

**VILNIAUS UNIVERSITETAS
KAUNO HUMANITARINIS FAKULTETAS**

INFORMATIKOS KATEDRA

Verslo informatikos studijų programa
Kodas

DARIUS GRAŽULIS

MAGISTRO BAIGIAMASIS DARBAS

**VEIKLOS TAISYKLĖMIS GRINDŽIAMOS
REIKALAVIMŲ KURIAMAI INFORMACINEI
SYSTEMAI SPECIFIKACIJOS SAUGOJIMO SISTEMA**

Kaunas, 2007

**VILNIAUS UNIVERSITETAS
KAUNO HUMANITARINIS FAKULTETAS**

INFORMATIKOS KATEDRA

Verslo informatikos studijų programa
Kodas

DARIUS GRAŽULIS

MAGISTRO BAIGIAMASIS DARBAS

**VEIKLOS TAISYKLĖMIS GRINDŽIAMOS
REIKALAVIMŲ KURIAMAI INFORMACINEI
SISTEMAI SPECIFIKACIJOS SAUGOJIMO SISTEMA**

Leidžiama ginti _____

Magistrantas _____

(parašas)

Darbo vadovas _____

(parašas)

Daktaras, docentas Rimantas Butleris

(darbo vadovo mokslinis laipsnis, mokslo pedagoginis
vardas, vardas, pavardė)

Darbo įteikimo data _____

Registracijos Nr. _____

Kaunas, 2007

TURINYS

SANTRUMPŲ SARAŠAS	4
PAVEIKSLŲ SARAŠAS.....	5
LENTELIŲ SARAŠAS.....	6
ĮVADAS.....	7
1 VEIKLOS TAISYKLIŲ METODO PRICIPAI.....	9
1.1 Veiklos taisyklė.....	9
1.2 Veiklos taisyklių metodo vystymosi prielaidos	9
1.3 Bendri reikalavimai veiklos taisyklėms	10
1.4 Veiklos taisyklių klasifikacija.....	11
1.5 Veiklos taisyklių specifikuojamo proceso.....	16
2 VEIKLOS TAISYKLĖMIS GRINDŽIAMAS VARTOTOJO POREIKIŲ SPECIFIKAVIMAS.....	20
2.1 Veiklos taisyklių struktūrizavimo modelis	20
2.2 Vartotojo reikalavimų specifikuojamo proceso.....	22
2.2.1 Veiklos konteksto apibrėžimas ir aktorių nustatymas.....	23
2.2.2 Funkcijų ir funkcijų sekų atskleidimas	24
2.2.3 Struktūrinių veiklos taisyklių atskleidimas.....	25
2.2.4 CRUD matricos kūrimas.....	26
2.2.5 Veiklos taisyklių šaltinių fiksavimas	28
2.2.6 Nestrukūrinė veiklos taisyklių atskleidimas.....	28
2.3 Užfiksuotų vartotojo reikalavimų specifikuojamo dokumentacija.....	32
2.4 Pagrindinės gairės saugyklos vartotojo sąsajai.....	32
3 VEIKLOS TAISYKLĖMIS GRINDŽIAMOS REIKALAVIMŲ KURIAMAI INFORMACINEI SISTEMAI SPECIFIKACIJOS SAUGOJIMO SISTEMOS EKSPERIMENTINĖ REALIZACIJA	35
3.1 Projekto tikslas.....	35
3.2 Projekto apimtys	36
3.3 Duomenų bazės schema.....	36
3.3.1 Duomenų šaltinių posistemė	38
3.3.2 Funkcijų ir sprendimų posistemė.....	41
3.3.3 Konceptualaus duomenų modelio (struktūrinių veiklos taisyklių) posistemė.....	44
3.3.4 Veiklos taisyklių šablonų posistemė.....	50
3.3.5 Nestrukūrinė veiklos taisyklių posistemė.....	54
3.4 Realizacijos langų architektūra	59
3.5 Programuotojo vadovas	70
3.6 Eksperimentinės realizacijos komentarai.....	71
IŠVADOS IR PASIŪLYMAI.....	73
LITERATŪROS SARAŠAS.....	75
SANTRAUKA	76
PRIEDAI.....	77
1 Priedas: ISA Framework.....	78
2 Priedas: RuleSpeak šablonai anglų kalba	80
3 Priedas: Straipsnis.....	83

SANTRUMPŲ SĄRAŠAS

IT — informacinės technologijos.

IS — informacinė sistema.

DB — duomenų bazė.

DBVS — duomenų bazių valdymo sistema.

BRS — Business Rules Solutions.

VT — veiklos taisyklė.

ER — esybių ryšių diagrama.

PVM — pridėtinės vertės mokestis.

CRUD — create, read, update, delete.

PAVEIKSLŲ SĄRAŠAS

1 pav. Veiklos taisyklių darbo grupės siūlomas VT metamodelis.....	13
2 pav. Veiklos taisyklių specifikuojami žingsniai.....	16
3 pav. Funkcijos vykdymas darant veiklos sprendimus.....	17
4 pav. Ronald G. Ross siūlomas veiklos modelio kūrimo procesas.....	17
5 pav. VT grindžiamas poreikių specifikuojamas procesas.....	23
6 pav. Metamodelio fragmentas informacijai apie aktorius saugoti.....	24
7 pav. Metamodelio fragmentas funkcijų hierarchijai saugoti.....	25
8 pav. Metamodelio fragmentas conceptualaus duomenų modeliui saugoti.....	26
9 pav. Metamodelio fragmentas CRUD matricoms saugoti.....	27
10 pav. Metamodelio fragmentas VT šaltiniams saugoti.....	28
11 pav. Metamodelio fragmentas VT šablonams saugoti.....	29
12 pav. Metamodelio fragmentas VT saugoti.....	30
13 pav. Veiklos taisyklėmis grindžiamos reikalavimų kuriamai Informacinei Sistemai specifikacijos saugojimo sistemos metamodelis.....	31
14 pav. Vartotojo poreikių saugyklos DB schema.....	37
15 pav. DB ryšių schema Access platformoje.....	38
16 pav. Startinis saugyklos langas.....	60
17 pav. Funkcijų nustatymo veiklų diagrama.....	61
18 pav. Funkcijų hierarchijos duomenų įvedimo langas.....	62
19 pav. Sprendimų įvedimo langas.....	62
20 pav. Esybių ir jų sinonimų įvedimo langas.....	63
21 pav. Esybių atributų ir jų sinonimų įvedimo langas.....	64
22 pav. Reikšmių ir terminų įvedimo langas.....	64
23 pav. Ryšių įvedimo langas.....	65
24 pav. Asmenų įvedimo langas.....	66
25 pav. Pareigybių įvedimo langas.....	66
26 pav. Funkcijų susiejimo su aktoriais įvedimo langas.....	67
27 pav. Dokumentų įvedimo langas.....	67
28 pav. Programinių kodų įvedimo langas.....	68
29 pav. Veiklos taisyklių šablonų įvedimo langas.....	68
30 pav. Veiklos taisyklių šablonų elementų įvedimo langas.....	69
31 pav. Veiklos taisyklių įvedimo lango prototipas.....	70

LENTELIŲ SĄRAŠAS

1 lentelė VT klasifikacijų pavyzdžiai.....	11
2 lentelė Sprendimų lentelės pavyzdys	15
3 lentelė BRS RuleSpeak šablonai Lietuvių kalba.....	20
4 lentelė CRUD matricos pavyzdys.	27
5 lentelė DB lentelės Asmuo aprašas.	38
6 lentelė DB lentelės Pareigybe aprašas.....	39
7 lentelė DB lentelės Dokumentas aprašas.	40
8 lentelė DB lentelės PrograminisKodas aprašas.....	40
9 lentelė DB lentelės Funkcija aprašas.....	41
10 lentelė DB lentelės FunkcijosAtvaizdas aprašas.....	42
11 lentelė DB lentelės FunkcijaAktorius aprašas.....	43
12 lentelė DB lentelės Sprendimas aprašas.....	44
13 lentelė DB lentelės FunkcijaSprendimas aprašas.....	44
14 lentelė DB lentelės Esybe aprašas.....	45
15 lentelė DB lentelės Rysys aprašas.....	46
16 lentelė DB lentelės Atributas aprašas.....	47
17 lentelė DB lentelės ReiksmeTerminas aprašas.....	48
18 lentelė DB lentelės Sinonimas aprašas.....	49
19 lentelė DB lentelės CRUDSantykis aprašas.....	49
20 lentelė DB lentelės VTTipas aprašas.	50
21 lentelė DB lentelės Sablonas aprašas.	50
22 lentelė DB lentelės SblElementoTipas aprašas.	51
23 lentelė DB lentelės SubjektoTipas aprašas.....	52
24 lentelė DB lentelės SablonoElementas aprašas.....	52
25 lentelė DB lentelės GalimasElementoTipas aprašas.....	53
26 lentelė DB lentelės RezervuotasTekstas aprašas.....	53
27 lentelė DB lentelės VeiklosTaisykle aprašas.	54
28 lentelė DB lentelės VTIsraiskosKomponentas aprašas.....	55
29 lentelė DB lentelės SusijesModelioElementas aprašas.....	56
30 lentelė DB lentelės SprendimuLentele aprašas.	57
31 lentelė DB lentelės SLEilutesReiksme aprašas.....	57
32 lentelė DB lentelės SLStulpelioReiksme aprašas.....	58
33 lentelė DB lentelės SLReiksme aprašas.....	58
34 lentelė DB lentelės VTSaltinis aprašas.	59
35 lentelė DB lentelės SprendimoVT aprašas.....	59

ĮVADAS

Šiandieninis sparčiai besivystantis, konkurencijos sąlygomis veikiantis verslas reikalauja lanksčios programinės įrangos, kuri galėtų būti pritaikyta besikeičiantiems verslo reikalavimams. Draudimo bendrovės, finansinės institucijos ir kitos nuo veiklos politikos priklausančios organizacijos turi vystyti taikomąją programinę įrangą, kuri leistų lengvai modifikuoti ir įgyvendinti veiklos bei kontrolės politiką. Telekomunikacijų kompanijos savo santykių su klientais valdymo bei sąskaitų apdorojimo sistemas turi projektuoti taip, kad šios leistų dažnai pasiūlyti naujus produktus ar servigus. Taikomųjų programų serviso tiekėjai privalo turėti galimybę prisitaikyti prie kiekvieno individualaus kliento unikalių reikalavimų. Vis daugiau kompanijų ieškos sprendimo tokioms problemoms atsigręžia į veiklos taisykles.

Maždaug prieš 20 metų prasidėjo veiklos taisyklių požiūrio vystymasis. Veiklos taisyklių požiūris – tai formalus veiklos taisyklių formalizavimo bei valdymo būdas siekiant užtikrinti tokį veiklos elgesį ir vystymąsi, kokio siekia verslo vadovai. Remiantis šiuo požiūriu kuriamos informacinės sistemos, kurios ne tik leidžia veiklą vykdyti tiksliai pagal taisykles, bet ir jas keisti reaguojant į aplinkos ar vidaus pokyčius. Veiklos taisyklių požiūris leidžia automatizuoti verslo logiką. Per 15 teorijos vystymo metų mokslininkai išsamiai aprašė modelius ir informacinių sistemų, besiremiančių veiklos taisyklėmis, kūrimo procesą.

Viena iš problemų, su kuria susiduria informacinių sistemų kūrėjai, tai veiklos taisyklių atradimas. Tai procesas, kurio metu identifikuojamos veiklos taisyklės, įtakojančios kuriamą informacinę sistemą. Veiklos taisyklių atradimas gali trukti visą IS kūrimo laiką: daugelis jų atrandama specifikuojant vartotojų poreikius, taip pat projektuojant IS modelius. Derėtų nepamiršti, kad IS kūrimas – tai procesas, išsidėstęs laike, todėl tuo metu pati veikla, savaime aišku, nestovi vietoje ir gali generuoti naujas taisykles ar senų pakeitimus. Taigi taisyklės atrandamos viso proceso metu ir netgi po jo. Kurgi dėti, kaip dokumentuoti naujai atrastą taisyklę? Daugelis mokslininkų siūlo susikurti tam tikros formos blanką ir jame užrašyti veiklos taisyklės specifikaciją. Be abejo tai gali būti ir kompiuterinis failas.

Šio darbo objektas veiklos taisyklėmis grindžiamas vartotojo reikalavimų kuriamai informacinei sistemai specifikavimas.

Darbo tikslas yra pasiūlyti veiklos taisyklėmis grindžiamo vartotojo poreikių specifikavimo metodą ir CASE priemonę jam palengvinti.

Iškelti uždaviniai yra tokie:

- Apžvelgti veiklos taisyklių koncepciją;
- Aptarti veiklos taisyklių struktūrizavimo metodus;
- Apžvelgti veiklos taisyklių struktūrizavimo proceso eigą;

- Realizuoti pasirinktą veiklos taisyklėmis grindžiamos vartotojo reikalavimų saugyklos modelį;
- Pateikti išvadas ir pasiūlymus tolesniam sistemos vystymui.

Metodai, kuriuos ketinama iširti ir panaudoti darbe: mokslinėje literatūroje paskelbtų koncepcijų ir išvadų sisteminė analizė, palyginamoji ir apibendrinimo analizė, bendrieji mokslinio tyrimo metodai.

Laukiamas rezultatas veikianti vartotojo reikalavimų saugykla.

Teorinėje darbo dalyje pateikiama bendra VT koncepcijos analizė, aptariami ir palyginami svarbiausi žinomi VT struktūrizavimo metodai. Analitinėje darbo dalyje aprašomas vartotojo reikalavimų specifikavimo procesas ir siūlomas VT saugyklos modelis, būtinas proceso išsamumui užtikrinti. Trečiojoje, eksperimento, dalyje aprašomas pasiūlyto struktūrizavimo proceso modelio prototipo realizacija. Darbo pabaigoje pateikiamas pasiūlymo įvertinimas bei išvados. Papildoma reikšminga informacija bei autoriaus paskelbtas straipsnis pateikiami prieduose. Darbo apimtis: 70 puslapių neįskaitant literatūros sąrašo ir priedų, darbe pateiktos 35 lentelės ir 31 paveikslas.

1 Veiklos taisyklių metodo principai

1.1 Veiklos taisyklė

Veiklos taisyklių metodo tobulinimui dar 1993 metais buvo įkurta organizacija, šiuo metu žinoma kaip Veiklos taisyklių darbo grupė (Business Rules Group). Ši grupė jungia daugelį mokslininkų dirbančių veiklos taisyklių srityje, taip pat ir Ronald G. Ross (sukūrė specialią notaciją veiklos taisyklėms atvaizduoti duomenų modeliuose, tačiau modeliai gavosi pernelyg dideli, sudėtingi daugeliui žmonių, taip pat pasiūlė veiklos taisyklių šablonizavimo principus, kurie šiame darbe panaudoti ir Lietuvių kalba užrašomoms taisyklėms) bei Barbara von Halle (parašė vieną išsamiausių knygų apie veiklos taisyklių taikymą, tačiau ji linksta prie komercinių produktų, veiklos taisyklių saugyklų, naudojimo, paskelbė daug straipsnių aiškinančių veiklos taisyklių metodus) kurių darbais daugiausiai remtasi šiame baigiamajame magistro darbe.

Sistemų analitikai jau seniai praktikavo detalų organizacijos struktūros (išreikštos duomenų modeliais), bei organizacijos atliekamų funkcijų aprašymą. Tačiau dažnai linkstama prie apribojimų kurie veikia organizacijos veiklą ignoravimo. Apribojimai nebūdavo aiškiai išreikšiami iki pat momento, kai ateidavo laikas juos išreikšti programiniu kodu. Veiklos taisyklių darbo grupės tikslas paskatinti visų veiklos taisyklių dokumentavimą [6 Business Rules Group. 2000].

Ši grupė veiklos taisyklę apibrėžia kaip teiginį kuris apibrėžia arba apriboja, suvaržo kokį nors veiklos aspektą. Jis skirtas išreikšti veiklos struktūrą arba kontroliuoti ar įtakoti veiklos elgesį [6 Business Rules Group. 2000].

Šiuo metu veiklos taisyklių teoretikai labiau koncentruojasi ties informacinių sistemų kūrimą liečiančiomis veiklos taisyklėmis, taigi veiklos taisyklės reiškia apribojimus duomenų informacinėje sistemoje kūrimui, keitimui, bei naikinimui, trynimui [7 Business Rules Group. 2005].

1.2 Veiklos taisyklių metodo vystymosi prielaidos

Šiuolaikiniame verslo pasaulyje veiklos efektyvumo ir konkurencingumo užtikrinimui tenka imtis skirtingų veiklos keitimo projektų. Pagrindinės tokių projektų kategorijos pagal R. Ross'ą [8 Ross, R. 2003].:

- Reinžinerija — Veiklos procesų atnaujinimas ir automatizavimas. Veiklos taisyklės šiame procese gali vaidinti svarbų vaidmenį „nuleidžiant iš viršaus“ veiklos strategijas, bei apibūdinant veiklos procesų veikimo logiką.
- Atgaivinimas — jau egzistuojančių IT sistemų modifikavimas pasikeitus vidinėms arba išorinėms tvarkoms. Šis skausmingas procesas palengvėtų veiklos taisyklės iškėlus į atskirą sluoksnį.

- Perėjimas prie naujų technologinių platformų paskutiniu metu vis dažnėja, dėl didėjančios naujausių technologijų kaitos. Veiklos taisyklių metodas galėtų pagreitinti veiklos logikos perkėlimą iš vienos aplinkos į kitą.
- Žinių išsaugojimo problema aštrėja dėl darbuotojų kaitos, senų IT sistemų dokumentacijos netobulumo. Veiklos taisyklių metodo taikymas leistų dokumentuoti veiklos logiką, sumažinti riziką prarasti žinias.
- Veiklos valdymo galios perkėlimas iš IT į veiklos vykdytojų rankas. Turima omenyje IT projektų vykdymo trukmės sumažinimas, greitesnis veiklos logikos pokyčių įgyvendinimas. Veiklos taisyklių metodas leidžia veiklos taisykles administruoti patiems veiklos vykdytojams, o IT projektuose stipriai sutrumpėja arba išvis išnyksta vartotojų poreikių analizė.

Kaip matome veiklos procesai neatsiejami nuo IT procesų, tačiau ankstesni IT procesų kūrimo metodai įtvirtino atotrūkį tarp pačios veiklos specialistų ir IT specialistų. Dažnai susidarydavo situacija kai veiklą reguliuodavo IT (pvz.: stabdydama jos pokyčius). Veiklos taisyklių metodo taikymas padeda panaikinti šį atotrūkį [4 Hay, D. 2003].

ISA Framework kontekste (žr. 1 priedą) kalbant apie veiklos taisyklių modeliavimą informacinių sistemų inžinerijai apsiribojama tik duomenų ir motyvacijos stulpeliais. Taip pat apsiribojama trečiąja eilute kas reiškia, kad domimasi tik tomis taisyklėmis kurios apibūdina operacijas su duomenimis [6 Business Rules Group. 2000].

1.3 Bendri reikalavimai veiklos taisyklėms

Norint geriau suvokti VT esmę, būtina žinoti joms keliamus reikalavimus. Reikalavimų, užtikrinančių išskirtųjų taisyklių tinkamumą tolesniam apdorojimui, rinkinys paprastai yra pateikiamas visuose VT klasifikavimo modeliuose, tačiau yra išskiriami ir bendri reikalavimai taisyklėms, kurie nepriklauso nuo naudojamo modelio [11 Business Rules Solutions. 2002]. Taisyklės turi būti glaustos ir pateiktos tokia forma, kuri užtikrintų formalizavimo ir taisyklių saugyklos sudarymo proceso sklandumą [7 Business Rules Group. 2005]..

Nustatyta, jog registruojamoji taisyklė privalo būti [11 Business Rules Solutions. 2002]:

1. Deklaratyvi — VT nėra formuluojama procedūriškai – ji tiesiog deklaruojama kaip atskiras teiginys. Tiesa, realizuojant taisykles, jos gali būti aprašomos ir procedūriškai.
2. Tiksliai — VT privalo turėti tik vieną interpretaciją. Pastebėjus, jog taisyklė gali būti traktuojama keliais būdais, būtina pakeisti šios taisyklės formuluotę.
3. Nedaloma (logiškai nedaloma) — VT turi sudaryti ne daugiau negu viena baigta mintis. Taisyklė privalo būti nedaloma, t.y. skaidant taisyklę į dalis yra prarandama informacija.

4. Nuosekli — VT aibėje negali būti viena kitai prieštaraujančių ar kitaip nesuderintų taisyklių.
5. Neperteklinė — VT aibėje negali būti tą pačią informaciją konstatuojančių taisyklių.
6. Orientuota į veiklą — VT privalo būti formuluojama vartojant terminus, suprantamus konkrečios veiklos srities atstovams. Šie terminai turi būti prasmingi ir pripažinti tam tikros veiklos srityje.
7. Priklausanti veiklai — VT formuluoti, modifikuoti arba paskelbti negaliojančia gali tik konkrečios veiklos srities atstovai, o ne IS kūrėjai ar analitikai.

Taisyklė gali vartoti kitas taisykles. Sąvoka “atominė taisyklė” nereiškia, jog taisyklė negali vartoti kitų taisyklių. Detaliame ir nedviprasmiškame modelyje bus labai mažai VT, kurios nevirtotų kitų taisyklių (pvz.: taisyklę gali sudaryti keletas terminų ar keletas faktų) [8 Ross, R. 2003]..

1.4 Veiklos taisyklių klasifikacija

Toliau pateiktoje lentelėje matomos įvairios klasifikacijos sukurtos įvairių autorių. Tai tikrai ne galutinis sąrašas. Čia jį pateikiau tam, kad parodyti kokia požiūrių įvairovė egzistuoja.

1 lentelė

VT klasifikacijų pavyzdžiai

Šaltinis	Klasifikacija
GUIDE Veiklos taisyklių projektas	<ul style="list-style-type: none"> • Struktūrinis teiginys (terminas, faktas) • Veiksmo teiginys (sąlyga, apribojimas, leidimas) • Išvestis (matematinis skaičiavimas, loginė išvada)
Ronald Ross, Business Rule Solutions, Inc.	<ul style="list-style-type: none"> • Terminai • Faktai (sąryšių tipai, potipiai, atributai) • Taisyklės (apskaičiavimai, draudimai, projektoriai ir kt.)
Meta Data Coalition kartu su Microsoft	<ul style="list-style-type: none"> • Sąvoka • Faktas • Veiksmas • Draudimas
James Odell, nepriklausomas konsultantas	<ul style="list-style-type: none"> • Apribojimas: (stimulas/atsakas, operacija, sandara) • Išvestis: (išvada, apskaičiavimas)
Tom Romeo, IBM	<ul style="list-style-type: none"> • Struktūrinės (sąryšiai, domenai, kardinalumas, būtinumas) • Pagrįstos elgsena (pradinės sąlygos, galinės sąlygos, išvestis)
Margaret Thorpe, Tangram	<ul style="list-style-type: none"> • Apibrėžimai • Baziniai integralumo apribojimai • Bendri deklaratyvūs apribojimai • Procedūriniai apribojimai • Išvadų tipo • Išvestys
Barbara von Halle, Knowledge Partners	<ul style="list-style-type: none"> • Apibrėžimai • Faktai • Apribojimai • Išvestys • Išvados

Šaltinis	Klasifikacija
Usoft Corporation	<ul style="list-style-type: none"> • Suvaržymas (apribojimas, domenas) • Išvada • Pagrista elgsena • Atvaizdavimas
Dan Tasker, Air New Zealand	<ul style="list-style-type: none"> • Veiksmo suvaržymas • Veiksmo iškvietimas • Apribojimas
Brightware	<ul style="list-style-type: none"> • Veiklos reikalas • Politika (strategija) • Veiklos srautas • Sprendimo priėmimo tvarka
Vision Software	<ul style="list-style-type: none"> • Patikrinimas • Išvestis • Sąryšis • Sąlyginis veiksmas

Šaltinis: sudaryta autoriaus pagal [2 Kapočius K. 2005] ir [12].

Nors šios klasifikacijos iš pirmo žvilgsnio skiriasi viena nuo kitos, tačiau visos jos skirtos modeliuoti sritį — veiklą, kas reiškia, kad visose galima rasti atitikimų. Visose galima rasti kategorijas skirtas duomenų struktūrai, motyvacijai, veiklos procesams, rolėms ir t.t. išreikšti.

Be abejo, iš pateiktų klasifikacijų yra sukurti ir metamodeliai. Kaip pavyzdį pateikiu veiklos taisyklių darbo grupės siūlomą modelį.

- Terminas (terms). Bazinis veiklos taisyklių elementas yra kalba naudojama joms išreikšti. Pats termino apibrėžimas yra veiklos taisyklė kuri apibrėžia tai kaip žmonės galvoja ir kalba apie daiktus. Taigi žodžių reikšmių apibrėžimas (terminai) yra atskira veiklos taisyklių kategorija.
- Faktas (facts). Organizacijos struktūra gali būti apibūdinta kaip faktai kurie sieja terminus vieną su kitu. Pavyzdžiui, teiginys, kad klientas gali pateikti užsakymą yra veiklos taisyklė.
- Apribojimai (constraints). Kiekvienoje įmonėje elgesys yra kaip nors ribojamas, tai glaudžiai siejasi su apribojimais duomenų įvedimui (kas gali ir kas negali įvedinėti duomenis). Draudimas įvesti įrašą yra tas pat kaip draudimas kokiam nors veiksmui įvykti.
- Išvedimo taisyklės (derivations). Veiklos taisyklės nustato kaip vienos rūšies žinios gali būti transformuojamos į kitos rūšies žinias. Išvestas yra atributas kuris gaunamas iš kitų atributų ar sisteminių kintamųjų.

Pažymėtina, kad terminas tai konkreti, vienintelė žodžiui ar frazei priskirta reikšmė. Taigi terminai panaikina tam tikras problemas iš pokalbių. Žodžiai ir frazės gali turėti kelias reikšmes kurios priklauso nuo konteksto. Taigi veiklos taisyklė “terminas” yra vienas žodis (arba frazė) turintis vieną ir tik vieną reikšmę. Jei žodis turi kelias reikšmes ir jos reikalingos, kuriami keli terminai [4 Hay, D. 2003].

Su terminais susiduriama tokiomis formomis [4 Hay, D. 2003]:

- Esybė. Įmonei reikšmingas dalykas apie kurį saugoma informacija.
- Atributas (jo pavadinimas). Apibūdina esybes.
- Kiti veiklos terminai. Terminai apibūdinantys kokį nors gyvenimo aspektą , tačiau kurių formalizuoti nebūtina.

Terminus galima atvaizduoti paprastomis Esybių ryšių (ER) diagramomis.

Faktas tai kažko atributas naudojamas apibūdinimui: atliekama rolė ar koks kitas deskriptorius. Duomenų modelyje gali būti pavaizduoti trijų rūšių faktai [4 Hay, D. 2003]:

- Ryšiai (rolės). A tyri ryšį su (atlieka rolę) B.
- Atributai. A yra B atributas
- Supertipai/subtipai. Jei yra A tai taip pat yra ir B.

Faktus galima atvaizduoti kartu su terminais esybių ryšių (ER) diagramose.

Apribojimų ir išvedimo taisyklės yra sudėtingesnės. Buvo bandymų sukurti notacijas grafiškai vaizduojančias šias taisyklių rūšis, tačiau jie gavosi labai sudėtingi ir šioje srityje neišsilavinusiam žmogui lengvai neperprantami. Siūloma izeiti naudoti kokią nors struktūrinę kalbą jiems aprašyti.

Ronald G. Ross siūlo naudoti kompanijos Business Rule Solutions sukurtą veiklos taisyklių aprašymo metodiką RuleSpeak. Ši metodika leidžia [8 Ross, R. 2003].:

- Veiklos taisykles išreikšti aiškiai, nedviprasmiška ir gerai struktūrizuota anglų kalba.
- Išryškinti veiklos logiką.
- Pagerinti komunikaciją tarp verslo ir IT.
- Susieti veiklos analizę su sistemų kūrimu.

Taigi leidžia išvengti pagrindinių vartotojų poreikių specifikuojimo problemų.

RuleSpeak — tai sprendimas apibrėžiantis kaip reikia išreikšti veiklos taisykles aiškiai ir nedviprasmiškai. Dažniausiai šia kalba naudosis IT darbuotojai, tačiau struktūrizuota anglų kalba leidžia taisykles suprasti (bei “išstarti”) ir likusiems veiklos nariams [8 Ross, R. 2003].

2 priede pateikiu Ronald G. Ross sukurtus šablonus. Jie skirti anglų kalbai ir jų gali būti daugiau. Taip pat analogiškus šablonus galima sukurti ir kitoms kalboms, pavyzdžiui, lietuvių [8 Ross, R. 2003].

RuleSpeak taip pat numatyta galimybė naudoti sprendimų lenteles. Tokios lentelės leidžia vienodą paskirtį ir struktūrą turinčių taisyklių rinkinį atvaizduoti paprasčiau [8 Ross, R. 2003]. Lengviausia tai paaiškinti pavyzdžiu. Taigi specifikuodami veiklos taisykles galime susidurti su tokiu jų rinkiniu :

- PVM tarifas turi būti 20% jei metai 1999
- PVM tarifas turi būti 18% jei metai 2000
- PVM tarifas turi būti 17,5% jei metai 2001
- PVM tarifas turi būti 16% jei metai 2002
- PVM tarifas turi būti 15,75% jei metai 2003

Matome, kad visos šios taisyklės rodo koks turi būti PVM tarifas skirtingais metais. Pažymėtina, kad šių taisyklių neįmanoma atvaizduoti formule. Štai kaip toks taisyklių rinkinys atrodys sprendimų lentelėje:

- PVM tarifas turi būti lygus procentiniam dydžiui iš 2 lentelės atitinkančiam duotus metus.

2 lentelė

Sprendimų lentelės pavyzdys

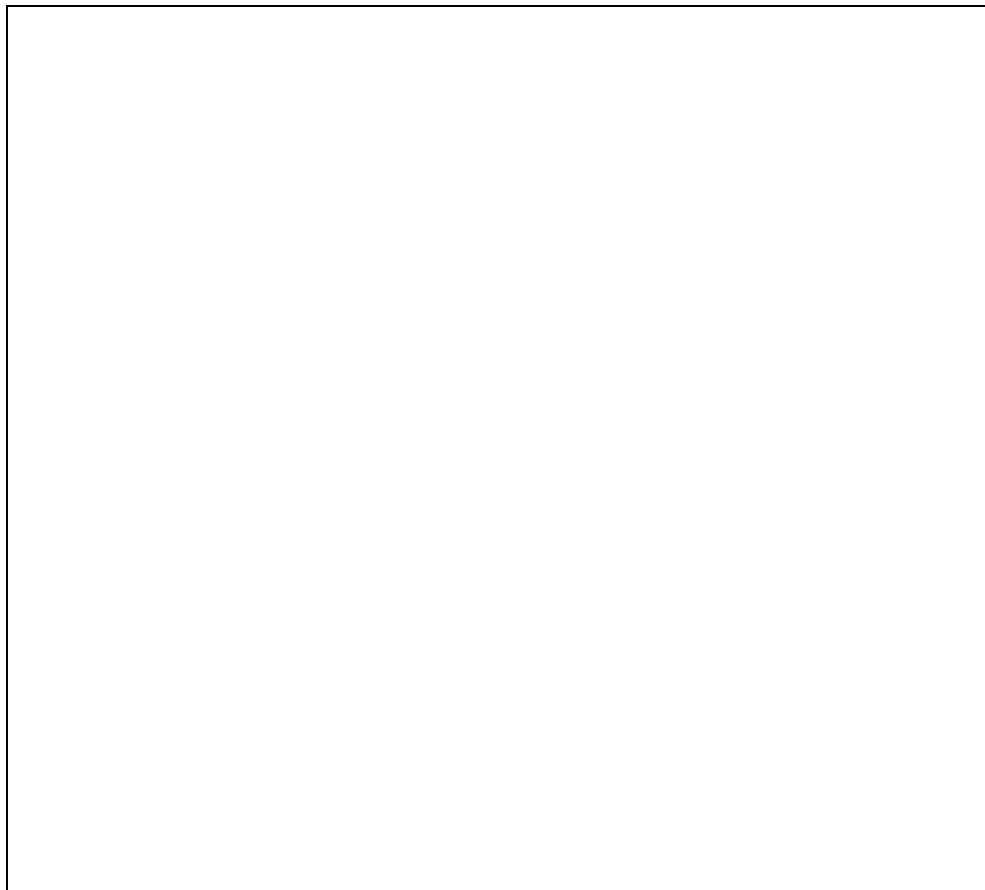
Metai	PVM tarifas
1999	20%
2000	18%
2001	17,5%
2002	16%
2003	15,75%

Šaltinis: sudaryta autoriaus.

Taigi vietoje penkių taisyklių turime vieną sprendimų lentelę.

1.5 Veiklos taisyklių specifikuojimo procesas

Barbara von Halle savo knygoje (literatūros sąrašė nr. 12) išskiria tokius veiklos taisyklių specifikuojimo proceso žingsnius:



Šaltinis: [12 Von Halle, B. 2002]

2 pav. Veiklos taisyklių specifikuojimo žingsniai.

Šio proceso išeiga yra [12 Von Halle, B. 2002]:

- Veiklos taisyklės išreikštos verslo kalba ir kategorizuotos pagal esybes arba veiklos objektus.
- Veiklos taisyklių saugykla.
- Veiklos taisyklių išgavimo iš programinio kodo metodai.
- Prioritetizuoti veiklos taisyklių šaltiniai.
- Veiklos taisyklių tvarkymo galimybė.
- Duomenų failų/lentelių ataskaitos.

Kaip matome, duomenų modelio ir veiklos taisyklių specifikuojimas vykdoma kartu. Tai yra visiškai natūralu, nes iš esmės duomenų modelio elementai (esybės, atributai, ryšiai) priskiriami prie veiklos taisyklių.

Taip pat norėčiau parodyti, kaip ši mokslininkė siūlo sieti veiklos funkcijas (veiklos procesus įvykius) ir veiklos taisykles per sprendimus [12 Von Halle, B. 2002]:



Šaltinis: sukurta autoriaus pagal [12 Von Halle, B. 2002].

3 pav. Funkcijos vykdymas darant veiklos sprendimus.

Ronald'o G. Ross'o siūlomas veiklos modelio kūrimo procesas matomas paveiksle.



Šaltinis: sukurta autoriaus pagal [9Business Rules solutions. 2005]

4 pav. Ronald G. Ross siūlomas veiklos modelio kūrimo procesas.

Pirmajame žingsnyje apibrėžiama veiklos sritis. Aiškiai apibrėžiama veiklos misija bei galutinis efektas arba tikslai kurių veikla siekia. Nustatomi veiklos įvykiai, kuriems reikalingi darbų sekų modeliai. Nustatomi pagrindiniai aktoriai iš išorės veikiantys veiklą. Veikla skirta kurti

pridėtinę vertę jiems. Taip pat nustatomi pagrindinės veiklos koncepcijos, terminai. Šie terminai bus pagrindas veiklos taisyklėms ir veiklos proceso reinžinerijai. Nustatomos pagrindinės pridėtinę vertę kuriančios transformacijos (kitai vadinamos funkcijomis). Sukuriamas fizinių ir virtualių lokacijų, kur bus atliekamas darbas, sąrašas. Tai vėliau leis atlikti jų optimizavimo ir ryšių analizę [9 Business Rules solutions. 2005].

Antrajame žingsnyje suformuojama veiklos taktika. Ruošiantis veiklos taktikos užfiksavimui patartina įvertinti jau egzistuojančią dokumentaciją, apklausti pagrindinius žaidėjus. Tai padės geriau suprasti kontekstą kuriame sprendimas bus kuriamas. Reikia peržvelgti verslo problemas veikiančias kompaniją. Taip pat reikia peržiūrėti veiklos kryptis ateityje ir tai kaip jos siejasi su kuriu sprendimu. Peržvelgiamos ir pagrindinės jau egzistuojančios tvarkos, bei įvertinamas jų efektyvumas. Tada parengiama pradinė aukšto lygio taktika — veikimo būdai kuriuos naudos veikla, kai bus įsisavintas sprendimas. Po to nustatomi rimčiausi barjerai, rizikos aukšto lygio taktikos veikimui. Kiekvienam identifikuotam rizikos šaltiniui valdyti numatoma žemesnio lygio taktika arba taisyklės. Tada analizuojamos rizikos šioms taktikoms ir t.t. kol identifikuojami visi rizikos šaltiniai bei parengiamas rizikos valdymo planas susidedantis iš taisyklių bei taktikos aprašymų [9 Business Rules solutions. 2005].

Trečiame žingsnyje formuojama darbų seka (workflow). Iš pradžių dokumentuojamos jau veikiančios darbų sekos, identifikuojamos užduotys, aktoriai, įvykiai. Tai leidžia išanalizuoti kontekstą, kuriame darbų sekos modelis bus kuriamas. Surandamos problemos, analizuojami veikimo principai, rolės bei atsakomybė, o taip pat ir galimybės pagerinti, pertvarkyti, racionalizuoti ir pan. Tada sudaromas darbų sekos modelis kuris bus diegiamas. Sudarant šį modelį stengiamasi išspręsti problemas rastas jau veikiančiuose procesuose. Apibrėžiami darbų sekų modelio darbai įskaitant įeigą, transformacijas ir išeigą. Detaliai apibūdinama, kaip atliekamas darbas. Toliau reikia peržiūrėti naująjį modelį ir rasti aktorius kurių rolės ir atsakomybė aiškiai atitinka modelio užduotis. Tai leis paskirstyti užduotis ir pareigas naujajame modelyje. Išplėskite naująjį modelį atsižvelgdami į scenarijų įvairovę, ypač į tuos kurie apibūdina išimtines sąlygas. Tokiu būdu užtikrinamas modelio išbaigtumas. Gerai paruoštas darbų sekų modelis sudaro pagrindą fundamentalių taisyklių reguliuojančių veiklos procesą apibrėžimui. Modelio ir užduočių apibrėžimų analizė leidžia atrasti ir iširti šias taisykle. Peržiūrima aktorių rolės ir atsakomybė naujo modelio ir taisyklių kontekste. Nustatoma kaip darbus paskirstyti ir organizuoti efektyviau. Taip pat reikia patikrinti kaip šios perskirstytos rolės ir atsakomybės paveiks organizaciją [9 Business Rules solutions. 2005].

Ketvirtajame žingsnyje suformuojama terminų ir faktų aibė. Surandami ir dokumentuojami terminai ir jų apibrėžimai iš industrijos, korporacijos žodynų, projektų dokumentacijos, instrukcijų, tvarkų, saugyklų, ir kitų prieinamų šaltinių. Įvertinamas rastų terminų ir faktų aibės išbaigtumas bei

kokybė, pasistengiama užpildyti rastas spragas. Peržiūrimi egzistuojantys faktų ir terminų modeliai. Juos galima rasti ankstesnių projektų dokumentacijoje, arba egzistuojančių duomenų bazių struktūrose. Taip apibrėžiamas naujo faktų modelio kontekstas, randami egzistuojančio modelio netobulumai, problemos, ir tokiu būdu identifikuojamos tobulintinos modelio vietos. Reikia atrinkti ir išanalizuoti realius terminus, tai yra tokius kurie reiškia žmones, vietas ar daiktus. Išskirti jų kategorijas, bendrumus, persidengimus, parinkti tinkamus pavadinimus ir įvesti apibrėžimus. Taip pat reikia atrinkti ir išanalizuoti abstrakčius terminus, tai yra tokius kurie reiškia koncepcijas ar įvykius. Juos taip pat suskirstyti į kategorijas, sugrupuoti, permastyti ką jie reiškia, parinkti tinkamus pavadinimus ir įsivesti apibrėžimus. Tada jau galima sukurti naują faktų modelį su terminais ir faktais bei kur reikia kategorijomis ir atributais. Tai leidžia susidaryti bendrą vaizdą apie sistemos veikimą, bei sudarys pagrindą visam veiklos procesui. Kiekvienam terminui reikia sukurti į verslą orientuotą apibrėžimą ir organizuoti apibrėžimus koncepcijų kataloge. Šiame žingsnyje galima patikrinti faktų modelio teisingumą ir išbaigtumą sugretinant koncepcijas ir faktus su ankstesnių žingsnių išeiğa. Taip pat reikia peržiūrėti užfiksuotas iki šiol taisykles ir patikrinti ar visa jose esanti terminologija yra išreikšta faktų modelyje. Remiantis faktų modeliu galima suformuluoti ir pagrindines veiklą reguliuojančias taisykles [9 Business Rules solutions. 2005].

Penktajame žingsnyje nustatomos veiklos vykdymo gairės. Iš pradžių peržiūrimas faktų modelis ir identifikuojami terminai tinkami gairių analizei. Tai paprastai būna abstraktūs terminai padedantys koordinuoti darbą tarp užduočių ir darbų sekų bei išsidėstymą laike. Nustatomos pasirinkto termino būsenos arba gairės rodančios normalų arba norimą susijusio darbo režimą, ir joms suteikiami pavadinimai bei apibrėžimai. Nustatomos pasirinkto termino nenormalios būsenos atspindinčios nepageidautinas, nenorimas arba ypatingas sąlygas. Šioms gairėms taip pat suteikiami pavadinimai ir apibrėžimai. Nustatomos pagrindinės taisyklės koordinuojančios termino gairių režimą. Šios taisyklės gali nustatyti kada pasikeitimai reikalingi arba neleidžiami. Taip pat nustatomos taisyklės koordinuojančios termino būsenas skirtingos užduotyse ir darbų sekose [10 Business Rules solutions. 2001].

Šeštajame žingsnyje analizuojamos veiklos taisyklės. Įvertinamas taisyklių atomiškumas, tai yra ar taisyklė apibūdina vieną veiklos loginį elementą. Jei ne taisyklę reikia skaidyti. Specifikuojama informacija apibūdinanti taisykles: taisyklės kategorizuojamos, nustatomi ryšiai tarp pačių taisyklių ir tarp taisyklių bei kitų veiklos modelio elementų. Permaštoma kiekviena taisyklė, jos prielaidos, ar terminai ir aktoriai optimaliai jas atitinka. Peržiūrimas taisyklių rinkinys ieškant pasikartojimų, identiškumų, semantinių atitikimų ar konfliktų, bei išsprendžiamos šios problemos. Taip pat ieškomos taisyklės išimtys. Analizuojami „kas būtų, jeigu“ scenarijai, taip pat įvykiai kuriems taisyklė taikoma. Priklausomai nuo rastų išimčių pobūdžio taisyklė pataisoma arba sukuriama naujos taisyklės išimčių valdymui [10 Business Rules solutions. 2001].

2 Veiklos taisyklėmis grindžiamas vartotojo poreikių specifikuavimas

2.1 Veiklos taisyklių struktūrizavimo modelis

Remiantis GUIDE projektu veiklos taisyklės gali būti skirstomos į struktūrines ir nestruktūrines. Struktūrinės veiklos taisyklės tai terminai ir faktai aprašantys ryšius tarp terminų. Iš esmės struktūrinėmis taisyklėmis aprašomas duomenų modelis. Kadangi vartotojo reikalavimų specifikuavimo procese detalus duomenų modelis dar neprivalo būti suformuotas, tai KTU informacijos Sistemų katedros siūlomame metode apsiribojama tik konceptualaus duomenų modelio formavimu [2 Kapočius K. 2005].

Nestruktūrinėms veiklos taisyklėms fiksuoti pasirinktas natūralios kalbos sakinių šablonais paremtas Business Rules Solutions RuleSpeak (BRS RuleSpeak) metodas. Pagal šį metodą terminai saugomi Esybių-Ryšiu modelyje. BRS RuleSpeak taisyklės suskirstytos į funkcines kategorijas. Kiekvienos kategorijos taisyklės užrašyti naudojama po vieną ar keletą šablonų. Šablonų pritaikytų lietuvių kalbai pavyzdžius matote 3 lentelėje [1 Kapočius, K. 2006].

3 lentelė

BRS RuleSpeak šablonai Lietuvių kalba

Kategorija	Neformalus apibrėžimas	Galimi subjekto tipai	Šablonas	VT pavyzdys
1.1 Atmetimo taisyklė/atmetimas/apribojimo taisyklė (<i>rejector</i>)	Apribojimas, kurio nepažeidžiant užtikrinamas duomenų teisingumas (neprieštaringumas)	Terminas, faktas, duomenų elementas	<i>b1.</i> <Subjektas> PRIVALO/NEPRIVALO/TU RĖTŲ/NETURĖTŲ <faktas> [jeigu/kol <sąlyga>].	Užsakymas privalo turėti pristatymo datą. Studentas neprivalo dalyvauti daugiau nei keturiuose kursuose kol stažuojasi.
			<i>b2.</i> <Subjektas> gali/turėtų <faktas> TIK jeigu/kol <sąlyga>.	Klientas gali pateikti užsakymą TIK jeigu klientas turi atsidaręs sąskaitą.
			<i>b3.</i> <Subjektas> gali/turėtų <faktas> TIK <sąlyga>	Fiksuotą atlyginimą gaunantis darbuotojas gali dirbti TIK biudžetiniame skyriuje.
1.2 Leidimo taisyklė (<i>permission statement</i>)	Strateginė nuostata ar paaiškinimas, leidžiantis vykdyti veiklą.	Terminas, faktas, taisyklė, procesas (funkcija???) , duomenų elementas	<i>b4.</i> <Subjektas> GALI <faktas/VT raktažodis> [jeigu/kol <sąlyga>].	Užsakymas, kurio suma yra \$1000 ar mažiau, gali būti priimtas iš kliento net jeigu kliento kredito likutis nepatikrintas.
			<i>b5.</i> <Subjektas> NETURI <faktas/VT raktažodis> [jeigu/kol <sąlyga>].	Automobilis neturi būti apdraustas nuo vagystės.

Kategorija	Neformalus apibrėžimas	Galimi subjekto tipai	Šablonas	VT pavyzdys
2.1 Skaičiavimo taisyklė (<i>computation rule</i>)	Teiginys ar aritmetinė formulė, nurodanti, kaip apskaičiuoti tam tikrą skaitmeninę reikšmę.	Apskaičiuojamas terminas (reikšmė), duomenų elementas	<i>b6.</i> <Subjektas> turi/neturi/turėtų/neturėtų BŪTI APSKAIČIUOJAMAS/-a/-i/-os kaip <matematinė formulė> [jeigu/kol <sąlyga>].	Apmokėjimo už užsakymą suma apskaičiuojama kaip visų užsakymui priklausančių dalių apmokėjimų suma.
			<i>b7. Sutrumpintas variantas:</i> <Subjektas> = <matematinė formulė> [jeigu/kol <sąlyga>].	Apmokėjimo už užsakymą suma = visų užsakymo dalių apmokėjimų suma.
2.2 Išvedimo taisyklė (<i>derivation rule</i>)	Teiginys ar loginė išraiška, nusakanti, kaip nustatyti taip/ne pobūdžio rezultatą.	Išvestinis terminas (atributas, esybė, reikšmė), duomenų elementas	<i>b8.</i> <Subjektas> turi/neturi/turėtų/neturėtų REIŠTI, KAD <loginė išraiška> [jeigu/kol <sąlyga>].	Stambi prekė turi reikšti, kad prekės kaina viršija \$500.
			<i>b9. Sutrumpintas variantas:</i> <Subjektas> REIŠKIA, KAD/NEREIŠKIA, KAD <loginė išraiška> [jeigu/kol <sąlyga>].	Stambi prekė reiškia, kad prekės kaina viršija \$500.
3.1.1 Išvados taisyklė (<i>inference rule</i>)	Taisyklė, pagal kurią, atsižvelgiant į tam tikras aplinkybes, padaroma išvada.	Terminas (reikšmė, atributas, esybė), duomenų elementas	<i>b10.</i> <Subjektas> turi/neturi/turėtų/neturėtų BŪTI LAIKOMAS/-a/-i/-os <terminas> [jeigu/kol <sąlyga>].	Asmuo turėtų būti laikomas moterimi jeigu asmens lytis yra moteris ir asmens amžius yra virš 21 metų.
			<i>b11. Sutrumpintas variantas:</i> <Subjektas> YRA/NĖRA <terminas> [jeigu/kol <sąlyga>].	Asmuo yra moteris jei asmens lytis yra moteris ir asmens amžius yra virš 21 metų.
3.1.2 Taisyklės jungiklis (<i>rule toggle</i>)	Taisyklė, "įjungianti" ar "išjungianti" kitą taisyklę, priklausomai nuo konkrečių aplinkybių. Naudojama taisyklių išimtims nusakyti.	Taisyklė	<i>b12. Neformalus variantas:</i> <Taisyklės teiginys> NEBENT/IŠSKYRUS <sąlyga>.	Bibliotekos kortelė gali priklausyti daugiausiai vienam skolintojui, nebent vienas iš skolintojų, kuriems priklauso bibliotekos kortelė, yra Bilas Geitsas.
			<i>b13. Formalus variantas:</i> <Taisyklės pavadinimas/kodas> turi/neturi/turėtų/neturėtų BŪTI TAIKOMA [jeigu/kol <sąlyga>].	Taisyklė-kortelė-priklauso-tik-vienam-skolintojui neturi būti taikoma, jeigu vienas iš skolintojų kuriems priklauso bibliotekos kortelė yra Bilas Geitsas.
3.1.3 Proceso jungiklis (<i>process toggle</i>)	Taisyklė, esant tam tikroms aplinkybėms "įjungianti" ar "išjungianti" procesą.	Procesas, procedūra	<i>b14.</i> <Subjektas> turi/neturi/turėtų/neturėtų BŪTI ĮGALINTAS / BŪTI UŽDRAUSTAS [jeigu/kol <sąlyga>].	Įspėjimo-dėl-susitikimo-siuntimas turi būti uždraustas, jeigu kliento adresas nežinomas.

Kategorija	Neformalus apibrėžimas	Galimi subjekto tipai	Šablonas	VT pavyzdys
3.1.4 Duomenų jungiklis (<i>data toggle</i>)	Taisyklė, esant tam tikroms aplinkybėms šalinanti (ar sukurianti atsitiktinius) duomenis.	Duomenų elementas.	<i>b15.</i> <Duomenų elementas> turi/neturi/turėtų/neturėtų BŪTI SUKURTAS/IŠTRINTAS jeigu/kol < sąlyga>.	Kiekvienas nenagrinėtas bylos klausimas turi būti ištrintas, jeigu byla uždaroma. Loterijos-laimintys-skaičiai turi būti sukurti, jeigu loterijos-data lygi šiandienai.
3.2.1 Reikšmės priskyrimo taisyklė (<i>imprint rule</i>)	Taisyklė, priskirianti saugomam duomenų elementui tam tikrą reikšmę.	Terminas (reikšmė/atributas/esybė), faktas, duomenų elementas	<i>b16.</i> <Subjektas> turi/neturi/turėtų/neturėtų [GYTI REIKŠMĘ <terminas/reikšmė> [kai/jeigu < sąlyga>].	Studento-semestro-skola turi įgyti reikšmę \$3065, kai studentas registruojamas semestru. Normali-mokesčių-deklaracijos-apmokėjimo-data turi įgyti reikšmę Balandžio 15. Automobilio-draudimo-mokestis turi įgyti reikšmę iš sprendimų lentelės T1 pagal automobilio modelį, pagaminimo metus ir kilometražą.
3.2.2 Pateikimo (atvaizdavimo) taisyklė (<i>presentation rule</i>)	Taisyklė, kuria nustatoma duomenų pateikimo forma (ekrane, ataskaitoje ir pan.).	Terminas (atributas, esybė, reikšmė), faktas, duomenų elementas	<i>b17.</i> <Subjektas> turi/neturi/turėtų/neturėtų BŪTI PATEIKIAMAS/-a/-i/-os [<vaizdavimo terpė>] <vaizdavimo būdas> [jeigu/kol < sąlyga>].	Užsakymas turi būti pateikiamas ekrane raudonai jei užsakymas laiku neatliktas. Potencialūs tiekėjai turi būti pateikiami Potencialių-tiekėjų-ataskaitoje išrikiuoti pagal abėcėlę.
3.3.1 Proceso trigeris (<i>process trigger</i>)	Taisyklė, pagal kurią, susidarius tam tikroms aplinkybėms, automatiškai vykdomas procesas ar procedūra.	Procesas, procedūra.	<i>b18.</i> <Subjektas> turi/neturi/turėtų/neturėtų BŪTI VYKDOMAS/-a , kai < sąlyga>.	Siųsti-išankstinį-įspėjimą užsakymui turi būti vykdomas, kai užsakymas išsiųstas.
3.3.2 Taisyklės trigeris (<i>rule trigger</i>)	Taisyklė, pagal kurią, susidarius tam tikroms aplinkybėms, automatiškai paleidžiama kita taisyklė.	Taisyklė	<i>b19.</i> <Taisyklės pavadinimas/kodas> turi/neturi/turėtų/neturėtų BŪTI PALEISTA , kai < sąlyga>.	Taisyklė-Numatoma-pristatymo-data turi būti paleista, kai krovinyš išvedamas į ekraną.

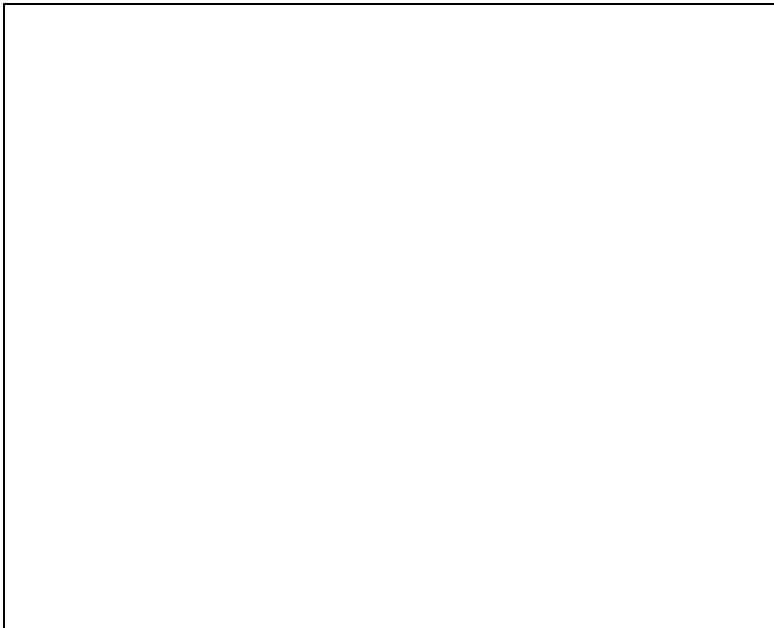
Šaltinis: sudaryta autoriaus pagal [1 Kapočius, K. 2006] ir [8 Ross, R. 2003]..

Visuose šablonuose pirmasis elementas yra subjektas, kuris priklausomai nuo kategorijos gali būti terminas, faktas, konkreti reikšmė, kita taisyklė ir t.t. Raktažodžiais paremta šablono struktūra leidžia taisyklės išskaidyti ir saugoti specialioje saugykloje [13].

2.2 Vartotojo reikalavimų specifavimo procesas

Nagrinėjant ankstesniame skyriuje paminėtų autorių medžiagą Kauno Technologijos universiteto Informacijos sistemų katedroje buvo sukurta veiklos taisyklėmis grindžiama poreikių kuriamai informacinei sistemai specifavimo koncepcija. Poreikių specifavimo proceso schemą

matote 5 paveiksle. Procesas prasideda apibrėžiant veiklos kontekstą, bei aktorių, kaip būsimos sistemos vartotojų arba poreikių informacijos šaltinio, registravimas. Toliau specifikuojamos aktorių atliekamos funkcijos bei jų atlikimo sekos. Taip pat registruojami kiekvienos funkcijos atlikimo metu priimami sprendimai bei jų eiliškumas [1 Kapočius, K. 2006].



Šaltinis: [2 Kapočius K. 2005]

5 pav. VT grindžiamas poreikių specifavimo procesas.

Lygiagrečiai vyksta conceptualaus duomenų modelio formavimas, tai yra fiksuojamos duomenų esybės jų atributai bei ryšiai tarp esybių (struktūrinės veiklos taisyklės: terminai ir faktai). Svarbi reikalavimų specifikacijos dalis yra CRUD matricos formavimas, tai yra užfiksavimas kokios funkcijos naudoja kokius terminus ir koks to panaudojimo pobūdis: kopijuojama, skaitoma [13].

Pasiekus nestruktūrinių veiklos taisyklių fiksavimo fazę. Kiekvienai veiklos taisyklei yra fiksuojamas šaltinis: asmuo, dokumentas arba programinis kodas.

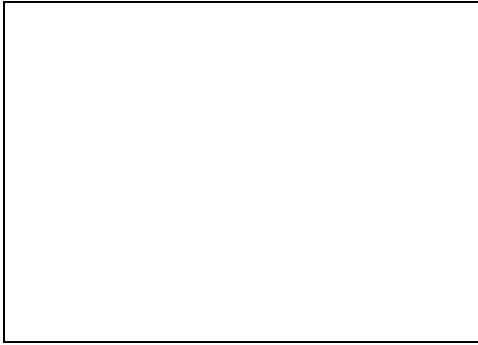
Sudarius veiklos taisyklių rinkinį jis tikrinamas. Pradžioje automatinėmis priemonėmis, vėliau verifikuojamas būsimų IS vartotojų. Kiekviename žingsnyje fiksuojama informacija išsaugoma vartotojo reikalavimų saugykloje [13].

Toliau kiekvienas žingsnis aprašomas detaliau.

2.2.1 Veiklos konteksto apibrėžimas ir aktorių nustatymas

Į veiklos konteksto apibrėžimą įeina organizacijos strategijų, tikslų, tvarkų, misijų dokumentacija. Ši informacija kol kas nefiksuojama siūlomoje saugykloje, todėl šios informacijos surinkimo proceso taip pat nedetalizuosiu.

Į aktorių nustatymą įeina organizacijos personalo duomenų surinkimas. Čia įeina ne vien vadovaujantis personalas, bet ir žemesnio rango darbuotojai, kurie gali būti naudingi specifikuojant poreikius ir/arba yra potencialūs sistemos vartotojai ateityje [1 Kapočius, K. 2006]. Taip pat turėtų būti fiksuojama ir pareigybinė struktūra (pavaldumo ryšiai). Patartina surinkti ir vadovybės kontaktinę informaciją.



Šaltinis: sukurta autoriaus.

6 pav. Metamodelio fragmentas informacijai apie aktorius saugoti

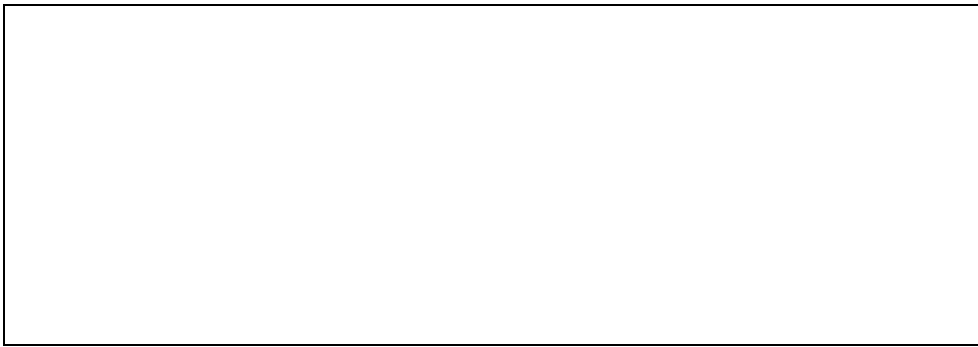
6 paveiksle matome metamodelio fragmentą kuriame saugoma informacija apie aktorius. Lentelėje Pareigybe saugoma hierarchinė struktūra (egzistuoja ryšys su pačia savimi), o lentelėje Asmuo saugomi asmens duomenys.

2.2.2 Funkcijų ir funkcijų sekų atskleidimas

Funkcijų atskleidimo proceso rezultatas yra Funkcijų hierarchijos diagramos. Laikoma, kad atskleisti reikalinga tik kompiuterizuojamas funkcijas. Keletas pastabų funkcijų hierarchijos sudarymui [1 Kapočius, K. 2006]:

- Iš pradžių fiksuojamos aukšto lygmens, o vėliau žemesnio lygmens funkcijos. Sudėtingos funkcijos turi būti pateikiamos atskirose diagramose.
- Funkcijos susiejamos su organizacijos asmenimis arba pareigybėmis jas atliekančiais.
- Naudinga priimti taisyklę, kad už funkcijos atlikimą atsakingi aktoriai taip pat atsakingi ir už žemesnio lygio funkcijų arba kitaip subfunkcijų atlikimą. Todėl aktoriai turėtų būti priskiriami kuo aukštesniame funkcijų hierarchijos lygyje.
- Funkcijų hierarchijos diagramos turėtų būti tikrinamos atitinkamų organizacijos atstovų.

Funkcijos turėtų būti saugomos vartotojo reikalavimų specifikacijos saugykloje. 7 paveiksle matomas atitinkamas saugyklos metamodelio fragmentas.



Šaltinis: sukurta autoriaus.

7 pav. Metamodelio fragmentas funkcijų hierarchijai saugoti.

Funkcijų aprašai saugomi lentelėje `Funkcija`, o funkcijų hierarchija saugoma atskiroje lentelėje su rekursiniu ryšiu į save pačią — `FunkcijosAtvaizdas`. Pastebėtina, kad pasirinkta struktūra leidžia konkrečias funkcijas „pakabinti“ keliose funkcijų hierarchijos vietose. Kiekvienas funkcijos atvaizdas/kopija yra laikomas atskira funkcija. Lentelėje `FunkcijosAtvaizdas` saugoma ir jų vykdymo seka. Tai reikalinga tam, kad išsaugotumėme tėvinės funkcijos atlikimo tvarką, o tai savo ruožtu turės tiesioginę įtaką su jomis susijusių veiklos taisyklių taikymo sekai. Su aktoriais (asmenimis arba pareigybėmis) funkcijų atvaizdai susiejami per lentelę `FunkcijaAktorius`.

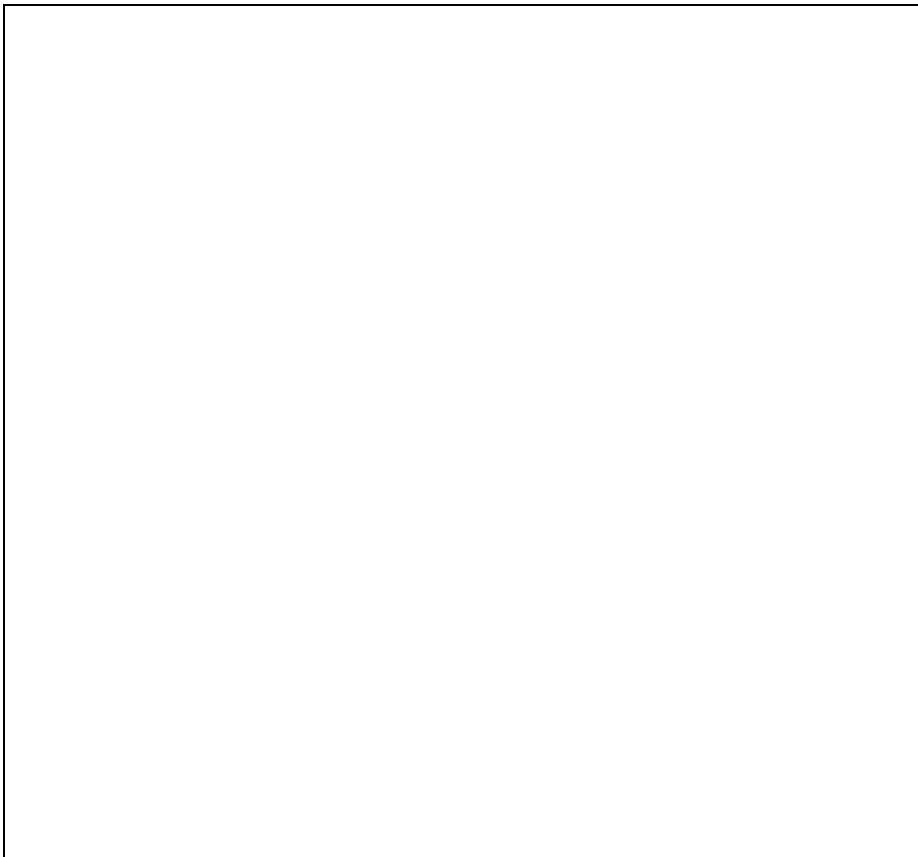
Funkcijos kodas formuojamas iš lauko `FunkcijosAtvaizdas.Kodas` reikšmių atskirtų taškais, „keliaujant“ rekursiniu ryšiu iki hierarchijos viršaus.

2.2.3 Struktūrinių veiklos taisyklių atskleidimas

Siūlomame metamodelyje struktūrinės veiklos taisyklės saugomos kaip konceptuali esybių ryšių diagrama. Tekstinė struktūrinių taisyklių išraiška patikrinimui gali būti pateikta naudojantis diagramos elementų aprašymais [1 Kapočius, K. 2006].

Esybių ir jų atributų atskleidimas vykdomas analizuojant funkcijas. Kiekvienoje funkcijoje galima rasti su veikla susijusių ar bendrų terminų. Kiekvienas terminas gali turėti sinonimų. Taip pat analizuojant funkcijas galima rasti ir veiklos logiką — elementarius faktus, apibūdinančius ryšius tarp terminų.

Atrastos esybės, jų atributai ir sinonimai bei ryšiai tarp esybių turėtų būti saugomi vartotojo reikalavimų specifikacijos saugykloje. Atitinkamas saugyklos metamodelio fragmentas matomas 8 paveiksle.



Šaltinis: sukurta autoriaus.

8 pav. Metamodelio fragmentas konceptualaus duomenų modeliui saugoti.

Kaip matome modelis leidžia saugoti ne tik esybes, atributus, ryšius bei sinonimus, bet ir fiksuotas atributų reikšmes, konstantas bei terminus kurie negali būti įtraukti į duomenų modelį (lentelė ReiksmeTerminas).

2.2.4 CRUD matricos kūrimas

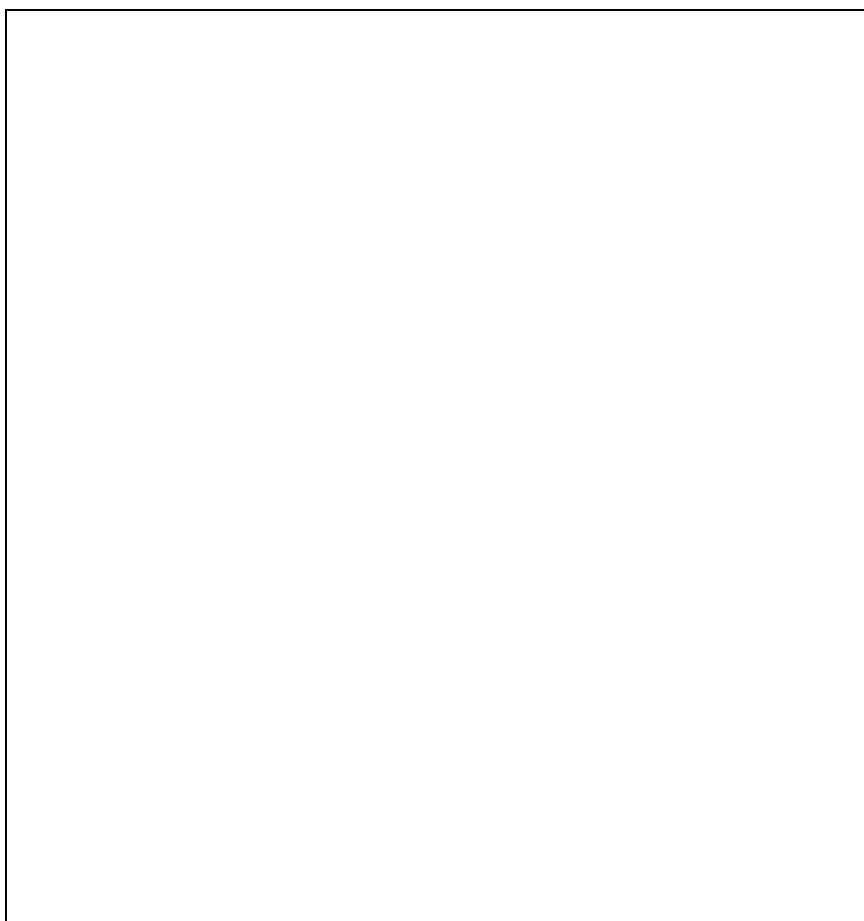
CRUD (create, read, update, delete) matrica atskleidžia, kaip terminas naudojamas vykdant funkciją. Sisteminiai įvykiai yra nepageidaujami veiklos taisyklių išraiškose ir negali būti jų subjektuose, todėl CRUD matrica vėliau gali būti (pvz.: programuotojo) naudojama kai prireikia nustatyti kokie įvykiai duomenų posistemėje vyksta vykdant funkciją. Taip pat CRUD matrica galės būti panaudota ir surandant veiklos taisyklių įtakojamus duomenų įvykius per šių taisyklių įtakojamas funkcijas. 4 lentelėje matoma CRUD matricos pavyzdys. Jame parodyta, kad nustatinėjant automobilio vertę perskaitomos keturių esybės automobilis atributų reikšmės ir sukuriami arba atnaujinami atributo vertė reikšmė.

CRUD matricos pavyzdys.

Terminas	Funkcija: Nustatyti automobilio vertę			
	C	R	U	D
Modelis		*		
Pagaminimo metai		*		
Rida		*		
Vertė	*	*	*	

Šaltinis: sukurta autoriaus.

Paprastai tik žemiausio lygio funkcijos turėtų būti įtraukiamos į matricą. CRUD matricos saugomos vartotojo reikalavimų specifikacijos saugykloje. Atitinkamas saugyklos metamodelio fragmentas matomas 9 paveiksle.



Šaltinis: sukurta autoriaus.

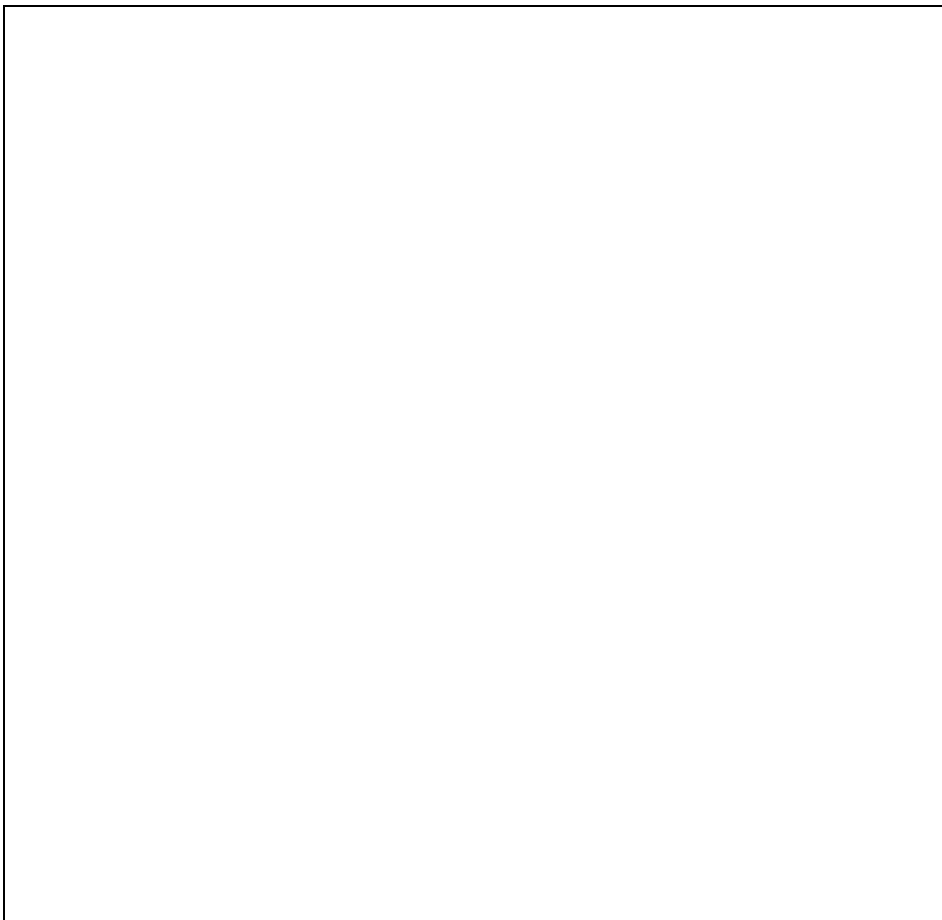
9 pav. Metamodelio fragmentas CRUD matricoms saugoti.

Pažymėtina, kad CRUD matricos sukūrimas reikalauja visiško funkcijos supratimo. Tačiau dažniausiai tai neįmanoma. Pavyzdžiui, susidūrus su funkcija „Nustatyti automobilio vertę“ aišku, kad automobilio atributas vertė bus sukurtas arba atnaujintas, tačiau netišku kokios charakteristikos

bus panaudotos skaičiuojant vertę. Ši informacija atrandama atskleidžiant susijusias nestruktūrines veiklos taisykles. Todėl CRUD matricos dažniausiai fiksuojamos lygiagrečiai su veiklos taisyklėmis [1 Kapočius, K. 2006].

2.2.5 Veiklos taisyklių šaltinių fiksavimas

Laikoma, kad kiekviena veiklos taisyklė turi savo šaltinį, t.y. analitikas apie ją iš kažkur sužino. Siūlomame metamodelyje išskirtos trys veiklos taisyklių šaltinių rūšys: asmenys, dokumentai ir programinis egzistuojančios informacinės sistemos kodas. Duomenys apie veiklos taisyklių šaltinius saugomi metamodelyje. Susijęs metamodelio fragmentas matomas 10 paveiksle.



Šaltinis: sukurta autoriaus.

10 pav. Metamodelio fragmentas VT šaltiniams saugoti.

Kaip matome iš fragmento, šaltinį asmenį turi ir dokumentai bei programinis kodas. Šis ryšys reikalingas vėliau jei prireikia ką nors patikslinti dėl dokumentų ar programinio kodo.

2.2.6 Nestruktūrinių veiklos taisyklių atskleidimas

Nestruktūrinės veiklos taisyklės atskleidžiamos analizuojant veiklos sprendimus. Barbara von Halle savo metode parodė, kad veiklos sprendimai yra geras ryšys tarp veiklos įvykių ir

taisyklių. Pagal mūsų siūlomą metodą veiklos sprendimai taikomi nes pačios funkcijos negali talpinti savyje pakankamai informacijos apie jas veikiančias taisykles. Taigi nestruktūrinių veiklos taisyklių atskleidimo procesas prasideda veiklos sprendimų analize [2 Kapočius K. 2005].

Sprendimai išreiškiami klausimais į kuriuos reikia atsakyti tam, kad galėtum įvykdyti atitinkamą funkciją. Sprendimai susieti su funkcija turi galioti ir visoms tos funkcijos subfunkcijoms. Funkcijos turinčios subfunkcijų negali būti susietos su sprendimais, tuo tarpu žemiausio lygmens funkcijos privalo turėti bent vieną sprendimą. Sprendimai turi būti saugomi metaduomenų bazėje.

Nestruktūrinės veiklos taisyklės atskleidžiamos kiekvienam veiklos sprendimui. Taisyklėms išreikšti pasirinktas BRS RuleSpeak metodas. Šis metodas reikalauja galimybės keisti veiklos taisyklių šablonus. Siūlomas veiklos taisyklių šablonų metaduomenų saugojimo modelis matomas 11 paveiksle.



Šaltinis: sukurta autoriaus.

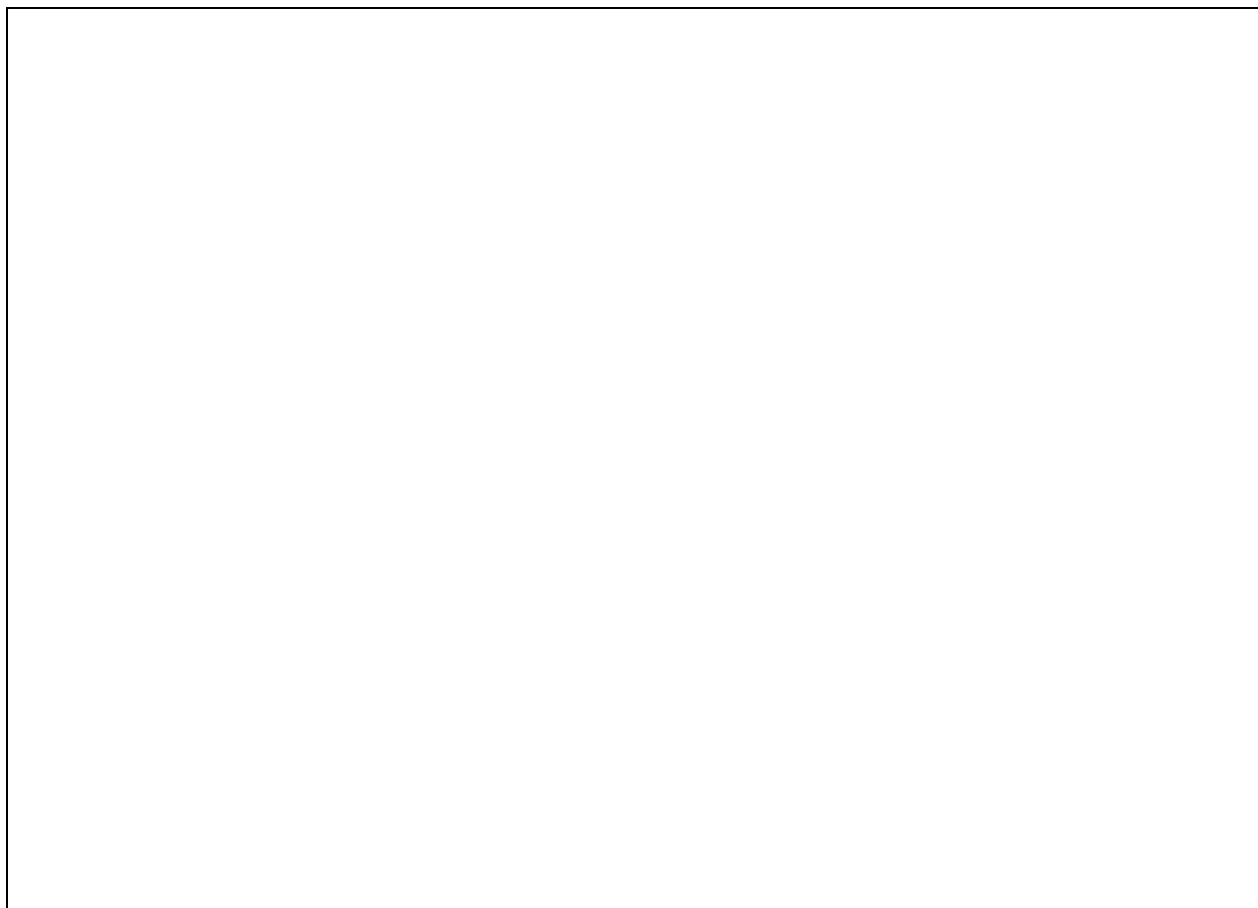
11 pav. Metamodelio fragmentas VT šablonams saugoti.

Šis šablonų saugojimo modelis optimizuotas BRS RuleSpeak veiklos taisyklių struktūrizavimo metodikai. Šis modelis taip pat leidžia dinamiškai generuoti taisyklių įvedimo į saugyklą formas. Tai reiškia, kad sistemos vartotojas/analitikas gali papildyti šablonų rinkinį savo nuožiūra, o naujo šablono struktūros taisyklių įvedimo langas bus pateikiamas automatiškai.

Atskleidžiant veiklos taisykles jos turi būti siejamos su duomenų modeliu (kuris neprivalo iš karto būti išbaigtas). Kitaip tariant nestruktūrinės veiklos taisyklės turi būti susiję su struktūrinėmis veiklos taisyklėmis. Todėl jeigu atrandama taisyklė su naujais terminais ir faktais, tai prieš ją fiksuojant saugykloje reikia atnaujinti duomenų modelį.

Kiekvienas struktūrinės taisyklės panaudojimas nestruktūrinėje taisyklėje turi būti fiksuojamas saugykloje susiejant atitinkamas esybes, atributus ir ryšius su taisyklės elementais. Taip pat patartina, kad nestruktūrinėje veiklos taisyklėje minimo duomenų elemento išraiška sutaptų su išraiška užfiksuota duomenų modelyje [2 Kapočius K. 2005].

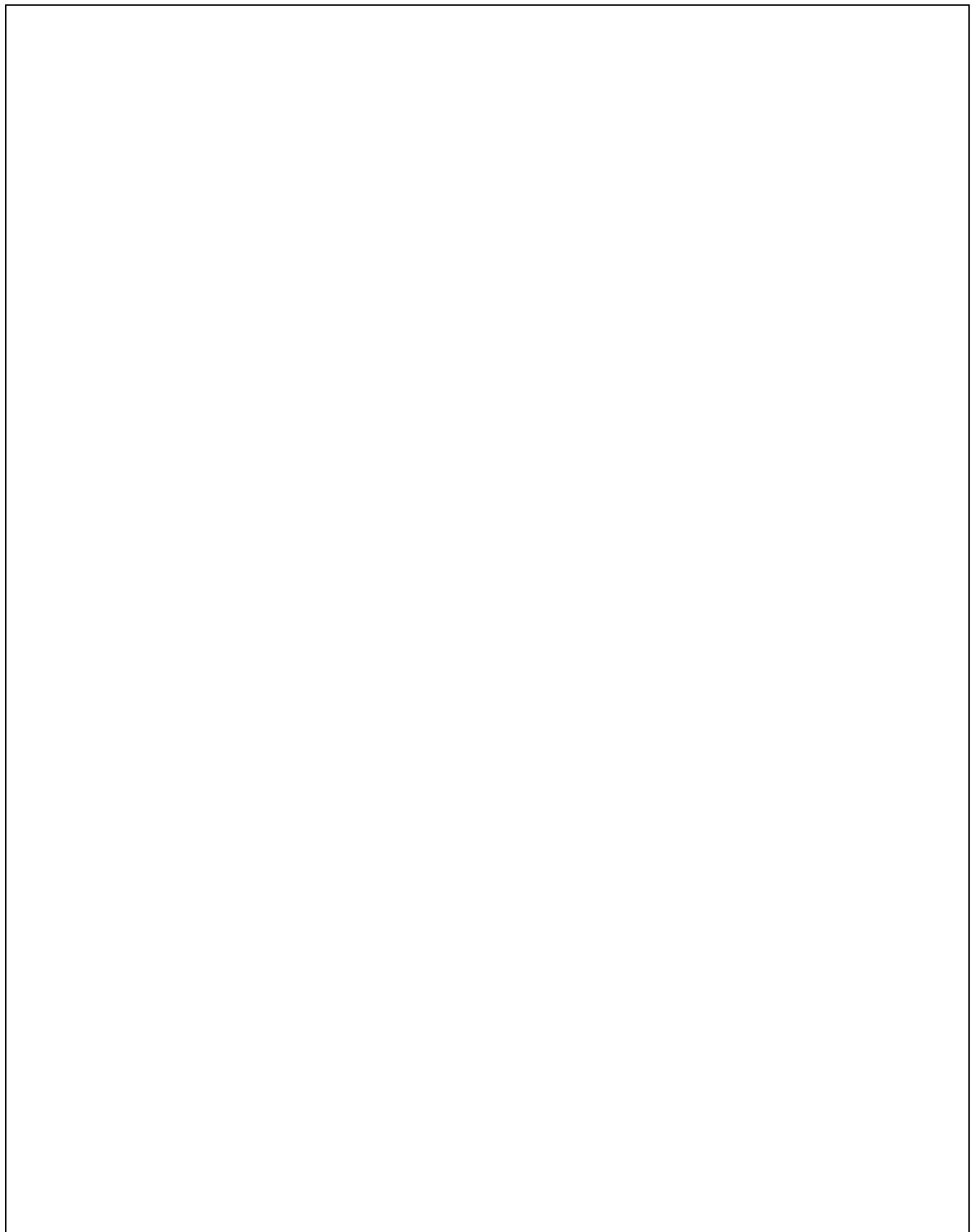
12 paveiksle matomas siūlomo metamodelio fragmentas kuriame saugomos nestruktūrinės veiklos taisyklės.



Šaltinis: sukurta autoriaus.

12 pav. Metamodelio fragmentas VT saugoti.

13 paveiksle pateikta viso siūlomo metamodelio schema.



Šaltinis: sukurta autoriaus.

13 pav. Veiklos taisyklėmis grindžiamos reikalavimų kuriamai Informacinei Sistemai specifikacijos saugojimo sistemos metamodelis.

2.3 Užfiksuotų vartotojo reikalavimų specifikacijos dokumentacija

Vartotojo poreikių kuriamai informacinei sistemai saugykla gali bet kuriuo metu sugeneruoti dokumentai skirtus užfiksuotų poreikių patikrinimui, verifikavimui. Tikrinti reikia konceptualų duomenų modelį, nestruktūrines veiklos taisykles, funkcijas ir sprendimus. Kadangi visi šie elementai saugykloje susieti su aktoriais tai nesunku atrinkti, kas turėtų atlikti patikrinimą. Būtent ši patikrinimo stadija atskleidžia siūlomo vartotojo poreikių fiksavimo metodo privalumą — užfiksuotus vartotojo poreikius gali tikrinti patys vartotojai be analitiko pagalbos, kas ženkliai padidina tikrinimo efektyvumą ir savo ruožtu būsimos informacinės sistemos kokybę.

Ataskaitos ir diagramos vaidina labai svarbią rolę poreikių specifikacijoje. Štai keletas dokumentacijos formų kurias galima būtų realizuoti:

- Funkcijų hierarchijos diagrama.
- Funkcijų-VT arba VT-Funkcijų hierarchijos diagramos.
- Funkcijų-VT lentelės.
- Sprendimų-VT lentelės.
- ER diagrama.
- Terminų žodynas
- Sprendimų lentelės.

2.4 Pagrindinės gairės saugyklos vartotojo sąsajai

Bendri realizacijai keliami reikalavimai [13]:

- Siekiant neperkrauti langų įvairiomis informacijos įvedimo priemonėmis turi būti stengiamasi rodyti tik tuo metu reikalingas. Tam bus pasitelkiami lango puslapiai (Tab), elementų aktyvavimas/deaktyvavimas, bei paslėpimas/parodymas.
- Įvedamos informacijos korektiškumas turi būti kontroliuojamas nuo pat pirmo žingsnio, tai yra iškart įvedinėjant ją į laukus. Visur kur įmanoma bus apribotas galimų įvesti informacijos variantų kiekis. Taip nebus leidžiama nekorektiškai informacijai patekti į duomenų bazę.
- Patogiai įvestos informacijos peržiūrai turi būti realizuoti įvairūs filtrai, navigacijos ir paieškos priemonės.
- Languose turi būti išvedama tik vartotojui prasminga informacija (t.y. nerodomi identifikacijos ir kiti “tarnybiniai” laukai).
- Keletas modelio lentelių turi ryšius pačios su savimi. Vadinasi įrašai jose gali būti vaizduojami hierarchinės struktūros. Tam tinka medžio tipo vaizdavimo būdas.

- Medis bus panaudojamas ne tik kaip vizualizacijos, bet ir kaip įrašų navigacijos priemonė. Tai yra paspaudus ant konkrečios medžio šakos įrašų lentelėse bus pažymimi atitinkami įrašai.

Reikalavimai struktūrinių veiklos taisyklių posistemės

- Kai kurios esybės tarpusavyje yra susietos "Is A" ryšiu. Ši priklausomybė lengviau būtų suprantama ir pastebima atvaizdavus ją medžio struktūroje.
- Kadangi esybės paprastai turi riboto dydžio atributų skaičių, tai minėtame medyje turi būti galimybė vaizduoti ir esybių atributus. Atributų vaizdavimas privalo skirtis nuo esybių vaizdavimo.
- Esybių bei atributų sinonimai nėra nuolat reikalinga informacija, todėl ji dažniausiai bus paslėpta. Specialus mygtukas leis įjungti lango sinonimų peržiūros ir redagavimo posistemį. Čia bus rodomi ir redaguojami tik tuo metu pažymėtos esybės ar atributo sinonimai.
- Ryšių tarp esybių patogesniai registravimui reikalingi du esybių medžiai, tik šiuo atveju jau be atributų.
- Kardinalumų reikšmes turi būti leidžiama pasirinkti iš sąrašo arba įvesti ranka.

Reikalavimai veiklos taisyklių šaltinių įvedimo posistemės realizacijai:

- Pareigybių hierarchija vaizduojama kaip medis, taip pat leidžiama pasirinkti ar rodyti medyje pareigybių asmenis.
- Tiek pareigybės tiek asmenys gali atlikti tam tikras funkcijas. Šiems ryšiams užmegzti reikalingas funkcijų hierarchiją vaizduojantis medis.
- Dokumentai ir programinis kodas susiejami su asmenimis leidžiant juos pasirinkti iš sąrašo.

Reikalavimai funkcijų įvedimo langui:

- Kadangi funkcijos saugomos dvejose lentelėse: vienoje funkcijos aprašymas, kitoje funkcijų hierarchija, tai informacijos apie funkcijas pateikimas praktiškai visada turi būti šių dviejų lentelių kombinavimas. Funkcijų hierarchija vaizduojama medžio struktūros, kuri saugoma funkcijų atvaizdų lentelėje, o medžio šakų pavadinimai saugomi funkcijų lentelėje. Medis padeda ir pačios hierarchijos formavime: kuriant funkcijos atvaizdą galima medyje pasirinkti tėvinę funkciją.
- Šiame lange vyksta ir sprendimų įvedimas bei jų susiejimas su funkcijomis. Funkcijų susiejimas su sprendimais taip pat vyksta dalyvaujant medžiui.

Reikalavimai CRUD santykio fiksavimo langui.

- Lange pateikiamos dvi hierarchijos: funkcijų ir terminų (esybių bei atributų). CRUD santykis užmezgamas tarp pažymėtų medyje įrašų. Tikrinama ar ryšys nebuvo užmezgtas prieš tai. Jei taip leidžiama jį redaguoti.
- CRUD ryšių peržiūrai naudojamos CrossTable tipo lentelės. Pateikiamą informaciją galima filtruoti pagal funkciją arba terminą.

Reikalavimai veiklos taisyklių šablonų įvedimo posistemei [13]:

Privalo leisti kurti bei koreguoti jau sukurtus veiklos taisyklių šablonus. Šablonai susideda iš tam tikra seka išdėstytų elementų. Šie elementai turi savo tipus bei raktažodžius. Todėl būtinos priemonės papildyti arba pakoreguoti tipų bei raktažodžių lenteles. Šablonai kuriami tam tikram veiklos taisyklių tipui. Kiekvienas veiklos taisyklės tipas gali turėti keletą šablonų. Šablono pavadinimą patartina formuoti taip, kad atspindėtų jį sudarančius elementus. Pradinį šablono pavadinimą pasiūlo pati sistema. Formuojant šabloną sistema išveda jo elementus sakinio struktūroje, kad galima būtų įsivaizduoti, kaip galėtų atrodyti pagal jį suformuota veiklos taisyklė.

Reikalavimai veiklos taisyklių įvedimo posistemei [13]:

Veiklos taisyklės įvedamos pasirinkus šabloną. Sistema dinamiškai pagal šablono elementus formuoja įvedimo laukus bei pasirinkimo sąrašus, išdėstydamą juos pagal elementų eilės tvarką. Susijęs modelio elementas pasirenkamas iš sąrašų (jei elementai sudaro hierarchiją tai iš medžio tipo sąrašų). Veiklos taisyklės išraiškos komponentų yra tiek kiek ir šablono elementų. Veiklos taisyklės išraiškos komponentų išraiškos priklauso nuo tipo, jei tai raktažodis tai išraiška bus konkretus nuo šablono elemento

3 Veiklos taisyklėmis grindžiamos reikalavimų kuriamai Informacinei Sistemai specifikacijos saugojimo sistemos eksperimentinė realizacija

3.1 Projekto tikslas

Šio projekto pagrindinis tikslas sukurti veiklos taisyklėmis grindžiamą reikalavimų kuriamai Informacinei Sistemai specifikacijos saugojimo sistemą. Pagrindinį tikslą išskaidyčiau į tokius smulkesnius tikslus:

- Naujo, pažangesnio metodo vartotojo reikalavimų specifikacijos fiksavimui realizavimas.
- Duomenų bazių lentelių struktūros realizavimas Access duomenų bazių valdymo sistemoje.
- Navigacijos informacinėje sistemoje posistemio sukūrimas ir įgyvendinimas.
- Aktorių hierarchijos išsaugojimo posistemio sukūrimas ir įgyvendinimas.
- Funkcijų hierarchijos specifikacijos saugojimo posistemio sukūrimas ir įgyvendinimas.
- Funkcijų sprendimų ir jų sekų specifikacijos saugojimo posistemio sukūrimas ir įgyvendinimas.
- Struktūrinių veiklos taisyklių (esybių ir ryšių) specifikacijos saugojimo posistemio sukūrimas ir įgyvendinimas.
- Esybių sinonimų išsaugojimo ir valdymo posistemio sukūrimas ir įgyvendinimas.
- Esybių su fiksuotomis reikšmėmis arba terminų saugojimo modulio sukūrimas ir įgyvendinimas.
- CRUD matricos posistemio sukūrimas ir įgyvendinimas.
- Veiklos taisyklių šaltinių sąrašų valdymo posistemio sukūrimas ir įgyvendinimas.
- Nestruktūrinių veiklos taisyklių šablonų kūrimo ir valdymo posistemio sukūrimas ir įgyvendinimas.
- Nestruktūrinių veiklos taisyklių kūrimo ir valdymo posistemio sukūrimas ir įgyvendinimas.
- Sprendimų lentelių posistemio sukūrimas ir įgyvendinimas.
- Veiklos taisyklių rinkinio verifikavimo posistemio sukūrimas ir įgyvendinimas.
- Veiklos modelio schemų generavimo posistemio sukūrimas ir įgyvendinimas.

Kadangi projektas yra didelės apimties ir magistriniame darbe visus tikslus aprėpti sudėtinga, tai dalis tikslų pilnai įgyvendinti šiame etape nebus. Bus apsiribota eksperimentinių prototipų realizavimu.

3.2 Projekto apimtys

Šis projektas susideda iš tokių dalių:

- Analizė
 - Surinkta prieinama veiklos taisyklių požiūrį aprašanti informacija.
 - Surinkta informacija išanalizuota orientuojantis į vartotojo reikalavimų specifikacijos procesą.
 - Išanalizuoti prieinami specifikacijų saugojimo metamodeliai ir pasirenkamas vienas iš jų įgyvendinimui.
- Projektavimas.
 - Duomenų struktūros projektavimas.
 - Sistemos architektūros projektavimas.
 - Ataskaitų sistemos projektavimas
- Projekto įgyvendinimas
 - Duomenų struktūros perkėlimas iš modelio į duomenų bazę
 - Sistemos realizavimas pasirinktoje programavimo terpėje.
 - Testinių duomenų įvedimas į sistemą.
 - Vartotojo reikalavimų modelių ataskaitų generavimas.
 - Vartotojo instrukcijos paruošimas.
 - Vartotojų mokymo plano paruošimas.

Projekto vykdymo eigoje gali paaiškėti, kad jo apimtis yra per didelė magistro darbo apimčiai.

3.3 Duomenų bazės schema

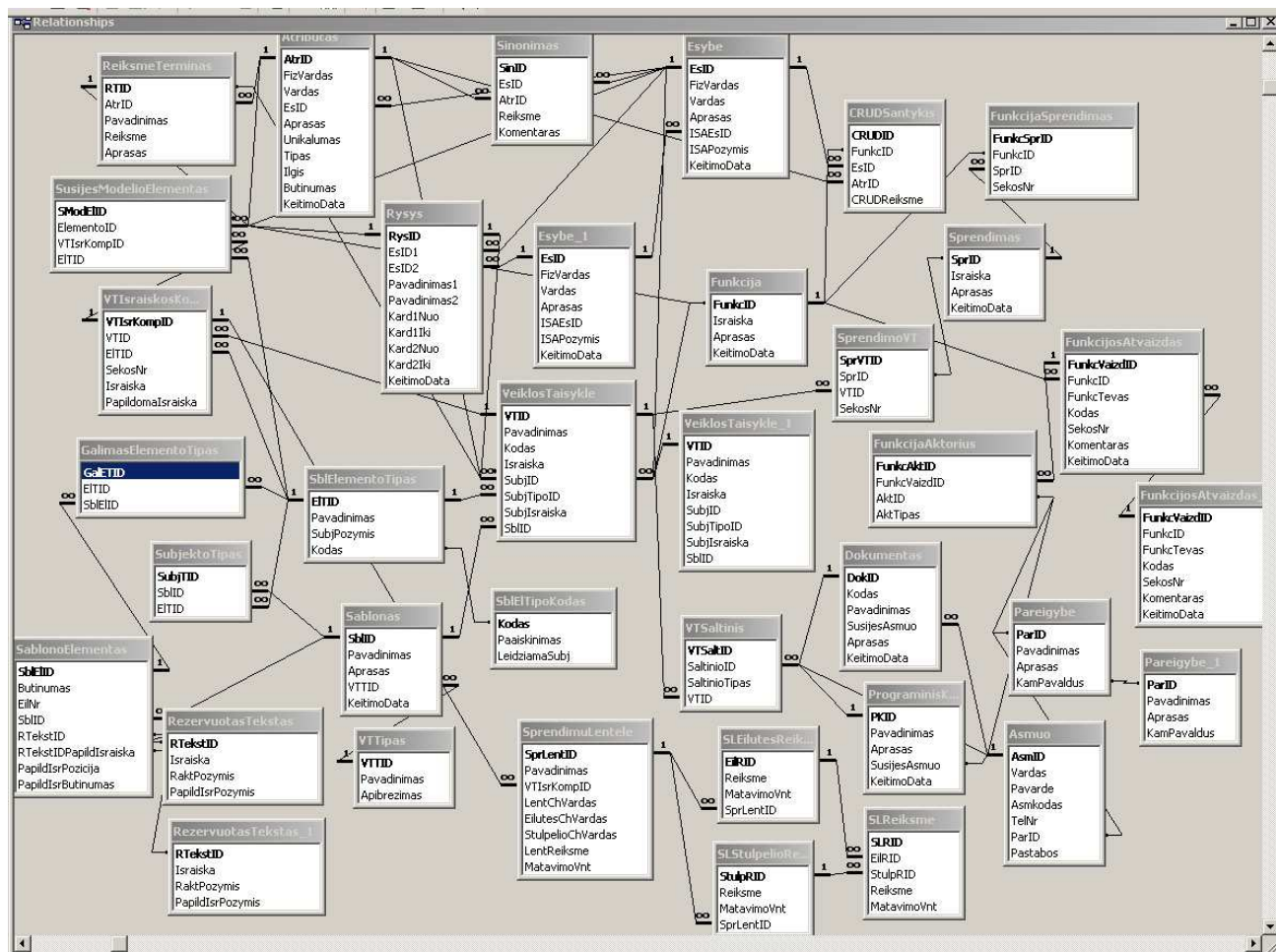
14 paveiksle matome kuriamos vartotojo poreikių kuriamai informacinei sistemai saugyklos duomenų bazės schemą.



Šaltinis:

14 pav. Vartotojo poreikių saugyklos DB schema.

15 paveiksle parodyta kaip atrodo šios duomenų bazės realizuotos Microsoft Access 2000 platformoje ryšių diagrama.



15 pav. DB ryšių schema Access platformoje.

Schemose lentelės pasistengta sugrupuoti pagal posistemas. Sistemos realizavimas bus vykdomas panašia tvarka, kaip ir reikalavimų specifikuojimo proceso metu. Duomenų bazė susideda iš 32 lentelių, 4 iš jų turi rekursinius ryšius su savimi, t.y. duomenys siejasi hierarchiškai.

Toliau pateikiu duomenų bazės lentelių aprašymus suskirstytus pagal posistemas.

3.3.1 Duomenų šaltinių posistemė

Asmuo. Duomenys apie asmenis, kurie gali būti / yra naudingi specifikuojant reikalavimus ir/arba yra potencialūs kuriamos IS vartotojai (aprašas – 5 lentelėje).

5 lentelė

DB lentelės Asmuo aprašas.

Atributas	Aprašas	Galimos reikšmės	Būtinai	Tipas
<i>AsmID</i>	Unikalus aktoriaus identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonom
<i>Vardas</i>	Asmens vardas.	Vardas (jeigu yra keli, tai pirmasis).	✓	chr(20)
<i>Pavarde</i>	Asmens pavardė.	Pavardė.	✓	chr(20)
<i>AsmKodas</i>	Asmens kodas.	Skaitmenų seka.		chr(11)

Atributas	Aprašas	Galimos reikšmės	Būti nas	Tipas
<i>TelNr</i>	Darbo/mobilaus telefono numeris.	Skaitmenų seka.		chr(13)
<i>ParID</i>	Pareigybės, kurią organizacijos viduje užima asmuo, id.	Pasirinkta <i>ParID</i> reikšmė iš lentelės <i>Pareigybe</i> .		num(10)
<i>Pastabos</i>	Laisvos formos komentaras.	Bet koks tekstas.		chr(200)
Vienas-su-daug ryšys su <esybė>.<atributas>				
Vienas-su-daug ryšys su <esybė>.<atributas>		Reikšmė	Būti nas	
<i>FunkcijaAktorius.AktID</i>		Nuorodų į funkcijų atvaizdus, su kuriais siejamas asmuo, aibė.		
<i>VTSaltinis.SaltinioID</i>		Nuorodų į veiklos taisykles, kurių kilmės šaltiniu (ar vienu iš šaltinių) yra asmuo, aibė.		
<i>PrograminisKodas.SusijęsAsmuo</i>		Programinio kodo elementų, kurių autoriumi, pateikėju ar pan. yra asmuo, aibė.		
<i>Dokumentas.SusijęsAsmuo</i>		Dokumentų, kurių autoriumi, pateikėju ar pan. yra asmuo, aibė.		

Pareigybe. Duomenys apie analizės prasme svarbias pareigybes, egzistuojančias organizacijoje, kuriai yra kuriama IS (aprašas 6 lentelėje).

6 lentelė

DB lentelės Pareigybe aprašas.

Atributas	Aprašas	Galimos reikšmės	Būti nas	Tipas
<i>ParID</i>	Unikalus pareigybės identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonum
<i>Pavadinimas</i>	Pareigybės pavadinimas.	Bet koks tekstas.	✓	chr(20)
<i>Aprašas</i>	Pareigybės aprašas.	Bet koks tekstas.		chr(50)
<i>KamPavaldus</i>	Pareigybės, kuriai pagal organizacijos administracinę hierarchiją pavaldūs šios pareigybės atstovai, id.	Tokia pasirinkta <i>ParID</i> reikšmė iš jau esančių lentelėje <i>Pareigybe</i> , kad <i>ParID</i> ≠ <i>KamPavaldus</i> .		num(10)
Vienas-su-daug ryšys su <esybė>.<atributas>				
Vienas-su-daug ryšys su <esybė>.<atributas>		Reikšmė	Būti nas	
<i>Asmuo.ParID</i>		Aktorių, vykdančių/užimančių šią pareigybę, aibė.		
<i>Pareigybe.KamPavaldus</i>		Pareigybių, kurių atstovai yra tiesiogiai pavaldūs šios pareigybės atstovams, aibė.		
<i>FunkcijaAktorius.AktID</i>		Nuorodų į funkcijų atvaizdus, su kuriais siejama pareigybė, aibė.		

Dokumentas. Duomenys apie spausdintinius, rašytinius, elektroninius ar daugialypės terpės (*angl. multimedia*) dokumentus, kuriuose yra informacijos, atskleidžiančios, paaiškinančios ar vaizduojančios reikalavimus kuriamai IS (aprašas 5 lentelėje).

DB lentelės Dokumentas aprašas.

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<i>DokID</i>	Unikalus dokumento identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonom
<i>Kodas</i>	Dokumento identifikacinis kodas arba numeris (jeigu toks yra žinomas).	Bet koks tekstas.		chr(20)
<i>Pavadinimas</i>	Dokumento pavadinimas.	Bet koks tekstas.	✓	chr(50)
<i>SusijęsAsmuo</i>	Asmens (aktoriaus), kuris aiškiausiai susijęs su dokumentu (yra autorius, teikėjas, aiškintojas ir pan.), id. Jeigu tokių asmenų yra keli, analitikas parenka tik vieną iš jų.	Pasirinkta <i>AsmID</i> reikšmė iš lentelės <i>Asmuo</i> .		num(10)
<i>Aprašas</i>	Dokumento esmines savybes atskleidžiantis aprašymas.	Bet koks tekstas.		chr(200)
<i>KeitimoData</i>	Dokumento įvedimo į saugyklą arba paskutinio bent vienos dokumento charakteristikos redagavimo data. Taip pat nurodo, ar dokumentas nėra pašalintas (praradęs aktualumą).	Data ir tikslus laikas. Šalinant dokumentą, nustatoma reikšmė "1111.11.11"; kitos charakteristikos nepakinta.	✓	datetime
Vienas-su-daug ryšys su <esybė>.<atributas>				
		Reikšmė	Būtin as	
<i>VTSaltinis.SaltinioID</i>		Nuorodų į veiklos taisykles, kurių kilmės šaltiniu (ar vienu iš šaltinių) yra dokumentas, aibė.		

PrograminisKodas. Duomenys apie analizuojamoje veikloje jau naudojamų programų kodo elementus ar iškarpas, taip pat ir veikiančias programas ar jų elementus (langus, formas ir pan.), kuriuose yra informacijos, atskleidžiančios, paaiškinančios ar vaizduojančios reikalavimus kuriamai IS.

DB lentelės PrograminisKodas aprašas.

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<i>PKID</i>	Unikalus programinio kodo ar elemento identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonom
<i>Pavadinimas</i>	Programinio kodo ar elemento identifikatorius arba sutartinis pavadinimas.	Bet koks tekstas.	✓	chr(50)
<i>SusijęsAsmuo</i>	Asmens (aktoriaus), kuris aiškiausiai susijęs su programiniu kodu ar elementu (yra autorius, teikėjas, aiškintojas ir pan.),	Pasirinkta <i>AsmID</i> reikšmė iš lentelės <i>Asmuo</i> .		num(10)

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
	id. Jeigu tokių asmenų yra keli, analitikas parenka tik vieną iš jų.			
<i>Aprašas</i>	Programinio kodo ar elemento esmines savybes atskleidžiantis aprašymas.	Bet koks tekstas.		chr(200)
<i>KeitimoData</i>	Programinio kodo ar elemento įvedimo į saugyklą arba paskutinio bent vienos jo charakteristikos redagavimo data. Taip pat nurodo, ar šis įrašas nėra pašalintas (praradęs aktualumą).	Data ir tikslus laikas. Šalinant įrašą, nustatoma reikšmė "1111.11.11"; kitos charakteristikos nepakinta.	✓	datetime
Vienas-su-daug ryšys su <esybė>.<atributas>				
		Reikšmė	Būtin as	
<i>VTSaltinis.SaltinioID</i>	Nuorodų į veiklos taisykles, kurių kilmės šaltiniu (ar vienu iš šaltinių) yra programinis kodas ar elementas, aibė.			

3.3.2 Funkcijų ir sprendimų posistemė

Funkcija. Duomenys apie reikalavimų analizės metu fiksuojamas nagrinėjamoje veikloje vykdomas funkcijas.

9 lentelė

DB lentelės Funkcija aprašas.

Atributas	Aprašas	Galimos reikšmės	Būti nas	Tipas
<i>FunkcID</i>	Unikalus funkcijos identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonom
<i>Israiska</i>	Funkcijos pavadinimas (gali būti santrumpa).	Bet koks tekstas.	✓	chr(100)
<i>Aprašas</i>	Funkcijos aprašas.	Bet koks tekstas.		chr(200)
<i>KeitimoData</i>	Funkcijos įvedimo į saugyklą arba paskutinio bent vienos funkcijos charakteristikos redagavimo data. Taip pat nurodo, ar funkcija nėra pašalinta (praradus aktualumą).	Data ir tikslus laikas. Šalinant funkciją, nustatoma reikšmė "1111.11.11"; kitos charakteristikos nepakinta.	✓	datetime
Vienas-su-daug ryšys su <esybė>.<atributas>				
		Reikšmė	Būti nas	

Atributas	Aprašas	Galimos reikšmės	Būtinai	Tipas
FunkcijosAtvaizdas.FunkcID	Funkcijų, atvaizduojančių šią funkcijų hierarchijos diagramoje, aibė.			✓
FunkcijaSprendimas.FunkcID	Nuorodų į sprendimus, kuriuos reikia priimti (arba klausimus, į kuriuos reikia atsakyti) norint įvykdyti funkciją, aibė.			
CRUDSantykis.FunkcID	CRUD santykių, kuriais apibrėžiama funkcijos vykdymo įtaka esybėms arba atributams (terminams), aibė.			

FunkcijosAtvaizdas. Duomenys apie Funkcijų hierarchijos (FH) diagramoje vaizduojamą nagrinėjamos veiklos funkciją. Funkcijos atvaizdas paveldi visas atvaizduojamos funkcijos (lentele Funkcija) charakteristikas.

10 lentelė

DB lentelės FunkcijosAtvaizdas aprašas.

Atributas	Aprašas	Galimos reikšmės	Būtinai	Tipas
<i>FunkcVaizdID</i>	Unikalus funkcijos atvaizdo identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonum
<i>FunkcID</i>	Funkcijos, kurią atvaizduoja šis atvaizdas, id.	Pasirinkta <i>FunkcID</i> reikšmė iš lentelės <i>Funkcija</i> .	✓	num(10)
<i>FunkcTevas</i>	Funkcijos, kurios vaikinė funkcija pagal FH diagramą yra ši funkcija, id.	Tokia pasirinkta <i>FunkcVaizdID</i> reikšmė iš jau esančių lentelėje <i>FunkcijosAtvaizdas</i> , kad $FunkcVaizdID \neq FunkcTevas$.		num(10)
<i>Kodas</i>	Funkcijos numeris konkrečios tėvinės funkcijos vaikinių funkcijų tarpe. <u>Informacijos išvedimo (ataskaitose, diagramose ir pan.) metu nurodomas funkcijos kodas gaunamas sudėjus visų aukštesnio lygio tėvinių funkcijų (iki pirminės funkcijos) kodus atskiriant juos tašku “.”.</u>	Skaičius nuo 1 iki 99. Jeigu $FunkcSekosNr \neq 0$, tai <i>FunkcKodas</i> reikšmės perskaičiuojamos automatiškai sunumeruojant <i>FunkcSekosNr</i> didėjimo tvarka. Konkrečios tėvinės funkcijos vaikinės funkcijos kodai turi būti unikalūs.	✓	num(2)
<i>SekosNr</i>	Funkcijos eilės numeris tėvinės funkcijos vykdymo sekoje. Būtina numeruoti arba visas vaikinės funkcijas, arba nei vienos. Jeigu funkcijos vykdomos lygiagrečiai, joms gali būti suteikiamas toks pats sekos numeris.	Skaičius nuo 1 iki 99.		num(2)
<i>Komentaras</i>	Papildomas komentaras, būdingas tik konkrečiam funkcijos atvaizdui ir nepersidengiantis su	Bet koks tekstas.		chr(100)

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
	atvaizduojamos funkcijos aprašu..			
<i>KeitimoData</i>	Funkcijos atvaizdo įvedimo į saugyklą arba paskutinio bent vienos funkcijos atvaizdo charakteristikos redagavimo data. Taip pat nurodo, ar funkcijos atvaizdas nėra pašalintas (praradęs aktualumą).	Data ir tikslus laikas. Šalinant funkcijos atvaizdą, nustatoma reikšmė “1111.11.11”; kitos charakteristikos nepakinta.	✓	datetime
Vienas-su-daug ryšys su <esybė>.<atributas>				
		Reikšmė	Būtin as	
<i>FunkcijaAktorius.</i> <i>FunkcVaizdID</i>	Nuorodų į aktorius, su kuriais siejama funkcija, aibė.			
<i>FunkcijosAtvaizdas.</i> <i>FunkcTevas</i>	FH diagramos funkcijų, kurios yra “vaikinės” šiai funkcijai (arba į kurias detalizuojama ši funkcija), aibė.			

FunkcijaAktorius. Sąryšio daug-su-daug tarp lentelių *Aktorius* arba *Pareigybė* ir *FunkcijosAtvaizdas* pagalbinė lentelė. Kiekvienas aktorius arba pareigybė gali būti siejami su daugiau nei viena funkcija, o kiekviena funkcija savo ruožtu gali sietis su daugiau nei vienu asmeniu ar pareigybe (aktoriumi). Sąsaja užmezgama, jeigu aktorius ar neapibrėžtas organizacijos pareigūnas/darbuotojas atlieka/atliks funkciją arba gali suteikti / jau suteikė informacijos apie funkciją arba jai aktualius sprendimus, veiklos taisykles ir kt.

11 lentelė

DB lentelės *FunkcijaAktorius* aprašas.

Atributas	Aprašas	Galimos reikšmės	Būti nas	Tipas
<i>FunkcVaizdID</i>	Funkcijos atvaizdo identifikacinis numeris.	Pasirinkta <i>FunkcVaizdID</i> reikšmė iš lentelės <i>FunkcijosAtvaizdas</i> .	✓	num(10)
AktID	Aktoriaus identifikacinis numeris.	Vienas iš dviejų variantų: - pasirinkta <i>AsmID</i> reikšmė iš lentelės <i>Asmuo</i> , - pasirinkta <i>ParID</i> iš lentelės <i>Pareigybe</i> .	✓	num(10)
AktTipa	Kodas, nurodantis, ar aktorius yra asmuo, ar neapibrėžtas pareigūnas/darbuotojas.	- “A”: asmuo (<i>AsmID</i>), - “P”: pareigybė (<i>ParID</i>).	✓	chr(1)

Sprendimas. Duomenys apie sprendimus. Sprendimas – tai klausimas, į kurį reikia atsakyti norint įvykdyti tam tikrą(-as) funkciją(-as). Nagrinėjant sprendimus yra nustatomos nestruktūrinės veiklos taisyklės.

DB lentelės Sprendimas aprašas.

Atributas	Aprašas	Galimos reikšmės	Būti nas	Tipas
<i>SprID</i>	Unikalus sprendimo identifikacinis numeris.	Pasirinkta <i>FunkcVaizdID</i> reikšmė iš lentelės <i>FuncijosAtvaizdas</i> .	✓	autonum
Israiska	Sprendimo išraiška (klausimo forma).	Bet koks tekstas.	✓	chr(200)
Aprasas	Sprendimą paaiškinantis ar aprašantis komentaras.	Bet koks tekstas.		chr(200)
<i>KeitimoData</i>	Sprendimo įvedimo į saugyklą arba paskutinio bent vienos sprendimo charakteristikos redagavimo data. Taip pat nurodo, ar sprendimas nėra pašalintas (praradęs aktualumą).	Data ir tikslus laikas. Šalinant sprendimą, nustatoma reikšmė "1111.11.11"; kitos charakteristikos nepakinta.	✓	datetime
Vienas-su-daug ryšys su <esyb>.<atributas>				
		Reikšmė	Būti nas	
<i>FunkcijaSprendimas.SprID</i>	Nuorodų į atvaizdus funkcijų, kurioms įvykdyti reikia atsakyti į šį klausimą (priimti šį sprendimą), aibė.		✓	
<i>SprendimoVT.SprID</i>	Nuorodų į veiklos taisykles, kurios nustatytos nagrinėjant sprendimą, aibė.			

FunkcijaSprendimas. Sąryšio daug-su-daug tarp lentelių *FuncijosAtvaizdas* ir *Sprendimas* pagalbinė lentelė. Kiekvienai konkrečiai funkcijai įvykdyti gali reikėti priimti vieną ar daugiau sprendimų, o kiekvienas sprendimas gali būti aktualus vienam ar daugiau funkcijų.

DB lentelės *FunkcijaSprendimas* aprašas.

Atributas	Aprašas	Galimos reikšmės	Būti nas	Tipas
<i>FunkcSprID</i>	Funkcijos sprendimo unikalus identifikacinis numeris.	Automatiškai priskiriamas skaičius.	✓	autonum
<i>SprID</i>	Sprendimo identifikacinis numeris.	Pasirinkta <i>SprID</i> reikšmė iš lentelės <i>Sprendimas</i> .	✓	num(10)
<i>FunkcID</i>	Funkcijos (bazinės) identifikacinis numeris.	Pasirinkta <i>FunkcID</i> reikšmė iš lentelės <i>Funcija</i> .	✓	num(10)
<i>SekosNr</i>	Sprendimo eilės numeris santykinėje sprendimų priėmimo sekoje, jeigu tokios sekos fiksavimas yra reikalingas.	Skaičius nuo 1 iki 99. Numeravimas kiekvienam funkcijos atvaizdai pradedamas iš naujo.		num(2)

3.3.3 Konceptualaus duomenų modelio (struktūrinių veiklos taisyklių) posistemė

Esybe. Duomenys apie konceptualaus duomenų modelio esybes.

DB lentelės Esybe aprašas.

Atributas	Aprašas	Galimos reikšmės	Būti nas	Tipas
<i>EsID</i>	Unikalus esybės identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonum
FizVardas	Esybės fizinis vardas, naudojamas duomenų modelyje.	Bet koks žodis ar simbolių junginys be tarpų.	✓	chr(25)
<i>Vardas</i>	Esybės pavadinimas natūralia kalba.	Bet koks tekstas.	✓	chr(50)
<i>Aprašas</i>	Aprašas, atskleidžiantis esybės savybes, kurių negalima nuspėti pagal vardą.	Bet koks tekstas.		chr(200)
<i>ISAEsID</i>	Esybės, kuri yra šios esybės potipis, poklasis, rūšis ar pan. (ISA ryšys, apibendrinimo ryšys), id.	Pasirinkta tokia <i>EsID</i> reikšmė iš lentelės <i>Esybe</i> , kad <i>EsID</i> ≠ <i>ISAEsID</i> .		num(10)
ISAPozymis	Požymis, nurodantis, ar esybė yra apibendrinanti (t.y., ar egzistuoja ISA sąryšiu su esybe susijusių esybių).	Taip (true, 1) arba Ne (false, 0).		boolean
<i>KeitimoData</i>	Esybės įvedimo į saugyklą arba paskutinio bent vienos esybės charakteristikos redagavimo data. Taip pat nurodo, ar esybė nėra pašalinta (praradus aktualumą).	Data ir tikslus laikas. Šalinant esybę, nustatoma reikšmė "1111.11.11"; kitos charakteristikos nepakinta.	✓	datetime
Vienas-su-daug ryšys su <esybė>.<atributas>				
Vienas-su-daug ryšys su <esybė>.<atributas>		Reikšmė	Būti nas	
<i>Rysys.EsID1</i>		Sąryšių tarp dviejų esybių, kuriuose viena iš besisiejančių esybių yra ši esybė, aibė.		
<i>Rysys.EsID2</i>		Sąryšių tarp dviejų esybių, kuriuose viena iš besisiejančių esybių yra ši esybė, aibė.		
<i>Esybe.ISAEsID</i>		Esybių, kurios yra šios esybės potipiai, poklasiai rūšys ir pan. (siejasi su šia esybe ISA sąryšiu), aibė.		
<i>Atributas.EsID</i>		Atributų, kurie aprašo pagrindines šios esybės savybes ir charakteristikas, aibė.		
<i>Sinonimas.EsID</i>		Esybės vardo sinonimų (žodžių ar išraiškų) aibė.		
CRUDSantykis.EsD		CRUD santykių, kuriais apibrėžiamas esybės panaudojimo pobūdis vykdant funkcijas, aibė.		
VeiklosTaisykle.SubjektoID		Veiklos taisyklių, kuriuose esybė atlieka subjekto vaidmenį, aibė.		
SusijęsModelioElementas.ElementoID		Nuorodų į nustatytų VT išraiškų komponentus, kuriuose naudojamas esybės pavadinimas, jo forma ar sinonimas.		

Rysys. Duomenys apie sąryšius tarp dviejų esybių.

DB lentelės Rysys aprašas.

Atributas	Aprašas	Galimos reikšmės	Būti nas	Tipas
<i>RysID</i>	Unikalus ryšio identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonum
EsID1	Esybės, kuri vienas-su-daug sąryšyje yra sąlyginėje kairėje pusėje (arba tiesiog yra pirmoji esybė sąryšyje dėl to, kad nurodoma pirmoji), id.	Pasirinkta <i>EsID</i> reikšmė iš lentelės <i>Esybe</i> .	✓	num(10)
<i>EsID2</i>	Esybės, kuri vienas-su-daug sąryšyje yra sąlyginėje dešinėje pusėje (arba tiesiog yra pirmoji esybė sąryšyje dėl to, kad nurodoma pirmoji), id.	Pasirinkta <i>EsID</i> reikšmė iš lentelės <i>Esybe</i> . <i>EsID2</i> gali sutapti su <i>EsID1</i> .	✓	num(10)
<i>Pavadinimas1</i>	Sąryši iš esybės1 į esybę2 charakterizuojanti veiksmožodinė frazė.	Bet koks tekstas.	✓	chr(50)
<i>Pavadinimas2</i>	Sąryši iš esybės2 į esybę1 charakterizuojanti veiksmožodinė frazė.	Bet koks tekstas.	✓	chr(50)
Kard1Nu o	Sąryšio kardinalumo esybės1 pusėje apatinė riba.	- sveikas skaičius 0 ... 999.	✓	chr(3)
Kard1Iki	Sąryšio kardinalumo esybės1 pusėje viršutinė riba.	- sveikas skaičius 0 ... 999, - simbolis "*" (reikšmė – "daug").	✓	chr(3)
Kard2Nu o	Sąryšio kardinalumo esybės2 pusėje apatinė riba.	- sveikas skaičius 0 ... 999.	✓	chr(3)
Kard2Iki	Sąryšio kardinalumo esybės2 pusėje viršutinė riba.	- sveikas skaičius 0 ... 999, - simbolis "*" (reikšmė – "daug").	✓	chr(3)
<i>KeitimoData</i>	Ryšio įvedimo į saugyklą arba paskutinio bent vienos ryšio charakteristikos redagavimo data. Taip pat nurodo, ar ryšys nėra pašalintas (praradęs aktualumą).	Data ir tikslus laikas. Šalinant ryšį, nustatoma reikšmė "1111.11.11"; kitos charakteristikos nepakinta.	✓	datetime
Vienas-su-daug ryšys su <esybė>.<atributas>				
SusijęsModelioElementas. ElementoID		Reikšmė		Būti nas
Nuorodų į nustatytų VT išraiškų komponentus, kuriuose naudojamas vienas iš fakto pavadinimų ar jo forma.				
VeiklosTaisykle.SubjektoID		Veiklos taisyklių, kuriose ryšys atlieka subjekto vaidmenį, aibė.		

Atributas. Duomenys apie konceptualaus duomenų modelio esybių atributus.

DB lentelės Atributas aprašas.

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

ReiksmeTerminas. Duomenys apie dalykinėje srityje svarbias esybių charakteristikų (atributų) reikšmes, taip pat ir apie savarankiškas reikšmes. Daugelis reikšmių gali būti fiksuojamos tik atitinkamose taisyklėse, nekuriant įrašo šioje lentelėje, tačiau jeigu reikšmės yra svarbios (pvz.: rodikliai, fiksuoti dydžiai, kurie naudojami sąlygų išraiškose), tuomet jos bus saugojamos šioje lentelėje. Lentelėje paprastai nebus saugojamos reikšmės, kurios priskiriamos atributui esant tam tikroms sąlygoms. Savarankiškos reikšmės pavyzdys: taisyklėse gali būti naudojama reikšmė “fizinių pajamų mokestis”, kuri nėra nei atributas, nei esybė – tokios reikšmės vertė būtų saugoma šioje lentelėje, o ją naudojančiose taisyklėse būtų nurodomas tik reikšmės pavadinimas. Atributo reikšmės pavyzdys: automobiliams gali būti žinoma “populiariausia modifikacija” (esybės “automobilis” atributo “modifikacija” reikšmė), kurios reali reikšmė gali keistis ir būtų saugoma šioje lentelėje, o taisyklėse būtų nurodomas tik reikšmės pavadinimas.

Lentelėje taip pat saugomi su konceptualių duomenų modeliu nesietini, tačiau svarbūs terminai, kurie neturi jokių specifinių reikšmių ir analitikui svarbu tik suprasti jų reikšmę. Tokių terminų pavyzdžiai: “darbo etika”, “difuzija” ir pan.

17 lentelė

DB lentelės ReiksmeTerminas aprašas.

Atributas	Aprašas	Galimos reikšmės	Būti nas	Tipas
<i>RTID</i>	Unikalus reikšmės identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonum
<i>AtrID</i>	Atributo, kurio reikšmė saugoma, id.	Pasirinkta <i>AtrID</i> reikšmė iš lentelės <i>Atributas</i> .		num(10)
<i>Pavadinimas</i>	Reikšmės pavadinimas, kuris bus naudojamas formuluojant veiklos taisyklės ir kitus reikalavimus.	Bet koks tekstas.	✓	chr(50)
<i>Reiksme</i>	Reikšmės išraiška iš vieno ar daugiau žodžių, skaičių ar pan.	Bet koks tekstas.		chr(50)
<i>Aprasas</i>	Aprašas arba komentaras, atskleidžiantis termino arba reikšmės prasmę.	Bet koks tekstas.		chr(100)
Vienas-su-daug ryšys su <esybė>.<atributas>				
Vienas-su-daug ryšys su <esybė>.<atributas>		Reikšmė	Būti nas	
SusijęsModelioElementas. ElementoID		Nuorodų į nustatytų VT išraiškų komponentus, kuriuose naudojamas reikšmės pavadinimas, jo forma ar sinonimas, aibė.		
VeiklosTaisykle.SubjektoID		Veiklos taisyklių, kuriuose reikšmė ar terminas atlieka subjekto vaidmenį, aibė.		

Sinonimas. Duomenys atributų ar esybių pavadinimų sinonimus.

DB lentelės Sinonimas aprašas.

Atributas	Aprašas	Galimos reikšmės	Būti nas	Tipas
<i>SinID</i>	Unikalus sinonimo identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonum
EsID	Esybės, kurios papildoma išraiška yra šis sinonimas, id.	Pasirinkta <i>EsID</i> reikšmė iš lentelės <i>Esybe</i> . - jeigu <i>EsID</i> ≠ Null, tai <i>AtrID</i> = Null; - jeigu <i>EsID</i> = Null, tai <i>AtrID</i> ≠ Null.		num(10)
<i>AtrID</i>	Atributo, kurio papildoma išraiška yra šis sinonimas, id.	Pasirinkta <i>AtrID</i> reikšmė iš lentelės <i>Atributas</i> . - jeigu <i>AtrID</i> ≠ Null, tai <i>EsID</i> = Null; - jeigu <i>AtrID</i> = Null, tai <i>EsID</i> ≠ Null.		num(10)
<i>Reiksmė</i>	Sinonimo išraiška iš vieno ar daugiau žodžių.	Bet koks tekstas.	✓	chr(50)
<i>Komentaras</i>	Paaiškinimas, apibrėžiantis sinonimo naudojimo vietoj pagrindinės išraiškos specifiką.	Bet koks tekstas.		chr(100)

CRUDSantykis. Duomenys apie terminų (esybių arba atributų pavadinimų) santykį su funkcijomis. Konkretaus termino CRUD santykiu su konkrečia funkcija apibrėžiamas šio termino panaudojimo vykdant funkciją fizinis pobūdis (CRUD: create, read, update, delete – sukurti, skaityti, atnaujinti, šalinti).

DB lentelės CRUDSantykis aprašas.

Atributas	Aprašas	Galimos reikšmės	Būti nas	Tipas
CRUID	Unikalus CRUD santykio identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonum
FunkcID	Funkcijos, kurios CRUD santykis su esybe arba atributu yra fiksuojamas, ID.	Pasirinkta <i>FunkcID</i> reikšmė iš lentelės <i>Funkcija</i> .	✓	num(10)
EsID	Esybės, kurios CRUD santykis su funkcija yra fiksuojamas, id.	Pasirinkta <i>EsID</i> reikšmė iš lentelės <i>Esybe</i> . - jeigu <i>EsID</i> ≠ Null, tai <i>AtrID</i> = Null; - jeigu <i>EsID</i> = Null, tai <i>AtrID</i> ≠ Null.		num(10)
<i>AtrID</i>	Atributo, kurio CRUD santykis su funkcija yra fiksuojamas, id.	Pasirinkta <i>AtrID</i> reikšmė iš lentelės <i>Atributas</i> . - jeigu <i>AtrID</i> ≠ Null, tai <i>EsID</i> = Null; - jeigu <i>AtrID</i> = Null, tai <i>EsID</i> ≠ Null.		num(10)

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<i>CRUDReiksme</i>	CRUD santykių apibrėžianti sutartinio formato išraiška.	Tekstinė išraiška, susidedanti iš nesikartojančių raidžių “C”, “R”, “U”, “D”, atskirtų kableliu. Įtraukiamos tos raidės, kurios atspindi santykio esmę.	✓	chr(7)

3.3.4 Veiklos taisyklių šablonų posistemė

VTTipas. Veiklos taisyklių tipai (pagal BRS RuleSpeak modelį).

Pastaba: vieno tipo taisyklėms užrašyti gali būti naudojama vienas ar daugiau šablonų.

20 lentelė

DB lentelės VTTipas aprašas.

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<i>VTTID</i>	Unikalus VT tipo identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonom
<i>Pavadinimas</i>	VT tipo pavadinimas.	Sutartiniai <i>BRS RuleSpeak</i> modelyje numatytų VT tipų pavadinimai ir kiti pavadinimai, jeigu įvedami nauji VT tipai.	✓	chr(30)
<i>Apibrėžimas</i>	VT tipo apibrėžimas.	Bet koks tekstas (esamiems <i>BRS RuleSpeak</i> VT tipams gali būti naudojami esamų apibrėžimų vertimai).	✓	chr(200)
Vienas-su-daug ryšys su <esybė>.<atributas>		Reikšmė	Būtin as	
Sablonas.VTTID	VT šablonų, naudojamų duoto tipo taisyklėms užrašyti, aibė.		✓	

Sablonas. Duomenys apie veiklos taisyklių užrašymo natūralia (lietuvių) kalba šablonus (pagal BRS RuleSpeak modelį).

21 lentelė

DB lentelės Sablonas aprašas.

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<i>SbUID</i>	Unikalus VT šablono identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonom
<i>Pavadinimas</i>	VT šablono pavadinimas.	Sutartiniai <i>BRS RuleSpeak</i> modelyje numatytų VT šablonų pavadinimai ir kiti pavadinimai, jeigu įvedami nauji VT šablonai. Šablonų pavadinimus rekomenduojama konstruoti iš šablone užduotų esminių žodžių ir raktažodžių.	✓	chr(40)

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<i>Aprašas</i>	VT šablono aprašymas, paaiškinimas ar laisvo pobūdžio komentarai.	Bet koks tekstas.		chr(200)
<i>VTTID</i>	VT tipo, kurio taisyklėms užrašyti skirtas šablonas, id.	Pasirinkta <i>VTTID</i> reikšmė iš lentelės <i>VTTipas</i> .	✓	num(10)
<i>KeitimoData</i>	Šablono įvedimo į saugyklą arba paskutinio bent vienos šablono charakteristikos redagavimo data. Data keičiasi ir pasikeitus su šablonu susijusių komponentų charakteristikoms (įskaitant ir raktinių žodžių pasikeitimus), pasikeitus šablono subjektų tipams. Taip pat nurodo, ar šablonas nėra pašalintas (praradęs aktualumą).	Data ir tikslus laikas. Šalinant šabloną, nustatoma reikšmė "1111.11.11"; kitos charakteristikos nepakinta.	✓	datetime
Vienas-su-daug ryšys su <esybė>.<atributas>				
		Reikšmė	Būtin as	
<i>SubjektoTipas.SblID</i>		Nuorodų į šablone leistinus subjekto tipus aibė.	✓	
<i>SablonoElementas.SblID</i>		Šabloną sudarančių elementų aibė.	✓	
<i>VeiklosTaisykle.SblID</i>		Pagal šabloną užrašytų veiklos taisyklių aibė.		

SblElementoTipas. Duomenys apie veiklos taisyklių užrašymo šablonuose naudojamų galimų komponentų tipus.

22 lentelė

DB lentelės *SblElementoTipas* aprašas.

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<i>EITID</i>	Unikalus VT šablono elemento tipo identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonum
<i>Pavadinimas</i>	VT šablono elemento tipo pavadinimas.	<i>BRS RuleSpeak</i> modelyje numatytų VT šablonų elementų/komponentų pavadinimai ir kiti pavadinimai, jeigu kuriami nauji šablonai, kuriuose įvedami nauji elementai. Pradinė reikšmių aibė: {"užduotas tekstas", "raktažodis", "ryšys", "esybė", "atributas", "reikšmė", "kitoks terminas", "VT vardas", "duomenų vienetas", "mat. formulė", "loginė išraiška", "sąlyga", "prielaida", "išvedimo būdas", "išvedimo forma"}.	✓	chr(20)
<i>SubjPozymis</i>	Požymis, nurodantis, ar elemento tipas gali būti	Loginė reikšmė: - <i>taip (true)</i> – subjektia gali būti	✓	boolean

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
	taikomas veiklos taisyklės subjektui, t.y. ar taisyklės subjektas gali būti šio tipo.	šio tipo; - <i>ne (false)</i> – subjektai negali būti šio tipo.		
Vienas-su-daug ryšys su <esybė>.<atributas>	Reikšmė			Būtin as
VeiklosTaisykle.SubjTipoID	Veiklos taisyklių, kurių subjektas yra šio tipo, aibė.			
SusijesModelioElementas.ElTID	Nuorodų į su veiklos taisyklių komponentais susijusius modelio elementus aibė.			✓
SubjektoTipas.ElTID	Nuorodų į šablonus, kuriuose subjektu gali būti šio tipo elementas, aibė.			
VTIsraiskosKomponentas.ElTID	Nustatytų veiklos taisyklių komponentų, priklausančių šiam tipui, aibė.			
GalimasElementoTipas.ElTID	Nuorodų į VT šablonų elementus, kurie yra šio tipo, aibė.			✓

SubjektoTipas. Sąryšio daug-su-daug tarp lentelių Sablonas ir SblElementoTipas pagalbinė lentelė. Kiekvienas VT šablonas turi subjektą, o galimi subjekto tipai yra apibrėžti BRS RuleSpeak (ar kitaip, jeigu naudojami papildomi ar kiti šablonai). Tokių tipų gali būti keletas. To paties tipo subjektai gali būti naudojami daugiau nei viename šablone.

23 lentelė

DB lentelės SubjektoTipas aprašas.

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<i>ElTID</i>	Šablonų elemento tipo identifikacinis numeris.	Pasirinkta <i>ElTID</i> reikšmė iš lentelės <i>SblElementoTipas</i> .	✓	num(10)
<i>SblID</i>	VT šablono identifikacinis numeris.	Pasirinkta <i>SblID</i> reikšmė iš lentelės <i>Sablonas</i> .	✓	num(10)

SablonoElementas. Duomenys apie veiklos taisyklių užrašymo šablonus sudarančias sudedamąsias dalis – elementus.

24 lentelė

DB lentelės SablonoElementas aprašas.

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<i>SblElID</i>	Unikalus VT šablono elemento identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonum
<i>Butinumas</i>	Požymis, nurodantis, ar šis šablono elementas yra privalomas užrašant VT.	Taip (<i>true</i> , 1) arba Ne (<i>false</i> , 0)	✓	boolean
<i>EilNr</i>	Šablono elemento vieta šablone (vis šablono elementai bus numeruojami nuo 1).	Sveikas skaičius.	✓	num(2)
<i>RTekstID</i>	Šablone naudojamo rezervuoto teksto ID.	Pasirinkta <i>RTekstID</i> reikšmė iš lentelės <i>RezervuotasTekstas</i> .		num(10)

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<i>SblID</i>	VT šablono, kurio dalis yra elementas, identifikacinis numeris.	Pasirinkta <i>SblID</i> reikšmė iš lentelės <i>Sablonas</i> .	✓	num(10)
<i>PapildomaIsraiska</i>	Papildoma šablono elemento tekstinė išraiška, jeigu tokia yra.	Tekstas, nurodytas BRS RuleSpeak ir bet koks kitoks tekstas, jeigu įvedamas naujas šablonas. Gali egzistuoti ir jeigu išraiškos nėra. Galimi variantai atskiriami simboliu “/”.		chr(20)
<i>PapildIsrPozicija</i>	Papildomos išraiškos pozicija.	Pagrindinės galimos reikšmės (ne baigtinė aibė): - 0 – prieš pagrindinę išraišką; - 1 – po pagrindinės išraiškos.		Num(1)
<i>PapildIsrButinumas</i>	Požymis, nurodantis, ar papildoma išraiška yra privaloma užrašant VT.	Taip (<i>true</i> , 1) arba Ne (<i>false</i> , 0)		boolean
Vienas-su-daug ryšys su <esybė>.<atributas>				
		Reikšmė	Būtin as	
<i>SblElementoRaktazodis</i> . <i>SblEIID</i>	Nuorodų į šablono elemente, jeigu jis yra <i>Keyword</i> tipo (raktinis žodis arba raktinė išraiška), galimus raktažodžius aibė. Jeigu šablono elementas yra raktažodis, gali būti, kad jis nebus apibrėžtas o pasirenkamas iš dviejų ar daugiau galimybių, pvz.: <i>igalinti / išjungti</i> .			
<i>GalimasElementoTipas</i> . <i>EITID</i>	Nuorodų į šablono elementų tipus, kurie yra galimi (kaip alternatyvos) šio elemento tipai, aibė.		✓	

GalimasElementoTipas. Duomenys VT šablono elemente galimą elemento tipą. Pvz.: šablone (elementai atskirti “||”)

<Term> || must/should [not] || BE SET to|| <term/value> || [when/if <condition>]

ketvirtas elementas (<term/value>) gali būti arba terminas, arba reikšmė.

25 lentelė

DB lentelės *GalimasElementoTipas* aprašas.

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<i>SblEIID</i>	Šablono elemento identifikacinis numeris.	Pasirinkta <i>SblEIID</i> reikšmė iš lentelės <i>SablonoElementas</i> .	✓	num(10)
<i>EITID</i>	Šablonų elemento tipo identifikacinis numeris.	Pasirinkta <i>EITID</i> reikšmė iš lentelės <i>SblElementoTipas</i> .	✓	num(10)

RezervuotasTekstas. Duomenys VT šablonuose galimus raktinius žodžius, raktines išraiškas ir iš anksto nustatytas tekstines išraiškas.

26 lentelė

DB lentelės *RezervuotasTekstas* aprašas.

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
-----------	---------	------------------	----------	-------

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<i>RTekstID</i>	Unikalus rezervuoto teksto identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonom
<i>Israiska</i>	Raktinis žodis arba žodžių junginys.	Tekstas, nurodytas BRS RuleSpeak ir bet koks kitoks tekstas, jeigu įvedamas naujas šablonas. Galimi variantai atskiriami simboliu “/”.	✓	chr(100)
<i>RaktPozymis</i>	Požymis, nurodantis, ar išraiška yra raktažodis, t.y. žodis ar žodžių junginys, kuris unikalus kiekvienam veiklos taisyklių tipui.	Loginė reikšmė: - <i>taip (true)</i> – išraiška yra raktažodis; - <i>ne (false)</i> – išraiška nėra raktažodis.	✓	boolean
Vienas-su-daug ryšys su <esybė>.<atributas>				
		Reikšmė	Būtin as	
SablonoElementas. RTekstID		Nuorodų į šablonų elementus, kuriuose gali būti naudojamas raktinis žodis ar kitokia užduota tekstinė išraiška, aibė.	✓	

3.3.5 Nestruktūrinių veiklos taisyklių posistemė

VeiklosTaisykle. Pagrindiniai duomenys apie nustatytas veiklos taisykles.

27 lentelė

DB lentelės VeiklosTaisykle aprašas.

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<i>VTID</i>	Unikalus VT identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonom
<i>Pavadinimas</i>	Trumpas veiklos taisyklės pavadinimas.	Bet koks tekstas.		chr(30)
<i>Kodas</i>	Sutartinis VT kodas, naudojamas sudarant VT grindžiamus modelius.	Skaitmeninė išraiška, sudaryta pagal nustatyta metodą, be tarpų. Galimas variantas – aptikimo eilės numeris.	✓	num(4)
<i>Israiska</i>	Visa veiklos taisyklė, užrašyta pagal šabloną. Saugoma, kad nereikėtų kaskart “perskaičiuoti” VT išraiškos pagal šabloną ir saugomus komponentus.	Bet koks tekstas.	✓	chr(500)
<i>SubjID</i>	Subjekto id.	Galimi keturi variantai: 1. pasirinkta <i>EsID</i> reikšmė iš lentelės <i>Esybe</i> ; 2. pasirinkta <i>AtrID</i> reikšmė iš lentelės <i>Atributas</i> ; 3. pasirinkta <i>RysID</i> reikšmė iš lentelės <i>Rsys</i> ; 4. pasirinkta <i>RTID</i> iš lentelės	✓	num(10)

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas															
		<i>ReiksmeTerminas;</i> 5. pasirinkta <i>VTID</i> reikšmė iš jau esančių lentelėje <i>VeiklosTaisykle</i> .																	
<i>SubjTipID</i>	VT subjekto tipo id.	Pasirinkta <i>ELITID</i> reikšmė iš lentelės <i>SblElementuTipai</i> .	✓	num(10)															
<i>SubjIsraiska</i>	VT subjekto išraiška, kokia ji naudojama VT išraiškoje. Pildoma net jeigu išraiška sutampa su esybės/atributo/ryšio pavadinimu.	Bet koks tekstas.	✓	chr(50)															
<i>SblID</i>	Šablono, pagal kurį užrašyta VT, id.	Pasirinkta <i>SblID</i> reikšmė iš lentelės <i>Sablonas</i> .	✓	num(10)															
<i>KeitimoData</i>	VT įvedimo į saugyklą arba paskutinio bent vienos VT charakteristikos redagavimo data. Taip pat nurodo, ar VT nėra pašalinta (praradus aktualumą).	Data ir tikslus laikas. Šalinant VT, nustatoma reikšmė "1111.11.11"; kitos charakteristikos nepakinta.	✓	datetime															
<table border="1"> <thead> <tr> <th>Vienas-su-daug ryšys su <esybė>.<atributas></th> <th>Reikšmė</th> <th>Būtin as</th> </tr> </thead> <tbody> <tr> <td>VTIsraiskosKomponentas.VTID</td> <td>Komponentų, iš kurių susideda VT, aibė (be subjekto).</td> <td>✓</td> </tr> <tr> <td>VeiklosTaisykle.SubjID</td> <td>Veiklos taisyklių, kurių subjektas yra ši VT, aibė.</td> <td></td> </tr> <tr> <td>SprendimoVT.VTID</td> <td>Nuorodų į funkcijoms priskirtus sprendimus, nagrinėjant kuriuos buvo nustatyta ši taisyklė, aibė.</td> <td>✓</td> </tr> <tr> <td>VTSaltinis.VTID</td> <td>Duomenų šaltinių, su kuriais siejama VT, aibė.</td> <td></td> </tr> </tbody> </table>					Vienas-su-daug ryšys su <esybė>.<atributas>	Reikšmė	Būtin as	VTIsraiskosKomponentas.VTID	Komponentų, iš kurių susideda VT, aibė (be subjekto).	✓	VeiklosTaisykle.SubjID	Veiklos taisyklių, kurių subjektas yra ši VT, aibė.		SprendimoVT.VTID	Nuorodų į funkcijoms priskirtus sprendimus, nagrinėjant kuriuos buvo nustatyta ši taisyklė, aibė.	✓	VTSaltinis.VTID	Duomenų šaltinių, su kuriais siejama VT, aibė.	
Vienas-su-daug ryšys su <esybė>.<atributas>	Reikšmė	Būtin as																	
VTIsraiskosKomponentas.VTID	Komponentų, iš kurių susideda VT, aibė (be subjekto).	✓																	
VeiklosTaisykle.SubjID	Veiklos taisyklių, kurių subjektas yra ši VT, aibė.																		
SprendimoVT.VTID	Nuorodų į funkcijoms priskirtus sprendimus, nagrinėjant kuriuos buvo nustatyta ši taisyklė, aibė.	✓																	
VTSaltinis.VTID	Duomenų šaltinių, su kuriais siejama VT, aibė.																		

VTIsraiskosKomponentas. Duomenys apie nustatytų konkrečių veiklos taisyklių atskirus komponentus. Kiekviena veiklos taisyklė saugoma dalimis.

28 lentelė

DB lentelės VTIsraiskosKomponentas aprašas.

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<i>VTIsrKompID</i>	Unikalus VT išraiškos komponento identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonum
<i>VTID</i>	Veiklos taisyklės, kuriai priklauso komponentas, id.	pasirinkta <i>VTID</i> reikšmė iš jau esančių lentelėje <i>VeiklosTaisykle</i> .	✓	num(10)
<i>ELITID</i>	Komponento tipo id.	Pasirinkta <i>ELITID</i> reikšmė iš lentelės <i>SblElementuTipai</i> .	✓	num(10)
<i>SekosNr</i>	Komponento eilės numeris komponentų išdėstymo VT išraiškoje sekoje.	Rekomenduojamas skaitmuo nuo 1 iki 9, nes ilgesni nei 9 komponentų šablonai nebus naudojami.	✓	num(1)
<i>Israiska</i>	Komponento pagrindinė išraiška.	Bet koks tekstas.	✓	chr(200)
<i>PapildomaIsraiska</i>	Komponento papildoma išraiška.	vienas iš teksto variantų, kurie numatyti naudojamame šablone šio komponento sudėtyje.		chr(20)

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
Vienas-su-daug ryšys su <esybė>.<atributas>	Reikšmė			Būtin as
SusijęsModelioElementas. VTIsrKompID	Nuorodų į VT komponente naudojamus poreikių modelio elementus aibė. Ši aibė nebus tuščia tik kuomet komponento tipas yra vienas iš: <i>ryšys</i> , <i>esybė</i> , <i>atributas</i> , <i>reikšmė</i> , <i>taisyklės pavadinimas</i> , <i>duomenų vienetas</i> , <i>matematinė formulė</i> , <i>sąlyga</i> , <i>prielaida</i> .			
SprendimuLentele. VTIsrKompID	Sprendimų lentelių, į kurias nurodoma VT komponente, aibė. Aibė turės >1 narį tik tuomet, kai sprendimų lentelė bus sudaroma pagal tris charakteristikas (t.y. bus sudaromas sprendimų lentelių masyvas).			

SusijęsModelioElementas. Sąryšio daug-su-daug tarp lentelės VTIsraiskosKomponentas ir lentelių Esybe/Atributas/Rysys pagalbinė lentelė. Kivkiename VT komponente gali būti naudojami esybių, atributų ar ryšių pavadinimai.

29 lentelė

DB lentelės SusijęsModelioElementas aprašas.

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
SModEIID	Unikalus įrašo identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonum
<i>ElementoID</i>	VT komponento išraiškoje naudojamo ar kitaip su taisykle susijusio komponento id.	Galimi keturi variantai: 1.pasirinkta <i>EsID</i> reikšmė iš lentelės <i>Esybe</i> ; 2.pasirinkta <i>AtrID</i> reikšmė iš lentelės <i>Atributas</i> ; 3.pasirinkta <i>RysID</i> reikšmė iš lentelės <i>Rysys</i> ; 4.pasirinkta <i>RTID</i> iš lentelės <i>ReiksmeTerminas</i> .	✓	num(10)
VTIsrKompID	VT komponento id.	Pasirinkta <i>VTIsrKompID</i> reikšmė iš lentelės <i>VTIsraiskosKomponentas</i> .	✓	num(10)
<i>EITID</i>	susijusio elemento tipo id.	Pasirinkta <i>EITID</i> reikšmė iš lentelės <i>SblElementoTipas</i> . Galimas pasirinkimas: “ryšys”, “esybė”, “atributas”, “reikšmė”.	✓	num(10)

SprendimuLentele. Pagrindiniai duomenys apie nustatytoje veiklos taisyklėse naudojamas sprendimų lenteles (kitais tariant, apie lentelių antraštes). Sprendimų lentelė – tai tam tikros charakteristikos reikšmės kitimo priklausomai nuo 1-3 kitų charakteristikų atvaizdavimo forma. Sprendimų lentelės, kuriose reikšmė nustatoma pagal tris charakteristikas, vadinamos sprendimų lentelių masyvu.

DB lentelės SprendimuLentele aprašas.

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
SprLentID	Unikalus sprendimų lentelės identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonom
<i>Pavadinimas</i>	Sprendimų lentelės trumpas pavadinimas arba kodas.	Bet koks tekstas.	✓	chr(10)
<i>EilutesChVardas</i>	Sprendimų lentelės eilutės charakteristikos pavadinimas (bus visada).	Bet koks tekstas.	✓	chr(50)
<i>StulpelioChVardas</i>	Sprendimų lentelės stulpelio charakteristikos pavadinimas (tik jeigu sprendimų lentelė yra dviejų arba trijų charakteristikų)	Bet koks tekstas.		chr(50)
<i>LentelesChVardas</i>	Sprendimų lentelės bendros charakteristikos pavadinimas (tik jeigu yra sprendimų lentelių masyvas).	Bet koks tekstas.		chr(50)
<i>LentReiksme</i>	Sprendimų lentelės bendros charakteristikos reikšmė.	Bet koks tekstas.		chr(50)
<i>MatavimoVnt</i>	Sprendimų lentelės bendros reikšmės matavimo vienetas (gali būti ir, tarkim, loginis – true, false).	Bet koks tekstas.		chr(20)
VTIsrKompID	VT komponento, kuriame naudojama nuoroda į sprendimų lentelę, id.	Pasirinkta <i>VTIsrKompID</i> reikšmė iš lentelės <i>VTIsraiskosKomponentas</i> .	✓	num(10)
Vienas-su-daug ryšys su <esybė>.<atributas>				
		Reikšmė	Būtin as	
SIEilutesReiksme.SprLentID	Sprendimų lentelės eilučių reikšmių aibė.		✓	
SISulpelioReiksme.Sp rLentID	Sprendimų lentelės stulpelių reikšmių aibė.			

SLEilutesReiksme. Duomenys apie sprendimo lentelių eilučių reikšmes (t.y. reikšmes, esančias eilučių pradžioje; pirmosios charakteristikos reikšmes).

DB lentelės SLEilutesReiksme aprašas.

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
EiIRID	Unikalus sprendimų lentelės eilutės reikšmės (arba tiesiog eilutės) identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonom
<i>Reiksme</i>	Eilutės reikšmė.	Bet koks tekstas.	✓	chr(30)
<i>MatavimoVnt</i>	Eilutės reikšmės matavimo vienetas (gali būti ir, tarkim, loginis – true, false).	Bet koks tekstas.		chr(10)

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<i>SprLentID</i>	Sprendimų lentelės, kuriai priklauso eilutės reikšmė, id.	Pasirinkta <i>SprLentID</i> reikšmė iš lentelės <i>SprendimuLentele</i> .	✓	num(10)
Vienas-su-daug ryšys su <esybė>.<atributas>		Reikšmė	Būtin as	
SIReiksme.EilRID	Pagal sprendimų lentelę nustatomų reikšmių, kurios bus įgaunamos, kuomet yra ši konkreti eilutės reikšmė, aibė.		✓	

SLStulpelioReiksme. Duomenys apie sprendimo lentelių stulpelių reikšmes (t.y. reikšmes, esančias stulpelių viršuje; antrosios charakteristikos reikšmes).

32 lentelė

DB lentelės SLStulpelioReiksme aprašas.

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<i>StulpRID</i>	Unikalus sprendimų lentelės stulpelio reikšmės (arba tiesiog stulpelio) identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonom
<i>Reiksme</i>	Stulpelio reikšmė.	Bet koks tekstas.	✓	chr(30)
<i>MatavimoVnt</i>	Stulpelio reikšmės matavimo vienetas (gali būti ir, tarkim, loginis – true, false).	Bet koks tekstas.		chr(10)
<i>SprLentID</i>	Sprendimų lentelės, kuriai priklauso stulpelio reikšmė, id.	Pasirinkta <i>SprLentID</i> reikšmė iš lentelės <i>SprendimuLentele</i> .	✓	num(10)
Vienas-su-daug ryšys su <esybė>.<atributas>		Reikšmė	Būtin as	
SIReiksme.StulpRID	Pagal sprendimų lentelę nustatomų reikšmių, kurios bus įgaunamos, kuomet yra ši konkreti stulpelio reikšmė, aibė.		✓	

SLReiksme. Duomenys apie sprendimo lentelių eilučių ir stulpelių susikirtime esančias reikšmes (t.y. reikšmes, kurioms nustatyti sudaroma sprendimų lentelė).

33 lentelė

DB lentelės SLReiksme aprašas.

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<i>EilRID</i>	Sprendimų lentelės stulpelio reikšmės id.	Pasirinkta <i>EilRID</i> reikšmė iš lentelės <i>SLStulpelioReiksme</i> .	✓	num(10)
<i>StulpRID</i>	Sprendimų lentelės eilutės reikšmės id.	Pasirinkta <i>StulpRID</i> reikšmė iš lentelės <i>SLStulpelioReiksme</i> .	✓	num(10)
<i>Reiksme</i>	Sprendimo lentelės eilutės ir stulpelio susikirtime esanti reikšmė.	Bet koks tekstas.	✓	chr(30)
<i>MatavimoVnt</i>	Reikšmės matavimo vienetas.	Bet koks tekstas.		chr(10)

VTSaltinis. Sąryšio daug-su-daug tarp lentelės VeiklosTaisykle ir lentelių Asmuo/Dokumentas/PrograminisKodas pagalbinė lentelė. Kiekviena VT gali turėti kelis kilmės šaltinius, o kiekvienas iš šių šaltinių gali būti siejamas su viena ar daugiau VT.

34 lentelė

DB lentelės VTSaltinis aprašas.

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<i>VTSaltID</i>	Unikalus VT šaltinio identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	autonom
<i>SaltinioID</i>	Veiklos taisyklės šaltinio (programinio kodo/elemento, dokumento arba aktoiaus) id.	Galimi trys variantai: 1. pasirinkta <i>AsmID</i> reikšmė iš lentelės <i>Asmuo</i> ; 2. pasirinkta <i>PKID</i> reikšmė iš lentelės <i>PrograminisKodas</i> ; 3. pasirinkta <i>DokID</i> reikšmė iš lentelės <i>Dokumentas</i> .	✓	num(10)
<i>VTID</i>	Veiklos taisyklės id.	Pasirinkta <i>VTID</i> reikšmė iš lentelės <i>VeiklosTaisykle</i> .	✓	num(10)
<i>SaltinioTipas</i>	Šaltinio tipo kodas.	Pasirenkamas vienos raidės kodas. Galimos reikšmės: - "A" – aktorius; - "D" – dokumentas; - "P" – programinis kodas/elementas.		chr(1)

SprendimoVT. Sąryšio daug-su-daug tarp lentelių FunkcijaSprendimas ir VeiklosTaisykle pagalbinė lentelė. Ta pati VT gali būti nustatyta nagrinėjant daugiau nei vieną sprendimą, o kiekvienas sprendimas gali sąlygoti vienos ar daugiau VT nustatymą.

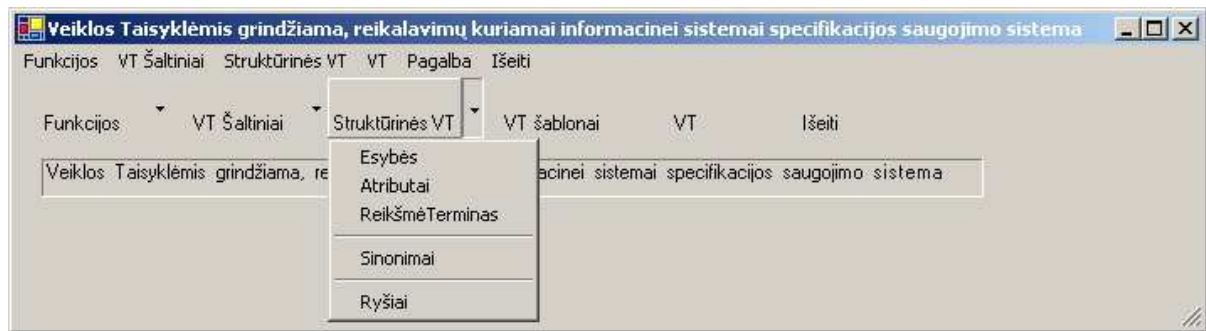
35 lentelė

DB lentelės SprendimoVT aprašas.

Atributas	Aprašas	Galimos reikšmės	Būtin as	Tipas
<i>SprID</i>	Sprendimo id.	Pasirinkta <i>SprID</i> reikšmė iš lentelės <i>Sprendimas</i> .	✓	num(10)
<i>VTID</i>	Veiklos taisyklės id.	Pasirinkta <i>VTID</i> reikšmė iš lentelės <i>VeiklosTaisykle</i> .	✓	num(10)
<i>SekosNr</i>	VT sekos numeris, jeigu konkrečiam sprendimui pasiekti taisyklės reikia vykdyti tam tikra eilės seka.	Sveikas skaičius nuo 1 iki 99		num(2)

3.4 Realizacijos langų architektūra

Šiame poskyryje pateikiamos eksperimentinės informacinės sistemos langai, kuriuos reikės paruošti įgyvendinant veiklos taisyklėmis grindžiamos reikalavimų kuriamai Informacinei Sistemai specifikacijos saugojimo sistemą.

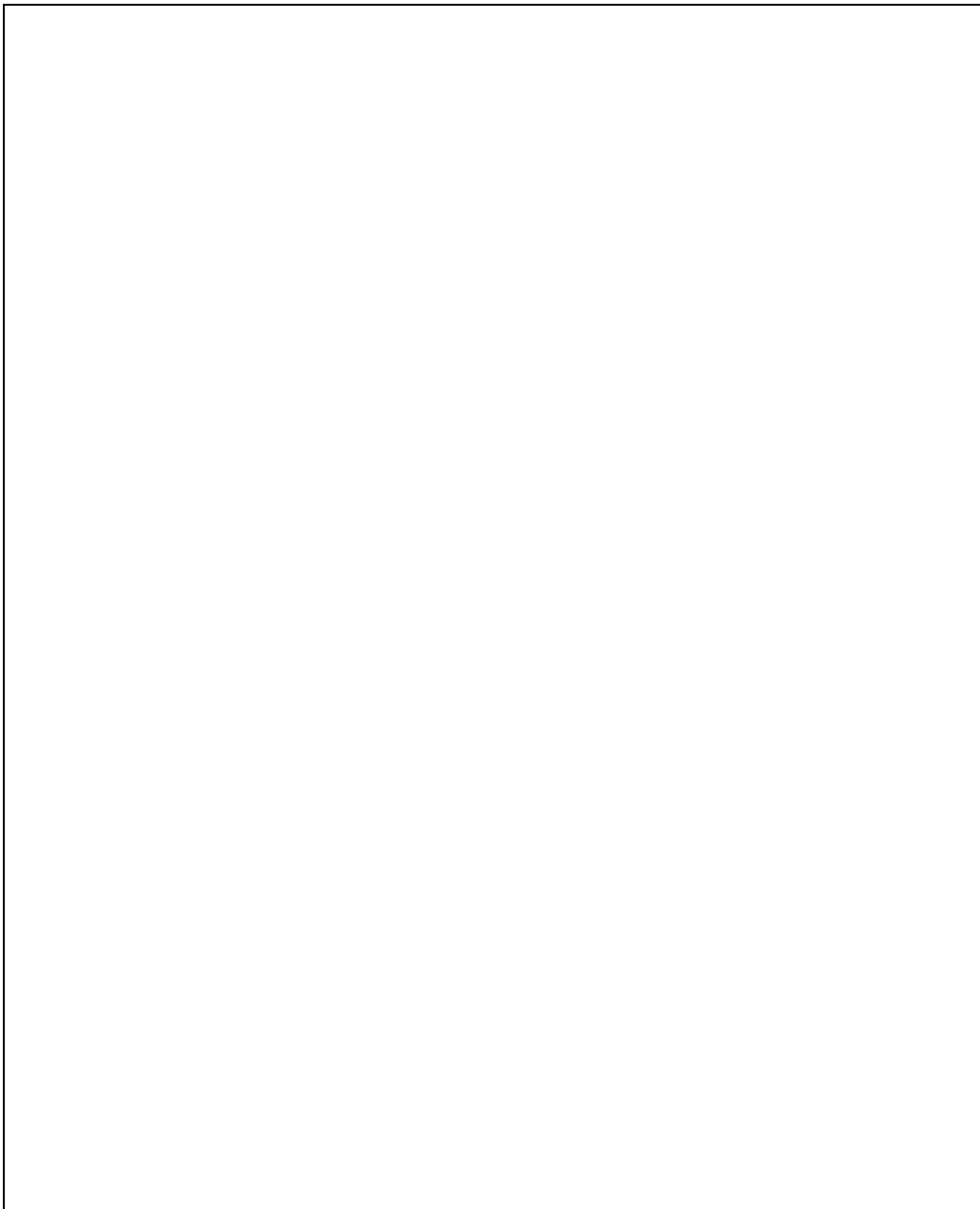


Šaltinis: sukurta autoriaus.

16 pav. Startinis saugyklos langas.

16 paveiksle matome startinį sistemos langą skirtą navigacijai IS viduje. Iš jo paleidžiami kiti langai, taip pat suteikiama galimybė iš karto pereiti į reikalingą atidaromo lango puslapį.

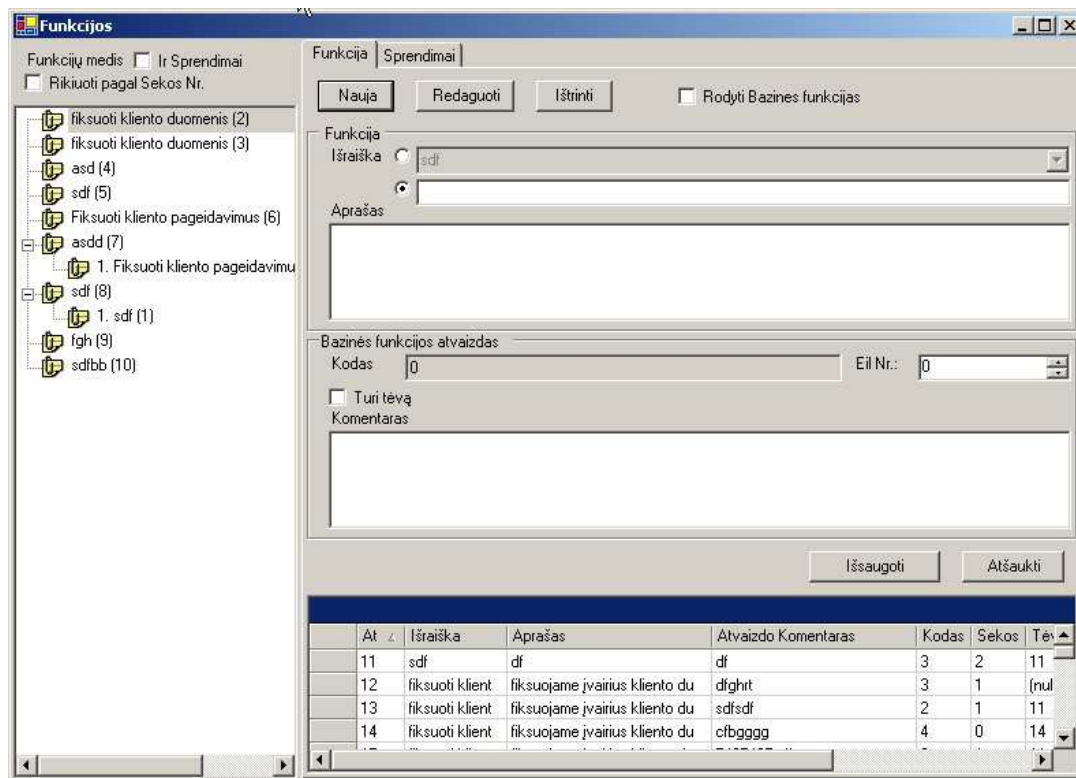
17 paveiksle parodyta funkcijų hierarchijos fiksavimo veiklų diagrama. Didžioji dalis veiklų atliekama 18 paveiksle matomame lange, funkcijų susiejimas su aktoriais atliekamas 26 paveiksle parodytame lange.



Šaltinis:

17 pav. Funkcijų nustatymo veiklų diagrama.

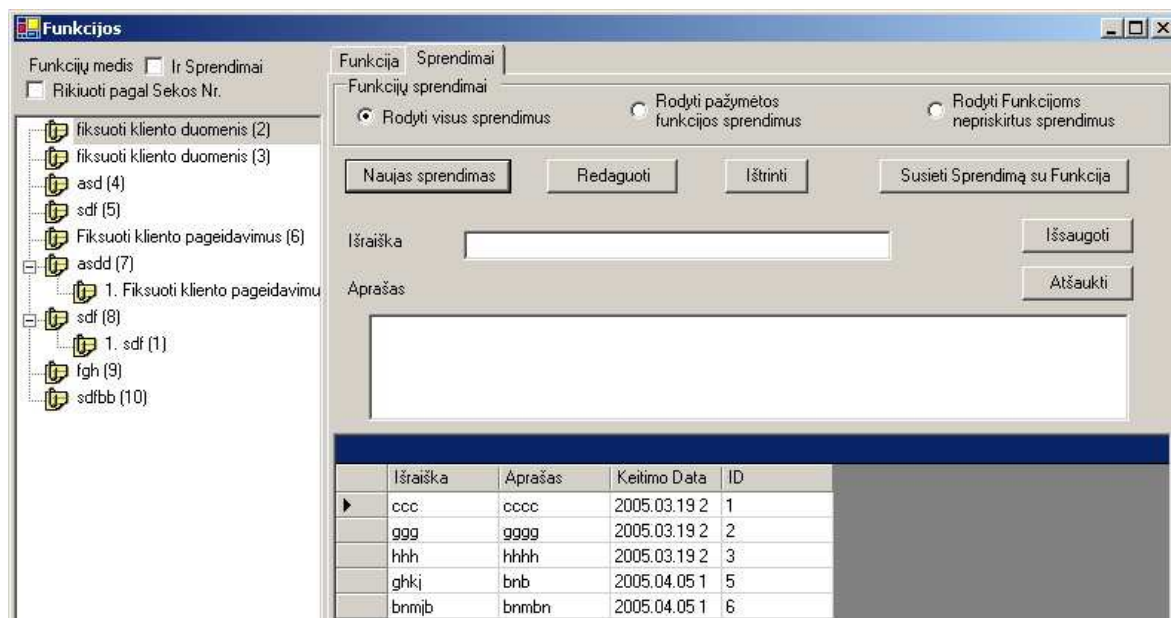
Kaip matome 18 paveiksle, funkcijų hierarchijos įvedimo lange panaudotas standartinis programinis vartotojo sąsajos komponentas treeview gerai tinkantis hierarchinių struktūrų atvaizdavimui. To pasėkoje analitikas funkcijų hierarchiją mato iš karto, nereikalingas dažnas funkcijų hierarchijos diagramų spausdinimas.



Šaltinis: sukurta autoriaus.

18 pav. Funkcijų hierarchijos duomenų įvedimo langas.

19 paveiksle matomas kitas funkcijų hierarchijos formavimo lango puslapis, kuriame galima suvesti funkcijas atliekant vykdomus sprendimus.

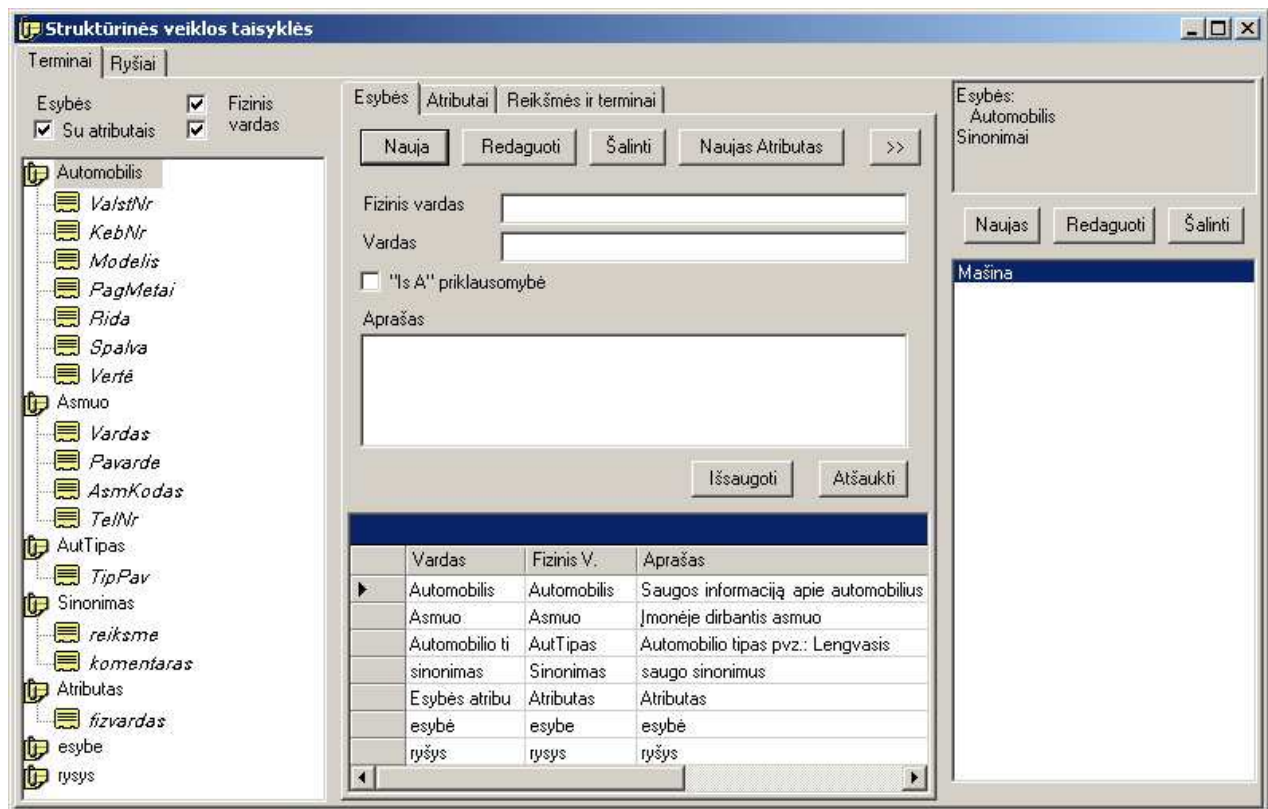


Šaltinis: sukurta autoriaus.

19 pav. Sprendimų įvedimo langas.

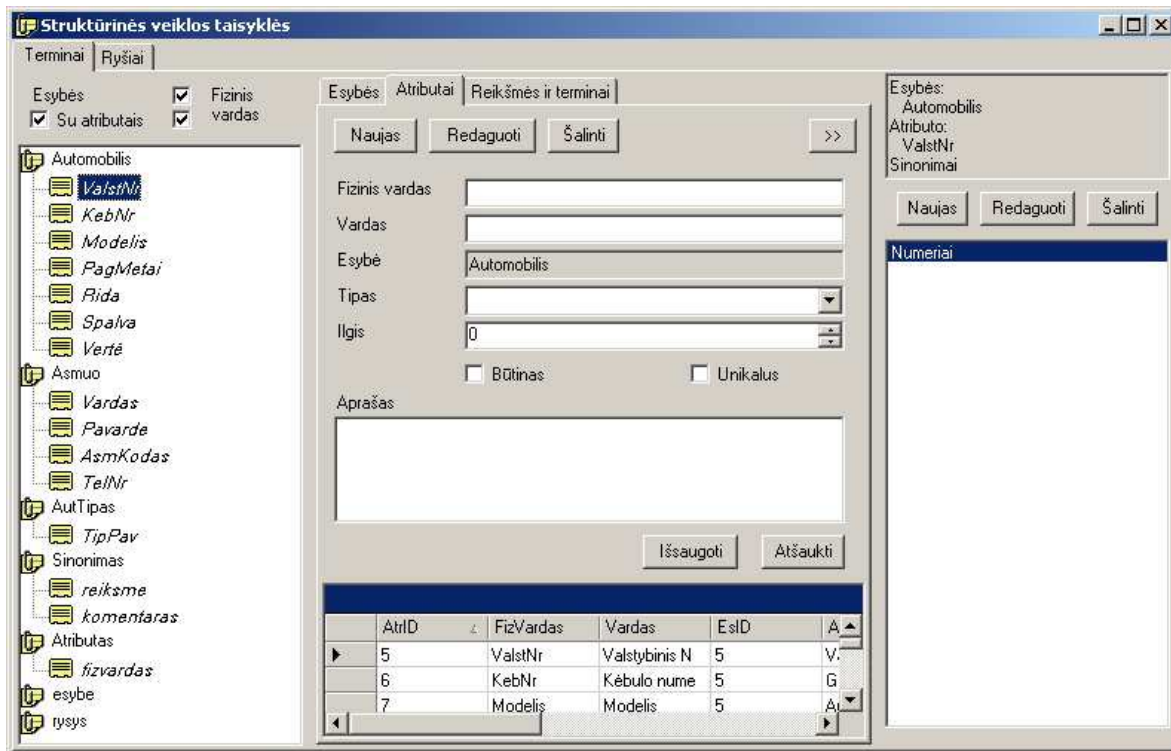
Sprendimai priskiriami funkcijų medyje pažymėtai funkcijai.

20, 21, 22 ir 23 paveiksluose matoma koncepcinio duomenų modelio (ER) informacijos suvedimo langai. Juose galima suvesti esybes ir jų atributus bei ryšius tarp esybių.



Šaltinis: sukurta autoriaus.

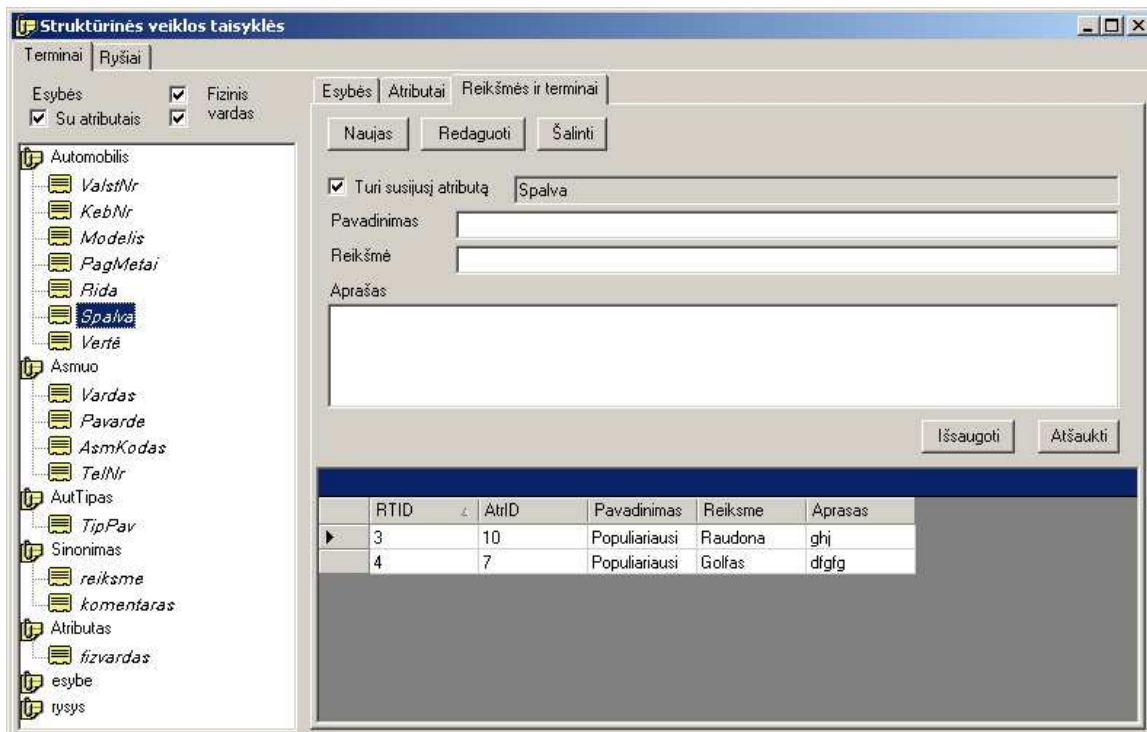
20 pav. Esybių ir jų sinonimų įvedimo langas.



Šaltinis: sukurta autoriaus.

21 pav. Esybių atributų ir jų sinonimų įvedimo langas.

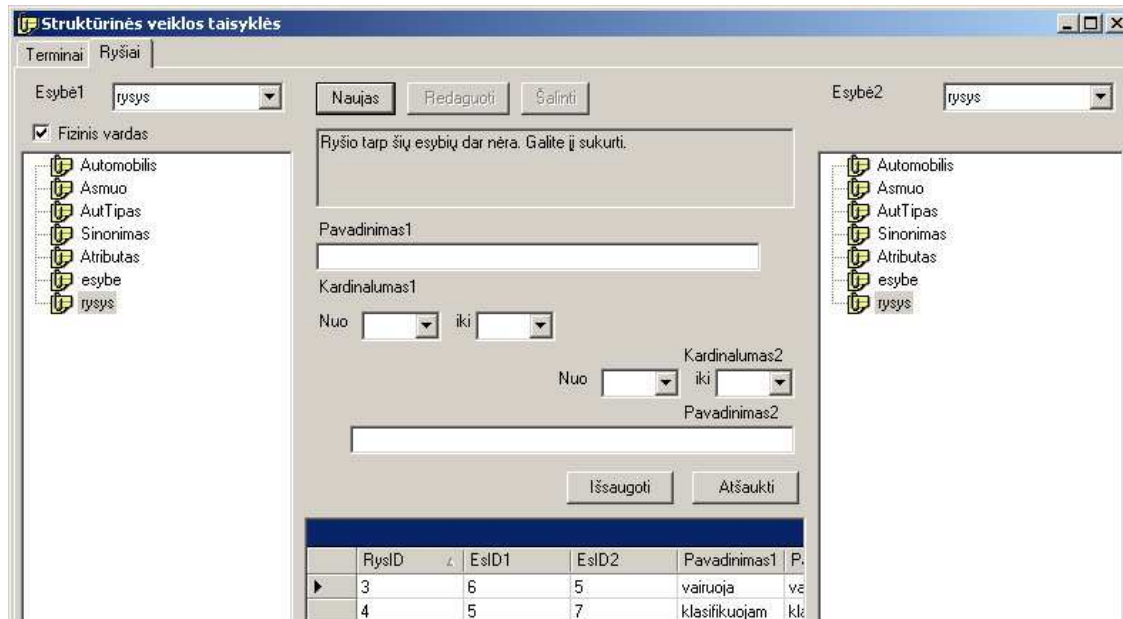
Kaip matome iš 20, 21 paveikslų esybės ir atributai gali turėti sinonimų.



Šaltinis: sukurta autoriaus.

22 pav. Reikšmių ir terminų įvedimo langas.

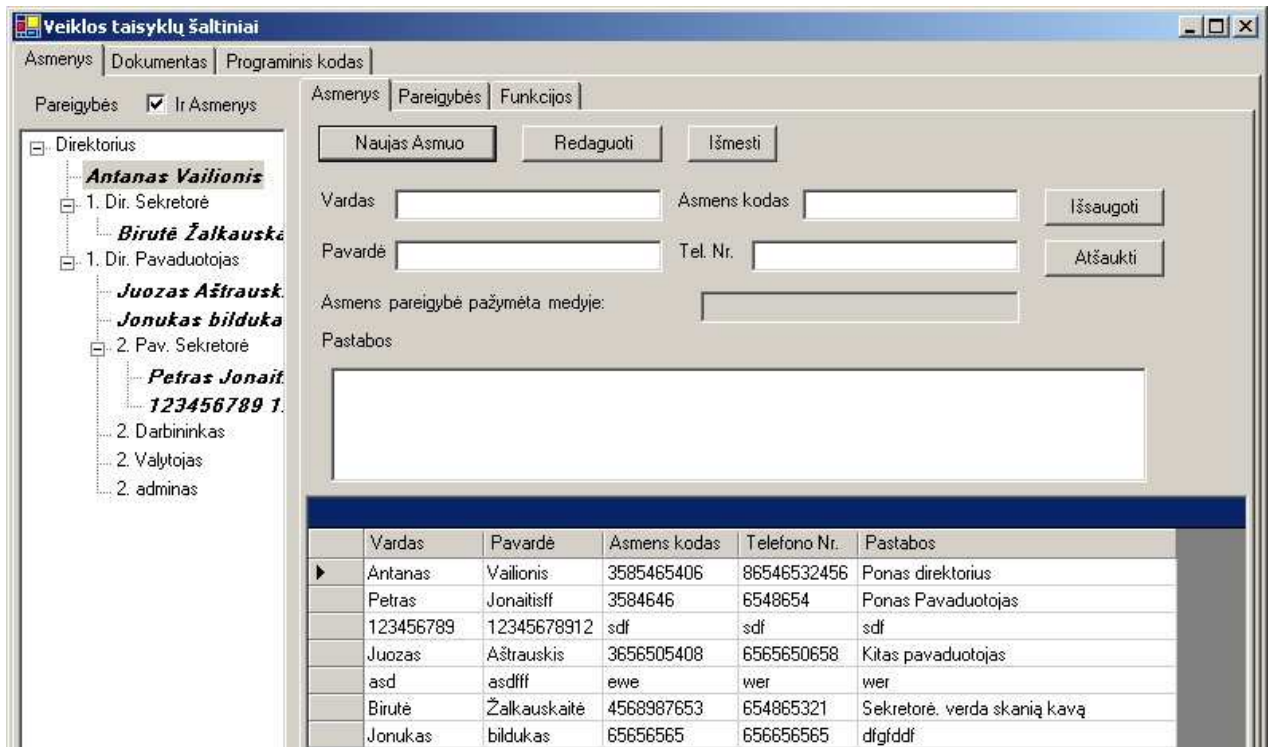
22 paveiksle suteikiama galimybė suvesti konstantas, terminus, atributų reikšmių aibes. Ši informacija nors ir nepriklauso duomenų modeliui tačiau ji gali būti reikalinga specifikuojant veiklos taisykles.



Šaltinis: sukurta autoriaus.

23 pav. Ryšių įvedimo langas.

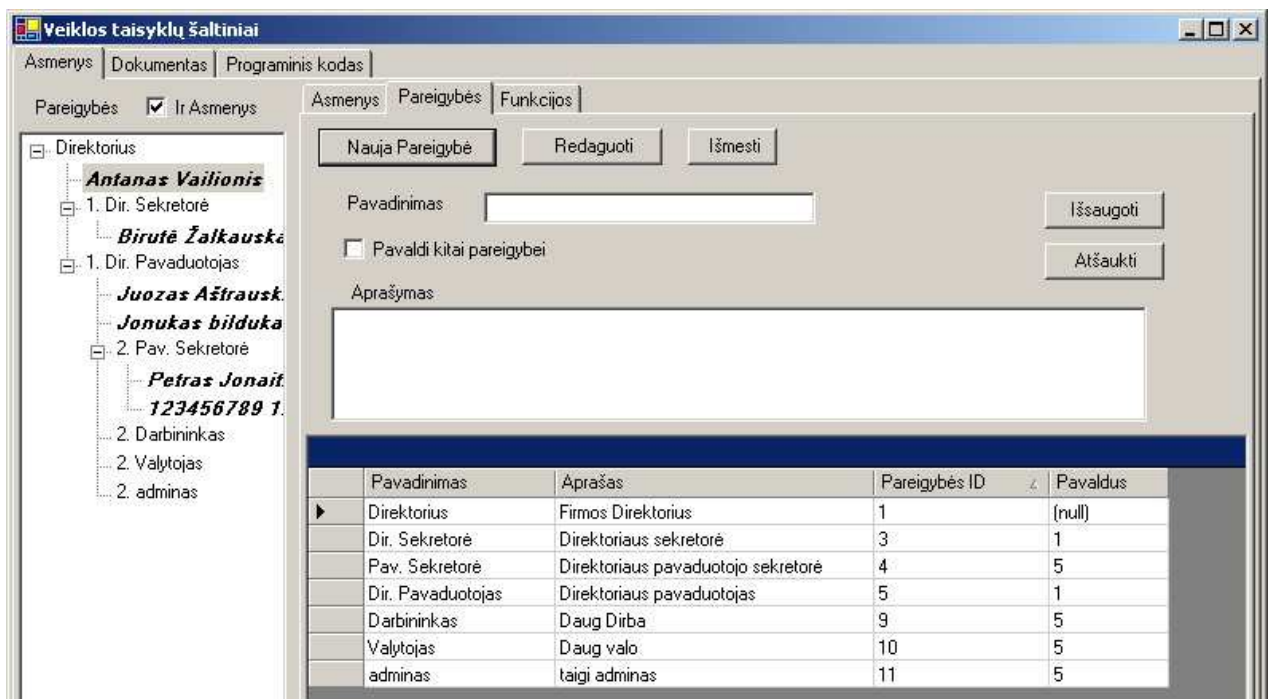
24, 25, 27, 28 paveiksluose matomi langai veiklos taisyklių šaltinių suvedimui. 24, 25 languose suvedami aktoriai, t.y. asmenys ir pareigybių hierarchija. Ši informacija naudojama ir kaip kontaktų sąrašas analitikui. Taip pat asmenų sąrašas ir pareigybių hierarchija gali būti susiejami su funkcijų hierarchija (žr. 26 paveikslą).



Šaltinis: sukurta autoriaus.

24 pav. Asmenų įvedimo langas.

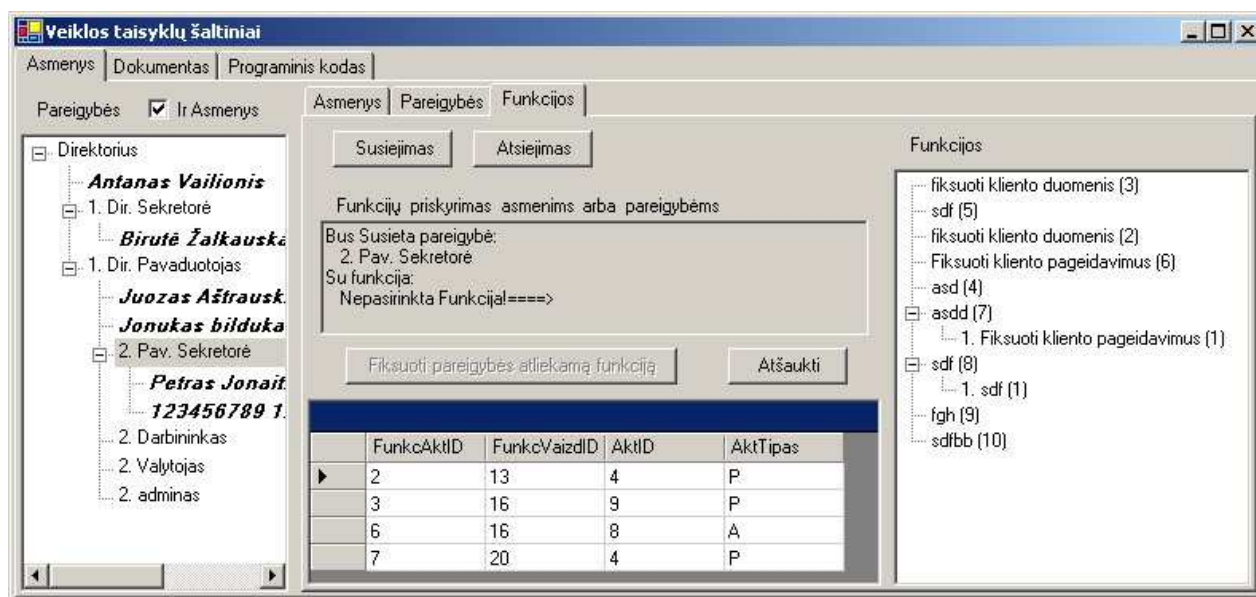
Kaip matome 24 ir 25 paveiksluose asmenys ir pareigybės atvaizduojamos hierarchinėje struktūroje. Tai palengvina duomenų teisingumo ir pilnumo užtikrinimą.



Šaltinis: sukurta autoriaus.

25 pav. Pareigybių įvedimo langas.

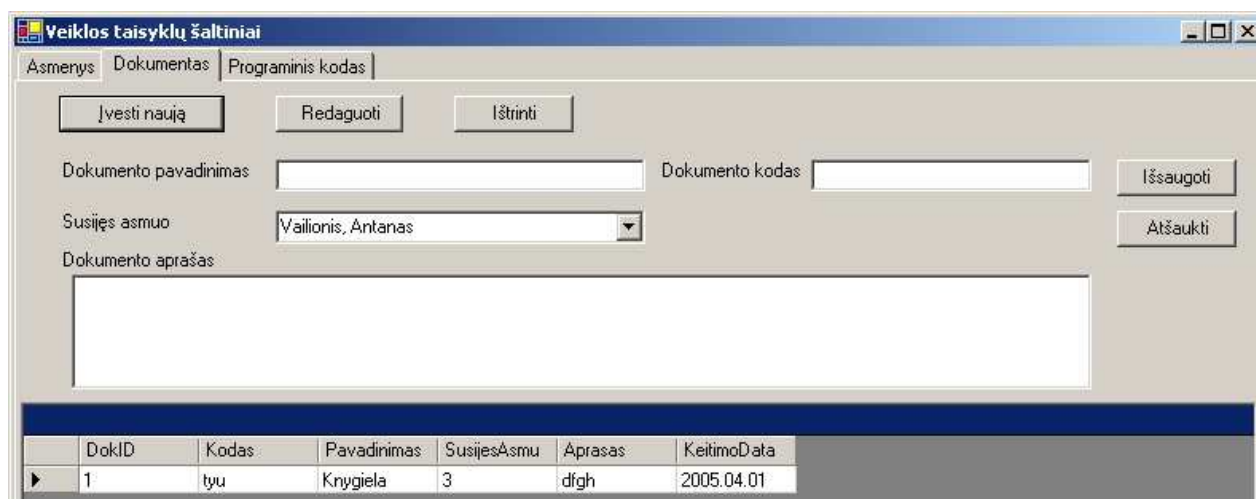
26 paveiksle matomame lange galima susieti dvi hierarchijas pareigybių ir asmenų hierarchiją bei funkcijų. Tai atliekama labai paprastai. Skirtinguose lango šonuose atvaizduotose hierarchijose pažymėjus norimus susieti aktorių ir funkciją paspaudžiamas mygtukas „Fiksuoti pareigybės atliekamą funkciją“.



Šaltinis: sukurta autoriaus.

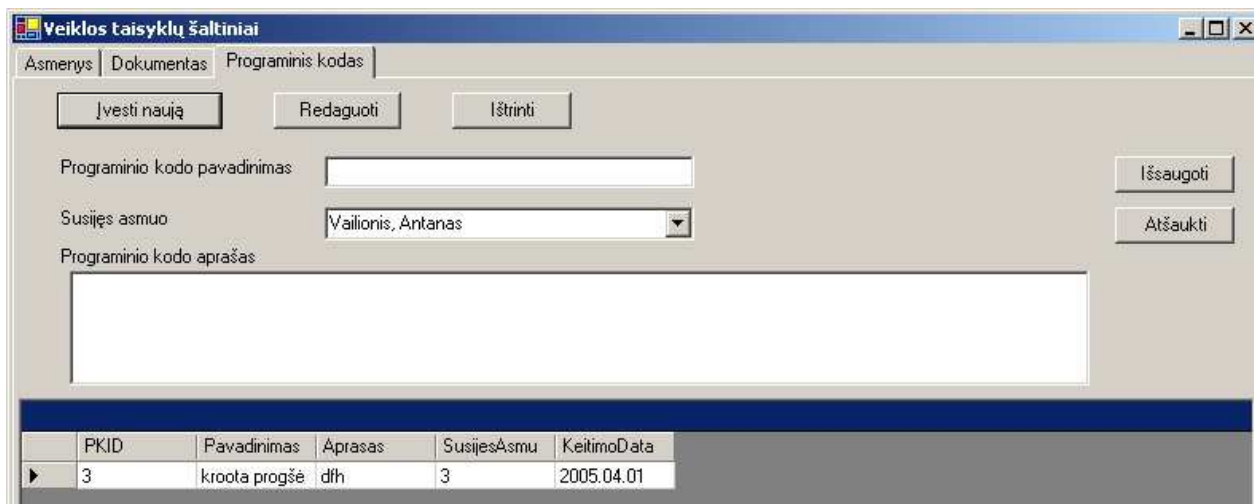
26 pav. Funkcijų susiejimo su aktoriais įvedimo langas.

27 ir 28 paveiksluose matome veiklos taisyklių šaltinių dokumentų ir programinio kodo suvedimo langus. Juose galima aprašyti veiklos taisyklių šaltinius bei užfiksuoti susijusius asmenis (kontaktai galintys suteikti informacijos apie šaltinius).



Šaltinis: sukurta autoriaus.

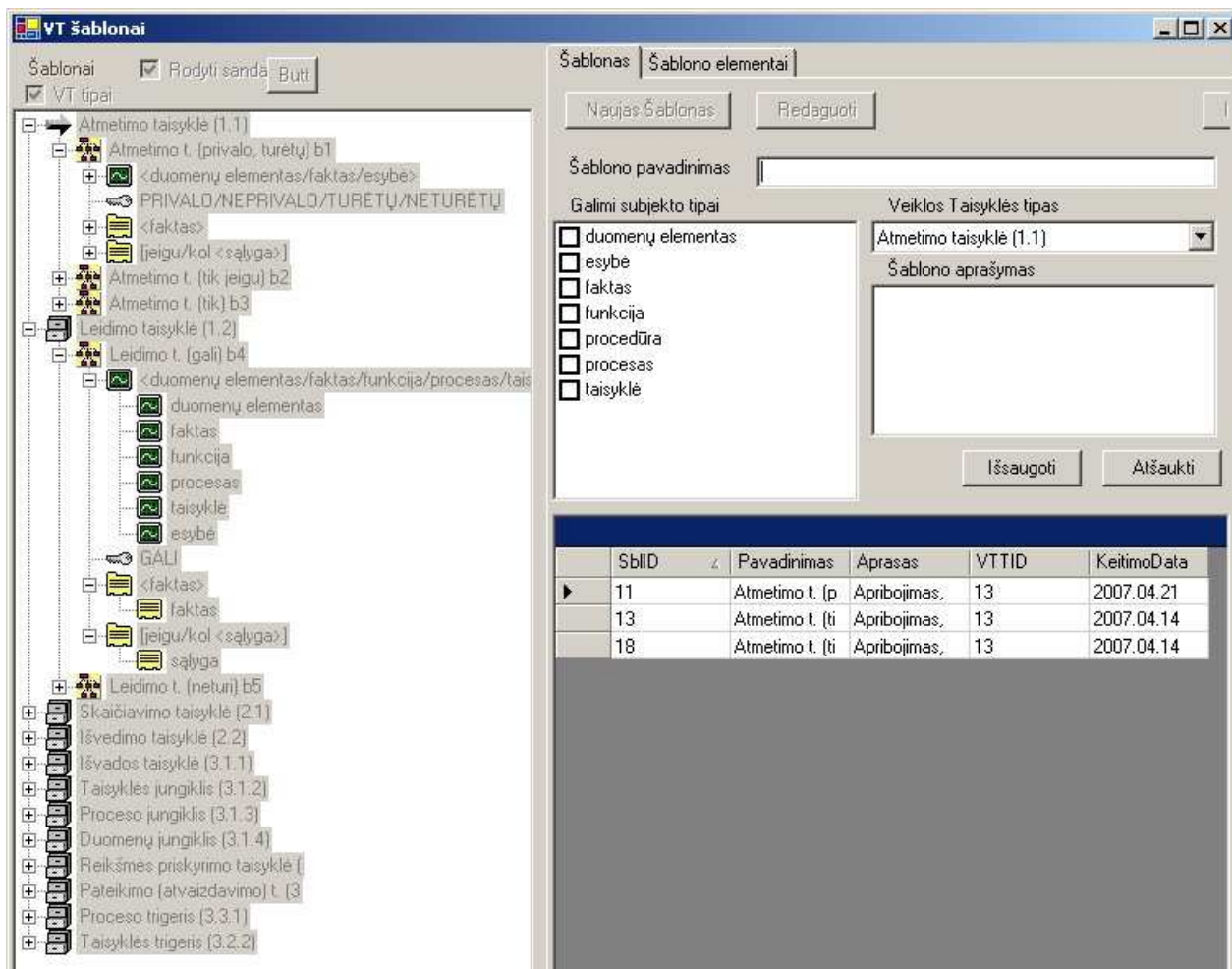
27 pav. Dokumentų įvedimo langas.



Šaltinis: sukurta autoriaus.

28 pav. Programinių kodų įvedimo langas.

29, 30 paveiksluose matomi langai veiklos taisyklių šablonų įvedimui.

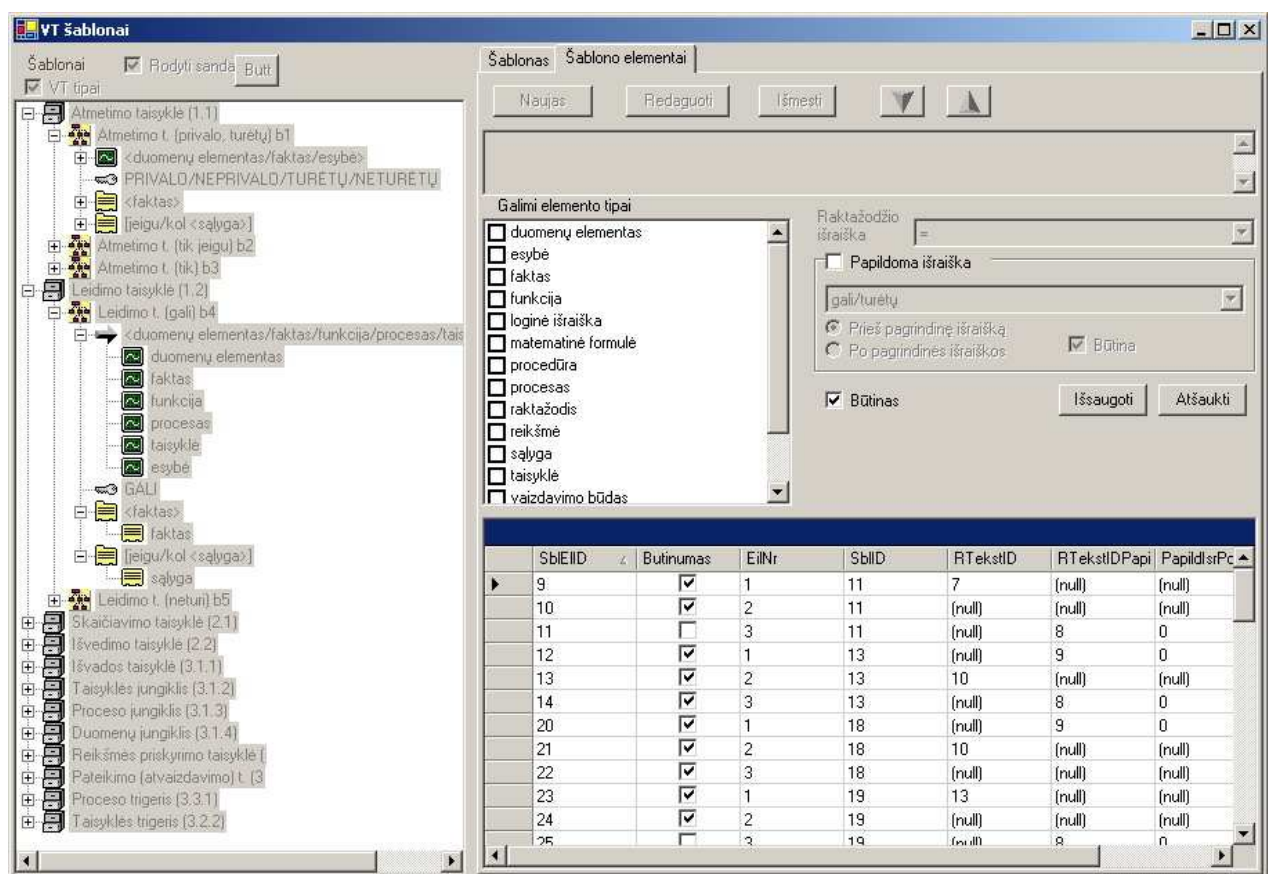


Šaltinis: sukurta autoriaus.

29 pav. Veiklos taisyklių šablonų įvedimo langas.

Aprašant šabloną reikia pasirinkti veiklos taisyklės tipą, galimus subjekto tipus bei įvesti šablono pavadinimą ir šablono aprašymą. Galimų subjektų tipų sąrašą galima redaguoti paspaudus dešinę pelės klavišą ir pasirinkus norimą operaciją (redaguoti pasirinktą tipą, įrašyti naują tipą, išmesti pasirinktą tipą). Redagavimas vyksta lango viduryje išsiskleidus galimų subjektų tipų redagavimo laukų blokui. Analogiškai redaguojami ir veiklos taisyklių tipai.

Kadangi šablonai aprašomi daugeliu skirtingose lentelėse saugomų įrašų tai bendrą vaizdą perteikti reikalinga hierarchinė jų atvaizdavimo struktūra. Kaip matome 29 paveiksle atvaizduotame lange kairėje lango dalyje yra medžio tipo struktūra atvaizduojanti sistemoje išsaugotų šablonų informaciją. Užvedus pelę ant bet kurio elemento iššoka geltonas informacinis langelis su visa informacija užfiksuota įrašant šį elementą įskaitant detalius aprašus (kuriuose gali būti pvz.: taisyklių pavyzdžiai).

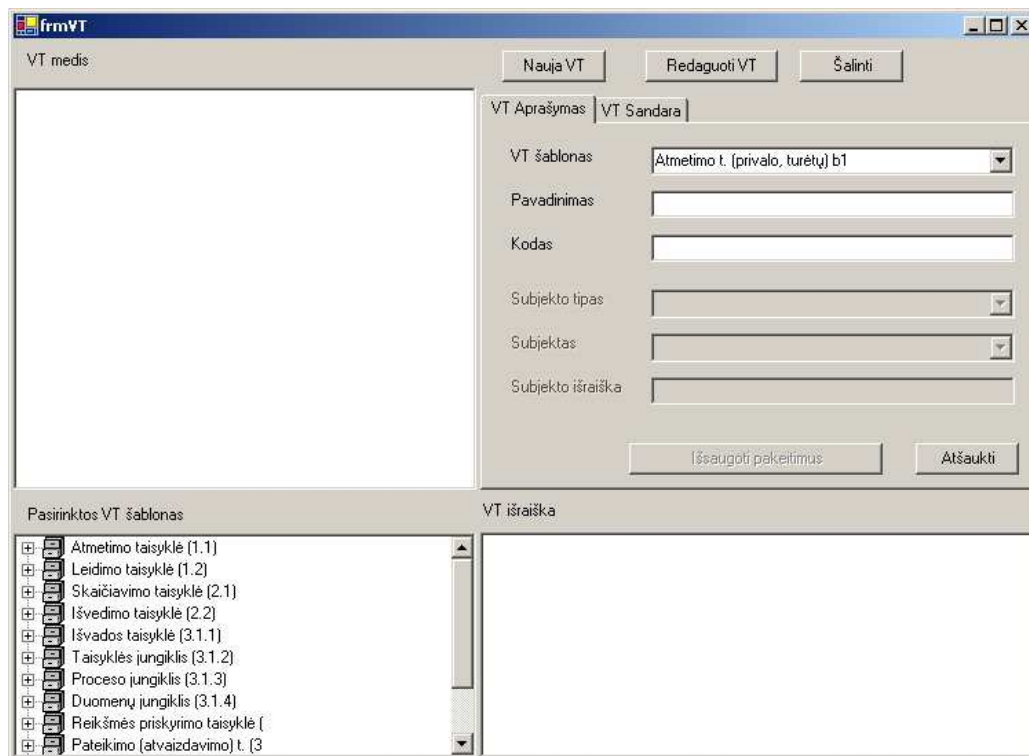


Šaltinis: sukurta autoriaus.

30 pav. Veiklos taisyklių šablonų elementų įvedimo langas.

30 paveiksle matome šablonų elementų įvedimo langą. Jis leidžia fiksuoti elemento tipus bei papildomas išraiškas bei koreguoti jų eiliškumą, bei būtinumą. Šiame lange esantys galimų elementų tipų, raktažodžio išraiškų ir papildomų išraiškų sąrašai redaguojami analogiškai aukščiau aprašytam galimų subjektų tipų sąrašo redagavimui.

31 paveiksle matomas siūlomas veiklos taisyklių fiksavimo langas.



Šaltinis: sukurta autoriaus.

31 pav. Veiklos taisyklių įvedimo lango prototipas.

Veiklos taisyklės fiksuojamos užpildant veiklos taisyklės išraiškos lauką bei suformuojant įrašus atitinkamose DB lentelėse atspindinčius šiame lauke užrašyto sakinio struktūrą. Ši struktūra turi atitikti pasirinktą VT šabloną. VT sandaros lange suvedami komponentai kurie siejasi su sakinio užrašyto VT išraiškos lauke dalimis. Kokią informaciją čia reikia suvesti priklauso nuo pasirinkto šablono.

3.5 Programuotojo vadovas

Programos duomenų bazė realizuota Microsoft Access 2000 duomenų bazių valdymo sistemos pagrindu. Duomenų bazės faile yra tik su duomenų struktūra susiję elementai, t.y. programos logika yra atskirta.

Programos vartotojo sąsaja sukurta naudojantis Microsoft Visual Basic .NET sistema. Surišimas su duomenų failu vyksta kiekvienos formos load procedūroje sakiniiais:

```
cnSaugykla.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;User
ID=Admin;Data Source=" Application.StartupPath & "\Saugykla.mdb"
cnSaugykla.Open()
```

Taigi duomenų failas gali būti patalpintas bet kur ir jo pavadinimas gali būti koreguojamas, tačiau reikia atitinkamai pakoreguoti ir minėtą prisijungimo sakinį.

Minimalūs reikalavimai:

- Palaikomos operacinės sistemos: TabletPC; Windows 2000; Windows 2000 Service Pack 2; Windows 2000 Service Pack 3; Windows 2000 Service Pack 4; Windows 98; Windows 98 Second Edition; Windows ME; Windows NT; Windows Server 2003; Windows XP; Windows XP Media Center Edition
- Suinstaliuoti Visual studio .NET 2003 versiją.
- Suinstaliuoti MS Access 2000 versiją;
- Reikalavimai techninei įrangai: Pentium III 600 Mhz, 256 MB RAM, 3 Gb laisvos vietos kietame diske;

Pateikti reikalavimai minimalūs.

3.6 Eksperimentinės realizacijos komentarai

Veiklos taisyklių metodikos propaguotojų galutinis tikslas yra dinaminis veiklos taisyklių keitimas informacinėje sistemoje ir tą atlikti turėtų ne IS, o veiklos specialistai. Pasirinkta veiklos taisyklių saugyklos realizacijos metodika tai apsunkintų dėl sudėtingo ir ilgai trunkančio, neintuityvaus darbo su sistema proceso. Kitaip tariant be tam tikro informacinių sistemų išsilavinimo žmogus su sistema dirbti negalėtų.

Pavyzdžiui, funkcijų hierarchija. Jų specifikuojimas vis dar gana techniškai, informaciją suvedinėti į reikia į kelis laukus, jų sekų ir hierarchijų sudarymo principai nėra savaime suprantami.

Kitas pavyzdys struktūrinės veiklos taisyklės. Ši posistemė akivaizdžiai daugiau orientuota į IS analitiką ar programuotoją: isA priklausomybė, fizinis vardas, sąvoka esybė, būtinumas, unikalumas, kardinalumas ir pan. nėra įprastinės veiklos dalyviams koncepcijos. Realizuojant šią posistemę stengiasi išreikšti duomenų modelį tekstu, kad galutinis vartotojas galėtų ją verifikuoti. Tačiau manau, kad tradiciniai duomenų modelio kūrimo metodai braižant diagramas yra daug spartesni ir iššaukia mažiau klaidų. O taip pat leidžia fiksuoti papildomus elementų aprašus, kurie galėtų būti verifikuojami paprastų vartotojų.

Dar vienas sudėtingo darbo su sistema pavyzdys: šablonų įvedimas. Siekiant leisti vartotojui susikurti šablonus pačiam jie tapo stipriai išskaidyti ir kiekvienas šablonas išsaugomas 8-iose lentelėse. Todėl kiekvieno šablono formavimas užtrunka nemažai laiko ir reikalauja didelio atidumo ir nuoseklumo, daugybės operacijų. Ši situacija atrodo dar keisčiau, kai pereinama prie veiklos taisyklių fiksavimo naudojant šiuos šablonus. Egzistuoja tam tikras atotrūkis tarp šablono ir veiklos taisyklės. Veiklos taisyklė užrašoma, kaip sakinytis ir saugoma viename lentelės lauke. O štai kiti jau techniniai laukai modeliuoja jos struktūrą ir užmezga sąsajas su likusiais specifikacijos elementais. Tai reiškia, kad pagal dabartinį procesą su taisyklės modelio elementais galima susieti bet kokį šablono leidžiamą likusios specifikacijos elementą, tačiau neįmanoma kontroliuoti ar tas elementas tikrai yra naudojamas toje taisyklėje.

Manau, kad tokioje veiklos taisyklėmis grindžiamoje sistemoje reikalinga tiksliai apibrėžti į koki vartotoją bus orientuojamasi. Jeigu be išsilavinimo informacinių sistemų srityje, tai mano nuomone sistemoje informacija turi būti fiksuojama, kaip laikantis tam tikrų taisyklių užrašytas tekstas. Šį tekstą pati sistema turi gebėti konvertuoti, pavyzdžiui į duomenų modelį. Veiklos taisyklės fiksavimas turėtų prasidėti nuo jos teksto analizės ir pastangų pritaikyti turimą šabloną, o jei toks dar neegzistuoja tai pasiūlyti fiksuoti naują. Toliau likusius vartotojo poreikių specifikacijos elementus sieti su šablono elementais per tikrąją veiklos taisyklės tekstą, t.y. kad galima būtų bakstelėti į žodį veiklos taisyklėje ir sistema gebėtų parodyti, kad tai yra pavyzdžiui kažkokio termino sinonimas.

IŠVADOS IR PASIŪLYMAI

1. Analizuojant VT koncepciją pastebėta, kad veiklos taisyklių metodu grindžiamos IS projektavimo priemonės nėra paplitę taip kaip tradiciniai metodai.
2. VT metodo pagrindinis privalumas galimybė išskirti veiklos logiką išreiškiančią IS dali ir dinamiškai ją keisti.
3. Veiklos taisyklių struktūrizavimo srityje dirba nemažai mokslininkų ir kuriamos skirtingos koncepcijos, tai reiškia, kad kol kas nėra vieno geriausio metodo ir nauji darbai šioje srityje yra labai reikalingi.
4. Ryškiausias bandymas unifikuoti veiklos taisyklių metodiką apjungiant daugelio mokslininkų įdirbį yra Business Rules Group projektas, siekiantis patvirtinti ir standartizuoti VT metodą.
5. Eksperimentinei realizacijai pasirinktas Business Rules Group projekte dirbančių mokslininkų darbais grindžiamas KTU Informacinių sistemų katedroje vystomas veiklos taisyklių struktūrizavimo procesas.
6. Pasirinktas procesas orientuojasi ne vien į veiklos taisyklėmis grindžiamų informacinių sistemų kūrimą, bet siūlo naują metodą užrašyti vartotojo reikalavimus bet kokio tipo veiklos informacinei sistemai.
7. pasirinktas vartotojo reikalavimų specifikavimo metodas unikalus tuo, kad jis apjungia dalykinės srities konteksto apibrėžimą, funkcijų nustatymą, duomenų modelio sudarymą ir veiklos taisyklių specifikavimą į vientisą procesą ir siūlo visą informaciją saugoti vienoje integruotoje saugykloje.
8. Modelio pagrindu pasirinktas Ronald G. Ross'o veiklos taisyklių struktūrizavimo pasitelkiant natūralios kalbos šablonus metodas.
9. VT saugyklos struktūra pritaikyta būtent natūralios kalbos šablonais aprašomoms veiklos taisyklėms.
10. Apsibrėžtos pagrindinės tokios saugyklos realizacijos vartotojo sąsajos gairės, atsižvelgiant į tai, kad naudojamas metodas reikalauja didelio realizacijos lankstumo, nes vartotojui leidžiama pačiam susikurti nestruktūrinių veiklos taisyklių klasifikaciją bei struktūrizavimo šablonus.
11. Pasirinkto veiklos taisyklių saugyklos modelio pagrindu bei laikantis apsibrėžtų vartotojo sąsajos reikalavimų atlikta eksperimentinė sąsaja, bei pastebėta, kad modelis iš principo gali būti realizuotas ir veikia leisdamas išsaugoti pagrindinius vartotojo reikalavimus kuriamai informacinei sistemai. Tačiau pasirinkta vartotojo sąsajos koncepcija reikalauja daug vartotojo veiksmų, o duomenų modelio arba veiklos taisyklių šablonų specifikavimo procesas yra ilgai trunkantis ir daug atidumo reikalaujantis.

12. Atliktos analizės ir eksperimentinės realizacijos pagrindu nustatyta, kad KTU informacinių sistemų katedros siūlomas vartotojo reikalavimų kuriamai informacinei sistemai specifikavimo metodas tinkamas naudoti IS projektavime.

LITERATŪROS SĄRAŠAS

1. Kapočius Kęstas, Butleris Rimantas. Repository for Business Rules Based IS requirements. Informatica, Vol. 17, No.4, p. 503-518., 2006
2. Kapočius Kęstas, Butleris Rimantas. Business rules driven approach for elicitation of IS requirements. Proceedings of 9 th World Multi-Conference on Systemics, Cybernetics and Informatics, July 10-13, 2005, Orlando, USA, 2005, Vol IV, ISBN:980-6560-56-6, p.276-281, 2005
3. Zachman, John. Zachman Framework for Enterprise Architecture. [interaktyvus] Žiūrėta 2007 m. balandžio 29 d. Prieiga per internetą: <<http://www.zifa.com/framework.pdf>>.
4. David C. Hay Modeling Business Rules: What Data Models Do. (2003) [interaktyvus] Žiūrėta 2007 m. balandžio 11 d. Prieiga per internetą: <<http://www.tdan.com/i027fe03.htm>>.
5. Čaplinskas Albertas (1998) Programų sistemų inžinerijos pagrindai. II dalis. Matematikos ir informatikos institutas, Vilnius. 276 p.
6. The Business Rules Group. Defining Business Rules ~ What Are They Really? (2000). [interaktyvus]. Žiūrėta 2007 m. gegužės 4 d. Prieiga per internetą: <http://www.businessrulesgroup.org/first_paper/BRG-whatBR_3ed.pdf>
7. The Business Rules Group. Business Rules Manifesto. (2005). [interaktyvus]. Žiūrėta 2007 m. gegužės 4 d. Prieiga per internetą: <http://www.businessrulesgroup.org/first_paper/BRG-whatBR_3ed.pdf>
8. Ronald G. Ross. Principles of the Business Rule Approach. Boston: Addison Wesley, 2003. 352 p. ISBN 0-201-78893-4.
9. Business rules solutions. The Steps of Business Rule Methodology. (2005) [interaktyvus] žiūrėta 2007 m. Gegužės 10 d. Prieiga per internetą: <<http://www.brsolutions.com/metamodel/sld001.htm>>
10. Business rules solutions. Developing the System Model. (2001) [interaktyvus] žiūrėta 2007 m. Gegužės 09 d. Prieiga per internetą: <<http://www.brsolutions.com/proteus.zip>>
11. Business rules solutions. BRS RuleSpeak Do's and Don'ts. (2002) [interaktyvus] žiūrėta 2007 m. Gegužės 10 d. Prieiga per internetą: <<http://www.brsolutions.com/rulespeak.zip>>
12. Von Hale Barbara. Business rules applied: building better systems using the business rules approach. New York: John Wiley & Sons, Inc., 2002. 495 p. ISBN 0-471-41293-7
13. Gražulis Darius; Kapočius, Kęstutis; Butleris, Rimantas. Reikalavimai veiklos taisyklėmis grindžiamos vartotojo poreikių saugyklos realizacijai // Informacinės technologijos: X tarpuniversitetinė magistrantų ir doktorantų konferencija: konferencijos pranešimų medžiaga. Kaunas: Technologija, 2005. ISBN 9955-09-852-X. p. 159-162.

SANTRAUKA

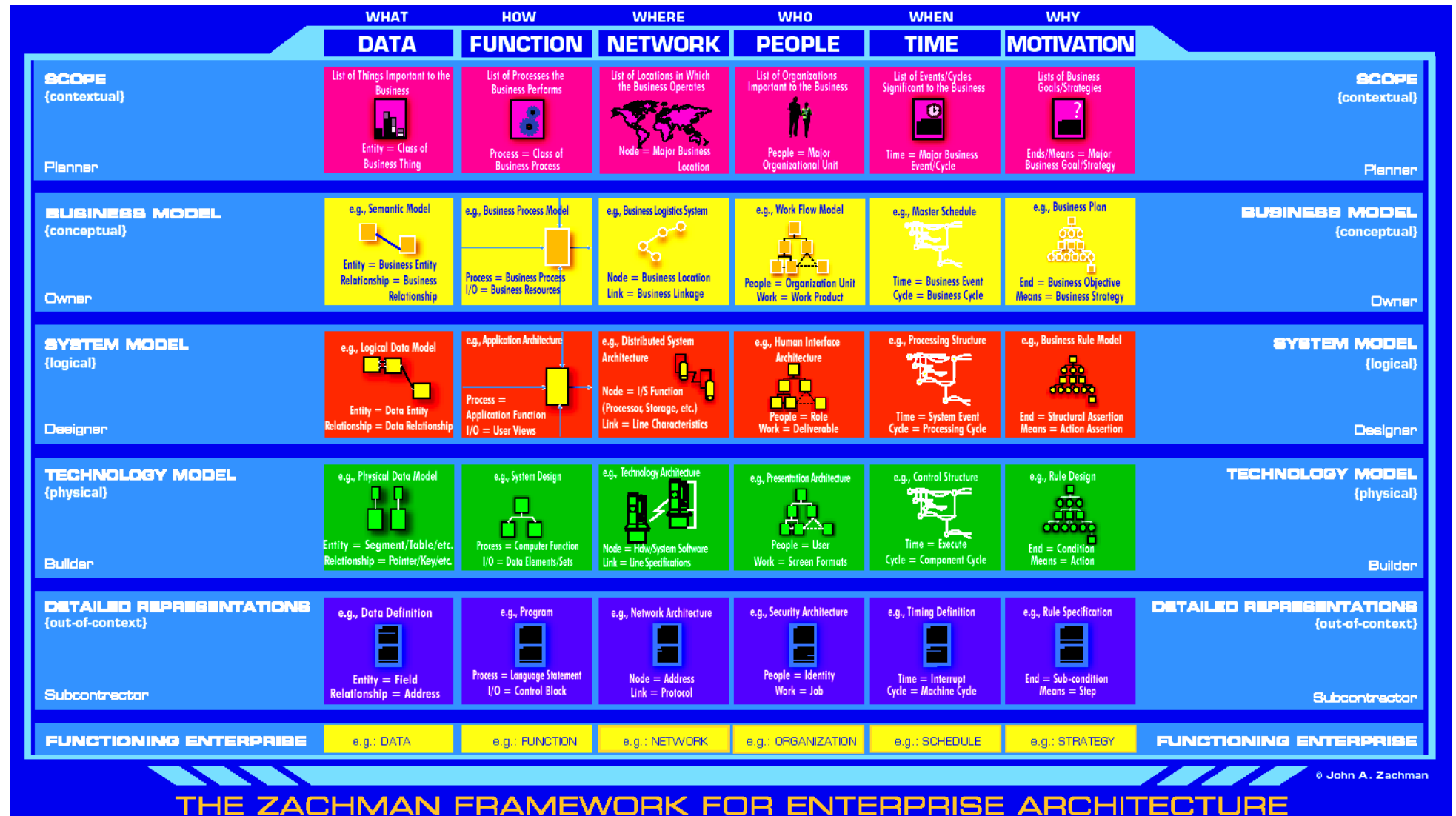
Gražulis Darius. (2007). Business rules based user requirements for IS specification repository. Master's Work in Business Informatics. Graduation Paper. Kaunas: Vilnius University, Kaunas Faculty of Humanities, Department of Informatics. 87 p.

SUMMARY

The success of requirements analysis often decides the quality of the finished information system (IS). Application of ideas behind the business rules approach for requirements analysis could help narrow the gap between business representatives and analysts thus paving the way to a better requirements specification. In this paper a framework for business rules driven IS requirements specification is presented. The proposed approach is based on discovery of functions, business decisions and underlying business rules. The management of captured requirements as well as the generation of relevant documentation is performed using the repository system, major implementation guidelines of which are also discussed. The prototype of the repository system has been implemented. The problems of the proposed approach that were discovered during the experimental implementation are also discussed.

PRIEDAI

1 Priedas: ISA Framework



Šaltinis: Zachman Institute 2001 p1 [3].

1 pav. Džono Zachmano sukurtas ISA Framework.

1987 m. IBM darbuotojas John Zachman, turintis 26-ių metų taikomųjų informacinių sistemų projektuotojo patirtį, sukūrė Zachman Framework for Enterprise Architecture – organizacinių sistemų architektūros modelį, skirtą organizacijų aprašymui ir klasifikavimui, vykdant jų kompiuterizavimą. Šiuo metu daugelis organizacijų naudoja J.Zachman modelį savo informacijos infrastruktūros aprašymui [3].

Organizacijos informacijos sistemos architektūros (ISA) modelis yra bazinė struktūra, kuri atvaizduoja visų paskirčių informacinių sistemų (IS), veikiančių organizacijoje, architektūrą keliais svarbiausiais aspektais, būtinais organizuojant IS kūrimą, jas integruojant, valdant, vystant ir analizuojant jų veikimą. Tokie skirtingus tyrimo aspektus atitinkantys architektūros atvaizdavimai yra vadinami artifaktais. ISA modelis sukuria dvimatį architektūros modelį: organizacijos veikla dekomponuojama dvimatėje erdvėje. Vertikalios ašies kryptimi išdėstyti bendri visiems (organizacijos veiklos aspektu) architektūrų atvaizdavimams aprašymo lygiai, horizontalios ašies kryptimi išdėstyta architektūrų klasifikacija [3].

2 Priedas: RuleSpeak šablonai anglų kalba

Ronald G. Ross savo knygoje (literatūros sąrašė nr. 8) pateikia tokius veiklos taisyklių šablonus:

Kategorija	Apibūdinimas/tikslas	Veiklos taisyklės subjektas	Šablonas	Pavyzdys
Rejector	A constraint for maintaining correctness (consistency) by preventing violations	Term or fact (data item also permitted)	<Subject> MUST/should [not] <fact> [if/while <condition>].	An order MUST indicate the date it was received
Rule keywords: MUST			<Subject> may/should <fact> ONLY if/while <condition>.	A student MUST not take more than four courses while on probation.
ONLY			<Subject> may/should <fact> ONLY <preposition> <condition>.	A customer may place an order ONLY if the customer holds an account. A salaried employee may work ONLY in a budgeted department.
Permission statement	A policy or clarification permitting a business practice	Term, fact, rule, or process (data item also permitted)	<Subject> MAY <fact/rule keyword> [if/while <condition>].	An order on credit totaling \$1,000 or under MAY be accepted from a customer even if the customers credit has not been checked.
Rule keywords: MAY			<Subject> NEED NOT <fact/rule keyword> [if/while <condition>].	A customer NEED NOT place any orders.
NEED NOT				
Computation rule	A statement or arithmetic formula indicating how to calculate a numeric value	Computed term (data item also permitted)	<Subject> must/should [not] BE COMPUTED as <mathematical formula> [if/while <condition>].	The amount paid for an order must BE COMPUTED as the sum of all payment amounts applied to the order.
Rule keywords: BE COMPUTED			Shorthand: <Subject> = <mathematical formula> [if/while <condition>].	The amount paid for an order = the sum of all payment amounts applied to the order.
Derivation rule	A statement or logical expression indicating how to determine a yes/no (true/false)	Derived term	<Subject> must/should [not] BE TAKEN TO MEAN <logical expression> [if/while	Big-ticket item must BE TAKEN TO

Rule keywords:	result	(data item also permitted)	<condition>].	MEAN the items cost exceeds \$500.
BE TAKEN TO MEAN			Shorthand:	Big-ticket item MEANS the items cost exceeds \$500.
MEANS			<Subject> MEANS [not] <logical expression> [if/while <condition>].	
Inference rule	A rule that infers a conclusion from a particular set of circumstances	Term	<Subject> must/should [not] BE CONSIDERED [a] <term> if/while <condition>.	A person must BE CONSIDERED a woman if the person is female and the persons age is 21 or over.
Rule keywords:		(data item also permitted)	Shorthand:	A person is a woman if the person is female and the persons age is 21 or over.
BE CONSIDERED			<Subject> is [not] [a] <term> if/while <condition>.	
Rule toggle	A rule that turns another rule on or off in a particular set of circumstances, especially for making exceptions	Rule	Informal:	A library card may be held by at most one borrower UNLESS one of the borrowers who hold the library card is Bill Gates.
Rule keywords:			<Rule statement>, UNLESS/EXCEPT <condition>.	
UNLESS			Formal:	The one-borrower-per-library-card rule must not BE ENFORCED if one of the borrowers who hold the library card is Bill Gates.
EXCEPT			<Rule name> must/should [not] BE ENFORCED if/while <condition>.	
BE ENFORCED				
Process toggle	A rule that turns a process on or off in a particular set of circumstances	Process or procedure	<Subject> must/should [not] BE ENABLED/DISABLED if/while <condition>.	Send-appointment-notice must BE DISABLED if the client's address is unknown.
Rule keywords:				
BE ENABLED				
BE DISABLED				
Data toggle	A rule that deletes data (or creates it randomly) in a particular set of circumstances	Data item	<Data item> must/should [not] BE CREATED/DELETED if/while <condition>.	Each outstanding case issue must BE DELETED when the case is closed.
Rule Keywords:				

BE CREATED

BE DELETED

Imprint rule	A rule that sets a stored data item to a particular value	Term or fact	<Term> must/should [not] BE SET to <term/value> [when/if <condition>].	A students-semester-fees-owed must BE SET to \$3,065 when the student registers for a semester.
Rule keywords:		(data item also permitted)		

BE SET

Presentation rule	A rule that requires data to be presented in a certain manner (for example, on a screen or in a report)	Term or fact	<Subject> must/should [not] BE DISPLAYED [to/on/in <media>] <display manner> [if/while <condition>].	An order must BE DISPLAYED to the screen in red if the order is overdue.
Rule keywords:		(data item also permitted)		

BE DISPLAYED

Process trigger	A rule that automatically executes a process or procedure in a given set of circumstances	Process or procedure	<Subject> must/should BE EXECUTED when <condition>.	Send-advance-notice must BE EXECUTED for an order when the order is shipped.
Rule keywords:				

BE EXECUTED

Rule trigger	A rule that automatically fires another rule in a given set of circumstances	Rule	<Rule name> must/should BE FIRED when <condition>.	The projected-shipment-date-rule must BE FIRED when a shipment is displayed to the screen.
Rule keywords:				

BE FIRED

3 Priedas: Straipsnis

Gražulis Darius; Kapočius, Kęstutis; Butleris, Rimantas. Reikalavimai veiklos taisyklėmis grindžiamos vartotojo poreikių saugyklos realizacijai // Informacinės technologijos: X tarpuniversitetinė magistrantų ir doktorantų konferencija: konferencijos pranešimų medžiaga. Kaunas: Technologija, 2005. ISBN 9955-09-852-X. p. 159-162.

Veiklos taisyklėmis grindžiamos IS reikalavimų saugyklos realizavimas Darius Gražulis¹, Kęstas Kapočius², Rimantas Butleris²

¹ VU KHF Informatikos katedra,

Muitinės g. 8, LT-44280, Kaunas, Lietuva, darius.gr@gmail.com

² KTU Informacijos Sistemų katedra,

Studentų g. 50, LT-51368, Kaunas, Lietuva.

Tradiciniai IS reikalavimų specifikuojimo metodai neužtikrina pakankamai glaudaus santykio tarp kuriamos sistemos dalykinės srities specialistų ir analitikų. Dėl to nukenčia sistemos reikalavimų specifikuojimo, o tuo pačiu ir galutinės sistemos, kokybė. Straipsnyje atskleidžiama veiklos taisyklių panaudojimo vartotojo poreikių specifikuojimo procese nauda. Apžvelgiamas KTU Informacinių sistemų katedroje kuriamas taisyklėmis grindžiamas vartotojo reikalavimų kuriamai IS specifikuojimo proceso modelis bei reikalavimų saugyklos modelis. Straipsnyje pristatomas reikalavimų saugyklos IS realizavimo planas ir aptariami pagrindiniai kiekvieno saugyklos posistemio realizavimo aspektai.

1 Įvadas

Populiarūs reikalavimų kuriamai Informacijos sistemai (IS) fiksavimo ir modeliavimo metodai nesuteikia galimybės išreikšti juos ne IS specialistui suprantama forma. Dėl šios priežasties IS analitiko sukurtas modelis dažnai negali būti patikrintas dalykinės srities specialistų. Dėl to nukenčia reikalavimų specifikuojimo, o tuo pačiu ir galutinės sistemos kokybė. Tam, kad sumažinti šį atotrūkį tarp IS analitiko ir dalykinės srities specialisto, modelis turėtų būti išreikštas kuo natūraliau būdu.

KTU Informacijos sistemų katedroje vystomas vartotojo reikalavimų specifikuojimo metodas grindžiamas veiklos taisyklėmis. Vartotojo reikalavimai saugomi struktūrizuota tekstone forma. Tai reiškia, kad tekstu išreikštą modelį turėtų suprasti kiekvienas skaityti mokantis dalykinės srities specialistas.

Šiame straipsnyje trumpai pristatomas minėtas metodas bei apibrėžiami pagrindiniai reikalavimų specifikuojimo saugyklos sistemai keliami reikalavimai.

2 Reikalavimų specifikuojimo metodo apžvalga

2.1 VT struktūrizavimo modelis

Remiantis GUIDE projektu veiklos taisyklės gali būti skirstomos į struktūrines ir nestruktūrines. Struktūrinės veiklos taisyklės tai terminai ir faktai aprašantys ryšius tarp terminų. Iš esmės struktūrinėmis taisyklėmis aprašomas duomenų modelis. Kadangi vartotojo reikalavimų specifikuojimo procese detalus duomenų modelis dar neprivalo būti suformuotas, tai KTU informacijos Sistemų katedros siūlomame metode apsiribojama tik konceptualaus duomenų modelio formavimu.

Nestruktūrinėms veiklos taisyklėms fiksuoti pasirinktas natūralios kalbos sakinių šablonais paremtas Business Rules Solutions RuleSpeak (BRS RuleSpeak) metodas. Pagal šį metodą terminai saugomi Esybių-Ryšų modelyje. BRS RuleSpeak taisyklės suskirstytos į funkcines kategorijas. Kiekvienos kategorijos taisyklės užrašyti naudojama po vieną ar keletą šablonų. Šablono pavyzdys:

<Subjektas> TURI/NETURI <faktas> [kol/jeigu <sąlyga>].

Taisyklės pavyzdys būtų "Studentas turi mokėti už mokslą, jeigu neišlaikė egzamino". Visuose šablonuose pirmasis elementas yra subjektas, kuris priklausomai nuo kategorijos gali būti terminas, faktas, konkreti reikšmė, kita taisyklė ir t.t. Raktažodžiais paremta šablono struktūra leidžia taisyklės išskaidyti ir saugoti specialioje saugykloje.

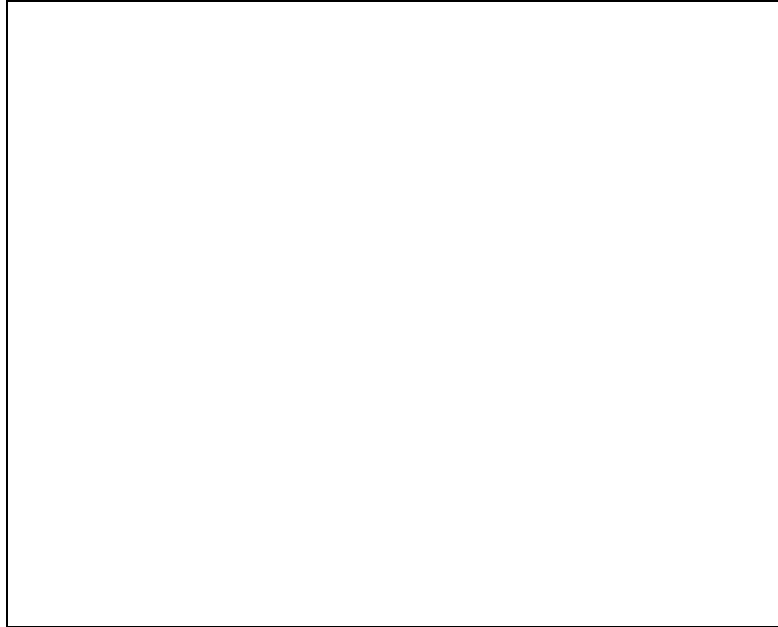
2.1.1 Vartotojo reikalavimų specifikuojimo procesas

Reikalavimų specifikuojimo proceso schemą matome 1 paveiksle. Procesas prasideda apibrėžiant veiklos kontekstą, bei aktorių, kaip būsimos sistemos vartotojų arba poreikių informacijos šaltinio, registravimas. Toliau specifikuojamos aktorių atliekamos funkcijos bei jų atlikimo sekos. Taip pat registruojami kiekvienos funkcijos atlikimo metu priimami sprendimai bei jų eiliškumas.

Lygiagrečiai vyksta konceptualaus duomenų modelio formavimas, tai yra fiksuojamos duomenų esybės jų atributai bei ryšiai tarp esybių (struktūrinės veiklos taisyklės: terminai ir faktai). Svarbi reikalavimų specifikacijos dalis yra CRUD matricos formavimas, tai yra užfiksavimas kokios funkcijos naudoja kokius terminus ir koks to panaudojimo pobūdis: kopijuojama, skaitoma.

Pasiekus nestruktūrinių veiklos taisyklių fiksavimo fazę. Kiekvienai veiklos taisyklei yra fiksuojamas šaltinis: asmuo, dokumentas arba programinis kodas.

Sudarius veiklos taisyklių rinkinį jis tikrinamas. Pradžioje automatinėmis priemonėmis, vėliau verifikuojamas būsimų IS vartotojų.



1 pav. Reikalavimų specifikavimo procesas.

Kiekviename žingsnyje fiksuojama informacija išsaugoma vartotojo reikalavimų saugykloje kurios duomenų bazės lentelių modelį matome 2 paveiksle.

3 Saugyklos realizavimo principai/gairės

Saugyklos duomenų bazės lentelių modelį matome 2 paveiksle. Jame lentelės pasistengta sugrupuoti pagal posistemas. Sistemos realizavimas bus vykdomas panašia eilės tvarka kaip ir reikalavimų specifikavimo proceso metu.



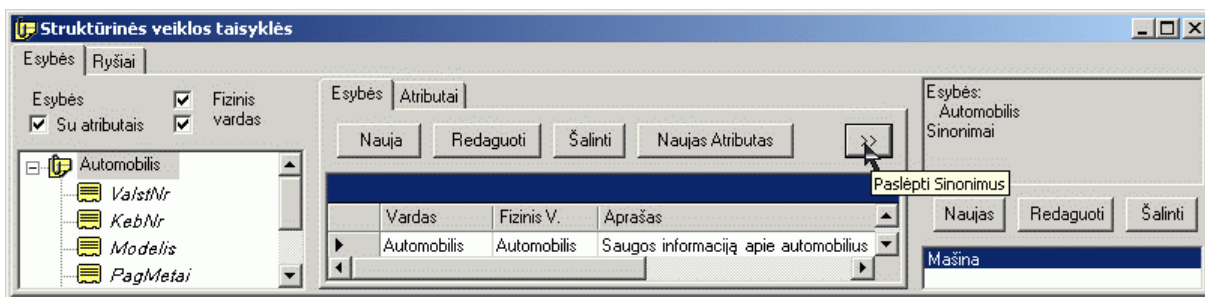
2 pav. Saugyklos duomenų bazės modelis.

Bendri realizacijai keliami reikalavimai:

- Siekiant neperkrauti langų įvairiomis informacijos įvedimo priemonėmis turi būti stengiamasi rodyti tik tuo metu reikalingas. Tam bus pasitelkiami lango puslapiai (Tab), elementų aktyvavimas/deaktyvavimas, bei paslėpimas/parodymas.
- Įvedamos informacijos korektiškumas turi būti kontroliuojamas nuo pat pirmo žingsnio, tai yra iškart įvedinėjant ją į laukus. Visur kur įmanoma bus apribotas galimų įvesti informacijos variantų kiekis. Taip nebus leidžiama nekorektiškai informacijai patekti į duomenų bazę.
- Patogiai įvestos informacijos peržiūrai turi būti realizuoti įvairūs filtrai, navigacijos ir paieškos priemonės.
- Languose turi būti išvedama tik vartotojui prasminga informacija (t.y. nerodomi identifikacijos ir kiti “tarnybiniai” laukai).
- Keletas modelio lentelių turi ryšius pačios su savimi. Vadinasi įrašai jose gali būti vaizduojami hierarchinės struktūros. Tam tinka medžio tipo vaizdavimo būdas.
- Medis bus panaudojamas ne tik kaip vizualizacijos, bet ir kaip įrašų navigacijos priemonė. Tai yra paspaudus ant konkrečios medžio šakos įrašų lentelėse bus pažymimi atitinkami įrašai.

Reikalavimai struktūrinių veiklos taisyklių posistemės realizacijai (galimo lango pavyzdys pateiktas 3 paveiksle):

- Kai kurios esybės tarpusavyje yra susietos “Is A” ryšiu. Ši priklausomybė lengviau būtų suprantama ir pastebima atvaizdavus ją medžio struktūroje.
- Kadangi esybės paprastai turi riboto dydžio atributų skaičių, tai minėtame medyje turi būti galimybė vaizduoti ir esybių atributus. Atributų vaizdavimas privalo skirtis nuo esybių vaizdavimo.
- Esybių bei atributų sinonimai nėra nuolat reikalinga informacija, todėl ji dažniausiai bus paslėpta. Specialus mygtukas leis įjungti lango sinonimų peržiūros ir redagavimo posistemę. Čia bus rodomi ir redaguojami tik tuo metu pažymėtos esybės ar atributo sinonimai.
- Ryšių tarp esybių patogesniai registravimui reikalingi du esybių medžiai, tik šiuo atveju jau be atributų.
- Kardinalumų reikšmės turi būti leidžiama pasirinkti iš sąrašo arba įvesti ranka.



3 pav. Galimas Struktūrinių veiklos taisyklių langas.

Reikalavimai veiklos taisyklių šaltinių įvedimo posistemės realizacijai:

- Pareigybių hierarchija vaizduojama kaip medis, taip pat leidžiama pasirinkti ar rodyti medyje pareigybių asmenis.
- Tiek pareigybės tiek asmenys gali atlikti tam tikras funkcijas. Šiems ryšiams užmegzti reikalingas funkcijų hierarchiją vaizduojantis medis.
- Dokumentai ir programinis kodas susiejami su asmenimis leidžiant juos pasirinkti iš sąrašo.

Reikalavimai funkcijų įvedimo langui:

- Kadangi funkcijos saugomos dvejose lentelėse: vienoje funkcijos aprašymas, kitoje funkcijų hierarchija, tai informacijos apie funkcijas pateikimas praktiškai visada turi būti šių dviejų lentelių kombinavimas. Funkcijų hierarchija vaizduojama medžio struktūros, kuri saugoma funkcijų atvaizdų lentelėje, o medžio šakų pavadinimai saugomi funkcijų lentelėje. Medis padeda ir pačios hierarchijos formavime: kuriant funkcijos atvaizdą galima medyje pasirinkti tėvinę funkciją.
- Šiame lange vyksta ir sprendimų įvedimas bei jų susiejimas su funkcijomis. Funkcijų susiejimas su sprendimais taip pat vyksta dalyvaujant medžiui.

Reikalavimai CRUD santykio fiksavimo langui.

- Lange pateikiamos dvi hierarchijos: funkcijų ir terminų (esybių bei atributų). CRUD santykis užmezgamas tarp pažymėtų medyje įrašų. Tikrinama ar ryšys nebuvo užmegztas prieš tai. Jei taip leidžiama jį redaguoti.
- CRUD ryšių peržiūrai naudojamos CrossTable tipo lentelės. Pateikiamą informaciją galima filtruoti pagal funkciją arba terminą.

1 lentelė. CRUD matricos pavyzdys.

Terminas	Funkcija: Nustatyti automobilio vertę			
	C	R	U	D
Modelis		*		
Pagaminimo metai		*		
Rida		*		
Vertė	*	*	*	

Reikalavimai veiklos taisyklių šablonų įvedimo posistemėi:

Privalo leisti kurti bei koreguoti jau sukurtus veiklos taisyklių šablonus. Šablonai susideda iš tam tikra seka išdėstytų elementų. Šie elementai turi savo tipus bei raktažodžius. Todėl būtinos priemonės papildyti arba pakoreguoti tipų bei raktažodžių lenteles. Šablonai kuriami tam tikram veiklos taisyklių tipui. Kiekvienas veiklos taisyklės tipas gali turėti keletą šablonų. Šablono pavadinimą patartina formuoti taip, kad atspindėtų jį sudarančius elementus. Pradinį šablono pavadinimą pasiūlo pati sistema. Formuojant šabloną sistema išveda jo elementus sakinio struktūroje, kad galima būtų įsivaizduoti, kaip galėtų atrodyti pagal jį suformuota veiklos taisyklė.

Reikalavimai veiklos taisyklių įvedimo posistemėi:

Veiklos taisyklės įvedamos pasirinkus šabloną. Sistema dinamiškai pagal šablono elementus formuoja įvedimo laukus bei pasirinkimo sąrašus, išdėstydama juos pagal elementų eilės tvarką. Susijęs modelio elementas pasirenkamas

iš sąrašų (jei elementai sudaro hierarchiją tai iš medžio tipo sąrašų). Veiklos taisyklės išraiškos komponentų yra tiek kiek ir šablono elementų. Veiklos taisyklės išraiškos komponentų išraiškos priklauso nuo tipo, jei tai raktažodis tai išraiška bus konkretus nuo šablono elemento

Literatūros sąrašas

- [1] **K. Kapočius, R. Butleris.** Business Rules driven approach for elicitation of IS requirements. Proceedings of World Multiconference on Systemics, Cybernetics and Informatics '2005. Priimta spaudai.
- [2] **Business Rules Group, The.** Defining Business Rules ~ What Are They Really? (final report). http://www.businessrulesgroup.org/first_paper/BRG-whatIsBR_3ed.pdf. 2000.
- [3] **R. G. Ross.** Principles of the Business Rules Approach. Addison-Wesley. 2003.
- [4] **B. von Halle.** Business Rules Applied: Building Better Systems Using the Business Rules Approach. John Wiley & Sons, New York. 2001.

Implementation of the repository system for the IS requirements specification based on the business rules approach

The success of requirements analysis often decides the quality of the finished information system (IS). Application of ideas behind the business rules approach for requirements analysis could help narrow the gap between business representatives and analysts thus paving the way to a better requirements specification. In this paper a framework for business rules driven IS requirements specification is presented. The proposed approach is based on discovery of functions, business decisions and underlying business rules. The management of captured requirements as well as the generation of relevant documentation is performed using the repository system, major implementation guidelines of which are also discussed.