

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA

Saulius Baškevičius

**Veiklos taisyklių struktūrizavimo pagal modifikuotą  
R. Roso metodą tyrimas**

Magistro darbas

Darbo vadovas  
doc. dr. R. Butleris

Kaunas, 2006

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA

Saulius Baškevičius

**Veiklos taisyklių struktūrizavimo pagal modifikuotą  
R. Roso metodą tyrimas**

Magistro darbas

Kalbos konsultantė

Lietuvių k. katedros  
doc. dr. J. Mikelionienė  
2006-05

Vadovas

doc. dr. R. Butleris  
2006-05

Recenzentas

doc. Regina Misevičienė  
2006-05

Atliko

IFM-0/2 gr. stud.  
Saulius Baškevičius  
2006-05

Kaunas, 2006

# Turinys

1	Įvadas .....	6
2	Veiklos taisyklių struktūrizavimo analizė.....	8
2.1	Veiklos taisyklių samprata.....	8
2.2	Veiklos taisyklių motyvacinis modelis .....	9
2.3	Veiklos taisyklių kategorijos .....	9
2.4	Roso metodas .....	10
2.5	Veiklos taisyklių formalizavimas pagal Roso metodą.....	14
2.6	Struktūrizuotų veiklos taisyklių saugyklos architektūra.....	17
2.6.1	Veiklos taisyklių koncepcijos prasmė.....	17
2.6.2	Veiklos taisyklių saugyklos vaidmuo informacinėje sistemoje.....	18
2.6.3	Pagal Roso metodą formalizuotų veiklos taisyklių saugyklos modelis.....	20
3	Veiklos taisyklių struktūrizavimo modelis .....	23
3.1	Panaudojimo atvejai.....	24
3.1.1	VT modelio įkėlimas iš saugyklos.....	24
3.1.2	Papildomų duomenų įvedimas.....	25
3.1.3	VT išsaugojimas saugykloje .....	26
3.1.4	VT redagavimas .....	27
3.2	Duomenų bazės modelis .....	28
3.3	Programinių komponentų architektūra .....	30
4	Veiklos taisyklių struktūrizavimo įrankio programinės realizacijos aprašymas .....	32
4.1	Programų sistemos kūrimui naudoti įrankiai .....	32
4.2	Įrankio principinė schema.....	32
4.3	Veiklos taisyklių struktūrizavimo įrankio funkcinis aprašymas.....	33
4.4	Programų sistemos įdiegimas .....	37
5	VT struktūrizavimo įrankio eksperimentas.....	39
5.1	Eksperimento koncepcija.....	39
5.2	Dalykinė sritis .....	39
5.2.1	Dalykinės srities parinkimas .....	39
5.2.2	Dalykinės srities charakteristika .....	39
5.3	Veiklos taisyklių specifikavimas .....	40
5.3.1	Veiklos taisyklių rinkinio sudarymas .....	40
5.3.2	Veiklos taisyklių formalizavimas pagal Roso metodą.....	40
5.4	Veiklos taisyklių saugyklos panaudojimas VT struktūrizavimo įrankio pagalba.....	42

5.4.1	Veiklos taisyklių saugyklos pildymas.....	42
5.4.2	Informacijos nuskaitymas iš veiklos taisyklių saugyklos.....	45
5.5	VT struktūrizavimo įrankio vertinimas.....	45
5.5.1	Veiklos taisyklių saugyklos pildymo programa.....	46
5.5.2	Veiklos taisyklių struktūrizavimo įrankis.....	48
5.5.3	Įrankių palyginimo apibendrinimas.....	50
6	Išvados.....	52
7	Literatūra.....	53
8	Summary.....	55
9	Terminų ir santrumpų žodynas.....	56
10	Priedai.....	57
10.1	IS projektavimo fazėje struktūrizuotų VT saugyklos loginio modelio lentelių aprašymai.....	57
10.2	Straipsnio kopija.....	68

# 1 ĮVADAS

Kuriant informacijos sistemas (IS) pagal veiklos taisyklių (VT) koncepciją, taisyklių identifikavimo, klasifikavimo, užrašymo fiksuota forma ir susiejimo su kitais IS projekto objektais klausimas yra ypač aktualus. Minėtus VT tvarkymo veiksmus apimantis procesas vadinamas VT struktūrizavimu. Iš eilės žinomų teorinių VT struktūrizavimo modelių išsiskiria Roso (*Ronald Ross*) metodas, leidžiantis VT išreikšti grafinėmis diagramomis [5]. Šis metodas yra konkretizuotas ir praplėstas, sukuriant pagal jį struktūrizuotų VT, duomenų modelio elementų ir kitos svarbios informacijos saugyklos loginį modelį [2], [3], taip pat įrankį, pildantį sukurtąją veiklos taisyklių saugyklą. Šiame darbe pristatomas VT saugyklą palaikantis grafinio taisyklių modeliavimo įrankis, leidžiantis visapusiškai įvertinti šio VT struktūrizavimo modelio gyvybingumą ir praktinę reikšmę.

Darbo antrame skyriuje apžvelgiami veiklos taisyklių ir Roso metodo pagrindai. Paaškinama VT samprata, kaip jos atsispindi, pavyzdžiui įmonės veikloje. Pristatomas motyvacinis modelis, jo panaudojimo sritys. Yra išskiriamos ir paaškinamos keturios veiklos taisyklių kategorijos:

1. Veiklos terminų aprašymas.
2. Faktai susiejantys terminus.
3. Apribojimai.
4. Išvedimai.

Roso metodas iš esmės apibrėžia nestruktūrinių veiklos taisyklių klasifikavimą ir formalų užrašymą, kitaip tariant modeliavimą, naudojant savitą grafinę notaciją. Pagal šį metodą taisyklės gali būti *nedalomos* ir *išvestinės*. Taip pat Rosas yra aiškiai išskirstęs abiejų tipų taisykles į atskiras tipų grupes.

VT modeliavimas pagal Roso metodą susideda iš šešių pagrindinių funkcijų:

1. Nustatyti ar taisyklė yra integralumo apribojimas, ar sąlyga.
2. Nustatyti taisyklės bazę.
3. Nustatyti taisyklės korespondentą.
4. Nustatyti taisyklės tipą.
5. Nustatyti taisyklės bazės ir korespondento ryšį.
6. Nustatyti papildomas veiklos taisyklės modelio ypatybes.

Atsižvelgiant į Roso veiklos taisyklių modeliavimo ypatybes yra sudarytas saugyklos loginis modelis, kurį sukūrė K. Kapočius ir R. Butleris. Modelyje naudojama duomenų modelio elementų saugojimo posistemė. Jos struktūra visiškai atitinka Roso metodu

naudojamam duomenų modeliui keliamiems reikalavimams. Sukurtoje saugykloje yra saugoma tik veiklos taisyklės diagramoje atsispindinti informacija.

Trečiame darbo skyriuje apžvelgiamas VT struktūrizavimo modelis. Pagal modifikuotą Roso metodą veiklos taisyklių saugyklai pildyti buvo sukurtas veiklos taisyklių struktūrizavimo įrankis. Jis gali būti panaudojamas formaliai užrašytų taisyklių išsaugojimui saugykloje, kurios duomenys gali būti panaudojami projektuojant IS.

Pateiktame duomenų bazės modelyje, aiškiai apibrėžiama kurios lentelės yra pildomos, kuriose saugoma kita VT modeliui reikalinga informacija. Programinių komponentų architektūros modelyje aiškiai pavaizduoti VT struktūrizavimo įrankio programiniai moduliai ir jiems priklausančios procedūros ir funkcijos, formos.

Veiklos taisyklių struktūrizavimo įrankio realizacija pateikta ketvirtame skyriuje, aprašyti kūrimui naudoti įrankiai. Iš VT struktūrizavimo įrankio principinės schemos matyti kaip tarpusavyje yra susijusi darbo aplinka, saugykla, programinis kodas. Funkciniame aprašyme pateiktos visos įrankio funkcijos.

Penktame darbo skyriuje apžvelgiamos įrankio panaudojimo galimybės. Šiam tikslui sukurta konkreti dalykinė sritis. Atliktas veiklos taisyklių specifikuojimas, užrašant VT neformalia ir formalia forma. Įsitikinta, kad įrankis pilnai atlieka savo funkcijas, t. y. veiklos taisyklės išsaugomos saugykloje, taip pat sugeneruojamas VT grafinis modelis, nuskaitant duomenis iš saugyklos. Galiausiai atliktas VT struktūrizavimo įrankio vertinimas parodo, jog grafinis veiklos taisyklių struktūrizavimas yra patogesnis, nei naudojant atributinę sąsają.

Prieduose pateikta VT struktūrizavimo įrankio pildomos saugyklos lentelių aprašymai (10.1), konferencijoje „Informacinės technologijos verslui – 2006“ pristatyto straipsnio kopija (10.2).

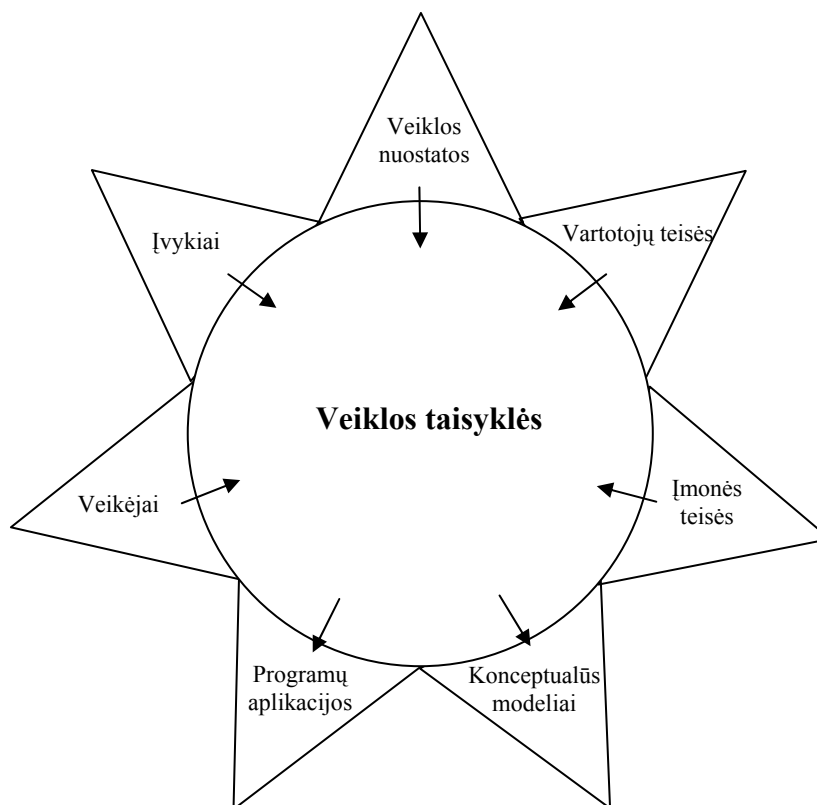
## 2 VEIKLOS TAISYKLIŲ STRUKTŪRIZAVIMO ANALIZĖ

### 2.1 Veiklos taisyklių samprata

Veiklos taisyklė – tai formuluotė, kuri charakterizuoja ar nusako kai kuriuos veiklos aspektus. Tai tvirtai numatyta veiklos struktūra, kontrolė ar poveikis veiklos funkcionavime. Pagal situaciją, kiekviena veiklos taisyklės formuluotė gali būti suskaidyta į daugiau veiklos taisyklių [1]. Veiklos taisyklė nesusieja kontrolinių srautų formuluočių, esančių programoje kaip kad duomenų bazės atnaujinimai, rezultatai.

Veiklos taisyklė yra nepriklausoma nuo modeliavimo ar techninių platformų. Taisyklė yra nustatyta ir kontroliuojama veiklos dalyvių.

Pavyzdžiui, komercinės įmonės. Jos apima tūkstančius taisyklių kombinacijų, kurios dirba operaciniame veiklos lygmenyje. Veiklos taisyklės apibrėžia ir kontroliuoja produktų gyvavimą, servisą ir infrastruktūrų palaikymą. Veiklos taisyklės nusako įmonės pirkimus, pardavimus, užsakymus, darbuotojus, ataskaitas, planus ir kita. Jos tarsi įmonės branduolys (1 pav.) [9].



1 pav. Veiklos taisyklių vieta įmonėje

## 2.2 Veiklos taisyklių motyvacinis modelis

Veiklos taisyklių motyvacinį modelį galima panaudoti kaip schemą arba struktūrą formulavimui, susiejimui ir veiklos plano valdymui. Veiklos taisyklių motyvacinis modelis suteikia:

1. Nustato faktorius, kurie motyvuoja veiklos planų sukūrimą.
2. Nustato ir apibrėžia veiklos planų elementus.
3. Nurodo, kaip visi faktoriai ir elementai yra susiję [12].

Yra dvi pagrindinės veiklos taisyklių motyvacinio modelio sritys:

- Pirmoji yra veiklos plano pabaiga ir reikšmė. Pabaigoje yra dalykų, kuriuos įmonė tikisi pasiekti – pavyzdžiui, tikslai ir siekiai. Reikšmėse yra dalykų, kuriuos įmonė siekia susieti su pabaigomis – pavyzdžiui, strategija, taktika, veiklos taisyklės.
- Kitas poveikis, kurio pavidalu veiklos plano elementai ir vertinimas gaunami iš pabaigos ir reikšmės poveikio. Pavyzdžiui, stipriosios pusės, silpnosios pusės, galimybės, rizikos.

Pabaiga, reikšmė ir poveikis yra susiję vieni su kitais:

1. Įmonės tikslų pasiekimas.

Tam reikia, suskirstyti detalius veiklos planų elementus – kitaip sakant, nebūtina reikšmė, kad pasiektumėme numatytą pabaigą.

2. Veiklos plano elementai.

Nustatant detalią pabaigą, kurios reikšmė teikiama ir poveikis, kuris pabrėžia svarbų pasirinkimą – yra motyvacija.

Veiklos taisyklės turi didelę reikšmę veiklos planų vystymuisi. Pavyzdžiui, jie gali siekti sekančių tikslų:

- Daryti veiklos taktikai pakankamai įtakos, duodant didelę įtaką darbui.
- Suteikti išankstines pozicijas, kada kai kurie veiklos plano elementai žlunga.
- Išvengti konfliktų, kada viena veiklos pabaiga konfliktuoja su kita.

Visi veiklos taisyklių motyvacinio modelio elementai yra kuriami remiantis veiklos tikslais. Pagrindinė idėja yra plėtoti veiklos modelį iš veiklos plano elementų prieš sistemos projektavimą, arba techninio kūrimo pradžią. Šiuo metodu, veiklos planas gali išsivystyti iš veiklos kūrimo, sujungiant sistemos sprendimus su veiklos tikslais [12].

## 2.3 Veiklos taisyklių kategorijos

Veiklos taisyklės klasifikuojamos į keturias kategorijas:

1. Veiklos terminų aprašymas.
2. Faktai susiejantys terminus.



3. Apribojimai.

4. Išvedimai.

*Veiklos terminų aprašymas.* Pats paprasčiausias veiklos taisyklės elementas yra išreiškiamas kalba. Terminų aprašymas jau yra veiklos taisyklė, kuri aprašo, kaip žmonės galvoja ir kalba apie daiktus. Šie terminų aprašymai yra tiksli veiklos taisyklės kategorija. Terminai tradiciškai turi būti aprašyti dokumentuose, arba numatyti objektinių klasių diagramose.

*Faktai susiejantys terminus.* Natūralios ar valdomos organizacijos struktūros gali būti aprašytos faktais, kurie susiejami terminais. Pasakymas, kad klientas gali duoti užsakymą - yra veiklos taisyklė. Pats faktas gali būti aprašytas dokumentais, natūralia kalba arba ryšiais, atributais ir apibendrinančiomis struktūromis grafiniuose modeliuose.

*Apribojimai.* Kiekvienoje įmonėje yra tam tikri nusistovėję apribojimai, ir tai yra glaudžiai susiję su tuo, ar informacija gali būti atnaujinama ar ne. Tai yra daroma tam, kad uždraustumėme įrašų įrašymą, ir kai kuriais kitais atvejais.

*Išvedimai.* Veiklos taisyklės apibrėžia, kaip vieno formato žinios gali būti transformuojamos į kitas žinias, tikėtina, ir į skirtingą formatą.

## **2.4 Roso metodas**

Metodo autorius iš esmės apibrėžia nestruktūrinių VT klasifikavimo ir formalaus užrašymo arba modeliavimo, naudojantis savita grafine notacija, metodiką. Roso sukurta VT klasifikacija gali būti traktuojama kaip labiau konceptualaus GUIDE modelio [6] papildinys, detalizuojantis veiksmo teiginių VT kategoriją.

Pagal Roso metodą taisyklės gali būti *nedalomos* ir *išvestinės*. Yra išskiriami 32 nedalomų taisyklių tipai, kurie grupuojami į 7 tipų grupes. Išvestinė taisyklė – tai taisyklė, kuri išreiškiama kitų taisyklių aprašais. Išvestinės taisyklės nėra nedalomos ir gali būti sudarytos iš keleto nedalomų taisyklių arba kitų išvestinių taisyklių. Yra išskiriami 58 išvestinių taisyklių tipai, kurie pagal panaudojimo sritį yra grupuojami į 12 tipų grupių. Būtina pažymėti, kad ypač daug dėmesio R. Rosas skiria taisyklių tipų aprašymui, todėl šis modelis klasifikavimo požiūriu yra ypač tikslus.

<i>Nedalomų taisyklių grupės</i>	<i>Išvestinių taisyklių tipų grupės</i>
<ol style="list-style-type: none"> <li>1. Egzempliorių patvirtintojai</li> <li>2. Tipo patvirtintojai</li> <li>3. Pozicijos patvirtintojai</li> <li>4. Funkciniai patvirtintojai</li> <li>5. Lyginamieji įvertintojai</li> <li>6. Matematiniai įvertintojai</li> <li>7. Projekcijos valdikliai</li> </ol>	<ol style="list-style-type: none"> <li>1. Egzempliorių testavimo</li> <li>2. Pozicijos patikrinimo</li> <li>3. Atributų modifikavimo</li> <li>4. Sekos valdymo</li> <li>5. Sekos specifikavimo</li> <li>6. Kompozicijos struktūrų testavimo</li> <li>7. Sąlyginio laiko įvertinimo</li> <li>8. Atnaujinimų įvertinimo</li> <li>9. Veiklos koordinavimo</li> <li>10. Egzempliorių įgalinimo</li> <li>11. Egzempliorių kopijavimo</li> <li>12. L. Egzempliorių iškvietimo</li> </ol>

*Nedalomos taisyklės* yra skirstomos į septynias tipų grupes [Ross, 1997] (žr. 2 lentelė). Galima pastebėti šios lentelės panašumą su Mendelejevo cheminių elementų lentele. Taisyklės gali būti sujungiamos įvairiomis kombinacijomis tokiu būdu gaunant *sudėtines* taisykles. Skaidymas į tipų grupes turi konkrečią prasmę:

- *Kiekviena tipų grupė gali atlikti tam tikros rūšies patikrinimą, kurio negali kitų grupių taisyklės.*

Kiekviena taisyklė *skaičiuoja*, t. y. atlikus taisyklėje numatytą patikrinimą visada gaunamas rezultatas. Šis rezultatas dažniausiai yra nematomas. Tačiau rezultatas gali būti nesunkiai materializuotas siekiant panaudoti jį kitose taisyklėse, tuomet jis traktuojamas kaip papildomas taisyklės atributas *išeigos reikšmė*.

- *Išeigos reikšmės tipas vienareikšmiškai susijęs su taisyklės tipu. Kiekviena tipų grupė pasižymi unikaliu šios reikšmės tipu.*

2 lentelė. Nedalomų taisyklių tipai pagal Roso metodą (lietuvių kalba)

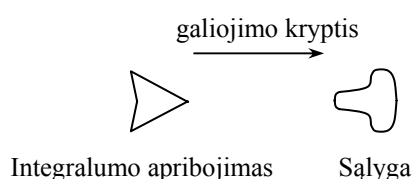
I. Egzempliorių patvirtinimo	II. Tipo patvirtinimo	III. Pozicijos patvirtinimo	IV. Funkcinio patvirtinimo	V. Palyginamojo įvertinimo	VI. Matematinio įvertinimo	VII. Projekcijos Valdikliai
<b>P</b> privaloma <i>EK</i>	<b>B</b> bendra <i>TK</i>	<b>POZ</b> pozicinė <i>IF</i>	<b>FNKC</b> funkcinė <i>EN</i>	<b>LG</b> lygu <i>PV</i>	<b>APSK</b> apskaičiuota <i>REZ</i>	<b>IGL</b> igalinta <i>IEK</i>
<b>A</b> apribota <i>EK</i>	<b>BA</b> bendrai-atskirianti <i>TK</i>	<b>ZEM</b> žemiausia <i>IF</i>	<b>UNIK</b> unikali <i>EN</i>	<b>NLG</b> nelygu <i>PV</i>	<b>SUM</b> susumuota <i>REZ</i>	<b>KOP</b> nukopijuota <i>KEK</i>
	<b>BS</b> bendraisusiejanti <i>TK</i>	<b>AUK</b> aukščiausia <i>IF</i>	<b>SV</b> svyruojanti <i>EN</i>	<b>DU</b> daugiau-už <i>PV</i>	<b>ATM</b> atimta <i>REZ</i>	<b>IVK</b> įvykdyta <i>VEK</i>
	<b>BU</b> bendrai-uždraudžianti <i>TK</i>	<b>CHRO</b> chronologinė <i>DL</i>	<b>AUG</b> auganti <i>EN</i>	<b>DLG</b> daugiau-arba-lygu <i>PV</i>	<b>DGB</b> sudauginta <i>REZ</i>	
		<b>SEN</b> seniausia <i>DL</i>	<b>KRE</b> krentanti <i>EN</i>	<b>MU</b> mažiau-už <i>PV</i>	<b>PDL</b> padalinta <i>REZ</i>	
		<b>NAU</b> naujausia <i>DL</i>	<b>NAT</b> neatnaujinama <i>EN</i>	<b>MLG</b> mažiau-arba-lygu <i>PV</i>	...	<i>REZ</i>

*Išvestinės taisyklės* – tai taisyklės, kurios išreiškiamos kitų taisyklių aprašais. Išvestinės taisyklės nėra nedalomos. Išvestinė taisyklė gali būti sudaryta iš keleto nedalomų taisyklių arba kitų išvestinių taisyklių. Šio tipo taisyklėms būdingos nedalomų taisyklių savybės, iš jų ir specifinio tipo išeities vertė.

Formaliai užrašant VT, reikia remtis dalykinės srities duomenų modeliu. Pagrindiniai šio modelio elementai yra esybės, atributai, asociacijos, sąryšiai, potipiai [5]. VT modeliavimas susideda iš šešių pagrindinių funkcijų:

1. Nustatyti ar taisyklė yra integralumo apribojimas, ar sąlyga. T. y. taisyklė turi būti vykdoma ar išreikšta kaip patikrinimas?

Integralumo apribojimas – tai taisyklė, kurios rezultatas visada privalo būti loginė tiesa. Sąlyga – tai taisyklė, kurios rezultatas gali būti tiek loginė tiesa, tiek loginis melas. Taip pat rezultatas gali būti nežinomas. Kuriant VT grafinį modelį, integralumo apribojimams ir sąlygoms žymėti naudojami specialūs žymėjimai (2 pav.).



2 pav. Taisyklės vykdymo ir taisyklės tikrinimo žymėjimas (pagal [5])

2. Nustatyti taisyklės bazę. Kuriam tipui duomenų modelyje reikėtų priskirti nagrinėjamą taisyklę?

Dažniausia VT baze būna duomenų modelio objektas – esybė, atributas, ryšys. Tačiau baze taip pat gali būti kita VT ar jos išeišos reikšmė. Kiekviena VT privalo turėti bazę [5]. Išeinantis iš bazės ryšys veiklos taisyklės grafiniame modelyje vaizduojamas punktyrine linija.

3. Nustatyti taisyklės korespondentą. Kokie tipai duomenų modelyje būtini, norint patikrinti taisyklę?

Korespondentu gali būti duomenų modelio objektas, taip pat kita veiklos taisyklė ar jos išeišos reikšmė. Kiekviena VT privalo turėti bent vieną korespondentą. Įeinantis į korespondentą ryšys grafiniame veiklos taisyklės modelyje vaizduojamas punktyrine linija.

Kiekviena taisyklė turi vieną ar daugiau korespondentų. Galima išskirti šias ypatybes:

- ✓ Nedalomos taisyklės, priklausančios tipo patvirtintųjų šeimai, privalo turėti du arba daugiau korespondentų, nes veikia kaip loginiai operatoriai IR ar ARBA.
- ✓ Nedalomos taisyklės, priklausančios matematinių įvertintųjų šeimai, gali turėti daugiau negu vieną korespondentą,
- ✓ Kitoms šeimoms priklausančios nedalomos (atominės) taisyklės gali turėti tik vieną ir ne daugiau korespondentų.

4. Nustatyti taisyklės tipą, t. y. kokius testus pritaiko taisyklė.

Taisyklė gali priklausyti nedalomų arba išvestinių VT grupei, kurių kiekviena skaidoma į eilę tipų [5]. Visi taisyklių tipai tinka tiek integralumo apribojimams, tiek ir sąlygoms. Taisyklės tipo santrumpa įrašoma apribojimo arba sąlygos simbolio viduje.

5. Nustatyti taisyklės bazės ir korespondento ryšį, t. y., kaip dalykinės srities duomenų modelio elementai, reikalingi taisyklei patvirtinti, susiję tarpusavyje.

Svarbu išskirti teisingą sąryšį tarp bazės ir korespondento. Šis sąryšis privalo sekti iš duomenų modelio.

6. Nustatyti papildomas veiklos taisyklės modelio ypatybes.

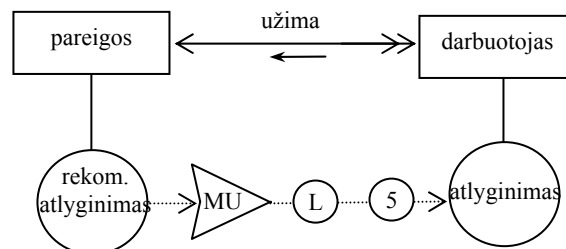
Roso notacijoje yra numatyta galimybė naudoti papildomus simbolius, galinčius pakeisti standartinę tam tikro tipo taisyklės interpretaciją. Šie simboliai skirstomi į dvi pagrindines grupes [5]:

- Interpretatoriai – tai simboliai, nurodantys, jog taisyklę jos bazės atžvilgiu reikia interpretuoti ne taip, kaip reikėtų remiantis tik jos tipu. Šie simboliai yra vaizduojami ant bazės ryšio rodyklės.

- Apribojimai – tai simboliai, nurodantys, jog taisyklei jos korespondento(-ų) egzempliorių atžvilgiu turi būti taikomi specialūs apribojimai. Šie simboliai vaizduojami ant korespondento ryšio rodyklės.

Be specialiųjų simbolių VT diagramose dar gali būti vartojamos konstantos, taisyklių išeigos reikšmės ir kitos papildomos priemonės. Paprastai taisyklės išeigos reikšmė yra „nematoma“ ir naudojama taisyklės viduje tam tikram testavimui atlikti arba apribojimui pritaikyti. Tačiau kai kurios taisyklės turi tiesiogiai tikrinti kitų taisyklių išeigos reikšmes (tai labiau būdinga sąlygos tipo taisyklėms). Tuomet veiklos taisyklės išeigos reikšmė atsispindės ir taisyklės, naudojančios šią reikšmę, diagramoje. Kaip jau buvo minėta, taisyklės išeigos reikšmė gali būti kitos taisyklės bazė arba korespondentas.

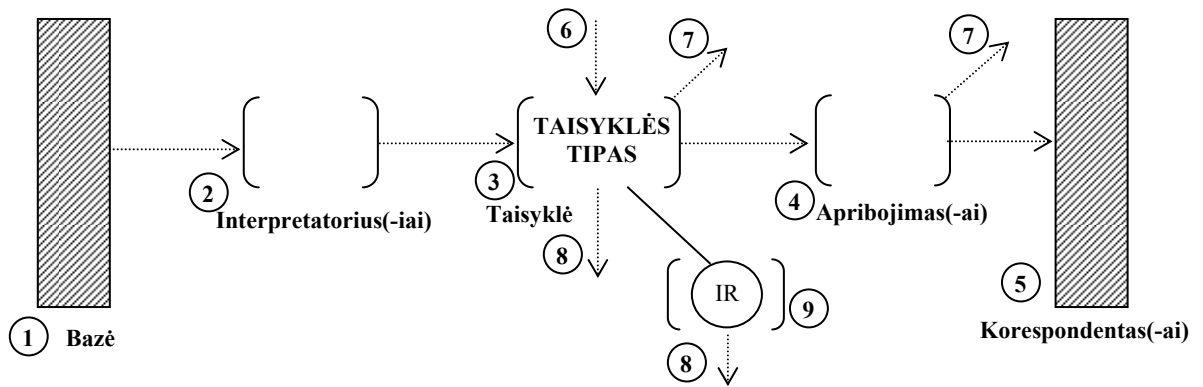
Paprastos taisyklės pavyzdys pateiktas 3 pav. Ji apibrėžia tokį apribojimą: *„Rekomenduojamas pareigų atlyginimas turi būti mažesnis už bent penkių darbuotojų, užimančių šias pareigas, atlyginimą“* [2]. Šios taisyklės bazė yra dalykinės srities duomenų modelio esybės pareigos atributas rekom.atlyginimas (rekomenduojamas atlyginimas). Taisyklės korespondentas – duomenų modelio esybės darbuotojas atributas atlyginimas. Mus domina tik tos pareigos, kurias užima darbuotojai, todėl į diagramą įtrauktas ir šis duomenų modelio sąryšis (faktas). Taisyklės tipas yra **Mažiau-už** (kodas MU). Diagramoje taip pat naudojami ir du apribojimai: žemesniojo slenksčio apribojimas (simbolizuoja rutuliukas su raide L) ir numeratorius (simbolizuoja rutuliukas su skaičiumi 5).



3 pav. Veiklos taisyklės grafinio modelio iliustracija [2]

## 2.5 Veiklos taisyklių formalizavimas pagal Roso metodą

VT formalizavimą reikėtų pradėti prieš tai preliminariai patvirtinus taisyklės teiginį. Šiame etape naudojama Roso modelyje apibrėžta taisyklių užrašymo specialių diagramų pavidalu metodika. Priimama, jog kiekviena pagal Roso metodą sumodeliuota VT pasižymi tokia pačia struktūra [Ross, 1997], kuri pateikta 4 pav.



4 pav. Veiklos taisyklės formalizavimo pagal Roso metodą sintaksė

Veiklos taisyklės diagramos sudedamosios dalys yra šios (žr. 4 pav.):

1. *Privaloma*. Gali būti: duomenų modelio tipas, į duomenų modelyje neapibrėžta konstanta, kita VT arba kitos VT išeišgos reikšmė.
2. *Nebūtina*. Teisingas panaudojimas priklauso nuo VT tipo.
3. *Privaloma*. Gali būti integralumo apribojimas arba sąlyga.
4. *Nebūtina*. Teisingas panaudojimas priklauso nuo VT tipo.
5. *Privaloma*. Gali būti: duomenų modelio tipas, į duomenų modelyje neapibrėžta konstanta, kita VT arba kitos VT išeišgos reikšmė.
6. *Nebūtina*. Vaizduoja įeišgos taisyklę.
7. *Leistina daugiau negu vienas*. Taisyklė gali turėti daugiau negu vieną korespondentą.
8. *Nebūtina*. Vaizduoja išeišgos taisyklę.
9. *Nebūtina*. Vaizduoja taisyklės išeišgos reikšmę.

Be šių elementų VT diagramoje turi matytis ir visi jos interpretacijai svarbūs duomenų tipai.

Formalizuojant VT būtina laikytis šių principų:

1. Į taisyklės diagramą turi būti įtraukti tik tie duomenų tipai, kurie yra vienareikšmiškai svarbūs specifikuojamai taisyklei.
  - Naudojant tik vienareikšmiškai svarbius duomenų tipus, specifikuojama taisyklė bus tiksliau interpretuojama bei bus išvengta nesusipratimų, kuriuos galėtų sukelti pertekliniai VT diagramoje matomi elementai.
  - VT diagramoje turi būti pateikti visi duomenų tipai, kurie susieti su formalizuojamu taisyklės teiginiu reikalavimų specifikuojimu pagal siūlomą metodą metu. Didelė tikimybė, jog šis tipų rinkinys bus išsamus, tačiau projektuotojas privalo įsitikinti, ar nėra trūkstamų komponentų.

- VT modeliavimo metu naudojamas programinis įrankis turėtų užtikrinti VT diagramoje naudojamų duomenų tipų atitikimą bazėje saugomai struktūrai bei neliesti įvesti logiškai neišsamių diagramų (pvz.: naudojančių atributų simbolius, neturinčius sąsajų su diagramoje matomomis esybėmis, arba ryšius tarp esybių neįtraukiant pačių esybių ir pan.).
2. Būtina laikytis principo, jog interpretuojant taisyklę ji bus taikoma tik tiems korespondento egzemplioriams, kurie susiję su bazės egzemplioriumi.
    - Kiekvienas į taisyklės diagramą įtrauktas ryšys tarp bazės ir korespondento(-ų) turi būti svarbus taisyklės interpretacijai. Nesvarbūs ryšių tipai į diagramą neįtraukiami.
  3. Taisyklės turi būti tiksliai identifikuotos.
    - Veiklos taisyklės ir jas atitinkančios diagramos turi būti identifikuojamos pagal tam tikrą sistemą parinktais vienodais identifikatoriais (pvz.: visos taisyklės gali būti numeruojamos, tą patį numerį priskiriant tiek taisyklės teiginiui, tiek ir jos diagramai). Rekomenduojama naudoti reikalavimų specifikavimo metu VT suteiktus kodus.
  4. Kiekviena VT gali būti priskirta tik vienam tipui.
  5. Kiekviena VT diagramoje vaizduojama nedaloma taisyklė gali turėti tik vieną bazę.
  6. Taisyklė turi būti specifikuojama kaip įmanoma bendresniam duomenų modelio elementui.
    - Specifikuojant taisyklę reikia išskirti tokį bazės tipą, kad jis apimtų visus taisyklės apibrėžiamus potipius. Priimama, jog visi šio tipo potipiai paveldi šią taisyklę. Taigi, kiekvienam iš potipių išskirti atskiros taisyklės nereikėtų.
    - Būtina įsitikinti, ar tikrai visi parinktos bazės potipiai paveldi taisyklę. Taip pat potipiui negali būti išskirta taisyklė, kuri pažeidžia bendresniam tipui, į kurį įeina nagrinėjamas potipis, taikomą taisyklę.
  7. Taisyklės Įgalintojai (ar kitų tipų taisyklės) neturėtų būti painiojamos su trigeriais, naudojamais duomenų bazių valdymo sistemose.
    - Trigeris gali būti naudojamas iškviešti tam tikrai procedūrai, pritaikančiai tam tikrą taisyklę, kuomet įvyko tam tikras įvykis. Tai tik viena nedidelė taisyklės realizavimo proceso dalis. VT neregamentuojama, kaip ar kada programiniu požiūriu jos turėtų būti vykdomos. VT reglamentuoja pačią veiklą ir tai, kas svarbu šios veiklos dalyviams.

Remiantis Roso metodu ir pasiūlymo specifika suformuluoti baziniai reikalavimai formaliai užrašytoms VT pateikti žemiau:

1. Veiklos taisyklė privalo būti priskirta vienam ir tik vienam tipui.
2. Veiklos taisyklė privalo turėti vieną ir tik vieną bazę.
3. Veiklos taisyklės bazė gali būti arba duomenų tipas, arba konstanta, arba kita VT, arba kitos VT išeišigos reikšmė.
4. Veiklos taisyklė privalo turėti ne mažiau negu 1 korespondentą.
5. Veiklos taisyklės korespondentas gali būti arba duomenų tipas, arba konstanta, arba kita VT, arba kitos VT išeišigos reikšmė.

## **2.6 Struktūrizuotų veiklos taisyklių saugyklos architektūra**

Tradiciniai informacijos sistemų (IS) projektavimo metodai, didėjant operatyvaus informacijos apdorojimo poreikiui dinamiškai kintančioje aplinkoje, nėra pakankamai efektyvūs. Šias problemas padeda spręsti sąlygiškai naujas informacijos sistemų projektavimo būdas, pagrįstas veiklos taisyklių koncepcija. Veiklos taisyklių koncepcijos pranašumą, palyginti su tradiciniu požiūriu, lemia tokie veiksniai:

1. Tradiciniais metodais sukurtos IŠ pakeitimo ar pritaikymo pasikeitusiai situacijai išlaidos yra pernelyg didelės. Veiklos taisyklėmis pagrįstoje sistemoje taisyklės prirėikus nesunkiai gali keisti patys veiklos srities atstovai, programuotojų ar analitikų pagalba čia beveik nereikalinga.

2. Egzistuojantys informacinių sistemų specifėkavimo ir projektavimo modeliai blogai išreiškia dalykinėje srityje galiojančias taisykles, jos yra padrikai paskirstytos po visą sistemą. Taikant veiklos taisyklių nepriklausomo sluoksnio koncepciją, galima išspręsti šią problemą, nes taisyklių identifikavimas ir automatizavimas yra išreiškiamas formaliais metodais, o pačios taisyklės saugomos atskirai nuo kitų sistemos komponentų.

Sistemoje, pagrįstoje veiklos taisyklių principais, taisyklės yra saugomos autonomiškoje veiklos taisyklių saugykloje, kuri kartu su specialia veiklos taisyklių interpretavimo sistema užtikrina informacinės sistemos adaptyvumą kintančios veikos sąlygomis. Dabar egzistuoja keli veiklos taisyklių klasifikavimo ir modeliavimo metodai, todėl bendro saugyklos modelio nėra [2].

### **2.6.1 Veiklos taisyklių koncepcijos prasmė**

Tradicinėje kliento-serverio architektūroje veiklos taisyklės sudaro atskirą sluoksnį, esantį tarp duomenų valdymo ir atvaizdavimo sluoksnių (5 pav.). Iš šių sluoksnių veiklos taisyklių sistema perima ir daugumą duomenų tikrinimo, algoritmų logikos, apsaugos bei kitų funkcijų. Reikia pabrėžti, kad kiekvienas iš trijų sluoksnių funkcionuoja visiškai nepriklausomai. Atskyrus veiklos taisyklės nuo kitų architektūros sluoksnių, sistemos modifikavimui, pritaikant ją prie



pasikeitusių veiklos sąlygų, nereikia atlikti sudėtingų struktūros pakeitimų ar perrašyti programos modulių. Taisyklių pasikeitimus gali fiksuoti patys sistemos vartotojai, naudodamiesi paprastomis priemonėmis. Visi sistemos moduliai, kurių veikimui daro įtaką pasikeitusi taisyklė, įvertina šiuos pasikeitimus, perimdami pakitusius funkcionalumo apribojimus iš jos specifikacijos.

Pažymėtina, jog veiklos taisyklės turi būti traktuojamos ne kaip veiklos objektai, duomenų bazės trigeriai ar apribojimai, pranešimai, procedūros, veiksmai. Taisykles reikia nagrinėti kaip visiškai savarankišką modeliavimo objektą, kuris savo ruožtu yra susijęs 5 pav. pateiktais sistemos veikimo aspektais [2].

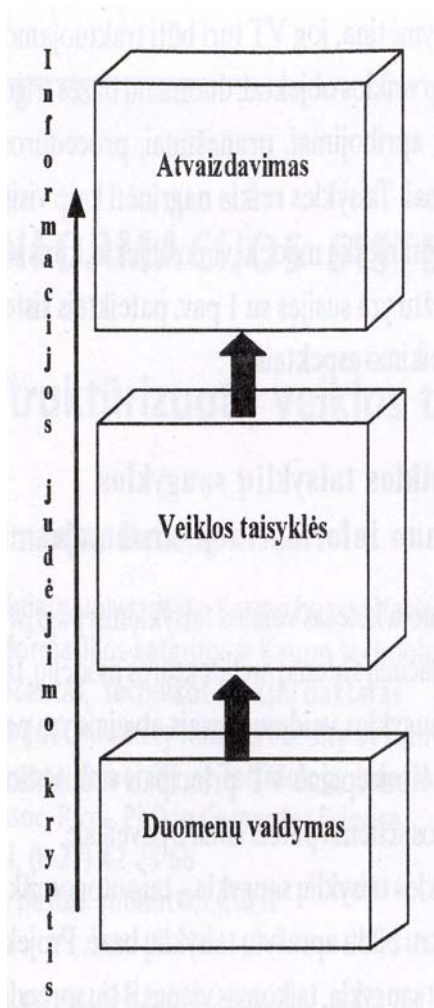
## **2.6.2 Veiklos taisyklių saugyklos vaidmuo informacinėje sistemoje**

Egzistuoja keletas veiklos taisyklėmis pagrįstų informacinių sistemų architektūros modelių, tačiau saugyklos vaidmuo visais atvejais yra panašus. Konceptinė veiklos taisyklių principais veikiančios sistemos schema pateikiama 6 pav.

Veiklos taisyklių saugykla – tai autonomiška tam tikru būdu aprašytų taisyklių bazė. Projektuojant saugyklą, taikomas vienas iš šių sprendimų:

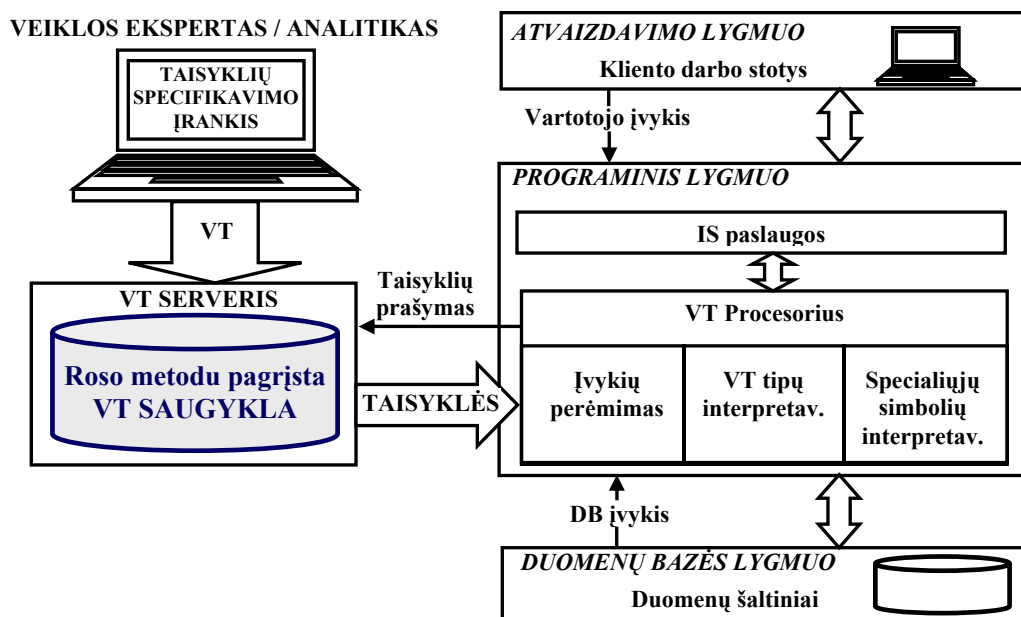
- 1) parametrais pagrįstas sprendimas. Tai dažniausiai naudojamas metodas. Šiuo atveju taisyklės saugomos tiesiog duomenų bazėje, o kiekviena jų savybė apibūdinama šios bazės lentelių atributais;
- 2) teiginiais pagrįstas sprendimas. Ši metodologija leidžia užrašyti taisykles sąlyginiais arba faktiniais teiginiais. Toks sprendimas nusileidžia parametrais pagrįstai struktūrai, kuri yra kur kas lankstesnė.

Veiklos taisyklių saugyklą pildyti ar keisti jau esamas veiklos taisykles gali tiek analitikas, tiek modeliuojamos veiklos ekspertas. Veikiančioje sistemoje taisykles interpretuoja speciali programa, vadinama veiklos taisyklių interpretavimo sistema arba mechanizmu. Taigi saugykla yra atskirta nuo sistemos ir gali būti naudojama visur, kur naudojamas analogiškas duomenų modelis ir yra atitinkama interpretavimo sistema.



- ekrano langai, objektų išdėstymas
  - meniu struktūra ir judėjimas tarp ekranų
  - langų valdymas
  - pelės ir klaviatūros įvykių apdorojimas
  - įvestų reikšmių sintaksinė kontrolė
  - situacijų „kas bus, jeigu...“ analizė
- Bulio logika
  - apibrėžimo srities ir reikšmių srities patikrinimas
  - duomenų tarpusavio priklausomybės patikrinimas
  - būsenos kitimo eiliškumo patikrinimas
  - duomenų vientisumo apribojimų užtikrinimas
  - veiksmų trigerių logika
  - algoritmų vykdymas
  - vartotojų autentifikavimas
  - pranešimų apdorojimas
- duomenų kreipties ir manipuliavimo kalba
  - lygiagretaus duomenų naudojimo valdymas
  - SQL interpretavimas
  - duomenų apsauga

5 pav. Veiklos taisyklių vieta ir funkcijos kliento – serverio architektūroje



6 pav. Veiklos taisyklių saugyklos vaidmuo taikomojoje sistemoje

### 2.6.3 Pagal Roso metodą formalizuotų veiklos taisyklių saugyklos modelis

Kuriant VT saugyklos loginės struktūros modelį, jam buvo iškelti šie baziniai reikalavimai:

1. Informacijos išsaugojimo lygis turi būti toks, kad būtų galima į duomenų bazę įvesti duomenis apie bet kokio tipo ar struktūros VT, formalizuotą pagal Roso metodą.
2. Saugyklos duomenų bazėje saugomos informacijos apie VT turi pakakti, norint visiškai atkurti jos diagramą.
3. Aprašant VT diagramą išsaugoma tik informacija, kuri matoma tiesiogiai diagramoje arba neformaliai ją charakterizuoja. Informacija apie kiekvieno iš VT tipų ar specialiųjų elementų interpretacijos principus nesaugoma.
4. Be formalią VT apibūdinančių duomenų taip pat turi būti saugoma ir papildoma informacija, palengvinanti taisyklių peržiūrą ir jų supratimą (grupavimo, kilmės ir kitokio pobūdžio duomenys).

Taigi, saugykloje turi būti saugoma tik VT diagramoje atspindinti informacija. Priimama prielaida, jog informacija apie kiekvieno taisyklės tipo interpretaciją bus užkoduota tiesiogiai VT interpretavimo sistemoje. VT atskyrimas nuo jų interpretavimo yra vienas iš VT požiūrio principų [Business, 2003].

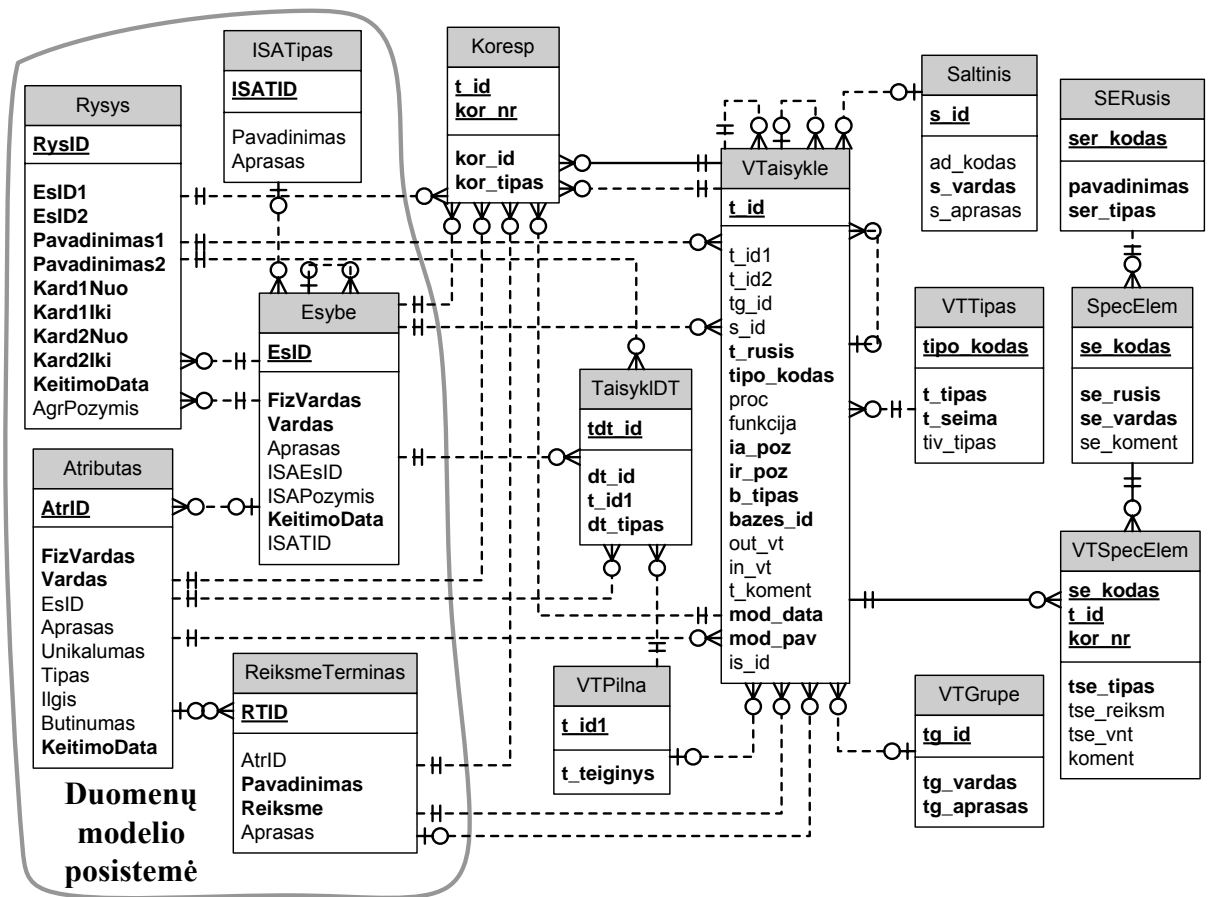
Atsižvelgiant į Roso VT modeliavimo metodo ypatybes sudarytas saugyklos loginis modelis pateiktas 7 pav. Šis modelis yra gautas tobulinant ankstesnius pasiūlymo variantus, pristatytus [Butleris, Kapočius, 2001; Butleris, Kapočius, 2002].

Modelyje naudojama domenų modelio elementų saugojimo posistemė. Jos struktūra visiškai atitinka kartu su Roso metodu naudojamam duomenų modeliui keliamus reikalavimus. Be duomenų modelio elementų išsaugojama ir reikšmių lentelė `ReiksmeTerminas`, kurioje bus saugomos VT diagramose naudojamos konstantos.

Kiekviena logiškai vientisa VT, išreiškiama viena diagrama, `VTPilna` gali susidėti iš keleto veiklos taisyklių `VTaisykle`, priklausančių vienam iš Roso metode numatytų tipų `VTTipas`. Kiekviena pilna taisyklė gali būti susijusi su daugeliu dalykinę sritį charakterizuojančių duomenų modelio elementų iš lentelių `Esybe`, `Rsys`, `Atributas`, `ReiksmeTerminas`. Kiekviena veiklos taisyklė `VTaisykle`, turi bazę ir vieną ar daugiau korespondentų `Korespondentas`. Korespondentų, kaip ir bazę, gali būti duomenų modelio elementas, konstanta kita veiklos taisyklė `VTaisykle` arba kitos VT išeigos reikšmė. Veiklos taisyklės `VTaisykle` diagramoje gali būti naudojami vienas ar daugiau specialiųjų simbolių `SpecElem`, kurie sugrupuoti pagal rūšį `SERusis`.

Be tiesiogiai taisyklės diagramoje atsispindinčios informacijos yra saugomi ir papildomi duomenys apie VT kilmės šaltinį `Saltinis`, be to, taisyklės gali būti grupuojamos į įvairaus pobūdžio grupes `VTGrupe`.

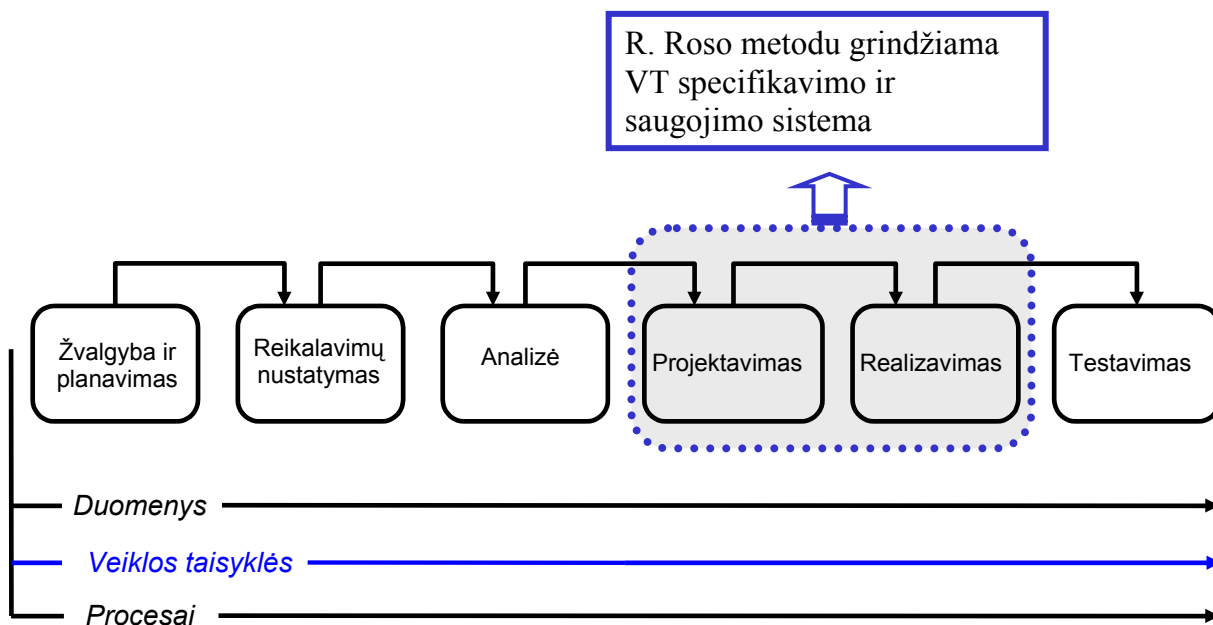
Kiekvienos modelio lentelės aprašai pateikti 10.1 šio darbo priede.



7 pav. Modifikuotu Roso metodu formalizuotų VT saugyklos loginės struktūros modelis

### 3 VEIKLOS TAISYKLIŲ STRUKTŪRIZAVIMO MODELIS

Veiklos taisyklių struktūrizavimo įrankis gali būti panaudojamas sudarant formaliai aprašytą veiklos taisyklių saugyklą, kurioje esantys duomenys gali būti panaudojami projektuojant IS. Taip pat pati saugykla gali būti specifinės veiklos taisyklėmis grindžiamos IS architektūros dalimi (8 pav.).



8 pav. Kuriamo įrankio vieta IS kūrimo procese [3]

Pirminė pritaikymo sritis – VT specifیکavimas pagal R. Roso metodą IS projektavimo metu, remiantis jau anksčiau sudaryta IS reikalavimų specifیکacija. Veiklos taisyklių saugyklos sukūrimo tikslas – kompiuterizuoti Roso VT struktūrizavimo metodiką, remiantis KTU Informacijos sistemų katedroje sukurtu VT saugojimo fizinės struktūros modeliu. Toks (prototipinis) kompiuterizavimas būtinas, norint toliau vystyti kuriamą VT grindžiamą IS kūrimo metodą, kurio dalimi yra sukurtas VT struktūrizavimo modelis.

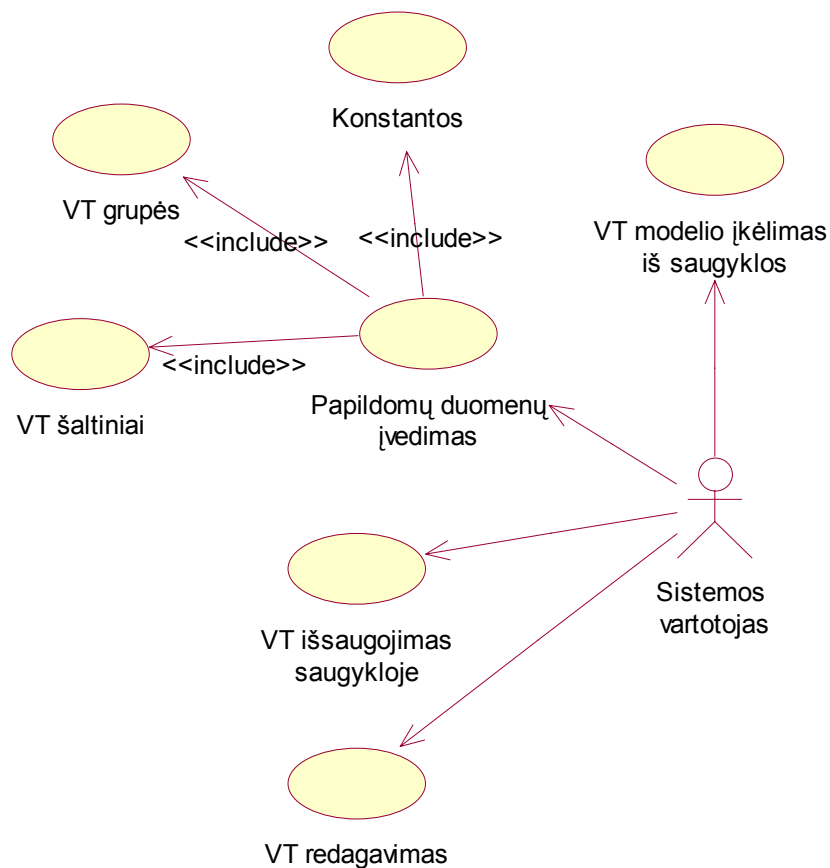
Kompiuterizuojamas VT struktūrizavimo modelis naudojamas IS projektavimo metu, o ne reikalavimų specifیکavimui. Jis galėtų būti integruojamas ir į reikalavimų specifیکavimo procesą, tačiau šioje IS kūrimo stadijoje detalumas ir formalumas, kurį siūlo Roso metodas, nėra naudingas. Taip formaliai užrašytų taisyklių nesupras užsakovai (apsunkinamas patikrinimas). Tačiau jos gali būti labai detalios siejamos su duomenų modelio elementais ir interpretuojamos vienareikšmiškai. Dėl šios priežasties modelis tinkamas projektavimo fazei, kurioje eliminuojamos visos dviprasmybes. Kaip matyti 6 pav., saugykla galėtų būti integruojama į VT grindžiamos sistemos architektūrą ir naudojama kaip VT saugykla.

Realizacijos lygyje, kai sistema jau veikia naudodama sukauptas VT, įrankis galėtų būti naudojamas redaguojant taisykles ar sukuriant naujas.

### 3.1 Panaudojimo atvejai

Iš panaudojimo atvejų diagramos (9 pav.) matyti, kad sistemoje yra keturi pagrindiniai panaudojimo atvejai:

1. Veiklos taisyklės grafinio modelio generavimas, nuskaitant duomenis iš saugyklos.
2. Papildomų veiklos taisyklės duomenų įvedimas, t. y. veiklos taisyklės šaltinis ir grupė, konstanta.
3. Struktūrizuotos veiklos taisyklės išsaugojimas saugykloje.
4. Išsaugotos struktūrizuotos veiklos taisyklės grafinis redagavimas.



9 pav. Panaudojimo atvejų diagrama

Toliau pateiktas pagrindinių panaudojimo atvejų aprašymas.

#### 3.1.1 VT modelio įkėlimas iš saugyklos

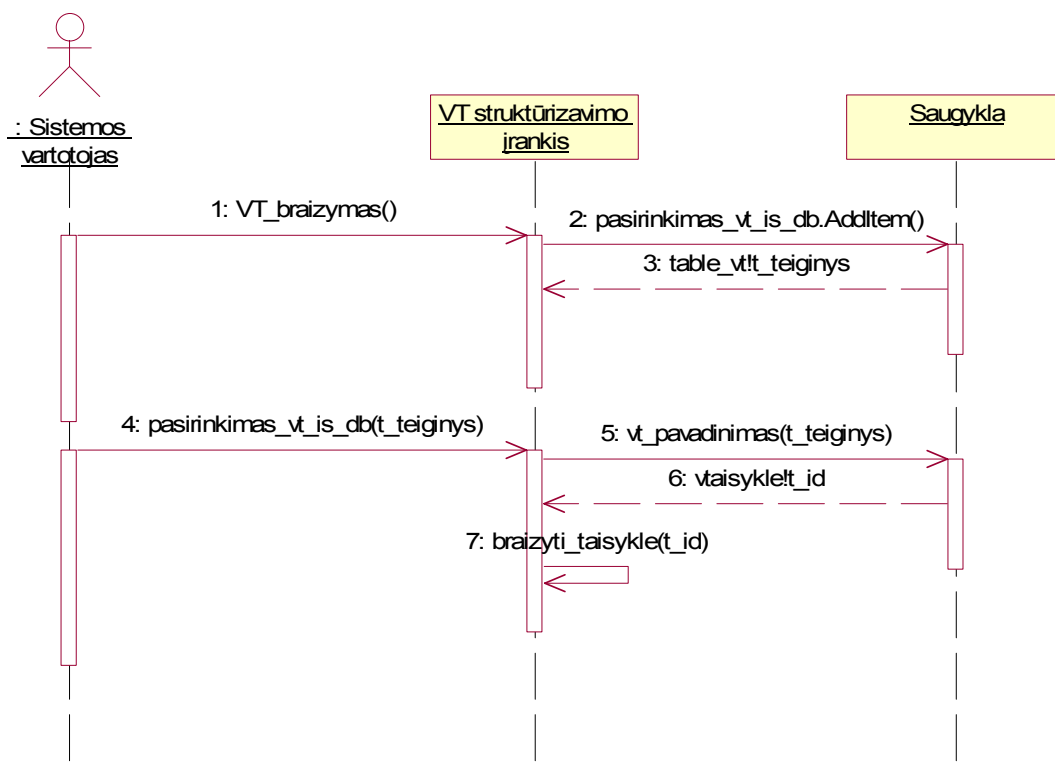
Tikslas: Sugeneruoti veiklos taisyklės grafinį modelį, nuskaitant duomenis iš saugyklos.

Aktoriai: Sistemos vartotojas.

Pagrindinis scenarijus:

1. Pasirenkamas meniu punktas *Nubraižyti VT grafiką iš DB*.
2. Visų taisyklių esančių saugykloje nuskaitymas.
3. Pasirenkama taisyklė, kuri bus atvaizduota grafiškai.
4. Veiklos taisyklės grafinis generavimas, nuskaitant duomenis iš saugyklos.

Panaudojimo atvejo sekų diagrama:



10 pav. VT modelio įkėlimo iš saugyklos veiksmų sekų diagrama

### 3.1.2 Papildomų duomenų įvedimas

Tikslas: Įvesti veiklos taisyklei priklausančius papildomus duomenis: grupę, šaltinį, konstantą.

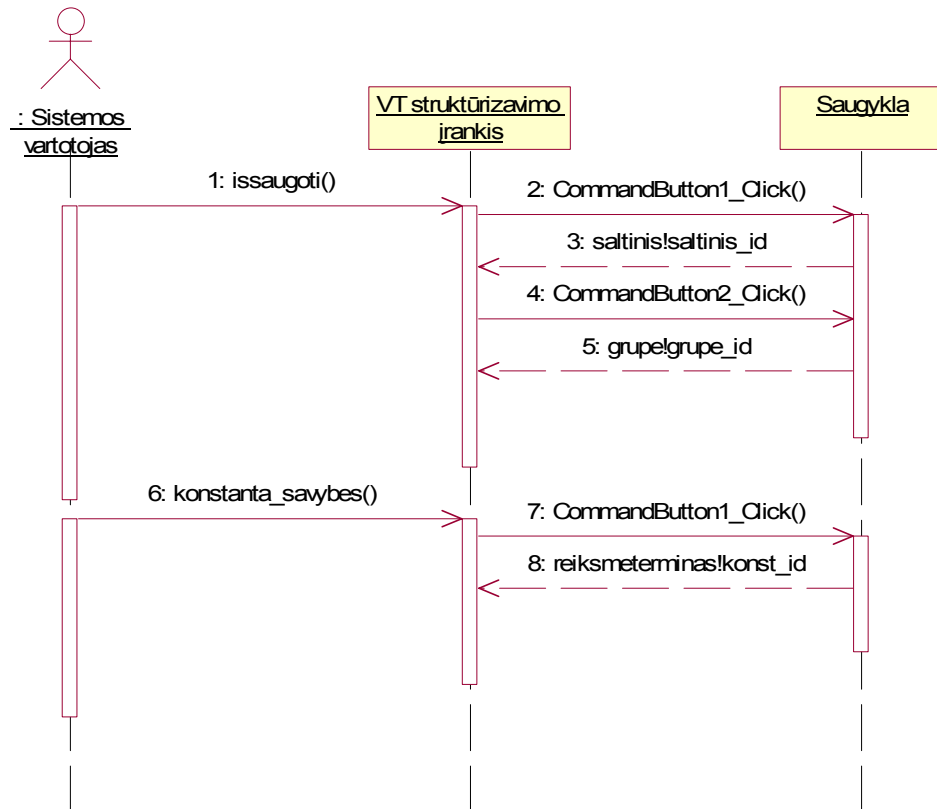
Aktoriai: Sistemos vartotojas.

Pagrindinis scenarijus:

1. Pasirenkamas meniu punktas *Išsaugoti VT grafiką į DB*.
2. Atidaroma veiklos taisyklės išsaugojimo forma.
3. Įvedamas naujas VT šaltinis ir grupė.
4. Pažymėjus konstantą pasirenkamas meniu punktas *Savybės*.
5. Įvedama nauja konstanta.

Panaudojimo atvejo sekų diagrama:





11 pav. Papildomų duomenų įvedimo veiksmų sekų diagrama

### 3.1.3 VT išsaugojimas saugykloje

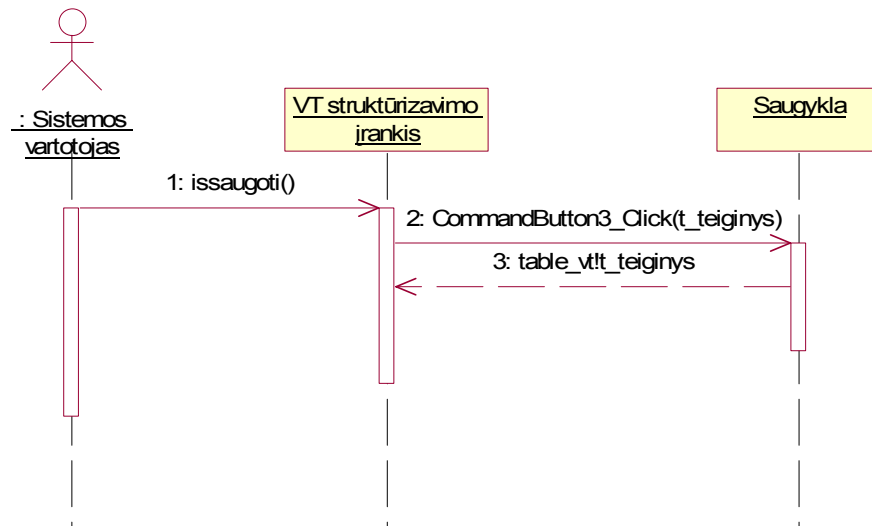
Tikslas: Išsaugoti veiklos taisyklės grafinį modelį saugykloje.

Aktoriai: Sistemos vartotojas.

Pagrindinis scenarijus:

1. Pasirenkamas meniu punktas *Išsaugoti VT grafiką į DB*.
2. Atidaroma veiklos taisyklės išsaugojimo forma.
3. Įvedama VT formuluotė.
4. Pasirenkami papildomi duomenys.
5. Išsaugojama VT saugykloje.

Panaudojimo atvejo sekų diagrama:



12 pav. VT išsaugojimo saugykloje veiksmų sekų diagrama

### 3.1.4 VT redagavimas

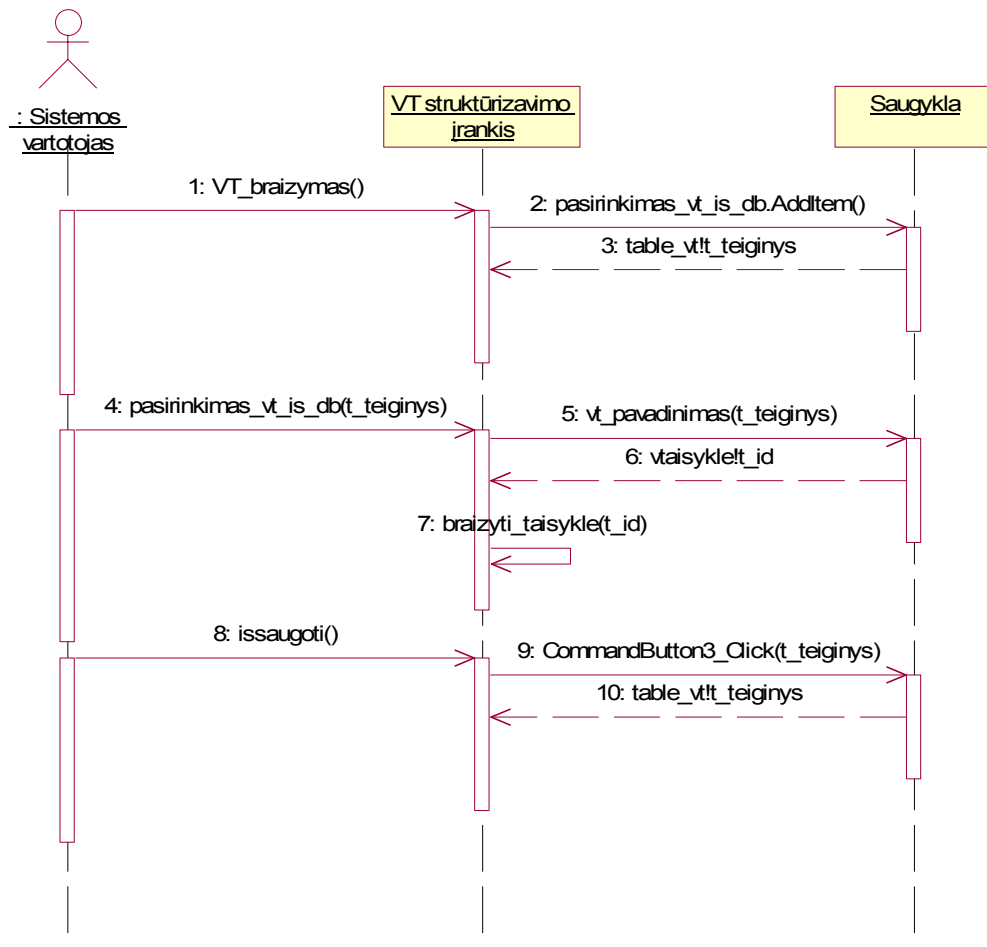
Tikslas: Išsaugotos veiklos taisyklės grafinis redagavimas.

Aktoriai: Sistemos vartotojas.

Pagrindinis scenarijus:

1. Pasirenkamas meniu punktas *Nubraižyti VT grafiką iš DB*.
2. Visų taisyklių esančių saugykloje nuskaitymas.
3. Pasirenkama taisyklė, kuri bus atvaizduota ir po to redaguojama grafiškai.
4. Veiklos taisyklės grafinis generavimas, nuskaitytant duomenis iš saugyklos.
5. VT modelio grafinis redagavimas.
6. Pasirenkamas meniu punktas *Išsaugoti VT grafiką į DB*.
7. Atidaroma veiklos taisyklės išsaugojimo forma.
8. Jeigu reikia koreguojami papildomi duomenys.
9. Išsaugojama redaguota VT saugykloje.

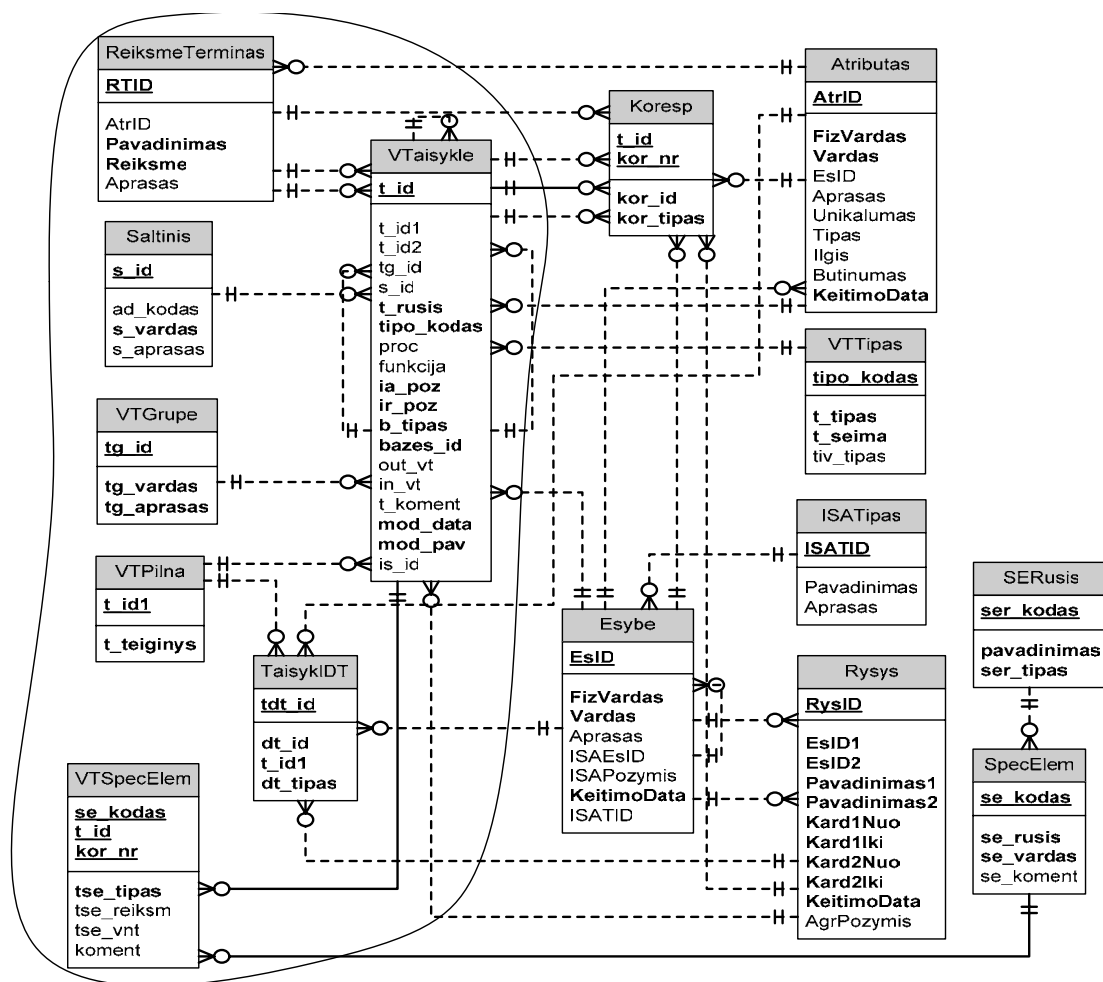
Panaudojimo atvejo sekų diagrama:



13 pav. VT redagavimas veiksmų sekų diagrama

### 3.2 Duomenų bazės modelis

Veiklos taisyklių struktūrizavimo įrankis naudoja *MS Access* duomenų bazę. 14 pav pavaizduota įrankio naudojama duomenų bazės schema. Ši saugykla naudojama informacijos saugojimui bei išgavimui.



14 pav. VT saugyklos modelis

VT struktūrizavimo įrankis informacijos saugojimo atveju pildo ne visas duomenų bazės lenteles. Išsaugoma informacija šiose lentelėse (14 pav. apibrėžtos linija):

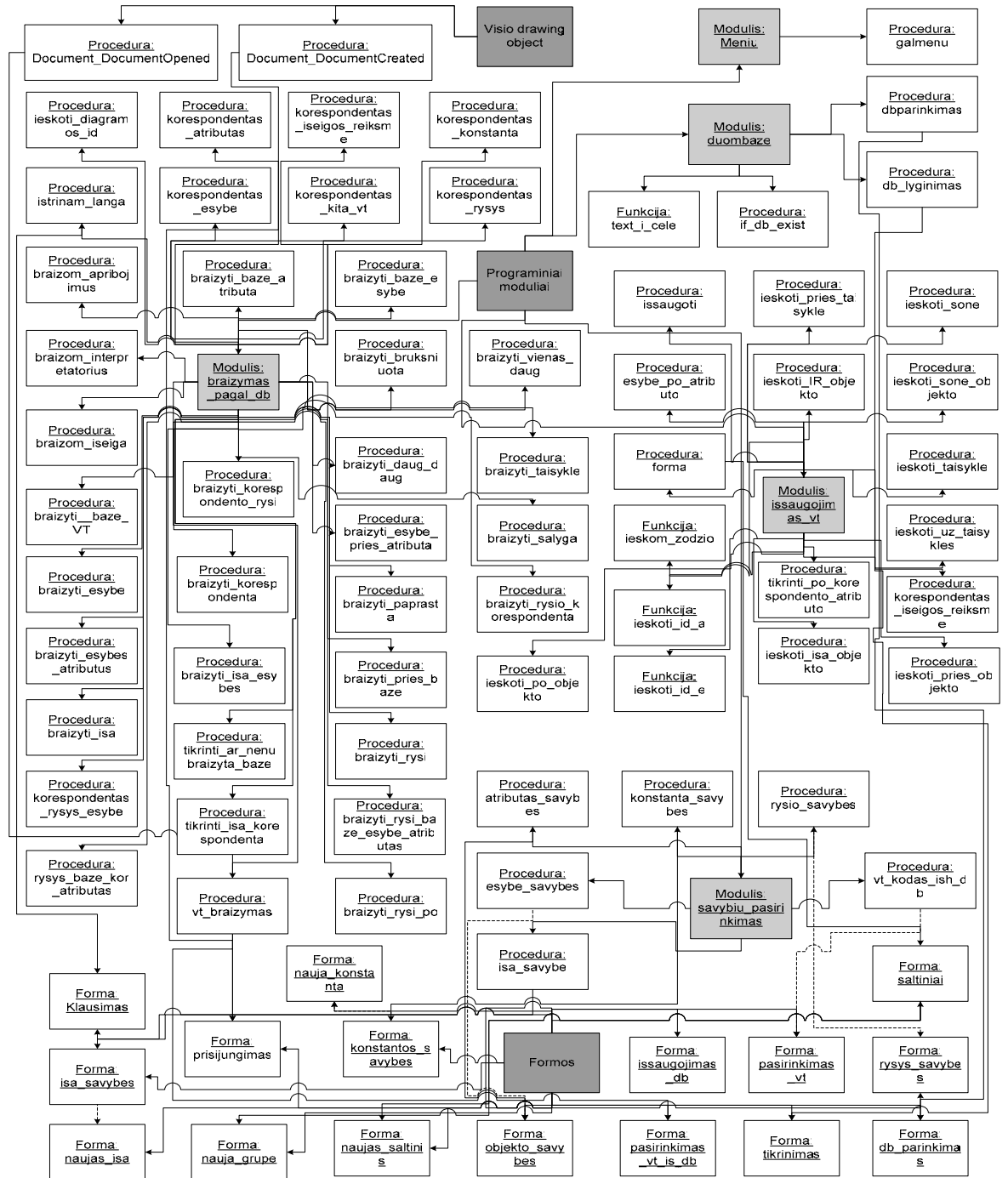
1. *VTPilna*. Veiklos taisyklės modelio formuluotė.
2. *VTaisykle*. Taisyklės tipas, bazės, korespondento, šaltinio, grupės, konstantos identifikaciniai numeriai. Nurodoma ar modeliui priklauso specialieji elementai, išieigos reikšmė.
3. *TaisyklIDT*. Veiklos taisyklės modeliui priklausančių papildomų elementų (korespondentas, bazė, ryšys, ISA esybė) identifikaciniai kodai.
4. *ReiksmeTerminas*. Veiklos taisyklės modeliui priklausanti konstanta.
5. *Saltinis*. Veiklos taisyklės šaltinis.
6. *VTGrupe*. Veiklos taisyklės grupė.
7. *VTSpecElem*. Veiklos taisyklės modeliui priklausančių specialiųjų elementų identifikaciniai kodai

Likusios lentelės nepildomos, jose saugoma taisyklių tipai *VVTipus*, korespondentai ir bazė *Esybe* ir *Atributas*, specialieji elementai *SpecElem* ir *SERusis*, ryšio tipai *Rsys*, ISA esybės *ISATipus*.

VT struktūrizavimo įrankis informacijos išgavimo atveju, t. y. veiklos taisyklės grafinio modelio generavimo metu, naudoja visas arba beveik visas saugyklos esybes. Tai priklauso nuo to, ar modelyje yra specialieji elementai, išeigos reikšmės, konstantos, ISA esybės.

### **3.3 Programinių komponentų architektūra**

Veiklos taisyklių struktūrizavimo įrankio programinių komponentų architektūra pavaizduota 15 pav. Šiame modelyje parodyti pagrindiniai įrankio programiniai komponentai - Visio drawing object, moduliai, formos ir jiems priklausomi komponentai bei jų sąsajos. Komponentų Visio drawing object, moduliai ir formos ir jiems priklausančių komponentų sąsajos pavaizduotos ištisinėmis linijomis. Nuorodos į formas pavaizduotos punktyrinėmis linijomis.



15 pav. Programinių komponentų architektūra

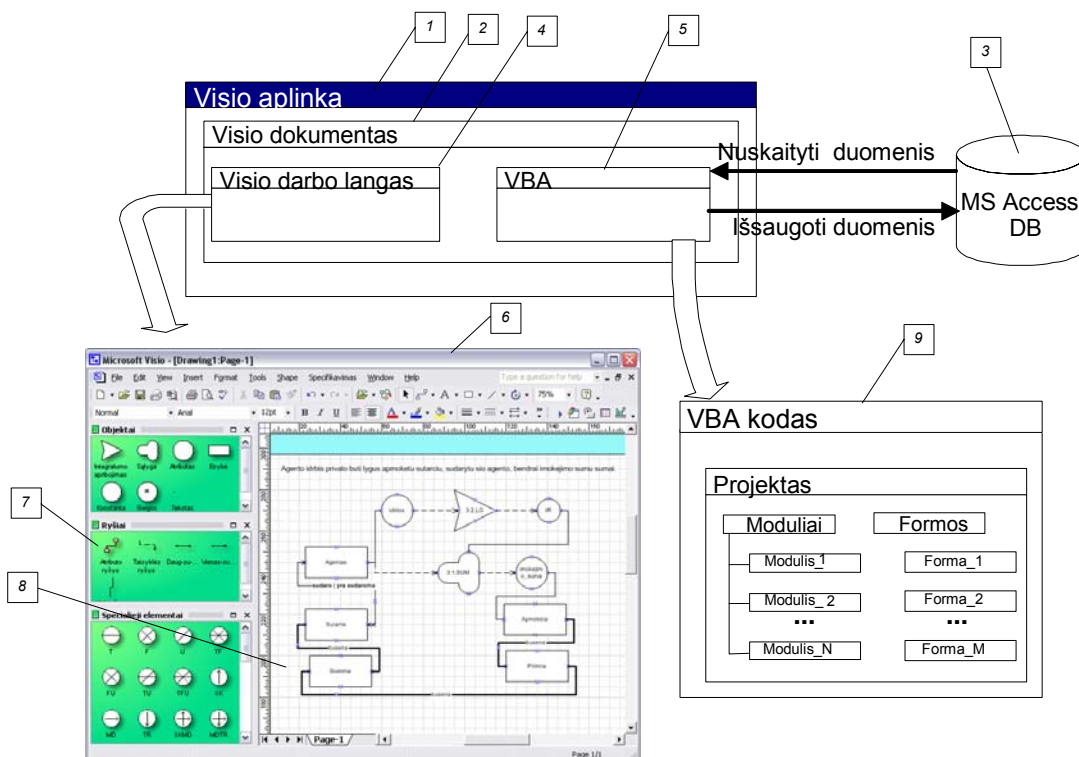
## 4 VEIKLOS TAISYKLIŲ STRUKTŪRIZAVIMO ĮRANKIO PROGRAMINĖS REALIZACIJOS APRAŠYMAS

### 4.1 Programų sistemos kūrimui naudoti įrankiai

Sukurtas veiklos taisyklių struktūrizavimo įrankis yra savarankiška sistema. Programa sukurta *MS Windows* operacinės sistemos pagrindu. *Microsoft Visio 2000* paketas panaudotas kaip grafinė aplinka, kadangi palaiko *Visual Basic for Applications* programavimo kalbą ir tenkina kuriamai sistemai keliamus reikalavimus. Programavimo kalba *Visual Basic for Applications (VBA)*. Veiklos taisyklėms saugoti sukurta *Microsoft Access* duomenų bazė.

### 4.2 Įrankio principinė schema

Realizuojant įrankį, kaip grafinė aplinka pasirinktas *Microsoft Visio 2000* paketas. Specifikacijai sudaryti ir saugoti grafinėje notacijoje naudojamas šiam tikslui paruoštas failas-šablonas. Visa specifikacija taip pat yra išsaugoma specialiai sukurtoje veiklos taisyklių saugykloje. Principinė įrankio struktūros schema pateikta 16 pav.



16 pav. VT struktūrizavimo įrankio principinė schema

- 1 – *Ms Visio* aplinka;
- 2 – *Ms Visio* dokumentas;
- 3 – *Ms Access* duomenų bazė;
- 4 – *Ms Visio* dokumento grafinė dalis;

5 – *Ms Visio* dokumento programinė dalis;

6 – *Ms Visio* dokumento grafinė dalis sudaryta iš tokių dalių:

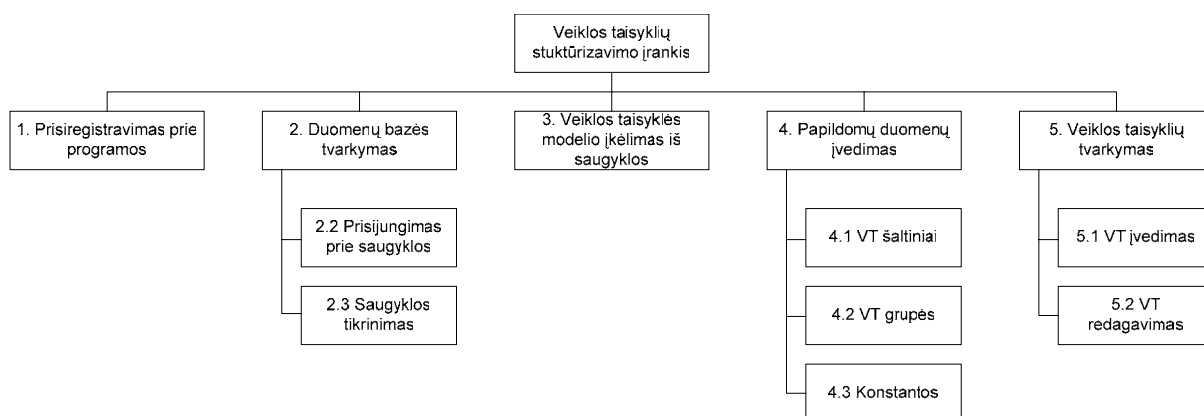
7 - įrankių juosta, kurių gali būti kelios ir saugomos atskiruose failuose;

8 - vaizduojamoji dalis, skirta veiksmams atlikti.

9 – *VBA* kodas, sudarantis programinę dalį. Su *MS Visio* dokumentu susijęs vienas projektas, kurį sudaro programiniai moduliai ir ekrano formos.

### 4.3 Veiklos taisyklių struktūrizavimo įrankio funkcinis aprašymas

Kuriamas veiklos taisyklių struktūrizavimo įrankis atlieka funkcijas pavaizduotas 17 pav.

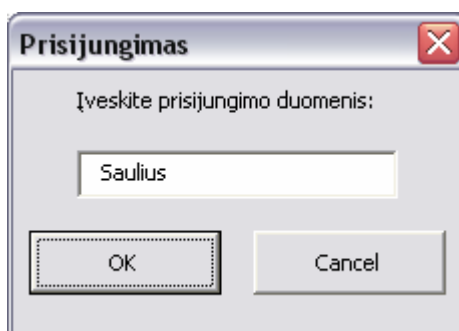


17 pav. VT struktūrizavimo įrankio atliekamos funkcijos

Toliau pateikiamas kiekvienos funkcijos aprašymas.

1. Prisijungimas prie programos.

VT struktūrizavimo įrankis paleidžiamas pasirinkus failą *vt\_specifikavimas.vst*. Tuo pačiu metu yra iškviečiama prisiregistravimo prie saugyklos funkcija. Programa išveda naują langą, kuriame reikia įrašyti dirbančiojo su sistema vartotojo vardą (18 pav.).



18 pav. Registravimosi prie programos langas

2. Duomenų bazės tvarkymas.

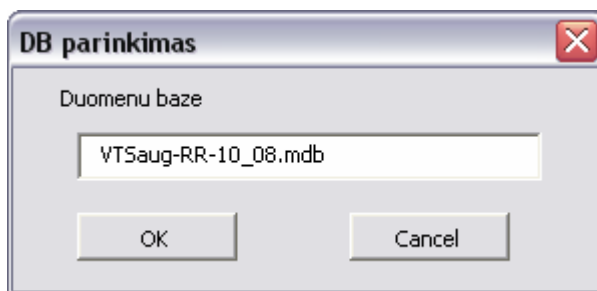
Duomenų bazės tvarkymas susideda iš dviejų funkcijų:

2.1. Prisijungimas prie saugyklos.

Šios funkcijos vykdymo metu yra prisijungiama prie veiklos taisyklių saugyklos. Funkcija iškviečiama pasirenkant programos meniu punktą *Specifikavimas->Pasirinkti DB*. Programa



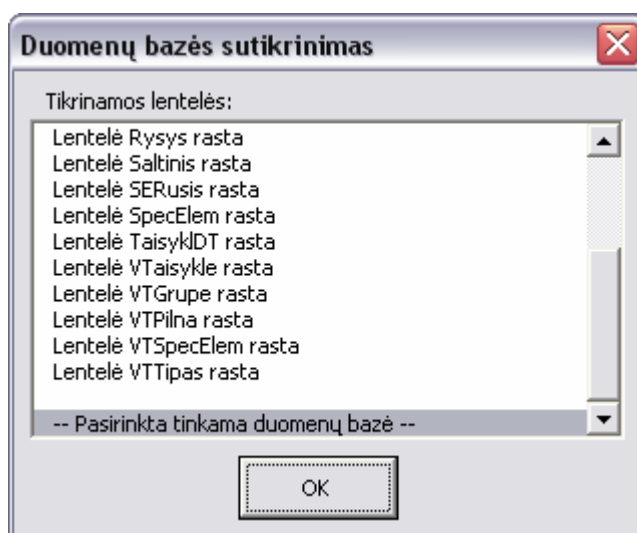
išveda naują langą, kuriame reikia įrašyti duomenų bazės vardą ir pilną kelią (19 pav.). Jeigu saugyklos failas yra tame pačiame kataloge kaip ir programa, pakanka įrašyti tik saugyklos failo pavadinimą.



19 pav. Prisijungimo prie saugyklos forma

## 2.2. Saugyklos tikrinimas.

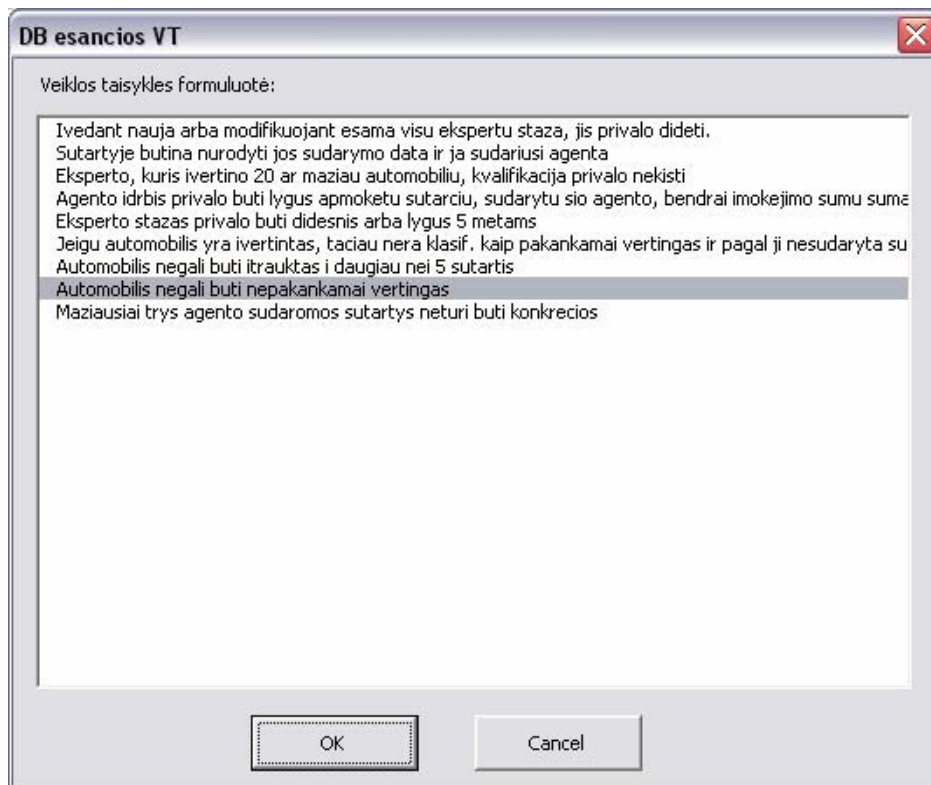
Šios funkcijos pagalba galima nustatyti ar prie veiklos taisyklių saugyklos yra prisijungta. Funkcija iškviečiama pasirenkant programos meniu punktą *Specifikavimas->Tikrinti DB*. Programa išveda naują langą, kuriame yra informacija apie patikrintas saugyklos lenteles (20 pav.).



20 pav. Saugyklos patikrinimo rezultatų langas

## 3. Veiklos taisyklės modelio įkėlimas iš saugyklos.

Tai VT grafinio modelio generavimas, nuskaitant duomenis iš saugyklos. Ši funkcija iškviečiama pasirenkant programos meniu punktą *Specifikavimas->Nubraižyti VT grafiką iš DB*. Tuomet programa išveda naują langą, kuriame yra visų VT formuluotės, kurios saugomos saugyklos lentelėje *VTPilna* (21 pav.). Pasirinkus reikiamą formuluotę, sugeneruojamas VT grafinis modelis. Generuojant modelį, programa kreipiasi į beveik visas saugyklos lenteles – tai priklauso nuo to, ar modelyje yra specialieji elementai, išieigos reikšmes, konstantos ir kt.



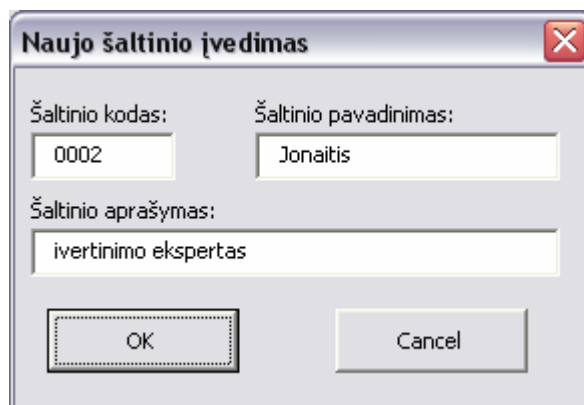
21 pav. VT pasirinkimo langas

#### 4. Papildomų duomenų įvedimas.

Šios funkcijos vykdymo metu gali būti įvedama informacija apie:

##### 4.1. VT šaltinius.

Prieš išsaugant naują arba redaguojant jau esančią saugykloje VT, būtina nurodyti taisyklės šaltinį. Informacija apie jį saugoma saugyklos lentelėje *Saltinis*. Funkcija iškviečiama, kai išsaugojamas VT modelis. Tuomet programos lange galima pasirinkti iš saugykloje jau esančių šaltinių arba įvesti naują. Įvedant naują VT šaltinį reikia nurodyti jo kodą, pavadinimą ir trumpą aprašymą (22 pav.).

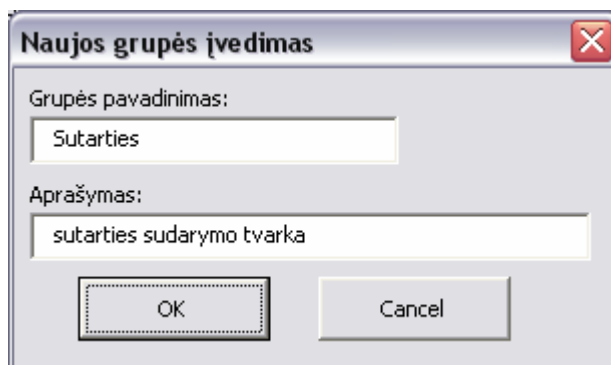


22 pav. Naujo šaltinio įvedimo langas

##### 4.2. VT grupes.

VT grupę, kaip ir jos šaltinį, būtina nurodyti prieš išsaugant naują arba redaguojant esamą grafinį VT modelį. Informacija apie grupę saugoma saugyklos lentelėje *VTGrupe*. Funkcija

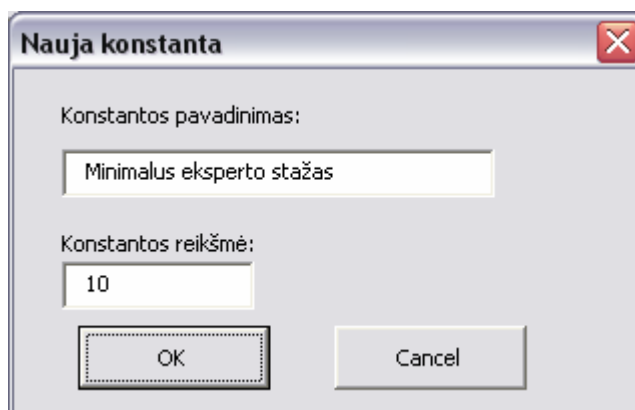
iškviečiama tuomet, kai išsaugojamas VT modelis. Tuomet programos lange galima pasirinkti iš jau anksčiau užfiksuotų grupių arba įvesti naują. Įvedant naują veiklos taisyklės grupę, būtina nurodyti jos pavadinimą ir trumpą aprašymą (23 pav.).



23 pav. Naujo šaltinio įvedimo langas

#### 4.3. Konstantas.

Įkėlus konstantos tipo elementą į VT modeliavimo langą, reikia nustatyti konstantos reikšmę. Visa informacija apie konstantas saugoma saugyklos lentelėje *ReiksmeTerminas*. Funkcija iškviečiama dukart spragtelėjus ant įkelto duomenų modelio elemento arba pasirenkant jo funkciją *Savybės*. Tuomet programa atidaro langą, kuriame galima pasirinkti konstantos reikšmę iš esamų saugykloje arba įvesti naują. Įvedant naują reikšmę, reikia nurodyti ir konstantos pavadinimą (24 pav.).



24 pav. Naujos konstantos įvedimo langas

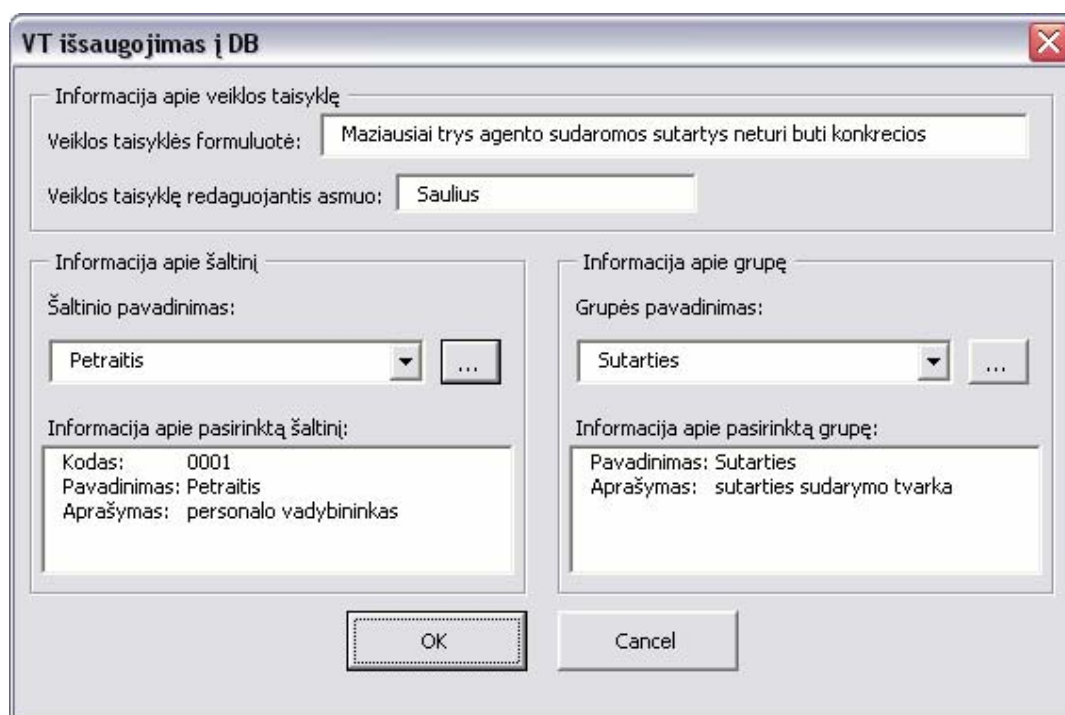
### 5. Veiklos taisyklių tvarkymas.

Veiklos taisyklių grafinių modelių tvarkymas susideda iš dviejų funkcijų:

#### 5.1. VT įvedimas.

Pasirinkus šią funkciją, saugykloje yra išsaugojama nauja struktūrizuota VT. Funkcija iškviečiama pasirinkus meniu punktą *Specifikavimas->Išsaugoti VT modelį į DB*. Programa atidaro naują langą, kuriame būtina nurodyti naujos VT formuluotę ir papildomus duomenis – VT šaltinį ir grupę (25 pav.). Išsaugant modelį, informacija saugoma į saugyklos lenteles

*VTPilna, VTaisykle, TaisyklDT, Koresp.* Jei VT grafiniame modelyje yra specialiųjų elementų, informacija išsaugojama lentelėje *VTSpecElem*, konstantų – *ReiksmeTerminas*.



25 pav. VT išsaugojimo saugykloje programos langas

## 5.2. VT redagavimas.

Prieš redaguojant saugykloje jau saugomą VT, ją reikia atvaizduoti grafiškai. Tai atliekama pasirenkant pirmąją funkciją – veiklos taisyklės modelio įkėlimas iš saugyklos. Tuomet, atlikus pakeitimus, galima VT išsaugoti į saugyklą. Funkcija iškviečiama pasirinkus meniu punktą *Specifikavimas->Išsaugoti VT modelį į DB*. Programa atidaro naują langą, kuriame jau yra įrašyta VT formuluotė, grupė ir šaltinis (25 pav.). Grupę ir šaltinį galima keisti naudojant funkcijas *VT šaltiniai* ir *VT grupės*. Išsaugant pakeistą grafinį veiklos taisyklių modelį, informacija saugoma tose pačiose lentelėse, kaip ir funkcijos *VT įvedimas* atveju.

## 4.4 Programų sistemos įdiegimas

Programų sistemą sudaro šie failai:

- 1) *objektai.vss*;
- 2) *ryšiai.vss*;
- 3) *specelem.vss*;
- 4) *vt\_specifikavimas.vst*;
- 5) *vt\_saug.mdb*.

Programų sistemos paleidžiamasis failas yra failas *vt\_specifikavimas.vst*.

Kompiuteryje, kuriame bus diegiama sukurta programų sistema, turi būti įdiegta *Windows* operacinė sistema. Kompiuteryje turi būti įdiegtas *Microsoft Visio 2000* paketas. Jis naudojamas kaip grafinė aplinka sukurtam produktui. Taip pat kompiuteryje turi būti įdiegtas *MS Access* paketas.

Techninei įrangai specialių reikalavimų nekeliami, tačiau viena iš kompiuterio, kuriuo realizuojamas projektas, charakteristikų – operatyvinės atminties turi būti 256 RAM ar daugiau. Tai užtikrina greitesnį ir patogesnį darbą, nes esant mažesniai operatyvios atminties kiekiui, dirbti su *Microsoft Visio* paketu nėra taip patogiu.

## 5 VT STRUKTŪRIZAVIMO ĮRANKIO EKSPERIMENTAS

### 5.1 Eksperimento koncepcija

Eksperimentas buvo atliekamas siekiant praktiškai išbandyti sukurtą VT struktūrizavimo įrankį. Pasiruošimas eksperimentui susidėjo iš šių etapų:

1. Dalykinės srities parinkimas.
2. Pasiūlyto VT saugyklos modelio programinis realizavimas.

Eksperimentas buvo atliekamas tokia tvarka:

1. Remiantis dalykinės srities charakteristika buvo sudaromas testinis VT rinkinys. Išskirtos VT buvo užrašomos neformaliais teiginiais.
2. Remiantis VT modeliavimo metodika, išskirtos taisyklės buvo užrašomos formaliu pavidalu, t.y. diagramomis.
3. Formaliai aprašytos VT buvo įvedamos į sukurtą taisyklių saugyklos duomenų bazę naudojant veiklos taisyklių struktūrizavimo įrankį.

### 5.2 Dalykinė sritis

#### 5.2.1 Dalykinės srities parinkimas

Eksperimentui buvo nuspręsta pasirinkti abstraktų automobilių techninės apžiūros (TA) procesą. Charakterizuojant pasirinktą sritį nebuvo stengiamasi pasiekti aukštą išsamumo ar tikslumo lygį. Pagrindinė dalykinės srities paskirtis – inicijuoti įvairaus pobūdžio VT, leidžiančias įvairiapusiškai ištestuoti Roso taisyklių modeliavimo metodiką naudojant VT struktūrizavimo įrankį bei užpildyti taisyklių saugyklą.

#### 5.2.2 Dalykinės srities charakteristika

Nagrinėjama konceptuali automobilių techninės apžiūros sistema. Norintis atlikti automobilio techninę apžiūrą asmuo (toliau *savininkas*) bendrauja su TA bendrovės atstovu (toliau *ekspertas*). Savininkas kreipiasi į TA kasą (toliau *kasa*) su prašymu užregistruoti jo automobilį. Kasa prašymo pagrindu pateikia užduotį ekspertui, išrašydama pažymą. Jeigu automobilis tikrinamas pirmą kartą, ekspertas patikrina pilnai automobilį, priešingu atveju, tikrina ar pašalinti anksčiau aptikti gedimai. Ekspertui nustačius, kad automobilis tvarkingas, jis išduoda TA patikrinimo taloną.

## **5.3 Veiklos taisyklių specifikuojimas**

### **5.3.1 Veiklos taisyklių rinkinio sudarymas**

Pagal pasirinktą dalykinę sritį buvo sudarytas VT formuluočių rinkinys. Išskiriant testines taisykles nebuvo siekiama išsamiai ir tiksliai specifikuoti dalykinę sritį. Pagrindinis tikslas buvo parinkti nedaug skirtingo pobūdžio VT, kurios atspindėtų kuo įvairesnes struktūrines kombinacijas.

Buvo suformuluoti šie VT teiginiai:

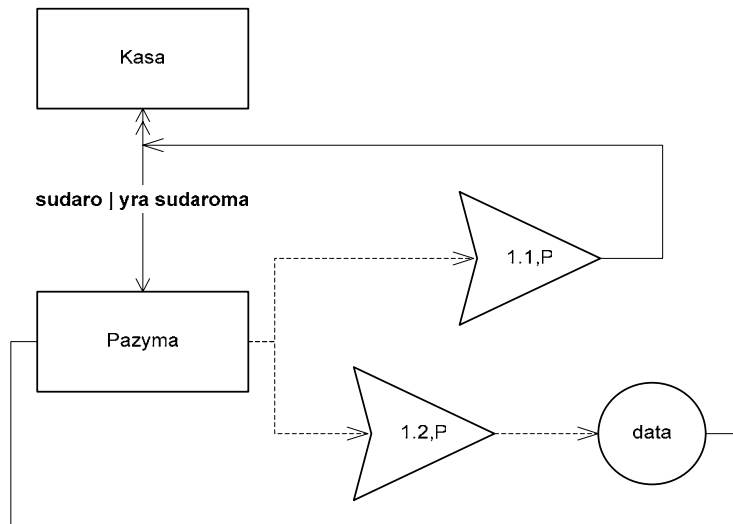
1. TA apžiūros pažymoje būtina nurodyti jos atlikimo datą ir ją sudariusią kasą.
2. Ekspertas, kuris patikrino 100 ar mažiau automobilių, kvalifikacija privalo nekisti.
3. Eksperto stažas privalo būti didesnis arba lygus 1 metams.
4. Atliekant naują arba pakartotinę TA, apžiūrų kiekis privalo didėti.
5. Automobilis negali turėti daugiau nei 1 TA patikrinimo taloną.

### **5.3.2 Veiklos taisyklių formalizavimas pagal Roso metodą**

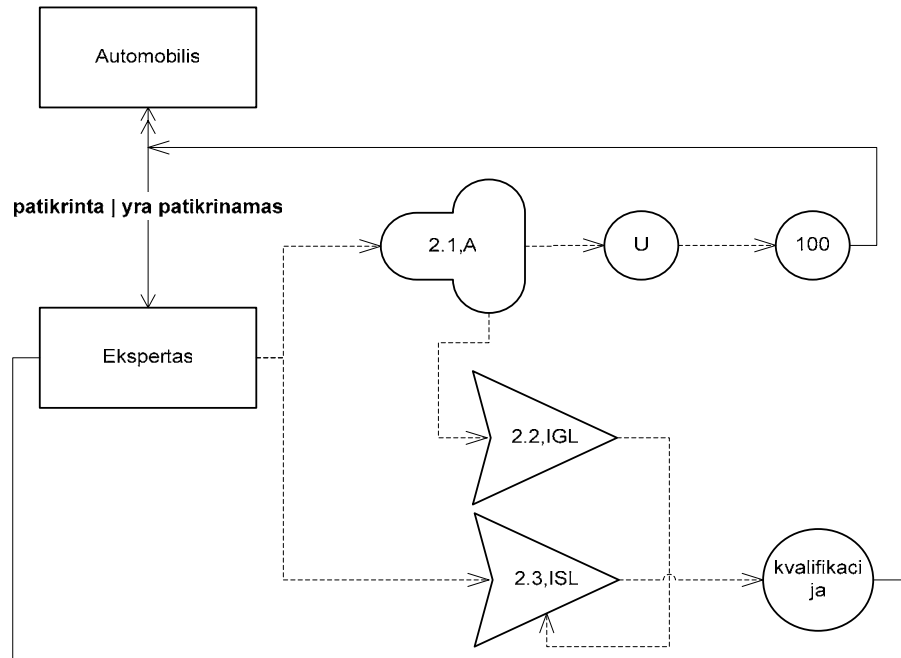
Vadovaujantis dalykinės srities charakteristika ir suformuluotais veiklos taisyklių teiginiais buvo braižomos šių taisyklių diagramos, naudojant VT struktūrizavimo įrankį. Visos taisyklės yra numeruojamos, įrašant numerį prie taisyklės tipo. Jei veiklos taisyklės modelį sudaro kelios atskiros taisyklės, tai jos numeruojamos skaičiumi iš dviejų dalių:

- pirmoji numerio dalis arba skaičius iki taško – tai logiškai nedalomos VT, kurios dalimi yra konkreti taisyklė, numeris;
- antroji numerio dalis arba skaičius po taško – tai sąlyginis šioje VT unikalus numeris.

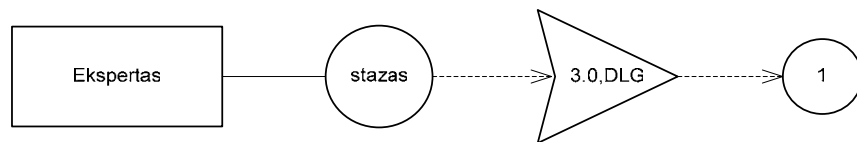
Žemiau pateikiamos veiklos taisyklių grafiniai modeliai, sumodeliuoti naudojant VT struktūrizavimo įrankį.



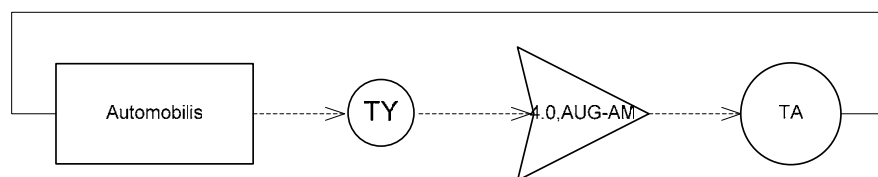
26 pav. 1 VT grafinis modelis



27 pav. 2 VT grafinis modelis

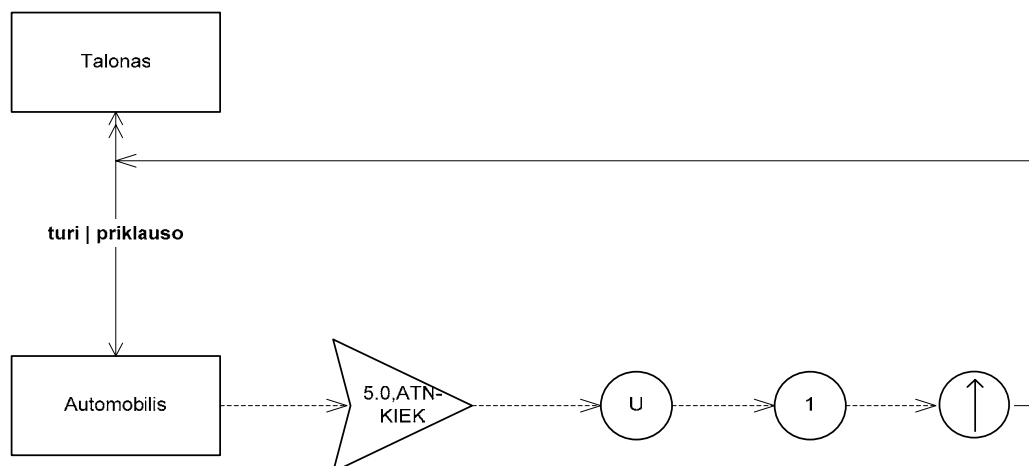


28 pav. 3 VT grafinis modelis



29 pav. 4 VT grafinis modelis





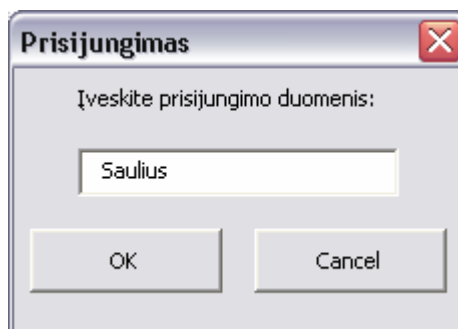
30 pav. 5 VT grafinis modelis

## 5.4 Veiklos taisyklių saugyklos panaudojimas VT struktūrizavimo įrankio pagalba

### 5.4.1 Veiklos taisyklių saugyklos pildymas

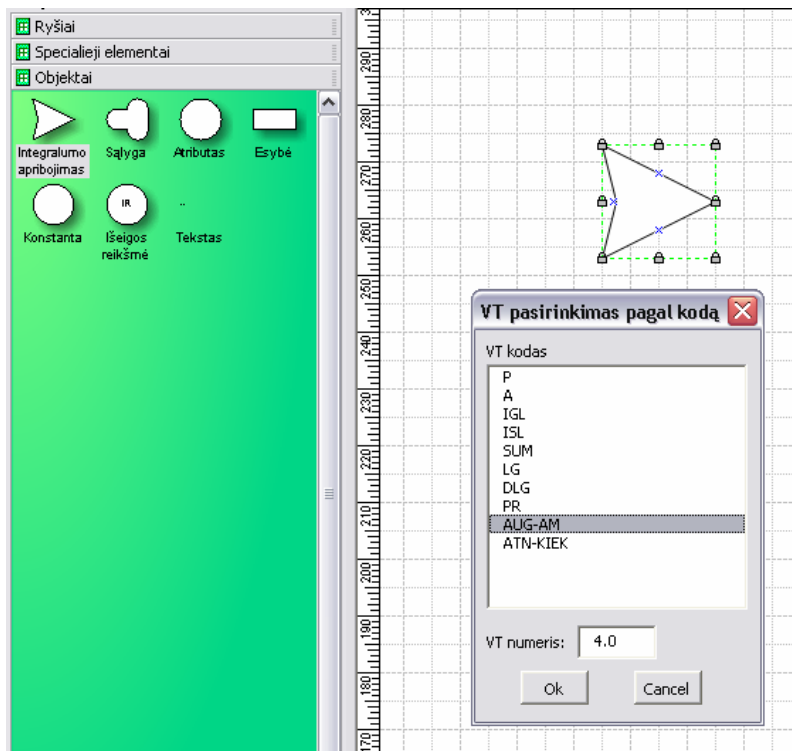
Išsaugant veiklos taisyklės grafinį modelį saugykloje, įvedama visa informacija matoma modelyje. Šiam tikslui buvo sukurtas VT struktūrizavimo įrankis.

Prieš pradėdamas darbą, vartotojas privalo prisiregistruoti (31 pav.).



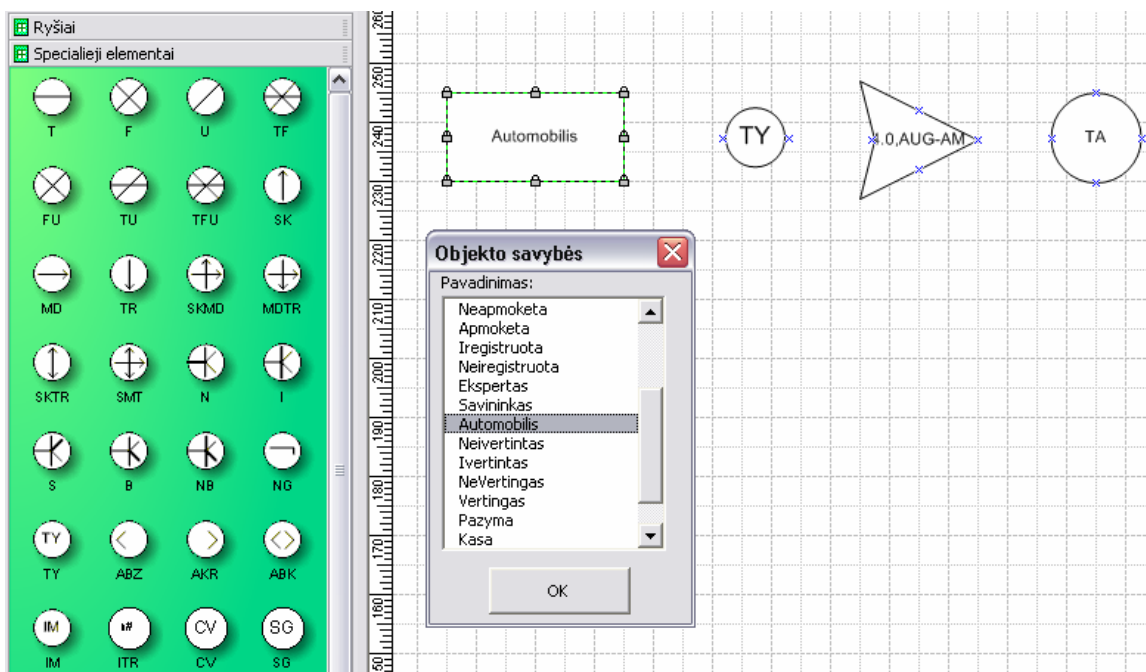
31 pav. Prisiregistravimo forma

Pirmiausia į programos darbo langą įkeliami ir specifikuojama taisyklė (sąlyga arba integralumo apribojimas). Būtina nurodyti taisyklės tipą ir numerį (32 pav.)



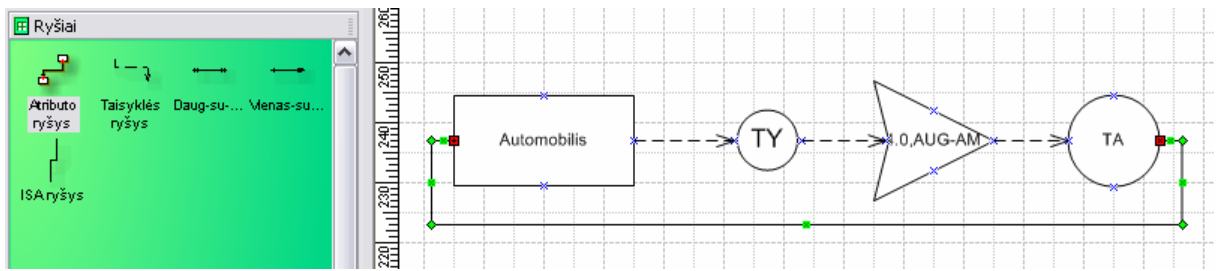
32 pav. Integralumo apribojimo įkėlimas ir specifikuojimas

Toliau į programos darbo langą įkeliama ir specifikuojama bazė, korespondentas (-ai). Jei veiklos taisyklei priklauso įkeliami specialieji elementai, išieigos reikšmė, konstantos (33 pav.).



33 pav. Papildomų objektų įkėlimas ir specifikuojimas

Įkėlus visus objektus į programos darbo langą ir juos specifikuojus, belieka sudėti ryšius (34 pav.).



34 pav. VT grafinis modelis

Pilnai nubraizius veiklos taisyklės grafinį modelį, galime jį įvesti į saugyklą. Informacijos apie VT įvedimą forma pateikta 35 pav. Čia yra nurodomos šios savybės:

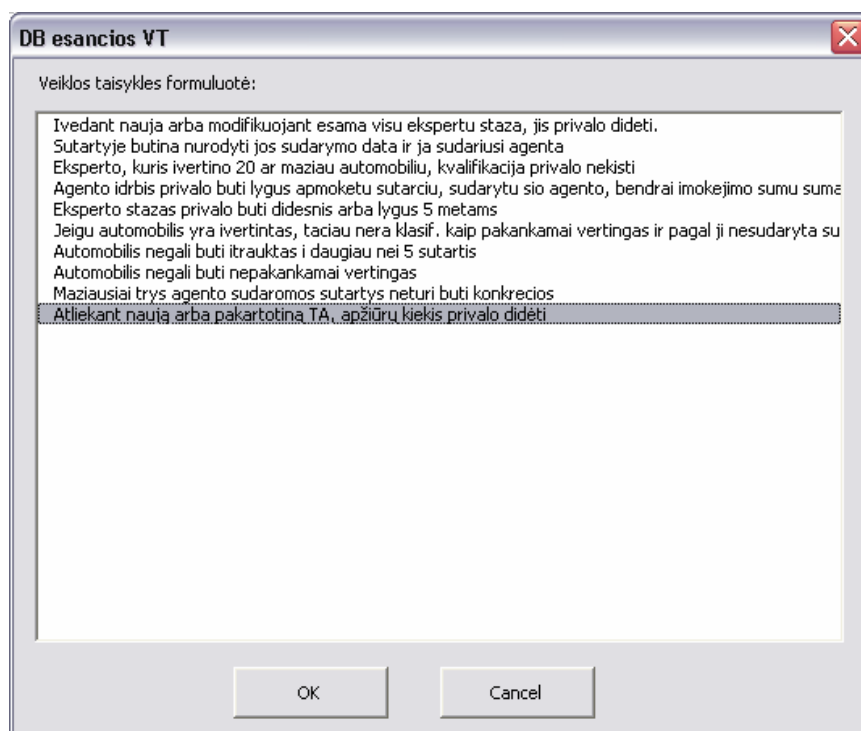
- Veiklos taisyklės formuluotė. Ji gali būti vienoda keletui VT, jeigu jos priklauso tai pačiai logiškai nedalomai taisyklei.
- Veiklos taisyklę redaguojantis asmuo. Čia automatiškai įrašoma informacija apie vartotoją, kuri buvo pateikta prisiregistruojant prie programos.
- Šaltinis, kuriam priklauso veiklos taisyklė. Pasirenkamas iš jau esančių saugykloje arba įvedamas naujas.
- Grupė, kuriai priklauso veiklos taisyklė. Pasirenkama iš jau esančių saugykloje arba įvedama nauja.

35 pav. VT išsaugojimo į saugyklą forma

## 5.4.2 Informacijos nuskaitymas iš veiklos taisyklių saugyklos

Generuojant grafinį veiklos taisyklės modelį, naudojama tik VT saugykloje esanti informacija. Šiam tikslui buvo sukurta forma. Kaip ir saugyklos pildymo atveju, pirmiausia reikia prisiregistruoti prie programos (31 pav.).

VT modelio generavimo forma pateikta 36 pav. Čia yra rodoma informacija apie visas veiklos taisykles, išsaugotas saugykloje. Naudojant VT struktūrizavimo įrankį, galima pasirinkti bet kurią veiklos taisyklę ir sugeneruoti jos grafinį modelį.



36 pav. VT modelio generavimo forma

## 5.5 VT struktūrizavimo įrankio vertinimas

Yra sukurta veiklos taisyklių valdymo įrankių, tokių kaip *ILOG Rules 6* [18], *BRS Rule Track 3.0* [17]. Tačiau analizuojant veiklos taisyklių specifikavimo metodus galima pastebėti, jog nėra sukurta vieninga taisyklių saugyklos struktūra, kuri leistų saugoti pagal griežtą klasifikaciją surinktas taisykles. Taip pat neišku kokia tvarka ir metodais tas taisykles išgauti bei talpinti į saugyklą, kad ateityje jas būtų galima lengvai modifikuoti, papildyti. Šiam tikslui pagal modifikuotą Roso metodą yra sukurta VT saugykla, taip pat įrankiai, skirti darbui su šia saugykla:

1. Veiklos taisyklių saugyklos pildymo programa (K. Kapočius).
2. Veiklos taisyklių struktūrizavimo įrankis (S. Baškevičius).

Toliau pateiktas veiklos taisyklės išsaugojimo saugykloje funkcijos palyginimas, dirbant su kiekviena programa.

### 5.5.1 Veiklos taisyklių saugyklos pildymo programa

Programa realizuota naudojant FoxPro programinį paketą. Veiklos taisyklėms išsaugoti saugykloje yra sukurta forma, susidedanti iš keturių langų.

Pagrindinės informacijos apie VT įvedimo lange (37 pav.) nurodomos šios savybės:

- Veiklos taisyklės numeris.
- Veiklos taisyklės teiginys (formuluotė).
- Taisyklės rūšis.
- Taisyklės tipas.
- VT šaltinis ir grupė.
- VT bazė.
- Išeigos reikšmė.
- Taisyklėje naudojama procedūra arba funkcija.

37 pav. Pagrindinės informacijos apie VT įvedimas

Antrajame VT išsaugojimo lange įvedama informacija apie korespondentus (38 pav.). Čia nurodomos šios savybės:

- Pasirenkamas korespondento pobūdis.
- Nurodomas atitinkamas korespondentas iš jau esančių saugykloje elementų.
- Korespondentui priskiriamas eilės numeris.

**Veiklos taisyklės**

Pagrindinė informacija | **Korespondentai** | Specialieji elementai | Taisyklės duomenų tipai

Eil. Nr.	Pobūdis	ID

**Lauko "Pobūdis" reikšmės:**

- D - korespondentas yra duomenų tipas
- C - korespondentas yra konstanta
- T - korespondentas yra kita taisyklė
- I - korespondentas yra kitos taisyklės išieigos reikšmė

Pobūdis:

Duomenų tipas:

Taisyklė:

Konstanta:

Korespondento numeris:

Konstanta:

Skaičius  Tekstas  Data  Laikas

Vienetai:

Išsaugoti korespondenta | Atšaukti | Išsaugoti konstantą

Įvesti naują korespondentą | Įvesti naują konstantą

38 pav. VT korespondentų įvedimo langas

Trečiajame lange įvedama VT priklausantys specialieji elementai (39 pav.). Įvedama informacija:

- Pasirenkamas specialaus elemento kodas.
- Specialaus elemento tipas.
- Specialus elemento reikšmė ir matavimo vienetai (jeigu reikia).
- Korespondento numeris.

**Veiklos taisyklės**

Pagrindinė informacija | Korespondentai | **Specialieji elementai** | Taisyklės duomenų tipai

Kodas	Tipas	Reikšmė	Eil. nr.	Korespondento nr.	Komentaras

**Lauko "Tipas" reikšmės:**

I - elementas yra interpretatorius A - elementas yra apribojimas

Specialaus elemento kodas:

Tipas: **TY** Lygio pakėlejas  A - Apribojimas (ant bazės ryšio)

Reikšmė: **SK** sukūrimas

Matavimo vienetai:

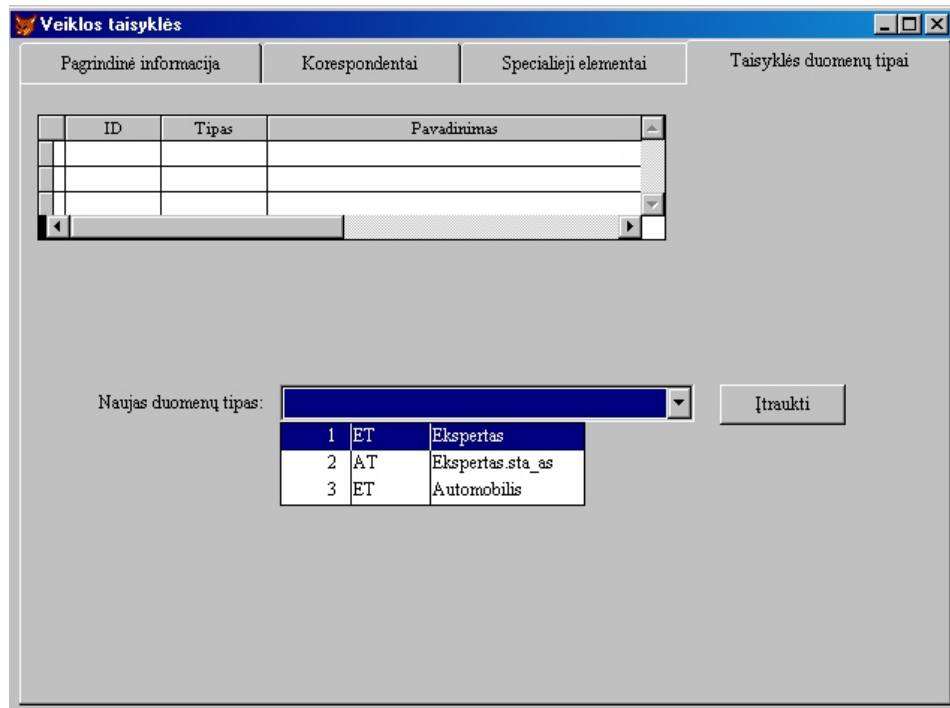
Korespondento numeris (jeigu apribojimas):

Komentaras:

Įvesti naują | Išsaugoti elementą | Atšaukti

39 pav. VT priklausančių specialiųjų elementų įvedimo langas

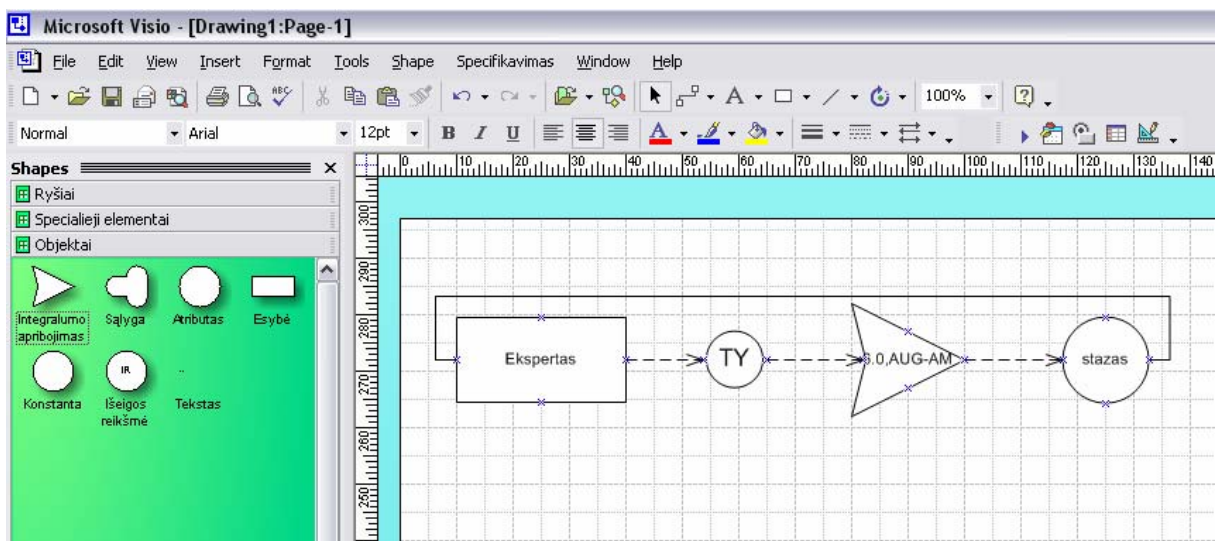
Paskutiniame programos lange įvedama informacija apie diagramoje naudojamus duomenų tipus (40 pav.). Norint įvesti duomenų tipą, užtenka pasirinkti vieną iš jau esančių saugykloje įrašų.



40 pav. VT naudojamų duomenų tipų įvedimo langas

### 5.5.2 Veiklos taisyklių struktūrizavimo įrankis

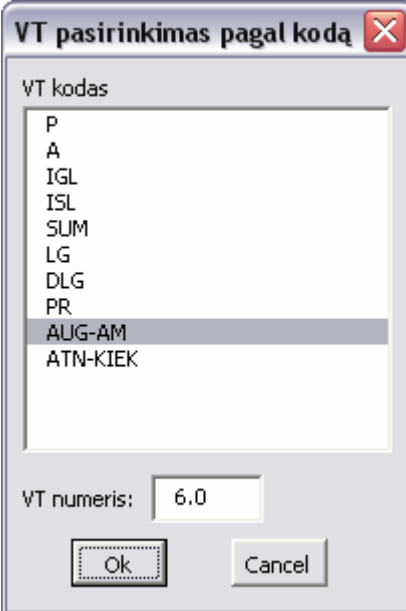
Veiklos taisyklių struktūrizavimo įrankis realizuotas *MS Visio*, *VBA* ir *MS Access* programų paketus. Norint išsaugoti veiklos taisyklę, pirmiausia reikia nubraižyti VT grafinį modelį (41 pav.).



41 pav. VT grafinio modelio braižymo langas

Įkėlus bet kurį objektą į programos langą, jį būtinai reikia specifiuoti, pasirenkant objekto meniu punktą *Savybės*. Šiam tikslui yra sukurtos formos. 42 pav. pateiktas taisyklės (integralumo apribojimo) savybių parinkimo forma, kurioje nurodyti informaciją:

- Pasirinkti taisyklės tipą.
- Įvesti taisyklės numerį.



The image shows a Windows-style dialog box with the title "VT pasirinkimas pagal kodą". Inside the dialog, there is a list box labeled "VT kodas" containing the following items: P, A, IGL, ISL, SUM, LG, DLG, PR, AUG-AM (which is highlighted with a grey background), and ATN-KIEK. Below the list box, there is a text input field labeled "VT numeris:" containing the value "6.0". At the bottom of the dialog, there are two buttons: "Ok" and "Cancel".

42 pav. Taisyklės specifkavimo forma

Įkėlus į programos langą specialųjį elementą, papildomai specifiuoti nereikia, nes grafinių elementų kodai atitinka kodams, esantiems saugykloje.

Nubraižius veiklos taisyklės grafinį modelį, belieka išsaugoti saugykloje. Šiam tikslui sukurtas VT išsaugojimo langas (43 pav.), kuris iškviečiamas pasirinkus meniu punktą *Specifikavimas->Išsaugoti VT grafiką į DB*. Lange reikia nurodyti šią informaciją:

- Veiklos taisyklės formuluotę.
- Pasirinkti arba įvesti naują VT grupę ir šaltinį.



43 pav. VT išsaugojimo langas

### 5.5.3 Įrankių palyginimo apibendrinimas

Programų vertinimui atlikti buvo pasirinkta viena iš pagrindinių funkcijų – tai veiklos taisyklės išsaugojimas saugykloje (3 lentelė). Vertinant pateiktų kriterijų atžvilgiu, VT struktūrizavimo įrankis yra pranašesnis, nes veiklos taisyklės grafinis modelis yra paprasčiau skaitomas, geriau suprantamas, patogesnis.

3 lentelė. Įrankių palyginimas

Palyginimo kriterijus	VT struktūrizavimo įrankis	VT saugyklos pildymo programa
Darbo aplinka	Naudojami specialūs grafiniai simboliai VT modelio kūrimui.	Sukurtas atributinė programos forma su keturiais langais, kuriuose įvedama informacija apie VT.
Taisyklės savybių apibrėžimas	Pasirinkus taisyklės grafinio simbolio <i>Savybių</i> meniu, pasirenkamas taisyklės tipas, užrašomas numeris.	Pirmajame formos lange įrašomas taisyklės numeris ir rūšis, pasirenkamas tipas.
Specialiųjų elementų savybių apibrėžimas	Pasirenkamas reikiamas specialusis elementas iš įrankių juostos. Papildomų savybių nustatyti nereikia.	Trečiajame formos lange pasirenkamas specialaus elemento kodas, jei reikia įrašoma reikšmė, išsaugojamas įrašas.
Bazės savybių apibrėžimas	Pasirinkus bazės grafinio simbolio <i>Savybių</i> meniu, pasirenkamas bazės tipas.	Pirmajame formos lange pasirenkamas bazės pobūdis ir duomenų tipas.
Korespondento savybių apibrėžimas	Pasirinkus korespondento grafinio simbolio <i>Savybių</i> meniu,	Antrajame formos lange pasirenkamas korespondento

<b>Palyginimo kriterijus</b>	<b>VT struktūrizavimo įrankis</b>	<b>VT saugyklos pildymo programa</b>
	pasirenkamas bazės tipas.	pobūdis, tipas. Įrašomas korespondento numeris.
VT šaltinio ir grupės įvedimas	VT išsaugojimo formoje pasirenkamas arba įvedamas naujas šaltinis ir grupė.	Pirmajame formos lange pasirenkamas šaltinis ir grupė.

## 6 IŠVADOS

Atlikta veiklos taisyklių struktūrizavimo analizė, aptarti VT sudarymo metodai, pristatytas saugyklos modelis, VT struktūrizavimo pagal Roso metodą modelis, modelio grafinė notacija, sudarymo principai.

Sukurtas veiklos taisyklių struktūrizavimo įrankis, kurio pagalba galima kurti, redaguoti, tvarkyti veiklos taisykles grafinėje aplinkoje.

Įsitikinta, kad nagrinėjamas veiklos taisyklių struktūrizavimo įrankis gali būti taikomas praktikoje ir pilnai tenkina užsibrėžtus Roso metodu užrašytų VT saugojimo modelio reikalavimus.

Nustatyta, jog įrankis leidžia išsaugoti grafinius VT modelius saugyklos duomenų bazėje bei juos be praradimų atkurti tik pagal saugykloje užfiksuotą informaciją. Tai iš dalies patvirtina, jog panaudotas VT struktūrizavimo modelis gali būti taikomas praktikoje, o VT diagramos gali būti išsaugomos atributiniame pavidale.

Pirminė veiklos taisyklių struktūrizavimo įrankio pritaikymo sritis – VT specifikuojamas pagal modifikuotą Roso metodą IS projektavimo metu, remiantis jau anksčiau sudaryta IS reikalavimų specifikacija.

Atliktas veiklos taisyklių struktūrizavimo įrankio vertinimas ir įsitikinta, kad įrankis yra pranašesnis, nes veiklos taisyklės struktūrizavimas naudojantis grafinėmis priemonėmis atitinka formalius taisyklių užrašymo reikalavimus, keliamus Roso metode. Taip pat grafinis VT struktūrizavimas patogesnis kuriant naujas ar redaguojant esančias saugykloje veiklos taisykles.

Veiklos taisyklių struktūrizavimo įrankis pristatytas konferencijoje „Informacinės technologijos verslui – 2006“

## 7 LITERATŪRA

- [1] Business Rules Group. Defining Business Rules ~ What Are They Really? (3<sup>rd</sup> edition). Business Rules Group [interaktyvus]. 2000 [žiūrėta 2004-10-15]. Prieiga per internetą: <http://www.BusinessRulesGroup.org>.
- [2] Butleris, R.; Kapočius, K. *Struktūrizuotų veiklos taisyklių saugyklos architektūra*. Informacijos mokslai 17 Vilnius, Vilniaus universiteto leidykla, 2001: pp. 46-57.
- [3] Butleris, R.; Kapočius, K. *The Business Rules Repository for Information Systems Design*. The 6<sup>th</sup> East-European Conference ADBIS'2002. Konferencijos pranešimų medžiaga. Bratislava, Slovakia, Vydavatel'stvo STU, Vol.2, 2002: pp.64-77.
- [4] Elmasri, R.; Navathe, S. B. *Fundamentals of Database Systems*, Third Edition. Addison-Wesley, 2002.
- [5] Ross, R. G. *The Business Rule Book: Classifying, Defining an Modeling Rules*. Business Rule Solutions, Houston, 1997.
- [6] GUIDE Business Rules Project. Final Report (revision 1.2) [interaktyvus]. 2000 [žiūrėta 2005 11 11]. Prieiga per internetą: <http://www.softerra.com>.
- [7] Dr. Dorsey, P. The Business Rules Approach to Systems Development. Dulcian, Inc. [interaktyvus]. 2002 [žiūrėta 2006 05 04]. Prieiga per internetą: <http://www.dulcian.com>.
- [8] Ross, R. G. Principles of the Business Rule Approach [interaktyvus]. Addison Wesley Professional, 2003 [žiūrėta 2006 05 03]. Prieiga per internetą: <http://www.awprofessional.com>.
- [9] Gottesdiener, E. Business Rules show power, promise [interaktyvus]. 2001 [žiūrėta 2006-05-03]. Prieiga per internetą: <http://www.adtmag.com>.
- [10] Gottesdiener, E. Discovering an Organization's Knowledge: Facilitating Business Rules Workshops [interaktyvus]. 1999 [žiūrėta 2006-05-03]. Prieiga per internetą: <http://amauta.org>.
- [11] Hurwitz, J. When rules meet development [interaktyvus]. 2000 [žiūrėta 2006-05-03]. Prieiga per internetą: <http://www.dbmsmag.com>.
- [12] Ross, R. G.; Healy, K. A. Organizing Business Plans [interaktyvus]. 2000 [žiūrėta 2006-05-03]. Prieiga per internetą: <http://www.businessrulesgroup.org>.
- [13] Hale Von, B. *Business rules applied*. New York, 2002.
- [14] Morgan, T. *Business rules and information systems*. Indianapolis, 2003.
- [15] Healy, K. A.; Eulenberg, M. Response to MDC/Microsoft Business Rules Metamodel by the Business Rules Group (BRG) [interaktyvus]. 1999 [žiūrėta 2006-05-03]. Prieiga per internetą: <http://www.businessrulesgroup.org>.

- [16] Perkins, A. Business Rules Are Meta Data [interaktyvus]. 2002 [žiūrėta 2006-05-03]. Prieiga per internetą: <http://www.BRCommunity.com>.
- [17] Business Rule Solutions. Commercial business rules support products [interaktyvus]. 2006 [žiūrėta 2006-04-06]. Prieiga per internetą: <http://www.brsolutions.com>.
- [18] ILOG Inc. ILog JRules [interaktyvus]. 2006 [žiūrėta 2006-05-01]. Prieiga per internetą: <http://www.ilog.com>.

## **8 SUMMARY**

### **Research of business rules structuring by modified Ross method**

Designing information systems (IS) of business rules (BR) conception, rules identification, classification, enrolment fixed form and linking with other IS objects of project question is very important. Referred business rules ordering operation inclusive process is BR structuring. Of any known business rules theoretical models the Ronald Ross method is stand out of susceptible BR graphic notation. This method are concretized and extended, by designing structured method-based BR, data models and other important information repository model and tool witch fill this designed repository using graphic notation.

The models and technologies for business rules structuring are analyzed. BR designing using Ross method-based model are discussed.

The master work presents the modified Ross method-based model for business rules structuring during the information systems design as well as a software implementation of this model. The developed tool provides the designer with possibilities to create graphical business rules models, relating them with the specific elements of universe of discourse data model. The models are then processed and stored in a specialized repository. The presentation includes the overview of the model as well as the analysis of the specific features of the developed tool.

The experiments of implemented tool and program of BR repository filling are described.

## 9 TERMINŲ IR SANTRUMPŲ ŽODYNAS

*VT* – veiklos taisyklė.

*IS* – informacinė sistema.

*Case priemonės (CASE)* – priemonės informacinės sistemos kūrimui.

*VBA (Visual Basic for Applications)* – programavimo įrankis.

*Microsoft Visio* – įrankis, kurio grafinė aplinka naudojama CASE priemonių funkciniam reikalavimams specifiuoti.

*Microsoft Visio* – programa, kuri naudojama duomenų bazėms kurti.

## 10 PRIEDAI

### 10.1 IS projektavimo fazėje struktūrizuotų VT saugyklos loginio modelio lentelių aprašymai

#### 10.1.1 Atributas

Duomenys apie dalykinės srities duomenų modelio esybių atributus.

*Struktūra analogiška reikalavimų saugyklos modelio tos pačios lentelės struktūrai.*

Atributas	Aprašas	Galimos reikšmės	Būtinai	Tipas
<i>AtrID</i>	Unikalus atributo identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonum
<i>FizVardas</i>	Atributo fizinis vardas, naudojamas duomenų modelyje.	Bet koks žodis ar simbolių junginys be tarpų.	✓	chr(20)
<i>Vardas</i>	Atributo pavadinimas natūralia kalba.	Bet koks tekstas.	✓	chr(50)
<i>EsID</i>	Esybės, kurią charakterizuoja/aprašo atributas, ID.	Pasirinkta <i>EsID</i> reikšmė iš lentelės <i>Esybe</i> .		num(10)
<i>Aprasas</i>	Aprašas, atskleidžiantis atributo savybes, kurių negalima nuspėti pagal atributo charakteristikas.	Bet koks tekstas.		chr(200)
<i>Unikalumas</i>	Požymis, nurodantis, ar atributo reikšmė privalo būti unikali (negali kartotis lentelės viduje).	Taip (true, 1) arba Ne (false, 0).		boolean
<i>Tipas</i>	Atributo fizinis tipas.	- <i>num</i> - <i>char</i> - <i>boolean</i> - <i>date</i> - <i>time</i> - ...		chr(10)
<i>Ilgis</i>	Atributo reikšmės maksimalus leistinas ilgis (jeigu taikytina pagal tipą).	Sveikas teigiamas skaičius.		num(3)
<i>Butinumas</i>	Požymis, nurodantis, ar atributo reikšmė yra privaloma.	Taip (true, 1) arba Ne (false, 0).		boolean
<i>KeitimoData</i>	Atributo įvedimo į saugyklą arba paskutinio bent vienos atributo charakteristikos redagavimo data. Taip pat nurodo, ar atributas nėra pašalintas (praradęs aktualumą).	Data ir tikslus laikas. Šalinant atributą, nustatoma reikšmė "1111.11.11"; kitos charakteristikos nepakinta.	✓	datetime
<b>Vienas-su-daug ryšys su &lt;esybė&gt;.&lt;atributas&gt;</b>		<b>Reikšmė</b>	<b>Būtinai</b>	
<b>ReiksmeTerminas.AtrID</b>		Dalykinei sričiai svarbių šio atributo reikšmių aibė.		



<b>Koresp.dt_id</b>	Veiklos taisyklių korespondentų, esančių šiuo atributu, aibė.	
<b>TaisyklDT.dt_id</b>	Nuorodų į išsamias VT, kurioms yra svarbus šis atributas, aibė.	
<b>VTaisykle.baze_id</b>	Veiklos taisyklių, kurių baze yra šis atributas, aibė.	

### 10.1.2 Esybe

Duomenys apie dalykinės srities duomenų modelio esybes.

*Struktūra analogiška reikalavimų saugyklos modelio tos pačios lentelės struktūrai.*

<b>Atributas</b>	<b>Aprašas</b>	<b>Galimos reikšmės</b>	<b>Būtinai</b>	<b>Tipas</b>
<i>EsID</i>	Unikalus esybės identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonum
<i>FizVardas</i>	Esybės fizinis vardas, naudojamas duomenų modelyje.	Bet koks žodis ar simbolių junginys be tarpų.	✓	chr(25)
<i>Vardas</i>	Esybės pavadinimas natūralia kalba.	Bet koks tekstas.	✓	chr(50)
<i>Aprasas</i>	Aprašas, atskleidžiantis esybės savybes, kurių negalima nuspėti pagal vardą.	Bet koks tekstas.		chr(200)
<i>ISAEsID</i>	Esybės, su kuria ši esybė susijusi ISA (apibendrinimo) ryšiu, id.	Pasirinkta tokia <i>EsID</i> reikšmė iš lentelės <i>Esybe</i> , kad <i>EsID ≠ ISAEsID</i> .		num(10)
<i>ISATID</i>	ISA sąryšio, kuriuo ši esybė susijusi su kita esybe, tipo id.	Pasirinkta <i>ISATID</i> iš lentelės <i>ISATipas</i> .		num(10)
<i>ISAPozymis</i>	Požymis, nurodantis, ar esybė yra apibendrinanti (t.y., ar egzistuoja ISA sąryšiu su esybe susijusių esybių).	Taip (true, 1) arba Ne (false, 0).		boolean
<i>KeitimoData</i>	Esybės įvedimo į saugyklą arba paskutinio bent vienos esybės charakteristikos redagavimo data. Taip pat nurodo, ar esybė nėra pašalinta (pradus aktualumą).	Data ir tikslus laikas. Šalinant esybę, nustatoma reikšmė "1111.11.11"; kitos charakteristikos nepakinta.	✓	datetime

<b>Vienas-su-daug ryšys su &lt;esybė&gt;.&lt;atributas&gt;</b>	<b>Reikšmė</b>	<b>Būtinai</b>
<b>Koresp.dt_id</b>	Veiklos taisyklių korespondentų, esančių šia esybe (arba ISA sąryšiu, išeinančiu iš šios esybės), aibė.	
<b>TaisyklDT.dt_id</b>	Nuorodų į išsamias VT, kurioms yra svarbi ši esybė, aibė.	
<b>VTaisykle.baze_id</b>	Veiklos taisyklių, kurių baze yra ši esybė (arba ISA sąryšis, išeinantis iš šios esybės), aibė.	
<i>Rysys.EsID1</i>	Sąryšių tarp dviejų esybių, kuriuose viena iš besisiejiančių esybių yra ši esybė, aibė.	

<i>Rsys.EsID2</i>	Sąryšių tarp dviejų esybių, kuriuose viena iš besisiejiančių esybių yra ši esybė, aibė.	
<i>Esybe.ISAEsID</i>	Esybių, kurios yra šios esybės potipiai, poklasiai rūšys ir pan. (siejasi su šia esybe ISA sąryšiu), aibė.	
<i>Atributas.EsID</i>	Atributų, kurie aprašo pagrindines šios esybės savybes ir charakteristikas, aibė.	

### 10.1.3 ISATipas

Duomenys apie galimus ISA ryšio tarp esybių tipus.

*Struktūra analogiška reikalavimų saugyklos modelio tos pačios lentelės struktūrai.*

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<b>ISATID</b>	Unikalus ISA ryšio tipo identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonum
<i>Pavadinimas</i>	ISA ryšio tipo pavadinimas.	ISA ryšio pobūdį atspindintis tekstas. Tipai kuriami pagal realų poreikį, atsirandantį kuriant duomenų modelį. Galima (nebaigtinė) reikšmių aibė: {"tipas", "rūšis", "kategorija", "būseną", "statusas", ...}	✓	chr(30)
<i>Aprašas</i>	ISA ryšio tipo aprašas.	Bet koks tekstas.	✓	chr(200)
<b>Vienas-su-daug ryšys su &lt;esybė&gt;.&lt;atributas&gt;</b>		<b>Reikšmė</b>	<b>Būtin as</b>	
<i>Esybe.ISATID</i>	Dalykinės srities duomenų modelio esybių, kurios yra susietos ISA ryšiu su kita esybe(-ėm), aibė.			

### 10.1.4 Koresp

Konkrečios VT korespondentas. Korespondentu gali būti duomenų tipas, konstanta, kita VT arba kitos VT išieigos reikšmė.

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<i>t_id</i>	Veiklos taisyklės, kurios korespondentu yra šis įrašas, identifikacinis numeris.	Pasirinkta <i>t_id</i> reikšmė iš lentelės <i>VTaisykle</i> .	✓	autonum
<i>kor_nr</i>	Korespondento eilės numeris. Kartu su atributu <i>t_id</i> sudaro unikalią kombinaciją. Naudojamas, jeigu specialaus elemento tipas yra <i>apribojimas</i> ir veiklos taisyklė, su kuria siejamas šis specialusis elementas, turi du ar daugiau korespondentų.	Dviženklis sveikas skaičius.	✓	num(2)
<i>kor_tipas</i>	Korespondento pobūdis.	<b>E</b> – korespondentas yra esybė. <b>A</b> – korespondentas yra atributas. <b>R</b> – korespondentas yra ryšys. <b>S</b> – korespondentas yra ISA ryšys. <b>C</b> – korespondentas yra konstanta.	✓	char(1)

		<b>T</b> – korespondentas yra kita VT negu ta, kuriai priskiriamas šis korespondentas. <b>I</b> – korespondentas yra kitos VT negu ta, kuriai priskiriamas šis korespondentas, išieigos reikšmė.		
<i>kor_id</i>	Korespondento id. Tai gali būti konstantos, duomenų tipo arba kitos taisyklės id (priklausomai nuo <i>kor_tipas</i> ).	Vienas iš šių variantų: Pasirinkta <i>EsID</i> reikšmė iš lentelės <i>Esybe</i> (kai <i>kor_tipas</i> = E, S), Pasirinkta <i>AtrID</i> reikšmė iš lentelės <i>Atributas</i> (kai <i>kor_tipas</i> = A), Pasirinkta <i>RysID</i> reikšmė iš lentelės <i>Rysys</i> (kai <i>kor_tipas</i> = R), Pasirinkta <i>RTID</i> reikšmė iš lentelės <i>ReiksmeTerminas</i> (kai <i>kor_tipas</i> = C), Pasirinkta <i>t_id</i> reikšmė iš lentelės <i>VTaisyklė</i> (kai <i>kor_tipas</i> = T, I).	✓	num(10)

### 10.1.5 ReiksmeTerminas

Duomenys apie dalykinėje srityje svarbias esybių charakteristikų (atributų) reikšmes, taip pat ir apie savarankiškas reikšmes. Viso veiklos taisyklių diagramose naudojamos konstantos ir išieigos reikšmės bus saugojamos šioje lentelėje. Lentelėje paprastai nebus saugojamos reikšmės, kurios priskiriamos atributui esant tam tikroms sąlygoms. Savarankiškos reikšmės pavyzdys: taisyklėse gali būti naudojama reikšmė “fizinų pajamų mokestis”, kuri nėra nei atributas, nei esybė – tokios reikšmės vertė būtų saugoma šioje lentelėje, o ją naudojančiose taisyklėse būtų nurodomas tik reikšmės pavadinimas. Atributo reikšmės pavyzdys: automobiliams gali būti žinoma “populiariausia modifikacija” (esybės “automobilis” atributo “modifikacija” reikšmė), kurios reali reikšmė gali keistis ir būtų saugoma šioje lentelėje, o taisyklėse būtų nurodomas tik reikšmės pavadinimas.

*Struktūra analogiška reikalavimų saugyklos modelio tos pačios lentelės struktūrai.*

Atributas	Aprašas	Galimos reikšmės	Būti nas	Tipas
<i>RTID</i>	Unikalus reikšmės identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonum
<i>AtrID</i>	Atributo, kurio reikšmė saugoma, id.	Pasirinkta <i>AtrID</i> reikšmė iš lentelės <i>Atributas</i> .		num(10)
<i>Pavadinimas</i>	Reikšmės pavadinimas, kuris bus naudojamas formuluojant veiklos taisyklės ir kitus reikalavimus.	Bet koks tekstas.	✓	chr(50)
<i>Reiksme</i>	Reikšmės išraiška iš vieno ar daugiau žodžių, skaičių ar pan.	Bet koks tekstas.		chr(50)
<i>Aprasas</i>	Aprašas arba komentaras, atskleidžiantis termino arba reikšmės prasmę.	Bet koks tekstas.		chr(100)
<b>Vienas-su-daug ryšys su &lt;esybė&gt;.&lt;atributas&gt;</b>		<b>Reikšmė</b>	<b>Būti nas</b>	
<i>Koresp.dt id</i>		Veiklos taisyklių korespondentų, esančių šiuo atributu, aibė.		
<i>VTaisykle.baze id</i>		Veiklos taisyklių, kurių baze yra šis atributas, aibė.		
<i>VTaisykle.is id</i>		Veiklos taisyklių, kurių išieigos reikšme yra ši reikšmė, aibė.		

## 10.1.6 Rysys

Duomenys apie dalykinės srities duomenų modelyje užfiksuotus sąryšius tarp dviejų esybių.  
*Struktūra analogiška reikalavimų saugyklos modelio tos pačios lentelės struktūrai.*

Atributas	Aprašas	Galimos reikšmės	Būtinai	Tipas
<i>RysID</i>	Unikalus ryšio identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonum
<i>EsID1</i>	Esybės, kuri vienas-su-daug sąryšyje yra sąlyginėje kairėje pusėje (arba tiesiog yra pirmoji esybė sąryšyje dėl to, kad nurodoma pirmoji), id.	Pasirinkta <i>EsID</i> reikšmė iš lentelės <i>Esybe</i> .	✓	num(10)
<i>EsID2</i>	Esybės, kuri vienas-su-daug sąryšyje yra sąlyginėje dešinėje pusėje (arba tiesiog yra pirmoji esybė sąryšyje dėl to, kad nurodoma pirmoji), id.	Pasirinkta <i>EsID</i> reikšmė iš lentelės <i>Esybe</i> . <i>EsID2</i> gali sutapti su <i>EsID1</i> .	✓	num(10)
<i>Pavadinimas1</i>	Sąryšį iš esybės1 į esybės2 charakterizuojanti veiksmožodinė frazė.	Bet koks tekstas.	✓	chr(50)
<i>Pavadinimas2</i>	Sąryšį iš esybės2 į esybės1 charakterizuojanti veiksmožodinė frazė.	Bet koks tekstas.	✓	chr(50)
<i>Kard1Nuo</i>	Sąryšio kardinalumo esybės1 pusėje apatinė riba.	- sveikas skaičius 0 ... 999.	✓	chr(3)
<i>Kard1Iki</i>	Sąryšio kardinalumo esybės1 pusėje viršutinė riba.	- sveikas skaičius 0 ... 999, - simbolis "*" (reikšmė – "daug").	✓	chr(3)
<i>Kard2Nuo</i>	Sąryšio kardinalumo esybės2 pusėje apatinė riba.	- sveikas skaičius 0 ... 999.	✓	chr(3)
<i>Kard2Iki</i>	Sąryšio kardinalumo esybės2 pusėje viršutinė riba.	- sveikas skaičius 0 ... 999, - simbolis "*" (reikšmė – "daug").	✓	chr(3)
<i>AgrPozymis</i>	Požymis, nurodantis, ar ryšys yra agregavimo (t.y. gali būti ivardinamas išraiška "yra dalis", "is a part of").	Loginė reikšmė: - <i>taip (true)</i> – ryšys yra agregavimo; <i>esybė1</i> yra <i>esybės2</i> dalis; šiuo atveju atributai <i>Pavadinimas1</i> ir <i>Pavadinimas2</i> gali būti ir nepildomi. - <i>ne (false)</i> – ryšys nėra agregavimo.	✓	boolean
<i>KeitimoData</i>	Ryšio įvedimo į saugyklą arba paskutinio bent vienos ryšio charakteristikos redagavimo data. Taip pat nurodo, ar ryšys nėra pašalintas (praradęs aktualumą).	Data ir tikslus laikas. Šalinant ryšį, nustatoma reikšmė "1111.11.11"; kitos charakteristikos nepakinta.	✓	datetime
<b>Vienas-su-daug ryšys su &lt;esybė&gt;.&lt;atributas&gt;</b>		<b>Reikšmė</b>	<b>Būtinai</b>	
<i>Koresp.dt id</i>		Veiklos taisyklių korespondentų, esančių šiuo ryšiu, aibė.		
<i>TaisyklDT.dt id</i>		Nuorodų į išsamias VT, kurioms yra svarbus šis ryšys, aibė.		

<i>VTaisykle.baze_id</i>	Veiklos taisyklių, kurių baze yra šis ryšys, aibė.	
--------------------------	--	--

### 10.1.7 Šaltinis

Konkrečios veiklos taisyklės kilmės šaltinis. Šaltiniu gali būti asmuo, dokumentas (įstatymas, specifikacija, reglamentas, dokumentacija ar kt.) ar kitoks objektas.

Atributas	Aprašas	Galimos reikšmės	Būtinai	Tipas
<i>s_id</i>	Unikalus šaltinio identifikacinis numeris.	Automatiškai priskiriamas skaičius.	✓	autonum
<i>a_d_kodas</i>	Šaltinio kodas.	Asmens arba dokumento ar kitokio objekto, esančio šaltiniu, kodas		char(10)
<i>s_vardas</i>	Šaltinio pavadinimas.	Pavardė (jeigu asmuo) arba pavadinimas daiktavardine forma (jeigu kita)	✓	char(20)
<i>s_aprasas</i>	Šaltinio aprašas. Naudojamas tuo atveju, jeigu kodo ir pavadinimo nepakanka šaltiniui apibūdinti.	Bet koks tekstas.		char(80)
<b>Vienas-su-daug ryšys su &lt;esybė&gt;.&lt;atributas&gt;</b>		<b>Reikšmė</b>		
<i>VTaisykle.s_id</i>		Taisyklių, kilusių iš šio šaltinio, aibė. Ši aibė gali būti tuščia.		

### 10.1.8 SERusis

Roso metode numatytų specialiųjų elementų grupių charakteristikos [13]. Pastarajame metode naudojami specialieji elementai suskirstyti į penkias grupes (plius tam tikros variacijos).

Atributas	Aprašas	Galimos reikšmės	Būtinai	Tipas
<i>ser_kodas</i>	Unikali specialaus elemento grupės pavadinimo santrumpa (kodas).	Roso metode numatytų spec elementų grupių pavadinimų santrumpos. Jų gali būti daugiau, tačiau mažiausiai privalo būti įtrauktos šios reikšmės: <b>IB</b> – <i>iškvietimo būsenų</i> grupė, <b>PG</b> – <i>pagrindų</i> grupė, <b>AK</b> – <i>aktivatorių</i> grupė, <b>SI</b> – <i>specinterpretatorių</i> grupė, <b>SA</b> – <i>pecapribojimų</i> grupė.	✓	char(2)
<i>pavadinimas</i>	Specialaus elemento pavadinimas.	Galimos reikšmės numatytos Roso metode.	✓	char(20)
<i>ser_tipas</i>	Kodas, nurodantis, kur VT diagramoje gali būti naudojami šios specialiųjų elementų grupės elementai.	Galimos reikšmės: <b>I</b> – grupės elementai yra interpretatoriai, naudojami tik prieš taisyklės simbolį (kai <i>ser_kodas</i> = SI), <b>A</b> – grupės elementai yra apribojimai, naudojami tik po taisyklės simbolio (kai <i>ser_kodas</i> = SA), “-” – grupės elementai gali būti naudojami ir kaip apribojimai, ir kaip interpretatoriai (tai reikšmė pagal nutylėjimą arba, kai <i>ser_kodas</i> = IB, PG, AK).	✓	char(1)
	Specialaus elemento tipas, t.y. ar jis yra			

	interpretatorius (naudojamas prieš taisyklės simbolį), ar apribojimas (naudojamas po taisyklės simbolio).			
<b>Vienas-su-daug ryšys su &lt;esybė&gt;.&lt;atributas&gt;</b>		<b>Reikšmė</b>		
<i>SpecElem.se_rusis</i>		Specialiujų elementų, kurie yra šios rūšies, aibė.		

### 10.1.9 SpecElem

Roso metode numatyti specialieji elementai [13]. Iš viso pastarajame metode yra naudojami 25 specialieji elementai (plius tam tikros variacijos).

Atributas	Aprašas	Galimos reikšmės	Būtinai	Tipas
<i>se_kodas</i>	Unikali specialaus elemento pavadinimo santrumpa (kodas).	Galimų reikšmių sąrašas pateikiamas 7 priede.	✓	char(4)
<i>se_rusis</i>	Specialaus elemento rūšies kodas.	Pasirinkta <i>ser_kodas</i> reikšmė iš lentelės <i>SERusis</i> .	✓	char(2)
<i>se_vardas</i>	Specialaus elemento pavadinimas.	Galimos reikšmės numatytos Roso metode.	✓	char(40)
<i>se_koment</i>	Specialųjį elementą apibūdinantis komentaras. Naudojamas, jeigu kitų atributų informacijos nepakanka.	Bet koks tekstas.		char(50)
<b>Vienas-su-daug ryšys su &lt;esybė&gt;.&lt;atributas&gt;</b>		<b>Reikšmė</b>		
<i>VTSpecElem.se_kodas</i>		Nuorodų į VT, kuriose yra naudojamas šis specialusis elementas, aibė. Ši aibė gali būti tuščia.		

### 10.1.10 TaisykiDT

Sąryšio daug-su-daug tarp lentelių Esybe/Atributas/Rsys ir VTPilna pagalbinė lentelė. Logiškai nedalomai veiklos taisyklei gali būti svarbūs įvairūs duomenų tipai, taip pat ir duomenų tipai gali būti svarbūs įvairioms taisyklėms.

Atributas	Aprašas	Galimos reikšmės	Būtinai	Tipas
<i>vdt_id</i>	Taisyklės duomenų tipo unikalus identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonum
<i>dt_id</i>	Duomenų tipo identifikacinis numeris.	Vienas iš šių variantų: Pasirinkta <i>EsID</i> reikšmė iš lentelės <i>Esybe</i> (kai <i>dt_tipas</i> = E, S), Pasirinkta <i>AtrID</i> reikšmė iš lentelės <i>Atributas</i> (kai <i>dt_tipas</i> = A), Pasirinkta <i>RysID</i> reikšmė iš lentelės <i>Rsys</i> (kai <i>dt_tipas</i> = R).	✓	num(10)
<i>t_idl</i>	Identifikacinis numeris.	Pasirinkta <i>t_idl</i> reikšmė iš lentelės <i>VTPilna</i> .	✓	num(4)
<i>dt_tipas</i>	Veiklos taisyklei svarbaus duomenų tipo pobūdis.	Galimos reikšmės: E – esybė,	✓	char(1)

		<b>A</b> – atributas, <b>R</b> – ryšys, <b>S</b> – ISA ryšys.		
--	--	---	--	--

### 10.1.11 VTaisykle

Informacija apie atskiros veiklos taisyklės diagramą. Ši informacija neapima taisyklės korespondentų (saugomi lentelėje *Koresp*) ir svarbių duomenų tipų (saugomi lentelėje *TaisyklDT*).

Atributas	Aprašas	Galimos reikšmės	Būti nas	Tipas
<i>t_id</i>	Unikalus veiklos taisyklės identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonom
<i>t_id1</i>	Pilnos logiškos veiklos taisyklės, kurios dalimi yra ši veiklos taisyklė, identifikacinis numeris.	Įvedant taisyklę į DB jau žinoma pirmoji unikalios taisyklės numerio dalis (pilnas numeris arba skaičius iki taško). Pvz.: jeigu taisyklės numeris 42, tai <i>t_id1</i> bus 42, jeigu 43.2 – tai 43.	✓	num(4)
<i>t_id2</i>	Antroji unikalios taisyklės numerio dalis (skaičius po taško).	Įvedant taisyklę į DB jau žinomas. Pvz.: jeigu taisyklės numeris 42, tai <i>t_id2</i> bus 0, jeigu 43.2 – tai 2. Vienodos šio atributo ir atributo <i>t_id1</i> reikšmių kombinacijos neleistinos.	✓	num(2)
<i>tg_id</i>	Grupės, kuriai priskiriama taisyklė, identifikacinis numeris.	Pasirinkta <i>tg_id</i> reikšmė iš lentelės <i>VTGrupe</i> .		num(10)
<i>s_id</i>	Šaltinio, inicijavusio taisyklę, id.	Pasirinkta <i>s_id</i> reikšmė iš lentelės <i>Saltinis</i> .		num(10)
<i>t_rusis</i>	Taisyklės pobūdis, rūšis.	<b>A</b> – Taisyklė yra apribojimas. <b>S</b> – Taisyklė yra sąlyga.	✓	char (1)
<i>tipo_kodas</i>	Tipo, kuriam priskiriama ši veiklos taisyklė, kodas.	Pasirinkta <i>tipo_kodas</i> reikšmė iš lentelės <i>VTTip</i> . Reikšmė nustatoma tiesiogiai pagal taisyklės diagramą.	✓	char(8)
<i>proc</i>	Procedūros, naudojamos veiklos taisyklėje, pavadinimas. Turi prasmę tik <i>Funkcinių patikrintojų</i> tipų grupei priklausančiose nedalomose veiklos taisyklėse ir kai kuriose išvestinėse taisyklėse.	Bet koks naudojamoje programinėje terpėje reikiamą procedūrą identifikuojantis tekstas.		char(30)
<i>funkcija</i>	Funkcijos, naudojamos veiklos taisyklėje, pavadinimas. Turi prasmę tik <i>Funkcinių patikrintojų</i> tipų grupei priklausančiose nedalomose veiklos taisyklėse ir kai kuriose išvestinėse taisyklėse.	Bet koks naudojamoje programinėje terpėje reikiamą funkciją identifikuojantis tekstas (pvz.: cosine, sine, log, ir t.t.)		char(20)
<i>ia_poz</i>	Požymis, nurodantis, ar veiklos taisyklėje naudojami interpretatoriai arba apribojimai (specialieji elementai).	Loginė reikšmė: <i>Loginis vienetas</i> (loginė tiesa) – Specialieji elementai naudojami. <i>Loginis nulis</i> (loginė netiesa) – Specialieji elementai nenaudojami.	✓	Logical
<i>ir_poz</i>	Požymis, nurodantis, ar veiklos taisyklė turi išorinę išieigos reikšmę.	Loginė reikšmė: <i>Loginis vienetas</i> (loginė tiesa) – taisyklė turi išorinę išieigos reikšmę.	✓	Logical

		<i>Loginis nulis</i> (loginė netiesa) – taisyklė išorinės išeigos reikšmės neturi.		
<i>b_tipas</i>	Veiklos taisyklės bazės pobūdis.	Galimos reikšmės: E – bazė yra esybė, A – bazė yra atributas, R – bazė yra ryšys, S – bazė yra ISA ryšys, C – bazė yra konstanta, T – bazė yra kita VT, I – bazė yra kitos VT išeigos reikšmė.	✓	char(1)
<i>bazės_id</i>	Bazės identifikacinis numeris. Tai gali būti konstantos, duomenų tipo arba kitos taisyklės identifikacinis numeris. (priklausomai nuo bazės pobūdžio).	Vienas iš šių variantų: Pasirinkta <i>EsID</i> reikšmė iš lentelės <i>Esybe</i> (kai <i>b_tipas</i> = E, S), Pasirinkta <i>AtrID</i> reikšmė iš lentelės <i>Atributas</i> (kai <i>b_tipas</i> = A), Pasirinkta <i>RysID</i> reikšmė iš lentelės <i>Rysys</i> (kai <i>b_tipas</i> = R), Pasirinkta <i>RTID</i> reikšmė iš lentelės <i>ReiksmeTerminas</i> (kai <i>b_tipas</i> = C), Pasirinkta <i>t_id</i> reikšmė iš lentelės <i>VTaisyklė</i> (kai <i>b_tipas</i> = T, I). Negali sutapti su to paties įrašo <i>t_id</i> , <i>in_vt</i> ir <i>out_vt</i> reikšmėmis).	✓	num(10)
<i>out_vt</i>	Išeigos taisyklės unikalus identifikacinis numeris (jeigu yra išeigos taisyklė).	Pasirinkta <i>t_id</i> reikšmė iš lentelės <i>VTaisykle</i> . Negali sutapti su to paties įrašo <i>t_id</i> , <i>in_vt</i> ir <i>bazės_id</i> (jeigu bazės pobūdis yra kita taisyklė arba kitos taisyklės išeigos reikšmė).		num(10)
<i>in_vt</i>	Įeigos taisyklės unikalus identifikacinis numeris (jeigu yra įeigos taisyklė).	Pasirinkta <i>t_id</i> reikšmė iš lentelės <i>VTaisykle</i> . Negali sutapti su to paties įrašo <i>t_id</i> , <i>out_vt</i> ir <i>bazės_id</i> (jeigu bazės pobūdis yra kita taisyklė arba kitos taisyklės išeigos reikšmė).		num(10)
<i>is_id</i>	Išeigos reikšmės identifikacinis numeris (jeigu yra išeigos reikšmė).	Pasirinkta <i>RTID</i> reikšmė iš lentelės <i>ReiksmeTerminas</i> .		num(10)
<i>t_koment</i>	Informaciją apie veiklos taisyklę papildantis komentaras.	Bet koks tekstas.		char(60)
<i>mod_data</i>	Veiklos taisyklės paskutinio modifikavimo (arba įvedimo, jeigu dar nemodifikuota) data.	Lentelėje atlikus bent vieno lauko pakeitimą automatiškai nustatoma esama data.	✓	date
<i>mod_pav</i>	Įvedusio veiklos taisyklę arba ją pakeitusio asmens pavardė.	Bet koks žodis ar žodžių junginys.	✓	char(20)
<b>Vienas-su-daug ryšys su &lt;esybė&gt;.&lt;atributas&gt;</b>		<b>Reikšmė</b>		
<i>Koresp.t_id</i>		Veiklos taisyklės korespondentų aibė. Joje privalo būti bent vienas narys.		
<i>Koresp.kor_id</i>		Kitų veiklos taisyklių korespondentų, kuriais yra ši veiklos taisyklė, aibė. Ši aibė gali būti tuščia.		
<i>VTSpecElem.t_id</i>		Specialiųjų elementų, naudojamų veiklos taisyklės diagramoje, aibė. Ši aibė gali būti tuščia.		



<i>VTaisykle.out_vt</i>	Kitų veiklos taisyklių, kurių išeiga yra ši taisyklė, aibė. Ši aibė gali būti tuščia.
<i>VTaisykle.in_vt</i>	Kitų veiklos taisyklių, kurių įeiga yra ši taisyklė, aibė. Ši aibė gali būti tuščia.
<i>VTaisykle.baze_id</i>	Kitų veiklos taisyklių, kurių baze yra ši taisyklė, aibė. Ši aibė gali būti tuščia.

### 10.1.12 VTGrupe

Duomenys apie abstrakčią veiklos taisyklių grupę.

Atributas	Aprašas	Galimos reikšmės	Būti nas	Tipas
<i>tg_id</i>	Unikalus taisyklių grupės identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonom
<i>tg vardas</i>	Taisyklių grupės pavadinimas.	Bet koks žodis ar žodžių junginys, pageidautina daiktavardinės formos.	✓	char(20)
<i>tg aprašas</i>	Taisyklių grupės aprašas.	Bet koks tekstas.	✓	chr(100)
<b>Vienas-su-daug ryšys su &lt;esybė&gt;.&lt;atributas&gt;</b>		<b>Reikšmė</b>		
<i>VTaisykle.t_id</i>	Veiklos taisyklių, priklausančių šiai grupei, aibė. Ji gali būti tuščia.			

### 10.1.13 VTPilna

Bendra informacija apie logiškai nedalomą taisyklę. Vienos pilnos veiklos taisyklės diagramoje gali būti kelios veiklos taisyklės.

Atributas	Aprašas	Galimos reikšmės	Būti nas	Tipas
<i>t_id1</i>	Unikalus veiklos taisyklės numeris.	Sveikas skaičius. Įvedant taisyklę į DB jau žinomas. Pirmoji unikalios taisyklės numerio dalis (pilnas numeris arba skaičius iki taško).	✓	num(4)
<i>t_teiginys</i>	Tiksliai suformuluotas veiklos taisyklės teiginys.	Teiginio tekstas.	✓	chr(200)
<b>Vienas-su-daug ryšys su &lt;esybė&gt;.&lt;atributas&gt;</b>		<b>Reikšmė</b>		
<i>TaisyklDT.t_id1</i>	Nuorodų į duomenų tipus, kurie yra svarbūs šiai VT, aibė. Ši aibė gali būti tuščia.			
<i>VTaisykle.t_id1</i>	Veiklos taisyklių, įeinančių į šios pilnos taisyklės diagramą, aibė. Šioje aibėje privalo būti bent vienas narys.			

### 10.1.14 VTSpecElem

Ryšio daug-su-daug tarp lentelių *SpecElem* ir *VTaisykle* pagalbinė lentelė. Informacija apie tam tikrų veiklos taisyklių diagramose naudojamus specialiuosius elementus.

Atributas	Aprašas	Galimos reikšmės	Būti nas	Tipas
<i>se_ikodas</i>	Specialaus elemento kodas (pavadinimo santrumpa).	Pasirinkta <i>se_kodas</i> reikšmė iš lentelės <i>SpecElem</i> .	✓	char(4)
<i>t_id</i>	Veiklos taisyklės identifikacinis numeris.	Pasirinkta <i>t_id</i> reikšmė iš lentelės <i>VTaisykle</i> .	✓	num(10)
<i>kor_nr</i>	Korespondento, kuriam	Sveikas skaičius, neviršijantis taisyklės,	✓	num(2)

	galioja specialusis elementas, numeris. Turi prasmę tik tuomet, kai specialusis elementas yra apribojimas (A).	kurios diagramoje naudojamas specialusis elementas apribojimas, korespondentų skaičiaus. Jeigu galioja visiems korespondentams, reikšmė lygi 0 (reikšmė pagal nutylėjimą 0).		
<i>tse_tipas</i>	Specialaus elemento tipas, t.y. ar jis yra interpretatorius (naudojamas prieš taisyklės simbolį), ar apribojimas (naudojamas po taisyklės simbolio).	<b>I</b> – Specialusis elementas yra interpretatorius. <b>A</b> – specialusis elementas yra apribojimas	✓	char(1)
<i>tse_reiksm</i>	Specialaus elemento reikšmė (jeigu ji tikslinga)	Sveikas skaičius.		num(3)
<i>tse_vnt</i>	Specialaus elemento reikšmės matavimo vienetas.	Bet koks tekstas, pvz.: minutes, valandos ar pan.		char(10)
<i>koment</i>	Informaciją papildantis komentaras.	Bet koks tekstas.		char(30)

### 10.1.15 VTTipas

Saugomi visi Roso metode numatyti veiklos taisyklių tipai [13]. Iš viso 91 įrašas.  
Visi duomenys iš anksto žinomi.

Atributas	Aprašas	Galimos reikšmės	Būtinai	Tipas
<i>tipo_kodas</i>	Unikali veiklos taisyklės tipo pavadinimo santrumpa.	Roso metode numatytos tipų santrumpos (lietuviškos). Pilnas tipų sąrašas pateiktas 3 ir 5 lentelėse.	✓	char(8)
<i>t_tipas</i>	Visas taisyklės tipo pavadinimas.	Roso metode nustatyti taisyklės tipų pavadinimai (lietuviški). Visas tipų sąrašas pateiktas 3 ir 5 lentelėse.	✓	char(45)
<i>t_seima</i>	Visas taisyklės tipų grupės pavadinimas.	Roso metode nustatyti taisyklės tipų grupių pavadinimai (lietuviški). Visos taisyklių tipų grupės pateiktos 3 ir 5 lentelių titulinėse eilutėse.	✓	char(35)
<i>tiv_tipas</i>	Galimos išeiigos vertės tipo santrumpa.	Roso metode nustatyti išeiigos vertės tipai. Visi išeiigos vertės tipai nurodyti 4 lentelėje.		char(5)
<b>Vienas-su-daug ryšys su &lt;esybė&gt;.&lt;atributas&gt;</b>		<b>Reikšmė</b>		
<i>VTaisykle.tipo_kodas</i>		Veiklos taisyklių, priklausančių šiam tipui, aibė. Ši aibė gali būti tuščia.		

## 10.2 Straipsnio kopija

# VEIKLOS TAISYKLIŲ STRUKTŪRIZAVIMO INFORMACIJOS SISTEMŲ PROJEKTAVIMO METU ĮRANKIO REALIZACIJA

Saulius Baškevičius<sup>1</sup>, Kęstutis Kapočius<sup>2</sup>

<sup>1</sup>*KTU Informacijos Sistemų katedra,  
Studentų g. 50, Kaunas, Lietuva, sbaskevicius@gmail.com*

<sup>2</sup>*KTU Informacijos Sistemų katedra,  
Studentų g. 50, Kaunas, Lietuva, kestutis.kapocius@ktu.lt*

Straipsnyje apžvelgiamas modifikuotu Roso metodu grindžiamas veiklos taisyklių struktūrizavimo informacijos sistemų projektavimo metu modelis ir pristatoma programinė jo realizacija. Sukurtasis įrankis leidžia projektuotojui kurti detalius grafinius veiklos taisyklių modelius, susiejant juos su konkrečiais dalykinės srities duomenų modelio elementais. Pristatyme apžvelgiamas panaudotas modelis bei akcentuojami sukurtojo įrankio ypatumai, lemiantys sėkmingą šio modelio taikymą praktikoje.

## 1. ĮVADAS

Kuriant informacijos sistemas (IS) pagal veiklos taisyklių (VT) koncepciją, taisyklių identifikavimo, klasifikavimo, užrašymo fiksuota forma ir susiejimo su kitais IS projekto objektais klausimas yra ypač aktualus. Minėtus VT tvarkymo veiksmus apimantis procesas vadinamas VT struktūrizavimu. Iš eilės žinomų teorinių VT struktūrizavimo modelių išsiskiria Roso (*Ronald Ross*) metodas, leidžiantis VT išreikšti grafinėmis diagramomis [5]. Šis metodas buvo konkretizuotas ir praplėstas, sukuriant pagal jį struktūrizuotų VT, duomenų modelio elementų ir kitos svarbios informacijos saugyklos loginį modelį bei VT struktūrizavimo atributinėje sąsajoje įrankį [2, 3]. Šiame straipsnyje pristatomas VT saugyklą palaikantis grafinio taisyklių modeliavimo įrankis, leidžiantis visapusiškai įvertinti šio VT struktūrizavimo modelio gyvybingumą ir praktinę reikšmę.

Taigi, straipsnio 2 skyriuje apžvelgiami Roso metodo pagrindai bei pagal šį metodą struktūrizuotų VT ir duomenų modelio elementų saugyklos loginė architektūra. Straipsnio 3 skyriuje pristatomas VT struktūrizavimo grafinėje aplinkoje įrankis, apibrėžiamos esminės jo savybės. Straipsnis baigiamas išvadamis ir ateities darbų įvertinimu.

## 2. VEIKLOS TAISYKLIŲ STRUKTŪRIZAVIMO MODELIS, GRINDŽIAMAS MODIFIKUOTU ROŠO METODU

### 2.1 Roso metodo apžvalga

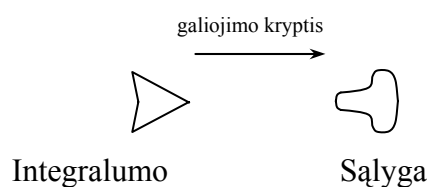
Metodo autorius iš esmės apibrėžia nestruktūrinių VT klasifikavimo ir formalaus užrašymo arba modeliavimo, naudojantis savita grafine notacija, metodiką. Roso sukurta VT

klasifikacija gali būti traktuojama kaip labiau konceptualaus GUIDE modelio [1] papildinys, detalizuojantis veiksmo teiginių VT kategoriją.

Formaliai užrašant VT, reikia remtis dalykinės srities duomenų modeliu. Pagrindiniai šio modelio elementai yra esybės, atributai, asociacijos, sąryšiai, potipiai [5]. VT modeliavimas susideda iš šešių pagrindinių funkcijų:

1. Nustatyti ar taisyklė yra integralumo apribojimas, ar sąlyga. T. y. taisyklė turi būti vykdoma ar išreikšta kaip patikrinimas?

Integralumo apribojimas – tai taisyklė, kurios rezultatas visada privalo būti loginė tiesa. Sąlyga – tai taisyklė, kurios rezultatas gali būti tiek loginė tiesa, tiek loginis melas. Taip pat rezultatas gali būti nežinomas. Kuriant VT grafinį modelį, integralumo apribojimams ir sąlygoms žymėti naudojami specialūs žymėjimai (1 pav.).



*1 pav. Taisyklės vykdymo ir taisyklės tikrinimo žymėjimas (pagal [5])*

2. Nustatyti taisyklės bazę. Kuriam tipui duomenų modelyje reikėtų priskirti nagrinėjamą taisyklę?

Dažniausia VT baze būna duomenų modelio objektas – esybė, atributas, ryšys. Tačiau baze taip pat gali būti kita VT ar jos išeigos reikšmė. Kiekviena VT privalo turėti bazę [5]. Išeinantis iš bazės ryšys veiklos taisyklės grafiniame modelyje vaizduojamas punktyrine linija.

3. Nustatyti taisyklės korespondentą. Kokie tipai duomenų modelyje būtini, norint patikrinti taisyklę?

Korespondentu gali būti duomenų modelio objektas, taip pat kita veiklos taisyklė ar jos išeigos reikšmė. Kiekviena VT privalo turėti bent vieną korespondentą. Įeinantis į korespondentą ryšys grafiniame veiklos taisyklės modelyje vaizduojamas punktyrine linija.

4. Nustatyti taisyklės tipą, t. y., kokius testus pritaiko taisyklė.

Taisyklė gali priklausyti nedalomų arba išvestinių VT grupei, kurių kiekviena skaidoma į eilę tipų [5]. Visi taisyklių tipai tinka tiek integralumo apribojimams, tiek ir sąlygoms. Taisyklės tipo santrumpa įrašoma apribojimo arba sąlygos simbolio viduje.

5. Nustatyti taisyklės bazės ir korespondento ryšį, t. y., kaip dalykinės srities duomenų modelio elementai, reikalingi taisyklei patvirtinti, susiję tarpusavyje.

Svarbu išskirti teisingą sąryšį tarp bazės ir korespondento. Šis sąryšis privalo sekti iš duomenų modelio.

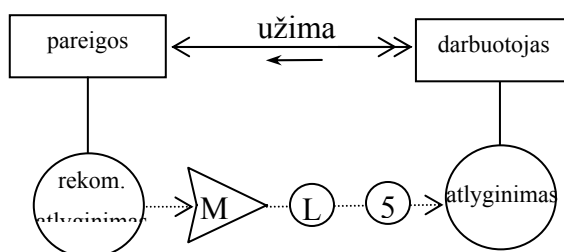
## 6. Nustatyti papildomas veiklos taisyklės modelio ypatybes.

Roso notacijoje yra numatyta galimybė naudoti papildomus simbolius, galinčius pakeisti standartinę tam tikro tipo taisyklės interpretaciją. Šie simboliai skirstomi į dvi pagrindines grupes [5]:

- a) Interpretatoriai – tai simboliai, nurodantys, jog taisyklę jos bazės atžvilgiu reikia interpretuoti ne taip, kaip reikėtų remiantis tik jos tipu. Šie simboliai yra vaizduojami ant bazės ryšio rodyklės.
- b) Apribojimai – tai simboliai, nurodantys, jog taisyklei jos korespondento(-ų) egzempliorių atžvilgiu turi būti taikomi specialūs apribojimai. Šie simboliai vaizduojami ant korespondento ryšio rodyklės.

Be specialiųjų simbolių VT diagramose dar gali būti vartojamos konstantos, taisyklių išeišios reikšmės ir kitos papildomos priemonės.

Paprastos taisyklės pavyzdys pateiktas 2 pav. Ji apibrėžia tokį apribojimą: *“Rekomenduojamas pareigų atlyginimas turi būti mažesnis už bent penkių darbuotojų, užimančių šias pareigas, atlyginimą”* [2]. Šios taisyklės bazė yra dalykinės srities duomenų modelio esybės pareigos atributas rekom.atlyginimas (rekomenduojamas atlyginimas). Taisyklės korespondentas – duomenų modelio esybės darbuotojas atributas atlyginimas. Mus domina tik tos pareigos, kurias užima darbuotojai, todėl į diagramą įtrauktas ir šis duomenų modelio sąryšis (faktas). Taisyklės tipas yra **Mažiau-už** (kodas MU). Diagramoje taip pat naudojami ir du apribojimai: žemesniojo slenksčio apribojimas (simbolizuoja rutuliukas su raide L) ir numeratorius (simbolizuoja rutuliukas su skaičiumi 5).



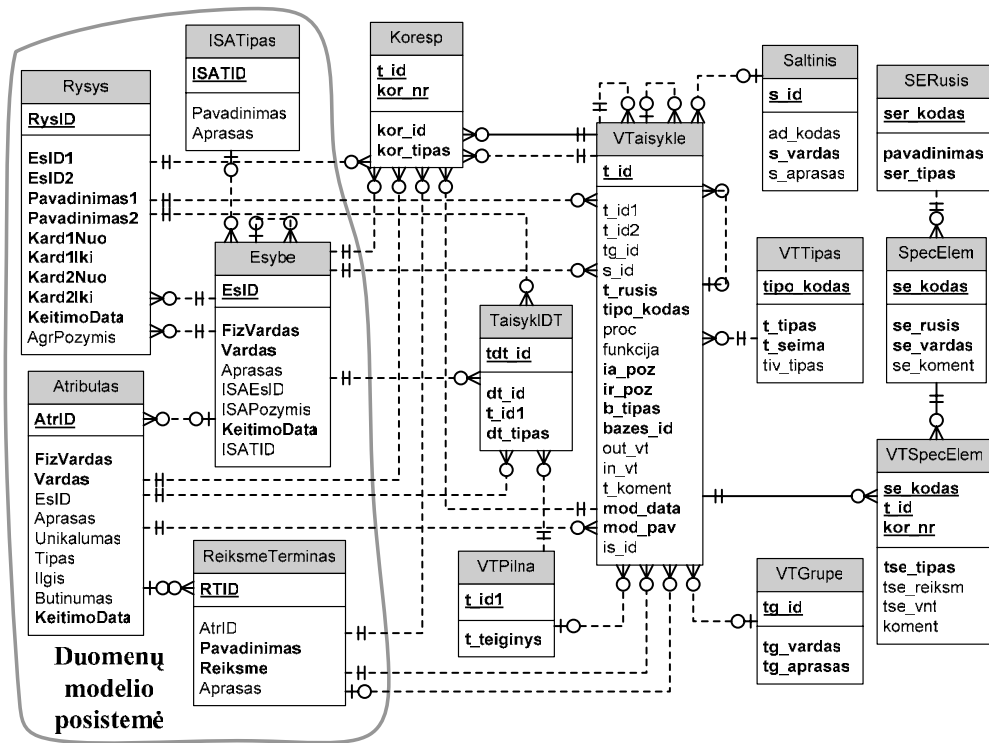
2 pav. Veiklos taisyklės grafinio modelio iliustracija [2]

Remiantis Roso metodu ir pasiūlymo specifika suformuluoti baziniai reikalavimai formaliai užrašytoms VT pateikti žemiau.

1. Veiklos taisyklė privalo būti priskirta vienam ir tik vienam tipui.
2. Veiklos taisyklė privalo turėti vieną ir tik vieną bazę.
3. Veiklos taisyklės bazė gali būti arba duomenų tipas, arba konstanta, arba kita VT, arba kitos VT išeišios reikšmė.
4. Veiklos taisyklė privalo turėti ne mažiau negu 1 korespondentą.
5. Veiklos taisyklės korespondentas gali būti arba duomenų tipas, arba konstanta, arba kita VT, arba kitos VT išeišios reikšmė.

## 2.2 Pagal modifikuotą Roso metodą formalizuotų veiklos taisyklių saugyklos modelis

Saugyklos loginis modelis buvo kuriamas atsižvelgiant į tai, jog Rosas ar kiti žinomi autoriai tokio modelio neaprašo. Sukurtų VT modelių, duomenų modelio elementų ir susijusios informacijos saugyklos loginis modelis pateiktas 3 paveiksle. Ankstyvosios modelio versijos pristatytos [2, 3]. Modelio diagramoje išskirtos dalykinės srities duomenų modelio elementų saugojimo posistemės lentelės, kurios sukurtos atsižvelgiant į tai, jog šis modelis bus kuriamas išplėstinės esybių ryšių (EER) diagramos pavidalu [2].



3 pav. Modifikuotu Roso metodu formalizuotų VT saugyklos loginės struktūros modelis

Kuriant VT struktūrizavimo grafinėje aplinkoje įrankį buvo siekiama realizuoti šių lentelių pildymą:

1. *VTPilna*. Šioje lentelėje saugoma logiškai nedalomos VT formulotė (kiekviena VT diagramoje gali būti atvaizduojama keliomis semantiškai nedalomomis taisyklėmis).
2. *Vtaisykle*. Tai pagrindinė lentelė, kurioje be kita ko yra saugoma tokia informacija apie VT:
  - a. ar taisyklė yra sąlyga, ar integralumo apribojimas;
  - b. taisyklės tipas;
  - c. bazės tipas ir identifikacinis kodas;
  - d. korespondento identifikacinis kodas;
  - e. ar VT diagramoje yra specialiųjų elementų;
  - f. ar VT turi išeišgos reikšmę;
  - g. neformalios VT grupės ir kilmės šaltinio identifikaciniai kodai;
  - h. VT modifikavusio vartotojo vardas;

- i. VT išsaugojimo data.
3. *TaisyklDT*. Lentelėje saugoma informacija apie VT modelyje naudojamus papildomus duomenų modelio elementus, pvz.: ISA ryšius, esybes ir kt.
4. *Koresp*. Lentelėje saugomos nuorodos į korespondentą(-us), esantį(-ius) VT diagramoje.
5. *VTSpecElem*. Lentelėje saugomi VT diagramoje naudojamų specialiųjų elementų identifikatoriai bei požymis, nurodantis, ar šie elementai diagramoje naudojami kaip interpretatoriai, ar kaip apribojimai.
6. *Saltinis*. Lentelėje saugoma informacija apie VT kilmės šaltinius, kuriais gali būti asmenys, dokumentai, įstatymai, tam tikros įrangos dokumentacija, interneto nuorodos ir pan.
7. *VTGrupe*. Lentelėje saugomi duomenys apie neformalias grupes, kurioms gali būti priskiriamos VT.

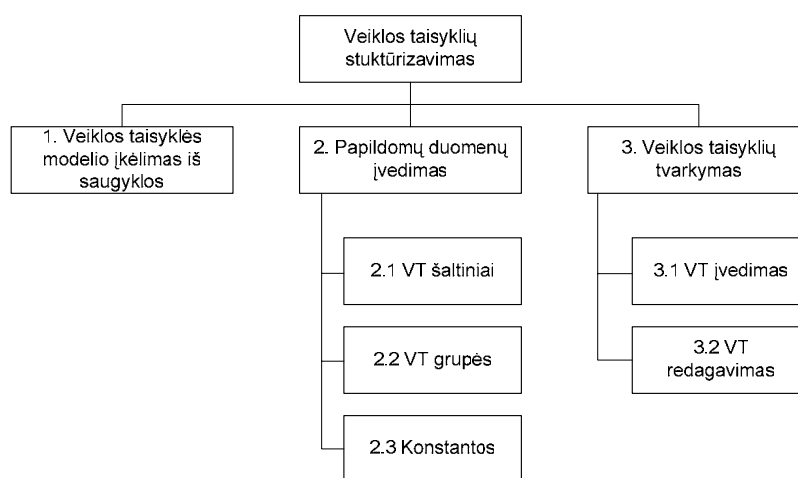
Kitos lentelės yra pildomos naudojantis saugyklos pildymo atributinėje sąsajoje įrankiu, pristatytu [3]. Tai informacija apie taisyklių tipus, specialiuosius elementus bei dalykinės srities duomenų modelio objektus.

### 3. VEIKLOS TAISYKLIŲ STRUKTŪRIZAVIMO ĮRANKIS

#### 3.1 Realizuotos funkcijos

Veiklos taisyklių struktūrizavimo įrankis realizuotas *Microsoft Visio 2000* aplinkoje ir jam buvo keliami šiai aplinkai būdingi vartotojo sąsajos reikalavimai. Prieš pradėdant darbą su įrankiu, vartotojui reikia prisiregistruoti. Tai atliekama vartotojui įvedant savo vardą. Ši funkcija reikalinga tam, kad ateityje būtų galima nustatyti, kas įvedė ar redagavo VT modelį. Prisiregistravus, būtina nurodyti saugyklą, kuri turi būti *Microsoft Access* formato. Tai galima atlikti pasirinkus meniu punktą *Specifikavimas->Pasirinkti DB*.

Pagrindinės VT struktūrizavimo įrankyje realizuotos funkcijos yra šios (žr. 4 pav.):



4 pav. Pagrindinės VT struktūrizavimo įrankio atliekamos funkcijos

1. Veiklos taisyklės modelio įkėlimas iš saugyklos.

Tai VT grafinio modelio generavimas, nuskaitant duomenis iš saugyklos. Ši funkcija iškviečiama pasirenkant programos meniu punktą *Specifikavimas->Nubraižyti VT grafiką iš DB*. Tuomet programa išveda naują langą, kuriame yra visų VT formuluočių, kurios saugomos saugyklos lentelėje *VTPilna*. Pasirinkus reikiamą formulotę, sugeneruojamas VT grafinis modelis. Generuojant modelį, programa kreipiasi į beveik visas saugyklos lenteles – tai priklauso nuo to, ar modelyje yra specialieji elementai, išeigos reikšmės, konstantos ir kt.

## 2. Papildomų duomenų įvedimas.

Šios funkcijos vykdymo metu gali būti įvedama informacija apie:

### 2.1. VT šaltinius.

Prieš išsaugant naują arba redaguojant jau esančią saugykloje VT, būtina nurodyti taisyklės šaltinį. Informacija apie jį saugoma saugyklos lentelėje *Saltinis*. Funkcija iškviečiama, kai išsaugojamas VT modelis. Tuomet programos lange galima pasirinkti iš saugykloje jau esančių šaltinių arba įvesti naują. Įvedant naują VT šaltinį reikia nurodyti jo kodą, pavadinimą ir trumpą aprašymą.

### 2.2. VT grupes.

VT grupę, kaip ir jos šaltinį, būtina nurodyti prieš išsaugant naują arba redaguojant esamą grafinį VT modelį. Informacija apie grupę saugoma saugyklos lentelėje *VTGrupe*. Funkcija iškviečiama tuomet, kai išsaugojamas VT modelis. Tuomet programos lange galima pasirinkti iš jau anksčiau užfiksuotų grupių arba įvesti naują. Įvedant naują veiklos taisyklės grupę, būtina nurodyti jos pavadinimą ir trumpą aprašymą.

### 2.3. Konstantas.

Įkėlus konstantos tipo elementą į VT modeliavimo langą, reikia nustatyti konstantos reikšmę. Visa informacija apie konstantas saugoma saugyklos lentelėje *ReikšmeTerminas*. Funkcija iškviečiama dukart spragtelėjus ant įkelto duomenų modelio elemento arba pasirenkant jo funkciją *Savybės*. Tuomet programa atidaro langą, kuriame galima pasirinkti konstantos reikšmę iš esamų saugykloje arba įvesti naują. Įvedant naują reikšmę, reikia nurodyti ir konstantos pavadinimą.

## 3. Veiklos taisyklių tvarkymas.

Veiklos taisyklių grafinių modelių tvarkymas susideda iš dviejų funkcijų:

### 3.1. VT įvedimas.

Pasirinkus šią funkciją, saugykloje yra išsaugojama nauja struktūrizuota VT. Funkcija iškviečiama pasirinkus meniu punktą *Specifikavimas->Išsaugoti VT modelį į DB*. Programa atidaro naują langą, kuriame būtina nurodyti naujos VT formulotę ir papildomus duomenis – VT šaltinį ir grupę. Išsaugant modelį, informacija saugoma į



saugyklos lenteles *VTPilna*, *VTaisykle*, *TaisyklDT*, *Koresp*. Jei VT grafiniame modelyje yra specialiųjų elementų, informacija išsaugojama lentelėje *VTSpecElem*, konstantų – *ReiksmeTerminas*.

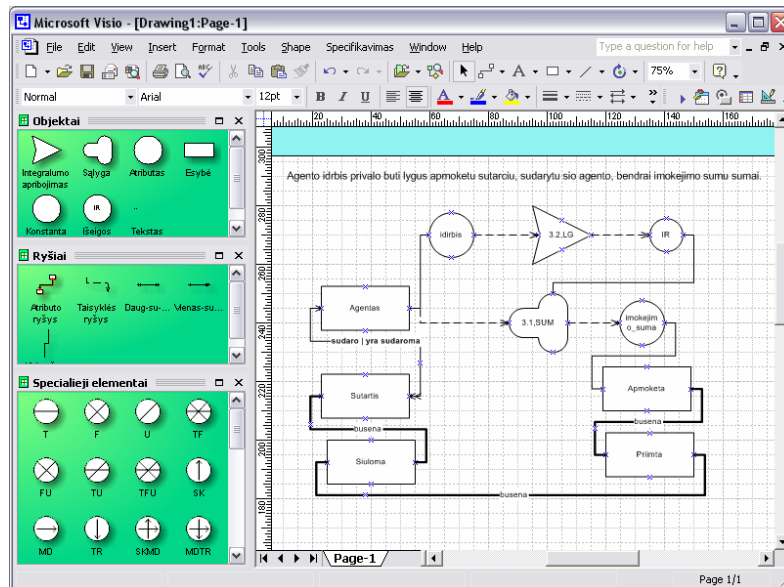
### 3.2. VT redagavimas.

Prieš redaguojant saugykloje jau saugomą VT, ją reikia atvaizduoti grafiškai. Tai atliekama pasirenkant pirmąją funkciją – veiklos taisyklės modelio įkėlimas iš saugyklos. Tuomet, atlikus pakeitimus, galima VT išsaugoti į saugyklą. Funkcija iškviečiama pasirinkus meniu punktą *Specifikavimas->Išsaugoti VT modelį į DB*. Programa atidaro naują langą, kuriame jau yra įrašyta VT formuluotė, grupė ir šaltinis. Grupę ir šaltinį galima keisti naudojant funkcijas *VT šaltiniai* ir *VT grupės*. Išsaugant pakeistą grafinį veiklos taisyklių modelį, informacija saugoma tose pačiose lentelėse, kaip ir funkcijos *VT įvedimas* atveju.

VT įvedimo ir VT redagavimo metu yra atliekamas veiklos taisyklės grafinio modelio teisingumo patikrinimas. Tikrinama ar VT modeliavimo lange nėra įkelta nepanaudotų taisyklių simbolių, duomenų modelio elementų, ryšių, specialiųjų elementų, konstantų. Taip pat atliekamas modelio teisingumo patikrinimas. Nustatoma, ar VT modelis turi bazę ir korespondentą. Tikrinama, ar apribojimų grupei priklausančių specialiųjų elementų nėra įkelta tarp taisyklės ir korespondento, taip pat ar nėra įkelta interpretatorių grupei priklausančių specialiųjų elementų tarp bazės ir taisyklės. Įrankiui nustačius klaidas, yra išvedami atitinkami pranešimai, o klaidingi elementai nudažomi raudona spalva.

### 3.2 Grafinių elementų rinkiniai

Kad būtų galima atlikti VT modelių kūrimą, išsaugojimą ir redagavimą, sukurti specialūs grafinių elementų rinkiniai (angl.: *stencils*) (fragmentai matomi 5 pav.):



5 pav. Veiklos taisyklių struktūrizavimo įrankio langas

1. Ryšiai – tai taisyklės grandinės ryšys bei duomenų modelio ryšiai, numatyti EER diagramos notacijoje.
  - a. *Atributo ryšys*. Šis ryšys naudojamas duomenų modelio esybei ir jos atributu susieti.
  - b. *Taisyklės ryšys*. Tarpusavyje susieja bazę, taisyklę, korespondentą ir specialiuosius elementus.
  - c. *Daug su daug* ir *Vienas su daug* ryšiai. Tai ryšiai tarp esybių. Įkėlus tokį ryšį jį galima pasirinkti iš esančių saugykloje (pasirinkus įrankio meniu punktą *Savybės* arba du kartus spragtelėjus ant įkelto objekto) – reikiamos duomenų modelio esybės bus nubraižomos automatiškai.
  - d. *ISA ryšys*. Susieja duomenų modelio esybę ir jos potipius [4].
2. Specialieji elementai – tai visi 36 Roso metode numatyti specialieji grafiniai elementai (interpretatoriai ir apribojimai). Jų sąrašas ir kodai atitinka saugykloje saugomą informaciją.
3. Objektai – tai Roso notacijoje numatyti VT simboliai (integralumo apribojimo ir sąlygos) bei terminų klasės duomenų modelio elementams (esybėms, atributams) vaizduoti skirti simboliai.
  - a. *Integralumo apribojimas* ir *Sąlyga*. Tai VT grafinį modelį sudarančios nedalomos taisyklės. Įkėlus taisyklės, būtina nustatyti ir jų tipus bei numerius. Tai padaroma pasirenkant meniu punktą *Savybės* arba du kartus spragtelėjus ant įkelto elemento.
  - b. *Atributas*. Duomenų modelio elementas, esybės atributas. Įkėlus atributo grafinį elementą, būtina pasirinkti vieną iš saugykloje saugomų reikšmių. Tai galima padaryti pasirinkus meniu punktą *Savybės* arba du kartus spragtelėjus ant įkelto elemento. Pasirinkus atributą, susijusi esybė nubraižoma automatiškai.
  - c. *Esybė*. Duomenų modelio esybė. Kaip ir atributo atveju, esybės elemento reikšmė nustatoma pasirenkant iš saugykloje saugomų reikšmių. Tai atliekama pasirenkant meniu punktą *Savybės* arba du kartus spragtelėjus ant įkelto elemento.
  - d. *Konstanta*. Vaizduoja VT modelyje naudojamas konstantas. Norint nustatyti konstantos reikšmę, reikia pasirinkti meniu punktą *Savybės* arba du kartus

spragtelėti ant įkelto objekto. Galima pasirinkti konstantos reikšmę iš jau esamų saugykloje arba įsivesti naują.

e. *Išeigos reikšmė*. Taisyklės išeigos reikšmė. Papildomai specifikuoti nereikia.

Sukurtasis įrankis buvo preliminariai testuojamas sukuriant skirtingomis charakteristikomis pasižyminčių VT modelius bei generuojant jų diagramas pagal saugykloje išsaugotą informaciją. 5 paveiksle pavaizduotas sukurtojo įrankio veiklos taisyklių grafinio modeliavimo langas, kuriame matoma sumodeliuota VT.

#### 4. IŠVADOS

Straipsnyje pristatytas VT struktūrizavimo pagal modifikuotu Roso metodu grindžiamą modelį programinis įrankis, leidžiantis kurti nestruktūrinių VT modelius grafinėje aplinkoje.

Nustatyta, jog įrankis leidžia be išsaugoti grafinius VT modelius saugyklos duomenų bazėje bei juos be praradimų atkurti tik pagal saugykloje užfiksuotą informaciją. Tai iš dalies patvirtina, jog panaudotas VT struktūrizavimo modelis gali būti taikomas praktikoje, o VT diagramos gali būti išsaugomos atributiniame pavidale. Priimama prielaida, jog tokiu pavidalu išsaugojus VT modelius galima sukurti specifinį VT variklį, leisiantį atlikti automatinį VT interpretavimą jau realizuotoje IS [2, 3]. Tokiu būdu parodoma, jog veiklos modeliavimui skirtas Roso metodas yra tinkamas ir VT struktūrizavimui IS projektavimo fazėje atlikti.

Ateityje bus atliekamas išsamus eksperimentas bei formuluojamos darbo išvados.

#### LITERATŪROS SĄRAŠAS

- [1] Business Rules Group. Defining Business Rules ~ What Are They Really? (3<sup>rd</sup> edition). Business Rules Group interneto puslapis [interaktyvus], 2000 [žiūrėta 2004-10-15], prieiga per internetą: <<http://www.BusinesRulesGroup.org>>.
- [2] Butleris, R., Kapočius, K. Struktūrizuotų veiklos taisyklių saugyklos architektūra. Informacijos mokslai 17 Vilnius, Vilniaus universiteto leidykla, 2001: pp. 46-57.
- [3] Butleris, R., Kapočius, K. The Business Rules Repository for Information Systems Design. The 6<sup>th</sup> East-European Conference ADBIS'2002. Konferencijos pranešimų medžiaga. Bratislava, Slovakia, Vydavatel'stvo STU, Vol.2, 2002: pp.64-77.
- [4] Elmasri, R., Navathe, S. B. Fundamentals of Database Systems, Third Edition. Addison-Wesley, 2002.
- [5] Ross, R. G. The Business Rule Book: Classifying, Defining and Modeling Rules. Business Rule Solutions, Houston, 1997.

The Tool for Business Rules Structuring during the Information Systems Design

The paper presents the modified Ross method-based model for business rules structuring during the information systems design as well as a software implementation of this model. The developed tool provides the designer with possibilities to create graphical business rules

models, relating them with the specific elements of universe of discourse data model. The models are then processed and stored in a specialized repository. The presentation includes the overview of the model as well as the analysis of the specific features of the developed tool.