo aplinka.

Veiklos politika (policy) – organizacijos veiklos aspektą apibendrinanti charakteristika.

Veiklos sąvoka – sąvoka, turinti konkrečią reikšmę nustatytame veiklos kontekste.

Veiklos taisyklė (business rule) – teiginys, kuris apibrėžia arba nustato tam tikrą organizacijos veiklos aspektą. Ji pagrindžia organizacijos struktūrą, valdo arba įtakoja organizacijos veiklą (elgesį).

Veiklos taisyklės sakiny (business rule statement) – tai deklaratyvus sakiny, nusakantis veiklai nustatytą struktūrą arba apribojimą.

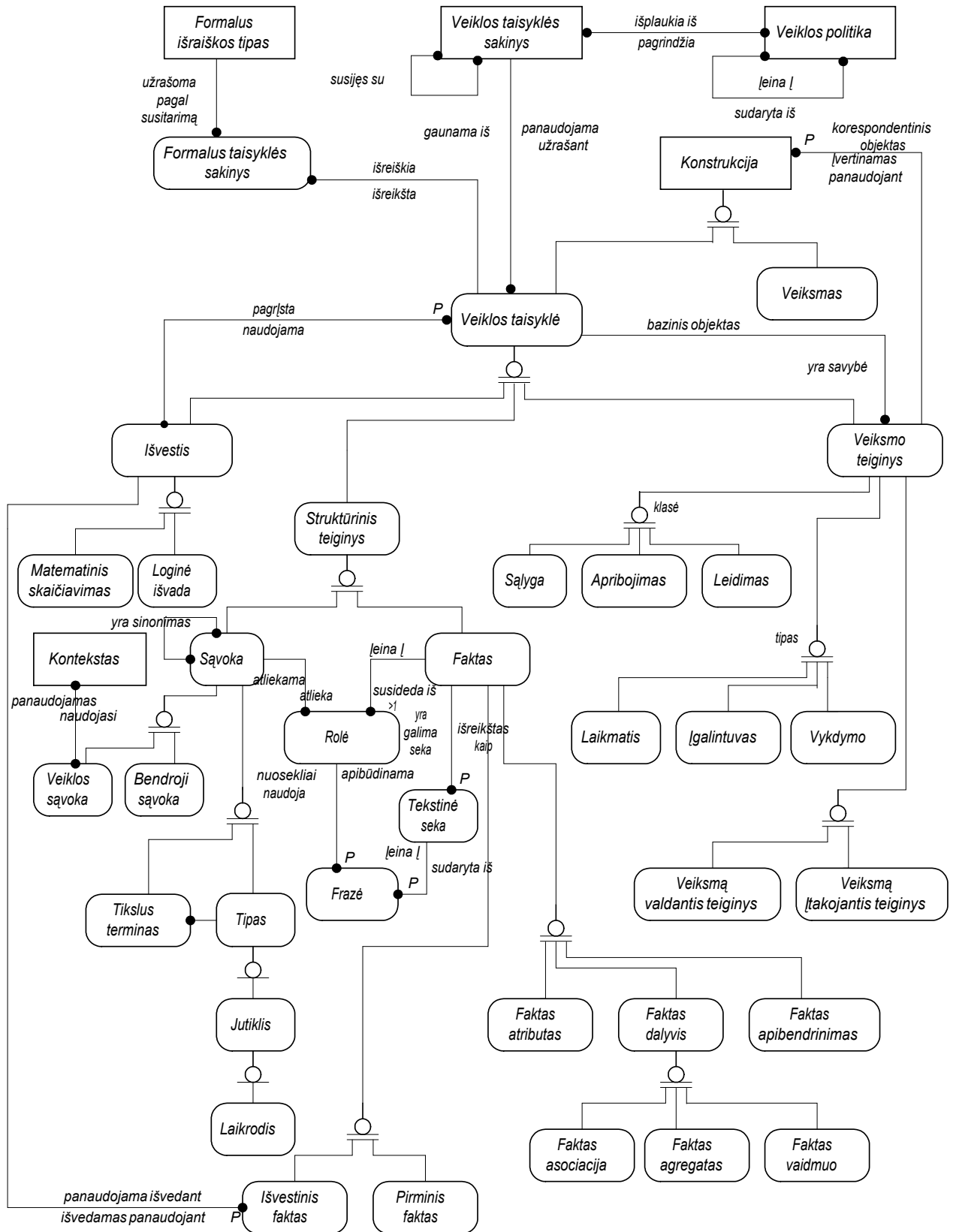
Veiksmo teiginys (action assertion) – apribojimą ar sąlygą reiškiantis sakiny, kuris nustato arba valdo organizacijos veiksmus.

Vykdytojas – veiksmo teiginys, kuris reikalauja (arba iššaukia) vieno arba daugiau veiksmų atlikimą.

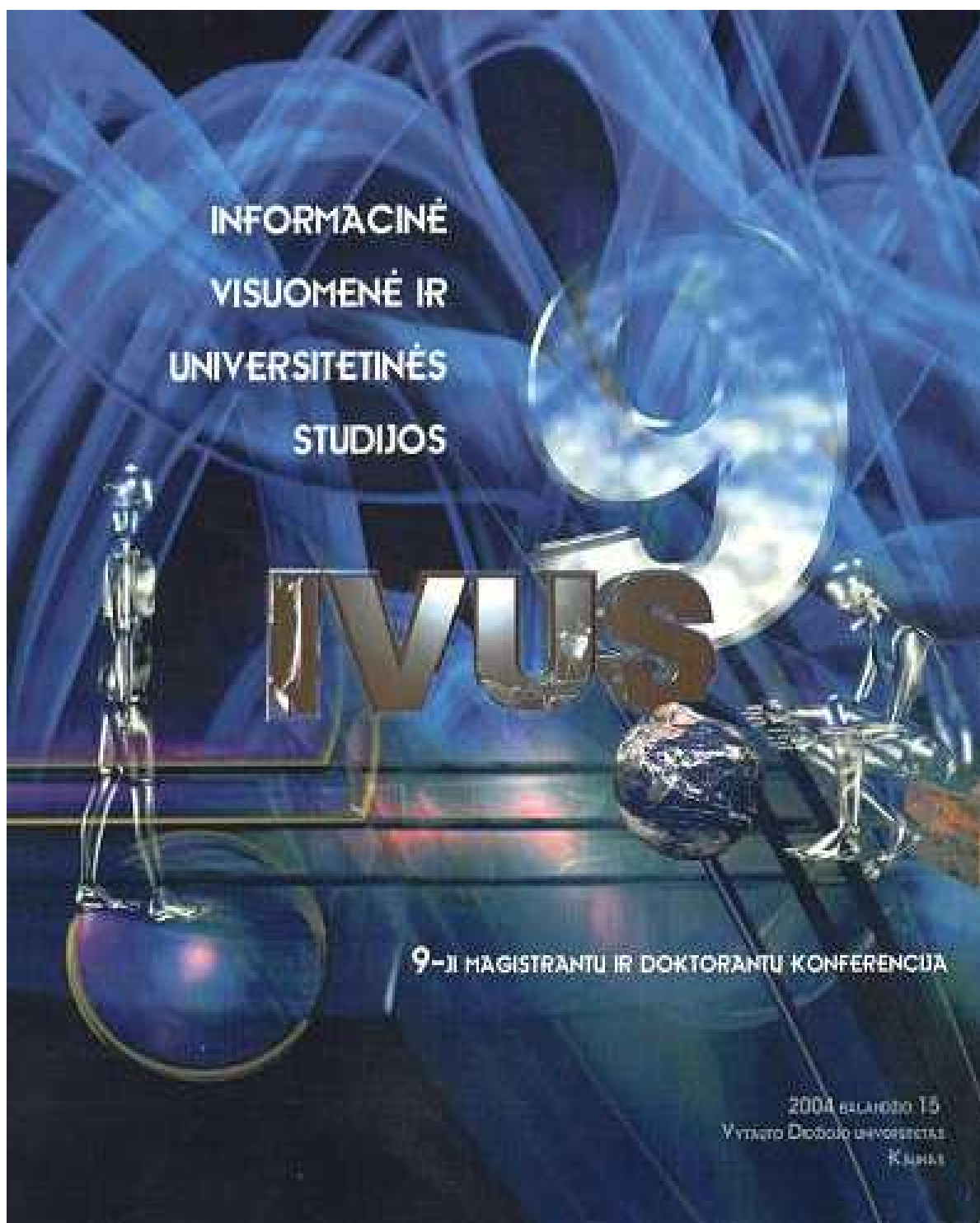
VT – Veiklos taisyklė.

9. PRIEDAI

1 priedas Veiklos taisyklių modelis pagal GUIDE



2 priedas Straipsnis „Veiklos taisyklių šablonai struktūriniais teiginiais specifikuoti“



VEIKLOS TAISYKLIŲ ŠABLONAI STRUKTŪRINIAMS TEIGINIAMS SPECIFIKUOTI

Aurimas Laurikaitis
Rimantas Butleris

Kauno technologijos universitetas Informacijos sistemų katedra

Šiame straipsnyje nagrinėjama veiklos taisyklių šablonų nauda bei jų pritaikymas veiklos taisyklių, klasifikuotų pagal GUIDE metodologiją, užrašymui. Pateikiama keletas veiklos taisyklių šablonų pavyzdžių. Pristatoma saugykla šablonų pagrindu specifikuotiems struktūriniams teiginiams išsaugoti.

1. Įvadas

Veiklos taisyklių (VT) išgavimas, surinkimas ir specifikuojimas tapo įprastas modeliuojant verslą bei projektuojant informacines sistemas [3,7]. Šis procesas būtinas, kad teisingai suprasti veiklos procesus. Egzistuoja keletas išvystytų metodų, kaip surinkti veiklos taisykles, tačiau išsmaus, plačiai paplitusio metodo, kaip fiksuoti surinktas veiklos taisykles visiems priimtina forma trūksta. Šiame straipsnyje nagrinėjamas šablonų panaudojimas veiklos taisyklių fiksavimui, taip pat surinktų veiklos taisyklių saugyklos sudarymas pagal GUIDE metodologiją [2].

2. Šablonų privalumai

Struktūrizuotioms VT užrašyti tikslinga panaudoti šablonus. *Šablonas* – pakartotinai panaudojamas modelis [5], nustatantis gairių, kaip sukurti tam tikrą produktą ar atlikti veiklą, rinkinį. VT fiksavimui naudojamų šablonų privalumai:

- Šablonai yra pirmasis žingsnis formalizuojant VT

VT idėja yra gerai išnagrinėta ir pristatyta, tačiau taisyklių formalizavimas ir fiksavimas vis dar tyrimams ir tobulinimui atvira sritis. Atslūgus pirmajam susižavėjimui, sistemų analitikai ir užsakovai ima suprasti, kad reikalingas nuoseklus priėjimas užrašyti išgautas taisykles. Šablonų panaudojimas suteikia VT fiksavimo procesui darnos ir parodo kelią link technologinių sprendimų.

- Šablonai sumažina užfiksuotų taisyklių informacijos atvaizdavimo prieštaringumą

Geriausiai matoma nauda, susijusi su šablonais, yra prieštaringumo sumažinimas. Visai komandai naudojantis tuo pačiu formatu VT fiksuoti, jos nariai gali pasidalinti taisyklėmis su bendradarbiais, nesirūpindami dėl taisyklių fiksavimo struktūros.

- Šablonai pašalina dviprasmiškumus ir įgalina atlikti VT patvirtinimą (*validation*)

Vienas iš pagrindinių VT metodo privalumų yra patvirtinimo paprastumas. Priešingai duomenų ar proceso modelio diagramoms, VT sąrašas gali būti lengvai naudojamas be papildomų paaiškinimų ar dokumentacijos. Šablonai dar labiau išplečia galimybes, suteikdami neprieštariną naudojamų taisyklių aprašo formą. Be to šablonai, orientuoti į atskirą taisyklę, leidžia patvirtinti kiekvieną iš jų atskirai.

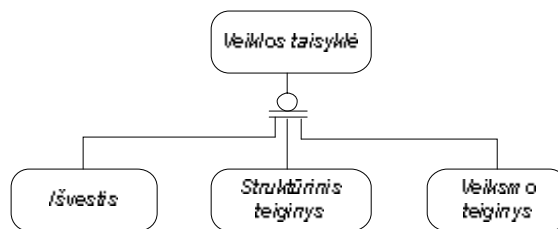
- Sistemos kūrėjams šablonas tarnauja kaip pirminė sistemos specifikuojimo struktūra.

Šablonų neprieštarinumo savybę sistemų kūrėjai gali panaudoti kaip pirminę sistemos specifikuojimo struktūrą. Kadangi kiekviena veiklos taisyklėse naudojama sąvoka yra esybė, atributas ar atributo reikšmė, šablono forma pateiktos taisyklės leidžia žymiai lengviau surasti sprendimus kuriant sistemą.

3. Veiklos taisyklių šablonai

Šablonai turi būti aprašyti kiekvienam klasifikuotam VT tipui. GUIDE modelyje [2] yra išskiriama keletas VT tipų (1 pav.):

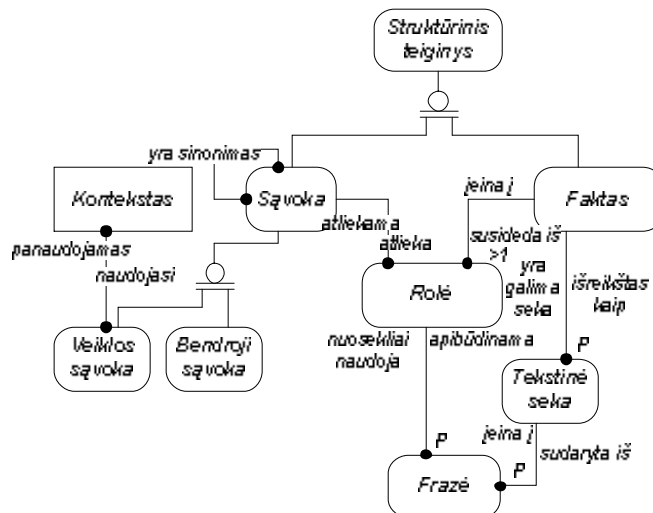
- *Struktūrinis teiginys* – sąvokos apibrėžimas arba fakto formuluotė, kuri išreiškia tam tikrą organizacijos struktūros aspektą.
- *Veiksmo teiginys* – apribojimą arba sąlygą nusakantis sakiny, kuris apriboja arba valdo organizacijos veiksmus.
- *Išvestis* – sakiny, apibūdinantis žinias, kurios yra išvedamos iš kitų veiklos žinių.



1 pav. Veiklos taisyklių tipai

Struktūrinis teiginys nustato veiklai svarbius aspektus, kurie arba egzistuoja kaip koncepcinės esybės, arba siejasi su kitais reikšmingais dalykais. *Struktūrinis teiginys* - tipas, kuris sudaro visi kitų VT tipų pagrindą. Šiame straipsnyje bus detalizuojami šio tipo taisyklių šablonai ir jų specifikacijos saugykla.

Struktūriniai teiginiai skirstomi į *Sąvokas* ir *Faktus* (2 pav.).



2 pav. Sąvokos ir faktai

Sąvoka – tai žodis arba frazė, turinti konkrečią reikšmę veikloje. *Sąvoka* yra elementari VT dalis, kuri nebegali būti išskaidyta į dar mažesnes dalis. *Sąvoka* užrašoma, kaip simbolių seka, išskyrus kitų taisyklių tipų šablonų bazinius žodžius.

Pvz. *Sąvoka Gedimino pilis*. *Sąvoką* sudaro atitinkama seka simbolių, kurie logiškai nebegali būti išskaidyti į atskiras dalis, nes būtų prarandama informacija.

Faktas nusako ryšį tarp dviejų ar daugiau *sąvokų*. *Faktas* apima dvi arba daugiau *sąvokų*, o *sąvoka* gali būti panaudojama viename ir daugiau *faktų*.

Pvz. *Faktas Automobilis yra transporto priemonė* apima dvi *sąvokas*: *Automobilis* ir *Transporto priemonė*. Taip pat *sąvoka* gali būti panaudota keliuose *faktuose*. Pvz. *sąvoka Automobilis* gali būti panaudota ir kitame *fakte Spalva yra automobilio savybė*.

Faktai yra skirstomi į tipus. Kiekvieno tipo *faktai* užrašomi pagal nustatytą šabloną. Šabloną sudaro griežtai nustatyti eiliškumu pateikiami *raktiniai žodžiai* bei *sąvokos*. *Faktas* atkuriamas į šabloną nustatyti eiliškumu įstačius *raktinius žodžius* bei *fakto sąvokas*. Šablono sudėtinės dalys:

- *Raktinis žodis* – kiekvienam VT tipui nustatyta ir užfiksuota alfabетinių simbolių seka, kuri atskiria vienas *fakto sąvokas* nuo kitų. Vienas tipas gali turėti kelis *raktinius žodžius*.
- *Sąvoka* – kintanti *fakto* dalis. Kad būtų paprasčiau apibūdinti įvairių tipų *faktų* šablonus, *sąvokos fakte* skirstomos:
 - *Pagrindas* – *sąvoka*, panaudojama *fakte*. Vienas *fakto* tipas turi tik vieną *pagrindą*, o *pagrindą* sudaro tik viena *sąvoka*.
 - *Pirmoji kompozicija* - *sąvoka*, panaudojama *fakte*. Vienas *fakto* tipas turi tik vieną *pirmąją kompoziciją*, tačiau *kompoziciją* gali sudaryti kelios *sąvokos*.

- *Antroji kompozicija* - sąvoka, panaudojama *fakte*. Vienas *fakto* tipas turi tik vieną *antrąją kompoziciją*, o *kompoziciją* sudaro tik viena sąvoka. *Antroji kompozicija* galima tik tuo atveju, kai *faktas* turi *pirmąją kompoziciją*, tačiau semantine prasme šios dvi sąvokos nesusijusios.
- *Veiksmazodis* - sąvoka, panaudojama *fakte*. Vienas *fakto* tipas turi tik vieną veiksmazodį, o pagrindą sudaro tik viena sąvoka.

Fakto tipai:

- *Atributas* yra nurodymas, kad tam tikra sąvoka yra kitos sąvokos atributas.

Atributo sudėtinės dalys:

1. Sąvoka apibūdinanti kitą sąvoką (*pirmoji kompozicija*);
2. Pirmasis *raktinis žodis* („yra“);
3. Apibūdinama sąvoka (*pagrindas*);
4. Antrasis *raktinis žodis* („savybė“).

Pvz. **Spalva yra automobilio savybė**. Čia **spalva** – *pirmoji kompozicija*, **yra** – pirmasis *raktinis žodis*, **automobilis** – *pagrindas*, **savybė** – antrasis *raktinis žodis*.

- *Apibendrinimas* yra nurodymas, kad tam tikra sąvoka (supertipas) yra vienos ar daugiau kitų sąvokų (subtipų arba potipių) apibendrinimas.

Apibendrinimą sudaro:

1. Apibendrinama sąvoka (*pirmoji kompozicija*);
2. *Raktinis žodis* („yra“);
3. Sąvoka apibendrinanti kitą sąvoką (*pagrindas*).

Pvz. **Automobilis yra transporto priemonė**. Čia **automobilis** – *pirmoji kompozicija*, **yra** – *raktinis žodis*, **transporto priemonė** – *pagrindas*.

- *Dalyvis* yra tam tikra sąvoka, reiškianti ryšį tarp kitų sąvokų. *Faktas* – *dalyvis* dar skirstomas:

- *Agregatas* - *faktas* - *dalyvis*, kuris nusako “įeina į/susideda iš” sąryšį.

Agregatą sudaro:

1. Detalizuojama sąvoka (*pagrindas*);
2. Pirmasis *raktinis žodis* („susideda“);
3. Antrasis *raktinis žodis* („iš“);
4. Sąvoka detalizuojanti kitą sąvoką (*pirmoji kompozicija*).

Pvz. **Kompiuteris susideda iš sisteminio bloko, vaizduoklio, pelės, klaviatūros**. Čia **kompiuteris** – *pagrindas*, **susideda** – pirmasis *raktinis žodis*, **iš** – antrasis *raktinis žodis*, **sisteminis blokas, vaizduoklis, pelė, klaviatūra** – *pirmoji kompozicija*, sudaryta iš keturių sąvokų.

- *Vaidmuo* - *faktas* - *dalyvis*, aprašantis, kaip viena sąvoka tampa veikėju (kita sąvoka), sąveikaudama su savo aplinka.

Vaidmenį sudaro:

1. Sąveikaujanti sąvoka (*pagrindas*);
2. Pirmasis *raktinis žodis* („gali“);
3. Antrasis *raktinis žodis* („būti“);
4. Sąvoka – veikėjas (*antroji kompozicija*);
5. Sąvoka – veiksmazodis;
6. Sąvoka – veiksmas (*pirmoji kompozicija*).

Pvz. **Asmuo gali būti bedarbis, netekęs darbo**. Čia **asmuo** – *pagrindas*, **gali** – pirmasis *raktinis žodis*, **būti** – antrasis *raktinis žodis*, **bedarbis** – *antroji kompozicija*, **netekęs** – veiksmazodis, **darbas** – *pirmoji kompozicija*.

- *Asociacija - faktas - dalyvis*, tiesiog atspindintis bet kokį kitą ryšio tipą (ne vaidmuo ir ne agregatas).

Asociaciją sudaro:

1. Sąveikaujanti *sąvoka* (*pagrindas*);
2. *Raktinis žodis* („gali“);
3. *Sąvoka – veiksmožodis*;
4. *Sąvoka – veiksmas* (*kompozicija*).

Pvz. **Asmuo gali nusipirkti automobilį**. Čia **asmuo** – *pagrindas*, **gali** – *raktinis žodis*, **nusipirkti** – *veiksmožodis*, **automobilis** – *kompozicija*.

4. Veiklos taisyklių saugyklos modelis

Turint šablonų pagrindu sudarytą VT specifikaciją, VT galima kaupti duomenų saugykloje, kur jomis galima lengvai manipuliuoti [6]. Kiekviena VT suformuojama iš saugykloje išsaugotų jos sudėtinių dalių ir tipo raktinių žodžių.

Sąvokai saugoti naudojama lentelė **Sąvoka** (3 pav.), kurios atributai: **s_id** – unikalus identifikacinis *sąvokos* kodas; **frazė** – *sąvoką* nusakantis žodis arba frazė.

Sąvoka
s_id frazė

3 pav. Lentelė **Sąvoka**

Sąvokos lentelę taip pat reikia papildyti *sąvokos* tipo atributu, nes pagal GUIDE veiklos taisyklių modelį [2] *sąvokos* skirstomos:

- *Veiklos sąvokos* – žodžiai ar frazės, kurios turi tam tikrą reikšmę nurodytame veiklos kontekste.
- *Bendrosios sąvokos* – kasdienės kalbos žodžiai, vartojami pagal visuotinai priimtą reikšmę. Jos priimamos kaip aksiominės, kad išvengtų ciklinių apibrėžimų.

Veiklos sąvokos skiriasi nuo *bendrujų sąvokų* tuo, kad jos aiškiai ir tiksliai apibrėžiamos išreiškiant jas vienu ar keliais *faktais*. *Bendrujų sąvokų* prasmė paprastai yra visiems suprantama ir nereikalauja tikslų apibrėžimų.

Sąvokos tipui išsaugoti lentelėje **Sąvoka** (4 pav.) yra pridamas atributas **vs_poz** - *sąvokos* tipas, kuris gali įgyti tokias reikšmes:

- *Veiklos sąvoka*, jei *sąvoka* apibrėžiama išreiškiant ją bent vienu *faktu*;
- *Bendroji sąvoka*, jei *sąvokos* neapibrėžia nei vienas *faktas*.

Sąvoka
s_id frazė vs_poz

4 pav. Modifikuota lentelė **Sąvoka**

Pvz. Turime veiklos taisyklę **Automobilis yra transporto priemonė**. Specifikacijoje reikia išsaugoti dvi sąvokas: **Automobilis**, **transporto priemonė**. Pirmoji *sąvoka* (**automobilis**) yra apibrėžiama panaudojant antrąją *sąvoką* (**transporto priemonė**), todėl ji yra *Veiklos sąvoka*. Antroji *sąvoka* (**transporto priemonė**) šiuo atveju yra bendroji *sąvoka*, tačiau jeigu bent vienas *faktas* apibrėžtų šią *sąvoką*, ji taptų *Veiklos sąvoka*.

Faktui saugoti yra naudojama lentelė **Faktas** (5 pav.), kurios atributas: **f_id** – unikalus identifikacinis *fakto* kodas.

Faktas
f_id

5 pav. Lentelė Faktas

Fakta sudaro *savokų rolės*, kurios aprašo *fakte* dalyvaujančių *savokų* atliekamą semantinį vaidmenį. Kiekviena *savoka* gali atlikti vieną ar daugiau *savokų rolių*, o kiekviena *savokų rolė* turi panaudoti vieną *savoką* viename *fakte*.

Pvz. *Faktas* **Automobilis yra transporto priemonė** turi dvi *savokų roles*, kurios aprašo *savokų* **Automobilis** ir **Transporto priemonė** semantinį vaidmenį *fakte*. *Savokų rolėms* saugoti yra naudojama lentelė **Rolė** (6 pav.), kurios atributai: **f_id** – unikalus identifikacinis *fakto* kodas, **s_id** – unikalus identifikacinis *savokos* kodas.

Role
f_id
s_id

6 pav. Lentelė Rolė

Fakta dažnai galima išreikšti keliais būdais. Netgi dvi *savokos* apimantis *faktas* gali būti nusakomas mažiausiai dviem būdais, priklausomai nuo *savokų rolių* vietos *fakte*. Pvz. *fakta* **Automobilis yra transporto priemonė**, sukeitus *savokas* vietomis, galima išreikšti taip: **Transporto priemonė yra automobilis**. Šiuos abu *faktus* sudaro tos pačios *savokos*, tačiau jų semantinė prasmė yra visiškai skirtinga. Veiklos taisyklių modelyje tai atvaizduojama *tekstine seka*. Kiekviena *tekstinė seka* sudaroma iš *frazių*, kuriose nuosekliai panaudojamos *savokų rolės*. Kad išsaugotume *tekstines sekas* ir *frazes* reikalingos dvi naujos lentelės. Tačiau modifikavus lentelę **Rolė** (7 pav.) ir pridėjus du papildomus laukus **r_tipas** – rolės tipas, nusakantis *savokos* vietą *fakte*, **r_nr** – *savokos* eilės numeris tarp to paties tipo *savokų*, galima supaprastinti VT saugyklos struktūrą.

Role
f_id
s_id
r_tipas
r_nr

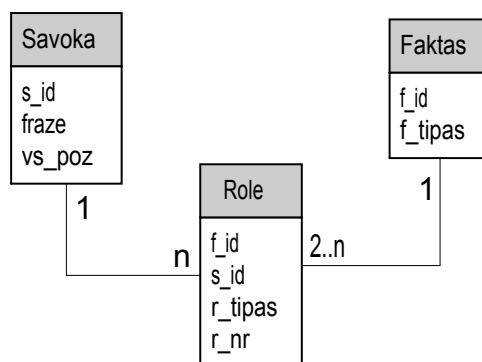
7 pav. Modifikuota lentelė Rolė

Faktai yra skirstomi į tipus. Kad šie tipai būtų atspindėti, lentelėje *Faktas* (8 pav.) reikia pridėti lauką **tipas**.

Faktas
f_id
f_tipas

8 pav. Modifikuota lentelė Faktas

Visas saugyklos modelis, išsaugantis *struktūrinių teiginių* tipo VT, pagal GUIDE veiklos taisyklių metodologiją yra pavaizduotas 9 pav.



9 pav. Saugyklos modelis

Siūloma veiklos taisyklių saugykla yra skirta saugoti tik *struktūrinių teiginių* tipo VT. Saugyklos modelį galima nesunkiai praplėsti įtraukiant naujas lenteles bei atributus, kurie apimtų ir kitų VT tipus. Kuriant VT saugyklos modelį buvo atsisakyta įvairios taisyklę identifikuojančios informacijos, kuri nėra tiesiogiai susijusi su pačios taisyklės semantinės reikšmės išsaugojimu. Realioje sistemoje saugykloje galėtų būti saugomi papildomi duomenys [1,4], tokie kaip informacija apie VT galiojimo datą, modifikavimo istoriją ir pan.

5. Išvados

Žinomuose VT struktūrizavimo modeliuose paprastai daugiau dėmesio skiriama taisyklių klasifikacijai, o jų užrašymo formos ir saugojimo galimybės nagrinėjamos menkai. Ne išimtis yra ir GUIDE veiklos taisyklių metodas, kurio autoriai taisyklių formalaus aprašymo, o kartu ir VT saugyklos sukūrimo galimybių nenagrinėja. Šiame darbe pasiūlyta GUIDE metodui pritaiktų šablonų panaudojimo galimybės bei šablonų pagrindu specifikuotų VT saugyklos architektūra.

Literatūros sąrašas

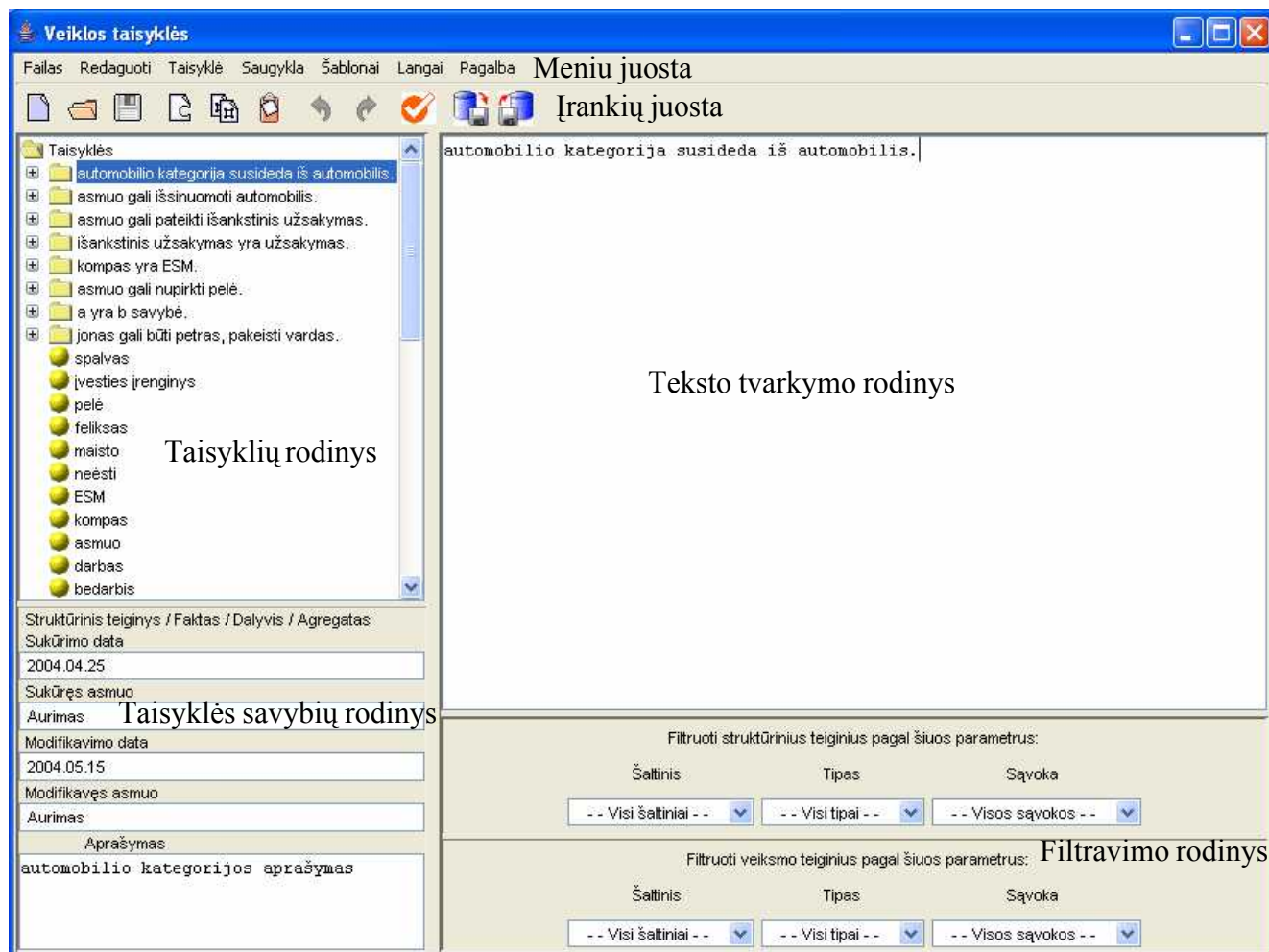
- [1] **Butleris R., Kapočius K.** Struktūrizuotų veiklos taisyklių architektūra//*Informacijos mokslai* . ISSN 1392-0561. 2001, Nr. 17, p. 46 – 56.
- [2] **Hay D., Healy K.A.** Defining Business Rules ~ What Are They Really?//*The Business Rules Group* [interaktyvus]. 2000, [žiūrėta 2003-02-18]. Prieiga per internetą: http://www.businessrulesgroup.org/first_paper/BRG-whatBR_3ed.pdf
- [3] **Kapočius K.** Veiklos taisyklėmis grindžiamo poreikių specifikuavimo principai//*Informacinės technologijos 2004* . ISBN 9955-09-588-1. 2004, p 189 – 195.
- [4] **Motiejūnas L., Butleris R., Kapočius K.** Veiklos taisyklių specifikuavimo ir manipuliavimo modelis//*Informacinės technologijos verslui - 2002* . 2002, p 24 – 30.
- [5] **Reeder J.** Templates for Capturing Business Rules//*The Business Rules Community* [interaktyvus]. 2000, [žiūrėta 2004-03-09]. Prieiga per Internetą: <http://www.brcommunity.com/cgi-bin/x.pl/print/p-b056.html>
- [6] **Vaštakaitė R.** Veiklos taisyklių specifikuavimo modelis šablonų pagrindu//*Informacinės technologijos ir mokslų integracija* . ISBN 9955-09-427-3. 2003, p. 129 – 132.
- [7] **Wilson K.D.** Implementing Business Rules with Inferencing//*The Business Rules Community* [interaktyvus]. 1999, [žiūrėta 2003-02-26]. Prieiga per Internetą: <http://www.brcommunity.com/cgi-bin/x.pl/print/p-a422a.html>

Using Business Rules Templates for Action Assertion Specification Process

This article discusses the benefits of using Business Rule templates and their appliance for Business Rules recording, which are classified using GUIDE methodology. Several examples of templates for various classifications of Business Rules are provided. The repository schema for action assertion rules, which are specified using templates, is presented

3 priedas Veiklos taisyklių sistemos pagrindinis langas

Sistemos pagrindinis langas pateiktas 9.3.1 pav.



9.3.1 pav. Sistemos pagrindinis langas

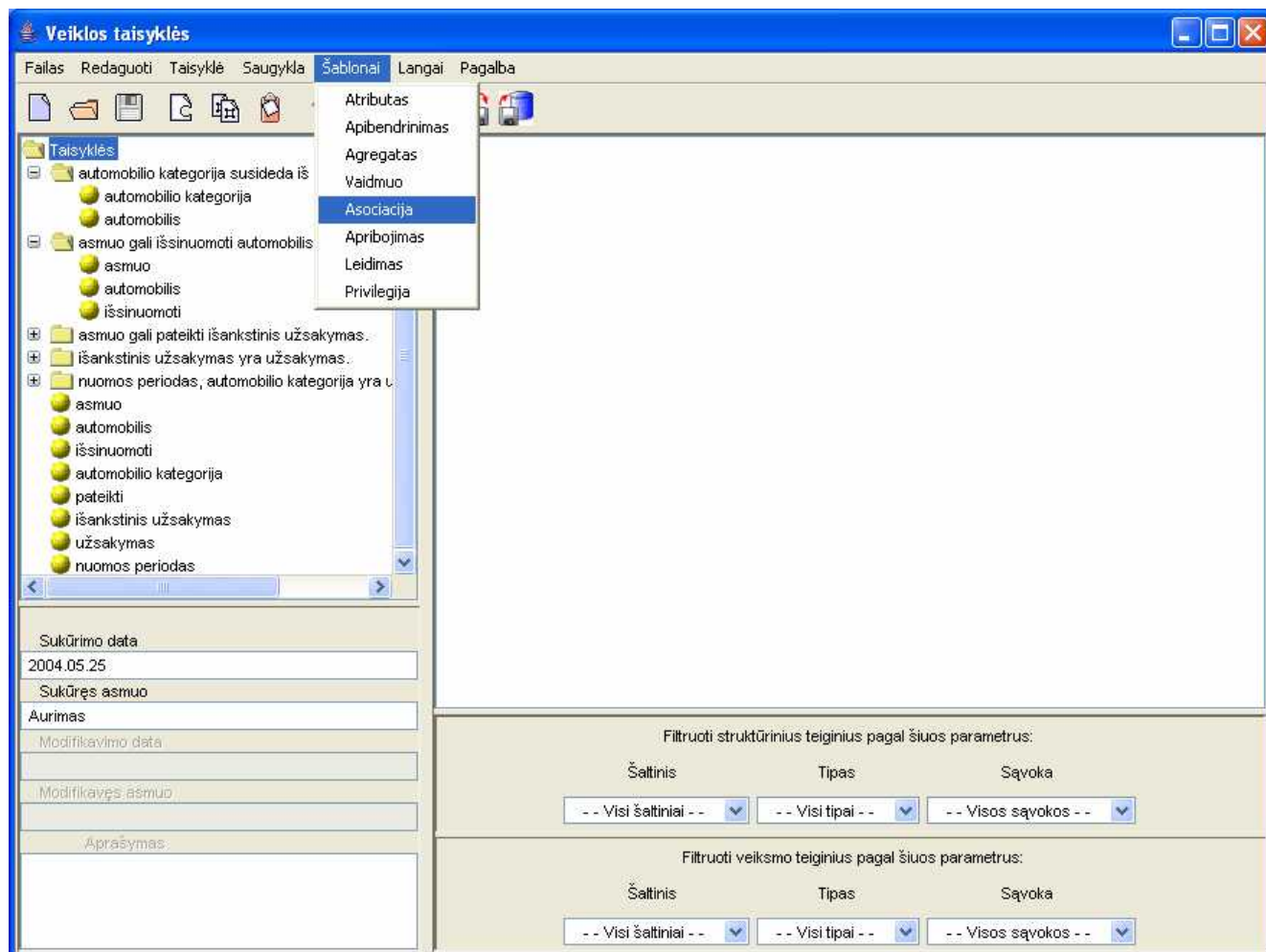
Sistemos pagrindinio lango sudėtinės dalys:

- *Meniu juosta* yra skirta atidaryti arba paleisti tam tikras sistemos funkcijas.
- *Įrankių juosta* yra skirta greitesniam ir patogesniam svarbiausių sistemos funkcijų paleidimui.
- *Taisyklių rodinys* yra išvedamas sistemoje esančių taisyklių sąrašas. Čia galima peržiūrėti sąvokas, faktus ir veiksmo teiginius, pasirinkti redagavimui taisyklę, peržiūrėti taisyklės sudedamąsias dalis.
- *Taisyklės savybių rodinys* yra pateikiamos taisyklių rodinys pasirinktos taisyklės savybės: *Sukūrimo data*, *Sukūręs asmuo*, *Modifikavimo data*, *Modifikavęs asmuo*, *Aprašymas*.
- *Teksto tvarkymo rodinys* yra skirtas taisyklės įvedimui, peržiūrėjimui ir redagavimui.
- *Filtravimo rodinys* leidžia filtruoti taisyklių rodinys išvestas veiklos taisykles pagal šaltinį, tipą arba jas sudarančias sąvokas.

4 priedas Veiklos taisyklės išsaugojimas sistemoje

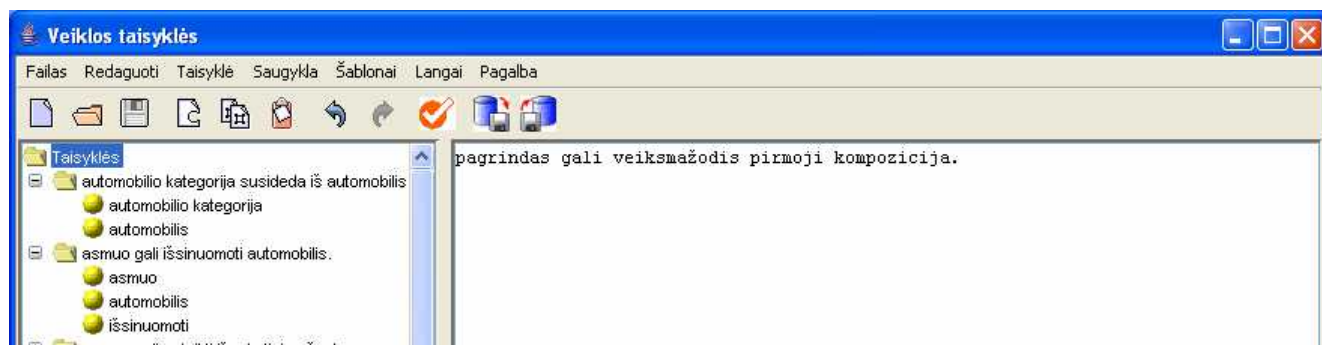
Tarkime, kad reikia įvesti veiklos taisyklę: **Asmuo gali pateikti skubų užsakymą**. Ši taisyklė yra Struktūrinis teiginys / Faktas / Dalyvis / Asociacija.

1. Reikiamo šablono pasirinkimas. Asociacijos šablonas pasirenkamas iš meniu punkto *Šablonai* (9.4.1 pav.). Gerai išanalizavus galimus taisyklių šablonus šį žingsnį galima praleisti ir pereiti iš karto prie 3 žingsnio.



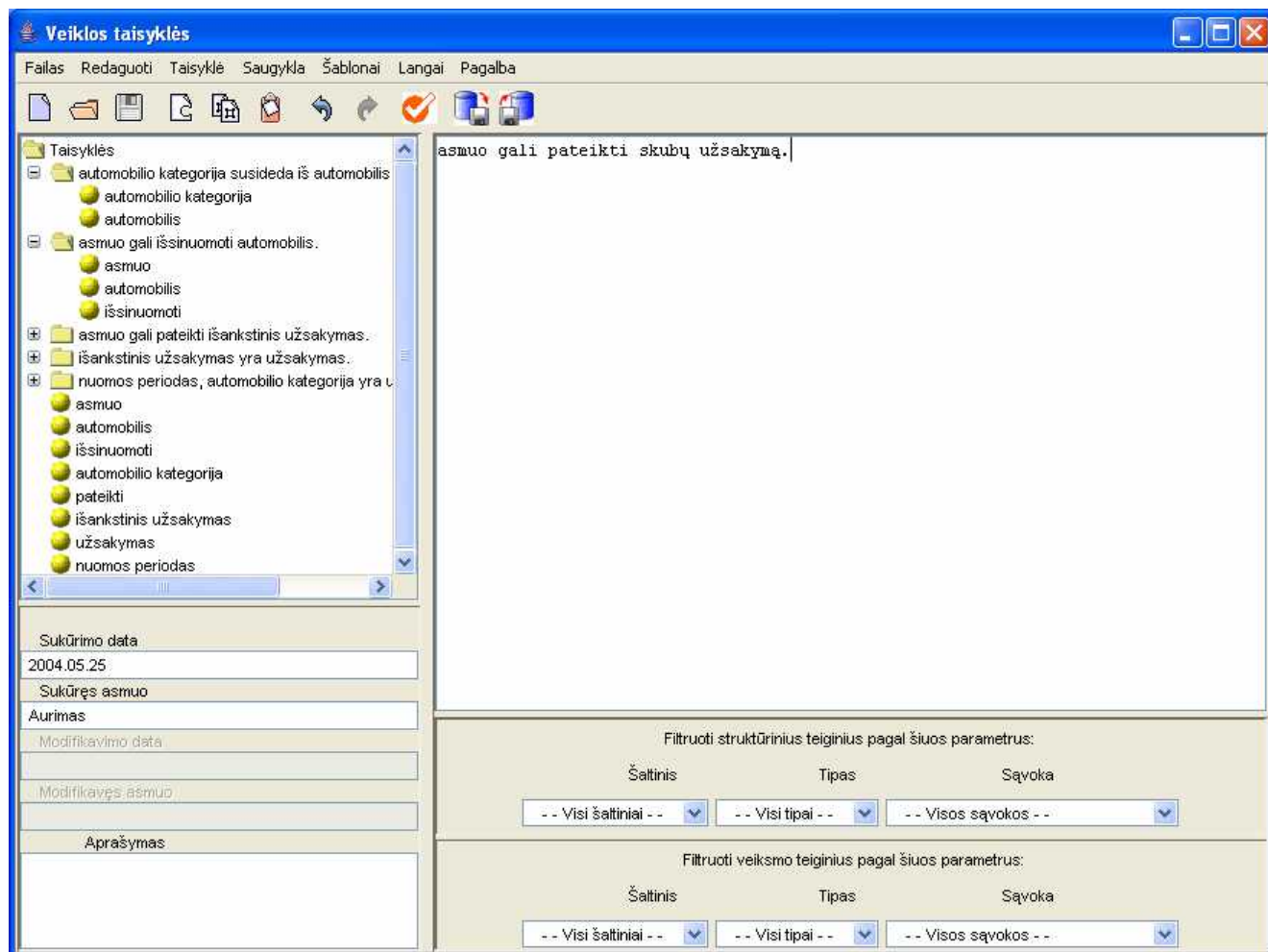
9.4.1 pav. Šablono pasirinkimas

2. Šablono teksto redagavimas. Pasirinkto šablono (9.4.2 pav.) kintamos dalys modifikuojamos: pagrindas, veiksmožodis ir pirmoji kompozicija turi būti pakeisti į konkrečias sąvokas.



9.4.2 pav. Asociacijos šablonas

3. Teksto tvarkymo rodyneje užrašyta taisyklė pateikta 9.4.3 pav.



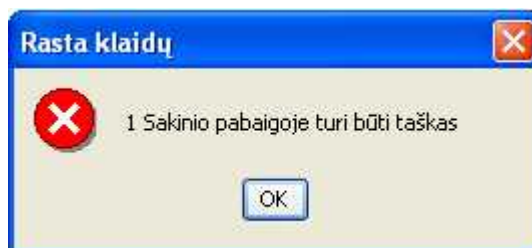
9.4.3 pav. Teksto tvarkymo rodyneje užrašyta taisyklė

4. Taisyklės patikrinimas. Užrašius taisyklę, ji turi būti patikrinta. Patikrinimas atliekamas paspaudus tikrinimo mygtuką (9.4.4 pav.).

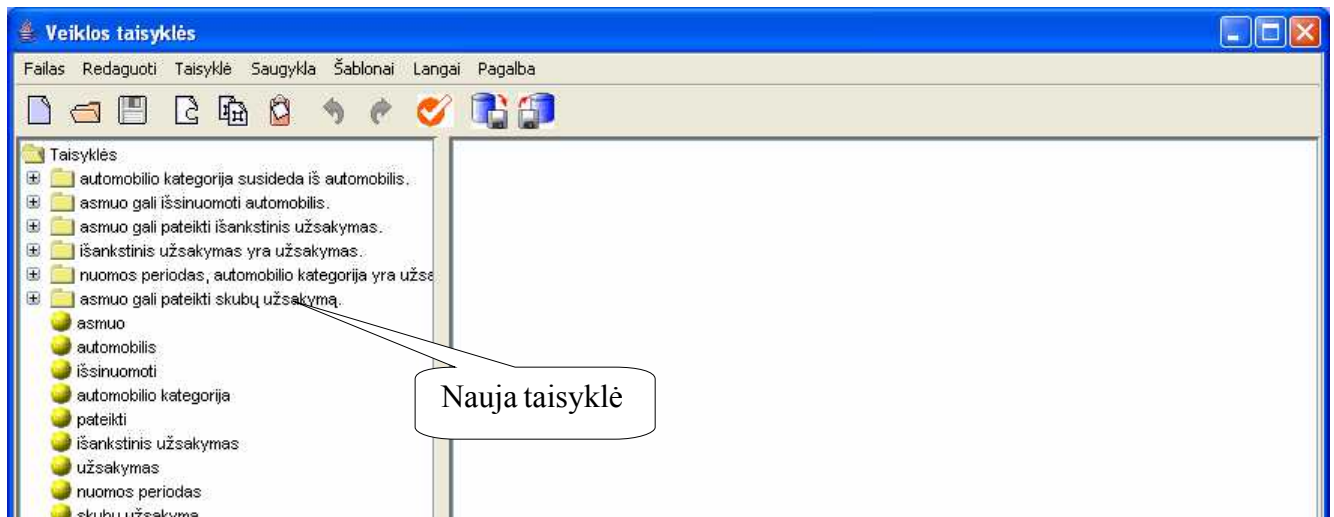


9.4.4 pav. Taisyklės tikrinimo mygtukas

5. Jeigu tikrinant taisyklę buvo rasta sintaksinių klaidų – išvedamas pranešimas apie klaidą (9.4.5 pav.), jeigu taisyklė užrašyta teisingai, taisyklė įvedama į sistemą (9.4.6 pav.).

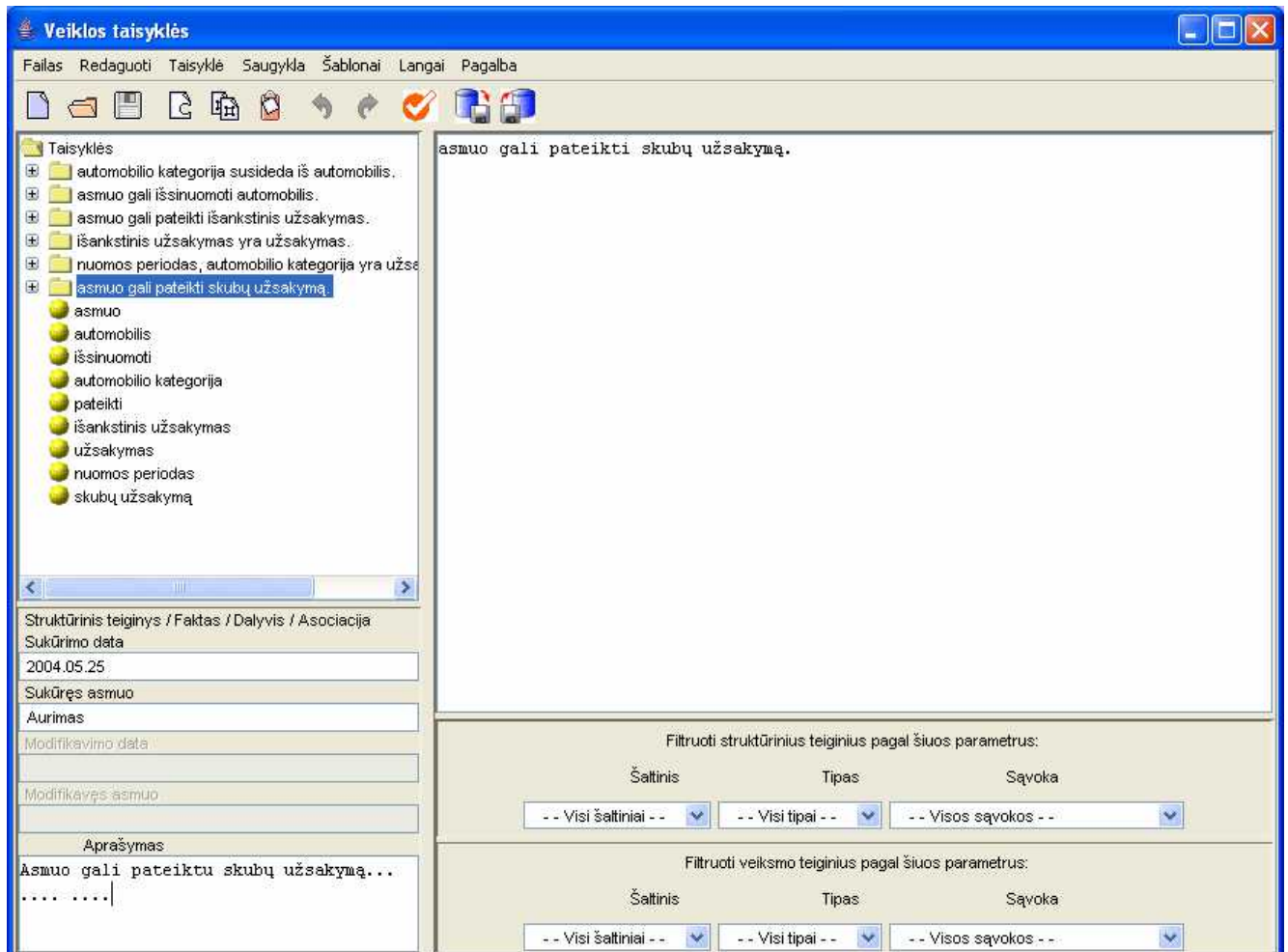


9.4.5 pav. Galimas klaidos pavyzdys



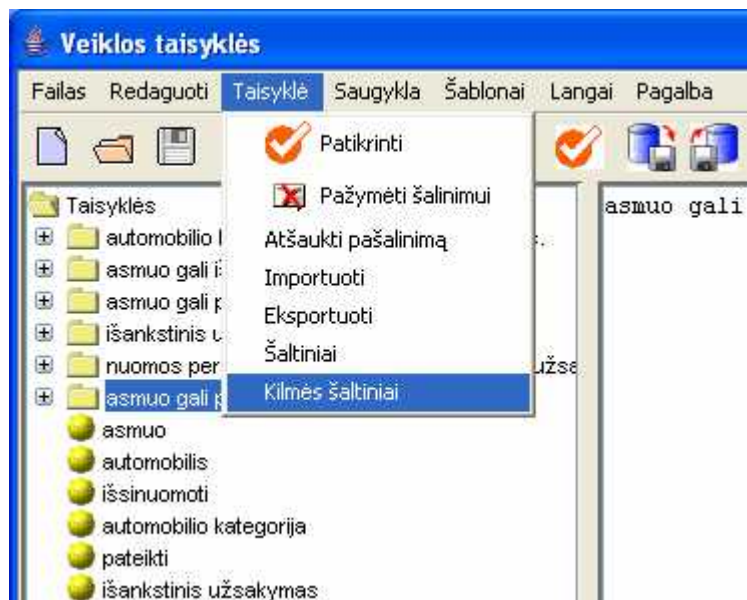
9.4.6 pav. Naujai įvesta taisyklė

6. Papildomų duomenų įvedimas (9.4.7 pav.). Jeigu netenkina papildomi taisyklę apibūdinantys duomenys, taisyklę pažymėjus, juos galima pakeisti (pvz. pridėti taisyklės tekstinį aprašymą).



9.4.7 pav. Papildomų duomenų įvedimas

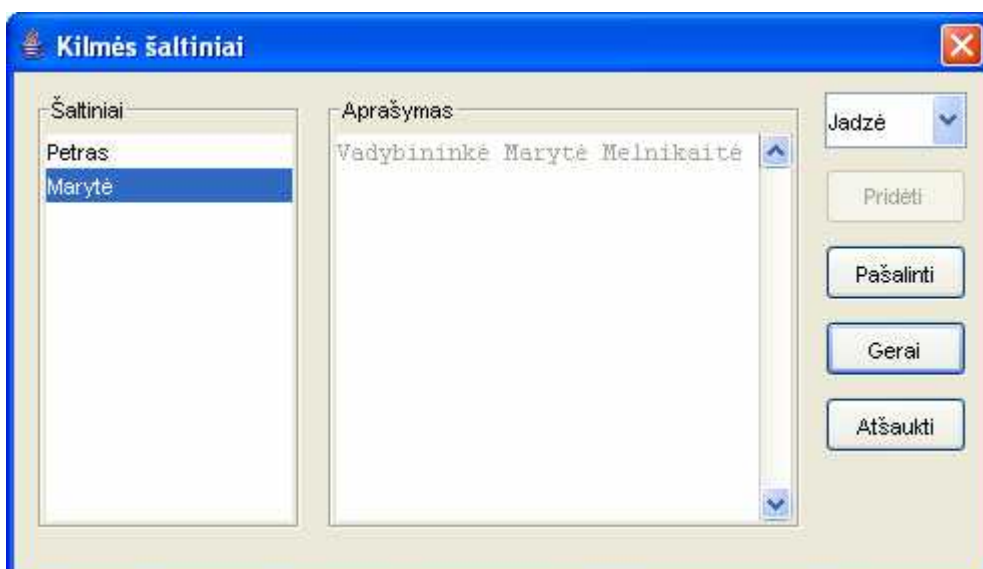
7. Taisyklės informacijos šaltinio pasirinkimas (9.4.8 – 9.4.10 pav.). Taisyklei galima priskirti informacijos šaltinį, vėliau naudojant filtrą galima peržiūrėti tik konkrečiam šaltiniui priskirtas veiklos taisykles.



9.4.8 pav. Kilmės šaltinių įvedimo pasirinkimas

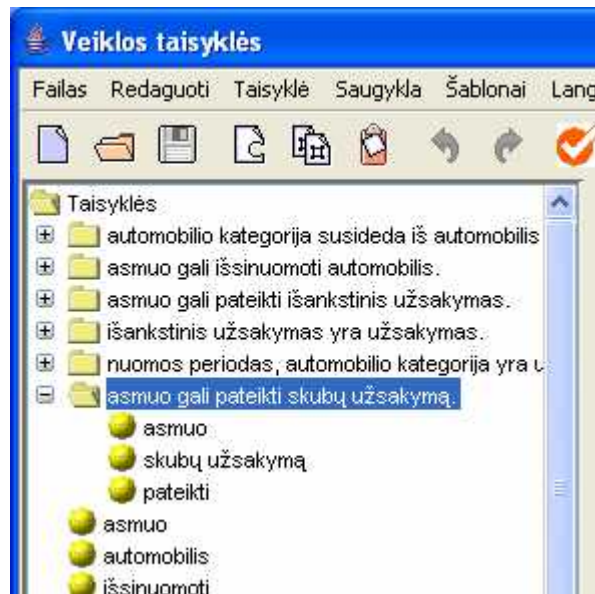


9.4.9 pav. Tuščias kilmės šaltinių langas

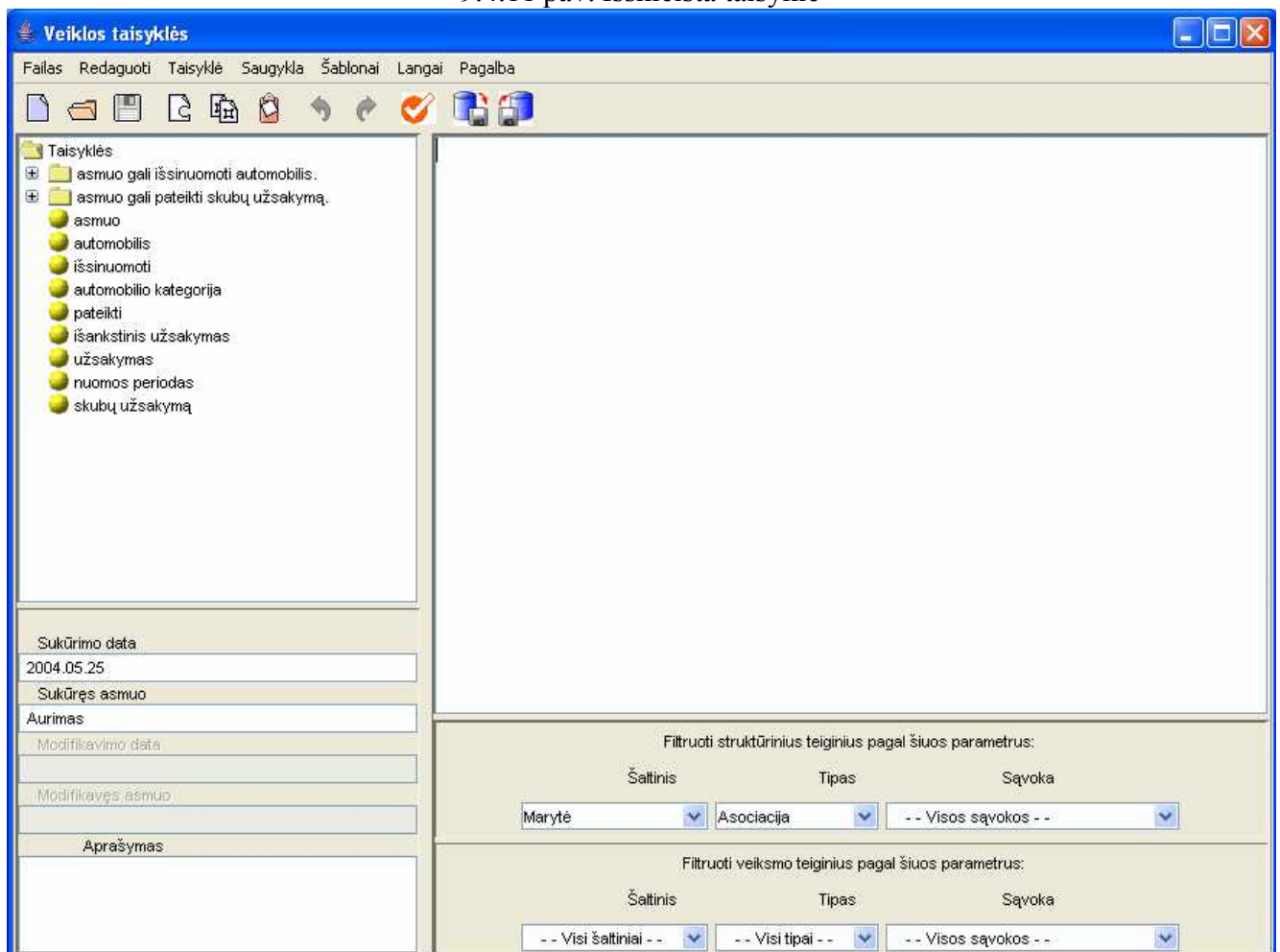


9.4.10 pav. Taisyklei priskirti kilmės šaltiniai

7. Manipuliavimas veiklos taisyklių rinkiniu. Taisyklės galima pakeisti, pašalinti, eksportuoti, importuoti, peržiūrėti taisyklės sudedamąsias dalis (9.4.11 pav.). Galima peržiūrėti tik tas taisyklės, kurios atitinka kriterijus, nurodytus filtravimo rodinyje (9.4.12 pav.).



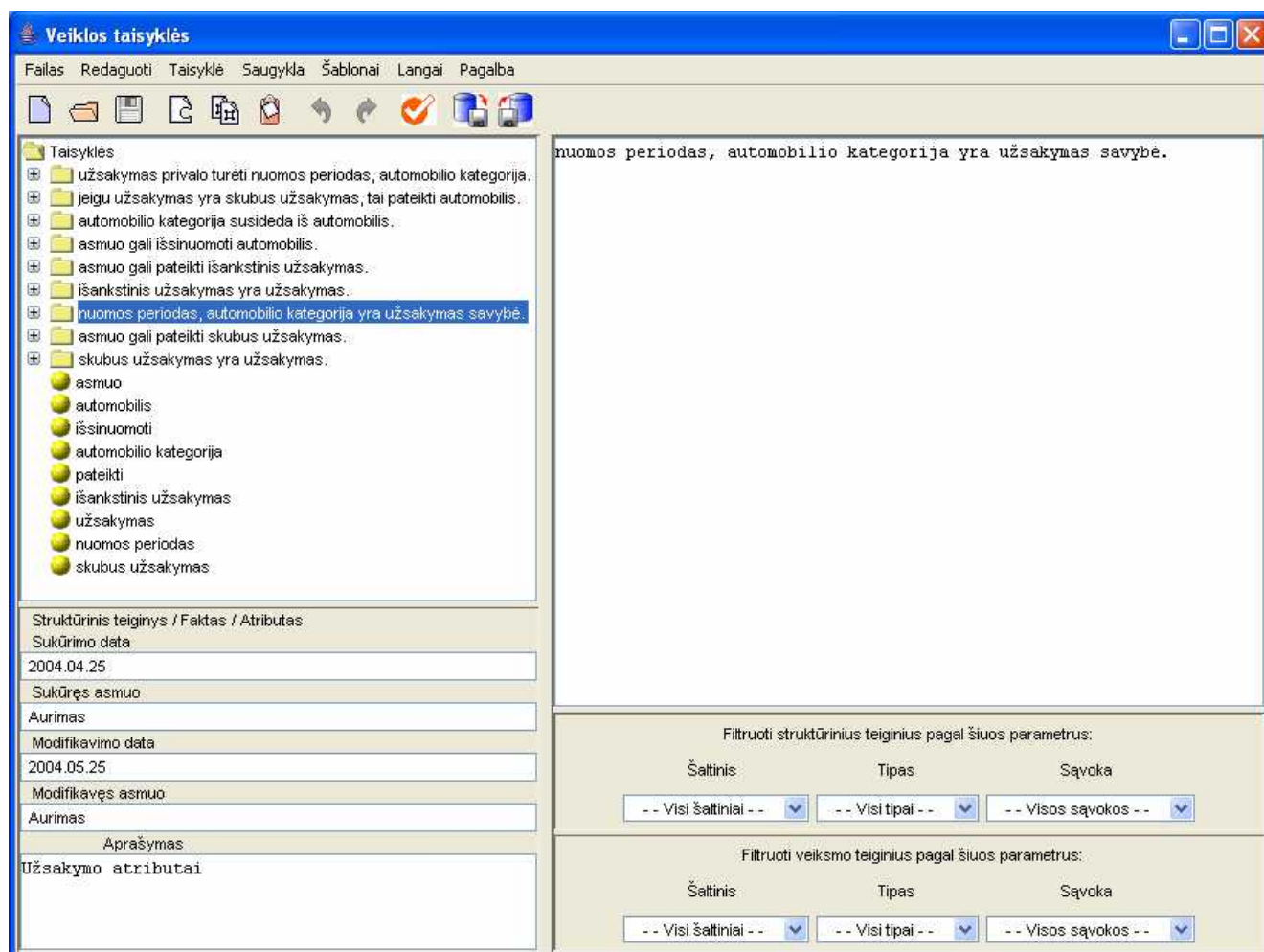
9.4.11 pav. Išskleista taisyklė



9.4.12 pav. Įjungtas filtras pagal šaltinį ir tipą

5 priedas Eksperimento metu į sistemą įvestos veiklos taisyklės

Eksperimento metu į sistemą įvestos veiklos taisyklės pateiktos 9.5.1 pav.



9.5.1 pav. Eksperimento metu į sistemą įvestos taisyklės