

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACINIŲ SISTEMŲ KATEDRA

Rasa Miklovienė

## **Veiklos taisyklių redaktorius**

Magistro darbas

Darbo vadovas:

doc. S. GUDAS

Konsultantas:

Dokt. Tomas Skersys

Kaunas  
2004

# TURINYS

<b>1. ĮVADAS</b> .....	<b>4</b>
<b>2. ANALIZĖS DALIS</b> .....	<b>5</b>
2.1. TYRIMO SRITIS, OBJEKTAS IR PROBLEMA.....	5
2.1.1. Veiklos taisyklės sąvoka.....	5
2.1.2. Veiklos taisyklių realizacija.....	5
2.1.2. Veiklos taisyklių svarba IS kūrime.....	6
2.2. ANALIZĖS METODŲ, PRIEMONIŲ PARINKIMAS.....	7
2.3. ORGANIZACIJOS VEIKLOS ANALIZĖ.....	8
2.3.1. Analizės tikslas.....	8
2.3.2. Veiklos sąveikų modelis.....	9
2.3.3. Veiklos tikslų modelis.....	9
2.3.4. Veiklos panaudojimo atvejų modelis.....	10
2.3.5. Veiklos objektų modelis.....	10
2.3.6. Veiklos procesų modelis.....	11
2.3.7. Vartotojų bei jų poreikių analizė.....	13
2.3.8. Informacijos sistemai keliami nefunkciniai reikalavimai ir apribojimai.....	13
2.4. PASAULIO BEI LIETUVOS LITERATŪROS ŠALTINIUIOSE PATEIKTŲ SPRENDIMŲ PROBLEMAI SPRĖSTI LYGINAMOJI ANALIZĖ.....	14
2.4.1. Atskirų mokslininkų siūlomi sprendimai.....	14
2.4.2. Pasaulyje pateiktų sprendimų problemai spręsti lyginamoji analizė.....	14
2.5. ANALIZĖS IŠVADOS.....	15
<b>3. PROJEKTO DALIS</b> .....	<b>17</b>
3.1. PROJEKTO TIKSLAS IR JO PAGRINDIMAS, KOKYBĖS KRITERIJŲ APIBRĖŽIMAS.....	17
3.2. PROJEKTAVIMO METODŲ, PRIEMONIŲ PARINKIMAS.....	20
3.2.1. Projektavimo metodų analizė.....	20
3.2.2. Metodai, aprašantys veiklos taisykles.....	22
3.3. KOMPIUTERIZUOJAMOS SISTEMOS VARIANTO PARINKIMAS.....	24
3.4. TECHNINĖ UŽDUOTIS.....	25
3.4.1. Magistrinio darbo tema.....	25
3.4.2. Kompiuterizuojamas objektas.....	25
3.4.3. Kompiuterizuojamos funkcijos.....	25
3.4.4. Sistemos naudojimo dokumentai.....	25
3.4.5. Reikalavimai projektavimui, programinei ir techninei įrangai.....	25
3.4.6. Reikalavimai darbo pristatymui.....	25
3.5. REIKALAVIMŲ MODELIS.....	26
3.5.1. Vartotojų panaudojimo atvejų diagramos.....	26
3.5.1.1. Aukšto lygio panaudojimo atvejų diagrama.....	26
3.5.1.2. Detalesnio lygio panaudojimo atvejų diagrama.....	26
3.5.2. Specifikacijos kiekvienam panaudojimo atvejui.....	27
3.5.2.1. Terminų pildymo diagramos.....	27
3.5.2.2. Faktų pildymo diagramos.....	28
3.5.2.3. Natūralios veiklos taisyklės įvedimo diagramos.....	30
3.5.2.4. Veiklos taisyklių formalizavimo diagramos.....	32
3.5.3. Dalykinės srities klasių diagrama.....	36
3.5.4. Vartotojų interfeiso modelis.....	37
3.5.5. Reikalavimai sistemos funkcionalumui.....	38
3.6. SISTEMOS PROJEKTAS.....	38
3.6.1. Projekto tikslas.....	38
3.6.2. Sistemos panaudojimo atvejų ir sekų diagramos.....	39

3.6.3 Duomenų bazės modelis.....	45
3.6.4 Realizacijos modelis.....	48
3.6.5 Paskirstymo diagrama.....	48
3.6.6 Testavimo modelis bei duomenys, kontrolinis pavyzdys.....	49
3.6.7 Veiklos taisyklių redaktoriaus diegimo priemonių planas.....	51
3.6.8 Formalizuotas tiriama metodo, algoritmų ar jų elementų, atspindinčių darbo idėją bei naujumą, aprašas.....	52
3.7 PROJEKTO IŠVADOS.....	59
<b>4. EKSPERIMENTINIS TYRIMAS.....</b>	<b>59</b>
4.1 SUKURTOS SISTEMOS KOKYBĖS TYRIMAS.....	59
4.2 TOLIMESNIO SISTEMOS TOBULINIMO, PLĖTOJIMO GALIMYBĖS.....	61
<b>5. IŠVADOS.....</b>	<b>62</b>
<b>6. LITERATŪRA.....</b>	<b>64</b>
<b>7. TERMINŲ IR SANTRAUKŲ ŽODYNAS.....</b>	<b>65</b>
<b>8. SUMMARY.....</b>	<b>66</b>
<b>9. PRIEDAI.....</b>	<b>67</b>
9.1 PRIEDAS 1.....	68
9.2 PRIEDAS 2. PASAULYJE PATEIKTŲ PROGRAMŲ VEIKLOS TAISYKLIŲ KŪRIMO LANGAI.....	72
9.3 PRIEDAS 3. SISTEMOS VARTOTOJO VADOVAS.....	75

## 1. ĮVADAS

Šiuo metu jau beveik visos privačios ir valstybinės organizacijos yra labai priklausomos nuo programinės įrangos. Dėl joms būdingo veiklos aplinkos dinamiškumo programinės įrangos vystymas ir tobulinimas neišvengiamas dalykas. Veiklos politikos ir pačios veiklos suformuoti pokyčiai perduodami į programinę įrangą. Didelė dalis bendro gyvavimo ciklo kaštų yra paskiriama naujų reikalavimų pateikimui, ir egzistuojančių pašalinimui arba pakeitimui. Daugelis tyrimo projektų bando surasti tinkamiausią būdą kurti lanksčiai prie pokyčių prisitaikančias informacines sistemas. Jie siekia sukurti programinės įrangos modelį arba architektūrą, kuri turėtų galimybę sumažinti pokyčių daromą poveikį, taip pat įtraukti reikalavimus į modelį. Daugelis jų panaudoja jau sukurtą objektiškai orientuotą, paskirstytą sistemų, programinės įrangos architektūros ir komponentinių technologijų privalumus [13].

Veiklos taisyklių metodologijos taikymas kuriant informacijos sistemas (IS) įtakoja ne tik techninius realizacijos aspektus, bet ir visą kūrimo eigą. Proceso išskirtinumą lemia tai, jog veiklos taisyklės jau sukurtoje sistemoje turi būti aiškiai atskirtos, suklasifikuotos ir paruoštos galimiems jų pakeitimams. Siekiant šio tikslo būtina koreguoti tradicinį požiūrį atsižvelgiant į eilę rekomendacijų, kurios leistų sėkmingai tirti dalykinę sritį, atlikti analizę, sistemos projektavimą ir realizaciją. Svarbu pastebėti, jog daugiausia dėmesio pastaruoju metu susilaukia IS fizinių modelių projektavimo ir jos realizavimo fazės, rinkoje pasirodo vis daugiau produktų, leidžiančių įvairiai realizuoti veiklos taisyklių principais veikiančias sistemas. [1]. Tuo tarpu reikalavimų kuriamai sistemai nustatymo ir analizės metodika aprašyta gana menkai, nėra nusistovėjusio unifikuoto požiūrio į šiuos IS kūrimo etapus.

Išnagrinėjus probleminę sritį ir suvokus veiklos taisyklių egzistavimo svarbą, nuspręsta kurti veiklos taisyklių redaktorių, bent jau iš dalies realizuojantį šios problemos sprendimą. Todėl magistrinio darbo tikslas – *veiklos taisyklių redaktoriaus sukūrimas*. Galima išskirti sekančius etapus:

- veiklos taisyklių duomenų bazės projektavimas;
- veiklos taisyklių redaktoriaus sukūrimas.

Magistrinio darbo aprašas bus sudarytas iš analizės, projekto ir eksperimentinio tyrimo dalių. Analizės dalyje išanalizuota probleminė sritis, apibrėžtos projekte naudojamos sąvokos, jų svarba informacijos sistemų inžinerijoje. Naudojantis UML notacija nubraižytos diagramos, iliustruojančios nagrinėjamą dalykinę sritį. Projekto modelyje projektuojama būsima sistema. Nagrinėjama vartotojų sąsaja, duomenų bazės modelis, realizacijos modelis bei pateikiamos projekto dalies išvados. Eksperimentinio tyrimo dalyje aprašomi reikalavimai sistemos funkcionalumui ir numatoma tolimesnio sistemos tobulinimo strategija. Magistrinio

darbo aprašo pabaigoje pateikiamos viso darbo išvados, kuriose atsispindi visame projekte atlikti darbai, inžinerinė nauda bei sistemos privalumai ir trūkumai.

## 2. ANALIZĖS DALIS.

### 2.1. Tyrimo sritis, objektas ir problema.

#### 2.1.1. Veiklos taisyklės sąvoka.

Veiklos taisyklė yra teiginys, kuris apibrėžia ar apriboja tam tikrus veiklos aspektus. Veiklos taisykle nusakoma tam tikros veiklos struktūra, kontroliuojama arba įtakojama tam tikra veikla [4].

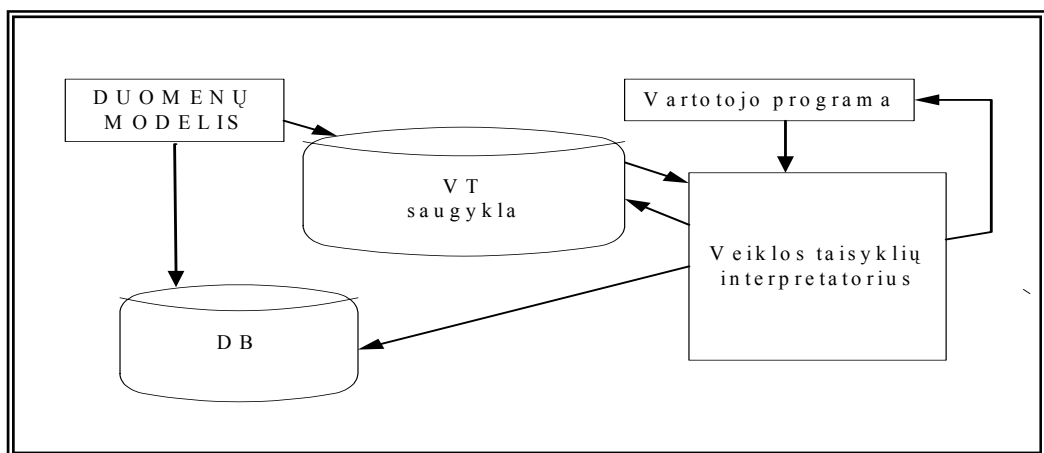
Kai veiklos taisyklė nusakoma veiklos žmogaus, tai ji dažniausiai įvardijama dviprasmiškai ir negriežtai. Dažniausiai veiklos taisykles galima išskaidyti iki elementarių taisyklių, iki kol atsiribojama viena išbaigta mintimi. Elementari veiklos taisyklė, parašyta deklaratyviu būdu, naudojant standartinę natūralios kalbos gramatiką, kurią veiklos žmonės lengvai supranta, nėra dviprasmiška. Taigi, veiklos taisyklės yra elementarios, deklaratyvios, nepriklausančios nuo loginės struktūros, orientuotos į veiklą ir priklausančios veiklai [4].

Kad veiklos taisykles būtų lengva keisti, reikia, kad jos būtų logiškai atskirtos nuo duomenų bei programos struktūros. Dažniausiai jos įsimenamos taisyklių redaktoriuje, kuris yra priemonė, leidžianti peržiūrėti, pakeisti ar tvarkyti tas taisykles.

Veiklos taisyklės gali būti labai įvairios. Pagal skirtingus metodus jos skirtingai ir klasifikuojamos, tačiau ir tuose skirtumuose galima atrasti bendrumo.

#### 2.1.2 Veiklos taisyklių realizacija.

Veiklos taisyklių realizacija susideda iš trijų dalių: veiklos taisyklių procesoriaus, saugyklos ir interpretatoriaus (manager) [3] (1 pav.)



1 pav. Veiklos taisyklių realizacija.

### **Veiklos taisyklių procesorius**

Veiklos taisyklių procesorius įdiegia taisykles į duomenų modelį. Pirmiausia jis nuskaityt taisyklių saugyklą, kurioje saugomos visos veiklos taisyklės. Po to procesorius įterpia išankstines sąlygas (pre-conditions) ir veiksmus (action) į įvykių valdiklį, susietą su veiklos taisyklės įvykiais. Veiklos taisyklių procesorius yra kritiškiausia ir sudėtingiausia veiklos taisyklių sistemos dalis, todėl kad ji yra vienintelė sieja taisyklių sistemą su veiklos modeliu. Taisyklių procesorius privalo būti taip suprogramuotas, kad galėtų keisti taisykles, išreikštas kalba panašia į natūralią kalbą ir saugomas reliacinėje duomenų bazėje objektiškai orientuotame kode [3].

### **Veiklos taisyklių saugykla**

Veiklos taisyklių saugyklos paskirtis yra saugoti aprašytas taisykles ir pakartotinai naudojamus taisyklių komponentus. Remiantis skirtingais metodais, tokios saugyklos architektūra gali būti labai įvairi. [3]

### **Veiklos taisyklių interpretatorius (manager)**

Veiklos taisyklių interpretatorius yra sąsaja su veiklos taisyklių saugykla. Šis komponentas parodo egzistuojančias taisykles, o taip pat parodo egzistuojančių taisyklių komponentus, kuriuos galima panaudoti kuriant naujas taisykles. Veiklos taisyklių interpretatoriaus dar viena iš funkcijų yra ta, kad vartotojas jo dėka gali nurodyti taisyklių procesoriui realizuoti veiklos taisykles veiklos modelyje. Jei procesorius aptinka klaidą, jis apie ją informuoja interpretatorių. Jame galima aprašyti veiklos taisykles natūralia kalba. [3]

#### **2.1.2. Veiklos taisyklių svarba IS kūrime.**

Veiklos IS kūrime egzistuoja trys elementai: duomenys, procesai ir taisyklės. Tuo tarpu kai pirmieji du jau integruoti, naudojant objektiškai orientuotus metodus, taisyklių vis dar yra nepaisoma ir besąlygiškai paliekama programos kode. Apleton (1984) iškėlė problemą, kad trūksta ryšio tarp aukšto lygio duomenų architektūros ir projekto lygio fizinio duomenų kūrimo arba duomenų apdorojimo. Pasak Apleton klausimas yra "Kuri ontologija yra mūsų kompiuteriuose? 1980 metais buvo procesų ontologija, t.y. rinkodara ir produktų kūrimas, o vėliau 1990 metais prasidėjo organizacijos arba verslo ontologija?" Pratęsiant mokslininko klausimą, galima jį papildyti: 2000 metais prasidėjo verslo tinklo ontologija, kur veiklos atnaujinimo procese, veiklos taisyklės turi būti dinamiškos tam, kad apdorotų nuolat besikeičiančią verslo aplinką.

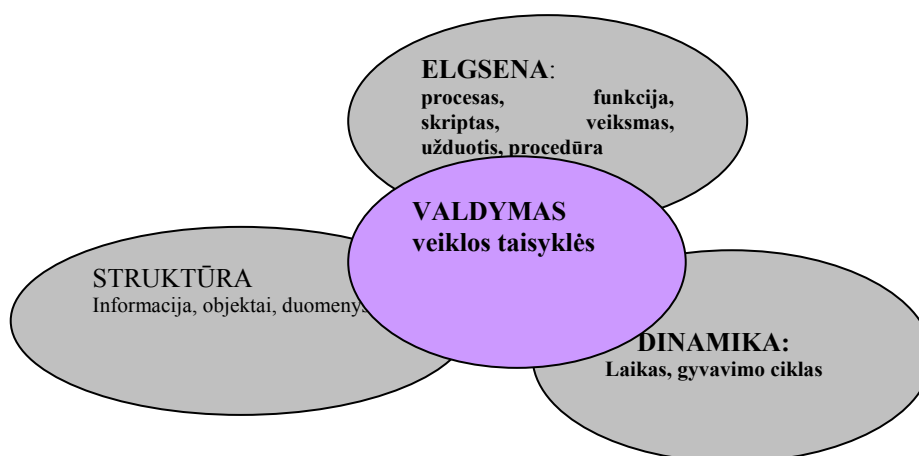
Veiklos taisyklių svarba ir populiarumas išaugo pastaraisiais metais. Jos pripažintos kaip atskira sąvoka, vaidinanti pagrindinį vaidmenį kuriant taikomas programas, kurios yra lanksčios ir besąlygiškos pokyčiams. Galima išskirti tokius veiklos taisyklių etapus: taisyklių analizė, klasifikavimas, išreiškimas ir formalizavimas.

## 2.2 Analizės metodų, priemonių parinkimas.

Įvairių procesų vyksmą reglamentuoja veiklos taisyklės, kurių identifikavimas, struktūrizavimas bei specifikavimas yra pagrindiniai probleminės srities analizės ir reikalavimų kuriamai IS specifikacijos sudarymo uždaviniai.

Į veiklos taisyklės žiūrint iš IS inžinerijos pozicijų, tai jos yra pačiame struktūrinių, elgsenos ir dinaminių modelių centre. Visus šiuos modelius projektuotojas sukuria analizės ir projektavimo etapuose, nepaisydamas to, ar jis naudoja į duomenis, procesus, objektus ar komponentus orientuotus metodus.

Kompanija EBGconsulting, užsiimanti veiklos bei veiklos taisyklių modeliavimo konsultacijomis, pateikė 2 paveiksle parodytą schemą, kurioje matyti, kad veiklos taisyklės atlieka kontrolės vaidmenį ir apjungia visus – elgsenos, struktūros ir dinamikos – sluoksnius. [4]



2 pav. Veiklos taisyklių padėtis kitų modelių atžvilgiu. [4]

Jau daugelį metų nagrinėjamos veiklos taisyklės, tačiau vis kyla ir kyla naujų idėjų ir pasiūlymų jų modeliavimui ir formalizavimui. Šiuo klausimu dirba daug pasaulio mokslininkų, kurie pateikia savo sukurtas realizacijas arba tik išsamias studijas. 1 lentelėje pateikiama skirtingų veiklos taisyklės nagrinėjančių mokslininkų klasifikacijos.

### Veiklos taisyklių klasifikacija ( pagal jas nagrinėjusius mokslininkus )

1 lentelė

ŠALTINIS	TAKSONOMIJA
Guide Business Rules Project	<ul style="list-style-type: none"> <li>• <b>struktūrinės veiklos taisyklės</b> (terminai, faktai)</li> <li>• <b>veiksmo veiklos taisyklės</b> (integralumo apribojimai, sąlygos, autorizacija)</li> <li>• <b>išvesties veiklos taisyklės</b> (skaičiavimai, išvados)</li> </ul>
Ron Ross, Database Research Group	<ul style="list-style-type: none"> <li>• <b>terminai</b></li> <li>• <b>faktai</b> (duomenų ryšių tipai, sub-tipai, atributai)</li> <li>• <b>taisyklės</b> (skaičiavimo, atmetimo, kūrimo)</li> </ul>
James Odell, independent consultant	<ul style="list-style-type: none"> <li>• <b>apribojimai:</b> (poveikis/reakcija, operacija, struktūra)</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>išvedimas:</b> (išvedimo, skaičiavimo)</li> </ul> Pastaba: taisyklės gali būti globalios, vietinės ir/arba laikinos.
Tom Romeo, IBM	<ul style="list-style-type: none"> <li>• <b>struktūrinės</b> (duomenų ryšių, srities, kardinalumo)</li> <li>• <b>elgsenos</b> (prieš-sąlygos, po-sąlygos, išvedimo)</li> </ul>
Margaret Thorpe, Tangram	<ul style="list-style-type: none"> <li>• <b>apibrėžimas</b></li> <li>• <b>pagrindiniai integralumo apribojimai;</b></li> <li>• <b>bendri deklaratyvūs apribojimai;</b></li> <li>• <b>procedūriniai apribojimai;</b></li> <li>• <b>išvesties;</b></li> <li>• <b>išvedimo.</b></li> </ul>
Barbara von Halle, Knowledge Partners	<ul style="list-style-type: none"> <li>• <b>apibrėžimas;</b></li> <li>• <b>faktai;</b></li> <li>• <b>apribojimai;</b></li> <li>• <b>išvedimo;</b></li> <li>• <b>rezultato.</b></li> </ul>
Usoft Corporation	<ul style="list-style-type: none"> <li>• <b>ribojimo</b> (apribojimai, srities);</li> <li>• <b>išvados darymo;</b></li> <li>• <b>elgsenos;</b></li> <li>• <b>pristatymo.</b></li> </ul>
Dan Tasker, Air New Zealand	<ul style="list-style-type: none"> <li>• <b>veiksmų apribojimo;</b></li> <li>• <b>veiksmų sužadinimo;</b></li> <li>• <b>apribojimų.</b></li> </ul>
Brightware	<ul style="list-style-type: none"> <li>• <b>verslo;</b></li> <li>• <b>politikos;</b></li> <li>• <b>darbų srautų;</b></li> <li>• <b>sprendimų euristicos.</b></li> </ul>
Vision Software	<ul style="list-style-type: none"> <li>• <b>validavimo;</b></li> <li>• <b>išvedimo;</b></li> <li>• <b>duomenų ryšių;</b></li> <li>• <b>sąlyginių veiksmų.</b></li> </ul>

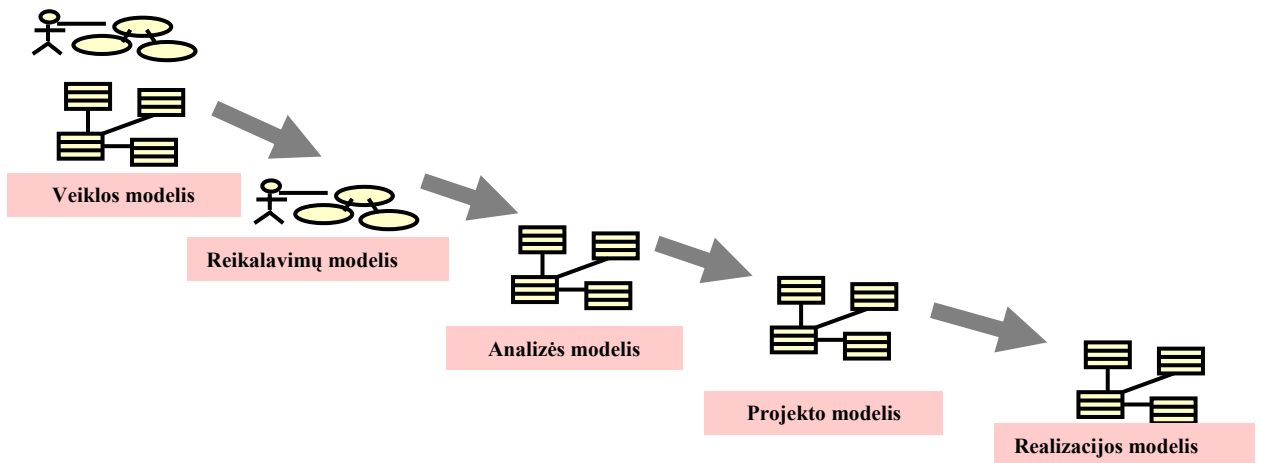
## 2.3 Organizacijos veiklos analizė.

### 2.3.1 Analizės tikslas.

Šiame darbe nebus kuriama informacinė sistema, todėl ir organizacija kaip objektas nebus nagrinėjama. Atliksime tik pačių veiklos taisyklių surinkimo, įvedimo, formalizavimo ir saugojimo veiklos taisyklių redaktoriuje veiksmų analizę. Modeliai bus pateikiami naudojant UML notaciją Rational Rose pakete.

Modelius galima struktūrizuoti įvairiais būdais, priklausomai nuo projektavimo metodo ir pasirinktos struktūrizavimo strategijos. Taikant IS projektavimui skirtą **RUP**, sukuriama eilė modelių (3 pav.):

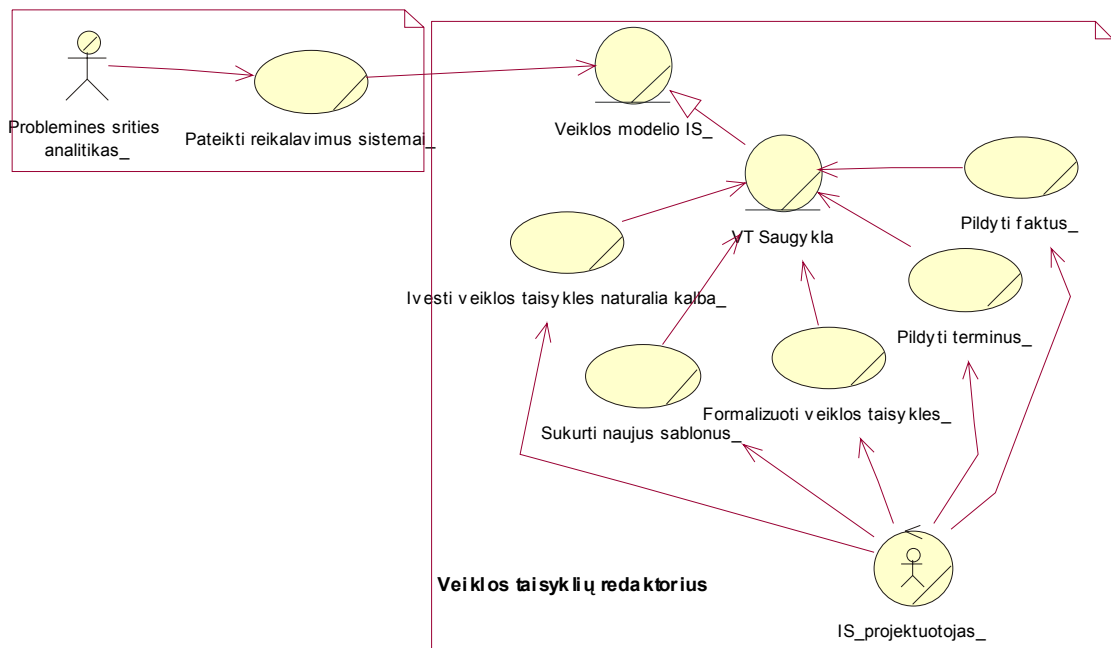




3 pav. RUP metodo modeliai.

### 2.3.2 Veiklos sąveikų modelis.

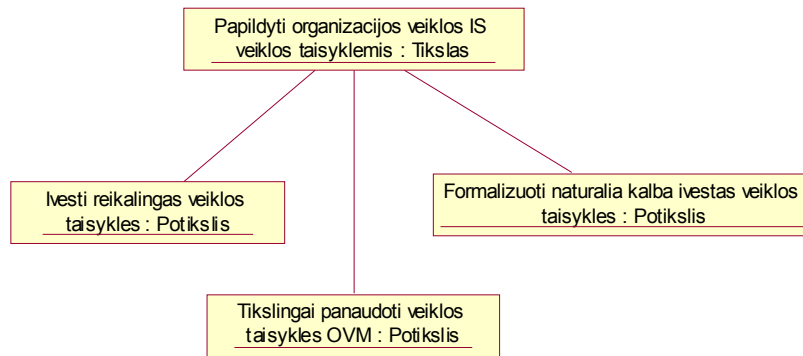
Kuriant veiklos panaudojimo atvejų diagramas, naudojami specialūs stereotipai: veiklai išorinis aktorius (*business actor*), veiklos panaudojimo atvejis (*business use case*), vidinis veiklos aktorius – *business worker*. Jie atspindi ne kompiuterinės sistemos panaudojimo atvejus, bet analizuojamoje organizacijoje vykdomus veiklos procesus. Žemiau pateiktame 4 paveiksle veiklos sąveikų modelis parodo veiklos ryšius su išoriniais aktoriais



4 pav. Veiklos sąveikų modelis.

### 2.3.3 Veiklos tikslų modelis.

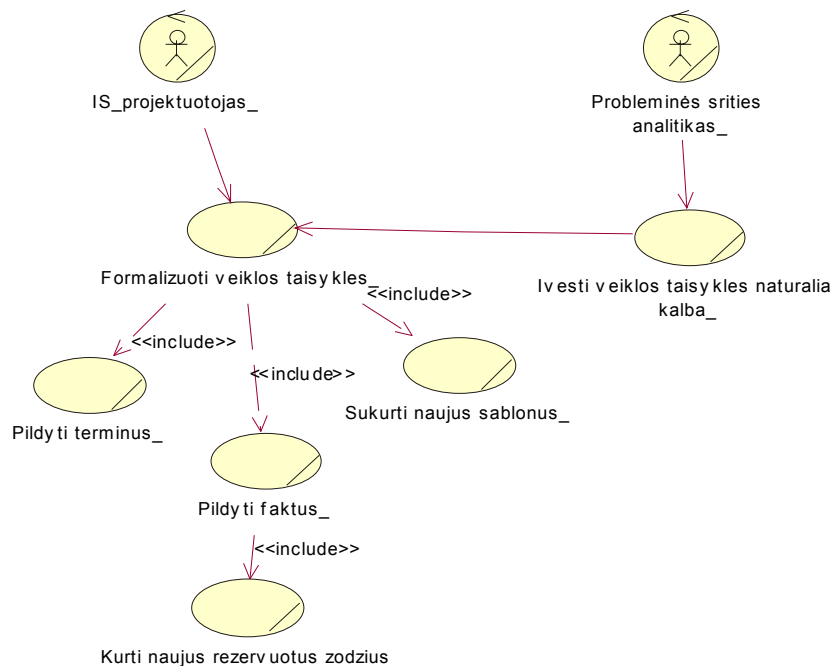
5 paveiksle pateikta nagrinėjamos srities išskirti tikslai, kurie gali būti išskaidomi į daug smulkesnius, konkretesnius teiginius.



5 pav. Veiklos tikslų modelis.

### 2.3.4 Veiklos panaudojimo atvejų modelis.

Veiklos panaudojimo atvejų modelyje pateikiami visi IS projektuotojo veiksmai, kurie atliekami organizacijoje. Kita spalva pažymėti panaudojimo atvejai, kuriuos planuojama kompiuterizuoti (6 pav.).

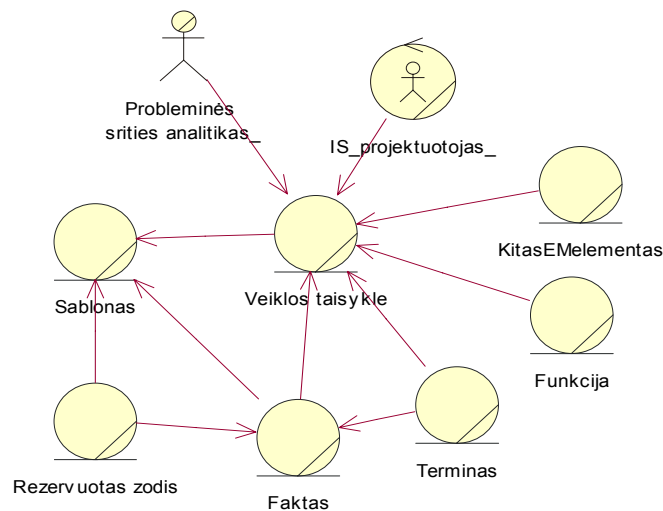


6 pav. Veiklos panaudojimo atvejų modelis.

### 2.3.5 Veiklos objektų modelis.

Veiklos modelyje klasių diagramos naudojamos veiklos objektams – dalykinės srities esybėms – modeliuoti. Veiklos objektų modeliu galima aprašyti tik pagrindinius konceptus ir jų ryšius, nenurodant nei atributų, nei operacijų, nei ryšių kardinalumo. Toks modelis tarnauja kaip žodynas, tolesniuose projektavimo etapuose, jis tikslinamas ir detalizuojamas. Tačiau, esant reikalui, veiklos objektų modelis gali būti pakankamai tikslus. Jį galima tiesiai perkelti į

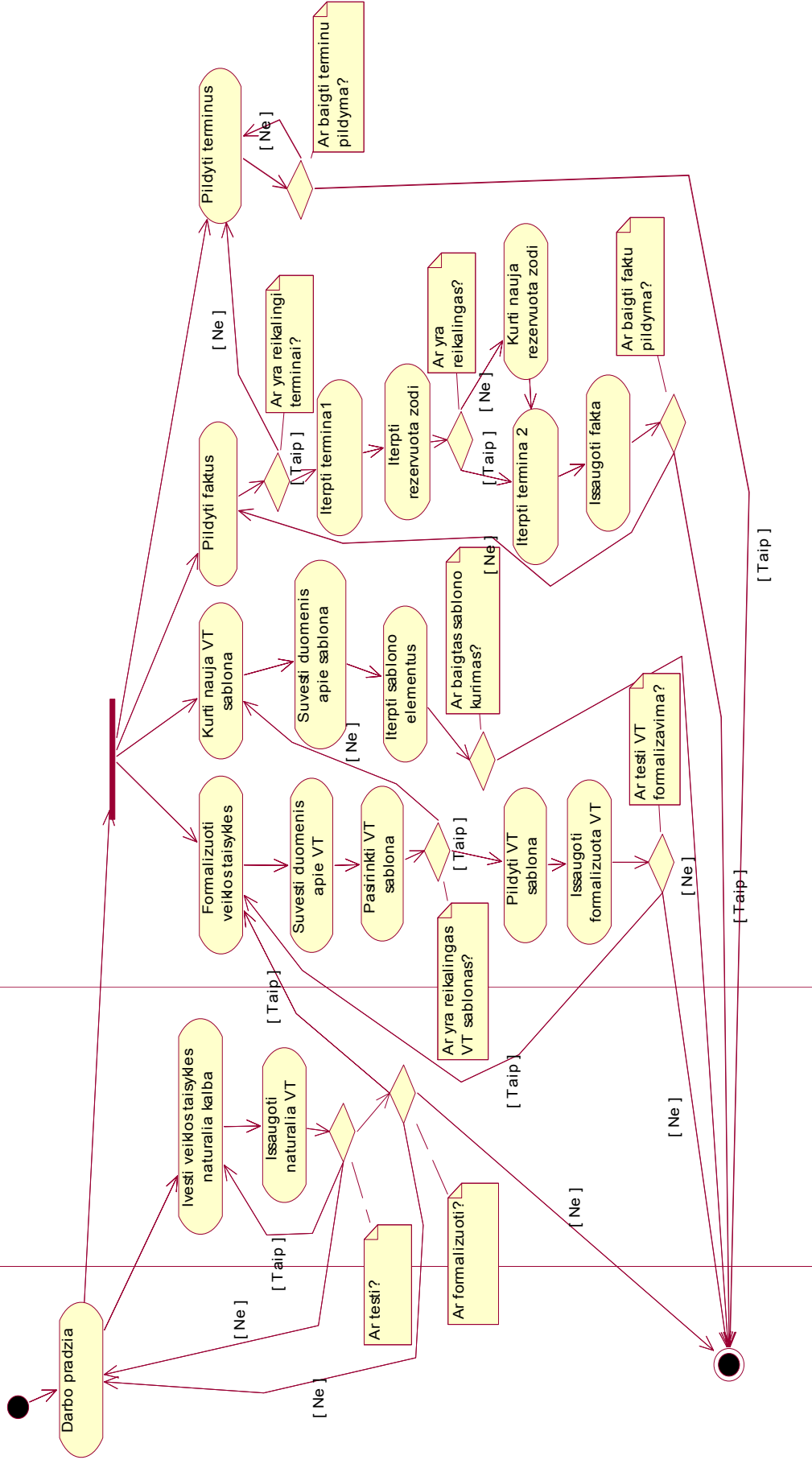
dalykinės srities klasių modelį ir panaudoti duomenų bazės projektavimui. 7 paveiksle pateiktas veiklos objektų modelis.



7 pav. Veiklos objektų modelis.

### 2.3.6 Veiklos procesų modelis.

IS projektavimo metu veiklos diagrama naudojama daugelyje modelių. Veiklos panaudojimo atveju modelyje veiklos diagramą galima panaudoti veiklos procesų modeliavimui. Detali veiklos diagrama naudojama veiklos modeliavimo etape, aprašant darbų sekų procesus. Sprendimo taškai įgalina aprašyti proceso vykdymo veiklos taisykles. Jei sinchronizavimo juosta naudojama perėjimų išskaidymui, tai reiškia, kad turi būti vykdomi visi iš jos išeinantys perėjimai. Tačiau galima aprašyti perėjimų taisykles, kad būtų vykdomi tik tie perėjimai, kurių vykdymo sąlygos išpildytos. Perėjimo sąlygas reikia aprašyti taip, kad jos teisingai atspindėtų veikos logiką (8 pav.).



8 pav. Veiklos procesų modelis.

### 2.3.7 Vartotojų bei jų poreikių analizė.

Vartotojo sąsaja su sistema kuriant ją yra bene svarbiausias dalykas. Kad būtų visiškai patenkinti vartotojo reikalavimai sistemai, būtina į tai atsižvelgti. Su kuriama sistema galės dirbti ne bet kuris vartotojas, kaip tai galima daryti su daugeliu sistemų, nes ji yra specializuota. Ši kuriama sistema skirta jau egzistuojančios informacinės sistemos vienai iš dalių užpildyti, ir todėl kad tiksliai ir kokybiškai formalizuoti veiklos taisykles vartotojas turėtų būti informacijos sistemų srities specialistas. Šis vartotojas analizuotų gautas iš probleminės srities analitiko veiklos taisykles.

Kuriamoje sistemoje bus tik du vartotojai:

- ❖ *probleminės srities analitikas* – analizuoja visą veiklos sritį, surenka keliamus reikalavimus informacijos sistemai ir suveda į Veiklos taisyklių redaktorių veiklos taisykles tik natūralia kalba;
- ❖ *IS projektuotojas* – išsirenka natūralia kalba suvestas veiklos taisykles, jas labai atidžiai analizuoja, nagrinėja ir formalizuoja, t.y. aprašo formalia kalba naudodamas šablonus. Šablonas – tai konkretaus tipo veiklos taisyklės struktūra. Jei nėra sukurto šablono pagal konkrečią veiklos taisyklę, tuomet sukuria tam tikram tipui reikalingą šabloną. Pagrindines funkcijas su sistema ir atliks IS projektuotojas.

Vartotojo sąsaja bus draugiška vartotojui, nes darbas su sistema bus nesudėtingas, aiškūs meniu punktai.

### 2.3.8 Informacijos sistemai keliami nefunkciniai reikalavimai ir apribojimai.

Pirmiausia analizuojant ir kuriant informacijos sistemą, reikalinga išnagrinėti keliamus nefunkcinius reikalavimus ir apribojimus, nuo kurių priklausys būsimos informacijos sistemos funkcionalumas ir savybės.

- ❖ patikimumas;
- ❖ patogumas;
- ❖ efektyvumas;
- ❖ priežiūros savybės;
- ❖ perkeliamumas.

## **2.4 Pasaulio bei Lietuvos literatūros šaltiniuose pateiktų sprendimų problemai spręsti lyginamoji analizė.**

### **2.4.1 Atskirų mokslininkų siūlomi sprendimai.**

Veiklos taisyklėmis orientuotas koceptualus modeliavimas (BROCOM) pateikia metamodelį, kuriame formalizuojamos veiklos taisyklės konceptualiai. BROCOM metamodelyje veiklos taisyklės susideda iš trijų komponentų: įvykio, kuris sužadina veiklos taisyklės, sąlygų, kurios turi būti tenkinamos prieš atliekant veiksmą ir veiksmo, kuris aprašo reikalingą įvykdyti užduotį. Vadinasi, veiklos taisyklės yra apibrėžiamos naudojant ECA struktūrą. Be to, BROCOM taip pat palaiko modelį, kuris susieja veiklos taisyklės su organizaciniais vienetais. Business Rules Group (BRG) sugrupavo veiklos taisyklės į tris grupes, t.y. struktūrinius apribojimus, veiksmo apribojimus ir išvedimo taisyklės. Struktūriniai apribojimai yra teiginiai, apibrėžiantys sąvokas arba ryšius tarp verslo sudėtinių dalių. Galima išskirti du struktūrinių apribojimų tipus: terminai ir faktai. Terminas yra žodis arba frazė, turinti specifinę reikšmę veikloje. Faktas apibrėžia asociacijas tarp dviejų arba daugiau terminų. Veiksmo apribojimai yra įtakojami dinaminių veiklos aspektų. Jie aprašo sąlyginius veiksmus, integralumo apribojimus ir paprastus veiksmus. Ir išvedimo veiklos taisyklės yra išvedimo faktas, kuris yra kuriamas išvadų arba matematinių skaičiavimų, naudojant terminus, faktus, kitas išvestis arba veiksmo apribojimus, pagrindu. Matematiniai skaičiavimai gaunami iš faktų, naudojant specifinius matematinius algoritmus.

Von Halle suklasifikavo veiklos taisyklės į terminus, faktus, griežtus apribojimus, negriežtus apribojimus, veiksmus, skaičiavimo ir išvesties. Ši klasifikacija gali būti laikoma kaip BRG klasifikacinės schemos supaprastinimas.

Morgan siūlo veiklos taisyklių teiginius išskaidyti į šablonus, kuriuos vėliau būtų galima transformuoti į formalią logiką. Pagrindinė šablonų forma yra *<subjektas> privalo <apribojimas>*. Ši pagrindinė forma gali būti išplečiama iki daug sudėtingesnių, kurios sudarytos iš penkių šablonų, vadinamų pagrindiniai apribojimai, sąrašinių apribojimų, klasifikavimo, skaičiavimo ir išvardijimo. Morgan taip pat siūlo susieti veiklos taisyklės su faktų modeliu, kuriame yra veiklos objektai, jų tarpusavio ryšiai ir atributai. Kad pasiekti dar didesnio lankstumo, jis dar pasiūlė išsaugoti faktų modelyje klases ir ryšius kuo paprastesnius ir parodyti jų apribojimus kaip veiklos taisyklės.

### **2.4.2 Pasaulyje pateiktų sprendimų problemai spręsti lyginamoji analizė.**

Veiklos taisyklės iš natūralios kalbos sakinio turi būti formalizuojamos, t.y. užrašomos specialia kalba. Tačiau tokių kalbų yra daug ir kiekvienoje veiklos taisyklę galima užrašyti skirtingai.

- Pavyzdžiui, MS Excel programoje veiklos taisyklę, naudodami IF funkciją, galime užrašyti taip:

**IF(logical\_test,value\_if\_true,value\_if\_false)**

kur, logical\_test - konkreti reikšmė ar išraiška, kuri įvertinama TRUE arba FALSE;  
value\_if\_true – tai reikšmė, kurią logical\_test grąžina kaip TRUE;  
value\_if\_false – tai reikšmė, kurią logical\_test grąžina kaip FALSE.

Taip pat IF funkcija aprašoma ir MS Access programoje.

- Programoje SmartCost veiklos taisyklės gali būti aprašomos kitokia tvarka. Joje veiklos taisyklę sudaro pavadinimas, kuris turi būti unikalus, po to nurodomi atributai/elementai, jų aprašas ir pats veiklos taisyklės tekstas.

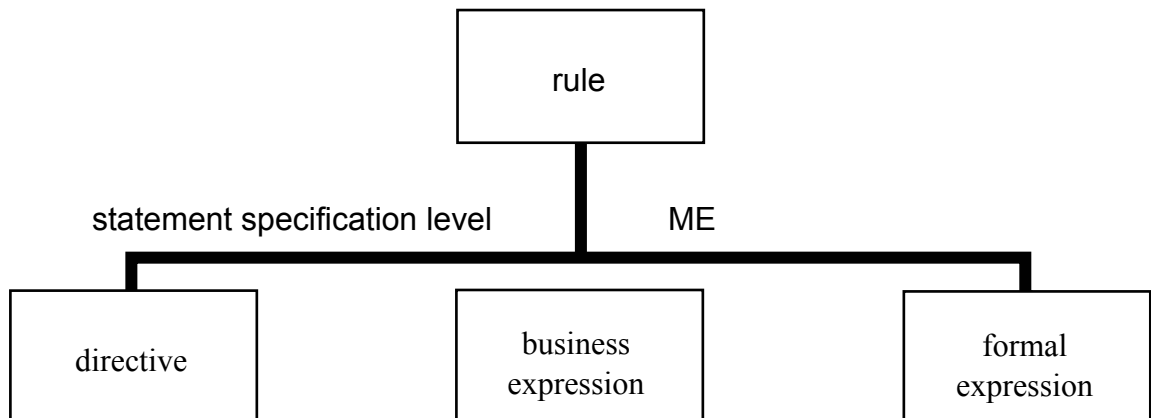
```
If (type_of_gas="toxic",2300
If (type_of_gas="flammable",1400,1000
)
)
```

Čia panašiai kaip ir MS Excel suformuojamas IF funkcijos sakiny – sąlyga, veiksmas, kai rezultatas TRUE ir veiksmas, kai FALSE.

Priede 9.2.1 pateikta keletas langų iš SmartCost programos, kaip galima sukurti naują ir taisyti jau suformuotas veiklos taisykles.

- RuleTrack – tai dar vienas iš išnagrinėtų pavyzdžių, kaip galima modeliuoti veiklos taisykles. Šioje programoje formuojama veiklos taisyklė, atsakant į tokius klausimus:
  - *Kas*: kas tai yra taisyklė?
  - *Kaip*: kokie yra palaikantys komponentai?
  - *Kur*: kuriai modelio daliai ši taisyklė yra skirta?
  - *Kas*: kokios dalys sudaro veiklos taisyklę?
  - *Kada*: kokie pokyčiai įtakoja taisyklę?
  - *Kodėl*: kokios priežastys sukėlė poreikį sukurti taisyklę?

9 paveiksle parodyta, kaip RuleTrack programa atsako į klausimą “Kas tai yra taisyklė?”



9 pav. Kas tai yra taisyklė? **2.5 Analizės išvados.**

- ❖ Išanalizuota daugelis siūlytų veiklos taisyklių klasifikacijų ir apibendrinus sukurta šiame darbe naudojama nauja klasifikacija.
- ❖ Analizuojant probleminę sritį buvo naudojami tokie modeliai:
  - Veiklos sąveikų modelis;*
  - Veiklos tikslų modelis;*
  - Veiklos procesų modelis;*
  - Veiklos objektų modelis;*
  - Veiklos panaudojimo atvejų modelis.*
- ❖ Naudojami modeliai padėjo įvertinti vartotojų atliekamas funkcijas, jų poreikius bei atsakomybes.
- ❖ Išanalizuoti pasaulyje pateikti sprendimai veiklos taisyklių formalizavimo problemai spręsti. Prieduose pateikta keletas vartotojų sąsajos langų iš nagrinėtų programų.
- ❖ Aprašyti šablonai, kurie bus naudojami veiklos taisyklių formalizavimui.
- ❖ Parinkta programinė įranga Java programavimo kalba ir MySQL duomenų bazių sistema, kurių pagalba kuriamas veiklos taisyklių redaktorius.
- ❖ Suformuluota techninė užduotis projektavimui.



### 3. PROJEKTO DALIS.

#### 3.1 Projekto tikslas ir jo pagrindimas, kokybės kriterijų apibrėžimas.

Priešingai nei tradicinėse IS, kuriose yra naudojamos veiklos taisyklės, IS galima nagrinėti ne dviem, o trimis lygiais – duomenų valdymo, realizacijos (arba vartotojo) ir veiklos taisyklių lygiu. Yra daug skirtingų veiklos taisyklių pagrindu sukurtų IS architektūros modelių, bet veiklos taisyklių vaidmuo išlieka toks pat. Jis padeda lengviau valdyti veiklos taisykles. Paprastai taisyklės yra realizuojamos naudojantis procesų valdymo metodais, kur taisykles galima rasti programos kode. Tai padaro veiklos taisyklių redaktorių labai sunkiai pritaikomą prie pokyčių. Veiklos taisyklių realizavimas rankiniu būdu arba sąrašais suteikia daugiau lankstumo, tačiau tai nepalaiko IS vartotojo veiksmų valdymo.

Veiklos taisyklių redaktorius yra autonominis veiklos taisyklių rinkinys, kuris gali būti keičiamas bet kuriuo metu panaudojant sąlyginai paprastas priemones. Veiklos taisykles galima saugoti dviem būdais [15]:

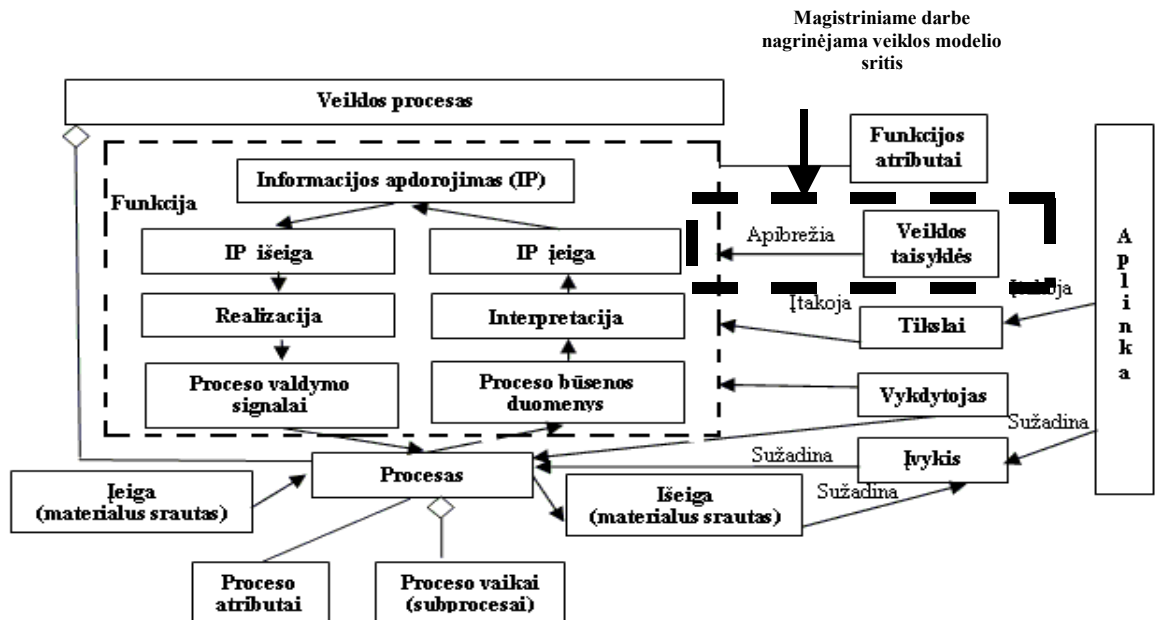
- ❖ Parametrais pagrįstu metodu. Šiuos atveju taisyklės yra saugomos duomenų bazėje, kurioje jos yra aprašomos įvairių atributų reikšmėmis. Pasak tyrinėtojų, taisyklių saugykla gali būti suprojektuota arba kaip nepriklausoma duomenų bazė, arba kaip dalis pagrindinio loginio modelio. Tačiau pirmasis sprendimas suteikia daugiau lankstumo ir daugiau galimybių sudėtingų veiklos taisyklių saugojimui.
- ❖ Nepriklausomas nuo procesų metodas. Šis metodas yra panašus į tradicines metodologijas, kuriose taisyklės yra realizuotos tiesiogiai programos kode, tik šiuo atveju kodas, atvaizduojantis taisykles, yra saugomas nepriklausomai nuo kitų IS lygių, todėl taisyklės yra išreikštos tik vieną kartą visoje informacijos sistemoje.

Esminis šio projekto akcentas yra tas, kad veiklos taisyklių redaktorius bus kuriamas ir siejamas su organizacijos veiklos modeliu (OVM). Jame atsispindi svarbiausi konstruktai, reikalingi tam, kad būtų sukurta kokybiška ir tenkinanti vartotojų poreikius informacinė sistema. Vienas iš tų konstrukčių yra veiklos taisyklės, kurias naudinga būtų kaupti ir saugoti atskirai nuo visos informacinės sistemos. Kaupimas ir saugojimas atskirai nuo visos programos kodo suteikia galimybę redaguoti veiklos taisykles bet kuriuo metu, neįsivieliant į dideles jų paieškas visame kode. 10 paveiksle pateiktas veiklos metamodelis sukurtas remiantis[1] ir pagal kurį bus kuriama programa:

- Porter Veiklos grandinės modeliui;
- Hendersono veiklos ir IT sąsajos modeliui;

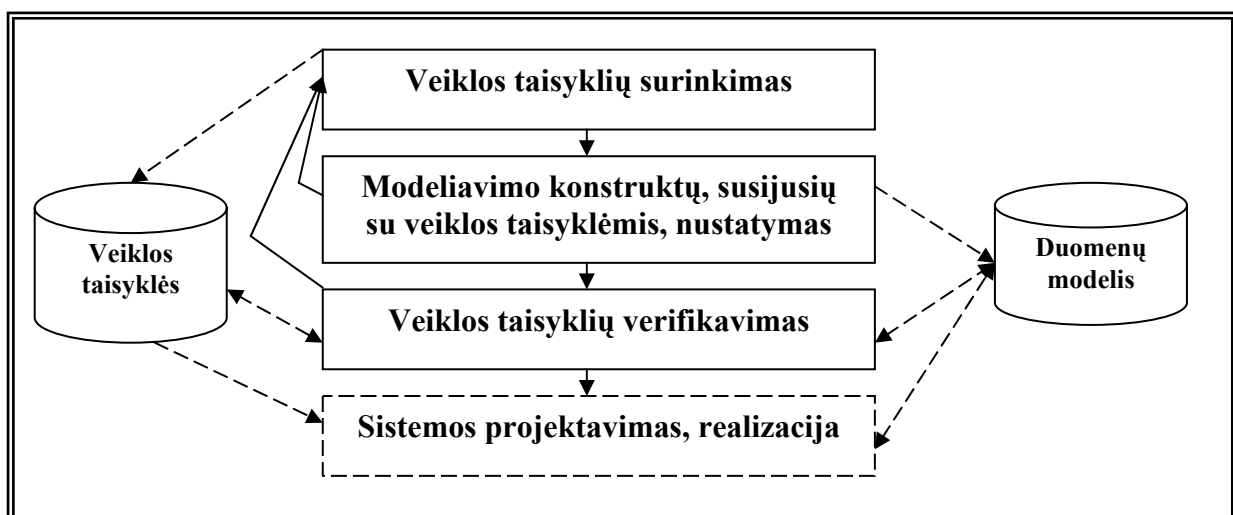
- Tarptautiniais veiklos modeliavimo standartais (UEML, ENV 12204, ENV 40003, IDEF3).

10 paveikslas iliustruoja projekte nagrinėjamą veiklos modelio sritį.



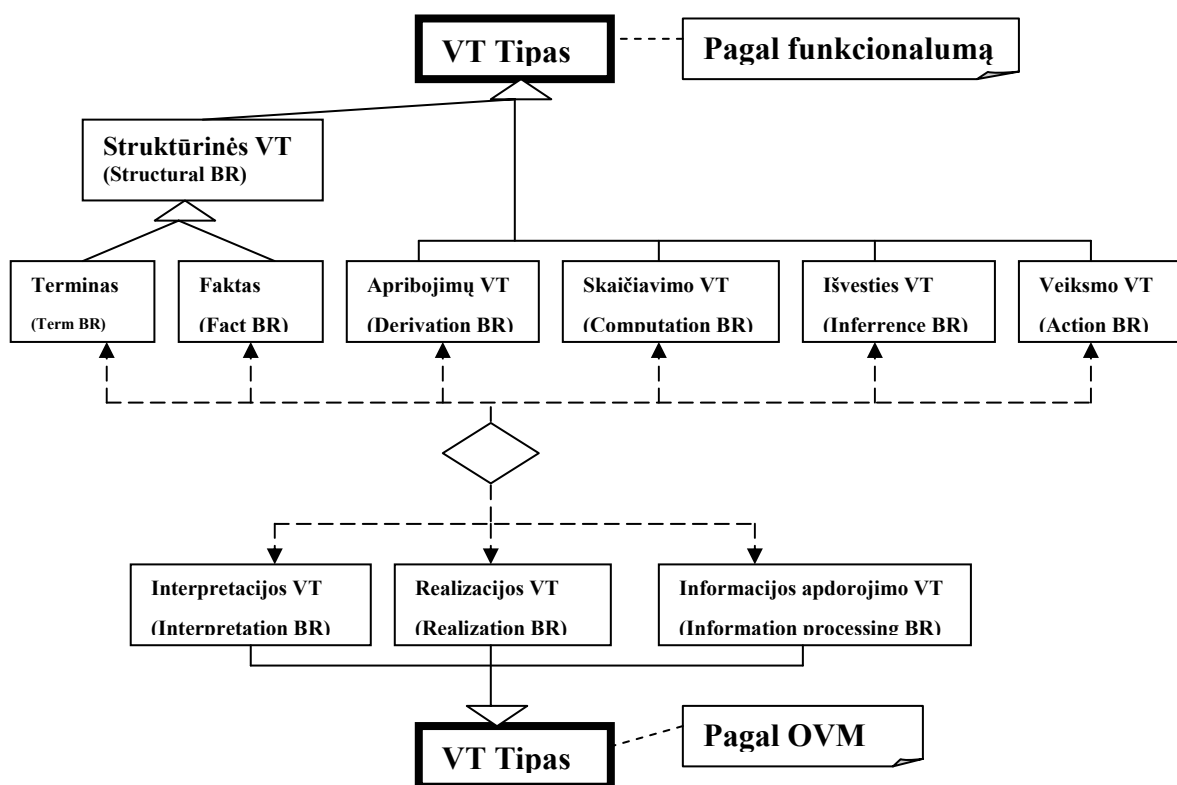
10 pav. Organizacijos veiklos metamodelis.

Pagrindinis sistemos analizės tikslas yra surinkti visą svarbiausią informaciją apie probleminę sritį. Šie faktai pirmiausia yra apie procesus bei apie duomenų objektų struktūrą ir elgseną. Veiklos taisyklės apima abu aspektus, todėl mes laikome, kad jos yra pagrindinis sistemos analizės elementas ir todėl adaptuojame sistemos analizės procesą. Šiame procese faktai surenkami realiame pasaulyje (imant interviu, dokumentacijos) ir apibrėžiant, kad tai veiklos taisyklės [18]. 11 paveikslas iliustruoja viso veiklos taisyklių apdorojimo principus, t.y. kad pirmiausia veiklos taisyklės bus surenkamos, po to nustatomi modeliavimo konstruktai, vėliau veiklos taisyklės verifikuojamos, o galų gale kuriama sistema. Magistriniame darbe nėra realizuojami paskutiniai du žingsniai.



11 pav. Sistemos analizė sukoncentruota ties veiklos taisyklėmis

Išnaginėjus mokslininkų pasiūlytas veiklos taisyklių klasifikacijas, jų idėjas, galima pateikti apibendrintą klasifikaciją (12 pav.), kurią būtų galima kompiuterizuoti šiame darbe.



12 pav. Veiklos taisyklių klasifikacija.

12 paveiksle punktyrinė linija reiškia, kad veiklos taisyklių tipai (VT Tipas2) pagal organizacijos veiklos modelį gali būti bet kurio iš veiklos taisyklių tipų (VT Tipas).

Atlikus magistrinį darbą bus įvertinami šie vertinimo kriterijai:

- tinkamumas, saugumas, tikslumas;
- patikimumas, užbaigtumas, tolerancija klaidoms, atstatomumas;
- patogumas, suprantamumas, išmokstamumas, efektyvumas;
- perkeliamumas, pakeičiamumas.

## 3.2 Projektavimo metodų, priemonių parinkimas.

### 3.2.1 Projektavimo metodų analizė.

Šiuo metu yra sukurta daugybė veiklos taisyklės ir/arba apribojimus aprašančių kalbų, tačiau jos nėra praktiškos dėl jų abstraktumo ir formalizmo. Sprendimas įtraukti vartotoją į IS kūrimo analizės etapą reikalauja aiškaus ir visiškai suprantamo metodo veiklos taisyklių formalizavimui. Metodai, kurie aprašo veiklos taisyklės grafiškai, iš dalies yra tinkami.

Lyginamoji analizė pateikiama 2 lentelėje. Atlikta ji šių dviejų kriterijų pagrindu [11]:

- Kurie konstruktai suteikia galimybę konkrečiam metodui modeliuoti veiklos taisyklės?
- Kokio tipo veiklos taisyklės gali būti modeliuojamos?

2 lentelė

Metodas	Modeliavimo konstruktai					Veiklos taisyklių tipai	
	Objektai	Taisyklės	Įvykiai	Sąlygos	Veiksmas	Nuoseklumo taisyklės	Veiklos taisyklės
DFD of SA	duomenų saugojimo, išoriniai objektai	---	iš dalies duomenų srautų	---	procesai	tik minidalyse (minispecs)	labai ribotos
CPM of Merise	---	---	išoriniai/vidiniai įvykiai	sinchronizacija	operacijos	taip (nenumatomos)	taip
STD	---	---	(užklausų lentelės)	užklausų lentelės	užklausų lentelės	taip (nenumatomos)	taip
Transition Graph	---	---	---	užklausų lentelės	---	tranzityvios, dinaminės	ne
Ext.PN	---	---	mazgai	mazgai	užklausos	taip (nenumatomos)	taip
ERM, NIAM	esybių tipai, ryšių tipai (atributai)	---	---	kardinalumai, raktai, sritys	---	statinės (ribotos)	ne
ER-RM	esybių tipai, ryšių tipai (atributai)	taisyklių tipas	nurodytos sritys	persidengia su taisyklių tipu	nurodytos sritys	taip	labai ribotos
BIER	esybių tipai, ryšių tipai (atributai)	---	mazgai	---	užklausos	tranzityvios, statinės, dinaminės	taip
ELH of SSADM	esybės tipas (skirtas vienam esybės tipui)	---	įvykis (išorinis)	---	operacijos	tranzityvios (ribotai)	labai ribotos
Event	---	trigierių	įvykio tipas	valdymo	operacijos	taip	taip

schema of OOA&D	(siejasi su objektų modeliu)	taisyklė		sąlygos		(nenumatomos)	
-----------------	------------------------------	----------	--	---------	--	---------------	--

Šioje dalyje išanalizuota ir akcentuota veiklos taisyklių kaip IS elemento svarba. Taip pat išnagrinėta keletas programinės įrangos inžinerijos metodų, tikėdamasi, kad jie padės modeliuoti veiklos taisykles. Aiškiausias veiklos taisyklių palaikymo būdas - į modeliavimą įtraukti tikslūs taisyklės konstruktai. Be to, veiklos taisyklės taip pat gali būti išreikštos savais komponentais, t.y. įvykiais, sąlygomis ir veiksmis. Modeliavimui tinkamas metodas turėtų palaikyti nors vieną iš šių savybių. Taip pat toks metodas turėtų turėti galimybę specifikuoti veiklos taisyklių poveikį objektams ir taisyklių tarpusavio priklausomybes.

Daugelis gerai žinomų IS modeliavimo metodų neturi reikalingų konstrukčių tam, kad būtų galima formalizuoti veiklos taisykles. Šiuo požiūriu iš dalies tokie yra DFD ir ERM metodai, kurie yra struktūrinės sistemų analizės branduolys. Metodai, aprašantys IS dinamiką, tokie kaip STD ir (išplėstiniai) Petri tinklai, tam tinka labiau, net jei jie ir nepalaiko atitinkamų konstrukčių. Juos galima tiesiogiai naudoti, kad aprašyti seką, kuria veiklos taisyklės yra vykdomos, tačiau šie metodai mažiau tinkami veiklos taisyklių poveikio objektams specifikuojimui. Yra vienintelis būdas išvengti šios problemos. Tai tinklų priskyrimas modeliui, kai pasikeičia objekto būsenos. Šis metodas panašus į ELH idėją. Jo trūkumas tik toks, kad veiklos taisyklės dažnai veiks atskirus skirtingų tipų objektus, kurie nėra atvaizduojami diagramoje.[11]

Kitas būdas veiklos taisyklių formalizavimui yra šablonų naudojimas. Jų naudojimas labai palengvina darbą. Šablonas yra daug kartų panaudojama struktūra, kuri tarnauja kaip pradinis atskaitos taškas. Šablonų naudojimas veiklos taisyklių surinkimui suteikia keletą privalumų [8]:

1) ***Šablonai yra pirmas "Veiklos teiginių" (Business Statements) formalizavimo žingsnis.***

Veiklos taisyklių idėja dažnai suvokiama iki tol, kol nepradedama nagrinėti, kaip taisyklės bus dokumentuotos. Šablonai suteikė VT surinkimo tikrumą ir pademonstravo, kad tai nėra laikinas dalykas.

2) ***Šablonai užtikrina informacijos, surinktos su veiklos taisyklėmis, pateikimo nuoseklumą.***

Tai didžiausia nauda, kurią galima paaiškinti. Jei visi grupės nariai naudoja tą patį šabloną veiklos taisyklių surinkimui, tai jie gali pasidalinti šablonais su bet kuo, kas gali bendrauti, neiškeliant klausimo apie tai, kokią struktūrą naudoti dokumentuojant taisykles.

3) *Biznio dalyje šablonai pašalina dviprasmiškumus ir sudaro sąlygas patikrinimui (validavimui).*

Vienas iš veiklos taisyklių metodo tikslų yra nesudėtingas patikrinimas (validation). Skirtingai nei duomenų ar procesų diagramose VT sąrašas gali keliauti tarp keleto grupių be jokių paaiškinimų ar papildomų dokumentacijų. Šablonai įtraukia šį žingsnį vėliau, tikrindami cirkuliuojančio VT sąrašo nuoseklumą. Be to šablonai susikoncentruoja vienoje dalyje, kuri užtikrina kiekvienos taisyklės validavimą atskirai.

4) *Sistemos dalyje šablonai gali būti fiksuojami pseudo-kodu (Pseudo-Code).*

Kai kiekvienas terminas, naudojamas veiklos taisyklėse, yra susietas su esybėmis, atributais arba atributų reikšmėmis nagrinėjamoje duomenų bazėje, tuomet šablono formoje atvaizduotos taisyklės suteikia kūrėjams daug informacijos.

### 3.2.2. Metodai, aprašantys veiklos taisykles.

Projekto tikslus galima įgyvendinti įvairiais programinių komponentų kūrimo paketais. Konkretaus paketo pasirinkimas turėtų priklausyti nuo numatomos veiklos taisyklių saugyklos bei veiklos taisyklių interpretatoriaus panaudojimo srities. Jei planuojama veiklos taisyklėmis pagrįstą sistemą įdiegti nedidelėje įstaigoje, kurioje veiklos taisyklių kiekis ir duomenų srautai nėra labai dideli, galima naudoti bet kurią šiuolaikinę IS kūrimo sistemą, tačiau didelėse organizacijose, kuriose tenka labai dažnai reaguoti į įvairius situacijos pakitimus ir juos operatyviai pritaikyti, realizacijai reiktų naudoti kitokias kūrimo priemones.

Viena iš metodologijų, naudojama veiklos taisyklių realizavime yra **IDEA**. Tai metodologija, kuri naudoja objektus ir veiklos taisykles duomenų bazėje. Aktyvių taisyklių taikymas - tai originaliausias indėlis IDEA metodologijoje. Susikoncentruojama pritaikant individualias taisykles ir principus pagal analizės poreikius. Vadovaujamosi aktyvios taisyklės nuolatinio pakeisto varianto patikrinimu, kuris sutelkia dėmesį į tiriamą elgesį nustatčius tam tikrą taisyklę, ar jų grupę.[7]

Kitas metodas dažnai naudojamas modeliuoti veiklos taisykles yra **IDEF9**. Šis biznio apribojimų atradimo metodas buvo sukurtas padėti atrasti ir analizuoti biznio sistemos apribojimus. Kartą įtraukus apribojimus į sąrašą jie gali būti sistemiškai analizuojami ir, jeigu reikia, derinami ar pakeičiami, siekiant pagerinti sistemos efektyvumą.[5]

Dar vienas iš dažniau naudojamų metodų yra **OCL** (Object Constrain Language) – kalba padedanti kuo tiksliau išreikšti veiklos taisykles. Tai UML semantikos dalis. Modeliuotojai, kurie įpratę naudoti UML kalbą savo veiklos modeliavimui, gali labai sėkmingai naudoti OCL tam, kad aprašytų tik tai sričiai būdingas taisykles [6]. OCL gali

būti naudojama tiksliai apibrėžti klasių ir tipų kintamuosius klasių modelyje; apibrėžti tipų kintamuosius stereotipams; nusakyti operacijų ir metodų išankstines ir vėlesnes sąlygas; apibūdinti apribojimus; kaip navigacijos kalba; tiksliai apibrėžti operacijų apribojimus.

### Lyginamoji veiklos taisyklių sudėties analizė.

Veiklos taisykles gali sudaryti labai įvairūs komponentai, o jų klasifikacija, remiantis skirtingais metodais ir standartais, taip pat gali būti sudaroma daugeliu būdų. 3 lentelėje pateikta lyginamoji nagrinėtų standartų analizė.

3 lentelė

	IDEA	IDEF9	OCL
Veiklos taisyklių klasifikacija	<ul style="list-style-type: none"> <li>---skaičiavimo taisyklės;</li> <li>---aktyvios taisyklės: <ul style="list-style-type: none"> <li>---vidinės;</li> <li>---išorinės vidinės;</li> <li>---išorinės.</li> </ul> </li> </ul>	<p>Apribojimai:</p> <ul style="list-style-type: none"> <li>---įgalinantys;</li> <li>---ribojantys.</li> </ul> <p>Taip pat:</p> <ul style="list-style-type: none"> <li>---in – vidiniai;</li> <li>---on – išoriniai;</li> <li>---of – vykdomi sistemos arba valdomi sistemoje;</li> <li>---between – sujungiantys verslo sistemas kaip objektus su didesnių sistemų formomis.</li> </ul>	<ul style="list-style-type: none"> <li>---prieš sąlyga;</li> <li>---po sąlyga;</li> <li>---apsauginės;</li> <li>---skaičiavimo;</li> <li>---bendra išraiška;</li> <li>---pastovios.</li> </ul>
Veiklos taisyklių komponentai	<ul style="list-style-type: none"> <li>---įvykis;</li> <li>---sąlyga;</li> <li>---veiksmas;</li> <li>---pirmenybė.</li> </ul>	<ul style="list-style-type: none"> <li>---apribojimas;</li> <li>---kontekstas;</li> <li>---požymis;</li> <li>---apribojimo efektas;</li> <li>---simptomas;</li> <li>---sistema;</li> <li>---apribojimų loginis pagrindas;</li> <li>---tikslas.</li> </ul>	<ul style="list-style-type: none"> <li>---savybės (propierties)</li> <li>---atributai;</li> <li>---tipas;</li> </ul>
Proceso etapai	<ul style="list-style-type: none"> <li>---analizė;</li> <li>---projektavimas;</li> <li>---greitas prototipų kūrimas (testavimas konceptualiame lygyje);</li> <li>---realizacija (Oracle, Illustra, DB2/ODE, validavimas)</li> </ul>	<ul style="list-style-type: none"> <li>---rinkimas – reikalauja stebėjimų ir apribojimų požymių šaltinio.</li> <li>---klasifikavimas – individualizuoti kontekstą, objektus, objektų tipus, savybes ir ryšius.</li> <li>---hipotezės iškėlimas – pradinių galimų apribojimų iš duomenų ir požymių suradimas.</li> <li>---įrodymas – sukurti ar surinkti pavyzdžius, kurie apibrėžtų, kurie galimi apribojimai gali būti parodyti kaip apribojimai.</li> </ul>	

		---iššūkis – įtraukti srities specialistus testuojant išvadas ir analizuojant jų pagrįstumą. ---tobulinimas – filtruojama, gerinama, pritaikoma ir papildoma detalėmis, kurios charakterizuoja apribojimus.	
--	--	---	--

### 3.3 Kompiuterizuojamos sistemos varianto parinkimas.

Pagrindinis šio darbo tikslas – suprojektuoti ir sukurti veiklos taisyklių redaktorių. Jis skiriasi nuo veiklos taisyklių interpretatoriaus tuo, kad jame VT tik formalizuojamos ir saugomos, taip pat tikrinamos pre- ir post- sąlygos.

Šiame magistriniame darbe kompiuterizuojami bus tokie uždaviniai:

- veiklos taisyklių surinkimas ir saugojimas duomenų bazėje;
- terminų ir faktų pildymas;
- veiklos taisyklių formalizavimas:
  - pildomi duomenys apie veiklos taisyklę;
  - pasirenkamas veiklos taisyklės tipas;
  - pasirenkamas OVM elementas, kuriam skirta veiklos taisyklė;
  - pasirenkamas veiklos taisyklės tipas pagal OVM (jei VT skirta elementui „Funkcija“);
  - užfiksuojamos PRE ir POST sąlygos;
  - pildomas šablonas, kuris priklauso nuo veiklos taisyklės tipo pasirinkimo.
- VT redagavimas veiklos taisyklių redaktoriuje.

Natūralios veiklos taisyklės surenkamos kartu su organizacijos darbų sekų modelio (Workflow model) kūrimu. Sukurtoji programa skirsis iš kitų jau egzistuojančių tuo, kad:

- joje bus naudojama lietuvių kalba;
- joje nebus tikrinamas veiklos taisyklių veikimas, kaip tai realizuojama veiklos taisyklių interpretatoriuose;
- bus tiesiogiai susieta su organizacijos veiklos modeliu (aprašytu 2.5 skyriuje).
- kituose darbuose bus naudojama metodų generavimui.

Veiklos taisyklių redaktoriui kurti bus naudojamas Java programavimo kalba. Duomenų bazė, kurioje saugomos veiklos taisyklės, realizuota su MySQL paketu. Šios priemonės projekto realizavimui pasirinktos dėl jų plačių galimybių. Java programavimo kalba suprogramuotą projektą galima pritaikyti labai plačiai.



### **3.4 Techninė užduotis.**

Atliko: IFM-8/4 gr. studentė

Rasa Miklovienė

2003-09-16

SUDERINTA:

Vadovas: doc. S. Gudas

2003-09-16

#### **3.4.1 Magistrinio darbo tema.**

Veiklos taisyklių redaktorius.

#### **3.4.2 Kompiuterizuojamas objektas.**

Veiklos taisyklių surinkimo, papildant jomis veiklos modelį ir formalizavimo kompiuterizavimas.

#### **3.4.3 Kompiuterizuojamos funkcijos.**

Veiklos taisyklių surinkimo, papildant jomis veiklos modelį ir formalizavimo kompiuterizavimas apima tokius uždavinius:

3.4.3.1 Terminų pildymas;

3.4.3.2 Faktų pildymas;

3.4.3.3 Natūralių veiklos taisyklių įvedimas;

3.4.3.4 Veiklos taisyklių formalizavimas;

3.4.3.5 Naujų veiklos taisyklių šablonų kūrimas;

#### **3.4.4 Sistemos naudojimo dokumentai**

3.1.4.1 Vartotojo vadovas;

#### **3.4.5 Reikalavimai projektavimui, programinei ir techninei įrangai.**

3.1.5.1. Projektavimo priemonės: Rational Rose Enterprise Edition 2002 paketas

3.1.5.2 Programinė įranga: Operacinė sistema – Windows;

Programos – JBuilder 9, MySQL.

Techninė įranga: Personalinis kompiuteris (minimalūs reikalavimai) – PC pentium, 8MB RAM, SVGA 1028x768

#### **3.4.6 Reikalavimai darbo pristatymui.**

3.1.6.1 Pateikti darbo aprašą pagal pateiktą magistrinio darbo struktūrą;

3.1.6.2 Pateikti CD su programos kodu, kontrolinio darbo duomenimis, magistrinio darbo tekstu;

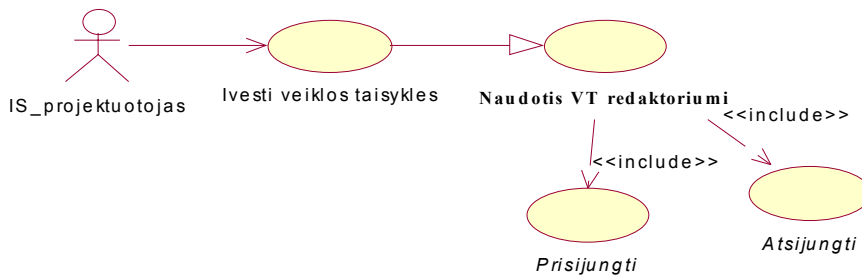
3.1.6.3 Darbo gynymui pateikti Power Point skaidres;

### 3.5 Reikalavimų modelis.

#### 3.5.1 Vartotojų panaudojimo atvejų diagramos.

##### 3.5.1.1 Aukšto lygio panaudojimo atvejų diagrama.

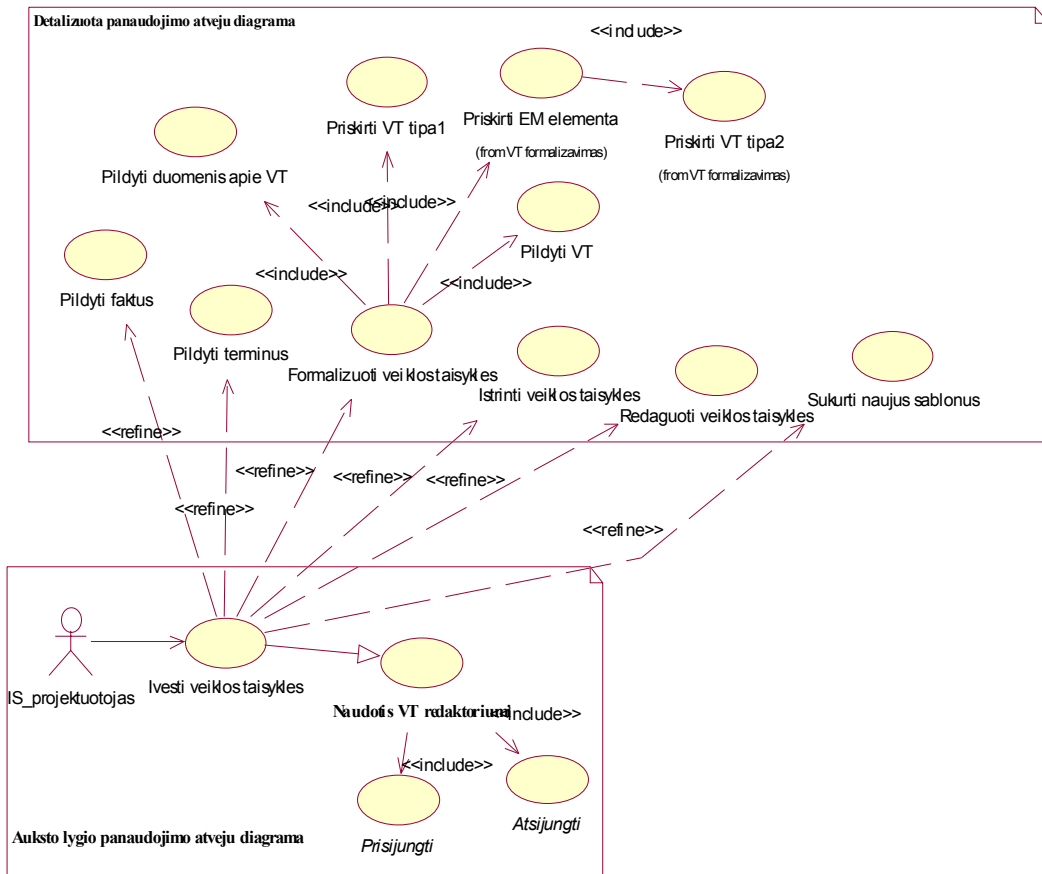
13 paveiksle pateikta aukšto lygio panaudojimo atvejų diagrama. Ji atvaizduoja tik patį pagrindinį veiksma, kuris charakterizuoja visos sistemos darbą.



13 pav. Aukšto lygio panaudojimo atvejų diagrama.

##### 3.5.1.2 Detalesnio lygio panaudojimo atvejų diagrama.

Šioje diagramoje (14 pav.) pagrindinis panaudojimo atvejis išskaidytas į smulkius, sistemoje kompiuterizuotus panaudojimo atvejus.



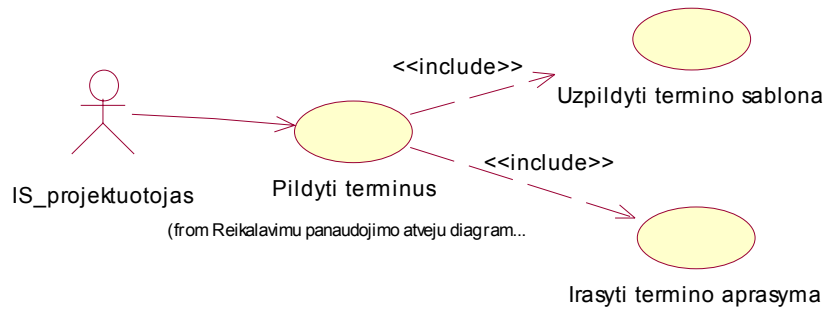
14 pav. Detalesnio lygio panaudojimo atvejų diagrama.

### 3.5.2 Specifikacijos kiekvienam panaudojimo atvejui.

#### 3.5.2.1 Terminų pildymo diagramos.

##### 3.5.2.1.1 Terminų pildymo panaudojimo atvejų diagrama.

Terminų pildymas kuriamoje programoje nėra sudėtingas procesas, tačiau neužpildžius terminų nebus galima toliau tęsti darbo. 15 paveiksle pateikta terminų pildymo panaudojimo atvejų diagrama.



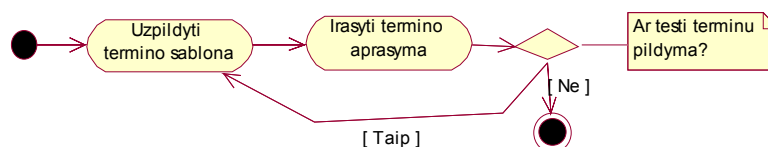
15 pav. Terminų pildymo panaudojimo atvejų diagrama.

##### 3.5.2.1.2 Terminų pildymo veiklos diagrama.

#### Panaudojimo atvejo „Pildyti terminus“ specifikacija

4 lentelė

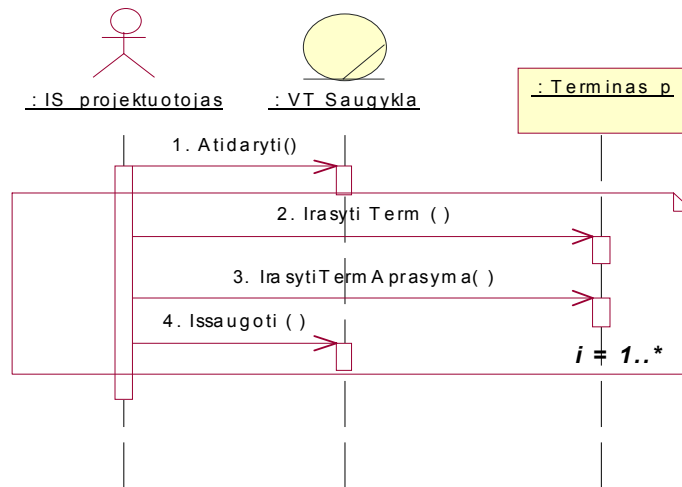
<b>Panaudojimo atvejis</b>	Pildyti terminus
<b>Aktorius</b>	IS projektuotojas
<b>Sistema</b>	Veiklos taisyklių redaktorius
<b>Sužadavimo įvykis</b>	IS projektuotojas gauna papildomus reikalavimus informacijos sistemai.
<b>Prieš sąlyga</b>	Turi būti gauti papildomi reikalavimai.
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Vartotojas pasirenka meniu punktą „Terminų pildymas“ (šis žingsnis apima panaudojimo atvejį „Pildyti terminus“)	1.1. Sistema paleidžia terminų pildymo formą.
2. Užpildyti termino šablona.	2.1 Įrašomas į šablona terminas, kuris vėliau bus naudojamas sistemoje.
3. Įrašyti termino apibūdinimą.	3.1 Užpildoma informacija apie terminą.
4. Išsaugoti ir įvesti kitą terminą.	5.1 Jei nevedamas kitas terminas, baigiamas terminų pildymas.
5. Baigti terminų pildymą.	
<b>Po sąlyga</b>	Sukuriamas naujas terminas, kuris vėliau naudojamas faktų ir veiklos taisyklių pildymui.



16 pav. Terminų pildymo veiklos diagrama.

### 3.5.2.1.3 Terminų pildymo sekų diagrama.

Pirmiausia terminas yra užpildomas pagal vienintelį šabloną, o po to išsaugomas veiklos taisyklių saugykloje ( 17 pav.).

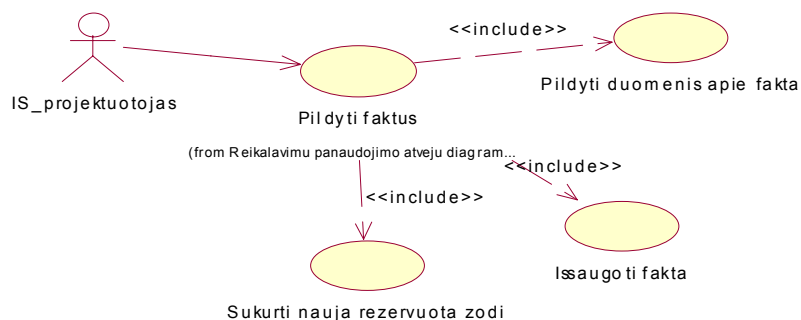


17 pav. Terminų pildymo sekų diagrama.

### 3.5.2.2 Faktų pildymo diagramos.

#### 3.5.2.2.1 Faktų pildymo panaudojimo atvejų diagrama.

Faktų pildymas priklauso nuo terminų užpildymo ir rezervuotų žodžių egzistavimo. Faktų pildymo šablonas taip pat yra vienas. Jame skiriasi tik rezervuotų žodžių reikšmė. Šabloną sudaro du terminai ir rezervuotas žodis tarp jų. Jei nėra užpildytų reikalingų terminų, juos galima užpildyti lygiagrečiai su faktų pildymu pasirinkus kitą meniu punktą (18 pav.).



18 pav. Faktų pildymo panaudojimo atvejų diagrama.

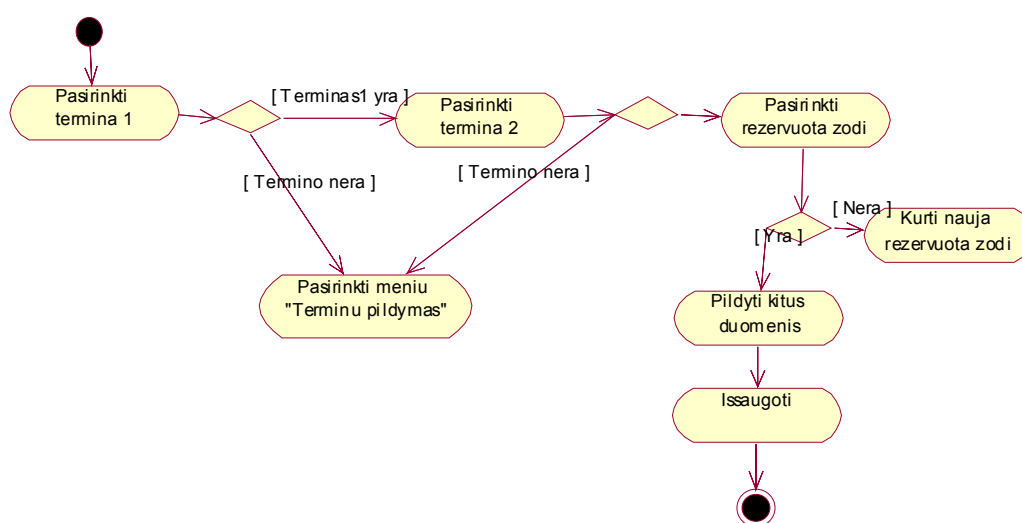
#### 3.5.2.2.2 Faktų pildymo veiklos diagrama.

Reikalavimų modelyje naudojamos veiklos diagramos, susietos su panaudojimo atvejais ir aprašančios panaudojimo atvejo žingsnius. Panaudojimo atvejui “Pildyti faktus” sudaroma veiklos diagrama.

Panaudojimo atvejo “Pildyti faktus” specifikacija

<b>Panaudojimo atvejis</b>	Pildyti faktus
<b>Aktorius</b>	IS projektuotojas
<b>Sistema</b>	Veiklos taisyklių redaktorius
<b>Sužadinimo įvykis</b>	IS projektuotojas gauna papildomus reikalavimus informacijos sistemai.
<b>Prieš sąlyga</b>	Turi būti užpildyti reikalingi faktų pildymui terminai.
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1.Pasirinkti meniu punktą “Faktų pildymas” (šis žingsnis apima panaudojimo atvejį “Pildyti faktus”) 2.Pasirinkti pirmą terminą.  3.Pasirinkti rezervuotą žodį.  4.Pasirinkti antrą terminą.  5.Sukurti naują rezervuotą žodį. 6.Įrašyti fakto aprašymą. 7.Įrašyti ryšio prasmę. 8.Išsaugoti ir pildyti sekantį. 9.Baigti faktų pildymą.	1.1. Sistema paleidžia faktų pildymo formą.  2.1 Jei termino nėra, sistema lygiagrečiai leidžia pildyti terminus pasirinkus meniu punktą „Terminų pildymas“. 3.1 Sistema tikrina, ar yra reikalingas rezervuotas žodis. Jei toks yra, sistema įrašo į faktą, o jei nėra pereina į 5 žingsnį. 4.1 Jei termino nėra, sistema lygiagrečiai leidžia pildyti terminus.  8.1 Sistema išsaugo užpildytą faktą. 9.1 Jei nepildomas sekantis faktas, baigiamas jų pildymas.
<b>Po sąlyga</b>	Sukuriamas naujas faktas, kuris vėliau naudojamas veiklos taisyklių pildymui.
<b>Alternatyvos (nesėkmės atvejai)</b>	2.1 ir 4.1 a. Nėra reikalingo termino. Galima susikurti naujus terminus, pasirinkus meniu punktą „Terminai“.

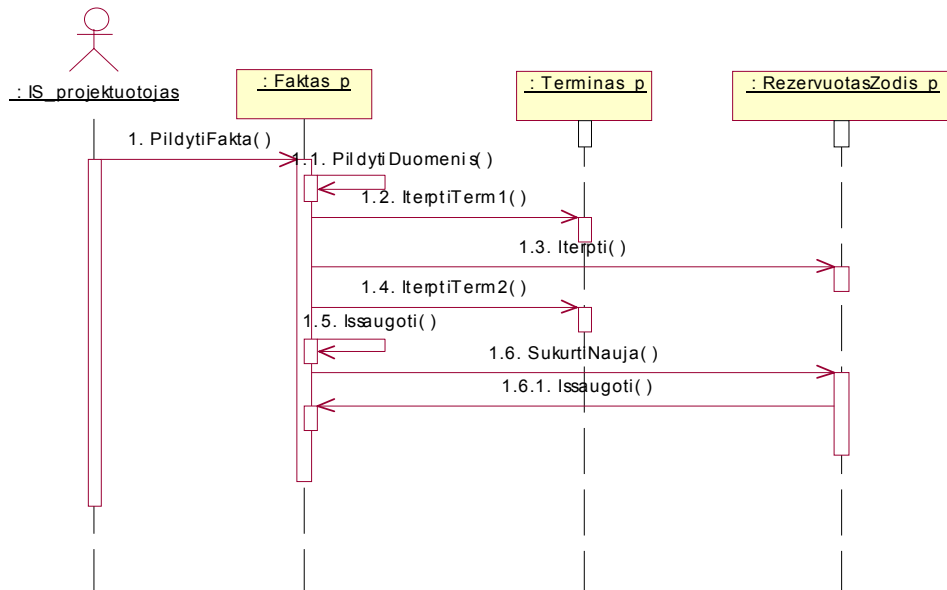
Aparašytam scenarijui sudaroma veiklos diagrama (19 pav.):



19 pav. Faktų pildymo veiklos diagrama.

### 3.5.2.2.3 Faktų pildymo sekų diagrama.

Panaudojimo atvejų specifikaciją galima atvaizduoti ir sekų diagramomis. 20 paveikslas iliustruoja faktų pildymo seką.

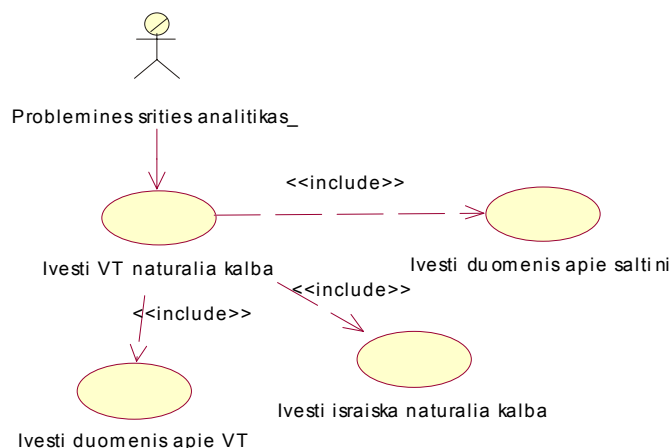


20 pav. Faktų pildymo sekų diagrama.

### 3.5.2.3 Natūralios veiklos taisyklės įvedimo diagramos.

#### 3.5.2.3.1 Natūralios veiklos taisyklės įvedimo panaudojimo atvejų diagrama.

Probleminės srities analitikas išanalizavęs sistemos probleminę sritį, suveda natūralias veiklos taisykles. IS projektuotojas vėliau jas savo ruožtu nagrinėja, analizuoja ir formalizuoja (21 pav.).

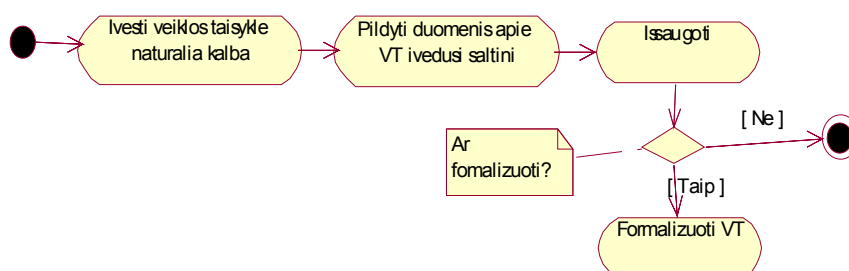


21 pav. Natūralios veiklos taisyklės įvedimo panaudojimo atvejų diagrama.

#### 3.5.2.3.2. Natūralios veiklos taisyklės įvedimo veiklos diagrama.

Panaudojimo atvejo “Ivesti VT natūralia kalba” specifikacija.

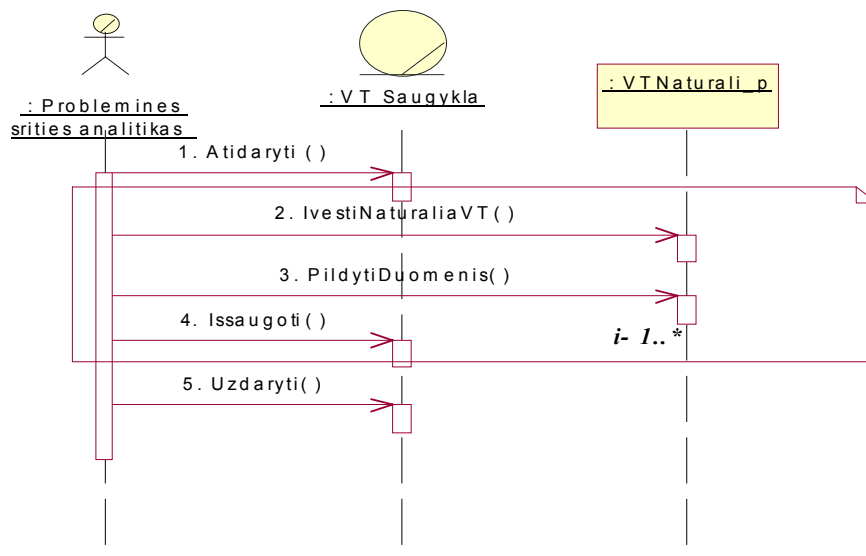
<b>Panaudojimo atvejis</b>	Ivesti VT natūralia kalba
<b>Aktorius</b>	Probleminės srities analitikas
<b>Sistema</b>	Veiklos taisyklių redaktorius
<b>Sužadavimo įvykis</b>	Probleminės srities analitikas išanalizuoja pateiktus reikalavimus sistemai.
<b>Prieš sąlyga</b>	Turi būti žinomi vartotojo reikalavimai sistemai.
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
1. Įrašyti veiklos taisyklę natūralia kalba. 2. Pildyti duomenis apie veiklos taisyklę įvedusį šaltinį. 3. Išsaugoti. 4. Formalizuoti veiklos taisyklę. 5. Baigti veiklos taisyklių įvedimą natūralia kalba.	1.1. Sistema pateikia formą, kurioje įrašoma veiklos taisyklės išraiška natūralia kalba. 2.1 Įvesti šaltinio pavadinimą ir identifikavimo numerį.  3.1 Sistema užklausia, ar išsaugoti įrašytą ir įvesti kitą veiklos taisyklę natūralia kalba. 4.1 Sistema užklausia, ar veiklos taisyklė bus formalizuojama iš karto. Jei ne, sistema užbaigia veiklos taisyklių įvedimą natūralia kalba.
<b>Po sąlyga</b>	Natūrali veiklos taisyklė toliau sistemoje formalizuojama.
<b>Alternatyvos (nesėkmės atvejai)</b>	3.1 a. Įvedinėjama kita veiklos taisyklė natūralia kalba. Sistema pasiūlo įvesti naują veiklos taisyklę natūralia kalba. Jei ne, tai einame į 5 žingsnį, jei taip grįžtame į 1 žingsnį.



22 pav. Natūralios veiklos taisyklės įvedimo veiklos diagrama.

### 3.5.2.3.3. Natūralios veiklos taisyklės įvedimo sekų diagrama.

Sekų diagramoje atvaizduota veiksmų eiga, kaip natūralios veiklos taisyklės įvedamos į sistemą (23 pav.).

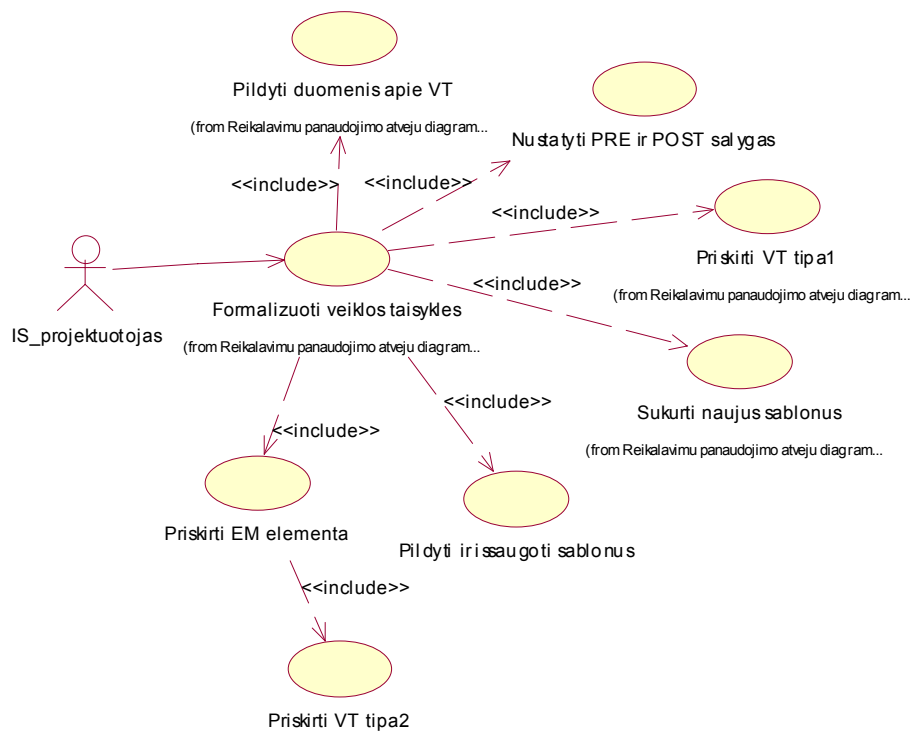


23 pav. Natūralios veiklos taisyklės įvedimo sekų diagrama.

### 3.5.2.4 Veiklos taisyklių formalizavimo diagramos.

#### 3.5.2.4.1 Veiklos taisyklių formalizavimo panaudojimo atvejų diagrama.

24 paveiksle pateikta veiklos taisyklių formalizavimo panaudojimo atvejų diagrama. Joje aprašyti veiksmai, kurie atliekami norint formalizuoti veiklos taisykles.



24 pav. Veiklos taisyklių formalizavimo panaudojimo atvejų diagrama.

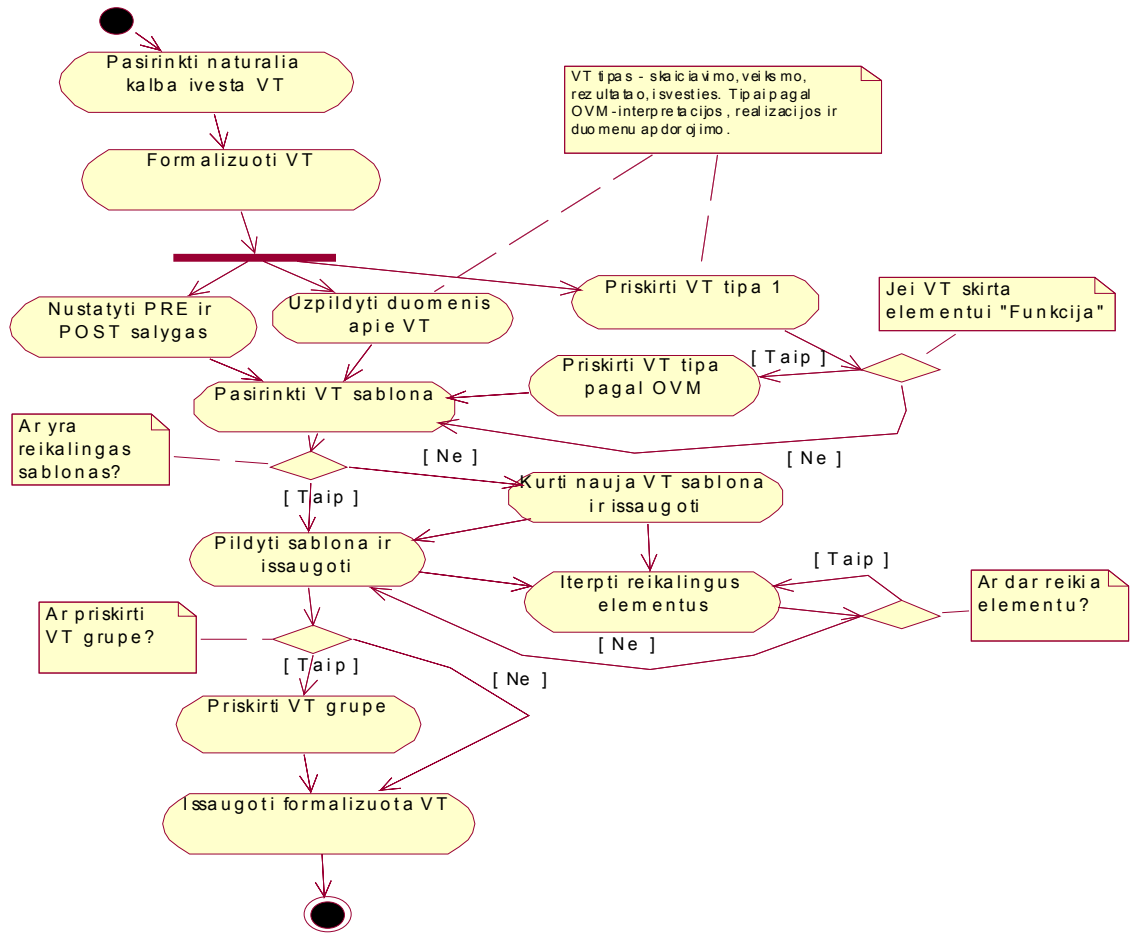


3.5.2.4.1 Veiklos taisyklių formalizavimo veiklos diagrama.

Panaudojimo atvejo “Formalizuoti veiklos taisykles” specifikacija.

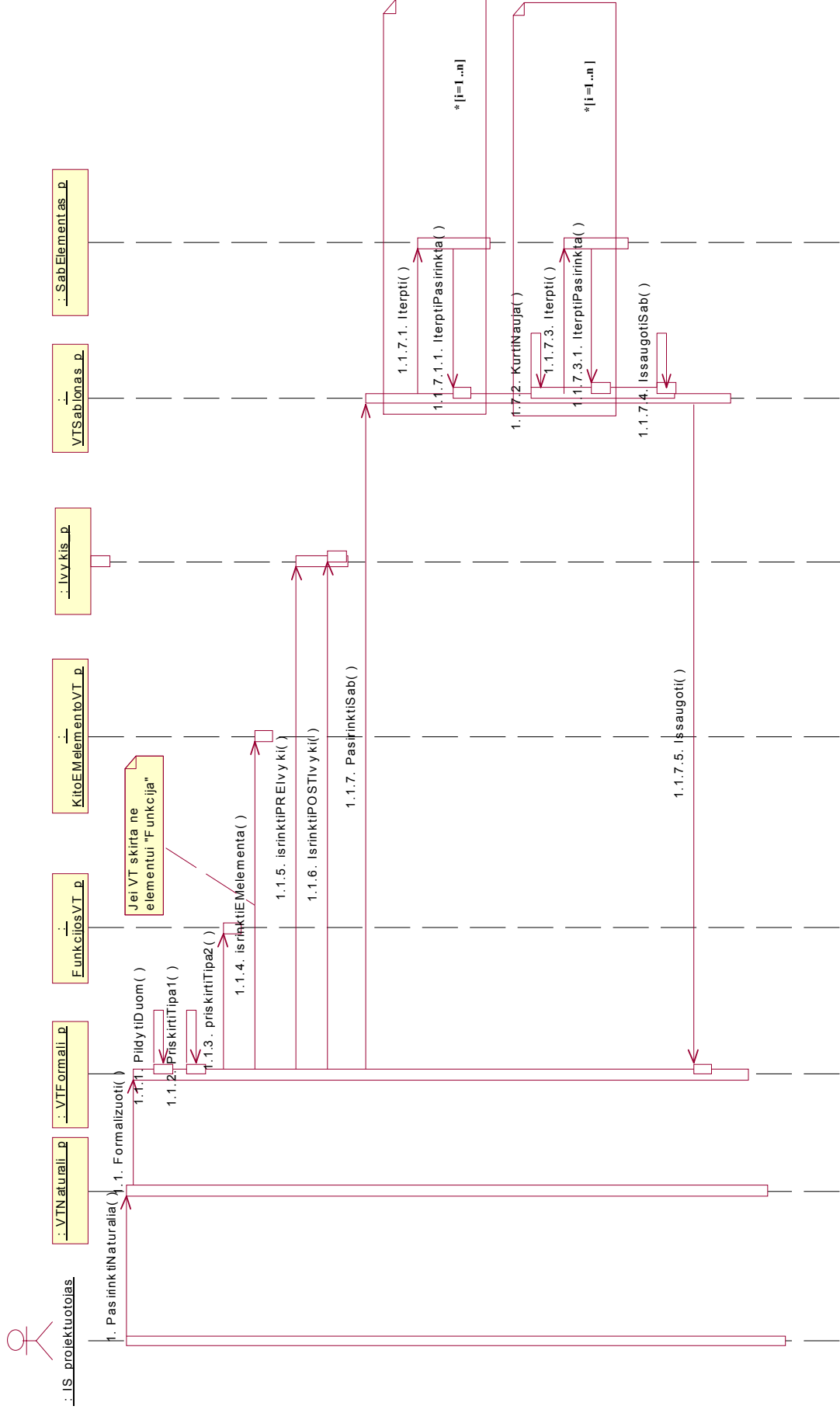
7 lentelė

<b>Panaudojimo atvejis</b>	Formalizuoti veiklos taisykles
<b>Aktorius</b>	IS projektuotojas
<b>Sistema</b>	Veiklos taisyklių redaktorius
<b>Sužadinimo įvykis</b>	IS projektuotojas įveda veiklos taisyklę natūralia kalba.
<b>Prieš sąlyga</b>	Turi būti bent viena veiklos taisyklė, įvesta natūralia kalba.
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
<p>1. Pasirinkti natūralia kalba įvestą veiklos taisyklę.</p> <p>2. Formalizuoti pasirinktą veiklos taisyklę.</p> <p>3. Užpildyti duomenis apie veiklos taisyklę.</p> <p>4. Priskirti veiklos taisyklės tipą.</p> <p>5. Nustatyti PRE ir POST sąlygas.</p> <p>6. Pasirinkti veiklos taisyklės šabloną.</p> <p>7. Pildyti šabloną ir išsaugoti.</p> <p>8. Įterpti reikalingus elementus.</p> <p>9. Priskirti VT grupę.</p> <p>10. Išsaugoti formalizuotą VT.</p> <p>11. Veiklos taisyklės formalizavimo pabaiga.</p>	<p>1.1. Sistema išveda pasirinktą natūralią veiklos taisyklę ir leidžia formalizuoti.</p> <p>3.1 Įvesti veiklos taisyklės aprašymą, pavadinimą, išraišką, sukūrimo, redagavimo ir galiojimo datas. Priskirti elementą iš OVM (organizacijos veiklos modelis).</p> <p>4.1 Priskirti veiklos taisyklės tipą. Jei veiklos taisyklė skirta elementui “Funkcija”, priskirti tipą pagal OVM.</p> <p>5.1 Nustatomos PRE ir POST įvykiai ir veiklos taisyklės, kurioms įvykus galima būtų vydyti formalizuojamą veiklos taisyklę.</p> <p>6.1 Parenkamas šablonas pagal pasirinktą veiklos taisyklės tipą.</p> <p>7.1 Užpildomas šablonas pagal įrašytą veiklos taisyklės išraišką ir išsaugomas.</p> <p>8.1 Šablonas pildomas tikrinant šablonų pildymo medį ir įterpinėjant reikalingus elementus (faktus, rezervuotus žodžius, formules, kitas veiklos taisykles) į pasirinktą šabloną.</p> <p>9.1 Jei reikalinga priskirti veiklos taisyklės grupę, pagal kurią jos būtų grupuojamos.</p> <p>10.1 Išsaugoti formalią veiklos taisyklės išraišką.</p> <p>11.1 Sistema užklausia, ar tęsti veiklos taisyklių formalizavimą. Jei taip, tai pereina į 1 žingsnį, o jei ne nutraukia veiklos taisyklių formalizavimą.</p>
<b>Po sąlyga</b>	Formali veiklos taisyklės išraiška išsaugoma duomenų bazėje ir galima naudoti organizacijos veiklos modelyje.
<b>Alternatyvos (nesėkmės atvejai)</b>	<p>5.1 a. Nėra reikalingo šablono pagal natūralią veiklos taisyklės išraišką. Sistema pasiūlo sukurti naują šabloną, sukuriant jį panašiu principu kaip ir pildymo atveju, t.y. įterpinėjant elementus į kuriamą struktūrą.</p> <p>6.1 a. Nėra reikalingo fakto ar rezervuoto žodžio. Sistema leidžia lygiagrečiai su veiklos taisyklių formalizavimu pildyti faktus ir kurti reikalingus rezervuotus žodžius.</p>
<b>Veiklos taisyklės</b>	Vartotojas turi teisingai įvesti duomenis pagal natūralia kalba įvestos veiklos taisyklės išraišką.



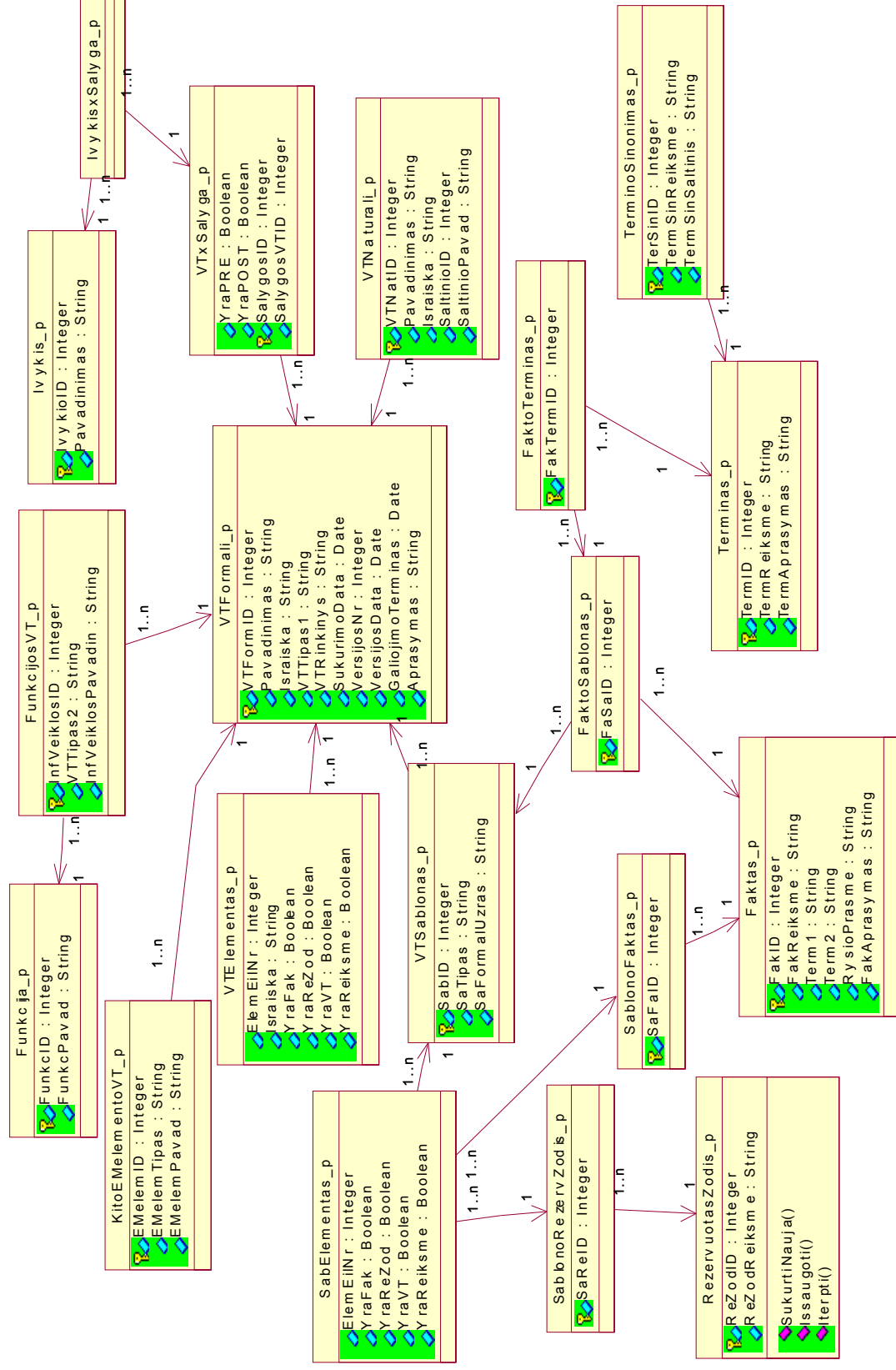
25 pav. Panaudojimo atvejo „Formalizuoti veiklos taisykles“ veiklos diagrama.

### 3.5.2.3.4 Veiklos taisyklių formalizavimo sekų diagrama.



26 pav. Veiklos taisyklių formalizavimo sekų diagrama.

### 3.5.3 Dalykinės srities klasių diagrama.



27 pav. Dalykinės srities klasių diagrama.



### 3.5.5 Reikalavimai sistemos funkcionalumui.

Kuriamos sistemos funkcionalumui keliami tokie reikalavimai:

- ❖ tikslumas (rezultato) – rezultatas turėtų būti pateikiamas kuo tikslesnis, kadangi šios sistemos rezultatas bus naudojamas kitose sistemose;
- ❖ sąveika su kitomis sistemomis – ši sistema sąveikaus su organizacijos veiklos modelio pagrindu sukurta informacijos sistema, todėl turi būti numatytas apjungimas per OVM duomenų bazės lentelę „Veiklos taisyklė“;
- ❖ patikimumas – sistema turėtų būti užbaigta, nes neužbaigtumas mažina jos patikimumą;
- ❖ apsimokamumas – ši programa finansiniu požiūriu turėtų būti brangi ir nereikalaujanti daug resursų, nes jos kūrimui planuojama naudoti arba visiškai nemokamas, arba nebrangias programas;
- ❖ suprantamumas – planuojama sukurti sistemą, kuri būtų kuo lengviau suprantama, nes tai tik palengvina darbą su ja;
- ❖ efektyvumas
  - laiko parametrai – manoma, kad sistema veiks pakankamai greitai, nes duomenų bazėje yra ne daug testavimo duomenų, tačiau ir su nemažu kiekiu duomenų ji turėtų veikti gana greitai.
  - išteklių naudojimas – kuriamai sistemai nereikės labai daug resursų ir kompiuterio atminties.
- ❖ priežiūros savybės
  - analizės savybės – bus suteiktos galimybės analizuoti kuriamą sistemą, nes visose formose lentelės pavidale bus pateikiami egzistuojantys duomenų bazėje įrašai;
  - pakeitimų galimybės – pakeitimus visiškai nesudėtingai galima atlikti, turint Java Web Start paketą ir mokant programuoti Java programavimo kalba;
  - testavimo savybės – testavimui galima įvedinėti visokius duomenis, kuriuos vėliau galima pašalinami.
- ❖ perkeliamumas
  - įdiegimo savybės – numatyta, kad šios sistemos įdiegimas bus paprastas, paleidžiant vieną vykdomąjį failą;
  - prisitaikymo savybės – sistema turės veikti visose operacinėse aplinkose, nes Java programavimo kalba pritaikyta naudojimui jose visose.

## 3.6 Sistemos projektas.

### 3.6.1 Projekto tikslas.

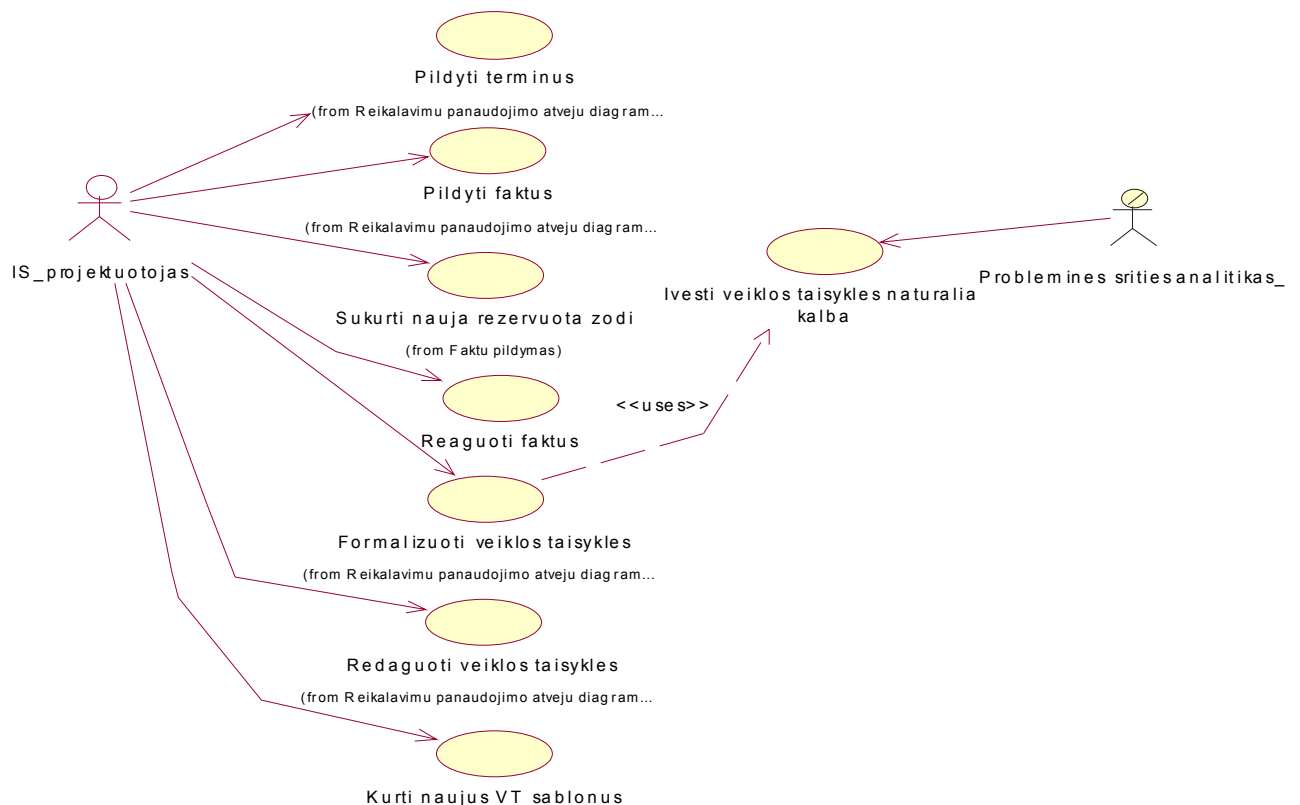
Magistrinio darbo projekto tikslas yra sukurti programą, kuri suteiktų galimybę:

- ❖ įvesti veiklos taisykles natūralia kalba;
- ❖ formalizuoti veiklos taisykles naudojant šablonus;

- ❖ sukurti naujus veiklos taisyklių šablonus;
- ❖ pildyti terminus ir faktus;
- ❖ kurti naujus rezervuotus žodžius.

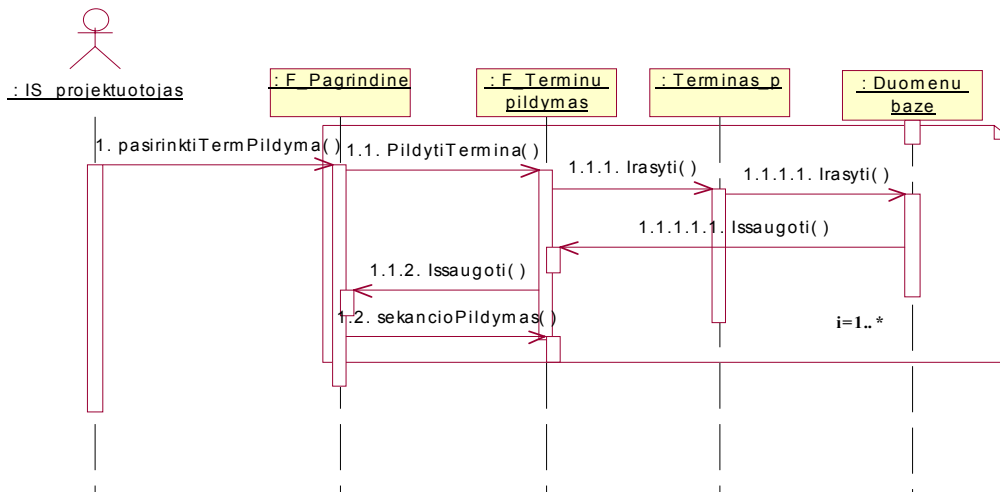
### 3.6.2 Sistemos panaudojimo atvejų ir sekų diagramos.

Reikalavimų modelyje sekų diagramomis galima atvaizduoti panaudojimo atvejų scenarijus. Tokios sekų diagramos dar vadinamos sistemos sekų diagramomis – jos rodo aktoriaus sąveikas su sistema. Panašiomis sekų diagramomis galima vaizduoti sistemų sąveiką. 29 paveiksle pateikta sistemos panaudojimo atvejų diagrama. Joje atvaizduoti aktoriai ir jų atliekami veiksmai sistemoje.



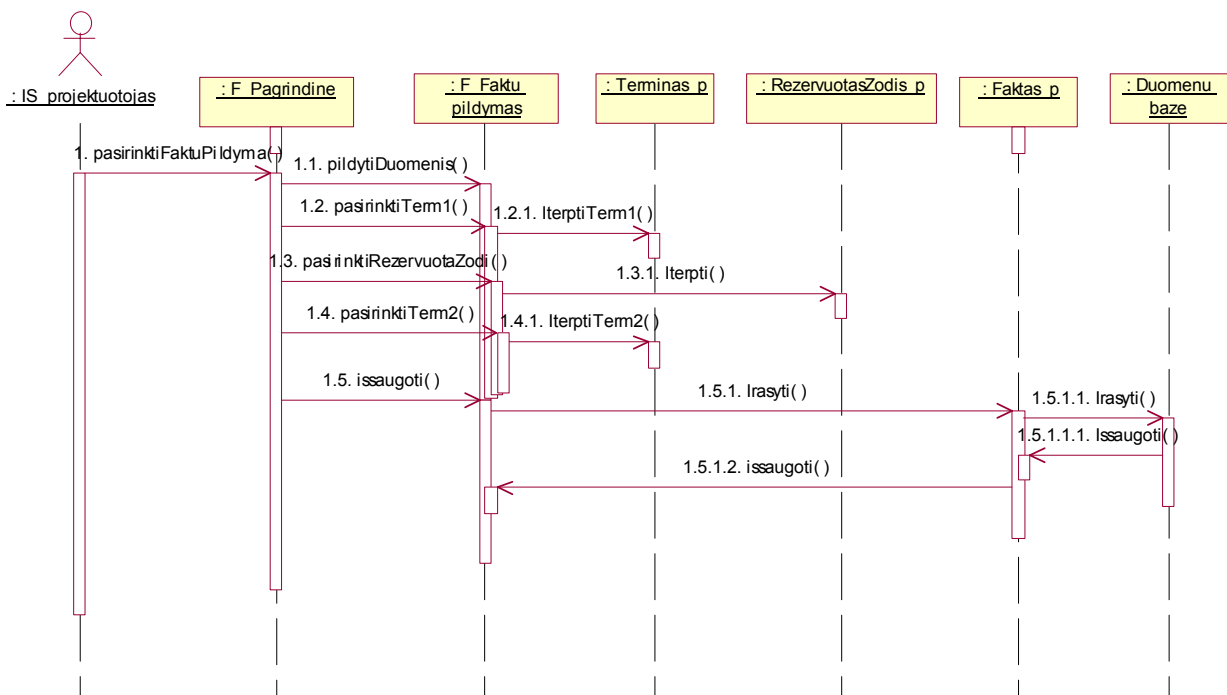
29 pav. Sistemos panaudojimo atvejų diagrama.

Terminų pildymo sekų diagrama pateikta 30 paveiksle. Joje egzistuoja ciklas, nes terminus galime vesti ne po vieną, o iš karto visus, kiek reikalinga veiklos taisyklių formalizavimui.



30 pav. Terminų pildymo sekų diagrama.

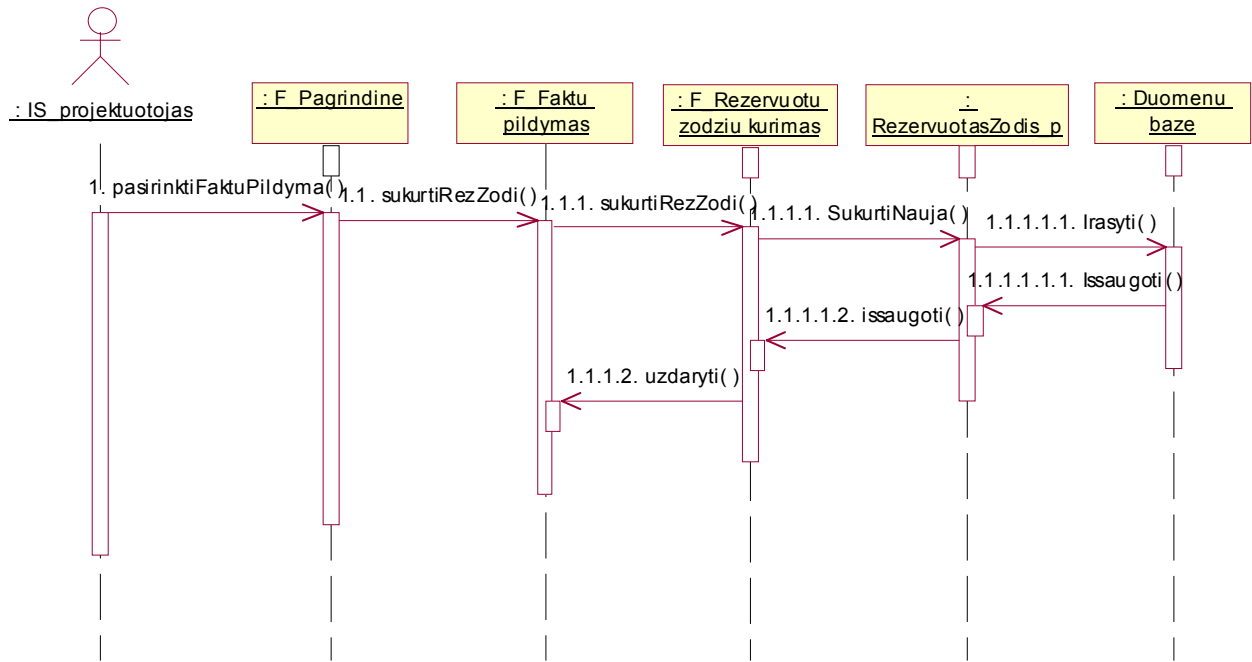
31 paveiksle pateikta faktų pildymo sekų diagrama. Faktai pildomi pagal vienintelį šabloną:  $\langle Term1 \rangle \langle rezervuotas \text{ žodis} \rangle \langle Term2 \rangle$ . Faktų pildymas labai svarbus dalykas šioje sistemoje, nes nuo jų korektiško užpildymo priklauso visos veiklos taisyklių formalizavimas.



31 pav. Faktų pildymo sekų diagrama.

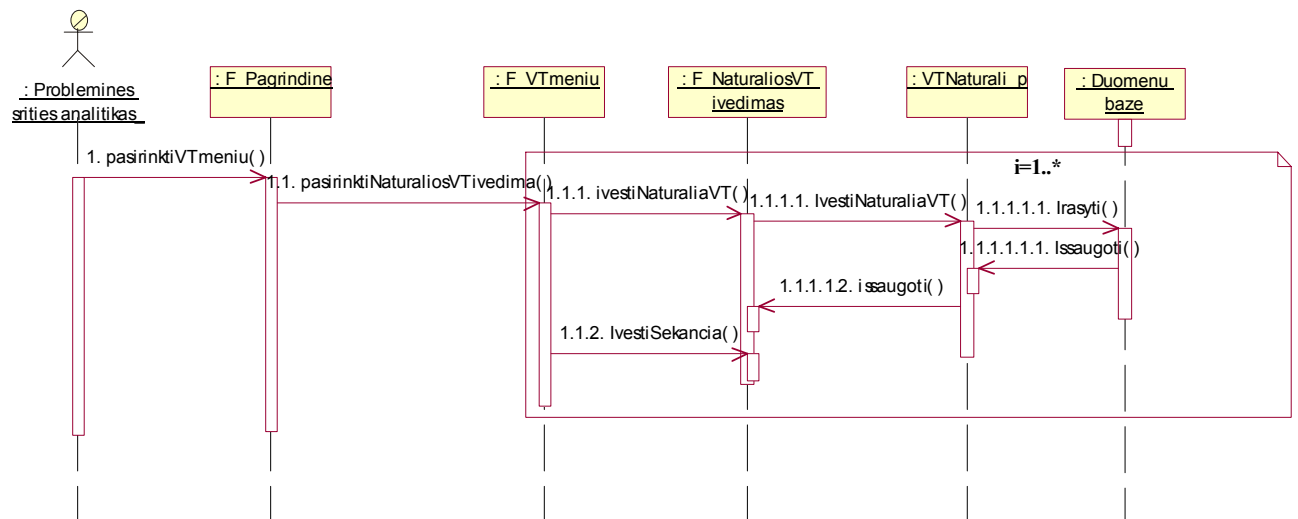
Pildant faktus ir formalizuojant veiklos taisykles, dažnai prireiks naujų rezervuotų žodžių, kurių dar nėra duomenų bazėje. Tai galima padaryti pasirinkus meniu punktą “Naujo rezervuoto žodžio kūrimas”. Jų kūrimas visiškai nesudėtingas, tereikia įvesti pačio žodžio reikšmę ir aprašymą, kuris jį charakterizuoja (32 pav.).





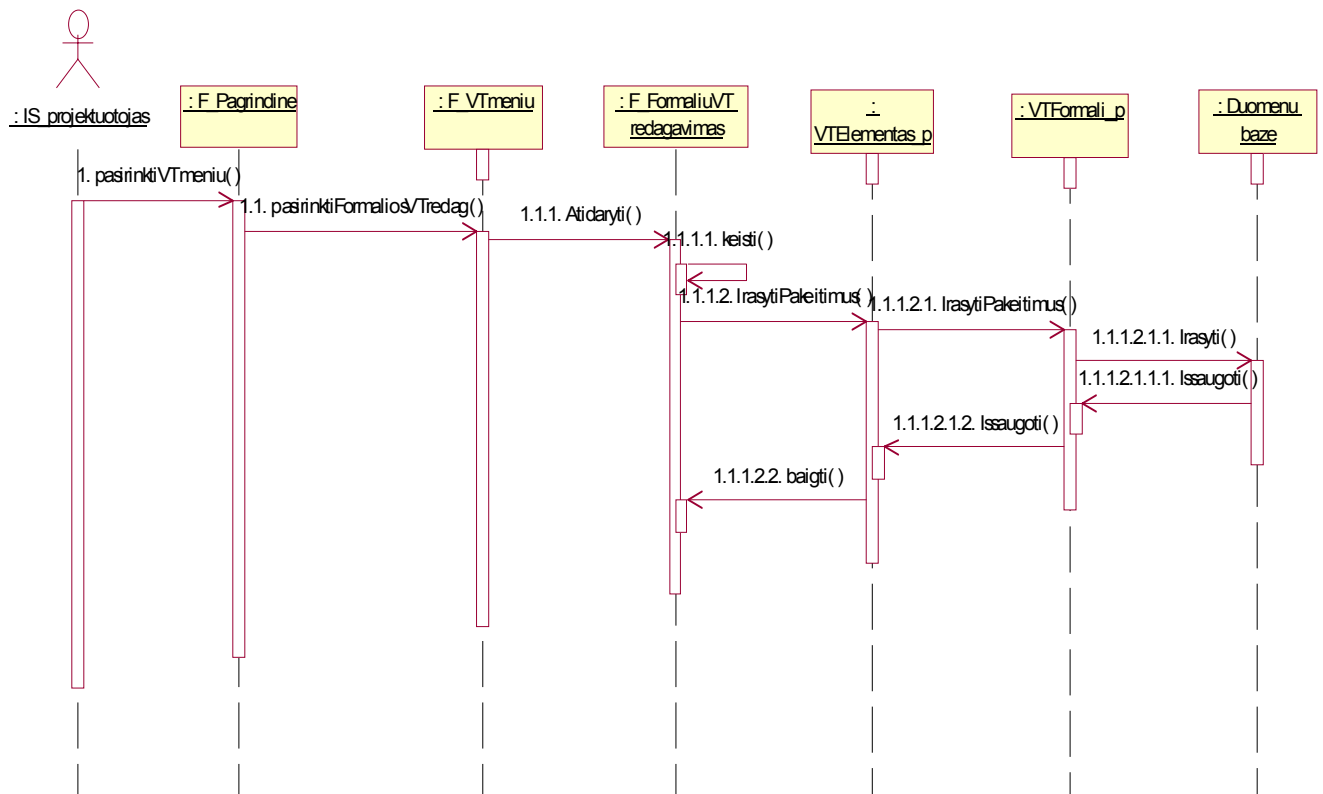
32 pav. Naujo rezervuoto žodžio kūrimo sekų diagrama.

Natūralių veiklos taisyklių suvedimą į sistemą atlieka probleminės srities analitikas, nes jis analizuoja iš vartotojų gautus reikalavimus sistemai. Jis nesiaiškina tų veiklos taisyklių sudėties, jų tipo. Probleminės srities analitikas tiesiog įveda veiklos taisyklę natūralia kalba, neformalizuotai ir užpildo informacija apie veiklos taisyklę įvedusį šaltinį. 33 paveiksle pateikta veiklos taisyklių įvedimo natūralia kalba sekų diagrama.



33 pav. Natūralių veiklos taisyklių įvedimo sekų diagrama.

Formalių veiklos taisyklių redagavimo sekų diagrama pateikta 34 paveiksle.

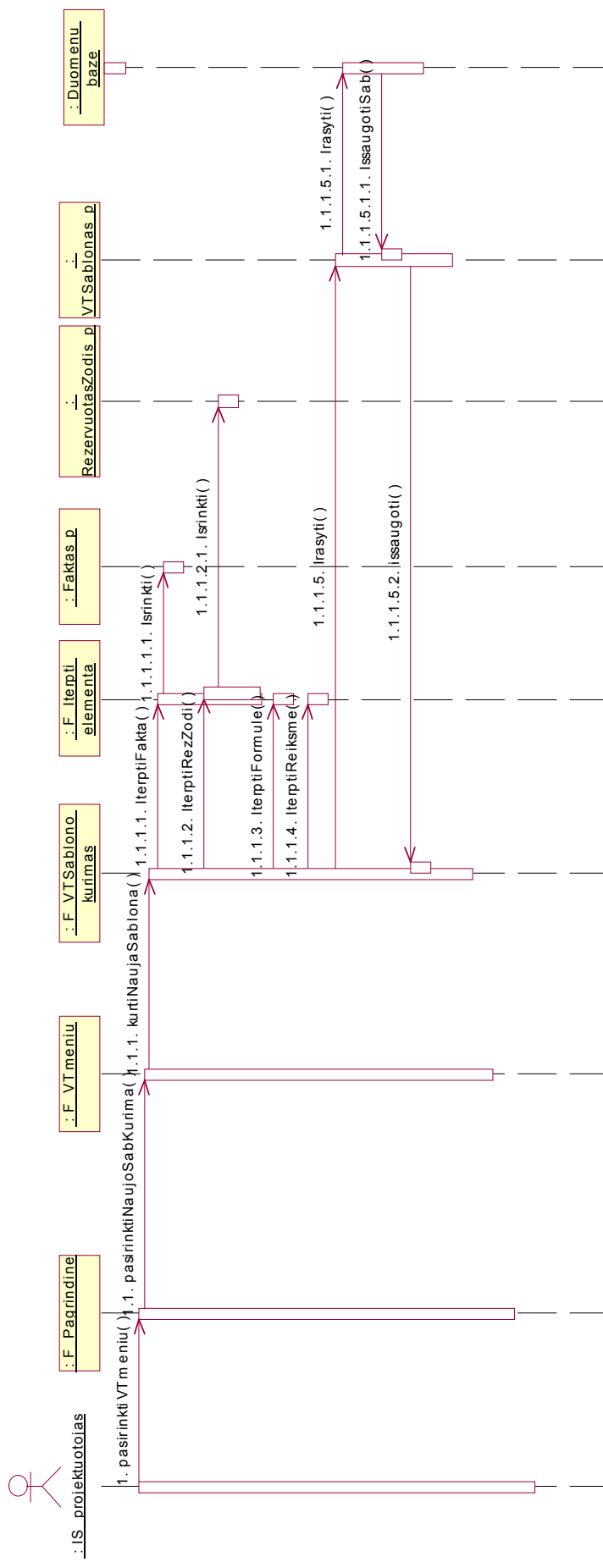


34 pav. Formalių veiklos taisyklių redagavimo sekų diagrama.

Natūralių veiklos taisyklių formalizavimas – sudėtingiausias veiksmas, atliekamas kuriamoje sistemoje. Po to kai probleminės srities analitikas suveda visas išanalizuotas veiklos taisykles (nebūtinai visas), IS projektuotojas ima jas išsamiai analizuoti, kas ir vadinama veiklos taisyklių formalizavimu. 35 paveiksle parodyta veiklos taisyklių formalizavimo sekų diagrama.

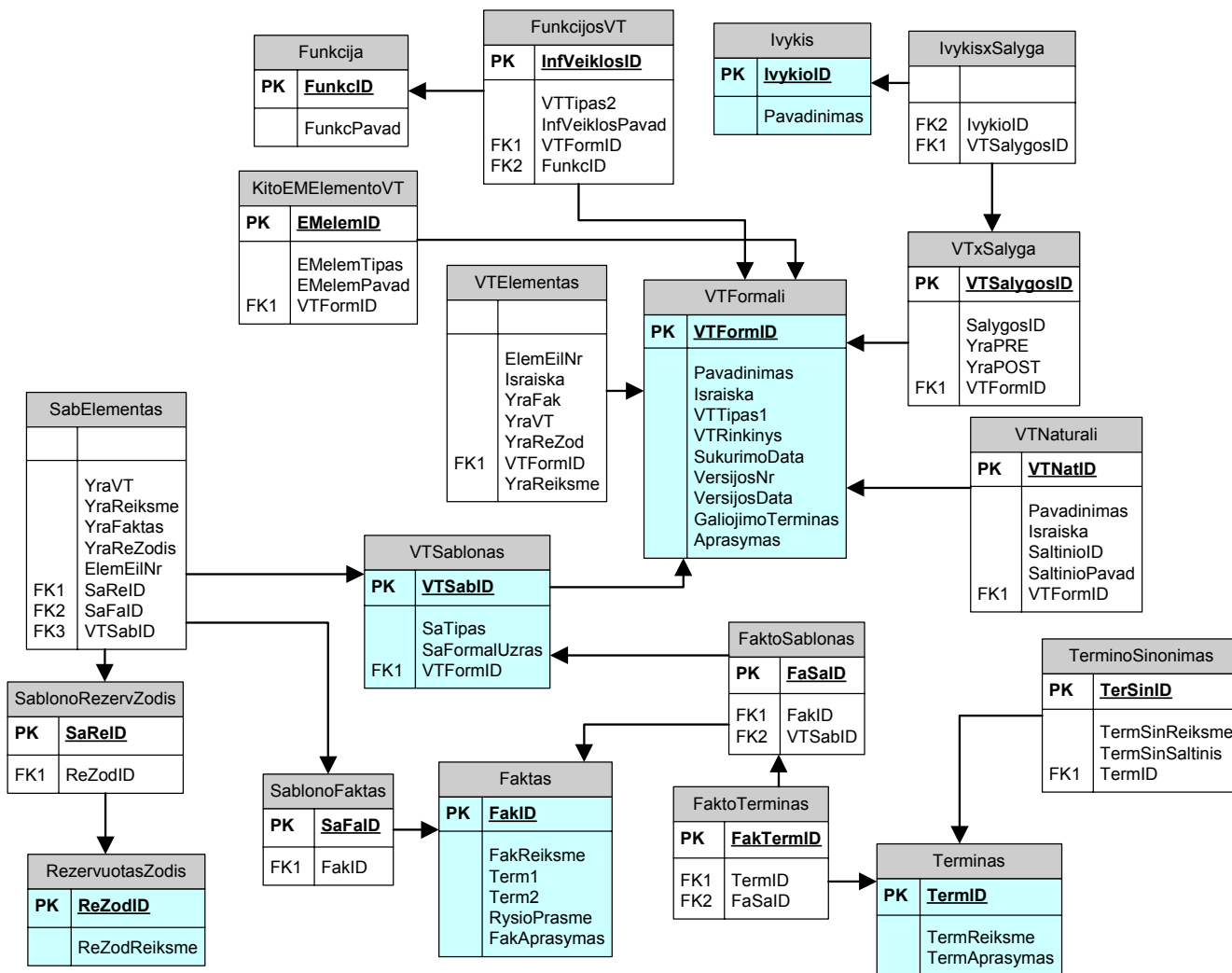


Jei formalizuojant veiklos taisyklės paaiškėja, kad nėra reikalingo šablono, galima susikurti naują. Kūrimas vyksta kaip ir formalizavimas, įterpinėjant reikalingus šablono elementus – faktus, rezervuotus žodžius, formules, reikšmes arba kitas veiklos taisyklės. Po sukūrimo tą šablona galima iš karto užpildyti reikalinga veiklos taisykle ir išsaugoti duomenų bazėje (36 pav.).



36 pav. Naujo veiklos taisyklės šablono kūrimas.

### 3.6.3 Duomenų bazės modelis.



37 pav. Duomenų bazės modelis.

8 lentelė

Esybė	Atributai	Aprašymas	
<b>VTNaturaliai</b>		<b>Šioje lentelėje saugomos veiklos taisyklės, užrašytos natūralia kalba.</b>	
	VTNatID	Natūralios veiklos taisyklės ID numeris (identifikatorius), užpildomas automatiškai.	<b>integer</b>
	Pavadinimas	Natūralios veiklos taisyklės pavadinimas.	<b>string</b>
	Israiska	Natūralios veiklos taisyklės pilnas tekstas.	<b>string</b>
	SaltinioID	Šaltinio, įvedusio natūralią veiklos taisyklę, ID numeris (identifikatorius). Užpildomas automatiškai.	<b>integer</b>
	SaltinioPavad	Šaltinio, įvedusio veiklos taisyklę, pavadinimas.	<b>string</b>
<b>VTFormali</b>		<b>Šioje lentelėje saugomos formalizuotos veiklos taisyklės.</b>	
	VTFormID	Formalios veiklos taisyklės ID numeris (identifikatorius). Užpildomas automatiškai.	<b>integer</b>
	Pavadinimas	Formalios veiklos taisyklės pavadinimas.	<b>string</b>
	Israiska	Formalios veiklos taisyklės pilnas tekstas.	<b>string</b>

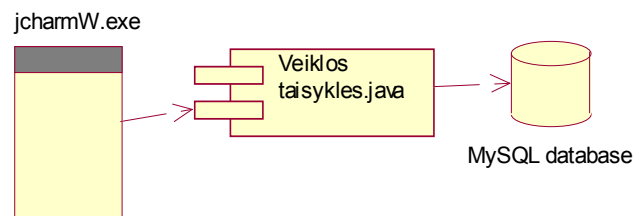
	VTipas1	Veiklos taisyklės tipas pagal paskirtį.	string
	VTRinkinys	Veiklos taisyklių rinkinys identifikuoja, kokiam OVM elementui priklauso veiklos taisyklė. Galimos reikšmės: procesui, aktoriui ir kt.	string
	SukurimoData	Veiklos taisyklės sukūrimo data. Užpildoma automatiškai.	date
	VersijosNr	Veiklos taisyklės redagavimo versijos numeris. Gali būti užpildoma automatiškai.	integer
	VersijosData	Veiklos taisyklės redagavimo versijos data.	date
	GaliojimoTerminas	Veiklos taisyklės galiojimo terminas, iki kurio ji privalo įtakoti tam tikrai veiklos taisyklei priskirtą elementą.	date
	Aprasymas	Konkretus veiklos taisyklės aprašymas.	string
<b>FunkcijosVT</b>		<b>Šioje lentelėje saugomos informacinės veiklos sudarančios funkcija.</b>	
	InfVeiklosID	Informacinės veiklos ID numeris (identifikatorius). Užpildomas automatiškai.	integer
	VTipas2	Veiklos taisyklės tipas pagal organizacijos modelio informacinės veiklas.	string
	InfVeiklosPavad	Tai informacinių veiklų pilnas pavadinimas	string
<b>Funkcija</b>		<b>Šioje lentelėje saugoma informacija apie OVM egzistuojančias funkcijas.</b>	
	FunkcID	Funkcijos ID numeris (identifikatorius). Užpildomas automatiškai.	integer
	FunkcPavad	Funkcijos pavadinimas.	string
<b>KitoElementoVT</b>		<b>Šioje lentelėje saugomi iš OVM gaunami elementai.</b>	
	EMelemID	Elemento ID numeris (identifikatorius). Užpildomas automatiškai.	integer
	EMelemTipas	Tai organizacijos veiklos modelyje egzistuojantis elemento tipas. Galimos reikšmės: procesas, aktorius, tikslas ir kt.	string
	EMelemPavad	Tai konkreti OVM elemento reikšmė. Jei OVM elementas – aktorius, tai galima reikšmė tuomet “darbininkas”.	string
<b>Ivykis</b>		<b>Ši lentelė skirta įvykių saugojimui.</b>	
	IvykioID	Įvykio ID numeris (identifikatorius). Užpildomas automatiškai.	integer
	Pavadinimas	Įvykio pavadinimas.	string
<b>IvykisxSalyga</b>		<b>Tarpinė lentelė skirta “daug-su-daug” ryšiui realizuoti tarp lentelių <u>Ivykis</u> ir <u>VTxSalyga</u>.</b>	
<b>VTxSalyga</b>		<b>Tarpinė lentelė, skirta “daug-su-daug” ryšiui realizuoti tarp lentelių <u>IvykisxSalyga</u> ir <u>VTFormali</u>.</b>	
	VTsalygosID	PRE ir POST sąlygos konkrečiai VT ID numeris (identifikatorius). Užpildomas automatiškai.	integer
	SalygosID	PRE ir POST sąlygos ID numeris (identifikatorius). Užpildomas automatiškai.	integer
	YraPRE	Atributas nusako, ar tai PRE sąlyga. Jei taip, tai reikšmė “TRUE”.	boolean
	YraPOST	Atributas nusako, ar tai POST sąlyga. Jei taip, tai reikšmė “TRUE”.	boolean
<b>VTElementas</b>		<b>Šioje lentelėje saugomi konkretūs veiklos taisyklių egzemplioriai.</b>	
	ElemEilNr	Elemento veiklos taisyklėje eilės numeris.	integer
	Israiska	Konkreto veiklos taisyklės egzemplioriaus išraiška.	string

	YraFak	Jei tai Faktas konkrečiame veiklos taisyklės egzemplioriuje, tai TRUE reikšmė.	<b>boolean</b>
	YraReZod	Jei tai Rezervuotas žodis konkrečiame veiklos taisyklės egzemplioriuje, tai TRUE reikšmė.	<b>boolean</b>
	YraReiksme	Jei tai reikšmė įrašoma ranka konkrečiame veiklos taisyklės egzemplioriuje, tai TRUE reikšmė.	<b>boolean</b>
<b>SabElementas</b>		<b>Šioje lentelėje saugoma informacija apie bet kuriame šablone esančių elementų eilės tvarką.</b>	
	ElemEilNr	Elementų eilės numeris šablone.	<b>integer</b>
	YraFak	Jei tai Faktas bet kuriame šablone, tai TRUE reikšmė.	<b>boolean</b>
	YraReZod	Jei tai Rezervuotas žodis bet kuriame šablone, tai TRUE reikšmė.	<b>boolean</b>
	YraReiksme	Jei tai reikšmė įrašoma ranka kuriamame šablone, tai TRUE reikšmė.	<b>boolean</b>
<b>VTSablonas</b>		<b>Lentelėje kaupiama informacija apie šablonus ir jų tipus.</b>	
	SabID	Šablono ID numeris (identifikatorius). Užpildomas automatiškai.	<b>integer</b>
	SaTipas	Šablono tipas.	<b>string</b>
	SaFormalUzras	Formalus kiekvieno šablono užrašas.	<b>string</b>
<b>SablonoRezervZodis</b>		<b>Tarpinė lentelė, kuri realizuoja ryšį n:m tarp lentelių RezervuotasZodis ir VTSablonas.</b>	
	SaReID	Šablono rezervuoto žodžio ID numeris (identifikatorius). Užpildomas automatiškai.	<b>integer</b>
<b>RezervuotasZodis</b>		<b>Lentelėje saugomi šablonuose panaudojami rezervuoti žodžiai.</b>	
	ReZodID	Rezervuoto žodžio ID numeris (identifikatorius). Užpildomas automatiškai.	<b>integer</b>
	ReZodReiksme	Kokrečiau rezervuoto žodžio reikšmė.	<b>string</b>
<b>SablonoFaktas</b>		<b>Tai pagalbinė lentelė, pašalinanti “daug su daug” ryšį DB loginėje schemoje tarp lentelių SabElementas ir Faktas. Ji apibūdina tai, kad šablonuose naudojami faktai.</b>	
	SaFaID	Šablono fakto ID numeris (identifikatorius). Užpildomas automatiškai.	<b>integer</b>
<b>FaktoSablonas</b>		<b>Tai pagalbinė lentelė, pašalinanti “daug su daug” ryšį DB loginėje schemoje tarp lentelių VTSablonas ir Faktas. Ji apibūdina tai, kad faktams sudaryti reikalingi šablonai.</b>	
	FaSaID	Fakto šablono ID numeris (identifikatorius). Užpildomas automatiškai.	<b>integer</b>
<b>Faktas</b>		<b>Šioje lentelėje saugomi visi faktai.</b>	
	FakID	Fakto ID numeris (identifikatorius). Užpildomas automatiškai.	<b>integer</b>
	FakReiksme	Trumpai nusakoma fakto reikšmė.	<b>string</b>
	Term1	Identifikuoja pirmąjį fakto terminą (bendru atveju tai gali būti kažkokia Esybė).	<b>string</b>
	Term2	Identifikuoja antrąjį fakto terminą (tai gali būti kita Esybė arba esybės Savybė, t.y. Atributas).	<b>string</b>
	RysioPrasme	Nusako, koks ryšio tipas tarp dviejų Fakto terminų (galimos reikšmės: asociacija tarp esybių, apibendrinimo ryšys tarp esybių, savybės ryšys).	<b>string</b>
	FakAprasymas	Faktą charakterizuojantis aprašymas.	<b>string</b>
<b>FaktoTerminas</b>		<b>Pagalbinė lentelė, pašalinanti “daug su daug” ryšį loginėje DB schemoje tarp lentelių Terminas ir FaktoSablonas. Joje saugomi terminai, įeinantys į konkrečių faktų sudėtį.</b>	

	FakTermID	Fakto termino ID numeris (identifikatorius). Užpildomas automatiškai.	<b>integer</b>
<b>Terminas</b>		<b>Lentelėje saugomi visi terminai ir jų aprašymai.</b>	
	TermID	Termino ID numeris (identifikatorius). Užpildomas automatiškai.	<b>integer</b>
	TermReiksme	Termino reikšmė.	<b>integer</b>
	TermAprasymas	Terminą charakterizuojantis aprašymas.	<b>integer</b>
<b>TerminoSinonimas</b>		Lentelėje saugomi terminų sinonimai.	
	TerSinID	Termino sinonimo ID numeris (identifikatorius). Užpildomas automatiškai.	<b>integer</b>
	TermSinReiksme	Termino sinonimo reikšmė.	<b>string</b>
	TermSinSaltinis	Šaltinio pavadinimas, iš kurio gautas termino sinonimas.	<b>string</b>

### 3.6.4 Realizacijos modelis.

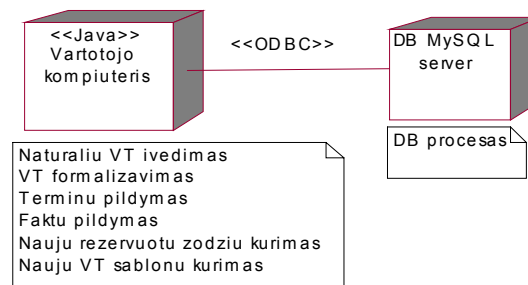
Šios sistemos realizacijos modelis nėra sudėtingas. Pirmiausia reikalinga paleisti pagrindinį vykdomąjį failą, kuris kreipiasi į Jbuilder 9 programą, o pastaroji į MySQL duomenų bazę, iš kurios ima ir kurioje saugo duomenis. Ši diagrama pateikta 38 paveiksle.



38 pav. Veiklos taisyklių redaktoriaus realizacijos schema.

### 3.6.5 Paskirstymo diagrama.

Paskirstymo diagrama (39 pav.) atvaizduoja, kokie įrenginiai dalyvauja veiklos taisyklių redaktoriaus veikimo metu. Pakanka tiesiog vartotojo kompiuterio ir duomenų bazių serverio, kuris taip pat gali būti instaliuotas tame pačiame vartotojo kompiuteryje.



39 pav. Paskirstymo diagrama.



### 3.6.6 Testavimo modelis bei duomenys, kontrolinis pavyzdys.

Testuosime sukurtą programą tam, kad įsitikinti kad jos funkcionalumas toks, koks buvo numatytas. Kontrolinį pavyzdį pasirinkime tokį:

Įveskime veiklos taisyklę natūralia kalba:” *Mažoji laivo ilgis turėtų būti ne didesnis negu 24m.*”

Tuomet turime susikurti

- ❖ terminus “Mažasis laivas”, “ilgis”;

	id_terminas	termreiksme	termaprasymas
1	1	Laivas	transporto priemonė
2	2	mazasis laivas	transporto priemonė
3	3	ilgis	ismatavimas

40 pav. Terminų sukūrimo langas.

- ❖ rezervuotus žodžius: “turi”, “ne didesnis negu”;

	id_rezervuotaszodis	rezodreiksme	rysioprasme
1	1	yra	savybe
2	2	ne didesnis negu	
3	3	turi	savybe

41 pav. Rezervuotų žodžių kūrimo langas.

- ❖ faktus: “Laivas turi ilgį”, “Mažasis laivas turi ilgį”;

Rezervuoti žodžiai **Iveskite reikalingus duomenis apie faktą:**

Terminas 1: Laivas      Rezervuotas žodis:      Terminas 2: ilgis

Ryšio prasmė: savybe

Fakto reikšmė: Laivo ilgis

Fakto aprašymas: laivo ilgis

	id_faktas	termreiksme	rezodreiksme	fakreiksme	rysioprasme	fakaprasymas
1	1	Laivas	ilgis	yra Laivo ilgis	savybe	laivo ilgis
2	2	mazasis laivas	ilgis	turi Mazojo laivo ilgis	savybe	savybes apibudir

42 pav. Faktų kūrimo langas.

- ❖ apribojimo šabloną: “faktas ne didesnis negu 24 m.”;

**Šablonai:**

Šablono tipas: apribojimo      Kurti šablono elementus

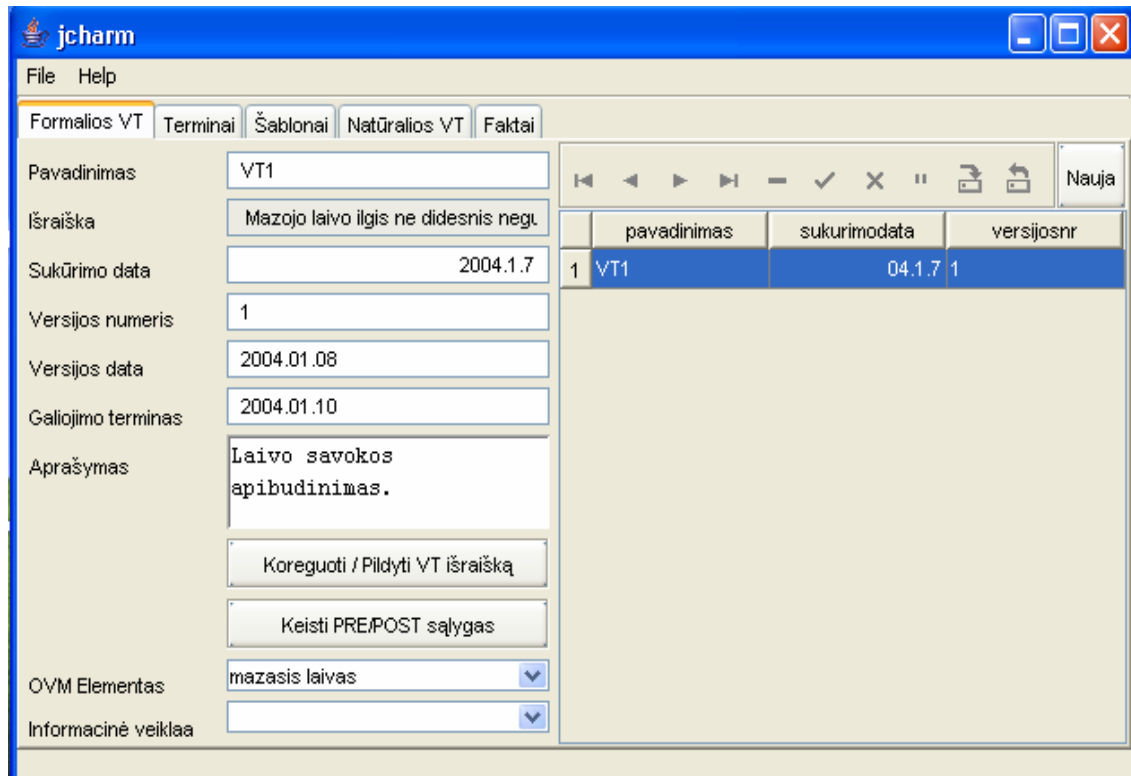
Šablono išraiška: faktas ne didesnis negu 24m.

Išsaugoti

	id_vtsablonas	satipas	saformaluzras
1	1	apribojimo	faktas ne didesnis negu 24m.

43 pav. Šablono sukūrimo langas.

Tai formalizuodami gausime veiklos taisyklę su išraiška “Mažojo laivo ilgis ne daugiau negu 24 m”.



44 pav. Formalizuotos veiklos taisyklės langas.

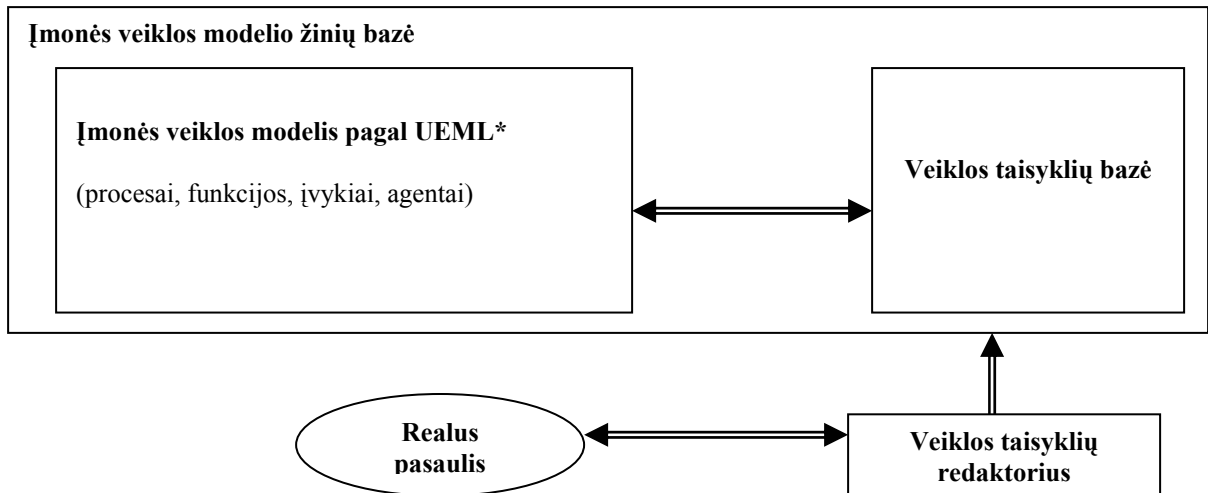
### 3.6.7 Veiklos taisyklių redaktoriaus diegimo priemonių planas.

Prieš pradėdant darbą su šia sistema, ją reikia įdiegti darbu paruoštame kompiuteryje.

- ❖ Pirmiausia reikia suinstaliuoti MySQL paketą. Jis suinstaliuojamas paleidus *setup.exe* failą iš katalogo “mysql”.
- ❖ Po to suinstaliuojame programą Mascon, kuri bus naudojama duomenų bazės peržiūrai ir jos redagavimui. Joje reikia susikurti kol kas tuščią duomenų bazę vardu “DB007” ir ją sugeneruoti pasirinkus failą “Setup” iš katalogo “install”. Jame reikia įrašyti duomenų bazės pavadinimą ir pasirinkti mygtuką “Tikrinti”, o gavus pranešimą “OK”, paspausti mygtuką “Install database”. Jei viskas pavyko, bus pateikiamas pranešimas “Duomenų bazė perkrauta”.
- ❖ Po to suinstaliuojame Java Web Start paketą, pasirinkdami “j2re-1\_4\_2\_02-windows-i586-p.exe” failą.
- ❖ Programa paleidžiama pasirinkus vykdomąjį failą “jcharmW.exe”.

### 3.6.8 Formalizuotas tiriama metodo, algoritmų ar jų elementų, atspindinčių darbo idėją bei naujumą, aprašas.

Šio projekto metu bus sukurtas veiklos taisyklių redaktorius, iš kurio veiklos taisyklės bus perduodamos į veiklos taisyklių bazę. 45 paveiksle pateiktas projekto realizacijos modelis.



45 pav. Projekto realizacijos modelis

Kad surenkamos veiklos taisyklės būtų konkrečios ir tokiame formate, kuriame būtų lengva jas aprašyti ir formalizuoti, saugomos taisyklės turi būti [14]:

- ❖ deklaratyvios;
- ❖ tikslios – veiklos taisyklės privalo būti vienprasmėms interpretuojamos. Jei taisyklė bus daugiapasmė, bus reikalinga ją performuluoti.
- ❖ atominės – veiklos taisyklė reiškia vieną išbaigtą mintį. Ji turi būti nedaloma; jei bandoma ją išskaidyti, galima prarasti svarbios informacijos.
- ❖ nuoseklios – veiklos taisyklės negali prieštarauti viena kitai.
- ❖ neperteklinės – rinkinyje veiklos taisyklės neturi kartotis.
- ❖ orientuotos į veiklą – veiklos taisyklės aprašomos veiklos žmonėms suprantamais terminais.
- ❖ jos gali būti naudojamos kitose taisyklėse – veiklos taisyklės gali būti konstruojamos viena iš kitos.

Veiklos taisyklės surenkamos ir saugomos redaktoriuje šablonų pagalba. Pildant veiklos taisykles ir kuriant naujus veiklos taisyklių šablonus, bus tikrinami šablonų struktūrų medžiai. Jie sudaryti remiantis pasaulio mokslininkų pateiktais siūlymais.

**Šablonų klasifikavimas [8].**

Taip kaip yra VT kategorijos, taip yra ir gausybė VT šablonų. Pradinis šablonų rinkinys atitinka VT kategorijas. Žemiau pateikiama po vieną ar daugiau šablonų, skirtą **Terminų (Term)**, **Faktų (Fact)**, **Skaičiavimo (Computation)**, **Apribojimų (Constraints)**, **Išvesties (Inferred Knowledge)** ir **Veiksmo įgalinimo (Action enabler)** veiklos taisyklėms. Tačiau yra galimybė šablonus sudaryti ir iš vieno ar daugiau paprastų šablonų.

Kai veiklos taisyklių šablonai yra užbaigti ir patikrinti, tuomet gali būti išskirti keletas šablonų kategorijų kaip meta duomenys duomenų modeliui. Daugelyje CASE įrankių yra įdiegta keletas šablonų tipų, bet mažai tēra tokių, kurie galėtų valdyti juos visus.

### **Terminai naudojami VT šablonuose**

Kai sudarinėjame šabloną, labai svarbu yra aiškiai apibrėžti terminus, kurie sudaro jį. Čia pateiktuose šablonuose yra naudojami tokie terminai:

- Terminas (term) – bet koks daiktavardis ar daiktavardžių frazė, kurioje yra Taisyklė;
- Operatorius (operator) – jungtis (connector) ar jungties frazė, kuri susieja du terminus. Jungtis gali būti EQ, NE, GT, GE, LT, LE, Is, Has, Has At Least n, Has At Most n, Has Exactly ir t.t.
- Termino reikšmė (Term value) – Termino konkreti reikšmė (Term=Supplier; Term Value=DAMA);
- Termino reikšmių rinkinys (Term value Set) – reikšmių rinkinys termino, kuris yra išskirtinis ir gali būti išsamus;
- Sąlygos terminas (Conditonal Term) – bendri terminai, kurie yra kaip parametras skirtingose taisyklėse;
- Išvestis (Inferred Knowledge) – žinios, kurios gaunamos įvertinus vieną ar kelias taisyklių frazes.
- Veiksmas (Action) – veiksmas, kuris gaunamas įvertinus vieną ar kelias taisyklių frazes.

Šie pagrindiniai šablonų tipai yra apibūdinti ir iliustruoti:

1. Terminų VT šablonas (The Term Business Rule Template);
2. Faktų VT šablonas (The Fact Business Rule Template);
3. Skaičiavimo VT šablonas (The Computation Business Rule Template);
4. Apribojimų VT šablonas (The Constrain Business Rule Template);
5. Išvadų VT šablonas (The Inferred Knowledge Business Rule Template);
6. Veiksmo įgalinimo VT šablonas (The Action Enabler Business Rule Template);

### **Terminų veiklos taisyklės**

Terminų VT yra viena iš daugelio pagrindinių biznio analizės ir/arba modeliavimo uždavinių kotsravimo bloką. Iki kol biznio terminai turi tikslius ir priimtus apibrėžimus, šių terminų naudojimas kitose VT gali įnešti nemažai rizikos.

Terminų VT yra surenkamos kaip esybių ir atributų apibrėžimai tradicinėje duomenų bazėje.

### Termino šablonas

<Terminas> yra <tekstinė išraiška>



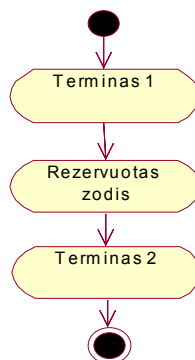
46 pav. Termino šablonas.

### Faktų veiklos taisyklės

Faktai yra teiginiai, sujungiantys terminus į prasmingas, tiesiogiai su bizniu susijusias grupes. Paprastai faktas apima tik du terminus ir abu iš jų turi būti apibrėžti. Faktai apibūdina ryšius tarp dviejų lentelių arba tarp lentelės ir atributo. Faktų VT yra užfiksuojamos ryšiuose, supertipo/ subtipo struktūrose ir atributuose tradicinėje duomenų bazėje.

### Fakto šablonas

<Terminas1> gali būti<veiksmazodis> <Terminas2>



47 pav. Faktų šablono struktūra.

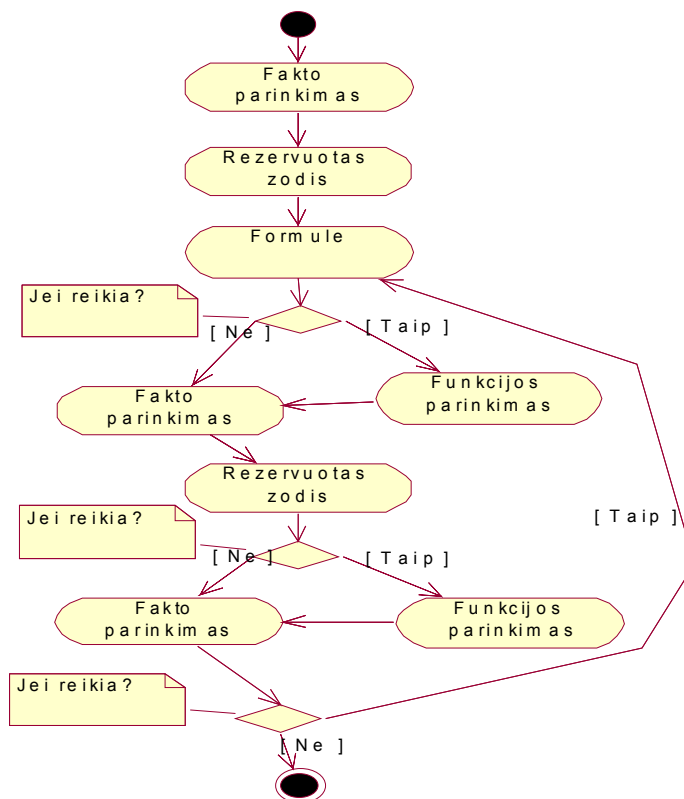
## Skaičiavimo veiklos taisyklės

Skaičiavimas yra veiklos taisyklė, kuri duoda algoritmą tam, kad suteiktų reikšmę naujam terminui. Algoritmams gali būti - suma, atimtis, sandauga, dalyba, kiekis, maksimumo ir minimumo paieška, vidurkio ar santykio skaičiavimas. Terminai naudojami formulėse taip pat turi būti apibrėžti.

Skaičiavimo terminai dažnai išskiriami kaip išvestiniai (Derived) atributai tradicinėje duomenų bazėje; formulė yra dalis aprašymo. Gerai sukurtose CASE priemonėse, jie taip pat gali būti surenkami trigeryje.

## Skaičiavimo šablonas

<Faktas> apskaičiuojamas <formulė>



48 pav. Skaičiavimo veiklos taisyklių šablono struktūra.

## Apribojimų veiklos taisyklės

Apribojimai yra teiginiai, kurie išreiškia aplinkybes (įvertina teigiamai), kurios privalo būti įvykdytos arba ne. Jie apibrėžia tarpusavio ryšių kardinalumą, priklausomybes tarp atributo reikšmių arba tarp atributo reikšmių ir konstantų, o taip pat nustato atributų reikšmes.

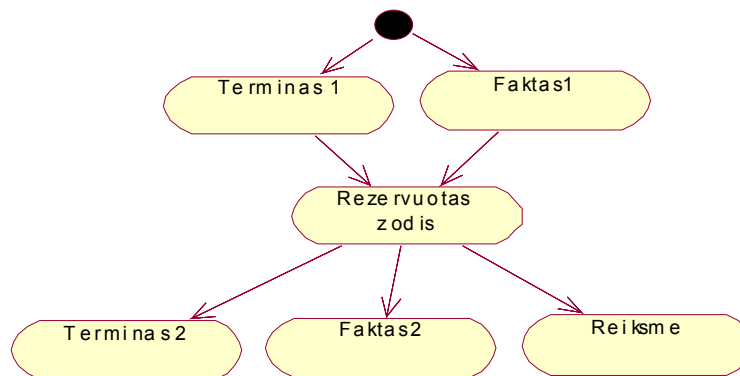
Apribojimų VT yra dokumentuojamos meta duomenyse. Ryšių kardinalumo apribojimai paprastai yra pateikiami kaip ryšių objektų savybės CASE priemonėse. Palyginimai yra dokumentuojami kaip ir Apribojimai.

### Apribojimų šablonai

<Faktas 2>privalo turėti <mažiausiai, daugiausiai, tiksliai n iš> <Fakto 1>

<Faktas1> privalo <palyginti> {<Faktas2>, Termino reikšmė, konstanta}

<Faktas1> turi būti sąrašė <sąrašas>



49 pav. Apribojimų veiklos taisyklių šablono struktūra.

### Išvesties Veiklos Taisyklės

Išvesties šablonai yra sąlygų rinkinys, kuris, jei tenkinamas, sukuria naują terminą. Prieš tai visi Išvesties VT terminai naudojami taisyklės frazėse turi būti apibrėžti.

Daugelyje CASE priemonių aprašyti sąlygines taisykles yra sudėtinga. Tam gali būti panaudojami trigerio objektai.

### Išvesties VT šablonai

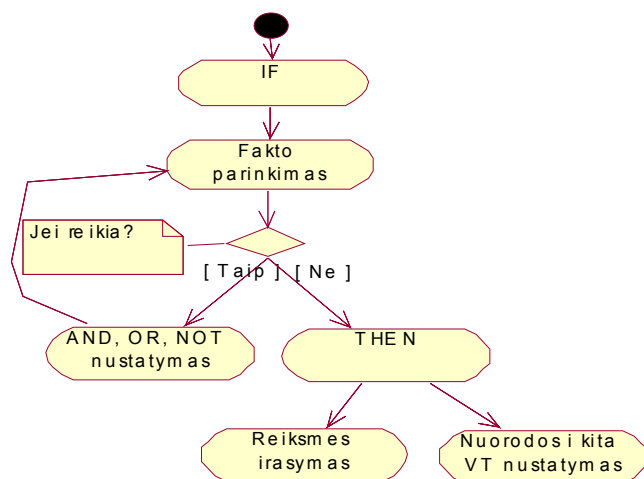
*IF <faktas>*

*[AND < faktas>*

*AND < faktas>...]*

*THEN <kita VT arba reikšmės įrašymas >*





50 pav. Išvesties veiklos taisyklių šablono struktūra.

### Veiksmo įgalinimo taisyklių šablonai

Veiksmo įgalinimo taisyklių šablonai yra sudaryti tokiu būdu, kaip ir Išvesties šablonai, bet šių šablonų rezultatas yra veiksmas arba pasiūlymas atlikti standartinę procedūrą. Tiems kas analizuoja VT, įvykiai ir sprendimai dažnai yra puikus būdas VT nagrinėjimui. Daugeliu atvejų veiksmas, kaip veiksmo įgalinimo taisyklės rezultatas, yra įvykis, nukreipiantis į kitą veiklos dalį.

Veiksmo įgalinimo VT yra aprašomos panašiai kaip ir Išvesties VT.

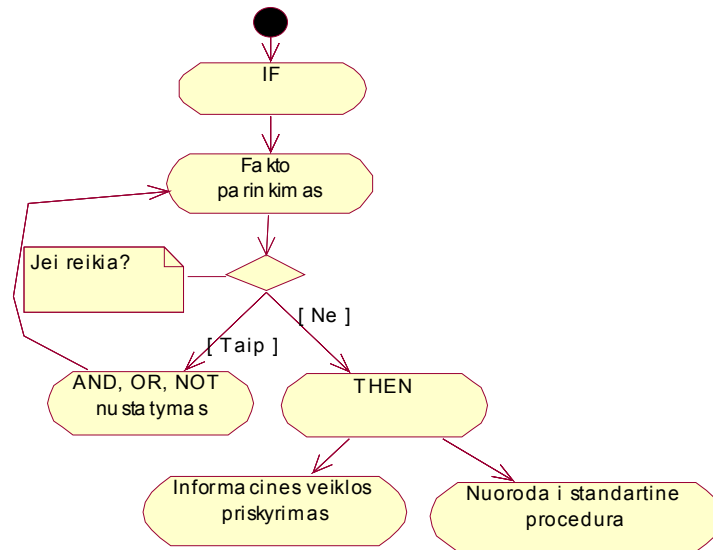
### Veiksmo įgalinimo VT šablonai

**IF <faktas, taisyklės frazė>**

**[AND < faktas, taisyklės frazė >**

**AND < faktas, taisyklės frazė >...]**

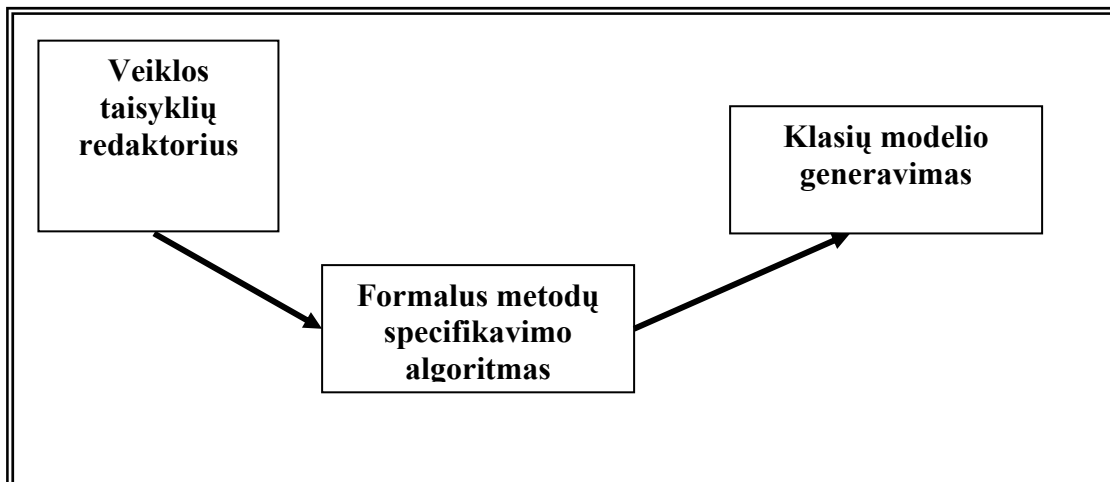
**THEN <informacinės veiklos vykdymas arba standartinė procedūra>**



51 pav. Veiksmo įgalinimo veiklos taisyklių šablono struktūra.

Šios programos naujovės ir privalumai:

- ❖ Šiame projekte sukurtas veiklos taisyklių redaktorius, kuris naudojamas veiklos taisyklių saugojimui ir redagavimui. Jos yra atskirtos nuo programos kodo, todėl paieškos procesas labai supaprastėja. Taip pat šis redaktorius yra naudojamas jo paties tobulinimo procese, t.y. yra naudojamas veiklos modelio klasių generavime.
- ❖ Šios sistemos naujumas tame, kad veiklos taisyklių formalizavimui naudojami šablonai. Jų naudojimas supaprastina veiklos taisyklių formalizavimą ir tolimesnį jų panaudojimą organizacijos informacinėje sistemoje.
- ❖ Inžinerinė šio magistrinio darbo nauda yra ta, kad veiklos taisyklių redaktorius bus toliau naudojamas kaip pagalbinis komponentas metodų specifikavimui ir generavimui (52 pav.).



### 3.7 Projekto išvados.

Šio magistrinio darbo projekto dalyje:

- ❖ Suformuluotas projekto tikslas ir jis pagrįstas;
- ❖ Reikalavimų modelyje nubraižytos diagramos pagal UML notaciją:
  - vartotojo panaudojimo atvejų diagramos, kontekstinė diagrama;
  - kiekvieno panaudojimo atvejo specifikacijos;
  - veiklos diagramos;
  - dalykinės srities klasių diagrama;
  - vartotojų interfeiso modelis.
- ❖ Parengta projekto techninė užduotis, pagal kurią bus kuriama sistema.
- ❖ Kuriama sistema detaliai išnagrinėta sistemos projekte. Pateikti visi reikalavimai sistemos funkcionalumui palaikyti, apribojimai. Taip pat pateikti reikalavimai techninei bei programinei įrangai.
- ❖ Magistrinio darbo aprašo prieduose pateiktas sistemos vartotojų vadovas ir sistemos įdiegimo dokumentas, kuriuose detaliai aprašyta kaip dirbti su sistema, kaip ją įdiegti savuose kompiuteriuose.
- ❖ Sistema sukurta naudojant JAVA programavimo kalbą ir MySQL duomenų bazių sistemą.

## 4. Eksperimentinis tyrimas.

### 4.1 Sukurtos sistemos kokybės tyrimas.

Visada sukūrę naują sistemą, pirmiausia ją turime detaliai ištestuoti, aprašyti jos savybes:

- ❖ tikslumas – rezultatas pateikiamas pakankamai tiksliai, priklauso nuo vartotojo (IS projektuotojo) išsilavinimo ir sugebėjimo analizuoti;
- ❖ patikimumas – patikimumas šioje programoje nėra didelis, nes kol kas nėra realizuota duomenų kontrolė;
- ❖ sąveika su kitomis sistemomis – ši sistema sąveikaus su organizacijos veiklos modelio pagrindu sukurta informacijos sistema, todėl yra numatomas apjungimas per OVM duomenų bazės lentelę „Veiklos taisyklė“;
- ❖ užbaigtumas – sistema dar visiškai neužbaigta. Ją galima dar labai tobulinti ir gerinti.

- ❖ tolerancija klaidoms – kol kas šioje sistemoje nėra pilnai realizuota duomenų kontrolė, todėl joje galima įvesti bet kokią informaciją, nes netikrinamas jos teisingumas.
- ❖ atstatomumas – ši savybė būdinga sukurtajai sistemai. Ją galima atstatyti iš po nenumatytų nutikimų.
- ❖ patogumas – sistema patogi tuo, kad jos valdymas pateiktas išsiskleidžiančiais meniu punktais, todėl nereikia vaikščioti po sistemą ir ieškoti reikalingo meniu punkto, kad atlikti norimą veiksmą.
- ❖ suprantamumas – sistema yra pakankamai sudėtinga paprastam vartotojui, bet jei su ja dirba informacinių sistemų specialistas, ji gana lengvai perprantama ir suvokiama joje įvedinėjama informacija;
- ❖ išmokstamumas – jei vartotojas informacijos sistemų specialistas, tai jam nebus sudėtinga perprasti šios sistemos veikimą ir darbą su ja.
- ❖ efektyvumas – ši sistema rezultatai turėtų pateikti gana efektyviai, nes duomenų bazė nėra ypatingai didelė:
  - laiko parametrai - sistema veikia pakankamai greitai, nes duomenų bazėje yra ne daug testavimo duomenų, tačiau ir su nemažu kiekiu duomenų sistema veiks gana greitai.
  - išteklių naudojimas – veiklos taisyklių redaktorius nenaudoja daug kompiuterio resursų. Pakanka kad kompiuteryje veiktų MySQL duomenų bazė ir Java Web Start vykdomasis failas.
- ❖ apsimokamumas – ši programa finansiniu požiūriu nėra brangi ir reikalaujanti daug resursų, nes jos kūrimui naudotos arba visiškai nemokamos, arba nebrangios programos;
- ❖ pakeičiamumas – šią sistemą galima lengvai keisti mokant Java programavimo kalbą;
- ❖ priežiūros savybės
  - analizės savybės – analizavimui suteiktos puikios galimybės, nes visose formose lentelės forma pateikiami egzistuojantys įrašai;
  - pakeitimų galimybės – pakeitimus visiškai nesudėtingai galima atlikti, turint JBuilder 9 paketą ir mokant programuoti Java programavimo kalbą;
  - testavimo savybės – testavimui galima įvedinėti įvairius duomenis, kuriuos vėliau galima naikinti.

## **4.2 Tolimesnio sistemos tobulinimo, plėtojimo galimybės.**

Šiame magistriniame darbe sukurta programa nėra iki galo išbaigta. Ją bus galima dar labai plačiai tobulinti. Ji neišbaigta todėl, kad jau kūrimo metu šios programos pagrindas buvo organizacijos veiklos modelis, kuris taip pat lygiagrečiai baigiama kurti. Pagal organizacijos veiklos modelį sukurtos informacijos sistemos dalis ir bus veiklos taisyklių redaktorius, kuriame atskirai nuo programos kodo bus saugomos visos konkrečią organizacijos veiklą įtakojančios ar apibrėžiančios veiklos taisyklės.

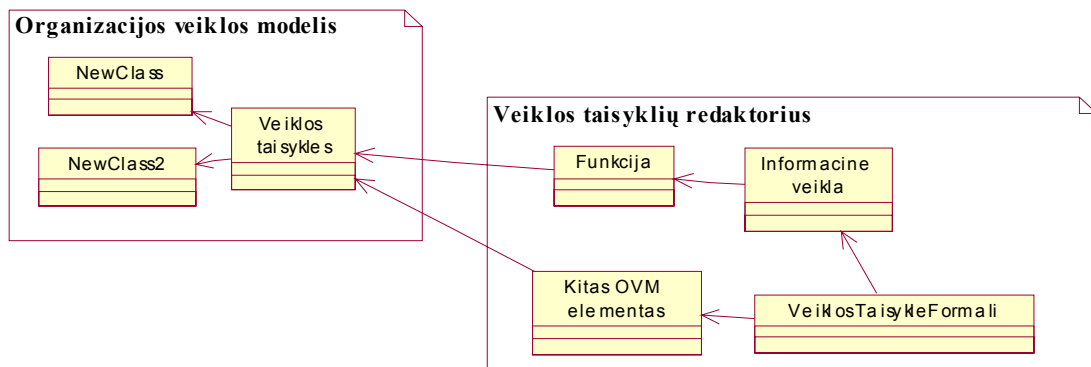
Kita sukurtos sistemos tobulinimo kryptis yra jos panaudojimas klasių arba metodų generavimui. Ji bus kaip tarpinis elementas tarp organizacijos informacijos sistemos ir klasių modelio. Metodų generavimui be veiklos taisyklių redaktoriaus dar bus reikalingas ir matematiškai aprašytas metodų specifikuojantis algoritmas.

Dar viena iš sistemos tobulinimo galimybių yra suprogramuoti įvedamų ir saugomų duomenų kontrolę. Tai paprastai visose programose būna nemaža ir nelengva dalis viso darbo. Tačiau be duomenų kontrolės sistema niekada neveiks visiškai patikimai. Duomenų ir informacijos kontrolė būtina.

## 5. Išvados.

Magistrinis darbas buvo atliekamas organizacijos veiklos modelio pagrindu. Šios programos naujovės ir privalumai:

- ❖ Veiklos taisyklių redaktorius yra integruojanti priemonė, kurios pagalba veiklos taisyklės iš redaktoriaus integruojamos į organizacijos veiklos modelį (53 pav.) per pagalbines lenteles.



53 pav. Veiklos taisyklių integravimas į organizacijos veiklos modelį.

- ❖ Šiame projekte sukurtas veiklos taisyklių redaktorius, kuris naudojamas veiklos taisyklių saugojimui ir redagavimui. Jos yra atskirtos nuo programos kodo, todėl paieškos procesas labai supaprastėja. Taip pat šis redaktorius yra naudojamas jo paties tobulinimo procese, t.y. yra naudojamas veiklos modelio klasių generavime.
- ❖ Šios sistemos naujumas tame, kad veiklos taisyklių formalizavimui naudojami šablonai. Jų naudojimas supaprastina veiklos taisyklių formalizavimą ir tolimesnį jų panaudojimą organizacijos informacinėje sistemoje.
- ❖ Inžinerinė šio magistrinio darbo nauda yra ta, kad veiklos taisyklių redaktorius bus toliau naudojamas kaip pagalbiniis komponentas metodų specifikavimui ir generavimui.

### Analizės etapo išvados:

- ❖ Išanalizuota daugelis siūlytų veiklos taisyklių klasifikacijų ir apibendrinus sukurta šiame darbe naudojama klasifikacija.
- ❖ Analizuojant probleminę sritį buvo naudojami tokie modeliai:

*Veiklos sąveikų modelis;*

*Veiklos tikslų modelis;*

*Veiklos procesų modelis;*

*Veiklos objektų modelis;*

*Veiklos panaudojimo atvejų modelis.*

- ❖ Naudojami modeliai padėjo įvertinti vartotojų atliekamas funkcijas, jų poreikius bei atsakomybes.
- ❖ Aprašyti šablonai, kurie bus naudojami veiklos taisyklių formalizavimui.
- ❖ Parinkta programinė įranga JBuilder 9 ir MySQL , kurios pagalba kuriamas veiklos taisyklių redaktorius.

### **Projekto modelio išvados:**

- ❖ Naudojantis UML notacija, projektavimo etape nubraižytos diagramos aprašančios:
  - kuriamoje sistemoje egzistuojančius panaudojimo atvejus ir jų specifikacijas;
  - sistemos elgseną;
  - dalykinės srities klasių diagramą;
  - realizacijos modelį;
  - testavimo modelį.
- ❖ Suformuoti veiklos taisyklių šablonai pagal jų tipą:
  - terminų;
  - faktų;
  - apribojimų VT;
  - skaičiavimo VT;
  - veiksmo įgalinimo VT;
  - išvesties VT.

Šiame magistriniame darbe atlikta išsami veiklos taisyklių analizė pagal skirtingus kriterijus. Suformuota nauja jų klasifikacija, parinkti atitinkami veiklos taisyklių pildymo šablonai. Šios sistemos naujovės:

- ❖ sistemoje naudojama tik lietuvių kalba;
- ❖ veiklos taisyklės pildomos naudojantis šablonais;
- ❖ aiškus perėjimas nuo natūralia kalba įvestų veiklos taisyklių prie formalizuotų veiklos taisyklių;
- ❖ veiklos taisyklė tiksliai priskiriama konkrečiam elementui iš organizacijos veiklos modelio.

Sukurtos sistemos privalumai:

- ❖ nesudėtinga, aiški ir suprantama vartotojų sąsaja;
- ❖ naudojama lietuvių kalba;
- ❖ galima naudoti bet kurioje operacinėje aplinkoje.

Sukurtos sistemos trūkumai:

- ❖ su sistema gali dirbti tik informacinių sistemų specialistai;
- ❖ nėra įgyvendinta duomenų kontrolė.

## 6. LITERATŪRA.

- [1] – S.Gudas, A.Lopata, T.Skersys Domain knowledge integration for information systems engineering. - Konferencijos pranešimų medžiaga “Informacinės technologijos verslui – 2002”, Kaunas, Technologija, 2002, p. 56-59.
- [2] - Kapočius K.,Butleris R. „Reikalavimų kuriamai IS nustatymo ir analizės ypatumai veiklos taisyklių metodologijoje“// Informacinės technologijos ‘2003. Konferencijos pranešimų medžiaga. Kaunas. Technologija, 2003, psl. XIV-74 – XVI-81.
- [3] – Leo Hsu, Regina Obe Business Rules and Object Orientation, 1996, [žiūrėta 2003-02-12] Prieiga per internetą: <http://www.businessrulesmod.com/>
- [4] - EBGconsulting company, Business Rules, 2002. [žiūrėta 2003-04-02] Prieiga per internetą: [http://www.ebgconsulting.com/services\\_consulting\\_businessrules.html](http://www.ebgconsulting.com/services_consulting_businessrules.html).
- [5] - Richard J. Mayer, Michael K. Painter, Madhavi Lingineni „Information integration for concurrent engineering (IICE) toward a method for business constrains discovery“, Knowledge Based Systems, Inc., 1995, [žiūrėta 2003-04-05] Prieiga per internetą: <http://www.idef.com/Downloads/pdf/Idef9.pdf>
- [6] - Object Constraint Language Specification, Rational Software Corporation, 1997.
- [7] - Idea Methodology , [žiūrėta 2003-04-10] Prieiga per internetą: [http://www.cc.gatech.edu/classes/AY2002/cs6421\\_spring/presentations/IDEA\\_Methodology.pdf](http://www.cc.gatech.edu/classes/AY2002/cs6421_spring/presentations/IDEA_Methodology.pdf)
- [8] - Templates for Capturing Business Rules, [žiūrėta 2003-10-03] Prieiga per internetą: <http://www.BRCommunity.com>.
- [9] - "Defining Business Rules ~ What Are They Really?" GUIDE Business Rules Project, Final Report. July, 2000. [žiūrėta 2002-10-25]. Prieiga per internetą: <http://www.businessrulesgroup.org/>.
- [10] – Dr. Silvie Spreeuwenberg, “Using Verification and Validation Techniques for High-quality Business Rules”, [žiūrėta 2003-11-15] Prieiga per internetą: <http://www.brcommunity.com/cgi-bin/x.pl/print/p-b132.html>.
- [11] – H. Herbst, G. Knolmayer, T. Myrach and M. Schlesinger, “The specification of business rules: a comparison of selected methodologies”; IFIP Working Group 8.1 Conference CRIS 94, University of Limburg, Maastricht. Published in: A.A. Verijn-Stuart, T.-W. Olle (Eds.), Methods and Associated Tools for the Information System Life Cycle, Amsterdam et al.: Elsevier 1994, pp. 29-46.
- [12] – Ellen Gottesdiener „Business Rules”, Straipsnis iš žurnalo tomas 4., numeris. 3; [žiūrėta 2003-03-12]; Prieiga per internetą: [http://www.ebgconsulting.com/powerpromise\\_article.htm](http://www.ebgconsulting.com/powerpromise_article.htm)
- [13] - Wan M. N. Wan Kadir, Pericles Loucopoulos: Relating Evolving Business Rules to Software Design. Software Engineering Research and Practice 2003: pp.129-134.
- [14] – “What makes a business rules good?”; Intelligent ENTERPRISE. March 9, 1999.[žiūrėta 2003-05-12].
- [15] – Kapočius K., Butleris R. „The Business Rules Repository for Information Systems Design // The 6th East-European Conference ADBIS’2002“. Konferencijos pranešimų medžiaga. ISBN: 80-227-1744-4. Bratislava, Slovakia. Vydavatel’stvo STU, 2002, Vol.2, p.64-77.
- [16] – Andrew N.Edmonds, „Why metarule?“ [žiūrėta 2003-04-17]; Prieiga per internetą: [www.scientio.com](http://www.scientio.com).
- [17] - Ronald G. Ross, Gladys S. W. Lam “RuleSpeak™ Sentence Templates - Developing Rule Statements Using Sentence Patterns”; [žiūrėta 2003-04-17]; Prieiga per internetą: [www.BRSolutions.com](http://www.BRSolutions.com).
- [18] - H. Herbst, “A Meta-Model for Business Rules in Systems Analysis”// J. Iivari, K. Lyytinen, M. Rossi (Eds.), Proceedings of the Seventh Conference on Advanced Information Systems Engineering (CAiSE '95), Berlin et al.: Springer 1995, pp. 186 – 199.



## 7. TERMINŲ IR SANTRAUKŲ ŽODYNAS.

**VT** – veiklos taisyklė;

**OVM** – organizacijos veiklos modelis;

**Probleminės srities analitikas** – asmuo, analizuojantis informacijos sistemos probleminę sritį ir suvedantis veiklos taisykles natūralia kalba;

**IS projektuotojas** – asmuo, analizuojantis suvestas veiklos taisykles natūralia kalba ir formalizuojantis jas. Jis atlieka visas pagrindines šios sistemos funkcijas.

**Terminas** – tai informacijos sistemos sąvoka, aprašanti kiekvieną joje egzistuojantį objektą.

**Faktas** – tai struktūra, susidedanti iš dviejų terminų ir rezervuoto žodžio tarp jų. Ji dažniausia charakterizuoja objektų savybes;

**Skaičiavimo veiklos taisyklė** – tai veiklos taisyklė, kurioje atliekami skaičiavimai;

**Veiksmo įgalinimo veiklos taisyklė** – tai veiklos taisyklė, kurios rezultatas yra arba veiksmas, arba standartinė procedūra.

**Išvesties veiklos taisyklė** – tai veiklos taisyklė, kurios rezultatas yra – arba informacinė veikla, arba išvedama nauja reikšmė, arba nuoroda į kitą veiklos taisyklę;

**Apribojimų veiklos taisyklė** – tai veiklos taisyklė, kuri nusako kokį tai apribojimą informacijos sistemos objektams.

## 8. SUMMARY.

Nowadays, nearly all of the commercial and government organizations are highly dependent on software systems. Due to the inherent dynamic nature of their business environment, software evolution is inevitable. The changes generated by business policies and operations are propagated onto software system. A large portion of total lifecycle cost is devoted to introduce new requirements, and remove or change existing requirements. However, the evolution of a successful software system is inevitable for the software to remain useful in its environment.

Many research projects attempt to find a more applicable way for building a software system that is adaptive to changes. They strive to propose a software model, or architecture, that has the ability to minimize the effect of changes as well as providing requirements traceability in their model. Most of them utilize the benefit of object-oriented, distributed system, software architecture and component-based technologies. Our research attempts to address these issues in a slightly different way namely by considering business rules as volatile part of a software system. In this position paper, we introduce a software architecture which includes a link model that aims to reduce the effects of software changes by improving requirements traceability.

When we have analyse this problem, we decide to make the program, which will have ability to resolve business rules isolation from all program code problem. The main purpose in this project is – to create a *business rules editor*, in which we will have ability to formalize business rules using templates. We can display this stages of the project:

- ❖ design the business rules database;
- ❖ creating the business rules templates;
- ❖ creating the business rules editor.

## **9. PRIEDAI.**

9.1 PRIEDAS 1. Straipsnio “Veiklos taisyklių taikymas informacijos sistemų inžinerijoje” medžiaga. Staripnis perskaitytas konferencijoje “Informacinės technologijos – 2003”, vykusiame 2003 metų gegužės 23 dieną Kaune Vilniaus Universiteto Kauno humanitariniame fakultete.

9.2 PRIEDAS 2. SmartCost ir RuleTrack programų veiklos taisyklių kūrimo langai.

9.3 PRIEDAS 3. Sistemos vartotojo vadovas.

9.4 Darbo pristatymas MS Power Point formatu.

9.5 Sukurtos programos išeities kodas pateikiamas CD diske.

## 9.1 PRIEDAS 1.

### VEIKLOS TAISYKLIŲ TAIKYMAS INFORMACIJOS SISTEMŲ INŽINERIJOJE

Rasa Mikloviėnė

Magistrantė, Informacijos sistemų katedra, Kauno technologijos universitetas

*Straipnyje aptariama veiklos taisyklių koncepcija bei pagrindiniai žinomi modeliai. Analizuojama veiklos taisyklių vieta informacijos sistemų inžinerijoje. Pateiktas veiklos taisyklių saugyklos modelis. Pasiūlytas planuojamas realizacijos modelis.*

Vadovas Saulius Gudas

#### 1. Įvadas.

Per pastaruosius trisdešimt metų buvo ypatingai įdomu stebėti, kaip keičiasi ir tobulėja informacijos sistemų kūrimo metodika. Su vos tik pasirodžiusiu programavimu atrodė buvo sukuriama viskas. Augant vartotojų poreikiams pasirodė, kad programuojant galima buvo sukurti tik dalį to, ko reikėjo vartotojui. Vėliau atsirado informacijos inžinerija ir objektiškai orientuotas programavimas. Bet ir to buvo per maža, kad patenkinti vartotojų poreikius. Tam, kad visiškai patenkinti vartotojų reikalavimus buvo atrasta tokia žinių dalis, kuri yra gyvybiškai svarbi organizacijos egzistavimui – veiklos taisyklės.

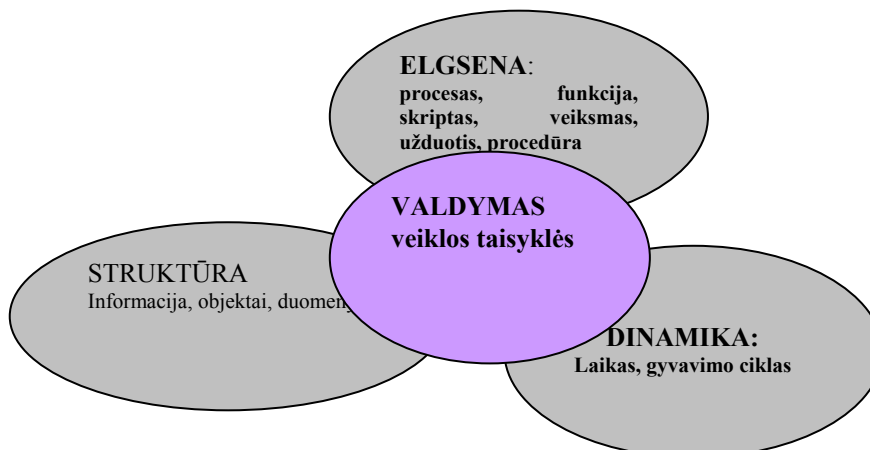
Kuriant informacijos sistemas (IS), viena iš problemų yra veiklos taisyklių struktūrizavimas. Analizuojant probleminę sritį ir vartotojo reikalavimus kuriamai IS, tenka struktūrizuoti informaciją apie veiklos procesus bei jiems keliamus apribojimus.

#### 2. Veiklos taisyklių modeliavimo apžvalga.

Įvairių procesų vyksmą reglamentuoja veiklos taisyklės, kurių identifikavimas, struktūrizavimas bei specifikavimas yra pagrindiniai probleminės srities analizės ir reikalavimų kuriamai IS specifikacijos sudarymo uždaviniai.

Į veiklos taisyklės žiūrint iš IS inžinerijos pozicijų, tai jos yra pačiame struktūrinių, elgsenos ir dinaminių modelių centre. Visus šiuos modelius projektuotojas kuria analizės ir projektavimo etapuose, nepaisydamas to, ar jis naudoja į duomenis, procesus, objektus ar komponentus orientuotus metodus.

Kompanija EBGconsulting, užsiimanti veiklos bei veiklos taisyklių modeliavimo konsultacijomis, pateikė 1 paveikslėlyje parodytą schemą, kurioje matyti, kad veiklos taisyklės atlieka kontrolės vaidmenį ir apjungia visus – elgsenos, struktūros ir dinamikos – sluoksnius. [2]



1 pav. Veiklos taisyklių padėtis kitų modelių atžvilgiu. [2]

Kokią naudą duoda veiklos taisyklių modeliavimas? [4]

- **Veiklos taisyklių modeliavimas supaprastina pačios veiklos suvokimą.** Veiklos taisyklių modeliavimas pateikia visas taisykles, apribojimus, veiklos sritis, nurodymus ir kitokias sąlygas egzistuojančias konkrečioje veikloje. Tai taip pat apima ir struktūrinius apribojimus, ir procesų/ sekų apribojimus ir pateikia juos projektuotojui darbų sekų diagramose.
- **Leidžia našiau, efektyviau projektuoti.** Jei sistema projektuojama remiantis veiklos taisyklių architektūra, tuomet galima tikėtis tokios naudos:
  - pritaikomumo – nesudėtinga keisti veiklos taisykles;
  - aiškumo, suprantamumo – lengviau suvokiama, kam reikalingas konkrečioje vietoje tam tikras elementas ar sistemos dalis;

- **Sistema tampa suprantamesnė.** Suformuoja pagrįstus žiniomis sprendimus apie sistemos struktūrą ir veikimo principą. Tokie sprendimai įtakoja suvokimą ir patikimumą tuo atveju, kai pasikeičia veiklos taisyklių rinkinys.
- Veiklos taisyklių modeliavimas gali būti realizuotas “juodojoje dėžėje” – paprasčiausiai stebdami (išorinę) sistemos ar organizacijos elgseną mes galime formuluoti jų valdomas taisykles. Tai reiškia, kad tai yra aiškus būdas paaiškinti, kaip klientai ir konkurentai elgiasi, egzistuojant įvairioms sąlygoms. Tai taip pat reiškia, kad mes galime mąstyti apie vis atsirandančias naujas sudėtingos sistemos savybes, nežinodami apie detalius komponentų ir paslaugų, iš kurių susideda sistema, struktūras ir mechanizmus.

### 3. Veiklos taisyklių modelių savybių palyginimas.

Veiklos taisyklė yra teiginys, kuris apibrėžia ar apriboja tam tikrus veiklos aspektus. Veiklos taisykle nusakoma tam tikros veiklos struktūra, kontroliuojama arba įtakojama tam tikra veikla. [6]

Kai veiklos taisyklė nusakoma veiklos žmogaus, tai ji dažniausiai įvardijama dviprasmiškai ir negiežtai. Dažniausiai veiklos taisykles galima išskaidyti iki elementarių taisyklių, iki kol atsiribojama viena išbaigta mintimi. Elementari veiklos taisyklė, parašyta deklaratyviu būdu, naudojant standartinę natūralios kalbos gramatiką, kurią veiklos žmonės lengvai supranta, nėra dviprasmiška. Taigi, veiklos taisyklės yra elementarios, deklaratyvios, nepriklausančios nuo loginės struktūros, orientuotos į veiklą ir priklausančios veiklai.

Kad veiklos taisykles būtų lengva keisti, reikia, kad jos būtų logiškai atskirtos nuo duomenų bei programos struktūros. Dažniausiai jos išimamos taisyklių interpretatoriuje, kuris yra priemonė, leidžianti peržiūrėti, pakeisti ar tvarkyti tas taisykles.

Veiklos taisyklės struktūrizuojamos skirtingai. Įvairūs autoriai pateikia skirtingus metodus, kuriais naudojantis galima suskirstyti veiklos taisykles į kategorijas ir rūšis. Labiausiai išplėtoti ir pripažinti yra GUIDE projekto ir Ronaldo Roso modeliai.

GUIDE projekte išskiriami trys pagrindiniai veiklos taisyklių tipai. Kiekviena taisyklė turi priklausyti kuriam nors iš šių tipų:

- *Struktūrinis sakiny* – apibrėžta mintis, išreiškianti tam tikrą organizacijos sandaros aspektą.
- *Veiksmo teiginys* – sakiny, reiškiantis apribojimą ar sąlygą ir nustatantis arba valdantis organizacijos veiksmus.
- *Išvestis* – sakiny, išreiškiantis žinias, kurios išvedamos iš kitų žinių apie veiklą.

Labai detalų veiklos taisyklių klasifikavimo metodą sukūrė Ronaldas Rosas. Pagrindinė šio metodo idėja – suskirstyti taisykles į standartinius taisyklių tipus, kurie nurodo, kokį patikrinimą turi atlikti taisyklės. Taisyklės gali būti nedalomos ir išvestinės. Nedalomos taisyklės suskirstytos į 32 tipus, kurie papildomai grupuojami į 7 tipų šeimas. Išvestinės taisyklės yra sudaromos iš kitų taisyklių. Išskiriami 58 išvestinių taisyklių tipai, kurie pagal panaudojimo sritį yra grupuojami į 12 tipų šeimų.

Ronaldas Rosas pasiūlė penkis klausimus, į kuriuos reikia atsakyti prieš pradėdant modeliuoti taisykles. Šiuos klausimus galima interpretuoti kaip 5 modeliavimo etapus:

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

Veiklos procesų specifikuojant kalbose veiklos taisyklės skirstomos į tris klases:

- *Veiksmo taisyklės* - taisyklės, kurios reiškia, kad turi būti vykdomas kažkoks veiksmas susidarius tam tikroms sąlygoms, jos atvaizduojamos taip:  
sąlyga ⇒ veiksmas
- *Kilmės taisyklės* - taisyklės, kurios reiškia, kad patvirtinami tam tikri faktai, jei egzistuoja grupė kitų faktų, jos atvaizduojamos taip:  
patvirtinta sąlyga ⇒ sąlyga
- *Apribojimo taisyklės* - taisyklės, pasakančios, kad tam tikros taisyklės neturi būti pažeistos. Jos yra naudojamos įvertinant atliekamus veiksmus.

### 4. Metodai, aprašantys veiklos taisykles.

Projekto tikslus galima įgyvendinti įvairiais programinių komponentų kūrimo paketais. Konkretaus paketo pasirinkimas turėtų priklausyti nuo numatomos veiklos taisyklių saugyklos bei veiklos taisyklių

interpretatoriaus panaudojimo srities. Jei planuojama veiklos taisyklėmis pagrįstą sistemą įdiegti nedidelyje įstaigoje, kurioje veiklos taisyklių kiekis ir duomenų srautai nėra labai dideli, galima naudoti bet kurią šiuolaikinę IS kūrimo sistemą, tačiau didelėse organizacijose, kuriose tenka labai dažnai reaguoti į įvairius situacijos pakitimus ir juos operatyviai pritaikyti, realizacijai reiktų naudoti kitokias kūrimo priemones. [6]

Viena iš metodologijų, naudojama veiklos taisyklių realizavime yra **IDEA**. Tai metodologija, kuri naudoja objektus ir veiklos taisykles duomenų bazėje. Aktyvių taisyklių taikymas - tai originaliausias indėlis IDEA metodologijoje. Susikoncentruojama pritaikant individualias taisykles ir principus pagal analizės poreikius. Vadovaujamosi aktyvios taisyklės nuolatiniu pakeisto varianto patikrinimu, kuris sutelkia dėmesį į tiriamą elgesį nustačius tam tikrą taisyklę, ar jų grupę.[9]

Kitas metodas dažnai naudojamas modeliuoti veiklos taisykles yra **IDEF9**. Šis biznio apribojimų atradimo metodas buvo sukurtas padėti atrasti ir analizuoti biznio sistemos apribojimus. Kartą įtraukus apribojimus į sąrašą jie gali būti sistemiskai analizuojami ir, jeigu reikia, derinami ar pakeičiami, siekiant pagerinti sistemos efektyvumą.[5]

Dar vienas iš dažniau naudojamų metodų yra **OCL** (Object Constrain Language) – kalba padedanti kuo tiksliau išreikšti veiklos taisykles. Tai UML semantikos dalis. Modeliuotojai, kurie įpratę naudoti UML kalbą savo veiklos modeliavimui, gali labai sėkmingai naudoti OCL tam, kad aprašytų tik tai sričiai būdingas taisykles [8]. OCL gali būti naudojama tiksliai apibrėžti klasių ir tipų kintamuosius klasių modelyje; apibrėžti tipų kintamuosius stereotipams; nusakyti operacijų ir metodų išankstines ir vėlesnes sąlygas; apibūdinti apribojimus; kaip navigacijos kalba; tiksliai apibrėžti operacijų apribojimus.

## 5. Veiklos taisyklių saugykla.

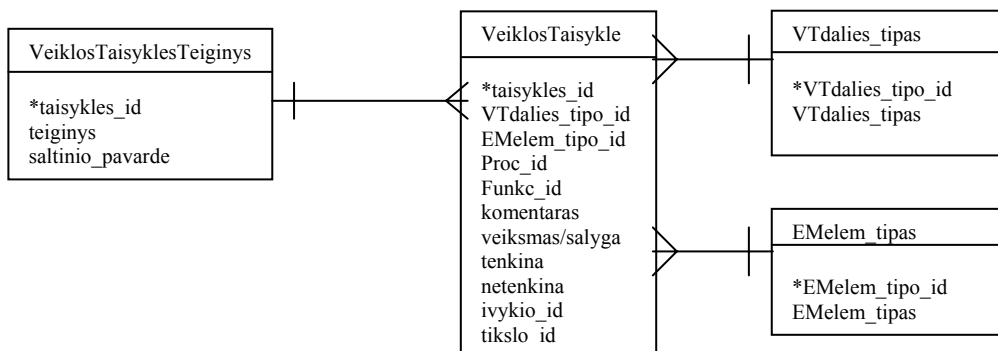
Įšnagrinėjus keletą projektų (GUIDE, Rosso) galima suklasifikuoti veiklos taisykles ir projektuoti duomenų bazę. Tačiau šie autoriai nemodeliavo veiklos taisyklių atsižvelgdami į konkretų veiklos modelį (enterprise model). 2 paveiksle pateikta veiklos taisyklių duomenų bazės dalis. Šiame paveiksle nepateikta sąsaja su pačiu veiklos modeliu.

---Lentelėje **VTdalties\_tipas** kaupiama informacija apie tai, kokie veiklos taisyklių tipai gali būti, t.y. veiksmas, sąlyga, apribojimas, įvykis.

---Lentelėje **EMelem\_tipas** kaupiama informacija apie tai, kurią veiklos valdymo dalį kontroliuos veiklos taisyklės. Galimos reikšmės yra veiklos valdymo funkcijos etapai ir jų informacinės sąsajos (veiklos valdymo modelis aprašytas [1]):

- DA - duomenų apdorojimas;
- IN - interpretacija;
- RE - realizacija;
- P\_out – proceso ir materialaus srauto būsenos atributai;
- P\_in – proceso materialaus srauto valdymo poveikiai;
- IP\_in – informacijos apdorojimo proceso įeigos srautas;
- IP\_out – informacijos apdorojimo proceso išeigos srautas.

---Lentelėje **VeiklosTaisyklesTeiginys** bus įrašomas taisyklės tekstas natūralia kalba. Jį gali pateikti konkretus asmuo arba taisyklę išgauti iš dokumentų, blankų ir pan.

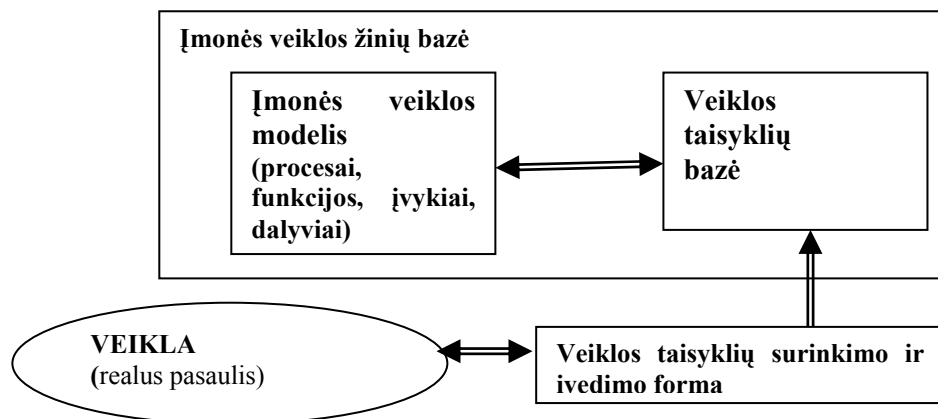


2 pav. Veiklos taisyklių duomenų bazės dalis.

Pirmiausia bus įvedamas veiklos taisyklės teiginys natūralia kalba, o vėliau siejamas su visa veiklos taisyklių duomenų baze.

Pastaraisiais metais pasaulyje ir Lietuvoje jau labai plačiai nagrinėjama veiklos taisyklių modeliavimo problema. Todėl jau yra sukurta daugelis taikomųjų programų, kurių pagalba galima rinkti ir kaupti veiklos taisykles. Tai SmartCost, RuleTrack. Šiose programose jau sukurtos formos, kurių pagalba įvedamos veiklos taisyklės apribojančios funkcijas ar procesus.

Šio projekto metu bus sukurta veiklos taisyklių surinkimo įvedimo forma, iš kurios jos bus perduodamos į veiklos taisyklių bazę. 3 paveiksle pateiktas projekto realizacijos modelis.



3 pav. Projekto realizacijos modelis

## 7. Išvados.

Struktūrizuojant pradinę informaciją apie probleminę sritį ir sudarant veiklos taisyklių specifikaciją, ja galima remtis įvairiuose informacinės sistemos gyvavimo ciklo etapuose. Tikslios veiklos taisyklės specifikacijos sudarymas įgalina pagreitinti projekto realizavimą bei padidina sukurtos sistemos adekvatumą vartotojo reikalavimams ir probleminės srities apribojimams. Vienas pagrindinių uždavinių šiems tikslams pasiekti – VT identifikavimo, specifikavimo bei panaudojimo metodikos, atitinkančios probleminės srities analizės procesą, sudarymas. Tuo tikslu, būtina parinkti ne tik atitinkamoms veiklos taisyklėms adekvačias specifikavimo formas, bet ir detalai aprašyti procesus, kurių metu veiklos taisyklių specifikacija bus sudaroma. Parenkant specifikavimo formas būtina įvertinti tai, kad veiklos taisyklės specifikacija galėtų naudotis ne tik tiesioginiai IS kūrėjai, bet ir kompiuterizuojamos veiklos ekspertai bei būsimieji sistemos vartotojai.

Šio projekto naujovė ta, kad tai bus ne atskira informacijos sistema surenkanti ir analizuojanti veiklos taisykles, bet planuojama sukurti CASE priemonės įrankį. Šio įrankio pagalba būtų galima detalai modeliuoti organizacijos veiklą, į projektavimą įvedant veiklos taisykles.

## 9.2 Pasulyje pateiktų programų veiklos taisyklių kūrimo langai.

### ERL kalbos pavyzdys

Pateiksime vieną iš veiklos procesų specifikuojančių pavyzdžių ( lentelė), kuris aprašo sandėlininko veiklos EVC. ERL kalba (*external rules language*) panaudota interpretavimo ir realizavimo procesų specifikuojimui.

lentelė

<b>EVC komponentė</b>	<b>Formalus aprašas</b>
EVC technologinis procesas TP	$MP_N = (PAJAMUOTI(PrekesXSiuunta), SANDELIUOTI(PrekesXSiuunta))$
EVC įeigos materialus srautas MS	PrekesXSiuunta = (Pavadinimas, Kiekis, Kaina, Tiekejas)
Interpretavimo procesas IN	Prekes_kodas = RASTI(Pavadinimas) IF Prekes_kodas = 0 THEN prekes_kodas = PAPILDYTI(pavadinimas) END ELSE RASyti I DB(prekes_kodas, kaina, kiekis, tiekejas)
Informacinis procesas IP	DUOMENU IRASYMO PROCESAS, PREKIU DUOMENU SRAUTAS, SAUGYKLA
Realizavimo procesas RE	SPAUSDINTI ZINIARASTI (prekes_kodas)
EVC išeigos materialus srautas MS*	PajamuotuPrekiuXSiuunta = (Ziniarastis)

### RuleTrack programos veiklos taisyklių kūrimo langai.





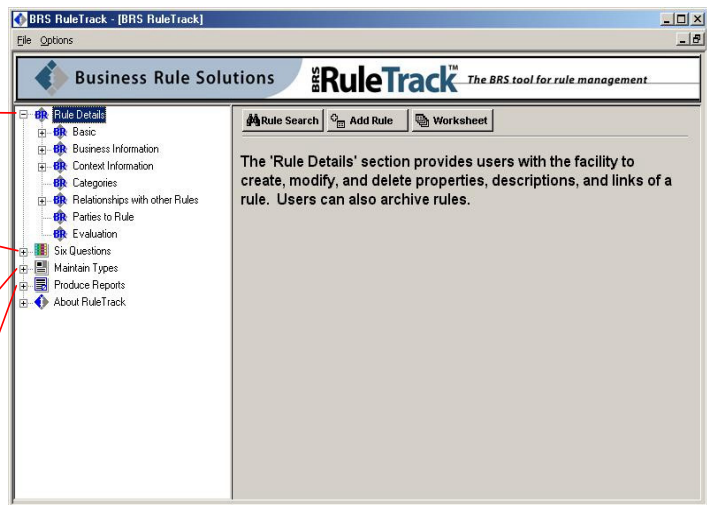
## Overview: What Does BRS RuleTrack Do For You?

**Rule Details...** Record your business rules, and specify their properties and links. Manage your business terms interactively. Archive old versions.

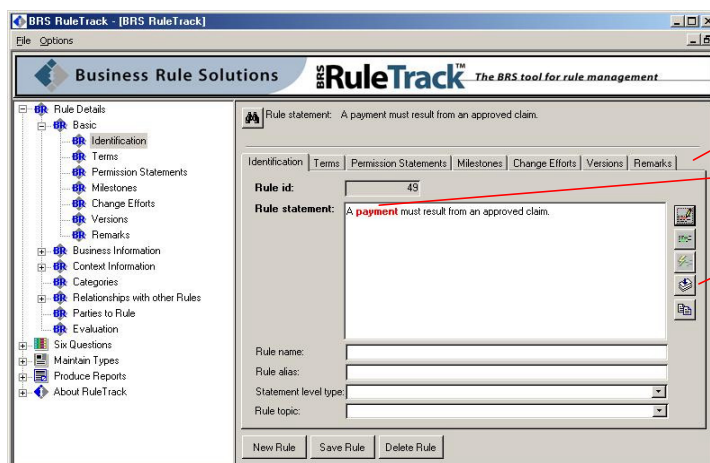
**Six Questions...** Define elements of the business and systems context that can be linked to your business rules.

**Maintain Types...** Manage current valid values for type data, including extensive predefined categories.

**Produce Reports...** Trace your business rules based on user-selected filtering criteria.



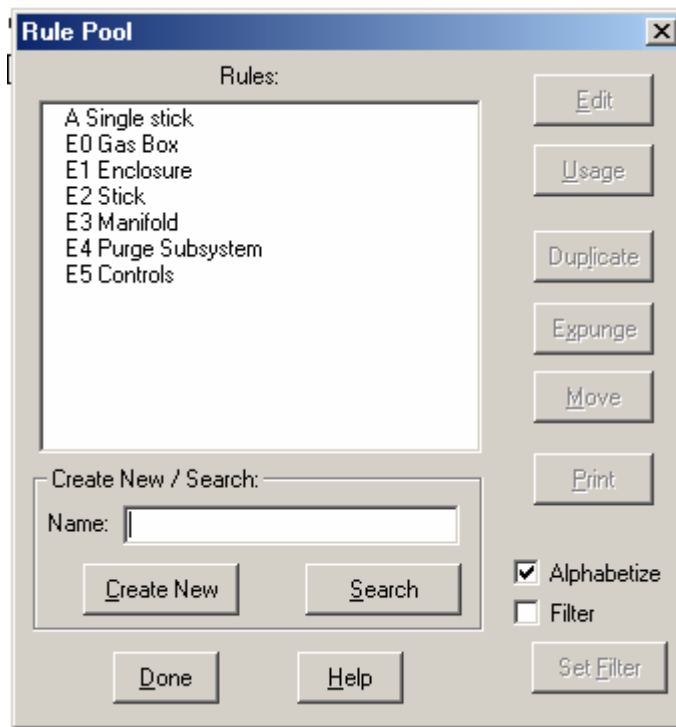
## Overview: What Does BRS RuleTrack Do For You?



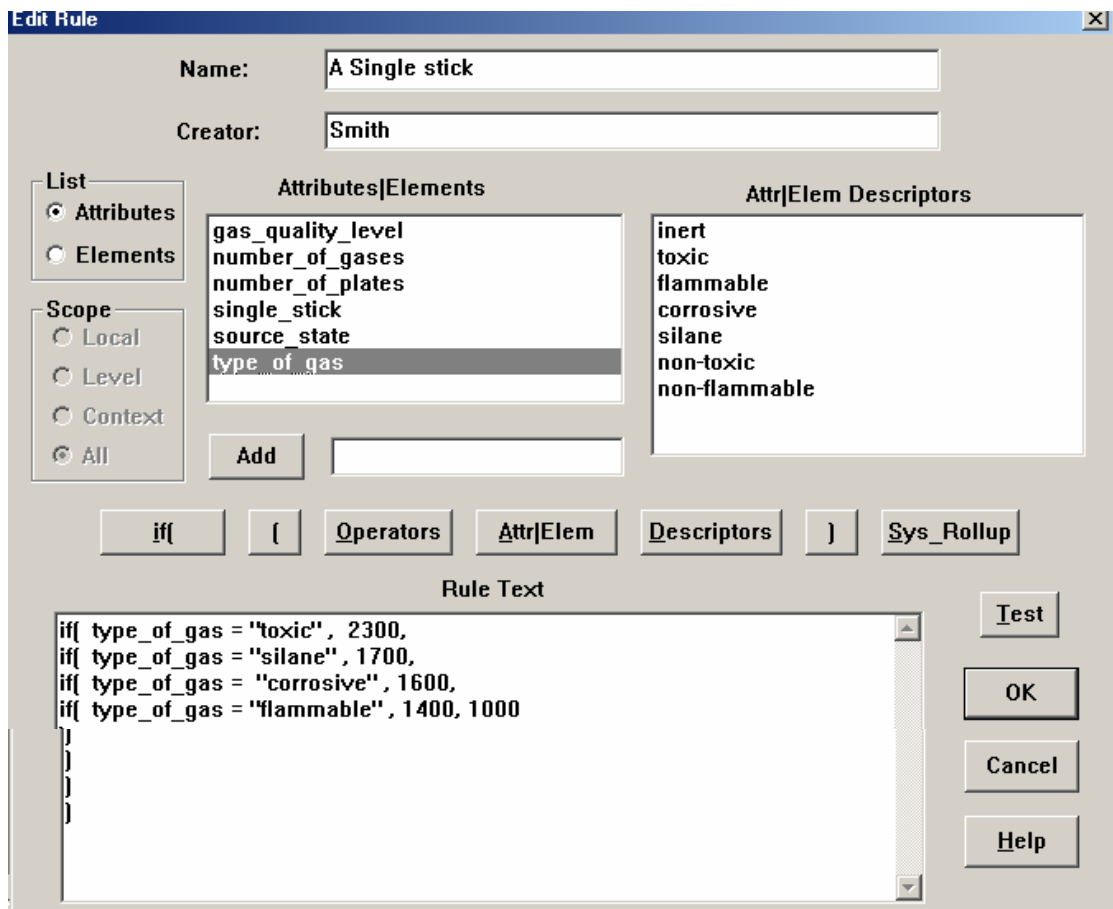
Record your business rules.

Manage your business terms interactively.

Archive old version.



pav. SmartCost programos veiklos taisyklių kūrimo langas



pav. SmartCost programos veiklos taisyklių kūrimo langas.

### **9.3 Priedas 3. Sistemos vartotojo vadovas.**

## **Veiklos taisyklių redaktoriaus vartotojo vadovas**

# TURINYS

<b>1. VEIKLOS TAISYKLIŲ REDAKTORIAUS PASKIRTIS.....</b>	<b>77</b>
<b>2. VEIKLOS TAISYKLIŲ REDAKTORIAUS SUDEDAMOSIOS DALYS.....</b>	<b>77</b>
2.1. VEIKLOS TAISYKLIŲ REDAKTORIAUS MENIU DALYS.....	77
2.2 VEIKLO TAISYKLIŲ REDAKTORIAUS VALDYMO BLOKAS.....	78
<b>3.REIKALAVIMAI TECHNINEI IR PROGRAMINEI ĮRANGAI.....</b>	<b>79</b>
<b>4.DARBO PRADŽIA SU VEIKLOS TAISYKLIŲ REDAKTORIUMI.....</b>	<b>79</b>
4.1 VEIKLOS TAISYKLIŲ REDAKTORIAUS ĮDIEGIMAS.....	79
4.2 PRISIJUNGIMAS IR ATSIJUNGIMAS IŠ VEIKLOS TAISYKLIŲ REDAKTORIAUS.....	79
<b>5. VEIKLOS TAISYKLIŲ REDAKTORIAUS FUNKCIJOS.....</b>	<b>80</b>
5.1 MENIU PUNKTŲ FUNKCIJOS.....	80
5.1.1 Meniu punkto “Naturalios VT” funkcijos.....	80
5.1.2 Meniu punkto “Terminai” funkcijos.....	81
5.1.3 Meniu punkto “Faktai” funkcijos.....	82
5.1.4 Meniu punkto “Formalios VT” funkcijos.....	84
5.1.5 Meniu punkto “Šablonai” funkcijos.....	86
<b>6. SISTEMOS PRIVALUMAI IR TRŪKUMAI.....</b>	<b>88</b>

## 1. Veiklos taisyklių redaktoriaus paskirtis.

Veiklos taisyklių redaktorius – tai veiklos taisyklių kaupimo, jų formalizavimo ir saugojimo duomenų bazėje sistema. Ši sistema suteikia galimybę atlikti tokius veiksmus:

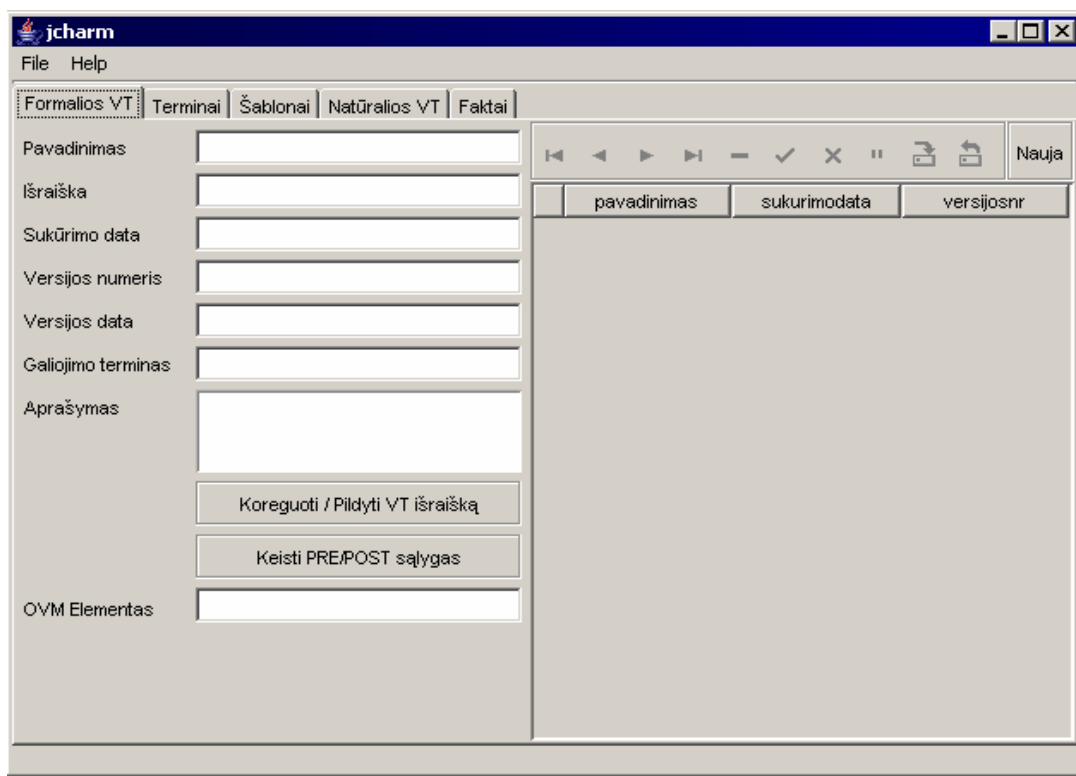
- ❖ įvesti veiklos taisykles natūralia kalba;
- ❖ pildyti terminus;
- ❖ pildyti faktus;
- ❖ kurti rezervuotus žodžius;
- ❖ formalizuoti veiklos taisykles, įvestas natūralia kalba;
- ❖ redaguoti formalizuotas veiklos taisykles;
- ❖ sukurti naujus šablonus.

## 2. Veiklos taisyklių redaktoriaus sudedamosios dalys.

### 2.1. Veiklos taisyklių redaktoriaus meniu dalys.

Veiklos taisyklių redaktorių sudaro tokie meniu punktai:












- ❖ **Terminai** – pasirinkus šį meniu punktą bus galima apsirašyti visus veiklos taisyklių formalizavimui reikalingus terminus. Taip pat šio meniu punkto pagalba galima redaguoti jau įvestus terminus.
- ❖ **Faktai** – pasirinkus šį meniu punktą apsirašomi faktai pagal vieną šabloną - *<Terminas1><rezervuotas žodis><Terminas2>*. Čia rezervuotas žodis gali skirtis. Jei reikalingas naujas rezervuotas žodis, tuomet pasirenkame mygtuką “Rezervuoti žodžiai”.
- ❖ **Natūralios VT** – probleminės srities analitikas suveda veiklos taisykles natūralia kalba, neformalizuotai. Jei reikalinga, gali jas ir redaguoti.
- ❖ **Formalios VT** – tai pagrindinis meniu punktas. Jo pagalba galima formalizuoti veiklos taisykles. Pateiktoje formoje suvedama informacija apie formalizuojamą veiklos taisyklę, pasirenkamas šablonas, pagal kurį bus pildoma VT.
- ❖ **Šablonai** – šioje formoje sukuriami reikalingi veiklos taisyklių formalizavimui šablonai. Jie kuriami įterpinėjant reikalingus šablono elementus.



58 pav. Veiklos taisyklių redaktoriaus pagrindinis langas.

## 2.2 Veiklo taisyklių redaktoriaus valdymo blokas.

Šioje sistemoje valdymo blokas pateikiamas kiekvienoje formoje. Jį sudaro:

-  ❖ rodyti pirmą įrašą;
-  ❖ rodyti ankstesnį įrašą;
-  ❖ rodyti sekantį įrašą;
-  ❖ rodyti paskutinį įrašą;
-  ❖ įterpti naują eilutę;
-  ❖ ištrinti pasirinktą įrašą;
-  ❖ išsaugoti pakeitimus pasirinktame įrašė;
-  ❖ neišsaugoti prieš tai padarytų pakeitimų pasirinktame įrašė;
-  ❖ nukopijuoti ankstesnį įrašą;
-  ❖ išsaugoti pakeitimus;
-  ❖ atnaujinti duomenis.



### **3.Reikalavimai techninei ir programinei įrangai.**

Norint dirbti su veiklos taisyklių redaktoriumi, reikia, kad būtų tenkinamos šios sąlygos:

- ❖ Vartotojo personaliniame kompiuteryje turi būti įdiegta MySQL duomenų bazių valdymo sistema tam, kad būtų galima matyti duomenų bazės schemą ir joje kaupiamus duomenis. Be jos neveikia visa sistema.
- ❖ Vartotojo personaliniame kompiuteryje turi būti instaliuotas Java Web Start paketas.
- ❖ Vartotojų personaliniai kompiuteriai turi tenkinti jiems keliamus reikalavimus.

#### Reikalavimai vartotojo kompiuteriui:

Pentium II 450MHz, 128MB RAM, 6.4 – 10 GB SCSI HDD, 10/100 Ethernet card, 2 MB Video card, 32x CD-ROM.

### **4.Darbo pradžia su veiklos taisyklių redaktoriumi.**

#### **4.1 Veiklos taisyklių redaktoriaus įdiegimas.**

Prieš pradėdant darbą su šia sistema, ją reikia įdiegti darbui paruoštame kompiuteryje.

- ❖ Pirmiausia reikia suinstaliuoti MySQL paketą. Jis suinstaliuojamas paleidus *setup.exe* failą iš katalogo “*mysql*”.
- ❖ Po to jame reikia susikurti kol kas tuščią duomenų bazę vardu “*DB007*”.
- ❖ Po to suinstaliuojame Java Web Start paketą, pasirinkdami “*j2re-1\_4\_2\_02-windows-i586-p.exe*” failą.
- ❖ Sukurta tuščia duomenų bazė sugeneruojama, pasirinkus failą “*Setup*”iš katalogo “*install*”. Jame reikia įrašyti duomenų bazės pavadinimą ir pasirinkti mygtuką “Tikrinti”, o gavus pranešimą “OK”, paspausti mygtuką “Install database”. Jei viskas pavyko, bus pateikiamas pranešimas “Duomenų bazė perkrauta”.
- ❖ Programa paleidžiama pasirinkus vykdomąjį failą “*jcharmW.exe*”.

#### **4.2 Prisijungimas ir atsijungimas iš veiklos taisyklių redaktoriaus.**

Prisijungimas prie veiklos taisyklių redaktoriaus vyksta paleidžiant vykdomąjį Java programavimo kalba suprogramuotą failą pavadinimu “*jcharmW.exe*”. Papildomų prisijungimo operacijų nenumatyta. Nereikalaujama nei vartotojo identifikacinio vardo, nei slaptažodžio.

Atsijungimas nuo veiklos taisyklių redaktoriaus taip pat paprastas. Tereikia pasirinkti meniu punktą “File” ir “Exit” arba tiesiog paspausti kryželį lango kamputyje.

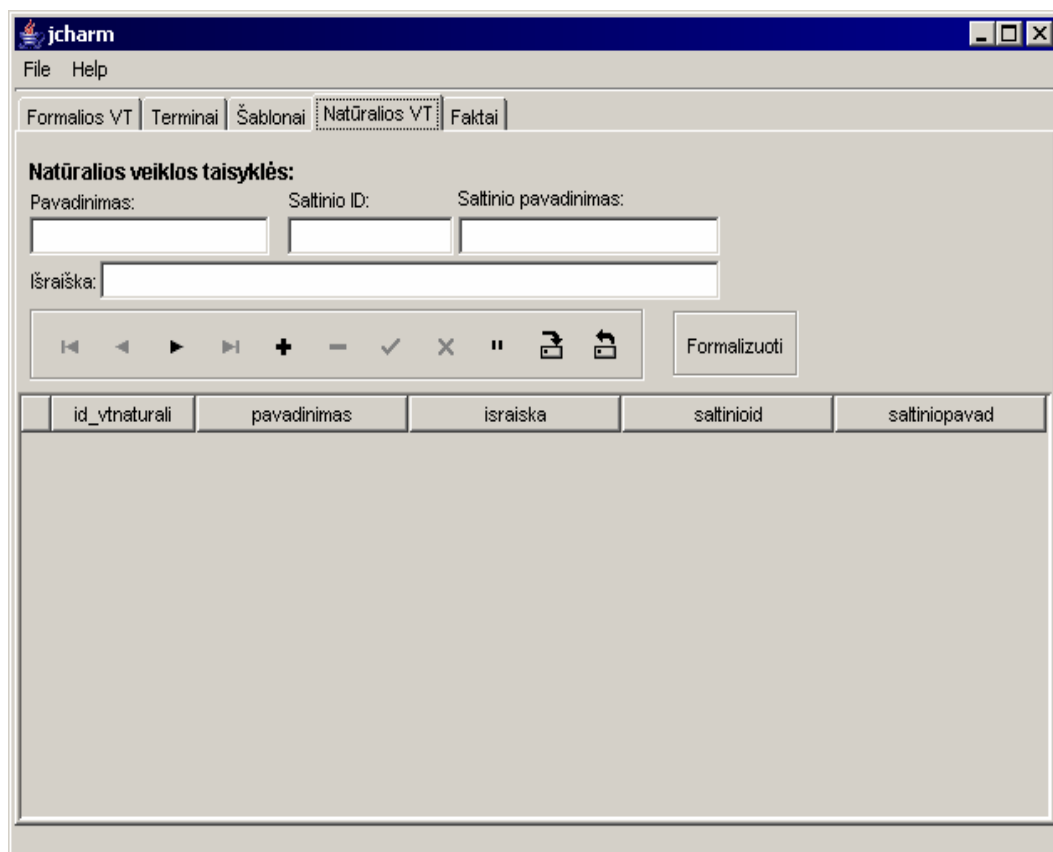
## 5. Veiklos taisyklių redaktoriaus funkcijos.

Šiame skyriuje aprašyta visos veiklos taisyklių redaktoriuje galimi atlikti veiksmai.

### 5.1 Meniu punktų funkcijos.




Šiame skyriuje bus aprašytos išsamiai išnagrinėtos meniu punktų funkcijos.

#### 5.1.1 Meniu punkto “Natūralios VT” funkcijos.




60 pav. Veiklos taisyklių redaktoriaus langas, pasirinkus meniu punktą “Natūralios VT”.

Šioje formoje (60 pav.) probleminės srities anallitikas suveda veiklos taisyklės natūralia kalba, atlikdamas sekančius veiksmus:

- ❖ Jei tai jau ne pirma veiklos taisyklė, tuomet reikia paspausti mygtuką  , kad įterpti naują eilutę;
- ❖ Pirmiausia įvedamas natūralios veiklos taisyklės pavadinimas;
- ❖ Po to duomenys apie šaltinį, t.y. identifikacinis numeris ir pavadinimas.
- ❖ Galiausiai įvedama natūralios veiklos taisyklės išraiška. Ją gali sudaryti keletas neformalizuotų sakinių, iš kurių vėliau bus formalizuojama veiklos taisyklė.
- ❖ Paspaudžiame  mygtuką , kad išsaugoti sukurtą veiklos taisyklę natūralia kalba;
- ❖ Paskutinis veiksmas – paspaudžiame mygtuką  , kad atnaujinti duomenis



lentelėje. Šio etapo metu susikuria identifikacinis natūralios veiklos taisyklės numeris.

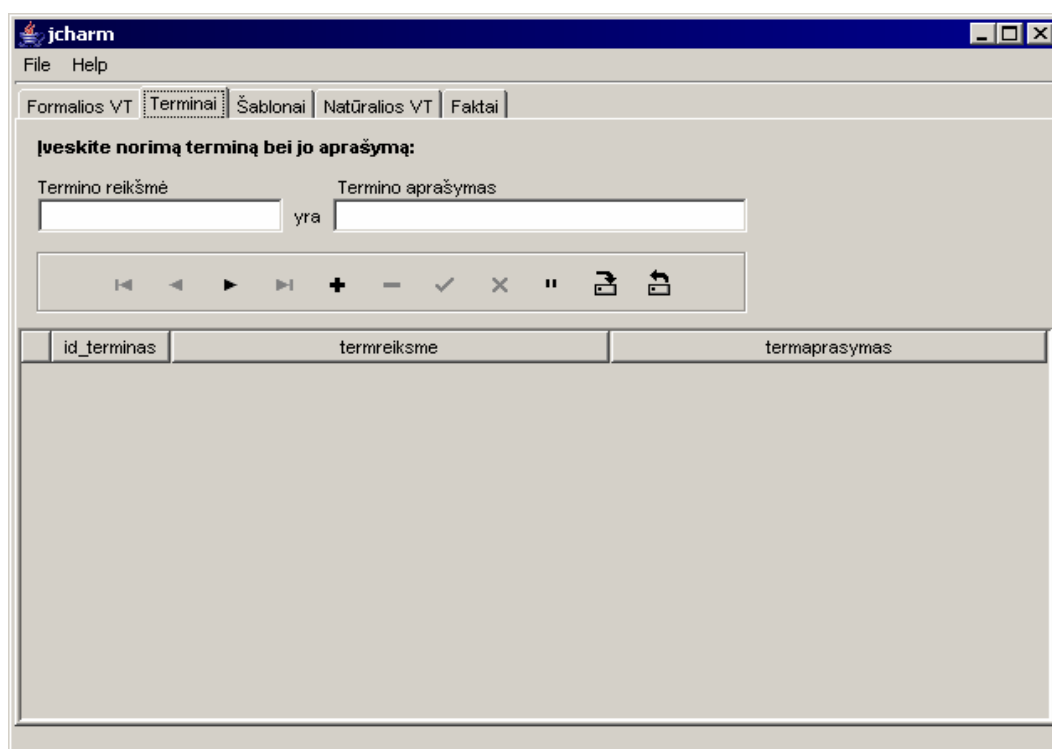
- ❖ Jei norime ištrinti nereikalingą natūralią veiklos taisyklę, pažymime ją ir pasirenkame  mygtuką .

- ❖ Jei norime redaguoti kurią tai natūralią veiklos taisyklę, pažymime ją, pakoreguojame kas reikalinga ir  paspaudžiame mygtuką .

Probleminės srities analitikas tik suveda veiklos taisyklės natūralia kalba, tačiau nesiima jų formalizuoti. Tai jau yra IS projektuotojo pagrindinė funkcija.


### 5.1.2 Meniu punkto “Terminai” funkcijos.






Terminus pildo IS projektuotojas, prieš tai išsiaiškinęs, kokių jų reiks faktų pildymui ir veiklos taisyklių formalizavimui. Terminai pildomi pagal vienintelį šabloną <Termino reikšmė><yra><Termino apibūdinimas>. 61 paveiksle pateiktas langas, pasirinkus meniu punktą “Terminai”.



61 pav. Sistemos langas, pasirinkus meniu punktą “Terminai”.

Terminų pildymo veiksmų seka:






- ❖ Pasirenkame meniu punktą “Terminai”;
- ❖ Jei tai jau ne pirmas terminas, tuomet reikia paspausti  mygtuką , kad įterpti naują eilutę;
- ❖ Įrašoma termino reikšmė;
- ❖ Įrašomas termino apibūdinimas. Rezervuotas žodis visada išlieka tas pats;

- ❖ Paspaudžiame  mygtuką , kad išsaugoti sukurtą terminą;
- ❖ Paskutinis veiksmas – paspaudžiame mygtuką , kad atnaujinti duomenis lentelėje. Šio etapo metu susikuria identifikacinis termino numeris.
- ❖ Jei tęsiame terminų pildymą, tai vėl paspaudžiame  mygtuką .
- ❖ Jei norime ištrinti nereikalingą terminą, pažymime jį ir pasirenkame  mygtuką .
- ❖ Jei norime redaguoti kurį tai terminą, pažymime jį, pakoreguojame kas reikalinga ir paspaudžiame mygtuką  .

### 5.1.3 Meniu punkto “Faktai” funkcijos.

Faktų pildymui reikalingi terminai ir rezervuoti žodžiai. Pasirinkus šį meniu punktą, matomas mygtukas “Rezervuoti žodžiai”, kurį paspaudus galima bus kurti naujus reikalingus rezervuotus žodžius. Jei pildant faktą paaiškėja, kad nėra reikalingų terminų, tai tiesiog uždaroma faktų pildymo forma, pasirenkamas meniu “Terminai”, sukuriamas reikalingas terminas, o po to vėl sugrįžtama į meniu punktą “Faktai”, kur pasirinkus nebaigtą pildyti faktą, galima užbaigti jo pildymą.

#### Rezervuotų žodžių kūrimo eiga (62 pav.):

- ❖ Pasirenkamas meniu punktas “Faktai”, mygtukas “Rezervuoti žodžiai”;
- ❖ Jei tai jau ne pirmas rezervuotas žodis, tuomet reikia paspausti mygtuką , kad įterpti naują eilutę;
- ❖ Įrašoma rezervuoto žodžio reikšmė;
- ❖ Įrašoma ryšio prasmė (jei tai savybę charakterizuojantis rezervuotas žodis);
- ❖ Paspaudžiame mygtuką , kad išsaugoti sukurtą rezervuotą žodį;
- ❖ Paskutinis veiksmas – paspaudžiame mygtuką , kad atnaujinti duomenis lentelėje. Šio etapo metu susikuria identifikacinis rezervuoto žodžio numeris.
- ❖ Jei norime ištrinti nereikalingą rezervuotą žodį, pažymime jį ir pasirenkame mygtuką  .
- ❖ Jei norime redaguoti kurį tai rezervuotą žodį, pažymime jį, pakoreguojame kas reikalinga ir paspaudžiame mygtuką  .

**Rezervuoti žodžiai**

**Įveskite norimą rezervuotą žodį:**






Rezervuoto žodžio reikšmė:

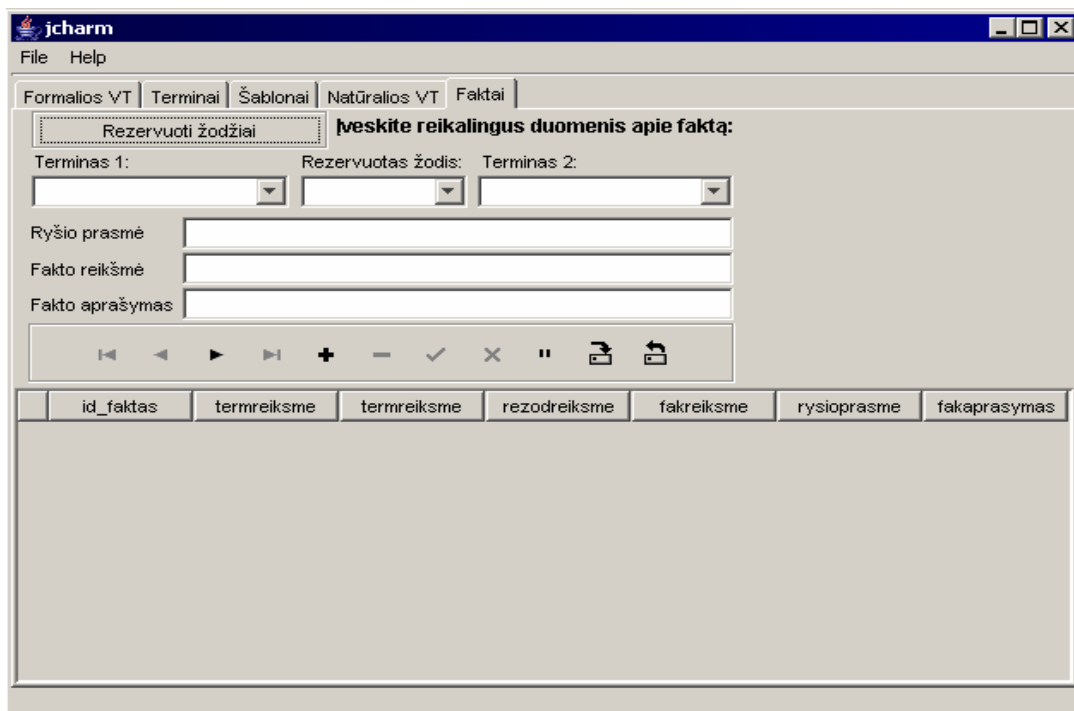
Ryšio prasmė:

	id_rezervuotaszodis	rezodreiksme	rysioprasme
1	1	turi	savybe
2	2	yra	savybe
3	3	+	
4	4	-	
5	5	*	
6	6	/	
7	7	=	
8	8	daugiau negu	
9	9	maziau negu	

62 pav. Sistemos langas pasirinkus meniu punktą "Faktai" ir mygtuką "Rezervuoti žodžiai".

#### Faktų pildymo seka:

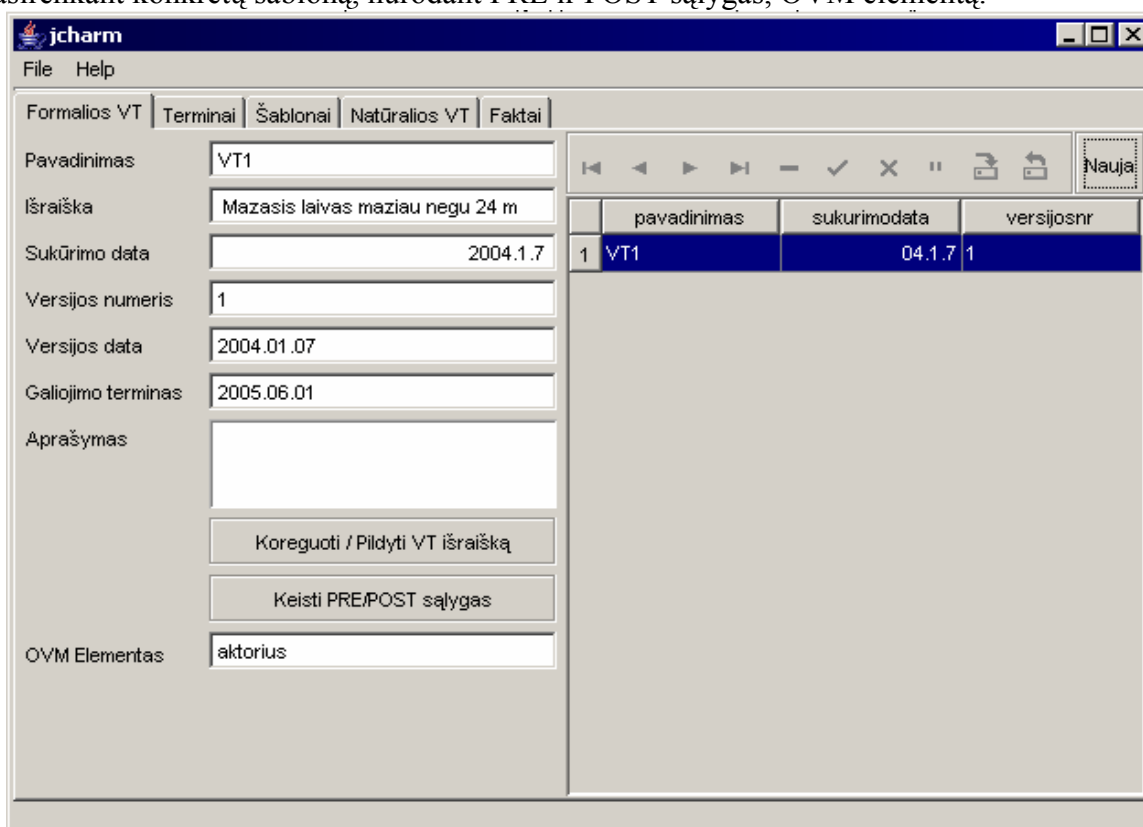
- ❖ Pasirenkamas meniu punktas "Faktai";
- ❖ Jei tai jau ne pirmas rezervuotas žodis, tuomet reikia paspausti mygtuką , kad įterpti naują eilutę;
- ❖ Pasirenkamas pirmas terminas, rezervuotas žodis ir antras terminas;
- ❖ Užpildomi laukai: "Ryšio prasmė", "Fakto reikšmė", "Fakto aprašymas".
- ❖ Paspaudžiame mygtuką , kad išsaugoti sukurtą faktą;
- ❖ Paskutinis veiksmas – paspaudžiame mygtuką , kad atnaujinti duomenis lentelėje. Šio etapo metu susikuria identifikacinis fakto numeris.
- ❖ Jei norime ištrinti nereikalingą faktą, pažymime jį ir pasirenkame mygtuką .
- ❖ Jei norime redaguoti kurį tai faktą, pažymime jį ir pakoreguojame kas reikalinga ir paspaudžiame mygtuką .



63 pav. Sistemos langas, pasirinkus meniu punktą "Faktai".

#### 5.1.4 Meniu punkto "Formalios VT" funkcijos.

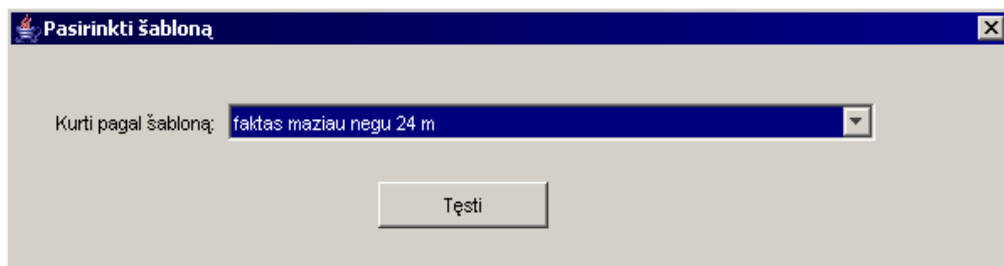
Tai pagrindinis sistemos meniu punktas. Jame vyksta veiklos taisyklių formalizavimas pasirinkant konkretų šabloną, nurodant PRE ir POST sąlygas, OVM elementą.



64 pav. Sistemos langas pasirinkus meniu punktą "Formalios VT".

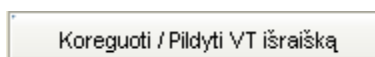
Formalių veiklos taisyklių pildymo seka:

- ❖ Pirmiausia pasirenkame meniu punktą “Naturalios VT” ir išsirenkame veiklos taisyklę, kurią norime formalizuoti;
- ❖ Mums pateikia šablonų pasirinkimo langą (65 pav.)



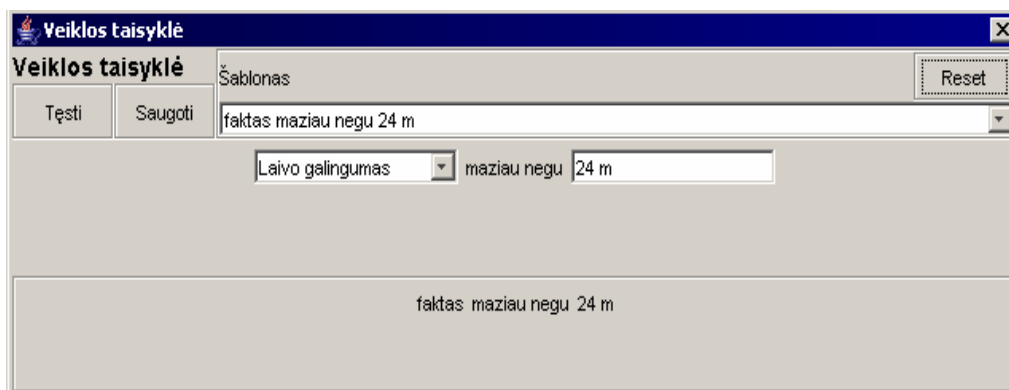
65 pav. Šablonų pasirinkimo langas.

- ❖ Paspaudus mygtuką “Tęsti”, grįžtama į pagrindinį formalių veiklos taisyklių pildymo langą;
- ❖ Užpildomi laukai: Pavadinimas, Versijos nr, Versijos data, Galiojimo terminas ir Aprašymas.
- ❖ Laukai “Sukūrimo data” ir “Išraiška” užsipildo automatiškai.
- ❖ Po to pasrenkame mygtuką “Koreguoti/Pildyti VT išraišką”

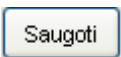
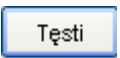


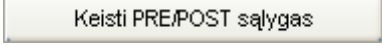
66 pav. Mygtukas “Koreguoti/ Pildyti VT išraišką”.

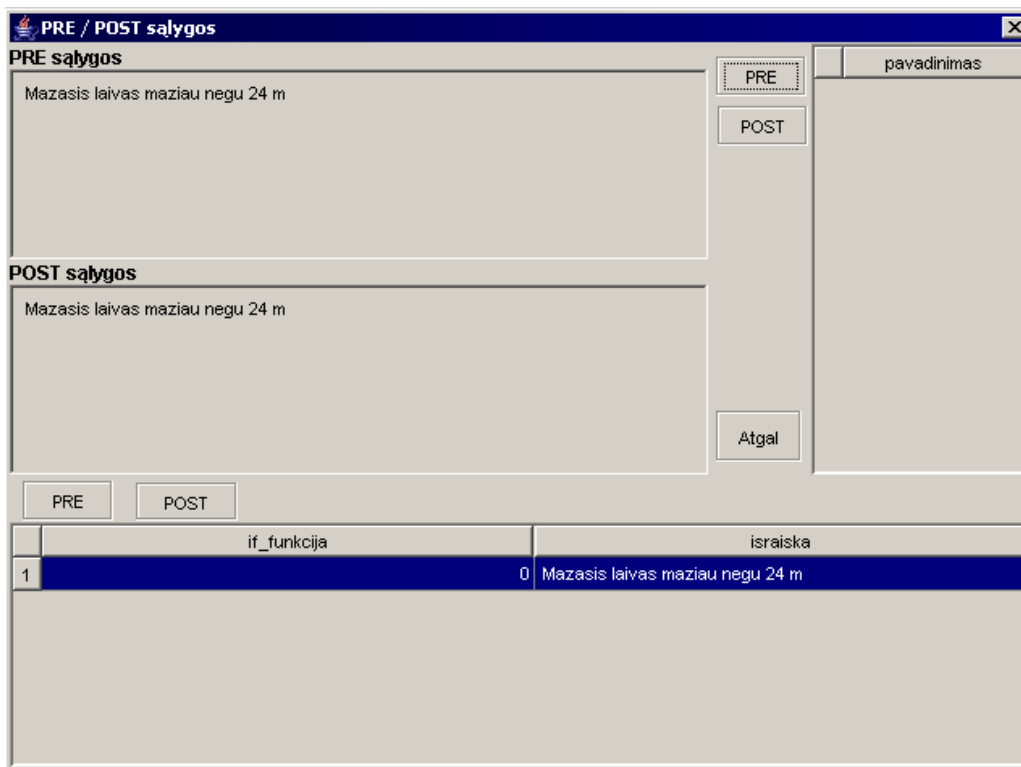
- ❖ Pažymėjus norimą pildyti/ koreguoti veiklos taisyklę, pateikiama forma “Veiklos taisyklė”:



67 pav. Veiklos taisyklių pildymo/koregavimo langas.





Joje reikia paspausti mygtuką “Reset”, kad būtų galima pildyti/koreguoti šabloną konkrečiomis reikšmėmis. Lango apačioje matoma pildoma/ koreguojama šablono išraiška. Užpildžius šabloną konkrečiomis reikšmėmis, spaudžiame mygtuką  ir išsaugome užpildytą/koreguotą veiklos taisyklę. Po to spaudžiame mygtuką  ir tęsiame veiklos taisyklių formalizavimą.

- ❖ Pasirenkame mygtuką , kad galėtume nustatyti egzistuojančias PRE ir POST sąlygas, kurios gali būti įvykiai ir kitos veiklos taisyklės. Paspaudus šį mygtuką pateikiama PRE ir POST sąlygų pildymo forma:



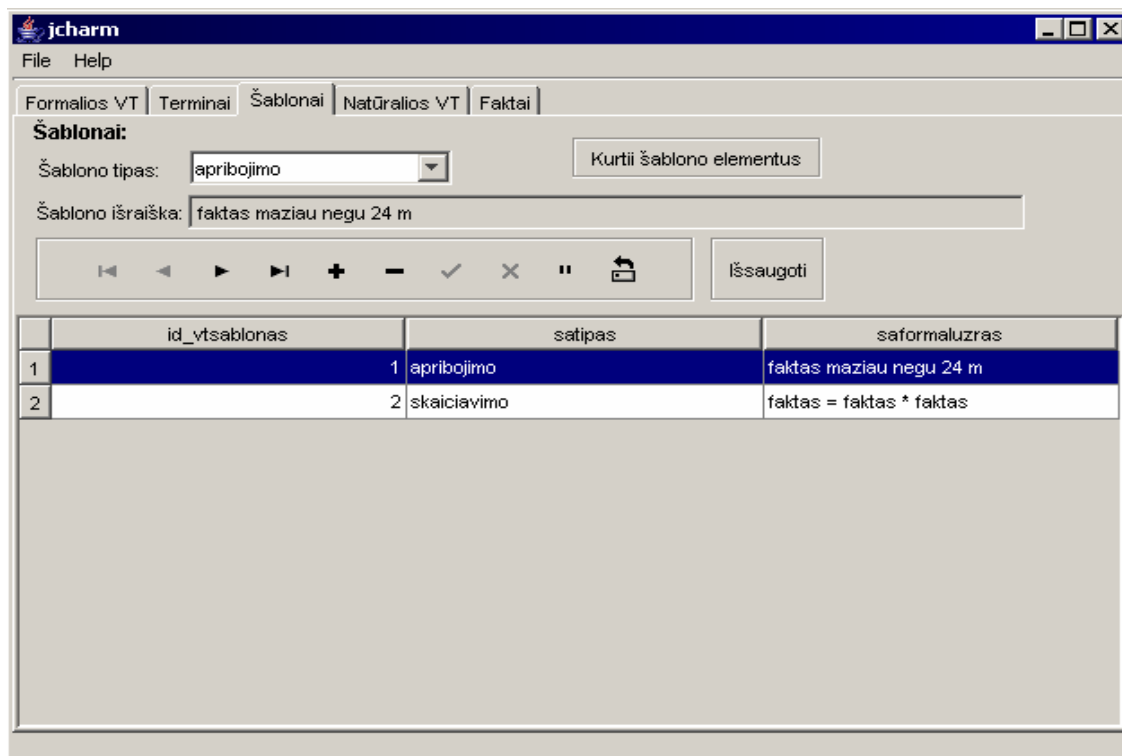
	if_funkcija	israiska
1	0	Mazasis laivas maziau negu 24 m

68 pav. PRE ir POST sąlygų pildymo langas.

- ❖ Galiausiai užpildomas OVM elementas, kuris įrašomas ranka.
- ❖ Paspaudžiame mygtuką , kad išsaugoti sukurtą formalią veiklos taisyklę;
- ❖ Paskutinis veiksmas – paspaudžiame mygtuką , kad atnaujinti duomenis lentelėje. Šio etapo metu susikuria identifikacinis formalios veiklos taisyklės numeris.
- ❖ Jei norime ištrinti nereikalingą formalią veiklos taisyklę, pažymime ją ir pasirenkame mygtuką .
- ❖ Jei norime redaguoti kurią tai formalią veiklos taisyklę, pažymime ją, pakoreguojame kas reikalinga ir paspaudžiame mygtuką .


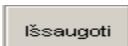

### 5.1.5 Meniu punkto “Šablonai” funkcijos.

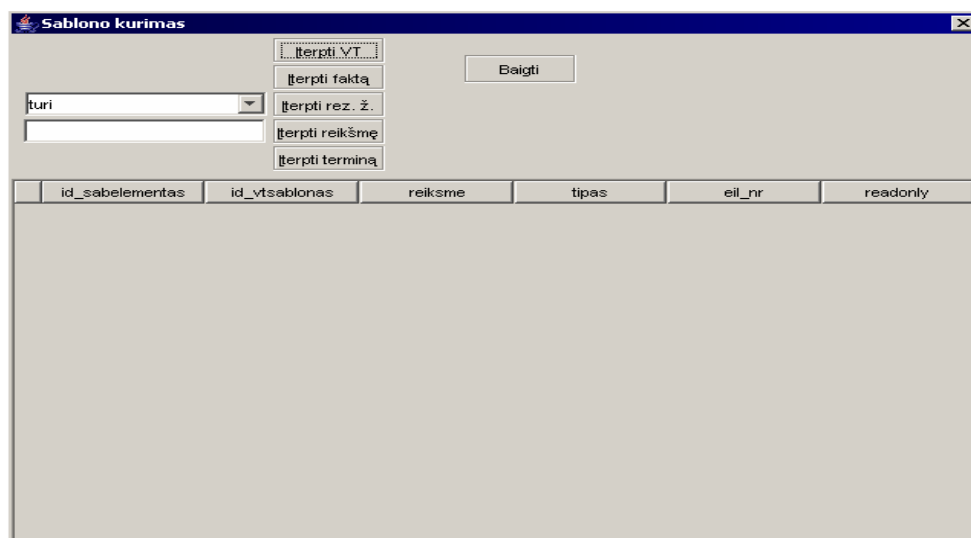
Pasirinkus meniu punktą “Šablonai”, bus galima kurti naujus šablonus.



69 pav. Šablonų kūrimo/ redagavimo langas.

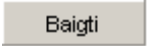



Šablonų pildymo veiksmų seka:

- ❖ Pasirenkamas meniu punktas “Šablonai”;
- ❖ Jei tai jau ne pirmas šablonas, tuomet reikia paspausti mygtuką , kad įterpti naują eilutę;
- ❖ Pasirenkamas šablono tipas ir spaudžiamas mygtukas  tam, kad susikurtų naujo šablono identifikacinis numeris;
- ❖ Po to kuriame šablono elementus, paspaūžiame mygtuką .
- ❖ Pateikiama forma „Šablono kūrimas“.



70 pav. Šablono elementų kūrimo langas.

Šablono elementų įterpimo tvarka priklauso nuo to, koks yra pasirinktas šablono tipas. Tikrinama pildymo seka, pagal sudarytus šablono struktūros medžius. Elementai, kuriuos galima įterpti, pateikiami ryškesni. Įterpinėjant faktą, terminą arba kitą veiklos taisyklę, nurodoma tik tai, kad toks elementas bus šablone (nurodomas tik kaip tipas), o įterpinėjant rezervuotus žodžius ir reikšmę, galima iš karto nurodyti konkrečią reikšmę.

- ❖ Baigti įterpinėti šablona sudarančius elementus galima dviem būdais:
  - arba paspaudus kryžiuką kampe (neteisingas išėjimas);
  - arba leidžiama baigti įterpinėti elementus, kai pasidaro ryškus mygtukas  (teisingas išėjimas).
- ❖ Paspaudžiame mygtuką , kad išsaugoti sukurtą šablona;
- ❖ Paskutinis veiksmas – paspaudžiame mygtuką , kad atnaujinti duomenis lentelėje.
- ❖ Jei norime ištrinti nereikalingą šablona, pažymime jį ir pasirenkame mygtuką .
- ❖ Redaguoti kurį tai šablona neleidžiama. Jei nėra reikalingo šablono, tuomet susikuriame naują.

## 6. Sistemos privalumai ir trūkumai.

Sukurta sistema skirta veiklos taisyklių kaupimui, formalizavimui, saugojimui duomenų bazėje ir jų redagavimui.

### Sistemos privalumai:

- ❖ sistemoje egzistuoja lietuvių kalba;
- ❖ pakankamai patogi, aiški ir nesudėtinga vartotojų sąsaja;
- ❖ nesudėtingas sistemos įdeigimas.

### Sistemos trūkumai:

- ❖ ne iki galo išvystyta programa programavimo atžvilgiu;
- ❖ neatlikta pilna duomenų kontrolė.