

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Vaidas Pečiulis

**Veiklos taisyklių saugyklos
valdymo sistema**

Magistro darbas

Darbo vadovas
lekt. dr. T. Skersys

Kaunas, 2008

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Vaidas Pečiulis

**Veiklos taisyklių saugyklos
valdymo sistema**

Magistro darbas

Recenzentas

2008-01-14

doc. dr. G. Vilutis

Vadovas

2008-01-14

lekt.dr. T. Skersys

Atliko

2008-01-11

IFM-2/4 gr. stud.
Vaidas Pečiulis

Kaunas, 2008

TURINYS

SUMMARY	5
1 ĮVADAS	6
2 VEIKLOS TAISYKLIŲ SAUGYKLOS VALDYMO SISTEMŲ ANALIZĖ	8
2.1 VT specifikuavimo metodai	8
2.1.1 VT specifikuavimas naudojant DSL kalbas	9
2.1.2 VT specifikuavimas naudojant grafines specifikuavimo kalbas	9
2.1.3 Veiklos taisyklių struktūrizavimas naudojant šablonus	12
2.2 Veiklos taisyklių valdymo sistemų analizė	14
2.2.1 JBoss Rules	15
2.2.2 Mandarax	15
2.2.3 Oracle Business Rules	15
2.2.4 Visual Rules	16
2.2.5 Quick Rules	16
2.2.6 ILOG Rules	16
2.2.7 Versata Logic Suite	17
2.2.8 Blaze Advisor	17
2.3 Veiklos taisyklių valdymo sistemų palyginimas	18
2.4 Sistemos architektūros ir įgyvendinimo priemonių analizė	20
2.5 Siekiamos sistemos apibrėžimas	21
2.6 Darbo tikslas ir siekiami privalumai	22
2.7 Kompiuterizuojamos sistemos funkcijos	22
2.8 Reikalavimai duomenims	24
2.9 Analizės išvados	25
3 VEIKLOS TAISYKLIŲ SAUGYKLOS REIKALAVIMŲ SPECIFIKACIJA	26
3.1 Nefunkciniai reikalavimai ir apribojimai	26
3.1.1 Reikalavimai standartams	26
3.1.2 Reikalavimai veikimui	26
3.1.3 Suderinamumas su operacinėmis, kitomis sistemomis	26
3.2 Sistemos rizikos faktoriai	27
3.2.1 Sistemos neatitikimas vartotojo reikalavimams:	27
3.2.2 Sistemos stabilumo stoka:	27
3.2.3 Sistemos klaidingas veikimas:	27
3.3 Rezultato kokybės kriterijai	27
3.3.1 Vartotojo reikalavimų išpildymas:	27
3.3.2 Paprastumas:	27

3.3.3	Veikimo greitis:	28
3.4	Funkcinių reikalavimų specifikacija.....	28
3.4.1	Nestruktūrizuotos veiklos taisyklės įvedimo funkcija.....	28
3.4.2	Struktūrizuotos veiklos taisyklės įvedimo funkcija.....	29
3.4.3	Struktūrizuotos VT modifikavimo funkcija.....	31
3.4.4	Veiklos taisyklių peržiūrėjimo funkcija.....	32
3.4.5	Veiklos taisyklių naikinimo funkcija.....	34
3.4.6	Šablono kūrimo funkcija.....	35
3.4.7	Šablono modifikavimo funkcija.....	37
3.4.8	Šablonų peržiūrėjimo funkcija.....	39
3.4.9	Šablono naikinimo funkcija.....	40
3.5	Sistemos duomenų klasių modelis.....	41
3.6	VTS valdymo sistemos klasių modelis.....	46
3.7	Reikalavimų specifikacijos apibendrinimas	46
4	VEIKLOS TAISYKLIŲ SAUGYKLOS VALDYMO SISTEMOS PROJEKTO SPECIFIKACIJA.....	48
4.1	Sistemos sprendimas.....	48
4.2	Sistemos architektūra - statinės struktūros modelis.....	48
4.3	Loginė visos sistemos architektūra	48
4.4	Vartotojo paslaugos	49
4.5	Veiklos paslaugos	50
4.6	Duomenų paslaugos.....	51
4.7	Sistemos elgsenos modelis.....	53
4.7.1	Veiklos taisyklės įvedimas.....	53
4.7.2	Šablono sukūrimas	57
4.7.3	Sistemos objektų būsenų kaitos diagramos	61
4.7.4	Projektavimo etapo apibendrinimas.....	64
5	FIZINĖ DUOMENŲ BAZĖ.....	65
5.1	Realizacijos modelis	68
6	SISTEMOS TESTAVIMAS.....	69
6.1	Testavimo modelis.....	69
6.2	Sistemos testavimo scenarijai	69
6.3	Sistemos testavimo protokolas.....	82
6.4	Testavimo apibendrinimas.....	83
7	IŠVADOS	85
8	LITERATŪRA	86
9	PRIEDAI.....	88

Business Rules Repository Management System

SUMMARY

In order to remain concurrent and reliable organization's information system (IS) must be adaptable to the rapidly evolving business environment. According to business rule (BR) approach all business (and information) systems are based on some kind of rules. Maintaining these rules during all their lifecycle is the key element to IS vitality and reliability. Business rules must be placed in custom repository and maintained by rule management tool in order to keep them up to date. KTU Department of Information Systems works in the area of business rules modeling and requires some engineering solutions in order to prove their theoretical results in practice. One of such solutions is business rules management system, which is the main engineering result of the thesis.

The main goal of this work is to improve the business rule specification and maintenance process by developing specialized business rule management system. Exceptional feature of the system is the possibility to specify BR using natural language-based structured templates. System's functionality also includes BR modification, versioning and deletion. Moreover, the system also enables the construction of the business rules templates as well as provides all other functions needed to maintain existing templates and other business rule repository elements.

Research includes three main sections: problem domain analysis, project specification and experimental section. Analysed business rules specification methods are presented in the analysis section; this section also includes some brief overview of the BR management systems and analyzes some technical solutions for the implementation of our system. Project specification part presents specification of the implemented system. It includes formal UML activity and sequence diagrams as well as main GUI forms and their descriptions. Section also includes logical and physical database schemas, robustness and other diagrams. Experimental section presents developed test scenarios and system testing protocol.

1 ĮVADAS

Veiklos taisyklės apibrėžimas. Iki šiol nėra vieningos nuomonės įvairiais klausimais, susijusiais su veiklos taisyklių (VT) identifikavimu, atvaizdavimu, jų integravimu IS kūrimo procese, diegimu taikomiosiose programose ir kt. Nėra ir vieningo VT apibrėžimo, notacijos bei jų klasifikacijos. Pateikiame keletą VT apibrėžimų:

- ✓ Veiklos taisyklės – tai taisyklės, nustatančios, kaip turi būti vykdoma organizacijos veikla [1].
- ✓ Veiklos taisyklė – tai teiginys, kuris apibrėžia arba sąlygoja tam tikrą veiklos aspektą. VT skirta apibrėžti veiklos struktūrai arba valdyti, įtakoti veiklos elgseną [2].
- ✓ Veiklos taisyklė – tai atominis, deklaratyviai specifikuotas, daugkartinio panaudojimo veiklos logikos fragmentas [3].

Šiame darbe veiklos taisyklei apibrėžti naudojamas toks apibrėžimas: „Veiklos taisyklė yra teiginys, kuris kontroliuoja arba įtakoja veiklos procesų vykdymą arba apibrėžia resursų, reikalingų veiklai vykdyti, struktūrą“ [4].

Aktualumas. Didėjant verslo aplinkos kitimo tempams labai svarbu, kad organizacijos informacinė sistema spėtų adaptuotis prie pakitusių veiklos sąlygų. Remiantis veiklos taisyklių ideologija, kiekviena verslo sistemą realizuojanti informacinė sistema yra grindžiama veiklos taisyklėmis, todėl užtikrinti šių taisyklių valdymą viso jų gyvavimo ciklo metu yra kritiškai svarbus organizacijos informacinės sistemos gyvybingumo ir patikimumo užtikrinimo elementas. Veiklos taisyklių valdymo funkcijoms užtikrinti yra būtina sukurti veiklos taisyklių saugyklą ir tos saugyklos valdymo sistemą (VTVS). KTU Informacijos sistemų katedroje pradėti vykdyti moksliniai tyrimai veiklos taisyklių srityje taip pat pareikalavo atitinkamų inžinerinių sprendimų, vienas kurių yra būtent veiklos taisyklių valdymo sistema.

Tyrimo sritis ir objektas. Tyrimo sritis yra veiklos taisyklių specifikuojimo ir jų valdymo procesas, tam naudojant kompiuterizuotas priemones. Tyrimo objektas yra veiklos taisyklių saugyklos valdymo sistema.

Darbo tikslas ir uždaviniai. Darbo tikslas yra pagerinti veiklos taisyklių specifikuojimo ir valdymo procesą, sukuriant specializuotą veiklos taisyklių valdymo sistemą. Sukurtosios sistemos ypatumas yra galimybė veiklos taisykles specifikuoti, naudojant natūraliai Lietuvių kalbai artimus struktūrizuotus šablonus, ir užtikrinti šių taisyklių valdymą (versijavimą, modifikavimą, šalinimą ir pan.).

Darbo tikslo įgyvendinimui išskelti šie uždaviniai:

1. Atlikti veiklos taisyklių specifikuojimo būdų analizę;

2. Ištirti pasirinktas veiklos taisyklių valdymo sistemas ir atlikti jų lyginamąją analizę, siekiant nustatyti būsimosios (autorinės) valdymo sistemos funkcionalumą;
3. Atliktos analizės pagrindu sukurti vartotojo reikalavimus atitinkančią veiklos taisyklių saugyklos valdymo sistemos projektinę specifikaciją;
4. Argumentuotai pasirinkti sistemos realizavimo priemonės ir nustatyti sistemos rizikos faktorius;
5. Realizuoti suprojektuotąją sistemą, taikant iteracinį sistemos kūrimo metodą;
6. Sukurti sistemos testavimo scenarijus bei pritaikyti juos realizuotos sistemos testavimui.

Darbo sudėtis. Darbą sudaro trys pagrindinės dalys: analitinė dalis, projektinė dalis bei eksperimentinė dalis. Analitinėje dalyje analizuojami veiklos taisyklių specifikavimo būdai, taisyklių valdymo įrankiai, bei VT valdymo sistemų realizavimo priemonės. Projektinėje darbo dalyje yra pateikiama sukurtos sistemos projektinė specifikacija, kurios pagrindą sudaro: pagrindinių funkcijų UML veiklos ir sekų diagramos ir jų aprašymai, o taip pat grafinės vartotojo sąsajos langai su aprašymais. Darbo eksperimentinėje dalyje pateikiami sukurti sistemos testavimo scenarijai ir sistemos testavimo rezultatai.

Darbo aprobavimas. Pagrindiniai darbo rezultatai yra aprašyti straipsnyje (anglų kalba), kuris pateiktas tarptautinei konferencijai „Information technologies 2008“. Straipsnio tema: „Business Rules Specification using Natural Language-based Templates: Approach and Implementation“.

2 VEIKLOS TAISYKLIŲ SAUGYKLOS VALDYMO SISTEMŲ ANALIZĖ

2.1 VT specifavimo metodai

Veiklą nusakanti informacija saugo veiklos taisykles, išreikštas šnekamąja kalba. Tam, kad taisykles būtų galima valdyti, modifikuoti bei klasifikuoti jos pagal VT manifestą turi būti[5]:

- ✓ Deklaratyvios, ne procedūrinės - taisyklės yra atsietos nuo bet kokio reikalavimo jas vykdyti. Taisyklės ir jų vykdymas yra skirtingi dalykai.
- ✓ Tikslios – veiklos taisyklė turi turėti tik vieną reikšmę. Jei taisyklė gali būti interpretuojama dvejopai ją reikia performuluoti.
- ✓ Atominės – veiklos taisyklė nusako vieną išbaigtą mintį, bet ne daugiau kaip vieną. Taisyklė turi būti nedali. Bandant susmulkinti veiklos taisyklė bus prarasta informacija.
- ✓ Suderintos – veiklos taisyklių rinkinyje neturi būti viena kitai prieštaraujančių taisyklių.
- ✓ Neperteklinės – veiklos taisyklių rinkinyje neturi būti taisyklių saugančių tą pačią informaciją.
- ✓ Orientuotos į veiklą – taisyklės suformuluotos taip, kad jas suprastų veiklos dalyviai. Taisyklės turi naudoti veiklos aplinkoje vartojamus terminus.
- ✓ Priklausančios veiklai – veiklos taisykles formuoja veiklos dalyviai ir jie yra tikrieji taisyklių valdytojai galintys jas modifikuoti ar pašalinti.
- ✓ Naudojančios kitas taisykles – veiklos taisyklės paremtos kitomis veiklos taisyklėmis.
- ✓ Norint, kad taisyklės atitiktų visus aukščiau išvardintus kriterijus jas būtina struktūrizuoti. Struktūrizavimo procesui naudojama keletas skirtingų būdų:
 - ✓ Veiklos taisyklių struktūrizavimas naudojant šablonus
 - ✓ Struktūrizavimas naudojant DSL (domain specific language) kalbas
 - ✓ VT struktūrizavimas naudojant grafines specifavimo kalbas

Yra keletas taisyklių teiginių skirstymo į grupes metodų, vienas iš jų teiginius skirsto į tokias grupes[2]:

- ✓ Veiklos termino apibrėžimo
- ✓ Faktų siejančių apibrėžimus
- ✓ Apribojimų
- ✓ Išvedimo

Savo ruožtu veiklos taisyklės pagal tą patį metodą skirstomos į tris tipus[2]:

- ✓ Struktūrinius teiginius
- ✓ Elgesio teiginius
- ✓ Derivacijos (išvedimo)

2.1.1 VT specifikavimas naudojant DSL kalbas

Dalis VT struktūrizavimui naudojamų kalbų realizuotos naudojant struktūrinės kalbas (XML) [6], kita dalis sukurta specialiai VT specifikavimui, kompanijų dirbančių VT valdymo srityje. Kiekviena kalba turi savitus tik joms būdingus VT užrašymo būdus. Norint perprasti ar suspecifikuoti VT naudojant viena iš šių kalbų būtina įsisavinti jų savitumą.

VT struktūrizavimui naudojamos kalbos:

- ✓ Quick Rules Markup Language (QRML)
- ✓ Extensible Markup Language (XML)
- ✓ Structured Rule Language (SRL)
- ✓ Business Action Language (BAL)
- ✓ Technical Rule Language (TRL)
- ✓ Drools Rule Language (DRL)

Didelė specifikavimo kalbų įvairovė apsunkina taisyklių užrašymą bei supratimą. Tuo pačiu kiekviena kalba yra specifinė pritaikyta konkrečiai problematikai spręsti, todėl dažnai gali iškilti taisyklių užrašymo problemų. Reikalingi konkrečios DSL kalbos specialistai galintys teisingai specifikuoti pateikiamas veiklos taisykles.

Taisyklės užrašius specifine kalba, o vėliau jas norint panaudoti kitoje sistemoje atsiranda nesuderinamumo problema, taisyklės tenka perrašyti sistemai suprantama kalba.

2.1.2 VT specifikavimas naudojant grafines specifikavimo kalbas

Vaizdingam VT specifikavimui naudojamos grafines specifikacijos. Grafinis veiklos taisyklių pateikimas akcentuoja funkcijas, informaciją arba objektus. Tokiu būdu specifikuotos VT aiškiai ir paprastai suvokiamos, tačiau tokie specifikavimo būdai turi savitą žymėjimą ir norint suprasti jų pagalba specifikuotas taisykles būtina žinoti naudojamų simbolių reikšmę bei tarpusavio sąveiką[7].

Grafinės VT specifikacijos[8]:

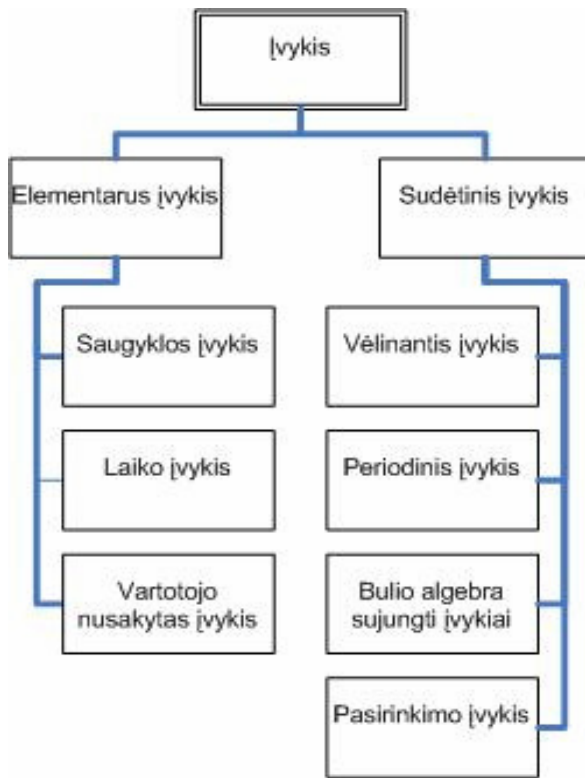
1. Į funkcijas orientuotos specifikacijos
 - ✓ Duomenų srautų diagramomis (DSD)
 - ✓ Merise struktūros pagrindu (Merise)
 - ✓ Būsenos keitimo modeliu (BKM), keitimo grafais (KG) ir būsenų lentelėmis

- ✓ Petri tinklais (PT)
2. Į informaciją orientuotos specifikacijos
 - ✓ Objektas - ryšys – modelis (ORM) ir NIAM metodas
 - ✓ Praplėstas Objektas - ryšys – modelis metodas
 - ✓ Objektų ryšių taisyklių modelis (OR)
 - ✓ Integruotos elgsenos objektų ryšių modelis (IEOR)
 - ✓ Objektų gyvenimo ciklo modelis (OGC)
 3. Į objektą orientuotos specifikacijos (OO)

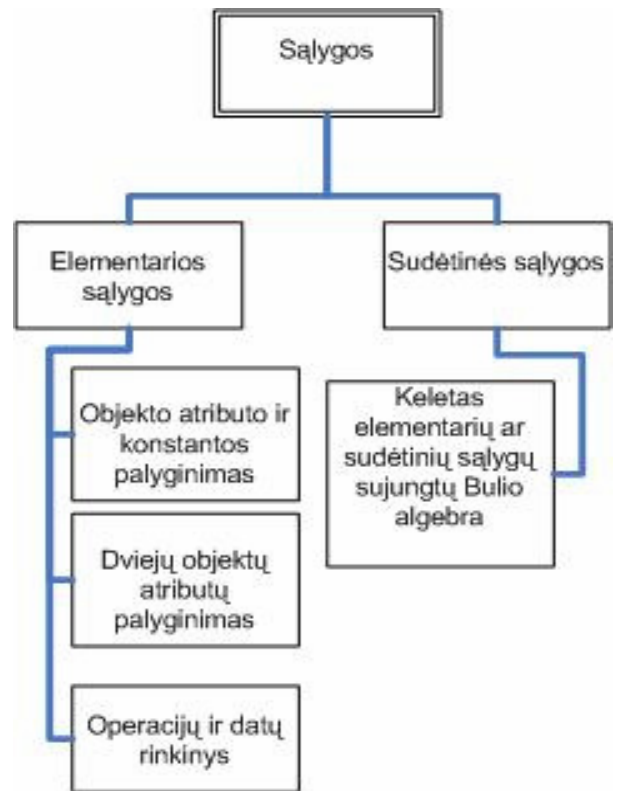
lent. 2.1 Grafinių VT specifikuojamųjų metodų palyginimas [8]

Metodas	Modeliavimo dariniai					Veiklos taisyklių tipai	
	Objektai	Taisyklės	Įvykiai	Sąlygos	Veiksmai	Nuoseklumo taisyklės	Veiklos taisyklės
DSD	Saugykla, išorinis objektas	-----	Dalinai informacijos srautais	-----	Procesai (detaliai mini specifikacijose)	Tik mini specifikacijose	Labai ribota
Merise	-----	-----	Išoriniai/vidiniai įvykiai	Sinchronizacija	Operacija	Taip (nenumatyta)	Taip
BKM	-----	-----	(būsenos keitimo pavad.)	Būsenos keitimo pavadinimais	Būsenos keitimo pavadinimais	Taip (nenumatyta)	Taip
KG	----- (apribota iki vieno objekto)	-----	-----	Būsenos keitimo pavadinimais	-----	Būsenos keitimo, dinaminės	Ne
PT	-----	-----	Skiltimis	Skiltimis	Būsenų pasikeitimai	Taip (nenumatyta)	Taip
ORM, NIAM	Objekto tipo, ryšių tipo, (atributai)	-----	-----	Kardinalumas, raktai, domenai	-----	Statinės (ribotos)	Ne
OR	Objekto tipo, ryšių tipo, (atributai)	Taisyklių tipo	Tiesioginiais lankais	Nusakomas taisyklės tipo	Tiesioginiai lankai	taip	Labai ribota
IEOR	Objekto tipo, ryšių tipo, (atributai)	-----	Skiltimis	-----	Būsenų pasikeitimai	Būsenos keitimo, statinės	Taip
OGC	Objekto tipo, (susiję su vienu objekto tipu)	-----	Įvykis (išorinis)	-----	Operacija	Būsenos keitimo (ribotos)	Labai ribota
OO Įvykių schema	- (nuorodos į objektų modelį)	Trigerio taisyklė	Įvykio tipas	Valdymo būsenos	Operacija	Taip (nenumatyta)	Taip

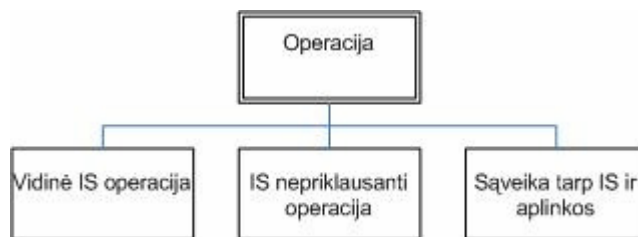
Taikant grafinius specifikuojamųjų būdus VT suskaidomos į įvykius, sąlygas ir veiksmus. Įvykiai, tai momentiniai atsitikimai vykstantys laiko erdvėje. Sąlygos nurodo kas turi būti patikrinta, o veiksmi nurodo kas turi būti padaryta. Įvykiai ir sąlygos skaidomi į elementarius ir sudėtingus, o šie dar smulkesnius (pav. 2.1, pav. 2.2). Operacijos skaidomos į vidines IS, IS nepriklausančias ir sąveikaujančias tarp IS ir aplinkos (pav. 2.3).



pav. 2.1 *Ivykių klasifikacija*



pav. 2.2 *Sąlygų klasifikacija*



pav. 2.3 *Sąlygų klasifikacija*

2.1.3 Veiklos taisyklių struktūrizavimas naudojant šablonus

Struktūrizuojant veiklos taisykles šablonų pagalba būtina išmanyti aprašinėjama veiklos sritį. Šablonai užtikrina taisyklių vienodumą, taip pat naudojant veiklai būdingą žodyną užtikrinama, kad VT turėtų vienintelę interpretaciją [9]. Yra nustatyta keletas šablonų tipų, bet visuomet gali atsirasti taisyklių neatitinkančių nė vieno šablono, tad būtina turėti galimybę susikurti savo šablonus. Šabloninis specifikavimo būdas artimiausias šnekamąja kalba išreiškiamų VT prigimčiai, tad yra lengvai suprantamas ir įsisavinamas [10].

Galima išskirti dviejų tipų šablonus pagal jų funkcionalumą:

- ✓ Šablonai užfiksuoja VT struktūrą, bei elementų reikšmes ir suteikiantys galimybę redaguoti tik keletą, VT elementų reikšmes.

- ✓ Šablonai užfiksuojantys tik VT struktūrą, o visų elementų reikšmes leidžiantys pasirinkti iš reikšmių aibės.

[vairioje literatūroje pateikiami skirtingi veiklos taisyklių šablonų tipai.

Veiklos taisyklių šablonų tipai[11]:

- ✓ Faktų
- ✓ Terminų
- ✓ Apribojimų
- ✓ Teiginių
- ✓ Nurodymų
- ✓ Apskaičiavimo
- ✓ Išvadų

Nagrinėjamoje duomenų saugykloje numatyti šablonų tipai:

- ✓ Skaičiavimo
- ✓ Išvesties
- ✓ Veiksmo
- ✓ Apribojimo
- ✓ Struktūros (faktų)

Keletas šablonų pavyzdžių pateikta lent. 2.2

lent. 2.2 Veiklos taisyklėm specifikuoti naudojamų šablonų tipai[12]








Šablono tipas	Apibrėžimas ir formalizavimas
Terminas	Žodis ar frazė susijusi su veikla
Faktas	Teiginys tvirtinantis, kad yra ryšys (atributas/apibendrinimas/grupavimas) tarp dviejų terminų. <terminas><ryšys>[of] <terminas>
Apribojimas	Teiginys nusakantis privalomą kriterijų (veiklos elgseną ar charakteristiką) kuris turi būti patenkintas veiklos objekto. <subjektas>must[not] <savybė>
Nurodymai	Teiginys specifikuojantis nebūtiną kriterijų kuris turi būti patenkintas veiklos objekto. Pažeidžiant šią taisyklę sistema tik įspėja, bet neuždraudžia transakcijos. <subjektas>should[not] <savybė>
Veiksmas, teiginys	Teiginys nusakantis sąlygą kuri pradės veiklos veiksmą. if<sąlyga>then<veiksmas>
Apskaičiavimas	Teiginys gaunantis reikšmę naudojant algoritmą.

Šablono tipas	Apibrėžimas ir formalizavimas
	<reikšmė>is computed as<algoritmas>
Išvada	Teiginys įgaunantis reikšmę naudojant vietinę išvadą ar indukciją If<sąlyga>then<faktas>

2.2 Veiklos taisyklių valdymo sistemų analizė

Taisyklių saugyklų valdymo sistemos įvairios, skirtos skirtingo tipo taisyklėms valdyti. Pagrindiniai valdymo sistemų tipai: VT saugyklų valdymo sistemos, dirbtinių intelektualių taisyklių saugyklų valdymo sistemos, įvykių apibendrinimo taisyklių saugyklų valdymo sistemos ir duomenims orientuotų taisyklių saugyklų valdymo sistemos[13].

VT saugyklų sistemoms priklauso tokie įrankiai (pav. 2.4) : ILOG, Blaze Advisor, Veisual Rules, JBoss ir kt.

Įrankis	Logo	Nuoroda
Blaze Advisor		www.fairisaac.com
ILOG		www.ilog.de
JBoss		labs.jboss.com
Mandarax		www.mandarax.org , mandarax.sourceforge.net
Oracle		www.oracle.com
QuickRules		www.yasutech.com
Versata		www.versata.com
Visual Rules		www.visual-rules.de

pav. 2.4 Įrankiai, jų logotipai ir nuorodos

2.2.1 JBoss Rules

JBoss Rules[14] įrankis yra atviro kodo priemonė naudojanti Java technologiją. Šis įrankis naudoja sprendimų priėmimo lentes greitam veiklos strategijos nustatymui. JBoss Rules yra ne vienintelis veiklos taisyklių valdymui skirtas komponentas, jis yra veiklos taisyklių tvarkymo paketo dalis.

JBoss Rules pagrįstas Java ir taisyklių kalba vadinama DROOLS[15]. Darbo aplinkai naudojamas Eclipse įrankis su DROOLS įskiepiu. Taisyklės užrašomos naudojant DROOLS kalbą, o kitos funkcijos realizuojamos Java kalboje.

Paketo komponentai:

1. JBoss Rules Workbench (Eclipse įskiepis)
2. Projektavimo aplinka Eclipse
3. Taisyklių redaktorius
4. JBoss Application serveris.

2.2.2 Mandarax

Mandarax [16] yra atviro kodo Java klasių biblioteka skirta taisyklėmis pagrįstų sistemų tvarkymui. Įrankis pagrįstas PRO LOG programavimo kalbos algoritmais. Paprastas išorinės informacijos integravimas leidžia prijungti iki 45 duomenų bazių ar duomenų failų. Pakete kurį nemokamai galima parsisiųsti iš Mandarax svetainės pateikiama keletas programų pavyzdžių.

Mandarax nėra pilna veiklos taisyklių valdymo sistema, tai tik Java klasių rinkinys skirtas taisyklių apdorojimui. Paketas neteikia taisyklių redagavimo aplinkos, tam naudojami teksto redaktoriai tokie kaip RuleML ar dalinai grafinis Oryx taisyklių redaktorius.

Paketo komponentai:

1. Licencija : Mandarax: GNU LGPL, Oryx: GPL/Commercial Licence
2. Java: JDK 1.4 (Mandarax 3.4)
3. Klasių biblioteka

2.2.3 Oracle Business Rules

Oracle Business Rules [17] yra priemonė suteikianti galimybę lengvai panaudoti veiklos taisykles Java ar į servisus orientuotoje aplinkoje. Oracle RuleAuthors pagalba suteikia galimybę peržiūrėti ar pakeisti veiklos taisykles atskirai nuo taikomosios programos.

Taisyklės RuleAuthor pateikiamos naudojant IF – THEN darinius. Sukurtos taisyklės konvertuojamos į Oracle naudojamą RL taisyklių užrašymo kalbą ir patalpinamos VT saugykloje. Norint redaguoti taisykles būtina tai daryti Java kode, tiesioginiai pakeitimai VT

saugykloje negalimi. Saugykloje užtikrinamas taisyklių versijavimas tam, kad būtų galima peržvelgti taisyklės atsiradimo istoriją ar grįžti prie ankstesnės versijos.

Be RuleAuthor Oracle dar pateikia išplėstą redaktorių JDeveloper kaip taisyklių projektavimo priemonę.

2.2.4 Visual Rules

Veiklos taisyklių valdymo priemonė Visual Rules [18] suteikia galimybę administruoti, dokumentuoti, ir valdyti veiklos taisykles. Pagrindinės savybės:

1. Grafinis taisyklių redaktorius
2. Testavimas ir taisyklių derinimas su aiškiu klaidų išryškinimu
3. Stebėjimo ir statistikos elementai analizei atlikti
4. Automatinis taisyklių modelio dokumentacijos generavimas įvairiais formatais
5. Kodo generatoriai įvairioms platformoms ir programavimo kalboms
6. Visų J2EETM aplikacijų serverių palaikymas
7. Prisijungimas prie duomenų bazės su veiksmiais ir funkcijomis skirtomis informacijos apdorojimui RDMS tipo sistemose.

Produktą galima įdiegti į Eclipse, SAP NetWeaver™ ar IBM WebSphere studio Workbench.

2.2.5 Quick Rules

Quick Rules yra YASU Technologies [19] kompanijos produktas skirtas VT valdymui. Sistema turi dvi realizacijas: .NET ir J2EE platformose. Produkte naudojama specifinė verslo taisyklių kalba QRML, o taisyklės saugomos XML formatu. Pagrindinės funkcijos: versijų kontrolė, sistemos naudotojui patogi taisyklių užrašymo kalba, taisyklių konfliktų sprendimas, taisyklių apibrėžimas naudojant XML, priėjimo ir versijų kontrolė, bei sprendimų lentelės.

Paketo komponentai:

- ✓ Quick Rules Business Rule Engine
- ✓ Quick Rules Builder
- ✓ Quick Rules Web Editor
- ✓ Rule Editor API

2.2.6 ILOG Rules

ILOG [20] kompanijos produktai ILOG Rules, ILOG JRules ir ILOG for .NET sukurti naudojant skirtingas platformas atitinkamai C++, Java ir .NET. ILOG produktai naudoja ECA tipo taisyklių aprašymą. Galima naudoti ir BAL (Business), TRL (Technical Rule Language) ar IF-THEN-ELSE formuluotės sakinius veiklos taisyklių formalizavimui. Pagrindinės

sistemos funkcijos: VT versijų kontrolė, taisyklių specialistui suprantama ir patogi VT formavimo kalba ir konfliktų sprendimo įrankis.

Paketo komponentai:

- ✓ Rule Builder
- ✓ Business Rule Repository
- ✓ Business Rule Editor
- ✓ Business Rule Management

2.2.7 Versata Logic Suite

Versata Inc. [21] produktas Versata 6 BRMS. Sistema apjungia veiklos taisyklių kūrimo ir diegimo įrankius. Versata produktas naudoja Eclipse 3.0 aplinką kuri artima Java pagrįstų įrankių aplinkai. Sistemos funkcijos: VT specifikavimas, transakcijų ir uždavinių sekų nustatymas, progreso sekimas, išimčių nustatymas.

Paketo komponentai:

- ✓ Versata Process Logic Designer
- ✓ Versata Process Logic Engine
- ✓ Versata Workbench
- ✓ Versata Business Rules Engine

2.2.8 Blaze Advisor

Fair Isaac Corporation [22] produktas Blaze Advisor sukurtas Java pagrindu. Sistema naudoja struktūrinę taisyklių užrašymo kalbą (Structured Rule Language) SRL. Suteikiama galimybė naudoti šablonus suteikiančios taisyklėms bendras struktūras. Pagrindinės produkto funkcijos: VT versijų kontrolė ir vartotojų autorizacija, suderinamumas su išoriniais duomenų šaltiniais (XML, COBRA & COM), taisyklių valdymas nenaudojant programavimo, sudėtingų daugiažingsnių taisyklių kūrimas, sprendimo lentelės, bei taisyklių grupavimas ir testavimas.

Paketo komponentai:

- ✓ Blaze Advisor Integrated Development Environment
- ✓ Blaze Advisor Innovator
- ✓ Rule Repository
- ✓ Blaze Advisor Rule Engine

2.3 Veiklos taisyklių valdymo sistemų palyginimas

VT valdymo sistemas galima vertinti pagal dviejų tipų kriterijus:

- ✓ Nepriklausomi nuo programos tipo
- ✓ Priklausomi nuo programos tipo

Vieni kriterijai naudojami visų tipų programoms palyginti, nes jie atskleidžia sistemos kainos, greičio, našumo ir panašias kiekvienai programai būdingas savybes.

Kiti tuo tarpu yra naudojami tik tam tikro tipo programoms palyginti ir yra sufokusuoti į pasirinkto tipo programų svarbiausias funkcijas. Aprašytų sistemų palyginimo lentelėje (lent. 2.3 *Veiklos taisyklių sistemų palyginimas*[23]) pateikiamas jų palyginimas.

lent. 2.3 *Veiklos taisyklių sistemų palyginimas*[23]

Kriterijai nepriklausomi nuo programos							
	Blaze Advisor	ILOG	JBOSS Rules	Mandarax	Oracle	QuickRules	Visual Rules
Licencijos/kainos	0	4	5	5	2	1	3
Vykdyimo aplinka	5	5	5	5	4	5	5
Našumas	5	5	4	3	4	4	5
Išteklių poreikis	3	5	4	4	4	5	4
Praktiškumas	4	3	3	3	4	4	4
Gebėjimas prisitaikyti	0	4	2	3	2	2	0
Kriterijai orientuoti į programą							
Taisyklių pateiktis	4	4	3	3	4	3	5
Taisyklių apdorojimas	4	4	4	4	4	4	4
Faktinė bazė	5	5	5	4	5	5	3
Pagalba kuriant	4	3	3	2	3	5	3
Taisyklių diegimas	5	4	2	1	3	4	4
Saugumas	5	5	2	1	3	4	4
Taškų suma	43	53	42	38	42	44	45

Dalis veiklos taisyklių valdymo sistemų sukurtos kaip Eclipse įrankio įskiepai. Šio tipo programos dažniausiai yra nemokamos ir suteikia ribotas galimybes VT specifikuojimo srityje. Patogu, bet tik Java kalbą išmanančiam vartotojui suprantamą VT užrašymo būdą pateikia įrankis JBoss Rules. Jis naudoja Java programavimo kalbą, bei VT specifikuojimo kalbą DROOLS IF-THEN tipo taisyklėms suformuoti. Deja, toks VT užrašymo būdas programavimo kalbos neperpratusiam vartotojui gali būti sudėtingas ir keliantis papildomų problemų.

Grafinę VT specifikuojimo kalbą naudojantis įrankis Visual Rules suteikia galimybę specifikuoti VT sudėliojant ją iš grafinių elementų. Tokiu būdu suformuotai VT nustačius kiekvieno komponento parametrus sukuriama tekstinis atitikmuo. Šio įrankio teigiama savybė ta, kad vizualizuotas VT kūrimas yra išraiškesnis ir suteikia daugiau aiškumo. Bet tokiu būdu specifikuojant VT užtrunkama daugiau laiko, be to toks VT fiksavimo būdas daug artimesnis sistemos analitikui nei srities specialistui.

Blaze Advisor įrankis naudoja VT sprendimams priimti. Veiklos taisyklių specifikuojimo įrankio pagalba atliekama keliais būdais. Taisyklės gali būti sugeneruotos kuriant sprendimų lenteles ar sprendimų medžius. VT jungiamos į rinkinius yra galimybė nustatyti VT prioritetus. Atskirų taisyklių užrašymas komplikuoatas ir reikalauja programavimo patirties, taip pat specifinės įrankio naudojamos kalbos išmanymo. Šio įrankio licencijų kaina yra pati didžiausia iš nagrinėjamų, tai mažina jo vartotojų ratą.

Dauguma nagrinėtų įrankių naudoja specialias kalbas VT specifikuojimui. Norint naudoti vieną iš šių įrankių būtina perprasti naudojamos kalbos specifika, dalis šių kalbų reikalauja programavimo patirties ar bent programavimo pagrindų. Norint supaprastinti VT specifikuojimą būtina naudoti vartotojui lengvai perprantamą VT surinkimo būdą. Natūralios kalbos sakiniams artimiausias VT specifikuojimo būdas yra VT struktūrizavimas naudojant šablonus. Tokiu būdu specifikuojant VT vartotojas gali natūralia kalba užrašytas taisykles įvesti į sistemą pagal iš anksto nustatytą struktūrą. Kadangi taisyklių įvairovė labai didelė būtina suteikti vartotojui galimybę šablonus sudaryti iš nustatytos elementų aibės.

2.4 Sistemos architektūros ir įgyvendinimo priemonių analizė

VT saugyklų valdymo sistemos realizacijai naudojama serverio – kliento tipo architektūra. Tokia sistema suteikia galimybę valdyti VT saugyklą iš nutolusio kompiuterio. Šis valdymo būdas reikalingas tam, kad bazė būtų nuolatos atnaujinama ir ją galėtų valdyti keletas skirtingų asmenų.

Duomenų saugykla turi užtikrinti galimybę eksportuoti bei importuoti duomenų bazės modelį. Taip pat turi būti galimybė prisijungti vartotojams su įvairaus lygio teisėmis, kas užtikrintų duomenų apsaugą.

VT saugyklų realizavimui naudojamos duomenų saugyklos:

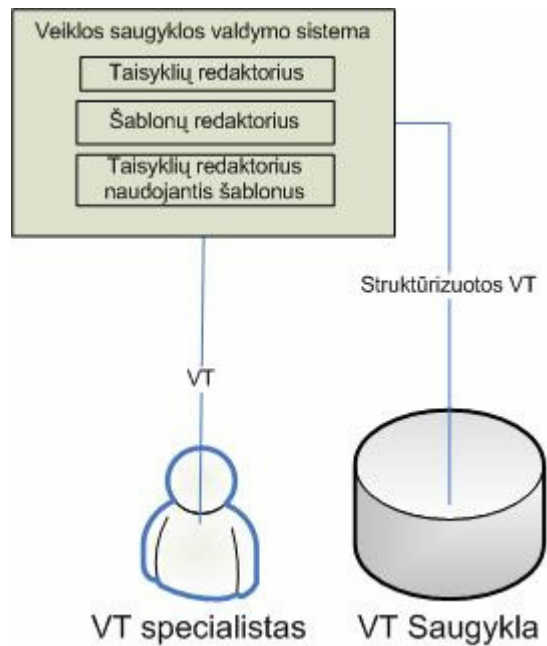
- ✓ Microsoft SQL Server
- ✓ MySQL Server
- ✓ Sybase SQL Server
- ✓ Specialiai atskirom sistemoms sukurtos DB

Kalbos naudojamos VT saugyklų valdymo sistemoms kurti:

- ✓ Visual Basic
- ✓ C++/ C
- ✓ Java
- ✓ PHP
- ✓ ASP

Sistema naudojama duomenų bazei esančiai lokaliame Microsoft SQL Serveryje valdyti. Programa instaliuojama vartotojo kompiuteryje ir yra Windows aplikacija bendraujanti su serveriu. Siunčianti bei gaunanti iš saugyklos struktūrizuotas ir nestruktūrizuotas VT, jų šablonus, papildomą informaciją. Bendravimui su serveriu naudojamos SQL užklauskos. Į veiklos taisyklių valdymo sistemos apibendrintą architektūrą (pav. 2.5) įeina VT specialistas, jo naudojama VTVS, bei valdoma taisyklių saugykla.

Windows aplikacijai realizuoti naudojama Visual Basic kalba. Ji pasirinkta, todėl, kad turi patogų aplikacijų kūrimo įrankį Visual Studio. Šio įrankio pagalba greit ir patogiai sudaromi duomenų rinkiniai vėliau naudojami programai realizuoti. Duomenų rinkinių naudojimas sumažina duomenų bazės apkrovimą ir tuo pat metu paspartina programinės įrangos veikimą.



pav. 2.5 Apibendrinta numatomos VTS valdymo sistemos architektūra

2.5 Siekiamos sistemos apibrėžimas

Siekama sukurti lanksčią sistemą įgalinančią VT specialistą valdyti veiklos taisyklių saugyklą. VT valdymo priemonė suteikia galimybę formalizuoti VT šablonų pagalba, jas redaguoti, peržiūrėti ir atnaujinti. Tam, kad būtų įmanoma sustruktūrizuoti visų tipų VT vartotojui suteikiama galimybė kurti savo šablonus naudojant VT ar tiesiog sudėliojant šabloną iš elementų. Kadangi sistema kuriama specifinės duomenų bazės valdymui, tai jai būdingos savitos funkcijos.

Sistema diegiama vartotojo kompiuteryje, o struktūrizuotos VT ir šablonai, bei papildoma informacija apie juos naudojant SQL užklausas įterpiama ar išgaunama iš saugyklos esančios lokaliame serveryje. Sistemos struktūrą (pav. 2.6) sudaro vartotojo kompiuteris su įdiegta VTVS, taisyklių saugyklos serveris, bei jame saugoma VTS.



pav. 2.6 Sistemos struktūrinė diagrama

2.6 Darbo tikslas ir siejami privalumai

Darbo tikslas yra sumodeliuoti ir sukurti veiklos taisyklių saugyklos valdymo sistemą, kuri įgalintų saugyklos vartotojus naudotis saugyklos teikiamais privalumais.

Verslo taisyklių saugojimo VT saugykloje privalumai [24]:

- ✓ VT saugomos atskirai nuo projektuojamos IS
- ✓ Visos VT saugomos vienoje vietoje
- ✓ Jau įvestų VT panaudojimas naujų kūrimui
- ✓ VT formalizavimas ir dviprasmiškumų vengimas
- ✓ VT grupavimas ir prieštaravimų išvengimas
- ✓ Patogus ir greitas taisyklių redagavimas

Programos pagrindiniai privalumai:

- ✓ Intuityvi vartotojo sąsaja
- ✓ VT formalizavimas šablonų pagalba, grupavimas, modifikavimas
- ✓ Šablonų kūrimo, modifikavimo ir naudojimo galimybės
- ✓ Dalinai automatizuotas VT surinkimas

2.7 Kompiuterizuojamos sistemos funkcijos

1. Klasifikatorių kūrimas

2. Veiklos taisyklių valdymas:

- ✓ Kūrimas (naudojant / nenaudojant šablonus)
- ✓ Modifikavimas

- ✓ Peržiūrėjimas
- ✓ Naikinimas
- ✓ Grupavimas

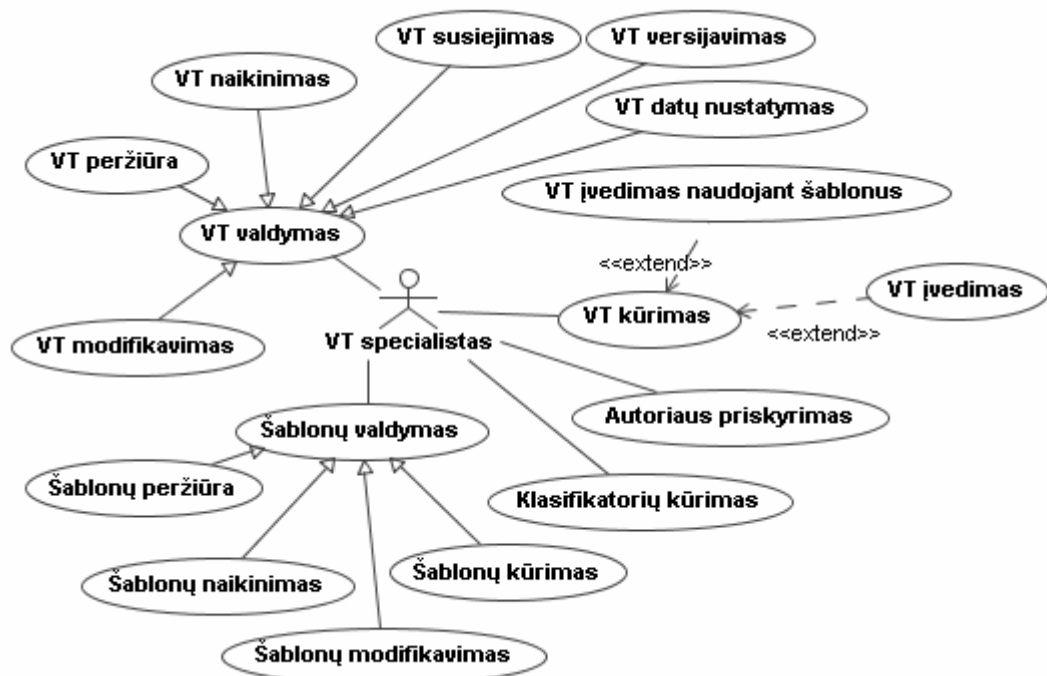
3. Šablonų valdymas:

- ✓ Kūrimas
- ✓ Peržiūrėjimas
- ✓ Naudojimas
- ✓ Modifikavimas
- ✓ Naikinimas
- ✓ Grupavimas

4. Papildomos funkcijos:

- ✓ VT versijavimas
- ✓ VT įvedimo/galiojimo datos nustatymas
- ✓ Autoriaus priskyrimas
- ✓ Nestruktūrizuotos ir struktūrizuotos VT susiejimas

Papildomos funkcijos naudingos prižiūrint veiklos taisyklių vystymosi procesą, VT kaitą vykdant nuolatinį VTS atnaujinimą. Visos veiklos taisyklių valdymo sistemos funkcijos (pav. 2.7) suteikia vartotojui galimybę pilnai valdyti VTS.

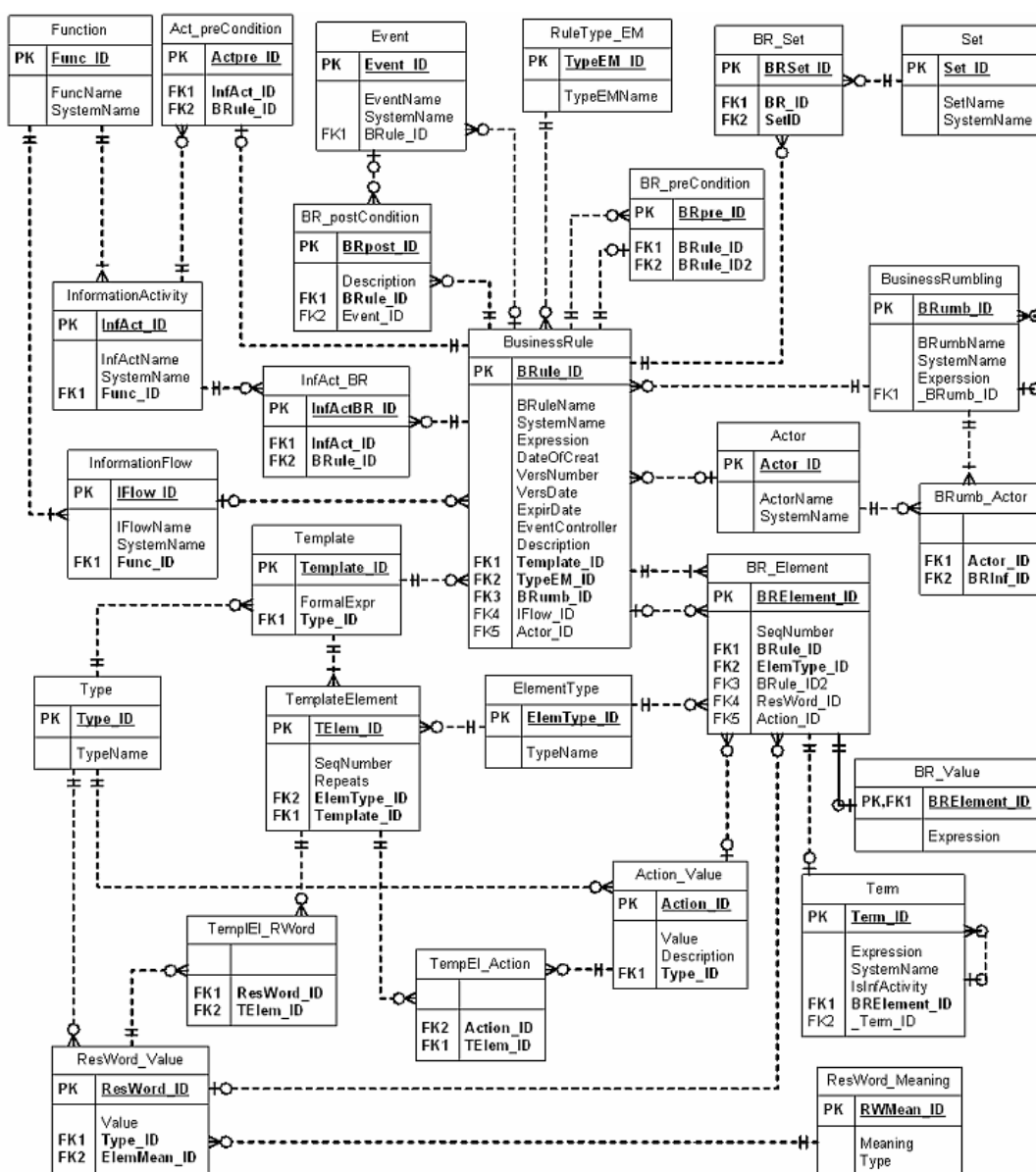


pav. 2.7 VT saugyklos valdymo sistemos panaudojimo atvejų modelis

2.8 Reikalavimai duomenims

Valdymo sistema kuriama suprojektuotai VT saugykloi kurioje realizuota galimybė saugoti VT ir jų šablonus. Saugykloje taip pat saugomi struktūrizuotų VT elementai ir nestruktūrizuotos VT. Duomenys skirti VT valdymo sistemai yra veiklos taisyklės kurias sudaro žodžiai skaičiai, bei simboliai. Įvedant taisyklės pagal šablonus, žodžiai sakiniuose išdėstomi šablono nustatyta tvarka. Sakinių logiką tikrina taisyklės įvedantis vartotojas. Modelio naudojamos VT saugyklos loginė duomenų bazės schema pateikta pav. 2.8

Šablonai kaip ir taisyklės išskaidomi į elementus ir siejami su taisyklėmis sudarytomis jų pagalba. Duomenyse taip pat atsispindi VT bei šablonų grupės. Veiklos taisyklių elementų reikšmės apriboja VT šablonų tipai.



pav. 2.8 Veiklos taisyklių saugyklos loginė duomenų bazės schema [26]

2.9 Analizės išvados

1 Išnagrinėjus KTU Informacijos sistemų katedroje naudojamo IS kūrimo metodo principus nuspręsta, kad reikalinga veiklos taisyklių saugyklos valdymo sistema skirta suprojektuotai VT saugyklai valdyti.

2 Ištyrus verslo taisyklių specifikavimo metodus, taisyklių kūrimui pasirinktas šablonus naudojantis VT specifikavimo metodas. Suformuluotas siekiamos sukurti sistemos apibrėžimas, parinktos realizavimo priemonės.

3 Išanalizavus taisyklių valdymo sistemas nustatyti jų privalumai bei trūkumai, išryškintos pagrindinės jų teikiamos funkcijos. Įvertinus svarbiausias sistemų teikiamas funkcijas, bei jas praplėtus papildomomis suformuluotos būsimos sistemos funkcijos.

4 Išsiaiškinus valdymo sistemų naudojamą architektūrą, technologijas ir saugyklų įvairovę suprojektuota sistemos architektūra ir realizavimui parinktas VisualStudio įrankis, bei Microsoft SQL serveris.

5 Įvertinus veiklos taisyklių saugyklos valdymo sistemai būdingas savybes ir jų svarbą sudaryti realizuojamos sistemos vertinimo kriterijai, nustatyti nefunkciniai reikalavimai.

3 VEIKLOS TAISYKLIŲ SAUGYKLOS REIKALAVIMŲ SPECIFIKACIJA

Išanalizavus projektuojamos sistemos aplinką, būsimus jos vartotojus, bei analogiškų sistemų teikiamas funkcijas pradėtas reikalavimų specifikavimo bei analizės etapas. Reikalavimų specifikavimui pasirinkta UML (Unified Modeling Language, Vieninga modeliavimo kalba) ir ją naudojantis MagicDraw 11.5 versijos modeliavimo paketas. Specifikavimo metu nustatytos pagrindinių funkcijų veiksmų sekos. Sudarytos vartotojo ir sistemos bendravimo sekų diagramos.

3.1 Nefunkciniai reikalavimai ir apribojimai

Nefunkciniai reikalavimai apibrėžia reikiamą sistemos patikimumo ir parengtumo lygį. Jie apima standartus kurių laikomasi kuriant sistemą, taip pat apibrėžia veikimą bei suderinamumą su kitomis sistemomis. Nefunkciniuose reikalavimuose apibrėžiami sistemos rizikos faktoriai ,bei rezultatų kokybės kriterijai.

3.1.1 Reikalavimai standartams

VT saugyklos valdymo sistema remiasi Object Management Group sukurtu veiklos žodyno ir veiklos taisyklių (SBVR) standartu[27]. Verslo taisyklės nagrinėjamos vadovaujantis GUIDE VT projekto dokumentacija - viena iš pagrindinių VT standartų[2].

3.1.2 Reikalavimai veikimui

VT saugyklos valdymas turi būti stabilus tam, kad netikėtai sutrikus programos veikimui saugyklos informacija nebūtų sugadinta. Stabilumo stoka taip pat neigiamai veikia sistemos vartotojo norą ja naudotis. Kadangi sistema skirta VTS valdymui, tai jos stabilumas priklauso ir nuo serverio kuriame patalpinta VTS stabilumo, bei ryšio su serveriu patikimumo.

3.1.3 Suderinamumas su operacinėmis, kitomis sistemomis

Valdymo sistema turi būti suderinta su Windows operacine sistema. Realizuota autentifikuoto prisijungimo prie lokalaus Microsoft SQL Server galimybė.

Sistema kuriama suprojektuotai VT saugykloi, tad turi būti realizuota taip, kad turėtų visas funkcijas reikalingas VT saugyklos valdymui, bei integruotų savitą saugyklos logiką.

3.2 Sistemos rizikos faktoriai

3.2.1 Sistemos neatitikimas vartotojo reikalavimams:

Retas konsultavimasis su produkto užsakovais, netinkamas srities išmanymas, per mažas įsigilinimas į kompiuterizuojamą sritį, visa tai sąlygoja klaidingą problemos sprendimo parinkimą. Net jei sistema ir išpildys pagrindinius vartotojo reikalavimus ji gali būti atmesta kaip netinkama jei vartotojas nuspręs, kad ji netenkina visų jo reikalavimų. Norint išvengti tokių pasekmių būtina nuolat bendrauti su vartotoju ir pateikti dalines sistemos funkcijas, tobulinti vartotojo sąsają.

3.2.2 Sistemos stabilumo stoka:

Sistemai tobulėjant ir plečiantis gali atsirasti sistemos modulių suderinamumo problemos sąlygojančios sistemos nestabilumą. Atsiradęs nestabilumas gali padaryti sistemą visiškai netinkamą naudojimui. Norint to išvengti būtina naudoti aiškiai nusakytus realizavimo metodus. Metodai tarpusavyje turi būti suderinti, o jų veikimo korektiškumas patikrintas testavimo metu.

3.2.3 Sistemos klaidingas veikimas:

Net programai veikiant stabiliai gali atsirasti programavimo klaidų kurios įtakos programos gaunamų rezultatų teisingumą. Kad taip neatsitiktų būtina nuolat testuoti tarpinius duomenis ir rezultatus, bei pasitelkus kitus, ne sistemoje apskaičiuotus rezultatus patikrinti sistemos rezultatų teisingumą. Privalomas nuolatinis pavyzdinių duomenų įvedimas bei gautų rezultatų korektiškumo patikrinimas.

3.3 Rezultato kokybės kriterijai

3.3.1 Vartotojo reikalavimų išpildymas:

Nustatytų vartotojo reikalavimų išpildymas būtina sąlyga nusakanti vartotojo norų ir sukurtos sistemos atitikimo laipsnį. Siekiant išgauti tikslius vartotojo reikalavimus būtina sudaryti vartotojo bei sistemos bendravimo sekų diagramas. Diagramos sudaromos bendraujant su vartotojais ir fiksuojant jų norus bei pageidavimus. Sistema gali tapti nenaudojama neišpildžius vartotojo reikalavimų.

3.3.2 Paprastumas:

Paprastumas nusako vartotojo ir programos bendravimo lengvumą. Reikiamų komponentų išdėstymas vartotojui patogiose vietose, jų paskirties aiškumas didina programos naudojimo paprastumą. Taip pat komponentų savalaikis pateikimas analogiškas vartotojui

įprastai darbo tĕkmei daug greiĉiau suprantamas ir įsisavinamas. Funkcijų vykdymo tvarka turi bŭti nekomplikuota ir lengvai įsisavinama.

3.3.3 Veikimo greitis:

Veikimo greitis tai kiekybinis programos darbo laiko matas. Jis nusako kaip greit sistema apdoroja duomenis, pateikia rezultatus. Veikimo greitis priklauso nuo sistemos sudĕtingumo, todĕl įvertinant veikimo greitį būtina lyginti analogiškas sistemas. Kuriamos sistemos funkcijų vykdymo greitis turi bŭti toks, kad neverstų vartotojo laukti ir dvejoti dĕl galimo sistemos gedimo.

3.4 Funkcinių reikalavimų specifikacija

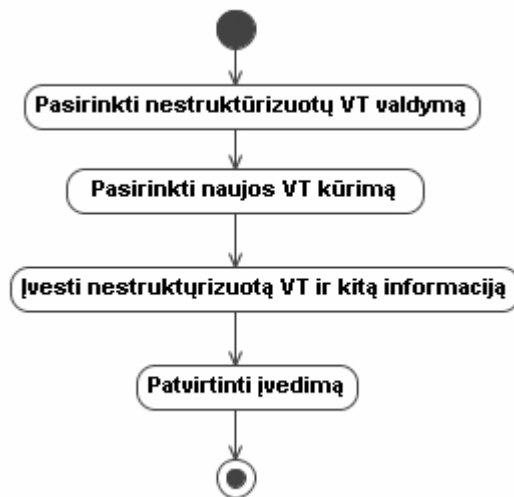
Specifikuojant reikalavimus kiekvienam sistemos panaudojimo atvejui sudaromos funkcijų atlikimo sekų bei specifikacijų diagramos. Jose atskleidžiama funkcijų atlikimo tvarka bei apribojimai. Sekų diagramos vaizduoja vartotojo sąveiką su veiklos taisyklių valdymo sistema. Diagramose vaizduojami tik pagrindiniai veiksmai. Sekų diagramos suteikia galimybę suprasti vartotojo atliekamų veiksmų seką ir numatyti elementų pateikimo tvarką.

3.4.1 Nestruktūrizuotos veiklos taisyklės įvedimo funkcija

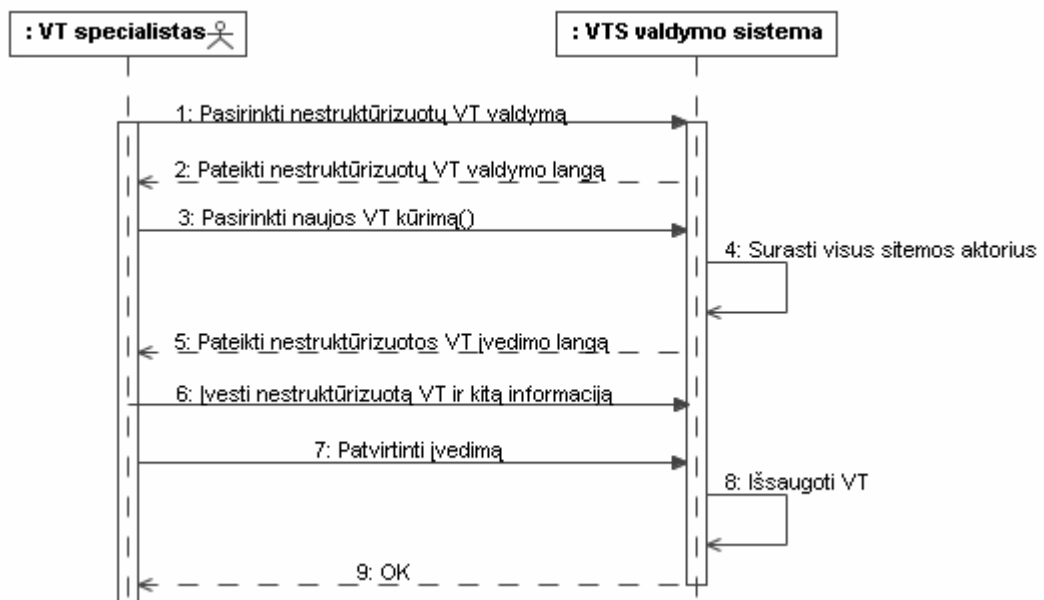
Nestruktūrizuotos VT įvedimas skirtas nestruktūrizuotų, natūralia kalba užrašytų taisyklių fiksavimui. Įvedant nestruktūrizuotas VT taip pat fiksuojama papildoma informacija (pavadinimas, sisteminis vardas, apibūdinimas ir kt.)

Nestruktūrizuotos VT įvedimas (pav. 3.1, pav. 3.2) susideda iš:

- ✓ Nestruktūrizuotų VT valdymo pasirinkimas – nestruktūrizuotų VT įvedimas pradedamas VT valdymo pasirinkimu. Pasirinkus VT valdymą pateikiamas langas atvaizduojantis VTS esanĉias nestruktūrizuotas VT.
- ✓ Naujos VT kūrimo pasirinkimas – atvertame nestruktūrizuotų VT valdymo lange pasirenkamas naujos VT įvedimas ir pateikiamas VT informacijos įvedimo langas, kuriame pateikiamas VTS esanĉių aktorių sąrašas bei laukai reikalingi VT įvesti.
- ✓ Nestruktūrizuotos VT ir papildomos informacijos įvedimas – naujos VT įvedimo metu natūralia kalba įvedama nestruktūrizuota VT, jos apibūdinimas bei papildoma nestruktūrizuotai VT būdinga informacija.
- ✓ Įvedimo patvirtinimas – įvedus visą reikiamą informaciją ir įsitikinus jos teisingumu VT įvedimas patvirtinamas. Po patvirtinimo sistemoje sukaupta informacija išsaugoma VTS.



pav. 3.1 Nestructūrizuotos VT įvedimo specifikacija



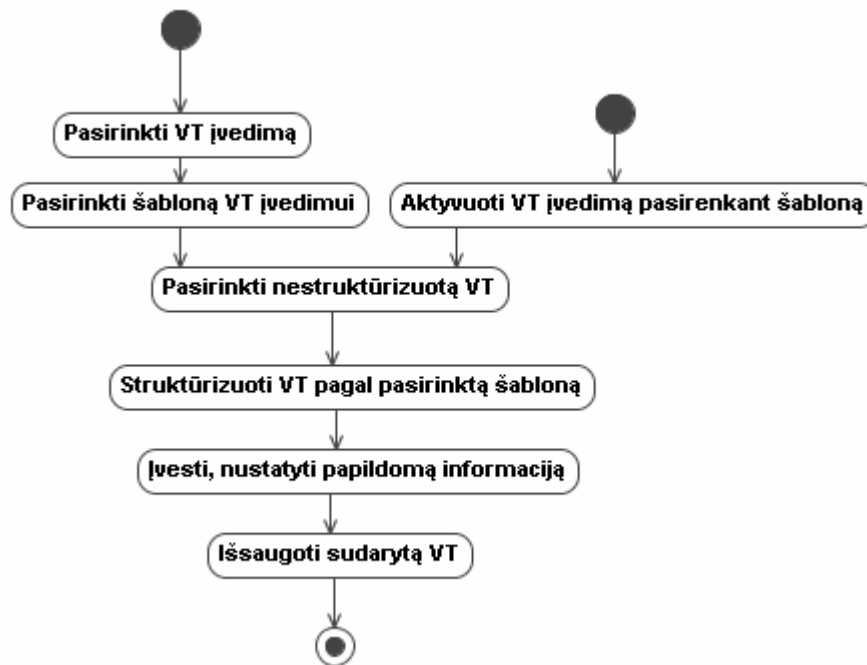
pav. 3.2 Nestructūrizuotos VT įvedimo sekų diagrama

3.4.2 Struktūrizuotos veiklos taisyklės įvedimo funkcija

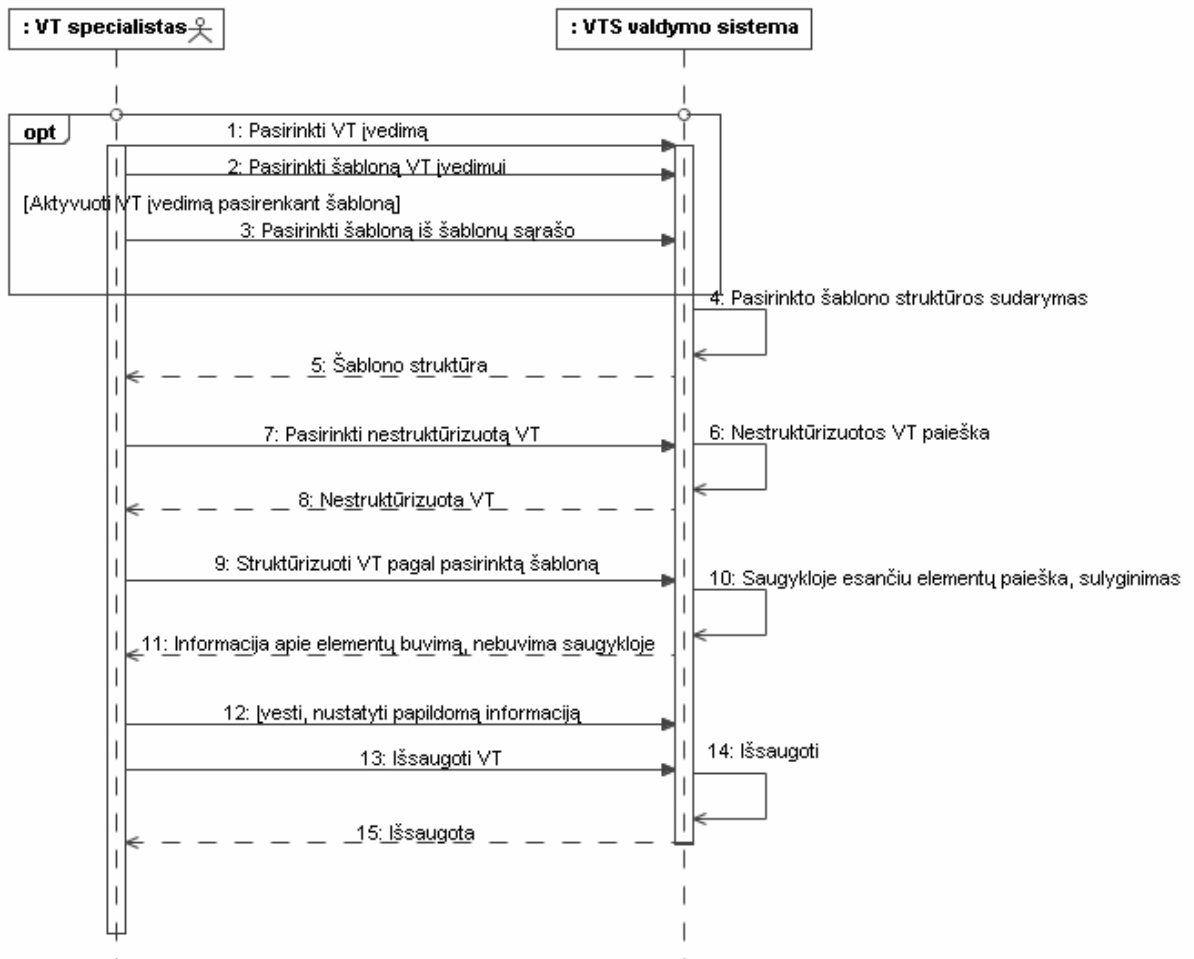
Struktūrizuotos VT įvedimas naudoja sistemoje esančius šablonus. Tokiu būdu įvedinėjant VT reikia pasirinkti šabloną kurį naudosime taisyklės struktūrizavimui. Pasirinkta šablono struktūra apsaugo vartotoją nuo neteisingo įvedimo, taip pat pateikia sistemoje esančių taisyklių elementų reikšmes, kurias galima pasirinkti įvedus vos kelis taisyklės elemento simbolius. Sistemoje neesant įvedinėjamam elementui suteikiama galimybė jį įvesti. Struktūrizuotos VT įvedimas (pav. 3.3, pav. 3.4) susideda iš:

- ✓ VT struktūrizavimo proceso atyvavimo būdai:
 - Iš meniu pasirinkus VT struktūrizavimą, ir pasirinkus šabloną iš sistemos pateikto šablonų sąrašo
 - Aktyvavus VT struktūrizavimą pasirenkant šabloną

- ✓ Nestruktūrizuotos VT pasirinkimas – nestruktūrizuota VT pasirenkama iš sistemos pateikto VTS saugomų nestruktūrizuotų VT sąrašo.
- ✓ VT struktūrizavimas pagal pasirinktą šabloną – VT struktūrizavimo proceso metu pasirinkta nestruktūrizuota VT informacija struktūrizuojama pasirinkto šablono pagalba. Struktūrizavimo proceso metu pateikiamos VTS esančių elementų reikšmės. Struktūrizavimas vykdomas tol kol pilnai išpildoma šablono struktūra.
- ✓ Papildomos informacijos įvedimas, nustatymas – struktūrizuotai VT nustatoma papildoma informacija (galiojimo data, apibūdinimas, sisteminis vardas ir kt.)
- ✓ Sudarytos VT išsaugojimas – įsitikinus, kad visa proceso metu įvesta ar nustatyta informacija yra teisinga VT struktūrizavimas baigiamas išsaugojant sudarytą VT saugykloje.



pav. 3.3 Struktūrizuotos VT įvedimo specifikacija



pav. 3.4 Struktūrizuotos VT įvedimo sekų diagrama

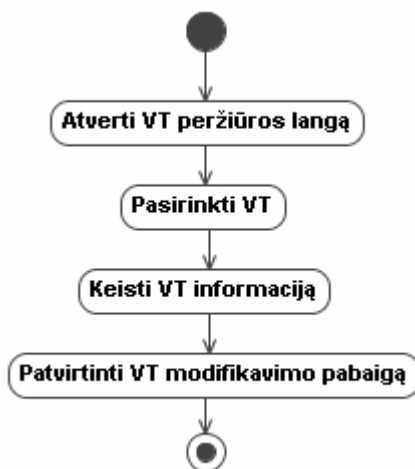
3.4.3 Struktūrizuotos VT modifikavimo funkcija

Atliekant struktūrizuotos VT modifikavimą leidžiama keisti tik elementų reikšmes, bet ne jų kiekį ar išdėstymo tvarką. Elementų kiekio, bei išdėstymo tvarkos koregavimas neleidžiamas, siekiant išlaikyti VT struktūrą susietą su VT priskirtu šablonu. Šio proceso metu taip pat galima koreguoti papildomą VT informaciją.

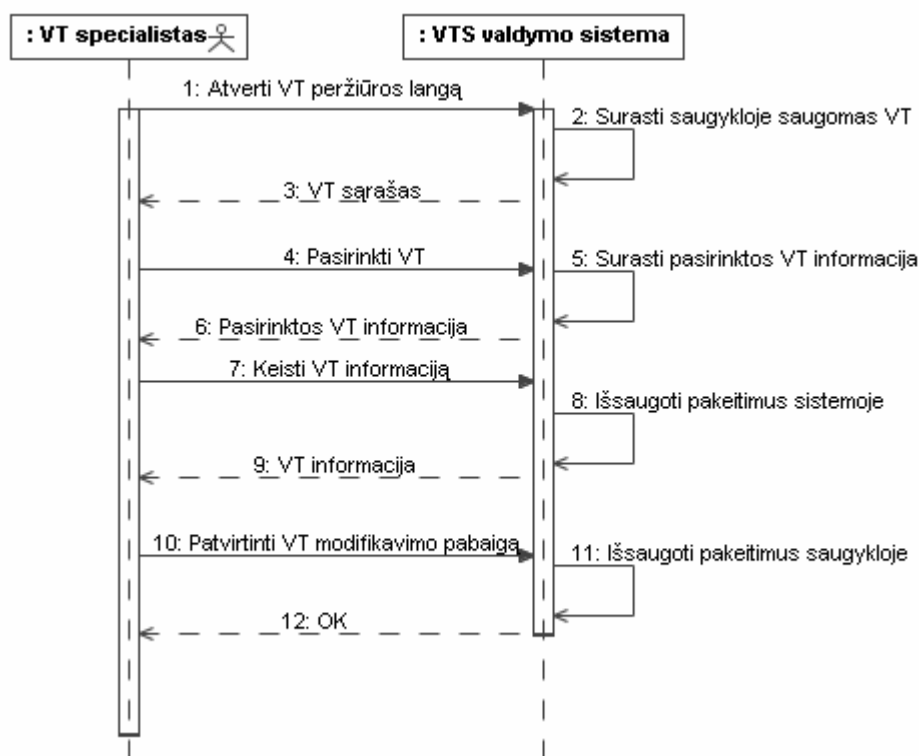
Struktūrizuotos VT modifikavimo procesas (pav. 3.5, pav. 3.6) susideda iš:

- ✓ VT peržiūros lango atvėrimas – norint pradėti VT modifikavimą, būtina pasirinkti struktūrizuotą VT esančią VTS. VTS esančių VT sąrašas pateikiamas atvėrus VT peržiūros langą.
- ✓ VT pasirinkimas – modifikuojama VT pasirenkama iš sistemos pateikto VTS esančių VT sąrašo. Sistemos pateiktas VT sąrašas gali kisti priklausomai nuo nustatytų VT paieškos parametrų.
- ✓ VT informacijos keitimas – pasirinkus VT atliekamas norimos VT informacijos koregavimas. Sistema leidžia keisti struktūrizuotos VT elementų reikšmes, susijusią nestruktūrizuotą VT bei papildomą informaciją.

- ✓ VT modifikavimo pabaigos patvirtinimas - atlikus visus norimus pakeitimus ir įsitikinus jų teisingumu struktūrizuotos VT modifikavimas užbaigiamas proceso pabaigos patvirtinimu ir VT išsaugojimu VTS.



pav. 3.5 VT modifikavimo specifikacija



pav. 3.6 VT modifikavimo sekų diagrama

3.4.4 Veiklos taisyklių peržiūrėjimo funkcija

Sistemoje esančių taisyklių peržiūrėjimas yra nuolatos vartotojo naudojama funkcija. Prieš atliekant naujų VT įvedimą ar struktūrizavimą būtina įsitikinti, kad sistemoje dar nėra taisyklės kuria norima papildyti sistemą. Taip pat prieš atliekant VT modifikavimą būtina

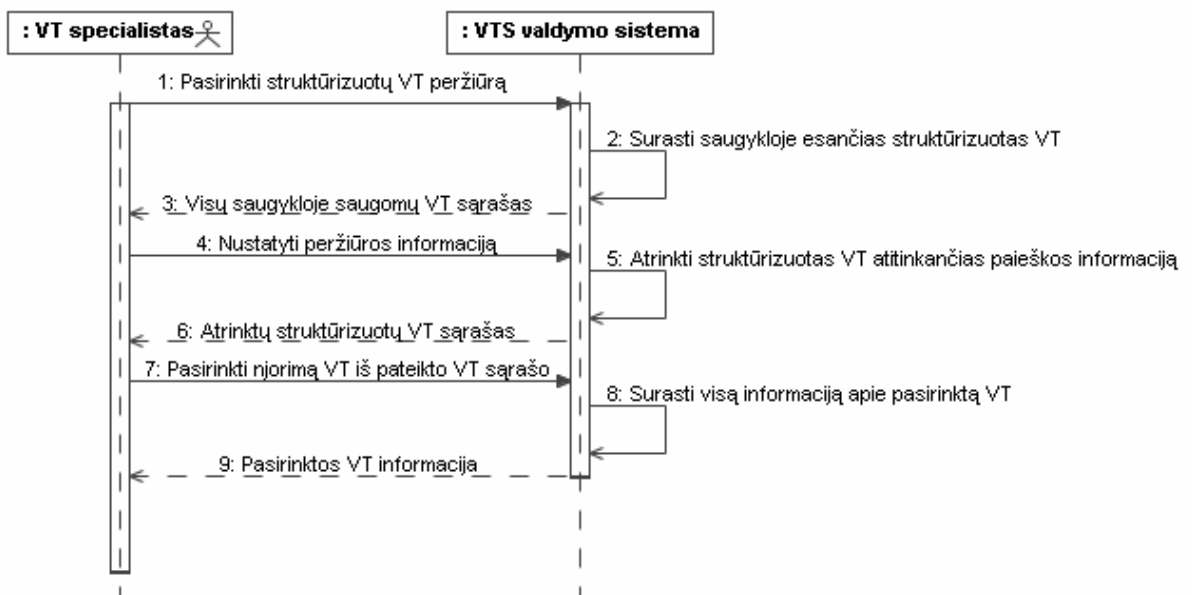
susirasti VT kuri bus modifikuojama. Peržiūra atliekama naudojant vieną iš taisyklės atributų, taisyklės filtruojamos pagal nustatytą kriterijų.

Peržiūrėjimo procesas (pav. 3.7, pav. 3.8) susideda iš:

- ✓ VT peržiūros pasirinkimas – peržiūros procesas pradedamas VT peržiūros pasirinkimu iš sistemos meniu. Proceso pradžioje atveriamas langas kuriame pateikiamos visos VTS saugomos VT.
- ✓ Peržiūros informacijos nustatymas – norint sumažinti sistemos pateikiamo VT sąrašo dydį nustatoma peržiūros informacija. Pasirenkamas kriterijus bei įvedamas raktinis žodis.
- ✓ Norimos VT pasirinkimas – iš pateikto VT sąrašo pasirenkama VT kurios detalesnę informaciją norima peržiūrėti.
- ✓ VT informacijos peržiūrėjimas – pasirinkus norimą sąrašo įrašą pateikiama detali informacija apie įrašą atitinkančią VT.



pav. 3.7 VT peržiūrėjimo specifikacija



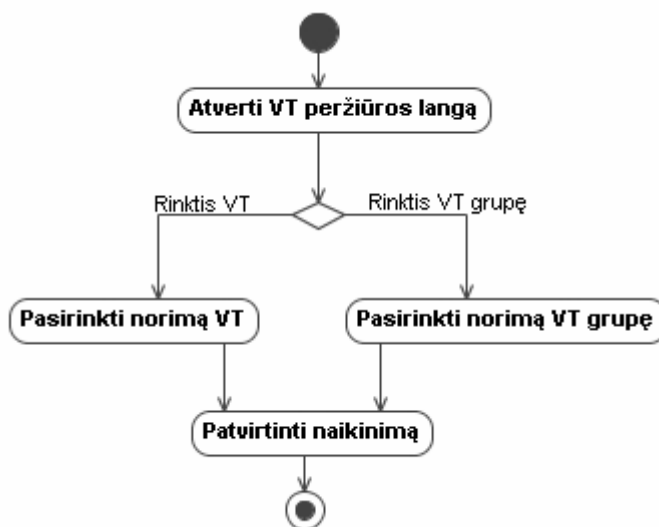
pav. 3.8 VT peržiūrėjimo sekų diagrama

3.4.5 Veiklos taisyklių naikinimo funkcija

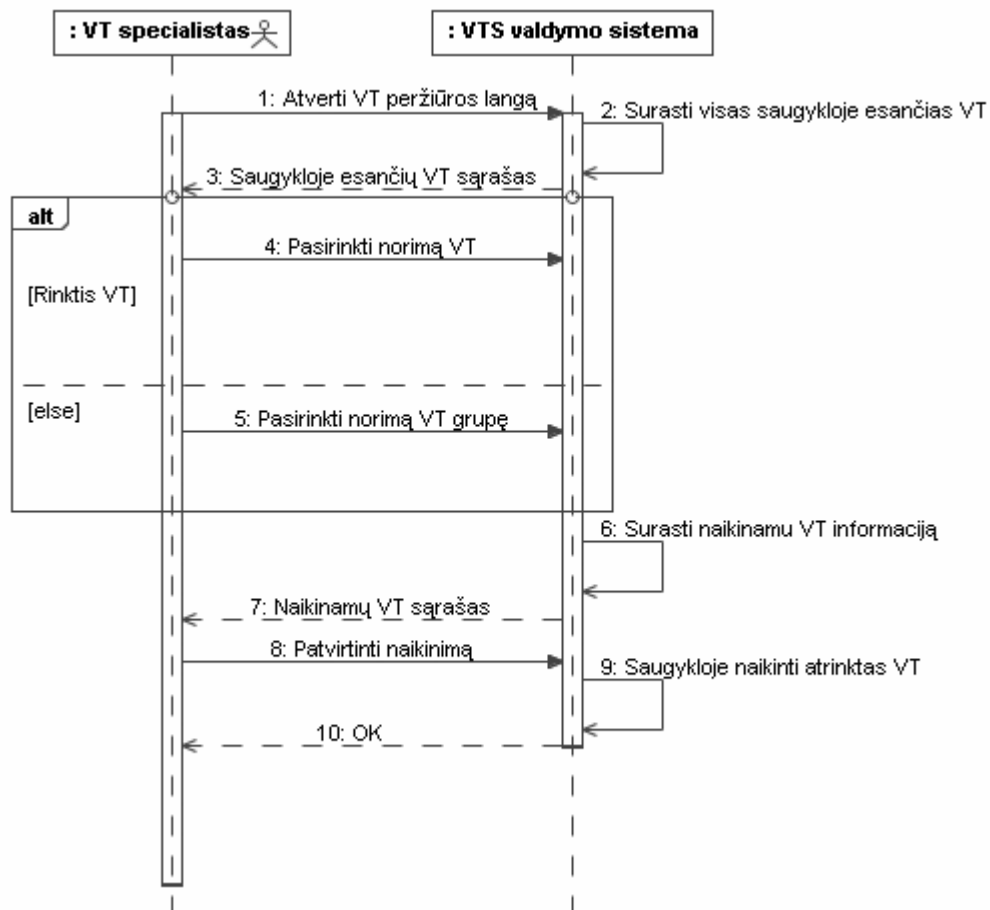
Taisyklių naikinimo procesas skirtas nenaudojamų, IS nebeįtakojančiu ar pasenusių VT pašalinimui iš sistemos. Taisyklės atrenkamos pagal nustatytą kriterijų, trynimą galima atlikti ir taisyklių grupėms, vienu metu iš sistemos pašalinant visą taisyklių rinkinį.

VT naikinimo procesas (pav. 3.9, pav. 3.10) susideda iš:

- ✓ VT peržiūros lango atvėrimas – naikinimo procesas pradedamas VTS esančių VT pasirinkimu. Tam, kad pasirinkti trinamas VT atveriamas VT peržiūros langas.
- ✓ Trinamų VT pasirinkimas:
 - Vienos VT pasirinkimas – sistemos pateikiamą VT sąrašą galima sumažinti nustatčius paieškos kriterijus, tuomet pasirenkama VT kuri bus pašalinta iš VTS.
 - VT grupės pasirinkimas – trinant VT grupę pasirenkamas grupės pavadinimas iš pateikto VTS esančių VT grupių sąrašo.
- ✓ Naikinimo patvirtinimas – pasirinkus naikinamą VT ar VT grupę atliekamas naikinimo patvirtinimas, po kurio seka pasirinktų VT pašalinimas iš VTS.



pav. 3.9 VT naikinimo specifikacija



pav. 3.10 VT naikinimo sekų diagrama

3.4.6 Šablono kūrimo funkcija

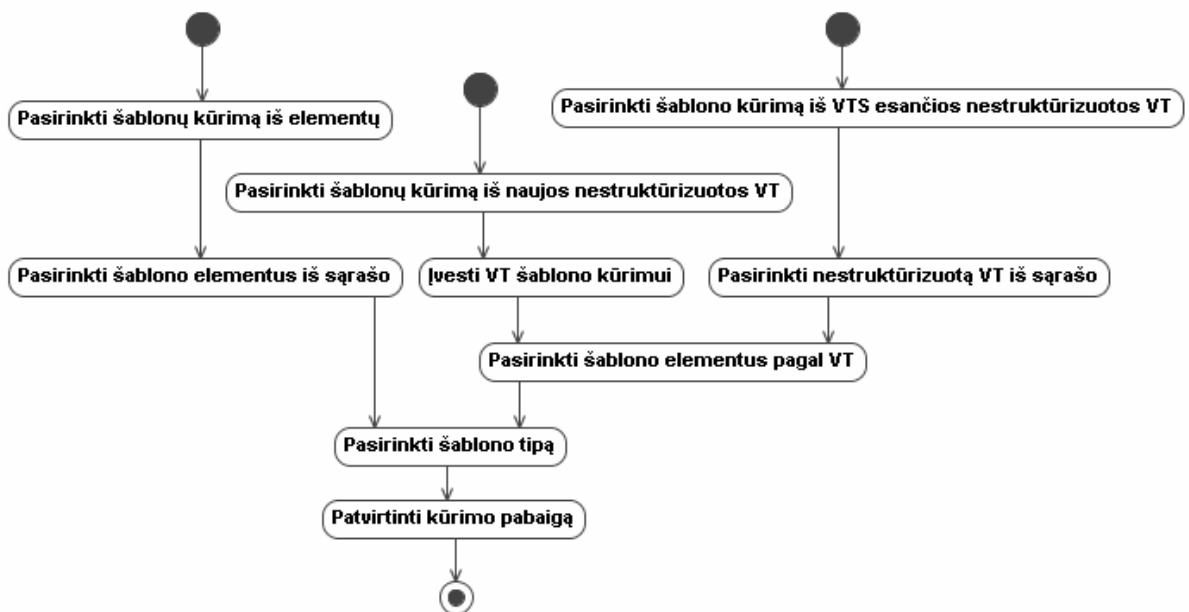
Šablonų skirtų VT struktūrizavimui kūrimas viena pagrindinių sistemos funkcijų. Norint suteikti kuo didesnę lankstumą šablonų įvedimo funkcijai suteikiami trys būdai ją inicijuoti.

Pirmas būdas (šablono sudarymas iš sistemos pateiktų elementų) paprastesnis, bet sudarantis didesnę galimybę suklysti. Antruoju ir trečiuoju būdais sudarinėjant šabloną naudojamas kontekstinis meniu kuris gali būti valdomas trumposiomis klavišų kombinacijomis. Toks daugialypis kūrimo būdas suteikia vartotojui daugiau laisvės ir pasirinkimo galimybių.

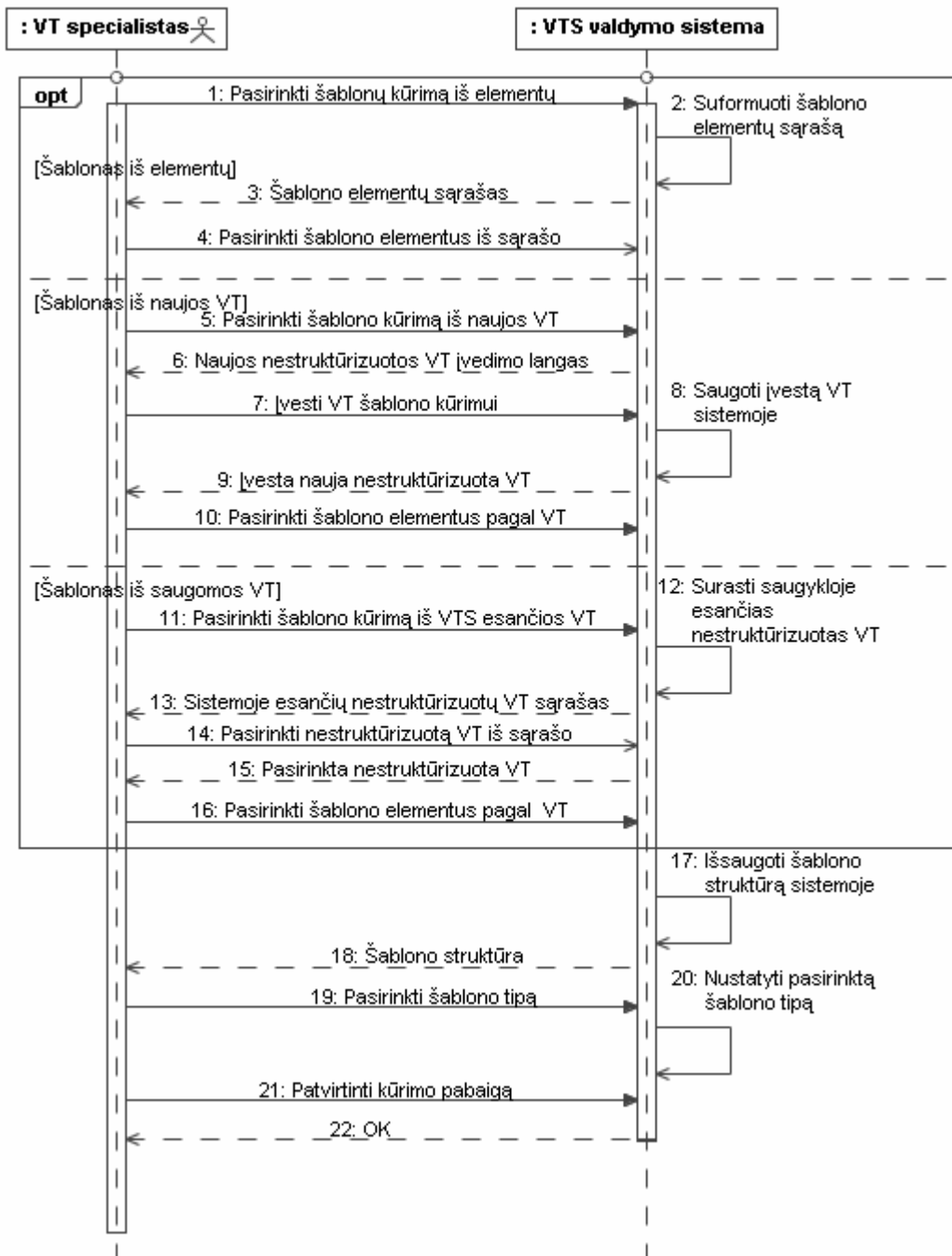
Šablonų kūrimo procesas (pav. 3.11, pav. 3.12) susideda iš:

- ✓ Šablonų struktūros sudarymo būdo pasirinkimas:
 1. Laisvai pasirenkant šablono elementus iš sistemos sukurto elementų sąrašo
 2. Pasirenkant šablono elementus pagal sistemoje esančią nestructūrizuotą VT
 3. Pasirenkant šablono elementus pagal naujai įvestą nestructūrizuotą VT
- ✓ Šablono sudarymui skirtos VT įvedimo ar pasirinkimas:

- ✓ Antruoju būdu sudarinėjant šablono struktūrą sistema pateikia VTS saugomų nestructūrizuotų VT sąrašą iš kurio galima pasirinkti VT pagal kurią bus sudarinėjamas šablonas.
- ✓ Trečiuoju būdu sudarinėjant šablono struktūrą sistema pareikalauja natūralia kalba įvesti nestructūrizuotą VT pagal kurią bus sudarinėjamas šablonas.
- ✓ Šablono struktūros sudarymas:
 - Pirmuoju būdu sudarinėjant šabloną elementai ir jų išdėstymo tvarka pasirenkami laisvai.
 - Antruoju ir trečiuoju būdu formuojant šablono struktūrą naudojamosi nestructūrizuotos VT elementais. Pažymėjus VT elementą jo šabloninis atitikmuo pasirenkamas iš kontekstinio meniu arba trumpųjų klavišų pagalba.
- ✓ Šablono tipo pasirinkimas – sudarius šablono struktūrą iš sistemos pateikto šablonų tipų sąrašo parenkamas šablono tipas, bei įvedamas šablono apibūdinimas.
- ✓ Kūrimo proceso pabaigos patvirtinimas – įsitikinus, kad visa įvesta informacija korektiška ir šablono struktūra atitinka norimą aptvirtinama šablono proceso pabaiga. Po patvirtinimo šablonas ir papildoma informacija išsaugoma VTS.



pav. 3.11 Šablono kūrimo specifikacija



pav. 3.12 Šablono kūrimo sekų diagrama

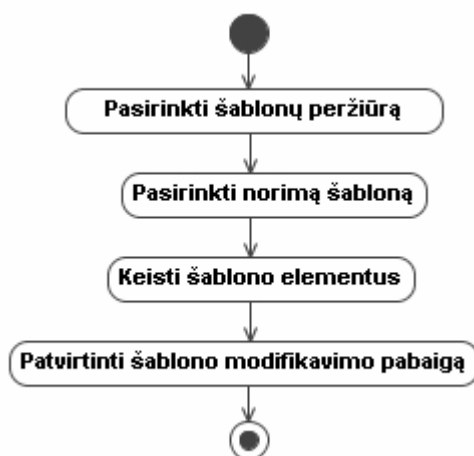
3.4.7 Šablono modifikavimo funkcija

Prieš pradėdant modifikuoti šabloną būtina nepamiršti, kad jis gali būti naudojamas ir susietas su VTS saugomomis VT. Modifikuojant šabloną susietą su bent viena struktūrizuota VT naujai gautas šablonas išsaugomas kaip naujas šablonas. Modifikuojant nenaudojamą šabloną pakoreguojama VTS saugoma šablono struktūra.

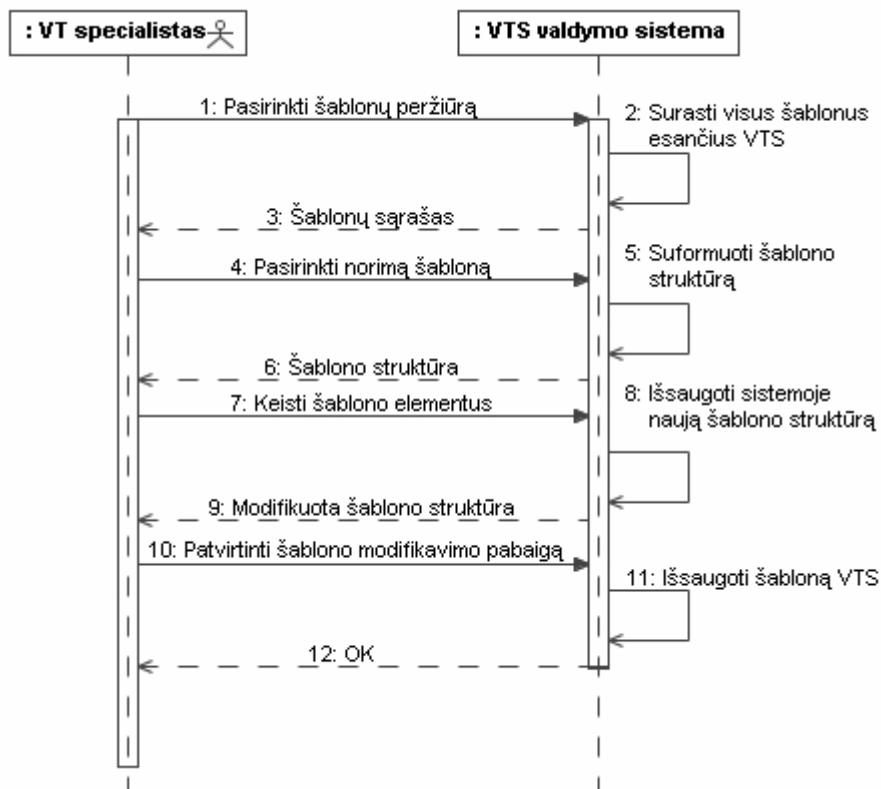
Šablono modifikavimo procesas (pav. 3.13, pav. 3.14) susideda iš:

- ✓ Šablonų peržiūros pasirinkimas – šablono modifikavimo proceso pradžioje atveriamas šablonų peržiūros langas kuriame pateikiami visi VTS saugomi šablonai.

- ✓ Norimo šablono pasirinkimas – šablonų sąrašo dydis gali būti keičiamas nustatant šablonų tipą. Iš sistemos pateikto sąrašo pasirenkamas šablonas kuris bus modifikuojamas.
- ✓ Šablono modifikavimas – modifikuojant šabloną galima keisti elementų kiekį, bei jų išdėstymo tvarką.
- ✓ Modifikavimo pabaigos patvirtinimas – nustačius visą norimą informaciją ir įsitikinus jos teisingumu patvirtinama modifikavimo pabaiga. Po patvirtinimo šablonas išsaugomas VTS.



pav. 3.13 Šablono modifikavimo specifikacija



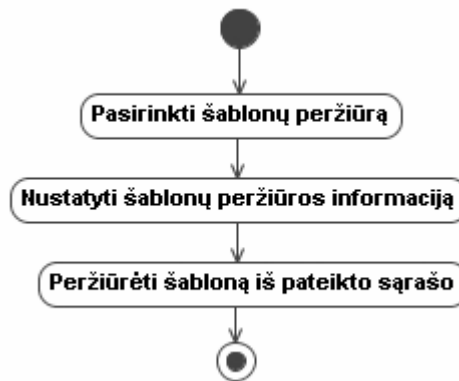
pav. 3.14 Šablono modifikavimo sekų diagrama

3.4.8 Šablonų peržiūrėjimo funkcija

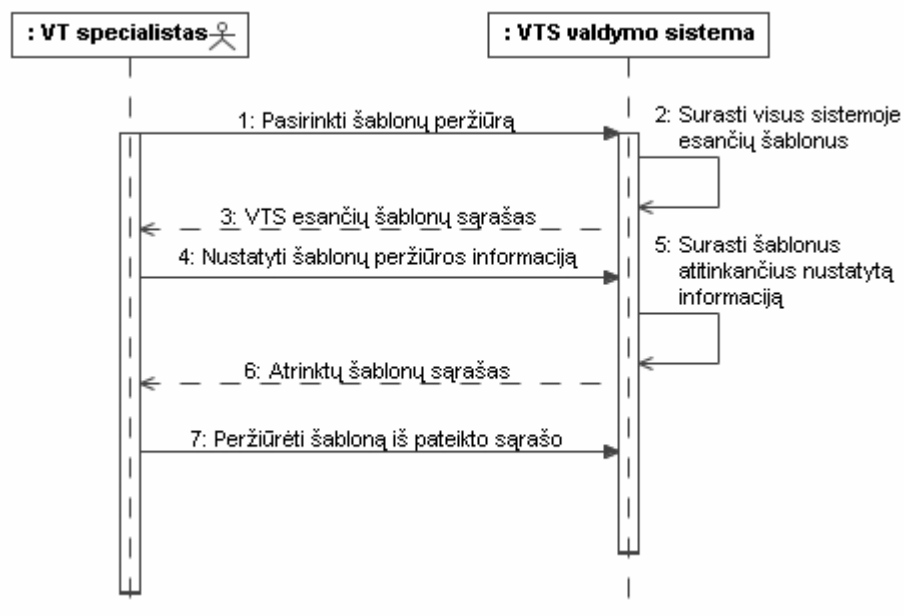
Šablonų peržiūra naudojama pasirenkant šabloną struktūrizuotų taisyklių įvedimui, prieš įvedant naują šabloną, bei šablono modifikavimo proceso pradžioje. Peržiūra atliekama pagal šablonų tipus.

Šablonų peržiūros procesas (pav. 3.15, pav. 3.16) susideda iš:

- ✓ Šablonų peržiūros pasirinkimas – šablonų peržiūros proceso pradžioje atveriamas langas kuriame pateikiamas visų VTS esančių šablonų sąrašas.
- ✓ Šablonų peržiūros informacijos nustatymas – norint sumažinti rodomų šablonų sąrašą pasirenkamas vienas iš sistemos pateiktų šablonų tipų. Tuomet pateikiamas pasirinkto tipo šablonų sąrašas.
- ✓ Šablono informacijos peržiūrėjimas – pateiktame šablonų sąrašė matoma visa kiekvieno šablono informacija.



pav. 3.15 Šablono peržiūrėjimo specifikacija



pav. 3.16 Šablono peržiūrėjimo sekų diagrama

3.4.9 Šablono naikinimo funkcija

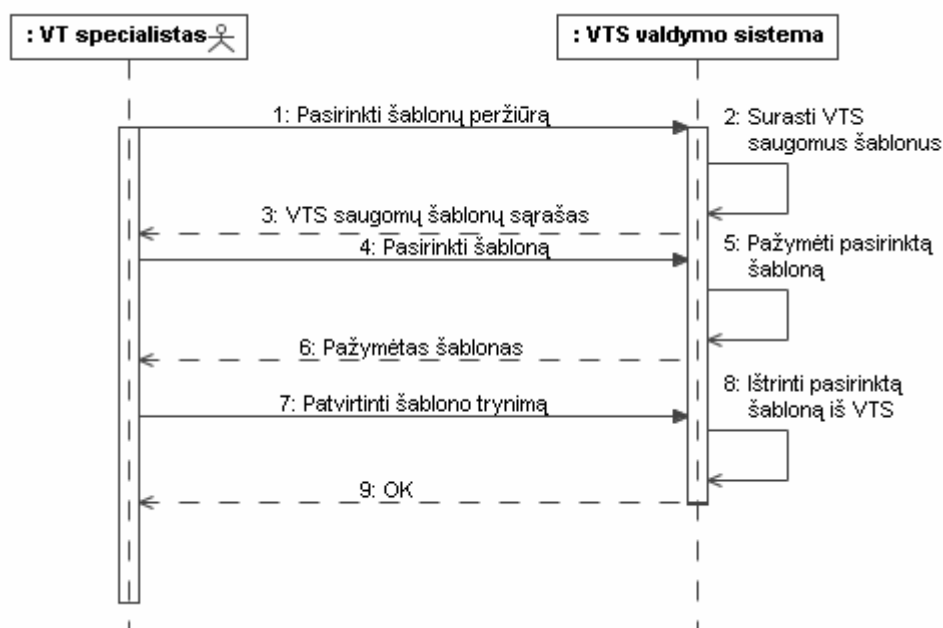
Trynimasis kaip ir modifikavimas šablonams yra ribotas. Siekiant užtikrinti naudojamų šablonų ir pagal juos specifiikuotų VT suderinamumą modifikuoti naudojamus šablonus draudžiama. Prieš pradėdant bet kokius šablono pakeitimus sistema patikrina šablono būseną, jei jis neturi susietų VT tai trynimasis leidžiamas.

Šablono naikinimo procesas (pav. 3.17, pav. 3.18) susideda iš:

- ✓ Šablonų peržiūros pasirinkimas - pradėdant šablonų naikinimo procesą atveriamas šablonų peržiūros langas kuriame pateikiami visi VTS esantys šablonai.
- ✓ Šablono pasirinkimas – iš sistemos pateikto sąrašo pasirenkamas šablonas kuris bus trinamas iš VTS. Sąrašo sudėtis gali būti keičiama pasirenkant vieną iš sistemos pateiktų šablonų tipų.
- ✓ Šablono trynimo pasirinkimas – nurodžius norimą šabloną pasirenkama trynimo funkcija.
- ✓ Šablono trynimo patvirtinimas – sistemai patikrinus, kad nurodytas šablonas nenaudojamas jis pašalinamas iš VTS.



pav. 3.17 Šablono trynimo specifikacija



pav. 3.18 Šablono trynimo sekų diagrama

3.5 Sistemos duomenų klasių modelis

Sistema projektuojama IS projektavimui skirtu modelio naudojamos duomenų bazės pagrindu, tad dalykinės srities modelis buvo gautas taikant atvirkštinę inžineriją. Šiam etapui buvo panaudota MagicDraw 11.5 v teikiama klasių modelio generavimo iš turimos duomenų bazės funkcija. Gauta esybių klasių diagrama pateikta pav. 3.19

Pereinant prie reikalavimų, veiklos modelio konceptai transformuojami į reikalavimų etapo esybių konceptus. Projekto klasės aprašo priimtus architektūrinius sprendimus (programinius modulius, komponentus, klases, interfeisus, vartotojo interfeiso klases, duomenų bazės lenteles, jų tarpusavio ryšius) ir detalias kompiuterizuotos sistemos objektų savybes (atributus, operacijas, duomenų tipus).[25]

Pagrindinės sistemos esybės (lent. 3.1), tai BusinessRule (veiklos taisyklė) nusakanti veiklos taisyklę ir jos papildomą informaciją. BR_Element (veiklos taisyklės elementas) atvaizduojantis VT elementą, bei Template (šablonas) ir TemplateElement (šablono elementas) atitinkamai realizuojantys šablono ir jo elemento objektus.

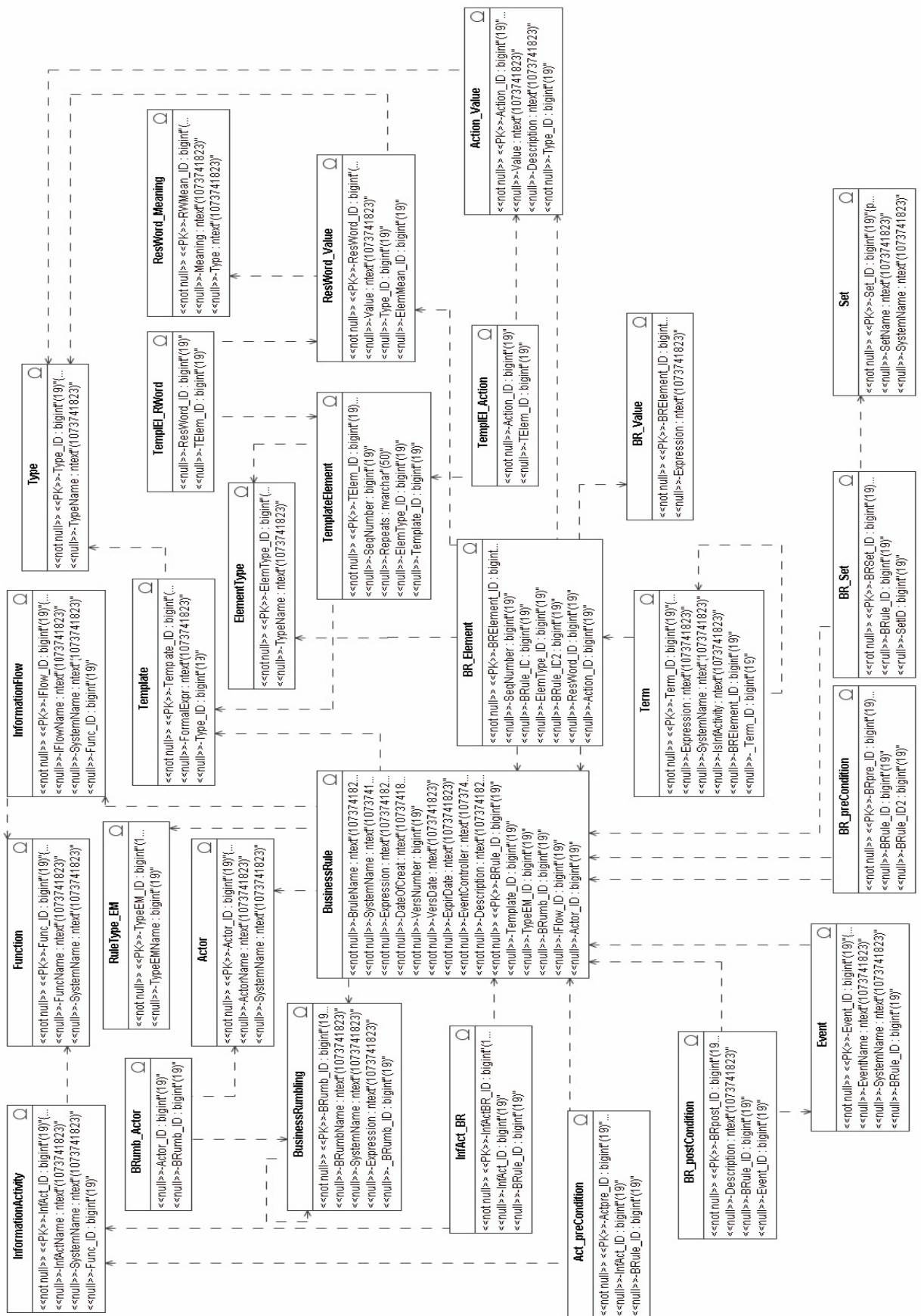
lent. 3.1 Veiklos taisyklių saugyklos esybių klasių ir laukų paaiškinimas [26]

Lentelė	Laukas	Paiškinimas
Actor		Aktorius, kuris gali būti žinių šaltiniu, atliekant VT analizę, arba atsakingu už VT įvykdymą, priežiūrą (Aktorius taip pat yra VM elementas)
	Actor_ID	Aktoriaus identifikatorius
	ActorName	Aktoriaus vardas

Lentelė	Laukas	Paaiškinimas
	SystemName	Sisteminis aktoriaus vardas
Function	Func_ID	Valdymo funkcija (Funkcija taip pat yra VM elementas) Funkcijos identifikatorius
	FuncName	Funkcijos vardas
	SystemName	Sisteminis funkcijos vardas
InformationActivity	InfAct_ID	Informacinė veikla (Informacinė veikla taip pat yra VM elementas) Informacinės veiklos identifikatorius
	InfActName	Informacinės veiklos vardas
	SystemName	Sisteminis informacinės veiklos vardas
Act_preCondition	Actpre_ID	Informacinės veiklos prieš-sąlyga Informacinės veiklos prieš-sąlygos identifikatorius
InformationFlow	IFlow_ID	Informacinis srautas (Informacinis srautas taip pat yra VM elementas) Srauto identifikatorius
	IFlowName	Srauto vardas
	SystemName	Sisteminis srauto vardas
Event	Event_ID	Įvykis (Įvykis taip pat yra VM elementas) Įvykio identifikatorius
	EventName	Įvykio vardas
	SystemName	Sisteminis įvykio vardas
BusinessRule	BRule_ID	Formalizuota veiklos taisyklė VT (Veiklos taisyklė taip pat yra VM elementas) VT identifikatorius
	BRuleName	VT vardas
	SystemName	Sisteminis VT vardas
	Expression	VT išraiška
	DateOfCreat	VT sukūrimo data
	VersNumber	VT versijos numeris
	VersDate	VT versijos data
	ExpirDate	Data, iki kurios VT galioja
	EventController	Loginė išraiška iš įvykių vardų ir loginių operatorių, skirta veiklos taisyklei inicijuoti
	Description	VT aprašymas
Set	Set_ID	Veiklos taisyklių rinkinys VT rinkinio identifikatorius
	SetName	VT rinkinio vardas
	SystemName	Sisteminis VT rinkinio vardas

Lentelė	Laukas	Paaiškinimas
RuleType_EM	TypeEM_ID TypeEMName	Veiklos taisyklių tipai pagal veiklos modelio sudėtį (interpretavimo, realizavimo, sprendimų priėmimo) VT tipo identifikatorius VT tipo vardas
BR_postCondition	BRpost_ID Description	Veiklos taisyklės po-sąlyga Veiklos taisyklės po-sąlygos identifikatorius Veiklos taisyklės po-sąlygos aprašymas
BR_preCondition	BRpre_ID	Veiklos taisyklės prieš-sąlyga Veiklos taisyklės prieš-sąlygos identifikatorius
BR_Element	BRElement_ID SeqNumber	Veiklos taisyklės elementas (VT yra sudaryta iš VT elementų) VT elemento identifikatorius VT elemento eilės numeris taisyklėje
BR_Value	BRElement_ID Expression	VT elementas, kurio tipas yra „Value“ VT elemento identifikatorius Reikšmė
Term	Term_ID Expression SystemName IsInfActivity	Terminas (terminų žodynas) Termino identifikatorius Termino išraiška Termino sisteminis vardas Ar terminas identifikuoja funkcijos informacinę veiklą
BusinessRumbling	BRumb_ID BRumbName SystemName Expression	Neformalizuota veiklos taisyklė Neformalizuotos VT identifikatorius Neformalizuotos VT vardas Sisteminis neformalizuotos VT vardas Neformalizuotos VT išraiška
Template	Template_ID FormalExpr	Veiklos taisyklės šablonas VT šablono identifikatorius Formalizuota išraiška, parodanti šablono sudėtį
TemplateElement	TElem_ID SeqNumber Repeats	VT šablono elementas VT šablono elemento identifikatorius VT šablono elemento sekos numeris Ar šablono elementas šablone gali kartotis
ElementType	ElemType_ID TypeName	Šablono elemento ir veiklos taisyklės elemento tipas Elemento tipo identifikatorius Elemento tipo vardas

Lentelė	Laukas	Paaiškinimas
Type	Type_ID TypeName	Šablono ir veiklos taisyklės tipas VT ir VT šablono tipo identifikatorius VT ir VT šablono tipo vardas
Action_Value	Action_ID Value Description	VT ir VT šablono elementas, kurio tipas yra „Action“ Veiksmo tipo elemento identifikatorius Veiksmo tipo elemento reikšmė Aprašymas
ResWord_Value	ResWord_ID Value	VT ir VT šablono elementas, kurio tipas yra „ReservedWord“ Rezervuoto žodžio identifikatorius Rezervuoto žodžio reikšmė
ResWord_Meaning	RWMean_ID Meaning Type	Rezervuoto žodžio reikšmės prasmė Rezervuoto žodžio reikšmės prasmės identifikatorius Rezervuoto žodžio reikšmės prasmė (ar šis žodis išreiškia sąryšį tarp dviejų objektų ar parodo savybės priklausomybę kokiam nors objektui) Jei rezervuoto žodžio reikšmės prasmė yra ryšys tarp dviejų objektų, tada šiame lauke nustatomas ryšio tipas: asociacija, apibendrinimas, agregavimas arba kompozicija



pav. 3.19 Esybių klasių diagrama

3.6 VTS valdymo sistemos klasių modelis

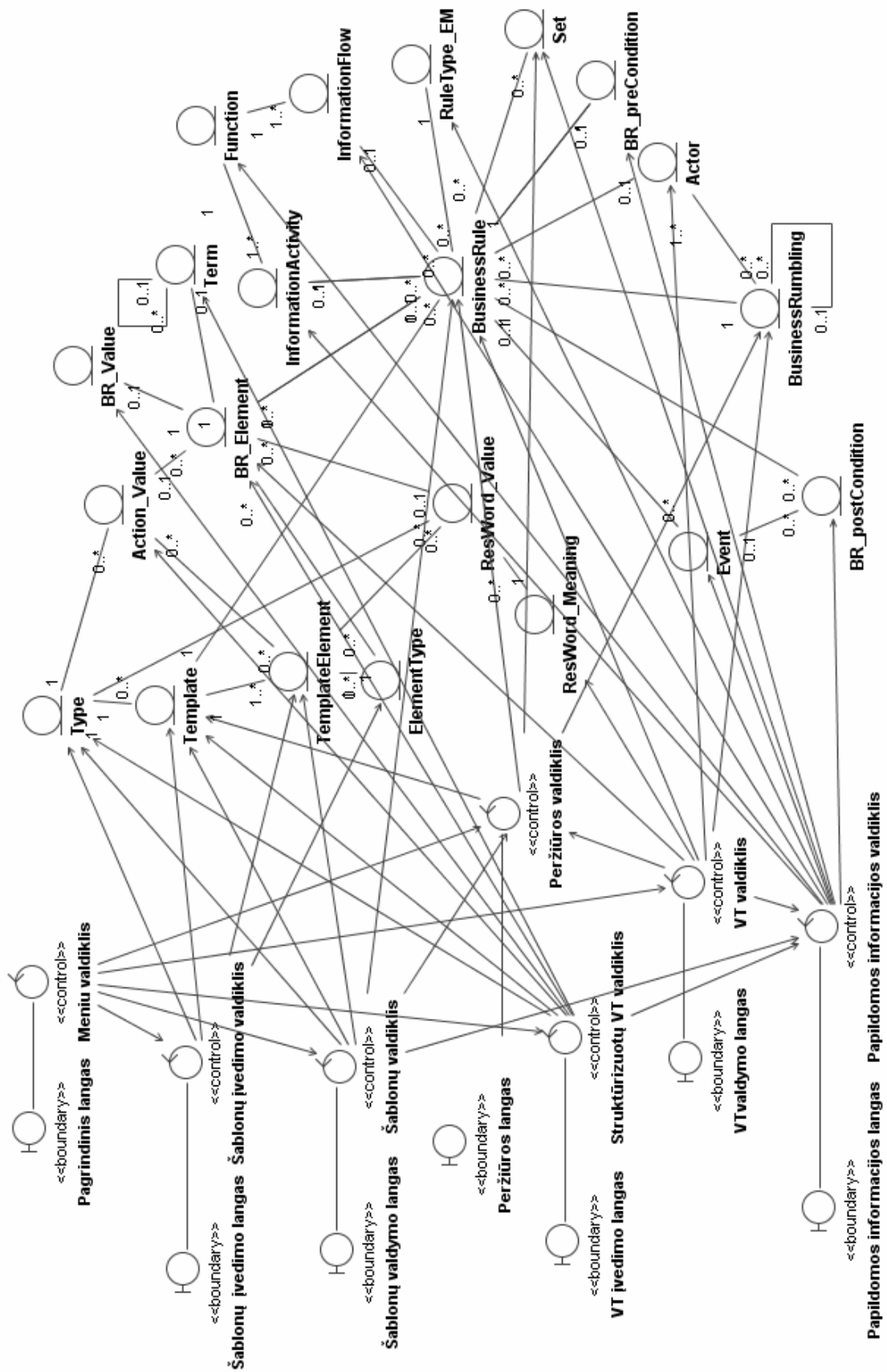
Atlikus reikalavimų specifikavimą, bei sudarius dalykinės srities modelį atliekama analizė. Jos metu sudaroma sistemos klasių diagrama (pav. 3.20). Joje vaizduojamos ribinės klasės atitinkančios vartotojui pateikiamus langus ir teikiančios vartotojo paslaugas. Taip pat vaizduojami valdikliai vykdantys bendravimą tarp sistemos ir veiklos taisyklių saugyklos. Trečio tipo vaizduojamos klasės atspindi VTS saugomas esybes kurios nuolat pateikia reikiamą informaciją, taip pat yra atnaujinamos ar papildomos valdiklių pagalba. Šios klasės sujungtos į bendrą diagramą suteikia galimybę sudaryti bendrą sistemos vaizdą.

Modelis parodo vartotojo naudojamų formų bendravimą su valdikliais. Taip pat pateikia valdiklių ir esybių ryšius. Modelyje matomas galimas sistemos suskirstymas į posistemius.

3.7 Reikalavimų specifikacijos apibendrinimas

Atliekant reikalavimų specifikaciją sudaryti veiklos sekų modeliai. Modeliuose pateikiama vartotojo pageidaujama veiksmų seka kiekvienos pagrindinės funkcijos atlikimo metu. Pastebėta, kad dauguma struktūrizuotų VT bei šablonų valdymo funkcijų pradedamos sistemoje esančių įrašų peržiūra. Pagal sudarytas veiksmų sekas sukurtos vartotojo ir sistemos bendravimo sekų diagramos. Pagal sudarytas sekų diagramas nustatyta vartotojui pateikiamos informacijos paieškos tvarka. Sudarinėjant sistemos projektą būtina įsitikinti, kad suprojektuota sistema atitiks vartotojo pageidaujamas veiksmų sekas.

Sudarytas sistemos duomenų klasių modelis. Esybių klasių diagrama gauta iš IS kūrimui skirto modelio naudojamos duomenų bazės panaudojus atvirkštinę inžineriją. Gauto esybių modelio pagrindu projektavimo etape bus sudaromas duomenų bazės modelis. Tolimesniuose etapuose pasitelkus gautas žinias bus kuriamas sistemos projektas, bei realizacijos modeliai.



pav. 3.20 Robastiškumo diagrama

4 VEIKLOS TAISYKLIŲ SAUGYKLOS VALDYMO SISTEMOS PROJEKTO SPECIFIKACIJA

Sistemos projektas sudaromas atsižvelgiant į analizės metu nustatytą funkcionalumą. Reikalavimų etape šis funkcionalumas specifikuotas vartotojo reikalavimus sistemai atspindinčiais modeliais.

Sistemos projektas prasideda nuo sprendimo priėmimo ir bendro būsimos sistemos aprašo. Aiškumui ir detalumui padidinti sudaromi statinis bei elgsenos modeliai. Sistema suskirstoma į posistemius realizuojančius tam tikrą sistemos atliekamų funkcijų dalį.

4.1 *Sistemos sprendimas*

Atlikus analizę bei nustačius reikalavimus nuspręsta sistemą realizuoti programa instaliuojama vartotojo kompiuteryje. Toks sistemos realizavimo būdas suteikia plačias galimybes kurti lanksčią bei intelektualią programos sąsają.

Sistemos tikslas – patogus VT saugyklos valdymas bus pasiektas suteikiant vartotojui galimybę naudoti šablonus. Taisyklių įvedimo paprastumą padidina sistemoje esančių elementų reikšmių pateikimas taisyklės struktūrizavimo metu. Sistemos realizacijai bus naudojama vartotojo – serverio architektūra suteikianti galimybę pasirinkti VT duomenų bazę.

4.2 *Sistemos architektūra - statinės struktūros modelis*

Statinį struktūros modelį sudaro vartotojo paslaugų modelis, veiklos paslaugų modelis bei duomenų paslaugų modelis. Šie modeliai gauti iš reikalavimų analizės modelio. Atliekant tolimesnį sistemos detalizavimą naudojamas duomenų paslaugų modelis, kadangi jis nurodo sistemos naudojamą informaciją.

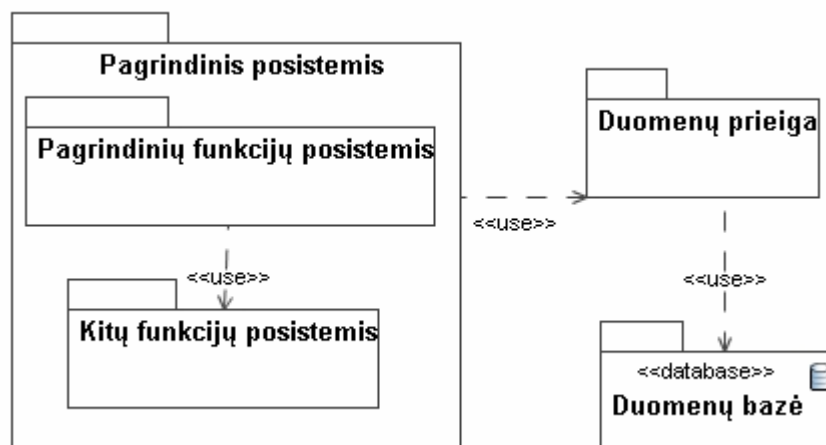
4.3 *Loginė visos sistemos architektūra*

Projektuojant sistemą ji suskirstoma į loginius vienetus – posistemius. Kiekvienas posistemis realizuoja atitinkamą sistemos teikiamų funkcijų dalį. Veiklos taisyklių saugyklos valdymo sistema suskirstyta į du posistemius (pav. 4.1).

Pagrindinių funkcijų posistemis teikia visas pagrindines šablonų valdymo funkcijas tokias kaip: šablonų kūrimas, modifikavimas, šalinimas, bei peržiūra. Jame taip pat realizuotos struktūrizuotų VT valdymo funkcijos: VT peržiūra, struktūrizavimas, modifikavimas, šalinimas.

Likusių VTS elementų valdymo funkcijos atskirtos į „Kitų funkcijų“ posistemį. Struktūrizuotų VT ir šablonų valdymo posistemis naudoja vieną vartotojo formą, o kitos funkcijos jas pasirinkus atsiveria naujuose languose. Siekiant atskirti skirtingai pateikiamas

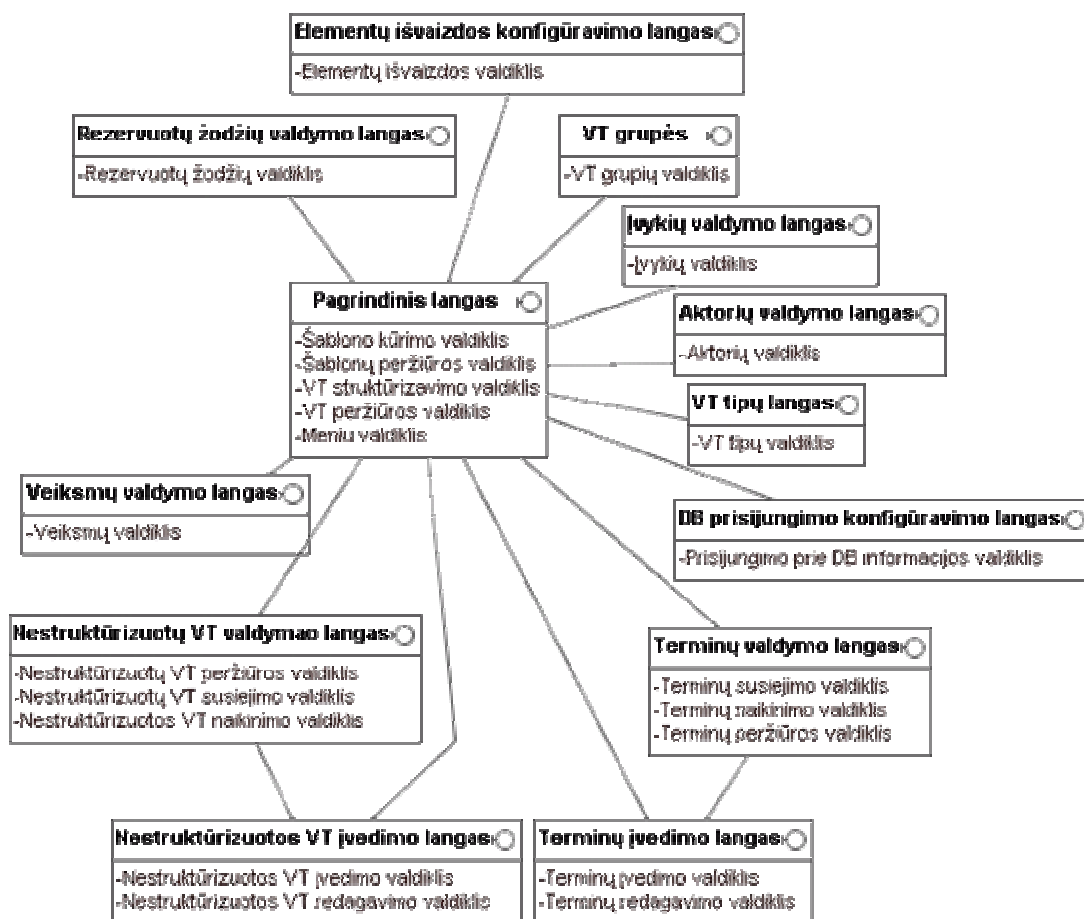
funkcijas jos ir buvo suskirstytos pagal į loginius sistemos posistemius. Sistemos prieiga prie nutolusiame serveryje patalpintos duomenų bazės vykdoma per TCP/IP protokolą naudojant duomenų prieigą.



pav. 4.1 Loginės sistemos architektūros diagrama

4.4 Vartotojo paslaugos

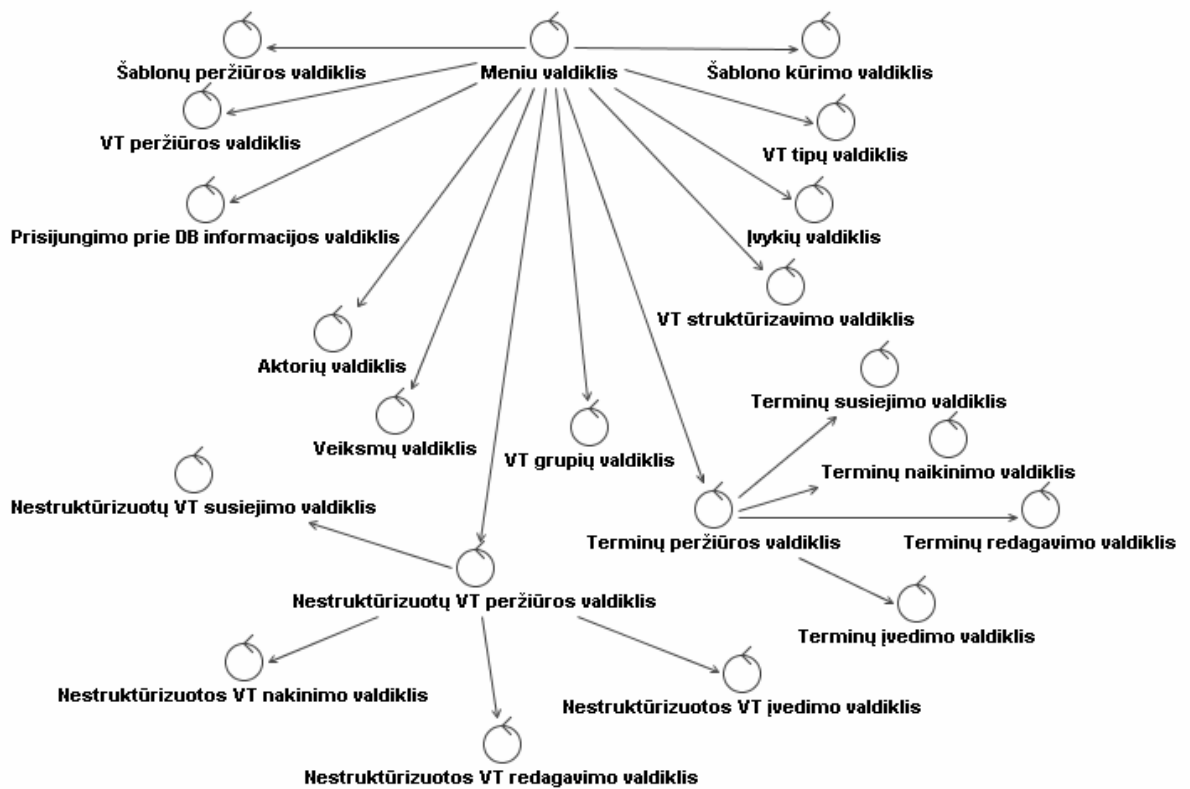
Vartotojo paslaugos, tai langai teikiantys informaciją apie vykstančius procesus. Vartotojas priklausomai nuo norimos atlikti funkcijos vadovaudamasis meniu atveria vieną ar kitą langą. Vartotojo sąsaja labai svarbi, kadangi ji yra pirminis objektas pagal kurį vartotojas vertina sistemą. Patogus ir savalaikis sistemos elementų pateikimas pasiekiamas kiekvienai svarbiai funkcijai sukuriant programos langą. Pagrindinis langas skirtas visų programa realizuotų langų apjungimui ir galimybei patekti į tuo metu reikiamą funkciją realizuojančią aplinką. Vartotojui pateikiamų sistemos langai, jų ryšiai bei langų naudojami valdikliai sudaro vartotojo paslaugas (pav. 4.2).



pav. 4.2 Vartotojo paslaugų diagrama

4.5 Veiklos paslaugos

Veiklos paslaugos, tai pagrindinis programos elementas tarpininkaujantis tarp vartotojo sąsajos ir duomenų. Jis realizuoja programos logiką ir visas numatytas funkcijas. Veiklos taisyklių saugyklos valdymo sistemai būtinas geras valdiklių tarpusavio bendradarbiavimas. Veiklos paslaugų diagrama (pav. 4.3) pateikia visus sistemoje naudojamus valdiklius bei jų ryšius. Joje aiškiai matoma, kad daugiausia ryšių turintis valdiklis yra meniu valdiklis. Taip yra todėl, kad jis yra pagrindinis, kitus sistemos valdiklius iškviečiantis valdiklis. Terminų peržiūros valdiklis sieja terminų valdymui skirtas funkcijas, analogiškai nestruktūrizuotų VT peržiūros valdiklis sieja funkcijas skirtas nestruktūrizuotų VT valdymui. Kiekvienas aktorių, įvykių, VT tipų, VT grupių bei veiksmų valdiklis susideda iš smulkesnių peržiūros, sukūrimo bei naikinimo valdiklių. Smulkesni valdikliai veiklos paslaugų diagramoje nevaizduojami siekiant diagramos paprastumo.

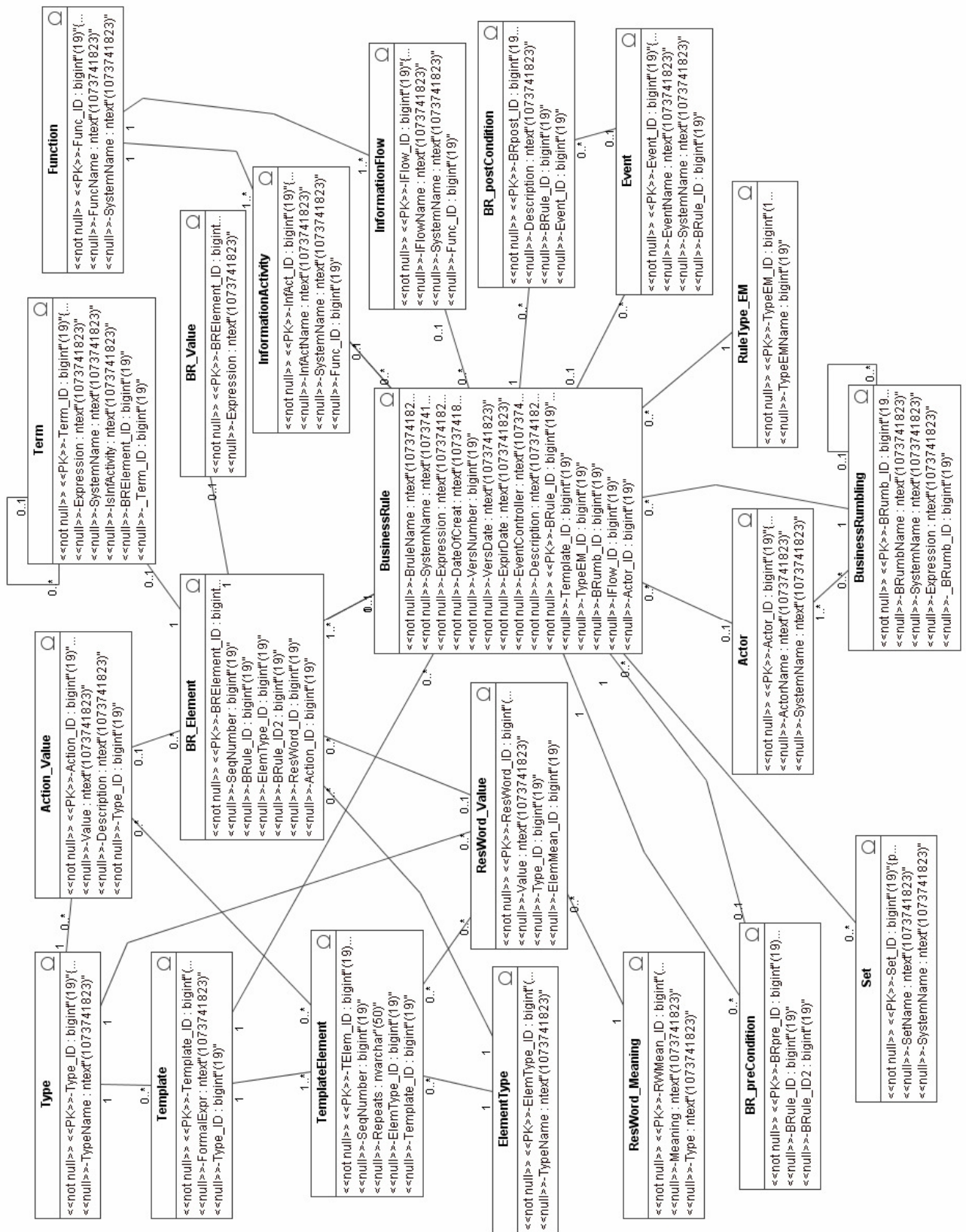


pav. 4.3 Veiklos paslaugų diagrama

4.6 Duomenų paslaugos

Duomenų paslaugos teikia informaciją kurią vėliau pagal vartotojo nurodytus veiksmus apdoroja sistemoje realizuoti valdikliai. Būtina užtikrinti nuolatinį duomenų integralumą, tad ryšiai tarp klasių riboja susijusių objektų pašalinimą.

Duomenų paslaugas VT valdymo sistemai teikia SQL serveris. Sudėtingiausias valdymas susijęs su daugiausia atributų ir ryšių turinčia veiklos taisyklę realizuojančia lentele – BusinessRule. Kadangi šią lentelę naudoja ne vienas valdiklis, tai būtina užtikrinti, kad duomenys nebūtų išgadinti netinkamos operacijos atlikimo metu.



pav. 4.4 Duomenų paslaugų modelis

4.7 Sistemos elgsenos modelis

Elgsenos modeliai naudojami aprašyti bendrą sistemos veikimą. Pilnai sistemos elgsenai nusakyti naudojamos veiklos diagramos (specifikacijos) ir sekų diagramos.

Veiklos diagramų tikslas – apibrėžti elgsenos mechanizmą. Veiklos diagrama nusako vykdomų veiksmų seką. Ši seka baigiasi tada, kai visi veiksmai įvykdyti arba įvyksta klaida, arba veiksmai nutraukiami.

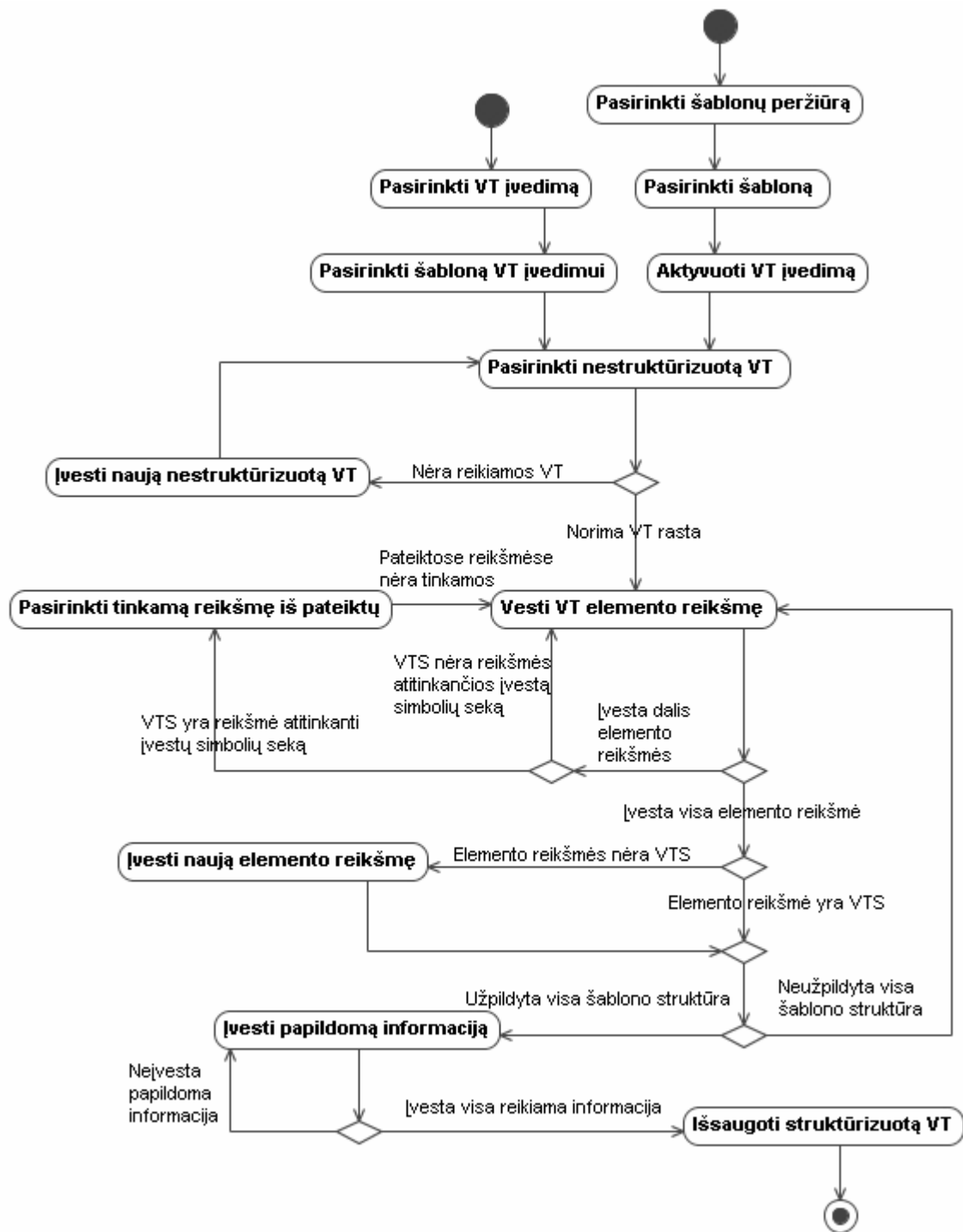
Sistemoje vykdomų procesų sekų diagramos vaizduoja sistemos atliekamus veiksmus. Jose vaizduojamos ribinės klasės, valdikliai bei duomenis teikiantys elementai. Jų tarpusavio bendravimas, informacijos išgavimo, pateikimo sekos. Priklausomai nuo vartotojo pasirinkimų ir sistemos apribojimų pateikiami alternatyvūs variantai, bei ciklai.

4.7.1 Veiklos taisyklės įvedimas

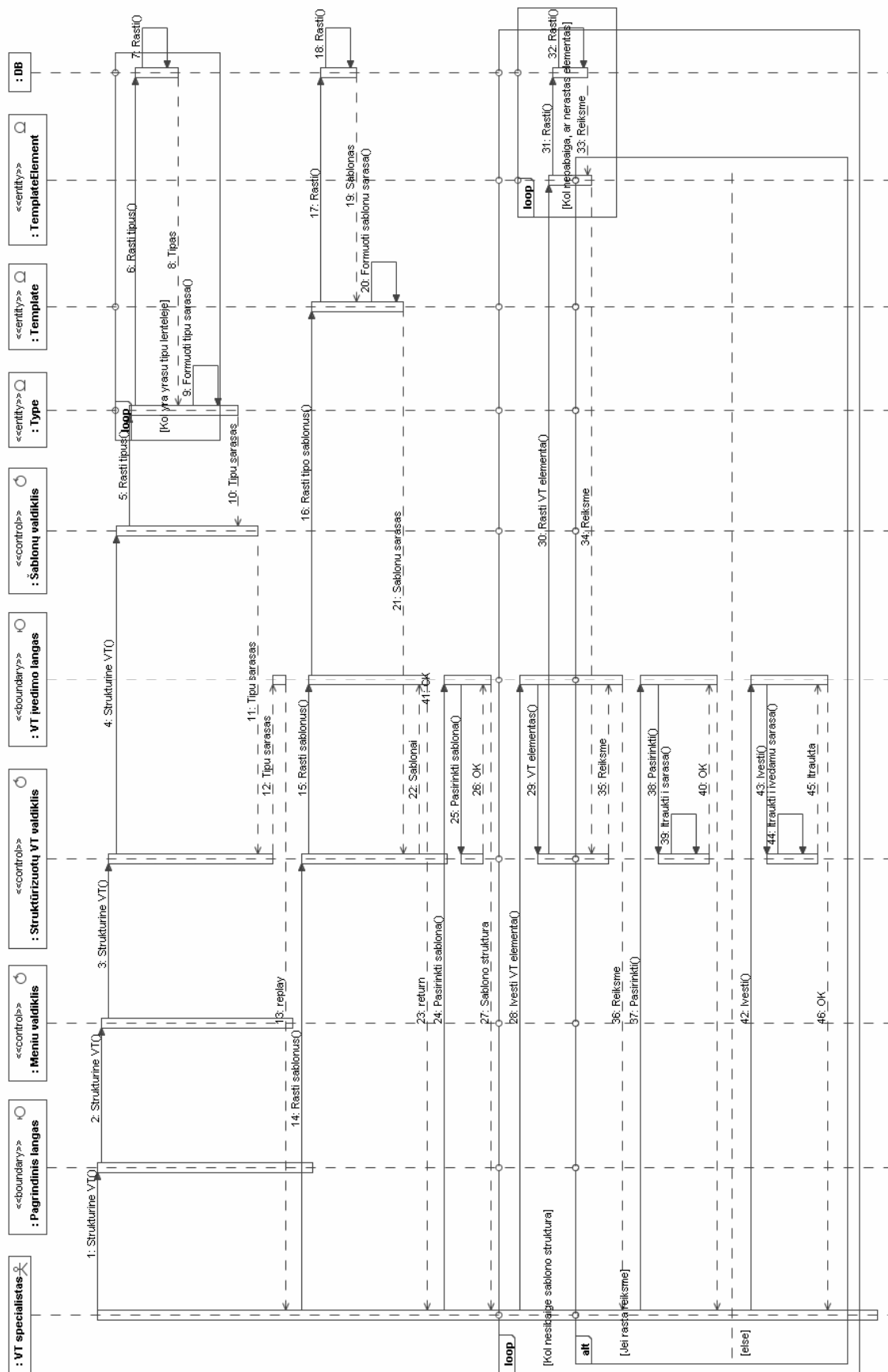
Struktūrizuotos veiklos taisyklės įvedimas viena iš sudėtingiausių sistemoje realizuojamų funkcijų. Ši funkcija realizuoja VT struktūrizavimą pagal sistemoje saugomą šabloną ir nestruktūrizuotą VT. Prieš įvedant taisyklę reikalaujama pasirinkti šabloną kuris nusakys įvedamos VT struktūrą. Taisyklė įvedinėjama tol, kol neužpildomi visi šablone numatyti struktūros elementai. Kiekvieno VT elemento įvedimo metu radus jo reikšmės atitikmenį VTS sistema siūlo pasirinkti esamą reikšmę, jei sistema VTS neranda įvedinėjamo elemento reikšmės pasiūloma ją įvesti.

Užpildžius visus šablono elementus būtina nurodyti papildomą VT informaciją, tokią kaip prieš ir po sąlygas, galiojimo datą, taisyklių rinkinį kuriam priklauso įvestoji VT ir kt..

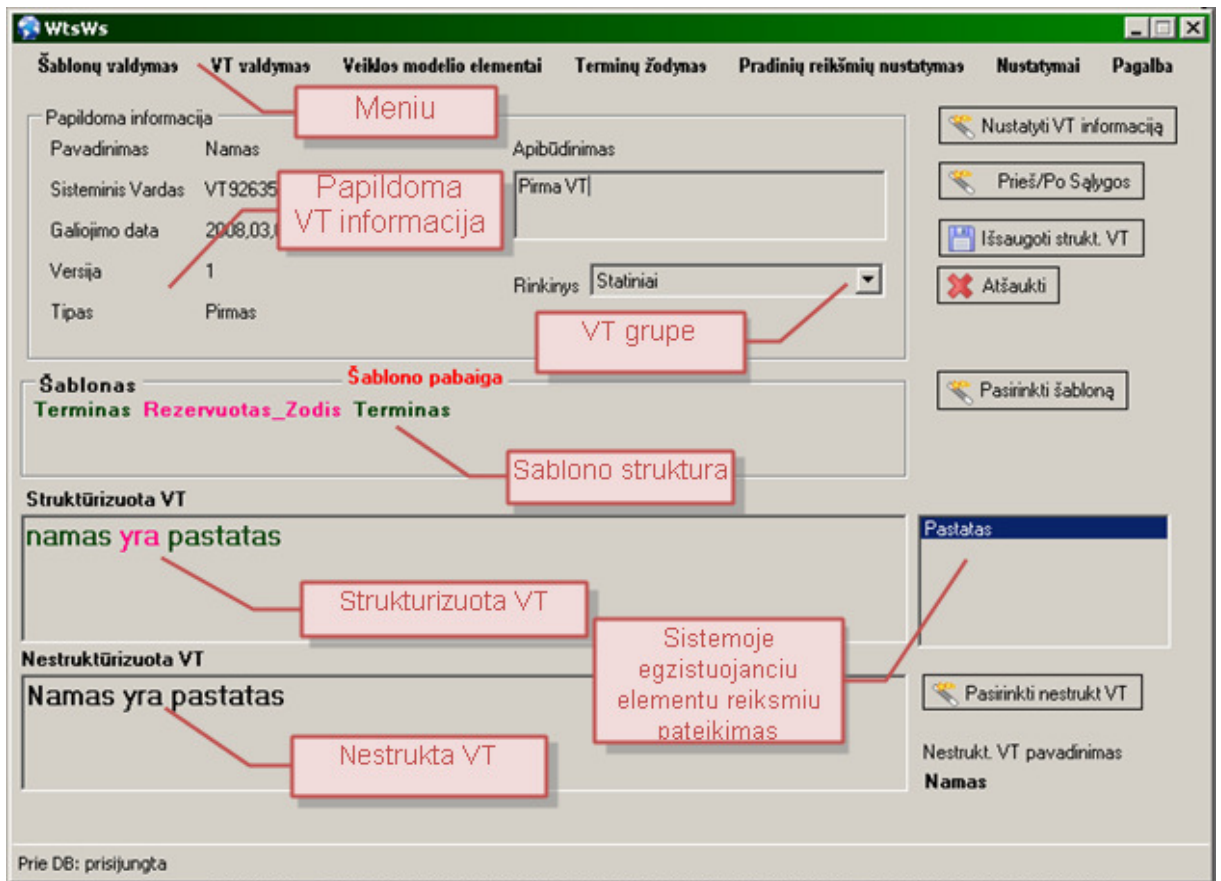
Struktūrizuotos VT įvedimo sekų diagramoje vaizduojamas šablono pagal kurį bus įvedinėjama taisyklė pasirinkimo procesas. Toliau parodoma kaip neradus sistemoje VT elemento reikšmės ji įterpiama į VTS sistemos pagalba.



pav. 4.5 Veiklos taisyklės įvedimo specifikacija



pav. 4.6 Strukturizotos veiklos taisyklēs sukūrimo sekū diagrama



pav. 4.7 Struktūrizuotos veiklos taisyklės kūrimo langas

Veiklos taisyklės struktūrizavimo lange (pav. 4.7) vartotojui pateikiama:

Menu juosta: Menu skirtas navigacijai po sistemą. Menu suteikia galimybę aktyvuoti norimą sistemos funkciją. Pasirinkus atitinkamą menu punktą atveriamas naujas sistemos langas, arba pagrindiniame lange sukuriama komponentai skirti kitai funkcijai atlikti.

Struktūrizuotos VT papildomos informacijos laukai: Struktūrizuotos VT papildoma informaciją sudaro jos pavadinimas, sisteminis vardas, galiojimo data, versija, apibūdinimas bei tipas. Šios informacijos nustatymas vyksta naujame lange kuris atveriamas spragtelėjus mygtuką „Nustatyti VT informaciją“

Struktūrizuotų VT grupių sąrašas: Sukuriant VT ją iš karto galima priskirti norimai VT grupei pasirenkant ją iš grupių sąrašo. Grupių sąrašą galima papildyti VT grupių valdymo lange.

Šablonas: Šablono struktūra nusako įvedinėjamos taisyklės struktūrą ir VT įvedimo metu parodo kuris šablono elementas įvedinėjamas esamu momentu. Šablono pabaigos indikatorius praneša šablono pilno išpildymo faktą. Toliau įvedinėjant elementus jie nebus tikrinami ir VT išsaugojimo metu nebus išsaugoti.

Nestruktūrizuotos VT išraiška: Nestruktūrizuota VT lange atvaizduojama tam, kad vartotojas galėtų naudodamasis ja įvedinėti norimus jos elementus struktūrizavimo lauke. Kartu su nestruktūrizuota VT pateikiamas jos pavadinimas.

Prieš/Po sąlygos: Norint nustatyti VT inicijavimo apribojimus bei po VT įvykdymo sekančius įvykius reikia spragtelėti mygtuką „Prieš/Po sąlygos“. Mygtukas atveria naują langą skirtą apribojimų bei įvykių nustatymui. Naujai atvertame lange taip pat galima suformuoti VT iniciavimo išraišką iš pasirinktų sąlygų.

Išsaugojimo, Atšaukimo mygtukai: Norint išsaugoti struktūrizuotą VT vartotojas turi spragtelėti mygtuką „Išsaugoti“. Tuomet sistemai įsitikinus, kad šablonas pilnai užpildytas ir įvesta visa reikiama informacija VT išsaugoma VTS. Norint išvalyti nustatytą informaciją spaudžiamas mygtukas „Atšaukti“.

4.7.2 Šablono sukūrimas

Šablonai – VT struktūrizavimo priemonė suteikianti galimybę struktūrizuoti natūralia kalba užrašytas VT. Sistemoje pateikiama galimybė sukurti įvairios struktūros šablonus iš nustatyto tipo elementų. Sistema suteikia galimybę kurti šablonus trim būdais.

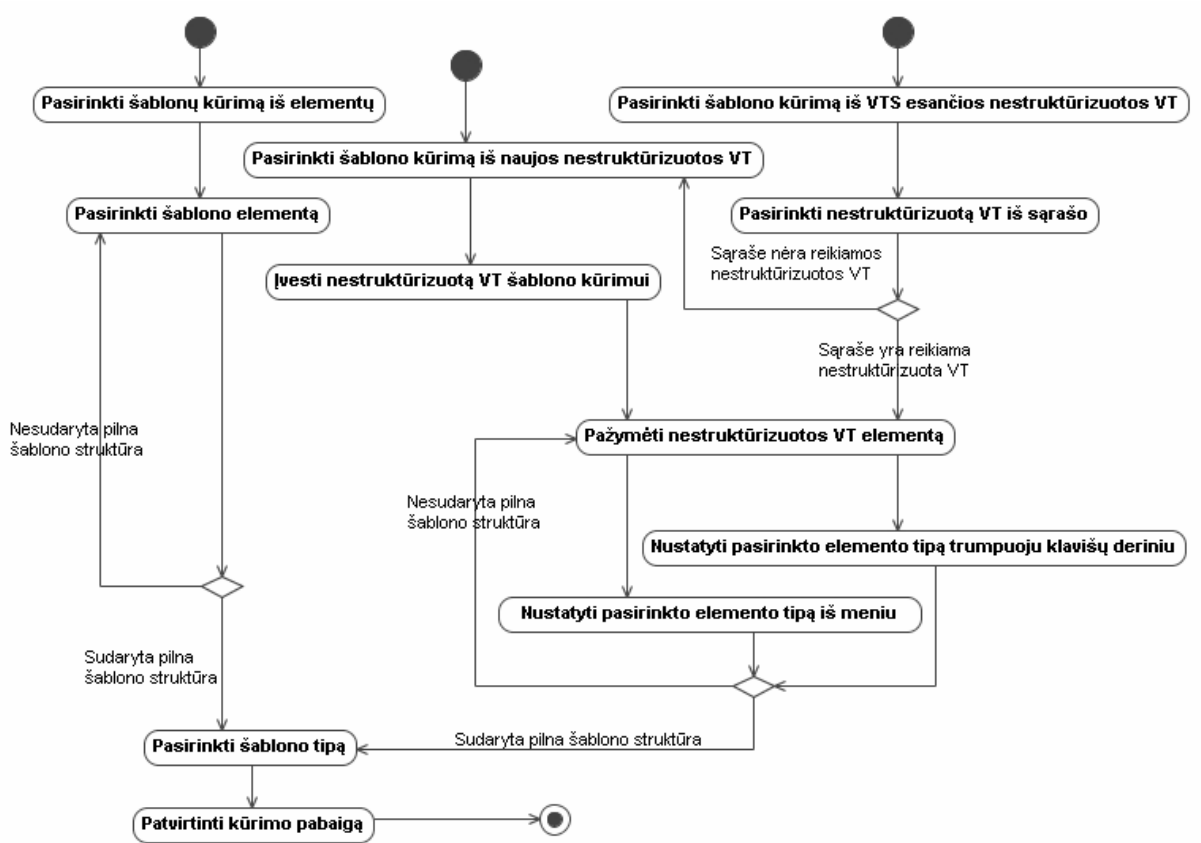
Pirmas šablono sukūrimo būdas skirtas šablono sudarymui iš elementų neturint pavyzdinės VT. Šablono struktūra sudaroma laivai pasirenkant sistemos siūlomus elementus ir juos išdėstant norima tvarka.

Antras šablono sukūrimo būdas naudojamas turint nestruktūrizuotą VTS saugomą pavyzdinę VT kuri neturi savo šablono atitikmens sistemoje. Taisyklė pasirenkama iš sistemos pateikto nestruktūrizuotų VTS saugomų VT sąrašo.

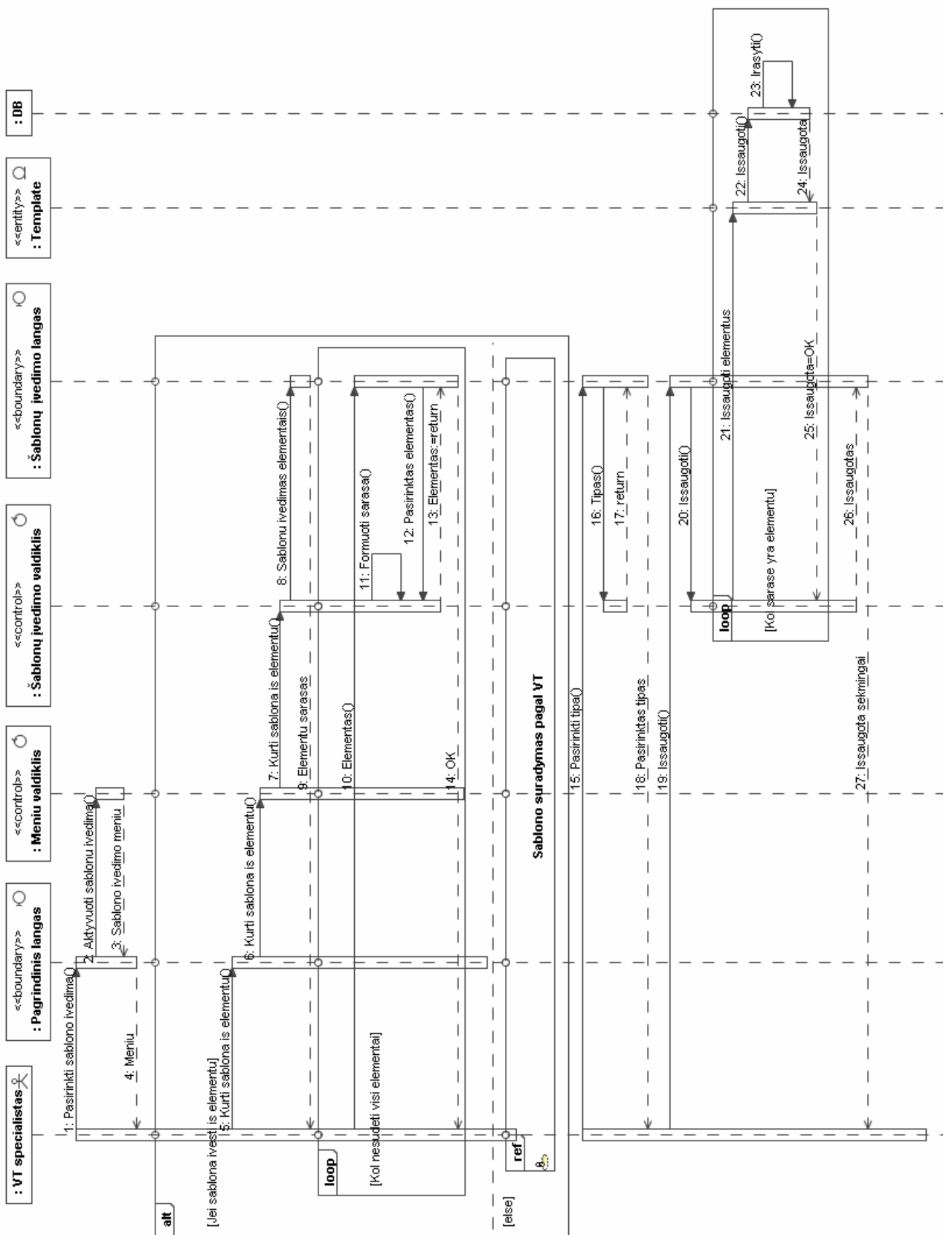
Trečias šablono sukūrimo būdas naudojamas kuriant šabloną pagal naują nestruktūrizuotą VT. Norint tokiu būdu sukurti šabloną į sistemą prieš kuriant šabloną teks įvesti nestruktūrizuotą VT užrašytą natūralia kalba.

Antrojo ir trečiojo šablonų kūrimo būdo metu kiekvienam nestruktūrizuotos VT elementui sukuriamas šabloninis atitikmuo. Šabloninius elementus pažymėtam VT žodžiui galima pasirinkti iš kontekstinio meniu ar naudojant trumpuosius klavišų derinius. VT elementų reikšmės papildo duomenų bazę.

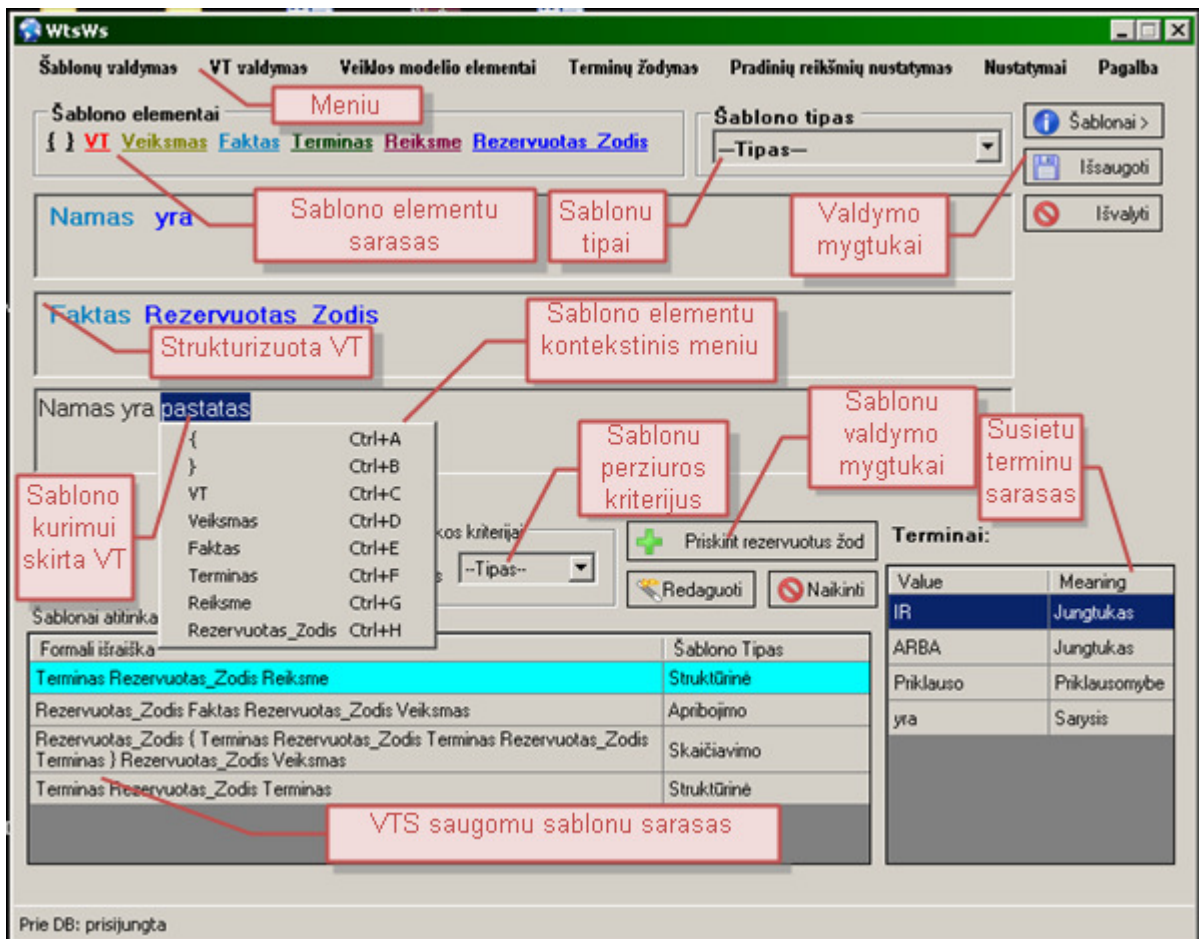
Šablono kūrimo sekų diagrama parodo veiksmų seką kurios metu atliekamas šablono įvedimo lango atvėrimas ir sistemoje saugomų elementų įkėlimas į vartotojui pateikiamą elementų sąrašą. Vartotojas nusprendžia ar kokiu būdu jis kurs šablono struktūrą. Priklausomai nuo vartotojo pasirinkimo jam pateikiamas vienokios ar kitokios konfigūracijos šablonų kūrimo langas.



pav. 4.8 Šablono sukūrimo specifikacija



pag. 4.9 Šablono sukūrimo sekų diagrama



pav. 4.10 Šablono kūrimo pagal nestructūrizuotą VT langas

Šablono kūrimo iš VTS saugomos nestructūrizuotos VT lange (pav. 4.10) vartotojui pateikiama:

Meniu juosta: meniu šablonų kūrimo lange atlieka tokias pat funkcijas kaip paminėta skyriuje 4.7.1 Veiklos taisyklės įvedimas.

Šablono elementų sąrašas: šablono elementų sąrašas pateikia sistemoje esančius šablono elementus. Spragtelėjus ant norimo elemento jis atsiranda šablono struktūros kūrimo lange.

Šablono elementų kontekstinis meniu: įvedinėjant šabloną pagal nestructūrizuotą VT pažymėjus VT elementą ir spragtelėjus dešinę pelės klavišą pateikiamas kontekstinis šablono elementų sąrašas. Kontekstiniame meniu pateikiamus elementus galima aktyvuoti ir trumposiomis klavišų kombinacijomis.

Strukturizuota VT: įvedinėjant šabloną pagal strukturizuotą VT kiekvieno elemento įterpimo metu formuojama ir iš naudojamos VT gauta strukturizuota taisyklė. Baigus kurti šabloną ir jį išsaugojus strukturizuotą VT galima įterpti į VTS prieš tai papildžius trūkstamą informaciją.

Šablono kūrimo VT: Šablonui kurti gali būti naudojama VTS saugoma nestruktūrizuota VT. Nestruktūrizuotą VT galima įvesti šablono proceso pradžioje, o baigus formuoti šabloną išsaugoti VTS.

Šablonų tipų sąrašas: Sukūrus šablono struktūrą būtina pasirinkti kokio tipo šablonas sukurtas. Nepasirinkus šablono tipo nebus galima išsaugoti sukurto šablono, apie tai informuoja pranešimas.

Kūrimo valdymo mygtukai: Nustaćius šablono struktūrą ir jo tipą šablono kūrimo proceso pabaiga patvirtinama mygtuku „išsaugoti“. Norint išvalyti šablono kūrimo langus spaudžiama mygtukas „išvalyti“. Šablonų peržiūros langui paslėpti/parodyti naudojamas mygtukas „Šablonai>“.

Šablonų peržiūros kriterijus: šablonų peržiūros lange pateikiami VTS saugomi šablonai. Pasirinkus peržiūros kriterijų pateikiamas sumažintas šablonų sąrašas.

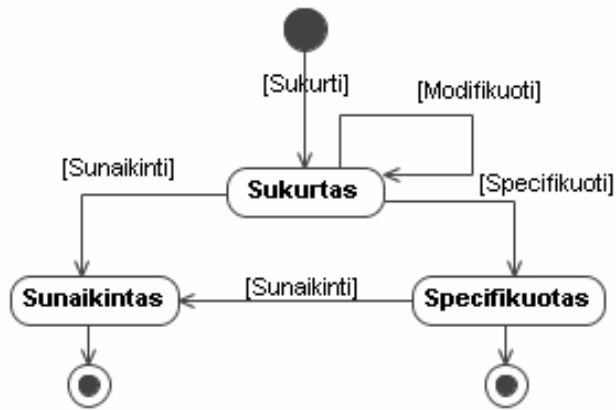
Šablonų valdymo mygtukai: Norint šablonų tipui priskirti rezervuotų žodžių aibę pažymimas atitinkamas šablonas ir spragtelimas mygtukas „Priskirti rezervuotus žod“, kuris atveria rezervuotų žodžių sąrašą. Siekiant pašalinti šabloną jis pažymimas ir spaudžiamas mygtukas „Naikinti“. Jei šablonas naudojamas bent vienos struktūrizuotos VT jo naikinimas neleidžiamas.

Susietų terminų sąrašas: Pasirinkus atitinkamo tipo šabloną pateikiamas su šiuo tipu susietų rezervuotų žodžių sąrašas. Norint sąrašą papildyti spaudžiamas rezervuotų žodžių priskyrimo mygtukas „Priskirti rezervuotus žod“

4.7.3 Sistemos objektų būsenų kaitos diagramos

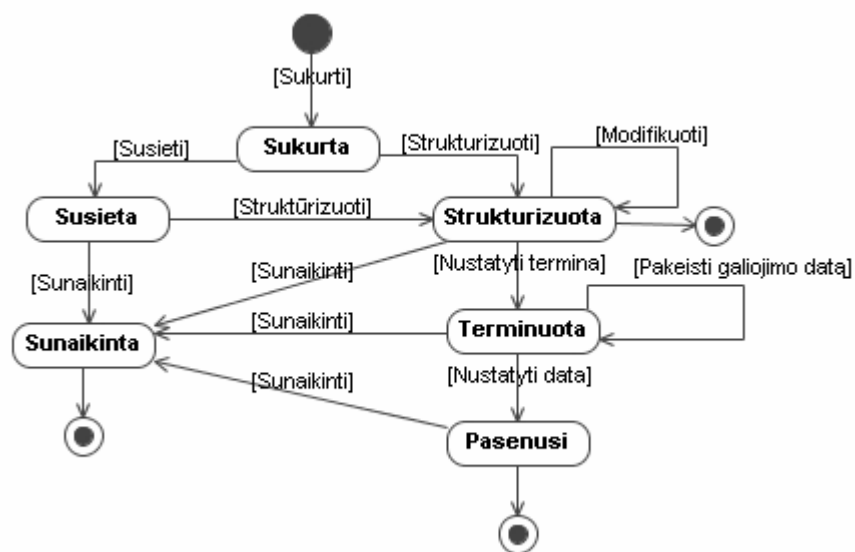
Sistemoje naudojamų objektų būsenos nuolat kinta. Visi elementai įgauna pirminę būseną sukurtas. Toliau objekto būseną kinta priklausomai nuo objekto paskirties ir elgsenos. Dalis objektų po būsenų pasikeitimo pasiekia galutinę sunaikinimo būseną, o dalis gali likti nesunaikinti visą sistemos gyvavimo laiką.

Veiklos taisyklės elementas turi tris būsenas. Įvedus nestruktūrizuotą VT į VTS visi jos elementai įgauna būseną „sukurtas“. Atlikus nestruktūrizuotos VT specifikavimą naudojant šabloną elementai įgauna būseną „specifikuotas“. Naikinant nestruktūrizuotą VT šablono elementas iš būsenos „sukurtas“ pereina į vieną iš dviejų galutinių būsenų „sunaikintas“. Naikinant nestruktūrizuotą VT šablono elementas iš būsenos „specifikuotas“ pereina į būseną „sunaikintas“. Veiklos taisyklės elemento būsenų kaita pavaizduota pav. 4.11



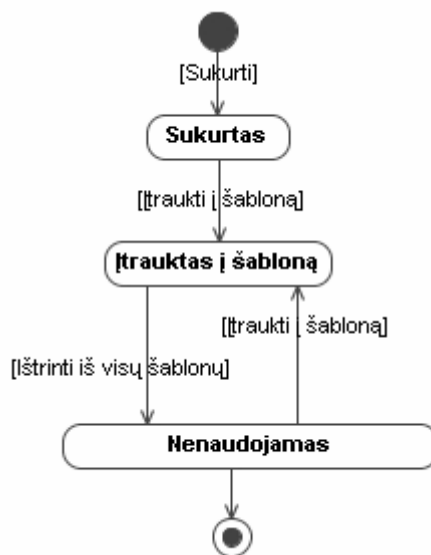
pav. 4.11 VT elemento būsenų modelis

Veiklos taisyklė gali įgauti vieną iš penkių būsenų. Pirmąją būseną VT įgauna į VTS įterpus nestructūrizuotą VT natūralia kalba. Susiejant kelias nestructūrizuotas VT būseną pasikeičia į „*susieta*“. Iš šios būsenos nestructūrizuotos VT naikinimo metu pereinama į vieną iš galutinių būsenų „*sunaikinta*“. Taip pat iš būsenos „*susieta*“ atlikus VT struktūrizavimą pagal šabloną būseną pasikeičia į „*struktūrizuota*“. Nenaikinant struktūrizuotos VT ir nenustatant jai galiojimo termino būseną „*struktūrizuota*“ gali tapti galutine. Naikinant struktūrizuotą VT ji pereina į būseną „*sunaikinta*“, o nustatant jai galiojimo terminą į būseną „*terminuota*“. Keičiant terminuotos VT galiojimo datą jos būseną nesikeičia, naikinant terminuotą VT ji pereina į būseną „*sunaikinta*“. Pasibaigus terminuotos VT galiojimo laikui VT būseną pasikeičia į „*pasenusi*“, tokioje būsenoje esančią VT galima palikti VTS arba sunaikinant pervesti į būseną „*sunaikinta*“. Veiklos taisyklės galimos būsenos, bei jų kaita pavaizduota pav. 4.12.



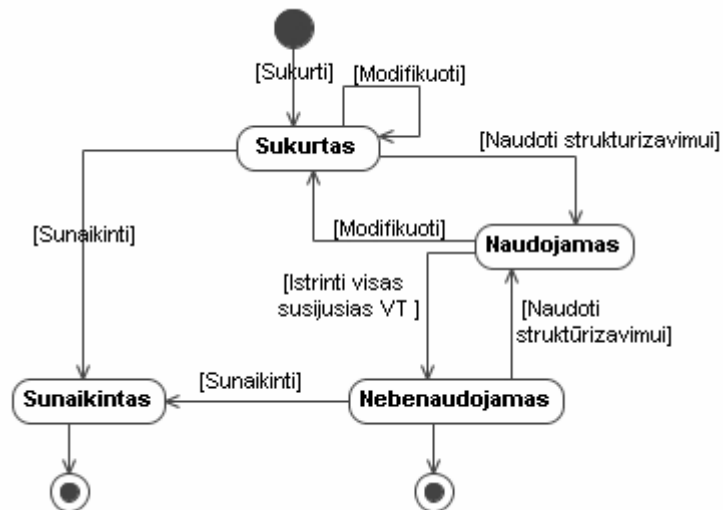
pav. 4.12 Veiklos taisyklės būsenų modelis

Šablono elementas gali įgauti vieną iš trijų būsenų. Pirmąją būseną „*sukurtas*“ jis įgauna jau VTS užpildymo pagrindiniais duomenimis metu. Kadangi šablono elementai viso VTS gyvavimo metu nenaikinami, tai jie gali įgauti būsenas „*įtrauktas į šabloną*“ ir „*ne naudojamas*“. Šablono elementas pereina į būseną „*įtrauktas į šabloną*“ tuomet, kai sukuriamas nors vienas šablonas. Ištrynus visus šablonus iš sistemos elementai pereina į būseną „*ne naudojamas*“. Šablono elemento galimos būsenos bei jų kaita pavaizduota pav. 4.13



pav. 4.13 Šablono elemento būsenų modelis

Šablonas gali įgauti vieną iš keturių būsenų. Vienu iš trijų būdų sukūrus šabloną jis įgauna būseną „*sukurtas*“. Atliekant nenaudojamo šablono modifikaciją jis nekeičia būsenos. Panaudojus šabloną VT struktūrizavimui jis pereina į būseną „*naudojamas*“. Atliekant naudojamą šablono modifikaciją sukuriamas naujas šablonas, tad jis pereina į būseną „*sukurtas*“. Ištrynus visas pagal šabloną struktūrizuotas VT šablono būseną pasikeičia į „*ne naudojamas*“. Pakartotinai panaudojus nenaudojamą šabloną jis grįžta į būseną „*naudojamas*“. Šablonas gali pereiti į būseną „*sunaikintas*“ tik tuo atveju jei VTS nėra pagal jį struktūrizuotų VT, taigi tik iš būsenų „*sukurtas*“ ir „*ne naudojamas*“. Būsenos „*sunaikintas*“ ir „*ne naudojamas*“ yra galutinės šablono būsenos. Šablono būsenos bei jų kaita pavaizduota pav. 4.14.



pav. 4.14 Šablono būsenų modelis

4.7.4 Projektavimo etapo apibendrinimas

Projektavimo etapo metu nuspręsta, kad sistema bus realizuojama vartotojo kompiuteryje diegiama programine įranga, o VTS patalpinama nutolusiame serveryje turinčiame duomenų bazės valdymo sistemą. Numatytas loginis sistemos suskirstymas posistemiais atskiriant pagrindines struktūrizuotų VT ir šablonų valdymo funkcijas.

Sudarytas vartotojui pateikiamų formų modelis su nurodytais kiekvienai formai priklausančiais valdikliais. Veiklos modelyje pavaizduoti valdiklių tarpusavio ryšiai, bei sudarytas duomenų paslaugų modelis atskleidžiantis būsimos duomenų bazės vaizdą. Sistemos elgsenos modelyje atskleistos sudėtingiausios funkcijos naudojant veiklos bei sekų diagramas, pateikti funkcijas atitinkantys vartotojo grafines sąsajos langai. Taip pat pateiktos sistemos objektų būsenų kaitos atskleidžiančios visas galimas objektų būsenas ir jų įgijimo kelius.

5 FIZINĖ DUOMENŲ BAZĖ

Kadangi realizacija buvo pradėta kartu su modeliavimu norint aiškiau nustatyti kintančius vartotojų reikalavimus, tai duomenų bazė jau buvo realizuota Microsoft SQL Server 2005 programine įranga. Tad norint gauti jau esamą saugyklos schemą buvo atliekama atvirkštinė inžinerija naudojant MagicDraw 11.5 v įrankį.

Sistema buvo kuriama iteraciniu kūrimo būdu, tad nuolatinis duomenų bazės pokytis neišvengiamas. Norint turėti SQL serveryje saugomos duomenų bazės modelį nuolat buvo atliekama atvirkštinė inžinerija. Naudojant MagicDraw 11.5 versiją susidurta su DB išgavimo iš serverio problemomis. Kadangi įrankis pritaikytas darbui su Microsoft SQL Server 2000, o sistemos kūrimui buvo naudojama naujesnė (Microsoft SQL Server 2005) serverio versija MagicDraw įrankis neturėjo galimybės prisijungti prie saerveryje saugomos DB.

Įdiegus naujesnę JDBC (Java Database Connectivity) prieigą prie serverio pavyko išgauti duomenų bazės schemą (pav. 5.1). Duomenų bazėje saugomos iš elementų sudarytos VT ir jų struktūrizavimui naudojami šablonai. Šablonų saugojimui naudojamos keturios lentelės, kurių pagalba atskirai saugomi šablonų elementai, jų tipai, bei patys šablonai ir šablonų tipai. Nestruktūrizuotos VT saugomos kaip natūralia kalba užrašyti sakiniai ir yra siejamos su jas įvedusiais aktoriais, bei struktūrizuotom VT. Struktūrizuotas VT sauganti VTS lentelė siejama su aktorių, šablonų, struktūrizuotos VT elementų, prieš bei po sąlygų lentelėmis.

Realizuojant veiklos taisyklių saugyklos valdymo sistemą duomenų bazė saugoma serveryje buvo koreguojama. Siekiant įgyvendinti galimybę šablonų tipui priskirti rezervuotų žodžių aibę DB papildyta lentelė (Type_ResWord). Ši lentelė sieja lentelę saugančią šablonų tipus (Type) ir lentelę saugančią rezervuotų žodžių reikšmes(ResWord_Value). Lentelė įterpta tam, kad realizuotų ryšį „daug-su-daug“. Taip papildžius DB atsiranda galimybė norimą rezervuotų žodžių poaibį priskirti vienam iš šablonų tipų. Šablonų tipam priskirtų rezervuotų žodžių poaibiai gali turėti vienodų elementų.

Tam, kad terminas galėtų suteikti reikšmę keletui VT elementų pakeistas ryšys tarp VT elementus (BRElement_ID) ir terminus (Term) saugančių lentelių. Taip pat terminų lentelėje panaikintas veiklos taisyklės elemento identifikavimo atributas(BRElement_ID), o VT elementų lentelėje įterptas terminą identifikuojantis (Term_ID) išorinis raktas.

Vykdamt sistemos realizaciją nustatyta, kad lentelės saugančios VT grupių informaciją pavadinimas (Set) negali būti naudojamas SQL užklausoje. Lentelės pavadinimas yra VTS saugoti pasirinktos Microsoft SQL server saugomos DB valdymo sistemos raktinis žodis. VT grupių informaciją saugančios lentelės pavadinimas pakeistas į „Sets“.

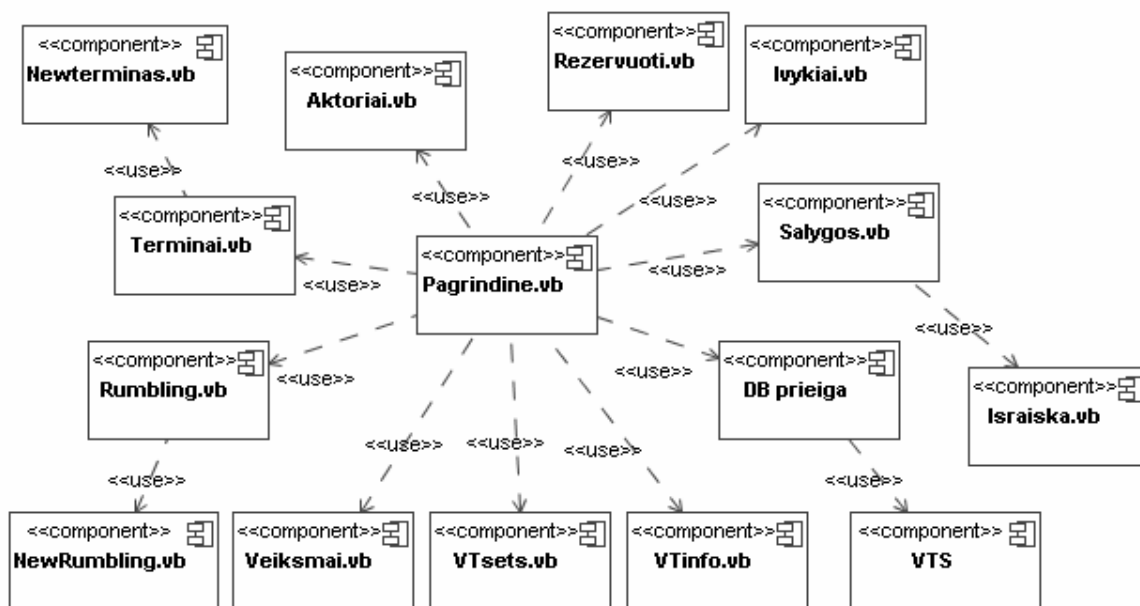
VT paleidžiantį įvykį prijungus prie sąlygų kurios turi būti tenkinamos paleidžiant VT panaikintas ryšys tarp VT (BusinessRule) ir įvykius (Event) saugančių lentelių. Sukurtas ryšys tarp įvykių (Event) ir VT prieš sąlygų (BR_preCondition) lentelių. Šių pakeitimų metu įvykių lentelėje panaikintas BRule_ID atributas identifikuojantis struktūrizuotą VT, o prieš sąlygų lentelėje įterptas įvykį identifikuojantis išorinis raktas „Event_ID“.

Siekiant padidinti duomenų bazėje saugomų elementų reikšmės aiškumą dauguma lentelių papildytos „Description“ lauku apibūdinančiu konkretų įrašą, laukas yra neprivalomas, tad jis gali būti pildomas tuomet kai elemento reikšmė reikalauja papildomo paaiškinimo.

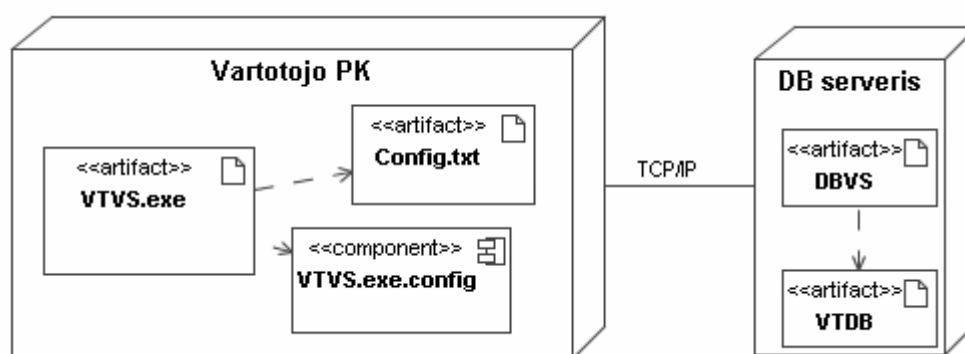
5.1 Realizacijos modelis

Realizacijos modelį sudaro sistemos komponentų ir sistemos įdiegimo modeliai. Sistemos programinių komponentų diagramoje (pav. 5.2) pateikiami sistemą sudarantys komponentai. Pagrindinis komponentas (Pagrindine.vb) apjungia struktūrizuotų VT bei šablonų valdymo funkcijas. Visi kiti komponentai aktyvuojami naudojant Pagrindine.vb. Komponentai „NewRumbling.vb“ ir „Newterminas.vb“ aktyvuojami naudojant atitinkamai „Rumbling.vb“ ir „Terminai.vb“. „Prisijungimas prie DB vyksta naudojant DB prieigos komponentą.

Įdiegimo modelyje (pav. 5.3) matomas programos išsidėstymas fiziniuose įrenginiuose. Programa instaliuojama vartotojo personaliniame kompiuteryje, o valdoma VTS patalpinama nutolusiame serveryje. VTS valdymui sistema naudoja serveryje esančia DBVS. Taip pat yra galimybė serverio programinę įrangą įdiegti vartotojo personaliniame kompiuteryje ir visą VT valdymo procesą vykdyti nenaudojant TCP/IP protokolo.



pav. 5.2 Sistemos programinių komponentų architektūros diagrama



pav. 5.3 Sistemos įdiegimo modelis

6 SISTEMOS TESTAVIMAS

6.1 Testavimo modelis

Testavimas būtinas kiekvienos sistemos patvirtinimo procesas. Norint pilnai ištestuoti sistemą būtina tikrinti kiekvieną ją sudarantį elementą. Kadangi sistema kuriama nuolat tobulinant sukurtą prototipą, tai nuolatinis testavimas yra neatsiejama sistemos kūrimo dalis.

Sistemai sudaryti testavimo duomenys turintys atskleisti silpnąsias sistemos vietas. Tik klaidų aptikęs testas skaitomas pavykusi, tad stengiamasi išnagrinėti visus galimus sistemos naudojimo atvejus.

Atlikus testavimą ir aptikus problemą atliekamas sistemos patobulinimas. Sistema testuojama ir taisoma tol, kol neaptinkama klaidų. Sistemos prototipe nebeaptikus klaidų atliekamas tolimesnis suprojektuotų funkcijų realizavimas.

6.2 Sistemos testavimo scenarijai

6.2.1. Meniu

Meniu suteikia galimybę vartotojui atverti sistemos langus skirtus įvairioms funkcijoms vykdyti. Testavimo metu būtina įsitikinti, kad visi meniu punktai veikia korektiškai ir pasirinkus atitinkamą meniu punktą atveriamas korektiškas funkcijos valdymo langas.

lent. 6.1 Meniu pagrindinis scenarijus

Nr.	Įėjimas	Laukiamas rezultatas
6.2.1 - 1	Pasirenkamas prisijungimo prie DB meniu punktas	Atveriamas prisijungimo prie DB informacijos langas
6.2.1 - 2	Pasirenkamas šablonų peržiūros meniu punktas	Atveriamas šablonų peržiūros langas
6.2.1 - 3	Pasirenkamas šablonų kūrimo meniu punktas	Atveriamas šablonų įvedimo naudojant elementus langas
6.2.1 - 4	Pasirenkamas šablonų kūrimo naudojant VT meniu punktas	Atveriamas šablonų įvedimo naudojant pavyzdinę VT langas
6.2.1 - 5	Pasirenkamas veiklos taisyklių valdymo meniu punktas	Atveriamas nestruktūrizuotų VT valdymo langas
6.2.1 - 6	Pasirenkamas struktūrizuotos VT įvedimo meniu punktas	Atveriamas struktūrizuotos VT įvedimo langas

Nr.	Įėjimas	Laukiamas rezultatas
6.2.1 - 7	Pasirenkamas terminų valdymo meniu punktas	Atveriamas terminų valdymo langas
6.2.1 - 8	Pasirenkamas aktorių valdymo meniu punktas	Atveriamas aktorių valdymo langas
6.2.1 - 9	Pasirenkamas rezervuotų žodžių valdymo meniu punktas	Atveriamas rezervuotų žodžių valdymo langas
6.2.1 - 10	Pasirenkamas veiksmų valdymo meniu punktas	Atveriamas veiksmų valdymo langas
6.2.1 - 11	Pasirenkamas VT grupių valdymo meniu punktas	Atveriamas VT grupių valdymo langas

6.2.2. Šablono įvedimo iš elementų

Šablonų įvedimas iš elementų suteikia galimybę greitai įvesti naujo šablono struktūrą. Tačiau tokiu būdu įvedinėjant šabloną kyla reali grėsmė suklysti, tad būtina, kad veiktų elementų šalinimas. Taip pat svarbu užtikrinti šablono tipo įvedimą.

lent. 6.2 Šablono įvedimo iš elementų pagrindinis scenarijus

Nr.	Įėjimas	Laukiamas rezultatas
6.2.2 - 1	Paspaudžiamas norimas šablono elementą atvaizduojantis nuorodos tipo meniu punktas	Elementas atvaizduojamas šablono sudarinėjimo formoje
6.2.2 - 2	Paspaudžiamas elementas atvaizduotas šablono sudarinėjimo formoje	Paspaustas elementas pašalinamas iš formos
6.2.2 - 3	Baigus sudaryti šabloną ir pasirinkus jo tipą spaudžiamas mygtukas „išsaugoti“	Šablonas išsaugomas ir parodomas informacinis pranešimas
6.2.2 - 4	Paspaudžiamas mygtukas „Išvalyti“	Išvaloma šablono sudarinėjimo forma

lent. 6.3 Šablono įvedimo iš elementų kiti scenarijai

Nr.	Įėjimas	Laukiamas rezultatas
6.2.2 - 5	Neįvedus nei vieno šablono elemento į formą spaudžiamas mygtukas „išsaugoti“	Išvedamas pranešimas informuojantis apie tai, jog šablonas

Nr.	Įėjimas	Laukiamas rezultatas
		nesuformuotas
6.2.2 - 6	Nepasirinkus šablono tipo paspaudžiamas mygtukas „išsaugoti“	Išvedamas pranešimas įspėjantis, jog nepasirinktas šablono tipas.
6.2.2 - 7	Šablono struktūroje neesant atidarancio skliausto spaudžiamas uždarymo skliausto elementas	Išvedamas pranešimas įspėjantis, kad šablone nėra atidarancio skliausto
6.2.2 - 8	Spaudžiamas ant mygtukas „Šablonai“	Šablonų įvedimo langas papildomas šablonų peržiūros langu

6.2.3. Šablono įvedimo naudojant VT

Šablono įvedimas naudojant VT suteikia galimybę vartotojui sukurti šabloną pagal sistemoje saugomą ar naujai įvestą VT. Sudarius šabloną pagal nestruktūrizuotą VT suteikiama galimybė išsaugoti struktūrizavimo metu gautą VT.

lent. 6.4 Šablono įvedimo naudojant VT pagrindinis scenarijus

Nr.	Įėjimas	Laukiamas rezultatas
6.2.3 - 1	Pažymėjus VT elementą iš kontekstinio meniu pasirenkamas šablono elemento tipas	Elementas atvaizduojamas šablono sudarinėjimo formoje
6.2.3 - 2	Pažymėjus VT elementą paspaudžiama trumpoji klavišų kombinacija atitinkanti tam tikrą elemento tipą	Elementas atvaizduojamas šablono sudarinėjimo formoje
6.2.3 - 3	Baigus sudaryti šabloną ir pasirinkus jo tipą spaudžiamas mygtukas „išsaugoti“	Šablonas išsaugomas ir parodomas informacinis pranešimas
6.2.3 - 4	Paspaudžiamas mygtukas „Išvalyti“	Išvaloma šablono sudarinėjimo forma
6.2.3 - 5	Išsaugojus šabloną spaudžiamas mygtukas „Išsaugoti taisyklę“	Atveriamas struktūrizuotos VT informacijos įvedimo langas

lent. 6.5 Šablono įvedimo naudojant VT kiti scenarijai

Nr.	Įėjimas	Laukiamas rezultatas
6.2.3 - 6	Nepažymėjus VT elemento iš kontekstinio meniu pasirenkamas šablono elemento tipas	Išvedamas pranešimas, jog nepažymėtas VT elementas
6.2.3 - 7	Nepažymėjus VT elemento paspaudžiama trumpoji klavišų kombinacija atitinkanti tam tikrą elemento tipą	Išvedamas pranešimas, jog nepažymėtas VT elementas
6.2.3 - 8	Neįvedus nei vieno šablono elemento į formą spaudžiamas mygtukas „išsaugoti“	Išvedamas pranešimas informuojantis apie tai, jog šablonas nesuformuotas
6.2.3 - 9	Nepasirinkus šablono tipo paspaudžiamas mygtukas „išsaugoti“	Išvedamas pranešimas įspėjantis, jog nepasirinktas šablono tipas.
6.2.3 - 10	Spaudžiamas ant mygtukas „Šablonai“	Šablonų įvedimo langas papildomas šablonų peržiūros langu
6.2.3 - 11	Neįvedus nei vieno šablono elemento į formą spaudžiamas mygtukas „Išsaugoti taisyklę“	Išvedamas pranešimas informuojantis apie tai, jog nesuformuota VT.

6.2.4. Nestruktūrizuotų VT valdymo

Nestruktūrizuotos VT yra pirminė informacija kuri vėliau struktūrizuojama pagal šablonus. Nestruktūrizuotų VT valdymą sudaro jų įvedimas, trynimasis, peržiūrėjimas, redagavimas, susiejimas su kitomis nestruktūrizuotomis VT. Būtina užtikrinti, kad su struktūrizuotomis VT susijusios nestruktūrizuotos VT nebūtų redaguojamos ar pašalinamos iš VTS.

lent. 6.6 Nestruktūrizuotos VT įvedimo pagrindinis scenarijus

Nr.	Įėjimas	Laukiamas rezultatas
6.2.4 - 1	Atidaromas nestruktūrizuotų VT valdymo langas.	Pateikiamas visų VTS esančių nestruktūrizuotų VT sąrašas.
6.2.4 - 2	Paspaudžiamas mygtukas „Sukurti“.	Atveriamas naujos nestruktūrizuotos VT įvedimo

Nr.	Įėjimas	Laukiamas rezultatas
		langas.
6.2.4 - 3	Paspaudžiamas mygtukas „Susijusios>>“.	Langas papildomas su nestruktūrizuota VT susijusių VT sąrašu.
6.2.4 - 4	Spragtelimas mygtuką „Susieti“.	Atveriamas su pažymėta nestruktūrizuota VT galimų susieti VT sąrašas.
6.2.4 - 5	Lange papildytam su nestruktūrizuota VT susijusių VT sąrašu pažymima kita VT.	Susijusių VT sąrašas atnaujinamas parodant su pasirinkta VT susijusias VT.
6.2.4 - 6	Atveriamas nestruktūrizuotų VT kūrimo langas	Sisteminio vardo lauke pateikiamas sugeneruotas vardas
6.2.4 - 7	Spragtelimas mygtukas „Naikinti“	Iš VTS pašalinama pažymėta įrašą atitinkanti nestruktūrizuota VT, VTS saugomų VT sąrašas atnaujinamas.
6.2.4 - 8	Spragtelimas mygtukas „Kurti šabloną“	Atveriamas šablonų kūrimo pagal nestruktūrizuotą VT langas
6.2.4 - 9	Spragtelimas mygtuką „Struktūrizuoti“	Atveriamas struktūrizuotų VT kūrimo langas.
6.2.4 - 10	VTS saugomų nestruktūrizuotų VT sąrašė pažymima norima VT	Pateikiamas pasirinktos VT apibūdinimas.
6.2.4 - 11	Spragtelimas mygtuką „Baigti“	Užveriamas nestruktūrizuotų VT valdymo langas

lent. 6.7 Nestruktūrizuotų VT valdymo kiti scenarijai

Nr.	Įėjimas	Laukiamas rezultatas
6.2.4 - 12	Nepasirinkus VT autoriaus paspaudžiamas mygtukas „Išsaugoti“	Išvedamas pranešimas, jog taisyklė išsaugota
6.2.4 - 13	Neįvedus VT spaudžiamas mygtukas „Išsaugoti“	Išvedamas pranešimas, jog neįvesta veiklos taisyklė
6.2.4 - 14	Neįvedus nestruktūrizuotos VT pavadinimo spaudžiamas mygtukas „Išsaugoti“	Išvedamas pranešimas, jog neįvestas taisyklės pavadinimas

Nr.	Įėjimas	Laukiamas rezultatas
6.2.4 - 15	Neįvedus nestructūrizuotos VT apibūdinimo spaudžiamas mygtukas „Išsaugoti“	Išvedamas pranešimas, jog taisyklė išsaugota
6.2.4 - 16	Naujos nestructūrizuotos VT įvedimo lange spragtelimas mygtukas „Baigti“	Užveriamas naujos VT įvedimo langas
6.2.4 - 17	Susiejamų nestructūrizuotų VT lange spaudžiamas mygtukas „Baigti“	Užveriamas susietinų VT langas.

6.2.5. Struktūrizuotos VT įvedimo

Struktūrizuotos VT įvedimas viena pagrindinių sistemos funkcijų. Jos metu nestructūrizuota VT pagal nustatytą šablono struktūrą įvedama į sistemą. Svarbu užtikrinti, kad struktūrizuojant VT šablono struktūra būtų pilnai išpildyta. Taip pat būtina patikrinti ar įvesta visa reikiama papildoma informacija. Struktūrizuotai VT taip pat galima nustatyti sąlygas kurios turi būti patenkintos norint ją inicijuoti.

lent. 6.8 *Struktūrizuotos VT įvedimo pagrindinis scenarijus*

Nr.	Įėjimas	Laukiamas rezultatas
6.2.5 - 1	VT struktūrizavimo langas atveriamas iš šablonų sąrašo pasirinkus norimą šabloną	Atvertam VT struktūrizavimo lange pateikiama pasirinkto šablono struktūra
6.2.5 - 2	VT struktūrizavimo langas atveriamas pasirinkus meniu punktą „Struktūrizuotos VT“	Atveriamas jokia informacija neužpildytas VT struktūrizavimo langas
6.2.5 - 3	Spragtelimas mygtukas „Nestructūrizuota VT“	Atveriamas langas su VTS saugomų nestructūrizuotų VT sąrašu
6.2.5 - 4	Spragtelimas mygtukas „Pasirinkti šabloną“	Atveriamas VTS saugomų šablonų sąrašas
6.2.5 - 5	Struktūrizuotos VT lauke įvedinėjamas elementas	Šablono struktūroje pažymimas įvedinėjamas elementas, sistemoje esančių reikšmių lauke pateikiamos galimo reikšmės

Nr.	Įėjimas	Laukiamas rezultatas
6.2.5 - 6	Įvestas VT elementas egzistuoja VTS	Įvestas elementas nuspalvinamas tam elementui būdinga spalva.
6.2.5 - 7	Įvestas VT elementas neegzistuoja VTS	Įvesto elemento fonas pažymimas raudonai
6.2.5 - 8	Užpildyta visa šabloną sudaranti struktūra	Parodomas pranešimas informuojantis apie šablono užpildymą
6.2.5 - 9	Spaudžiamas mygtukas „Nustatyti VT informaciją“	Atveriamas VT informacijos nustatymo langas
6.2.5 - 10	Spaudžiamas mygtukas „Prieš/Po sąlygos“	Atveriamas Prieš/Po sąlygų nustatymo langas.
6.2.5 - 11	Nustačius VT informaciją VT informacijos lange spaudžiamas mygtukas „Patvirtinti“	Nustatyta VT informacija atvaizduojama VT struktūrizavimo lange.
6.2.5 - 12	Užpildžius šablono struktūrą, bei nustačius visą reikiama VT informaciją spaudžiamas mygtukas „Išsaugoti“	Struktūrizuota VT išsaugoma VTS. Išvedamas pranešimas, kad VT išsaugota sėkmingai.
6.2.5 - 13	Paspaudžiamas mygtukas „Atšaukti“	Išvalomas VT struktūrizavimo lango komponentai

lent. 6.9 *Struktūrizuotos VT įvedimo kiti scenarijai*

Nr.	Įėjimas	Laukiamas rezultatas
6.2.5 - 14	Nebaigus pildyti šablono struktūros spaudžiamas mygtukas „Išsaugoti“	Išvedamas pranešimas, kad ne visas šablonas užpildytas
6.2.5 - 15	Neužpildžius papildomos informacijos spaudžiamas mygtukas „Išsaugoti“	Išvedamas pranešimas, jog neužpildyta papildoma VT informacija, atveriamas papildomos informacijos langas
6.2.5 - 16	Specifikuojamoje VT esant elementams su rausvu fonu spaudžiamas mygtukas „Išsaugoti“	Išvedamas pranešimas informuojantis kad VT sudaro VTS nesaugomi elementai
6.2.5 - 17	Nepažymėjus VT elemento su rausvu fonu iš kontekstinio meniu pasirenkama „išsaugoti“.	Išvedamas pranešimas, jog nepasirinktas elementas

Nr.	Įėjimas	Laukiamas rezultatas
6.2.5 - 18	Pažymėjus elementą be rausvo fono iš kontekstinio meniu pasirenkama „išsaugoti“	Išvedamas pranešimas, jog toks elementas sistemoje jau egzistuoja.
6.2.5 - 19	Nepasirinkus nestruktūrizuotos VT spaudžiamas mygtukas „Išsaugoti“	Išvedamas pranešimas, jog nenustatyta nestruktūrizuota VT
6.2.5 - 20	Nepasirinkus VT grupės spaudžiamas mygtukas „Išsaugoti“	Struktūrizuota VT išsaugoma VTS. Išvedamas pranešimas, kad VT išsaugota sėkmingai.
6.2.5 - 21	Neįvedus jokios informacijos spaudžiamas mygtukas „Išsaugoti“	Išvedamas pranešimas, jog būtina nustatyti reikiamą informaciją.

6.2.6. Terminų valdymo

Terminai vienas iš VT elementų. Jie yra dažniausiai naudojami, bei gali turėti sinonimų. Būtina užtikrinti, kad susiejant du terminus jų sinonimai taptų bendrais. Terminų valdymo forma apima jų peržiūrėjimą, įterpimą, redagavimą, naikinimą, bei susiejimą su sinonimais.

lent. 6.10 Terminų valdymo pagrindinis scenarijus

Nr.	Įėjimas	Laukiamas rezultatas
6.2.6 - 1	Pasirenkamas meniu punktas „Terminų žodynas“	Atveriamas terminų valdymo langas kuriame pateikiamas visų VTS saugomų terminų sąrašas
6.2.6 - 2	Spaudžiamas mygtukas „Sinonimai>>“	Terminų valdymo langas papildomas sinonimų langu.
6.2.6 - 3	Sinonimų sąrašu papildytame terminų valdymo lange pažymimas norimas terminas	Sinonimų sąrašė atvaizduojami termino sinonimai.
6.2.6 - 4	VTS saugomų terminų sąrašė pasirenkamas norimas terminas	Termino apibūdinimas pateikiamas apibūdinimo lauke.
6.2.6 - 5	Spaudžiamas mygtukas „Pridėti terminą“	Atveriamas terminų įvedimo langas

Nr.	Įėjimas	Laukiamas rezultatas
6.2.6 - 6	Terminų įvedimo lange įvedus reikiamą informaciją spaudžiamas mygtukas „Pridėti“	Terminas įterpiamas į VTS, išvedamas pranešimas, jog terminas sėkmingai išsaugotas
6.2.6 - 7	Terminų įvedimo lange spaudžiamas mygtukas „Baigti“	Užveriamas terminų įvedimo langas.
6.2.6 - 8	Spaudžiamas mygtukas „Pridėti sinonimą“	Atveriamas langas su galimų sinonimų sąrašu.
6.2.6 - 9	Iš galimų sinonimų sąrašo pasirinktas įrašas spragtelimas pele du kartus	Pasirinktas sinonimas pridedamas prie sinonimų sąrašo
6.2.6 - 10	Galimų sinonimų lange spragtelimas mygtukas „Baigti“	Užveriamas galimų sinonimų langas
6.2.6 - 11	Terminų sąrašė pasirenkamas norimas įrašas ir spaudžiamas mygtukas „Naikinti“	Iš VTS pašalinamas pasirinktą įrašą atitinkantis terminas
6.2.6 - 12	Terminų sąrašė pasirinkus norimą įrašą spaudžiamas mygtukas modifikuoti	Atveriamas langas su termino informaciją ir galimybe ją keisti
6.2.6 - 13	Terminų valdymo lange spaudžiamas mygtukas „Baigti“	Užveriamas terminų įvedimo langas

lent. 6.11 *Terminų valdymo kiti scenarijai*

Nr.	Įėjimas	Laukiamas rezultatas
6.2.6 - 14	Terminų įvedimo lange neįvedus termino spaudžiamas mygtukas „Pridėti“	Išvedamas, jog neįvesta termino išraiška
6.2.6 - 15	Terminų įvedimo lange neįvedus termino apibūdinimo spaudžiamas mygtukas „Pridėti“	Išvedamas pranešimas jog terminas išsaugotas sėkmingai

6.2.7. Rezervuotų žodžių valdymo

Rezervuotų žodžių valdymo forma apima jų įterpimą, redagavimą, naikinimą bei peržiūrėjimą. Rezervuoti žodžiai skirstomi į tipus. Žodžiai kuriems priskirtas tipas „Priklausomybė“ dar turi ir ryšio tipą. Visos rezervuotų žodžių valdymo funkcijos atliekamos tame pačiame lange.

lent. 6.12 *Rezervuotų žodžių valdymo pagrindinis scenarijus*

Nr.	Įėjimas	Laukiamas rezultatas
6.2.7 - 1	Menu pasirenkamas punktas „Rezervuoti žodžiai“	Atveriamas rezervuotų žodžių valdymo langas, kuriame pateikiamas VTS saugomų rezervuotų žodžių sąrašas
6.2.7 - 2	Iš tipų sąrašo pasirenkamas norimas rezervuoto žodžio tipas	Tipų sąrašė atvaizduojami pasirinktą tipą atitinkantys rezervuoti žodžiai
6.2.7 - 3	Iš tipų sąrašo pasirenkamas rezervuoto žodžio tipas „Priklausomybė“	Langas papildomas ryšio tipų sąrašu
6.2.7 - 4	Paspaudžiamas mygtukas „Baigti“	Užveriamas rezervuotų žodžių valdymo langas
6.2.7 - 5	Įvedus rezervuotą žodį ir nustačius jo tipą spaudžiamas mygtukas „Pridėti“	Rezervuotas žodis įterpiamas į VTS ir atvaizduojamas sąrašė.
6.2.7 - 6	Pasirinkus norimą rezervuotą žodį atitinkantį įrašą spaudžiamas mygtukas „Naikinti“	Pateiktame sąrašė pažymėtas rezervuotas žodis ištrinamas
6.2.7 - 7	Pasirinkus norimą rezervuotą žodį atitinkantį įrašą spaudžiamas mygtukas „Redaguoti“	Pasirinkto įrašo informacija atvaizduojama komponentuose
6.2.7 - 8	Paspaudžiamas mygtukas „Baigti“	Užveriamas rezervuotų žodžių valdymo langas

lent. 6.13 *Rezervuotų žodžių valdymo kiti scenarijai*

Nr.	Įėjimas	Laukiamas rezultatas
6.2.7 - 9	Neįvedus rezervuoto žodžio spaudžiamas mygtukas „Pridėti“	Išvedamas pranešimas, jog neįvestas rezervuotas žodis
6.2.7 - 10	Neįvedus rezervuoto žodžio apibūdinimo	Rezervuotas žodis įterpiamas į

Nr.	Įėjimas	Laukiamas rezultatas
	spaudžiamas mygtukas „Pridėti“	VTS ir atvaizduojamas sąrašė.
6.2.7 - 11	Nepasirinkus rezervuoto žodžio tipo spaudžiamas mygtukas „Pridėti“	Išvedamas pranešimas, jog nepasirinktas rezervuoto žodžio tipas
6.2.7 - 12	Pasirinkus rezervuoto žodžio tipą „Priklausomybė“, bet nepasirinkus ryšio tipo spaudžiamas mygtukas „Pridėti“	Išvedamas pranešimas, jog nepasirinktas rezervuoto žodžio ryšio tipas

6.2.8. Struktūrizuotų VT peržiūros

Struktūrizuotų VT peržiūra suteikia galimybę peržvelgti VTS saugomų VT informaciją. Šiame lange taip pat galima pasirinkti VT sąrašo filtravimo kriterijų. Pasirinkus norimą struktūrizuotą VT galima patekti į jos redagavimo langą, taip pat galima ištrinti bei perrikiuoti atvaizduojamas VT.

lent. 6.14 *Struktūrizuotų VT peržiūros pagrindinis scenarijus*

Nr.	Įėjimas	Laukiamas rezultatas
6.2.8 - 1	Pasirenkamas meniu punktas „VT peržiūra“	Atveriamas struktūrizuotų VT peržiūrėjimo langas
6.2.8 - 2	Pažymėjus norimą VT spaudžiamas mygtukas „Redaguoti“	Atveriamas VT redagavimo langas
6.2.8 - 3	Pažymėjus norimą VT spaudžiamas mygtukas „Naikinti“	Pasirinkta VT pašalinama iš VTS
6.2.8 - 4	Pasirenkamas norimas paieškos kriterijus ir įvedus paieškos raktą spaudžiamas mygtukas „Filtruoti“	Sąrašė atvaizduojamos paieškos filtrą atitinkančios VT

lent. 6.15 *Struktūrizuotų VT peržiūros kiti scenarijai*

Nr.	Įėjimas	Laukiamas rezultatas
6.2.8 - 5	Pasirenkamas peržiūros režimas	Suformuojamas atitinkamą režimą realizuojantis meniu
6.2.8 - 6	Pasirenkami peržiūros kriterijai ir	Išvedami įrašai atitinkantys

Nr.	Įėjimas	Laukiamas rezultatas
	spaudžiamas mygtukas „Peržiūrėti“	nustatyta filtrą.
6.2.8 - 7	Pažymėjus norimą įrašą iš kontekstinio meniu pasirenkamas redagavimo režimas	Atveriamas redagavimo langas
6.2.8 - 8	Paspaudžiamas mygtukas „Atšaukti“	Išvalomi filtro laukai

6.2.9. DB prisijungimo nustatymo

Duomenų bazės prisijungimo informacijos nustatymas labai svarbi funkcija. Prisijungimo duomenys nustatomi pirmo sistemos paleidimo metu. Jie taip pat gali būti koreguojami keičiantis VTS saugančio serverio informacijai.

lent. 6.16 *DB prisijungimo nustatymo pagrindinis scenarijus*

Nr.	Įėjimas	Laukiamas rezultatas
6.2.9 - 1	Pirmą kartą paleidžiama VTVS sistema	Atveriamas DB prisijungimo langas
6.2.9 - 2	Pasirenkamas meniu punktas „DB“	Atveriamas DB prisijungimo langas su nustatyta prisijungimo informacija
6.2.9 - 3	Nustačius visą reikiamą prisijungimo informaciją spaudžiamas mygtukas „Patikrinti ryšį“	Išvedamas pranešimas, jog prisijungimas pavyko
6.2.9 - 4	Nustačius visą norimą prisijungimo informaciją spaudžiamas mygtukas „Išsaugoti pakeitimus ir išeiti“	Uždaromas DB informacijos langas ir nustatyta informacija įrašoma į txt formato failą
6.2.9 - 5	Pažymėjus punktą „Nerodyti kitą kartą paleidžiant sistemą“ spaudžiamas mygtukas „Išsaugoti pakeitimus ir išeiti“	Uždaromas DB prisijungimo informacijos langas, jis nerodomas kito sistemos paleidimo metu.

lent. 6.17 DB prisijungimo kiti scenarijai

Nr.	Įėjimas	Laukiamas rezultatas
6.2.9 - 6	Nenustačius reikiamos prisijungimo informacijos spaudžiamas mygtukas „Patikrinti ryšį“	Išvedamas pranešimas, jog trūksta prisijungimo informacijos
6.2.9 - 7	Nenustačius reikiamos prisijungimo informacijos spaudžiamas mygtukas „Išsaugoti pakeitimus ir išeiti“	Išvedamas pranešimas, jog trūksta prisijungimo informacijos

6.2.10. Aktorių valdymo

Aktorių valdymas apima aktorių: įterpimą, redagavimą, peržiūrą bei naikinimą. Visi aktorių valdymo veiksmai atliekami viename lange. Analogiškai vykdomos VT tipų, VT grupių, Įvykių bei Veiksmų valdymo funkcijos.

lent. 6.18 Aktorių valdymo pagrindinis scenarijus

Nr.	Įėjimas	Laukiamas rezultatas
6.2.10 - 1	Pasirenkamas meniu punktas „Aktoriai“	Atveriamas aktorių valdymo langas su VTS saugomais įrašais
6.2.10 - 2	Įvedus reikiamą informaciją spaudžiamas mygtukas „Pridėti“.	Išvedamas pranešimas, jog įterpta sėkmingai
6.2.10 - 3	Pasirinkus norimą įrašą spaudžiamas mygtukas „Naikinti“	Išvedamas pranešimas, jog pašalintas pasirinktas įrašas
6.2.10 - 4	Pasirinkus norimą įrašą spaudžiamas mygtukas „Redaguoti“	Pasirinkto įrašo informacija pateikiama atitinkamuose laukuose.
6.2.10 - 5	Spaudžiamas mygtukas „Išvalyti“	Išvalomi aktoriaus informacijos laukai, sugeneruojamas naujas sisteminis vardas.
6.2.10 - 6	Spaudžiamas mygtukas „Baigti“	Užveriamas aktorių valdymo langas

lent. 6.19 Aktorių valdymo kiti scenarijai

Nr.	Įėjimas	Laukiamas rezultatas
6.2.10 - 7	Neįvedus aktorius spaudžiamas mygtukas „Pridėti“.	Išvedamas pranešimas, jog neįvestas aktorius
6.2.10 - 8	Neįvedus apibūdinimo spaudžiamas mygtukas „Pridėti“.	Išvedamas pranešimas, jog įterpta sėkmingai

6.3 Sistemos testavimo protokolas

Kadangi sistema buvo kuriama iteraciniu būdu, tai iteracijos metu realizuotų funkcijų testavimas buvo atliekamas iteracijos pabaigoje. Galutinės sistemos realizacijos testavimas buvo atliekamas naudojant sudarytus testavimo scenarijus. Pirmiausia buvo testuojama naudojant pagrindinį scenarijų, o vėliau išnagrinėjami likę scenarijai.

lent. 6.20 Sistemos testavimo rezultatai

Scenarijaus Nr.		Įvykdymo būseną	Atlikti pakeitimai
Nuo	Iki		
6.2.1-1	6.2.1-1	Testas atliktas sėkmingai	Pakeitimų neprireikė
6.2.2-1	6.2.2-4	Testas atliktas sėkmingai	Pakeitimų neprireikė
6.2.2-5	6.2.2-5	Atliekant testavimą aptikta klaida, išsaugotas tuščias šablonas	Šablono išsaugojimo funkcija papildyta šablono ilgio patikros procedūra.
6.2.2-6	6.2.2-8	Testas atliktas sėkmingai	Pakeitimų neprireikė
6.2.3-1	6.2.3-5	Testas atliktas sėkmingai	Pakeitimų neprireikė
6.2.3-6	6.2.3-7	Atliekant testavimą aptikta klaida, įterptas tuščias elementas	Šablono sudarymo funkcija papildyta patikros procedūra.
6.2.3-7	6.2.3-11	Testas atliktas sėkmingai	Pakeitimų neprireikė
6.2.4-1	6.2.4-11	Testas atliktas sėkmingai	Pakeitimų neprireikė
6.2.4-12	6.2.4-17	Testas atliktas sėkmingai	Pakeitimų neprireikė
6.2.5-1	6.2.4-13	Testas atliktas sėkmingai	Pakeitimų neprireikė
6.2.4-13	6.2.4-16	Testas atliktas sėkmingai	Pakeitimų neprireikė

Scenarijaus Nr.			
6.2.5-17	6.2.4-17	Atliekant testavimą aptikta klaida, įterptas tuščias elementas	VT struktūrizavimo funkcija papildyta elemento patikros procedūra.
6.2.5-18	6.2.4-21	Testas atliktas sėkmingai	Pakeitimų neprireikė
6.2.6-1	6.2.6-13	Testas atliktas sėkmingai	Pakeitimų neprireikė
6.2.6-14	6.2.6-14	Atliekant testavimą aptikta klaida, įterptas tuščias termino elementas	Terminų įvedimo funkcija papildyta patikros procedūra.
6.2.6-15	6.2.6-15	Testas atliktas sėkmingai	Pakeitimų neprireikė
6.2.7-1	6.2.7-8	Testas atliktas sėkmingai	Pakeitimų neprireikė
6.2.7-9	6.2.7-9	Atliekant testavimą aptikta klaida, įterptas tuščias rezervuotas žodis	Rezervuoto žodžio įvedimo funkcija papildyta patikros procedūra.
6.2.7-10	6.2.7-12	Testas atliktas sėkmingai	Pakeitimų neprireikė
6.2.8-1	6.2.8-8	Testas atliktas sėkmingai	Pakeitimų neprireikė
6.2.9-1	6.2.9-7	Testas atliktas sėkmingai	Pakeitimų neprireikė
6.2.10-7	6.2.10-7	Atliekant testavimą aptikta klaida, įterptas tuščias įrašas	Aktoriaus įvedimo funkcija papildyta patikros procedūra.
6.2.10-8	6.2.10-8	Testas atliktas sėkmingai	Pakeitimų neprireikė

6.4 Testavimo apibendrinimas

Testavimo metu sudaryti šablonai:

1. <Terminas><Rezervuotas žodis><Terminas> - fakto šablonas
2. <Faktas><Rezervuotas žodis>{ [Rezervuotas žodis] }(< Faktas>|<Reikšmė>) – apribojimo šablonas
3. <Rezervuotas žodis><Faktas ><Rezervuotas žodis >(<Faktas>|<Reikšmė>)[{ < Rezervuotas žodis > <Faktas><Rezervuotas žodis>(<Faktas>|<Reikšmė>)}]<Rezervuotas žodis><Faktas><Rezervuotas žodis>(<Faktas>| <Reikšmė>) – išvesties taisyklės šablonas
4. <Faktas><Rezervuotas žodis>(<Faktas>|<Reikšmė>){[<Rezervuotas žodis>(<Faktas>| <Reikšmė>)]} – skaičiavimo taisyklės šablonas

Testavimo metu įvestos VT:

1. Darbuotojas privalo turėti Identifikatorių – faktas
2. Darbuotojo Identifikatorius yra unikalus - apribojimas
3. Darbuotojo Atlyginimas yra didesnis už Darbuotojo Nustatytą minimalų atlyginimą – apribojimas
4. JEI Vartotojo Kredito limitas yra didesnis už Vartotojo Užsakymo sumą, TAI Vartotojo Užsakymo būseną = priimtas – išvesties taisyklė
5. Prekės Mažmeninė kaina = Prekės Didmeninė kaina + Prekės Antkainis – skaičiavimo taisyklė

Kiekvienos iteracijos testavimas sistemos kūrimo metu buvo atliekamas sistemos kūrėjo. Galutinis sistemos testavimas naudojant sudarytus scenarijus atliktas sistemos kūrėjo ir vadovo. Sistemoje aptikta keletas nedidelių klaidų. Didžioji dalis klaidų suteikė galimybę vartotojui į VTS įvesti tuščią ar nepilnai suformuotą įrašą. Aptiktos klaidos pašalintos klaidingai veikusias sistemos funkcijas papildžius tikrinimo procedūromis informuojančiomis vartotoją apie informacijos trūkumą ir užkertančiomis kelią blogai suformuoto įrašo įterpimui.

7 IŠVADOS

1. Atlikta veiklos taisyklių specifikuojimo būdų, naudojant DSL, grafines notacijas bei šablonus, analizė parodė, kad taisyklių specifikuojimas, naudojant natūralios kalbos pagrindu sudarytus šablonus, teikia bene daugiausiai privalumų: suprantamumas, glaudesnio ryšio su natūralia verslo aplinka (per toje aplinkoje naudojamus terminų žodynus) užtikrinimas, greitas vartotojų apmokymo procesas ir paprastumas naudoti, galimybė keisti šablonų konstrukcijas ir kt.
2. Atlikus veiklos taisyklių valdymo sistemų analizę nustatyta, kad dauguma jų taisyklių specifikuojimui naudoja specifines DSL kalbas, kurios yra išvestos iš objektinių aukšto lygio programavimo kalbų (pvz., Java). Natūralios kalbos pagrindu sukurtų šablonų taikymas veiklos taisyklių specifikuojimui komerciniuose įrankiuose vis dar išlieka komplikotas. Be to, yra naudojama išimtinai anglų kalba, kas nėra visiškai tinkama Lietuvoje (ar be kurioje kitoje šalyje, kur anglų kalba nėra nacionalinė).
3. Darbe pristatytas sukurtos veiklos taisyklių saugyklos valdymo sistemos projektas, kuriame specifikuotos visos pagrindinės sistemos funkcijos, užtikrinančios veiklos taisyklių saugyklos palaikymą. Sistemos funkcionalumas buvo nustatytas, įvertinus išanalizuotas veiklos taisyklių valdymo sistemas, šablonais grindžiamo taisyklių užrašymo specifika bei konkrečius KTU Informacinių sistemų katedros mokslo grupės, vykdančios tyrimus veiklos taisyklių srityje, reikalavimus.
4. Veiklos taisyklių saugyklos valdymo sistema buvo kuriama iteraciškai, atliekant sukurtų posistemų testavimą, tikslinant sistemos projektinę specifikaciją bei papildant sistemą nauju funkcionalumu. Sistemos kūrimo metu taip pat buvo pateikta keletas rekomendacijų, kaip patobulinti KTU ISK mokslo grupėje jau sukurtus projektinius sprendimus (pvz., VT saugyklos loginę DB schemą).
5. Realizuotoji sistema buvo patikrinta darbo metu sukurtų testavimo scenarijų pagrindu. Testavimo metu kritinių klaidų neaptikta.
6. Sukurtoji veiklos taisyklių saugyklos valdymo sistema prisideda plėtojant KTU Informacijos sistemos katedroje vykdomus mokslinius tyrimus veiklos taisyklių srityje.

8 LITERATŪRA

1. Hutt, A.T.F. Object Analysis and Design, Reference model. OMG Object Analysis and Design Special Interest Group, Draft 7.0, 1992.
2. Hay, D.; Healy, K.A. Defining Business Rules – What Are They Really?. The Business Rules Group, Final Report, 2000.
3. Ross, R.G.; Lam, G.S.W. Capturing Business Rules. Workbook for public seminar, Boston, MA, 2000.
4. H-E. Eriksson, M. Penker. Business Modelling with UML: Business Patterns at Work. J. Wiley & Sons, 2000
5. Business Rules Group. Business Rules Manifesto, the principles of rule independence 2003.
6. Daniela Rosca, John D'Attilio. Business Rules Specification, Enforcement and Distribution for Heterogeneous Environments, 2001
7. N. Haggerty, Modeling business rules using the UML and CASE. Business Rules Journal, 2000.
8. H. Herbst, G. Knolmayer, T. Myrach and M. Schlesinger. The Specification of Business Rules: A Comparison of Selected Methodologies. 1994.
9. Ross, R. G. and Gladys S. W. Lam. BRS RuleSpeak: The Do's and Don'ts of Expressing Business Rules. Business Rule Solutions, 2001.
<http://www.BRCommunity.com>
10. J. Reeder, Templates for Capturing Business Rules, 2003
<http://www.brcommunity.com/cgi-bin/x.pl/print/p-b056.html>
11. Ronald G. Ross and Gladys S. W. Lam. Rule Speak Sentence Templates: Developing Rule Statements Using Sentence Patterns. 2001
12. Wan M.N. Wan-Kadir, Pericles Loucopoulos. Relating Evolving Business Rules to Software Design. Software Engineering Research and Practice. 2003.
13. Ian Graham. Service Oriented Business Rules Management Systems, 2005
http://www.trireme.com/whitepapers/Business%20rules/Trireme_Report_Service_Oriented_Business_Rules_Management_Systems_Ver2b.pdf
14. JBoss Rules VT valdymo įrankio vartotojo dokumentacija, 2007
<http://labs.jboss.com/drools/documentation.html>
15. N. Alex Rupp. Introduction to the Drools rules engine, 2004
<http://www.theserverside.com/articles/article.tss?!=Drools>

16. Mandarax ir ORYX įrankių sudarančių taisyklių valdymo platformą apibūdinimas, 2007 <http://www-ist.massey.ac.nz/JBDietrich/presentations/mandarax-ebcr2004.pdf>
17. Oracle business rules VT valdymo įrankio techninė dokumentacija, 2007 http://www.oracle.com/technology/products/ias/business_rules/pdf/oraclebusinessrulestechnicalwhitepaper.pdf
18. Visual Rules. Veiklos taisyklių grafinio modeliavimo įrankio dokumentacija, 2007 http://www.visual-rules.de/en/pdf_en/en_visual_rules.pdf
19. YASU Technologies. YASU .NET ir YASU J2EE produktų specifikacijos, 2007 http://www.yasutech.com/products/quickrulesse/fact_sheet.htm
20. ILOG. ILOG Rules, ILOG JRules, ILOG for .NET produktu specifikacija, 2007 <http://www.ilog.com/products/businessrules/overview.cfm>
21. Versata inc. Versata 6 BRMS produkto specifikacija, 2007 http://www.versata.com/solutions/v6_brms.htm
22. Fair Isaac Corporation. Blaze Advisor produkto specifikacija, 2007 <http://www.fairisaac.com/Fairisaac/Solutions/Product+Index/Blaze+Advisor/>
23. Uwe Lämmel, Anatoli Beifert, Marcel Brätz, Stefan Brandenburg, Matthias Buse, Christian Höhn, Gert Mannheimer, Michael Rehfeld, Alexander Richter, Stefan Wissuwa. Business Rules Die Wissensverarbeitung erreicht die Betriebswirtschaft Einsatzmöglichkeiten und Marktübersicht 2007.
24. Alan Perkins, Business Rules Are Meta Data, Business Rules Journal, 2002 <http://www.BRCommunity.com/a2002/b097.html>
25. Lina Nemūraitė. Laboratorinių darbų su MagicDraw mokomoji medžiaga, 2007 <ftp://isd.ktu.lt/Isd/Nemuraite/Moduliai/T120M035-2007/LaboratoriniaiDarbaiSuMagicDraw/ISPLab2.doc>
26. Skersys, T.: Computer-aided information systems development method based on the business rules-extended enterprise model. Thesis. Kaunas University of Technology, Kaunas (2006)
27. Object Management Group. Semantics of Business Vocabulary and Business Rules (SBVR) 2006.

9 PRIEDAI

Priedas Nr. 1

Paiškinamasis žodynis

1. **DB** (Data base) – duomenų bazė yra organizuotas duomenų rinkinys, kuriuo galima individualiai naudotis elektroniniu ar kitu būdu.
2. **VT** – veiklos taisyklė.
3. **VTS** – veiklos taisyklių saugykla.
4. **DSL** (Domain specific language) – specifinė srities kalba skirta konkrečioms uždaviniams spręsti.
5. **UML** (Unified modeling language) – Vieninga modeliavimo kalba. Modeliavimo ir specifikacijų kūrimo kalba, skirta specifiuoti, atvaizduoti ir konstruoti objektiškai orientuotų programų dokumentus.
6. **RUP** (Rational unified process) – Racionalus vieningas procesas. Kartotinio programinės įrangos kūrimo metodika.
7. **VTVS** – veiklos taisyklių valdymo sistema.
8. **SBVR** (Semantics of Business Vocabulary and Business Rules) – veiklos žodyno semantika ir veiklos taisyklės.
9. **XML** (eXtensible Markup Language) - W3C rekomenduojama bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba.

Priedas Nr. 2

Paveikslėlių sąrašas

1. pav. 2.1 Įvykių klasifikacija.....	12
2. pav. 2.2 Sąlygų klasifikacija.....	12
3. pav. 2.3 Sąlygų klasifikacija.....	12
4. pav. 2.4 Įrankiai, jų logotipai ir nuorodos.....	14
5. pav. 2.5 Apibendrinta numatomos VTS valdymo sistemos architektūra.....	21
6. pav. 2.6 Sistemos struktūrinė diagrama.....	22
7. pav. 2.7 VT saugyklos valdymo sistemos panaudojimo atvejų modelis.....	23
8. pav. 2.8 Veiklos taisyklių saugyklos loginė duomenų bazės schema [26].....	24
9. pav. 3.1 Nestruktūrizuotos VT įvedimo specifikacija.....	29
10. pav. 3.2 Nestruktūrizuotos VT įvedimo sekų diagrama.....	29
11. pav. 3.3 Struktūrizuotos VT įvedimo specifikacija.....	30
12. pav. 3.4 Struktūrizuotos VT įvedimo sekų diagrama.....	31
13. pav. 3.5 VT modifikavimo specifikacija.....	32
14. pav. 3.6 VT modifikavimo sekų diagrama.....	32
15. pav. 3.7 VT peržiūrėjimo specifikacija.....	33
16. pav. 3.8 VT peržiūrėjimo sekų diagrama.....	33
17. pav. 3.9 VT naikinimo specifikacija.....	34
18. pav. 3.10 VT naikinimo sekų diagrama.....	35
19. pav. 3.11 Šablono kūrimo specifikacija.....	36
20. pav. 3.12 Šablono kūrimo sekų diagrama.....	37
21. pav. 3.13 Šablono modifikavimo specifikacija.....	38
22. pav. 3.14 Šablono modifikavimo sekų diagrama.....	38
23. pav. 3.15 Šablono peržiūrėjimo specifikacija.....	39
24. pav. 3.16 Šablono peržiūrėjimo sekų diagrama.....	39
25. pav. 3.17 Šablono trynimo specifikacija.....	40
26. pav. 3.18 Šablono trynimo sekų diagrama.....	41
27. pav. 3.19 Esių klasių diagrama.....	45
28. pav. 3.20 Robastiškumo diagrama.....	47
29. pav. 4.1 Loginės sistemos architektūros diagrama.....	49
30. pav. 4.2 Vartotojo paslaugų diagrama.....	50
31. pav. 4.3 Veiklos paslaugų diagrama.....	51
32. pav. 4.4 Duomenų paslaugų modelis.....	52
33. pav. 4.5 Veiklos taisyklės įvedimo specifikacija.....	54
34. pav. 4.6 Struktūrizuotos veiklos taisyklės sukūrimo sekų diagrama.....	55
35. pav. 4.7 Struktūrizuotos veiklos taisyklės kūrimo langas.....	56
36. pav. 4.8 Šablono sukūrimo specifikacija.....	58
37. pav. 4.9 Šablono sukūrimo sekų diagrama.....	59
38. pav. 4.10 Šablono kūrimo pagal nestruktūrizuotą VT langas.....	60
39. pav. 4.11 VT elemento būsenų modelis.....	62
40. pav. 4.12 Veiklos taisyklės būsenų modelis.....	62
41. pav. 4.13 Šablono elemento būsenų modelis.....	63
42. pav. 4.14 Šablono būsenų modelis.....	64
43. pav. 5.1 Fizinės duomenų bazės schema.....	67
44. pav. 5.2 Sistemos programinių komponentų architektūros diagrama.....	68
45. pav. 5.3 Sistemos įdiegimo modelis.....	68

Priedas Nr. 3

Lentelių sąrašas

1. lent. 2.1	<i>Grafinių VT specifikuojamų metodų palyginimas [8]</i>	11
2. lent. 2.2	<i>Veiklos taisyklėms specifikuoti naudojamų šablonų tipai [12]</i>	13
3. lent. 2.3	<i>Veiklos taisyklių sistemų palyginimas [23]</i>	18
4. lent. 3.1	<i>Veiklos taisyklių saugyklos esybių klasių ir laukų paaiškinimas [26]</i>	41
5. lent. 6.1	<i>Meniu pagrindinis scenarijus</i>	69
6. lent. 6.2	<i>Šablono įvedimo iš elementų pagrindinis scenarijus</i>	70
7. lent. 6.3	<i>Šablono įvedimo iš elementų kiti scenarijai</i>	70
8. lent. 6.4	<i>Šablono įvedimo naudojant VT pagrindinis scenarijus</i>	71
9. lent. 6.5	<i>Šablono įvedimo naudojant VT kiti scenarijai</i>	72
10. lent. 6.6	<i>Nestruktūrizuotos VT įvedimo pagrindinis scenarijus</i>	72
11. lent. 6.7	<i>Nestruktūrizuotų VT valdymo kiti scenarijai</i>	73
12. lent. 6.8	<i>Struktūrizuotos VT įvedimo pagrindinis scenarijus</i>	74
13. lent. 6.9	<i>Struktūrizuotos VT įvedimo kiti scenarijai</i>	75
14. lent. 6.10	<i>Terminų valdymo pagrindinis scenarijus</i>	76
15. lent. 6.11	<i>Terminų valdymo kiti scenarijai</i>	77
16. lent. 6.12	<i>Rezervuotų žodžių valdymo pagrindinis scenarijus</i>	78
17. lent. 6.13	<i>Rezervuotų žodžių valdymo kiti scenarijai</i>	78
18. lent. 6.14	<i>Struktūrizuotų VT peržiūros pagrindinis scenarijus</i>	79
19. lent. 6.15	<i>Struktūrizuotų VT peržiūros kiti scenarijai</i>	79
20. lent. 6.16	<i>DB prisijungimo nustatymo pagrindinis scenarijus</i>	80
21. lent. 6.17	<i>DB prisijungimo kiti scenarijai</i>	81
22. lent. 6.18	<i>Aktorių valdymo pagrindinis scenarijus</i>	81
23. lent. 6.19	<i>Aktorių valdymo kiti scenarijai</i>	82
24. lent. 6.20	<i>Sistemos testavimo rezultatai</i>	82