

**KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA**

Egidijus Baltrušaitis

**VEIKLOS TAISYKLIŲ KLASIFIKAVIMO IR
SPECIFIKAVIMO METODŲ TYRIMAS**

Magistro darbas

**Vadovas
doc. dr. R. Butleris**

KAUNAS, 2005

**KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA**

**TVIRTINU
Katedros vedėjas
doc. dr. R. Butleris
2005-05-**

**VEIKLOS TAISYKLIŲ KLASIFIKAVIMO IR
SPECIFIKAVIMO METODŲ TYRIMAS**

Informatikos mokslo magistro baigiamasis darbas

**Kalbos konsultantė
Lietuvių kalbos katedros lektorė
dr. J. Mikelionienė
2005-05-**

**Vadovas
doc. dr. R. Butleris
2005-05-**

**Recenzentas
doc. dr. V. Pilkauskas
2005-05-**

**Atliko
IFM-9/1 gr. stud.
E. Baltrušaitis
2005-05-**

KAUNAS, 2005

SUMMARY

The work is based on the research of business rules classification and specification methods. The basics of business rules approach are discussed. The most common business rules classification and modeling methods are analyzed. Business rules modeling techniques and tools for supporting them in the information systems are presented. Basing on the analysis results business rules classification method is proposed. Templates for every business rule type are presented. Business rules structuring process is shortly described. Business rules repository architecture, which is based on the proposed classification method, is presented. Proposed repository can store all types of business rules of defined classification method. Rules in the database are stored linked with events which are grouped by business functions. Such rules storing makes repository structure more flexible and more adaptive in information systems. Repository can function as an independent database for storing business rules and can be fully integrated in to the rule based information system. Rule specification tool for business rules registering and viewing was created. This tool has a simple user interface which allows register business rules and provides first steps in formalizing business statements. Finally conclusions are presented.

TURINYS

IVADAS.....	8
1. ŠIUOLAIKINĖ VEIKLOS TAISYKLĖS SAMPRATA.....	10
1.1. Veiklos taisyklės apibrėžimas	10
1.2. Veiklos taisyklių standartizavimas.....	11
1.3. Tyrėjai daugiausiai prisidėję prie veiklos taisyklių koncepcijos tobulinimo	11
1.4. Veiklos taisyklėms keliami reikalavimai	12
1.5. Apibendrinimas	15
2. VEIKLOS TAISYKLIŲ KLASIFIKAVIMO METODŲ APŽVALGA.....	16
2.1. Bendra apžvalga	16
2.2. GUIDE projektas.....	17
2.3. Roso metodas	18
2.4. MDC modelis	20
2.5. CommonKADS metodologija pagrįstas klasifikavimo metodas.....	20
2.6. RAP grupės pasiūlyta veiklos taisyklių klasifikavimo schema.....	23
2.7. Barbaros von Halle veiklos taisyklių klasifikacija	25
2.8. Kiti klasifikavimo metodai.....	26
2.9. Veiklos taisyklių integravimo sistemoje būdai	28
2.10. Klasifikavimo metodų apibendrinimas	28
3. VEIKLOS TAISYKLIŲ MODELIAVIMO PRIEMONIŲ APŽVALGA.....	29
3.1. Bendra veiklos taisyklių modeliavimo priemonių apžvalga	29
3.2. UML ir OCL.....	30
3.3. EER diagramos.....	33
3.4. Veiklos taisyklių modeliavimo priemonių apibendrinimas.....	33
4. PRIEMONĖS VEIKLOS TAISYKLĖMS REALIZUOTI INFORMACINĖSE SISTEMOSE	34
4.1. Veiklos taisyklių sistemos samprata ir jos naudojimo tikslingumas.....	34
4.2. Veiklos taisyklių valdymo sistemų apžvalga	35
4.3. Veiklos taisyklių saugyklos projektavimo koncepcija	37
4.4. Apibendrinimas	38
5. VEIKLOS TAISYKLIŲ KLASIFIKAVIMO METODO SUDARYMAS	39
5.1. Pasiūlytas veiklos taisyklių klasifikavimo metodas	39
5.2. Veiklos taisyklių specifikavimo būdo parinkimas	42
5.3. Veiklos taisyklių struktūrizavimo proceso aprašymas	42

6. VEIKLOS TAISYKLIŲ SAUGYKLOS PROJEKTAVIMAS	43
6.1. Sistemos kūrimo procesas veiklos taisyklių metodologijoje.....	43
6.2. Veiklos taisyklių sistemos architektūra	44
6.3. Veiklos taisyklių saugyklos architektūra.....	46
6.4. Veiklos taisyklių atvaizdavimas sukurtoje veiklos taisyklių saugykloje	51
6.5. Veiklos taisyklių įvedimas į saugyklą.....	53
6.6. Veiklos taisyklių saugyklos formalūs reikalavimai.....	55
6.7. Tolimesnio saugyklos vystymo ir panaudojimo perspektyvos.....	55
7. VEIKLOS TAISYKLIŲ SAUGYKLOS PROTOTIPO REALIZACIJOS APRAŠYMAS	57
7.1. Sistemos architektūra	57
7.2. Sistemos panaudojimo atvejų modelis	57
7.3. Vartotojo sąsajos modelis.....	58
7.4. Sistemos prototipo vartotojo sąsajos meniu aprašymas	58
7.5. Testinis darbo su sistema pavyzdys.....	59
8. PASIŪLYTO VEIKLOS TAISYKLIŲ KLASIFIKAVIMO METODO ĮVERTINIMAS KITŲ METODŲ KONTEKSTE	63
IŠVADOS.....	65
NAUDOTA LITERATŪRA	67
PRIEDAI	69
1 priedas. Veiklos taisyklių klasifikavimas ir saugyklos architektūra.....	69
2 Priedas. Sukurtos sistemos prototipo vartotojo sąsajos langų aprašymas	74
3 Priedas. Veiklos taisyklių atvaizdavimas veiklos taisyklių saugykloje.....	82

LENTELIŲ SĄRAŠAS

1.1 lentelė. Veiklos taisyklių apibrėžimų įvairovė.....	10
2.1 lentelė. Veiklos taisyklių klasifikavimo metodų sąrašas.....	16
2.2 lentelė. GUIDE projekto veiklos taisyklių tipų detalizavimas.....	17
2.3 lentelė. Veiklos taisyklių tipų šeimos pagal Roso metodą.....	18
2.4 lentelė. Veiklos taisyklių klasifikavimas pagrįstas CommonKADS metodologija.....	21
2.5 lentelė. CommonKADS metodologija pagrįsto metodo veiklos taisyklių atvaizdavimo būdai...22	
2.6 lentelė. Formalių specifikacijų terminų priskyrimas veiklos taisyklėms.....	24
2.7 lentelė. B. V. Halle veiklos taisyklių klasifikacija.....	25
5.1 lentelė. Pasiūlytos veiklos taisyklių klasifikacijos aprašymas.....	39
5.2 lentelė. Veiklos taisyklių šablonai ir pavyzdžiai.....	41
5.3 lentelė. Veiklos taisyklių specifikavimo metodo parinkimas.....	42
6.1 lentelė. Veiklos taisyklių saugyklos duomenų bazės lentelės ir jų aprašymai.....	48
6.2 lentelė. Išvesties veiklos taisyklės atvaizdavimas sukurtoje veiklos taisyklių saugykloje.....	51
6.3 lentelė. Fakto atvaizdavimas sukurtoje veiklos taisyklių saugykloje.....	52
8.1 lentelė. Veiklos taisyklių klasifikavimo metodų įvertinimas.....	63

PAVEIKSLŲ SĄRAŠAS

2.1 pav. GUIDE projekto veiklos taisyklių klasifikacija	17
2.2 pav. Integralumo apribojimo ir sąlygos žymėjimas Roso metode	19
2.3 pav. Bazinių ir veiksmo taisyklių klasifikavimas.....	26
2.4 pav. Veiklos taisyklių klasifikavimas projektavimo požiūriu	27
4.1 pav. Veiklos taisyklių sistemos projektavimo koncepcija	38
5.1 pav. Veiklos taisyklių klasifikacija pasiūlytu metodu.....	39
5.2 pav. Veiklos taisyklių struktūrizavimo procesas.....	42
6.1 pav. Sistemos kūrimo procesas veiklos taisyklių metodologijoje.....	43
6.2 pav. Veiklos taisyklių sistemos architektūra	45
6.3 pav. Veiklos taisyklių variklio ir taikomosios programos duomenų bazės sąryšis.....	45
6.4 pav. Veiklos taisyklių saugyklos architektūra pasiūlytam veiklos taisyklių klasifikavimo metodui.....	47
6.5 pav. Veiklos taisyklių saugyklos pildymo eiga.....	54
6.6 pav. Veiklos taisyklių saugyklos panaudojimo sritys	56
7.1 pav. Sistemos architektūra.....	57
7.2 pav. Sistemos panaudojimo atvejų modelis	57
7.3 pav. Vartotojo sąsajos modelis.....	58
7.4 pav. Pradinis sistemos prototipo puslapis	59
7.5 pav. Funkcijos įvedimo forma.....	59
7.6 pav. Įvykio įvedimo forma	60
7.7 pav. Veiklos taisyklės įvedimo forma	60
7.8 pav. Įvykio ir veiklos taisyklės susiejimo forma.....	60
7.9 pav. Taisyklės šablono sukūrimo forma.....	61
7.10 pav. Taisyklės šablono elementų įvedimo forma	61
7.11 pav. Šablonų tipų, taisyklių šablonų ir jų elementų ataskaita	62
7.12 pav. Funkcijų, įvykių ir taisyklių ataskaita	62

IVADAS

Darbe sprendžiama problema susijusi su veiklos taisyklių surinkimo, klasifikavimo, specifikuojimo ir panaudojimo metodais kuriant ir naudojant informacines sistemas. Šiuo metu egzistuoja nemažai veiklos taisyklių klasifikavimo metodų. Nėra vieningos nuomonės veiklos taisyklių klasifikavimo, surinkimo ir net apibrėžimo klausimais. Veiklos taisyklės specifikuojamos procedūriškai arba deklaratyviai. Gali būti naudojamas vienas arba kitas specifikuojimo būdas, bet dažnai aptinkami ir šių būdų junginiai. Kuriant programinę įrangą tradiciškai naudojamas procedūrinis veiklos taisyklių aprašymas. Deklaratyvusis yra naujesnis ir šiuo metu prie jo daugiau yra dirbama. Veiklos taisyklės gali būti geografiškai paskirstytos. Organizacijos, priklausomai nuo savo poreikių, turimomis priemonėmis pasiima veiklos taisykles, bei pritaiko jas savo reikmėms. Problemos iškyla kuomet reikia specifikuoti veiklos taisykles ir jas surinkti į vieną vietą. Taip pat svarbus dalykas yra taisyklių modifikavimas. Procedūriškai specifikuotas taisykles sunkiau modifikuoti, tuo tarpu deklaratyviai aprašytas taisykles sunkiau paimti ir pritaikyti programose. Reikalingi algoritmai, kurie paima veiklos taisykles ir jas įkomponuoja į programas. Savo ruožtu, pačios programos turi būti parašytos taip, kad galėtų paimti taisykles ir pritaikyti jas sistemos darbui. Taigi pagrindinis uždavinys yra išsiaiškinti, kaip tas taisykles surinkti, klasifikuoti ir specifikuoti, kokį metodą tam pasirinkti. Nuo metodo priklauso veiklos taisykles naudosiančios programos struktūra, naudojami algoritmai.

Pagrindinis darbo tikslas – veiklos taisyklių klasifikavimo ir specifikuojimo metodų tyrimas. Tyrinėjami veiklos taisyklių klasifikavimo ir specifikuojimo metodai. Atlikto tyrimo pagrindu pasiūlomas veiklos taisyklių klasifikavimo metodas, kuris yra palyginti paprastas ir apima daugumą veiklos taisyklių panaudojimo atvejų. Kiekvienam taisyklės pateikiami šablonai, kurie leidžia fiksuoti taisykles griežtai struktūrizuota forma. Pasiūlyto klasifikavimo metodo pagrindu sudaryta veiklos taisyklių saugykla, kuri leidžia kaupti sudaryto klasifikavimo metodo veiklos taisykles bei užtikrina sklandų veiklos taisyklių klasifikavimo ir struktūrizavimo procesą. Suprojektuota veiklos taisyklių saugyklos vartotojo sąsaja, leidžianti įvesti į duomenų bazę ir peržiūrėti duomenų bazėje specifikuotas veiklos taisykles. Saugykloje esančios informacijos detaliam nagrinėjimui formuojamos apibendrintų duomenų ataskaitos.

Darbas pateiktas skyriuose. 1 skyriuje pateikiama šiuolaikinė veiklos taisyklės samprata. 2 skyriuje aptariami geriausiai esamas veiklos taisyklių klasifikacijos charakterizuojantys metodai. 3 skyriuje apžvelgiamos veiklos taisyklių modeliavimo priemonės. Priemonės veiklos taisyklėms realizuoti informacinėse sistemose pateikiamos 4 skyriuje. 5 skyriuje aprašomas sudarytas veiklos taisyklių klasifikavimo metodas. Suprojektuota veiklos taisyklių saugykla ir jos panaudojimo

galimybės pateikiamos 6 skyriuje. 7 skyriuje pateikiamas realizuotas veiklos taisyklių saugyklos prototipas. Sudaryto veiklos taisyklių klasifikavimo metodo įvertinimas kitų klasifikavimo metodų kontekste pateiktas 8 skyriuje. Darbas baigiamas tyrimo išvadomis. 1 priede pateikiamas publikuotas straipsnis. Sukurtos sistemos vartotojo sąsajos detalus aprašymas pateikiamas 2 priede.

1. ŠIUOLAIKINĖ VEIKLOS TAISYKLĖS SAMPRATA

Įmonės, organizacijos kasdien susiduria su įvairiomis veiklos taisyklių (VT) rūšimis. Įvairios valstybinės institucijos, įmonės ir jų padaliniai, klientai, konkurentai ir pačios rinkos sąlygos reguliariai generuoja taisyklių pokyčius, kurie yra pervedami į veiklos taisykles [4]. VT daro įtaką ir vadovauja žmonių elgesiui (verslo perspektyva). Taip pat veikia informaciją, kuri yra kaupiama IT sistemoje, bei riboja šios informacijos keitimą (IT perspektyva).

1.1. Veiklos taisyklės apibrėžimas

Iki šiol nėra vieningos nuomonės klausimais, susijusiais su veiklos taisyklių klasifikavimu, identifikavimu, atvaizdavimu ir net pačiu veiklos taisyklės apibrėžimu. 1.1 lentelėje pateikta šiuo metu esanti veiklos taisyklių apibrėžimų įvairovė [21].

1.1 lentelė. Veiklos taisyklių apibrėžimų įvairovė

Šaltinis	Apibrėžimas
Business Rules: The missing Link, Daniel S. Appleton [1984]	„... aiškus apribojimo teiginys, kuris egzistuoja kartu su verslo ontologija“ [p. 146]
Entity Modeling: Techniques and Application, Ronald G. Ross [1987]	„... specialios taisyklės (ar verslo politika), kuri valdo ... elgseną [organizacijos] ir išskiria ją iš kitų ... šios taisyklės valdo organizacijos būsenos pasikeitimus...“ [p. 102]
The Business Rule Book (1 st ed.), Ronald G. Ross [1994]	„... diskreti, aktyvi veiklos politika ar praktika. VT gali būti skaitoma kaip vartotojo reikalavimas, kuris yra išreikštas neprocedūriška ir netechnine forma (dažniausiai tekstiniais teiginiais) ... VT vaizduoja teiginį apie verslo elgseną ...“ [p. 496]
GUIDE Business Rule Project Report [1995]	„... teiginys, kuris apibrėžia arba sąlygoja tam tikrą veiklos aspektą ... [kuris yra] skirtas veiklos struktūros įvertinimui, arba kontroliuoti ir veikti verslo elgseną. VT negali būti daloma ar išskaidoma toliau į kitas VT ... jei toliau VT bus „apkarpoma“, bus svarbios verslo informacijos praradimas.“ [p. 4-5]
The Business Rule Book (2 nd ed.), Ronald G. Ross [1997]	„Terminas, faktas (tipas) ar taisyklė, reprezentuojanti predikatą ...“ [p. 380]
Business Rules Group (žinoma kaip GUIDE Business Rule Project), [1998]	„Direktyva, skirta daryti įtaką ar valdyti verslo elgseną. Tokios direktyvos egzistuoja verslo politikos palaikymui, atsakas verslo rizikai, grėsmėms arba galimybėms.“
Capturing Business Rules, Ronald G. Ross and Gladys S. W. Lam [2000]	„Atominė dalis pakartotinio naudojimo veiklos logikos, specifikuotos deklaratyviai“
Managing Reference Data in Enterprise Databases, Malcolm Chisholm [2001]	„Atskiras teiginys, kuris paima duomenis ar informaciją, kurią valdo organizacija, ir išveda kitus duomenis ar informaciją iš šių duomenų, arba naudoja juos tam tikro veiksmo sužadinimui.“ [p. 365]
Business Rules Applied: Building	„... sąlygos, kurios valdo veiklos įvykius taip, kaip yra

Šaltinis	Apibrėžimas
Better Systems Using the Business Rule Approach, Barbara von Halle [2002]	priimtina verslui.“ [p. 28]
Business Rules and Information Systems, Tony Morgan [2002]	„Iš esmės, VT yra paprastas teiginys apibrėžiantis tam tikrą verslo aspektą ... Tai yra apribojimas ta prasme, kad VT yra tarp to kas turi ar neturi būti tam tikru atveju. Bet kuriuo atveju, turi būti galimybė nustatyti, kad sąlyga, išreikšta apribojimu yra logiškai teisinga; jei ne, tuomet reikalingas pataisomasis veiksmas. Toks interpretavimas, kuris gali būti aprašytas kaip Būlio logika (žiūrint iš programinės įrangos taško), ir yra pagrindinė priežastis, kodėl terminas <i>veiklos logika</i> yra dažnai naudojamas.“ [p. 5-6]

1.2. Veiklos taisyklių standartizavimas

Standartas, kuris vieningai reprezentuotų veiklos taisykles ir veiklos žodynus, šiuo metu yra kuriamas. Šį standartą ketina pristatyti OMG (*Object Management Group*), kuri yra tokių standartų kaip UML ir CORBA savininkė. Numatomas išleisti standartas leis pasikeitimą taisyklėmis ir jų žodynais tarp skirtingų produktų gamintojų ir suteiks brandumo visai veiklos taisyklių bendruomenei.

Papildomai yra kuriamas ISO 9000 standartas, kuris leis įvertinti veiklos taisyklių bei jų žodynų kokybę. Šis standartas yra svarbus VT apsikeitimui tarp organizacijų (pavyzdžiui, tarp valstybės institucijų ir visų kitų organizacijų, kurios turi paklusti šioms taisyklėms) ir tarp organizacijos atskirų departamentų (pavyzdžiui, verslo departamento ir IT departamento).

1.3. Tyrėjai daugiausiai prisidėję prie veiklos taisyklių koncepcijos tobulinimo

Yra grupė asmenų ypatingai prisidėjusių prie veiklos taisyklių evoliucijos, kurių pavardės nuolat figūruoja VT publikacijose. Ronald G. Ross, laikomas veiklos taisyklių „tėvu“. Jis yra išleidęs keletą su veiklos taisyklėmis susijusių knygų, bei pateikęs savo VT klasifikatorių (jis darbe bus aptartas vėliau). Toliau sektų Terry Moriarty, šiuo metu esantis Ross'o bendradarbis. Svarbus asmuo yra John A. Zachman. Jis sukūrė populiarią Zachman Framework metodologiją, kuri išreiškia savitą veiklos taisyklių koncepciją. Daug šioje srityje yra pasiekusi Barbara von Halle. Ji daug metų dirba konsultacinėje kompanijoje ir yra publikavusi nemažai straipsnių, susijusių su VT metodologija. Chris J. Date laikomas reliacinių duomenų bazių teorinių pagrindų pradininku. Jis išvystė savo teoriją link predikatų logikos ir veiklos taisyklių. Tony Morgan – termino „*extreme non-programming*“ savininkas, kuriuo veiklos taisyklių koncepciją jis išreiškia labai glaustu būdu. BRG (*Business Rules Group*) ir OMG (*Object Management Group – Business Rules Working Group*) yra svarbiausios veiklos taisyklių standartizavimo ir jų propagavimo organizacijos.

1.4. Veiklos taisyklėms keliami reikalavimai

Veikloms taisyklėms yra keliami tam tikri reikalavimai. Reikalavimų rinkiniai paprastai yra pateikiami daugelyje VT klasifikavimo modelių, bet yra išskiriami ir bendri reikalavimai. Palyginti bendrus reikalavimus pateikia C. J. Date [9]. Jis pateikia dvylika, kaip jis pats vadina, nurodymų (metataisyklių) veiklos taisyklėms.

1. Vykdomosios taisyklės. VT turi būti kompiliuojamos ir vykdomosios.
2. Deklaratyvios taisyklės. VT turi būti apibrėžiamos deklaratyviai. Veiklos taisyklių kalba, t. y. kalba arba sąsaja, kuri palaiko deklaratyvias specifikacijas, turi būti tiksliai išbaigta ir gerai suprojektuota (turi aiškiai atvaizduoti tiek sintaksinius, tiek semantinius aspektus ir koncepcinį integralumą).
3. Taisyklių rūšys. VT gali būti laisvai suskirstytos į tris rūšis:
 - Atvaizdavimo taisyklės, susijusios programos vartotojo veiksmams (jos apima taisyklės interaktyvių formų atvaizdavimui, jų užpildymui, klaidų išvedimui ir kt.);
 - Duomenų bazių taisyklės, susijusios su duomenų bazės duomenų apibrėžimu, duomenų išgavimu bei jų įrašymu į duomenų bazę apdorojant vartotojų užklausas ir kt.;
 - Programinės taisyklės, susijusios su programinių poreikių apdorojimu, kad teisingai įvykdyti tam tikrą organizacijos darbo elementą.

Duomenų bazių ir programinės taisyklės dar gali būti toliau skirstomos į skaičiavimų, apribojimų ir išvestines taisykles. Taip dar gali būti poveikio/atsako taisyklės (turinčios *IF...THEN...* formą).

4. Deklaravimo ir vykdymo eilės tvarka. Veiklos taisyklės, bendru atveju, priklauso viena nuo kitos. Taisyklės gali būti deklaruojamos bet kokia fizine eilės tvarka, o jų vykdymo tvarką kontroliuoja veiklos taisyklių variklis.
5. Veiklos taisyklių variklis yra duomenų bazių valdymo sistema (DBVS). Taisyklės yra išreiškiamos tam tikrais terminų konstruktais duomenų bazės schemeje. Žiūrint iš loginės pusės, jos yra būtina šios schemos dalis. Iš to seka, kad VT variklis yra tam tikra DBVS rūšis, ir taisyklės yra tam tikra duomenų (metaduomenų) rūšis. VT variklis gali veikti kaip tarpinė sistema, naudodama kitas DBVS ir/arba failų sistemas duomenų saugojimui (dažniausiai keletą posistemų vienu metu). Taisyklės egzistuoja išorinėse, koncepcinėse ir vidinėse formose:
 - Išorinė duotos taisyklės forma yra taisyklės šaltinio forma (t. y. forma, kuria jos buvo pateiktos VT varikliui).
 - Koncepcinė forma yra taisyklės atvaizdavimas kanonine forma.

- Vidinė forma yra forma arba formos, kurias VT variklis atrenka kaip tinkamas saugojimo ir vykdymo tikslams.

Šios trys formos turi būti griežtai atskiriamos ir viena su kita nemaišomos.

Jeigu teigiame, kad „taisyklės yra duomenys“, tai visi servisai taikomi duomenų bazės duomenims, gali būti taikomi ir taisyklėms, t. y. gali būti taikomos standartinės DBVS priemonės. Išskiriami pagrindiniai aspektai:

- Kiekvienas vartotojas turi dirbti tik su jį liečiančiomis taisyklėmis.
 - Taisyklės turi būti užklausiamosios ir papildomosios.
 - Turi būti išlaikytas taisyklių nuoseklumas.
 - Taisyklės turėtų turėti galimybę būti paskirstytoms tarp programų ir būti pakartotinai naudotinos.
6. Taisyklių variklis yra reliacinė DBVS. Konceptuali atžvilgiu VT variklis yra reliacinis. Žiūrint iš vartotojo taško jis veikia kaip abstraktus modelis.
7. Taisyklių valdymas. Kadangi taisyklės yra ne tik duomenys, tiksliau kalbant, metaduomenys, tai duomenų bazės dalis, kur jos yra laikomos, vadinama taisyklių katalogu. Taisyklių katalogas turi būti reliacinis. Autorizuoti vartotojai turi turėti galimybę prieiti prie šios katalogo, tiek informacijos gavimo, tiek papildymo tikslais. VT variklis turi:
- Aptikti ir panaikinti įvairius konfliktus VT kataloge;
 - Optimizuoti taisyklių perteklišumą VT kataloge;
 - Leisti katalogo papildymą be kitų sistemos dalių darbo sutrikimo.
8. Apribojimų rūšys. Integralumo apribojimai dalinami į keturias rūšis:
- tipų apribojimai;
 - atributų apribojimai;
 - ryšių apribojimai;
 - duomenų bazių apribojimai.
9. Išplečiamumas. Su bet kuria duotąja veiklos taisykle turi būti galima sužadinti esamas programas taip, lyg jos (taisyklės) būtų vidiniai operatoriai. Kitais žodžiais, tariant sistema turi būti išplečiama, ir programos, kaip ir taisyklės, turi būti visiems prieinamos ir pakartotinai naudotinos.
10. Platformos nepriklausomumas. Taisyklių kalba ir taisyklės išreikštos ta kalba turi būti nepriklausomos nuo programinės įrangos ir techninės platformos. VT variklis turi būti atsakingas už:
- taisyklių kompiliavimą į vykdomąjį kodą, kuris turi veikti ant bet kokios programinės ir techninės platformos;
 - priskyrimą vykdomojo kodo atitinkamiems procesoriams vykdomojoje aplinkoje.

Turi būti galimybė programai aprėpti keletą atskirų galutinių posistemių, kuriose veikia atskiros DBVS ir/arba yra skirtingos failinės sistemos. Reikalingi sujungimai turi būti specifiškai atitinkamomis veiklos taisyklėmis.

11. Apsisaugojimas nuo ardomųjų veiksmų. Jeigu yra palaikoma sąsaja, kuri suteikia prieigą prie duomenų bazės žemesniame nei konceptualiame lygyje, tuomet turi būti priemonės uždrausti naudotis šia sąsaja, norint apsaugoti sistemą. Turi būti galima apsaugoti nuo integralumo apribojimų apėjimo.
12. Pilnas automatizavimas. Programos sukurtos naudojant veiklos taisykles turi būti pilnai išbaigtos (viskas, kas gali būti automatizuota, turi būti automatizuota). Pilnas taisyklių rinkinys turi atspindėti išbaigtą organizacijos veiklos modelį. VT variklis turi palaikyti visos integruotos programinės sistemos vystymą.

Pagrindiniu dokumentu, apibrėžiančiu VT nepriklausomumą, yra laikomas Business Rules Group veiklos taisyklių manifestas [5]. Jame yra dešimt punktų (kurie dar toliau yra detalizuojami) labai aiškiai ir tiksliai apibrėžiančių VT. Pagrindiniai punktai yra šie:

1. Pirminiai reikalavimai, o ne antriniai.
2. Atskirtos nuo procesų, į juos neįtrauktos.
3. Gerai apgalvotos žinios, o ne šalutinis produktas.
4. Deklaratyvios, ne procedūrinės.
5. Gerai suformuota išraiška, nebūtinai konkrečiam tikslui.
6. Taisyklėmis paremta architektūra, o ne netiesioginis realizavimas.
7. Taisyklėmis paremti procesai, o ne išimtimis pagrįstas programavimas.
8. Verslo gerovei, ne technologijai.
9. *Iš, pagal ir dėl* verslo žmonių, o ne informacinių technologijų specialistams.
10. Valdymas verslo logikos, o ne techninės/programinės platformų.

Kadangi darbe bus akcentuojamas deklaratyvus ir procedūrinis veiklos taisyklių specifikavimas, todėl detaliau panagrinėtas ketvirtasis VT manifesto punktas, kuris šiuo aspektu yra aktualiausias (nors šio manifesto kiekvienas skirsnis, galima sakyti, yra labai svarbus). Taigi ketvirtas punktas, kuriame teigiama, kad VT turi būti deklaratyvos, o ne procedūrinės. Šis punktas skaidomas dar į septynis smulkesnius teiginius:

1. Taisyklės verslo auditorijai turi būti išreikštos deklaratyviai, natūralios kalbos sakiniiais.
2. Jeigu kas nors negali būti išreikšta, tai tas dalykas nėra taisyklė.
3. Teiginių rinkinys yra deklaratyvus tik tada, kada rinkinys neturi numanomų taisyklių sekų.

4. Taisyklių teiginiai, kurie reikalauja kitų konstrukcijų nei terminai ir faktai, išreiškia prielaidas apie sistemos realizavimą.
5. Taisyklė yra atskirta nuo bet kokio jai skirto priverstinio vykdymo reikalavimo. Taisyklė ir jos priverstinio vykdymo reikalavimai yra du skirtingi dalykai.
6. Taisyklės turi būti apibrėžiamos nepriklausomai nuo atsakomybės už *kas, kur, kada* arba *kaip* jas priverstinai vykdyt.
7. Išimtyms taisyklėms yra išreiškiamos kitomis taisyklėmis.

1.5. Apibendrinimas

Apžvelgus šiuo metu esančius veiklos taisyklių apibrėžimus, galima įsitikinti, jog nėra vieningos nuomonės šiuo klausimu. Pateikti bendri reikalavimai (nurodymai) veiklos taisyklėms. Smulkiai išanalizuoti C. J. Date reikalavimai (metataisyklės) veiklos taisyklėms. Apžvelgtas pagrindinis veiklos taisyklių nepriklausomumą apibrėžiantis BRG išleistas veiklos taisyklių manifestas, didesnę dėmesį skiriant vietai, kurioje akcentuojama, jog taisyklės turi būti deklaratyvios, o ne procedūrinės. Apibendrinant, galima teigti, kad veiklos taisyklė turi būti:

- logiškai nedali (atominė). Taisyklė negali būti toliau daloma. Skaidant veiklos taisyklės galimas informacijos praradimas.
- deklaratyvi. Taisyklė deklaruojama kaip atskiras teiginys.
- nuosekli. Sistemoje negali būti konfliktuojančių taisyklių.
- neperteklinė. Veiklos taisyklių sistemoje negali būti tą pačią informaciją apibrėžiančių taisyklių.
- priklausanti veiklai ir orientuota į konkrečią veiklą. Veiklos taisyklių formulavimą, keitimą ir jų paskelbimą negaliojančiomis turi atlikti tik konkrečios veiklos atstovai.
- tiksli. Veiklos taisyklė turi turėti tik vieną interpretaciją.

2. VEIKLOS TAISYKLIŲ KLASIFIKAVIMO METODŲ APŽVALGA

2.1. Bendra apžvalga

Šiuo metu yra daug veiklos taisyklių klasifikavimo metodų. 2.1 lentelėje pateiktas apibendrintas klasifikacijų sąrašas [2, 6], kuriame pateikiamos įvairių autorių siūlomos veiklos taisyklių klasifikavimo schemas.

2.1 lentelė. Veiklos taisyklių klasifikavimo metodų sąrašas

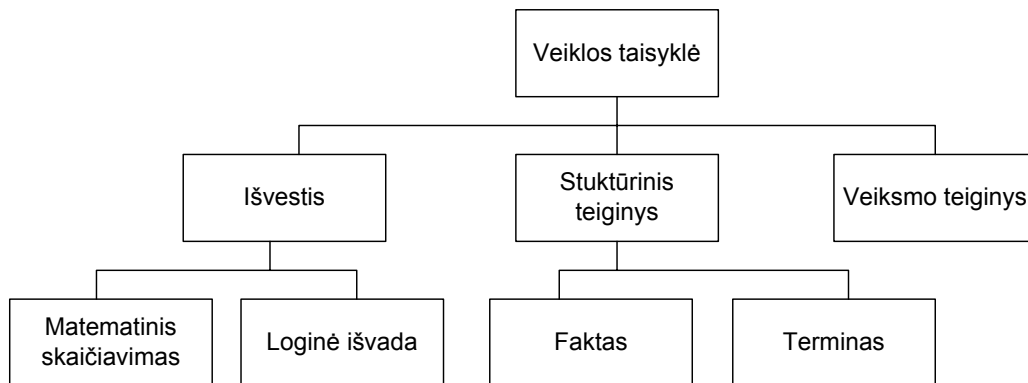
Šaltinis	Trumpas klasifikavimo aprašymas
GUIDE Business Rules Project	<ul style="list-style-type: none"> ▪ Struktūrinis teiginys (terminai, faktai) ▪ Veiksmo teiginys (integralumo apribojimai, sąlygos, autorizacija) ▪ Išvestys (skaičiavimai, išvados)
Ron Ross, Database Research Group	<ul style="list-style-type: none"> ▪ Terminai ▪ Faktai (ryšių tipai, subtipai, atributai) ▪ Taisyklės (skaičiavimai, išvestys, apribojimai, sekos, euristika)
James Odell, nepriklausomas konsultantas	<ul style="list-style-type: none"> ▪ Apribojimai (poveikis/atsakas, operacija, struktūra) ▪ Loginės išvados (išvestys, skaičiavimai) <p>Pastaba: taisyklės gali būti globalios, lokalsios arba laikinos</p>
Tom Romeo, IBM	<ul style="list-style-type: none"> ▪ Struktūrinės taisyklės (ryšiai, domenai, kardinalumas) ▪ Elgesio taisyklės (prieš-sąlygos, po-sąlygos, išvestys)
Margaret Thorpe, Tangram	<ul style="list-style-type: none"> ▪ Apibrėžimai ▪ Integralumo apribojimai ▪ Bendrieji deklaratyvūs apribojimai ▪ Procedūriniai apribojimai ▪ Išvestys ▪ Loginės išvados
Barbara von Halle, Knowledge Partners	<ul style="list-style-type: none"> ▪ Terminai ▪ Faktai ▪ Apribojimai ▪ Skaičiavimai ▪ Nurodymai ▪ Išvestys ▪ Veiksmo įgalintojai
Usoft Inc.	<ul style="list-style-type: none"> ▪ Apribojimai ▪ Dedukcija ▪ Elgesio taisyklės ▪ Atvaizdavimo taisyklės
Dan Tasker, Air New Zealand	<ul style="list-style-type: none"> ▪ Veiksmo apribojimas ▪ Veiksmo sužadinimas ▪ Apribojimas
Brightware	<ul style="list-style-type: none"> ▪ Veikla ▪ Strategija ▪ Darbų seka ▪ Sprendimų euristika

Šaltinis	Trumpas klasifikavimo aprašymas
Vision Software	<ul style="list-style-type: none"> ▪ Validavimo taisyklės ▪ Išvestys ▪ Ryšiai ▪ Sąlygos veiksmai
Versata Inc.	<ul style="list-style-type: none"> ▪ Išvestys ▪ Tikrinimai ▪ Apribojimai ▪ Atvaizdavimo ▪ Veiksmo/įvykių ▪ Nuorodinės taisyklės
SEEC korporacija	<ul style="list-style-type: none"> ▪ Pateikimo taisyklės ▪ Apribojimai (būsenos, perėjimo) ▪ Išvados (skaičiavimai, išvestys) ▪ Poveikio/atsako taisyklės ▪ Proceso taisyklės ▪ Saugumo ir audito taisyklės

2.2. GUIDE projektas

Detaliau aptarkime, vieną pagrindinių, GUIDE projekto klasifikaciją [11]. Veiklos taisyklė gali būti (2.1 pav.):

1. Išvestis – sakiny, išreiškiantis žinias, kurios yra išvedamos iš kitų veiklos žinių.
2. Struktūrinis teiginys – apibrėžta mintis ar fakto konstatavimas, kuris išreiškia tam tikrą organizacijos struktūros aspektą.
3. Veiklos teiginys – apribojimą arba sąlygą išreiškiantis sakiny, kuris riboja arba valdo organizacijos veiksmus.



2.1 pav. GUIDE projekto veiklos taisyklių klasifikacija

Kiekviena VT rūšis yra toliau detalizuojama. Pagrindiniai potipiai pateikti 2.2 lentelėje:

2.2 lentelė. GUIDE projekto veiklos taisyklių tipų detalizavimas

Išvestis	Struktūrinis teiginys	Veiksmo teiginys
<ul style="list-style-type: none"> ▪ Matematinis skaičiavimas; ▪ Loginė išvada. 	<ul style="list-style-type: none"> ▪ Terminas: <ul style="list-style-type: none"> - bendrasis; - veiklos. ▪ Faktas: (pagal kilnę): <ul style="list-style-type: none"> - pirminis; 	Klasės: <ul style="list-style-type: none"> - sąlyga; - apribojimas; - autorizacija. Tipai: <ul style="list-style-type: none"> - įgalintojas;

Išvestis	Struktūrinis teiginys	Veiksmo teiginys
	- išvestinis; (pagal tipą): - atributas; - dalyvis (asociacija, agregavimas, vaidmuo); - apibendrinimas.	- taimeris; - vykdytojas. Pobūdis: - veiksmą valdantis; - veiksmą įtakojantis.

2.3. Roso metodas

Vienas detaliausių veiklos taisyklių klasifikavimo ir modeliavimo aspektais yra Roso [22] metodas. Palyginus su GUIDE projektu jis pranašesnis daugeliu aspektų. Pagal Roso metodą taisyklės gali būti atominės ir išvestinės. Išskiriami 32 atominių taisyklių tipai, kurie yra grupuojami į 7 šeimas. Išvestinės taisyklės nėra atominės ir gali būti sudarytos iš keleto atominių arba kitų išvestinių taisyklių. Rosas išskiria net 58 išvestinių taisyklių tipus (todėl šiuo klasifikavimo požiūriu yra labai detalus), kurie pagal panaudojimo sritį yra grupuojami į 12 šeimų. Visos veiklos taisyklių šeimos pateiktos 2.3 lentelėje.

2.3 lentelė. Veiklos taisyklių tipų šeimos pagal Roso metodą

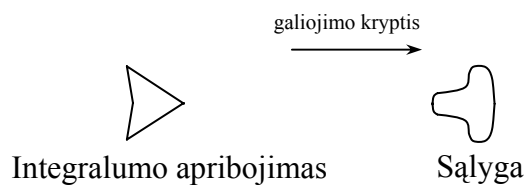
Atominių taisyklių šeimos	Išvestinių taisyklių šeimos
<ul style="list-style-type: none"> ▪ Egzempliorių patvirtintojai; ▪ Tipų patvirtintojai; ▪ Pozicijos patvirtintojai; ▪ Funkciniai patvirtintojai; ▪ Palyginamieji įvertintojai; ▪ Matematiniai įvertintojai; ▪ Projektacijos valdikliai. 	<ul style="list-style-type: none"> ▪ Egzempliorių testavimo; ▪ Pozicijos patikrinimo; ▪ Atributų modifikavimo; ▪ Sekos valdymo; ▪ Sekos specifikavimo; ▪ Kompozicijos struktūrų testavimo; ▪ Sąlyginio laiko įvertinimo; ▪ Atnaujinimų įvertinimo; ▪ Veiklos koordinavimo; ▪ Egzempliorių įgalinimo; ▪ Egzempliorių kopijavimo; ▪ Egzempliorių iškvietimo.

Roso metode be klasifikavimo schemas yra pasiūlyta veiklos taisyklių modeliavimo, užrašant jas grafine forma, metodika. Taisyklės teiginio pateikimas formalios taisyklės pavidalu susideda iš penkių etapų. Probleminės srities analitikas turi atsakyti į šiuos klausimus:

- ar taisyklė yra apribojimas ar sąlyga (ar taisyklė reikalauja vykdymo, ar yra išreikšta kaip patikrinimas?);
- kas yra taisyklės bazė, kuriam tipui duomenų modelyje priskirti taisyklę?
- kas yra taisyklės korespondentai, kokie tipai duomenų modelyje būtini norint patikrinti taisyklę?
- koks taisyklės tipas?
- koks ryšys yra tarp taisyklės bazės ir korespondento, kaip jie tarpusavyje susiję?

Atsakius į šiuos klausimus, veiklos taisyklė užrašoma grafiniu pavidalu (naudojant Roso pasiūlytą metodiką).

Taisyklės pagal Rosą gali būti dviejų rūšių: sąlyga arba integralumo apribojimas. Sąlygos rūšies taisyklės rezultatas gali būti arba loginis nulis, arba loginis vienetas. Rezultatas gali būti ir nežinomas. Jos naudingos, kai reikia nustatyti kokias taisykles vykdyti arba testuoti tam tikrais atvejais. Integralumo apribojimo rūšies taisyklės rezultatas visada yra loginis vienetas. Kadangi jos negali duoti rezultato loginis nulis, jos privalo būti vykdomos. Sąlygoms ir integralumo apribojimams Rosas siūlo grafinį žymėjimą parodytą 2.2 pav.



2.2 pav. Integralumo apribojimo ir sąlygos žymėjimas Roso metode

Simboliai gali būti sukami į bet kokią pusę, tik nereikia pamiršti galiojimo krypties. Kiekviena taisyklė visada apibrėžia tam tikrą į duomenų modelį įeinančio tipo egzempliorių. Šis egzempliorius vadinamas taisyklės baze. Taisyklės bazė gali būti duomenų objektas, atributas ar kita taisyklė. Kiekviena taisyklė privalo turėti bazę. Kiekviena taisyklė, be bazės remiasi dar bent vienu duomenų modelio egzemplioriumi. Šie egzemplioriai vadinami taisyklės korespondentais. Kiekviena taisyklė privalo turėti bent vieną korespondentą. Korespondentai yra tiesiogiai susiję su tikrinimu, kurį atlieka taisyklė. Korespondentas dažniausiai yra duomenų tipas (duomenų objektas, atributas), tačiau taip pat gali būti kita taisyklė ar veiksmas. Korespondentai turi šias savybes:

- atominės taisyklės, priklausančios patvirtinimo tipo šeimai, privalo turėti du arba daugiau korespondentus;
- atominės taisyklės, priklausančios matematinių įvertintojų šeimai gali turėti daugiau nei vieną korespondentą;
- kitoms šeimoms priklausančios atominės taisyklės gali turėti tik vieną korespondentą.

Taisyklės tipo parinkimas yra svarbiausias žingsnis interpretuojant taisyklę. Nuo taisyklės tipo priklauso, kokį patikrinimą atliks taisyklė. Taisyklių tipai parenkami iš standartinės atominių ir išvestinių taisyklių aibės. Visi taisyklių tipai gali būti naudojami tiek integralumo apribojimams, tiek sąlygoms. Svarbu dalykas yra kaip susieta bazė su korespondentu. Šis ryšys turi atsispindėti duomenų modelyje, taisyklė negali įvesti naujo ryšio. Duomenų modelį galima priimti kaip aibe faktų, kurie aprašo veiklos sritį. Faktai gali būti sąryšio, potipio ar atributų tipo.

2.4. MDC modelis

Vienas iš naujesnių veiklos taisyklių klasifikavimo modelių yra Meta Data Coalition (MDC) kartu su Microsoft sukurtas modelis. Išskirtinis šio modelio požymis yra tas, kad jis pasižymi detaliu saugojimo modelio aprašu. Iš karto po šio modelio pasirodymo, nes jis yra panašus į GUIDE projektą, pasirodė Business Rules Group atsakas [20], kuriame pateikti abiejų projektų skirtumai.

MDC modelyje vietoje termino veiklos taisyklės teiginys, naudojamas terminas veiklos taisyklės „slankiojimas“ (angl. *business rambling*), kuris GUIDE projekto grupei buvo nepriimtinas. Veiklos taisyklė MDC modelyje apibrėžia kaip pareiti nuo vienos būsenos prie kitos arba kaip uždrausti tokį perėjimą. Pagrindinis skirtumas yra VT tipai. MDC tipai:

- sąvoka;
- faktas;
- veiksmas;
- išvestis.

Veiklos taisyklės gali būti sugrupuotos į VT aibes. MDC modelis buvo kuriamas kaip programinio projekto Open Information Model (OIM) dalis [15]. Todėl daugiausiai dėmesio yra skiriama jo realizacijoms detalėms, nes GUIDE ir Roso modeliai realizacijos detalių nepateikia. MDC modelyje taisyklės saugomos parametrais pagrįstu principu. Kiekvienas taisyklės elementas yra saugomas atskirose lentelėse. Tokioje struktūroje yra numatyti kiekvieno VT modelio elemento atributai ir jų ryšys su pagrindinio duomenų modelio elementais. Bazinis duomenų modelis yra reikalingas, nes taisyklės yra išreiškiamos šio modelio terminais.

2.5. CommonKADS metodologija pagrįstas klasifikavimo metodas

Dar viena siūloma VT klasifikavimo schema [24] yra paremta populiaria CommonKADS metodologija. CommonKADS – žinių analizės ir žinių sistemų kūrimo standartas, kurį visą arba jo dalį naudoja dauguma didžiųjų bendrovių. Ši metodologija jau daugiau kaip dvidešimt metų daugelio Europos šalių vystomas produktas [13]. Jos pagrindu daugelis kompanijų kuria bankines, draudimo, programų kūrimo ir kt. sistemas. CommonKADS dažnai yra pagrindas įvairių vyriausybinių projektų, žinių valdymo, reikalavimų surinkimo ir veiklos procesų analizės kūrimo sistemų atveju.

Nagrinėjama klasifikacija yra orientuota daugiau semantiškai. Naudojant CommonKADS kaip rėminę konstrukciją (bazę), taisyklės klasifikuojamos remiantis jų veiklos rolėmis, kurias jos atlieka veiklos procese. Šioje klasifikavimo schemoje naudojami proceso ir užduoties terminai apibrėžiami taip:

- procesas yra bendriausias būdas veiklos elgsenai aprašyti. Procesai nėra susieti su konkrečiu aktoriumi ar vieta. Vienas procesas gali būti naudojamas keliems tikslams.

- užduotis yra apibrėžiama kaip elgsena, kuri gali būti susieta su tam tikru aktoriumi. Užduotys yra atliekamos procesų eigose. Užduoties (angl. *task*) ir veiklos (angl. *activity*) sąvokos šio modelio kontekste yra sutapatinamos.

2.4 lentelėje pateikta siūloma klasifikavimo schema [24]. Schema gali būti naudojama VT klasifikavimui, atsižvelgiant į jų semantines ypatybes (jų įtaką veiklos procese). Šioje schemoje yra penkiolika VT tipų. VT tipai yra sugrupuoti į tris kategorijas: struktūros, elgsenos ir valdymo. Šios kategorijos atspindi įvairius veiklos aspektus. Struktūrinės VT aprašo statinius veiklos aspektus. Elgsenos taisyklės aprašo sąlygas prie kurių yra atliekami tam tikri veiksmai. Valdymo taisyklės aprašo aukštesnio lygio veiklos apribojimus. Kiekvienam taisyklės tipui yra pateikiamas aiškinamasis klausimas, kuris gali būti naudojamas kaip pradinis taškas detaliam veiklos taisyklių studijavimui.

2.4 lentelė. Veiklos taisyklių klasifikavimas pagrįstas CommonKADS metodologija

Kategorija	Taisyklės tipas	Aiškinamasis klausimas
Struktūrinės	Koncepcijos struktūra	Kokios yra pagrindinės koncepcijos ir kokie jų ryšiai srityje?
	Išlaikymas	Kiek laiko informacija apie tam tikrą objektą turi būti laikoma?
	Kaupimas	Ar turi būti kaupiama objekto informacija?
Elgsenos	Informacijos srautų	Kokios informacijos reikia atlikti užduočiai iš kitų užduočių?
	Kontrolės srautų	Kurios taisyklės kontroliuoja kitų užduočių vykdymą?
	Prieš-sąlygos	Kokios sąlygos turi būti tenkinamos prieš atliekant tam tikrą veiksmą?
	Po-sąlygos	Kokios sąlygos įsigalioja po tam tikro veiksmo?
	Dažnumas	Kaip dažnai bus atliekamas tam tikras veiksmas?
	Trukmė	Kiek laiko reikia ar gali prireikti atlikti tam tikrą veiksmą?
	Užduoties žinios	Kokių žinių reikia tam tikros užduoties atlikimui?
Valdymo	Organizacinė	Kas gali būti pasakyta apie organizacijos vykdomą politiką?
	Tikslas ir Vertė	Koks proceso tikslas? Kokia tikėtina proceso vertė?
	Aktoriaus kompetencija	Kokius sugebėjimus turi turėti aktorius?
	Aktoriaus atsakomybė	Kokias užduotis aktorius gali atlikti?
	Resursai	Kokios taisyklės yra naudojamos resursų valdymui?

Šios klasifikavimo schemas kūrėjai taisyklių atvaizdavimui suformulavo tokius kriterijus:

- pagrindine priemone laikomas UML, kuomet jis yra tinkamas. Tekstinis formatas naudojamas tik tuomet, kai UML netinka.
- jeigu yra reikalingas tekstinis formatas, tai bandoma suformuluoti VT šabloną, kuris užtikrina, kad tam tikro tipo VT turės tą pačią struktūrą.
- jei šablonas nerandamas, tai rekomenduojamas bet koks pasirinktas tekstinis formatas.

2.5 lentelėje pateikiami siūlomi taisyklių atvaizdavimo būdai.

2.5 lentelė. CommonKADS metodologija pagrįsto metodo veiklos taisyklių atvaizdavimo būdai

Kategorija	Taisyklės tipas	Atvaizdavimas
Struktūrinės	Koncepcijos struktūra	<i>UML klasių diagrama.</i> Naudojamas stereotipas <<concept>>, parodymui, jog srities koncepcijos buvo pavaizduotos.
	Išlaikymas	<i>UML stereotipas.</i> Kuomet reikia specifikuoti daug išlaikymo informacijos, žemiau pateikti šablonai gali būti naudojami specifikuoti tekstines VT, kaip klasės ar atributo nuosavybę arba kaip <<business rule>> anotaciją susietą su klase ar atributu. <i>-[Information type/attribute] is valid for [period]</i> <i>-[Information type/attribute] must be destroyed after [period or task]</i>
	Kaupimas	<i>Loginis tipas arba klasių ir atributų stereotipas.</i> Kuomet reikia specifikuoti daug informacijos apie praeitus dalykus, žemiau pateiktas šablonas gali būti naudojamas specifikuoti tekstines VT, kaip klasės ar atributo nuosavybę arba kaip <<business rule>> anotaciją susietą su klase ar atributu. <i>All changes in the value of [information type or attribute] must be recorded</i>
Elgsenos	Informacijos srautų	<i>UML veiklos diagrama.</i>
	Kontrolės srautų	<i>UML veiklos diagrama arba būsenų diagrama.</i> Tekstinės alternatyvos: <i>-[Task X] can only be performed when [task Y] is completed</i> <i>-[Task] must be performed until / while [event]</i> <i>-If [event], perform [task]</i> <i>-If [task], then [event]</i> arba pseudo kodas
	Prieš- ir Po-sąlygos	Žemiau pateikti šablonai gali būti naudojami specifikuoti tekstines VT kaip veiklos/veiksmo arba kaip <<business rule>> anotaciją susietą su veikla/veiksmu. <i>-[Task] must only be performed when [condition]</i> <i>-After the execution of [task], [condition] must be true</i>
	Dažnumas	<i>UML sąveikų diagrama.</i> Papildomai gali būti naudojami šie šablonai: <i>-It is expected that [task] will be performed X times every [time measure]</i> <i>-[Task] will be performed by X [human actors]</i>
	Trukmė	Žemiau pateiktas šablonas gali būti naudojamas specifikuoti tekstines VT kaip veiklos/veiksmo arba kaip <<business rule>> anotaciją susietą su veikla/veiksmu. <i>[Task] should be performed within [time constraint]</i>
	Užduoties žinios	<i>Neapibrėžtas formatas, susijęs su konkrečia užduotimi.</i> Šių taisyklių modeliavimas išseina už VT modeliavimo ribų. Patariama šias taisykles sugrupuoti ir susilaikyti nuo detalaus jų modeliavimo.

Kategorija	Taisyklės tipas	Atvaizdavimas
Valdymo	Organizacinė	<i>Neapibrėžtas formatas.</i> Šios taisyklės iš prigimties yra labai skirtingos ir neįmanoma apibrėžti bendro šablono.
	Tikslas ir Vertė	<i>Neapibrėžtas formatas, susijęs su užduotimi arba procesu.</i> Šio tipo informacija gali būti išreikštas daugeliu būdų. Šios taisyklės turi būti susiejamos su užduotimi arba veikla, kurią jos aprašo.
	Aktorius kompetencija	<i>-[Actor] that performs [task] must have [skill or knowledge]</i> Aktorius kompetencijos taisyklės turi parodyti, kokias žinias ir sugebėjimus jis turi turėti, kad atlikti tam tikrą užduotį.
	Aktorius atsakomybė	<i>-If [condition], [task] must be performed by [actor]</i> Aktorius atsakomybės parodo, kas yra atsakingas už užduoties atlikimą. Tokios taisyklės, pavyzdžiui, gali aprašyti, kad normaliomis sąlygomis užduotį turi atlikti sistema. Jei sąlygos pasikeičia, tai skirtinga taisyklė galėtų teigti, kad tam tikras žmogiškasis aktorius tą užduotį turėtų atlikti.
	Resursai	<i>Neapibrėžtas formatas, susiejantis resursą su užduotimi.</i>

2.6. RAP grupės pasiūlyta veiklos taisyklių klasifikavimo schema

Dar vieną klasifikavimo schemą pasiūlė RAP (Rules and Patterns) grupė. Ši grupė užsiima įvairių su veiklos taisyklėmis susijusių dokumentų publikavimu [23]. Jų požiūris yra orientuotas į diskrečių įvykių ir duomenų modeliavimo, sistemų teorijos ir komponentais pagrįstos sistemos vystymo kombinaciją. Ši grupė priimtina laiko BRG (Business Rules Group) pasiūlytą veiklos taisyklių klasifikavimo schemą, kurioje VT skirstomos į sąvokas (terminus), faktus, apribojimus ir išvestis. Jie pasinaudojo šia klasifikacija, išplėsdami ją ir laisvai padalindami kiekvieną VT rūšį į struktūrines ir elgsenos dedamąsias. Dalijimo principas paremtas, galima sakyti, savo nuožiūra, t. y. kur, kokia taisyklė geriau tinka.

- Terminai. Yra dvi terminų rūšys.
 - Struktūrinio modelio terminai. Struktūriniai terminai yra klasių, atributų, ryšių vardai, kurie visada yra pastovūs.
 - Elgsenos modelio terminai. Elgsenos terminai yra įvykių ar operacijų vardai, kurie yra trumpalaikiai (pereinantys).
- Faktai. BRG schemoje faktas aprašo veiklos pobūdį faktų terminais, kurie susieja terminus vieną su kitu. Visi terminai BRG schemoje laikomi struktūriniais. RAP grupė pridėda elgsenos faktus, kurie susieja elgsenos ir struktūrinius terminus.
 - Struktūrinio modelio faktai. Struktūrinis faktas susieja struktūrinius terminus tam tikru struktūriniu sakiniu. Jis sujungia pastovius dalykus vienas su kitu.

- Elgsenos modelio faktai. Elgsenos faktas susieja elgsenos terminus su struktūriniais terminais tam tikru elgsenos sakiniu. Šie faktai parodo, kurie pastovūs dalykai yra paveikiami, kai kas nors įvyksta.
- Apribojimai. Struktūriniame modelyje apribojimai riboja atributų reikšmes arba ryšių skaičių. RAP schemeje papildomi apribojimai pridedami prie elgsenos modelių, kur yra specifikuojamos prieš-sąlygos. Išskiriamos dvi sąlygų rūšys:
 - fakto sąlyga. Pasirenkama viena proceso šaka. Pasirinkimas tarp veiksmų.
 - apribojimo sąlyga. Apsaugo procesą ir/arba vykdo jį atgalios tvarka bei inicijuoja pranešimą apie apribojimo pažeidimą.

Prieš-sąlygos susideda iš dviejų sąlygų rūšių, t. y. nepradėti proceso, kol netenkinamos tam tikros sąlygos, ir pradėti vykdyti procesą, bet jį sutraukti, jei tam tikros sąlygos yra tenkinamos.

Struktūriniai apribojimai yra prieš-sąlygos, kurios galioja kiekvienam galimam įvykiui. Jos galioja visą laiką, kol nepakeičiamos tam tikros taisyklės.

Elgsenos apribojimai yra prieš-sąlygos, kurios galioja vienam ar keliems įvykių tipams, bet ne visiems. Šie apribojimai galioja tol, kol prieštarauja kitų įvykių atėjimui į sistemą.

- Išvestys. BRG schemeje išvestys parodo, kaip žinios vienoje formoje gali būti transformuojamos į kitą formą, skaičiavimų pagalba. RAP schemeje išvestys yra struktūrinės ir elgsenos.
 - Struktūrinio modelio išvestys – invariantai. Struktūrinio modelio išvestys specifikuoja duomenų reikšmes kaip invariantinę esybės būseną, t. y. teisinga prieš ir po kiekvieno įvykio. Paprastai struktūrinės išvestimis yra specifikuojami objektai (klasės, atributai arba ryšiai) struktūriniame modelyje.
 - Elgsenos modelio išvestys – pereinančios po-sąlygos. Elgsenos išvestis yra po-sąlyga, kuri specifikuoja duomenų reikšmes po įvykio apdorojimo. Ji turi teigiamą reikšmę tik tuoj pat po įvykio. Paprastai elgsenos išvestimis yra specifikuojami objektų įvykiai ar operacijos elgsenos modelyje.

Šiame VT modelyje RAP grupė laisvai priskyrė formalių specifikacijų koncepcijas (invariantai, prieš-sąlygos ir po-sąlygos) veiklos taisyklių terminų, faktų, apribojimų ir išvesčių koncepcijoms. Iš tikrųjų nėra bendros priskyrimo schemas, nes formalių specifikacijų koncepcijos gali būti interpretuojamos keliais būdais. Pavyzdinė priskyrimo schema pateikta 2.6 lentelėje.

2.6 lentelė. Formalių specifikacijų terminų priskyrimas veiklos taisyklėms

	Struktūrinė taisyklė	Elgsenos taisyklė
Apribojimas	Invariantas	Prieš-sąlyga
Išvestis	Invariantas	Po-sąlyga

2.7. Barbaros von Halle veiklos taisyklių klasifikacija

Žemiau pateiktoje 2.7 lentelėje yra B. V. Halle pasiūlyta VT klasifikavimo schema ir jai skirti šablonų pavyzdžiai [10].

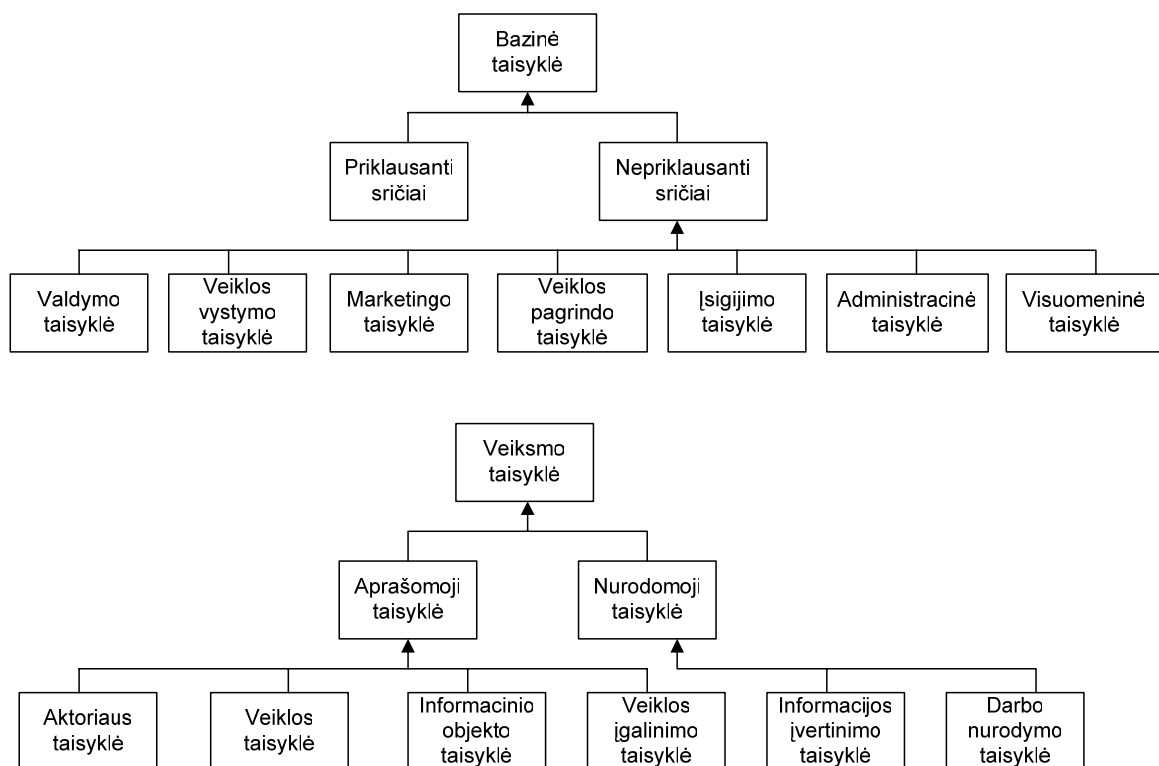
2.7 lentelė. B. V. Halle veiklos taisyklių klasifikacija

VT klasifikacija	Detalus VT paaiškinimas	VT šablonas
Terminas	Daiktavardis ar daiktavardinė frazė su iš anksto susitartu apibrėžimu. Pavyzdžiui: <ul style="list-style-type: none"> objekto klasė, esybė; elementas, atributas, reikšmė 	<i><term> IS DEFINED AS<text></i>
Faktas	Išraiška, sujungianti terminus prielinksnių ir veiksmažodžių pagalba į turinčias prasmę veiklos pastabas: <ul style="list-style-type: none"> esybė – esybė, esybė – atributas supertipo – subtipo sujungimas 	<i><term1> IS A<term2></i> <i><term1><verb><term2></i> <i><term1> IS COMPOSED OF<term2></i> <i><term1> IS A ROLE PLAYED BY<term2></i> <i><term1> HAS A PROPERTY OF<term2></i>
Skaičiavimas	Išraiška, nusakanti algoritmą skaitinei termino reikšmei apskaičiuoti. Skaičiuoja vieno termino reikšmę. Formulėje galima nurodyti reikiamą terminų skaičių, tačiau jie privalo būti iš anksto apibrėžti.	<i><term1> IS COMPUTED AS<formula></i>
Būtinumo apribojimas	Išraiška, išreiškianti sąlygas, kurios turi būti tenkinamos (arba netenkinamos) tam tikram veiklos atvejui	<i><term1> MUST HAVE<at least, at most, exactly n of><term2></i> <i><term1> MUST BE <comparison><term2>, <value>, <value list></i> <i><term1> MUST BE IN LIST<a,b,c></i> <i><term1> MUST NOT BE IN LIST<a,b,c></i> <i>IF<rule phrase(s)> THEN<constraint of any of the above types></i>
Rekomendacija	Išraiška, išreiškianti perspėjimą apie sąlygas, kurios turėtų būti tenkinamos (arba netenkinamos)	<i><term1> SHOULD HAVE<at least, at most, exactly n of><term2></i> <i><term1> SHOULD BE <comparison><term2>, <value>, <value list></i> <i><term1> SHOULD BE IN LIST<a,b,c></i> <i><term1> SHOULD NOT BE IN LIST<a,b,c></i> <i>IF<rule phrase(s)> THEN<constraint of any of the above guideline types></i>
Išvestis	Išraiška, kuri patikrina tam tikras sąlygas, ir joms esant teisingoms, patvirtina naujo fakto teisingumą	<i>IF <term1><operator><term2,value,value list> AND<again> THEN<term3></i> <i><operator><term4></i> <i>Where operator can be:</i> <i>Comparison (=, not =, =, >=, <>)</i> <i>Has quantity <at least n, at most n, exactly n>of</i>

VT klasifikacija	Detalus VT paaiškinimas	VT šablonas
Veiksmo įgalintojas	Išraiška, kuri patikrina tam tikras sąlygas ir inicijuoja kitą veiklos operaciją, pranešimą ar įvykį.	$IF<term1><operator><term2>THEN<action>$

2.8. Kiti klasifikavimo metodai

Yra pasiūlyta nemažai ir kitų autorių klasifikavimo metodų [2]. Vienas iš jų pateiktas [12] straipsnyje. Šiame straipsnyje autoriai siūlo VT klasifikuoti žiūrint į jas iš veiklos turinio ir veiklos proceso perspektyvų. Pirmą grupę taisyklių vadinama bazinėmis (angl. *intentional*) taisyklėmis. Jos išreiškia bendruosius organizacijos aspektus. Jų pagalba formuojamos veiksmo (angl. *operational*) taisyklės, kurios yra saugomos VT saugykloje. Šis klasifikavimas grafiškai pateiktas 2.3 pav.



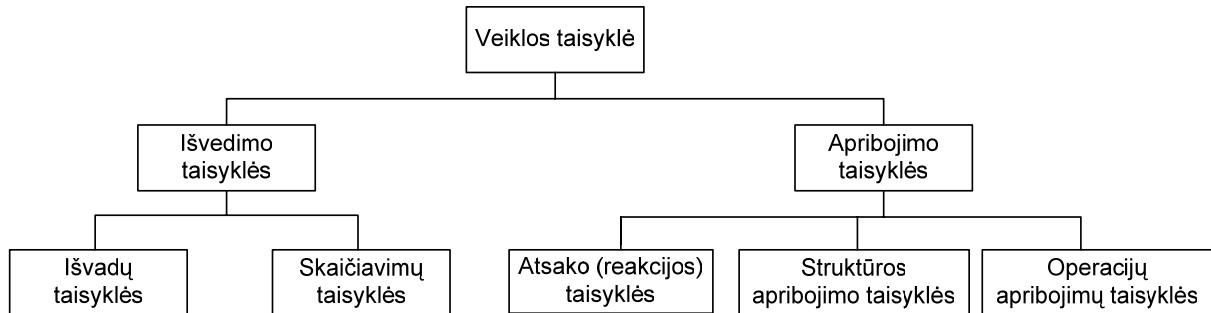
2.3 pav. Bazinių ir veiksmo taisyklių klasifikavimas

Dar vienas [18] VT klasifikavimo metodas veiklos taisykles grupuoja remiantis apdoravimo veiksmingumo ir sprendimo priėmimo lengvumo faktoriais:

- apdoravimo veiksmingumo faktorius:
 - nuorodos į apibrėžimus;
 - nuorodos į objektus;
 - nuorodos į faktus;
 - loginis.
- sprendimo priėmimo lengvumo faktorius:
 - loginis;

- kriterijų įvertinimai ir operacijos;
- veiklos politika.

Atsižvelgiant į tai, jog darbe numatoma sudaryti savo veiklos taisyklių klasifikavimo metodą, apžvelkime dar vieną klasifikavimo metodą, kuris daugiau tinka projektuotojui [17]. Šis klasifikavimo būdas panašus į GUIDE veiklos taisyklių projektą.



2.4 pav. Veiklos taisyklių klasifikavimas projektavimo požiūriu

Veiklos taisyklės dalinamos į dvi pagrindines taisyklių klases (2.4 pav.), t. y. išvedimo ir apribojimo taisyklės. Išvedimo taisyklės skaidomos į išvadų ir skaičiavimo taisykles. Apribojimo taisyklės skaidomos į atsako, operacijų ir struktūros apribojimo taisykles.

Atsako (reakcijos) taisyklės apriboja elgesį, nusako sąlygas, prie kurių galima vykdyti operaciją. Reakcijos taisyklės priklausomos nuo konteksto. Taisyklės *jeigu* sąlygą tikrinama tik tada, *kai* įvyksta tam tikro tipo įvykis. Tokios taisyklės sutinkamos darbų sekų arba būsenų perėjimo diagramose. Tokio tipo taisyklės dar vadinamos ECA (Event-Condition-Action) taisyklėmis.

Operacijų apribojimo taisyklės nurodo, kokios sąlygos turi būti tenkinamos prieš operaciją ir po jos, kad būtų užtikrintas vykdymo teisingumas. Tokie apribojimai yra svarbūs operacijos vykdymui ir nepriklauso nuo operacijos sužadinimo konteksto. Į šias taisykles galime žiūrėti kaip į kontraktą, siejantį operaciją su jos iškvietėjais. Operacijos apribojimo taisyklė atrodo taip: Jei iškviesi operaciją, kai įvykdyta prieš sąlyga, gausi galutinę būseną, kurioje bus patenkinta po-sąlyga.

Struktūros apribojimo taisyklės aprašo objektų ir jų sąryšių sąlygas, kurių negalima pažeisti. Struktūros apribojimo taisyklės savo ruožtu turi savo tipus: gali apriboti atributo reikšmes, esybių egzempliorius, ryšio kardinalumus. Šios taisyklės neturi nuorodų į operacijas, jos turi būti tenkinamos bet kuriomis sąlygomis.

Išvadų taisyklės aprašo, kokias išvadas galima padaryti iš tam tikrų faktų. Skaičiavimo taisyklės apibrėžia algoritmus, kuriuos vykdant galima gauti norimą rezultatą.

2.9. Veiklos taisyklių integravimo sistemoje būdai

Veiklos taisyklės iš prigimties yra nuolatos besikeičiančios, t. y. dinaminės [3]. Jos turi būti reguliariai atnaujinamos, kad atitiktų esamas rinkos sąlygas. Tradiciškai veiklos taisykles programuotojai naudoja procedūriškai – jos yra įdedamos į programos kodą. Toks darbo su veiklos taisyklėmis principas šiuo metu yra atgyvenęs ir neefektyvus, nes labai pasunkėja taisyklių redagavimas. Dažnai mažas pakeitimas reikalauja daug darbo ir resursų, bei iš esmės gali pakeisti visą sistemą.

Taisyklių atskyrimas nuo programinio kodo yra vienas svarbiausių žingsnių siekiant supaprastinti ir padaryti efektyvesniu VT valdymo mechanizmą. Naudojant VT valdymo sistemas, taisyklių organizavimą gali atlikti su organizacijos veikla susijęs analitikas, o ne vien tik programuotojas. VT saugyklos ir taikomųjų programų ryšys palaikomas per specialius programų kode įdėtus kreipinius. Dar dažnai naudojamas būdas, kuomet kodo generatoriai paimama veiklos taisykles iš saugyklos ir jos yra integruojamos į programos kodą, sugeneruotą tam tikrų projektinių modelių pagrindu. Tokiu būdu išsaugomas lankstus veiklos taisyklių kūrimo ir modifikavimo mechanizmas.

2.10. Klasifikavimo metodų apibendrinimas

Apibendrinant aptartus veiklos taisyklių klasifikavimo metodus galima teigti, kad GUIDE projektas [11] ir Roso modelis [22] yra naudingi kaip teorinė VT koncepcija, o MDC [15] modelis yra geriau pritaikytas praktiniam naudojimui. CommonKADS metodologijos pagrindu sudarytas metodas [24] naudingas bendresniam veiklos taisyklių supratimui. RAP [23] grupės pasiūlytas metodas, praplečia GUIDE metodo klasifikaciją, įvedant struktūrinės ir elgsenos dedamąsias. Barbaros von Halle klasifikacija [10] iš visų aptartųjų klasifikacijų yra aiškiausia ir priimtinausia. Paskutiniai trys aptarti veiklos taisyklių klasifikavimo metodai įrodo, kad nėra vieningos nuomonės VT klasifikavimo atžvilgiu. Kiekviena įmonė, organizacija ar pavieniai projektuotojai pagal poreikius kuria savo klasifikavimo metodus (dažnai remdamiesi visuotiniai pripažintais klasifikavimo metodais), kurie geriausiai atspindi jų veiklos sritį. Nepaskutinis dalykas yra ir jų realizacija, kurią projektuotojai laiko pirmoje vietoje, nors pagal visus veiklos taisyklių standartus klasifikuojant VT reiktų atsiriboti nuo realizacijos detalių.

Visi aptarti metodai, galima sakyti, yra deklaratyvūs ir visi autoriai pasisako už taisyklių specifikuojamą deklaratyviu būdu. Bet vienas taisykles geriau išreikšti ir saugoti deklaratyviai, o kitas procedūriškai.

3. VEIKLOS TAISYKLIŲ MODELIAVIMO PRIEMONIŲ APŽVALGA

3.1. Bendra veiklos taisyklių modeliavimo priemonių apžvalga

Veiklos taisyklės gali būti modeliuojamos daugeliu būdų [14]:

- natūralia kalba. Veiklos taisyklės pavaizduotos šiuo būdu gali būti interpretuojamos ir suprastos nevienareikšmiškai;
- naudojant šablonus. Veiklos taisyklės specifikuojamos naudojant pasirinktą šabloną;
- struktūriškai apibrėžta kalba. Naudojami natūralios kalbos komponentai, bet žodynas yra riboto dydžio. Veiklos taisyklės, užrašytos struktūriškai apibrėžta kalba, sunkiau yra suprantamos sistemos dalyviams. Sudėtingesnis jų teisingumo patikrinimas;
- formalios logikos sakiniais. Veiklos taisyklės užrašomos naudojant specialius simbolius;
- sprendimų lentelėmis. Lentelės sudarytos iš dviejų dalių. Viršutinėje dalyje užrašomos sąlygos, apatinėje – galimi veiksmai;
- sprendimų medžiais. Procesų logika pavaizduojama medžio pavidalo diagrama;
- objektų ribojimo kalba OCL (angl. *Object Constraint Language*). OCL yra UML notacija, skirta užrašyti specialioms ribojimams, kurių negalima specifikuoti UML diagramomis. OCL taip pat gali būti naudojama prieš- ir po-sąlygoms užrašyti;
- diagramomis (tame tarpe ir UML). VT gali būti vaizduojamos naudojant įvairias diagramas: ERA, IDEF, UML ir kt.;
- koncepciniais grafais. Veiklos taisyklės gali būti vaizduojamos naudojant naują kalbos „Knowledge base“ išplėtimo elementą „Rule base“.

Vienas dažniausiai naudojamų būdų yra VT specifikuojamas šablonais, nes šablonais užrašytos taisyklės yra dalinai formalizuotos, yra išlaikoma vienoda to paties tipo taisyklių struktūra, bei taisyklių šablonai laikomi pseudo-kodu [19], kas svarbu programuotojams. Šablonų naudojimas yra patogus būdas taisyklėms surinkti ir validuoti, tačiau gilesnei jų analizei reikėtų naudoti deklaratyvias formalias kalbas (pavyzdžiui, Z, OCL, Prolog ir kt.). Pagrindiniai šablonų tipai yra: terminų, faktų, skaičiuojamųjų, apribojimų, išvesties ir veiksmų veiklos taisyklių šablonai.

Formali logika yra formali kalba, kurią naudojant galima užrašyti sakinius, aprašančius faktus. Objektai ir ryšiai tarp jų užrašomi specialiais simboliais. Pavyzdžiui, simboliu A galima pažymėti veiklos agentų aibę, o konkretų veiklos agentą galima pažymėti simboliu X . Tada teiginį, kad agentas X yra veiklos agentas, galima pavaizduoti $X \in A$. Vaizduojant taisykles formalia logika dažniausiai, naudojami simboliai: \forall - naudojamas pavaizduoti, kad kiekvienas objektas tenkina tas pačias sąlygas; \wedge – loginis IR operatorius. \vee – loginis ARBA operatorius, $\bar{\quad}$ – loginis

neigimo operatorius. Formalūs modeliavimo metodai dažniausiai yra taikomi modeliuojant didelės svarbos sistemas.

Sprendimų lentelės sudarytos iš dviejų dalių: viršutinėje dalyje užrašomos sąlygos, apatinėje dalyje užrašomi galimi veiksmai. Jeigu yra n sąlygų (parametrų), kurios gali įgyti $k_1, k_2, k_3, \dots, k_n$ reikšmių, tai lentelėje bus $k_1 \times k_2 \times k_3 \times \dots \times k_n$ stulpelių. Lentelėje bus tiek eilučių, kiek yra galimų veiksmų. Šio vaizdavimo būdo pilnumas yra tame, kad išvardinamos visos galimos pradinių reikšmių kombinacijos.

Sprendimų medžiai kaip ir sprendimų lentelės gali būti naudojami procesų logikai užrašyti. Sprendimų medžio išsišakojimuose užrašomos sąlygos, o medžių lapai – veiksmai. Dažnai taisyklės pavaizduotas naudojant sprendimų lenteles, galima pavaizduoti ir sprendimų medžiais.

UML turi priemones aprašyti tam tikras taisykles, bet kitoms taisyklėms, kurių negalima pavaizduoti diagramomis, reikalinga formalizuota kalba. OCL yra formali ir tokiam atvejui tinkama UML komponentė, kuri įgalina vienareikšmiškai užrašyti VT.

Statinės taisyklės, aprašančios struktūros ribojimus, gali būti pavaizduotos UML klasių diagramomis. Šios taisyklės aprašomos kaip ribojimai klasių diagramose ir tampa metodų ar ryšių tarp klasių dalimi. Sudėtingi apribojimai dažnai užrašomi naudojant OCL. Veiklos taisyklės UML gali būti modeliuojamos naudojant užduočių, sekų, bendradarbiavimo, būsenų perėjimų, veiklos diagramas.

3.2. UML ir OCL

Dalis UML diagramų gali būti naudojamos veiklos taisyklių modeliavimui. Statinės taisyklės, aprašančios struktūros ribojimus, gali būti pavaizduotos UML klasių diagramomis. Šios taisyklės aprašomos kaip ribojimai klasių diagramose ir tampa metodų ar ryšių tarp klasių dalimi. Sudėtingi ribojimai dažnai užrašomi naudojant OCL. Klasių diagramose veiklos taisyklės gali būti specifikuotos kaip:

- klasių atributų duomenų tipai;
- klasių atributų reikšmės pagal nutylėjimą;
- ryšiai tarp klasių;
- ryšių tarp klasių kardinalumas.

Veiklos taisyklės UML gali būti modeliuojamos naudojant panaudojimo atvejų, sekų, bendradarbiavimo, būsenų perėjimų, veiklos diagramas. Dažnai sudėtingų taisyklių neįmanoma pavaizduoti, naudojant tik kurią nors vieną diagramą.

Dauguma taisyklių identifikuojamos specifikuojant funkcinius sistemos reikalavimus. Kadangi reikalavimų specifیکavimui naudojamos panaudojimo atvejų diagramos, tai šios diagramos gali būti naudojamos ir veiklos taisyklėms vaizduoti. Panaudojimo atvejų diagramose vaizduojama

kokie veiklos dalyviai, kokių užduočių vykdymą inicijuoja ir kaip tos užduotys yra susiję tarpusavyje. Užduotis šiose diagramose apibrėžiama, kaip dalykinė transakcija arba veiksmų, reikalingų kokiam nors tikslui pasiekti, visuma. Šio tipo diagramomis galima modeliuoti taisykles, apibrėžiančias sistemos aktorių pareigas ir kompetencijos ribas.

Sekų diagramomis detalizuojamos užduočių diagramoje pavaizduotos sistemos aktorių užduotys – aprašoma operacijų vykdymo logika, kurią nusako VT. Sekų diagrama pavaizduotas verslo taisyklių modelis aprašo sistemos elgseną. Šiose diagramose galima pavaizduoti operacijų eigą ir objektų tarpusavio sąveiką laiko atžvilgiu. Bendradarbiavimo diagramos mažai skiriasi nuo darbų sekų diagramų – jomis taip pat modeliuojama sistemos objektų sąveika (bendradarbiavimas). Vienintelis skirtumas, kad bendradarbiavimo diagramose nėra įvertinamas laikas.

Būsenų perėjimų diagramos naudojamos vaizduoti sistemos, sistemos dalies ar pasirinkto objekto elgseną, reaguojant į įvykius. Būsenų perėjimų diagramos dažniausiai naudojamos VT, pavaizduotoms panaudojimo atvejų diagramomis, detalizuoti. Taip vaizduojama VT logika, atsižvelgiant į klasių diagrama sumodeliuotą sistemos struktūrą. Būsenų perėjimo diagramos labiausiai tinka su verslo dinamika susijusioms taisyklėms vaizduoti, kadangi jose galima pavaizduoti verslo dinamiką, atsižvelgiant į verslo procesus įtakojančius įvykius.

Veiklos diagramos nedaug skiriasi nuo būsenų perėjimų diagramų. Būsenų perėjimų diagramose vaizduojamos objekto būsenos, kuriose jis būna savo gyvavimo cikle, ir perėjimai, kurie keičia objekto būsenas. Perėjimus inicijuoja veiklos, kurios vaizduojamos veiklų diagramose, ir susiję įvykiai. Veiklų diagramose specifikuojama operacijų seka. Kaip ir būsenų perėjimų diagramos, veiklų diagramos gali būti naudojamos verslo taisyklėms ir informacijos apdorojimo taisyklėms modeliuoti. Veiklų diagramos tai pat tinkamos detalizuoti VT, pavaizduotas užduočių diagramose, nes jose galima pavaizduoti ir sudėtingos operacijų sekos algoritmą. Tačiau naudojant veiklų diagramas sudėtinga pavaizduoti sistemos elgseną ar objektų tarpusavio sąveiką aprašančias taisykles. Dėl to jos negali pakeisti sekų, bendradarbiavimo ar būsenų perėjimų diagramų ir naudojamos tam, kad papildyti modelį.

Egzistuoja įvairių UML išplėtimų, kurie pasiūlyti siekiant pritaikyti UML veiklos taisyklėms modeliuoti. Tačiau UML išplėtimai sunkiai prigyja ir nėra plačiai taikomi dėl to, kad tokie išplėtimai nėra universalūs ir tinka tik tam tikro tipo taisyklėms modeliuoti, o komercinėse modeliavimo priemonėse, nėra siūlomų UML išplėtimų realizacijos.

UML turi specialiai taisyklių aprašymui sukurtą kalbą – OCL. OCL – yra formali logikos pagrindu paremta kalba, neturinti grafinės notacijos. OCL kalba tapo UML specifikacijos dalimi nuo 1.1 versijos. Ši kalba buvo sukurta IBM korporacijos laboratorijose ir naudojama papildomiems ribojimams objektiškai orientuotose modeliuose. Jos pagrindas – pirmos eilės

predikatų logika. Sakiniai teksto pavidalu gali būti prikabinėti prie UML klasės, arba klasės metodo.

OCL kalbos sakiniai negali egzistuoti be UML modelio, jų teisingumas gali būti tikrinamas tik atsižvelgiant į atitinkamą modelio realizaciją. OCL sakinio struktūra:

```
Context <Klasifikatorius> inv: <RibojimoPavadinimas>: Sąlyga
```

Kiekvienas OCL ribojimas prasideda nurodant kontekstą (angl. *context*), kuriame jis galioja. Kaip kontekstas gali būti naudojamas UML modelio klasifikatorius (pavyzdžiui, klasės pavadinimas ar atributas) arba klasifikatoriaus metodas. Priklausomai nuo ribojimo konteksto gali būti naudojama invariantas (raktinis žodis *inv*). Kai kontekstas klasifikatorius, arba prieš-sąlygos, raktinis žodis *pre*, ir po-sąlygos raktinis žodis *post*, kai kontekstas klasifikatoriaus metodas. Ribojimas visada baigiasi sąlyga. Invarianto sąlygą turi tenkinti kiekvienas modelio egzempliorius. Kuomet ribojimas uždedamas UML klasės metodei, prieš-sąlyga turi būti patenkinta prieš metodo vykdymą, o po-sąlyga turi būti teisinga po metodo vykdymo pabaigos.

OCL sakiniai aktyvioje duomenų bazėje atvaizduojami kelias skirtingais būdais, t. y. panaudojant tvirtinimo sakinius ir naudojant rodinius. Transformuojant OCL sakinius į ECA taisyklės (paprastai jie realizuojami kaip trigeriai) būtina įvertinti sekančius dalykus:

- trigerio sąlygą;
- trigerio suveikimo laiką;
- įvykį, kuris iššaukia trigerį;
- trigerio inicijavimo dažnį.

OCL invarianto sakinytis turi būti tenkinamas kiekvieno modelio egzemplioriaus, todėl būtina sekti kiekvieno OCL sąlygoje panaudoto atributo būklę. Trigeris turi būti iššauktas įterpianč/keičianč/trinant kortežus, kurių atributai yra panaudoti OCL sąlygoje. Trigerio suveikimo laikas numato kada bus patikrinta sąlyga: prieš įvykdant trigerio įvykį ar po jo. Įterpianč n įrašų iš eilės trigeris gali suveikti n arba 1 kartą, priklausomai nuo trigerio inicijavimo dažnio. Trigeriai gali turėti skirtingą inicijavimo dažnį.

Pažeidus OCL sakino ribojimą ECA taisyklė turi atlikti tam tikrą veiksmą. Šiuo atveju sistema gali tiesiog pranešti apie ribojimo pažeidimą arba atšaukti ribojimo pažeidimą. Toliau specifikuota transformacija įgyvendins transakcijos nutraukimą. Tai pat būtina atsižvelgti į įmanomus ribojimo pažeidimo atvejus transakcijos vykdymo metu, todėl visi ribojimai turi būti įvertinami prieš transakcijos pabaigą, t. y. turi būti naudojami taip vadinamos atidedami (angl. *deferred*) trigeriai.

3.3. EER diagramos

Šios diagramose veiklos taisyklės gali būti specifikuotos kaip:

- esybių/lentelių atributų duomenų tipai;
- esybių/lentelių atributų reikšmės pagal nutylėjimą;
- ryšiai tarp esybių/lentelių;
- ryšių tarp esybių/lentelių kardinalumas;
- pirminiai ir išoriniai raktai.

Keičiant EER diagramas standartizuotu reliacinės duomenų bazės aprašu galima patirti semantinius nuostolius, nes ne visos transformacijos išsaugo schemų semantiką. Yra trys geriausiai žinomos transformavimo taisyklės, leidžiančios tiksliausiai atvaizduoti reliacinės duomenų bazės dalykinės srities semantiką. Pirmai grupei priskirtos taisyklės, kuriomis EER diagrama pertvarkoma EER modelio ribose. Antrajai grupei priskirtos taisyklės, kuriomis eliminuojami apibendrinimo ryšiai ir EER modelis paverčiamas ER. Trečiajai grupei priskirtos taisyklės, kuriomis ER modelis verčiamas logine reliacinės duomenų bazės schema.

3.4. Veiklos taisyklių modeliavimo priemonių apibendrinimas

Susipažinta su veiklos taisyklių modeliavimo priemonėmis ir jų užrašymo būdais. Skirtingos modeliavimo kalbos ir jose naudojamos diagramos nevienodai tinkamos skirtingų klasių taisyklėms modeliuoti. Ne visų tipų taisyklės galima modeliuoti naudojant vieną modeliavimo kalbą ar joje naudojamą tam tikrą diagramą. Veiklos taisyklių modeliavimas naudojant UML ir OCL, bei jų realizacija duomenų bazių technologijomis yra vienas populiariausių būdų, bet iškyla problemų. OCL kalba neturi grafinės notacijos ir ją sunku suprasti verslo atstovams, bei siūlomi būdai nenagrinėja taisyklių, kurios specializuoja verslo algoritmus, procedūras ir jų automatinę realizaciją negalima. Taisyklių vaizdavimas EER diagramomis yra taip pat paplitęs, bet jų keitimas reliacinių duomenų bazių aprašais yra gana sudėtingas ir komplikotas.

4. PRIEMONĖS VEIKLOS TAISYKLĖMS REALIZUOTI INFORMACINĖSE SISTEMOSE

4.1. Veiklos taisyklių sistemos samprata ir jos naudojimo tikslingumas

Veiklos taisyklės dažniausiai realizuojamos taikomosiose programose ir duomenų bazių valdymo sistemose. Tai prieštarauja veiklos taisyklių koncepcijai, pagal kurią veiklos taisyklės turi užimti atskirą informacinės sistemos vienetą. Todėl veiklos taisyklėms projektuoti, redaguoti, trinti, papildyti ir valdyti naudojamos taip vadinamos veiklos taisyklių valdymo sistemos. Tokių sistemų esmė yra taisyklių atskyrimas (loginis ir fizinis) nuo taikomųjų programų bei duomenų bazių. Taisyklės projektuojamos atskirai bei saugomos veiklos taisyklių saugykloje. Tokių sistemų naudojimas sutrumpina taikomosios programos kūrimo laiką ir žymiai sumažina palaikymo bei tobulinimo kaštus. Taisyklės gali keisti ne tik programuotojai, bet ir kiti veiklos atstovai.

Veiklos taisyklių valdymo sistemos naudojamos tam, kad taisyklių vykdymo procesas būtų greitesnis ir efektyvesnis. Tokios sistemos naudoja sudėtingus algoritmus kompiuterinių resursų optimizavimui vykdant veiklos taisykles. Integruojant veiklos taisyklių valdymo sistemą į verslo sistemą kiekviena taisyklė užrašoma kaip nepriklausomas teiginys, aprašantis kokį nors veiklos aspektą. Tuomet veiklos taisyklės tuomet dirba kaip autonominis vienetas, atskirtas nuo taikomosios programos logikos. Pagrindinės priežastys, kodėl reikia naudoti veiklos taisyklių valdymo sistemas būtų:

- Veiklos taisyklių atskyrimas nuo taikomosios programos. Tai leidžia keisti veiklos logiką be programuotojų, nekeičiant taikomosios programos.
- Sutrumpina taikomosios programos kūrimo laiką ir žymiai sumažina palaikymo bei patobulinimo kaštus.

Veiklos taisyklių valdymo sistemas rekomenduojama naudoti kai:

- programa apima sprendimų priėmimą;
- veiklos taisyklės yra sudėtingos;
- taisyklės yra dažnai keičiamos;
- taisyklės turi būti naudojamos keliuose programose ir/arba organizacijose.

Veiklos taisyklių valdymo sistemų nerekomenduojama naudoti kai:

- taisyklės yra statinės ir išskaičiuojamos;
- taisyklės yra paprastos, vienodos, kartojamos;
- greitis ir našumas yra svarbesnis už lankstumą ir palaikymo kaštus.

Yra nemažai įvairių veiklos taisyklių valdymo sistemų. Jos atlieka skirtingas funkcijas, vykdo sprendimus įvairiausiuose lygmenyse, jų paskirtis taip pat yra skirtinga. Veiklos taisyklių valdymo sistemas galima klasifikuoti pagal tai, kokių tipų taisykles jos gali valdyti:

- paprastos veiklos taisyklės (angl. *simple business rules*) – taisyklės, kurios gali būti išreikštos naudojant paprastą žodyną ir išskviečiamos veiklos proceso arba programos kintamumo taškuose. Šiai grupei galima priskirti tokius produktus kaip ILOG, BlazeAdvisor, BRBeans, QuickRules ir kt.
- dirbtinės intelektualios taisyklės (angl. *artificial intelligence rules*) – taisyklės, kurios įtakoja algoritmus. Šiai grupei galima priskirti tokius produktus kaip Intelligent Mineramp, DB2Amp ir kt.
- įvykių apibendrinimo taisyklės (angl. *event correlation rules*) – taisyklės, kurios yra naudojamos įvykių apibendrinimui. Šiai grupei galima priskirti tokius sistemų valdymo produktus kaip Tivoliamp, Eventconsole ir kt.
- taisyklės orientuotos į duomenis (angl. *data-centric rules*) – taisyklės, valdančios duomenų užklausimą ir atnaujinimą. Šios taisyklės nustato kaip duomenys turi būti transformuojami, kas gali prie jų prieiti, užtikrina duomenų darną ir vientisumą. Šiai grupei galima priskirti tokius sistemų valdymo produktus kaip Versata.

4.2. Veiklos taisyklių valdymo sistemų apžvalga

Egzistuoja daug įvairiausių veiklos taisyklių valdymo sistemų. Jos atlieka skirtingas funkcijas, daro verslo sprendimus įvairiausiuose lygmenyse. Dažnai verslo atstovai veiklos taisyklių sistemas tapatina su veiklos taisyklių programomis [7], kurios pasižymi sekančiomis savybėmis:

- galimybe greitai pakeisti taisykles be programos kodo modifikavimo;
- turi taisyklių vykdymo mechanizmą, kuris lengvai integruojasi į programų architektūras;
- pasižymi aukšto lygio veiklos taisyklių kalba, kurią supranta ir verslo atstovai, ir sistemos vartotojai;
- turi taisyklių redaktorių.

Pagrindiniai veiklos taisyklių sistemų skirtumai yra kokio lygio sprendimams jos yra skirtos, kokiose platformose gali dirbti ir kokios naudojamos VT aprašymo kalbos. Kainos svyruoja nuo nulio iki kelių milijonų dolerių.

VT sistemų kūrėjai bando unifikuoti VT kūrimą, saugojimą ir apsikeitimą. Vis dažniau naudojamas XML (Quick Rules, ILOG Rules, Common Knowledge, CommonRules), kaip standartas VT saugojimui. Bandoma naudoti tas pačias kalbas ir taisyklių užrašymui. Taisyklių

apsikeitimas tarp skirtingų sistemų nėra dar realizuotas arba realizuotas labai prastai. Šiuo metu rengiamas OMG standartas [4], kaip tikimasi, padės išspręsti šias problemas.

Usoft VT specifikavimui naudoja ANSI SQL [10], kas labai palengvina bendravimą tarp reliacinių duomenų bazių. Versata VT surinkimui ir realizavimui naudoja deklaratyvias taisykles, kurios yra išreiškiamos naudojant struktūrinę anglų kalbą, o ne procedūrinį kodą. HNC Software produktuose VT yra specifikuojamos naudojant Advisor Structured Rule Language (SRL). SRL yra programavimo kalba, kurios sintaksė yra labai panaši į anglų kalbą, kas padaro ją suprantama ne vien programuotojams. Blaze Advisor taisyklėms aprašyti naudoja struktūrinę taisyklių kalbos sintaksę, taip pat leidžia kurti taisyklių šablonus bei sprendimų lenteles. Infrex taisyklės aprašomos deklaratyvia forma, galimi šablonai, be to, taisykles galima apibrėžti sprendimų medžiu, sprendimų lentele ar orientuotu grafu. ILOG naudoja Business Rule Language (BRL). ILOG taisyklės saugomos kaip Java ar C++ objektai, priklausomai nuo to koks VT variklis yra naudojamas. ILOG VT varikliai gali būti C++, Java ir EJB aplikacijos. Java privalumas yra jos sinchronizavimo galimybės, kurios leidžia daugiagijoms programoms dalintis objektais ir taisyklėmis, bei keliems VT varikliams dirbti su tais pačiais programų objektais. Kuomet reikalingas didesnis greitis rekomenduojama C++. Versata taisyklės yra kompiliuojamos ir saugomos kaip Enterprise JavaBeans arba CORBA objektai, t. y. procedūriškai. USoft aplinka leidžia sistemą įdiegti bet kurioje fizinėje aplinkoje, nereikalaujant didelių pokyčių veiklos modelyje. Kitais žodžiais tariant, sukurtos VT gerai veiks tiek ant Oracle ar DB/2. Taip pat galima įdiegti USoft aplikaciją ir pasauliniame žiniatinklyje. Visais atvejais taisyklių variklis yra nedidelis rinkinys binarinių vykdomųjų failų, sugeneruotų iš apibrėžimų saugomų reliacinėje saugykloje.

Veiklos taisyklių varikliai (angl. *business rules engine*) būna dviejų tipų [16]:

- pagrįsti duomenimis (angl. *data driven*). Tokiuose varikliuose taisyklės tiesiogiai nurodomos per duomenų modelį ir tuomet, kai vartotojas nori pakeisti duomenis, interpretatorius inicijuoja atitinkamus veiksmus (pavyzdžiui, Usoft, Infrex ir kt.);
- išvedimo mechanizmas (angl. *inference engine*). Tokie varikliai užfiksuoja žinias ir saugo jas taisyklių forma (pavyzdžiui, Blaze Advisor ir kt.).

Apibendrinami galime teigti, kad dauguma veiklos taisyklių valdymo sistemų:

- palaiko pilną transakcijų vykdymą;
- VT varikliai valdo prieigą prie duomenų bazės ir vykdo iš anksto numatytas taisykles, kuomet yra papildomi/keičiami duomenys;
- pirmiausia taisyklės yra specifikuojamos natūralia kalba, po to, deklaratyviai, produkto VT kalba;
- taisyklių projektavimas kartais reikalauja jų vykdymo sekos nurodymo;
- vartotojui yra suteikiama pagalba įvedant taisykles;

- VT produktas tarnauja kaip centrinis taškas taisyklių apibrėžime ir valdyme;
- vartotojui reikia labai mažai programuoti produkte naudojama VT specifikavimo kalba, nes dauguma reikalavimų yra išreiškiami deklaratyviai;
- dauguma taisyklių yra išreiškiami deklaratyviai (vartotojų patogumui), bet VT varikliai saugojimui jas transformuoja į procedūrinį kodą;
- turi aiškiai apibrėžtą metamodelį, kuris gali (bet ne visada) būti reikalui esant praplėstas.

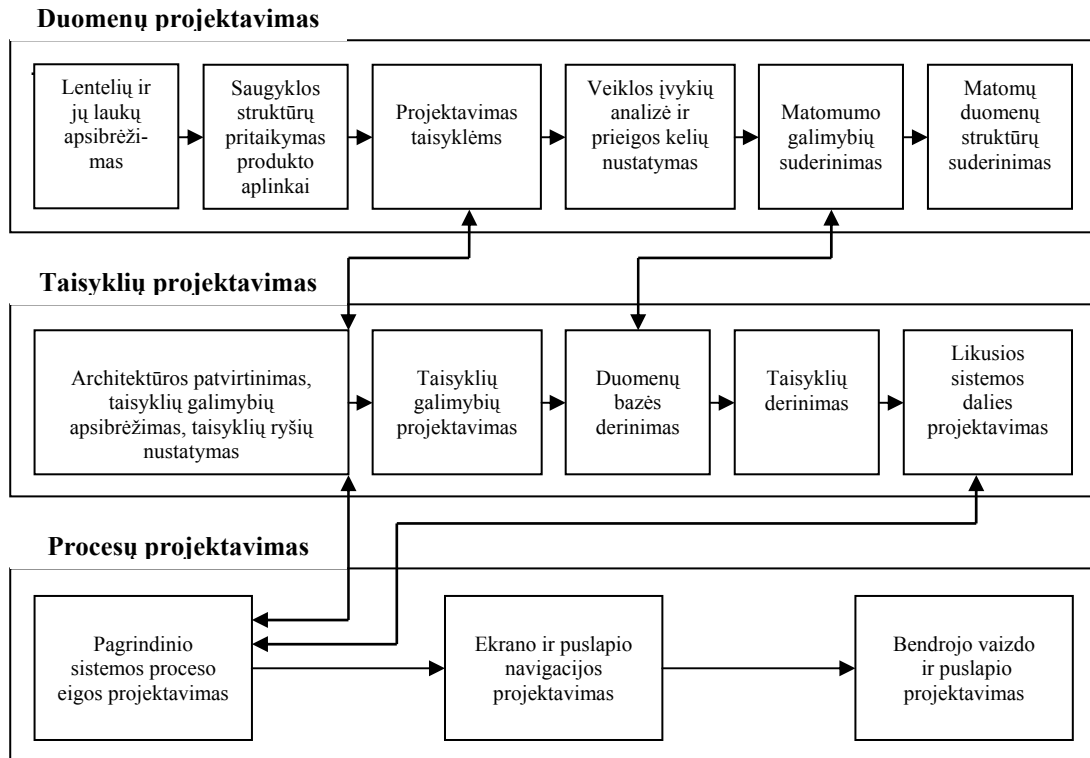
4.3. Veiklos taisyklių saugyklos projektavimo koncepcija

Yra daug būdų kaip projektuoti VT saugyklas. Visada reikia turėti omenyje, kad taisyklės turi būti prieinamos verslo atstovų auditorijai taip pat kaip ir sistemų vystymo profesionalams [10]. Veiklos taisyklių saugyklos pasirinkimo variantai:

- pačių suprojektuota veiklos taisyklių saugykla;
- praplėstas CASE įrankis;
- praplėstas komercinis duomenų bazės įrankis;
- komercinis taisyklių produktas.

Kiekvienu atveju reikia išanalizuoti kiekvieno varianto privalumus ir trūkumus bei pasirinkti sau tinkamiausią. Pagrindiniai kriterijai būtų: kitų modeliavimo technologijų turėjimas, biudžetas, esamos saugyklos galimybės (jei ji turima), turimo komercinio produkto saugyklos galimybės (jei toks produktas turimas).

Veiklos taisyklių saugyklos projektavimas yra labai sudėtingas dalykas apimantis visą taisyklių struktūrizavimo procesą. Yra įvairių metodikų tam atlikti. Aptarkime vieną iš jų. Jos autorius yra Barbara von Halle [10]. Autorė pateikia gana abstraktų aprašymą kaip apsiungia duomenų, procesų ir taisyklių projektavimas VT sistemos kūrimo metu (4.1 pav.). Pirmoji taisyklių projektavimo stadija vaizduoja susitelkimą ties taisyklių galimybėmis, kaip dalies technologinės ir programinės architektūros, planavimu. Antroji stadija vaizduoja, kad reikia priimti sprendimus susijusius su taisyklių galimybėmis. Šioje stadijoje loginis taisyklių modelis yra pertvarkomas į preliminarų taisyklių projektą. Preliminarus ar išankstinis taisyklių projektavimas turi būti nepriklausomas nuo būsimos realizavimo platformos, bet turi būti išskirti trys pagrindiniai lygmenys: procesų eigos, duomenų valdymo ir taisyklių vykdymo. Preliminarus taisyklių projektas parodo architektūrinius ryšius, bei duomenimis, taisyklėmis ir procesais paremtą projektavimo požiūrį. Kadangi technologijos nėra tobulos, tai trečioji stadija vaizduoja, kad pirmoje eilėje reikia apsvastyti duomenų bazės galimybės, o tik po to taisyklių projektavimą joje. Paskutinė stadija iliustruoja, likusios sistemos dalies projektavimą. Likusios sistemos dalies projektavimas, kuomet suprojektuota duomenų bazė ir taisyklės, apima procesų eigos projektavimą.



4.1 pav. Veiklos taisyklių sistemos projektavimo koncepcija

4.4. Apibendrinimas

Apžvelgtos priemonės veiklos taisyklėms realizuoti informacinėse sistemose, išskirtos jų bendrosios savybės. Dauguma taisyklių yra išreiškiamos deklaratyviai (vartotojų patogumui), bet dalis apžvelgtų VT variklių saugojimui jas transformuoja į procedūrinį kodą (objektus). Realizuojant taisykles jos gali būti aprašomos ir procedūriškai. Veiklos taisyklių valdymo sistemų funkcionalumas iš esmės yra gana panašus. Pagrindiniai šių sistemų skirtumai yra:

- kokiems veiklos taisyklių tipams jos yra skirtos;
- kokio lygio sprendimams;
- kokiose platformose gali dirbti;
- kokios naudojamos veiklos taisyklių aprašymo kalbos;
- koks naudojamas veiklos taisyklių variklis.

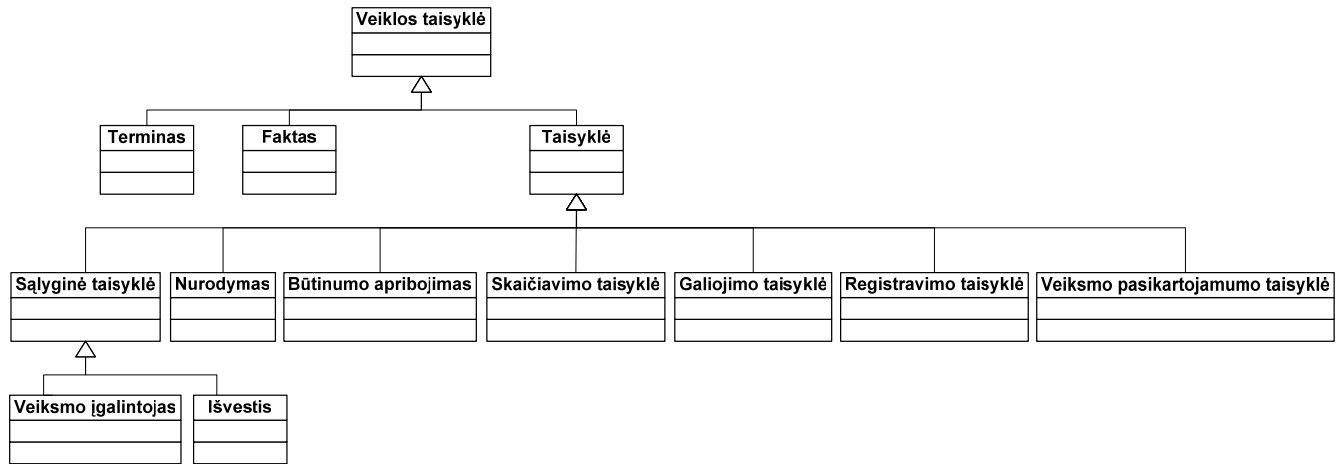
Veiklos taisyklių valdymo sistemų kaina gali svyruoti nuo nulio iki kelių milijonų dolerių, be to vartotojų skaičius, dirbančių su sistema gali būti iki kelių tūkstančių. Veiklos taisyklių apsisikeitimas tarp skirtingų sistemų arba nėra realizuotas, arba realizuotas labai prastai.

Pateikti veiklos taisyklių saugyklos pasirinkimo variantai, bei bendra veiklos taisyklių sistemos projektavimo koncepcija.

5. VEIKLOS TAISYKLIŲ KLASIFIKAVIMO METODO SUDARYMAS

5.1. Pasiūlytas veiklos taisyklių klasifikavimo metodas

Atlikus veiklos taisyklių klasifikavimo metodų analizę, buvo sudarytas VT klasifikavimo metodas, kurio pagrindą sudaro Barbaros von Halle [10] ir CommonKADS [24] veiklos taisyklių klasifikavimo principai. Sudarytas veiklos taisyklių klasifikavimo metodas pateiktas 5.1 pav. Kiekvieną taisyklę galima priskirti atskiram veiklos objektui (esybei) [2].



5.1 pav. Veiklos taisyklių klasifikacija pasiūlytu metodu

Pasiūlyta VT klasifikacija ir detalus kiekvienos taisyklės paaiškinimas pateiktas 5.1 lentelėje. Kadangi tolesniame tyrime didžiausias dėmesys bus skiriamas veiklos taisyklių užrašymu šablonais, tai lentelėje pateikiami ir atitinkamų veiklos taisyklių šablonai. Kaip teigia M. Bajec ir M. Krisper [1], kiekvienas veiklos taisyklės tipas turi turėti šabloną. Veiklos taisyklės šablonas parodo, kaip specifikuoti taisyklę, priklausančią tam tikram veiklos taisyklių tipui.

5.1 lentelė. Pasiūlytos veiklos taisyklių klasifikacijos aprašymas

VT klasifikacija	Detalus VT paaiškinimas	VT šablonas
Terminas	Daiktavardis ar daiktavardinė frazė su iš anksto susitartu apibrėžimu. Pavyzdžiui: <ul style="list-style-type: none"> objekto klasė, esybė; elementas, atributas reikšmė, reikšmių rinkinys 	<i><term> IS DEFINED AS<textual definition></i>
Faktas	Išraiška, sujungianti terminus prielinksnių ir veiksmažodžių pagalba į turinčias prasmę veiklos pastabas: <ul style="list-style-type: none"> ryšys esybė – esybė priskyrimas 	<i><term1> IS A<term2></i> <i><term1><verb><term2></i> <i><term1> MAY<verb><term2></i> <i><term1> IS COMPOSED OF<term2></i> <i><term1> IS A ROLE PLAYED BY<term2></i> <i><term1> HAS A PROPERTY OF<term2></i>

VT klasifikacija	Detalus VT paaiškinimas	VT šablonas
	esybė – atributas <ul style="list-style-type: none"> ▪ supertipo – subtipo sujungimas 	
Skaičiavimo taisyklė	Išraiška, nusakanti algoritmą skaitinei termino reikšmei apskaičiuoti. Skaičiuoja vieno termino reikšmę. Formulėje galima nurodyti reikiamą terminų skaičių, tačiau jie privalo būti iš anksto apibrėžti	$\langle term1 \rangle IS COMPUTED AS \langle formula \rangle$
Būtinumo apribojimas	Išraiška, išreiškianti sąlygas, kurios turi būti tenkinamos (arba netenkinamos) tam tikram veiklos atvejui	$\langle term1 \rangle MUST HAVE \langle at\ least, at\ most, exactly\ n\ of \rangle \langle term2 \rangle$ $\langle term1 \rangle MUST \langle compare\ to \rangle \{ \langle term2 \rangle, term\ value, constant \}$ $\langle term1 \rangle MUST BE \langle comparison \rangle \langle term2 \rangle, \langle value \rangle, \langle value\ list \rangle$ $\langle term1 \rangle MUST BE IN LIST \langle a, b, c \rangle$ $\langle term1 \rangle MUST NOT BE IN LIST \langle a, b, c \rangle$ $IF \langle rule\ phrase(s) \rangle THEN \langle constraint\ of\ any\ of\ the\ above\ types \rangle$
Nurodymas	Išraiška, išreiškianti perspektyvą, apie sąlygas, kurios turi būti tenkinamos (arba netenkinamos)	$\langle term1 \rangle SHOULD HAVE \langle at\ least, at\ most, exactly\ n\ of \rangle \langle term2 \rangle$ $\langle term1 \rangle SHOULD BE \langle comparison \rangle \langle term2 \rangle, \langle value \rangle, \langle value\ list \rangle$ $\langle term1 \rangle SHOULD BE IN LIST \langle a, b, c \rangle$ $\langle term1 \rangle SHOULD NOT BE IN LIST \langle a, b, c \rangle$ $IF \langle rule\ phrase(s) \rangle THEN \langle constraint\ of\ any\ of\ the\ above\ guideline\ types \rangle$
Išvestis	Išraiška, kuri patikrina tam tikras sąlygas, ir joms esant teisingoms, patvirtina naujo fakto teisingumą	$IF \langle term1 \rangle \langle operator \rangle \langle term2, value, value\ list \rangle AND \langle again \rangle THEN \langle term3 \rangle$ $\langle operator \rangle \langle term4 \rangle$ <i>Where operator can be:</i> <i>Comparison (=, not =, =<, >=, <>)</i> <i>In, not in</i> <i>Has quantity <at least n, at most n, exactly n> of</i>
Veiksmo įgalintojas	Išraiška, kuri patikrina tam tikras sąlygas, ir jei jos duoda teigiamą rezultatą, tai inicijuoja kitą veiklos operaciją, pranešimą ar kitą įvykį.	$IF \langle term1 \rangle \langle operator \rangle \langle term2 \rangle THEN \langle action \rangle$
Galiojimo taisyklė	Išraiška, kuri parodo kiek laiko turi būti laikoma (arba kada panaikinama) informacija apie tam tikrą objektą	$\langle term1 \rangle IS VALID FOR \langle period \rangle$ $\langle term1 \rangle MUST BE DESTROYED after \langle period, action \rangle$
Registravimo taisyklė	Išraiška, kuri nusako ar kaupti visas tam tikro atributo reikšmes	$All\ changes\ in\ the\ value\ of\ \langle term1 \rangle MUST\ BE\ RECORDED$

VT klasifikacija	Detalus VT paaiškinimas	VT šablonas
Veiksmo pasikartojamumo taisyklė	Išraiška, nurodanti kaip dažnai bus atliekamas tam tikras veiksmas	<i><action>WILL BE PERFORMED<number>TIMES EVERY<time measure></i>

Toliau pateikiami lietuviški veiklos taisyklių šablonų variantai ir pavyzdžiai (5.2 lentelė).

5.2 lentelė. Veiklos taisyklių šablonai ir pavyzdžiai

VT tipas	VT šablonas	Pavyzdys
Terminas	<i><terminas>YRA APIBRĖŽIAMAS KAIP<tekstas></i>	Rinkėjas YRA APIBRĖŽIAMAS KAIP pilietis, turintis rinkimo teisę
Faktas	<i><terminas1>YRA<terminas2> <terminas1>GALI<veiksmažodis><terminas2> <terminas1>YRA SUDARYTAS IŠ<terminas2> <terminas1>TURI SAVYBĘ<terminas2></i>	Studento numeris YRA studento atributas
Skaičiavimo taisyklė	<i><terminas1>YRA PASKAIČIUOJAMAS KAIP<formulė></i>	Galutinė kaina YRA PASKAIČIUOJAMA KAIP kaina + PVM
Būtinumo apribojimas	<i><terminas1>TURI TURĖTI<bent vieną, visus, ...><terminas2> <terminas1>TURI BŪTI<palyginimas> {<terminas2><reikšmė>/<reikšmių sąrašas>} <terminas1>TURI BŪTI SĄRAŠE<a, b, c> <terminas1>NETURI BŪTI SĄRAŠE<a, b, c> JEI<veiklos frazė>TUOMET<bet kuris iš aukščiau išvardintų apribojimų></i>	Automobilis TURI TURĖTI pirmos pagalbos vaistinėlę. Studento numeris TURI BŪTI unikalus. Galima reikšmė TURI BŪTI SĄRAŠE <100, 200, 300>
Nurodymas	<i><terminas1>TURĖTŲ TURĖTI<bent vieną, visus, ...><terminas2> <terminas1>TURĖTŲ BŪTI<palyginimas> {<terminas2>, <reikšmė>, <reikšmių sąrašas>} <terminas1>TURĖTŲ BŪTI SĄRAŠE<a, b, c> <terminas1>NETURĖTŲ BŪTI SĄRAŠE<a, b> JEI<verslo taisyklė>TUOMET<bet kuris iš aukščiau išvardintų apribojimų></i>	Bendrabutis TURĖTŲ TURĖTI budėtojus Laisvo kambario numeris TURĖTŲ BŪTI SĄRAŠE <201, 203, 304, 906>
Išvestis	<i>JEI <terminas1><operatorius><terminas2, reikšmė, reikšmių sąrašas>IR<...>TUOMET <terminas3><operatorius><terminas4> Kur operatorius gali būti: =, ≠, <, >, ≥, ≤ Yra, nėra Turi kiekį<mažiausiai n, daugiausiai n, lygiai n></i>	JEI kompiuterio kaina > 2500 IR pirkimo kiekis > 5 TUOMET nuolaida = 5 procentams
Veiksmo įgalintojas	<i>JEI<terminas1><operatorius><terminas2> TUOMET<veiksmas></i>	JEI kliento statusas = neaptarnautas TUOMET inicijuoti kliento aptarnavimo operaciją
Galiojimo taisyklė	<i><terminas1>GALIOJA<periodas> <terminas1>TURI BŪTI SUNAIKINTAS po <periodas, veiksmas></i>	Užsakymas GALIOJA 5 dienas Bylos TURI BŪTI SUNAIKINTOS po 5 metų

VT tipas	VT šablonas	Pavyzdys
Registravimo taisyklė	<i>Visi<terminas I><...>TURI BŪTI REGISTRUOJAMI</i>	Visi kliento užsakymai TURI BŪTI REGISTRUOJAMI
Veiksmo pasikartojamumo taisyklė	<i><veiksmas>BUS ATLIEKAMAS<skaičius> KARTŲ PER<laiko periodas></i>	Automobilio techninė apžiūra ATLIEKAMA 1 KARTĄ PER 2 metus

5.2. Veiklos taisyklių specifikuojimo būdo parinkimas

Deklaratyviai ar procedūriškai specifikuoti veiklos taisyklę? Tai yra vienas pagrindinių šio tyrimo klausimų. Žemiau pateiktoje 5.3 lentelėje yra pasiūlyta kaip tam tikrą taisyklės tipą geriau specifikuoti.

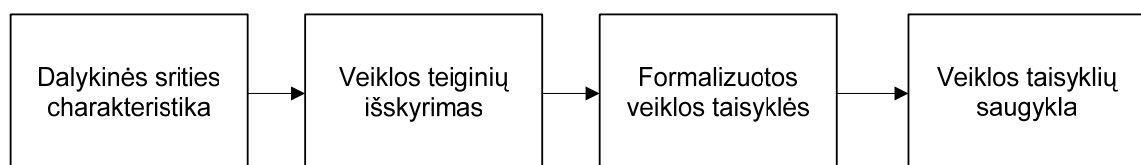
5.3 lentelė. Veiklos taisyklių specifikuojimo metodo parinkimas

VT tipas	Specifikavimo būdas
Terminas	Deklaratyvusis
Faktas	Deklaratyvusis
Skaičiavimo taisyklė	Deklaratyvusis ir(arba) procedūrinis
Būtinumo apribojimas	Deklaratyvusis
Nurodymas	Deklaratyvusis
Išvestis	Deklaratyvusis ir procedūrinis
Veiksmo įgalintojas	Deklaratyvusis ir procedūrinis
Galiojimo taisyklė	Deklaratyvusis ir procedūrinis
Registravimo taisyklė	Procedūrinis arba deklaratyvusis
Veiksmo pasikartojamumo taisyklė	Procedūrinis ir deklaratyvusis

5.3. Veiklos taisyklių struktūrizavimo proceso aprašymas

Sudarius VT klasifikavimo metodą turime apsirašyti bendrą veiklos taisyklių struktūrizavimo procesą, kuriuo remsimės tolesnio darbo metu. Sudaryto proceso eiga pateikta 5.2 pav. Struktūrizavimo procesas padalinamas į keturis etapus [2]:

1. Dalykinės srities charakteristika. Šiame etape atliekama įmonės ar organizacijos veiklos specialistų ir būsimų vartotojų įvairios apklausos, t. y. bandoma surinkti kuo daugiau informacijos apie tam tikrą sritį.
2. Veiklos taisyklių teiginių išskyrimas. Išskiriant teiginius remiamasi C. J. Date [9] reikalavimais bei veiklos taisyklių manifestu [5].
3. Veiklos taisyklių užrašymas formaliu pavidalu. Šiame etape remiamasi pasiūlytais veiklos taisyklių klasifikavimo šablonais.
4. Veiklos taisyklių saugyklos projektavimas. VT saugykla bus projektuojama norint kaupti visų tipų taisykles bei numatant tolimesnes jos vystymo ir panaudojimo perspektyvas.



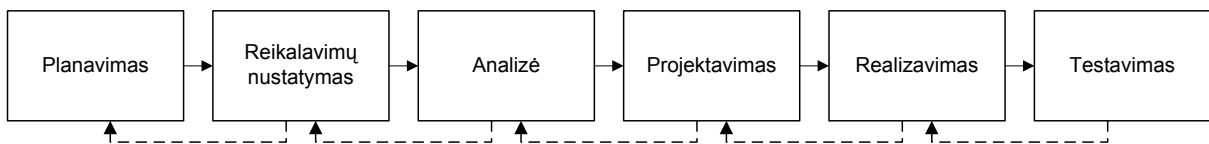
5.2 pav. Veiklos taisyklių struktūrizavimo procesas

6. VEIKLOS TAISYKLIŲ SAUGYKLOS PROJEKTAVIMAS

6.1. Sistemos kūrimo procesas veiklos taisyklių metodologijoje

Lyginant tradicinę informacinių sistemų projektavimo metodologiją su veiklos taisyklėmis paremta projektavimo metodologija ryškių skirtumų nėra. Tačiau yra skirtumų, kurie atsiranda projektavimo metu didesnę dėmesį skiriant veiklos taisyklėms. Veiklos taisyklėmis grįstos informacinės sistemos savitumą lemia tai, jog taisyklės yra laikomos kaip atskiros informacinės vertybės ir turi atskirą vietą sistemoje. Jos turi būti išreiškiamos tam tikra fiksuota forma. Taip pat jos turi būti lengvai keičiamos.

Sistemos kūrimo procesas veiklos taisyklių metodologijos atveju susideda iš šešių fazių (6.1 pav.): planavimo, reikalavimų nustatymo, analizės, projektavimo, realizavimo ir testavimo. Reikalavimų nustatymo, analizės, projektavimo ir realizavimo fazėse reikia įvertinti keturis aspektus: technologiją, procesus, taisykles ir duomenis.



6.1 pav. Sistemos kūrimo procesas veiklos taisyklių metodologijoje

Planavimo fazėje atliekamas aukšto lygio reikalavimų identifikavimas. Aukšto lygio reikalavimai apibrėžia projektuojamos sistemos pagrindinius tikslus, strategiją bei tolimesnį darbo planą.

Reikalavimų nustatymo fazės metu atliekama detali vartotojo poreikių (reikalavimų) analizė atskirai nagrinėjant procesų, duomenų ir taisyklių pobūdžio reikalavimus. Šioje fazėje priimamas sprendimas kokia veiklos taisyklių saugykla bus naudojama. Galimi saugyklos variantai: pačių suprojektuota veiklos taisyklių saugykla, tam tikras CASE įrankis ar praplėsta esama duomenų bazė.

Analizės fazės metu yra sutvarkoma ankstesnėse fazėse surinkta informacija ir projektuojamas loginis taisyklių modelis. Atliekama duomenų, taisyklių ir procesų analizė. Šio etapo metu paaiškėja prieš tai atliktų projektavimo fazių teisingumas ir galimi prieš tai atliktų fazių pataisymai. Kalbant konkrečiau, visi trys pirmieji projektavimo etapai gali būti vykdomi lygiagrečiai. Loginis taisyklių modelis, kaip ir loginis duomenų modelis, turi keletą komponentų. Pirmiausia, jis turi rinkinį aukšto lygio taisyklių, kaip kad loginis duomenų modelis turi aukšto lygio atributų rinkinį. Loginis taisyklių modelis taip pat vaizduoja taisyklių tarpusavio priklausomybes. Šios priklausomybės nėra tokios pat kaip grafiškai pavaizduotos priklausomybės tarp atributų duomenų

modelyje. Loginis taisyklių modelis susieja kiekvieną taisyklę su jos valdoma ar išvedama informacija.

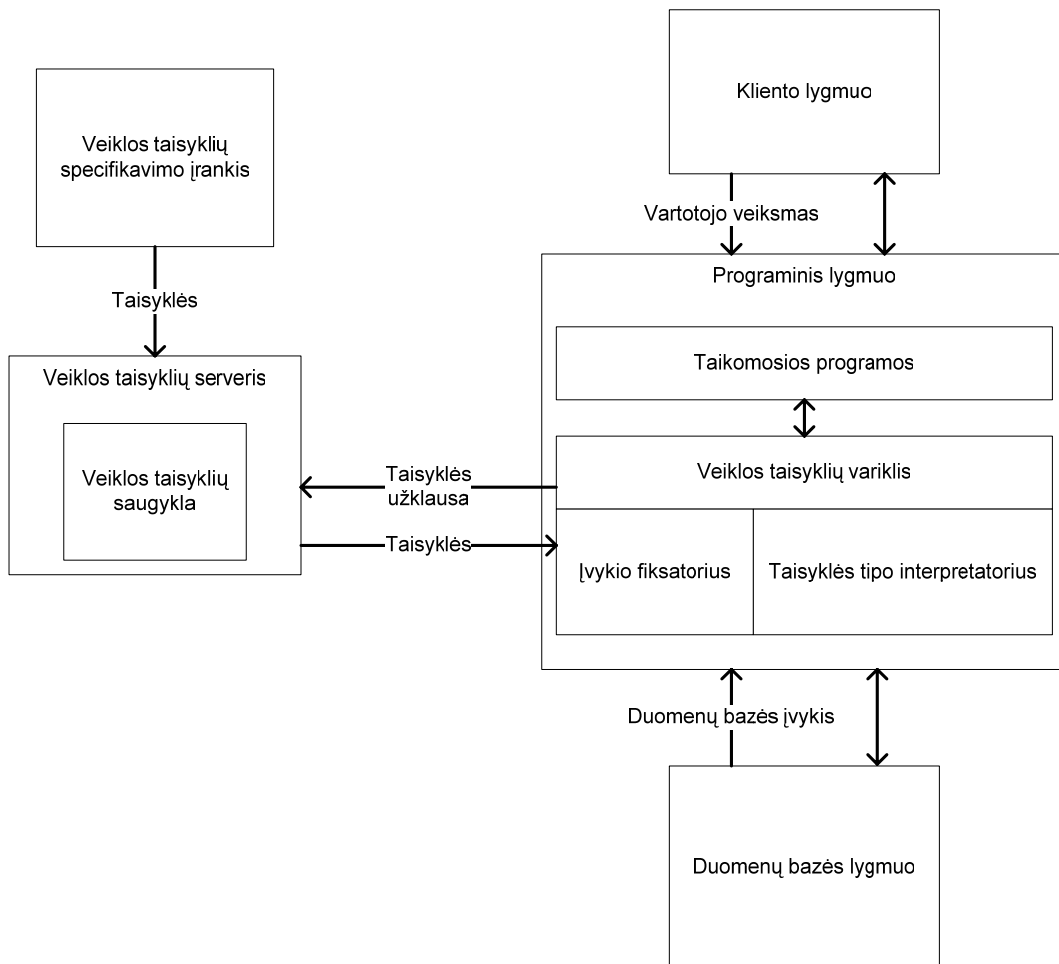
Projektavimo fazėje atliekamas sistemos realizavimo galimybių projektavimas. Projektavimo fazės rezultatas – galutinė sistemos specifikacija, apimanti fizinius, architektūrinius, vartotojo sąsajos, duomenų bazės ir programinius aspektus.

Realizavimo fazėje atliekama visos sistemos realizacija. Šios fazės rezultatas – nauja sistema ir jos priežiūros (palaikymo) planas.

Testavimo fazėje testuojama naujai sukurta sistema bei šalinamos aptiktos klaidos. Klaidų ištaisymui tenka grįžti į ankstesnes projektavimo fazes. Tai vaizduoja 6.1 pav. esančios punktyrinės rodyklės atgaline tvarka. Šios rodyklės parodo, kad po kiekvienos projektavimo fazės galimas grįžimas į ankstesnes sistemos kūrimo fazes, nes klaidos projektavimo metu yra neišvengiamos.

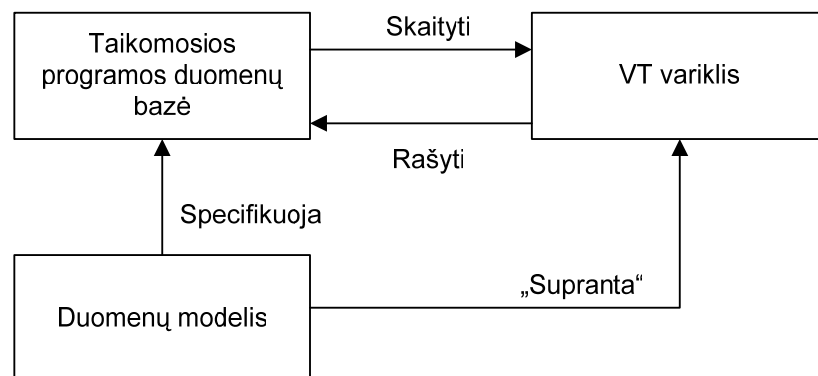
6.2. Veiklos taisyklių sistemos architektūra

Veiklos taisyklių sistemos architektūra pavaizduota 6.2 pav. Veiklos taisyklės kuriamos (ir redaguojamos) naudojant specialiai tam skirtą veiklos taisyklių specifikavimo įrankį. Jo architektūra priklauso nuo naudojamo specifikavimo metodo. Saugomos veiklos taisyklės sužadinamos VT variklio, kuris susideda iš dviejų dalių: įvykio fiksatoriaus ir veiklos taisyklės tipo interpretatoriaus. Veiklos taisyklės yra sužadinamos tam tikro veiksmo (įvykio), kuris gali būti sugeneruotas arba vartotojo, arba duomenų bazės trigerių pagalba. Tuomet veiklos taisyklių variklis generuoja taisyklės užklausa, kuri po to yra reikiamai apdorojama atitinkamų programinių servisų.



6.2 pav. Veiklos taisyklių sistemos architektūra

Reikia pastebėti, kad veiklos taisyklių variklio veikimas yra panašus į tam tikros taikomosios programos veikimą. Tuo būdu, kad jis pasiima informaciją iš programos naudojamos duomenų bazės ir ją atitinkamai apdorojęs paprastai įrašo atgal į duomenų bazę. VT variklis turi turėti prieigą prie duomenų bazės, t.y. iš jos skaityti ir į ją įrašyti. VT variklis turi mokėti „suprasti“ duomenų modelį, kurio pagrindu yra sudaryta naudojama programos duomenų bazė, kad galėtų dirbti su duomenų bazės duomenimis. Toks VT variklio ir taikomosios programos duomenų bazės sąryšis pateiktas 6.3 pav.



6.3 pav. Veiklos taisyklių variklio ir taikomosios programos duomenų bazės sąryšis

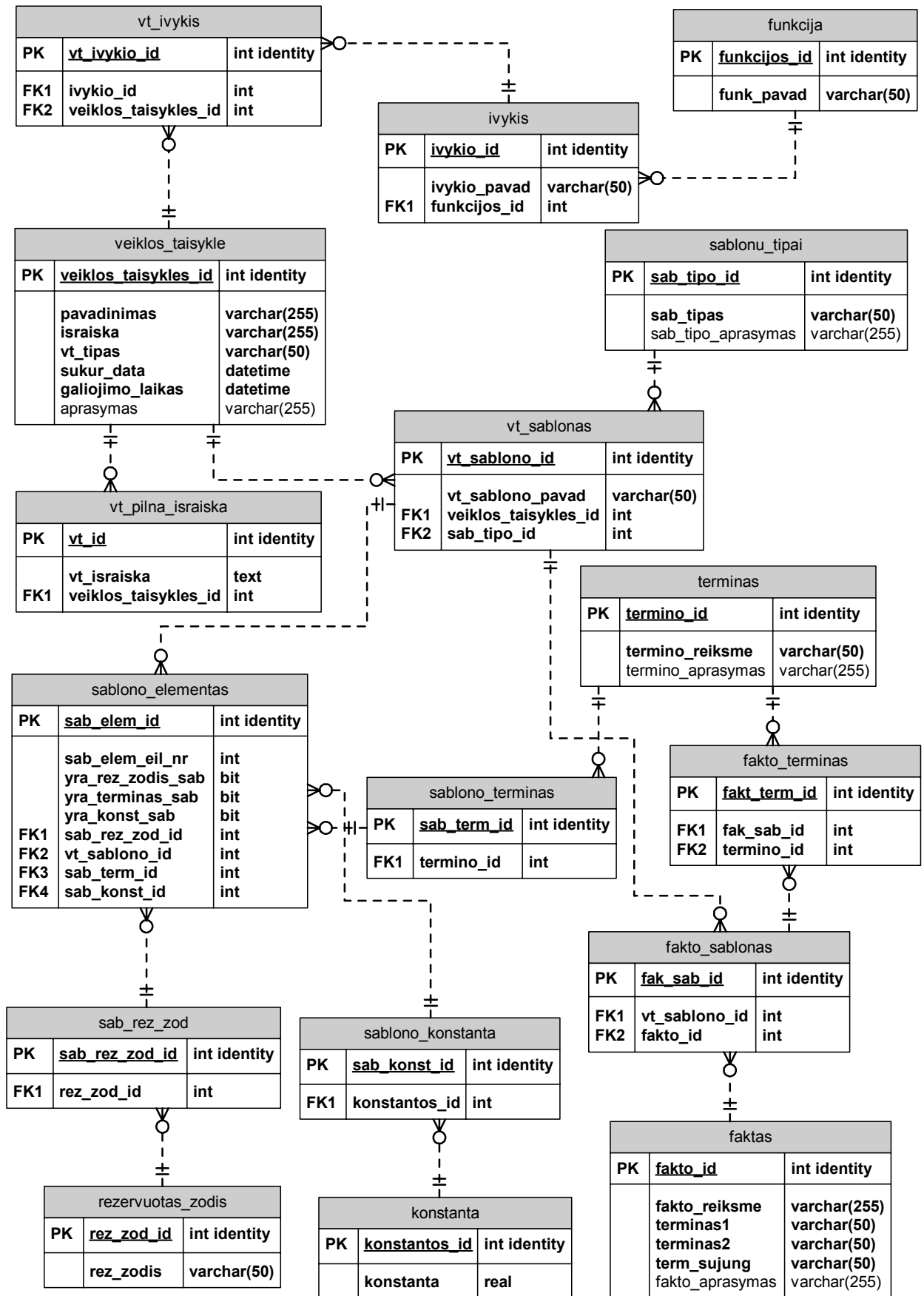
6.3. Veiklos taisyklių saugyklos architektūra

Remiantis aukščiau pateiktomis sistemos kūrimo fazėmis, projektuojama veiklos taisyklių saugykla, kuri leistų kaupti visas pasiūlytos klasifikacijos taisykles. Saugyklos kūrimo metu pereinamos planavimo, reikalavimų nustatymų ir analizės fazės. Šios fazės nėra pilnai detalizuojamos, nes šiame darbe nėra užsibrėžta suprojektuoti visą veiklos taisyklėmis pagrįstą informacinės sistemos modelį. Darbe apsiribojama saugyklos modeliu, kuris tarnautų sistemos analitikui surenkant ir specifikuojant veiklos taisykles. Tuo pačiu suprojektuota saugykla leis įvertinti pasiūlyto veiklos taisyklių klasifikavimo metodo efektyvumą.

Veiklos taisyklių saugykla projektuojama remiantis šiais principais:

- saugykla turi leisti išsaugoti ir atkurti visas, pasiūlytu klasifikavimo metodu, specifikuotas veiklos taisykles.
- veiklos taisyklė turi būti saugoma ir paprasta (t. y. natūralia) kalba, o ne vien tik formaliu pavidalu.
- informacija apie taisyklės interpretavimo (panaudojimo) galimybes turi būti nesaugoma (arba saugoma ši informacija turi būti minimali).

Suprojektuotą veiklos taisyklių saugyklos architektūrą vaizduoja 6.4 pav., o detalus kiekvienos duomenų bazės lentelės aprašas pateiktas 6.1 lentelėje. Pateiktoje saugykloje yra įvertintos jos tolesnio panaudojimo perspektyvos ir tai yra pagrindinis sukurtos saugyklos privalumas. Kiekviena taisyklė yra susiejama su įvykiu (įvykis gali turėti daug taisyklių) ir veiklos funkcija (funkcija gali turėti daug įvykių). Toks taisyklių kaupimo būdas įveda taisyklių grupavimą (kas yra svarbus dalykas veiklos taisyklių sistemose) bei palengvina tolimesnį saugyklos eksploatavimą. Saugykla gali būti eksploatuojama autonomiškai ir užtikrina ją naudosiančių programinių sistemų lankstumą.



6.4 pav. Veiklos taisyklių saugyklos architektūra pasiūlytam veiklos taisyklių klasifikavimo metodui

6.1 lentelė. Veiklos taisyklių saugyklos duomenų bazės lentelės ir jų aprašymai

Lentelė	Laukas	Aprašymas	Tipas
funkcija		Šioje lentelėje saugomi funkcijų aprašai	
	funkcijos_id	Funkcijos identifikacijos numeris	Sveikasis skaičius
	funk_pavad	Funkcijos pavadinimas	Tekstinė eilutė iki 50 simbolių
ivykis		Šioje lentelėje saugomi funkcijų įvykiai	
	ivykio_id	Įvykio identifikacijos numeris	Sveikasis skaičius
	ivykio_pavad	Įvykio pavadinimas	Tekstinė eilutė iki 50 simbolių
vt_ivykis		Pagalbinė lentelė, pašalinanti <i>daug su daug</i> ryšį tarp lentelių <i>veiklos_taisyklė</i> ir <i>ivykis</i> . Ji susieja įvykius ir taisykles	
	vt_ivykio_id	Įvykio veiklos taisyklės identifikacijos numeris	Sveikasis skaičius
	ivykio_id	Įvykio identifikacijos numeris	Sveikasis skaičius
	veiklos_taisyklės_id	Veiklos taisyklės identifikacijos numeris	Sveikasis skaičius
veiklos_taisykle		Šioje lentelėje saugoma pagrindinė veiklos taisyklių informacija	
	veiklos_taisyklės_id	Veiklos taisyklės identifikacijos numeris	Sveikasis skaičius
	pavadinimas	Veiklos taisyklės pavadinimas	Tekstinė eilutė iki 255 simbolių
	israiska	Veiklos taisyklės pilnas tekstas.	Tekstinė eilutė iki 255 simbolių
	vt_tipas	Veiklos taisyklės tipas pagal klasifikaciją	Tekstinė eilutė iki 50 simbolių
	sukurimo_data	Veiklos taisyklės sukūrimo data	Datos formatas
	galiojimo_laikas	Veiklos taisyklės galiojimo laikas, iki kurio ji turi būti aktyvi sistemoje	Datos formatas
	aprasymas	Veiklos taisyklės aprašymas	Tekstinė eilutė iki 255 simbolių
vt_pilna_israiska		Šioje lentelėje saugoma pilna veiklos taisyklės išraiška, t. y. veiklos taisyklė natūralia kalba	Tekstinė eilutė
	vt_id	Veiklos taisyklės natūralia kalba identifikacijos numeris	Sveikasis skaičius
	vt_israiska	Veiklos taisyklės išraiška natūralia kalba	Tekstinė eilutė
	vt_taisyklės_id	Veiklos taisyklės identifikacijos numeris	Sveikasis skaičius
	yra_terminas_vt	Jei tai terminas konkrečiame veiklos taisyklės egzemplioriuje, tai vienetinė reikšmė (priešingu atveju nulinė reikšmė)	Loginis tipas
	veiklos_taisykles_id	Veiklos taisyklės identifikacijos numeris	Sveikasis skaičius
terminas		Šioje lentelėje saugomi visi terminai ir jų aprašymai	

Lentelė	Laukas	Aprašymas	Tipas
	termino_id	Termino identifikacijos numeris	Sveikasis skaičius
	termino_reiksme	Termino reikšmė	Tekstinė eilutė iki 50 simbolių
	termino_aprasymas	Terminą charakterizuojantis aprašymas	Tekstinė eilutė iki 255 simbolių
vt_sablonas		Šioje lentelėje saugomi konkretūs taisyklių šablonai	
	vt_sablono_id	Šablono identifikacijos numeris	Sveikasis skaičius
	vt_sablono_pavad	Veiklos taisyklės šablono pavadinimas	Tekstinė eilutė iki 50 simbolių
	veiklos_taisyklės_id	Veiklos taisyklės identifikacijos numeris	Sveikasis skaičius
	sab_tipo_id	Šablono tipo identifikacijos numeris	Sveikasis skaičius
sablonu_tipai		Šioje lentelėje saugomi šablonų tipai	
	sab_tipo_id	Šablono tipo identifikacijos numeris	Sveikasis skaičius
	sab_tipas	Šablono tipas	Tekstinė eilutė iki 50 simbolių
	sab_tipo_aprasymas	Šablono tipo aprašymas	Tekstinė eilutė iki 255 simbolių
sablonu_elementas		Šioje lentelėje saugoma informacija apie šablone esančių elementų eilės tvarką	
	sab_elem_id	Šablono elemento identifikacijos numeris	Sveikasis skaičius
	sab_elem_eil_nr	Elementų eilės numeris šablone	Sveikasis skaičius
	yra_rez_zodis_sab	Jei tai rezervuotas žodis bet kuriame šablone, tai vienetinė reikšmė (priešingu atveju nulinė reikšmė)	Loginis tipas
	yra_terminas_sab	Jei tai terminas bet kuriame šablone, tai vienetinė reikšmė (priešingu atveju nulinė reikšmė)	Loginis tipas
	yra_konst_sab	Jei tai konstanta bet kuriame šablone, tai vienetinė reikšmė (priešingu atveju nulinė reikšmė)	Loginis tipas
	sab_rez_zod_id	Šablono rezervuoto žodžio identifikacijos numeris	Sveikasis skaičius
	vt_sablono_id	Šablono identifikacijos numeris	Sveikasis skaičius
	sab_term_id	Šablono termino identifikacijos numeris	Sveikasis skaičius
	sab_konst_id	Šablono konstantos identifikacijos numeris	Sveikasis skaičius
rezervuotas_zodis		Šioje lentelėje saugomi rezervuoti žodžiai	
	rez_zod_id	Rezervuoto žodžio identifikacijos numeris	Sveikasis skaičius
	rez_zodis	Rezervuoto žodžio reikšmė	Tekstinė eilutė iki 50 simbolių
sab_rez_zodis		Pagalbinė lentelė, pašalinanti daug su daug ryšį tarp lentelių <i>rezervuotas_zodis</i> ir <i>sablono_elementas</i> . Šioje lentelėje saugomi	

Lentelė	Laukas	Aprašymas	Tipas
		šablonuose panaudoti rezervuoti žodžiai.	
	sab_rez_zod_id	Šablono rezervuoto žodžio identifikacijos numeris	Sveikasis skaičius
	rez_zod_id	Rezervuoto žodžio identifikacijos numeris	Sveikasis skaičius
faktas		Šioje lentelėje saugomi faktai	
	fakto_id	Fakto identifikacijos numeris	Sveikasis skaičius
	fakto_reiksme	Fakto reikšmė	Tekstinė eilutė iki 255 simbolių
	terminas1	Identifikuoja pirmąjį fakto terminą	Tekstinė eilutė iki 50 simbolių
	terminas2	Identifikuoja antrąjį fakto terminą	Tekstinė eilutė iki 50 simbolių
	term_sujung	Nusako koks ryšio tipas tarp dviejų fakto terminų	Tekstinė eilutė iki 50 simbolių
	fakto_aprasymas	Faktą charakterizuojantis aprašymas	Tekstinė eilutė iki 255 simbolių
fakto_sablonas		Pagalbinė lentelė, pašalinanti <i>daug su daug</i> ryšį tarp lentelių <i>vt_sablonas</i> ir <i>faktas</i> . Ji apibūdina tai, kad faktams sudaryti reikalingi šablonai	
	fak_sab_id	Fakto šablono identifikacijos numeris	Sveikasis skaičius
	vt_sablono_id	Šablono identifikacijos numeris	Sveikasis skaičius
	fakto_id	Fakto identifikacijos numeris	Sveikasis skaičius
fakto_terminas		Pagalbinė lentelė, pašalinanti <i>daug su daug</i> ryšį tarp lentelių <i>terminas</i> ir <i>fakto_sablonas</i> . Šioje lentelėje saugomi terminai, įeinantys į konkrečių faktų sudėtį	
	fak_term_id	Fakto termino identifikacijos numeris	Sveikasis skaičius
	termino_id	Termino identifikacijos numeris	Sveikasis skaičius
	fak_sab_id	Fakto šablono identifikacijos numeris	Sveikasis skaičius
konstanta		Šioje lentelėje saugomos visos veiklos taisyklėse naudojamos konstantos (t. y. jų skaitinės reikšmės)	
	konstantos_id	Konstantos identifikacijos numeris	Sveikasis skaičius
	konstanta	Konstantos reikšmė	Realusis skaičius
sablono_konstanta		Pagalbinė lentelė, pašalinanti <i>daug su daug</i> ryšį tarp lentelių <i>konstanta</i> ir <i>sablono_elementas</i> . Šioje lentelėje saugomos šablonuose panaudotos konstantos.	
	sab_konst_id	Šablono konstantos identifikacijos numeris	Sveikasis skaičius
	konstantos_id	Konstantos identifikacijos numeris	Sveikasis skaičius
sablono_terminas		Pagalbinė lentelė, pašalinanti <i>daug su daug</i> ryšį tarp lentelių <i>terminas</i> ir <i>sablono_elementas</i> . Šioje lentelėje saugomi šablonuose panaudoti	

Lentelė	Laukas	Aprašymas	Tipas
		terminai.	
	sab_term_id	Šablono termino identifikacijos numeris	Sveikasis skaičius
	termino_id	Termino identifikacijos numeris	Sveikasis skaičius

6.4. Veiklos taisyklių atvaizdavimas sukurtoje veiklos taisyklių saugykloje

Pavyzdžiais iliustruokime kaip tam tikros veiklos taisyklės atsivaizduoja pasiūlytoje veiklos taisyklių saugykloje. Paimkime tokią veiklos taisyklę (išvesties): **Jei kompiuterio_kaina > 2500 ir pirkimo_kiekis > 5 tuomet nuolaida 5 procentai**. Terminai: *kompiuterio_kaina*, *pirkimo_kiekis*, *procentai* ir *nuolaida*. Rezervuoti žodžiai: *jei*, *>*, *ir*, *tuomet*. Konstantos: 2500 ir 5. Kaip tokia taisyklė atsivaizduoja suprojektuotoje veiklos taisyklių saugykloje parodyta 6.2 lentelėje. Dar vieno veiklos taisyklių tipo (būtinumo apribojimo) atvaizdavimas duomenų bazėje yra pateiktas 3 priede.

6.2 lentelė. Išvesties veiklos taisyklės atvaizdavimas sukurtoje veiklos taisyklių saugykloje

Lentelė <i>funkcija</i>								
funkcijos_id		funkc_pavad						
1		Užsakymo kainos skaičiavimas						
Lentelė <i>ivykis</i>								
ivykio_id		ivykio_pavad			funkcijos_id			
1		Nuolaidos skaičiavimas			1			
Lentelė <i>vt_ivykis</i>								
vt_ivykio_id		ivykio_id		veiklos_taisykles_id				
1		1		1				
Lentelė <i>veiklos_taisykle</i>								
veiklos_taisykles_id	pavadinimas	israiska	vt_tipas	sukurimo_data	galiojimo_laikas	aprasymas		
1	Kompiuterio nuolaidos skaičiavimas	Jei kompiuterio_kaina > 2500 ir pirkimo_kiekis > 5 tuomet nuolaida 5 procentai	išvestis	2005.03.18	2005.09.01			
Lentelė <i>vt_pilna_israiska</i>								
vt_id	vt_israiska				vt_taisykles_id			
1	Jei kompiuterio_kaina didesnė už 2500 ir pirkimo_kiekis daugiau kaip 5 tuomet nuolaida 5 procentai				1			
Lentelė <i>terminas</i>								
termino_id	termino_reiksme		termino_aprasymas					
1	kompiuterio_kaina							
2	pirkimo_kiekis							
3	nuolaida							
4	procentai							
Lentelė <i>vt_sablonas</i>								
vt_sablono_id		vt_sablono_pavad		veiklos_taisykles_id		sab_tipo_id		
1		Užsakymo nuolaidos skaičiavimas		1		1		
Lentelė <i>sablonu_tipai</i>								
sab_tipo_id		sab_tipas		sab_tipo_aprasymas				
1		išvestis						
Lentelė <i>sablono_elementas</i>								
sab_elem_id	sab_elem_eil_nr	yra_rez_zodis_sab	yra_terminas_sab	yra_konst_sab	sab_rez_zod_id	vt_sablono_id	sab_term_id	sab_konst_id
1	1	1	0	0	1	1	0	0
2	2	0	1	0	0	1	1	0
3	3	1	0	0	3	1	0	0
4	4	0	0	1	0	1	0	1
5	5	1	0	0	2	1	0	0
6	6	0	1	0	0	1	2	0
7	7	1	0	0	3	1	0	0
8	8	0	0	1	0	1	0	2
9	9	1	0	0	4	1	0	0
10	10	0	1	0	0	1	3	0
11	11	0	0	1	0	1	0	2
12	12	0	1	0	0	1	4	0

Lentelė <i>sablono terminas</i>	
sab term id	termino id
1	1
2	2
3	3
4	4
Lentelė <i>rezervuotas žodis</i>	
rez zod id	rez zodis
1	jei
2	ir
3	>
4	tuomet
Lentelė <i>sab_rez_zodis</i>	
sab rez zod id	rez zod id
1	1
2	2
3	3
4	4
Lentelė <i>konstanta</i>	
konstantos id	konstanta
1	2500
2	5
Lentelė <i>sablono konstanta</i>	
sab konst id	konstantos id
1	1
2	2

Pažiūrėkime kaip saugykloje atsivaizduotų toks faktas: **Studento numeris yra studento atributas**. Terminai: *studento numeris* ir *studento atributas*. Ryšys tarp terminų yra esybė – atributas (rezervuotas žodis yra).

6.3 lentelė. Fakto atvaizdavimas sukurtoje veiklos taisyklių saugykloje

Lentelė <i>funkcija</i>						
funkcijos id				funk pavad		
1				Studentų registravimas		
Lentelė <i>ivykis</i>						
ivykio id		ivykio pavad		funkcijos id		
1		Atributų kūrimas		1		
Lentelė <i>vt_ivykis</i>						
vt ivykio id		ivykio id		veiklos taisyklės id		
1		1		1		
Lentelė <i>veiklos taisykle</i>						
veiklos_tai_syklės id	pavadinimas	israiska	vt_tipas	sukurimo_data	galiojimo_laikas	aprasymas
1	Studento atributas	Studento numeris yra studento atributas	faktas	2005-03-21	2005-06-30	
Lentelė <i>vt_pilna_israiska</i>						
vt_id	vt_israiska				vt_taisyklės_id	
1	Studento bilieta numeris yra studento atributas				1	
Lentelė <i>terminas</i>						
termino id		termino reiksme		termino aprasymas		
1		studento numeris				
2		studento atributas				
Lentelė <i>vt_sablonas</i>						
vt sablono id		vt sablono pavad		veiklos taisyklės id		sab tipo id
1		Studento forma		1		1
Lentelė <i>sablonu tipai</i>						
sab_tipo_id		sab_tipas		sab_tipo_aprasymas		
1		faktas				
Lentelė <i>faktas</i>						
fakto_id	fakto_reiksme	terminas1	terminas2	term_sujung	fakto_aprasymas	
1	studento numeris yra studento atributas	studento numeris	studento atributas	yra		
Lentelė <i>fakto sablonas</i>						
fak sab id		vt sablono id		fakto_id		
1		1		1		

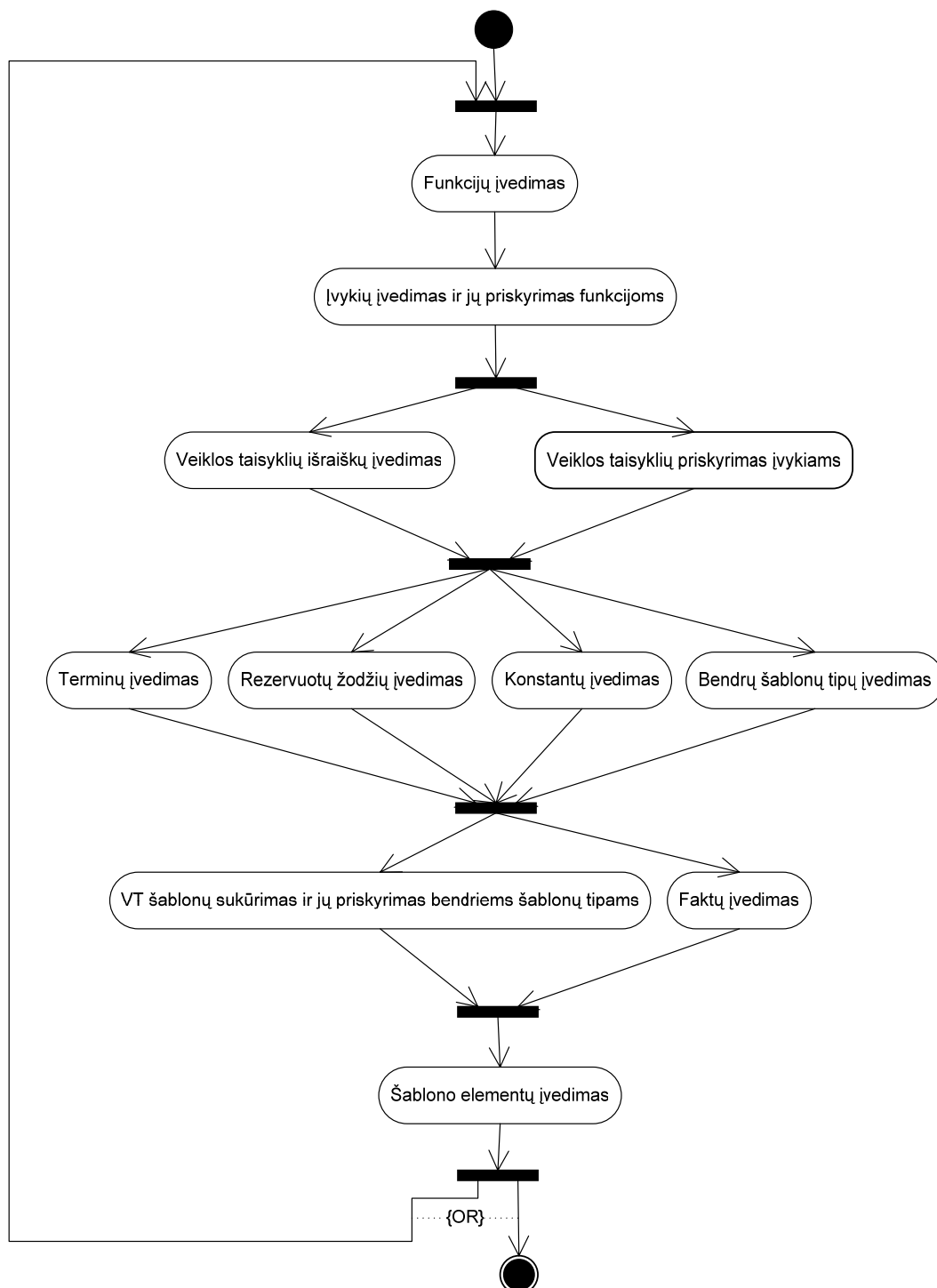
Lentelė <i>fakto terminas</i>		
fak term id	termino id	fak sab id
1	1	1
2	2	1

6.5. Veiklos taisyklių įvedimas į saugyklą

Suprojektuotą veiklos taisyklių saugyklos architektūrą vaizduoja 6.4 pav. Veiklos taisyklių įvedimas pagal pasiūlytą metodą turi prasidėti nuo funkcijų apibrėžimo, kurioms bus formuluojamos veiklos taisyklės. Jos suvedamos į lentelę *funkcija*. Funkcija šiame darbe laikomas tam tikras sistemos panaudojimo atvejis. Todėl pirmiausia patartina apibrėžti visus sistemos panaudojimo atvejus. Kiekvienai funkcijai toliau išskiriami įvykiai (lentelės *ivykis* ir *vt_ivykis*). Įvykis – tai konkretus sistemos įvykis, kuriam yra formuluojamos veiklos taisyklės. Lentelėje *veiklos_taisykle* saugoma pagrindinė veiklos taisyklių informacija, o jos pilna išraiška saugoma lentelėje *vt_pilna_israiska*. Lentelėje *terminas* saugomi visi terminai ir jų aprašymai. Lentelėje *vt_sablonas* saugoma informacija apie konkrečios taisyklės šablonus, o visi bendrieji veiklos taisyklių šablonų tipai saugomi lentelėje *sablonu_tipai*. Šablonai susideda iš apibrėžtų elementų (terminai, rezervuoti žodžiai ir konstantos). Tam realizuoti lentelėje *sablono_elementas* saugoma informacija apie šablone esančius elementus ir jų eilės tvarką. Rezervuoti žodžiai saugomi lentelėje *rezervuotas_zodis*. Lentelė *sab_rez_zod* yra pagalbinė, pašalinanti daug su daug ryšį tarp lentelių *rezervuotas_zodis* ir *sablono_elementas*. Joje saugomi šablonuose panaudoti rezervuoti žodžiai. Veiklos taisyklių konstantos saugomos lentelėje *konstanta*. Lentelė *sablono_konstanta* yra pagalbinė, pašalinanti daug su daug ryšį tarp lentelių *konstanta* ir *sablono_elementas*. Joje saugomos šablonuose panaudotos konstantos. Terminai ir jų aprašymai saugomi lentelėje *terminas*. Terminai, įeinantys į šablonus, saugomi lentelėje *sablono_terminas*. Faktai saugomi lentelėje *faktas*. Lentelė *fakto_sablonas* yra pagalbinė, pašalinanti daug su daug ryšį tarp lentelių *vt_sablonas* ir *faktas*. Ji apibūdina tai, kad faktams sudaryti reikalingi šablonai. Lentelėje *fakto_terminas* saugomi terminai, įeinantys į konkrečių faktų sudėtį.

Rekomenduojamą veiklos taisyklių, įvykių ir funkcijų įvedimo į saugyklą tvarką vaizduoja 6.5 pav. pateikta veiklos diagrama. Pirmiausiai yra įvedamos veiklos funkcijos. Po to įvedami įvykiai ir jie priskiriami funkcijoms. Toliau įvedama veiklos taisyklės išraiška ir galimas įvestos taisyklės susiejimas su įvykiu. Po to taisyklė yra struktūrizuojama ir suvedami joje esantys terminai, rezervuoti žodžiai ir konstantos. Jeigu įvedamas faktas, tai įvedami fakte naudojami terminai. Tame pačiame etape taip pat galimas bendrų taisyklių šablonų tipų įvedimas. Toliau yra sukuriamas konkrečios taisyklės šablonas ir jis susiejamas su taisykle bei bendru šablono tipu. Tame pačiame etape galimas ir faktų lentelės pildymas. Galiausiai suvedami visi šablono elementai atitinkama eilės tvarka.

Įvedimo tvarka nebūtinai turi būti tokia griežta, tačiau tokia tvarka rekomenduojama norint tvarkingai suvesti veiklos taisykles. Pavyzdžiui, jeigu iš karto pildomas šablonas, o dar nesuvesti jam reikalingi terminai ar rezervuoti žodžiai, tai iš karto kyla problemos specifikuojant šablono elementą. Todėl rekomenduojama laikytis aprašytos saugyklos pildymo tvarkos.



6.5 pav. Veiklos taisyklių saugyklos pildymo eiga

6.6. Veiklos taisyklių saugyklos formalūs reikalavimai

Suformuluojame pagrindinius sukurtos veiklos taisyklių saugyklos reikalavimus ir juos užrašome ir paprastu tekstiniu formatu, ir formalizuotai. Žemiau pateikti reikalavimai (metataisyklės) turi būti tenkinami visais atvejais, norint nepažeisti bendrųjų veiklos taisyklių saugyklos reikalavimų. Pagrindiniai reikalavimai yra šie:

1. Kiekvienas įvykis turi būti susietas su veiklos funkcija.

$$\forall iv(Ivykis(iv) \rightarrow \exists f(Veiklos_funkcija(f) \wedge Susieja(f, iv)))$$

2. Kiekvienas taisyklė turi būti susieta su įvykiu.

$$\forall t(Veiklos_taisyklė(t) \rightarrow \exists iv(Ivykis(iv) \wedge Susieja(iv, t)))$$

3. Kiekviena veiklos taisyklė per įvykį turi būti susieta su veiklos funkcija.

$$\forall t(Veiklos_taisyklė(t) \rightarrow \exists iv(Ivykis(iv) \wedge \exists f(Veiklos_funkcija(f) \wedge Nusako_sąryšį(f, iv, t)))$$

4. Kiekvienos veiklos taisyklės vykdymo eigą nusako veiklos funkcija.

$$\forall t(Veiklos_taisyklė(t) \rightarrow \exists f(Veiklos_funkcija(f) \wedge \wedge Nusako_vykdymo_eigą(f, t)))$$

5. Kiekviena veiklos taisyklė turi turėti šabloną.

$$\forall t(Veiklos_taisyklė(t) \rightarrow \exists s(\check{S}ablonas(s) \wedge Veiklos_taisyklės_šablonas(t, s)))$$

6. Veiklos taisyklės šablono elementas gali būti arba terminas, arba rezervuotas žodis, arba konstanta.

$$\forall s \forall se(\check{S}ablonas(s) \wedge \check{S}ablono_elementai(se) \wedge \check{S}ablonas_susideda_iš(s, se) \rightarrow \rightarrow (Terminas(se) \vee Rezervuotas_žodis(se) \vee Konstanta(se)))$$

7. Faktas nusako ryšį tarp dviejų terminų.

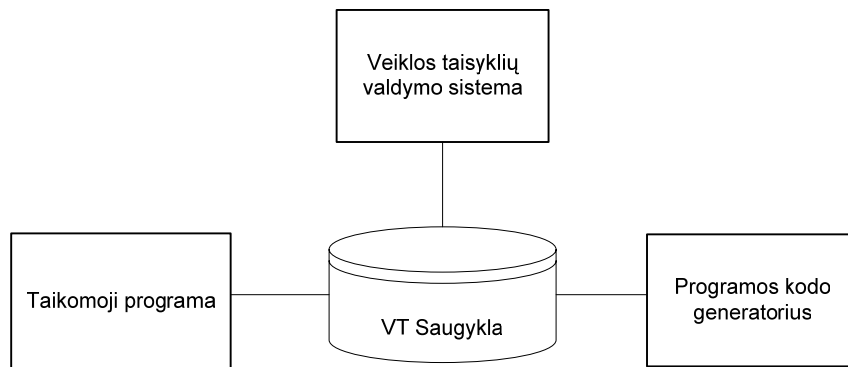
$$\forall x \forall y(Terminas(x) \wedge Terminas(y) \rightarrow \exists r(Ryšys(r) \wedge Nusako_ryšį(r, x, y)))$$

6.7. Tolimesnio saugyklos vystymo ir panaudojimo perspektyvos

Suprojektuota saugykla yra įrankis informacinės sistemos analitikui surenkant ir specifikuojant veiklos taisykles pasiūlytu klasifikavimo metodu. Žiūrint į sistemos kūrimo procesą (6.1 pav.) saugykla geriausiai tarnauja planavimo, reikalavimų nustatymų ir analizės fazėse. Kuomet reikia surinkti ir specifikuoti visas projektuojamos sistemos veiklos taisykles bei yra projektuojamas loginis taisyklių modelis. Veiklos taisyklių sistemoje (6.2 pav.) VT duomenų bazė užima svarbią vietą. Nuo jos struktūros priklauso veiklos taisyklių variklio veikimas ir pačio variklio projektavimas. Dabartinei saugyklos struktūrai veiklos taisyklių variklis gali būti suprojektuotas, bet ar jis būtų efektyvus paaiškėtų tik atlikus atitinkamus eksperimentinius tyrimus. Dabartinė saugykla yra skirta taisyklių struktūrizavimui ir jų kaupimui, o jų panaudojimas (iškvietimas, sužadinimas) konkrečioje sistemoje turėtų būti papildomai apsvarstytas. Turi būti

griežtinami taisyklių įvedimo reikalavimai bei suprojektuotas taisyklių sužadinimo procesas. Taisyklės yra sužadinamos prie tam tikrų įvykių. Dabartinėje saugyklos architektūroje įvykiai figūruoja, todėl tolimesnėje sistemos projektavimo eigoje reiktų juos susieti su veiklos taisyklių varikliu. Veiklos taisyklių variklis realiu laiku fiksuodamas tam tikrą įvykį, interpretuotą reikiama taisyklę (susijusią su įvykiu) ir ją iškviestą iš saugyklos. Po to taisyklė reikiamai apdorojama atitinkamų taikomųjų programų.

Konkrečias sukurtos veiklos taisyklių saugyklos panaudojimo sritis (galimybes) vaizduoja 6.6 pav. Esama saugykla gali būti integruojama į bendrą veiklos taisyklių valdymo sistemą. Ją gali naudoti taikomosios programos. Veiklos taisyklių saugyklos ir taikomųjų programų ryšys palaikomas per specialius programų kode įdėtus kreipinius. Taip pat saugyklą gali naudoti programų kodo generatoriai. Kodo generatoriai paimama veiklos taisyklės iš saugyklos ir jas yra integruoja į programos kodą, sugeneruotą tam tikrų projektinių modelių pagrindu.

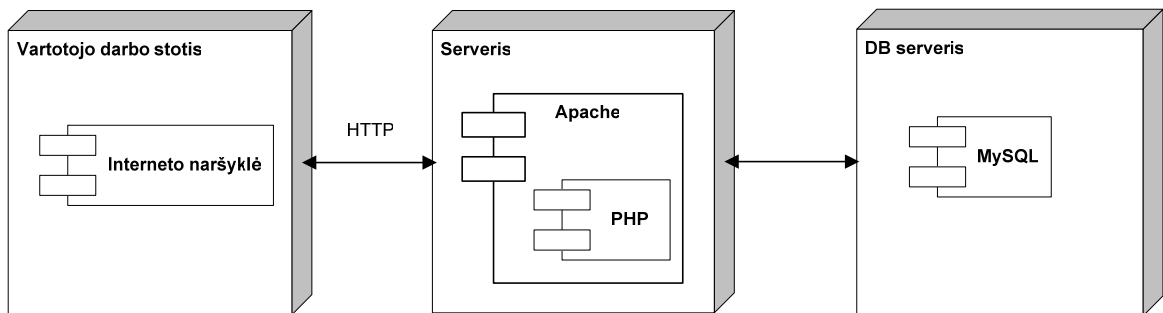


6.6 pav. Veiklos taisyklių saugyklos panaudojimo sritis

7. VEIKLOS TAISYKLIŲ SAUGYKLOS PROTOTIPO REALIZACIJOS APRAŠYMAS

7.1. Sistemos architektūra

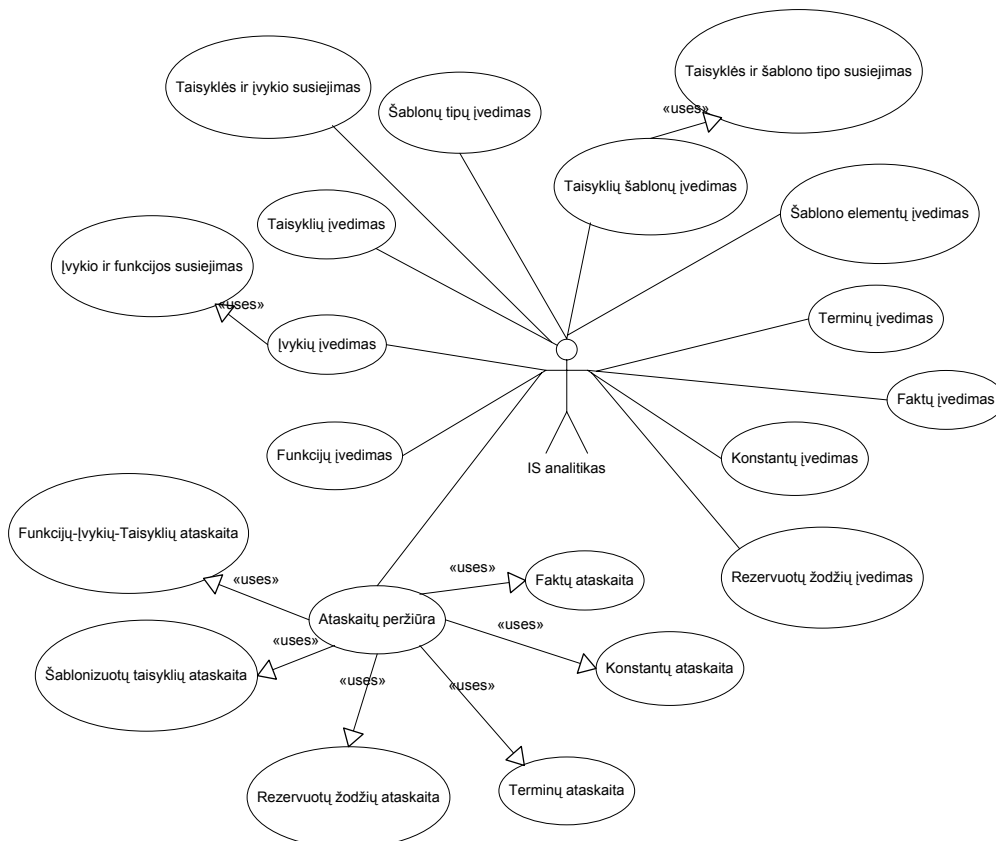
Naudojama tradicinė trijų lygių kliento – serverio architektūra (7.1 pav.). Vartotojo sąsaja – interneto naršyklė, programinės funkcijos – serveryje, duomenys – duomenų bazėje. Naudojama PHP programavimo kalba. Jos veikimui reikalingas Apache serveris. Duomenų bazė – MySQL. Tokios sistemos realizavimo priemonės pasirinktos todėl, kad yra nemokamos.



7.1 pav. Sistemos architektūra

7.2. Sistemos panaudojimo atvejų modelis

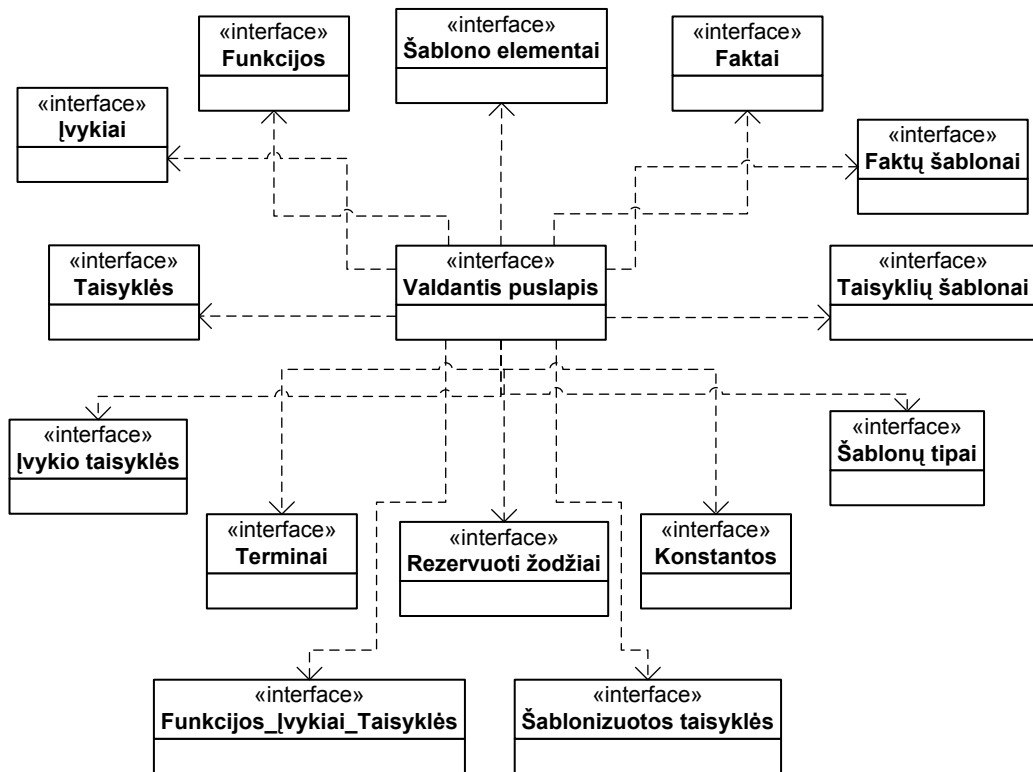
Suprojektuotos sistemos panaudojimo atvejų modelis pateiktas 7.2 pav. Diagramoje matome kokius veiksmus sistemoje gali atlikti sistemos vartotojas (informacinės sistemos analitikas).



7.2 pav. Sistemos panaudojimo atvejų modelis

7.3. Vartotojo sąsajos modelis

Vartotojo sąsajos modelis pavaizduotas 7.3 pav. Visi vartotojo veiksmai atliekami per interneto naršyklę, todėl visa sąsaja realizuota internetiniais puslapiais. Kiekvienai duomenų bazės lentelei (išskyrus veiklos taisyklę ir jos pilną išraišką) pildyti skiriamas atskiras puslapis. Taip išvengta sudėtingo sistemos valdymo bei kūrimo. Vartotojo veiksams kontroliuoti visoje sistemoje yra įdiegta informacinių laukų pildymo kontrolė, kuri informuoja vartotoją apie neužpildytus būtinus informacijos įvedimo laukus. Visur, kur įmanoma, apribotas galimų įvesti informacijos variantų kiekis. Patogiam informacijos įvedimui ir jos peržiūrėjimui realizuota filtravimo funkcija bei atskirai išvedamos pagrindinių duomenų ataskaitos. Įvedimo formose vartotojui leidžiama įvesti tik jam prasmingą informaciją. Identifikaciniai ir kiti galimi automatiškai užpildyti laukai, pildomi automatiškai.



7.3 pav. Vartotojo sąsajos modelis

7.4. Sistemos prototipo vartotojo sąsajos meniu aprašymas

Pradinis realizuotos sistemos puslapis pateiktas 7.4 pav. Meniu dirbant su sistema nesikeičia, t. y. vartotojas bet kuriuo metu gali pasirinkti tam tikrą meniu punktą. Meniu punktai atitinka duomenų bazės lentelių pildymo pasirinkimą. Detalus kiekvieno meniu punkto išskviečiamo lango ir funkcionalumo aprašas pateiktas 2 priede.



7.4 pav. Pradinis sistemos prototipo puslapis

7.5. Testinis darbo su sistema pavyzdys

Paimkime tokią taisyklę (išvesties): **Jei kompiuterio_kaina > 4500 ir pirkimo_kiekis > 5 tuomet nuolaida 5 procentai.** Terminai: *kompiuterio_kaina*, *pirkimo_kiekis*, *nuolaida* ir *procentai*. Rezervuoti žodžiai: *jei*, *>*, *ir*, *tuomet*. Konstantos: 2500 ir 5. Funkcija: *Užsakymo kainos skaičiavimas*. Įvykio pavadinimas: *Nuolaidos skaičiavimas*. Taisyklės pavadinimas: *Kompiuterio kainos nuolaidos skaičiavimas*.

Taisyklės įvedimo metu laikomės 6.5 pav. pateiktos įvedimo tvarkos. Visų pirma įvedame funkciją (menu pasirinkimas *Funkcija*):

7.5 pav. Funkcijos įvedimo forma

Po to įvedame įvykį ir jį priskiriame funkcijai (menu pasirinkimas *Įvykiai*):

LENTELĖ "IVYKIS"

Šioje lentelėje saugomi funkcijų įvykiai

Įvykio pavadinimas:

Funkcijos pavadinimas:

7.6 pav. Įvykio įvedimo forma

Po to suvedama taisyklė ir ją apibūdinanti informacija (menu pasirinkimas *Taisyklės*):

LENTELĖ "VEIKLOS_TAISYKLE"

Veiklos taisyklės pavadinimas:

Veiklos taisyklės pilnas tekstas:

Veiklos taisyklės tipas pagal klasifikaciją:

Veiklos taisyklės galiojimo laikas, iki kurio ji turi būti aktyvi sistemoje:
(pvz. 2005-07-01 12:20:30)

Veiklos taisyklės aprašymas:

7.7 pav. Veiklos taisyklės įvedimo forma

Taisyklė priskiriama įvykiui (menu pasirinkimas *Įvykio taisyklės*):

LENTELĖ "VT_IVYKIS"

Šioje lentelėje susiejami įvykiai ir taisyklės

Įvykio pavadinimas:

Veiklos taisyklės pavadinimas:

7.8 pav. Įvykio ir veiklos taisyklės susiejimo forma

Sukuriamas šablono tipas išvesties taisyklėms (jei jis dar nesukurtas) (menu pasirinkimas *Šablonų tipai*). Įvedami terminai (menu pasirinkimas *Terminai*), rezervuoti žodžiai (menu pasirinkimas *Rezervuoti žodžiai*) ir konstantos (menu pasirinkimas *Konstantos*). Veiklos taisyklei sukuriama šablonas ir priskiriamas bendram išvesties šablono tipui (menu pasirinkimas *Taisyklių šablonai*):

LENTELĖ "VT_SABLONAS"

Šioje lentelėje saugomi konkretūs taisyklių šablonai

Veiklos taisyklės šablono pavadinimas:

Veiklos taisyklės pavadinimas:

Šablono tipas:

7.9 pav. Taisyklės šablono sukūrimo forma

Galiausiai įvedami visi šablono elementai eilės tvarka (menu pasirinkimas *Šablono elementai*):

LENTELĖ "SABLONO_ELEMENTAS"

Šioje lentelėje saugoma informacija apie šablone esančius elementus
ir jų eilės tvarką šablone

Veiklos taisyklės šablono pavadinimas:

Elementų eilės numeris šablone:

Šablono elemento tipas:

Šablono rezervuotas žodis:

Šablono terminas:

Šablono konstanta:

7.10 pav. Taisyklės šablono elementų įvedimo forma

Įvestą informaciją geriausiai peržiūrėti ir patikrinti ataskaitose. Taip pat kiekvienos įvedimo formos apačioje (visoje sistemoje) yra išvedami visi atitinkamos duomenų bazės lentelės įrašai. Šablonizuota (struktūrizuota) ir atkurta taisyklė pateikiama ataskaitoje *Šablonizuotos taisyklės*:

ŠABLONŲ TIPAI, TAISYKLIŲ ŠABLONAI IR JŲ ELEMENTAI			
Šablono tipas	VT šablono pavadinimas	Elemento reikšmė	Elemento tipas
[2]Būtinumo apribojimas	[1]Draudimas	Automobilis	terminas
		Turi Būti	rezervuotas zodis
		Apdraustas	terminas
Automobilis Turi Būti Apdraustas			
[2]Būtinumo apribojimas	[17]Užsakymo užsakovas	Užsakymas	terminas
		Turi turėti	rezervuotas zodis
		Užsakovas	terminas
Užsakymas Turi turėti Užsakovas			
[4]Išvestis	[3]Užsakymo nuolaidos skaičiavimas	Jei	rezervuotas zodis
		Kompiuterio_kaina	terminas
		>	rezervuotas zodis
		4500	konstanta
		Ir	rezervuotas zodis
		Pirkimo_kiekis	terminas
		>	rezervuotas zodis
		5	konstanta
		Tuomet	rezervuotas zodis
		Nuolaida	terminas
		5	konstanta
		Procentai	terminas
		Jei Kompiuterio_kaina > 4500 Ir Pirkimo_kiekis > 5 Tuomet Nuolaida 5 Procentai	
[6]Galiojimas	[2]Užsakymo galiojimas	Užsakymas	terminas
		Galioja	rezervuotas zodis
		5	konstanta
		Dienas	terminas
Užsakymas Galioja 5 Dienas			

7.11 pav. Šablonų tipų, taisyklių šablonų ir jų elementų ataskaita

Funkcijos, įvykio ir taisyklės hierarchija pateikiama ataskaitoje *Funkcijos-Įvykiai-Taisyklės*:

FUNKCIJOS, ĮVYKIAI IR TAISYKLĖS		
Funkcijos pavadinimas	Įvykio pavadinimas	Taisyklės pavadinimas
[1]Studentų registravimas	[2]Atributų kūrimas	[4]Studento pažymėjimas
		[5]Studento vardas
		[6]Studento gyvenamoji vieta
[3]Kliento aptarnavimas	[1]Bendros sumos skaičiavimas	[1]Automobilio draudimas
		[2]Užsakymo galiojimas
		[0](Įvykis neturi taisyklės)
[4]Užsakymo priėmimas	[5]Užsakovo priskvrimas	[7]Užsakovo apribojimas
[10]Užsakymo kainos skaičiavimas	[4]Nuolaidos skaičiavimas	[3]Kompiuterio kainos nuolaidos skaičiavimas

7.12 pav. Funkcijų, įvykių ir taisyklių ataskaita

Terminus, rezervuotus žodžius ir konstantas galima peržiūrėti atitinkamose ataskaitose.

8. PASIŪLYTO VEIKLOS TAISYKLIŲ KLASIFIKAVIMO METODO ĮVERTINIMAS KITŲ METODŲ KONTEKSTE

Išanalizavimus nemažai veiklos taisyklių klasifikavimo metodų, bei pasiūlius savo klasifikavimo metodą, pateikiamas gautų rezultatų apibendrinimas, t. y. pasiūlytas metodas įvertinamas kitų metodų kontekste. Išskiriami kriterijai, kurių pagalba įvertintos analizuotų ir sudaryto modelio savybės. Įvertinimo skalę parenkame nuo 1 iki 5. Jei 1, tai metodas, galima teigti, neatitinka kriterijaus arba jis visai nevertinamas. Jei 5 – kriterijus pilnai išpildomas. Galimos ir tarpinės reikšmės priklausomai nuo kriterijaus išpildymo laipsnio. Įvertinimo rezultatai pateikti 8.1 lentelėje.

8.1 lentelė. Veiklos taisyklių klasifikavimo metodų įvertinimas

Metodas Kriterijus	GUIDE projektas	Roso metodas	MDC modelis	Common- KADS metodo- logijos atmaina	RAP klasifi- kacija	Halle klasifi- kacija	SEEC korporacijos metodas	Pasiūlytas klasifika- vimo metodas
Klasifikacijos užbaigtumas	5	5	2	3	2	3	3	3
Modeliavimo metodika	3	5	1	2	2	2	1	2
Saugojimo modelis	1	1	4	1	1	1	1	4
Teorinio panaudojimo tinkamumas	5	5	3	4	3	4	3	4
Praktinio panaudojimo tinkamumas	3	2	4	2	2	4	4	4

Kalbant apie klasifikacijos užbaigtumą, galima teigti, kad beveik visi analizuoti metodai (išskyrus Roso ir GUIDE metodus) pasižymi susiaurinta panaudojimo sritimi ir gali būti praplėsti tam tikrais taisyklių tipais. Veiklos taisyklės yra plačiai taikomos ir norint pilnai apimti visas taikymo sritis klasifikatorius turi būti ypač detalus. Iš to išplaukia jo praktinio panaudojimo problemos. Roso metodas, galima teigti, yra pilnai užbaigtas. Jame išskiriama net 90 taisyklių tipų, bet nėra pilnai atskleistas jo panaudojimo tikslingumas. Taisyklės priskyrimą kuriam nors tipui gali atlikti tik detalai šį metodą išanalizavęs IS analitikas. Detalia klasifikacija pasižymi ir Halle bei CommonKADS metodologijos atmainos klasifikatoriai. Pasiūlytas klasifikavimo metodas apima pagrindinius veiklos taisyklių panaudojimo atvejus, todėl galima teigti, kad klasifikacijos užbaigtumo kriterijus yra išpildomas.

Geriausią modeliavimo metodiką pateikia Roso metodas. Po to seka GUIDE projektas, o visi kiti metodai nepasižymi metodų modeliavimo metodikos pilnumu. Juose apsiribojama konkrečiais

klausimais, kurių atsakymai padeda išskirti VT. Pasiūlyto metodo pagrindu siūloma daugiau į sistemos realizaciją orientuota metodika, susiejant veiklos funkcijas, įvykius ir taisykles.

Saugojimo modelis, išskyrus MDC modelį, yra silpniausia visų klasifikavimo metodų vieta. Suprojektuota veiklos taisyklių saugykla leidžia kaupti visas pasiūlyto klasifikavimo metodo taisykles. Bet vien išsaugojimo neužtenka. Visada reikia turėti omenyje kaip saugykla bus naudojama, t. y. kaip taisyklės bus sužadinamos ir panaudojamos taikomiose programose. Sukurta saugyklos architektūra turi tokio panaudojimo galimybes, nes veiklos taisyklė yra susieta su įvykiais ir funkcijomis. Saugyklos panaudojimas priklauso tik nuo veiklos taisyklių variklio architektūros.

Paskutiniai du išskirti įvertinimo kriterijai yra metodų teorinio ir praktinio pritaikymo tinkamumas. Vieni metodai geriau tinkami teoriniam veiklos taisyklių studijavimui, o kiti – realizuoti praktiškai informacinėse sistemose. Teorinei studijai tinkami beveik visi aptarti metodai. Galima nebent labiau išskirti GUIDE, Roso, CommonKADS ir Halle metodus. Pasiūlytas metodas taip pat naudingas teoriniam studijavimui. Praktinis panaudojimo tinkamumas susijęs su metode išskiriamų taisyklių tipų realizacija konkrečioje sistemoje. Čia pasireiškia jau anksčiau minėtas metodo detalumas ir taisyklių pritaikymo atvejai. Geriausiai šį kriterijų išpildo GUIDE, Halle, SEEC bei MDC klasifikatoriai. Pasiūlytas metodas taip pat tinkamas praktiniam panaudojimui, nes apima pagrindines veiklos taisyklių taikymo sritis. Šio kriterijus išpildymą taip pat parodo sukurta veiklos taisyklių saugykla, kuri leidžia kaupti pasiūlyto klasifikavimo metodo taisykles.

IŠVADOS

- Išanalizuota šiuolaikinė veiklos taisyklės samprata ir jos pritaikymo galimybės. Taisyklių pagalba galima formaliai aprašyti veiklos operacijų semantiką, valdyti veiklos procesus. Šiuolaikinėse sistemose reikalingas taisyklių palaikymo mechanizmas, kuris leistų saugoti taisykles kaip atskirus informacinius vienetus bei juos modifikuoti. Veiklos taisyklių pritaikymo sritis vis dar yra pakankamai nauja, todėl naujų sprendimų paieška ir esančių tobulinimas yra aktualūs. Veiklos taisyklių principais veikiančios informacinės sistemos ir šiuos principus taikančios informacinių sistemų projektavimo priemonės nėra susilaukusios tokio pripažinimo kaip tradiciniai sistemų projektavimo sprendimai.
- Ištirtinėti geriausiai esamas veiklos taisyklių klasifikacijas charakterizuojantys metodai. GUIDE projektas ir Roso modelis yra naudingi kaip teorinė VT koncepcija, o MDC ir B. V. Halle modeliai yra geriau pritaikyti praktiniam naudojimui. CommonKADS metodologijos pagrindu sudarytas metodas geriausiai parodo taisyklių išskyrimo sritis veiklos procesuose. RAP grupės pasiūlytas metodas, praplečia GUIDE metodo klasifikaciją, įvedant struktūrines ir elgsenos dedamąsias. Bendras metodų įvertinimas pateiktas 8 skyriuje.
- Apžvelgti VT klasifikavimo metodai įrodo, kad nėra vieningos nuomonės VT identifikavimo ir klasifikavimo atžvilgiu. Kiekviena įmonė, organizacija ar pavieniai projektuotojai pagal poreikius kuria savo klasifikavimo metodus (dažnai remdamiesi visuotiniai pripažintais klasifikavimo metodais), kurie geriausiai atspindi jų veiklos sritį. Nepaskutinis dalykas yra ir jų realizacija, kurią projektuotojai laiko pirmoje vietoje, nors pagal visus VT standartus klasifikuojant VT reiktų atsiriboti nuo realizacijos detalių.
- Atlikus lyginamąją veiklos taisyklių klasifikavimo metodų analizę, nustatyta, kad kiekvienas iš jų gerai apibrėžia tik tam tikrą taisyklių struktūrizavimo proceso aspektą, nedetalizuoja viso veiklos taisyklių modeliavimo proceso.
- Skirtingos modeliavimo kalbos ir jose naudojamos diagramos nevienodai tinkamos skirtingų tipų taisyklėms modeliuoti. Ne visų tipų taisykles galima modeliuoti naudojant vieną modeliavimo kalbą ar joje naudojamą tam tikrą diagramą. Veiklos taisyklių modeliavimas naudojant UML ir OCL, bei jų realizacija duomenų bazių technologijomis yra vienas populiariausių būdų, bet išskyla problemų. OCL kalba neturi grafinės notacijos ir ją sunku suprasti veiklos atstovams, bei siūlomi būdai nenagrinėja taisyklių, kurios specializuoja verslo algoritmus, procedūras ir jų automatinė realizacija negalima. Taisyklių vaizdavimas EER diagramomis yra taip pat paplitęs, bet jų keitimas reliacinių duomenų bazių aprašais yra gana sudėtingas ir komplikotas.

- Veiklos taisyklių valdymo sistemų funkcionalumas iš esmės yra gana panašus. Veiklos taisyklių apsisikeitimas tarp skirtingų sistemų nėra realizuotas arba realizuotas prastai.
- Atlikus tyrimą, pasiūlytas veiklos taisyklių klasifikavimo metodas, kuris, lyginant su kitais metodais, pasižymi praktinio panaudojimo tinkamumu, saugojimo metodika ir apima daugumą veiklos taisyklių panaudojimo atvejų. Kiekvienam taisyklės tipui pateikti šablonai, leidžiantys fiksuoti taisykles griežtai struktūrizuota forma.
- Kiekvienas VT klasifikavimo metodas turi savų privalumų ir trūkumų. Didžiausias visų metodų trūkumas saugojimo modelio neapibrėžtumas bei modeliavimo metodikos neišbaigtumas. Siūlomas klasifikavimo metodas turi sudarytą saugyklos modelį bei į sistemos realizaciją orientuotą modeliavimo metodiką.
- Sudaryta VT saugykla, kuri leidžia kaupti pasiūlyto klasifikavimo metodo veiklos taisykles. Saugyklos privalumas yra tas, kad taisyklės saugomos susietos su įvykiais ir funkcijomis. Toks saugojimo ir specifikavimo būdas įveda VT grupavimą bei palengvina jų naudojimą taikomosiose programose. Saugyklai naudojama MySQL duomenų bazė.
- Sukurtos veiklos taisyklių saugyklos vartotojo sąsaja leidžia įvesti į duomenų bazę ir peržiūrėti duomenų bazėje specifiкуotas veiklos taisykles. Darbas su sistema vyksta per interneto naršyklę. Saugykloje esančios informacijos detaliam nagrinėjimui formuojamos apibendrintų duomenų ataskaitos, kurios yra daugiau skirtos informacinės sistemos analitikui, kuris surenka, klasifikuoja ir struktūrizuoja VT. Vartotojo sąsaja realizuota naudojant PHP programavimo kalbą bei jos veikimui reikalingą Apache serverį.
- VT specifikavimo įrankio prototipas, palyginus su kitais tokio tipo įrankiais, pasižymi taisyklių grupavimu pagal įvykius ir funkcijas, taisyklių struktūrizavimo, šablonų tipų praplėtimo galimybėmis.
- Suprojektuota sistema toliau gali būti vystoma. Pagrindinės tobulinamos sritys būtų esamos saugyklos informacijos redagavimo galimybės suteikimas, prieigos kontrolė, patogesnis taisyklių atvaizdavimo būdas.
- Veiklos taisyklių saugykla gali būti eksploatuojama ir autonomiškai, kaip atskiras įrankis taisyklių kaupimui, ir kaip veiklos taisyklių pagrindu funkcionuojančios sistemos sudėtinė dalis, tuo būdu užtikrinant programinių sistemų lankstumą.
- VT saugykla naudinga ne tik veiklos analitikui, bet ir projektuotojui ar programuotojui. Projektuotojas esamą saugyklos struktūrą gali integruoti į bendrą veiklos taisyklių sistemą, o programuotojas gali ją naudoti taikomosiose programose, naudojančiose veiklos taisykles.
- Tyrimo medžiagos pagrindu parašytas straipsnis „Veiklos taisyklių klasifikavimas ir saugyklos architektūra“, kuris buvo pristatytas Tarpuniversitetinėje magistrantų ir doktorantų konferencijoje Informacinės technologijos'05.

NAUDOTA LITERATŪRA

- [1] Bajec M., Krisper M. A Methodology and Tool Support for Managing Business Rules in Organizations. Article in Press.
- [2] Baltrušaitis E., Butleris R. Veiklos taisyklių klasifikavimas ir saugyklos architektūra, Informacinės technologijos'05, 10-oji tarpuniversitetinė magistrantų ir doktorantų konferencija, Konferencijos pranešimų medžiaga, Kaunas, Technologija, 2005, p. 112-116.
- [3] Bizagi. Business Rules [žiūrėta 2005 05 10]. Prieiga per internetą:
<http://www.visionsoftware.biz/english/resources/BusinessRules.htm>
- [4] Business Rules Background. European Business Rules Conference, 2005 [žiūrėta 2005 05 10]. Prieiga per internetą: <http://www.eurobizrules.org/download/ebrcbackground.pdf>
- [5] Business Rules Group. Business Rules Manifesto. The Principles of Rule Independence, version 2.0, 2003 [žiūrėta 2005 05 10]. Prieiga per internetą:
<http://www.businessrulesgroup.org/brmanifesto/BRManifesto.pdf>
- [6] Business Rules Taxonomies. EBG Consulting Inc, 2004 [žiūrėta 2005 05 10]. Prieiga per internetą: <http://dotnet.wikis.com/wc.dll?DotNet~BusinessRuleTaxonomies>
- [7] Business Rules. Powering Business and e-Business [žiūrėta 2005 05 10]. Prieiga per internetą:
http://www.kstec.co.kr/products/rules/wp_businessrules.pdf
- [8] Chisholm M. How to Build a Business Rules Engine. Morgan Kaufman Publishers, 2004.
- [9] Date C. J. Twelve Rules for Business Rules, 2000 [žiūrėta 2005 05 10]. Prieiga per internetą:
<http://www.alphora.com/12Rules-Date.pdf>
- [10] Halle B. V. Business Rules Applied: Building Better Systems Using the Business Rules Approach, John Wiley & Sons, 2001.
- [11] Hay D., Healy K. A., Kolber A. ir kt. GUIDE Business Rules Project: Final Report, revision 1.3, GUIDE International Corporation, 2000 [žiūrėta 2005 05 10]. Prieiga per internetą:
http://www.businessrulesgroup.org/first_paper/BRG-whatBR_3ed.pdf
- [12] Kardasis P., Loucopoulos P. Expressing and Organizing Business Rules. Article in Press.
- [13] Knowledge Engineering and Management. The CommonKADS Methodology [žiūrėta 2005 05 10]. Prieiga per internetą: <http://www.commonkads.uva.nl/ck-preface.pdf>
- [14] Lebedys E., Vasilecas O. Verslo taisyklių modeliavimo kalbų analizė. Konferencijos Informacinės technologijos 2004 medžiaga, Kaunas, Technologija, 2004, p. 487-494.
- [15] MDC Open Information Model (OIM) [žiūrėta 2005 05 10]. Prieiga per internetą:
<http://xml.coverpages.org/mdc-oim.html>

- [16] Motiejūnas L., Butleris R.. Veiklos taisyklių manipuliavimo mechanizmų analizė, „Informacinės technologijos 2003“, Kaunas, Technologija, 2003, p. 82-90.
- [17] Nemuraitė L., Ažubalis K. Veiklos taisyklių modeliavimas objektiškai orientuotose metoduose, „Informacinės technologijos 1999“, Kaunas, Technologija, 1999, p. 171-178.
- [18] Rai V. K., Anantaram C. Structuring Business Rules Interactions. Electronic Commerce Research and Applications, 2004.
- [19] Reeder J. Templates for Capturing Business Rules [žiūrėta 2005 05 10]. Prieiga per internetą: <http://www.brcommunity.com/b056.php>
- [20] Response to MDC/Microsoft Business Rules Metamodel by Business Rules Group [žiūrėta 2005 05 10]. Prieiga per internetą: <http://www.businessrulesgroup.org/brg-mdc/BRG-MDC.pdf>
- [21] Ross R. G. Principles of the Business Rules Approach. Addison-Wesley, 2003.
- [22] Ross R. G. The Business Rule Book (Second Edition). Business Rule Solution, Houston, 1997.
- [23] The RAP Group Papers. Part 2 Rules and Models [žiūrėta 2005 05 10]. Prieiga per internetą: <http://www.msg.uk.com/rappapers/introduction.htm>
- [24] Weiden M., Hermans L., Schreiber G., Zee S. Classification and Representation of Business Rules [žiūrėta 2005 05 10]. Prieiga per internetą: <http://neptune.irit.fr/Biblio/02-12-18.pdf>

PRIEDAI

1 priedas. Veiklos taisyklių klasifikavimas ir saugyklos architektūra

Baltrušaitis E., Butleris R. Veiklos taisyklių klasifikavimas ir saugyklos architektūra, Informacinės technologijos, 10-oji tarpuniversitetinė magistrantų ir doktorantų konferencija, Konferencijos pranešimų medžiaga, Kaunas, Technologija, 2005, p. 112-116.

VEIKLOS TAISYKLIŲ KLASIFIKAVIMAS IR SAUGYKLOS ARCHITEKTŪRA

Egidijus Baltrušaitis, Rimantas Butleris

*Informacijos sistemų katedra, Kauno technologijos universitetas,
Studentų g. 50-308, 51368 Kaunas*

Straipsnyje aptariami veiklos taisyklių klasifikavimo būdai. Jų pagrindu siūlomas veiklos taisyklių klasifikavimo metodas. Pateikiama metodika veiklos taisyklių struktūrizavimo procesui įgyvendinti. Sudaryto veiklos taisyklių klasifikavimo metodo pagrindu pateikiama veiklos taisyklių saugyklos architektūra.

1. Įvadas

Šiuo metu egzistuoja nemažai veiklos taisyklių klasifikavimo būdų. Nėra vieningos nuomonės veiklos taisyklių klasifikavimo, surinkimo ir net apibrėžimo klausimais. Dėl šios priežasties organizacijos, pavieniai mokslininkai, priklausomai nuo savo darbo specifikos, kuria savas veiklos taisyklių klasifikavimo schemas. Dažnai visuotino pripažinimo sulaukę metodai yra praplečiami ar susiaurinami paimant tam tikrą jų dalį, reikalingą tam tikrai veiklos sričiai.

Straipsnyje bus aptariami paplitę veiklos taisyklių klasifikavimo metodai. Jų pagrindu pasiūlytas veiklos taisyklių klasifikavimo metodas, kuris leidžia efektyviai ir daug šios srities žinių neturinčiam vartotojui surinkti ir specifikuoti veiklos taisykles. Siūlomo veiklos taisyklių klasifikavimo metodo pagrindu pateikiama veiklos taisyklių saugyklos architektūra.

2. Veiklos taisyklių klasifikavimo metodų apžvalga

2.1. Veiklos taisyklių klasifikavimo metodai

Šiuo metu yra daug veiklos taisyklių klasifikavimo metodų. 1 lentelėje pateiktas apibendrintas klasifikacijų sąrašas, kuriame pateikiamos įvairių autorių siūlomos veiklos taisyklių klasifikavimo schemas.

1 lentelė. Veiklos taisyklių klasifikavimo metodų sąrašas.

Šaltinis	Trumpas klasifikavimo aprašymas
GUIDE Business Rules Project [5]	<ul style="list-style-type: none"> ▪ Struktūrinis teiginys (terminai, faktai) ▪ Veiksmo teiginys (integralumo apribojimai, sąlygos, autorizacija) ▪ Loginės išvados (skaičiavimai, išvestys)
Ronald G. Ross [6]	<ul style="list-style-type: none"> ▪ 32 atominių taisyklių tipai, kurie grupuojami į 7 šeimas: egzempliorių patvirtintojai, tipo patvirtintojai, pozicijos patvirtintojai, funkciniai patvirtintojai, palyginamieji įvertintojai, matematiniai įvertintojai, projekcijos valdikliai ▪ 58 išvestinių taisyklių tipai, kurie yra grupuojami į 12 šeimų: egzempliorių testavimo, pozicijos patikrinimo, atributų modifikavimo, sekos valdymo, sekos specifikuojimo, kompozicijos struktūrų testavimo, sąlyginio laiko įvertinimo, atnaujinimų įvertinimo, veiklos koordinavimo, egzempliorių įgalinimo, egzempliorių kopijavimo, egzempliorių iškvietimo <p>Pastaba: išvestinės taisyklės gali būti sudarytos iš keleto atominių arba kitų išvestinių taisyklių</p>
CommonKADS metodologijos atmaina [8]	<ul style="list-style-type: none"> ▪ Struktūrinės taisyklės (konceptijos struktūra, galiojimas, kaupimas) ▪ Elgsenos taisyklės (informacijos srautų, kontrolės srautų, prieš-sąlygos, po-sąlygos, dažnumo, trukmės, užduoties žinių)

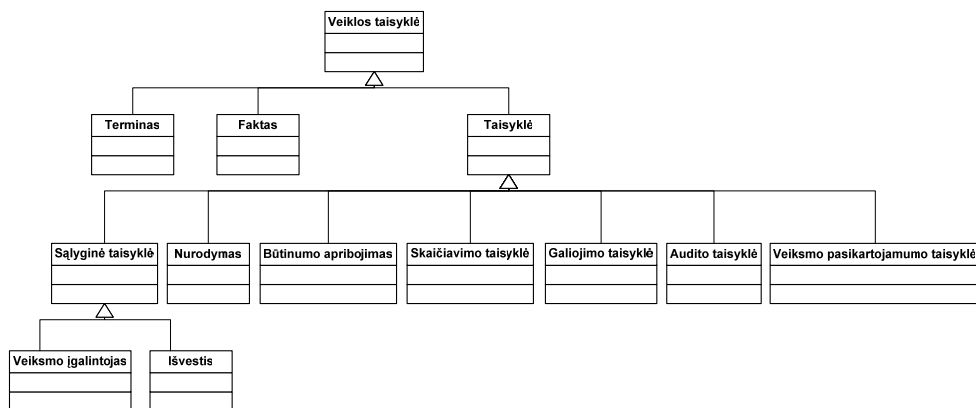
Šaltinis	Trumpas klasifikavimo aprašymas
	<ul style="list-style-type: none"> ▪ Valdymo taisyklės (organizacinė, tikslo ir vertės, aktoriaus kompetencijos, aktoriaus atsakomybės, resursų) Pastaba: kiekvienam taisyklės tipui yra pateikiamas aiškinamasis klausimas, kuris gali būti naudojamas kaip pradinis taškas detaliam VT studijavimui.
Rules and Patterns grupė [7]	<ul style="list-style-type: none"> ▪ Terminai (struktūriniai, elgsenos). Elgsenos terminai yra daiktų vardai, kurie yra trumpalaikiai (pereinantys). Tai įvykių ir operacijų vardai elgsenos modelyje ▪ Faktai (struktūriniai, elgsenos). Elgsenos faktas susieja elgsenos terminus su struktūriniais terminais tam tikru elgsenos sakiniu. ▪ Apribojimai (struktūriniai, elgsenos). Elgsenos apribojimai yra prieš-sąlygos, kurios galioja vienam ar keliems įvykių tipams, bet ne visiems. ▪ Išvestys (struktūrinės, elgsenos). Elgsenos išvestis yra po-sąlyga, kuri specifikuoja duomenų reikšmes po įvykio apdoravimo.
SEEC korporacija [1]	<ul style="list-style-type: none"> <li style="width: 50%;">▪ Atvaizdavimo taisyklės <li style="width: 50%;">▪ Poveikio/atsako taisyklės <li style="width: 50%;">▪ Apribojimai (būsenos, perėjimo) <li style="width: 50%;">▪ Proceso taisyklės <li style="width: 50%;">▪ Išvados (skaičiavimai, išvestys) <li style="width: 50%;">▪ Saugumo ir audito taisyklės
Barbara von Halle [4]	<ul style="list-style-type: none"> <li style="width: 50%;">▪ Terminai <li style="width: 50%;">▪ Nurodymai <li style="width: 50%;">▪ Faktai <li style="width: 50%;">▪ Išvestys <li style="width: 50%;">▪ Būtinumo apribojimai <li style="width: 50%;">▪ Veiksmo įgalintojai <li style="width: 50%;">▪ Skaičiavimai

2.2. Klasifikavimo metodų apibendrinimas

Apibendrinant aptartus VT klasifikavimo metodus galime teigti, kad GUIDE projektas [5] (dalina ir Roso metodas [6]) yra naudingas kaip teorinė VT koncepcija. SEEC ir Barbaros von Halle modeliai [1, 4] yra geriau pritaikyti praktiniam naudojimui. CommonKADS metodologijos pagrindu sudarytas metodas [8] naudingas bendresniam veiklos taisyklių supratimui. RAP grupės pasiūlytas metodas [7] praplečia GUIDE metodo klasifikaciją, įvedant struktūrinės ir elgsenos dedamąsias. Barbaros von Halle klasifikacija iš visų aptartųjų klasifikacijų yra aiškiausia ir priimtinausia. Visi aptarti metodai parodo, kad nėra vieningos nuomonės VT klasifikavimo atžvilgiu. Kiekviena įmonė, organizacija ar pavieniai projektuotojai pagal poreikius kuria savo klasifikavimo metodus (dažnai remdamiesi visuotiniai pripažintais klasifikavimo metodais), kurie geriausiai atspindi jų veiklos sritį. Nepaskutinis dalykas yra taisyklių realizacija, kurią projektuotojai laiko pirmoje vietoje, nors pagal visus veiklos taisyklių standartus klasifikuojant VT reiktų atsiriboti nuo realizacijos detalių.

3. Pasiūlytas veiklos taisyklių klasifikavimo metodas

Atlikus veiklos taisyklių klasifikavimo metodų analizę, buvo sudarytas VT specifikuojantis metodas, kurio pagrindą sudaro Barbaros von Halle [4] ir CommonKADS [8] veiklos taisyklių klasifikavimo principai. Sudarytas veiklos taisyklių klasifikavimo metodas pateiktas 1 pav. Kiekvieną taisyklę galima priskirti atskiram veiklos objektui (esybei).



1 pav. Veiklos taisyklių klasifikacija pasiūlytu metodu.

Pasiūlyta veiklos taisyklių klasifikacija ir detalus kiekvienos taisyklės paaiškinimas pateiktas 2 lentelėje. Kadangi tolesniame tyrime didžiausias dėmesys bus skiriamas VT šablonams, tai lentelėje pateikiami ir atitinkamų VT šablonai. Kaip teigia M. Bajec ir M. Krisper [2], kiekvienas veiklos taisyklės tipas turi turėti šabloną. VT šablonas parodo, kaip specifikuoti taisyklės, priklausančias tam tikram veiklos taisyklių tipui.

2 lentelė. Veiklos taisyklių tipų aprašymas.

VT tipas	Detalus VT paaiškinimas	VT šablonas
Terminas	Daiktavardis ar daiktavardinė frazė su iš anksto susitartu apibrėžimu. Pavyzdžiui: <ul style="list-style-type: none"> objekto klasė, esybė; elementas, atributas, reikšmė 	<terminas>YRA APIBRĖŽIAMAS KAIP<tekstas>
Faktas	Išraiška, sujungianti terminus prielinksnių ir veiksmazodžių pagalba į turinčias prasmę veiklos pastabas: <ul style="list-style-type: none"> esybė – esybė, esybė – atributas supertipo – subtipo sujungimas 	<terminas1>YRA<terminas2> <terminas1>YRA SUDARYTAS IŠ<terminas2> <terminas1>TURI SAVYBĘ<terminas2>
Skaičiavimo taisyklė	Išraiška, nusakanti algoritmą skaitinei termino reikšmei apskaičiuoti. Formulėje galima nurodyti reikiamą terminų skaičių, tačiau jie privalo būti iš anksto apibrėžti	<terminas1>YRA PASKAIČIUOJAMAS KAIP<formulė>
Būtinumo apribojimas	Išraiška, išreiškianti sąlygas, kurios turi būti tenkinamos (arba netenkinamos) tam tikram veiklos atvejui	<terminas1>TURI TURĖTI<bent vieną, visus, ...><terminas2> <terminas1>TURI BŪTI<palyginimas> {<terminas2><reikšmė>,<reikšmių sąrašas>} <terminas1>(NE)TURI BŪTI SĄRAŠE<a, b>
Nurodymas	Išraiška, išreiškianti perspėjimą apie sąlygas, kurios turėtų būti tenkinamos (arba netenkinamos)	<terminas1>TURĖTŲ TURĖTI<bent vieną, visus, ...><terminas2> <terminas1>TURĖTŲ BŪTI<palyginimas> {<terminas2>,<reikšmė>,<reikšmių sąrašas>} <terminas1>(NE)TURĖTŲ BŪTI SĄRAŠE<a>
Išvestis	Išraiška, kuri patikrina tam tikras sąlygas, ir joms esant teisingoms, patvirtina naujo fakto teisingumą	JEI <terminas1><operatorius><terminas2>, reikšmė, reikšmių sąrašas>IR<...>TADA <terminas3><operatorius><terminas4>
Veiksmo įgalintojas	Išraiška, kuri patikrina tam tikras sąlygas ir inicijuoja kitą veiklos operaciją, pranešimą ar įvykį.	JEI<terminas1><operatorius><terminas2> TADA<veiksmas>
Galiojimo taisyklė	Išraiška, kuri parodo kiek laiko turi būti laikoma (arba kada panaikinama) informacija apie tam tikrą objektą	<terminas1>GALIOJA<periodas> <terminas1>TURI BŪTI SUNAIKINTAS po <periodas, veiksmas>
Audito taisyklė	Išraiška, kuri nusako ar kaupiti visas tam tikro atributo reikšmes	Visi <terminas1>pakeitimai TURI BŪTI REGISTRUOJAMI
Veiksmo pasikartojamumo taisyklė	Išraiška, nurodanti kaip dažnai bus atliekamas tam tikras veiksmas	<veiksmas>BUS ATLIEKAMAS<skaičius> KARTŲ PER<laiko periodas>

4. Veiklos taisyklių saugyklos architektūra

Suprojektuotą veiklos taisyklių saugyklos architektūrą vaizduoja 2 pav. Veiklos taisyklių įvedimas pagal pasiūlytą metodą turi prasidėti nuo funkcijų apibrėžimo, kurioms bus formuluojamos veiklos taisyklės. Jos suvedamos į lentelę *funkcija*. Funkcija čia laikomas tam tikras sistemos panaudojimo atvejis. Todėl pirmiausia patartina apibrėžti visus sistemos panaudojimo atvejus. Kiekviena funkcija toliau skaidoma į įvykius (lentelės *ivykis* ir *vt_ivykis*). Įvykis - tai konkretus sistemos įvykis, kuriam yra formuluojamos veiklos taisyklės. Lentelėje *veiklos_taisykle* saugoma pagrindinė veiklos taisyklių informacija, o jos pilna išraiška saugoma lentelėje *vt_pilna_israiska*. Lentelėje *terminas* saugomi visi terminai ir jų aprašymai. Lentelėje *vt_sablonas* saugoma informacija apie šablonus, o visi šablonų tipai saugomi lentelėje *sablonu_tipai*. Šablonai susideda iš apibrėžtų elementų (terminai, rezervuoti žodžiai ir konstantos). Tam realizuoti lentelėje *sablono_elementas* saugoma informacija apie šablone esančius elementus ir jų eilės tvarką. Rezervuoti žodžiai saugomi lentelėje *rezervuotas_zodis*. Faktai saugomi lentelėje *faktas*. Lentelė *fakto_sablonas* yra pagalbinė, apibūdinanti tai, jog faktams sudaryti reikalingi šablonai. Lentelėje *fakto_terminas* saugomi terminai, įeinantys į konkrečių faktų sudėtį. Veiklos taisyklių konstantos saugomos lentelėje *konstanta*. Terminai, įeinantys į šablonus, saugomi lentelėje *sablono_terminas*.

5. Veiklos taisyklių struktūrizavimo proceso aprašymas

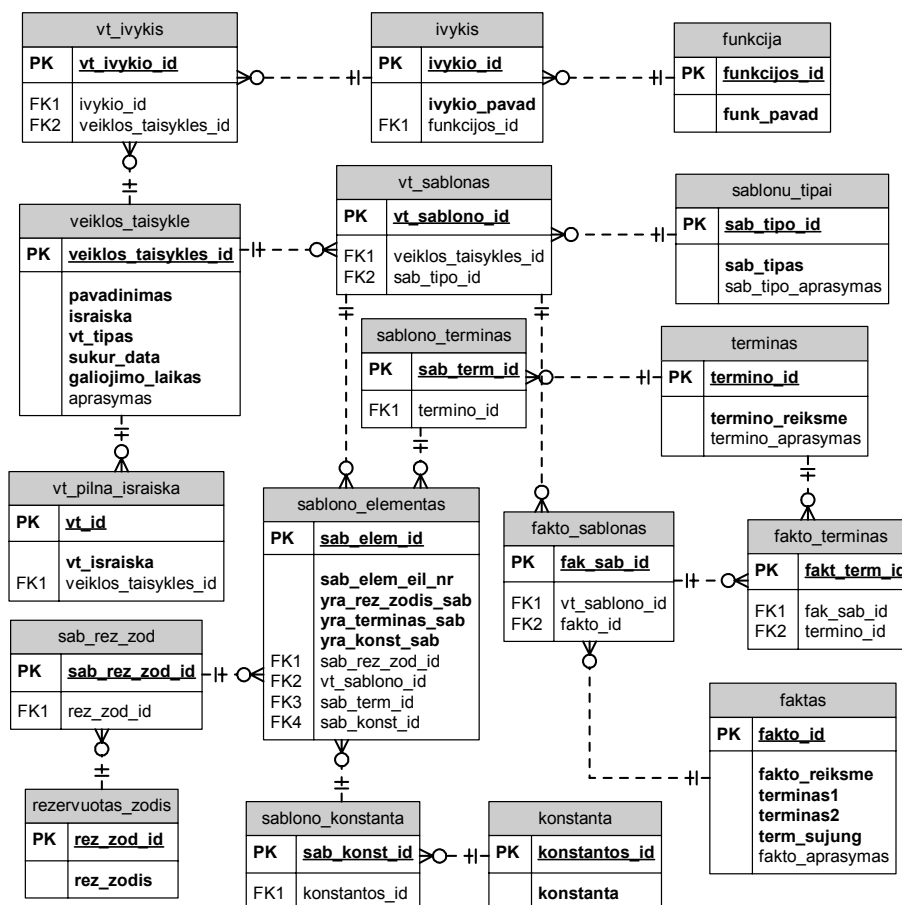
Visas veiklos taisyklių struktūrizavimo procesas gali būti padalintas į keturis etapus. Pirmiausia atliekama dalykinės srities analizė, kurios metu bandoma surinkti kuo daugiau informacijos apie veiklos sritį. Po to, remiantis

veiklos taisyklių manifestu [3], yra išskiriami veiklos taisyklių teiginiai. Trečiojo etapo metu veiklos taisyklės užrašomos formaliu pavidalu. Šiame etape remiamasi pasiūlytais veiklos taisyklių klasifikavimo šablonais. Paskutinis, ketvirtasis etapas, yra veiklos taisyklių saugyklos projektavimas. Veiklos taisyklių saugykla projektuojama remiantis pasiūlyto klasifikavimo metodo pagrindu.

6. Išvados

Straipsnyje aptarti keletas žinomiausių ir geriausiai charakterizuojančių esamas veiklos taisyklių klasifikacijas metodų. Kiekvienas iš jų turi savų privalumų ir trūkumų, todėl, atlikus tyrimą, pasiūlytas veiklos taisyklių klasifikavimo metodas, kuris yra paprastas ir aiškiai parodo daugumą veiklos taisyklių panaudojimo atvejų. Trumpai aptartas veiklos taisyklių struktūrizavimo procesas, kurio etapais reiktų remtis veiklos taisyklių sistemų projektavimo metu. Pasiūlyto klasifikavimo metodo pagrindu sudaryta veiklos taisyklių saugykla, kuri leidžia saugoti visas apibrėžtas formalizuotas veiklos taisykles.

Ateityje numatoma suprojektuoti vartotojo sąsają, kuri leis įvesti, redaguoti ir peržiūrėti duomenų bazėje specifikuotas veiklos taisykles. Esant vartotojo sąsajai, toliau bus bandoma pritaikyti tam tikras veiklos taisykles realiose situacijose, t.y. pasirinkti tam tikrą tyrimo objektą ir jo funkcijoms bei įvykiams pritaikyti veiklos taisykles, kurios būtų sužadinamos prie tam tikrų įvykių.



2 pav. Veiklos taisyklių saugyklos architektūra pasiūlytam veiklos taisyklių klasifikavimo metodui.

Literatūros sąrašas

- [1] Automated Business Rule Mining, Facilitating Rapid Transformation of Legacy Systems. *A SEEC, Inc. White Paper*, 2003. [žiūrėta 2005 04 02], prieiga per internetą: http://www.asera.com/reslibrary/SEEC_BusinessRuleMining.2003.pdf
- [2] **M. Bajec, M. Krisper**. A Methodology and Tool Support for Managing Business Rules in Organizations. *Article in Press*.
- [3] Business Rules Manifesto, The Principles of Rule Independence, version 2.0, *Business Rules Group*, 2003. [žiūrėta 2005 04 02], prieiga per internetą: <http://www.businessrulesgroup.org/brmanifesto/BRManifesto.pdf>

- [4] **B. V. Halle.** Business Rules Applied: Building Better Systems Using the Business Rules Approach. *John Wiley & Sons*, 2001.
- [5] **D. Hay, K. A. Healy ir kiti.** GUIDE Business Rules Project, Final Report – revision 1.3. *GUIDE International Corporation*, 2000. [žiūrėta 2005 04 02], prieiga per internetą: http://www.businessrulesgroup.org/first_paper/BRG-whatBR_3ed.pdf
- [6] **R. G. Ross.** The Business Rule Book: Classifying, Defining and Modeling Rules. *Business Rule Solution*, Second edition, 1997.
- [7] The RAP Group Papers. Part 2, Rules and Models. [žiūrėta 2005 04 02], prieiga per internetą: <http://www.msg.uk.com/rappapers/introduction.htm>
- [8] **M. Weiden, L. Hermans, G. Schreiber, S. Zee.** Classification and Representation of Business Rules. [žiūrėta 2005 04 02], prieiga per internetą: <http://neptune.irit.fr/Biblio/02-12-18.pdf>

BUSINESS RULES CLASSIFICATION AND REPOSITORY ARCHITECTURE

The most common business rules classification schemes are presented in this article. Basing on the analysis results business rules classification method is proposed. Also business rules structuring process is shortly described. Finally business rules repository architecture, which is based on the proposed classification method, is presented.

2 Priedas. Sukurtos sistemos prototipo vartotojo sąsajos langų aprašymas

Meniu pasirinkimas Funkcijos

Šiame puslapyje (1 pav.) įvedamos naujos bei išvedamos jau sistemoje esančios funkcijos.



1 pav. Funkcijų puslapis

Meniu pasirinkimas Įvykiai

Šiame puslapyje įvedami įvykiai (2 pav.). Įvedant įvykį, jis tuo pačiu turi būti priskirtas tam tikrai, jau sistemoje esančiai funkcijai. Todėl įvedimo metu patartina laikytis 6.5 pav. pateiktos saugyklos pildymo tvarkos. Viena funkcija gali turėti daug įvykių, todėl realizuota filtravimo funkcija: pasirinkus norimą funkcija ir paspaudus mygtuką *Filtruoti* yra išvedami visi pasirinktai funkcijai priskirti įvykiai. Norint vėl pamatyti visus sistemoje esančius įvykius reikia paspausti mygtuką *Visi*. Ta pati įrašų filtravimo ir peržiūrėjimo tvarka realizuota visoje sistemoje.



VEIKLOS TAISYKLIŲ SAUGYKLA

LENTELĖ "IVYKIS"

Šioje lentelėje saugomi funkcijų įvykiai

Ivykio pavadinimas:

Funkcijos pavadinimas:

Ivykio_ID	Ivykio pavadinimas	Funkcijos_ID
1	Bendros sumos skaičiavimas	3
2	Atributų kūrimas	1
3	Prekių kiekio patikrinimas	3
4	Nuolaidos skaičiavimas	10
5	Užsakovo priskyrimas	4

M E N I U

[Funkcijos](#)
[Ivykiai](#)
[Taisyklės](#)
[Ivykio taisyklės](#)
[Terminai](#)
[Rezervuoti žodžiai](#)
[Konstantos](#)
[Šablonų tipai](#)
[Taisyklių šablonai](#)
[Šablono elementai](#)
[Faktai](#)
[Faktų šablonai](#)

ATASKAITOS:
[Funkcijos-Ivykiai-Taisyklės](#)
[Šablonizuotos taisyklės](#)
[Faktai](#)
[Terminai](#)
[Rezervuoti žodžiai](#)
[Konstantos](#)

2 pav. Įvykių puslapis

Meniu pasirinkimas Taisyklės

Šiame puslapyje (3 pav.) įvedamos veiklos taisyklės ir jų pilnosios išraiškos (t. y. taisyklė natūralia kalba). Šios dvi lentelės pildomos viename puslapyje. Taisyklės pilna išraiška gali būti įvesta tuomet, kuomet jau yra įvesta pati taisyklė. Norint nekartoti tos pačios taisyklės išraiškos (ar peržiūrėti jau esamų taisyklių išraiškas) yra realizuotas filtravimas, kuomet pasirinkus veiklos taisyklę ir paspaudus mygtuką *Filtruoti* yra išvedama taisyklės pilna išraiška (jei ji yra).



VEIKLOS TAISYKLIŲ SAUGYKLA

LENTELĖ "VEIKLOS_TAISYKLE"

Veiklos taisyklės pavadinimas:

Veiklos taisyklės pilnas tekstas:

Veiklos taisyklės tipas pagal klasifikaciją:

Veiklos taisyklės galiojimo laikas, iki kurio ji turi būti aktyvi sistemoje:

Veiklos taisyklės aprašymas:

LENTELĖ "VT_PILNA_ISRAISKA"

Veiklos taisyklės išraiška natūralia kalba:

Veiklos taisyklės pavadinimas:

M E N I U

[Funkcijos](#)
[Ivykiai](#)
[Taisyklės](#)
[Ivykio taisyklės](#)
[Terminai](#)
[Rezervuoti žodžiai](#)
[Konstantos](#)
[Šablonų tipai](#)
[Taisyklių šablonai](#)
[Šablono elementai](#)
[Faktai](#)
[Faktų šablonai](#)

ATASKAITOS:
[Funkcijos-Ivykiai-Taisyklės](#)
[Šablonizuotos taisyklės](#)
[Faktai](#)
[Terminai](#)
[Rezervuoti žodžiai](#)
[Konstantos](#)

3 pav. Taisyklių puslapis

Meniu pasirinkimas Įvykio taisyklės

Šiame puslapyje (4 pav.) susiejamos taisyklės su įvykiais, t. y. įvykiams yra priskiriamos taisyklės. Realizuoti du pasirinkimo laukai. Vienas jų patiekia visus sistemoje esančius įvykius, kitas – visų sistemoje esančių veiklos taisyklių pavadinimus. Pasirinkus įvykį ir taisyklę, bei paspaudus mygtuką *Įvesti* įvykdomas įvykio ir taisyklės susiejimas. Realizuotas filtravimas pagal abu laukus, t. y. pasirinkus filtruoti pagal įvykį, išvedamos visos tam įvykiui priskirtos veiklos taisyklės. Atitinkamai vykdomas filtravimas pagal veiklos taisyklę.

VEIKLOS TAISYKLIŲ SAUGYKLA

LENTELĖ "VT_IVYKIS"

Šioje lentelėje susiejami įvykiai ir taisyklės

Įvykio pavadinimas:

Veiklos taisyklės pavadinimas:

Vt_ivykio_ID	Ivykio_ID	Veiklos taisyklės_ID
1	1	1
8	5	7
3	1	2
4	4	3
5	2	4
6	2	5
7	2	6

4 pav. Įvykio taisyklių puslapis

Meniu pasirinkimas Terminai, Rezervuoti žodžiai, Konstantos

Terminų, rezervuotų žodžių ir konstantų įvedimui yra skirti atskiri puslapiai. Jų visų struktūra yra panaši ir pateikta 5 pav. Įvedimo formos apačioje (kaip ir visoje sistemoje) yra išvedami visi atitinkamos duomenų bazės lentelės įrašai.

VEIKLOS TAISYKLIŲ SAUGYKLA

M E N I U

[Funkcijos](#)
[Ivykiai](#)
[Taisyklės](#)
[Ivykio taisyklės](#)
[Terminai](#)
[Rezervuoti žodžiai](#)
[Kontantos](#)
[Šablonų tipai](#)
[Taisyklių šablonai](#)
[Šablono elementai](#)
[Faktai](#)
[Faktų šablonai](#)

ATASKAITOS:
[Funkcijos-Ivykiai-Taisyklės](#)
[Šablonizuotos taisyklės](#)
[Faktai](#)
[Terminai](#)
[Rezervuoti žodžiai](#)
[Konstantos](#)

LENTELĖ "TERMINAS"

Šioje lentelėje saugomi visi terminai ir jų aprašymai

Termino reikšmė:

Terminą charakterizuojantis aprašymas:

Termino_ID	Termino reikšmė	Terminą charakterizuojantis aprašymas
1	Rinkėjas	
2	Klientas	
3	Automobilis	
4	Draudimas	
5	Apraustas	
6	Užsakymas	
7	Dienas	
8	Studento numeris	
9	Studento atributas	
10	Kompiuterio kaina	
11	Pirkimo kiekis	
12	Nuolaida	
13	Procentai	
14	Gynimas	
15	Kreditas	
17	Užsakovas	

5 pav. Terminų puslapis

Menu pasirinkimas Šablonų tipai

Šiame puslapyje (6 pav.) yra suvedami šablonų tipai kiekvienai veiklos taisyklių kategorijai.

VEIKLOS TAISYKLIŲ SAUGYKLA

M E N I U

[Funkcijos](#)
[Ivykiai](#)
[Taisyklės](#)
[Ivykio taisyklės](#)
[Terminai](#)
[Rezervuoti žodžiai](#)
[Kontantos](#)
[Šablonų tipai](#)
[Taisyklių šablonai](#)
[Šablono elementai](#)
[Faktai](#)
[Faktų šablonai](#)

ATASKAITOS:
[Funkcijos-Ivykiai-Taisyklės](#)
[Šablonizuotos taisyklės](#)
[Faktai](#)
[Terminai](#)
[Rezervuoti žodžiai](#)
[Konstantos](#)

LENTELĖ "ŠABLONU_TIPAI"

Šioje lentelėje saugomi taisyklių šablonų tipai

Šablono tipas:

Šablono tipo aprašymas:

Šablono tipo_ID	Šablono tipas	Šablono tipo aprašymas
1	Skaiciavimas	
2	Būtinumo apribojimas	
3	Nurodymas	
4	Išvestis	
5	Veiksmo igalintojas	
6	Galiojimas	
7	Registravimas	
8	Veiksmo pasikartojamumas	

6 pav. Šablonų tipų puslapis

Meniu pasirinkimas Taisyklių šablonai

Šiame puslapyje (7 pav.) veikos taisyklei yra sukuriamas šablonas: suvedamas veiklos taisyklės šablono pavadinimas, pasirenkama taisyklė bei taisyklių kategorijos šablono tipas. Pasiūlytoje VT klasifikacijoje kiekvienas taisyklės tipas gali turėti kelis šablonus (tai realizuoja lentelė *vt_sablonas*), priklausančius tam pačiam bendram taisyklių kategorijos šablonų tipui (lentelė *sablonu_tipai*).

VT Šablono_ID	Veiklos taisyklės_ID	VT Šablono pavadinimas	Šablono tipo_ID
1	1	Draudimas	2
2	2	Užsakymo galiojimas	6
3	3	Užsakymo nuolaidos skaičiavimas	4
17	7	Užsakymo užsakovas	2

7 pav. Taisyklių šablonų puslapis

Meniu pasirinkimas Šablonų elementai

Šiame puslapyje (8 pav.) taisyklės šablonas užpildomas elementais. Elementais gali būti terminai, rezervuoti žodžiai ir konstantos. Pasirenkamas veiklos taisyklės šablonas, kuris bus pildomas. Realizuotas pasirinkto taisyklės šablono elementų filtravimas, kad būtų matoma, kokius elementus jau turi pasirinktas taisyklės šablonas. Elemento eilės numeris suvedamas į tam skirtą formos lauką. Po to pasirenkamas elemento tipas ir, priklausomai nuo jo tipo, toliau iš atitinkamo sąrašo pasirenkama šio elemento reikšmė.

Pastaba: formoje rodomi visi sistemoje suvesti elementai (terminai, rezervuoti žodžiai ir konstantos), t. y. įvedimo metu formoje matomos jų pirmosios reikšmės. Bet jos nėra visos įrašomos mygtuko *Įvesti* paspaudimu (nes tuomet nebūtų įmanomas teisingas tokios taisyklės atkūrimas). Įrašoma tik to elemento reikšmė, kurio tipas buvo pasirinktas sąrašo *Šablono elemento tipas*.

VEIKLOS TAISYKLIŲ SAUGYKLA

LENTELĖ "SABLONO_ELEMENTAS"

Šioje lentelėje saugoma informacija apie šablone esančius elementus
ir jų eilės tvarką šablone

Veiklos taisyklės šablono pavadinimas:

Elementų eilės numeris šablone:

Šablono elemento tipas:

Šablono rezervuotas žodis:

Šablono terminas:

Šablono konstanta:

vt_sablono_id	sab_elem_id	elem_eil_nr	yra_rez_zodis_sab	yra_terminas_sab	yra_konst_sab	sab_rez_zod_id	sa
1	1	1	0	1	0	0	
1	2	2	1	0	0	4	
1	3	3	0	1	0	0	
2	4	1	0	1	0	0	
2	5	2	1	0	0	5	
2	6	3	0	0	1	0	
2	7	4	0	1	0	0	

8 pav. Šablono elementų puslapis

Meniu pasirinkimas Faktai

Šiame puslapyje (9 pav.) yra suvedama faktus apibūdinanti informacija.

VEIKLOS TAISYKLIŲ SAUGYKLA

LENTELĖ "FAKTAS"

Šioje lentelėje saugomi visi faktai

Fakto reikšmė:

Pirmasis fakto terminas:

Nusako koks ryšio tipas tarp fakto terminų:

Antrasis fakto terminas:

Faktą charakterizuojantis aprašymas:

Fakto_ID	Fakto reikšmė	Terminas1	term_sujung	Terminas2	Faktą charakterizuojantis aprašymas
1	Studento numeris yra studento atributas	Studento_numeris	yra	Studento_atributas	

9 pav. Faktų puslapis

Meniu Pasirinkimas Ataskaitos

Šiame sukurtos sistemos prototipe kiekvienos duomenų bazės lentelės turiniai yra pateikiami kiekvienos lentelės pildymo puslapiuose. Tačiau detalai saugykloje esančios informacijos analizei to nepakanka. Tuo tikslu yra formuojamos apibendrintos duomenų ataskaitos. Be apibendrintų ataskaitų dar pateikiamos saugykloje esančios faktų, terminų, rezervuotų žodžių ir konstantų lentelių ataskaitos.

Funkcijų-Įvykių-Taisyklių ataskaita

Šioje ataskaitoje pateikiamos visos funkcijos, jų įvykiai ir įvykių taisyklės (10 pav.). Jei įvykis neturi taisyklės (t. y. jam nėra priskirtos taisyklės) tai taisyklės pavadinimo grafoje išvedamas tekstas [0](Įvykis neturi taisyklės). Prieš kiekvieną funkcijos, įvykio ir taisyklės pavadinimą esančiuose laužtiniuose skliaustuose pateikiami atitinkami funkcijų, įvykių ir taisyklių identifikaciniai numeriai.

VEIKLOS TAISYKLIŲ SAUGYKLA		
MENIU		
Funkcijos Įvykiai Taisyklės Įvykio taisyklės Terminai Rezervuoti žodžiai Kontantos Šablonų tipai Taisyklių šablonai Šablono elementai Faktai Faktų šablonai		
FUNKCIJOS, ĮVYKIAI IR TAISYKLĖS		
Funkcijos pavadinimas	Įvykio pavadinimas	Taisyklės pavadinimas
[1]Studentų registravimas	[2]Atributų kūrimas	[4]Studento pažymėjimas
		[5]Studento vardas
		[6]Studento gyvenamoji vieta
[3]Kliento aptarnavimas	[1]Bendros sumos skaičiavimas	[1]Automobilio draudimas
		[2]Užsakymo galiojimas
	[3]Prekių kiekio patikrinimas	[0](Įvykis neturi taisyklės)
[4]Užsakymo priėmimas	[5]Užsakovo priskyrimas	[7]Užsakovo apribojimas
[10]Užsakymo kainos skaičiavimas	[4]Nuolaidos skaičiavimas	[3]Kompiuterio kainos nuolaidos skaičiavimas
ATASKAITOS:		
Funkcijos-Įvykiai-Taisyklės Šablonizuotos taisyklės Faktai Terminai Rezervuoti žodžiai Konstantos		

10 pav. Funkcijų-Įvykių-Taisyklių ataskaitos puslapis

Šablonizuotų taisyklių ataskaita

Šioje ataskaitoje pateikiami šablonų tipai, su jais susieti taisyklių šablonai ir taisyklės šablono elementai sakinio pavidalu, t.y. atkuriami struktūrizuoti (šablonizuota) veiklos taisyklė (11 pav.). Prieš kiekvieną šablono tipo ir veiklos taisyklės šablono pavadinimą esančiuose laužtiniuose skliaustuose pateikiami atitinkami šablono tipo ir veiklos taisyklės šablono tipo identifikaciniai numeriai.

VEIKLOS TAISYKLIŲ SAUGYKLA			
MENIU			
Funkcijos Ivykiai Taisyklės Ivykio taisyklės Terminai Rezervuoti žodžiai Kontantos Šablonų tipai Taisyklių šablonai Šablono elementai Faktai Faktų šablonai			
ŠABLONŲ TIPAI, TAISYKLIŲ ŠABLONAI IR JŲ ELEMENTAI			
Šablono tipas	VT šablono pavadinimas	Elemento reikšmė	Elemento tipas
[2]Būtinumo apribojimas	[1]Draudimas	Automobilis	terminas
		Turi Būti	rezervuotas zodis
		Apraustas	terminas
Automobilis Turi Būti Apraustas			
[2]Būtinumo apribojimas	[17]Užsakymo užsakovas	Užsakymas	terminas
		Turi turėti	rezervuotas zodis
		Užsakovas	terminas
Užsakymas Turi turėti Užsakovas			
[4]Išvestis	[3]Užsakymo nuolaidos skaičiavimas	Je	rezervuotas zodis
		Kompiuterio_kaina	terminas
		>	rezervuotas zodis
		4500	konstanta
		Ir	rezervuotas zodis
		Pirkimo_kiekis	terminas
		>	rezervuotas zodis
		5	konstanta
		Tuomet	rezervuotas zodis
		Nuolaida	terminas
		5	konstanta
		Procentai	terminas
Je			
Kompiuterio_kaina > 4500 Ir Pirkimo_kiekis > 5 Tuomet Nuolaida 5 Procentai			
[6]Galiojimas	[2]Užsakymo galiojimas	Užsakymas	terminas
		Galioja	rezervuotas zodis
		5	konstanta
		Dienas	terminas
Užsakymas Galioja 5 Dienas			

11 pav. Šablonizuotų taisyklių puslapis

3 Priedas. Veiklos taisyklių atvaizdavimas veiklos taisyklių saugykloje

Turime tokią veiklos taisyklę (būtinumo apribojimą): **Bendrabutis turi turėti budėtojus**. Terminai: *bendrabutis* ir *budėtojas*. Rezervuoti žodžiai: *turi turėti*. Kaip tokia taisyklė atsivaizduoja suprojektuotoje veiklos taisyklių saugykloje parodyta 1 lentelėje.

1 lentelė. Būtinumo apribojimo VT saugojimo atvaizdavimas sukurtoje veiklos taisyklių saugykloje

Lentelė <i>funkcija</i>								
funkcijos_id		funk_pavad						
1		Bendrabučių suvestinių pildymas						
Lentelė <i>ivykis</i>								
ivykio_id		ivykio_pavad			funkcijos_id			
1		Bendrabučio darbuotojų priskyrimas			1			
Lentelė <i>vt_ivykis</i>								
vt_ivykio_id		ivykio_id		veiklos_taisyklės_id				
1		1		1				
Lentelė <i>veiklos_taisykle</i>								
veiklos_taisyklės_id	pavadinimas	israiska	vt_tipas	sukurimo_data	galiojimo_laikas	aprasymas		
1	Bendrabučio budėtojai	Bendrabutis turi turėti budėtojus	būtinumo apribojimas	2005-03-18	2005-07-01			
Lentelė <i>vt_pilna_israiska</i>								
vt_id	vt_israiska				vt_taisyklės_id			
1	Bendrabutis turi turėti budėtojus				1			
Lentelė <i>terminas</i>								
termino_id	termino_reiksme		termino_aprasymas					
1	bendrabutis							
2	budėtojas							
Lentelė <i>vt_sablonas</i>								
vt_sablono_id		vt_sablono_pavad		veiklos_taisyklės_id		sab_tipo_id		
1		Bendrabučio budėtojai		1		1		
Lentelė <i>sablonu_tipai</i>								
sab_tipo_id		sab_tipas		sab_tipo_aprasymas				
1		Būtinumo apribojimas						
Lentelė <i>sablono_elementas</i>								
sab_elem_id	sab_elem_eil_nr	yra_rez_zodis_sab	yra_terminas_sab	yra_konst_sab	sab_rez_zod_id	vt_sablono_id	sab_term_id	sab_konst_id
1	1	0	1	0	0	1	1	0
2	2	1	0	0	1	1	0	0
3	3	0	1	0	0	1	2	0
Lentelė <i>sablono_terminas</i>								
sab_term_id					termino_id			
1					1			
2					2			
Lentelė <i>rezervuotas_zodis</i>								
rez_zod_id				rez_zodis				
1				turi turėti				
Lentelė <i>sab_rez_zodis</i>								
sab_rez_zod_id				rez_zod_id				
1				1				