

**VILNIAUS UNIVERSITETAS
KAUNO HUMANITARINIS FAKULTETAS**

INFORMATIKOS KATEDRA

Verslo informatikos studijų programa
Kodas 62109P101

RITA JONAITYTĖ

MAGISTRO BAIGIAMASIS DARBAS

**NATŪRALIOSIOS KALBOS ŠABLONAIŠ UŽRAŠYTŲ VEIKLOS
TAISYKLIŲ PANAUDOJIMAS PROJEKTUOJANT IR REALIZUOJANT
INFORMACINES SISTEMAS**

Kaunas 2009

VILNIAUS UNIVERSITETAS
KAUNO HUMANITARINIS FAKULTETAS
INFORMATIKOS KATEDRA

RITA JONAITYTĖ

MAGISTRO BAIGIAMASIS DARBAS

**NATŪRALIOSIOS KALBOS ŠABLONAIŠ UŽRAŠYTŲ VEIKLOS
TAISYKLIŲ PANAUDOJIMAS PROJEKTUOJANT IR REALIZUOJANT
INFORMACINES SISTEMAS**

Leidžiama ginti _____

Magistrantė _____
(parašas)

Darbo vadovas _____
(parašas)

doc.dr. Gintautas Garšva

Darbo konsultantas _____
(parašas)

lekt. dr. Kęstutis Kapočius

Darbo įteikimo data _____

Registracijos Nr. _____

Kaunas 2009

TURINYS

PAVEIKSLŲ SĄRAŠAS.....	4
LENTELIŲ SĄRAŠAS.....	5
SANTRUMPŲ SĄRAŠAS	6
SANTRAUKA	7
ĮVADAS.....	8
1. PROBLEMINĖS SRITIES ANALIZĖ	10
1.1 Literatūros šaltinių analizė	10
1.2 Natūralioji kalba VT specifikuojant, BRS RuleSpeak	12
1.2.1 BRS RuleSpeak modelis	13
1.2.2 VT grindžiamas reikalavimų specifikuojant metodas	14
1.2.3.1 Reikalavimų specifikuojant ir analizės stadijoje naudojami VT modeliai.....	15
1.2.3.2 Struktūrinių VT fiksavimo forma.....	15
1.2.3.3 Nestruktūrinių VT modelis.....	16
1.3 Alternatyvūs metodai	17
1.4 Formali kalba VT projektavime, Roso metodas.....	18
1.4.1 Roso metodas	18
1.4.2 Veiklos taisyklių formalizavimas pagal Roso metodą	22
1.4.3 Pagal Roso metodą formalizuotų veiklos taisyklių saugyklos modelis	24
1.4 Apibendrinimas	26
2. VT TRANSFORMAVIMO IŠ BRS RULESPEAK Į ROSO METODĄ KONCEPCIJA.....	29
2.1 Veiklos taisyklių transformavimo prielaidos	29
2.1.1 Perėjimo procesas.....	29
2.1.2 Išimtiniai atvejai	34
2.2 Perėjimo nuo specifikuojant prie projektavimo principas - metodika	36
2.2.2 Subjekto transformavimas.....	37
2.2.3 Sąlyga ar apribojimas	39
2.2.4 Fakto/termino kaip korespondento nustatymas	40
2.2.5 Roso VT tipo nustatymas pagal BRS RuleSpeak tipą.....	42
2.2.5.1 Atmetimo tipo taisyklės	43
2.2.5.2 Generavimo tipo taisyklės	46
2.2.5.3 Numatymo tipo taisyklės.....	47
2.2.5.4 VT tipų nustatymo sunkumai	49
2.2.6 Specialiųjų elementų identifikavimas	50
2.2.7 Papildomų duomenų modelio elementų išvedimas.....	51
2.3 VT transformavimo metodikos apibendrinimas.....	51
3. EKSPERIMENTINIS VT TRANSFORMAVIMO METODIKOS TYRIMAS.....	53
3.1 Eksperimento koncepcija	53
3.2 Eksperimento dalykinės srities charakteristika	53
3.3 Veiklos taisyklių specifikuojant pagal BRS RuleSpeak	54
3.4 Specifikuotų veiklos taisyklių transformavimas pagal sukurtą metodiką.....	56
3.5 Eksperimento rezultatai.....	69
3.5.1 Metodikos efektyvumo įvertinimas.....	69
3.5.2 Eksperimento rezultatų apibendrinimas	70
IŠVADOS.....	73
LITERATŪRA.....	74
1 PRIEDAS. Natūraliosios kalbos VT šablonai (modifikuotas BRS RuleSpeak).....	77
2 PRIEDAS. Nedalomų taisyklių tipų savybės pagal Roso metodą	80
3 PRIEDAS. Išvestinių taisyklių tipai pagal Roso metodą	83

PAVEIKSLŲ SARAŠAS

1 pav. Veiklos taisyklių tipai.....	10
2 pav. Veiklos taisyklių vieta trijų lygmenų IS.....	11
3 pav. Nestruktūrinių VT šablonų loginis modelis	17
4 pav. Taisyklės vykdymo ir taisyklės tikrinimo žymėjimas	20
5 pav. Veiklos taisyklės grafinio modelio iliustracija.....	22
6 pav. Veiklos taisyklės formalizavimo pagal Roso metodą sintaksė (pagal [21]).....	23
7 pav. Modifikuotu Roso metodu formalizuotų VT saugyklos loginės struktūros modelis	26
8 pav. Tyrimo objektų vieta IS kūrimo procese	28
9 pav. Apribojimo/atmetimo tipo taisyklė (šablonas b1) Roso metodu.....	30
10 pav. Apribojimo/atmetimo tipo taisyklė (šablonas b2) Roso metodu.....	31
11 pav. Leidimo tipo taisyklė (šablonas b5) Roso metodu	32
12 pav. Skaičiavimo tipo taisyklė (šablonas b6) Roso metodu.....	32
13 pav. Išvedimo tipo taisyklė (šablonas b9) Roso metodu.....	33
14 pav. Išvados tipo taisyklė (šablonas b10) Roso metodu.....	34
15 pav. Principinė transformavimo metodikos schema	36
16 pav. Subjekto transformavimas	39
17 pav. Apribojimo tipo taisyklė Roso metodu.....	43
18 pav. Bendro tipo taisyklė Roso metodu	44
19 pav. Unikalaus tipo taisyklės panaudojimo pavyzdys.....	44
20 pav. Įgalinimo tipo taisyklė Roso metodu.....	45
21 pav. Palyginamojo įvertinimo tipo taisyklė Roso metodu	45
22 pav. Matematinio įvertinimo (atimties) tipo taisyklė Roso metodu.....	46
24 pav. Įgalinimo tipo taisyklė Roso metodu.....	48
25 pav. Kopijavimo tipo taisyklė Roso metodu	48
26 pav. Paleidimo tipo taisyklė Roso metodu	49
27 pav. Koncepcinis dalykinės srities duomenų struktūros modelis.....	54
28 pav. Naudojamo įrankio prototipo VT braižymo aplinka	57
29 pav. VT 1 diagrama sugeneruota automatiškai (a) ir pilnai užbaigta (b).....	58
30 pav. VT 2 diagrama sugeneruota automatiškai (a) ir pilnai užbaigta (b).....	59
31 pav. VT 2 diagrama sugeneruota automatiškai (a) ir pilnai užbaigta (b).....	60
32 pav. VT 4 diagrama sugeneruota automatiškai (a) ir pilnai užbaigta (b).....	62
33 pav. VT 5 diagrama sugeneruota automatiškai (a) ir pilnai užbaigta (b).....	63
34 pav. VT 6 diagrama sugeneruota automatiškai (a) ir pilnai užbaigta (b).....	65
35 pav. VT 9 diagrama sugeneruota automatiškai (a) ir pilnai užbaigta (b).....	66
36 pav. VT 12 diagrama sugeneruota automatiškai (a) ir pilnai užbaigta (b).....	68

LENTELIŲ SĄRAŠAS

Lentelė 1. Galimi modifikuoto <i>BRS RuleSpeak</i> modelio VT šablonų elementų tipai.....	16
Lentelė 2. Nedalomų taisyklių tipai pagal Roso metodą	19
Lentelė 3. VT rinkinys pagal <i>BRS RuleSpeak</i> šablonus	55
Lentelė 4. Roso metodu nemodeliuojamos VT	56
Lentelė 5. 1 VT susiejimas su pagrindiniais šablono elementais	57
Lentelė 6. 2 VT susiejimas su pagrindiniais šablono elementais	59
Lentelė 7. 3 VT susiejimas su pagrindiniais šablono elementais	60
Lentelė 8. 4 VT susiejimas su pagrindiniais šablono elementais	61
Lentelė 9. 5 VT susiejimas su pagrindiniais šablono elementais	63
Lentelė 10. 6 VT susiejimas su pagrindiniais šablono elementais	64
Lentelė 11. 9 VT susiejimas su pagrindiniais šablono elementais	66
Lentelė 12. 12 VT susiejimas su pagrindiniais šablono elementais	67
Lentelė 13. Grafinių elementų pateikimo įvertinimas.....	69
Lentelė 14. Metodikos įvertinimas	71
Lentelė 15. VT šablonai pagal <i>BRS RuleSpeak</i>	77
Lentelė 16. Nedalomų taisyklių tipų savybės pagal Roso metodą.....	80
Lentelė 17. Išvestinių taisyklių tipai pagal Roso metodą.....	83

SĄNTRUMPŲ SĄRAŠAS

BRS RuleSpeak - kompanijos Business Rule Solutions (BRS) sukurta VT aprašymo panaudojant natūraliosios kalbos šablonus metodika.

MS Visio – Microsoft Visio tai pasaulyje pripažinta diagramų kūrimo programa.

OMG – (Object Management Group) tarptautinė standartų organizacija, kurios tikslas sukurti modeliavimo standartus programoms, sistemoms ir verslo procesams.

OCL – (Object Constraint Language) modeliavimo kalba UML meta-modeliams, kitiems OMG standartais apibrėžiamiems meta-modeliams apribojimams nusakyti.

PRR – (Production Rule Representation) OMG sukurtas produkcinų VT apibrėžimo standartas.

SBVR – (Semantics of Business Vocabulary and Business Rules) OMG sukurtas VT apibrėžimo standartas, orientuotas į semantinę natūraliosios kalbos sakinių, kuriais išreiškiamos VT, struktūrą.

VT – veiklos taisyklė, tam tikrus veiklos aspektus apibrėžiantis arba apribojantis teiginys, kuris reikalingas norint įvertinti veiklos struktūrą arba valdyti/įtakoti veiklą.

SANTRAUKA

JONAITYTĖ, Rita. (2009) *Application of the Natural Language Templates Based Business Rules During the IS Design and Implementation*. MBA Graduation Paper. Kaunas: Vilnius University, Kaunas Faculty of Humanities, Department of Informatics. 67 p.

SUMMARY

The main purpose of this paper is to create a methodology for the transition of formalized, natural language templates based business rules to formal Ross charts. For this purpose these tasks were determined :

- 1. Examine the concept of business rules, its specification using natural language templates and the BRS RuleSpeak model.*
- 2. Examine the modelling of business rules according to the Ross method.*
- 3. Create a methodology by which the rules expressed by the modified BRS RuleSpeak would be transformed into rules based on the modified Ross method.*
- 4. To test and evaluate the developed methodology.*

The natural language templates and Ross charts methods were compared and the similarities of their models, patterns, syntactic elements and types of business rules groups were distinguished. According to the results of the comparison the methodology defining the transformation of BRS RuleSpeak natural language templates based on business rules to the Ross charts expressed models was created.

The main results of the work are:

- The analysis of natural language templates expressed business rules according to BRS RuleSpeak method and their transition to Ross models was made;*
- A methodology for the transition of the requirements to the design specifications stage was created;*
- 6 steps of the transformation from BRS RuleSpeak natural language templates based BR on Ross charts were distinguished.*

The automation of the transition of the rule-based requirements to formal models is possible only in part. The design process is greatly accelerated.

The volume of the paper is 67 pages. In the main chapters there are 17 tables and 36 pictures.

ĮVADAS

Šiuolaikinės informacinės sistemos (IS) pasiekė tokį sudėtingumo lygį, kad joms įgyvendinti reikia viso informacinių technologijų (IT) komplekso, kurį sunku aprėpti ir valdyti be automatizuotų priemonių [17]. Bet kurios IS kūrimas susideda iš keleto etapų: reikalavimų rinkimo, analizės ir specifikavimo, IS projektavimo, realizavimo, testavimo ir palaikymo. Sėkmingas pradinių kūrimo etapų realizavimas dažniausiai nulemia ir viso projekto sėkmę.

Būtent reikalavimų specifikavimo ir projektavimo stadijose sukuriamų modelių kokybė yra akcentuojama taip vadinamoje veiklos taisyklių (VT) koncepcijoje, kurios esmė – atskirti įvairius apribojimus duomenų elementams bei procesams ir juos išreikšti, modeliuoti bei valdyti atskirame, veiklos taisyklių IS lygmenyje [5]. Idealiu atveju, taikant VT koncepciją, sistemos reikalavimų specifikavimo stadijoje užfiksuotos veiklos taisyklės užrašomos struktūrizuota natūraliąja kalba, taikant šablonus. Tuo tarpu projektavimo stadijoje taisyklių išraiškos formalizuojamos. Šiame darbe akcentuojamas tokio proceso variantas, kuomet reikalavimų specifikavimo metu naudojamas natūralios kalbos šablonais grindžiamas BRS RuleSpeak modelis [13, 23], o projektuojant IS, taisyklės modeliuojamos pagal grafine notacija grindžiamą Roso metodą [11, 2]. **Problema** yra perėjimas nuo reikalavimų prie projektavimo, t. y. nuo formalizuotų, natūraliąja kalba užrašytų VT prie formalių Roso taisyklių diagramų. Nors yra prielaidų, jog galima šį perėjimą dalinai automatizuoti, metodikų, aprašančių, kaip tai atlikti, neaptikta.

Darbo **objektas** – VT specifikavimo kuriant IS metodai ir jų apjungimas.

Darbo **tikslas** – pasiūlyti metodą, apjungiantį duotus dviejų formalumo lygmenų veiklos taisyklių specifikavimo modelius, sukuriant natūraliosios kalbos šablonais specifikuotų veiklos taisyklių transformavimo į formalias projektinio lygmens išraiškas metodiką.

Siekiant įvykdyti iškeltą tikslą, darbo metu buvo sprendžiami šie **uždaviniai**:

1. Išnagrinėti VT koncepcijos pagrindus, specifikavimo, panaudojant natūraliosios kalbos šablonus, modelį BRS RuleSpeak.
2. Išnagrinėti VT modeliavimą pagal Roso metodą.
3. Sukurti metodiką, pagal kurią modifikuotu BRS RuleSpeak išreikštos taisyklės galėtų būti transformuojamos į modifikuotu Roso metodu išreikštas taisykles.
4. Išbandyti ir įvertinti sukurtą metodiką.

Darbas susideda iš 3 pagrindinių dalių. Probleminės srities analizės skyriuje nagrinėjami darbe naudojami VT specifikavimo ir projektavimo metodai. VT transformavimo iš BRS RuleSpeak į Roso metodą skyriuje pateikiamos modelių susiejimo gairės ir koncepcija. Eksperimentiniame VT transformavimo metodikos tyrimo skyriuje aprašomas eksperimentas ir

gaunami rezultatai. Išvadų skyriuje aptariami gauti darbo rezultatai bei pateikiami pasiūlymai, pastebėjimai nagrinėta tema.

Rašant darbą pagrindiniai naudoti literatūros šaltiniai yra lietuvių autorių K. Kapočiaus ir R. Butlerio staipsniai, kurie yra įtraukti į ISI ir kitas tarptautines duomenų bazines. Taip pat naudoti pirminiai informacijos apie nagrinėjamus metodus šaltiniai, kurių autorius Ronald Ross ir kiti. Naudojamosi internete publikuojamais dokumentais, kuriuose apibrėžtos veiklos taisyklės ir jų specifikavimas. Medžiagos taip pat ieškota nekomercinių organizacijų, kurias sudaro patyrę verslo sistemų specialistai, nagrinėjantys veiklos taisykles, puslapiuose.

Tyrimo metu analizuojami duoti modifikuoti VT specifikavimo ir projektavimo metodai. Nagrinėjant metodus išvelgta struktūrų panašumų, bendrų bruožų jų susiejime su IS duomenų modelių elementais. Pagal šias išvalgas, sukurta transformavimo metodika. Eksperimento, kuriuo buvo siekiama išbandyti ir įvertinti metodiką, metu atliktas pasirinktos dalykinės srities VT modeliavimo imitavimas. Eksperimentas parodė, kad metodika neužtikrina visaverčio Roso VT modelio gavimo iš pagal modifikuotą BRS RuleSpeak užrašytų taisyklių. Taigi, perėjimo nuo reikalavimų specifikavimo metu užfiksuotų veiklos taisyklių prie projektinių modelių automatizavimas pagal paruoštą metodiką įmanomas tik dalinai.

Be šio įvado darbą sudaro trys pagrindiniai dėstymo skyriai (teorinis/analitinis, siūlomos metodikos, eksperimentinis), literatūros sąrašas ir 3 priedai. Darbe pateikiami 36 paveikslėliai ir 17 lentelių.

1. PROBLEMINĖS SRITIES ANALIZĖ

Norint atlikti išsikeltus uždavinius reikia apžvelgti situaciją IS kūrimo pradinėms stadijoms skirtų modelių įvairovę ir esamus sprendimus.

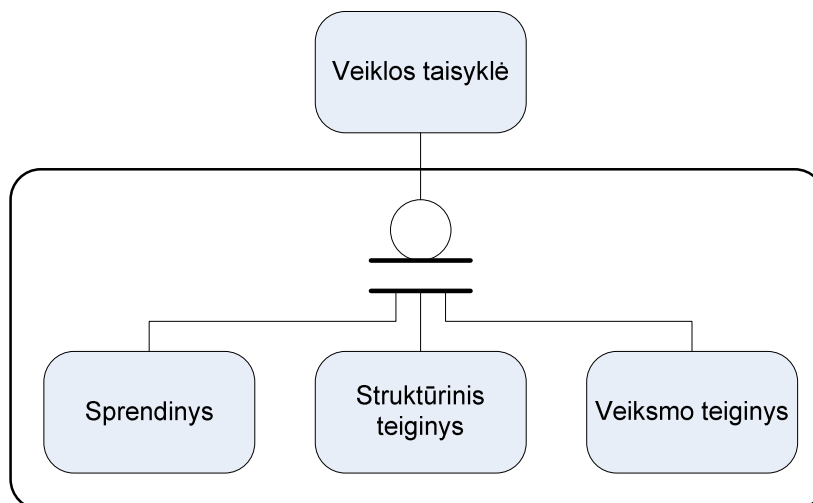
1.1 Literatūros šaltinių analizė

GUIDE projektas, dabar žinomas kaip Veiklos taisyklių grupė, pateikta tokį VT apibrėžimą:

Tam tikrus veiklos aspektus apibrėžiantis arba apribojantis teiginys, kuris reikalingas norint įvertinti veiklos struktūrą arba valdyti/įtakoti veiklą. Jokia VT negali būti suskaidyta į detalesnes VT neprarandant svarbios informacijos apie veiklą.

Pagal GUIDE apibendrinti VT tipai yra:

- *Struktūrinis teiginys (structural assertion)* – apibrėžtas teiginys arba koncepcija faktų, kurie išreiškia kurį nors verslo sistemos struktūros elementą.
- *Veiksmo teiginys (action assertion)* – apribojimo arba sąlygos teiginys, kuris riboja arba valdo verslo sistemos veiksmus.
- *Sprendinys (derivation)* – teiginys išvestas iš žinių turimų apie verslą [5].



Šaltinis: Sudaryta autoriaus pagal Business Rules Group. (2000). *Defining Business Rules ~ What Are They Really?* (3rd edition).

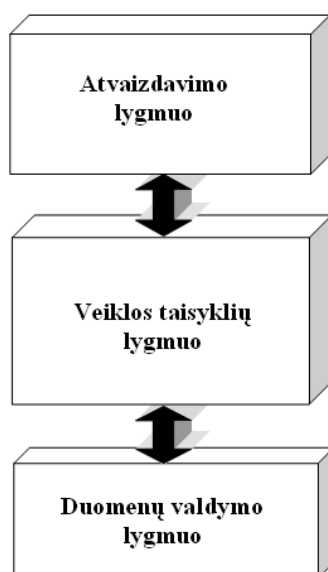
1 pav. Veiklos taisyklių tipai

Pagal šią apibrėžtį veiklos taisykle laikytini ne tik teiginiai ir apribojimai, reikalingi norint valdyti ar įtakoti veiklą, bet ir teiginiai, reikalingi norint apibrėžti ar įvertinti veiklos struktūrą [4].

Lina Nemuraitė habilitacijos mokslo darbų apžvalgoje tema „Modeliais grindžiamo informacinių sistemų projektavimo modeliai ir metodai“ [17] apžvelgia esamą IS kūrimo situaciją. Modeliais grindžiamo kūrimo metuose pagrindinis dėmesys sutelktas į paskutinius kūrimo proceso etapus. Pradiniai kūrimo proceso etapai – reikalavimų specifikavimas ir analizė, dalykinės srities tuo labiau veiklos modeliavimas, lieka nepaliesti. Formalizuotas perėjimas nuo reikalavimų prie projektinių specifikacijų yra labai svarbus ir turėtų būti kiek įmanoma automatizuotas, kad palengvinti projektuotojų darbą ir paspartintų sistemos kūrimą.

Pagal R. Butlerį ir K. Kapočių, tradicinė IS gali būti traktuojama kaip dvisluoksnė, t.y. susidedanti iš dviejų pagrindinių lygmenų – duomenų valdymo lygmens ir vartotojo arba atvaizdavimo lygmens [7]. Sukūrus tokią sistemą, jos VT yra realizuojamos programiniame kode. Vėliau norint jas pakeisti ar pridėti naujas, sudėtinga tokioje sistemoje atrasti konkrečiai reikiama vieta. Tokiu būdu sistemoje taisyklės gali būti realizuojamos daugiau nei vienoje vietoje, persidengti, nebeatitikti pirminio pavidalo. Sistemoje įterpus VT lygmenį, ją galima vadinti trisluoksne [11]. VT lygmuo yra nepriklausomas (nuo duomenų struktūrų ir procesų) ir tai suteikia sistemai lankstumo. Tai suteikia galimybę nesudėtingai keisti VT sistemos gyvavimo metu [21].

Iš dviejų tradicinių lygmenų bei tradicinio programinės įrangos lygmens VT pjūvis perima daugumą duomenų tikrinimo, algoritmų logikos, apsaugos bei kitų funkcijų [2].



Šaltinis: Butleris, R., Kapočius, K. (2001). *Struktūrizuotų veiklos taisyklių saugyklos architektūra*. Informacijos mokslai. Vilnius. Vilniaus universiteto leidykla, 17 tomas, 46-57.

2 pav. Veiklos taisyklių vieta trijų lygmenų IS

R. Butleris savo straipsnyje apžvelgia reikalavimų specifikavimo problemas ir išskiria tai, kad reikalavimus sistemai reikia rinkti viso IS kūrimo metu [1]. Straipsnyje minima, kad kuriami reikalavimų specifikavimo modelių prototipai, pateikiamas reikalavimų specifikacijos abstrahuoto

modelio sudarymo procesas. Į modelį įtraukiama ir VT specifikacija, tačiau VT kaip atskiras sistemos sluoksnis neišskirtos.

K. Kapočiaus ir R. Butlerio skelbtoje publikacijoje yra aprašomas VT grindžiamos IS reikalavimų saugyklos koncepcija [21]. Autoriai nagrinėja reikalavimų specifikavimo stadijos metodą bei VT modelį. Anot I. Valatkaitės ir O. Vasileco VT saugyklos modelio sudarymas yra daugiau apibendrinta vizija, nes praktikoje VT grindžiamos IS architektūros, kaip ir VT saugyklos struktūros, standartų faktiškai nėra [26]. ADBIS'2002 konferencijai skirtame straipsnyje K. Kapočius ir R. Butleris pateikia IS projektavimo VT modelį.

K. Kapočius parašęs disertaciją "*Veiklos taisyklių struktūrizavimo modeliai ir jų taikymas kuriant informacijos sistemas*". Tai pagrindinis informacijos apie modifikuotus BRS RuleSpeak ir Roso metodus šaltinis. Kaip pirminiai didžiosios dalies informacijos šaltiniai naudojamos R. Ross "*The Business Rule Book: Classifying, Defining and Modeling Rules*" ir "*Principles of the Business Rules Approach*" knygos.

Užsienio autoriai, R. Ross, T. Morgan, įvairios organizacijos, pvz., BRS (Business Rule Solutions), taip pat pateikia darbus, orientuotus tik į vieną IS kūrimo pradžios stadiją – reikalavimų specifikavimą arba projektavimą.

Atlikus literatūros analizę daroma išvada, kad yra daug dirbama prie IS reikalavimų specifikavimo ir projektavimo metodų. Tačiau gilinamasi yra tik į vieną stadiją, o perėjimas tarp jų lieka nenagrinėtas.

1.2 Natūralioji kalba VT specifikavime, BRS RuleSpeak

Užrašant VT struktūrizuota natūraliąja kalba, pasiekiamas sąlyginai žemas formalumo lygis, t.y. gaunamos formalizuotos, bet ne formalios VT išraiškos. Ši natūraliosios kalbos šablonų savybė lemia tai, jog šablonais grindžiami VT modeliai paprastai yra labiau tinkami IS reikalavimų specifikavimui arba tiesiog tam tikros veiklos srities analizei VT pagrindu. Šioje stadijoje naudojami VT šablonai padeda užtikrinti taisyklių rinkinio vientisumą ir bent minimalų loginį teisingumą [9]. Išreiškiant vartotojo reikalavimus ir poreikius struktūrizuota natūraliąja kalba užrašytomis taisyklėmis, taip pat yra sprendžiamos tokios vartotojo poreikių specifikavimo problemos:

1. Veiklos taisyklės išreiškiamos aiškia, nedviprasmiška ir gerai struktūrizuota natūraliąja kalba.
2. Išryškinama veiklos logika, nes neformalus sakinytis išspraudžiamas į formalizuotų šablonų rėmus.

3. Pagerinama komunikacija tarp verslo atstovų ir analitikų, nes visos VT užrašomos vienu pavidalu, suprantamu abiems pusėms.
4. Veiklos taisyklės susiejamos su IS kūrimu, nes įgydamos formalizuotą pavidalą jos gali būti skaidomos į atskiras dalis ir naudojamos specifikuojant reikalavimus, kuriuos turės tenkinti kuriama sistema [15].

VT šablonų schemas siūlo naudoti ne vienas VT srities autorius. Galima paminėti R. Ross [23], T.Morgan [16], Von Halle [27], Lam [22]. Šablonų sintaksė paremta EBNF standartu ir yra labai paprasta [10]:

- () – apriboja alternatyvių šablono elementų grupę,
- { } – apriboja pasikartojančių elementų grupę,
- [] – apriboja neprivalomų šablono elementų grupę,
- | – atskiria galimus alternatyvius variantus,
- < > – apriboja specialius šablono elementus (galimos reikšmės priklauso nuo šablonų modelio) [15].

1.2.1 BRS RuleSpeak modelis

Ronald G. Ross siūlo naudoti kompanijos Business Rule Solutions (BRS) sukurtą VT aprašymo panaudojant natūraliosios kalbos šablonus metodiką BRS RuleSpeak [22, 23]. Svarbiausia BRS RuleSpeak savybė – natūraliosios kalbos šablonai, bet metodika siūlo ir kitokių VT užrašymo patarimų ir gairių.

BRS RuleSpeak naudojama gana nesudėtinga VT klasifikavimo sistema. Priklausomai nuo reakcijos į įvykius pobūdžio, taisyklės gali būti priskiriamos vienai iš trijų kategorijų:

1. Atmetimo taisyklės, kitaip dar vadinamos apribojimais (angl.: *rejectors*). Tai bet kuri VT, kuri paprastai neleidžia įvykti įvykiui, jeigu tai iššauktų taisyklės pažeidimą [23].
2. Generavimo taisyklės (angl.: *producers*). Tai bet kuri taisyklė, kuri nei atmeta, nei numato įvykius, o tiesiog pagal tam tikrą matematinę funkciją apskaičiuoja arba išveda reikšmę [23]. Generavimo taisyklės gali būti šių tipų:

- 2.1. skaičiavimo taisyklė,
- 2.2. išvedimo taisyklė.

3. Numatymo taisyklės, kitaip dar vadinamos stimulo/reakcijos taisyklėmis (angl.: *projectors*). Tai bet kuri taisyklė, kuri paprastai iššaukia tam tikrą veiksmą (išskyrus atmetimo), kuomet įvyksta tam tikras įvykis [23]. Numatymo taisyklės skirstomos į šiuos tipus ir potipius:

- 3.1. Įgalinimo taisyklės. Priklausomai nuo informacijos, kuri įgalinama arba slopinama, įgalinimo taisyklė gali būti:
 - 3.1.1. išvados taisyklė;
 - 3.1.2. taisyklės jungiklis;
 - 3.1.3. proceso jungiklis;
 - 3.1.4. duomenų jungiklis.
- 3.2. Kopijuotoja. Priklausomai nuo atkartojamos informacijos pobūdžio kopijuotoja gali būti:
 - 3.2.1. reikšmės priskyrimo taisyklė (angl.: *imprint rule*);
 - 3.2.2. pateikimo taisyklė.
- 3.3. Vykdytoja. Priklausomai nuo to, kas yra iškviečiama vykdymui, vykdytoja gali būti:
 - 3.3.1. proceso trigeris;
 - 3.3.2. taisyklės trigeris.

BRS RuleSpeak VT klasifikacija apima dvi iš trijų GUIDE projekte išskirtų VT kategorijų – sprendinius (išvestis) ir veiksmo teiginius. Struktūriniai teiginiai (terminai ir faktai) BRS RuleSpeak yra traktuojami ne kaip veiklos taisyklės, bet kaip faktų modelio elementai.

BRS RuleSpeak metodu struktūrizuotos VT išraiškos pradžioje yra taisyklės *subjektas*, kuriuo gali būti terminas, faktas, duomenų vienetas, kita VT, procesas arba procedūra. Kiekvienam VT tipui siūloma naudoti nuo vieno iki kelių šablonų, kurių kiekviename naudojama tik konkrečiam VT tipui būdinga struktūra ir specifiniai raktiniai žodžiai. Pavyzdžiui, vienas iš atmetimo taisyklėms siūlomų šablonų yra toks:

<Subjektas> [ne]PRIVALO/TURĖTŲ <faktas> [kol/jeigu <sąlyga>].

Čia subjektu gali būti terminas arba faktas [2]. Pagal tokį šabloną užrašytos VT pavyzdys galėtų būti toks: “draudimo agentas neturėtų sudarinėti draudimo sutarčių kol atostogauja” [15].

1.2.2 VT grindžiamas reikalavimų specifikuojimo metodas

Reikalavimų specifikuojimo procesas pradedamas veiklos konteksto apibrėžimu. Tuo pačiu identifikuojami ir asmenys, kurie yra potencialūs kuriamos IS vartotojai arba tiesiog gali būti naudingi renkant reikalavimus kuriamai IS. Vėliau yra nustatomos funkcijos, veiklos sprendimai, struktūrinės ir nestruktūrinės VT ir jų šaltiniai, o rezultatai saugomi reikalavimų saugykloje. Nestruktūrinių VT nustatymas atliekamas analizuojant sprendimus, kurie turi būti priimti vykdant konkrečias funkcijas [15].

1.2.3.1 Reikalavimų specifikavimo ir analizės stadijoje naudojami VT modeliai

Struktūrinės ir nestrukūrinės VT turi būti užrašomos vadovaujantis skirtingais principais. Struktūrinės VT, arba terminai ir faktai, nuo pat savo gyvavimo pradžios turi būti ruošiamos integravimui į duomenų modelį. Reikalavimų specifikavimo stadijoje tai būtų koncepcinis duomenų modelis. Panašių apribojimų nestrukūrinių VT modeliui nėra – jos tiesiog privalo būti pakankamai aiškiai suklasifikuotos pagal savo funkcines savybes, o užrašymo šablonai turi būti tokie, kad analitikas galėtų nedviprasmiškai užrašyti bet kurio tipo nestrukūrinės taisykles [15].

1.2.3.2 Struktūrinių VT fiksavimo forma

Fiksuojant nustatytus terminus ir faktus kaip koncepcinio duomenų modelio elementus būtina apibrėžti, kokia forma šis modelis bus užrašomas. Struktūrinėms VT fiksuoti dažnai naudojamas faktų modelis. Yra priimama, kad terminai ir faktai fiksuojami kaip koncepcinio duomenų modelio, išreiškiamo EER (išplėstine ER [8]) diagrama, elementai:

- *Terminai* – tai duomenų modelio elementų pavadinimai.
 - esybės;
 - atributai;
 - potipiai;
 - charakteristikos.
- *Faktai* – tai duomenų modelio sąryšiai.
 - esybių sąryšiai;
 - potipių sąryšiai (ISA);
 - atributų sąryšiai.

Dalykinėje srityje svarbūs, tačiau į koncepcinį duomenų modelį nepatenkantys terminai (pvz.: techniniai terminai, specifinės sąvokos) registruojami atskirai. Taip pat registruojamos ir dalykinėje srityje svarbios fiksuotos reikšmės.

Fiksuojant struktūrinės VT koncepciniame duomenų modelyje EER diagrama išsprendžiama jų grafinio atvaizdavimo problema. Vis dėlto, ne visada EER diagrama bus gerai suprantama kuriamos IS užsakovams ir vien tik jos gali neužtekti nustatytos duomenų struktūros teisingumui užtikrinti. Sprendžiant šią problemą patogiu naudoti tekstines terminų ir faktų išraiškas, kurios būtų vienareikšmiškai suprantamos, o tai savo ruožtu leistų garantuoti efektyvų reikalavimų verifikavimo ir validavimo procesą. Nesant galimybei EER diagramoje pavaizduoti esybių ir jų atributų

apibrėžtis, jų sinonimus, sąryšius tarp esybių ir pan., tokią informaciją galima pateikti vartotojui naudojant tam tikrus duomenų modelio elementų užrašymo natūraliąja kalba šablonus [15].

1.2.3.3 Nestruktūrinių VT modelis

VT struktūrizavimas reikalavimų specifikavimo stadijoje atliekamas naudojant BRS RuleSpeak VT šablonų modelį. BRS RuleSpeak šablonai (lietuvių kalba) su pavyzdžiais pateikiami 1 priede. Atsižvelgiant į tai, kokie duomenys kaupiami pagal reikalavimų specifikavimo procesą, naudojamas modifikuotas galimų šablono elementų tipų sąrašas. Galimi šablono elementų tipai ir jų paaiškinimai pateikti 1 lentelėje. Yra apibrėžta, kokio tipo elementai gali būti VT subjektais.

Svarbi modifikuoto BRS RuleSpeak metodo esmė - kaip pagal šabloną užrašyta VT yra susiejama su joje naudojamomis struktūrinėmis VT, t.y. terminais ir faktais arba koncepcinio duomenų modelio elementais [15].

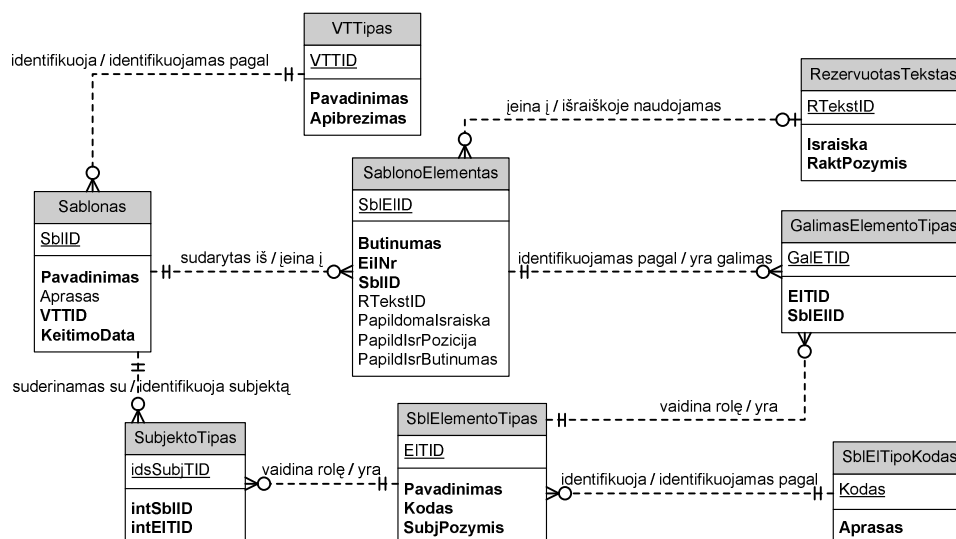
Lentelėje 1 pateikiami galimi modifikuoto BRS RuleSpeak modelio VT šablonų elementų tipai, o 3 paveikslėlyje pateikiamas nestruktūrinių VT šablonų loginis modelis.

Lentelė 1

Galimi modifikuoto *BRS RuleSpeak* modelio VT šablonų elementų tipai

Elemento tipas	Apibūdinimas	Atitikmuo <i>RuleSpeak</i>	Gali būti subjektu
Esybė	Koncepcinio duomenų modelio esybė.	Terminas	✓
Atributas	Koncepcinio duomenų modelio esybės atributas.	Terminas	✓
Ryšys	Koncepcinio duomenų modelio ryšys tarp esybių.	Faktas	✓
Reikšmė	Koncepcinio duomenų modelio atributo reikšmė, nepriklausoma reikšmė arba sprendimų lentelė.	Reikšmė	✓
Duomenų elementas	Tam tikras koncepcinio duomenų modelio esybe apibrėžto objekto vienetas.	<i>Toks pats</i>	✓
Funkcija	Veiklos funkcija.	Procedūra, procesas	✓
Raktažodis	Žodis ar žodžių junginys, identifikuojantis šabloną.	<i>Toks pats</i>	
Sąlyga	Laisva forma užrašyta sąlyga.	<i>Toks pats</i>	
Loginė išraiška	Tekstinė išraiška, kurioje naudojami loginiai operatoriai (pvz.: „daugiau“, „>“), sulyginimai.	<i>Toks pats</i>	
Matematinė formulė	Laisva forma užrašyta skaičiavimo formulė arba išraiška.	<i>Toks pats</i>	
Vaizdavimo terpė	Duomenų pateikimo vartotojui terpė (pvz.: „ekrane“, „ataskaitoje“).	<i>Toks pats</i>	
Vaizdavimo būdas	Duomenų pateikimo vartotojui būdas (pvz.: „raudonai“, „rikiuojant didėjimo tvarka“).	<i>Toks pats</i>	
VT pavadinimas	Veiklos taisyklės pavadinimas.	<i>Toks pats</i>	✓

Šaltinis: Kapočius, K. (2006) Disertacija "*Veiklos taisyklių struktūrizavimo modeliai ir jų taikymas kuriant informacijos sistemas*"



Šaltinis: Kapočius, K. (2006) Disertacija "Veiklos taisyklių struktūrizavimo modeliai ir jų taikymas kuriant informacijos sistemas"

3 pav. Nestruktūrinių VT šablonų loginis modelis

1.3 Alternatyvūs metodai

Kaip jau buvo pastebėta, reikalavimų analizės ir specifikavimo stadijoje yra labai patogu naudoti natūraliosios kalbos šablonus VT užrašinėti, nes natūralioji kalba yra lengvai suprantama tiek sistemos užsakovui, tiek analitikui. Kitaip tariant, specifikuojant reikalavimus natūraliosios kalbos šablonais garantuojamas maksimalus analitiko ir užsakovo sampratų apie būsimąją sistemą sutapimas. Galima teigti, kad kai turimos VT yra specifiкуotos pagal natūraliosios kalbos šablonus, jos jau yra formalizuotos. Vis tik pereinant prie projektavimo VT reikia suteikti formalią išraišką. Tam gali būti naudojamas, pavyzdžiui, grafinis VT formalizavimo Roso modelis, OCL [18], PRR [19], RuleML [25] ir pan.

Šiuo metu OMG grupė (Object Management Group) propaguoja keletą pakankamai naujų standartų, kurie galima sakyti yra alternatyva iki šiol naudojamiems metodams, skirtiems pradinėms IS kūrimo stadijoms. Tai yra semantikos veiklos žodynui ir veiklos taisyklėms standartas SBVR (*Semantics of Business Vocabulary and Business Rules*), atitinkantis specifikavimo natūraliosios kalbos šablonais metodą, nes naudoja vartotojui (IS užsakovui) suprantamą tekstinį pavidalą [20]. Jis leidžia aprašyti visas įmanomas taisykles, pavyzdžiui, duomenų modelį, veiklos taisykles ir t.t. Kitas propaguojamas standartas yra produkcinių veiklos taisyklių vaizdavimas PRR (*Production Rule Representation*) [19]. Jis naudojamas labai tikslioms specifikacijoms gauti projektavimo metu. Šis metodas nėra grafinis. Taigi, dvipakopio VT specifikavimo populiarinimas jau tampa visuotine tendencija, kurią remia pagrindinės IS kūrimo metodikas vystančios organizacijos.

Dvipakopis modeliavimo procesas įdomus tuo, kad yra artimesnis natūraliai reikalavimų raidai. Nebuvo rasta įrankių, kurie leistų dvipakopį VT specifikavimą atlikti viename įrankyje, t.y. šablonais (natūraliaja kalba) ir formaliu pavidalu. Kaip jau buvo minėta, šio darbo tikslas yra sukurti metodą, kuriame naudojami du konkretūs modeliai: VT formalizavimo natūraliosios kalbos šablonais (modifikuotas BRS RuelSpeak [13]) ir užrašymo formaliais modeliais (modifikuotas Roso metodas [3]). Šių modelių apjungimas yra svarbus IS kūrimo sėkmei užtikrinti.

1.4 Formali kalba VT projektavime, Roso metodas

Realizuoti IS galima tik turint formalų sistemos veikimo aprašymą. Vienas iš formalus VT užrašymo metodų yra pasiūlytas Ronaldo Roso (*Ronald Ross*) [21]. Terminai ir faktai yra fiksuojami duomenų modelyje.

1.4.1 Roso metodas

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. Kadangi R. Rosas labai daug dėmesio skyrė taisyklių tipų aprašymui, šis jo modelis yra ypač tikslus klasifikavimo požiūriu.

Nedalomos taisyklės yra skirstomos į septynias tipų grupes [21] (žr. 2 lentelę). 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.

Nedalomų taisyklių tipų grupės ir patys tipai paaiškinti 2 priede.

Lentelė 2.

Nedalomų taisyklių tipai pagal Roso metodą

I. Egzempliorių patvirtinimo	II. Tipo patvirtinimo	III. Pozicijos patvirtinimo	IV. Funkcinio patvirtinimo	V. Palyginamojo įvertinimo	VI. Matematinio įvertinimo	VII. Projekcijos Valdikliai
X privaloma <i>EK</i>	M bendra <i>TK</i>	POS pozicinė <i>IF</i>	FUNC funkcinė <i>EN</i>	EQ lygu <i>PV</i>	CALC apskaičiuota <i>REZ</i>	EA įgalinta <i>IEK</i>
LIM apribota <i>EK</i>	ME bendrai- atskirianti <i>TK</i>	LOW žemiausia <i>IF</i>	UQ unikali <i>EN</i>	NE nelygu <i>PV</i>	SUM susumuota <i>REZ</i>	COP nukopijuota <i>KEK</i>
	MI bendrai- susiejanti <i>TK</i>	HIGH aukščiausia <i>IF</i>	FLUC svyruojanti <i>EN</i>	GT daugiau-už <i>PV</i>	SUB atimta <i>REZ</i>	EX įvykdyta <i>VEK</i>
	MP bendrai- uždraudžianti <i>TK</i>	CHRO chronologinė <i>DL</i>	AS auganti <i>EN</i>	GE daugiau-arba- lygu <i>PV</i>	MULT sudauginta <i>REZ</i>	
		OLD seniausia <i>DL</i>	DE krentanti <i>EN</i>	LT mažiau-už <i>PV</i>	DIV padalinta <i>REZ</i>	
		NEW naujausia <i>DL</i>	NRE neatnaujinama <i>EN</i>	LE mažiau-arba- lygu <i>PV</i>	... <i>REZ</i>	

Pastaba: Nedalomų taisyklių lentelėje balti laukai atspindi bendrą atvejį konkrečioje taisyklių tipų grupėje.

Šaltinis: Kapočius, K. (2006) Disertacija "Veiklos taisyklių struktūrizavimo modeliai ir jų taikymas kuriant informacijos sistemas" (pakoreguota autoriaus)

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 išėities vertė.

Išvestinių taisyklių tipų grupės pateikiamos 3 priede.

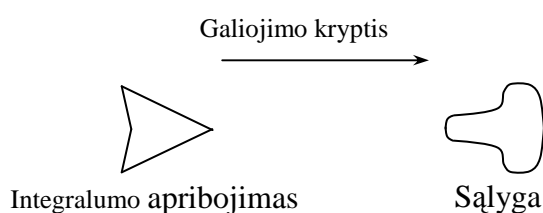
Be detalios taisyklių klasifikavimo sistemos, R.Roso metode yra pasiūlyta ir unikali VT modeliavimo užrašant jas grafiniu pavidalu metodika. Formaliai užrašant VT reikia remtis dalykinės srities duomenų modeliu. Konkretaus modelio naudojimas nėra nurodytas, tačiau yra apibrėžta, kokie duomenų tipai turi būti žinomi. Pagrindiniai jų yra esybės, atributai, asociacijos, sąryšiai, potipiai. Taisyklės teiginio užrašymas formaliu pavidalu susideda iš 6 pagrindinių etapų:

1. *Nustatymas, ar VT yra integralumo apribojimas, ar sąlyga.* Kitaip tariant, reikia nustatyti, ar taisyklė reikalauja vykdymo, ar yra išreikšta tik kaip patikrinimas.

Integralumo apribojimas – tai taisyklė, kurios rezultatas visada privalo būti loginis vienetas (loginė tiesa). Kadangi tokios taisyklės neturi duoti neigiamo rezultato, todėl **privalo** būti vykdomos.

Sąlyga – tai taisyklė, kurios rezultatas gali būti arba loginis vienetas (loginė tiesa), arba loginis nulis (loginis melas). Rezultatas taip pat gali būti ir nežinomas. Šios taisyklės naudingos, kuomet reikia nustatyti, kokias taisykles vykdyti arba kokius testavimus atlikti (t.y., kokias kitas taisykles iškviešti) atsižvelgiant į tam tikrą situaciją. Tokias taisykles galima išdėstyti nuosekliai, nurodant, kas turi vykti, **jeigu** aukštesnio lygio taisyklė yra tenkinama.

Modeliuojant taisykles apribojimams ir sąlygoms atskirti naudojami specialūs žymėjimai (4 paveikslėlis). Simboliai gali būti sukami į bet kurią pusę, tačiau būtina nepamiršti jų galiojimo krypties.



Šaltinis: Butleris, R., Kapočius, K. (2001). *Struktūrizuotų veiklos taisyklių saugyklos architektūra*. Informacijos mokslai. Vilnius. Vilniaus universiteto leidykla, 17 tomas, 46-57.

4 pav. Taisyklės vykdymo ir taisyklės tikrinimo žymėjimas

2. *VT bazės nustatymas*. Kitaip tariant, reikia nustatyti, kuriam objektui, vadinamam *base*, reikėtų “priskirti” nagrinėjamą VT. Dažniausiai bazė – tai duomenų modelio objektas (esybė, atributas ir pan.), tačiau tai gali būti ir kita VT ar jos išėigos reikšmė. Taisyklė gali būti traktuojama kaip jos bazės savybė.

Bazę privalo turėti kiekviena VT, nes tai yra svarbiausias elementas interpretuojant taisyklę. Bazės egzemplioriai gali būti įtraukti ir į taisyklės atliekamą patikrinimą. Toks patikrinimas atliekamas su kiekvienu bazės egzemplioriumi.

3. *Korespondento(-ų) nustatymas*. Visos VT be bazės paprastai apima dar bent vieną duomenų modelio objekto egzempliorių. Tokie objektai vadinami *korespondentais* ir jie yra būtini kiekvienoje taisyklėje. Korespondentai visuomet yra svarbūs atliekant taisyklės tipo apibrėžtą testą. Korespondentu gali būti ne tik duomenų modelio objektas, bet ir kita taisyklė, jos išėigos reikšmė arba veiksmas.

Kiekviena taisyklė turi vieną ar daugiau korespondentų. Yra išskirtos tokios ypatybės:

– Nedalomos taisyklės, priklausančios *tipo patvirtinimo* tipų grupei, privalo turėti du arba daugiau korespondentų, nes veikia kaip loginiai operatoriai IR ar ARBA.

- Nedalomos taisyklės, priklausančios *matematinio įvertinimo* tipų grupei, gali turėti daugiau negu vieną korespondentą.
- Likusioms tipų grupėms priklausančios nedalomos taisyklės gali turėti tik vieną ir ne daugiau korespondentų.

Tiek išeinantis iš bazės ryšys, tiek ir einantys į korespondentus ryšiai VT diagramoje visada vaizduojami punktyrine rodykle.

4. Taisyklių tipų parinkimas. Parinkti taisyklės tipą (arba nustatyti, kokius testus ji atlieka) yra būtina, nes tik esant nurodytam tipui galima teisingai interpretuoti taisyklę. Taisyklė gali priklausyti tiek nedalomų VT, tiek išvestinių VT grupės tipui. Visi taisyklių tipai tinka tiek integralumo apribojimams, tiek ir sąlygoms. VT diagramoje tipo santrumpa įrašoma apribojimo arba sąlygos simbolio viduje.

5. Bazės ir korespondento(-ų) susiejimas. Dažnai svarbu išskirti ir teisingą sąryšį tarp bazės ir korespondento. Šis sąryšis privalo sekti iš duomenų modelio. Galima priimti, jog duomenų modelis yra sudarytas iš *terminų* ir *faktų* apie probleminę sritį. Faktai gali būti šių kategorijų:

- Sąryšio tipai, pvz.: “klientas pateikia užsakymą”
- Potipių sąryšiai, pvz.: “automobilis yra prekė”
- Atributų sąryšiai, pvz.: “automobilis turi registracijos numerį”

Svarbu pabrėžti, jog VT niekuomet negali įvesti naujo fakto (sąryšio). Visi svarbūs faktai (sąryšiai) turi būti atspindėti duomenų modelyje.

Taigi, kiekvienas sąryšis tarp taisyklės bazės ir korespondento(-ų) turi būti paimtas iš duomenų modelio. Būtina išskirti ir kitą prielaidą: kiekvienas pavaizduotas sąryšis (fakto tipas) tarp taisyklės bazės ir korespondento(-ų) yra laikomas svarbiu taisyklės interpretacijai. Taigi, nesvarbūs sąryšiai į diagramą neturi būti įtraukiami.

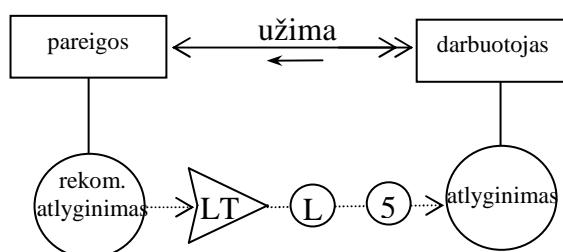
6. Papildomų VT modelio ypatybių nustatymas. Roso notacijoje yra numatyta galimybė naudoti papildomus simbolius, galinčius pakeisti standartinę tam tikro tipo taisyklės interpretaciją. Šie simboliai skirstomi į dvi pagrindines grupes:

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

Be specialiųjų simbolių dar gali būti vartojamos konstantos, taisyklių išėigos reikšmės ir kitos papildomos priemonės [21]. Paprastai taisyklės išėigos reikšmė yra “nematoma” ir vartojama taisyklės viduje tam tikram testavimui atlikti arba apribojimui pritaikyti. Tačiau tam tikrose taisyklėse reikia tiesiogiai tikrinti kitų taisyklių išėigos reikšmes (tai labiau būdinga

sąlygos tipo taisyklėms). Tuomet VT išeigos reikšmė atspindės ir taisyklės, vartojančios šią reikšmę, diagramoje. Kaip jau buvo minėta, taisyklės išeigos reikšmė gali būti kitos taisyklės baze arba korespondentu.

Paprastos taisyklės pavyzdys pateiktas 5 paveikslėlyje. Ji apibrėžia tokį apribojimą: *“Rekomenduojamas pareigų atlyginimas turi būti mažesnis už bent penkių darbuotojų, užimančių šias pareigas, atlyginimą”*. Šios taisyklės bazė yra 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 LT). Diagramoje taip pat naudojami ir du apribojimai: žemesniojo slenksčio apribojimas (simbolizuoja rutuliukas su raide L) ir numeratorius (simbolizuoja rutuliukas su skaičiumi 5).



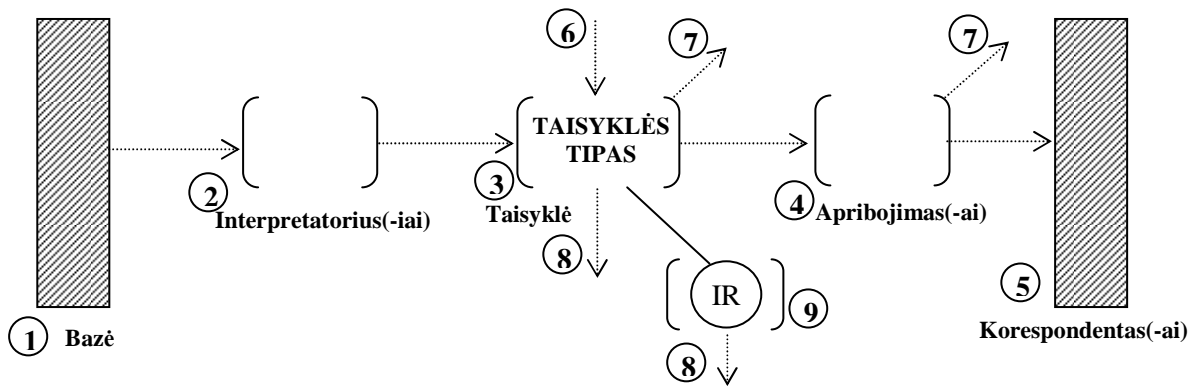
Šaltinis: Butleris, R., Kapočius, K. (2001). *Struktūrizuotų veiklos taisyklių saugyklos architektūra*. Informacijos mokslai. Vilnius. Vilniaus universiteto leidykla, 17 tomas, 46-57.

5 pav. Veiklos taisyklės grafinio modelio iliustracija

Taigi, galima teigti, jog Roso metodas leidžia atlikti išsamų VT struktūrizavimą pagal tipus. Roso VT modeliavimo metodika užtikrina vienareikšmišką VT susiejimą su duomenų modelio elementais, apribojamais VT ar tiesiog svarbiais jos interpretacijai. Be to, Roso VT modeliai leidžia vizualizuoti VT ir duomenų modelio sąsają [15].

1.4.2 Veiklos taisyklių formalizavimas pagal Roso metodą

VT formalizavimą geriausia pradėti iškart po jos patvirtinimo. Tam naudojama Roso modelyje apibrėžta taisyklių užrašymo specialių diagramų pavidalu metodika. Pagal Roso metodą sumodeliuotos VT pasižymi struktūra, pavaizduota 6 paveikslėlyje.



Šaltinis: Kapočius, K. (2006) Disertacija "Veiklos taisyklių struktūrizavimo modeliai ir jų taikymas kuriant informacijos sistemas"

6 pav. Veiklos taisyklės formalizavimo pagal Roso metodą sintaksė (pagal [21])

Veiklos taisyklės diagramos sudedamosios dalys [21]:

1. *Privaloma*. Gali būti: duomenų modelio tipas, į duomenų modelyje neapibrėžta konstanta, kita VT arba kitos VT išeiigos 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 išeiigos reikšmė.
6. *Nebūtina*. Vaizduoja įeiigos taisyklę.
7. *Leistina daugiau negu vienas*. Taisyklė gali turėti daugiau negu vieną korespondentą.
8. *Nebūtina*. Vaizduoja išeiigos taisyklę.
9. *Nebūtina*. Vaizduoja taisyklės išeiigos 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 specifikuojimo 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 neleisti 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.*
 - Triggeris 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 nereglamentuojama, kaip ar kada programiniu požiūriu jos turėtų būti vykdomos. VT reglamentuoja pačią veiklą ir tai, kas svarbu šios veiklos dalyviams [15].

1.4.3 Pagal Roso metodą formalizuotų veiklos taisyklių saugyklos modelis

VT saugyklos loginės struktūros modeliui buvo iškelti toki baziniai reikalavimai:

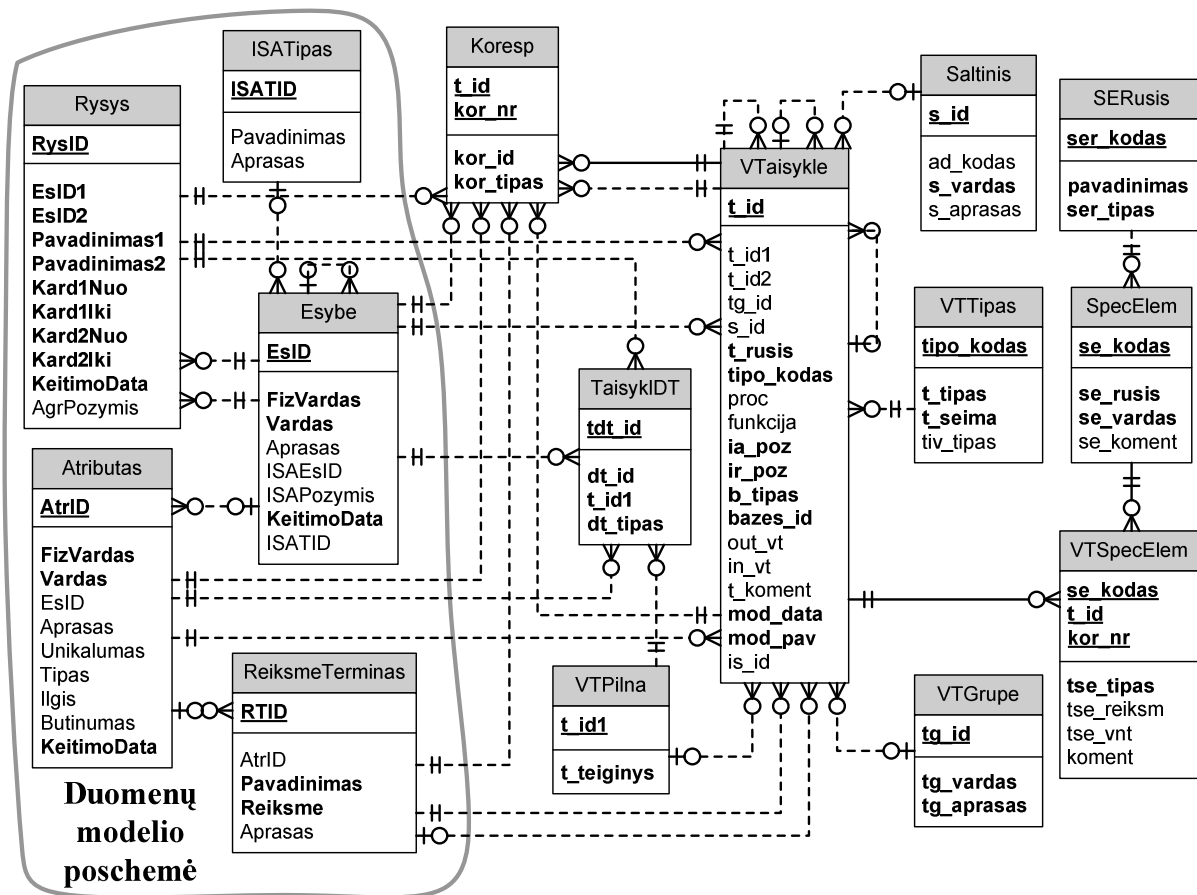
1. Informacijos išsaugojimo lygis turi būti toks, kad būtų galima į duomenų bazę įvesti duomenis apie bet kokio tipo ar struktūros VT, formalizuotą pagal Roso metodą.
2. Saugyklos duomenų bazėje saugomos informacijos apie VT turi pakakti, norint visiškai atkurti jos diagramą.
3. Aprašant VT diagramą 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, turimas saugyklos loginis modelis [12]. Saugykloje saugoma tik VT diagramoje atsispindinti informacija. Laikomasi prielaidos, kad jog informacija apie kiekvieno taisyklės tipo interpretaciją bus užkoduota tiesiogiai VT interpretavimo sistemoje [6].

Modelyje naudojama duomenų modelio elementų saugojimo poschemė. Be duomenų modelio elementų išsaugojama ir reikšmių lentelė `ReiksmeTerminas`, kurioje 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`, `Rysys`, `Atributas`, `ReiksmeTerminas`. Kiekviena veiklos taisyklė `VTaisykle`, turi bazę ir vieną ar daugiau korespondentų `Korespondentas`. Korespondentu, kaip ir baze, gali būti duomenų modelio elementas, konstanta kita VT `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 VT diagramoje atsispindinčios informacijos yra saugomi ir papildomi duomenys apie taisyklės kilmės šaltinį `Saltinis`, be to, taisyklės gali būti grupuojamos į įvairaus pobūdžio grupes `VTGrupe` [15].



Šaltinis: Kapočius, K. (2006) Veiklos taisyklių struktūrizavimo modeliai ir jų taikymas kuriant informacijos sistemas

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

1.4 Apibendrinimas

Veiklos taisyklės apibrėžia sistemos struktūros ribojimus, sistemos objektų elgseną, aktorių kompetencijos ribas ir pareigas. Pradiniai sistemų panaudojimo atvejai, taisyklės, vaizduojančios vartotojų poreikius, reikalavimų specifikavimo stadijoje yra užrašomos ir specifikuojamos (išsaugomos) panaudojant natūraliosios kalbos šablonus. Jos yra užrašomos natūraliais sakiniais, o tokia forma nėra visai formali, tačiau formalizuota, nes šablonų dėka turima vientisa sistema, kurioje taisyklės galima rūšiuoti pagal jų subjektą, pagal tipą ir panašiai.

Pabaigus reikalavimų specifikavimą ir perėjus prie projektavimo, veiklos taisyklės užrašomos formaliais būdais [16]. Tarkim panaudojant specializuotas kalbas (pseudokodą, OCL [18], labai tikslius šablonus ir pan.) arba braižant grafinius modelius. Šiame etape yra svarbu sukurti tokias VT išraiškas, iš kurių būtų galima generuoti programų kodą ar pagal kurias programuotojai galėtų vienareikšmiškai interpretuoti reikalavimus.

Struktūrizuojant VT IS kūrimo metu, jos turėtų įgyti dvi minėtas formas: formalizuotų natūraliosios kalbos šablonų bei formalių metodų. Taisyklės išreiškiant formalizuotais natūraliosios

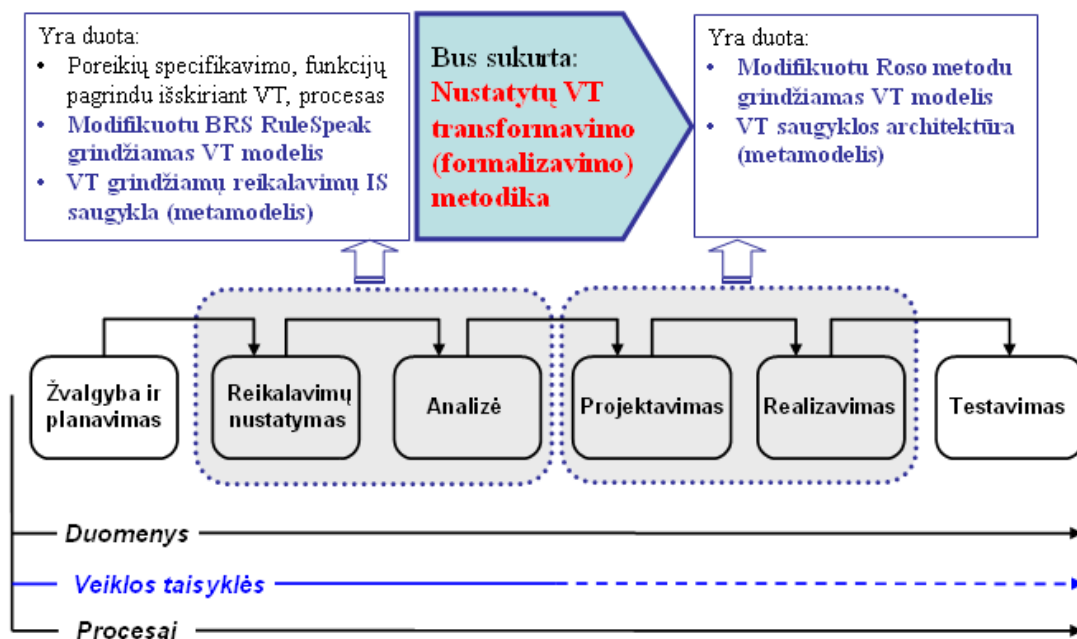
kalbos šablonais, jos yra užrašomos lengvai suprantama forma. Reikalavimų IS analizės ir specifikavimo metu surinktos taisyklės struktūrizuojamos tikrai pagal sąryšį su struktūrinės taisyklės atspindinčio koncepcinio duomenų modelio elementais.

Taisyklių išraiškos formaliais metodais yra tiesiogiai paveldimos iš pirminio jų varianto, t.y. užrašytų natūraliosios kalbos šablonais. Jos yra suklasifikuojamos pagal griežtą klasifikaciją ir užrašomos formaliai. Tokiu būdu jos tampa sunkiai suprantamos ne tos srities specialistui ir yra pritaikytos tolimesnių IS kūrimo etapų specialistams – projektuotojams, programuotojams. Formaliai išreikštos taisyklės yra saugomos VT saugykloje ir yra tinkamos IS realizavimui. Turėtų būti užtikrintas perėjimas tarp šių dviejų IS kūrimo stadijų, kad nebūtų prarandama informacija, gauta analizuojant ir specifikuojant taisykles.

Reikalavimų specifikavimas pradedamas veiklos apibrėžimu. Taip pat nustatomi galimi kuriamos IS vartotojai arba asmenys, kurie galėtų būti naudingi renkant reikalavimus kuriamai IS. Vėliau nustatomos funkcijos, veiklos sprendimai, struktūrinės ir nestructūrinės VT ir jų šaltiniai, o rezultatai saugomi reikalavimų saugykloje. Kaip jau minėta, prieš patekdamos į saugyklą, VT yra suklasifikuojamos. Taigi reikalavimų saugykloje galima išskirti dvi veiklos taisyklių poschemes (žinoma jų yra daugiau) – struktūrinių ir nestructūrinių VT.

Nagrinėjant duotus modelius, buvo išvelgta panašumų tarp modifikuoto BRS RuleSpeak ir modifikuoto Roso metodo. Todėl galima teigti, jog bent dalis perėjimo nuo reikalavimų specifikavimo į projektavimą (veiklos taisyklių modelių transformavimo) proceso automatizavimas yra įmanomas, o tai paspartintų visos IS kūrimą. Automatiškai projektuotojui pateikiant kiek įmanoma išsamesnius veiklos taisyklių modelių ruošinius pagal Roso metodą, būtų taupomas laikas ir didinamas darbo efektyvumas. Be to, būtų minimizuojama klaidų, galimų perkeliant reikalavimus iš natūralios kalbos į formalų lygmenį tikimybė.

Tyrimo objektų vietą IS kūrimo procese ir sprendimus galima pavaizduoti shcema pateikta 8 paveikslėlyje.



Šaltinis: sudaryta autoriaus pagal Kapočius, K., Butleris, R. (2003). Reikalavimų kuriamai IS nustatymo ir analizės ypatumai veiklos taisyklių metodikoje. *Informacinės technologijos '2003. Konferencijos pranešimų medžiaga*. Kaunas. Technologija, XIV-74 – XVI-81

8 pav. Tyrimo objektų vieta IS kūrimo procese

2. VT TRANSFORMAVIMO IŠ BRS RULESPEAK Į ROSO METODĄ KONCEPCIJA

Transformuojant VT iš BRS RuleSpeak į Roso metodu užrašytas taisykles svarbu daryti viską sistemingai. Reikia nustatyti kuo daugiau elementų, kuriuos būtų galima perkelti iš vienos saugyklos į kitą ir atlikti elementų susiejimą. Nagrinėjant BRS RuleSpeak šablonus ir Roso metodu užrašomas taisykles, buvo bandoma nustatyti kokie reikalavimų specifikavimo duomenų saugyklos esantys elementai gali būti atvaizduojami kokiais Roso saugyklos elementais. Kad tai nustatyti, reikia panagrinėti keleta pavyzdžių.

Yra žinoma, kad taisyklėse naudojamos sąlygos yra susietos su terminais ir faktais (duomenų modelio elementais), taip pat ir konstantomis, kurie naudojami tose sąlygose. Tuo remiantis galima teigti, kad jeigu taisyklėje yra sąlyga, tai į ekraną išvedus sąlygos elementą, reikia automatiškai išvesti tuos duomenų modelio elementus, su kuriais ta sąlyga yra susieta. Visus tuos elementus projektuotojas panaudotų braižydamas taisyklės modelį. Sąlygos susiejimas su jai priklausančiais duomenų modelio elementais yra atliekamas užrašant taisykles natūraliosios kalbos šablonais BRS RuleSpeak pagal KTU Informacijos sistemų katedroje sukurtą metodiką.

2.1 Veiklos taisyklių transformavimo prielaidos

Norint kuo tiksliau nustatyti VT užrašymo natūraliosios kalbos šablonais ir Roso metodu panašumus bei sukurti transformavimo metodiką, reikia detaliau panagrinėti perėjimo iš vienos stadijos į kitą metu gaunamus rezultatus. Nustatyti bendrus bruožus ir išskirti ypatumus.

2.1.1 Perėjimo procesas

Tarkim, turime taisyklę, užrašytą pagal BRS RuleSpeak 1.1 atmetimo/apribojimo tipo šabloną *b1*. <Subjektas> TURI|NETURI|TURĖTŪ|NETURĖTŪ <faktas> [(, jeigu |, kol) <sąlyga>]:

Draudimo sutartis turi turėti unikalų numerį.

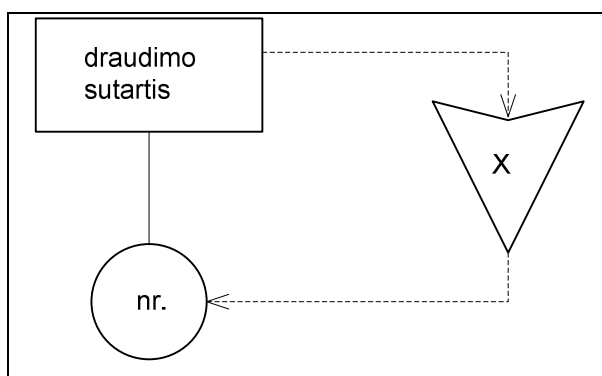
Transformuojant šią taisyklę į Roso metodu užrašytą taisyklę, subjektas turi tapti baze, korespondentu – taisyklės faktas. Taisyklės tipas turi būti nustatomas pagal raktinį žodį. Pagal tai ar pagal šabloną užrašytoje taisyklėje yra naudojama sąlyga ar ne, turėtų būti nustatomas Roso metodu užrašomos taisyklės pobūdis, t.y. sąlyga ar integralumo apribojimas. Šiuo atveju taisyklėje jokios

sąlygos nėra, tad į ekraną turėtų būti išvedamas integralumo apribojimo simbolis kuriame turės būti pažymėtas taisyklės tipas. Pagal raktinį taisyklės žodį TURI|NETURI|TURĖTŪ|NETURĖTŪ galima teigti, kad taisyklė galėtų būti egzempliorių patvirtinimo grupės viena iš taisyklių: privaloma (X) arba apribota (LIM).

Pagal tai, koks duomenų modelio elementas yra taisyklės subjektas, kuris bus vaizduojamas kaip bazė Roso metodu, į ekraną turi būti išvedamas atitinkamas simbolis: duomenų tipas, konstanta, kita VT ar kitos VT išeišgos reikšmė.

Šiuo atveju subjektas, tampantis baze, yra duomenų modelio esybė draudimo sutartis. VT taisyklės faktas <turėti unikalų numerį> pagal šabloną yra susietas su duomenų modelio esybės draudimo sutartis atributu nr, kuris ir tampa korespondentu Roso metodu užrašomoje taisyklėje. Nustatoma, kad taisyklė yra privaloma, tad apribojimo simboliuje įrašoma X.

Roso metodu sumodeliuotos taisyklės pavyzdys 9 paveikslėlyje.



Šaltinis: sudaryta autoriaus

9 pav. Apribojimo/atmetimo tipo taisyklė (šablonas b1) Roso metodu

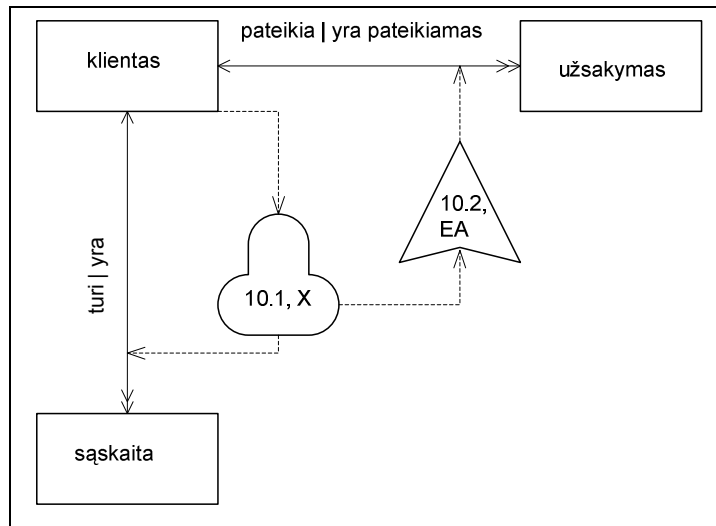
Jeigu turima taisyklė, kurioje yra tikrinama kažkokia sąlyga, tampa aišku, kad ir taisyklės užrašymo schemeje reikės atvaizduoti ir sąlygos, o ne tik integralumo apribojimo simbolį, t.y. pilnas taisyklės modelis pagal Roso notaciją susidės iš dviejų VT.

Pavyzdžiui, turime taisyklę, užrašyta pagal šabloną b2. <Subjektas> gali | turėtų <faktas> TIK jeigu | kol <sąlyga>:

Klientas gali pateikti užsakymą tik jeigu klientas turi atsidaręs sąskaitą internete.

Subjektas, šiuo atveju duomenų modelio esybė klientas, kaip ir pirmame pavyzdyje tampa baze, korespondentu nustatomas faktas <pateikti užsakymą> – ryšys tarp duomenų modelio esybių klientas ir užsakymas. Į ekraną turėtų būti išvedami atitinkami duomenų modelio elementai: esybės klientas ir užsakymas bei jas jungiantis ryšys pateikia, taip pat sąlygos ir integralumo apribojimo simboliai.

Roso metodu sumodeliuotos taisyklės pavyzdys 10 paveikslėlyje.



Šaltinis: sudaryta autoriaus

10 pav. Apribojimo/atmetimo tipo taisyklė (šablonas b2) Roso metodu

Taisyklių tipų grupės, kurios galėtų būti pasiūlomos projektuotojui, bus nagrinėjamos 2.2.5 skyriuje.

Šiuo atveju sąlygos taisyklė yra privaloma X, nes tik kai tenkinama nurodyta sąlyga, yra paleidžiama vykdymui įgalinimo taisyklė (EA), kuri priklauso projekcijos valdiklių tipų grupei.

Tarkime, turime taisyklę, užrašytą pagal BRS RuleSpeak leidimo tipo šabloną *b5* .
 <Subjektas> NETURI <faktas> | <VT raktažodis> [(, jeigu |, kol) <sąlyga>]:

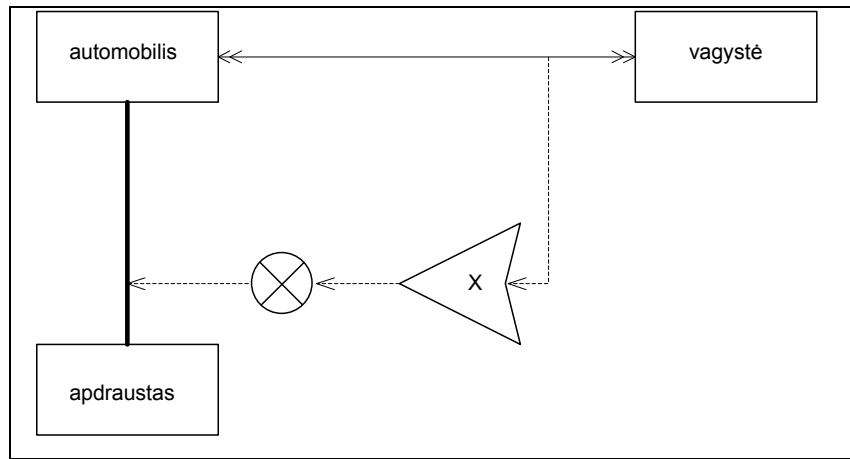
Automobilis neturi būti apdraustas nuo vagystės.

Subjektas duomenų modelio elementas esybė *automobilis* tampa baze, faktas <*būti apdraustas*>, kuris šiuo atveju yra duomenų modelio elementas ISA tipas *apdraustas*, tampa korespondentu. Pagal tai, kad taisyklėje nėra tikrinama sąlyga, į ekraną turėtų būti išvedamas tik apribojimo simbolis, kuris šiuo atveju įgyja reikšmę privaloma (X). Pagal taisyklės raktinį žodį, daroma išvada, kad Roso metodu užrašomoje taisyklėje turės būti panaudojamas specialusis elementas, leidžiantis kitaip interpretuoti taisyklę nei nusako jos tipas.

Specialiųjų elementų panaudojimas pagal raktinius žodžius bus aptartas 2.2.6 šio darbo skyriuje.

Šiuo atveju, kai taisyklėje yra neigimą reiškiantis raktinis žodis NETURI, išvedant taisyklės duomenų modelio elementus, turėtų būti pasiūlomas ir paneigimą išreiškiantis simbolis (*angl. false*).

Taisyklę iliustruoja 11 paveikslėlis.



Šaltinis: sudaryta autoriaus

11 pav. Leidimo tipo taisyklė (šablonas b5) Roso metodu

Kad būtų galima kurti kuo tikslesnę VT transformavimo metodiką, reikia panagrinėti daugiau BRS RuleSpeak šablonų nustatyti visoms taisyklėms (kiek tai yra įmanoma) bendrus bruožus.

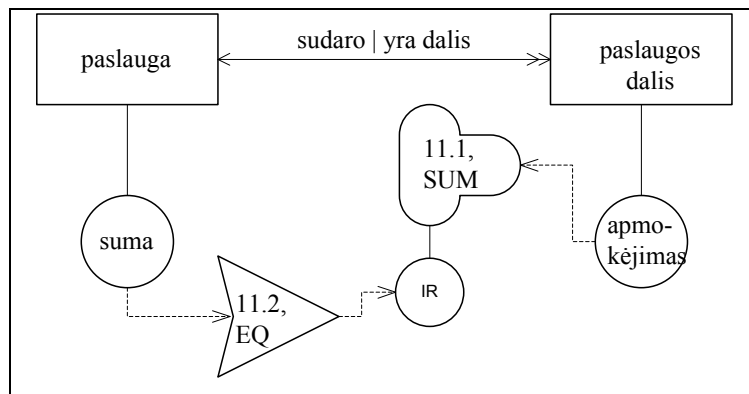
Nagrinėjame generavimo taisykles, kurios yra išskirtos į skaičiavimo ir išvedimo.

Turime skaičiavimo tipo taisyklę, užrašytą pagal b6. <Subjektas> turi|neturi|turėtų|neturėtų BŪTI APSKAIČIUOJAMAS|-a|-i|-os kaip <matematinė formulė> [(, jeigu |, kol) <sąlyga>]:

Apmokėjimo už paslaugą suma apskaičiuojama kaip visų paslaugai priklausančių dalių apmokėjimų suma.

Taisyklės subjektas duomenų modelio esybės paslauga atributas suma tampa baze. Matematinėje formulėje, kuri šiuo atveju užrašyta kaip <visų paslaugai priklausančių dalių apmokėjimų suma>, naudojamas duomenų modelio esybės paslaugos dalis atributas apmokėjimas, tampa korespondentu. Taisyklė yra apribojimas susumuota (SUM), kuri priklauso matematinio įvertinimo tipų grupei.

Taisyklė užrašyta Roso metodu atrodytų pavaizduota 12 paveikslėlis.



Šaltinis: sudaryta autoriaus

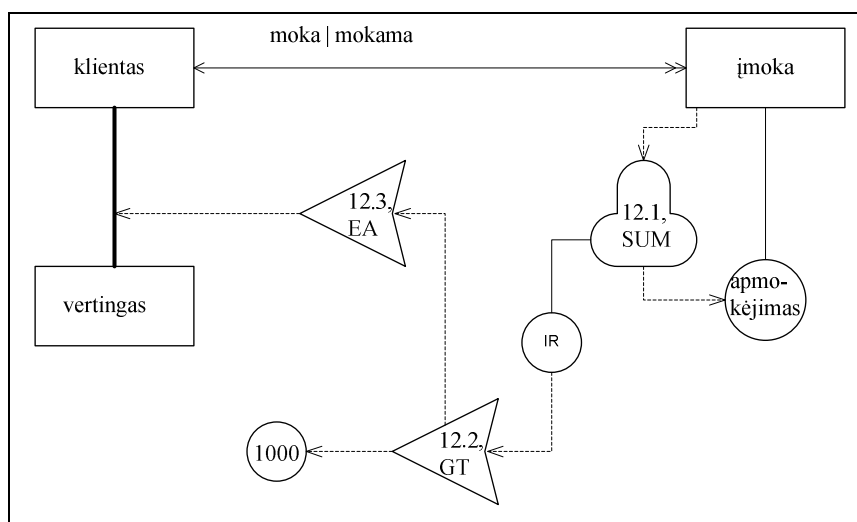
12 pav. Skaičiavimo tipo taisyklė (šablonas b6) Roso metodu

Turime išvedimo taisyklę, kuri užrašyta pagal b9.<Subjektas> REIŠKIA, KAD | NEREIŠKIA, KAD <loginė išraiška> [(, jeigu |, kol) <sąlyga>]:

Vertingas klientas reiškia, kad kliento metinių įmokų suma viršija 1000 Lt.

Taisyklės subjektu gali būti išvestinis terminas (atributas, esybė, reikšmė). Šiuo atveju taisyklė realizuojama dviem VT. Viena iš jų skaičiuoja kliento metinių įmokų sumą, kita naudoja tos taisyklės išėigą (apskaičiuotą rezultatą) ir tikrina ar ji viršija nurodytą minimalią sumą, t.y. 1000 Lt. Tokiu atveju vienoje taisyklėje baze yra duomenų modelio esybės įmoka atributas apmokėjimas, korespondentu tampa tos taisyklės išėiga (apskaičiuota reikšmė), taisyklės tipas yra matematinio įvertinimo (suma). Kitos taisyklės bazė yra pirmosios taisyklės išėiga, korespondentas yra ISA ryšys tarp duomenų modelio elemento klientas ir jo subtipo vertingas.

Sumodeliavus Roso metodu taisyklę atrodytų kaip pavaizduota 13 paveikslėlis.



Šaltinis: sudaryta autoriaus

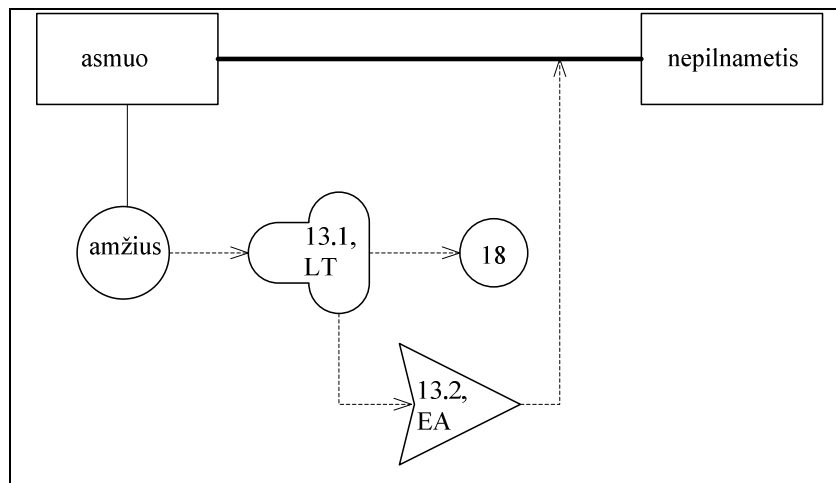
13 pav. Išvedimo tipo taisyklė (šablonas b9) Roso metodu

Šiuo atveju pagal BRS RuleSpeak šabloną užrašytos taisyklės subjektas tapo ne Roso metodu sumodeliuotos taisyklės baze, o korespondentu. Todėl galima daryti prielaidą, kad ne visada subjektas tampa baze.

Dar vienas nagrinėjamas atvejis - VT, užrašyta pagal b10. <Subjektas> turi|neturi|turėtų|neturėtų BŪTI LAIKOMAS|-a|-i|-os <terminas> (, jeigu |, kol) <sąlyga>:

Asmuo turėtų būti laikomas nepilnamečiu, jeigu asmens amžius yra mažiau 18-os metų.

Šiame šablone subjektu gali būti terminas ar duomenų elementas. Čia subjektu yra duomenų modelio elementas asmuo, terminas – nepilnametis. Modeluojant taisyklę Roso metodu, subjektu tampa duomenų modelio esybės asmuo atributas amžius, korespondentu tampa ISA ryšys tarp esybės asmuo ir jos subtipo nepilnametis. Taisyklės tipas Roso metode parenkamas palyginamojo įvertinimo mažiau-už (LT). Sumodeliuota taisyklė pavaizduota 14 paveikslėlis.



Šaltinis: sudaryta autoriaus

14 pav. Išvados tipo taisyklė (šablonas b10) Roso metodu

Šiuo atveju pats subjektas netapo baze Roso metode, o terminas netapo korespondentu. Tačiau baze tapo tai, kas buvo naudojama taisyklės sąlygoje, t.y. esybės `asmuo` atributas `amžius`. Korespondentu tapo ne tiesiogiai pats terminas `nepilnametis`, o ryšys tarp jo ir esybės `asmuo`.

2.1.2 Išimtiniai atvejai

Roso metodu nebraižomos VT

Ne visos taisyklės, užrašytos BRS RuleSpeak šablonais, gali būti modeliuojamos Roso metodu. Ši išvada seka iš to, kad Roso metodas yra skirtas aprašinėti toms VT, kurios užduoda paribojimus duomenų modelio elementams. Tokios specifikos, kaip vaizdavimo būdas „*pateikiamas raudonai*“ ar vaizdavimo terpė „*pateikiamas ekrane*“, procesas „*įspėjimo dėl susitikimo siuntimas*“ negali būti aprašomi Roso metodu – nėra tokios galimybės. Tokio tipo taisyklės, kurioms neįmanoma pritaikyti modeliavimo pagal Roso metodą, būtų šios:

3.1.3 Proceso jungiklis: šablonas *b14*;

3.2.2 Pateikimo (atvaizdavimo) taisyklė: šablonas *b17*;

3.3.1 Proceso trigeris: šablonas *b18*;

Aptikus šių tipų taisykles, jos galėtų būti transformuotos į papildomą informaciją, rekomendacijas dėl panaudojimo kituose, ne veiklos taisyklių modeliuose, tokiuose kaip vartotojo sąsajos specifikacijos, procesų specifikacijos, galbūt veiklos modeliuose (ypač jeigu subjektas yra procesas) priklaunomai nuo pasirinkto projektavimo metodo (pvz. objektinis, UML).

Taigi jeigu turima taisyklė yra funkcija (pagal BRS RuleSpeak tai gali būti procesas, procedūra), ji turi būti perduodama projektuotojui, kuris specifikuoja būtent tą funkciją.

Roso modelyje nėra galimybės sumodeliuoti tam tikrų duomenų sukūrimo. Todėl turint taisyklę pagal BRS RuleSpeak 3.1.4 Duomenų jungiklis: šablonas *b15*, jos taip pat Roso metodu nemodeliuojame. Taisyklės modeliavimas turėtų būti perduotas kitiems projektuotojams kaip ir aukščiau aptartos taisyklių išimtys.

Ypatingos VT, naudojančios jau esamas VT

Yra keletas BRS RuleSpeak taisyklių tipų, kuriose subjektas visada yra kita taisyklė. Galima išskirti šiuos tipus:

3.1.2 Taisyklės jungiklis: šablonai *b12*, *b13*,

3.3.2 Taisyklės trigeris: šablonas *b19*.

Pagal šiuos šablonus užrašytos taisyklės ypatingos tuo, kad jose naudojama kita taisyklė, kuri jau yra žinoma. Jeigu taisyklės, naudojamos šablone kaip subjektas, schema dar nėra sugeneruota, t.y. dar nėra saugoma saugykloje, tai turėtų būti atlikta pirmiausiai. Bet laikomasi principo, kad taisyklė-subjektas jau yra sumodeliuota. Tokiu atveju, į ekraną turėtų būti išvedama taisyklės-subjekto schema ir sąlygos tipo elementas, nes kaip jau buvo minėta anksčiau, šio tipo taisyklėse (taisyklės jungiklis) sąlyga yra privaloma. Į ekraną turi būti išvedami tie modelio elementai, kurie yra susieti su užrašyta sąlyga. Jeigu sąlygoje naudojami duomenų modelio elementai jau yra išvesti į ekraną (naudojami taisyklėje-subjekte), tokiu atveju antrą kartą jie nėra išvedami.

Jeigu taisyklėje-subjekte yra tik viena atominė Roso taisyklė, galima realizuoti sąlygos elemento susiejimą (punktyrine linija) su tos taisyklės simboliu. Jeigu taisyklėje-subjekte yra keletas taisyklių, sujungtų į logiškai nedalomą, tokiu atveju sąlygos elementas turėtų būti susiejamas su paskutiniu pagal loginę seką taisyklės simboliu. Bet kuriuo atveju projektuotojas visada gali pakeisti, jeigu jo manymu tai yra neteisinga.

Atliekant taisyklių generavimą pagal šiuos (*b12* ir *b13*) šablonus, pasikeičia taisyklės-subjekto taisyklės(-ių) simbolių numeracija. Pirmasis indeksuoja naują VT, antrasis – jos dalį. Jeigu taisyklė-subjektas turėjo tik vieną taisyklės simbolį ir jo numeris buvo pvz. 13.0, tai atliekant naujos taisyklės modeliavimą, šis numeris turėtų tapti pvz. 16.2, o naujos taisyklės sąlygos simbolis, bus numeruojamas kaip 16.1, tokiu būdu įgalinant taisyklės-subjekto „įjungimą“ ar „išjungimą“ pagal užduotą sąlygą. Pakitusi numeracija yra išsaugoma naujos VT sudėtyje. Keičiantis naudojamoms pirmosios VT struktūrai, generuojamoje VT ji liks nepakitus. Todėl atliekant modifikacijas pirmesnėje taisyklėje, reikia atlikti pakeitimus ir kitoje, galbūt sugeneruoti iš naujo.

Tokios pat VT transformavimo sąlygos galioja ir pagal BRS RuleSpeak šabloną *b19* modeliuojamoms Roso metodu taisyklėms. Pagal šį tipą užrašyta taisyklė-subjektas yra paleidžiama (ar sustabdoma – priklausomai nuo raktinio žodžio) tik kai yra tenkinama nurodyta sąlyga. Taigi

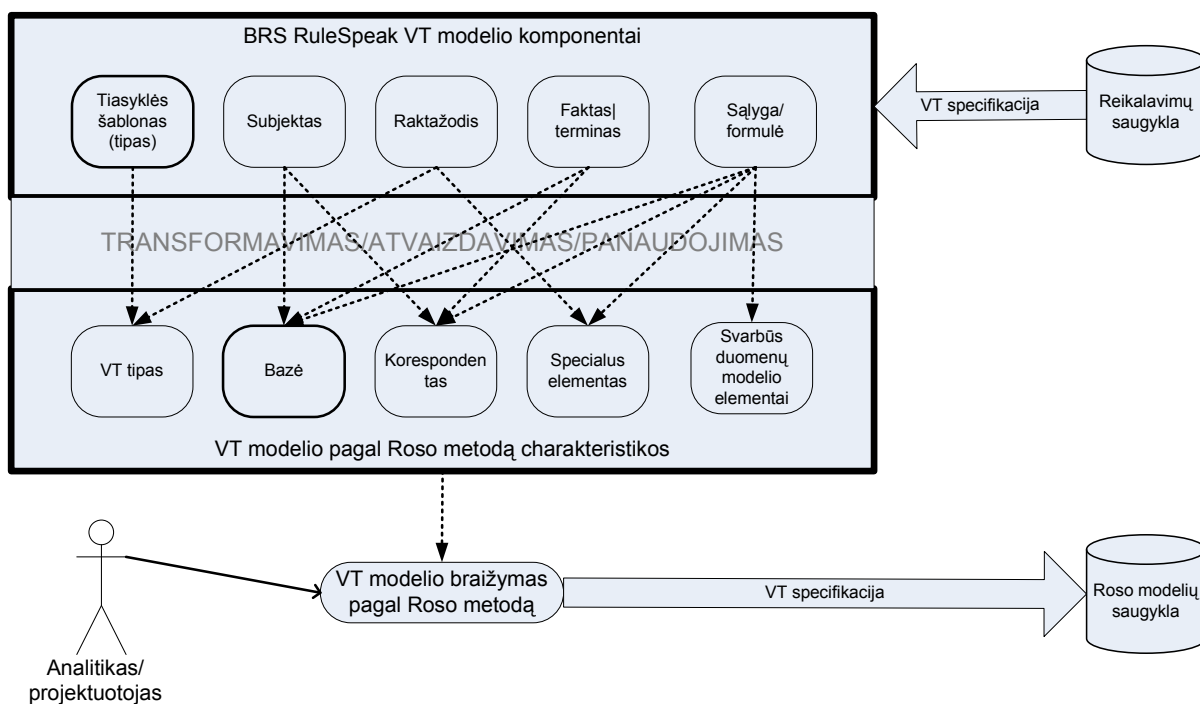
modeliuojant tokio tipo taisyklę, taisyklės-subjekto taisyklių simbolių numeracija turėtų būti keičiama atitinkamai pastumiant jos reikšmes vienu skaitmeniu į priekį, t.y. sąlyga tikrinama pirmiausiai, o tik po to sprendžiama ar nurodyta taisyklė bus (ne)vykdoma.

2.2 Perėjimo nuo specifikuojimo prie projektavimo principas - metodika

Metodika numato dalinį transformavimo automatizavimo metu įmanomą duomenų susiejimą Roso metodu su baze ar korespondentu, sąlyga ar apribojimu, t.y. pagrindiniais notacijos elementais. Tačiau projektuotojo valioje visada yra galimybė pakeisti šiuos duomenis. Tačiau ne tai, kas naudojama VT, bet naudojamų duomenų elementų susiejimą kaip Roso metodo elementų.

Būna atveju, kad pagal BRS RuleSpeak šablonus užrašyta VT generuojant ją Roso metodu tarsi apsiverčia ir yra interpretuojama nuo pabaigos. Kai taisyklės prasmė ne tai, kas galioja subjektui, ką jis turi (ne) turi daryti-atlikti-turėti, bet kas su juo (ne) turi būti daroma. Tokių išimčių generavimas paliekamas projektuotojui. Paleikant tik duomenų modelio elementų, susietų su VT, išvedimą į ekraną. Taip pat VT skaidymas į atomines Roso taisykles ir atitinkamas sąlygos ir/ar apribojimo simbolių kiekis taisyklėje yra tik projektuotojo valioje.

Principinė transformavimo schema pavaizduota 15 paveikslėlyje.



Šaltinis: sudaryta autoriaus

15 pav. Principinė transformavimo metodikos schema

Transformavimo eiliškumas yra sąlyginai lygiagretus ir nėra nustatyta griežta tvarka.

VT transformavimo metodiką galima būtų suskirstyti tokiais sąlyginiais žingsniais:

- Duomenų elementų, susietų su taisyklės subjektu, išvedimas į ekraną. Priskyrimas bazei.
- Nustatymas, ar Roso VT modelyje taisyklė yra sąlyga, ar apribojimas.
- Duomenų elementų, susietų su VT faktų/terminų išvedimas į ekraną (tik tuo atveju, jeigu tie duomenys dar nėra išvesti). Priskyrimas korespondentui.
- Tinkamų Roso VT tipų išskyrimas.
- Specialiųjų elementų poreikio identifikavimas (pagal raktinį VT žodį ar VT tipą).
- Sąlygoje naudojamų duomenų modelio elementų išvedimas į ekraną.

Toliau viskas priklauso nuo projektuotojo – ar bus daromi kažkokie loginiai pakeitimai pateiktame VT karkase (pvz. bazės ir korespondento sukeitimas), ar tik papildoma trūkstamais ryšiais, t.y. skaidoma į atomines Roso VT.

Ne visos VT gali būti realizuojamos Roso metodu. Todėl prieš pradėdant nagrinėti transformavimo metodiką pažingsniui, iškart išskirti VT tipai pagal BRS RuleSpeak kurie nebus modeliuojami Roso metodu. Apie tai parašyta skyriuje 2.1.2.

Svarbu pažymėti, jog atliekant modeliavimą Roso metodu reikėtų išlaikyti tokią pačią VT numeraciją kaip ir BRS RuleSpeak šablonais užrašytų taisyklių. Nors ji nebus nuosekli, turės trūkių, tačiau esant reikalui būtų patogiau atsekti reikiamas VT.

2.2.2 Subjekto transformavimas

Nagrinėjant modifikuotais BRS RuleSpeak šablonais užrašytų VT užrašymą modifikuotu Roso metodu, buvo pastebėta, kad beveik visais atvejais natūraliosios kalbos šablonais užrašytų VT subjektas tampa baze. Tačiau yra ir išimčių, kurios aptariamos šio darbo 2.1.2 skyriuje. Tokiu atveju, kai subjektas tikrai tampa baze, galima atlikti atitinkamų duomenų apie subjektą pritaikymą-perkėlimą bazei. Tai atliekama duomenų saugyklų lygmenyje.

Modifikuotoje reikalavimų specifikavimo saugykloje duomenys apie subjektą saugomi lentelėje *VeiklosTaisyklė* įrašais *SubjID* ir *SubjTipID*. *SubjTipID* nusako elemento tipą, kuris įrašytas *SubjID*. *SubjTipID* yra lygus *ELITID* reikšmei iš lentelės *SblElementoTipas*. Pastarojoje lentelėje saugoma informacija apie komponento tipą. Kaip žinia, subjektu parinktas duomenų modelio elementas, *SblElementoTipas.SubjPozymis* privalo turėti reikšmę *taip* (angl. true). Taigi subjekto elemento tipą nusako *SblElementoTipas.Kodas*, kuris gali būti atitinkamai (lentelės *SblELITipoKodas* atributas *Kodas*):

E – korespondentas yra esybė.

A – korespondentas yra atributas.

R – korespondentas yra ryšys.

S – korespondentas yra ISA ryšys.

C – korespondentas yra konstanta.

T – korespondentas yra kita VT negu ta, kuriai priskiriamas šis korespondentas.

I – korespondentas yra kitos VT negu ta, kuriai priskiriamas šis korespondentas, išeigos reikšmė.

Bazės tipas modifikuotoje Roso saugykloje *b_tipas* (lentelėje *VTaisykle*) turi būti toks pats, kaip aukščiau aptartas subjekto tipas.

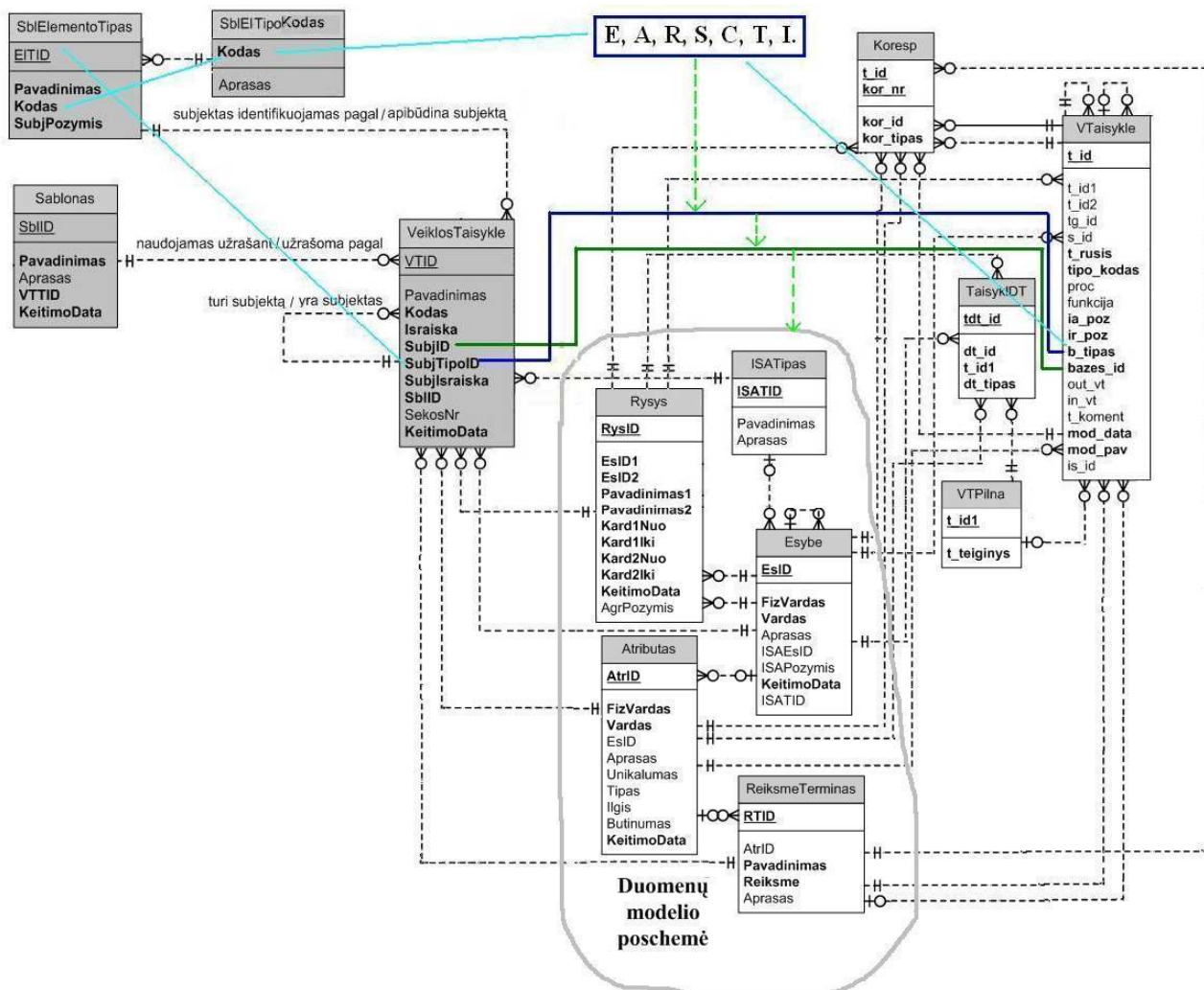
Lentelės *VeiklosTaisykle* atributas *SubjID* saugo subjekto id, t.y. atitinkamo duomenų modelio elemento id. Galimi variantai:

- pasirinkta *EsID* reikšmė iš lentelės *Esybe*;
- pasirinkta *AtrID* reikšmė iš lentelės *Atributas*;
- pasirinkta *RysID* reikšmė iš lentelės *Rsys*;
- pasirinkta *RTID* iš lentelės *ReiksmeTerminas*;
- pasirinkta *VTID* reikšmė iš jau esančių lentelėje *VeiklosTaisykle*.

Modifikuotoje reikalavimų specifikavimo saugykloje yra numatyta, kad *SubjID* gali įgyti reikšmę pasirinkta *FunkcID* iš lentelės *Funkcija*. Toks atvejis šioje metodikoje nenagrinėjamas, nes kaip numatyta skyriuje 2.1.2 ši metodika teigia, kad VT, kuriose subjektu yra funkcijos, t.y. procesai ir procedūros, Roso metodu nėra generuojamos.

Projektavimo saugyklos lentelės *VTaisykle* atributo *bazes_id* reikšmė turi būti tokia pati kaip ir nagrinėjamos VT reikalavimų specifikavimo saugyklos lentelės *VeiklosTaisykle* atributo *SubjID* reikšmė.

Kad aprašymas būtų aiškesnis, situaciją iliustruoja 16 paveikslėlis.



Šaltinis: sudaryta autoriaus

16 pav. Subjekto transformavimas

Bet, kaip jau minėta, ne visi subjekto atvejai gali būti iškart transformuojami į bazės egzempliorių, kadangi dėl skaidymo į atomines Roso taisyklės dažnai viskas tarsi apsverčia aukštyn kojom – subjektas tampa korespondentu ar pan. Tačiau dalis VT tokio transformavimo kaip aprašyta ir iliustruota aukščiau – įmanoma.

2.2.3 Sąlyga ar apribojimas

Ar Roso metodu užrašant VT ji bus sąlyga ar apribojimas galima spęsti pagal tai ar BRS RuleSpeak šablonais užrašytoje VT yra sąlygos elementai. Ne visuose šablonuose ji yra privaloma. Galima išskirti grupę šablonų, kuriuose sąlyga yra tikrai privaloma:

1.1 Atmetimo/apribojimo taisyklė: šablonai *b2*, *b3*;

3.1.1 Išvados taisyklė: šablonai *b10*, *b11*;

3.1.2 Taisyklės jungiklis: šablonai *b12*, *b13*;

3.3.2 Taisyklės trigeris: šablonas *b19*.

Taisyklės, užrašytas pagal šiuos šablonus, užrašant Roso metodu, į ekraną šalia visų duomenų modelio elementų turi būti išvedamas ir sąlygos simbolis. Integralumo apribojimo simbolis išvedamas bet kuriuo atveju – sąlyga taisyklės šablone yra ar nėra. Ši išvada daroma pagal tai, kad patikrinus sąlygą bus vykdoma kita taisyklė, kuri tikrai bus apribojimas ir vykdoma arba ne priklausomai nuo sąlygos.

Pagal tai, kad šablonais užrašytoje taisyklėje yra sąlyga, galima daryti išvadą, kad taisyklę užrašant Roso metodu, VT reikės skaidyti į atomines taisykles. Automatizuoti skaidymo procesą įmanoma tik taip, kad kai į ekrana išvedami sąlygos ir apribojimo simboliai, žinoma ir visi kiti su ta sąlyga susieti duomenų modelio elementai, projektuotojui gali būti pateikiama rekomendacija atkreipti į tai dėmesį. Automatiškai susieti sąlygos ir/ar apribojimo simbolius su kažkuriuo duomenų modelio elementu sudėtinga dėl didelio kiekio RuleSpeak modelyje „paslėptos“ neformalizuotos informacijos – susiejimą reikia palikti projektuotojui, kad tikrai neišvengtų logikos klaidų, kurios neatitiktų užrašytos VT.

Sąlygos ir/ar apribojimo simboliuje galėtų būti pasiūloma grupė taisyklių tipų tinkančių pagal taisyklės užrašymo šabloną. Tačiau tai įmanoma ne visada.

Tačiau kaip yra aptarta ankstesniame skyriuje 2.1.2, ne visos VT gali būti modeliuojamos Roso metodu, tad kai kuriems šablonams aprašytas transformavimo procesas nėra aktualus.

Nustatyta, kad VT pagal Roso metodą turinčios pozicijos patvirtinimo ar matematinio įvertinimo tipą, taisyklės visada bus sąlyga.

2.2.4 Fakto|termino kaip korespondento nustatymas

Korespondento nustatymas yra tik Roso metode. BRS RuleSpeak modifikuotame modelyje panašus šablono elementas nėra numatytas. Tačiau panagrinėjus BRS RuleSpeak šablonus, galima atrasti atvejų, kada tam tikras šablono elementas Roso metode galėtų būti priskiriamas kaip korespondentas.

Žinoma, kad korespondentu gali būti duomenų modelio objektas, kita taisyklė, jos išeigos reikšmė arba veiksmas. BRS RuleSpeak šablonuose subjekto dažniausiai sąlyginį korespondento vaidmenį atlieka tam tikras faktas, terminas (kaip šablonuose *b10*, *b11*, *b16*), reikšmė (kaip šablone *b16*), matematinė formulė (kaip šablonuose *b6*, *b7*), loginė išraiška (kaip šablonuose *b8*, *b9*) ar VT raktažodis (kaip *b4*, *b5* šablonuose). Galima daryti išvadą, kad šie išvardinti elementai modeliuojant VT Roso metodu turėtų tapti taisyklės korespondentu(-ais).

Išvedant į ekraną atitinkamą duomenų modelio elementą, kuris turėtų būti korespondentu, atitinkama lentelė Roso saugykloje *Koresp*, gali būti užpildoma šiomis reikšmėmis: *t_id*, kuri turi būti tokia pati kaip lentelėje *VTaisykle.t_id* – ši reikšmė yra žinoma, nes yra priskiriama automatiškai pradėjus generuoti taisyklę. Pagal tai, koks duomenų modelio elementas BRS RuleSpeak yra faktas, terminas, reikšmė, matematinė formulė, loginė išraiška ar VT raktažodis, *VTaisykle.kor_tipas* nustatomas atitinkamai:

E – korespondentas yra esybė.

A – korespondentas yra atributas.

R – korespondentas yra ryšys.

S – korespondentas yra ISA ryšys.

C – korespondentas yra konstanta.

T – korespondentas yra kita VT negu ta, kuriai priskiriamas šis korespondentas.

I – korespondentas yra kitos VT negu ta, kuriai priskiriamas šis korespondentas, išieigos reikšmė.

Pagal parinktą iš aukščiau nurodytų tipų lentelės *VTaisykle.kor_id* įrašas užpildomas:

- Pasirinkta *EsID* reikšmė iš lentelės *Esybe* (kai *kor_tipas* = E, S),
- Pasirinkta *AtrID* reikšmė iš lentelės *Atributas* (kai *kor_tipas* = A),
- Pasirinkta *RysID* reikšmė iš lentelės *Rysys* (kai *kor_tipas* = R),
- Pasirinkta *RTID* reikšmė iš lentelės *ReiksmeTerminas* (kai *kor_tipas* = C).

Korespondentu nustatomas faktas yra sujungiamas punktyrine linija su taisyklės simboliu išvestu į ekraną. Tačiau tai yra tik pirminis metodikos siūlomas variantas, kurį vėliau projektuotojas gali koreguoti.

Neįmanoma automatizuoti korespondento parinkimą, kai taisyklė yra skaidoma į atomines Roso taisykles ir to pasekoje atsiranda keletas VT, kuriose korespondentais tampa kiti duomenų modelio elementai, o tuo pačiu ir iš VT skaidymo išplaukiančios reikšmės kita atominė VT toje pačioje logiškai nedalomoje VT ar tokios taisyklės išieiga. Tai yra “viduje” taisyklės atsirandantys korespondentai turės būti parenkami projektuotojo. Pagal projektavimo saugykloje numatytas reikšmes negalima automatiškai užpildyti lentelės *Vtaisyklė.kor_id* reikšmėmis tuomet, kai pasirinkta *t_id* reikšmė iš lentelės *VTaisyklė* (kai *kor_tipas* = T, I).

Taigi, korespondento parinkimo automatizavimas įmanomas tik su pradinėmis fakto reikšmėmis iš reikalavimų specifikavimo šablonų.

2.2.5 Roso VT tipo nustatymas pagal BRS RuleSpeak tipą

Atliekant taisyklių transformavimą iš BRS RuleSpeak modifikuoto modelio į Roso modifikuotu metodu užrašomas taisyklės, labai svarbu yra nustatyti tinkamą taisyklės tipą. Nes kaip yra žinoma, taisyklės tipas nulemia atliekamus patikrinimus ar tam tikrus veiksmus (apribojimus) su duomenimis iš duomenų modelio.

BRS RuleSpeak modelyje taisyklės yra suskirstytos į tris dideles grupes – atmetimo, generavimo ir numatymo, kuriose iš viso yra 12 smulkesnių tipų. Roso metode yra išskirtos dvi sąlyginai didelės taisyklių grupės – nedalomos ir išvestinės. Jose atitinkamai yra 7 ir 12 tipų grupių (šeimų), kurios išskirtos atitinkamai į 32 ir 58 taisyklių tipus. Taigi iš viso turima 90 tipų. Konkrečiai kiekvienam taisyklių tipui pagal BRS RuleSpeak metodą priskirti kažkurį tipą pagal Roso metodą yra neįmanoma. Tačiau bent dalinis tipų grupių susiejimas palengvintų VT projektavimą ar dalinį jo automatizavimą. Kuomet projektuotojui būtų pasiūloma pasirinkti iš susiaurintos (apimančios tik galimus tipus) tipų ar jų grupių aibės, projektavimo procesas galėtų vykti žymiai sparčiau dėl sutrumpėjusio sprendimo, t.y. tinkamo taisyklės tipo parinkimo, priėmimo laiko.

Kaip jau buvo minėta ankstesniame skyriuje 2.1.2, ne visas VT galima sumodeliuoti Roso metodu. Tad tų BRS RuleSpeak taisyklių tipų atitikmenys Roso taisyklių tipuose nebus nagrinėjami, t.y. taisyklių tipai: proceso jungiklis (3.1.3), duomenų jungiklis (3.1.4), pateikimo taisyklė (3.2.2.) ir proceso trigeris (3.3.1).

Galima teigti, kad taisyklės tipą pagrinde nusako VT raktiniai žodžiai, kurie nurodo, kas turi būti atliekama su objektu.

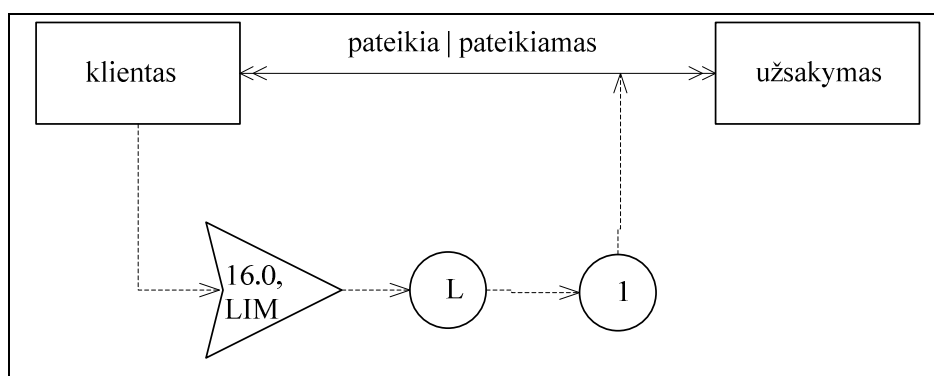
Keblumas bandant automatizuoti transformavimo proceso dalį, kuomet reikia parinkti VT tipą, yra tame, kad elementarių VT, kurios būtų realizuojamos tik viena atominė Roso taisyklė, būna ne tiek jau daug. Užrašant VT Roso metodu taisyklių skaidymas į atomines yra projektuotojo valioje. Numatyti skaidymo automatiškai yra neįmanoma. Tad ir nustatyti VT tipą kiekvienam naudojamam taisyklės simboliui pilnai ar bent iš dalies nėra įmanoma. Galima bandyti nuspėti, t.y. parinkti bent susiaurintą galimų VT tipų aibę, tik pagrindiniam taisyklės simboliui (kurio eiliškumas taisyklėje numatomas projektuotojo). Taip yra dėl to, kad neįmanoma automatizuotai numatyti kas gali būti naudojama sąlygose (VT užrašytoje pagal BRS RuleSpeak šabloną) ir kokius patikrinimus gali reikėti atlikti. Taigi numatyti galimą tipų grupę įmanoma tik vienam taisyklės simboliui.

Dalis šiame skyriuje pateikiamų pavyzdžių yra adaptuoti iš [21].

2.2.5.1 Atmetimo tipo taisyklės

Pagal atmetimo tipo grupės taisyklių (1.1. ir 1.2) raktinius žodžius TURI|NETURI|TURĖTU|NETURĖTU|GALI aiškiai matoma, kad subjektas turi tenkinti nurodytą faktą (priklausomai nuo raktažodžio – privalo arba neprivalo). Taisyklės tipas pagal Roso notaciją yra egzempliorių patvirtinimo šeimos ir gali būti arba privaloma (X), arba apribota (LIM) tipo. Privaloma tipo taisyklės pavyzdys jau anksčiau nagrinėtas skyriuje 2.1, 8 paveikslėlis. Pateikiamas apribojimo tipo taisyklės pavyzdys 17 paveikslėlyje.

Klientas turi pateikti bent vieną užsakymą.

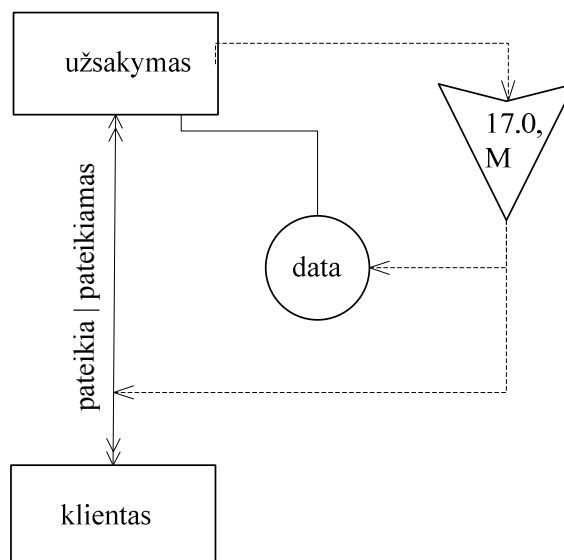


Šaltinis: sudaryta autoriaus

17 pav. Apribojimo tipo taisyklė Roso metodu

Gali būti atveju, kad atmetimo tipo taisyklėje išreiškiamas apribojimas, kurio nepažeidžiant užtikrinamas duomenų teisingumas, bus išreikštas dviem apribojimais, kurie logiškai sujungti IR ar ARBA jungtukais. Tai yra, taisyklė turės turėti arba visus nurodytus faktus-korespondentus, arba bent vieną iš jų – žinoma priklausomai nuo pačioje VT užduodamų sąlygų. Tokiu atveju, kai yra naudojami loginiai operatoriai IR ar ARBA, VT tipas turi būti tipo patvirtinimo: bendras (M), bendrai-atiskiriantis (ME), bendrai-susiejantis (MI) arb bendrai-uždraudžiantis (MP). Iliustruojantis pavyzdys 18 paveikslėlyje.

Užsakymas privalo turėti jo pateikimo datą ir jį pateikusį klientą.

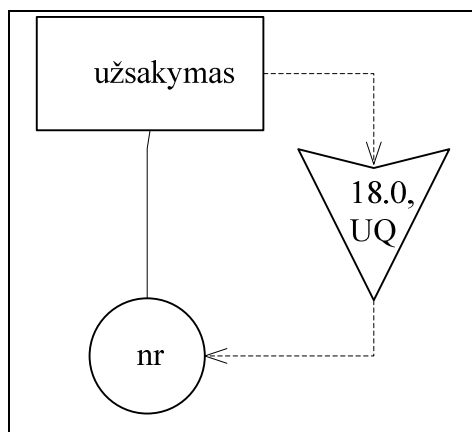


Šaltinis: sudaryta autoriaus, pagal Ross, R. G. (1997). *The Business Rule Book: Classifying, Defining and Modeling Rules*. Business Rule Solutions, Houston.

18 pav. Bendro tipo taisyklė Roso metodu

Atmetimo tipo taisyklės BRS RuleSpeak taip pat gali būti atvaizduojamos kaip funkcinio patvirtinimo tipo pagal Roso notaciją. Tokiu atveju, projektuotojas turėtų parinkti taisyklę atitinkantį tipą: funkcinį (FUNC), unikalų (UQ), svyruojantį (FLUCK), augantį (AS), krentantį (DE), neatnaujinamą (NRE).

Užsakymo numeris turi būti unikalus.

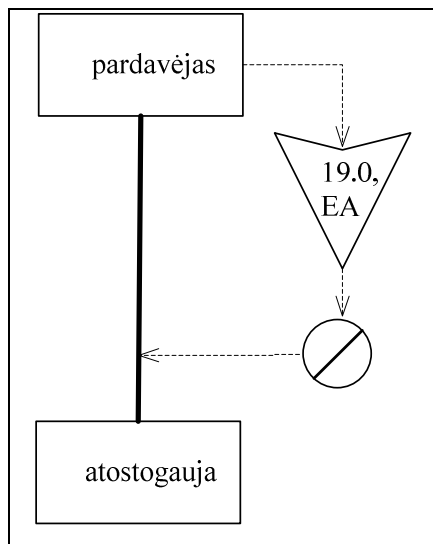


Šaltinis: sudaryta autoriaus

19 pav. Unikalaus tipo taisyklės panaudojimo pavyzdys

Taip pat BRS RuleSpeak atmetimo tipo taisyklės gali būti atitinkamai Roso metode projekcijos valdiklių įgalinimo tipo. Tokio tipo taisyklės pavyzdys pateiktas 20 paveikslėlyje.

Pardavėjas negali eiti atostogų.



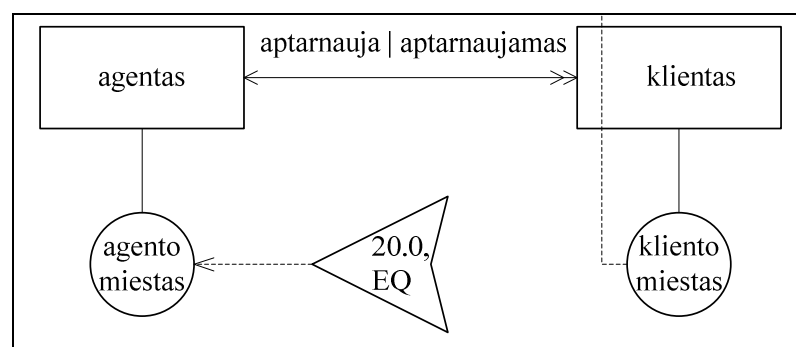
Šaltinis: sudaryta autoriaus pagal Ross, R. G. (1997). *The Business Rule Book: Classifying, Defining and Modeling Rules*. Business Rule Solutions, Houston.

20 pav. Įgalinimo tipo taisyklė Roso metodu

Šiuo atveju, galima daryti pastebėjimą, kad vietoj įgalinimo tipo (EA) taisyklės galima būtų naudoti tiesiog privaloma (X) tipo taisyklę, žinoma abi su neigimo reikšmę turinčiais simboliais. Tačiau įgalinimo taisyklės tipas čia tiksliau nusako kas yra draudžiama.

Atmetimo tipo taisyklė (BRS RuleSpeak) taip pat daugeliu atvejų gali būti ir palyginamojo įvertinimo tipo (Roso metode). Kaip yra žinoma pagal Roso notaciją, lyginamos gali būti tik tokio pat tipo reikšmės. Pavyzdys 21 paveikslėlyje, kur pateikiamas apribojimas kurio nepažeidžiant užtikrinamas duomenų teisingumas - tarpusavyje tiesiogiai lyginamos *kliento miesto* ir *agento miesto* reikšmės.

Klientas gali būti aptarnaujamas tik agento iš to pačio miesto.



Šaltinis: sudaryta autoriaus pagal Ross, R. G. (1997). *The Business Rule Book: Classifying, Defining and Modeling Rules*. Business Rule Solutions, Houston.

21 pav. Palyginamojo įvertinimo tipo taisyklė Roso metodu

Peržvelgus visus taisyklių tipus, galima daryti išvadą, kad atmetimo/leidimo tipo taisyklėms, užrašytoms pagal BRS RuleSpeak, transformacijos metu galima pritaikyti beveik visus Roso notacijos nedalomų taisyklių tipus, išskyrus matematinio įvertinimo.

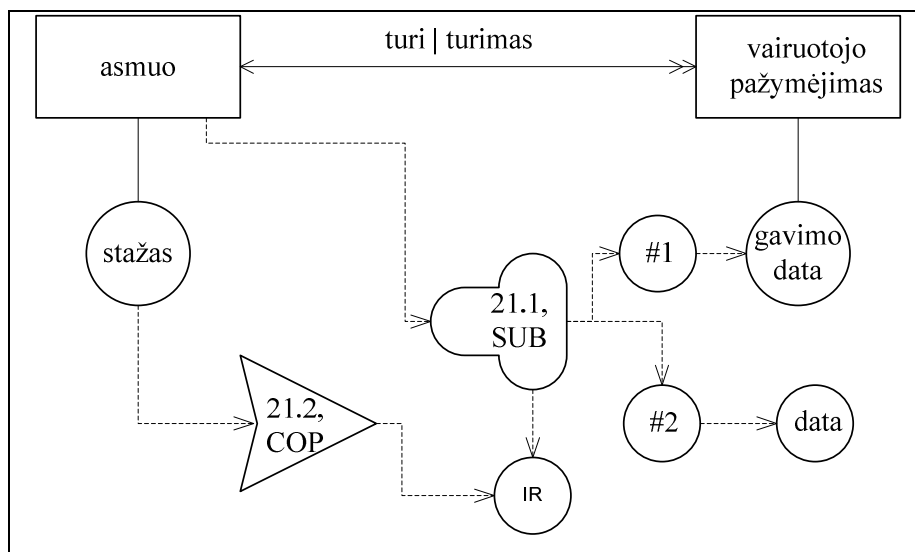
2.2.5.2 Generavimo tipo taisyklės

Generavimo tipo taisyklės (BRS RuleSpeak) atlieka skaičiavimo ir išvedimo veiksmus (2.1 ir 2.2). Kadangi jų atliekamos funkcijos susietos su matematiniais veiksmais (skaičiavimais, palyginimais), tai šio tipo taisyklėms modeliuojamoms Roso metodu turėtų būti parenkamas kažkuris palyginamojo įvertinimo ar matematinio įvertinimo tipo grupei priklausantis taisyklės tipas. Tai gali būti:

- palyginamojo įvertinimo – lygu (EQ), nelygu (NE), daugiau-už (GT), daugiau-arba-lygu (GE), mažiau-už (LT), mažiau-arba-lygu (LE);
- matematinio įvertinimo – apskaičiuota (CALC), susumuota (SUM), atimta (SUB), sudauginta (MULT), padalinta (DIV) ar bet koks matematinis veiksmas.

Pateikiama keletas taisyklės iliustruojančių pavyzdžių - 22 ir 23 paveikslėliai.

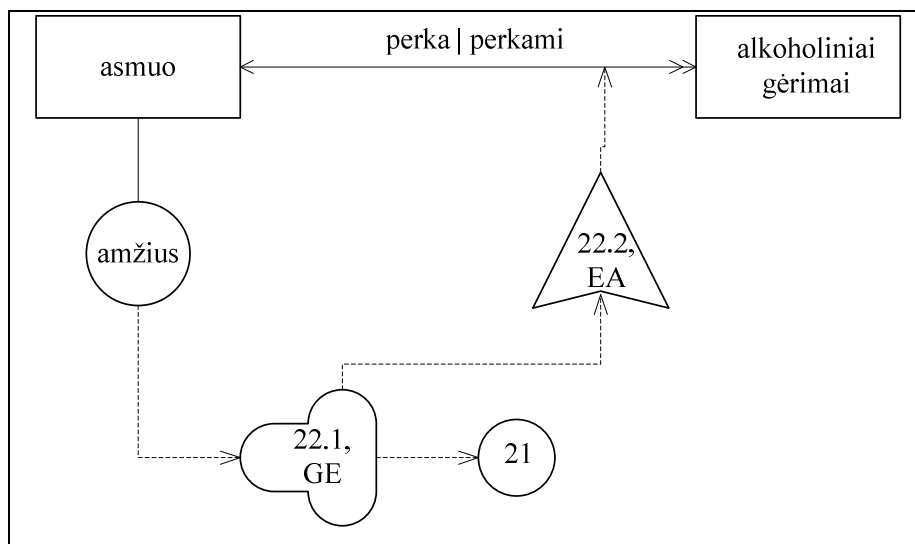
Asmens vairavimo stažas apskaičiuojamas kaip skirtumas tarp einamosios datos ir asmens vairuotojo pažymėjimo gavimo datos.



Šaltinis: sudaryta autoriaus

22 pav. Matematinio įvertinimo (atimties) tipo taisyklė Roso metodu

Asmuo gali pirkti alkoholinius gėrimus tik nuo 21-erių metų.



Šaltinis: sudaryta autoriaus

23 pav. Palyginamojo įvertinimo (daugiau-lygu) tipo taisyklė Roso metodu

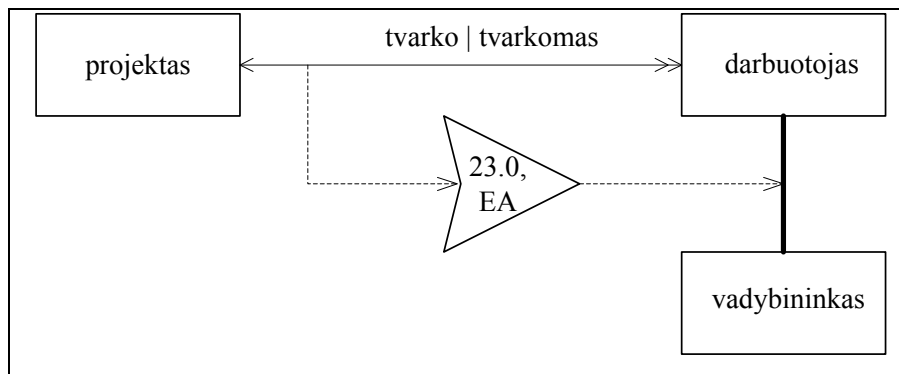
Tiksliau galima išskirti, kad BRS RuleSpeak šablonu 2.1 skaičiavimo tipo taisyklė generuojama Roso metodu visada turės matematinio įvertinimo tipo taisyklę, nes tik šio tipo taisyklės atlieka skaičiavimus pagal nurodytas formules. Išvedimo tipo taisyklė 2.2 pagal BRS RuleSpeak visada Roso metode taps palyginamojo įvertinimo tipo taisykle, nes tik šios taisyklės pritaiko tiesioginį reikšmių suliginimo testą, tuo būdu nusakant kaip nustatyti taip/ne pobūdžio rezultata.

2.2.5.3 Numatymo tipo taisyklės

BRS RuleSpeak numatymo tipo taisyklės gali būti transformuojamos į projekcijos valdiklius Roso metodu užrašomose taisyklėse. Ir BRS RuleSpeak, ir Roso metode, atitinkamai numatymo ir projekcijos valdiklių tipų grupių taisyklės suskirstytos į tris pagrindinius tipus. Abiejuose metoduose jų tipų pavadinimai ir atleikamos funkcijos sutampa: įgalinimo taisyklės, kopijuotoja, vykdytoja. Todėl galima teigti, kad atitinkamai toks pats taisyklės tipas bus ir po transformacijos.

Išvados taisyklė BRS RuleSpeak (3.1.1) visada bus įgalinimo tipo (EA) Roso metodu užrašytoje VT. Tai galima įrodyti pavyzdžiu 24 paveikslėlyje.

Darbuotojas turi būti laikomas vadybininku, jeigu jam paskirta tvarkyti projektą.



Šaltinis: sudaryta autoriaus pagal Ross, R. G. (1997). *The Business Rule Book: Classifying, Defining and Modeling Rules*. Business Rule Solutions, Houston.

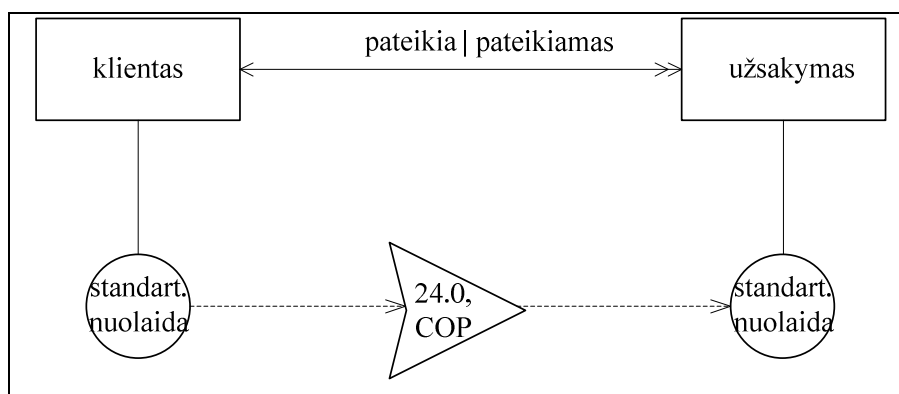
24 pav. Įgalinimo tipo taisyklė Roso metodu

Anksčiau nagrinėtas pavyzdys, BRS RuleSpeak *b10* šablonu užrašytos taisyklės modeliavimas Roso metodu, t.y. 13 paveikslėlis, įrodo tą patį teiginį. Žinoma ta taisyklė susideda iš dviejų atominių Roso metodu užrašomų taisyklių, bet esminis logiškai nedalomos taisyklės tipas - įgalinimo vykdyti VT, tik jei tenkinama nurodyta sąlyga.

Taisyklės jungiklio tipo (3.1.2, BRS RuleSpeak) VT projektuojamos Roso metodu turi tapti taip pat įgalinimo tipo VT. Tokia išvada daroma pagal formalaus 3.1.2 šablono varianto *b13* raktažodį turi|neturi|turėtų|neturėtų BŪTI TAIKOMA|-a ir VT tipą - įgalinimo. Pagal šią formuluotę matoma, kad šablone nurodyta taisyklė <Taisyklės pavadinimas|kodas> paleidžiama (arba nepaleidžiama – priklausomai nuo raktinio žodžio) tik kai patenkinama nurodyta sąlyga.

BRS RuleSpeak reikšmės priskyrimo tipo taisyklė 3.2.1 pagal *b16* šabloną taip pat turi/neturi įgyti tam tikrą nurodytą reikšmę. Tai galima realizuoti taip pat projekcijos valdiklių tipų grupės elementu – nukopijuotoja (COP). Pavyzdys pateikiamas 25 paveikslėlyje.

Užsakymo-nuolaida turi įgyti reikšmę standartinė-kliento-nuolaida.



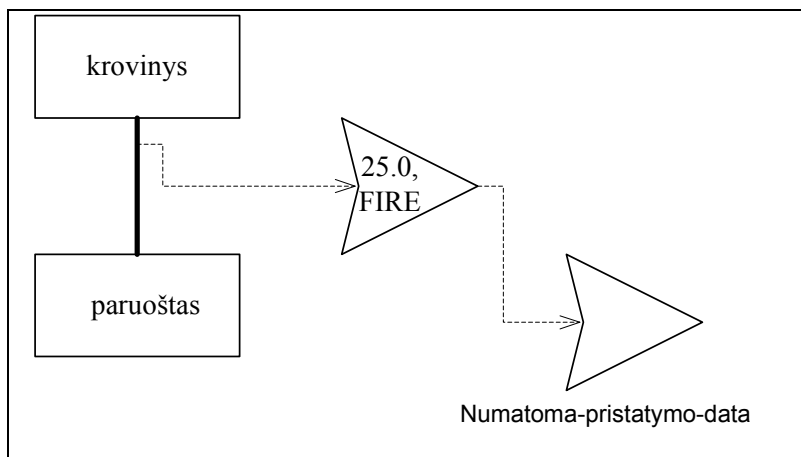
Šaltinis: sudaryta autoriaus pagal Ross, R. G. (1997). *The Business Rule Book: Classifying, Defining and Modeling Rules*. Business Rule Solutions, Houston.

25 pav. Kopijavimo tipo taisyklė Roso metodu

Taisyklės trigerio tipo (vykdytoja) taisyklė 3.3.2 (BRS RuleSpeak) pagal jos šablono raktinius žodžius gali būti taip pat vienu iš projekcijos valdiklių tipo (Roso metodu) – nedalomo tipo įvykdyta (EX) arba išvestinio tipo įvykdyta-su-atkūrimu (REX), paleista (FIRE) ar numatyta-įvykdymui (TEX).

Nurodyta VT <Taisyklės pavadinimas|kodas> atitinkamai pagal raktinį žodį turi|neturi|turėtų|neturėtų BŪTI PALEISTA|-as tik tuo atveju, jeigu tenkinama nurodyta sąlyga. Pavyzdys pateikiamas 26 paveikslėlyje.

Taisyklė Numatoma-pristatymo-data turi būti paleista, kai krovinys yra paruoštas.



Šaltinis: sudaryta autoriaus

26 pav. Paleidimo tipo taisyklė Roso metodu

Kaip ir anksčiau aptartuose pavyzdžiuose VT, užrašytos pagal bet kuri BRS RuleSpeak numatymo tipo šablona, modeliuojant Roso metodu gali būti skaidomos į atomines taisykles. Tokiu atveju, kaip jau buvo minėta anksčiau, VT turės ne tik kažkurį iš projekcijos valdiklių tipą, bet ir priklausomai nuo sąlygos bet kuri iš nedalomų ar išvestinių tipų.

2.2.5.4 VT tipų nustatymo sunkumai

Išnagrinėjus visus taisyklių tipus pagal BRS RuleSpeak notaciją ir sugretinus juos su taisyklių tipais pagal Roso notaciją, daroma išvada, kad susieti tipų grupes bent dalinai, įmanoma labai nedaug. Iš BRS RuleSpeak taisyklių tipų labiau išsiskiria generavimo ir numatymo tipo taisyklės. Generavimo tipo taisyklės Roso notacijoje atitinka matematinio įvertinimo ir palyginamojo įvertinimų tipų grupės. Numatymo tipo taisyklės BRS RuleSpeak atitinka Roso metodo projekcijos valdiklių tipų grupę.

Ypač daug Roso taisyklių tipų gali būti panaudota modeliuojant pagal atmetimo tipo (1.1 ir 1.2) šablonus užrašytas VT. Netinka tik pozicijos patvirtinimo ir matematinio įvertinimo tipo nedalomos taisyklės.

Tačiau šis VT tipo parinkimo numatymas naudingas tik jei taisyklė yra neskaidoma į atomines Roso taisykles. Tokiu atveju, kai taisyklės simbolis yra tik vienas, gali būti pasiūlyta susiaurinta galimų VT tipų grupė. Jeigu VT yra skaidoma – kitiems taisyklės simboliams (sąlygos ir/ar apribojimo) parinkti galimų VT tipų grupių nėra įmanoma.

Taip pat daroma išvada, kad dauguma (galbūt net visi) išvestiniai taisyklių tipai pagal Roso notaciją gali būti naudojami tik kaip papildomos pasirinkimo galimybės skaidant logiškai nedalomas taisykles į atomines Roso VT, t.y. gali išryškėti tik detalizuojant Roso VT modelį. Tokiais atvejais tipų parinkimas daugiau nei vienam taisyklės simboliui priklauso tik nuo to, kaip natūraliaja kalba užrašytoje taisyklėje užfiksuotą informaciją interpretuos projektuotojas.

2.2.6 Specialiųjų elementų identifikavimas

Modeliuojant VT Roso metodu pagal poreikį yra naudojami įvairūs specialūs elementai. Specieji simboliai gali leisti interpretuoti taisyklę ne pagal tiesioginį jos tipą arba naudojami pilnai išreikšti sąlygą.

Galima išskirti keletą specialiųjų elementų, kurių panaudojimą schemeje galima nuspręsti pagal VT naudojamus raktinius žodžius. Tai yra neigimo reikšmę turintys raktažodžiai, tokie kaip NETURI|NETURĖTŪ|NEREIŠKIA|NĖRA. Aptikus tokį raktažodį modeliuojamoje VT, būtų galima iškart su taisyklės simboliu (sąlyga ir/ar apribojimu) išvesti vieną iš tokią reikšmę turinčių simbolių: paneigimą išreiškiantis simbolis (*angl.* false) arba priešingos reikšmės simbolis (*angl.* negator).

Taisyklėms, kurių tipas pagal Roso metodą turi būti apribota (LIM), šalia taisyklės simbolio turėtų būti išvedami du papildomi specieji simboliai apribojimais, iš kurių pirmasis turėtų būti vieno iš numeratoriaus simbolio tipų:

L – žemesnis, mažesnis negu, ne daugiau (*angl.* lower),

U – didesnis, viršijantis, ne mažiau (*angl.* upper),

F – nustatytas, pastovus, nekintamas, fiksuotas (*angl.* fixed).

Antrajame simbolyje turi būti įrašoma konkreti reikšmė. Šie elementai turėtų būti iškart susieti su taisyklės simboliu (sujungti punktyrinėmis linijomis).

Daugiau specialiųjų elementų, kuriuos būtų galima parinkti automatiškai, nėra.

Specialiųjų elementų ribojimas

Roso notacija numato specialiųjų simbolių panaudojimą su tam tikrais VT tipais. Pavyzdžiui, neigimo simbolis *negator* yra negalimas egzempliorių patvirtinimo, matematinio įvertinimo, projekcijos valdiklio ir daugumoje kitų tipo patvirtinimo šeimos taisyklių. Su šiais ir likusiais VT tipais galima naudoti *truth value* specialųjį elementą. Pažymėtina, jog visų galimų kombinacijų lentelė yra sudaryta paties R.Roso, todėl kuriant taisyklių modelių generavimo sprendimą į ją turėtų būti atsižvelgiama.

2.2.7 Papildomų duomenų modelio elementų išvedimas

Ankstesniuose skyriuose numatyti transformavimo žingsniai skirti privalomų BRS RuleSpeak šablonų elementų išvedimo į ekraną aprašymui. Tačiau kaip jau buvo minėta, ne visuose šablonuose sąlygos elementas yra privalomas. Todėl atlikus subjekto, fakto|termino ir/ar sąlygos simbolių išvedimą į ekraną, šalia jų reikia išvesti tuos duomenų modelio elementus, kurie nėra panaudoti ir susieti su subjektu ar faktu, tačiau yra naudojami modeliuojamos VT šablono sąlygos elemente. Tokiu atveju projektuotojui reikės dar atidžiau nagrinėti turimą pagal šabloną užrašytą VT ir atlikti atitinkamus dalinai sugeneruoto taisyklės modelio patikslinimus, susiejant išvestus elementus tarpusavyje.

2.3 VT transformavimo metodikos apibendrinimas

Sudaryta VT transformavimo metodika susideda iš šešių sąlyginių žingsnių:

- Duomenų elementų, susietų su taisyklės subjektu, išvedimas į ekraną. Priskyrimas bazei.
- Nustatymas, ar Roso VT modelyje taisyklė yra sąlyga, ar apribojimas.
- Duomenų elementų, susietų su VT faktų|terminų išvedimas į ekraną (tik tuo atveju, jeigu tie duomenys dar nėra išvesti). Priskyrimas korespondentui.
- Tinkamų Roso VT tipų išskyrimas.
- Specialiųjų elementų poreikio identifikavimas (pagal raktinį VT žodį ar VT tipą).
- Sąlygoje naudojamų duomenų modelio elementų išvedimas į ekraną.

Ne visais atvejais pirminis bazės ir korespondento automatizuotas nustatymas yra tinkamas. Tačiau tai nesunkiai gali būti ištaisoma projektuotojo.

Kuomet taisyklėje, užrašytoje pagal BRS RuleSpeak, yra naudojama sąlyga, dažniausiai jos atvaizdavimo Roso modelyje specifika (t.y. taisyklės tipas, bazė, korespondentas(-ai)) yra nežinoma. Todėl automatizuojant projektavimo procesą dažniausiai įmanoma pateikti tik numanomą taisyklės simbolį (sąlygą ar apribojimą), kurio tipo parinkimas priklauso tik nuo projektuotojo, atliekančio VT modeliavimą. Automatiškai tam tikrais atvejais galima pateikti tik susiaurintą galimų taisyklės tipų sąrašą.

3. EKSPERIMENTINIS VT TRASFORMAVIMO METODIKOS TYRIMAS

Šioje darbo dalyje bus atliekamas praktinis sukurtos metodikos išbandymas ir įvertinimas.

3.1 Eksperimento koncepcija

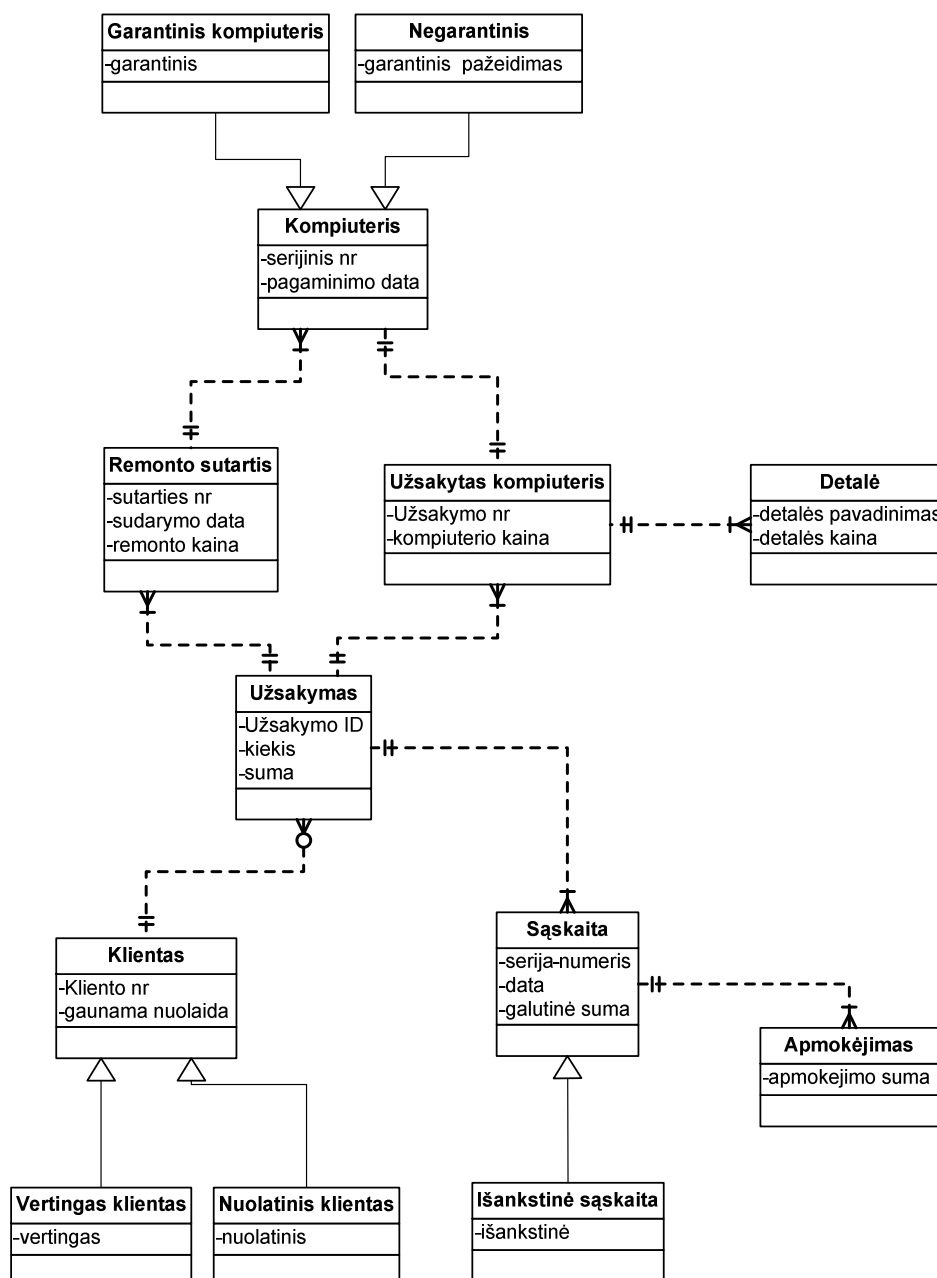
Atlikus preliminarą problemos analizę suformuluota metodika susidedanti iš 6 sąlyginių žingsnių: subjekto duomenų modelio elementų išvedimas į ekraną ir priskyrimas bazei, sąlygos ir/ar apribojimo išvedimas į ekraną, duomenų elementų susietų su faktais|terminais išvedimas į ekraną ir priskyrimas korespondentui, VT tipų grupių nustatymas, specialiųjų elementų panaudojimo nustatymas, sąlygoje naudojamų duomenų modelio elementų išvedimas į ekraną.

Siekiant išbandyti sukurtą metodiką priimtas sprendimas imituoti reikalavimų specifikavimą ir projektavimą pasirinktoje nedidelėje realioje dalykinėje srityje išskiriant reprezentatyvų rinkinį VT, kuriose būtų įvairių tipų taisyklių. Išskiriant testines taisykles nebuvo siekiama labai tiksliai ir išsamiai specifiuoti dalykinę sritį. Pagrindinis tikslas buvo parinkti kuo skirtingesnes VT, kurios būtų užrašomos kuo įvairesniais natūraliosios kalbos šablonais, t.y. bent po vieną iš priskiriamų skirtingiems VT tipams. Taip siekiama kuo tiksliau išbandyti metodikos veikimą ir nustatyti jos efektyvumą.

3.2 Eksperimento dalykinės srities charakteristika

Pasirinkta dalykinė sritis - kompiuterinės įrangos prekybos ir remonto paslaugas teikiančios įmonės veikla. Analizuojami bendri darbo su duomenimis atvejai. Įmonės klientas gali pateikti užsakymus. Pagal kliento užsakymo sumos dydį gali būti taikomos nuolaidos ar apribojimai dėl sąskaitų pateikimo. Taip pat pagal kliento tipą – vertingas jis ar ne, gali būti taikomi apribojimai nuolaidai ar sąskaitų pateikimui. Kompiuteris gali būti pristatytas remontui, kur yra nustatoma ar galioja garantija, ar ne. Klientui gali būti pasiūloma pirkti, pavyzdžiui, naują kompiuterį.

Pagal pateiktą dalykinės dalies aprašymą sudarytas duomenų modelis (27 paveikslėlis). Jame matomi duomenys naudojami tiek VT reikalavimų specifikavimo, tiek VT projektavimo metu.



Šaltinis: sudaryta autoriaus

27 pav. Konceptinis dalykinės srities duomenų struktūros modelis

3.3 Veiklos taisyklių specifikuojimas pagal BRS RuleSpeak

Pagal pasirinktą dalykinę sritį buvo suformuluotos VT, kurios specifikuotos pagal BRS RuleSpeak natūraliosios kalbos šablonus.

VT rinkinys pagal BRS RuleSpeak šablonus

VT nr.	BRS RuleSpeak šablonas, taisyklių tipų grupė	Veiklos taisyklė
1.	<i>b1</i> , atmetimo taisyklė	Remonto sutartis turi turėti unikalų numerį
2.	<i>b2</i> , atmetimo taisyklė	Išankstinė sąskaita rašoma tik jeigu klientas nėra įtrauktas į nuolatinių klientų sąrašą.
3.	<i>b6</i> , generavimo taisyklė	Kompiuterio kaina turi būti apskaičiuojama kaip visų jo detalių kainų suma.
4.	<i>b8</i> , generavimo taisyklė	Vertingas klientas turi reikšti, kad kliento visų apmokėjimų už paslaugas ir prekes suma viršija 3000 Lt.
5.	<i>b11</i> , numatymo taisyklė	Kompiuteris nėra garantinis, jeigu kompiuterio pagaminimo data yra daugiau nei dviem metais ankstesnės už šandienos datą arba yra pažeistas garantinis lipdukas.
6.	<i>b13</i> , numatymo taisyklė	Taisyklė Išankstinė-sąskaita-rašoma neturi būti taikoma, jeigu užsakymo kaina viršija 10 000 Lt.
7.	<i>b14</i> , numatymo taisyklė	Pranešimo-apie-remonto-pabaigą-siuntimas turi būti uždraustas, jeigu nėra žinomas kliento telefono numeris.
8.	<i>b15</i> , numatymo taisyklė	Kiekviena užsakymo prekė, kurios kiekis nenurodytas, turi būti ištrinta, kai užsakymas yra uždaromas.
9.	<i>b16</i> , numatymo taisyklė	Kliento nuolaida turi įgyti 10 procentų kai kliento užsakymo suma yra bent 1000 Lt.
10.	<i>b17</i> , numatymo taisyklė	Remonto-sutarties-numeris turi būti pateikiamas ekrane raudonai, jei šandienos data yra vėlesne ar lygi sutarties sudarymo datai plus 14 dienų.
11.	<i>b18</i> , numatymo taisyklė	Siųsti-išankstinį-įspėjimą užsakymui turi būti vykdomas kai užsakymas išsiųstas.
12.	<i>b19</i> , numatymo taisyklė	Taisyklė kliento-nuolaida neturi būti paleista kai klientas yra nevertingas.

Šaltinis: sudaryta autoriaus

Remiantis šiame darbe aprašyta metodika, VT užrašytos pagal BRS RuleSpeak šablonus *b14*, *b15*, *b17*, *b18*, nėra generuojamos. Pateikiamos rekomendacijos perduoti tokio tipo taisykles generuoti kitiems projektuotojams, kurie jas atvaizduotų tinkamu modeliu, pavyzdžiui., veiklos modeliu, procesų specifikacijos, vartotojo sąsajos ar pan.

Tokio tipo taisyklės yra išskirtos 4-oje lentelėje ir nebus generuojamos.

Roso metodu nemodeliuojamos VT

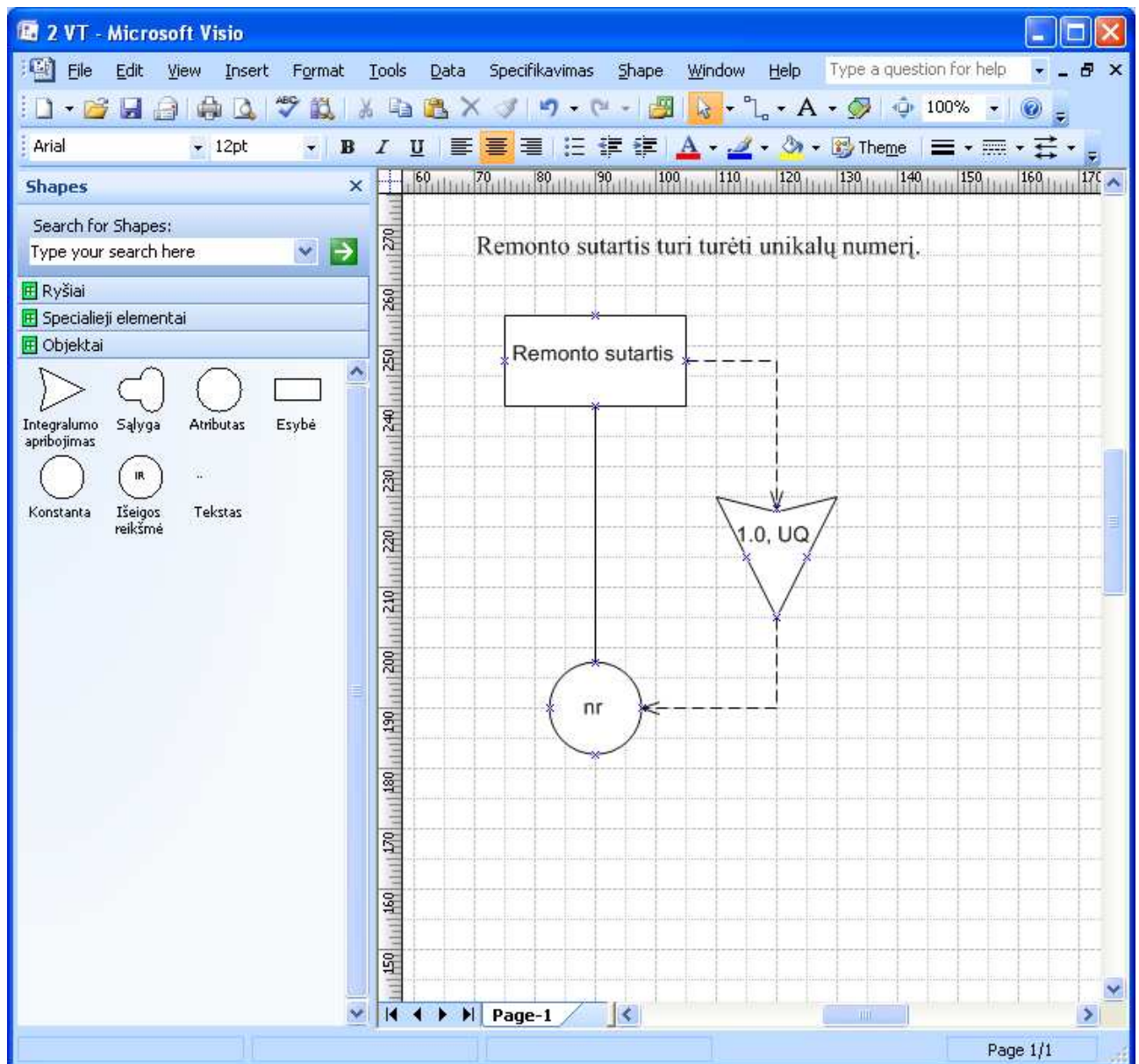
VT nr.	BRS RuleSpeak šablonas, taisyklių tipų grupė	Veiklos taisyklė
7.	<i>b14</i> , numatymo taisyklė	Pranešimo-apie-remonto-pabaigą-siuntimas turi būti uždraustas, jeigu nėra žinomas kliento telefono numeris.
8.	<i>b15</i> , numatymo taisyklė	Kiekviena užsakymo prekė, kurios kiekis nenurodytas, turi būti ištrinta, kai užsakymas yra uždaromas.
10.	<i>b17</i> , numatymo taisyklė	Remonto-sutarities-numeris turi būti pateikiamas ekrane raudonai, jei šiandienos data yra vėlesne ar lygi sutarties sudarymo datai plus 14 dienų.
11.	<i>b18</i> , numatymo taisyklė	Siųsti-išankstinį-įspėjimą užsakymui turi būti vykdomas kai užsakymas išsiųstas.

Šaltinis: sudaryta autoriaus

Visų likusių VT generavimas atliekamas MS Visio aplinkoje, Roso metodu užrašyti VT skirtu įrankio prototipu [14].

3.4 Specifikuotų veiklos taisyklių transformavimas pagal sukurtąją metodiką

Imituojamas automatizuotas sudarytų VT generavimas MS Visio įskiepyje, pritaikytame modeliuoti VT Roso metodu. Programos langas matomas 28 paveikslėlyje.



28 pav. Naudojamo įrankio prototipo VT braižymo aplinka

Atliekamas kiekvienos Roso metodu automatizuotai generuojamos ir jau pilnai sugeneruotos VT diagramos analizavimas.

VT 1 transformavimas. Užrašant natūraliąja kalba VT elementų priskyrimas šablono elementams pavaizduotas 5 lentelėje.

Lentelė 5

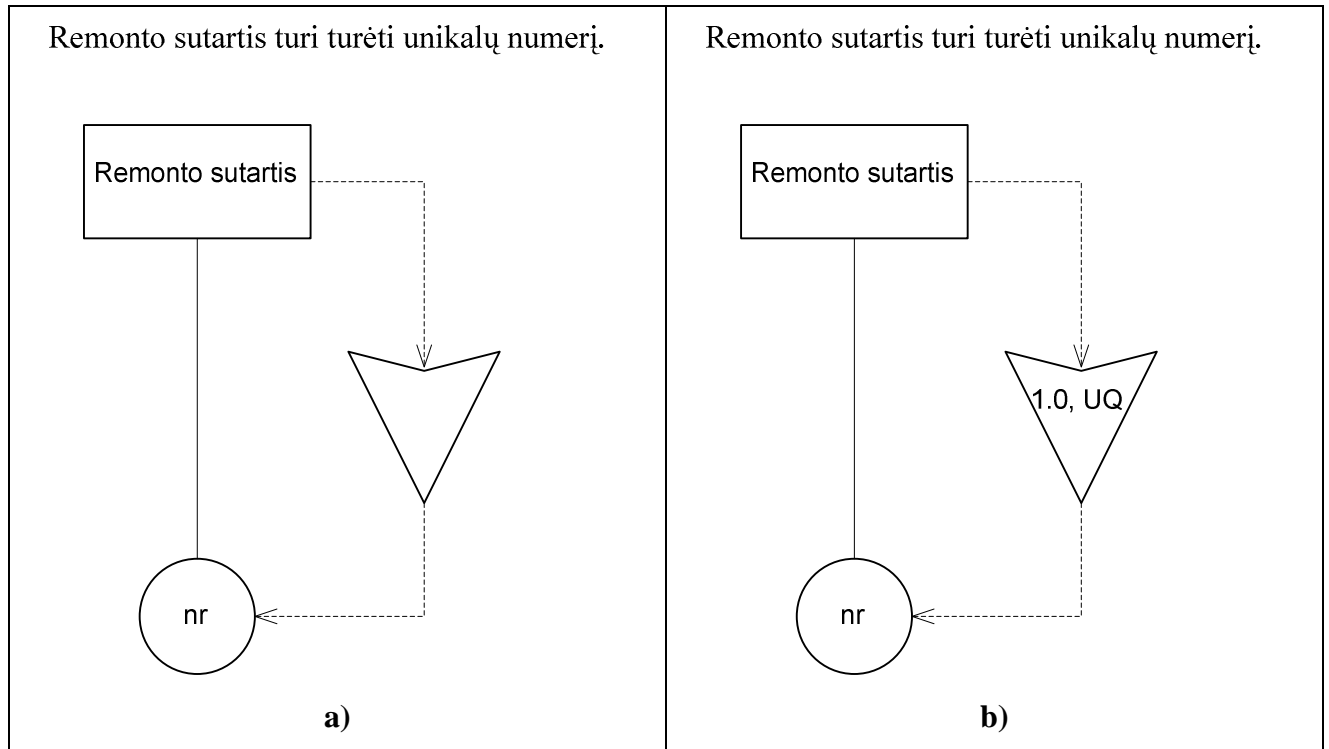
1 VT susiejimas su pagrindiniais šablono elementais

1. Remonto sutartis turi turėti unikalų numerį	
Taisyklė	Šablono elementas
Šablonas: b1.	
Remonto sutartis	subjektas
turi turėti	raktažodis
unikalų numerį	faktas

Šaltinis: sudaryta autoriaus

Pagal metodiką, generuojant VT, į ekraną išvedamas subjektas, integralumo apribojimas, faktas – atitinkami duomenų modelio elementai, kurie buvo susieti su šablono elementais specifikavimo metu.

Automatizuoto generavimo ir pilnai sugeneruotos VT diagramos Roso metodu pateikiamos 29 paveikslėlyje.



Šaltinis: sudaryta autoriaus

29 pav. VT 1 diagrama sugeneruota automatiškai (a) ir pilnai užbaigta (b)

Matome, kad ši taisyklė buvo sugeneruota gan tiksliai. Išvesti visi naudojami duomenų modelio elementai – esybė remonto sutartis ir jos atributas nr. Parinktas apribojimo simbolis, nes jokia sąlyga taisyklėje nėra tikrinama. BRS RuleSpeak šablono subjekto duomenų modelio elementas Remonto sutartis priskirtas bazei. Duomenų modelio elementas faktas, kuris čia yra esybės Remonto sutartis atributas nr, priskiriamas korespondentui. Buvo pateikti 6 grafiniai elementai – tiek kiek ir turėjo būti baigtoje diagramoje. Automatiškai generuojant nebuvo parinktas joks VT tipas. Kadangi VT pagal BRS RuleSpeak yra atmetimo tipo, tai Roso modelyje ji gali būti bet kuri iš nurodytų metodikoje. Projektuotojas, pabaigdamas braižyti diagramą turėtų tik parinkti tinkamą tipą iš siūlomų. Šiuo atveju VT yra unikalios tipo apribojimas.

Taisyklė 1.0

Bazė: duomenų tipas, esybė Remonto sutartis.

Korespondentas: duomenų tipas, atributas nr.

VT tipas: unikalus.

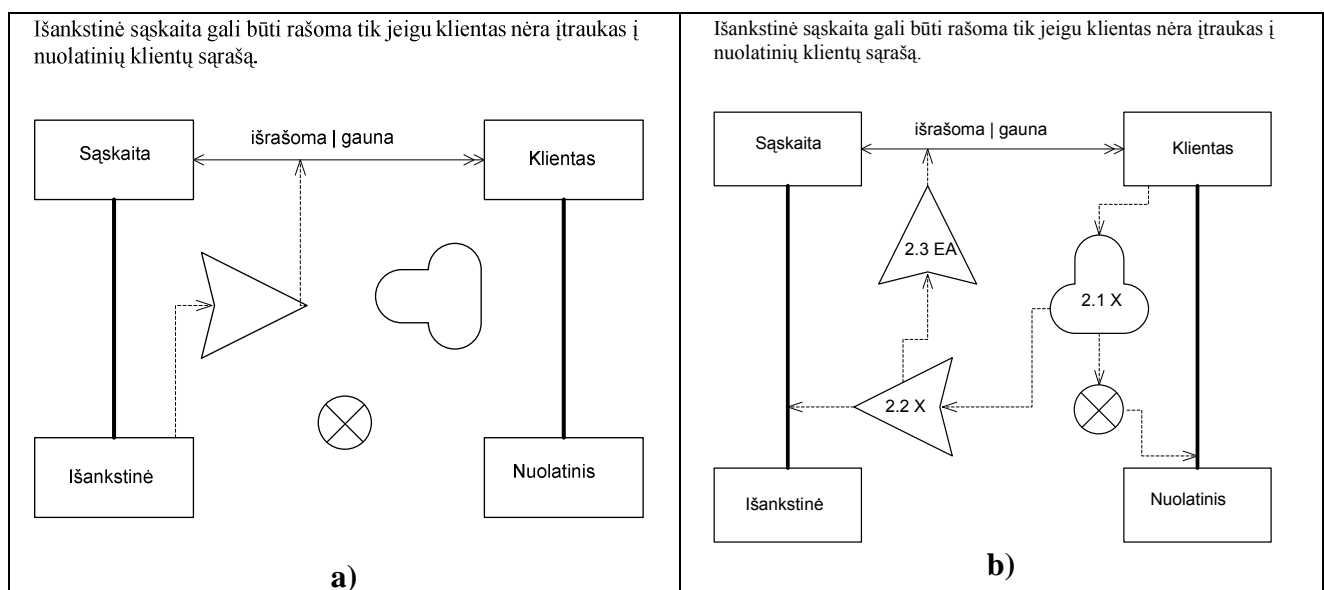
VT 2 transformavimas. Užrašant natūraliąja kalba VT elementų priskyrimas šablono elementams pavaizduotas 6 lentelėje.

Lentelė 6

2 VT susiejimas su pagrindiniais šablono elementais

2. Išankstinė sąskaita gali būti rašoma tik jeigu klientas nėra įtrauktas į nuolatinių klientų sąrašą.	
Taisyklė	Šablono elementas
Šablonas: b2.	
Išankstinė sąskaita	subjektas
gali būti išrašoma	faktas, ryšys
tik	raktažodis
jeigu klientas nėra įtrauktas į nuolatinių klientų sąrašą.	sąlyga

Šaltinis: sudaryta autoriaus



Šaltinis: sudaryta autoriaus

30 pav. VT 2 diagrama sugeneruota automatiškai (a) ir pilnai užbaigta (b)

Šios VT automatizuotas generavimas tokių gerų rezultatų, kaip ankstesniame pavyzdyje, nedavė. Pradinėje diagramoje pateikiama 2 esybės ir 2 jų subtipai, sujungti ISA ryšiais, ryšys, jungiantis esybes, sąlygos bei apribojimo elementai. Automatinio generavimo metu ekrane pateikti 12 grafinių elementų. Galutinėje diagramoje jų yra 18. Kaip matome, pateikiami visi duomenų modelio elementai, keletas taisyklių simbolių. Tačiau pati pagrindinė užduotis – tinkamas korespondento ir bazės nustatymas, bei VT tipo parinkimas nėra išgaunamas. Pirminis variantas, kai pagal metodiką baze parenkamas subjektas ir korespondentu ryšys, čia pasiteisina tik iš dalies. Taisyklės skaidymas į atomines Roso, teisingas jų numeravimas, t.y. loginės sekos numatymas, yra atliekamas projektuotojo. Po šių veiksmų, VT 2 išskaidoma į 3 atomines Roso taisykles. Vienos iš jų korespondentu tampa ryšys, jungiantis esybes Sąskaita ir Klientas.

Taisyklė 2.1 Bazė: duomenų tipas, esybė Klientas.

Korespondentas: duomenų tipas, ISA ryšys tarp esybės Klientas ir jos subtipo (taip pat esybės) Nuolatinis.

Yra išorinė išeigos reikšmė.

VT tipas: privaloma.

Taisyklė 2.2

Bazė: veiklos taisyklė 2.1.

Korespondentas: veiklos taisyklė 2.3.

VT tipas: privaloma.

Taisyklė 2.3

Bazė: veiklos taisyklė 2.2.

Korespondentas: duomenų tipas, sąryšis išrašoma.

VT tipas: įgalinama.

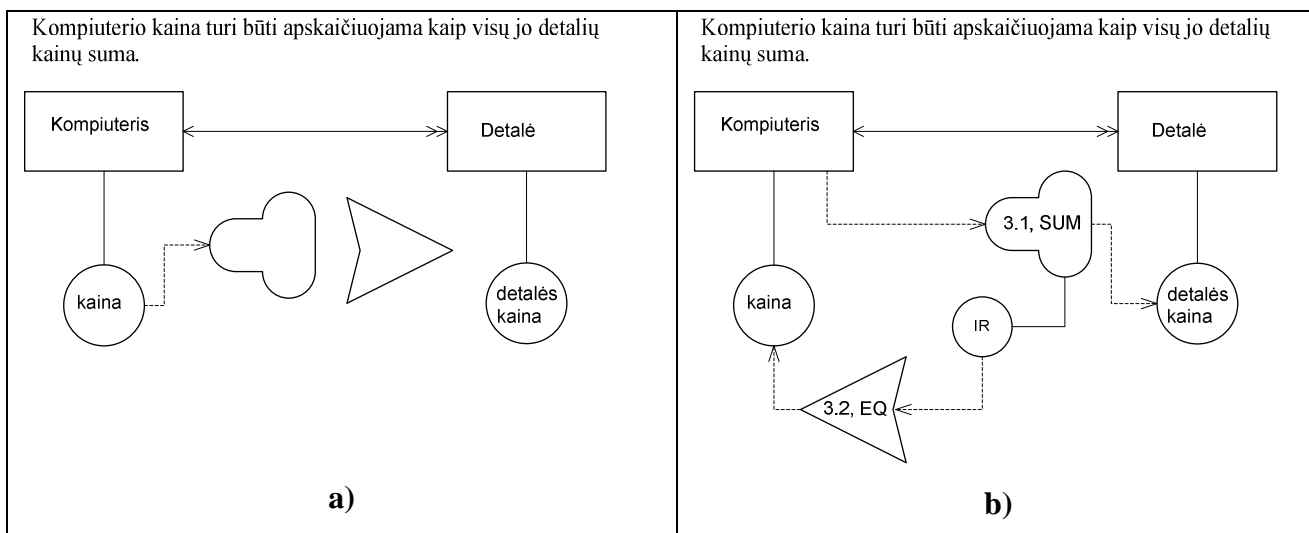
VT 3 transformavimas. Užrašant natūraliąja kalba VT elementų priskyrimas šablono elementams pavaizduotas 7 lentelėje.

Lentelė 7

3 VT susiejimas su pagrindiniais šablono elementais

3. Kompiuterio kaina turi būti apskaičiuojama kaip visų jo detalių kainų suma.	
Taisyklė	Šablono elementas
Šablonas: b6.	
Kompiuterio kaina	subjektas
turi būti apskaičiuojama kaip	raktažodis
Visų jo sudedamųjų dalių kainų suma	matematinė formulė

Šaltinis: sudaryta autoriaus



Šaltinis: sudaryta autoriaus

31 pav. VT 2 diagrama sugeneruota automatiškai (a) ir pilnai užbaigta (b)

Generavimo metu, kaip ir buvo tikėtasi, į ekraną išvesti visi VT naudojami duomenų elementai, taip pat salygos ir apribojimo simboliai, kurių panaudojimą aprašo metodika. Baze nustatytas subjektas duomenų elementas `kaina`, kuris yra esybės `Kompiuteris` atributas. Tačiau, pabaigus braižyti dalinai sugeneruotą diagramą, paaiškėja, kad pagal metodiką numatytas subjektas tampa ne VT baze, o antros atominės Roso taisyklės (3.2) korespondentu.

VT 3 generavimo metu į ekraną buvo išvesta 10 grafinių elementų iš 15. Nei vienam taisyklės simboliui nėra priskirta VT tipas. Bazės egzemplioriaus parinkimas neturi prasmės, kadangi galutinėje VT diagramoje jo nebelieka.

Taisyklė 3.1 Bazė: duomenų tipas, esybė `Kompiuteris`.

Korespondentas: duomenų tipas, esybės `Detalė` atributas `detalės kaina`. Yra išorinė išeišios reikšmė.

VT tipas: susumuota.

Taisyklė 3.2 Bazė: veiklos taisyklės 3.1 išeišios reikšmė.

Korespondentas: duomenų tipas, esybės `Kompiuteris` atributas `kaina`.

VT tipas: lygu.

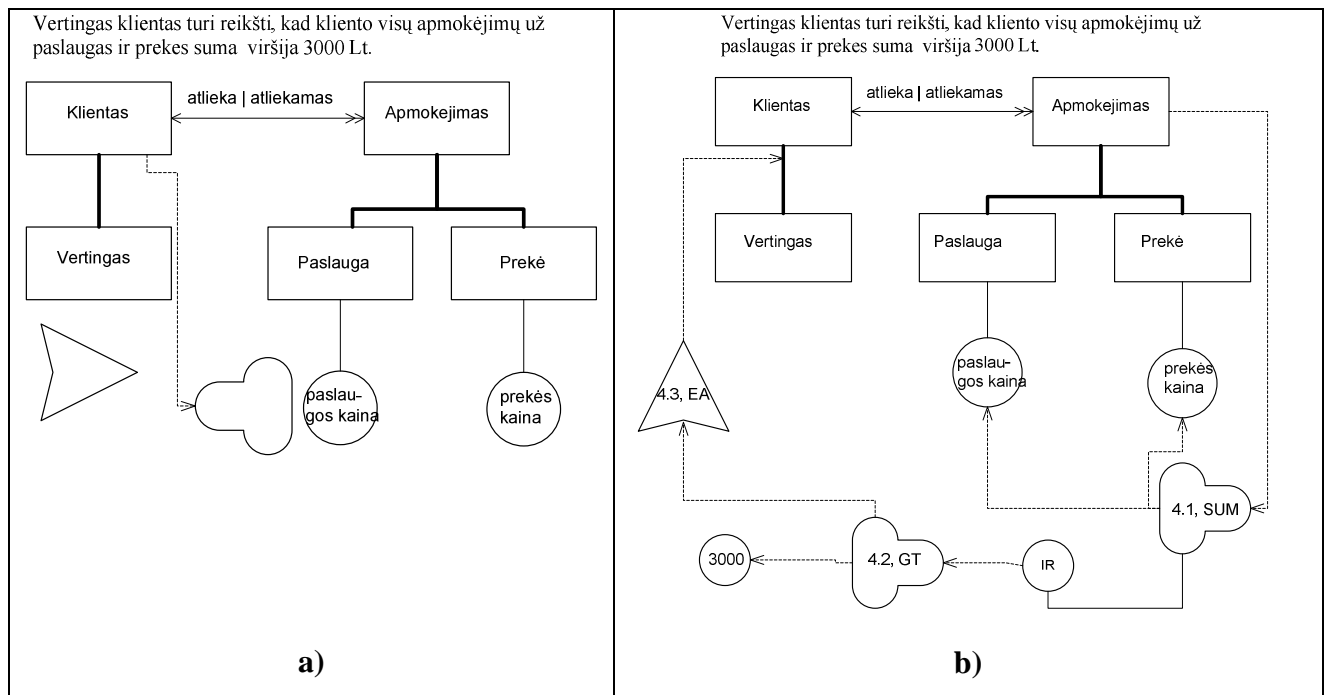
VT 4 transformavimas. Užrašant natūraliaja kalba VT elementų priskyrimas šablono elementams pavaizduotas 8 lentelėje.

Lentelė 8

4 VT susiejimas su pagrindiniais šablono elementais

4. Vertingas klientas turi reikšti, kad kliento visų apmokėjimų už paslaugas ir prekes suma viršija 3000 Lt.	
Taisyklė	Šablono elementas
Šablonas: b8.	
Vertingas klientas	subjektas
turi reikšti, kad	raktažodis
kliento visų apmokėjimų už paslaugas ir prekes suma viršija 3000 Lt	loginė išraiška

Šaltinis: sudaryta autoriaus



Šaltinis: sudaryta autoriaus

32 pav. VT 4 diagrama sugeneruota automatiškai (a) ir pilnai užbaigta (b)

VT 4 generavimo diagramoje matome 14 į ekraną išvestų grafinių elementų, po shemos pilno nubraižymo jų turi būti 26. VT loginė reikšmė įgauna prasmę tik tada, kai projektuotojas ją užbaigia. Automatizuotas generavimas įveda klaidingą ryšį - esybę Klientas priskiria bazei, kai iš tikrųjų, kaip paaiškėja po pilno VT diagramos nubraižymo, bazei tampa ISA ryšys jungiantis esybę Klientas ir jos subtipą Vertingas. Ši klaida ištaisoma projektuotojo. Stebint tokią situaciją, jau galima daryti prielaidą, kad metodikos sprendimas subjekto, duomenų elementą pagal BRS RuleSpeak, transformuoti į bazės elementą Roso modelyje yra ne visada teisingas.

Po diagramos užbaigimo VT 4 išskaidoma į 3 atominės Roso modelio taisykles.

Taisyklė 4.1 Bazė: duomenų tipas, esybė Apmokėjimas.

Korespondentas: duomenų tipas, esybės Paslauga atributas paslaugos kaina ir esybės Prekė atributas prekės kaina.

VT tipas: susumuota.

Yra išorinė išeigos reikšmė.

Taisyklė 4.2 Bazė: veiklos taisyklės 4.1 išeigos reikšmė.

Korespondentas: konstanta 3000 Lt.

VT tipas: daugiau-už.

Taisyklė 4.3 Bazė: veiklos taisyklė 4.2.

Korespondentas: duomenų tipas, ISA ryšys tarp esybių Klientas ir Vertingas.

VT tipas: įgalinta.

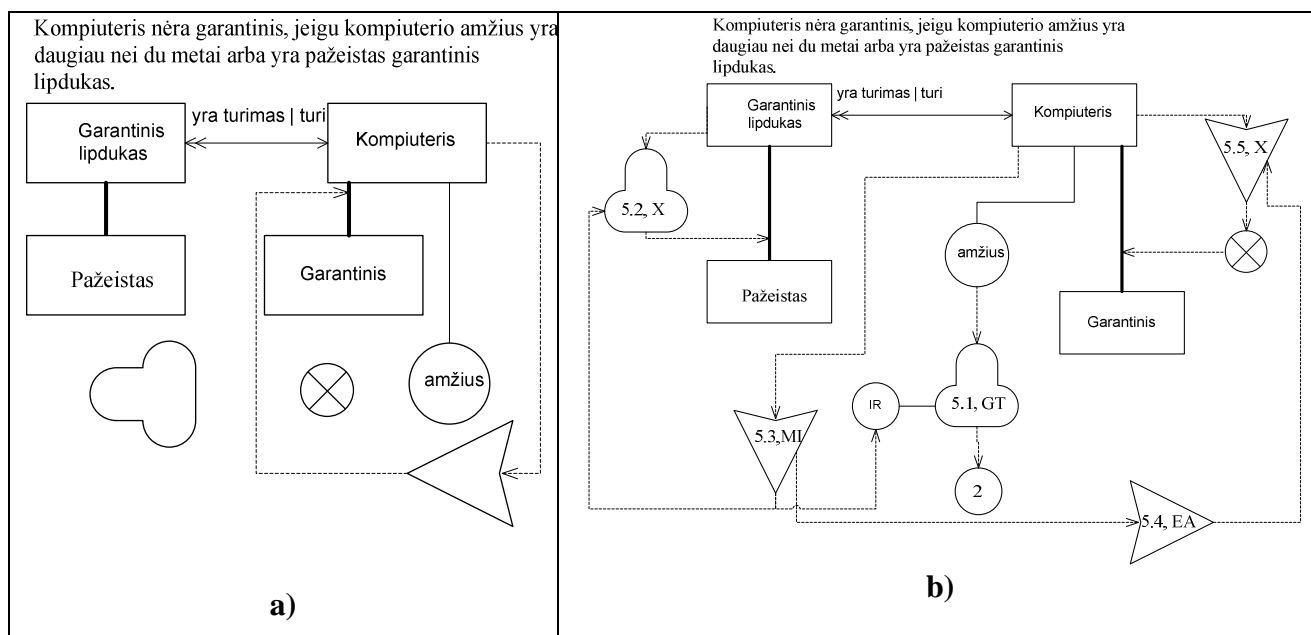
VT 5 transformavimas. Užrašant natūraliąja kalba VT elementų priskyrimas šablono elementams pavaizduotas 9 lentelėje.

Lentelė 9

5 VT susiejimas su pagrindiniais šablono elementais

5. Kompiuteris nėra garantinis, jeigu kompiuterio pagaminimo data yra daugiau nei dviem metais ankstesnės už šiandienos datą arba yra pažeistas garantinis lipdukas.	
Taisyklė	Šablono elementas
Šablonas: b11.	
Kompiuteris	subjektas
nėra	raktažodis
garantinis	terminas
jeigu kompiuterio pagaminimo data yra daugiau nei dviem metais ankstesnės už šiandienos datą arba yra pažeistas garantinis lipdukas	sąlyga

Šaltinis: sudaryta autoriaus



Šaltinis: sudaryta autoriaus

33 pav. VT 5 diagrama sugeneruota automatiškai (a) ir pilnai užbaigta (b)

Atlikus pilną VT 5 diagramos nubraižymą, matoma, kad VT susideda iš 5 atominių Roso modelio taisyklių. Automatinis generavimas numatė tik 2 – apribojimą ir sąlyga. Nubraižyta 13 grafinių elementų iš diagramai reikiamų 28. Projektuotojo nustatyta, kad modeliuojamoje VT yra privalomo, bendrai-susiejančio, įgalinimo ir palyginamojo įvertinimo atominių taisyklių tipai. Automatiškai baze parinktas duomenų modelio elementas Kompiuteris, terminas Garantinis priskirtas kaip korespondentas. Vienoje atominėje Roso taisyklėje šis priskyrimas

išlieka tik papildomai įterpiamas pasiūlytas neigimo simbolis, kuris paneigia VT tipą privaloma ir taisyklė skaitoma „nėra“.

Taisyklė 5.1 Bazė: duomenų tipas, esybės Kompiuteris atributas amžius.
Korespondentas: konstanta 2 metai.

Yra išorinė išeiigos reikšmė.

VT tipas: daugiau-už.

Taisyklė 5.2 Bazė: duomenų tipas, esybė Garantinis lipdukas.

Korespondentas: duomenų tipas, ISA ryšis tarp esybės Garantinis lipdukas ir jos subtipo Pažeistas.

VT tipas: privaloma.

Taisyklė 5.3 Bazė: duomenų tipas, esybė Kompiuteris.

Korespondentas: veiklos taisyklė 5.2. ir veiklos taisyklės 5.1. išeiigos reikšmė.

VT tipas: bendrai-susiejanti.

Taisyklė 5.4 Bazė: veiklos taisyklė 5.3.

Korespondentas: veiklos taisyklė 5.5.

VT tipas: įgalinta.

Taisyklė 5.5 Bazė: duomenų tipas, esybė Kompiuteris.

Korespondentas: duomenų tipas, ISA ryšis tarp esybės Kompiuteris ir jos subtipo Garantinis.

VT tipas: privaloma.

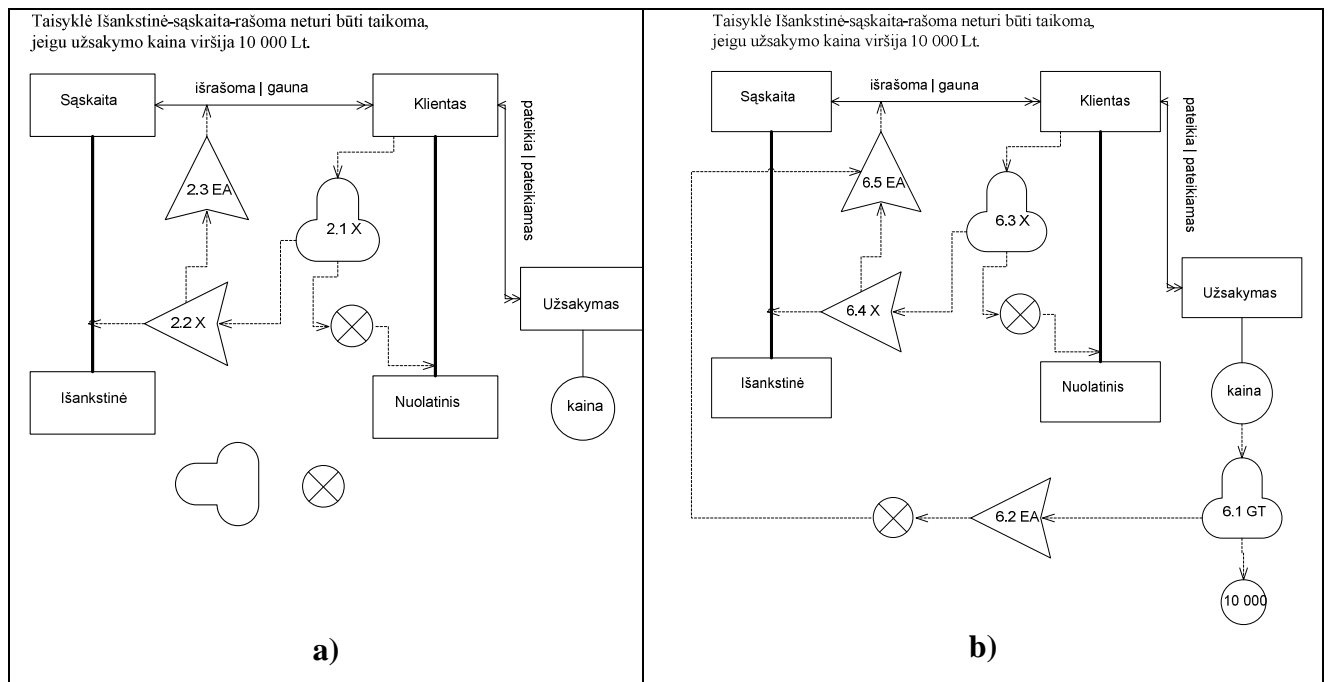
VT 6 transformavimas. Užrašant natūraliaja kalba VT elementų priskyrimas šablono elementams pavaizduotas 10 lentelėje.

Lentelė 10

6 VT susiejimas su pagrindiniais šablono elementais

6. Taisyklė Išankstinė-sąskaita-rašoma neturi būti taikoma, jeigu užsakymo kaina viršija 10 000 Lt.	
Taisyklė	Šablono elementas
Šablonas: b13.	
Taisyklė Išankstinė-sąskaita-rašoma	taisyklės pavadinimas
neturi būti taikoma	raktažodis
jeigu užsakymo kaina viršija 10 000 Lt	sąlyga

Šaltinis: sudaryta autoriaus



Šaltinis: sudaryta autoriaus

34 pav. VT 6 diagrama sugeneruota automatiškai (a) ir pilnai užbaigta (b)

Generuojant VT 6 į ekraną išvedama VT 2 (Išankstinė-sąskaita-rašoma), sąlygos elementas, duomenų modelio elementai, susieti su sąlyga, specialusis elementas. Iš viso pateikiama 12 grafinių elementų iš 30 reikiamų. Tai yra tik 40 procentų diagramos grafinės informacijos. Atliekant VT 6 generavimą, pasikeitė naudojamos VT 2 numeracija.

Taisyklė 6.1 Bazė: duomenų tipas, esybės Užsakymas atributas kaina.

Korespondentas: konstanta 10 000 Lt.

VT tipas: daugiau-už.

Taisyklė 6.2 Bazė: veiklos taisyklė 6.1.

Korespondentas: veiklos taisyklė 6.5.

VT tipas: įgalinta.

Taisyklė 6.3 Bazė: duomenų tipas, esybė Klientas.

Korespondentas: duomenų tipas, ISA ryšys tarp esybės Klientas ir jo subtipo Nuolatinis.

VT tipas: privaloma.

Taisyklė 6.4 Bazė: veiklos taisyklė 6.3.

Korespondentas: duomenų tipas, ISA ryšys tarp esybės Sąskaita ir jos subtipo Išankstinė.

VT tipas: privaloma.

Taisyklė 6.5 Bazė: veiklos taisyklė 6.4.

Korespondentas: duomenų tipas, ryšys tarp esybės Sąskaita ir esybės Klientas.

VT tipas: įgalinta.

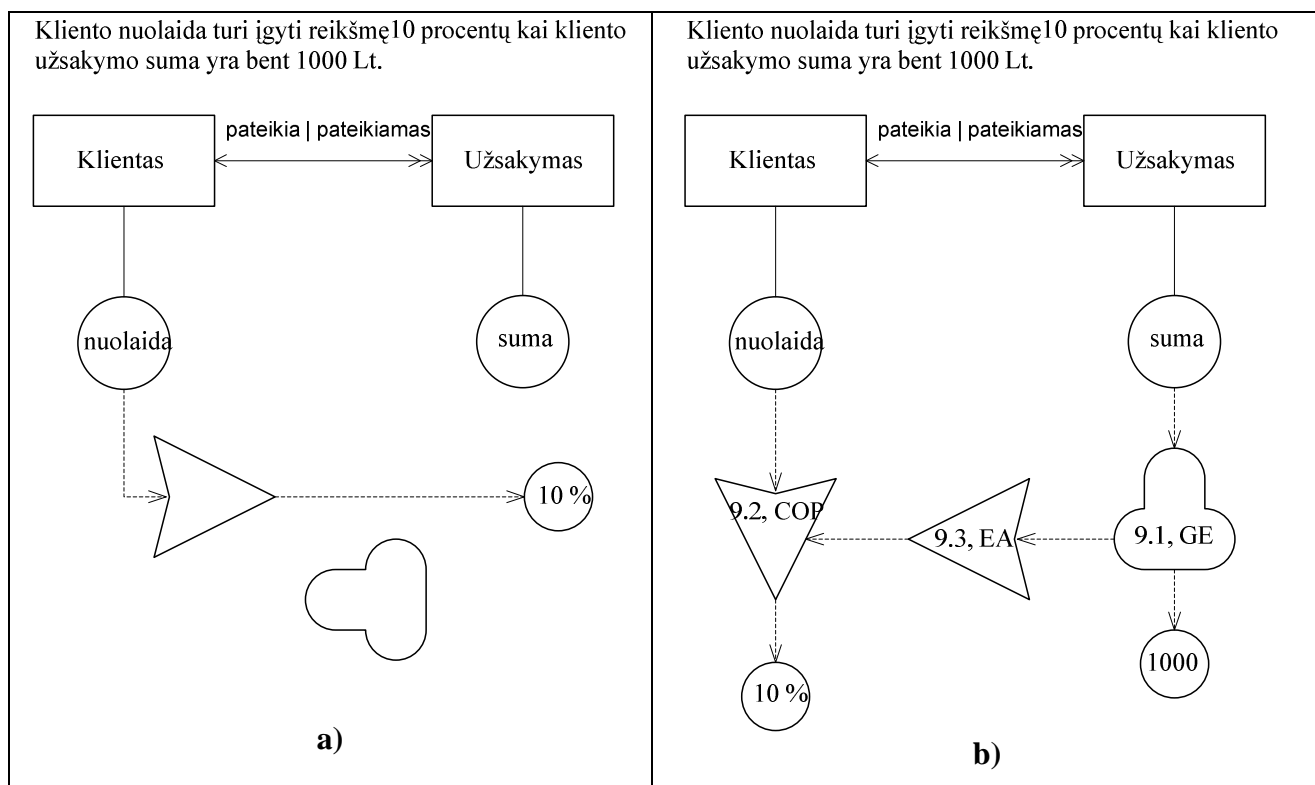
VT 9 transformavimas. Užrašant natūraliąja kalba VT elementų priskyrimas šablono elementams pavaizduotas 11 lentelėje.

Lentelė 11

9 VT susiejimas su pagrindiniais šablono elementais

Taisyklė	Šablono elementas
9. Kliento nuolaida turi įgyti reikšmę 10 procentų kai kliento užsakymo suma yra bent 1000 Lt.	
Šablonas: b19.	
Kliento nuolaida	subjektas
įgyti reikšmę	raktažodis
10 procentų	reikšmė
kai kliento užsakymo suma yra bent 1000 Lt	sąlyga

Šaltinis: sudaryta autoriaus



Šaltinis: sudaryta autoriaus

35 pav. VT 9 diagrama sugeneruota automatiškai (a) ir pilnai užbaigta (b)

VT 9 diagramos generavimo metu į ekrana pateikta 12 grafinių elementų, subjektas esybės Klientas atributas nuolaida priskirtas bazei, reikšmė – korespondentui. Pabaigus braižyti diagramą, grafinių simbolių skaičius padidėjo iki 18. Bazė ir korespondentas nebuvo pakeisti, tačiau

jie priskirti antrajai atominei Roso taisyklei. Generuojant šią VT dalis metodikos pasiteisino – teisingai parinkti bazės ir korespondento duomenų modelio elementai. Tačiau kaip ir anksčiau generuotose VT diagramose, VT tipas nėra parenkamas automatiškai. Projektuotojas gali rinktis tik iš dalinai susiaurintų tipų grupių.

Taisyklė 9.1 Bazė: duomenų tipas, esybės Užsakymas atributas suma.
Korespondentas: konstanta 1000 Lt.
VT tipas: daugiau-arba-lygu.

Taisyklė 9.2 Bazė: duomenų tipas, esybės Klientas atributas nuolaida.
Korespondentas: konstanta 10 procentų.
VT tipas: nukopijuota.

Taisyklė 9.3 Bazė: veiklos taisyklė 9.1.
Korespondentas: veiklos taisyklė 9.3.
VT tipas: įgalinta.

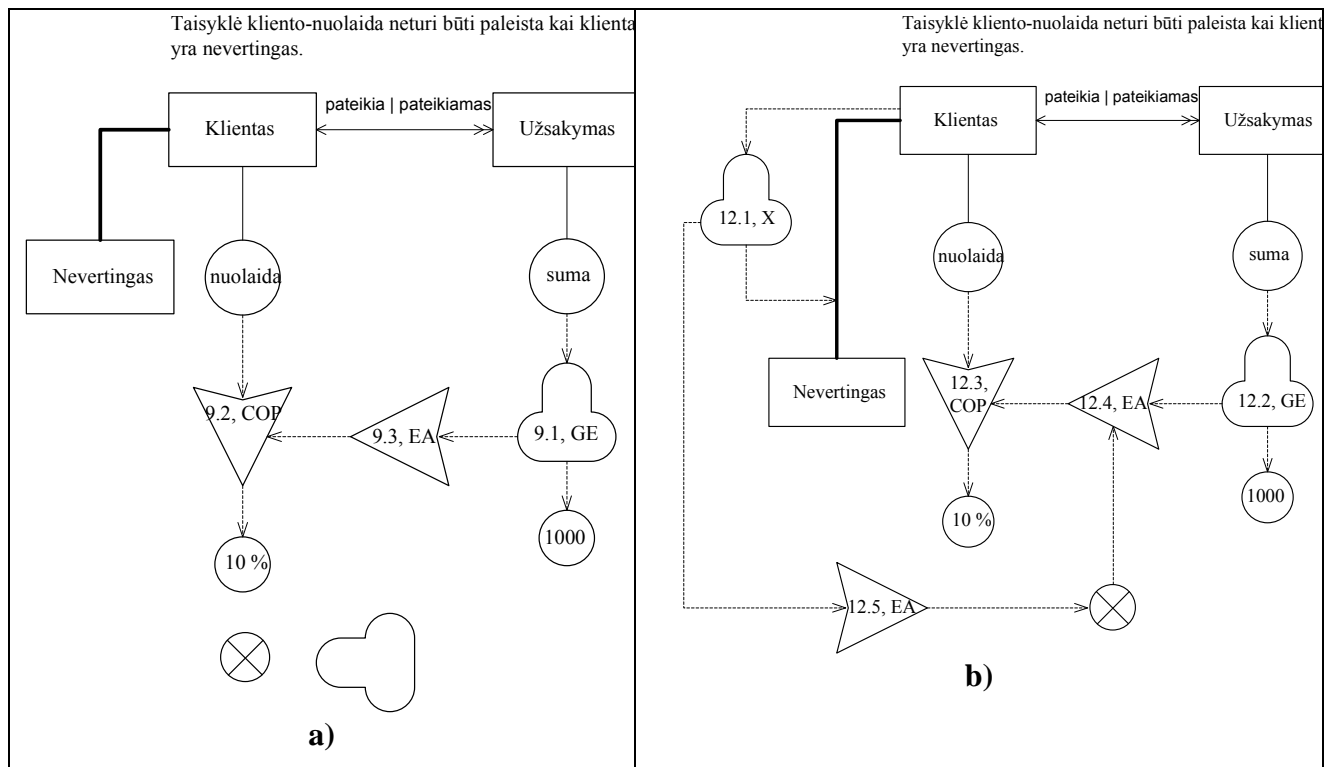
VT 12 transformavimas. Užrašant natūraliąja kalba VT elementų priskyrimas šablono elementams pavaizduotas 12 lentelėje.

Lentelė 12

12 VT susiejimas su pagrindiniais šablono elementais

12. Taisyklė kliento-nuolaida neturi būti paleista kai klientas yra nevertingas.	
Taisyklė	Šablono elementas
Šablonas: b19.	
Taisyklė kliento-nuolaida	taisyklės pavadinimas
neturi būti paleista	raktažodis
kai klientas yra nevertingas	sąlyga

Šaltinis: sudaryta autoriaus



Šaltinis: sudaryta autoriaus

36 pav. VT 12 diagrama sugeneruota automatiškai (a) ir pilnai užbaigta (b)

VT 12 diagramos generavimo metu pateikta anksčiau sugeneruota VT 9 diagrama, kurios paleidimas esant tam tikroms aplinkybėms yra negalimas, sąlygai priskirti duomenų modelio elementai, kurių nėra VT 9 – ISA ryšys tarp esybės *Klientas* ir jos subtipo *Nevertingas*, pagal raktažodį „neturi“ – neigimo specialusis elementas, sąlygos elementas. Rankiniu būdu pabaigus diagramos braižymą paaiškėja, kad reikalingas dar vienas apribojimo elementas, kuris uždraudžia paleisti anksčiau numatytos taisyklės VT 9, kurios elementų numeracija pakeista, vykdymą. Be VT 9 priklausančių 18 grafinių elementų į ekrana yra pateikiami dar 4 grafiniai elementai. Pabaigus VT 12 diagramos braižymą, grafinių elementų skaičius tampa 26. VT 9 priklausę elementuose įrašyti VT tipai lieka nepakitę.

Taisyklė 12.1 Bazė: duomenų tipas, esybė *Klientas*.

Korespondentas: duomenų tipas, ISA ryšys tarp esybės *Klientas* ir jos subtipo *Nevertingas*.

VT tipas: privaloma.

Taisyklė 12.2 Bazė: duomenų tipas, esybės *Užsakymas* atributas *suma*.

Korespondentas: konstanta 1000 Lt.

VT tipas: daugiau-arba-lygu.

Taisyklė 12.3 Bazė: duomenų tipas, esybės *Klientas* atributas *nuolaida*.

Korespondentas: konstanta 10 procentų.

	VT tipas: nukopijuota.
<u>Taisyklė 9.4</u>	Bazė: veiklos taisyklė 12.2. Korespondentas: veiklos taisyklė 12.3. VT tipas: įgalinta.
<u>Taisyklė 9.5</u>	Bazė: veiklos taisyklė 12.1. Korespondentas: veiklos taisyklė 12.4. VT tipas: įgalinta.

3.5 Eksperimento rezultatai

Siekiant įvertinti bandymo rezultatus, buvo atliekama jų analizė dviem pjūviais:

1. Metodikos kokybinis vertinimas, nustatant vidutinį automatinio Roso modelių generavimo efektyvumą;
2. Metodikos įvertinimas nustatant, kaip ji tenkina bazinius kriterijus bei sprendimui keltus reikalavimus.

3.5.1 Metodikos efektyvumo įvertinimas

Vienas iš kriterijų, leidžiančių vertinti metodikos naudą, tai grafinių elementų išvedimo į ekraną kiekio išpildymas. Kaip žinoma, VT skaidymą į atomines Roso taisykles gali atlikti tik projektuotojas – joks įrankis to nenumato, kadangi VT skaidymas yra loginio mąstymo reikalaujantis darbas. Todėl kuo tikslesnis diagramos užpildymas būtų privalumas. Analizuotas eksperimento metu atliktų VT diagramų generavimo metu gautų kiekybinių charakteristikų kiekis, t.y. pateiktų po automatinio generavimo ir po pilno VT diagramos sudarymo atvaizduotų grafinių elementų skaičius.

Grafiniu elementu vadinamas kiekvienas MS Visio aplinkoje nepriklausomas grafinis elementas: bet kuris objektas - integralumo apribojimas, sąlyga, atributas, esybė, konstanta, išeigos reikšmė, bet kuris ryšys - atributo ryšys, taisyklės ryšys, ryšys daug-su-daug, ryšys vienas-su-daug, ISA ryšys, bet kuris specialusis elementas.

Lentelė 13

Grafinių elementų pateikimo įvertinimas

VT	Sugeneruota grafinių elementų	Iš viso grafinių elementų	Generavimo našumas, %
VT 1	6	6	100,00

13 lentelės tęsinys

VT	Sugeneruota grafinių elementų	Iš viso grafinių elementų	Generavimo našumas, %
VT 2	12	18	66,67
VT 3	10	15	66,67
VT 4	14	26	53,85
VT 5	13	28	46,43
VT 6	12	30	40,00
VT 9	12	18	66,67
VT 12	12	26	46,15
Iš viso:			60,80

Šaltinis: sudaryta autoriaus

Atlikus analizę, nustatyta, kad nagrinėtuose pavyzdžiuose vidutiniškai automatiškai pateikiama apie 60 procentų grafinių elementų. Taigi daugiau nei pusė visų reikalingų. Kita dalis lieka projektuotojo valioje.

Reikia paminėti, kad į šį vertinimą nėra įtrauktos veiklos taisyklės, kurios nebuvo braižomos Roso metodu. Įvertinimui galima paskaičiuoti, kad iš 12 testinių taisyklių, generuojamos buvo tik 8, t.y. 66,67 proc. Galima teigti, kad metodikos našumas šiuo (testiniu) atveju yra tik apie 66 procentus.

Norint gauti tikslesnį metodikos įvertinimą - Roso metodu generuojamų ir negeneruojamų veiklos taisyklių santykį – reikėtų panagrinėti realios (galbūt ir pakankamai didelės) sistemos veiklos taisyklių projektavimo galimybes. Tačiau tai nėra šio darbo dalis.

3.5.2 Eksperimento rezultatų apibendrinimas

Tyrimo metu buvo siekiama įvertinti sukurtos metodikos naudą atliekant dalinai automatizuotą perėjimą iš natūraliosios kalbos šablonais pagal BRS RuleSpeak užrašytų VT realizavimą Roso metodu.

Metodikos įvertinimas pateikiamas 14 lentelėje. Vertinimas trijų balų sistemoje, kur: 1 – blogai, 2 – vidutiniškai, 3 – gerai.

- Duomenų elementų, susietų su taisyklės subjektu, išvedimas į ekraną. Priskyrimas bazei;
- Nustatymas, ar Roso VT modelyje taisyklė yra sąlyga, ar apribojimas;
- Duomenų elementų, susietų su VT faktųterminų išvedimas į ekraną (tik tuo atveju, jeigu tie duomenys dar nėra išvesti). Priskyrimas korespondentui;
- Tinkamų Roso VT tipų išskyrimas;

- Specialiųjų elementų poreikio identifikavimas (pagal raktinį VT žodį ar VT tipą);
- Sąlygoje naudojamų duomenų modelio elementų išvedimas į ekraną.

Lentelė 14

Metodikos įvertinimas

Kriterijus	Įvertinimas
Duomenų elementų, susietų su VT subjektu išvedimas į ekraną	3
Nustatymas, ar Roso VT modelyje taisyklė yra sąlyga, ar apribojimas	2
Duomenų elementų, susietų su VT faktųterminų išvedimas į ekraną	3
Tinkamų Roso VT tipų išskyrimas	1
Specialiųjų elementų poreikio identifikavimas	2
Sąlygoje naudojamų duomenų modelio elementų išvedimas į ekraną	3
VT bazes nustatymas	1
VT korespondento nustatymas	1

Šaltinis: sudaryta autoriaus

Kaip žinoma, vienas iš trijų pagrindinių ir visada privalomų VT atvaizdavimo Roso metodu elementų yra VT sąlygos ar apribojimo simbolis su nurodytu VT tipu. Jeigu bent vieną VT simbolių generuojamai diagramai įmanoma parinkti automatizuotai, tai VT tipo ne. Programa tik galėtų susiaurinti, bet ir tai labai nežymiai, VT tipų grupes, kurios būtų galimos panaudoti pagal konkretų šabloną. VT skaidymas į atomines Roso VT ir VT tipo parinkimas priklauso tik nuo projektuotojo. Vien dėl šios savybės pilnai automatizuoti generavimo procesą yra neįmanoma.

Kaip aptarta šiame darbe programiškai gali būti realizuojamas didžios dalies grafinių elementų išvedimas į ekraną. Bet projektuotojas vistiek gaišta laiką aiškindamasis ko dar gali trūkti toje dalinai sugeneruotoje diagramoje. Galbūt projektuotojui netgi greičiau ir aiškiau viskas būna kai diagramos braižymą atlieka pats ir nuo pradžių. Analizuojant dalinai sugeneruotą diagramą dėl klaidų atsiradimo, pavyzdžiui, kai netinkamai parenkami bazė ir korespondentas, galima įverti dar daugiau klaidų, kurių paieška ir taisymas užimtų nemažai laiko. Nors atsižvelgiant į tai, kad didžioji neišpildytų grafinių elementų dalis yra taisyklės ryšys, t.y. ryšys tarp bazės, korespondento, specialiųjų simbolių ir taisyklės simbolių – sąlygos ar apribojimo, galima teigti, kad duomenų modelio elementų išvedimas į ekraną turėtų įtakos darbo spartos pagreitėjimui.

Prototipinę realizaciją pasiūlytos metodikos galima būtų sukurti esamų ir tyrimo metu naudotų prototipų pagrindu. Reikalavimų specifikavimo įrankyje reikėtų sukurti naują funkcionalumą, kad pasirinkus VT atsidarytų MS Visio langas, kuriame būtų matoma taisyklė. Ją būtų galima papildyti ir išsaugoti.

Atsižvelgiant į pastebėjimus, daromus po eksperimento, galima daryti išvadą, kad metodikos realizavimas prototipiniu įrankiu perėjimą nuo reikalavimų specifikavimo prie projektavimo iš

dalis palengvintų. Paspirtėtų darbas dėl duomenų modelio elementų parinkimo nagrinėjamai VT.
Projektuotojui liekantys uždaviniai: VT skaidymas į atomines Roso taisykles, VT tipų parinkimas.

IŠVADOS

1. Struktūrizuojant veiklos taisykles IS kūrimo metu, jos įgyja du pavidalus: struktūrizuotos natūraliosios kalbos ir formalųjį. Taisyklių išraiškos formaliais metodais turėtų būti paveldimos iš pirminio jų varianto, t.y. užrašytų natūraliosios kalbos šablonais, tačiau metodikų, nusakančių, kaip tai atlikti, neaptikta. Formaliai išreikštos taisyklės yra saugomos VT saugykloje ir yra tinkamos IS realizavimui.

2. Atlikus preliminarią pagal BRS RuleSpeak užrašytų VT transformavimo į Roso taisyklių diagramas analizę nustatyta, jog tokia transformacija negalima pagal keturis iš devyniolikos galimų BRS RuleSpeak šablonų užrašytoms taisyklėms.

3. Apibendrinus analizės rezultatus, apibrėžta modifikuotus BRS RuleSpeak ir Roso metodą palaikanti veiklos taisyklių transformavimo metodika, susidedanti iš 6 sąlyginių žingsnių:

- Duomenų elementų, susietų su taisyklės subjektu, išvedimas į ekraną. Priskyrimas bazei;
- Nustatymas, ar Roso VT modelyje taisyklė yra sąlyga, ar apribojimas;
- Duomenų elementų, susietų su VT faktų/terminų išvedimas į ekraną (tik tuo atveju, jeigu tie duomenys dar nėra išvesti). Priskyrimas korespondentui;
- Tinkamų Roso VT tipų išskyrimas;
- Specialiųjų elementų poreikio identifikavimas (pagal raktinį VT žodį ar VT tipą);
- Sąlygoje naudojamų duomenų modelio elementų išvedimas į ekraną.

4. Atlikus sukurtosios metodikos bandymą realios dalykinės srities pagrindu nustatyta, kad ne visais atvejais metodika gali pasiūlyti teisingas Roso modelio taisyklės bazės ir korespondento reikšmes, tačiau šie trūkumai gali būti nesunkiai identifikuoti projektuotojo.

5. Dėl BRS RuleSpeak modelio savybių tikslus automatinis visų Roso modelyje esančių atominių taisyklių tipų nustatymas dažniausiai nėra įmanomas. Vis tik metodika leidžia efektyviai susiaurinti galimų Roso taisyklių tipų aibę ir tai palengvina transformaciją.

6. Apibendrinus eksperimento rezultatus nustatyta, jog metodika leidžia sugeneruoti vidutiniškai 60 % galutinės Roso taisyklės diagramos grafinių elementų. Tai ženkliai palengvina projektuotojo darbą kuriant pagal BRS RuleSpeak užrašytų VT grafinius modelius.

7. Atlikus sukurtosios metodikos vertinimą pagal kokybinius kriterijus nustatyta, jog sprendimas tenkina pagrindinius jam keltus reikalavimus ir jis gali būti realizuotas tai paspartinant perėjimą nuo taisyklėmis išreikštų reikalavimų prie formalių projektinių modelių.

LITERATŪRA

1. Butleris, R. (2006) Funkcinių reikalavimų specifikuojimas abstrahuoto ir detalizuoto reikalavimų modelių pagrindu. *Informacijos mokslai*, 168 - 177 psl. ISSN 1392-0561.
2. Butleris, R., Kapočius, K. (2001) *Struktūrizuotų veiklos taisyklių saugyklos architektūra*. Informacijos mokslai. Vilnius. Vilniaus universiteto leidykla, 17 tomas, 46-57.
3. Butleris, R., Kapočius, K. (2002) The Business Rules Repository for Information Systems Design. *The 6th East-European Conference ADBIS'2002*. Konferencijos pranešimų medžiaga. ISBN: 80-227-1744-4. Bratislava, Slovakia. Vydavatel'stvo STU, Vol.2, p.64-77.
4. Butleris R., Motiejūnas L. (2006) *Veiklos taisyklių saugykla, integruota su veiklos taisyklių iškvietimo mechanizmu*. Konferencijos pranešimų medžiaga. Kauno Technologijos Universitetas, Kaunas, Lietuva. Duomenų bazės ir modeliai, 2 Sekcija, 11 – 20 psl.
5. Business Rules Group. (2000). Defining Business Rules ~ What Are They Really? (3-rd edition). Iš Business Rules Group interneto puslapio [interaktyvus], [žiūrėta 2008 sausio 05 d.]. Prieiga per internetą: <http://www.businessrulesgroup.org/first_paper/BRG-whatBR_3ed.pdf>
6. Business Rules Group. (2003). Veiklos taisyklių manifestas. Taisyklių nepriklausomumo principai, VR. 2.0. (Redaktorius Ronald G. Ross). Business Rules Group interneto puslapis [interaktyvus], [žiūrėta 2007 lapkričio 24 d.]. Prieiga per internetą: <[http://businessrulesgroup.org/brmanifesto/BRManifestLithuanian\(v1.0\).pdf](http://businessrulesgroup.org/brmanifesto/BRManifestLithuanian(v1.0).pdf)>
7. Čaplinskas, A., Lupeikienė, A., Vasilecas, O. (2002) A Framework to Analyse and Evaluate Information Systems Specification Languages. *Lecture Notes In Computer Science, Vol. 2435. Advances in Databases and Information Systems: 6th East European conference proceedings*. Springer-Verlag Berlin Heidelberg, 248-262.
8. Elmasri, R., Navathe, S. B. (2002) *Fundamentals of Database Systems, Third Edition*. Addison-Wesley.
9. Gottesdiener, E. (2003) *Eliciting Business Rules in Workshops (part 2)*. Business Rules Journal, 2003 January, Vol. 4, No. 1, [interaktyvus], [žiūrėta 2007 spalio 14 d.]. Prieiga per internetą: <<http://www.BRCommunity.com/a2003/b121b.html>>
10. ISO. Tarptautinė standartizavimo organizacija. (1996). ISO/IEC 14977:1996. Informacinės technologijos – sintaksinė metakalba – išplėstas BNF, [interaktyvus], [žiūrėta 2008 kovo 15 d.]. Prieiga per internetą: <[http://standards.iso.org/ittf/PubliclyAvailableStandards/s026153_ISO_IEC_14977_1996\(E\).zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/s026153_ISO_IEC_14977_1996(E).zip)>

11. Kapočius, K., Butleris, R. (2003). Reikalavimų kuriamai IS nustatymo ir analizės ypatumai veiklos taisyklių metodikoje. *Informacinės technologijos '2003*. Konferencijos pranešimų medžiaga. Kaunas. Technologija, XIV-74 – XVI-81.
12. Kapočius, K., Butleris, R. (2005). Business Rules Driven Approach for Elicitation of IS Requirements. *Proceedings of 9th World Multi-Conference on Systemics, Cybernetics and Informatics*. Orlando, JAV, Vol. IV, 276-281.
13. Kapočius K., Butleris R. (2006) Repository for Business Rules Based IS requirements. *Informatica*, Vol. 17, No.4, p. 503-518.
14. Kapočius K., Baškevičius S. (2006) Veiklos taisyklių struktūrizavimo informacinių sistemų projektavimo metu įrankio realizacija. *Informacinės technologijos*. Tarpuniversitetinė magistrantų ir doktorantų konferencija. Pranešimų medžiaga. Vilnius. VU leidykla, 2 dalis, p. 91-96.
15. Kapočius, K. (2006) *Veiklos taisyklių struktūrizavimo modeliai ir jų taikymas kuriant informacijos sistemas*. Disertacija. Kaunas. Technikos mokslai, Informatikos inžinerija.
16. Morgan, T. (2002) *Business Rules and Information Systems*. Addison-Wesley.
17. Nemuraitė, L. (2006) *Modeliais grindžiamo informacinių sistemų projektavimo modeliai ir metodai* [interaktyvus], [žiūrėta 2007 lapkričio 27 d.]. Prieiga per internetą: <http://www.ktu.lt/habilitacija/nemuraite/Nemuraite_darbu_apzvalga.pdf>
18. Object Management Group (OMG). *Object Constraint Language Specification, version 2.0* [interaktyvus], [žiūrėta 2008 sausio 22 d.]. Prieiga per Internetą: <<http://www.omg.org/technology/documents/formal/ocl.htm>>
19. Object Management Group. (2007) *Production Rule Representation (PRR), Beta 1*. Iš Object Management Group internetinio puslapio [interaktyvus], [žiūrėta 2009 vasario 9 d.]. Prieiga per internetą: <<http://www.omg.org/docs/dtc/07-11-04.pdf>>
20. Object Management Group. (2008) *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*. Iš Object Management Group internetinio puslapio [interaktyvus], [žiūrėta 2009 vasario 9 d.]. Prieiga per internetą: <<http://www.omg.org/docs/formal/08-01-02.pdf>>
21. Ross, R. G. (1997). *The Business Rule Book: Classifying, Defining and Modeling Rules*. Business Rule Solutions, Houston.
22. Ross, R. G., Lam, G. S. W. (2001) *The Do's and Don'ts of Expressing Business Rules*. *Business Rule Solutions*, [interaktyvus], [žiūrėta 2008 kovo 15 d.]. Prieiga per internetą: <http://www.brsoptions.com/rulespeak_download.shtml>
23. Ross, R. G. (2003) *Principles of the Business Rules Approach*. Boston: Addison-Wesley.

24. Šarkiūnaitė, I; Simutis, R; Krikščiūnienė, D. (2007) *Magistro baigiamojo darbo rengimo tvarka. Metodiniai nurodymai. VU KHF informatikos katedros verslo informatikos ir verslo informacijos sistemų studijų programų studentams*. Vilnius: Vilniaus universiteto leidykla.

25. The Rule Markup Initiative. – RuleML iniciatyvinės grupės atviras tinklas [interaktyvus], [žiūrėta 2009 sausio 30 d.]. Prieiga per internetą: <<http://ruleml.org/papers/tutorial-ruleml-20050513.html>>

26. Valatkaitė, I., Vasilecas, O. (2005) On Business Rules Automation: The BR-Centric IS Development Framework. *Advances in Databases and Information Systems, 9th East European Conference, ADBIS 2005*, Tallinn, Estonia, September 12-15, 2005, Proceedings, J. Eder et al. (Eds.), *Lecture Notes in Computer Science, Vol. 3631*, Springer-Verlag Berlin Heidelberg, 349-364.

27. Von Halle B. (2001) *Business Rules Applied: Building Better Systems Using the Business Rules Approach*. Wiley.

1 PRIEDAS. Natūraliosios kalbos VT šablonai (modifikuotas BRS

RuleSpeak)

Darbe naudojamame VT grindžiamo reikalavimų specifikavimo metode naudojami VT šablonai, jų apibrėžimai ir taisyklių pavyzdžiai. VT šablonų komponentai rašomi iš naujos eilutės (išskyrus atvejus, kai komponentas nusikelia į naują eilutę).

Modifikuotame BRS RuleSpeak modelyje naudojamos savokos:

- terminas – koncepcinio duomenų modelio esybės, atributo ar reikšmės pavadinimas;
- faktas – koncepcinio duomenų modelio ryšys tarp esybių;
- procesas, procedūra – veiklos funkcija.

Lentelė 15

VT šablonai pagal BRS RuleSpeak

Kategorija	Neformalus apibrėžimas	Galimi subjekto tipai	Šablonas	VT pavyzdžiai iš automobilių draudimo veiklos (nevisur)
1.1 Atmetimo taisyklė/atmetimas/apribojimo taisyklė (<i>rejector</i>)	Apribojimas, kurio nepažeidžiant užtikrinamas duomenų teisingumas (neprieštarinumas)	Terminas, faktas, duomenų elementas	<i>b1.</i> <Subjektas> TURI NETURI TURĖTŪ NETURĖTŪ <faktas> [(, jeigu , kol) <sąlyga>].	Draudimo liudijimas turi būti išduotas, jeigu draudėjas sumokėjo draudimo įmoką ar jos dalį. Draudimo sutartis turi turėti unikalų numerį.
			<i>b2.</i> <Subjektas> gali turėtų <faktas> TIK jeigu kol <sąlyga>.	Klientas gali pateikti užsakymą tik jeigu klientas turi atsidaręs sąskaitą internete.
			<i>b3.</i> <Subjektas> gali turėtų <faktas> TIK <sąlyga>	Sutartis gali būti nutraukta, tik sutarties šalims nevykdant ar netinkamai vykdant nustatytas pareigas.
1.2 Leidimo taisyklė (<i>permission statement</i>)	Strateginė nuostata ar paaiškinimas, leidžiantis vykdyti veiklą.	Terminas, faktas, taisyklė, procesas, duomenų elementas	<i>b4.</i> <Subjektas> GALI <faktas VT raktažodis> [(,net jeigu ,kol) <sąlyga>].	Apmokėjimas už paslaugą, kurios suma yra 500 Lt ar mažiau, gali būti priimtas iš kliento net jeigu kliento kredito likutis nepatikrintas.
			<i>b5.</i> <Subjektas> NETURI <faktas> <VT raktažodis> [(, jeigu , kol) <sąlyga>].	Automobilis neturi būti apdraustas nuo vagystės.
2.1 Skaičiavimo taisyklė (<i>computation rule</i>)	Teiginys ar aritmetinė formulė, nurodanti, kaip apskaičiuoti tam tikrą skaitmeninę reikšmę.	Apskaičiuojamas terminas (reikšmė), duomenų elementas	<i>b6.</i> <Subjektas> turi neturi turėtų neturėtų BŪTI APSKAIČIUJAMAS -a -i -os kaip <matematinė formulė> [(, jeigu , kol) <sąlyga>].	Apmokėjimo už paslaugą suma apskaičiuojama kaip visų paslaugai priklausančių dalių apmokėjimų suma.
			<i>b7. Sutrumpintas variantas:</i> <Subjektas> = <matematinė formulė> [(, jeigu , kol) <sąlyga>].	Apmokėjimo už paslaugą suma = visų paslaugos dalių apmokėjimų suma.

Kategorija	Neformalus apibrėžimas	Galimi subjekto tipai	Šablonas	VT pavyzdžiai iš automobilių draudimo veiklos (nevisur)
2.2 Išvedimo taisyklė (<i>derivation rule</i>)	Teiginys ar loginė išraiška, nusakanti, kaip nustatyti taip/ne pobūdžio rezultata.	Išvestinis terminas (atributas, esybė, reikšmė), duomenų elementas	b8. <Subjektas> turi neturi turėtų neturėtų REIKŠTI, KAD <loginė išraiška> [(, jeigu , kol) <sąlyga>].	Vertingas klientas turi reikšti, kad kliento metinių įmokų suma viršija 1000 Lt.
			b9. <i>Sutrumpintas variantas:</i> <Subjektas> REIŠKIA, KAD NEREIŠKIA, KAD <loginė išraiška> [(, jeigu , kol) <sąlyga>].	Vertingas klientas reiškia, kad kliento metinių įmokų suma viršija 1000 Lt.
3.1.1 Išvados taisyklė (<i>inference rule</i>)	Taisyklė, pagal kurią ,atsižvelgiant į tam tikras aplinkybes, padaroma išvada.	Terminas, duomenų elementas	b10. <Subjektas> turi neturi turėtų neturėtų BŪTI LAIKOMAS -a -i -os <terminas> (, jeigu , kol) <sąlyga>.	Automobilio garantinis laikotarpis turi būti laikomas pasibaigusiu, jeigu automobilio pagaminimo data yra daugiau nei trimis metais ankstesnė už šiandienos datą arba automobilio kilometražas yra didenis už 100 000 km. Asmuo turėtų būti laikomas nepilnamečiu, jeigu asmens amžius yra mažiau 18-os metų.
			b11. <i>Sutrumpintas variantas:</i> <Subjektas> YRA NĖRA <terminas> (, jeigu , kol) <sąlyga>.	Automobilio garantinis laikotarpis yra pasibaigęs, jeigu automobilio pagaminimo data yra daugiau nei trimis metais ankstesnė už šiandienos datą arba automobilio kilometražas yra didesnis už 100 000 km. Asmuo yra nepilnametis, jei asmens amžius yra mažiau 18-os metų.
3.1.2 Taisyklės jungiklis (<i>rule toggle</i>)	Taisyklė, “įjungianti” ar “išjungianti” kitą taisyklę, priklausomai nuo konkrečių aplinkybių. Naudojama taisyklių išimtims nusakyti.	Taisyklė	b12. <i>Neformalus variantas:</i> <Taisyklės teiginys> NEBENT IŠSKYRUS <sąlyga>.	Bibliotekos kortelė gali priklausyti daugiausiai vienam skolintojui, nebent vienas iš skolintojų, kuriems priklauso bibliotekos kortelė, yra Bilas Geitsas.
			b13. <i>Formalus variantas:</i> <Taisyklės pavadinimas kodas> turi neturi turėtų neturėtų BŪTI TAIKOMA -as (, jeigu , kol) <sąlyga>.	Taisyklė-kortelė-priklauso-tik-vienam-skolintojui neturi būti taikoma, jeigu vienas iš skolintojų, kuriems priklauso bibliotekos kortelė, yra Bilas Geitsas.
3.1.3 Proceso jungiklis (<i>process toggle</i>)	Taisyklė, esant tam tikroms aplinkybėms “įjungianti” ar “išjungianti” procesą.	Procesas	b14. <Subjektas> turi neturi turėtų neturėtų BŪTI ĮGALINTAS -a BŪTI UŽDRAUSTAS -a (, jeigu , kol) <sąlyga>.	Įspėjimo-dėl-susitikimo-siuntimas turi būti uždraustas, jeigu kliento adresas nežinomas.
3.1.4 Duomenų jungiklis (<i>data toggle</i>)	Taisyklė, esant tam tikroms aplinkybėms šalinanti (ar sukurianti atsitiktinius)	Duomenų elementas	b15. <Duomenų elementas> turi neturi turėtų neturėtų BŪTI SUKURTAS -a IŠTRINTAS -a jeigu kol <sąlyga>.	Kiekvienas nenagrinėtas bylos klausimas turi būti ištrintas, jeigu byla uždaroma. Loterijos-laimintys-skaičiai turi būti sukurti, jeigu loterijos-data lygi

Kategorija	Neformalus apibrėžimas	Galimi subjekto tipai	Šablonas	VT pavyzdžiai iš automobilių draudimo veiklos (nevisur)
	duomenis.			šiandienai.
3.2.1 Reikšmės priskyrimo taisyklė (<i>imprint rule</i>)	Taisyklė, priskirianti saugomam duomenų elementui tam tikrą reikšmę.	Terminas, faktas, duomenų elementas	<i>b16.</i> <Subjektas> turi neturi turėtų neturėtų ĮGYTI REIKŠMĘ <terminas reikšmė> [kai jeigu <sąlyga>].	Franšizė-vagystės-atvejui turi įgyti reikšmę 10 %, jeigu draudimo automobilio sutartis su klientu sudaroma pirmą kartą. Automobilio-draudimo-mokestis turi įgyti reikšmę iš sprendimų lentelės T1 pagal automobilio modelį, pagaminimo metus ir kilometražą.
3.2.2 Pateikimo (atvaizdavimo) taisyklė (<i>presentation rule</i>)	Taisyklė, kuria nustatoma duomenų pateikimo forma (ekrane, ataskaitoje ir pan.).	Terminas, faktas, duomenų elementas	<i>b17.</i> <Subjektas> turi neturi turėtų neturėtų BŪTI PATEIKIAMAS -a -i -os [<i><vaizdavimo tarpė></i>] <vaizdavimo būdas> [jeigu kol <sąlyga>].	Draudimo-sutarties-numeris turi būti pateikiamas ekrane raudonai, jei sutarties galiojimo data vėlesnė arba lygi šiandienos datai minus 7 dienos. Sudarytos civilinio draudimo sutartys turi būti pateikiamos Sudarytų-civilinio-draudimo-sutarčių ataskaitoje išrikiuotos sutarties numerio didėjimo tvarka.
3.3.1 Proceso trigeris (<i>process trigger</i>)	Taisyklė, pagal kurią, susidarius tam tikroms aplinkybėms, automatiškai vykdomas procesas ar procedūra.	Procesas, procedūra	<i>b18.</i> <Subjektas> turi neturi turėtų neturėtų BŪTI VYKDOMAS -a , kai kol <sąlyga>.	Siųsti-išankstinį-įspėjimą užsakymui turi būti vykdomas, kai užsakymas išsiųstas.
3.3.2 Taisyklės trigeris (<i>rule trigger</i>)	Taisyklė, pagal kurią, susidarius tam tikroms aplinkybėms, automatiškai paleidžiama kita taisyklė.	Taisyklė	<i>b19.</i> <Taisyklės pavadinimas kodas> turi neturi turėtų neturėtų BŪTI PALEISTA -as , kai <sąlyga>.	Taisyklė-Numatoma-pristatymo-data turi būti paleista, kai krovinys išvedamas į ekraną.

Šaltinis: Kapočius, K. (2006) Disertacija "Veiklos taisyklių struktūrizavimo modeliai ir jų taikymas kuriant informacijos sistemas"

2 PRIEDAS. Nedalomų taisyklių tipų savybės pagal Roso metodą

Parengta pagal [Ross, 21] [15].

Lentelė 16

Nedalomų taisyklių tipų savybės pagal Roso metodą

TAISYKLIŲ TIPŲ GRUPĖ / TIPAS-SANTRUMPA	REIKŠMĖ	REIKŠMĖ, JEIGU VYKDOMA	REIKŠMĖ, JEIGU TIESIOG TESTUOJAMA
I. Egzempliorių patvirtinimo <i>Išskirtinė savybė:</i> tik šio tipo taisyklės gali skaičiuoti egzempliorius. <i>Išigos vertės tipas:</i> egzempliorių kiekis/EK.			
Privaloma – P (<i>angl. X</i>)	turi tam tikros savybės egzempliorių	privalo turėti	ar turi?
Apribota – A (<i>angl. LIM</i>)	turi nurodytą tam tikros savybės egzempliorių kiekį	egzempliorių kiekis privalo neviršyti nurodyto limito	ar egzempliorių kiekis neviršija nurodyto limito?
II. Tipų patvirtinimo <i>Išskirtinė savybė:</i> tik šio tipo taisyklės gali veikti kaip loginiai operatoriai IR arba ARBA. <i>Išigos vertės tipas:</i> tipų kiekis/TK.			
Bendra – B (<i>angl. M</i>)	turi nurodytą kiekį tipų	tipų kiekis privalo neviršyti nurodyto limito	ar tipų kiekis neviršija nurodyto limito?
Bendrai-atiskirianti – BA (<i>angl. ME</i>)	neturi dviejų ar daugiau tipų vienu metu	negali turėti daugiau negu vieną iš kelių skirtingų tipų vienu metu	ar neturi daugiau negu vieną skirtingą tipą vienu metu?
Bendrai-susiejanti – BS (<i>angl. MI</i>)	vienu metu turi bent vieną iš dviejų ar daugiau tipų	privalo turėti bent vieną iš dviejų ar daugiau tipų vienu metu	ar vienu metu turi bent vieną iš dviejų ar daugiau tipų?
Bendrai-uždraudžianti –BU (<i>angl. MP</i>)	vienu metu neturi visų iš dviejų ar daugiau tipų	negali turėti visų iš dviejų ar daugiau tipų vienu metu	ar neturi visų iš dviejų ar daugiau tipų vienu metu?
III. Pozicijos patvirtinimo <i>Išskirtinė savybė:</i> tik šio tipo taisyklės gali formuoti išvestinius faktus. <i>Išigos vertės tipas:</i> taisyklėms POZ, ZEM, AUK – išvestas faktas/IF; CHRO, SEN, NAU – data-laikas/DL.			
Pozicinė – POZ (<i>angl. POS</i>)	konkreiti vertė (arba vertės), apibrėžianti poziciją uždaroje sekoje	privalo būti nurodytoje pozicijoje	ar yra nurodytoje pozicijoje?
Žemiausia – ZEM (<i>angl. LOW</i>)	pirmoji arba mažiausia uždaroje reikšmių sekoje	turi būti žemiausia	ar žemiausia?
Aukščiausia – AUK (<i>angl. HIGH</i>)	paskutinė arba didžiausia uždaroje	privalo būti aukščiausia	ar aukščiausia?

TAISYKLIŲ TIPŲ GRUPĖ / TIPAS-SANTRUMPA	REIŠMĖ	REIŠMĖ, JEIGU VYKDOMA	REIŠMĖ, JEIGU TIESIOG TESTUOJAMA
Chronologinė – CHRO (<i>angl.</i> CHRO) Seniausia – SEN (<i>angl.</i> OLD) Naujausia – NAU (<i>angl.</i> NEW)	reikšmių sekoje konkreči amžiaus vertė uždaroje sekoje anksčiausia (seniausia) uždaroje sekoje pati naujausia uždaroje sekoje	privalo būti nurodyto amžiaus pozicijoje privalo būti anksčiausia privalo būti naujausia	ar yra nurodyto amžiaus pozicijoje? ar anksčiausia? ar pati naujausia?
IV. Funkcinio patvirtinimo			
<i>Išskirtinė savybė:</i> tik šio tipo taisyklės gali tikrinti duotojo duomenų tipo reikšmes kito duomenų tipo reikšmių sekoje (t.y., kaip funkcijos). <i>Išeigos vertės tipas:</i> egzemplioriaus numeris/EN			
Funkcinė – FNKC (<i>angl.</i> FUNC)	visi tipo egzemplioriai tenkina vartotojo pateiktą funkciją	privalo tenkinti vartotojo pateiktą funkciją	ar tenkina vartotojo pateiktą funkciją?
Unikali – UNIK (<i>angl.</i> UQ)	bet kurie du tipo egzemplioriai yra skirtingi	privalo būti unikalūs	ar unikalūs?
Svyruojanti – SV (<i>angl.</i> FLUC)	visi tipo egzemplioriai skiriasi nuo gretimų egzempliorių	privalo skirtis	ar skiriasi?
Auganti – AUG (<i>angl.</i> AS)	visi tipo egzemplioriai išdėstyti didėjimo tvarka	turi didėti	ar didėja?
Krentanti – KRE (<i>angl.</i> DE)	visi tipo egzemplioriai išdėstyti mažėjimo tvarka	turi mažėti	ar mažėja?
Neatnaujinama – NAT (<i>angl.</i> NRE)	visi panašūs tipo egzemplioriai seka vienas po kito	turi būti nuosekli	ar nuosekli?
V. Palyginamojo įvertinimo			
<i>Išskirtinė savybė:</i> jokio kito tipo taisyklės negali pritaikyti tiesioginio reikšmių suliginimo testų. <i>Išeigos vertės tipas:</i> palyginimo vertė/PV			
Lygu – LG (<i>angl.</i> EQ)	reikšmės lygios	privalo būti tokia pati, kaip	ar tokia pati?
Nelygu – NLG (<i>angl.</i> NE)	reikšmės nelygios	privalo būti ne tokia pati, kaip	ar ne tokia pati?
Daugiau-už – DU (<i>angl.</i> GT)	didesnė pagal reikšmę	privalo būti didesnė už	ar didesnė?
Daugiau-arba-lygu – DLG (<i>angl.</i> GE)	didesnė arba lygi pagal reikšmę	privalo būti didesnė arba lygi	ar didesnė už arba lygi?

TAISYKLIŲ TIPŲ GRUPĖ / TIPAS-SANTRUMPA	REIŠMĖ	REIŠMĖ, JEIGU VYKDOMA	REIŠMĖ, JEIGU TIESIOG TESTUOJAMA
Mažiau-už – MU (<i>angl.</i> LT)	mažesnė pagal reikšmę	privalo būti mažesnė pagal reikšmę	ar mažesnė?
Mažiau-arba-lygu – MLG (<i>angl.</i> LE)	mažesnė arba lygi pagal reikšmę	privalo būti mažesnė arba lygi	ar mažesnė už arba lygi?
VI. Matematinio įvertinimo <i>Išskirtinė savybė:</i> Jokio kito tipo taisyklės negali atlikti matematinių skaičiavimų. <i>Išigos vertės tipas:</i> rezultatas/REZ			
Apskaičiuota – APSK (<i>angl.</i> calc)	reikšmė tokia pati, kaip ir vartotojo nurodyto skaičiavimo rezultatas	privalo būti lygi skaičiavimo rezultatui	ar lygi rezultatui?
SUM, ATM, DGB, PDL ir t.t. (<i>angl.</i> SUM, SUB, MULT, DIV etc.)	reikšmė tokia pati, kaip ir nurodyto skaičiavimo rezultatas	privalo būti lygi rezultatui	ar lygi rezultatui?
VII. Projektijos valdikliai <i>Išskirtinė savybė:</i> jokio kito tipo taisyklės negali automatiškai “projektuoti” vienos vertės (pvz., loginio vieneto) į kitą vertę, pvz: automatiškai įgalinti, kopijuoti arba iškviesti vykdymą. <i>Išigos vertės tipas:</i> taisyklei IGL – įgalintų egzempliorių kiekis/IEK, KOP – nukopijuotų egzempliorių kiekis/KEK, IVK – įvykdytų egzempliorių kiekis/VEK.			
Įgalinta – IGL (<i>angl.</i> EA)	reikalauja papildomo būvio	privalo paskleisti būvį	ar būvis paskleistas?
Nukopijuota – KOP (<i>angl.</i> COP)	reikalauja vertės kopijavimo	privalo nukopijuoti vertę	ar vertė nukopijuota?
Įvykdyta – IVK (<i>angl.</i> EX)	reikalauja įvykdymo	privalo būti įvykdyta	ar įvykdyta?

Šaltinis: Kapočius, K. (2006) Disertacija "Veiklos taisyklių struktūrizavimo modeliai ir jų taikymas kuriant informacijos sistemas", pakoreguota autoriaus

Lentelė 17

Išvestinių taisyklių tipai pagal Roso metodą

A Egzempliorių testavimas	B Pozicijos patikrinimas	C Atributų modifikavimas	D Sekos valdymas	E Nuoseklumų specifikavimas	F Kompozicijos struktūrų testavimas	G Sąlyginio laiko įvertinimas	H Atnaujinimų įvertinimas	I Veiklos koordinavimas	J Egzempliorių įgalinimas	K Egzempliorių kopijavimas	L Egzempliorių iškvietimas
PRE egzistuojanti- anksčiau	POS-CM pozicinė- susiejanti-daug	INC didėjanti	IZ inicijuojanti	PAT šabloninė	R apribota	AGED trukmės	UP-C atnaujinimų- kiekio	CO-UP bendro- atnaujinimo	REA įgalinimo-su- atkūrimu	RCOP nukopijuota-su- atkūrimu	REX įvykdyta-su- atkūrimu
A pirmesnioji	LOW-CM žemiausia- susiejanti-daug	DEC mažėjanti	FW judėjimo-pirmyn		REFX atspindinti	PEND laukianti	UP-R atnaujinimų- dažnio	RE-UP pakartotinio- atnaujinimo	LREA įgalinimo-su- atkūrimu- apribota	LOW-RCOP žemiausia- nukopijuota-su- atkūrimu	FIRE paleista
EN besitęsianti	HIGH-CM aukščiausia- susiejanti-daug	UQ-MA unikali-atributų- modifikavimui	PRO progresyvi		CYCC cikliška	SHRT trumpiausia	FIX iššaldanti	SYNC sinchronizavimo	FLIP perjungimo	HIGH-RCOP aukščiausia- nukopijuota-su- atkūrimu	TEX numatyta- vykdimui
POR proporcinė	CHRO-CM chronologinė- susiejanti-daug	FLUC-MA svyruojanti- atributų- modifikavimui	RET regresyvi		TRAN tranzitinė	LONG ilgiausia	UFIX iššaldanti- vartotojams	FSQ paleidimo-sekos	RFLIP perjungimo-su- atkūrimu	OLD-RCOP seniausia- nukopijuota-su- atkūrimu	
MD bendrai- priklausoma	OLD-CM seniausia- susiejanti-daug	AS-MA auganti-atributų- modifikavimui	RIZ pakartotinio- inicijavimo		NODE mazgas				TI laiko-paskyrimo	NEW-RCOP naujausia- nukopijuota-su- atkūrimu	
FD funkciškai- priklausoma	NEW-CM naujausia- susiejanti-daug	DE-MA krentanti- atributų- modifikavimui	CYCL ciklinė						TUI intervalų-tarp- atnaujinimų- paskyrimo		
REL susieta		NRE-MA neatnaujinama- atributų- modifikavimui									

3 PRIEDAS. Išvestinių taisyklių tipai pagal Roso metodą

Šaltinis: Kapočius, K. (2006) Disertacija "Veiklos taisyklių struktūrizavimo modeliai ir jų taikymas kuriant informacijos sistemas", pakoreguota autoriaus