



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Neringa Norkutė

**VEIKLOS APRIBOJIMŲ AUTOMATIZUOTAS IŠGAVIMAS IŠ
REIKALAVIMŲ MODELIO**

Baigiamasis magistro projektas

Vadovė
lekt. dr. L. Bisikirskienė

KAUNAS, 2016

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

VEIKLOS APRIBOJIMŲ AUTOMATIZUOTAS IŠGAVIMAS
IŠ REIKALAVIMŲ MODELIO

Baigiamasis magistro projektas
Informacinių sistemų inžinerijos studijų programa (kodas 621E15001)

Vadovė

lekt. dr. L. Bisikirskienė
2016-05-20

Recenzentas

doc. dr. V. Rudžionis
2016-05-20

Projektą atliko

Neringa Norkutė
2016-05-20



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

(Fakultetas)

Neringa Norkutė

(Studento vardas, pavardė)

Informacinių sistemų inžinerijos studijų programa, 621E15001

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Veiklos apribojimų automatizuotas išgavimas iš reikalavimų modelio“

AKADEMINIO SAŽINGUMO DEKLARACIJA

20 16 m. gegužės 20 d.
Kaunas

Patvirtinu, kad mano, **Neringos Norkutės**, baigiamasis projektas tema „Veiklos apribojimų automatizuotas išgavimas iš reikalavimų modelio“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

Norkutė, Neringa. Veiklos apribojimų automatizuotas išgavimas iš reikalavimų modelio. Magistro baigiamasis projektas / vadovė lekt. dr. Lina Bisikirskienė; Kauno technologijos universitetas, Informatikos fakultetas.

Mokslo kryptis ir sritis: Informatikos inžinerija, technologijos mokslai

Reikšminiai žodžiai: *veiklos apribojimai, veiklos modelis, reikalavimų modelis*

Kaunas, 2016. 79 p.

SANTRAUKA

Veiklos procesų analitikas ne visuomet gali tiksliai nustatyti veiklos procese egzistuojančius apribojimus, todėl labai svarbu užtikrinti automatizuotą veiklos apribojimų identifikavimą ir pateikimą organizacijos veiklos modelyje. Taip pat, svarbu išlaikyti vieningą informaciją modeliuojant skirtingus organizacijos veiklos aspektus.

Automatinio veiklos apribojimų išgavimo iš reikalavimų modelio metodo sudarymui analizuojamos reikalavimų ir veiklos procesų modeliavimo galimybės, meta-modeliai, modeliavimo kalbos ir specialūs įrankiai. Sudaromas tam skirtas algoritmas panaudojant veiklos žodyno ir veiklos taisyklių generavimą. Siekiant įsitikinti automatizuoto veiklos apribojimų algoritmo veikimu atliekama prototipo realizacija, leidžianti iš reikalavimų modelio išgauti veiklos žodyno elementus.

Realizavus automatinį veiklos apribojimų išgavimo iš reikalavimų modelio metodą sistemos architektams sudaroma galimybė specifiškai teisingus veiklos modelius remiantis veiklos apribojimais. Be to, veiklos žodyno ir veiklos taisyklių elementai gali būti panaudojami modeliuojant ir kitus organizacijos aspektus. Sudarytas veiklos apribojimų automatizuoto formavimo iš reikalavimų modelio metodas. Metodo pagrindą sudaro detalios veiklos diagramos skirtingų elementų tekstų analizavimui, aprašančios, iš *SysML* reikalavimų modelių veiklos žodyno ir veiklos taisyklių sąrašo formavimą bei, naudojantis *SBVR* terminologija natūralia kalba leidžiančios jį pateikti vartotojui.

Realizuotas prototipas automatinio veiklos žodyno elementų formavimo algoritmas, kurio pagrindas yra reikalavimų modelio elementų tekstinės informacijos nuskaitymas, elementų pavadinimų ir tekstų nagrinėjimas pagal sudarytus algoritmus atsižvelgiant į anglų kalbos taisykles, reikalavimų modelio sudarymo apribojimus ir rekomendacijas. Išgauta veiklos apribojimų informacija pateikiama lentelių formoje ir sudaryta galimybė vartotojui rankiniu būdu koreguoti rezultatus.

Norkute, Neringa. Automatic Extraction of Business Constraints from Requirement Model: Master's thesis in Information Systems Engineering / supervisor dr. Lina Bisikirskienė. The Faculty of Informatics, Kaunas University of Technology.

Research area and field: Informatics Engineering, Technology Science

Key words: business constraints, business model, requirement model

Kaunas, 2016. 79 p.

SUMMARY

Business process analyst might not always accurately identify the constraints existing in the operational process, so it is very important to ensure that identification of the restrictions is automated and presented in the context of the organization's business model. Also, it is important to maintain a consistent information while modelling different aspects of the organization's activities.

In order to develop a method for automatic extraction of the restrictions from the requirements model, analysis requirements and business process modelling options, meta-models, modelling languages and special tools are analysed. Developed algorithm is using generated business vocabulary and business rules. In order to verify performance of the algorithm a prototype is created, able to extract business vocabulary items from requirements model.

Implementing automatic restrictions extraction from the requirements model enables system architects to specify the correct business models based on activity limitations. The method is based on detailed activity diagrams that describe a way to acquire business vocabulary and business rules by analysing the texts in elements of the SysML requirement models. Use of the SBVR terminology enables results to be presented to user in a natural language. Furthermore, the business vocabulary and business rules can be used to model other aspects of the organization.

Implemented prototype of algorithm automatically forms business vocabulary elements. The algorithm works by analysing text obtained from requirements model elements. Element names and texts are processed according to the English language rules and following the constraints and recommendations for compiling requirements model. Extracted restrictions are presented in tabular form and allows the user to manually correct the results.

TURINYS

Lentelių sąrašas	8
Paveikslų sąrašas	9
ĮVADAS	10
1. VEIKLOS APRIBOJIMŲ IŠGAVIMO IŠ REIKALAVIMŲ MODELIO METODŲ ANALIZĖ	12
1.1. Analizės tikslas	12
1.2. Tyrimo objekto naudotojų analizė	12
1.2.1. Naudotojų tipai ir charakteristikos	12
1.2.2. Naudotojų tikslai ir problemos	12
1.3. Reikalavimų modeliavimo analizė	13
1.3.1. Reikalavimų samprata	13
1.3.2. Reikalavimų specifikavimo šablonų analizė	14
1.3.3. Reikalavimų modeliavimo analizė	18
1.3.4. Reikalavimų valdymo įrankių analizė	22
1.3.5. Reikalavimų modeliavimo metodų palyginimas	24
1.4. Veiklos modeliavimo analizė	25
1.4.1. Veiklos samprata ir struktūra	25
1.4.2. Veiklos modeliavimo kalbų analizė	26
1.4.3. Veiklos modeliavimo metodų palyginimas	30
1.5. Veiklos apribojimų sudarymo analizė	31
1.5.1. SBVR standartas	32
1.6. Analizės išvados	34
2. VEIKLOS APRIBOJIMŲ AUTOMATIZUOTO IŠGAVIMO IŠ REIKALAVIMŲ MODELIO METODAS	35
2.1. Panaudojimo atvejų modelis	35
2.2. Duomenų struktūra	36
2.2.3. Reikalavimų duomenų struktūra	36
2.2.4. Veiklos žodyno ir veiklos taisyklių duomenų struktūra	37
2.2.5. Sistemos duomenų struktūra	38
2.3. Veiklos apribojimų automatizuoto išgavimo iš reikalavimų modelio metodo algoritmas	40
2.3.3. Įkelti reikalavimų modelį	41
2.3.4. Sudaryti veiklos žodyną	42
2.3.5. Išrinkti tinkamus reikalavimų modelio elementus	43
2.3.6. Redaguoti reikalavimų modelio elementų pasirinkimą	44
2.3.7. Suformuoti žodžių sąrašą iš elementų pavadinimų	45
2.3.8. Suformuoti žodžių sąrašą iš elementų tekstų	45
2.3.9. Suformuoti veiklos žodyno elementus	47
2.3.10. Redaguoti veiklos žodyno elementus	50
2.3.11. Sudaryti veiklos taisykles	51
2.4. Apribojimai ir rekomendacijos	56

3. VEIKLOS APRIBOJIMŲ AUTOMATIZUOTO IŠGAVIMO IŠ REIKALAVIMŲ MODELIO ALGORITMO TESTAVIMAS, REALIZACIJA IR REZULTATAI.....	58
3.1. Algoritmo testavimo etapai	58
3.2. Prototipo realizacija, funkcionalumas	59
3.2.3. Reikalavimų modelio duomenų nuskaitymas ir atvaizdavimas.....	59
3.2.4. Reikalavimų modelio elementų sąrašo redagavimas ir papildymas	61
3.2.5. Veiklos žodyno sudarymas.....	61
3.2.6. Veiklos žodyno faktų sudarymas.....	62
3.3. Prototipo testavimas	63
3.4. Prototipo eksperimentinis tyrimas kiekybiniais ir kokybiniais aspektais.....	63
3.4.3. Reikalavimų modelio „ <i>Library information system</i> “ analizė	65
3.4.4. Reikalavimų modelio „ <i>Book’s information sub-system</i> “ analizė	68
3.4.5. Reikalavimų modelio „ <i>User information sub-system</i> “ analizė	72
3.4.6. Apibendrinti eksperimento rezultatai	76
3.5. Prototipo kūrimo ir testavimo rezultatai	77
4. IŠVADOS	78
5. LITERATŪRA	79

LENTELIŲ SĄRAŠAS

2.3.1 lentelė Matematinųjų simbolių konvertavimo pavyzdžiai.....	46
2.3.2 lentelė Veiklos žodyno formavimo pavyzdys Nr.1	48
2.3.3 lentelė Veiklos žodyno formavimo pavyzdys Nr.2	48
2.3.4 lentelė Veiklos žodyno formavimo pavyzdys Nr.3	48
2.3.5 lentelė Veiklos žodyno formavimo pavyzdys Nr.4	48
2.3.6 lentelė Veiklos žodyno formavimo pavyzdys Nr.5	49
2.3.7 lentelė Fakto tipų formavimo pavyzdys Nr.1	49
2.3.8 lentelė Fakto tipų formavimo pavyzdys Nr.2	49
2.3.9 lentelė Fakto tipų formavimo pavyzdys Nr.3	49
2.3.10 lentelė Veiklos taisyklės formuluotės parinkimas	52
2.3.11 lentelė Veiklos taisyklės sudarymo pavyzdys Nr.1	52
2.3.12 lentelė Veiklos taisyklės sudarymo pavyzdys Nr.2	52
2.3.13 lentelė Veiklos taisyklės sudarymo pavyzdys Nr.3	52
2.3.14 lentelė Veiklos taisyklės sudarymo pavyzdys Nr.4	53
2.3.15 lentelė Veiklos taisyklės sudarymo pavyzdys Nr.5	53
2.3.16 lentelė Veiklos taisyklės sudarymo pavyzdys Nr.6	55
2.3.17 lentelė Veiklos taisyklės sudarymo pavyzdys Nr.7	55
2.4.1 lentelė SBVR elementų pernaudojimas sistemos modeliui	57
3.1 lentelė Veiklos žodyno elementų analizė iš reikalavimų modelio „ <i>Library information system</i> “	66
3.2 lentelė Veiklos žodyno elementų analizė iš reikalavimų modelio „ <i>Library information system</i> “	67
3.3 lentelė Veiklos žodyno elementų analizė iš reikalavimų modelio „ <i>Book's information sub-system</i> “	69
3.4 lentelė Veiklos žodyno elementų analizė iš reikalavimų modelio „ <i>Book's information sub-system</i> “	71
3.5 lentelė Veiklos žodyno elementų analizė iš reikalavimų modelio „ <i>User information sub-system</i> “	73
3.6 lentelė Veiklos žodyno elementų analizė iš reikalavimų modelio „ <i>User information sub-system</i> “	75
3.7 lentelė Apibendrinti eksperimento rezultatai	76

PAVEIKSLŲ SĄRAŠAS

1 pav. <i>Volere</i> šablono forma funkciniais ir nefunkciniais reikalavimams specifiuoti [11]	17
2 pav. <i>SysML</i> reikalavimo apibrėžimo modelis [1]	18
3 pav. <i>Doors</i> duomenų bazės struktūra [13].....	22
4 pav. veiklos proceso struktūra [4].....	25
5 pav. Pagrindinės <i>BPMN</i> elementų grupės [3].....	27
6 pav. <i>ArchiMate</i> modeliavimo karkasas [8].....	28
7 pav. <i>SBVR</i> standarto pagrindinė idėja.....	32
8 pav. Veiklos apribojimų automatizuoto išgavimo iš reikalavimų modelio UML panaudojimo atvejų modelis	35
9 pav. Reikalavimų duomenų struktūra.....	37
10 pav. Veiklos žodyno ir veiklos taisyklių duomenų struktūra	37
11 pav. Sistemos modelio duomenų struktūra.....	38
12 pav. Sistemoje saugomų reikalavimų duomenų struktūra	39
13 pav. Sistemoje saugomo veiklos žodyno ir veiklos taisyklių duomenų struktūra	39
14 pav. Apibendrintas sistemos veikimo algoritmas	40
15 pav. „1 Įkelti reikalavimų modelį“ algoritmas	41
16 pav. „2 Sudaryti veiklos žodyną“ algoritmas	42
17 pav. „2.1 Išrinkti tinkamus <i>SysML</i> diagramos elementus“ algoritmas	43
18 pav. „2.2 Redaguoti <i>SysML</i> diagramos elementų parinkimą“ algoritmas	44
19 pav. „2.3 Suformuoti žodžių sąrašą iš elementų pavadinimų“ algoritmas	45
20 pav. „2.4 Suformuoti žodžių sąrašą iš elementų tekstų“ algoritmas.....	46
21 pav. „2.5 Suformuoti veiklos žodyno elementus“ algoritmas	47
22 pav. „2.6 Redaguoti veiklos žodyno elementus“ algoritmas	50
23 pav. „3 Sudaryti veiklos taisykles “ algoritmas	51
24 pav. „3.4 Sudaryti sudėtinės veiklos taisykles“ algoritmas	54
25 pav. Sistemos vartotojo sąsajos langas	59
26 pav. Reikalavimų modelio parinkimas	60
27 pav. Reikalavimų modelio nuskaitymas	60
28 pav. Reikalavimų modelio elementų parinkimo langas.....	61
29 pav. Veiklos žodyno faktų sudarymas.....	62
30 pav. Reikalavimų diagrama „ <i>Library information system</i> “	65
31 pav. Reikalavimų diagrama „ <i>Book’s information sub-system</i> “	68
32 pav. Reikalavimų diagrama „ <i>User information sub-system</i> “	72

IVADAS

Šiais laikais didžioji dalis su informacinėmis sistemomis susijusių organizacijų siekia kuo konkrečiau apibrėžti organizacijos veiklą. Vienas efektyviausių metodų yra grafinių modelių panaudojimas. Labai svarbu, kad organizacijos veiklos modeliai atitiktų realaus pasaulio organizacijos veiklos procesus. Tačiau analitikai ne visais atvejais gali tiksliai sudaryti veiklos modelį. Taip yra dėl to, kad dažniausiai yra sudėtinga užtikrinti, kad bus specifikuoti visi veiklos procesų apribojimai.

Vadovaujantis korektišku veiklos procesų modeliu kiekvienas organizacijos darbuotojas gali suprasti savo ir visos organizacijos veiklą, suvokti visą veiklos procesą. Kadangi ne visuomet veiklos procesų analitikas gali tiksliai nustatyti veiklos procese egzistuojančius apribojimus, yra labai svarbu užtikrinti automatizuotą veiklos apribojimų identifikavimą ir pateikimą organizacijos veiklos modelyje. Taip pat, svarbu išlaikyti vieningą informaciją modeliuojant kitus organizacijos veiklos aspektus.

Dėl to, siekiant informacijos vieningumo skirtinguose organizacijos modelio aspektuose, reikia sudaryti organizacijos veiklos žodyną, kuris specifikuoja pagrindinius organizacijos faktus ir terminus bei organizacijos veiklos taisykles, kurios apibrėžia organizacijos veiklos apribojimus. Kadangi, organizacija turi didelės apimties reikalavimų specifikacijas, apibrėžiančias organizacijos veiklos reikalavimus, reikalingas automatinis veiklos apribojimų išgavimas iš reikalavimų modelių. Šiam tikslui pasiekti tikslinga automatizuotai sudaryti veiklos žodyną ir veiklos taisykles iš reikalavimų modelio be informacijos semantinių nuostolių.

Automatinio veiklos apribojimų išgavimo iš reikalavimų modelio metodo sudarymui analizuojamos reikalavimų ir veiklos procesų modeliavimo galimybės, meta-modeliai, modeliavimo kalbos ir specialūs įrankiai. Sudaromas tam skirtas algoritmas panaudojant veiklos žodyno ir veiklos taisyklių generavimą. Siekiant įsitikinti automatizuoto veiklos apribojimų algoritmo veikimu atliekama prototipo realizacija, leidžianti iš reikalavimų modelio išgauti veiklos žodyno elementus.

Realizavus automatinį veiklos apribojimų išgavimo iš reikalavimų modelio metodą sistemos architektams sudaroma galimybė specifikuoti teisingus veiklos modelius remiantis veiklos apribojimais. Be to, veiklos žodyno ir veiklos taisyklių elementai gali būti panaudojami modeliuojant ir kitus organizacijos aspektus.

Darbo tikslas ir uždaviniai

Darbo tikslas yra automatinio būdu išgauti veiklos apribojimus iš reikalavimų modelio. Šiam tikslui pasiekti suformuluoti uždaviniai:

1. Išanalizuoti reikalavimų modeliavimo standartus, struktūras bei įrankius. Išanalizuoti veiklos modeliavimo standartus, struktūras bei įrankius. Išanalizuoti veiklos apribojimų specifikavimo galimybes.
2. Sudaryti veiklos apribojimų automatinio išgavimo iš reikalavimų modelio principus ir metodą.
3. Sudaryti prototipą, leidžiantį įvertinti veiklos apribojimų išgavimo metodą.
4. Atlikti prototipo eksperimentinį tyrimą, leidžiantį įvertinti realizuoto metodo tikslumą.

Tyrimo objektas, sritis ir problema

Tyrimo sritis apima reikalavimų ir veiklos modeliavimą bei modeliavimui naudojamus standartus, šablonus ir įrankius.

Tyrimo objektas – automatinis veiklos apribojimų išgavimas iš reikalavimų modelio.

Tyrimo problemą sudaro veiklos modeliavimas remiantis reikalavimuose nurodyta informacija.

1. VEIKLOS APRIBOJIMŲ IŠGAVIMO IŠ REIKALAVIMŲ MODELIO METODŲ ANALIZĖ

1.1. Analizės tikslas

Analizuojamas tyrimo objektas bei tyrimo objekto naudotojų analizė. Analizės tikslas yra išsiaiškinti reikalavimų modeliavimo galimybes, meta-modelius, šablonus bei įrankius. Tokiu būdu nustatyti ir identifikuoti reikalavimuose saugomą informaciją, kuri aprašo veiklos apribojimus. Toliau, taip pat, išanalizuoti veiklos modeliavimo galimybes, standartus bei įrankius.

1.2. Tyrimo objekto naudotojų analizė

1.2.1. Naudotojų tipai ir charakteristikos

Suprojektuotas veiklos apribojimų automatinio išgavimo iš reikalavimų modelio prototipas skiriamas skirtingoms naudotojų grupėms, kurios atsiranda kuriant programinę įrangą ar informacinę sistemą. Šios naudotojų grupės glaudžiai vykdo savo specifinius darbus. Taigi, išskiriami tokie naudotojai:

- Analitikai;
- Sistemos architektai.

Analitikai apklausos būdais turi išgauti kuriamos programinės įrangos arba informacijos sistemos savybes, funkcijas bei kitus aktualius parametrus iš užsakovo, kuriais remdamiesi sudaro reikalavimų specifikaciją. Toliau, pagal patvirtintą būsimos sistemos reikalavimų dokumentą, analitikai turi sudaryti reikalavimų modelį. Tuomet, sistemos architektai remdamiesi reikalavimų modeliu turi sudaryti sistemos veiklos modelį, vėliau reikalingą sistemos kūrėjams. Šio modelio kūrimo etape architektams yra svarbu iš reikalavimų modelio išgauti visus veiklos apribojimus.

1.2.2. Naudotojų tikslai ir problemos

Kuriamo veiklos apribojimų automatinio išgavimo iš reikalavimų modelio prototipo pagrindiniai naudotojai yra analitikai ir architektai. Analitikų svarbiausias tikslas atlikti užsakovo poreikių analizę ir tiksliai sudaryti reikalavimų modelį, kuris aiškiai parodytų, kokie veiklos apribojimai aktualūs.

Sistemos architektai siekia pagal reikalavimų informaciją sudaryti tokį veiklos modelį, kuris atitiktų reikalavimus. Tačiau tikėtina, kad laikui bėgant užsakovo poreikiai pasikeis, o tada reikės rankiniu būdu perdaryti reikalavimų modelį ir atsekti, kuriems veiklos modeliams šie pasikeitimai turės įtaką. Siekiama išvengti rankinio poreikių specifikavimo ir netikslumų, kurie atsiranda tiesiogiai skaitant reikalavimų dokumentą ir rankiniu būdu sudarant veiklos modelius. Be to, siekiama sumažinti laiko sąnaudas, reikalingas veiklos modeliams sudaryti.

1.3. Reikalavimų modeliavimo analizė

1.3.1. Reikalavimų samprata

Reikalavimų inžinerija – tai procesas, apimantis sistemos ar programinės įrangos reikalavimų surinkimą, analizę, dokumentavimą ir valdymą sistemos kūrimo metu [6]. Reikalavimai gali būti išreiškiami formalia ir neformalia kalba. Prie neformalios kalbos galima priskirti ir paprasčiausią dialogą, kurio metu paslaugos teikėjas išklauso kliento norus paslaugos tipui, kokybei, kiekiui ir kt. Formalus reikalavimas yra apibrėžiamas formalia kalba, formulėmis ar funkcijomis, įteisinamas dokumentais ir pan. Reikalavimai gali būti nuo abstrakčių teiginių iki konkrečių detalizuotų matematinių funkcijų.

Reikalavimas – specifikacijoje, sutartyje ar standarte dokumentuota objekto ar proceso savybė [1].

IEEE (angl. *Institute of Electrical and Electronics Engineers*) terminologijos žodyne reikalavimas gali būti apibrėžiamas [1]:

- Sąlyga arba gebėjimas reikalingas vartotojui tam tikrai problemai išspręsti arba tam tikram tikslui pasiekti.
- Dokumentuota sąlygos arba gebėjimo reprezentacija.
- Sąlyga, kurią turi įvykdyti arba gebėjimas, kuri turi turėti sistema arba sistemos komponentė, kad patenkintų formaliai aprašytą specifikaciją, kontraktą ar standartą.

Iš esmės, reikalavimai gali būti suprantami kaip paslaugos, kurias sistema turi teikti ir apribojimai, kurių privalu laikytis sprendžiant tam tikras problemas.

Apibrėžimai apima ir vartotojo, ir vykdytojo požiūrius į reikalavimą. Reikalavimus galima klasifikuoti į vartotojo reikalavimus, kurie įvardijami aukštu abstrakcijos lygmeniu, ir sistemos ar programinės įrangos reikalavimus, kuriais detalai aprašoma, ką ir kaip realizuota sistema ar programinė įranga turėtų įgyvendinti. Sistemos ar programinės įrangos reikalavimai dar skirstomi į funkcinis ir nefunkcinis reikalavimus [6].

Funkciniai reikalavimai nusako sistemos funkcionalumą, apibrėžia, ką sistema turi daryti, kokios vartotojui svarbios funkcijos turi būti atliekamos ir kokiais duomenimis manipuluoti siekiant atlikti nurodytas funkcijas. Nefunkciniai reikalavimai – tai sistemos funkciniais reikalavimais numatytų funkcijų kokybinės savybės. Šie reikalavimai nusako sistemos savybes, kuriomis ji turi pasižymėti realizuodama savo funkcionalumą. Tokia klasifikacija leidžia skirtingiems interesantams pristatyti skirtingus aspektus.

1.3.2. Reikalavimų specifikavimo šablonų analizė

SRS šablonas (IEEE STD 830-1998 standartas)

Sistemų ir programinės įrangos reikalavimų specifikavimo *SRS* (angl. *Software Requirements Specification*) šablonas apibrėžiamas *IEEE STD 830-1998* standartu. *SRS* reikalavimų specifikavimo šabloną galima įvardyti kaip kliento reikalavimų supratimą iš organizacijos pusės tam tikru laiko momentu. Tai yra abipusis apsidraudimo dokumentas, užtikrinantis organizacijos ir kliento reikalavimų tarpusavio suvokimą.

Rekomenduojamas *SRS* šablono turinys apima pagrindinius sistemos kūrimo klausimus ir svarbiausias kuriamos sistemos specifikas [9, 10]:

- Įžanga (angl. *introduction*):
 - Tikslas (angl. *purpose*);
 - Sudėtis (angl. *scope*);
 - Apibrėžtys ir santrumpos (angl. *definitions, acronyms and abbreviation*);
 - Nuorodos (angl. *references*);
 - Apžvalga (angl. *overview*);
- Visapusiškas aprašas (angl. *overall description*):
 - Produkto perspektyva (angl. *product perspective*);
 - Produkto funkcijos (angl. *product functions*);
 - Naudotojo charakteristikos (angl. *user characteristics*);
 - Apribojimai (angl. *constraints*);
 - Prielaidos ir priklausomybės (angl. *assumptions and dependencies*);
 - Reikalavimų padalijimas (angl. *apportioning of requirements*);
- Specifiniai reikalavimai (angl. *specific requirements*):
 - Išorinės sąsajos (angl. *external interfaces*);
 - Funkcijos (angl. *functions*);
 - Eksploataciniai reikalavimai (angl. *performance requirements*);
 - Duomenų bazės reikalavimai (angl. *logical database requirements*);
 - Išvaizdos apribojimai (angl. *design constraints*);
 - Programinės įrangos atributai (angl. *software system attributes*);
 - Reikalavimų organizavimas (angl. *organizing the specific requirements*).

Remiantis šiuo šablonu reikalavimai įvardijami deklaratyviais teiginiais, kurių tekstas paremtas natūralia kalba. SRS šablonas apima tik funkcinis ir nefunkcinis sistemos reikalavimus, jame neaprašomi jokie dizaino sprendimo pasiūlymai, galimi technologiniai sprendimai ar kita dalykinės srities informacija, kurią klientas gali pateikti. Funkciniais reikalavimais yra nusakomi sistemos panaudojimo atvejai. SRS dokumento struktūra atspindi specifikuojamų reikalavimų specifiką. Taigi, specifikuoti reikalavimai skirstomi kategorijomis [13]:

1. Funkciniai (angl. *functional*) reikalavimai;
2. Nefunkciniai (angl. *non-functional*) reikalavimai:
 - Našumo (angl. *performance*);
 - Vartotojo sąsajos (angl. *interface*);
 - Veikimo (angl. *operational*);
 - Resursų (angl. *resource*);
 - Patvirtinimo (angl. *verification*);
 - Priėmimo (angl. *acceptance*);
 - Dokumentacijos (angl. *documentation*);
 - Kokybės (angl. *quality*);
 - Saugumo (angl. *safety*);
 - Patikimumo (angl. *reliability*);
 - Palaikymo (angl. *maintainability*).

Reikalavimų specifikacija, sudaryta pagal SRS šabloną, yra pilna, nedviprasmiška, nuosekli, reikalavimai pasižymi atsekamumu.

Volere šablonas

Volere (it. *volere* – norėti) šablonas yra ilgametis reikalavimų inžinerijos praktikos rezultatas. *Volere* reikalavimų specifikacija – tai reikalavimų konteineris, kuriame specifikuojami sistemos ar kuriamos programinės įrangos reikalavimai yra klasifikuojami skyriais, kurių kiekvienas turi apibrėžtus nepersidengiančius poskyrius [11]:

- Sistemos iniciatoriai (angl. *project drivers*):
 - Sistemos paskirtis (angl. *the purpose of the project*);
 - Suinteresuoti asmenys (angl. *the stakeholders*);
- Sistemos apribojimai (angl. *project constraints*):
 - Apribojimai reikalavimams (angl. *mandated constraints*);
 - Terminai ir apibrėžtys (angl. *naming conventions and terminology*);
 - Svarbūs faktai ir prielaidos (angl. *relevant facts and assumptions*);

- Funkciniai reikalavimai (angl. *functional requirements*):
 - Veiklos sudėtis (angl. *the scope of the work*);
 - Duomenų modelis ir jo elementai (angl. *business data model and data dictionary*);
 - Sistemos sudėtis (angl. *the scope of the project*);
 - Funkciniai reikalavimai (angl. *functional requirements*);
- Nefunkciniai reikalavimai (angl. *non-functional requirements*):
 - Reikalavimai sistemos išvaizdai (angl. *look and feel requirements*);
 - Reikalavimai panaudojamumui (angl. *usability and humanity requirements*);
 - Reikalavimai vykdymo savybėms (angl. *performance requirements*);
 - Reikalavimai veikimo sąlygoms (angl. *operational and environmental requirements*);
 - Reikalavimai sistemos priežiūrai (angl. *maintainability and support requirements*);
 - Reikalavimai saugumui (angl. *security requirements*);
 - Kultūriniai – politiniai reikalavimai (angl. *cultural and political requirements*);
 - Teisiniai reikalavimai (angl. *legal requirements*);
- Projekto išeiga (angl. *project issues*):
 - Atviri klausimai (angl. *open issues*);
 - Egzistuojantys sprendimai (angl. *off-the-shelf solutions*);
 - Naujos problemos (angl. *new problems*);
 - Uždaviniai (angl. *tasks*);
 - Pritaikymas naujame produkte (angl. *migration to the new product*);
 - Rizikos įvertinimas (angl. *risks*);
 - Kaina (angl. *costs*);
 - Vartotojo dokumentacija ir apmokymas (angl. *user documentation and training*);
 - Perspektyviniai reikalavimai (angl. *waiting room*);
 - Idėjos sprendimams (angl. *ideas for solutions*).

Esant reikalui, galima koreguoti šablono skyrius, panaikinant nereikalingus ar pridėdant naujus skyrius, kurie leistų aprašyti specifines dalykinės srities charakteristikas. Rekomenduojama, reikalavimus pateikti tekstone forma trumpais nesudėtiniais sakiniais siekiant reikalavimų specifikacijos pilnumo ir išvengti dubliavimosi ir dviprasmybių. Funkciniams ir nefunkciniams reikalavimams apibrėžti pateikiama speciali forma (1 pav.).

Reikalavimas #:	Reikalavimo tipas:	Įvykis/panaudojimo atvejis #:
Aprašymas:		
Pagrindimas:		
Šaltinis:		
Tikimo kriterijus:		
Užsakovo patenkinimas:	Užsakovo nepatenkinimas:	
Priklausomybės:	Konfliktai:	
Papildoma medžiaga:		
Istorija:		

1 pav. Volere šablono forma funkciniam ir nefunkciniam reikalavimams specifikuoti [11]

Volere reikalavimų specifikacijos formoje reikalavimams identifikuoti suteikiami trys atributai: unikalus ID numeris, nurodomas *Volere* reikalavimo tipas pagal šablono skyriaus numerį ir kiekvienas reikalavimas siejamas su panaudojimo atvejais pagal jų eilės numerį. Reikalavimo aprašymas skirtas nusakyti reikalavimo paskirčiai, o pagrindimas – reikalavimo reikalingumui pagrįsti. Tikimo kriterijus yra kvantifikuotas tikslas, kurį kuriama sistema turės tenkinti. Nors reikalavimas aprašomas vartotojo sąvokomis, tačiau tikimo kriterijus rašomas tiksliai nusakant jo matą, kad priimtus sprendimus būtų galima testuoti reikalavimo tenkinimo aspektu. Papildomai, specifikuojant reikalavimą nurodomos jo priklausomybės ir konfliktai su kitais egzistuojančiais reikalavimais, šaltinis ir kita svarbi papildoma medžiaga [11].

Volere šablonas pritaikytas darbui su reikalavimų valdymo įrankiais, tokiais kaip, *DOORS* (angl. *Dynamic object oriented requirements system*), *IBM Rational RequisitePro*. Specifikuojant reikalavimus tam tikruose *Volere* šablono poskyriuose, juos galima papildyti veiklos procesų, panaudojimo atvejų, duomenų srautų, esybių ryšių ir kitais modeliais [2].

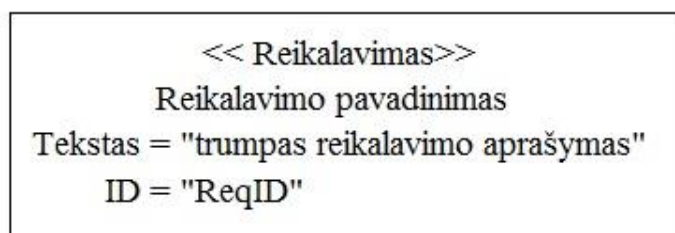
1.3.3. Reikalavimų modeliavimo analizė

SysML modeliavimo kalba

SysML (angl. *The Systems Modeling Language*) – standartizuota reikalavimų modeliavimo kalba, kuri palaiko techninės ir programinės įrangos, informacijos duomenų srautų, personalo, paslaugų bei procesų specifikavimą, dizainą, analizę ir verifikavimą. Be to, *SysML* apibrėžiama kaip *UML* (angl. *Unified Modeling Language*) standarto dialektas [1]. Ši kalba sukurta *UML* pagrindu, panaudojant jos konstrukcijas, tačiau pagrindinis skirtumas yra tas, kad ji pritaikyta būtent sistemų inžinerijai [6].

SysML vartotojo ar sistemos reikalavimams modeliuoti ir organizuoti pateikia specialią reikalavimų diagramą, kuri yra vienas didžiausių privalumų prieš *UML* modeliavimo kalbos teikiamas galimybes. *SysML* reikalavimų diagramoje reikalavimai yra vaizduojami standartizuota apibrėžta semantika, kurios laikantis aiškiai parodomi reikalavimų ryšiai. Dar vienas privalumas, kad yra apibrėžta semantika reikalavimų diagramos susiejimui su kitais sistemos modeliais. *SysML* užtikrina reikalavimų atsekamumą, kurio pasekmėje, galima užtikrinti, kad sistema ar jos komponentai reikalavimus įgyvendina. Vienas iš reikalavimų atsekamumo būdų yra *SysML* reikalavimų lentelės.

SysML modeliavimo kalboje reikalavimai vaizduojami kaip modelio elementai, kitaip tariant, kaip sistemos elementai. Toks reikalavimų modeliavimas papildo kitus *SysML* pateikiamus modelius. Tokiu būdu, diagrama užpildo spragas tarp pernelyg dviprasmiškų, natūralia kalba formuojamų, reikalavimų ir *UML* panaudojimo atvejų diagramos. *SysML* reikalavimų diagrama gali būti panaudojama reikalavimų specifikacijoms standartizuoti. Joje reikalavimams registruoti pateikiamas specialus reikalavimo apibrėžimo modelis (2 pav.), kuris sumažina daugumoje specifikacijų paliekamą reikalavimų apibrėžimo laisvę [6].



2 pav. SysML reikalavimo apibrėžimo modelis [1]

SysML reikalavimų diagrama numato kelis būdus pavaizduoti reikalavimų sąsajas – tai apima reikalavimų hierarchiją ir kitus reikalavimų ryšių tipus [1]:

- Sudėties (angl. *containment*);
- Išvesties (angl. *derive*);
- Kopijavimo (angl. *copy*);
- Tenkinimo (angl. *satisfy*);

- Papildymo (angl. *refine*);
- Verifikavimo (angl. *verify*);
- Atsekamumo (angl. *trace*);
- Pagrindimas (angl. *rationalte*);
- Problema (angl. *problem*).

SysML modeliavimo kalboje leidžiama sudėtingus reikalavimus skaidyti į smulkesnius sudarant jiems hierarchijos ryšius. Tokiu būdu, nuo pat sistemos kūrimo pradžios bet kokie aukšto lygio dalykinės srities reikalavimai gali būti palaipsniui dekomponuojami į detalius sistemos reikalavimus. Toks hierarchinis konceptas įgalina reikalavimų pernaudojimą. Pernaudojant reikalavimus sudaromas kopijavimo (angl. *copy*) ryšys, kuris aiškinamas kaip „vergvaldžio ir vergo“ ryšys, kuomet suprantama, kad nukopijuotasis yra „vergas“ ir originalusis reikalavimas – „vergvaldys“.

Išvesties (angl. *derive*) ryšiu parodoma, kurio reikalavimo pagrindu tam tikras reikalavimas yra iškeltas. Toks ryšys apjungia aukšto (vartotojo) ir žemesnių lygių (sistemos) reikalavimus, apibrėžiama priklausomybė tarp vartotojo ir sistemos reikalavimų. Patenkinimo (angl. *satisfy*) ryšiu parodoma priklausomybė tarp reikalavimo ir jį realizuojančio sistemos komponento. Testavimo atvejai siejami su reikalavimų verifikavimu, taigi, verifikavimo ryšiu parodoma, kaip testavimo atvejis turi verifikuoti reikalavimą. Papildymo (angl. *refine*) ryšiu vaizduojama, kaip konkretūs vienas ar keli modelio elementai gali papildyti reikalavimą.

Atsekamumo (angl. *trace*) ryšiui *SysML* modeliavimo kalboje nenurodomas apibrėžimas. Dažniausiai, jis gali būti naudojamas pavaizduoti ryšiui tarp reikalavimo ir bet kurio kito sistemos komponento. Pagrindimo (angl. *rationale*) ir problemos (angl. *problem*) ryšiams taip pat nenurodomi jokie apribojimai ar reikalavimai jų panaudojimui modeliuose. Bendrinio atveju, visi šie trys ryšiai pritaikomi pavaizduoti, kad modelio elementai tam tikru būdu siejasi tarpusavyje, kas paprastai gali likti ir nepastebėta.

Sistemos reikalavimų specifikavimui ir modeliavimui *SysML* reikalavimų diagrama pateikiama kartu su *SysML* reikalavimų lentele. Reikalavimų lentelės pasiteisina norint išskaidyti reikalavimą, pati lentelės forma pagerina reikalavimų atsekamumą, kuris labai svarbus kuriamos sistemos kokybės faktorius. Lentelės formoje aprašomi reikalavimų ryšiai ir reikalavimų ypatybės. Reikalavimų ryšiai taip pat gali būti parodomi reikalavimų matrica. Visa tai leidžia reikalavimus identifikuoti ir suteikti jiems prioritetus.

***i** metamodelis**

*i** metamodelis yra sukurtas į tikslus orientuotiems reikalavimams specifikuoti, šis metodas akcentuoja problemą, o ne į jos sprendimo būdus. *i** karkasas sutelkia dėmesį į ankstyvąsias reikalavimų modeliavimo ir analizės fazes pabrėždamas reikalavimų iškėlimo priežastis, o ne aprašymus, kokias funkcijas sistema turi patenkinti. Šis metodas sukurtas modeliuoti ir argumentuoti organizacines aplinkas ir informacines sistemas. *i** metodas pasižymi grafine notacija ir matematiniais modeliais, skirtingai nei *SysML* pateikiama modeliavimo kalba. Vadinasi, tai, kas išreikšta grafiškai gali būti aprašyta ir formaliai – taikant matematinius principus.

Pagrindinė sąvoka, naudojama *i**, yra „sąmoningas“ (angl. *intentional*) aktorius (angl. *actor*), kuris organizacijoje suprantamas, kaip turintis „sąmoningas“ savybes: tikslus, įsitikinimus, galimybes ir įsipareigojimus. Tokie aktoriai priklauso vieni nuo kitų atlikdami užduotis ir naudodami resursus. Be to, aktoriai strategiškai gali pakeisti priklausomybes siekdami pasinaudoti galimybėmis ar įveikti grėsmes. Tokiu būdu, išskiriami du modeliai:

- Strateginis priklausomybių SD (angl. *strategic dependency*);
- Strateginis pagrindimo SR (angl. *strategic rationale*).

Strateginis priklausomybių modelis aprašo aktorių priklausomybes organizaciniame kontekste. Priklausomybė apibrėžiama kaip dviejų aktorių – nepriklausomo ir priklausančiojo – susitarimas siekiant įgyvendinti tikslą, atlikti užduotį ar sukurti produktą. Priklausomybė gali būti ir miglotai apibrėžtas tikslas be aiškių jo išpildymo kriterijų, apribojimų. Tuo tarpu, strateginis pagrindimo modelis aprašo interesantų interesus ir rūpesčius, būtent, kaip jie yra adresuojami įvairioms sistemos konfigūracijoms ir aplinkoms. Pagrindimo modelis suteikia daugiau detalių priklausomybių modeliui, tarsi, pažvelgiant į strategines sąsajas „iš vidaus“ [7].

Taigi, *i** karkasas reprezentuoja strateginį agentų reikalavimų inžinerijos požiūrį. Pagrindiniai modelio elementai yra tikslai (angl. *goals*), neapibrėžtiniai tiksliai (angl. *softgoals*), užduotys (angl. *tasks*) ir resursai (angl. *resources*). Jie susisteminami aktoriams, kurie dar gali būti specifikuojami kaip agentai (angl. *agents*), rolės (angl. *roles*) ir postai (angl. *positions*). *i** modelis leidžia įvertinti, kada tikslas yra pasiekiamas. Tikslams vertinti *i** karkasas pateikia tikslo būsenas – vertinimo etiketes – todėl aukštesnio lygio tikslų gyvybingumas ir išpildymas gali būti įvertinamas remiantis žemesniojo lygio tikslais [12].

*i** karkasas įgalina skirtingų lygių ir tipų analizes, tokias kaip, užduočių be tikslų gyvybingumas, įgyvendinimas ir patikimumas.

KAOS metamodelis

KAOS (angl. *Knowledge Acquisition in automated specification* arba *Keep All Objectives Satisfied*) metodas palaiko visą reikalavimų specifikuojimo procesą nuo vartotojo poreikių iki sistemos reikalavimų, duomenų objektų ir operacijų. Lyginant su *i** karkasu, *KAOS* telkiamas dėmesys į vėlesnę reikalavimų specifikuojimo ir analizės fazę. *KAOS*, taip pat kaip ir *i** metodas, pasižymi grafine notacija ir matematiniais modeliais. Jis taip pat yra orientuotas į tikslus ir akcentuoja problemas, o ne jų sprendimo būdus.

KAOS metodas nepalaiko modelio motyvacijos, kodėl yra keliami reikalavimai, mažiau koncentruojamasi į aktoriaus intencijas. Pagrindinis *KAOS* meta-modelio konceptas yra tikslas, kuris apibrėžiamas kaip išankstinis ketinimų pareiškimas, kuriuos sistema kooperuodamasi su kitais agentais turi patenkinti. Čia agentas gali būti bet kuris tikslo patenkinimo aktorius: informacinė sistema, sukurtas produktas, žmogiškasis išteklius. *KAOS* meta-modelis sudaro keturi pagrindiniai modeliai:

- Tikslų hierarchija (angl. *goal modeling*);
- Atsakomybių modelis (angl. *responsibility modeling*);
- Duomenų modelis (angl. *object modeling*);
- Operacijų modelis (angl. *operation modeling*).

KAOS metodas leidžia tikslus nusakyti trimis abstrakcijos lygiais, aukštesni lygio tikslai su žemesniojo siejami papildymo ryšiais, kurie apibrėžia, kokie žemesniojo lygio tikslai reikalingi, kad būtų patenkintas aukštesniojo lygio tikslas. Taip pat, papildymo ryšiais gali būti nustatoma, kokiu tikslu žemesniojo lygio tikslas yra išskeltas.

Svarbus reikalavimų inžinerijos sprendimas yra, kuris tikslas gali būti automatizuotas ir kuris negali. Reikalavimu vadinamas sprendimas, kuris priskirtas būsimai sistemai, vadinasi, tuomet jis turi būti tenkinamas. Be to, tikslas keliamas sistemos aplinkai vadinamas galimybe ir sistemai nėra priskiriamas. *KAOS* meta-modelyje konflikto (angl. *conflict*) ryšiu galima parodyti, kada vieno tikslo įgyvendinimas stabdo kito tikslo įgyvendinimą. Kliūtis gali būti naudojama situacijai nusakyti.

KAOS metodas leidžia modeliuoti dalykinės srities savybes: hipotezes, apibūdinančias ypatybes, kurias tikimasi įgyti ir konstantas, apibūdinančias nekintančias savybes. *KAOS* palaiko skirtingas analizes, tokias kaip, reikalavimų atsekamumas, pilnumas, formalus patvirtinimas, papildymo tikrinimas, rizikų ir grėsmių analizė [7].

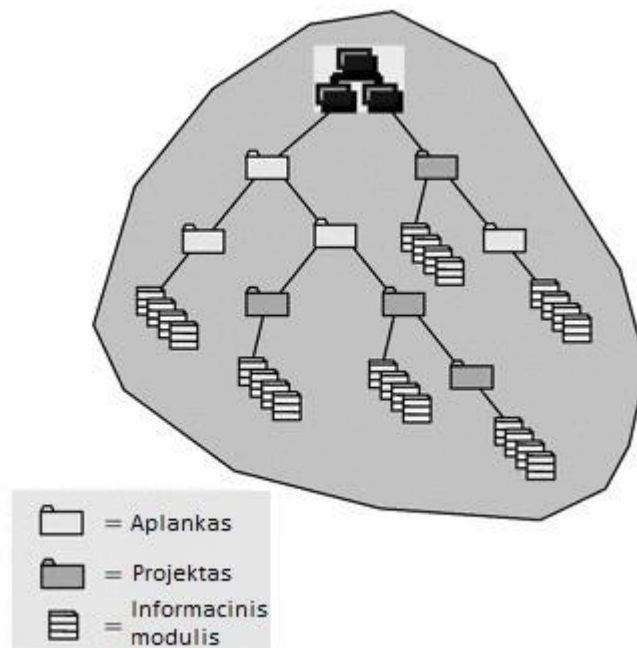
1.3.4. Reikalavimų valdymo įrankių analizė

Reikalavimų specifikavimui gali būti panaudojami paprasti teksto redagavimo įrankiai arba gali būti pereita prie išvystytų reikalavimų specifikavimo aplinkų, pavyzdžiui, *Doors*, *IBM Rational RequisitePro*.

Doors reikalavimų valdymo įrankis

Doors (angl. *Dynamic object oriented requirements system*) – daugiavfunkcinis reikalavimų informacijos valdymo ir reikalavimų atsekamumo įrankis. Šiuo įrankiu visą sistemos ar programinės įrangos kūrimo laikotarpį apdorojamos reikalavimų atsekamumo ataskaitos siekiant pagerinti reikalavimų kokybę ir sumažinti išlaidas.

Visi suinteresuotieji asmenys gali nuotoliniu būdu prisijungti prie serverio ir atlikti informacijos pakeitimus duomenų bazėje, pagal savo turimas vartotojo teises. Kiekvienas vartotojas gali pasidaryti savo asmeninę projekto kopiją kitoje *Doors* duomenų bazėje. *Doors* duomenų bazės projektai gali talpinti bet kokią informaciją, susijusią su reikalavimais.



3 pav. Doors duomenų bazės struktūra [13]

Doors duomenų bazėje (3 pav.), saugomoje serveryje, gali būti laikomi projektai, talpinantys reikalavimų modelius, reikalavimus ir jų ryšius. Reikalavimų informacija duomenų bazėje saugoma moduliais, kuriuos galima rūšiuoti pagal aplankus. Aplankuose gali būti patalpinti kiti aplankai, projektai ar moduliai. Kiekvienam aplankui suteikiamas pavadinimas ir aprašymas [13].

Doors įrankiu kiekvienas reikalavimas apdorojamas kaip atskiras objektas, galintis turėti neribotą atributų kiekį. Objektams gali būti priskiriamas dviejų tipų semantinis ryšys: išorinis arba vidinis. Vidiniu ryšiu *Doors* vienoje duomenų bazėje sujungiami du objektai, išoriniu ryšiu apjungiami skirtingų *Doors* duomenų bazių išoriniai objektai.

***IBM Rational RequisitePro* reikalavimų valdymo įrankis**

IBM Rational RequisitePro – tai reikalavimų valdymo įrankis, kuris skirtas modeliuoti ir sekti reikalavimus, kontroliuoti jų versijas. Įrankis labiausiai pritaikytas naudotojams, kurie siekia sekti ir kontroliuoti nuolat kintančius reikalavimus.

IBM Rational RequisitePro reikalavimų valdymo įrankis suteikia paprastą tekstinių ir grafinių elementų patalpinimą, leidžia tvarkyti reikalavimus naudojant panaudojimo atvejų diagramas. Reikalavimų informacija saugoma reliacinėje duomenų bazėje, patys reikalavimai gali būti grupuojami aplankais, kuriems suteikiamas pavadinimas ir aprašymas.

Reikalavimams suteikiami atributai, kuriais remiantis vėliau panaudojant užklausas galima greitai rasti reikiamą reikalavimų informaciją. Po kiekvieno reikalavimui atlikto pakeitimo, versijos istorijoje išsaugomas pakeitimas ir korekcijos data. Taip pat, kiekvienam reikalavimui gali būti nustatyti hierarchijos ir atsekamumo ryšiai.

IBM Rational RequisitePro reikalavimų valdymo įrankis pateikia statistinių metrikų sudarymo funkcijas. Papildomai, įrankis leidžia sudaryti reikalavimų atsekamumo matricas ir ryšių atvaizdavimo medžius. Gautus rezultatus atitinkamai pagal jų tipą galima saugoti įrankyje arba eksportuoti MS Word ir MS Excel formatais. Remiantis nustatytais ryšiais tarp reikalavimų, galima atliktų reikalavimų įtakos analizę, jeigu reikia nustatyti, kaip vienas naujas reikalavimas įtakoja kitus jau esančius reikalavimus.

IBM Rational RequisitePro įrankis gali būti integruojamas su *MS Word*, tuomet specifikacijoje esantys reikalavimai gali būti peržiūrėti, modifikuojami, atnaujinama jų informacija tiek *MS Word* įrankiu, tiek *IBM Rational RequisitePro* įrankiu. Taip pat yra internetinė prieiga – suteikianti galimybę peržiūrėti, redaguoti reikalavimų specifikaciją neturint įdiegto įrankio savo kompiuteryje.

IBM Rational RequisitePro reikalavimų valdymo įrankis suderinamas su duomenų bazių valdymo įrankiais: *MS Access*, *MS SQL*, *Oracle* [5].

1.3.5. Reikalavimų modeliavimo metodų palyginimas

Trumpai apžvelgiami reikalavimų specifikacijų sudarymo šablonai, reikalavimų specifikacijų valdymo įrankiai ir reikalavimų modelių sudarymo metodai. Analizuojamos jų pateikiamos galimybės arba galimi trūkumai ir panaudojimas, pritaikymo visapusiškumas arba pritaikymo specifinė sritis.

Reikalavimų specifikavimui, kalbant apie specifikavimo šablonus, atliekamas *SRS* ir *Volere* šablonų palyginimas. *IEEE* pateikiamas *SRS* šablonas apima pačios programinės įrangos ar informacinės sistemos funkcijų, vartotojo sąsajos, išvaizdos, duomenų bazės ir kitų sistemos atributų specifikavimą. Tuo tarpu, *Volere* šablonu, numatomos šios specifikacijos ir, papildomai, pateikiama galimybė apibrėžti ir nurodyti visus programinės įrangos ar kuriamos informacinės sistemos informacinius duomenis, kurie gali būti naudingi kūrimo, diegimo metu. *Volere* šablone galima išsamiai specifikuoti veiklos sudėtį prieš ją kompiuterizuojant, nurodyti panaudojimo atvejus, pagal kuriuos sudaromi produkto funkciniai reikalavimai ir vėliau nefunkciniai, taip pat, specifikuoti, kaip bus įgyvendinamas konkretus projektas, jo trukmę, kaštus, kaip bus migruojamas. Taigi, verčiau naudotis *Volere* šablonu, kadangi neaktualius skyrelius esant reikalui galima panaikinti, o vadovaujantis *SRS* šablonu svarbi informacija gali likti nedokumentuota.

Analizuojant reikalavimų modeliavimo kalbų ir metodų pateikiamas galimybes, lyginamos *SysML* modeliavimo kalbos galimybės bei *i** ir *KAOS* meta-modeliai. *i** ir *KAOS* meta-modeliai pasižymi tuo, kad abu turi grafinę notaciją ir yra pagrįsti matematiniu modeliu – grafiškai išreikštas modelis gali būti formaliai pateikiamas remiantis matematine logika ir principais. Palyginimui, *SysML* modeliavimo kalba grįstas modelis negali būti tiesiogiai pateikiamas matematiniais reiškiniiais. Taigi, čia reikėtų pasirinkti, kuria kryptimi eiti, atsižvelgiant į konkrečius poreikius, ar matematinis pagrindimas bus vėliau naudojamas.

KAOS ir *i** meta-modelius lyginant tarpusavyje, atkreipiamas dėmesys į tai, kad *KAOS* pateikia keturias diagramas, kuriomis sudaromi interesantų tikslų, atsakomybių ir sistemos naudojamų duomenų bei atliekamų operacijų modeliai. *i** meta-modelis koncentruotas į reikalavimų iškėlimo priežastis, strateginius interesantų poreikius, bet ne į sistemos funkcijas, kurias turėtų tenkinti. Taigi, *KAOS* meta-modeliu reikalavimai plačiau apimami, apibrėžiama platesnė bei šiuo atveju aktualesnė sritis.

Galiausiai, analizuojant reikalavimų valdymo įrankius *Doors* ir *IBM Rational RequisitePro*, vienas iš esminių kriterijų, lemiantis įrankio pasirinkimą yra tai, kad *IBM Rational RequisitePro* palaiko *Volere* šablono reikalavimų specifikacijos formatą.

1.4. Veiklos modeliavimo analizė

1.4.1. Veiklos samprata ir struktūra

Veiklos modeliavimas atlieka pagrindinę rolę siekiant suprasti veiklos procesus. Daugeliu atvejų, veiklos procesas yra toks pat išraiškingas kaip ir technika, kuri buvo panaudota tai veiklai modeliuoti. Taigi, veiklos modelio elementai ir galimybės atlieka svarbią rolę kalbant apie veiklos procesus.

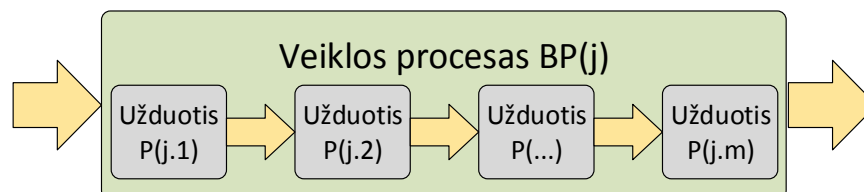
OMG (angl. *Object Management Group*) veiklos procesą apibūdina, kaip tiksliai nustatytą veiklos veiklų rinkinį, kuris apibrėžia žingsnius reikalingus pasiekti verslo tikslą. Į tai įeina informacijos ir resursų srautų naudojimas.

WFMC (angl. *The glossary of the WorkFlow Management Coalition*) aprašo veiklos procesą kaip rinkinį vienos ar daugiau procedūrų, arba kaip susijusias procedūras ar veiklas, kurios bendrai įgyvendina verslo tikslus ir politikas, paprastai esančias organizacijos struktūrą nusakančiose rolėse ir santykiuose.

Geary A. Rummler ir *Alan P. Brache* verslo procesą apibūdina kaip seriją žingsnių, skirtų sukurti produktą ar paslaugą. Jei rezultatai yra tiesiogiai naudingi klientui, procesas yra pirminis, kitu atveju, tai yra pagalbinis procesas.

Martyn Ould apibūdina verslo procesą, kaip nuoseklų verslo veiklų rinkinį, kurios vykdomos bendradarbiaujančioje grupėje siekiant bendro tikslo.

Autoriai *Howard Smith* and *Peter Fingar* verslo procesus aprašo panašiai, bet papildė juos keliomis charakteristikomis. Pagal juos, verslo procesas yra kompleksiškas, sudėtingas, paskirstytas ir ilgai trunkantis reiškinys [4].



4 pav. veiklos proceso struktūra [4]

Apibendrinant veiklos proceso apibrėžimus, veiklos procesas (4 pav.) sudarytas iš rinkinio užduočių, kurios yra tarpusavyje susijusios per įeigos ir išeigos srautus ir sudaro tam tikrą darbų seką. Veiklos procesas sukuria tam tikrą išėigą – produktą ar paslaugą, padeda įgyvendinti vieną ar daugiau organizacijos tikslų.

1.4.2. Veiklos modeliavimo kalbų analizė

UML modeliavimo kalba

UML (angl. *the Unified Modelling Language*) – unifikuota modeliavimo kalba. *UML* modeliavimo kalba specifikuoja programinės įrangos sistemas, detalizuoja sistemos artefaktus, aprašo ir struktūrizuoja sistemos loginę struktūrą.

UML pateikia tris pagrindines modeliavimo notacijas [16]:

- Elgsenos diagramos (angl. *behaviour diagrams*) – aukščiausio lygio programinės įrangos funkcionalumui aprašyti;
- Sąveikos diagramos (angl. *interaction diagrams*) – žemesnio lygio programinės įrangos funkcionalumui aprašyti – objektinėms sąveikoms;
- Struktūros diagramos (angl. *structure diagrams*) – pagrindinei programinės įrangos struktūrai specifikuoti įvairiais lygmenimis.

Veiklai modeliuoti pateikiamos – panaudojimo atvejų diagrama (angl. *use case diagram*) ir veiklos diagrama (angl. *activity diagram*). Veiklos diagramų elementai skirstomi į grupes [16]:

- Veiksmai (angl. *actions*);
- Objektai ir objektų srautai (angl. *objects and object flow*);
- Kontrolės mazgai (angl. *control nodes*).

BPMN metodika

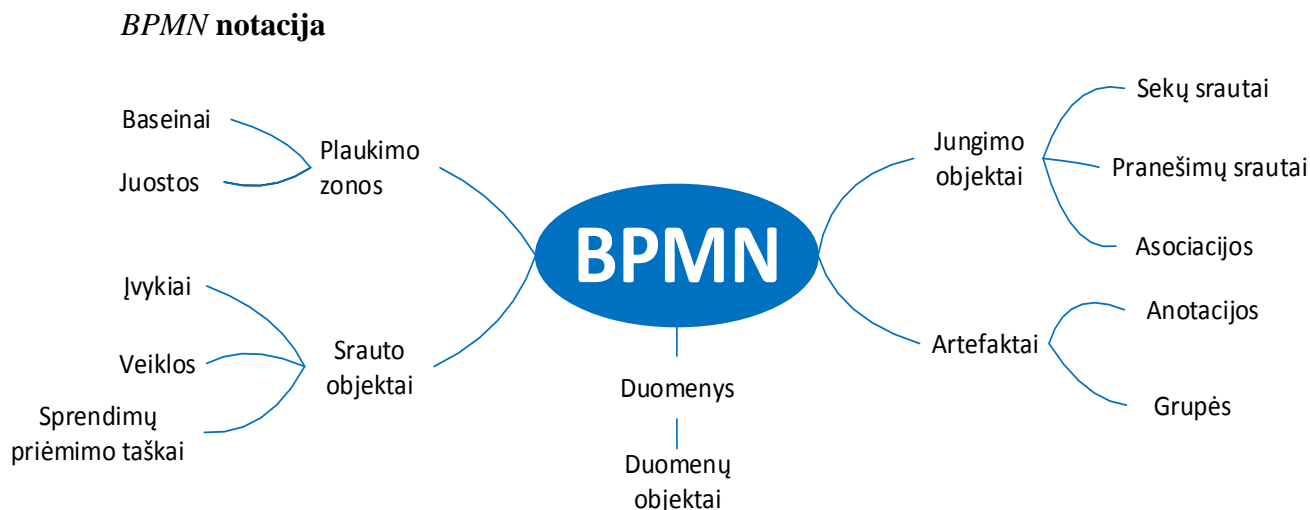
BPMN (angl. *Business Process Model and Notation*) – *OMG* veiklos procesų modeliavimo standartas. *BPMN* metodika palaiko veiklos procesų modeliavimą visą veiklos gyvavimo laikotarpį, pradedant sukūrimu iki vykdymo ir realizavimo. Naudojantis *BPMN* siekiama sukurti veiklos procesų modelius, kurie būtų vienodai suprantami visoms įmonės veiklos procesais besidominčioms žmonių grupėms, o ypač veiklos srities žmonėms.

BPMN modeliavimo kalbos pagrindinis privalumas, kad jos notacija lengvai suprantama įvairioms suinteresuotoms šalims, kurių technologinės žinios gali būti žemos. Tokia notacija sudaro galimybes bendradarbiauti ne tik skirtingiems interesantams, bet ir tarp skirtingų organizacijų. Nesunkiai aprašomi ir suprantami veiklos procesai, ciklai, sprendimų priėmimas, veiklos taisyklės.

BPMN diagramą galima panaudoti pagal esamą poreikį kaip:

- Vidinių veiklos procesų diagramą;
- Sąveikaujančių procesų diagramą;
- Bendradarbiaujančių procesų diagramą.

Iš kitos pusės, *BPMN* metodika gali būti suprantama, kaip procedūrinė žinių reprezentacija. Kiekviena *BPMN* diagrama perteikia persipynusias procedūras – veiklos procesus. Veiklų sekos diagramoms trūksta formalios modelio semantikos, todėl sudėtingesnė analizė gali tapti problemine. Be to, sąlyginai paprasta išvaizda reikalauja nemažos reprezentacinės erdvės. Taigi, konstruojamąsias deklaratyvios formos žinias, tokias kaip veiklos taisyklės, atvaizduoti nėra labai paprasta.



5 pav. Pagrindinės *BPMN* elementų grupės [3]

BPMN notacijoje išskiriamos šios pagrindinės elementų grupės (5 pav.):

- Srauto objektai (angl. *flow objects*);
- Jungimo objektai (angl. *connecting objects*);
- Plaukimo zonos (angl. *swimlanes*);
- Artefaktai (angl. *artifacts*);
- Duomenys (angl. *data objects*).

Pagrindiniai objektai, kurie apibrėžia veiklos procesus yra srauto objektai. Tai grafiniai elementai, skirti veiklos procesų elgsenai apibrėžti, kurie išskiriami į trijų tipų objektus:

- Įvykius (angl. *events*);
- Veiklas (angl. *activities*);
- Sprendimo priėmimo taškus (angl. *gateways*).

Jungimo objektai, skirti srauto objektus apjungti tarpusavyje ar prijungti papildomą informaciją. Jungimo objektai gali būti suskirstyti į:

- Sekos srautus (angl. *sequence flow*) – tai paprastieji perdavimai;
- Pranešimų srautus (angl. *message flow*) – pranešimų įvykių perdavimai;
- Asociacijas (angl. *association*) – papildomų duomenų, komentarų, informacijos asociacija su srauto objektais.

Artefaktai (angl. *artifacts*) naudojami papildomai procesų informacijai pateikti, nurodomi kelių tipai, tačiau esant reikalui galima pasipildyti:

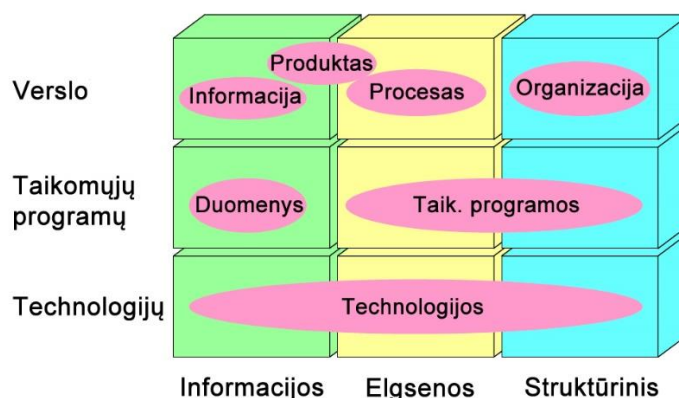
- Grupės (angl. *groups*);
- Anotacijos (angl. *annotations*).

Duomenys (angl. *data objects*) atskirai nuo visų artefaktų nurodomi, kokie yra naudojami įeinantys ir išeinantys duomenų objektai [3].

ArchiMate modeliavimo kalba

ArchiMate – tai į informacines technologijas orientuota modeliavimo kalba. Naudojantis *ArchiMate* kalba, skirtingų sričių specialistai gali pažvelgti į organizacijos architektūrą skirtingais aspektais. Skirtingi aspektai gali būti nagrinėjami atskirai arba susiejant du arba daugiau aspektų. Aspektai – puiki priemonė sutelkti dėmesį į konkrečius architektūros požiūrius. Jie nustatomi pagal suinteresuotų asmenų, su kuriais bendraujant yra kuriama architektūra, poreikius. Būtent suinteresuotieji asmenys sprendžia ką bus reikalinga rodyti galutiniame rezultate.

ArchiMate modeliavimo karkasas dviem ortogonaliomis dimensijomis išskaido organizacijos veiklą (6 pav.): sluoksniais, kurie nuosekliai nurodo modeliuojamą abstrakcijos lygį ir aspektais, kurie išreiškia skirtingus interesus.



6 pav. ArchiMate modeliavimo karkasas [8]

Sluoksnių dimensija išskiria tris pagrindinius sluoksnius [8]:

- Verslo (angl. *business layer*);
- Taikomųjų programų (angl. *application layer*);
- Technologijų (angl. *technology*).

Aspekto dimensija išskiria tokius pagrindinius modeliavimo aspektus [8]:

- Struktūrinis (angl. *structure*) aspektas, kuris nusako aktorius ir jų ryšius;
- Elgsenos (angl. *behaviour*) aspektas, kuris nusako aktorių elgseną, jų atliekamus veiksmus;
- Informacijos (angl. *information*) aspektas, kuris nusako aktorių naudojamas dalykinės srities žinias.

Kiekviena sankirta tarp sluoksnių ir aspektų vadinama konkrečiu aspektu, be to, priklausomai nuo požiūrio gali sutapti ir persidengti. Atitinkamai, persidengiantys aspektai sudaro persidengiančias sąvokas.

Tokiu būdu, *ArchiMate* modeliavimo kalba pateikia 18 modeliavimo aspektų, kurie yra [8]:

- Įvadinis aspektas (angl. *Introductory viewpoint*);
- Organizacijos aspektas (angl. *Organisation viewpoint*);
- Aktorių bendradarbiavimo aspektas (angl. *Actor co-operation viewpoint*);
- Veiklos funkcijų aspektas (angl. *Business function viewpoint*);
- Verslo procesų aspektas (angl. *Business process viewpoint*);
- Veiklos procesų bendradarbiavimo aspektas (angl. *Business process co-operation viewpoint*);
- Produkto aspektas (angl. *Product viewpoint*);
- Taikomųjų programų elgsenos aspektas (angl. *Application behavior viewpoint*);
- Taikomųjų programų bendradarbiavimo aspektas (angl. *Application co-operation viewpoint*);
- Taikomųjų programų struktūros aspektas (angl. *Application structure viewpoint*);
- Taikomųjų programų naudojimo aspektas (angl. *Application usage viewpoint*);
- Infrastruktūros aspektas (angl. *Infrastructure viewpoint*);
- Infrastruktūros naudojimo aspektas (angl. *Infrastructure usage viewpoint*);
- Įgyvendinimo ir diegimo aspektas (angl. *Implementation and deployment viewpoint*);
- Informacijos struktūros aspektas (angl. *Information structure viewpoint*);
- Paslaugų realizacijos aspektas (angl. *Service realization viewpoint*);
- Sluoksninis aspektas (angl. *Layered viewpoint*);
- Kraštovaizdžio žemėlapis aspektas (angl. *Landscape map viewpoint*).

ArchiMate sutelkia dėmesį į organizacijos išorines ir vidines savybes informacinių, elgesio bei struktūrines architektūros elementų kontekste. Išorinės savybės apibūdina produktus ir paslaugas, kurios yra siūlomos, o vidinės savybės aprašo, kaip jos yra siūlomos, ir kaip jos yra siūlomos priklausomai nuo procesų ir taikomųjų programų [7].

1.4.3. Veiklos modeliavimo metodų palyginimas

Analizuojamos *UML*, *BPMN* ir *ArchiMate* veiklos procesų modeliavimo kalbos, jų rekomenduojamos metodikos. Apžvelgiami jų panašumo požymiai arba skirtumai, kurie galėtų lemti vienos ar kitos pasirinkimą veiklos procesų modeliams kurti.

Apžvelgiant *UML* ir *BPMN* veiklos modeliavimo kalbas, jos pristato po vieną veiklos modelio diagramą. Nepaisant to, *BPMN* notacija ir diagramos daugiau formaliai apibrėžtos, taip pat, pateikiami naudojimo ir modeliavimo aprašymai. *BPMN* modelio notacija pritaikyta atvaizduoti daugybę skirtingų rūšių objektų, jų grupavimui, ryšių atvaizdavimui. *UML* modeliavimo kalbos tikslas unifikuoti sistemos modelį. *BPMN* modeliavimo kalbos tikslas, būtent, yra pateikti lengvai suprantamą daugiafunkcinę ir visuotinai pritaikomą veiklos procesų modelių notaciją, kuri palengvintų veiklos procesų supratimą.

Kalbant apie *ArchiMate*, ši modeliavimo kalba suteikia gausų aštuoniolikos diagramų pasirinkimą organizacijos architektūrai modeliuoti skirtingais aspektais ir skirtingais detalumo lygiais pagal suinteresuotų šalių poreikį. Tačiau, *ArchiMate* modeliavimo kalba nėra aiškiai ir formaliai apibrėžta, sudaroma laisvė modelių interpretacijoms. Jos notacija išraiškinga, skirtingiems objektams pateikiamos skirtingos iškart atpažįstamos spalvos ir formos, tad sukurti modeliai įvairiems vartotojams išlieka nesunkiai suvokiami.

Taigi, galima apsispręsti, kad, būtent, veiklos modeliavimui *UML* ir *BPMN* modeliavimo kalbos yra universalesnės. Ypač, *BPMN* universalus pritaikymas dėl jos apibrėžtumo ir tikslingumo veiklos modeliavimui. Ją naudojant galima aiškiai apibrėžti bet kurį aktualų veiklos procesą, o detalumo lygį turi nusistatyti pats naudotojas.

1.5. Veiklos apribojimų sudarymo analizė

Literatūroje kalbama apie skirtingas veiklos apribojimų sudarymo idėjas, tačiau nepateikiamas tam tikras aiškiai apibrėžtas metodas, kuris išspręstų problemą. Integralumo sprendimo metodai galėtų būti sudaromi remiantis panaudojimo atvejais, testavimo atvejais arba veiklos žodynu ir taisyklėmis.

Literatūroje kalbama apie kelis galimus reikalavimų ir veiklos modelių integravimo metodus:

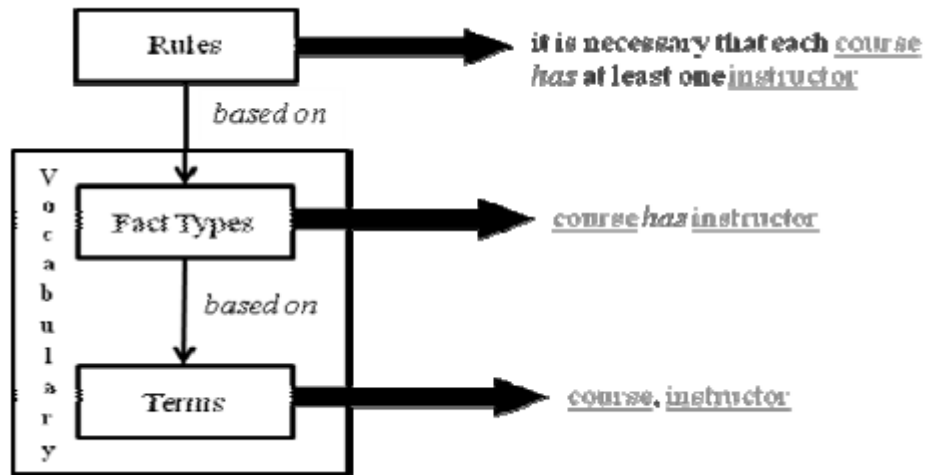
- Remiantis panaudos atvejais;
- Remiantis testavimo atvejais;
- Remiantis veiklos žodynais.

Kuomet remiamasi panaudojimo atvejų sudarymu, pirmiausiai atliekamas reikalavimų specifikavimas. Pagal tam tikrus principus iš reikalavimų specifikacijos išgaunamas panaudojimų atvejų modelis. Galiausiai, panaudojimo atvejų tekstas jau nusako būsimus veiklos procesus. Beveik analogiška situacija, kuomet remiamasi testavimo atvejais: - pagal reikalavimų informaciją gali būti surašomas testavimo atvejų dokumentas ir juo remiantis numatomi veiklos procesai.

Kuomet kalba apie veiklos žodyno *SBVR* (angl. *Semantics of Business Vocabulary and Business Rules*) panaudojimą, galima būtų pagal parašytą reikalavimų specifikaciją veiklos žodyne rinkti veiklos terminus – apibrėžtis, veiklos faktus ir veiklos taisykles. Tokiu būdu, iškart aiškiai apibrėžiama esminiai raktiniai reikalavimų žodžiai ir, taip pat, patogiai yra išskiriama reikalinga veiklos apribojimų informacija. Toliau, šiais sąrašais sukaupiama informacija, kuri bus naudojama veiklai modeliuoti. Taigi, veiklos žodyną pritaikymas yra vienas iš geriausių problemos sprendimo metodų.

1.5.1. SBVR standartas

SBVR (angl. *Semantics of Business Vocabulary and Business Rules*) – OMG veiklos žodyno ir veiklos taisyklių standartas. Šio standarto pagrindinė idėja yra, kad veiklos taisyklės paremtos faktais, o faktai paremti konceptais, kurie išreiškiami terminais (7 pav.) [15].



7 pav. SBVR standarto pagrindinė idėja

SBVR standartas veiklos žodynui ir veiklos taisyklės nusakyti turi savo terminologiją ir raktinius žodžius [14]. SBVR meta-modelio pagrindiniai elementai yra: terminai (angl. *terms*), vardai (angl. *names*), veiksmažodžiai (angl. *verbs*), fakto tipai (angl. *fact types*) ir raktiniai žodžiai (angl. *keywords*).

SBVR veiklos žodynas

SBVR veiklos žodynas apima terminus (angl. *term*), vardus (angl. *name*) ir faktus (angl. *fact*).

Terminas (angl. *term*) – tai daiktavardinis konceptas, kuris išreiškiamas vienu ar keliais žodžiais veiklos esybei nusakyti.

Vardas (angl. *name*) – vienu ar keliais žodžiais išreikštas individualus konceptas, kuris naudojamas konkrečiam terminui įvardinti.

Veiksmažodis (angl. *verb*) – veiksmažodinis konceptas, kuris išskiriamas sudarant faktų tipus, kaip fakto tipo struktūros dalis.

Fakto tipai (angl. *fact type*) yra tam tikri SBVR žodyno sakiniai, skirti nustatyti santykį tarp terminų ir vardų. Kiekvienas faktas gali būti įvardijamas sakinio *Term/Name-Verb-Term/Name* šablonine forma.

SBVR veiklos taisyklės

Veiklos taisyklės apibrėžia organizacijos veiklą – jos struktūrą ir elgseną. Taigi, veiklos taisyklės išskiriamos į du potipius:

- Struktūrinės (angl. *structural*) veiklos taisyklės – apibrėžiančios organizacijos struktūrą.
- Operatyvinės (angl. *operative*) veiklos taisyklės – atitinkamai, apibrėžiančios organizacijos elgseną.

SBVR veiklos taisyklėms sudaryti iš veiklos žodyno paimami terminai, vardai ir veiksmazodžiai, kurie buvo panaudoti veiklos žodynui sudaryti. Tuomet, pridėdant atitinkamus raktinius žodžius (angl. *keywords*) suformuojamos veiklos taisyklės. Raktažodžiai, kaip lingvistiniai elementai, naudojami frazėms konstruoti.

Raktinių žodžių klasifikacija [15]:

- Kiekybiniai raktažodžiai (angl. *quantification keywords*) – žodžiai, kuriais veiklos taisyklėse nusakoma terminų kiekybė (pvz., “*each*”, “*some*”, “*at least one*”, “*at least n*”, “*at most one*”, “*at most n*”, “*exactly one*”, “*exactly n*”, “*at least n at most m*”, “*more than one*”;

Čia *n* ir *m* – skaičiai).

- Loginiai raktažodžiai (angl. *logical keywords*) – žodžiai, kuriais veiklos taisyklėse išreiškiamos loginės operacijos.

(pvz., “*it is not the case that p*”, “*p and q*”, “*p or q*”, “*p or q but not both*”, “*if p then q*”, “*q if p*”, “*p if and only if q*”, “*not both p and q*”, “*neither p nor q*”, “*p whether or not q*”;

Čia *p* ir *q* – išraiškos arba teiginiai).

- Modaliniai raktažodžiai (angl. *modal keywords*) – žodžiai, kuriais veiklos taisyklėse aprašomos modalinės operacijos. Jie skirstomi į priešdėlinius (angl. *prefix*) ir įterptinius (angl. *embedded*). Be to, šie raktiniai žodžiai struktūrinėse ir operatyvinėse veiklos taisyklėse yra skirtingi.

(Priešdėlinių raktinių žodžių pvz.:

- Struktūriniai: “*it is necessary that*”, “*it is possible that*”, “*it is impossible that*”;
- Operatyviniai: “*it is obligatory that*”, “*it is permitted that*”, “*it is prohibited that*”.

Įterptinių raktinių žodžių pvz.:

- Struktūriniai: “*always*”, “*can*”, “*never*”;
- Operatyviniai: “*must*”, “*may*”, “*must not*”.)

- Kiti raktažodžiai (angl. *other keywords*) – pagalbiniai žodeliai, naudojami sudarant veiklos taisykles (pvz., “*the*”, “*a*”, “*an*”, “*another*”, “*a given*”).

1.6. Analizės išvados

1. Atlikta veiklos apribojimų automatinio išgavimo iš reikalavimų modelio prototipo naudotojų, naudotojų tipų, tikslų ir problemų analizė, paaiškintas prototipo kūrimo aktualumas.
2. Atlikta reikalavimų modeliavimo, reikalavimų struktūros ir jų specifikuavimo šablonų analizė. Palyginti *SRS* ir *Volere* reikalavimų specifikuavimo šablonai. Išanalizuotos *SysML* modeliavimo kalbos galimybės bei *i** ir *KAOS* meta-modeliai. Atlikta modeliavimo įrankių *Doors* ir *IBM Rational RequisitePro* analizė. Galima daryti išvadą, kad reikalavimai labiausiai tikėtinai pateikiami *SysML* modeliais. Pagal poreikius, reikalavimams modeliuoti galima būtų pasirinkti *RequisitePro*, nes jis suderinamas su *Volere* šablonu.
3. Atlikta veiklos modeliavimo analizė, *ArchiMate*, *UML* ir *BPMN* modeliavimo kalbų analizė ir palyginimas. Veiklos modeliavimui patogiausia rinktis *UML* arba *BPMN* modeliavimo kalbas. Konkrečiu atveju, siekiant išgauti veiklos apribojimus ir panaudoti veiklos modeliavimui, geriausia yra rinktis *UML* modeliavimo kalbą.

2. VEIKLOS APRIBOJIMŲ AUTOMATIZUOTO IŠGAVIMO IŠ REIKALAVIMŲ MODELIO METODAS

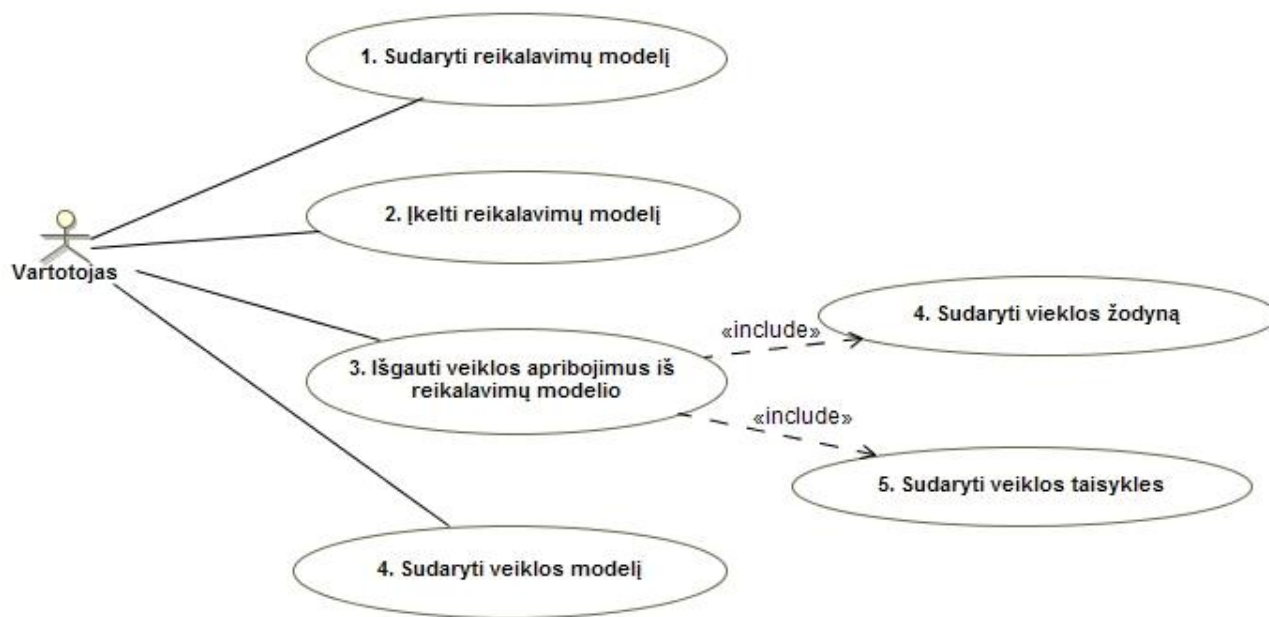
Veiklos apribojimų automatizuoto išgavimo iš reikalavimų modelio metodas skirtas tam, kad reikalavimų modelyje esanti informacija būtų efektyviai panaudojama veiklos modeliui modeliuoti. Metodas sumažina laiko sąnaudas, kurios reikalingos veiklos modeliui sudaryti.

Remiantis reikalavimų modeliu sudaromas veiklos žodynas ir veiklos taisyklės, kurie padeda vartotojui efektyviau formuoti veiklos modelį. Veiklos žodynas vienareikšmiškai apibrėžia sistemos terminiją pagal jos reikalavimus. Veiklos taisyklės dažniausiai parodo reikalingus sistemos parametrus ir apribojimus. Tokiu būdu vartotojas išvengia netikslumų iš naujo apibrėždamas apribojimus, greičiau sudaro veiklos modelį.

Suformuojamas metodo veikimo algoritmas, kuris aprašomas tolesniuose skyreliuose.

2.1. Panaudojimo atvejų modelis

Pateikiama veiklos apribojimų automatizuoto išgavimo iš reikalavimų modelio UML panaudojimų atvejų diagrama (8 pav.).



8 pav. Veiklos apribojimų automatizuoto išgavimo iš reikalavimų modelio UML panaudojimo atvejų modelis

Pagrindiniai šios sistemos panaudojimo atvejai:

- Sudaryti reikalavimų modelį – sistemos vartotojas prieš pradėdamas naudoti sistemą iš anksto turi pasidaryti *SysML* reikalavimų modelį.
- Įkelti reikalavimų modelį – sistemos vartotojas turi galėti įsikelti iš anksto parengtą *SysML* reikalavimų modelį, kurį sistema nuskaitytų ir patvirtintų, kad modelis korektiškas ir yra tinkamas tolesniam darbui.
- Išgauti veiklos apribojimus iš reikalavimų modelio – sistemos vartotojo tikslas yra efektyviai pasinaudoti *SysML* reikalavimų modelio informacija sudarant veiklos modelį.

Taigi, šis panaudojimo atvejis apima smulkesnius panaudojimo atvejus:

- 1) sudaryti veiklos žodyną;
 - 2) sudaryti veiklos taisykles.
- Sudaryti sistemos veiklos modelį – vartotojas pasinaudodamas sudarytais veiklos žodynu ir veiklos taisyklėmis turi galėti sudaryti sistemos veiklos modelį.

2.2. Duomenų struktūra

Veiklos apribojimų automatizuoto išgavimo iš reikalavimų modelio dalykinė sritis gali būti suskaidyta dalimis pagal panaudojimo atvejų eiliškumą bei prasmę. Taigi, dalykinės srities esybių modelis išskiriamas į tris pagrindines dalis:

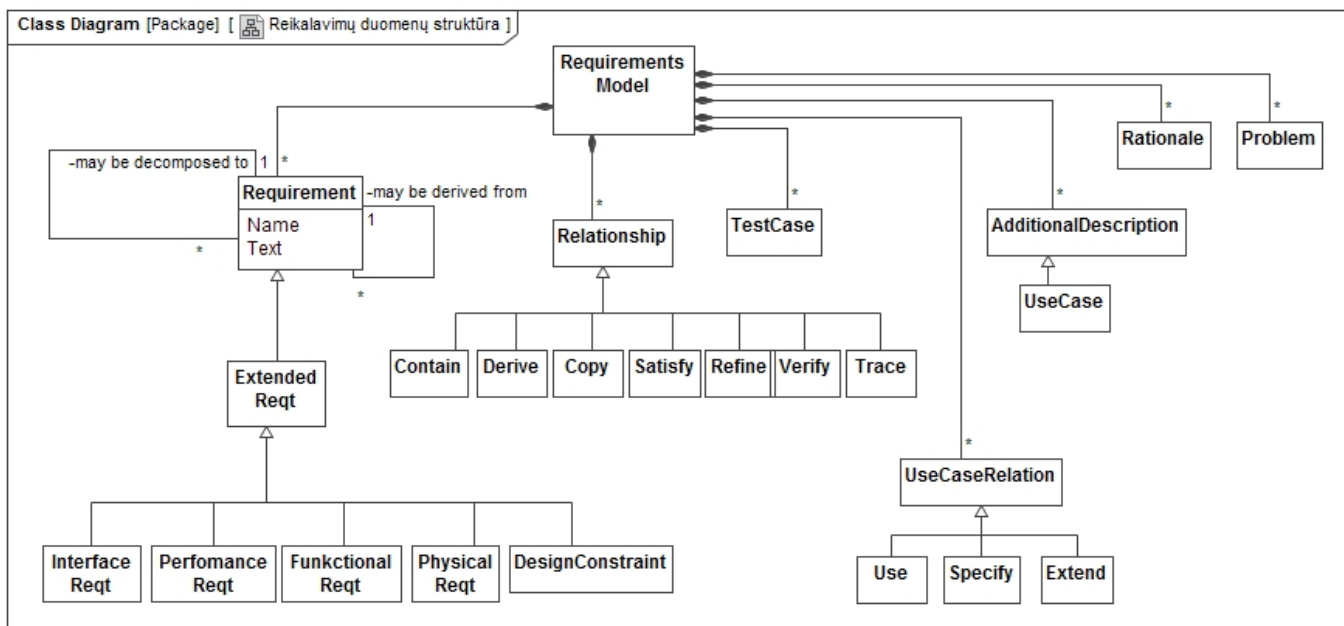
1. Reikalavimų dalis;
2. Veiklos žodyno ir taisyklių dalis;
3. Sistemos modelio dalis.

2.2.3. Reikalavimų duomenų struktūra

Pirmiausiai, nagrinėjamas reikalavimų modelis, kuris pateikiamas tam tikra *SysML* reikalavimų diagrama. Šios dalies dalykinės srities esybių modelis sudaromas remiantis *SysML* apibrėžiamu meta-modeliu.

SysML reikalavimų diagramoje reikalavimai gali būti specifikuoti (angl. *requirements*), testavimo atvejai (angl. *test cases*), reikalavimų pagrindimai (angl. *rationale*) ir problemos (angl. *problem*), papildomi aprašai – panaudojimo atvejai (angl. *use case*). Reikalavimai gali būti suskirstyti į tam tikrus potipius (pvz., angl. *Interface Reqt*, *Performance Reqt*, *Functional Reqt*, *Physical Reqt*, *Design Reqt*).

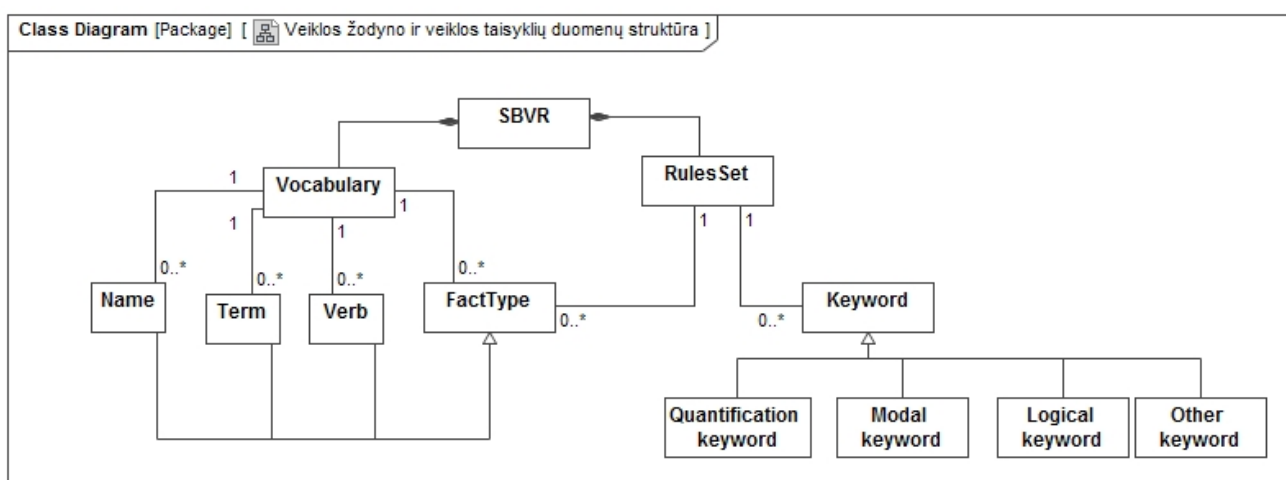
Tarp pačių reikalavimų diagramoje gali būti palaikoma hierarchija, nurodomas reikalavimas, iš kurio jis buvo išvestas ir nustatyti skirtingų tipų (pvz., angl. *contain*, *derive*, *copy*, *satisfy*, *refine*, *verify*, *trace*) kryptį turintys ryšiai (9 pav.).



9 pav. Reikalavimų duomenų struktūra

2.2.4. Veiklos žodyno ir veiklos taisyklių duomenų struktūra

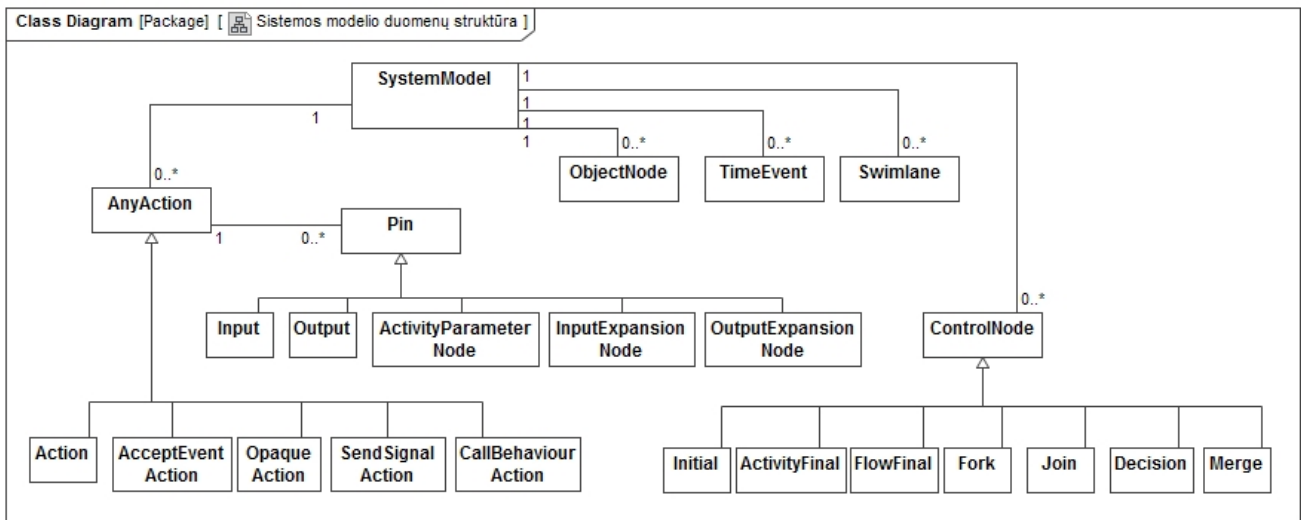
Toliau, iš reikalavimų modelio informacijos suformuojamas *SBVR* meta-modelio apimamas veiklos žodynas ir veiklos taisyklės (10 pav.). Taigi, šios dalies duomenų struktūra sudaroma remiantis *SBVR* apibrėžiamu meta-modeliu. *SBVR* meta-modelis nurodo tokius galimus sudaromo veiklos žodyno (angl. *vocabulary*) elementus: terminus (angl. *term*), faktus (angl. *fact*), faktų tipus (angl. *fact type*), vardus (angl. *name*), veiksmažodžius (angl. *verb*). Papildomai, sudaromas *SBVR* meta-modelio apibrėžtas veiklos taisyklių sąrašas (angl. *rules set*), kuris papildomas raktiniais žodžiais (angl. *keyword*), jie gali būti suskirstyti į potipius (pvz., angl. *quantification*, *modal*, *logical*, *other*).



10 pav. Veiklos žodyno ir veiklos taisyklių duomenų struktūra

Galiausiai, nagrinėjamas sudaromas sistemos veiklos procesų modelis, kuris pateikiamas kaip *SysML* diagrama (11 pav.). Taigi, šios dalies dalykinės srities esybių modelis sudaromas remiantis *SysML* apibrėžiamu meta-modeliu.

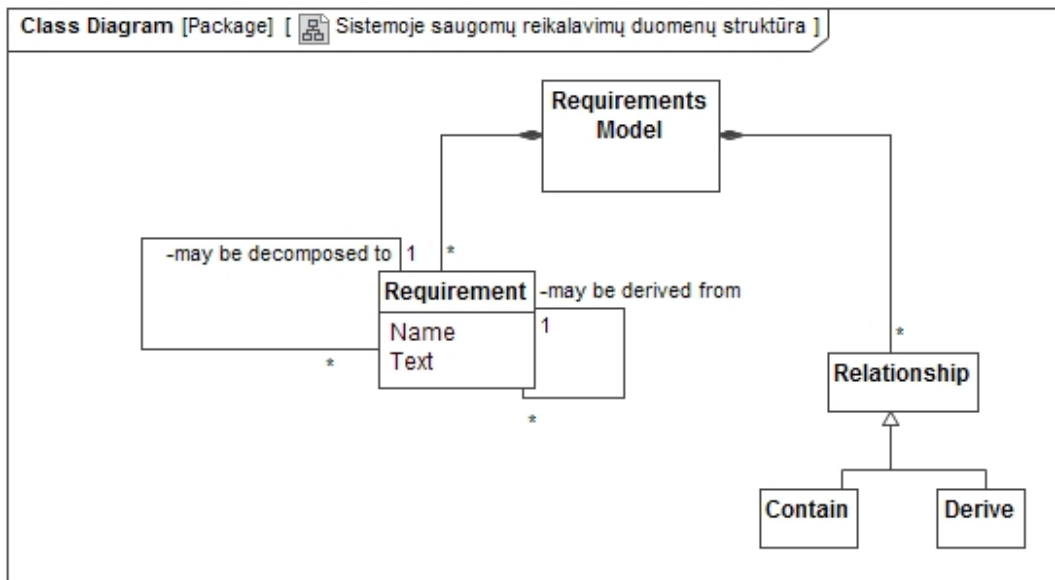
SysML meta-modelis diagramos elementus suskirsto į keturis pagrindinius tipus: veiksmus (angl. *any action*), veiksmų parametrus (angl. *pin*), objektus (angl. *object node*), kontrolės objektus (angl. *control node*) ir plaukimo juostas (angl. *swimlane*). Veiksmai, veiksmų parametrai ir kontrolės objektai gali būti suskirstomi į potipius.



11 pav. Sistemos modelio duomenų struktūra

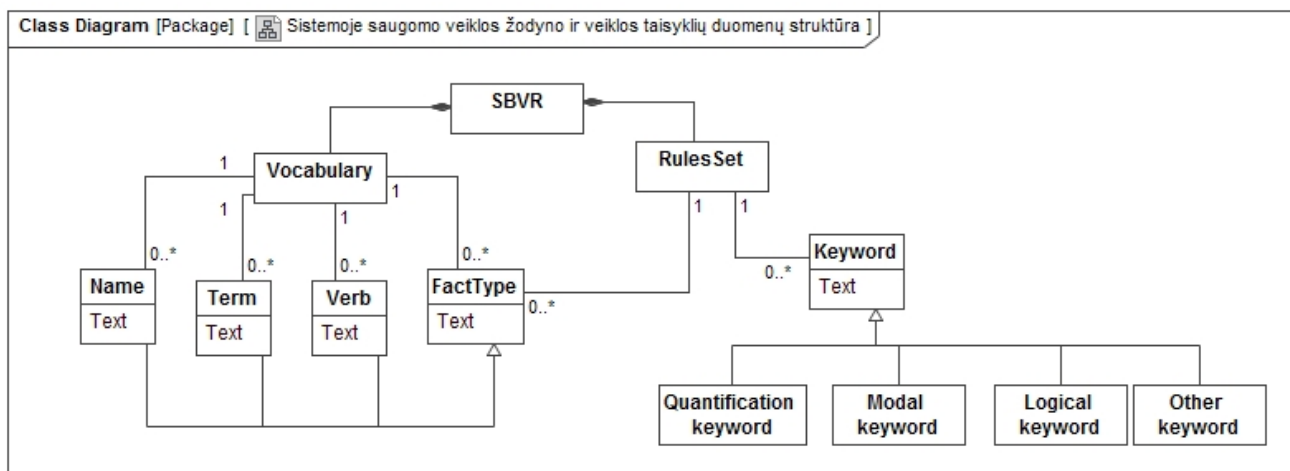
2.2.5. Sistemos duomenų struktūra

Iš vartotojo įkelto reikalavimų modelio sistemoje bus saugoma ne visa informacija, o tik ta, kuri vėliau bus panaudojama veiklos žodyno ir veiklos taisyklių formavimui. Saugomų duomenų struktūra pavaizduota žemiau esančioje diagramoje (12 pav.). Kuomet nuskaitomas reikalavimų modelis, sistemoje yra išsaugomi reikalavimų elementai ir juose kaupiama informacija – jų atributai – pavadinimas (angl. *name*) ir tekstas (angl. *text*), taip pat, ryšiai tarp reikalavimų – kompozicijos (angl. *contain*) ir išvedimo (angl. *derive*). Kiti meta-modelio ryšiai naudojami ne tarp dviejų reikalavimų, bet tarp reikalavimo ir papildomo modelio elemento, todėl – jie neaktualūs. Ryšiai tarp reikalavimų išsaugomi užfiksuojant, kokie reikalavimai yra išvesti iš tam tikro nuskaityto reikalavimo, bei į kokius reikalavimus jis gali būti suskaidomas.



12 pav. Sistemoje saugomų reikalavimų duomenų struktūra

Toliau, iš reikalavimų modelio informacijos yra suformuojamas *SBVR* meta-modelio apimamas veiklos žodynas ir veiklos taisyklės (13 pav.). Taigi, šios dalies duomenų struktūrą apibrėžia išsaugomi *SBVR* meta-modelio nurodytų veiklos žodyno (angl. *vocabulary*) ir veiklos taisyklių (angl. *rules set*), elementų atributai – jų tekstai (angl. *text*). Iš pradžių, suformuojami veiklos žodyno elementai, kuriais naudojantis sudaromos veiklos taisyklės.

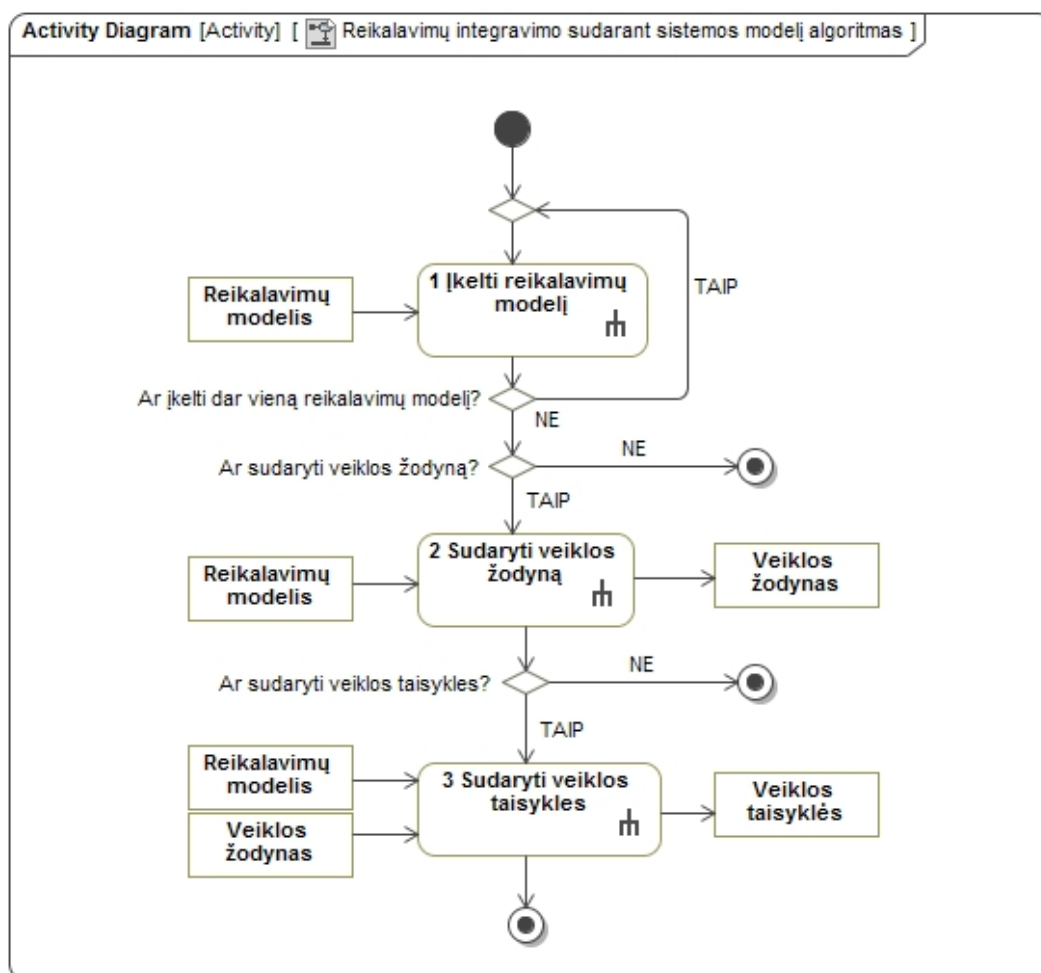


13 pav. Sistemoje saugomo veiklos žodyno ir veiklos taisyklių duomenų struktūra

2.3. Veiklos apribojimų automatizuoto išgavimo iš reikalavimų modelio metodo algoritmas

Veiklos apribojimų automatizuoto išgavimo iš reikalavimų modelio algoritmo kompiuterizuojamoji dalis susideda iš trijų pagrindinių žingsnių: reikalavimų modelio įkėlimo, veiklos žodyno sudarymo ir veiklos taisyklių sudarymo.

Paprastai, sistemos vartotojas gali turėti daugiau nei vieną *SysML* reikalavimų modelį, todėl į sistemą, taip pat, galima įkelti kelis atskirus reikalavimų modelius. Kuomet bent vienas reikalavimų modelis yra įkeltas, sistemos vartotojas gali inicijuoti veiklos žodyno sudarymą iš pasirinkto reikalavimų modelio arba baigti darbą su sistema. Sudarius veiklos žodyną, vartotojas gali inicijuoti veiklos taisyklių sudarymą iš pasirinkto reikalavimų modelio, tačiau joms sudaryti reikalingas suformuotas veiklos žodynas, arba gali baigti darbą su sistema nesudaręs veiklos taisyklių (14 pav.).



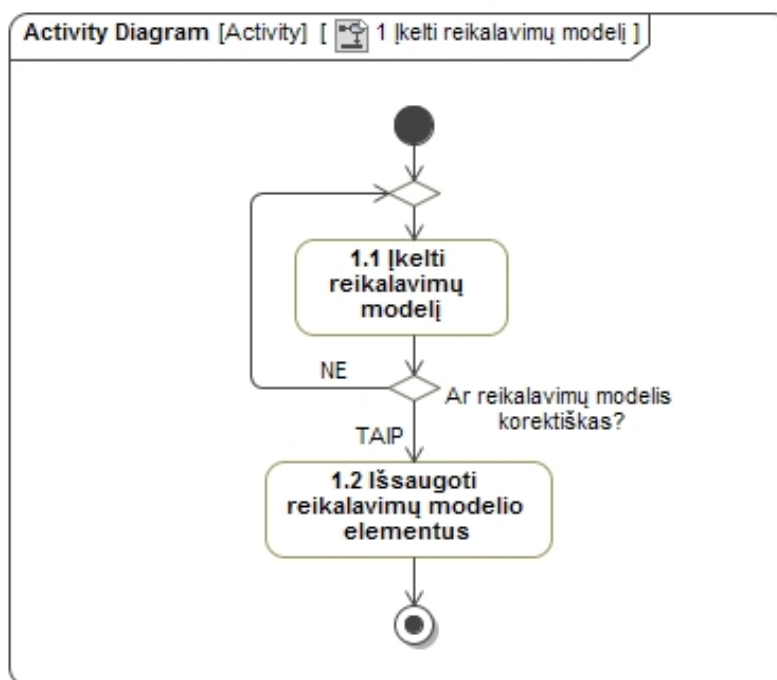
14 pav. Apibendrintas sistemos veikimo algoritmas

Pirmojo, antrojo ir trečiojo procesų įeigos srautas yra reikalavimų modelis, iš kurio antruoju atveju suformuojamas išeigos srautas – veiklos žodynas, o trečiuoju atveju – veiklos taisyklės. Trečiojo proceso įeigos srautas, taip pat, yra antrojo proceso sudarytas veiklos žodynas.

Jeigu yra įkeltas vienintelis reikalavimų modelis ir vartotojas tęsia darbą, tuomet pagal nutylėjimą toliau bus dirbama su tuo modeliu. Jeigu sistemoje yra įkelta daugiau nei vienas reikalavimų modelis, vartotojas gali pasirinkti, iš kurio modelio jis nori sudaryti veiklos žodyną ir veiklos taisykles. Jeigu vartotojui atsiranda poreikis atnaujinti veiklos žodyną ar veiklos taisykles įkeliant papildomą reikalavimų modelį, tuomet jis turėtų atskirai iš antrojo modelio sudaryti veiklos žodyną ir, jei reikia, veiklos taisykles. Po to, rankiniu būdu juos apjungti, o sistemoje automatiškai bus pašalinti besidubliuojantys elementai.

2.3.3. Įkelti reikalavimų modelį

Kuomet vartotojas įkelia į sistemą reikalavimų modelį (15 pav., 1.1 žingsnis), atliekamas automatinis sistemos patikrinimas, ar įkeltas reikalavimų modelis yra korektiškas. Jeigu modelis yra korektiškas, sistema nuskaityto jo elementus ir išsaugo elementų informaciją (15 pav., 1.2 žingsnis). Jeigu reikalavimų modelis nustatomas nekorektišku, vartotojas gali ištaisyti reikalavimų modelio trūkumus ir bandyti įkelti jį dar kartą arba įkelti kitą reikalavimų modelį.

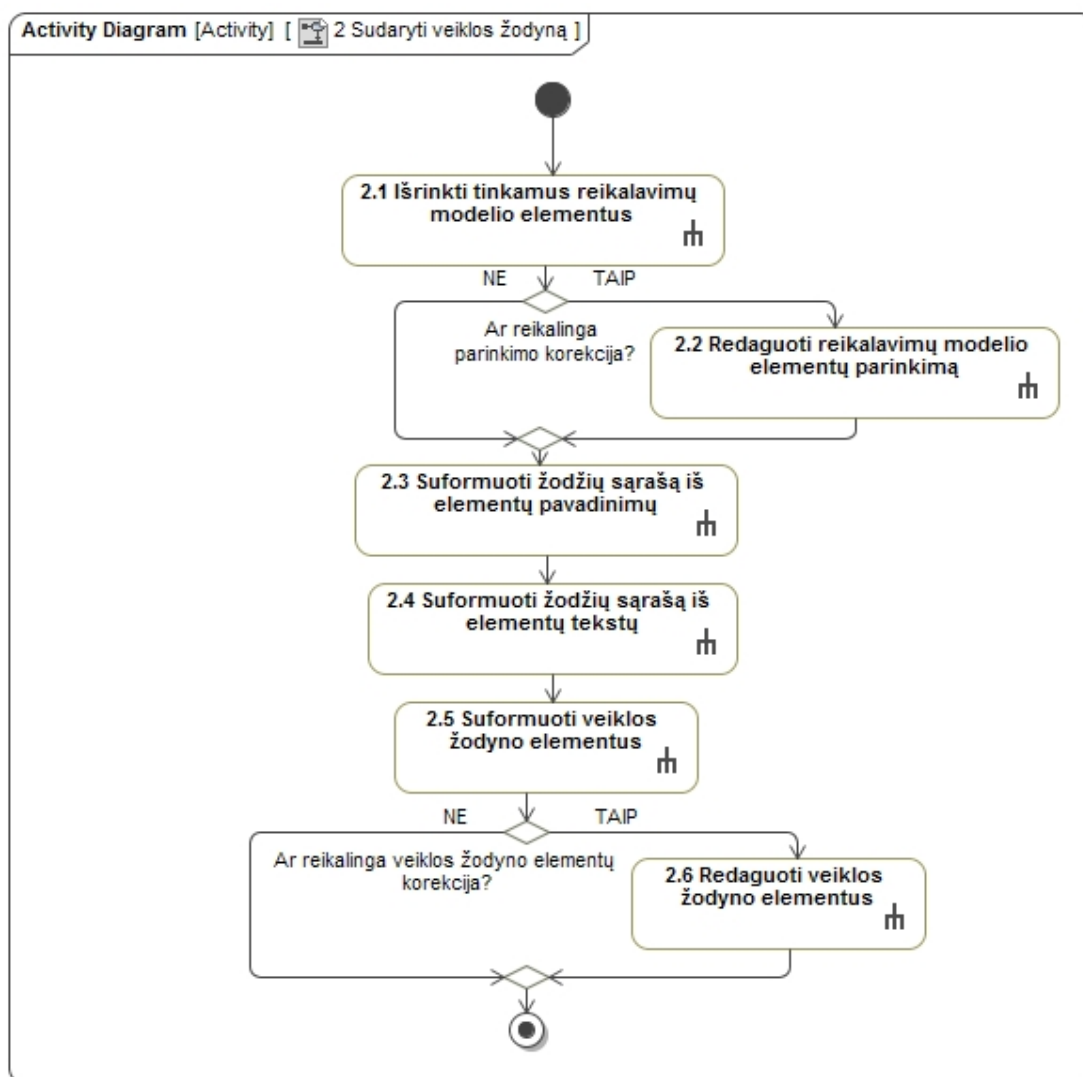


15 pav. „1 Įkelti reikalavimų modelį“ algoritmas

2.3.4. Sudaryti veiklos žodyną

Veiklos žodyno sudarymo algoritmas (16 pav.) išskaidomas į mažesnius algoritmo etapus, kurie pateikiami atskiromis veiklos diagramomis 17 - 22 pav. Algoritmo pagrindiniai žingsniai:

- Išrinkti tinkamus reikalavimų modelio elementus;
- Redaguoti reikalavimų modelio elementų pasirinkimą;
- Suformuoti žodžių sąrašą iš elementų pavadinimų;
- Suformuoti žodžių sąrašą iš elementų tekstų;
- Suformuoti veiklos žodyno elementus;
- Redaguoti veiklos žodyno elementus.



16 pav. „2 Sudaryti veiklos žodyną“ algoritmas

Algoritmo 2.3. ir 2.4. žingsniai (16 pav.) gali būti atliekami tiek nuosekliai, tiek ir lygiagrečiai. Reikalavimų modelio elementų redagavimas (16 pav., 2.2 žingsnis) bei veiklos žodyno elementų redagavimas (16 pav., 2.6 žingsnis) atliekami, jeigu sistemos vartotojas, peržiūrėjęs suformuotus elementų sąrašus, nusprendžia, kad reikia atlikti elementų korekcijas.

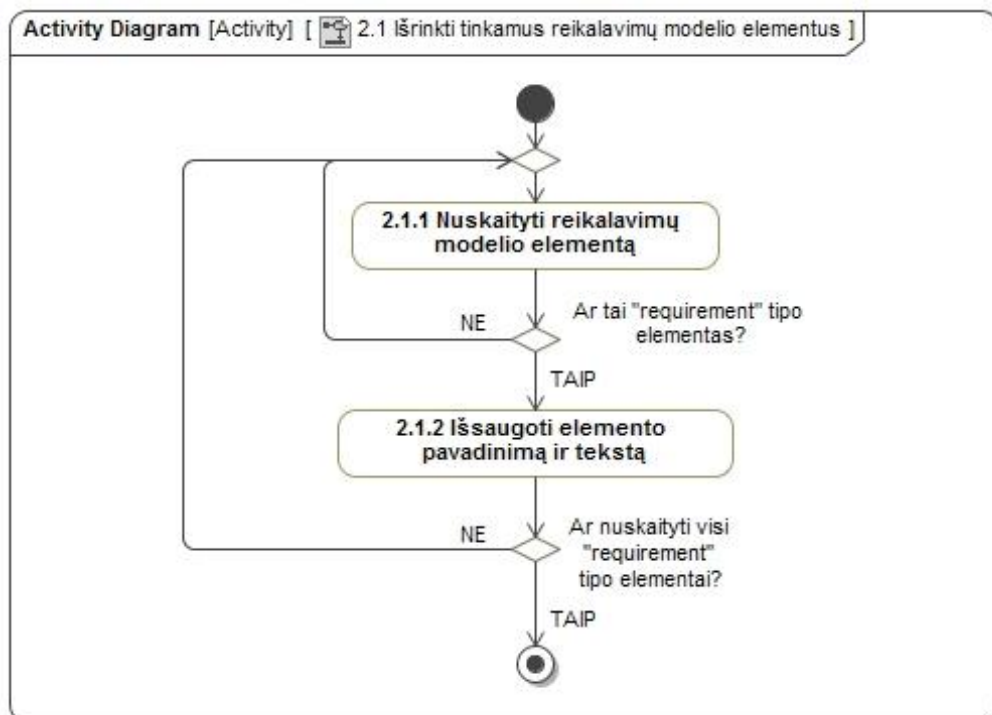
2.3.5. Išrinkti tinkamus reikalavimų modelio elementus

Taigi, pirmasis algoritmo žingsnis – išrinkti tinkamus *SysML* diagramos elementus. Sistema eilės tvarka nuskaito visus reikalavimų modelio elementus ir atrenka reikalavimo (angl. *requirement*) tipo elementus (17 pav.). Paprastai, *SysML* reikalavimų diagramoje reikalavimai gali turėti tokius atributų tipus:

1. Pavadinimas (angl. *name*);
2. ID;
3. Šaltinis (angl. *source*);
4. Tekstas (angl. *text*);
5. Tipas (angl. *kind*);
6. Verifikavimo metodas (angl. *verify method*);
7. Rizika (angl. *risk*).

Tačiau, atsižvelgiant į tai, kad ne visi reikalavimo atributai yra reikšmingi sudarant veiklos žodyną remiantis reikalavimų modeliu, tolesniam darbui su sistema bus išsaugomi tik šie reikalavimų elementų atributai:

1. Pavadinimas (angl. *name*);
2. Tekstas (angl. *text*).



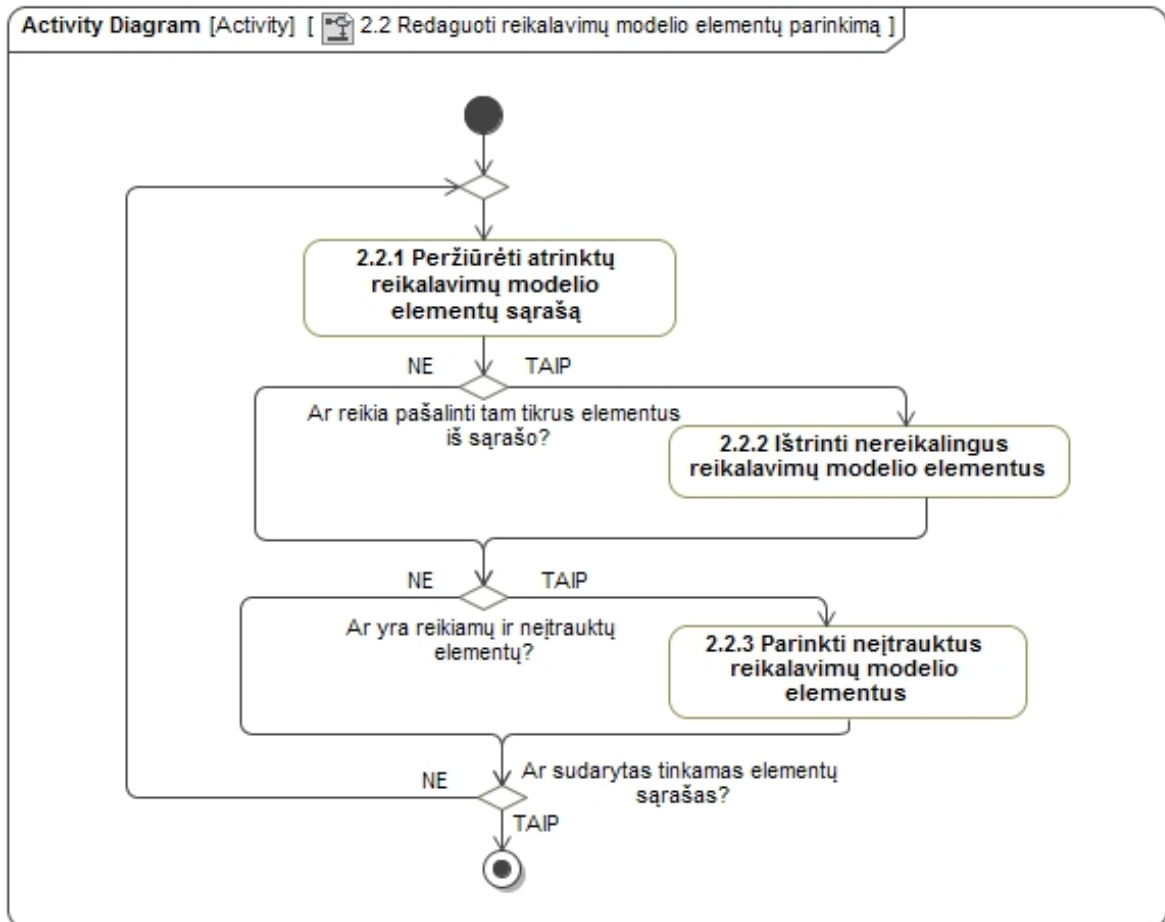
17 pav. „2.1 Išrinkti tinkamus *SysML* diagramos elementus“ algoritmas

Šis algoritmo žingsnis yra užbaigiamas, kuomet yra nuskaityti visi modelyje esantys reikalavimų tipo elementai.

2.3.6. Redaguoti reikalavimų modelio elementų pasirinkimą

Antruoju žingsniu, jeigu yra poreikis, vartotojas gali sutvarkyti atrinktų reikalavimo tipo elementų sąrašą (18 pav.). Jis gali būti nevykdomas visai, jeigu sistemos vartotojas nenori redaguoti parinktų elementų sąrašo.

Elementai sąrašė yra saugomi atskirai nurodant jų atributus. Tad, jeigu vartotojas nori dirbti tik su reikalavimų pavadinimais, arba tik su reikalavimų tekstais, šiame etape jis gali iš atrinktųjų elementų rankiniu būdu atsirinkti būtent tuos elementus ir jų atributus, su kuriais toliau nori tęsti darbą.



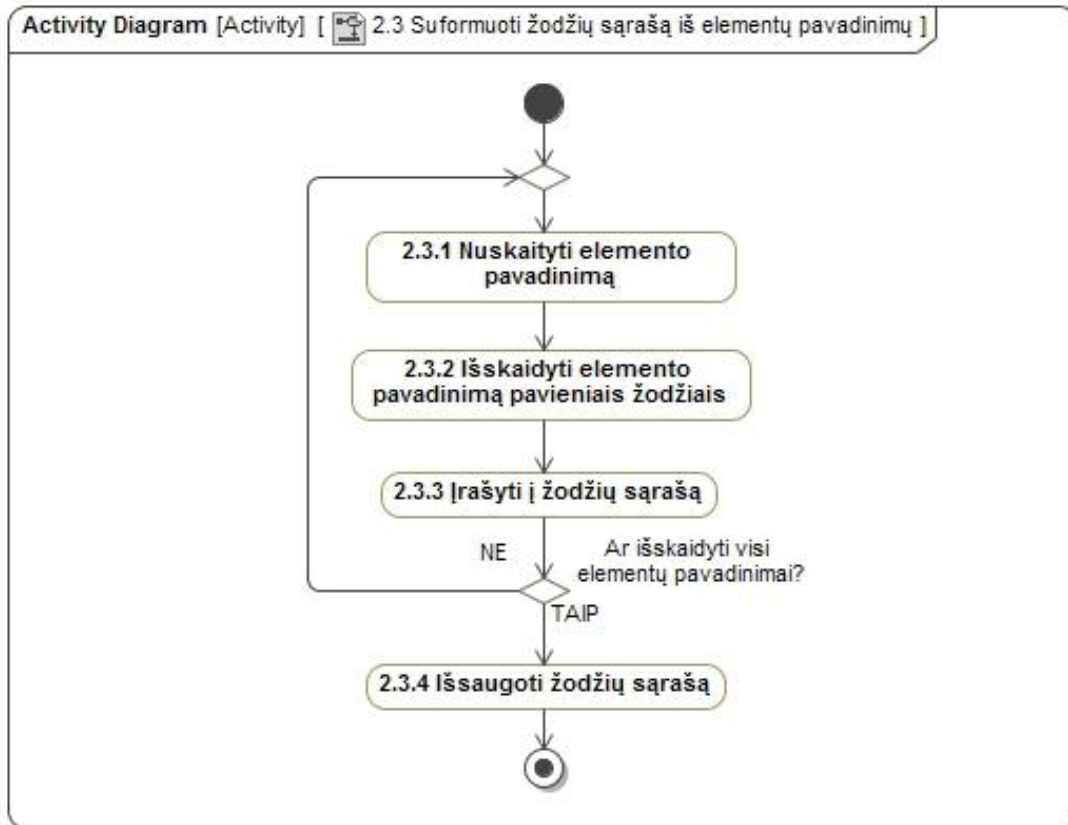
18 pav. „2.2 Redaguoti SysML diagramos elementų pasirinkimą“ algoritmas

Sistemos vartotojui leidžiama iš atrinktų elementų sąrašo pašalinti elementus, kurie jam neaktualūs, arba jeigu sistema padarė klaidų atrinkdama reikalavimų tipo elementus, jis gali rankiniu būdu įtraukti trūkstamus reikalavimo tipo elementus, kurių atributai bus išsaugomi vėlesniam darbui.

Šį algoritmo etapą sistemos vartotojas užbaigia tuomet, kai jį tenkina atrinktų elementų sąrašas.

2.3.7. Suformuoti žodžių sąrašą iš elementų pavadinimų

Trečiuoju žingsniu formuojamas žodžių sąrašas iš reikalavimų pavadinimų (19 pav.). Elementų pavadinimų skaidymas pavieniais žodžiais atliekamas paprastai – laikoma, kad žodžiai yra atskirti tarpo simboliais. Jeigu pavadinime yra naudojamas daiktavardžio kilmininko linksnis, kuris rašomas su apostrofu (pvz., *manager's*, *student's*), toks žodis skaitomas kaip vienas.



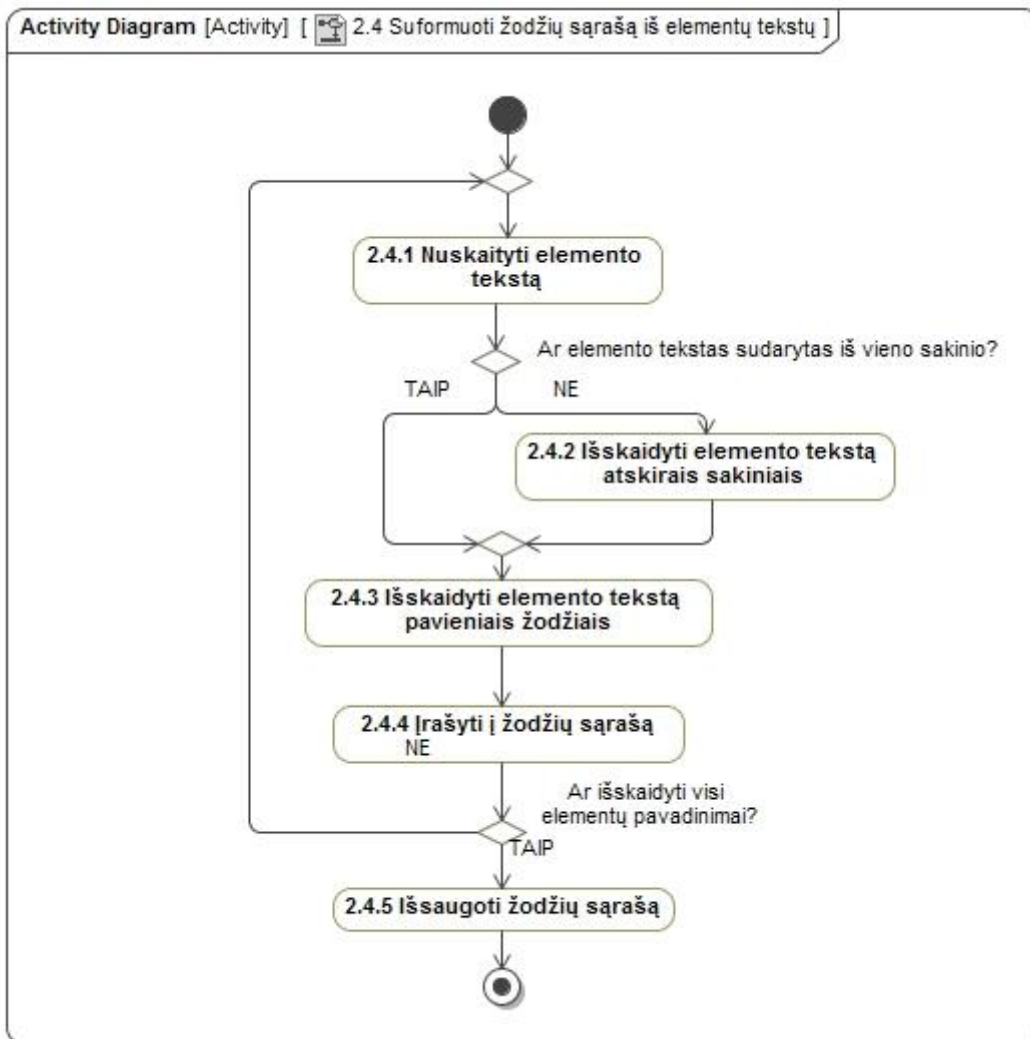
19 pav. „2.3 Suformuoti žodžių sąrašą iš elementų pavadinimų“ algoritmas

Šis algoritmo žingsnis yra užbaigiamas, kuomet yra nuskaityti visi reikalavimų tipo elementų pavadinimai, tuomet žodžių sąrašas yra išsaugomas sistemoje.

2.3.8. Suformuoti žodžių sąrašą iš elementų tekstų

Nepaisant to, ar veiklos žodyno sudarymo trečiasis ar ketvirtasis etapas atliekamas pirmiau, sistemoje sudaromas bendras žodžių sąrašas. Tai galioja ir šiuos etapus atliekant lygiagrečiai.

Ketvirtuoju algoritmo žingsniu formuojamas žodžių sąrašas iš reikalavimų tekstų (20 pav.). Pirmiausia, nustatoma, ar reikalavimų tipo elemento tekstas yra sudarytas iš kelių sakinių. Laikoma, kad atskiri sakiniai yra atskiri tašku ir kiekvienas sakinytis yra baigiamas tašku. Jeigu reikalavimo tekstas sudarytas iš kelių sakinių, tuomet iš pradžių jis skaidomas sakiniiais, o tik po to pavieniais žodžiais. Priešingu atveju, skaidymas sakiniiais yra praleidžiamas. Elementų tekstų suskaidymas pavieniais žodžiais atliekamas analogiškai, kaip ir elementų pavadinimų skaidymas pavieniais žodžiais.



20 pav. „2.4 Suformuoti žodžių sąrašą iš elementų tekstų“ algoritmas

Jeigu reikalavimo tekste yra nurodyti tam tikri nurodymai, apribojimai, kurie yra išreiškiami matematiniais simboliais ar išraiškomis, jie turi būti rankiniu būdu konvertuojami į tekstines išraiškas paliekant tik skaitmenis. Priešingu atveju, sakinius skaidant pavieniais žodžiais, simbolių rinkinys bus išskaidytas ir bus patiriami semantiniai nuostoliai. Toliau pateikiami pavyzdžiai, kaip turėtų būti atliekamas konvertavimas.

2.3.1 lentelė Matematinė simbolių konvertavimo pavyzdžiai

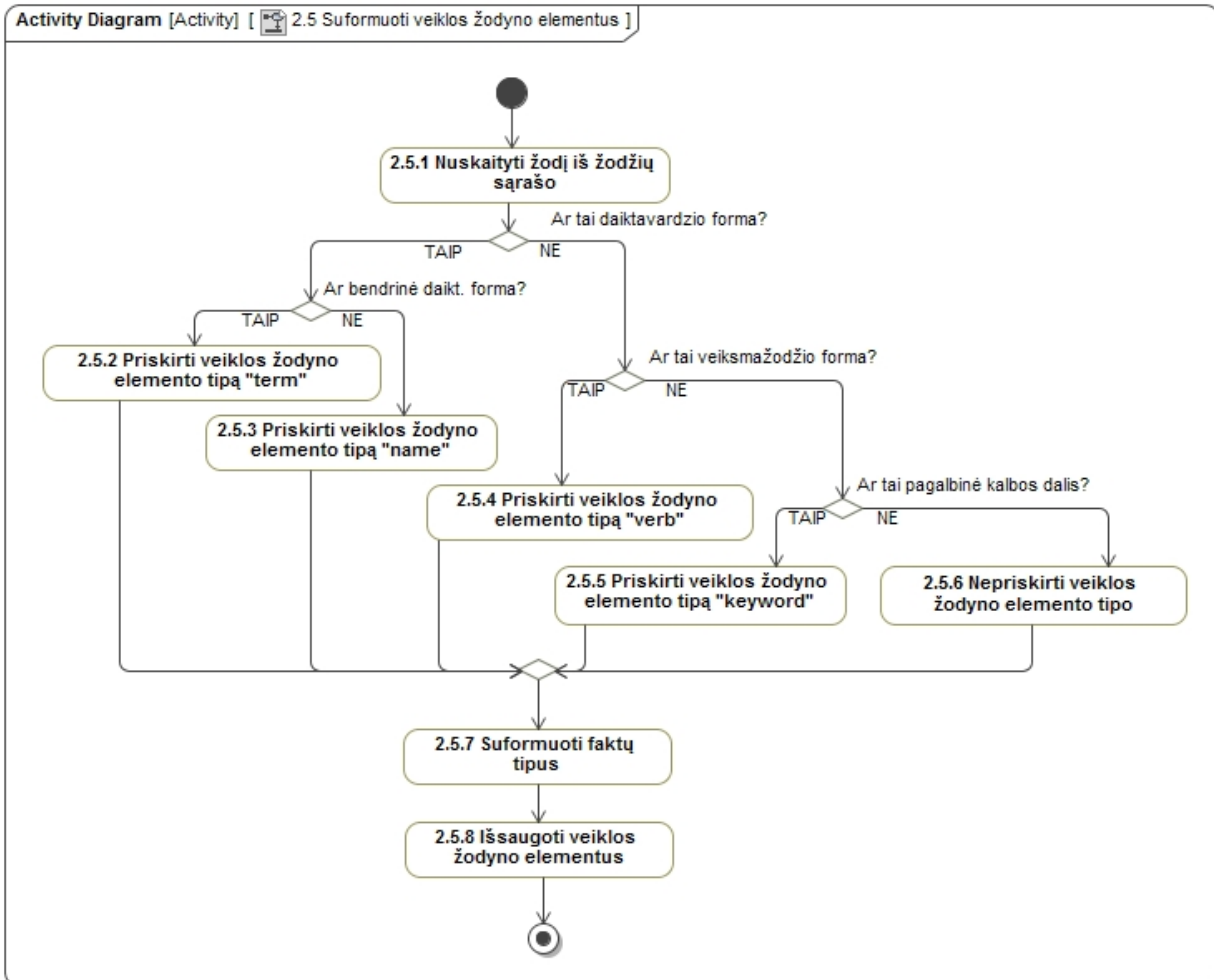
Matematinis simbolis	Tekstinė išraiška
$<; \leq$	<i>Less than; less than or equal to</i>
$>; \geq$	<i>Greater than; greater than or equal to</i>
$=; \neq; \approx$	<i>Equals; not equal to; approximately equals</i>

Žodžių sąrašo formavimas yra užbaigiamas, kuomet yra nuskaityti visi reikalavimų tipo elementų tekstai, tuomet sudarytas žodžių sąrašas yra išsaugomas sistemoje.

2.3.9. Suformuoti veiklos žodyno elementus

Veiklos žodyno formavimo etapas (21 pav.) susideda iš šių dviejų esminių žingsnių:

- reikalavimų pavadinimų ir tekstų žodžiams nustatomi veiklos žodyno tipai;
- suformuojami faktai ir nustatomi faktų tipai.



21 pav. „2.5 Suformuoti veiklos žodyno elementus“ algoritmas

Taigi, iš reikalavimų pavadinimų ir tekstų naudojantis *Stanford POS Tagger* programa sudaromi veiklos žodyno elementai, kurie skirstomi į vardus, terminus, veiksmažodžius ir raktažodžius tokiu būdu:

- Vardams priskiriami vienaskaitos ir daugiskaitos daiktavardiniai egzemplioriai (pvz., *Windows, Android*) ir skaitmenys.
- Terminams priskiriami vienaskaitos ir daugiskaitos bendriniai daiktavardiniai konceptai, kurie kai kuriais atvejais gali būti išreikšti ir keliais žodžiais (pvz., *user interface, order, client's list*). Paprastai, priskiriami keli iš eilės esantys daiktavardžiai.
- Veiksmažodžiams priskiriamos visos veiksmažodžio aktyvios ir pasyvios formos, kurios gali būti išreikštos ir keliais žodžiais (pvz., *are processing by, loading, create, is used by*).

- Raktažodžiams priskiriamos įvairios pagalbinės kalbos dalys: modaliniai veiksmažodžiai (pvz., *shall*), prielinksnio–būdvardžio–skaitvardžio (pvz., *at least one*) arba būdvardžio–skaitvardžio kombinacijos (pvz., *exclusively one*), jungtukai (pvz., *and, or, if*).

Išskiriama veiklos žodyno formavimas iš reikalavimų pavadinimų ir tekstų todėl, kad skiriasi šių atributų struktūros sudėtingumas. Formuojant veiklos žodyną iš reikalavimų pavadinimų gali būti sudaryti tik vardai/terminai ir veiksmažodžiai. Taip yra todėl, kad reikalavimų pavadinimus rekomenduojama įvardinti konkrečiai, kuo aiškiau ir trumpiau, vengiant kitų pagalbinių kalbos dalių. Toliau pateikiami veiklos žodyno formavimo iš reikalavimų pavadinimų pavyzdžiai (2.3.2 – 2.3.4 lentelės).

2.3.2 lentelė Veiklos žodyno formavimo pavyzdys Nr.1

Reikalavimo pavadinimas		<i>Model processing</i>
Žodis	Žodžio tipas	SBVR elemento tipas
<i>Model</i>	Daiktavardis	<i>Name</i>
<i>Processing</i>	Veiksmažodis	<i>Verb</i>

2.3.3 lentelė Veiklos žodyno formavimo pavyzdys Nr.2

Reikalavimo pavadinimas		<i>User interface</i>
Žodis	Žodžio tipas	SBVR elemento tipas
<i>User</i>	Daiktavardis	<i>Name</i>
<i>Interface</i>	Daiktavardis	<i>Name</i>

2.3.4 lentelė Veiklos žodyno formavimo pavyzdys Nr.3

Reikalavimo pavadinimas		<i>Project site loading performance</i>
Žodis	Žodžio tipas	SBVR elemento tipas
<i>Project</i>	Daiktavardis	<i>Name</i>
<i>Site</i>	Daiktavardis	<i>Name</i>
<i>Loading</i>	Veiksmažodis	<i>Verb</i>
<i>Performance</i>	Daiktavardis	<i>Name</i>

Formuojant veiklos žodyną iš reikalavimų tekstų gali būti sudaromi vardai, terminai, veiksmažodžiai ir raktažodžiai. Todėl, kad reikalavimų tekstai yra nurodyti natūralios kalbos sakiniais, kuriuose naudojama daugiau skirtingų kalbos dalių. Toliau pateikiami veiklos žodyno formavimo pavyzdžiai iš reikalavimų tekstų (2.3.5 – 2.3.6 lentelės).

2.3.5 lentelė Veiklos žodyno formavimo pavyzdys Nr.4

Reikalavimo tekstas		<i>Vehicle stops within ten seconds</i>
Žodis	Žodžio tipas	SBVR elemento tipas
<i>Vehicle</i>	Daiktavardis	<i>Name</i>
<i>Stops within</i>	Veiksmažodis	<i>Verb</i>
<i>Ten</i>	Skaitvardis	<i>Name</i>
<i>Seconds</i>	Daiktavardis	<i>Name</i>

2.3.6 lentelė Veiklos žodyno formavimo pavyzdys Nr.5

Reikalavimo tekstas		<i>System is preventing from mistakes</i>
Žodis	Žodžio tipas	SBVR elemento tipas
<i>System</i>	Daiktavardis	<i>Name</i>
<i>Is preventing</i>	Veiksmažodis	<i>Verb</i>
<i>From</i>	Prielinksnis	<i>Keyword</i>
<i>Mistakes</i>	Daiktavardis	<i>Name</i>

Fakto tipai vėliau yra panaudojami veiklos taisyklėms formuoti, kadangi, veiklos taisyklės nebus formuojamos iš reikalavimų pavadinimų, tačiau, fakto tipai, gali būti sudaromi ir iš reikalavimų pavadinimų.

Svarbu pastebėti, kad sudarant faktus yra ne praleidžiami modaliniai veiksmažodžiai. Fakto tipai sudaromi pagal šabloninę formą *Term/Name-Verb-Term/Name*, kurios ilgį apibrėžia vardu/terminų kiekis. Priklausomai nuo faktų sudarančių vardu/terminų kiekio faktai gali būti unariniai, dvinariai, trinariai ir tt. Taip pat, esant atitinkamai reikalavimo teksto struktūrai, gali būti sudaryti kategorijos (angl. *categorisation*), nuosavybės (angl. *property*), struktūros (angl. *partitive*) ir rūšies (angl. *assortment*) nurodymo faktai.

Toliau pateikiami pavyzdžiai, kaip iš reikalavimų tekstų gali būti sudaryti faktų tipai (2.3.7 – 2.3.9 lentelės).

2.3.7 lentelė Fakto tipų formavimo pavyzdys Nr.1

Reikalavimo tekstas	<i>Button closes application</i>
Faktas	Fakto tipas
<i>Button closes application</i>	Dvinaris

2.3.8 lentelė Fakto tipų formavimo pavyzdys Nr.2

Reikalavimo tekstas	<i>Manager sends product to customer if only payment is received</i>
Faktas	Fakto tipas
<i>Manager sends product to customer</i>	Trinariai
<i>Payment is received</i>	Unarinis

2.3.9 lentelė Fakto tipų formavimo pavyzdys Nr.3

Reikalavimo tekstas	<i>Employee sends item by mail if item is document</i>
Faktas	Fakto tipas
<i>Employee sends item by mail</i>	Trinariai
<i>Item is document</i>	Dvinaris; nuosavybės nurodymo

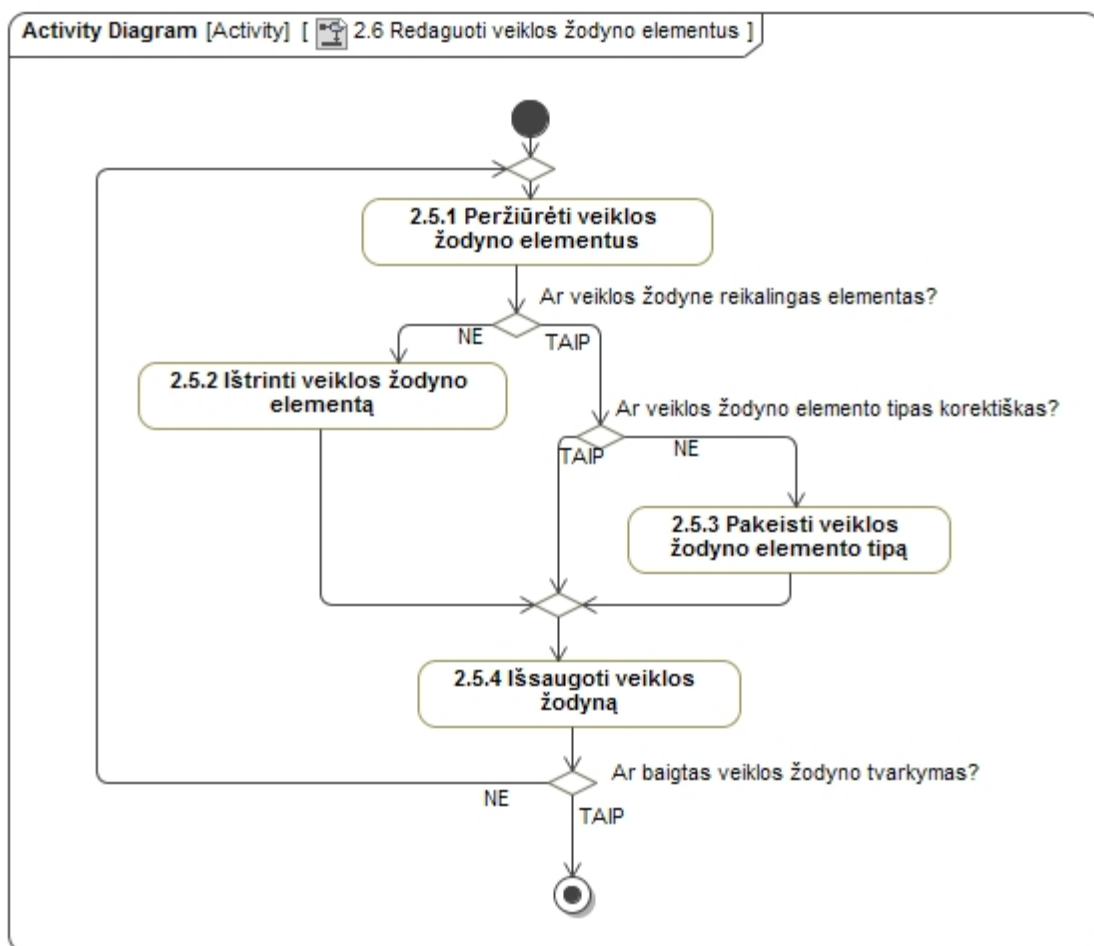
Su visais žodžių sąrašo elementais atlikus šiuos veiklos žodyno elementų formavimo ir fakto tipų nustatymo veiksmus veiklos žodynas yra išsaugomas. Papildomai, išsaugojimo metu sistema automatinio būdu pašalina pasikartojančius veiklos žodyno elementus.

2.3.10. Redaguoti veiklos žodyno elementus

Jeigu reikalavimų pavadinimų ir tekstų struktūra sudaryta pagal nurodytas rekomendacijas, veiklos žodyno algoritmas neturėtų padaryti klaidų nustatant veiklos žodyno elementų tipus. Tačiau, jeigu atsirado nekorektiškai nustatytų elementų, penktame veiklos žodyno algoritmo žingsnyje (22 pav.) sistemos vartotojas gali peržiūrėti praėjusiame etape arba ankstesnėje darbo su sistema sesijoje suformuotą veiklos žodyną.

Jeigu vartotojas pastebi veiklos žodyno elementų tipo nustatymo metu padarytų klaidų, jis gali rankiniu būdu pakeisti elemento tipą ir išsaugoti pakeitimus sistemoje. Taip pat, jeigu yra klaidingai priskirtų žodžių arba beprasmiškų žodyno elementų, kurie nėra aktualūs, tuomet sistemos vartotojas gali juos ištrinti iš sudaryto veiklos žodyno. Atlikus visas korekcijas, veiklos žodynas yra išsaugomas.

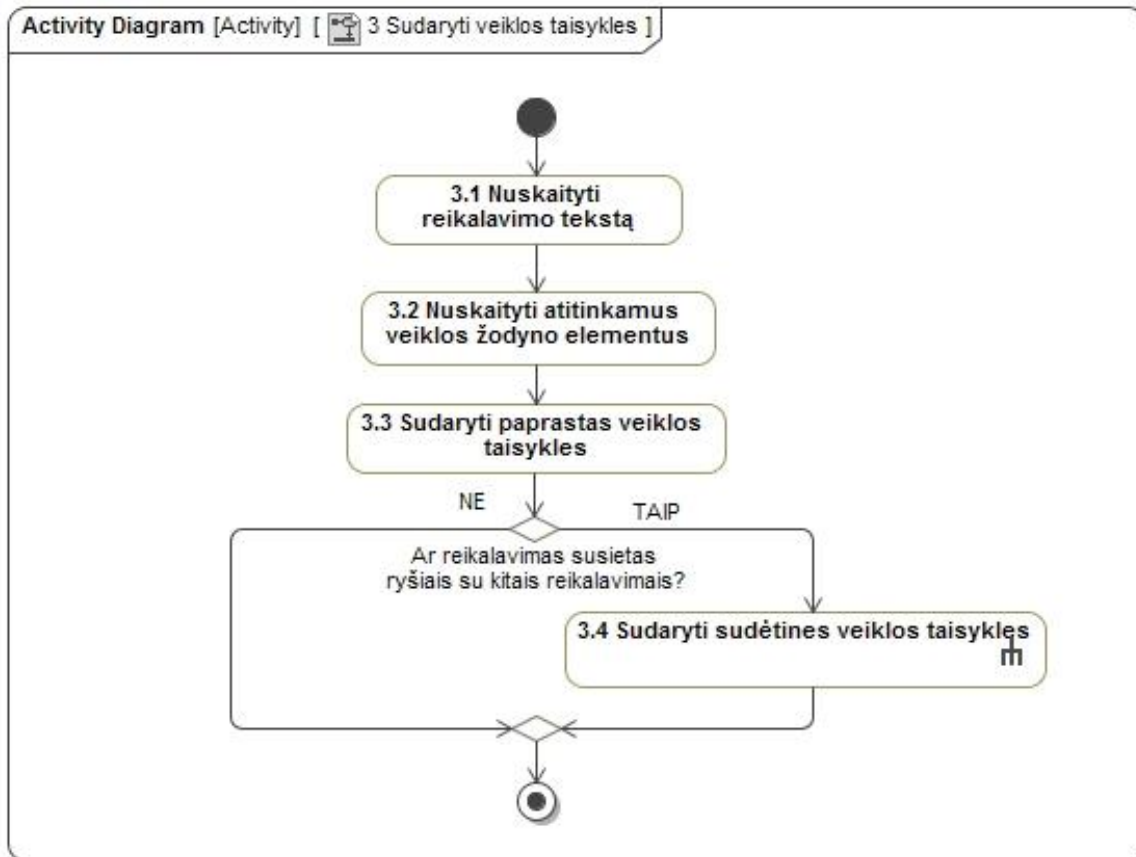
Taigi, šį algoritmo etapą sistemos vartotojas užbaigia tuomet, kai jį tenkina suformuotas veiklos žodyno elementų sąrašas.



22 pav. „2.6 Redaguoti veiklos žodyno elementus“ algoritmas

2.3.11. Sudaryti veiklos taisykles

Veiklos taisyklių sudarymo iš reikalavimų modelio elementų algoritmas pradedamas nuo reikalavimo teksto nuskaitymo (23 pav., 3.1 žingsnis) ir atitinkamų praeitime etape suformuotų veiklos žodyno elementų nuskaitymo (23 pav., 3.2 žingsnis). Kuomet nuskaitytas reikalavimo tekstas ir iš jo sudarytų veiklos žodyno elementų sąrašas, galima formuoti veiklos taisykles. Gali būti sudaromos paprastos veiklos taisyklės, ir jeigu reikalavimas turi ryšių su kitais reikalavimais, gali būti formuojamos sudėtinės veiklos taisyklės pagal esamus ryšius tarp reikalavimų.



23 pav. „3 Sudaryti veiklos taisykles“ algoritmas

Svarbu pastebėti, kad čia nuskaitytą veiklos žodyno elementus iš raktažodžių aibės išskiriami modaliniai raktažodžiai, kurie yra esmingai reikšmingas dėmuo veiklos taisyklių sudarymui. Leistinas modalinis raktažodis yra „*shall*“ arba, kitu atveju, geriau nenaudoti jokio.

Sudaryti paprastas veiklos taisykles

Paprastų veiklos taisyklių struktūra gali būti sudaroma tokiu būdu, priklausomai nuo reikalavimo teksto sakinio:

1. <taisyklės formuluotė> <faktas>, jei iš sakinio sudaromas 1 faktas;
2. <taisyklės formuluotė> <vardas1/terminas1/faktas1> <raktažodis>
<vardas2/terminas2/faktas2>, jei iš sakinio sudaromi 2 ir daugiau faktų;

Čia <raktažodis> – kiekybinis, loginis ar kitas, bet ne modalinis.

Veiklos taisyklės elementai (vardai, terminai, faktai ir raktažodžiai) atitinka reikalavimo veiklos žodyno elementus. Veiklos taisyklės formuluotė parenkama pagal tai, koks modalinis raktažodis buvo panaudotas reikalavimo sakinyje (2.3.10 lentelė).

2.3.10 lentelė Veiklos taisyklės formuluotės parinkimas

Veiklos taisyklės formuluotė	Reikalavimo modalinis raktažodis
<i>It is necessary that</i>	-
<i>It is obligatory that</i>	<i>Shall</i>

Toliau pateikiami paprastųjų veiklos taisyklių sudarymo pavyzdžiai (2.3.11 – 2.3.13 lentelės).

2.3.11 lentelė Veiklos taisyklės sudarymo pavyzdys Nr.1

Reikalavimo tekstas:	<i>Button closes application</i>
Faktas	Fakto tipas
<i>Button closes application</i>	Dvinaris
Veiklos taisyklė:	<i>It is obligatory that button closes application</i>

2.3.12 lentelė Veiklos taisyklės sudarymo pavyzdys Nr.2

Reikalavimo tekstas:	<i>Manager sends product to customer if only payment is received</i>
Faktas	Fakto tipas
<i>Manager sends product to customer</i>	Trinaris
<i>Payment is received</i>	Unarinis
Veiklos taisyklė:	<i>It is obligatory that manager sends product to customer if only payment is received</i>

2.3.13 lentelė Veiklos taisyklės sudarymo pavyzdys Nr.3

Reikalavimo tekstas:	<i>Employee sends item by mail if item is document</i>
Faktas	Fakto tipas
<i>Employee sends item by mail</i>	Trinaris
<i>Item is document</i>	Dvinaris; nuosavybės nurodymo
Veiklos taisyklė:	<i>It is prohibited that employee sends item by mail if item is document</i>

Jeigu reikalavimo tekstą sudaro daugiau nei vienas sakinys, veiklos taisyklės gali būti apjungiamos tokiu būdu:

1. *<taisyklės1 formuluotė> <taisyklės1 tekstas> AND <taisyklės2 tekstas>*, jei sakinių modaliniai raktažodžiai sutampa;
2. *<taisyklės1 formuluotė> <taisyklės1 tekstas> AND <taisyklės2 formuluotė> <taisyklės2 tekstas>*, jei sakinių modaliniai raktažodžiai skiriasi;

Čia *<taisyklės tekstas> = <vardas1/terminas1/faktas1>* arba

<taisyklės tekstas> = <vardas1/terminas1/faktas1> <raktažodis1> <vardas2/terminas2/faktas2> .

Toliau pateikiami tokių veiklos taisyklių sudarymo ir apjungimo pavyzdžiai (2.3.14 – 2.3.15 lentelės).

2.3.14 lentelė Veiklos taisyklės sudarymo pavyzdys Nr.4

Reikalavimo tekstas:	<i>Employee sends item by mail if item is document. Employee answers to document request.</i>
Faktas	Fakto tipas
<i>Employee sends item by mail</i>	Trinaris
<i>Item is document</i>	Dvinaris; nuosavybės nurodymo
<i>Employee answers to document request</i>	Trinaris
Veiklos taisyklė:	<i>It is obligatory that employee sends item by mail if item is document AND it is obligatory that employee answers to document request</i>

2.3.15 lentelė Veiklos taisyklės sudarymo pavyzdys Nr.5

Reikalavimo tekstas:	<i>Keyboards have fifty buttons. Mice have at least two buttons.</i>
Faktas	Fakto tipas
<i>Keyboards have fifty buttons</i>	Dvinaris
<i>Mice have two buttons</i>	Dvinaris
Veiklos taisyklė:	<i>It is obligatory that keyboards have fifty buttons AND mice have at least two buttons</i>

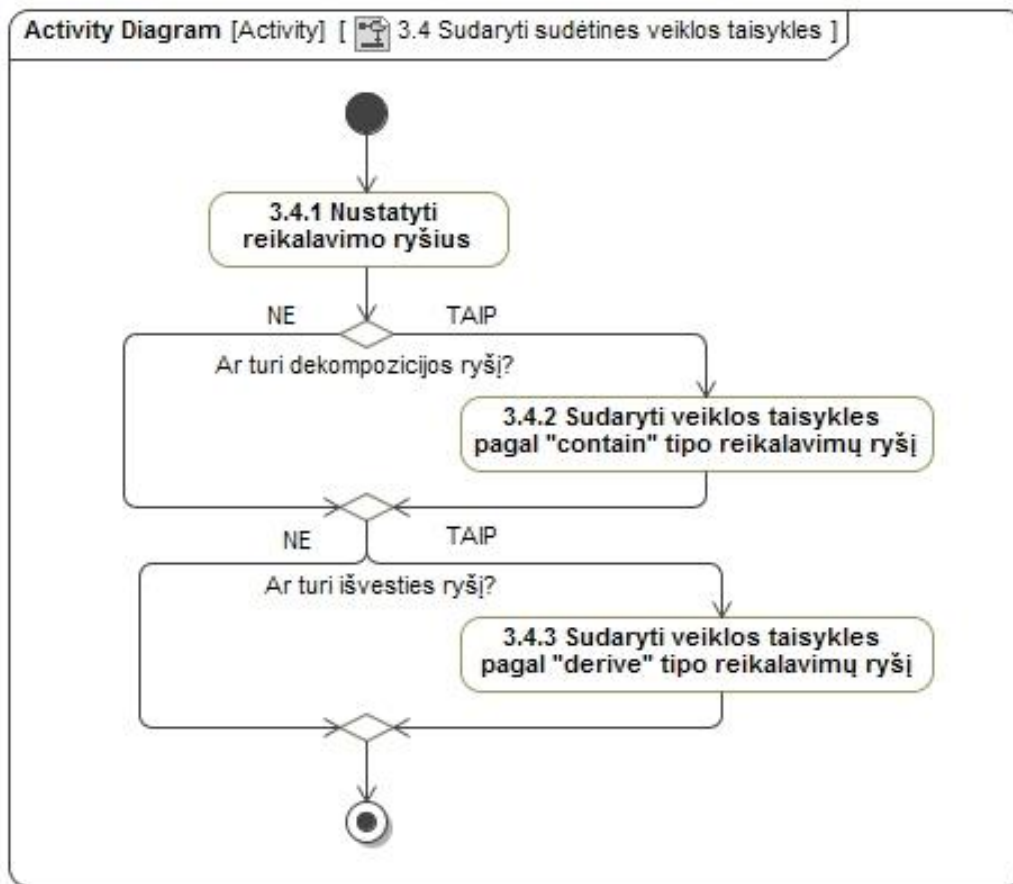
Sudaryti sudėtinės veiklos taisyklės

Sudėtinės veiklos taisyklės yra sudaromos tuomet, kai reikalavimų modelyje tarp reikalavimų yra ryšių, priešingu atveju, po paprastų veiklos taisyklių sudarymo algoritmas baigiamas. Kuomet yra ryšys tarp reikalavimų, pirmiausia nustatomas jo tipas, paskui pagal nustatytą tipą formuojamos sudėtinės veiklos taisyklės (24 pav.).

Taigi, tarp reikalavimų modelyje tarp dviejų reikalavimų gali būti specifikuoti tokie ryšiai:

- Hierarchinis (angl. *containment*) ryšys;
- Išvesties (angl. *derive*) ryšys.

Rekomenduojama reikalavimų modelyje tarp reikalavimų nedėti trasavimo arba vartotojo apibrėžtų ryšių. Sudarant sudėtinės veiklos taisykles toki ryšiai tiesiog yra praleidžiami, ignoruojami.



24 pav. „3.4 Sudaryti sudėtinės veiklos taisykles“ algoritmas

Jeigu tarp reikalavimų yra hierarchinis ryšys, sudėtinė veiklos taisyklė sudaroma tokiu būdu:

1. $\langle \text{taisyklė}_0 \text{ formulotė} \rangle \langle \text{taisyklė}_0 \text{ tekstas} \rangle, SO \langle \text{taisyklė}_1 \text{ formulotė} \rangle \langle \text{taisyklė}_1 \text{ tekstas} \rangle$, jeigu reikalavimų hierarchiją sudaro du reikalavimai;
2. $\langle \text{taisyklė}_0 \text{ formulotė} \rangle \langle \text{taisyklė}_0 \text{ tekstas} \rangle, SO \langle \text{taisyklė}_1 \text{ formulotė} \rangle \langle \text{taisyklė}_1 \text{ tekstas} \rangle AND \langle \text{taisyklė}_2 \text{ formulotė} \rangle \langle \text{taisyklė}_2 \text{ tekstas} \rangle AND .. AND \langle \text{taisyklė}_N \text{ formulotė} \rangle \langle \text{taisyklė}_N \text{ tekstas} \rangle$, jeigu reikalavimų hierarchiją sudaro daugiau nei du reikalavimai;

Čia *taisyklė₀* – aukščiausio hierarchinio lygio veiklos taisyklė sudaryta pagal aukščiausio lygio hierarchinį reikalavimą, *taisyklė₁*, *taisyklė₂*, .. *taisyklė_N* – taisyklė₀ sub-taisyklės, sudarytos pagal sub-reikalavimus;

Sub-taisyklių apjungimas *AND* operatoriumi atliekamas analogiškai, kaip taisyklių apjungimas, kuomet reikalavimas sudarytas iš kelių sakinių;

Jeigu sub-taisyklė sudaryta iš kelių paprastųjų taisyklių (nes reikalavimas sudarytas iš daugiau nei vieno sakinio) jos lygiaverčiai prijungiamos *AND* operatoriumi.

Toliau pateikiamas sudėtinių taisyklių pagal hierarchinį ryšį formavimo pavyzdys (2.3.16 lentelė)

2.3.16 lentelė Veiklos taisyklės sudarymo pavyzdys Nr.6

Reikalavimo tekstas:	<i>Calculator weights more than 1 kilogram</i>
Faktas	Fakto tipas
<i>Calculator weights 1 kilogram</i>	Dvinaris
Sub-reikalavimo tekstas:	<i>Battery weights less than 700 grams</i>
Faktas	Fakto tipas
<i>Battery weights 700 grams</i>	Dvinaris
Sudėtinė veiklos taisyklė:	<i>It is prohibited that calculator weights more than 1 kilogram, SO it is obligatory that battery weights less than 700 grams</i>

Jeigu tarp reikalavimų yra išvesties ryšys, sudėtinė veiklos taisyklė sudaroma tokiu būdu:

1. IF <taisyklės1 formuluotė> <taisyklės1 tekstas>, THEN <taisyklės2 formuluotė> <taisyklės2 tekstas>, jeigu išvestas reikalavimas sudarytas iš vieno sakinio;
2. IF <taisyklės1 formuluotė> <taisyklės1 tekstas>, THEN <taisyklės2 formuluotė> <taisyklės2 tekstas> AND <taisyklės3 formuluotė> <taisyklės3 tekstas> AND.. AND <taisyklėsN formuluotė> <taisyklėsN tekstas>, jeigu išvestasis reikalavimas sudarytas daugiau nei vieno sakinio;

Čia *taisyklė1* – sudaryta iš tėvinio reikalavimo,

taisyklė2, taisyklė3, .. , taisyklėN – sudarytos iš išvestojo reikalavimo sakinių.

Toliau, pateikiamas sudėtinių taisyklių pagal išvesties ryšį formavimo pavyzdys (2.3.17 lentelė).

2.3.17 lentelė Veiklos taisyklės sudarymo pavyzdys Nr.7

Reikalavimo tekstas:	<i>Keyboard has at least forty buttons.</i>
Faktas	Fakto tipas
<i>Keyboard has forty buttons</i>	Dvinaris
Išvestojo reikalavimo tekstas:	<i>Letters have buttons. Numbers have buttons.</i>
Faktas	Fakto tipas
<i>Letters have buttons</i>	Dvinaris
<i>Numbers have buttons</i>	Dvinaris
Sudėtinė veiklos taisyklė:	<i>IF it is obligatory that keyboard has at least forty buttons, THEN it is obligatory that letters have buttons AND numbers have buttons</i>

2.4. Apribojimai ir rekomendacijos

Apribojimai reikalavimų modeliui

Reikalavimų modelio apribojimus galima suskirstyti grupėmis:

- I. Apribojimai reikalavimų pavadinimų formavimui;
- II. Apribojimai reikalavimų tekstų formavimui;
- III. Apribojimai ryšiams tarp reikalavimų,
- IV. Rekomendacijos sistemos modelio sudarymui.

I. Apribojimai reikalavimų pavadinimų formavimui

Pirmiausiai, reikalavimo pavadinimas turi būti įvardintas kuo trumpiau ir konkrečiau. Pavadinimas turi neviršyti 6 – 8 žodžių ilgio, jeigu viršija trumpinti, konkretizuoti pavadinimą arba skaidyti į atskirus reikalavimus.

Pavadinimas turi būti sudarytas iš daiktavardžių ir veiksmažodžių konceptų, bet ne iš sakinių, pvz.: *user account control settings, system 's parameters setting up*.

II. Apribojimai reikalavimų tekstų formavimui

Reikalavimo tekstas turi būti įvardintas kuo aiškiau ir konkrečiau laikantis šių nurodymų:

1. Teksto struktūra turi būti kuo artimesnė struktūrai:

<Terminas> <Modalinis raktažodis> <Veiksmažodis> <Raktažodis> <Terminas>,
(pastaba, vietoje termino gali būti ir vardas) pvz.:

<i>data table must have exactly 5 columns = <data table> <must> <have> <exactly> <5 columns></i>
--

2. Gali būti panaudotas modalinis veiksmažodis *shall* prieš veiksmažodį.
3. Reikalavimo teksto sakinytis turi turėti sakinio gale tašką.

III. Apribojimai ryšiams tarp reikalavimų

Patartina, specifiuoti tik konkrečius *SysML* meta-modelyje apibrėžtus ryšius tarp reikalavimų (pvz., *derive, containment*). Nerekomenduojama naudoti trasavimo ryšių arba specifiuoti ryšių pavadinimus, siekiant, kad nebūtų semantinių nuostolių.

Patartina, vietoj sudėtingų reikalavimų sudaryti trumpų ir konkrečių reikalavimų hierarchiją, pvz.: vietoje reikalavimo *window layout design and appearance*, skaidyti į atskirus reikalavimus *window layout design* ir *window appearance*.

IV. Rekomendacijos veiklos modeliui sudaryti

Suformuoto veiklos žodyno ir taisyklių elementai gali būti pernaudojami sudarant tam tikrus veiklos modelio elementus. Tokiu būdu, sudaromas reikalavimų ir veiklos modelių integralumas, sumažinamos modeliavimui reikalingos laiko sąnaudos. Pateikiamas, plačiau aprašytas pritaikymas (2.4.1 lentelė).

2.4.1 lentelė *SBVR* elementų pernaudojimas sistemos modeliui

SBVR elementas	Veiklos modelio elementas
Vardas/Terminas	Sistemos/Bloko elemento pavadinimas
Faktas	Bloko elemento parametrai, apribojimai; apribojimų elemento atributai
Veiklos taisyklė	Bloko elemento parametrai, apribojimai; apribojimų elemento atributai

Veiklos žodyno terminai ir vardai gali būti panaudojami *SysML* blokų modelio elementų pavadinimams specifikuoti. Dažniausiai, jeigu yra konkrečiai ir aiškiai specifikuoti reikalavimų elementų pavadinimai iš daiktavardinių konceptų, jie gali ir pilnai atitikti sistemos blokų vardus. Faktai gali būti panaudojami sistemos veiklos diagramų modeliavimui, veiklos pavadinimams specifikuoti arba apribojimams apibrėžti.

Veiklos taisyklės gali būti panaudotos *SysML* apribojimams ir parametrms modeliuoti. Jeigu tam tikro elemento reikalavimas apibrėžia to elemento parametrus, tuomet iš jo išvestos veiklos taisyklės informacija gali būti pernaudojama sistemos blokų parametrms (angl. *values*) ir apribojimams (angl. *constraints*) modeliuoti.

3. VEIKLOS APRIBOJIMŲ AUTOMATIZUOTO IŠGAVIMO IŠ REIKALAVIMŲ MODELIO ALGORITMO TESTAVIMAS, REALIZACIJA IR REZULTATAI

3.1. Algoritmo testavimo etapai

Veiklos apribojimų automatizuoto išgavimo iš reikalavimų modelio algoritmas testuojamas pagal pagrindinius algoritmo žingsnius:

1. Reikalavimų modelio duomenų nuskaitymas ir atvaizdavimas;
2. Veiklos žodyno elementų išgavimas;
3. Veiklos žodyno faktų elementų formavimas.

Kiekvieno iš šių etapų testavimas atliekamas, kai yra realizuojamos jo vykdymui reikalingos funkcijos. Toliau apibrėžiamas kiekvienas iš etapų.

Reikalavimų modelio duomenų nuskaitymas ir atvaizdavimas

Testavimo metu *MagicDraw* įrankiu sukuriama skirtingi reikalavimų modeliai, kurie yra eksportuojami *.reqif* duomenų formatu. Eksportuoti dokumentai yra pagrįsti *.xml* kalba. Toliau, eksportuoti dokumentai yra po vieną įkeliami į sistemą. Sistemoje nuskaityta reikalavimų modelių informacija yra pateikiama reikalavimų pavadinimų ir tekstų sąrašu, kurį galima palyginti su sukurtais reikalavimų modeliais. Tokiu būdu patikrinama, ar teisingai veikia duomenų nuskaitymas ir atvaizdavimas.

Veiklos žodyno elementų išgavimas

Testavimo metu iš parinktos reikalavimų informacijos formuojami veiklos žodyno elementai: terminai, vardai ir veiksmažodžiai. Tikrinami reikalavimų pavadinimų ir tekstų žodžiai po vieną bei jų priklausomybė veiklos žodyno elemento tipui. Grafiniame vartotojo sąsajos lange žodžiai pateikiami išskaidyti ir priskirti veiklos žodyno elementų tipams. Gauti rezultatai palyginami su tikėtaisiais. Tokiu būdu patikrinamas gautas veiklos žodynas ir jo elementų teisingumas.

Veiklos žodyno faktų elementų formavimas

Testavimo metu iš parinktos reikalavimų informacijos formuojami pagal sudarytus veiklos žodyno elementus formuojami veiklos žodyno faktai ir nustatomi jų tipai. Grafiniame vartotojo sąsajos lange pateikiami sudaryti veiklos žodyno faktai ir jų tipai. Gauti rezultatai palyginami su tikėtaisiais. Tokiu būdu patikrinamas veiklos žodyno faktų ir jų tipų teisingumas.

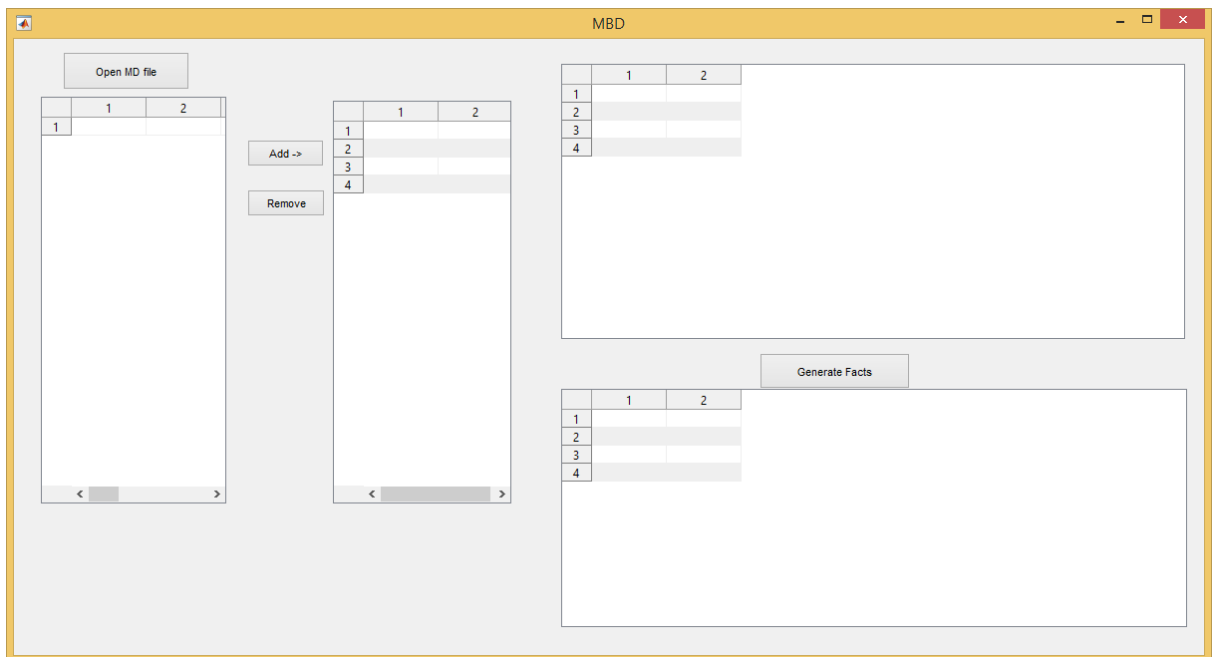
3.2. Prototipo realizacija, funkcionalumas

Sukurta vartotojo sąsaja, leidžianti įgyvendinti pagrindinius automatizuotus veiklos apribojimų veiksmus. Vartotojo sąsaja sudaryta iš vieno pagrindinio lango, kuriame vienu metu galima peržiūrėti ir tvarkyti visus turimus duomenis. Pagal sudarytą veiklos apribojimų automatinio išgavimo iš reikalavimų modelio metodą *Matlab* aplinkoje realizuotas prototipas, apimantis šias funkcijas:

1. Reikalavimų modelio duomenų nuskaitymas ir atvaizdavimas;
2. Reikalavimų modelio elementų sąrašo redagavimas ir papildymas;
3. Veiklos žodyno elementų sudarymas ir redagavimas;
4. Veiklos žodyno faktų elementų sudarymas.

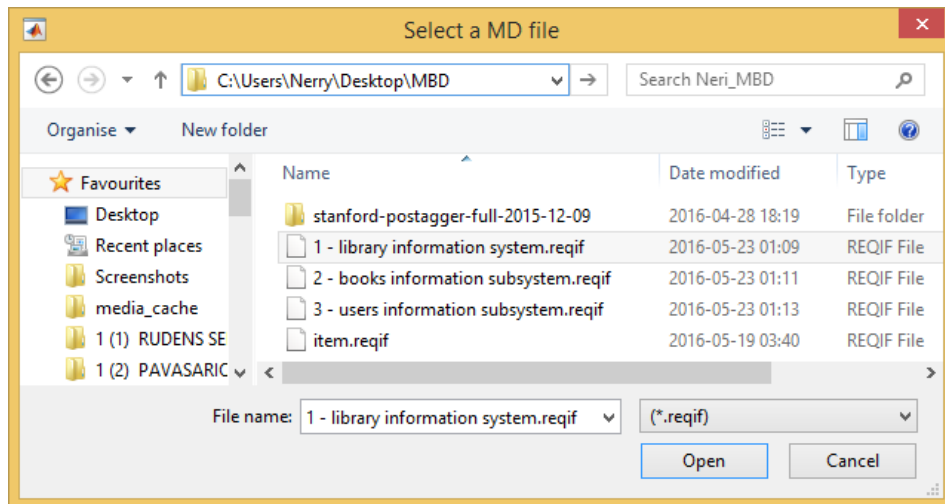
3.2.3. Reikalavimų modelio duomenų nuskaitymas ir atvaizdavimas

Prototipe realizuotas duomenų nuskaitymas ir gautų duomenų atvaizdavimas. Vartotojas gali nuskaityti eksportuoto reikalavimų modelio *.reqif* formatu elementus ir pasirinkti tuos, kuriuos nori analizuoti toliau. Reikalavimų modelis įkeliamas paspaudus „*Open MD file*“ mygtuką (25 pav.).



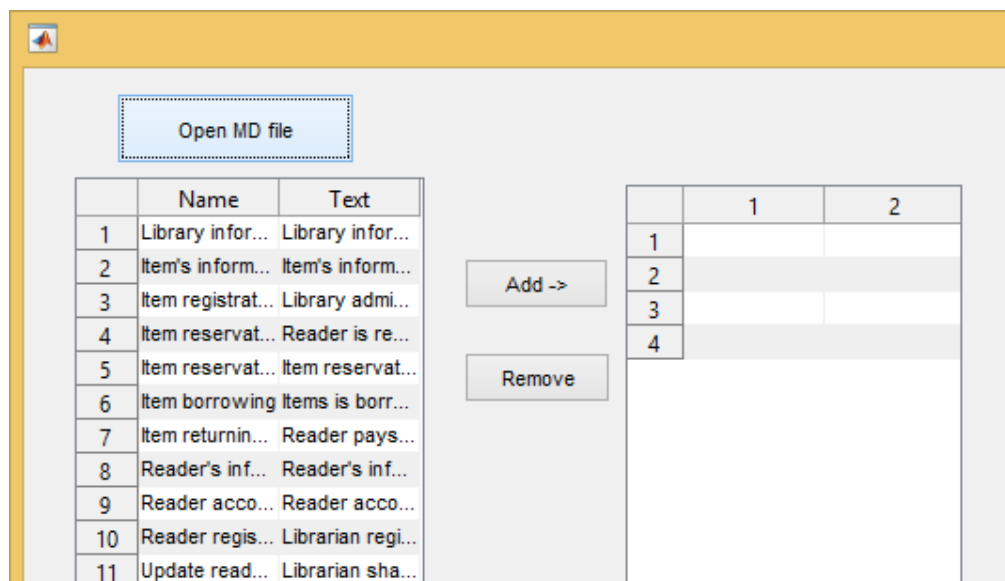
25 pav. Sistemos vartotojo sąsajos langas

Paspaudus reikalavimų modelio parinkimo mygtuką pateikiamas modelio parinkimo langas (26 pav.).



26 pav. Reikalavimų modelio parinkimas

Parinkus reikiamą eksportuotą modelį, kairėje lango pusėje atsiranda įkelto modelio reikalavimo tipo elementų sąrašas, pateikiamas jų pavadinimas ir tekstas (27 pav.).



27 pav. Reikalavimų modelio nuskaitymas

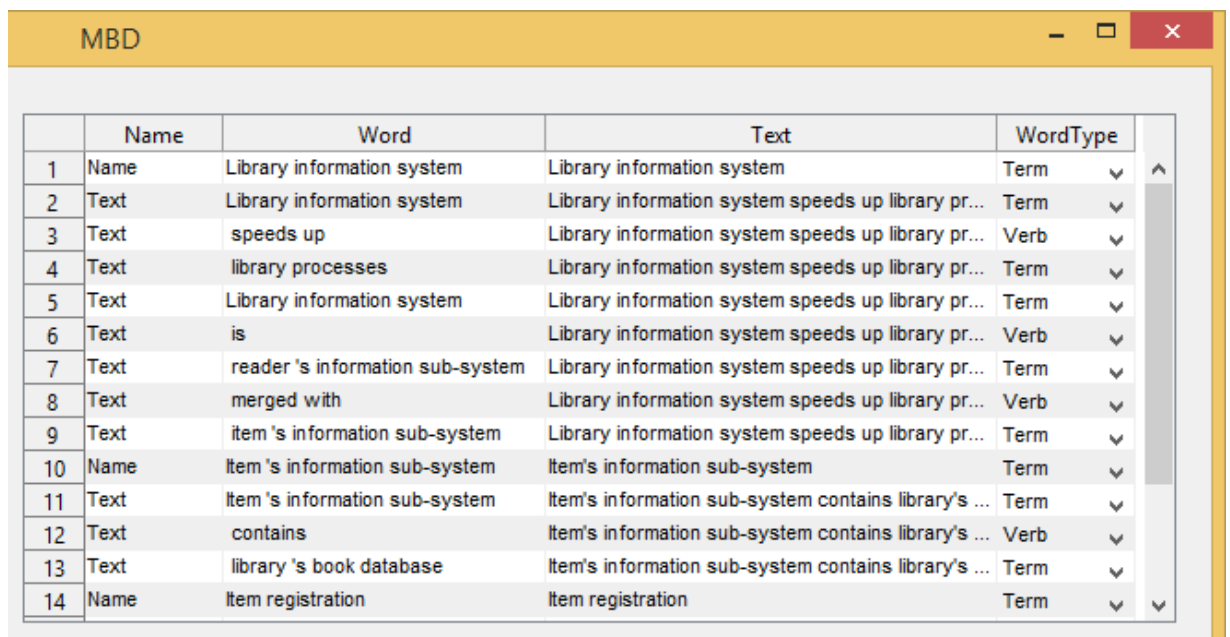
3.2.4. Reikalavimų modelio elementų sąrašo redagavimas ir papildymas

Prototipas automatiškai nuskaityto reikalavimo tipo elementus, esančius modelyje, ir pateikia iš jų gautą tekstinę informaciją. Vartotojas gali redaguoti atrinktų reikalavimų sąrašą spausdamas „Add - >“ arba „Remove“ mygtukus. Sąrašas pateikiamas pagal reikalavimo pavadinimą ir šalia pateikiamas nuskaitytas reikalavimo tekstas.

Jei reikia įsikelti dar vieną reikalavimų modelį ir reikalavimų sąrašą papildyti kitais elementais, vartotojas gali parinkti dar vieną modelį ir bus įkeliami jo elementai sąrašo gale. Toliau – eiga ta pati, kaip ir įkeliant pirmą kartą.

3.2.5. Veiklos žodyno sudarymas

Kuomet vartotojas pasirenka aktualius reikalavimų modelio elementus, sistema automatiškai sugeneruoja veiklos žodyno elementus: *name*, *term*, *verb*, ir *keyword*. Gautas veiklos žodyno elementų sąrašas pateikiamas dešinėje lango pusėje. Sudarytame veiklos žodyne nurodomas reikalavimo elemento tipas (pavadinimas arba tekstas), veiklos žodynui priskirti žodžiai, visas reikalavimo tekstas ir veiklos žodyno žodžio tipas (28 pav.).



	Name	Word	Text	WordType
1	Name	Library information system	Library information system	Term
2	Text	Library information system	Library information system speeds up library pr...	Term
3	Text	speeds up	Library information system speeds up library pr...	Verb
4	Text	library processes	Library information system speeds up library pr...	Term
5	Text	Library information system	Library information system speeds up library pr...	Term
6	Text	is	Library information system speeds up library pr...	Verb
7	Text	reader 's information sub-system	Library information system speeds up library pr...	Term
8	Text	merged with	Library information system speeds up library pr...	Verb
9	Text	item 's information sub-system	Library information system speeds up library pr...	Term
10	Name	Item 's information sub-system	Item's information sub-system	Term
11	Text	Item 's information sub-system	Item's information sub-system contains library's ...	Term
12	Text	contains	Item's information sub-system contains library's ...	Verb
13	Text	library 's book database	Item's information sub-system contains library's ...	Term
14	Name	Item registration	Item registration	Term

28 pav. Reikalavimų modelio elementų parinkimo langas

Jeigu pastebima sisteminė klaida sudarant veiklos žodyno elementus, prie kiekvieno veiklos žodyno elemento tipo, realizuotas iššokančiojo meniu vartotojo sąsajos elementas, skirtas veiklos žodyno elemento tipui redaguoti. Esant reikalui, tam pačiam lange veiklos žodyno lentelėje vartotojas gali koreguoti žodžiui priskirtą tipą. Visos korekcijos yra išsaugomos automatiškai - pakeistas elemento tipas yra automatiškai išsaugomas sistemoje.

3.2.6. Veiklos žodyno faktų sudarymas

Kuomet vartotojas pasitiktina visus veiklos žodyno elementus, gali sudaryti veiklos žodyno faktus spausdamas mygtuką „Generate Facts“. Tuomet iš jau sugeneruotų veiklos žodyno elementų pagal sakinio struktūrą sudaromi veiklos žodyno faktai. Sugeneruoti faktai ir jų tipai pateikiame dešinėje pusėje apatinėje lentelėje (29 pav.).

MBD

	Name	Word	Text	WordType
1	Name	Library information system	Library information system	Term
2	Text	Library information system	Library information system speeds up library pr...	Term
3	Text	speeds up	Library information system speeds up library pr...	Verb
4	Text	library processes	Library information system speeds up library pr...	Term
5	Text	Library information system	Library information system speeds up library pr...	Term
6	Text	is	Library information system speeds up library pr...	Verb
7	Text	reader 's information sub-system	Library information system speeds up library pr...	Term
8	Text	merged with	Library information system speeds up library pr...	Verb
9	Text	item 's information sub-system	Library information system speeds up library pr...	Term
10	Name	Item 's information sub-system	Item's information sub-system	Term
11	Text	Item 's information sub-system	Item's information sub-system contains library's ...	Term
12	Text	contains	Item's information sub-system contains library's ...	Verb
13	Text	library 's book database	Item's information sub-system contains library's ...	Term
14	Name	Item registration	Item registration	Term

Generate Facts

	Na...	Text	Fact	FactType
1	Text	Library infor...	Library information system speeds up library processes	2
2	Text	Library infor...	Library information system is reader 's information sub-system merged with ite...	4
3	Text	Library infor...	reader 's information sub-system merged with item 's information sub-system	2
4	Text	Item's inform...	Item 's information sub-system contains library 's book database	2
5	Text	Library admi...	Library administrator registers items	2

29 pav. Veiklos žodyno faktų sudarymas

3.3. Prototipo testavimas

Siekiant kuo tiksliau realizuoti sudarytą algoritmą, testavimas buvo vykdomas modeliuojant įvairius reikalavimų modelius ir keičiant reikalavimuose esančią informaciją. Kuomet pateikiamas teisingas reikalavimų modelis ir jame esanti informacija atitinka numatytus apribojimus, prototipas sėkmingai veikia. Toliau, testuojant buvo patikrinti galimi sudėtingi tekstinės informacijos atvejai.

Testavimo metu siekiama įvertinti suformuotų veiklos žodyno elementų kiekį ir tikslumą. Testavimo analizė ir rezultatai pateikiami toliau esančiame poskyryje, kur galima pastebėti, kad sistemos automatizuotu būdu suformuotame veiklos žodyne yra neatitikimų. Taip gali būti dėl kelių priežasčių:

1. Kai sakiniuose arba pavadinimuose panaudotos kalbos dalys, neatitinkančios veiklos žodyno elementų tipų;
2. Kai sakinio struktūra neatitinka veiklos žodyno faktų tipų sudarymui numatytos struktūros.

Taip pat, testavimo metu siekiama įvertinti suformuotų veiklos žodyno faktų kiekį ir tikslumą. Veiklos žodyno ir faktų elementai suformuoti iš tų pačių pavyzdžių. Testavimo metu patikrintas veiklos žodyno fakto tipų atitikimas. Fakto tipas yra nurodomas skaitmenini, kuris reiškia jį sudarančių *term/name* elementų kiekį. Toliau esančiame poskyryje galima pastebėti, kad sistemos automatizuotu būdu suformuoto veiklos žodyno faktų elementuose, taip pat, yra neatitikimų. Taip gali būti dėl ankstesniame žingsnyje blogai sudarytų veiklos žodyno elementų.

3.4. Prototipo eksperimentinis tyrimas kiekybiniais ir kokybiniais aspektais

Metodo ir prototipo pagrindinis tikslas iš reikalavimų modelio automatizuotu būdu išgauti veiklos apribojimus. Taigi, prototipo vertinimui pasirinkti kiekybiniai ir kokybiniai kriterijai, nustatantys, kad prototipas vartotojui yra naudingas:

1. Veiklos žodyno sudarytų elementų kiekis;
2. Veiklos žodyno sudarytų elementų teisingumas ir tikslumas.

Kokybės įvertinimui sudaryti reikalavimų modeliai su teisingais, ir su klaidingais duomenimis. Klaidingi duomenys – kuomet sakiniuose nesilaikoma numatytų apribojimų bei naudojamos papildomos kalbos dalys. Siekiant užtikrinti veiklos žodyno faktų teisingumą vartotojas pirmiausia turi peržiūrėti kitus sugeneruotus veiklos žodyno elementus ir atlikti reikalingas korekcijas.

Rezultatai – išsamus veiklos apribojimų sąrašas. Pateikiamas sugeneruotas veiklos žodynas kartu su reikalavimų modelio elementais, iš kurių jis buvo sugeneruotas. Kokybės užtikrinimui reikalavimų modelis turi atitikti realaus pasaulio reikalavimus.

Prototipo kokybės nustatymui ir įvertinimui atliekamas eksperimentinis prototipo tyrimas. Siekiant nustatyti kokybę sukurti trys tos pačios dalykinės srities testavimo atvejai:

- Reikalavimų modelis „*Library information system*“ (30 pav.);
- Reikalavimų modelis „*Book's information sub-system*“ (31 pav.);
- Reikalavimų modelis „*User information sub-system*“ (32 pav.).

Reikalavimų modeliuose naudojami skirtingo tipo elementai – „*requirement*“, „*extendedRequirement*“, „*interfaceRequirement*“. Reikalavimų informacija yra kaupiama reikalavimų pavadinimuose ir reikalavimų tekstuose.

Eksperimentiniu tyrimu vertinama veiklos apribojimų automatizuoto išgavimo iš reikalavimų modelio metodo kokybė atsižvelgiant į:

- Reikalavimų modelio elementų tipus ir jų kieki;
- Kiekvieno reikalavimo pavadinimą ir tekstą;
- Kiekvieno reikalavimo tekstinės informacijos formulavimą;
- Metode apibrėžtų apribojimų laikymąsi.

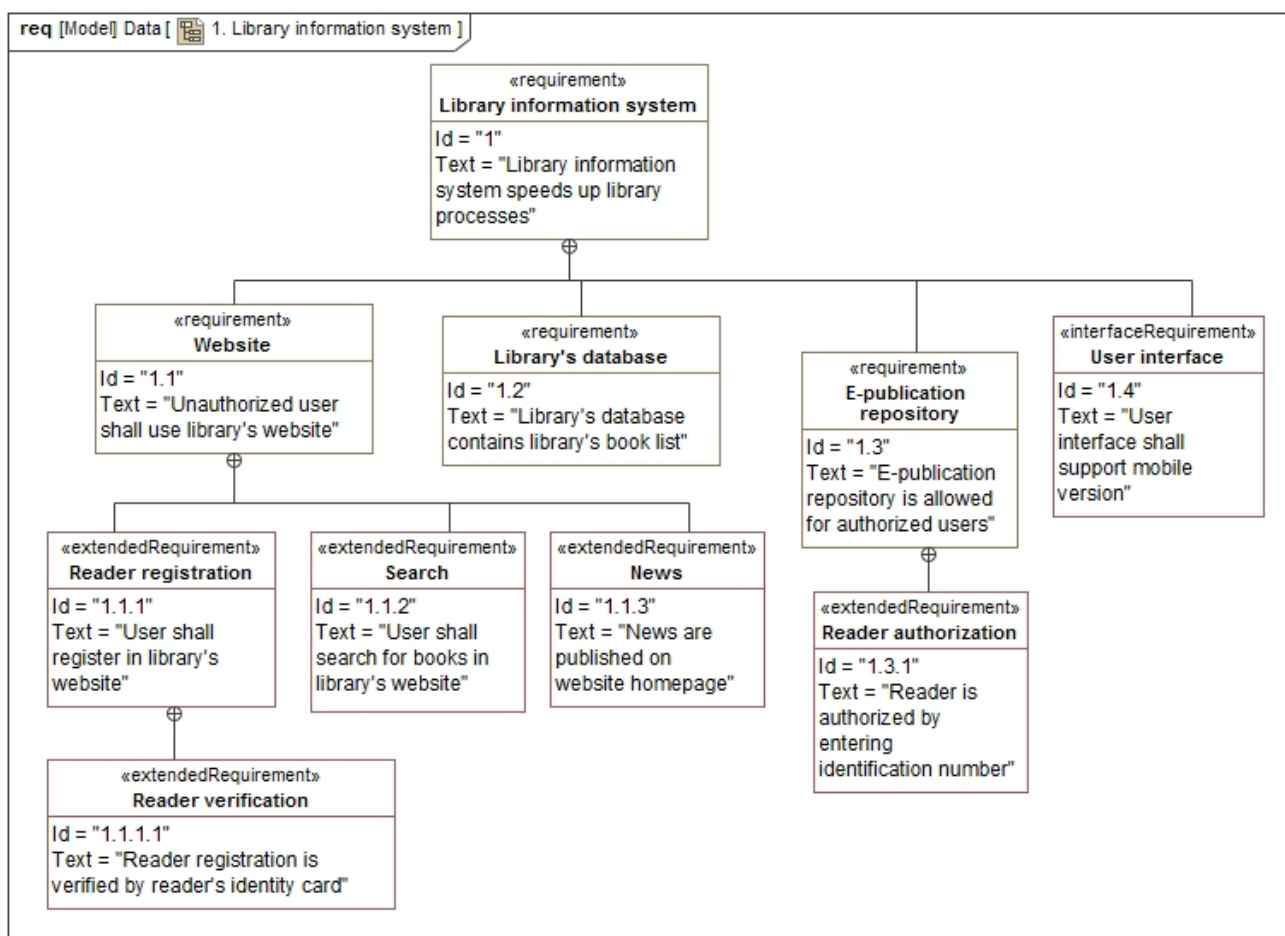
Sudarant reikalavimų modelius stengiamasi įvertinti, kad vartotojas ne visuomet laikosi numatytų apribojimų, tačiau sudaro gramatiškai pakankamai teisingus reikalavimus.

Pasinaudojant realizacijos funkcijomis nuskaityti ir išnagrinėti visų trijų modelių elementai atskirai ir pateikiami rezultatai. Gauti rezultatai pateikiami veiklos žodyno elementų lentelėmis ir veiklos žodyno faktų lentelėmis. Veiklos žodyno lentelėse pateikti elementai pagal reikalavimo informacijos tipą (reikalavimo pavadinimas arba tekstas), šalia pateikiamas pilnas reikalavimo dalies tekstas. Prie reikalavimo informacijos pateikiama sugeneruotas veiklos žodyno (VŽ) elementas ir jam priskirtas veiklos žodyno elemento tipas ir stulpelyje „atitikimas“ nurodoma, ar gautasis veiklos žodyno elementas kartu su tipu atitinka tikėtąjį elementą (+ reiškia atitinka, - reiškia neatitinka). „Atitikimo“ stulpelis pateikia rezultatus, pagal kuriuos galima nuspręsti, koku tikslumu sugeneruojami veiklos apribojimai iš reikalavimų modelio.

Atskirose lentelėse pateikiami faktų ir faktų tipų generavimo rezultatai. Taip pat, pateikiamas reikalavimo informacijos tipas (reikalavimo pavadinimas arba tekstas), šalia pilnas reikalavimo dalies tekstas, tuomet sugeneruotas faktas ir fakto tipas (1 – reiškia unarinis, 2 – reiškia binarinis, ir t.t.). Stulpelyje „atitikimas“ nurodoma, ar sugeneruotas faktas atitinka tikėtąjį faktą. Tokiu būdu galima nustatyti sugeneruotų veiklos apribojimų (faktų) tikslumą.

3.4.3. Reikalavimų modelio „Library information system“ analizė

Pateikiamas pirmasis nagrinėjamas reikalavimų modelis „Library information system“ (30 pav.).



30 pav. Reikalavimų diagrama „Library information system“

Toliau, pateikiama veiklos žodyno elementų analizės iš reikalavimų modelio analizės lentelė (3.1 lentelė) ir veiklos žodyno faktų analizės lentelė (3.2 lentelė).

3.4.1 lentelė Veiklos žodyno elementų analizė iš reikalavimų modelio „Library information system“

Tipas	Reikalavimo tekstas	VŽ elementas	VŽ elemento tipas	Atitikimas
Name	Library information system	Library information system	Term	+
Text	Library information system speeds up library processes.	Library information system	Term	+
		speeds up	Verb	+
		library processes	Term	+
Name	E-publication repository	E-publication repository	Term	+
Text	E-publication repository is allowed for authorized users.	E-publication repository	Term	+
		is allowed for	Verb	+
		authorized	Verb	+
		users	Term	+
Name	Reader authorization	Reader authorization	Term	+
Text	Reader is authorized by entering identification number.	Reader	Term	+
		is authorized by	Verb	+
		entering	Verb	+
		identification number	Term	+
Name	Library's database	Library 's database	Term	+
Text	Library's database contains library's book list.	Library 's database	Term	+
		contains library 's book	Verb	+
		list .	Term	+
Name	User interface	User interface	Term	+
Text	User interface shall support mobile version.	User interface	Term	+
		shall	Modal Keyword	+
		support	Verb	+
		mobile	[]	-
		version	Term	+
Name	Website	Website	Term	+
Text	Unauthorized user shall use library's website.	Unauthorized	[]	-
		user	Term	+
		shall	Modal Keyword	+
		use	Verb	+
		library 's website	Term	+
Name	News	News	Term	+
Text	News are published on website homepage.	News	Term	+
		are published on	Verb	+
		website homepage	Term	+
Name	Reader registration	Reader registration	Term	+
Text	User shall register in library's website.	User	Term	+
		shall	Modal Keyword	+
		register in	Verb	+
		library 's website	Term	+
Name	Reader verification	Reader verification	Term	+
Text		Reader registration	Term	+

	Reader registration is verified by reader's identity card.	is verified by	Verb	+
		reader 's identity card	Term	+
Name Search		Search	Verb	+
		User	Term	+
		shall	Modal Keyword	+
Text	User shall search for books in library's website.	search for	Verb	+
		books	Term	+
		in	[]	-
		library 's website	Term	+

3.1 lentelėje matoma, kad automatinio būdu suformuotame veiklos žodyne yra trys klaidos:

- veiklos žodynui nepriskirti žodžiai „mobile“ ir „unauthorized“;
- elementui „in“ nepriskirtas *keyword* elemento tipas.

Veiklos žodynui nepriskiriami žodžiai, kurie sakinyje vartojami vietoj būdvardžio – vartotojas nesilaikė reikalavimų specifikavimo apribojimų, todėl yra gautos klaidos automatinio būdu generuojant veiklos žodyną. „In“ nepriskyrimas *keyword* veiklos žodyno elementų tipui – sisteminė klaida, ją vartotojas turėtų ištaisyti prieš generuodamas veiklos žodyno faktus.

3.4.2 lentelė Veiklos žodyno elementų analizė iš reikalavimų modelio „Library information system“

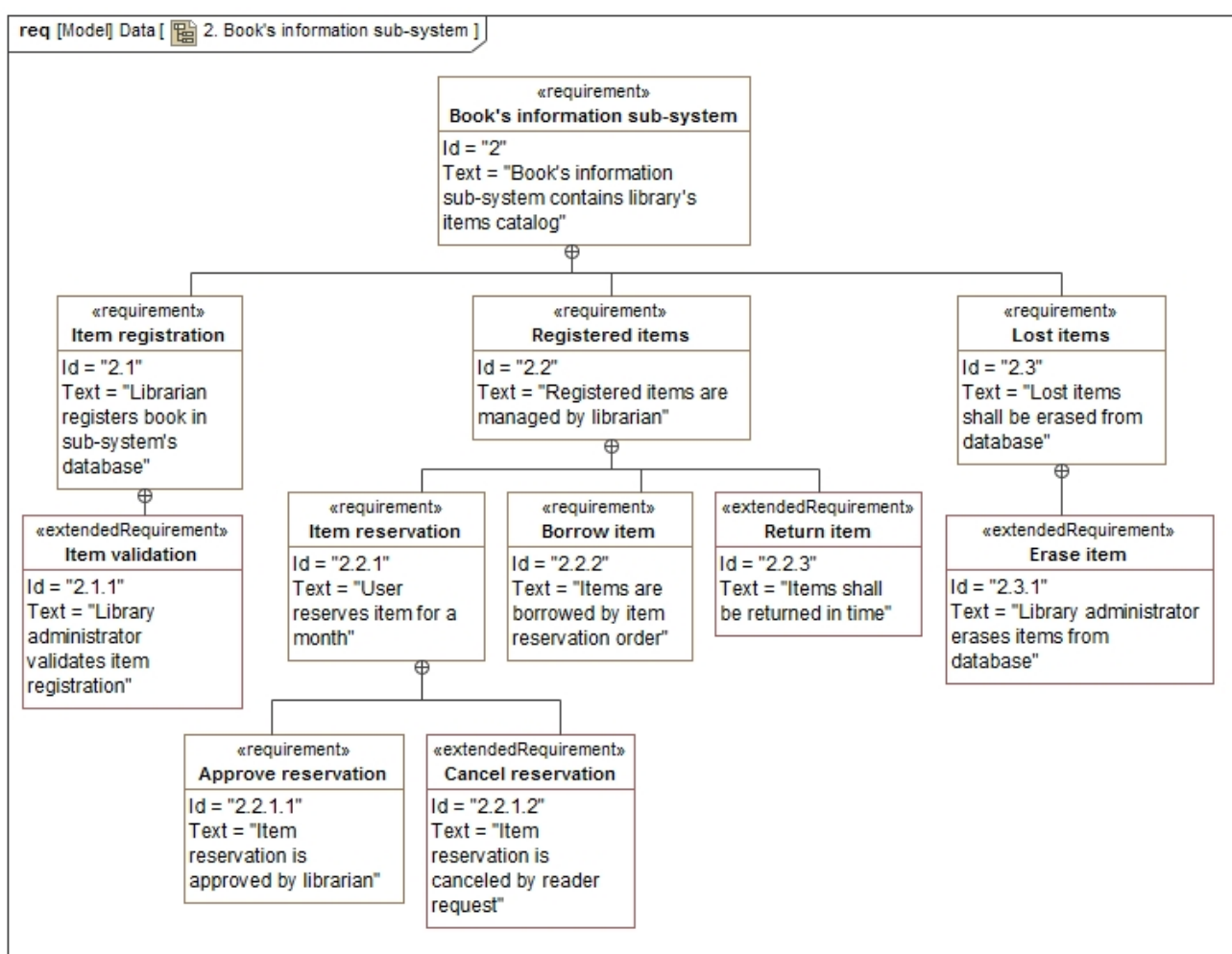
Tipas	Reikalavimo tekstas	VŽ Faktas	Fakto tipas	Atitikimas
Text	Library information system speeds up library processes.	Library information system speeds up library processes	2	+
Text	E-publication repository is allowed for authorized users.	E-publication repository is allowed for	1	-
Text	Reader is authorized by entering identification number.	Reader is authorized by	1	-
Text	Library's database contains library's book list.	Library 's database contains library 's book list .	2	+
Text	User interface shall support mobile version.	User interface shall support	1	-
Text	Unauthorized user shall use library's website.	user shall use library 's website	2	+
Text	News are published on website homepage.	News are published on website homepage	2	+
Text	User shall register in library's website.	User shall register in library 's website	2	+
Text	Reader registration is verified by reader's identity card.	Reader registration is verified by reader 's identity card	2	+

Text	User shall search for books in library's website.	User shall search for books	2	+
------	---	-----------------------------	---	---

3.2 lentelėje matoma, kad automatiniu būdu suformuotame veiklos žodyno faktų sąrašė yra keturios klaidos – išgauti ne pilni faktai, todėl neteisingai nustatyti ir faktų tipai. Iš esmės, klaidos susidarė todėl, kad faktai buvo generuojami iš nekoreguoto veiklos žodyno.

3.4.4. Reikalavimų modelio „Book’s information sub-system“ analizė

Pateikiamas pirmasis nagrinėjamas reikalavimų modelis „Book’s information sub-system“ (31 pav.).



31 pav. Reikalavimų diagrama „Book’s information sub-system“

Toliau, pateikiama veiklos žodyno elementų analizės iš reikalavimų modelio analizės lentelė (3.3 lentelė) ir veiklos žodyno faktų analizės lentelė (3.4 lentelė).

3.4.3 lentelė Veiklos žodyno elementų analizė iš reikalavimų modelio „Book's information sub-system“

Tipas	Reikalavimo tekstas	VŽ elementas	VŽ elemento tipas	Atitikimas
Name	Book's information sub-system	Book 's information sub-system	Term	+
Text	Book's information sub-system contains library's items catalog.	Book 's information sub-system	Term	+
		contains	Verb	+
		library 's items catalog	Term	+
Name	Item registration	Item registration	Term	+
Text	Librarian registers book in sub-system's database.	Librarian	Term	+
		registers	Verb	+
		book	Term	+
		in	[]	-
		sub-system 's database	Term	+
Name	Item validation	Item validation	Term	+
Text	Library administrator validates item registration.	Library administrator	Term	+
		validates	Verb	+
		item registration	Term	+
Name	Lost items	Lost	Verb	-
		items	Term	+
		Lost	Verb	-
		items	Term	+
Text	Lost items shall be erased from database.	shall	Modal	+
		be erased from	Verb	+
		database	Term	+
		Erase	Verb	+
Name	Erase item	item	Term	+
Text	Library administrator erases items from database.	Library administrator	Term	+
		erases	Verb	+
		items	Term	+
		from	[]	-
		database	Term	+
Name	Registered items	Registered items	Term	+
Text	Registered items are managed by librarian.	Registered items	Term	+
		are managed by	Verb	+
		librarian	Term	+
Name	Borrow item	Borrow	Verb	+
Text	Items are borrowed by item reservation order.	item	Term	+
		Items	Term	+
		are borrowed by	Verb	Verb
		item reservation order	Term	+

Name	Item reservation	Item reservation	Term	+
		User reserves item	Term	+
Text	User reserves item for a month.	for	[]	-
		a	Keyword	+
		month	Term	+
Name	Approve reservation	Approve	Verb	+
		reservation	Term	+
Text	Item reservation is approved by librarian.	Item reservation	Term	+
		is approved by	Verb	+
		librarian	Term	+
Name	Cancel reservation	Cancel	Verb	+
		reservation	Term	+
Text	Item reservation is canceled by reader request.	Item reservation	Term	+
		is canceled by	Verb	+
		reader request	Term	+
Name	Return item	Return item	Term	+
		Items	Term	+
Text	Items shall be returned in time.	shall	Modal Keyword	+
		be returned in	Verb	+
		time	Term	+

3.3 lentelėje matoma, kad automatiniu būdu suformuotame veiklos žodyne yra klaidos:

- veiklos žodynui nepriskirtas žodis „*lost*“;
- elementams „*in*“ ir „*from*“ nepriskirti *keyword* elemento tipai;
- elementas „*return item*“ priskirtas *term* elemento tipui.

Veiklos žodynui nepriskiriami žodžiai, kurie sakinyje vartojami vietoj būdvardžio – vartotojas nesilaikė reikalavimų specifikavimo apribojimų, todėl yra gauta klaida automatiniu būdu generuojant veiklos žodyną. „*In*“ ir „*from*“ nepriskyrimas *keyword* veiklos žodyno elementų tipui – sisteminė klaida, ją vartotojas turėtų ištaisyti prieš generuodamas veiklos žodyno faktus. „*Return item*“ priskirtas *term* elemento tipui, dėl to, kad galimai *Stanford POS Tagger* klaidingai atpažino kalbos dalį žodžiui „*return*“.

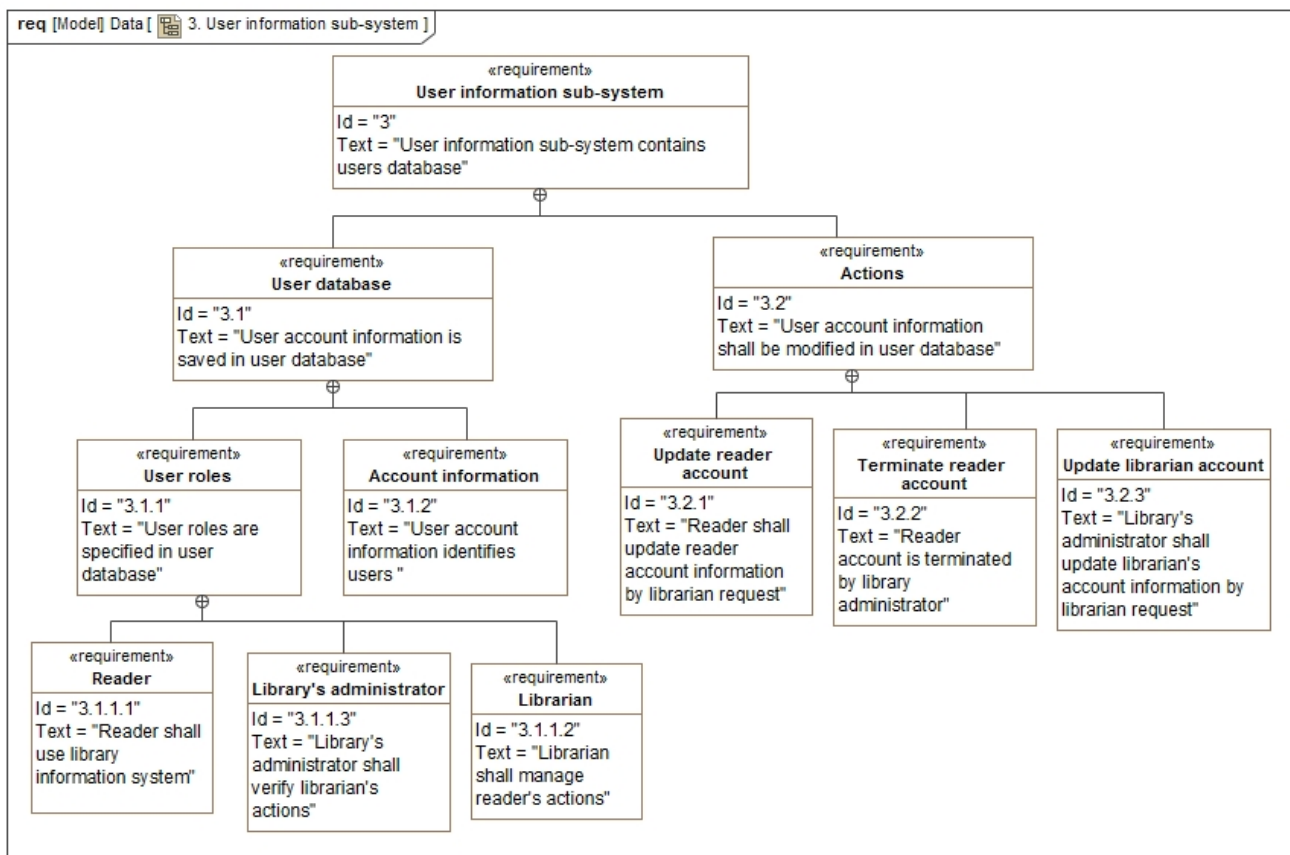
3.4.4 lentelė Veiklos žodyno elementų analizė iš reikalavimų modelio „*Book's information sub-system*“

Tipas	Reikalavimo tekstas	VŽ Faktas	Fakto tipas	Atitikimas
Text	Book's information sub-system contains library's items catalog.	Book 's information sub-system contains library 's items catalog	2	+
Text	Librarian registers book in sub-system's database.	Librarian registers book	2	-
Text	Library administrator validates item registration.	Library administrator validates item registration	2	+
Text	Lost items shall be erased from database.	items shall be erased from database	2	+
Text	Library administrator erases items from database.	Library administrator erases items	2	-
Text	Registered items are managed by librarian.	Registered items are managed by librarian	2	+
Text	Items are borrowed by item reservation order.	Items are borrowed by item reservation order	2	+
Text	Item reservation is approved by librarian.	Item reservation is approved by librarian	2	+
Text	Item reservation is canceled by reader request.	Item reservation is canceled by reader request	2	+
Text	Items shall be returned in time.	Items shall be returned in time	2	+

3.4 lentelėje matoma, kad automatiniu būdu suformuotame veiklos žodyno faktų sąrašė yra dvi klaidos – išgauti ne pilni faktai, todėl neteisingai nustatyti ir faktų tipai. Iš esmės, klaidos susidarė todėl, kad faktai buvo generuojami iš nekoreguoto veiklos žodyno.

3.4.5. Reikalavimų modelio „User information sub-system“ analizė

Pateikiamas pirmasis nagrinėjamas reikalavimų modelis „User information sub-system“ (32 pav.).



32 pav. Reikalavimų diagrama „User information sub-system“

Toliau, pateikiama veiklos žodyno elementų analizės iš reikalavimų modelio analizės lentelė (3.5 lentelė) ir veiklos žodyno faktų analizės lentelė (3.6 lentelė).

3.4.5 lentelė Veiklos žodyno elementų analizė iš reikalavimų modelio „User information sub-system“

Tipas	Reikalavimo tekstas	VŽ elementas	VŽ elemento tipas	Atitikimas
Name	User information sub-system	User information sub-system	Term	+
Text	User information sub-system contains users database.	User information sub-system	Term	+
		contains	Verb	+
		users database	Term	+
Name	Actions	Actions	Term	+
Text	User account information shall be modified in user database.	User account information	Term	+
		shall	Modal Keyword	+
		be modified in user database	Verb Term	+ +
Name	Terminate reader account	Terminate	Verb	+
Text	Reader account is terminated by library administrator.	reader account	Term	+
		Reader account is terminated by library administrator	Term Verb	+ +
		library administrator	Term	+
Name	Update librarian account	Update librarian account	Term	+
Text	Library's administrator shall update librarian's account information by librarian request.	Library's administrator	Term	+
		shall	Modal Keyword	+
		update	Verb	+
		librarian's account information	Term	+
		by	[]	-
Name	Update reader account	librarian request	Term	+
		Update reader account	Term	+
		Reader	Term	+
Text	Reader shall update reader account information by librarian request.	shall	Modal Keyword	+
		update	Verb	+
		reader account information	Term	+
		by	Term	-
		librarian request	Term	+
Name	User database	User database	Term	+
Text	User account information is saved in user database.	User account information	Term	+
		is saved in	Verb	+
		user database	Term	+
Name	Account information	Account information	Term	+

Tipas	Reikalavimo tekstas	VŽ elementas	VŽ elemento tipas	Atitikimas
Text	User account information identifies users.	User account information	Term	+
		identifies	Verb	+
		users	Term	+
Name	User roles	User roles	Term	+
		User roles are specified in user database.	Term	+
Text	User roles are specified in user database.	are specified in	Verb	+
		user database	Term	+
		Librarian	Term	+
Name	Librarian	Librarian	Term	+
		Librarian	Term	+
Text	Librarian shall manage reader's actions.	shall	Modal Keyword	+
		manage	Verb	+
		reader 's actions	Term	+
Name	Library's administrator	Library 's administrator	Term	+
		Library 's administrator	Term	+
Text	Library's administrator shall verify librarian's actions.	shall	Modal Keyword	+
		verify	Verb	+
		librarian 's actions	Term	+
Name	Reader	Reader	Term	+
		Reader	Term	+
Text	Reader shall use library information system.	shall	Modal Keyword	+
		use	Verb	+
		library information system	Term	+

3.5 lentelėje matoma, kad automatiniu būdu suformuotame veiklos žodyne yra klaidos:

- elementas „by“ nepriskirtas *keyword* elemento tipai;
- elementas „by“ priskirtas *term* elemento tipui.

Veiklos žodynui nepriskirtas žodis „by“ veiklos žodyno elementų tipui *keyword*– sisteminė klaida, ją vartotojas turėtų ištaisyti prieš generuodamas veiklos žodyno faktus. Antroji klaida – žodis „by“ priskirtas *term* elemento tipui, dėl to, kad galimai *Stanford POS Tagger* klaidingai atpažino kalbos dalį šiam žodžiui.

3.4.6 lentelė Veiklos žodyno elementų analizė iš reikalavimų modelio „User information sub-system“

Tipas	Reikalavimo tekstas	VŽ Faktas	Fakto tipas	Atitikimas
Text	User information sub-system contains users database.	User information sub-system contains users database	2	+
Text	User account information shall be modified in user database.	User account information shall be modified in user database	2	+
Text	Reader account is terminated by library administrator.	Reader account is terminated by library administrator	2	+
Text	Library's administrator shall update librarian's account information by librarian request.	Library 's administrator shall update librarian 's account information	2	+
Text	Reader shall update reader account information by librarian request.	Reader shall update reader account information	2	+
Text	User account information is saved in user database.	User account information is saved in user database	2	+
Text	User account information identifies users.	User account information identifies users	2	+
Text	User roles are specified in user database.	User roles are specified in user database	2	+
Text	Librarian shall manage reader's actions.	Librarian shall manage reader 's actions	2	+
Text	Library's administrator shall verify librarian's actions.	Library 's administrator shall verify librarian 's actions	2	+
Text	Reader shall use library information system.	Reader shall use library information system	2	+

3.6 lentelėje matoma, kad automatiniu būdu suformuotame veiklos žodyno faktų sąrašė visi faktai ir jų tipai nustatyti be klaidų. Taip yra todėl, kad sugeneruotame veiklos žodyne nebuvo esminių klaidų, kliudančių teisingai išgauti faktus.

3.4.6. Apibendrinti eksperimento rezultatai

Pateikiama apibendrinta eksperimento rezultatų analizės lentelė (3.7 lentelė).

3.4.7 lentelė Apibendrinti eksperimento rezultatai

Modelis VŽ elementas	„Library information system“	„Book's information sub-system“	„User information sub-system“
<i>Term</i>	30/30	35/35	35/35
<i>Verb</i>	13/13	14/14	12/12
<i>Modal Keyword</i>	4/4	2/2	6/6
<i>Keyword</i>	0/1	1/4	0/2
<i>Fact types</i>	7/10	8/10	10/10

3.7 lentelėje pateikiama statistinė suvestinė, kiek teisingai gauta tam tikrų veiklos žodyno elementų ir, kiek buvo tikimasi kiekvieno reikalavimų modelio atveju.

Fact types skiltyje pateikiamas apibendrintas faktų tipų kiekis ir kiek iš viso jų buvo sugeneruota iš pasirinktų eksperimento atvejų. Eksperimento atvejais dažniausiai pasitaikantys faktų tipai yra unarinis ir binarinis. Be to, akivaizdu, kad fakto tipai teisingiausiai sugeneruoti trečiojo reikalavimų modelio „User information sub-system“ atveju. Taip yra todėl, kad kokybiškas faktų formavimas iš veiklos žodyno elementų yra priklausomas nuo kokybiško veiklos žodyno elementų generavimo, ypač, nuo teisingai identifikuoto raktažodžio sakinyje.

3.5. Prototipo kūrimo ir testavimo rezultatai

1. Prototipo kūrimas apibrėžtas keturiais pagrindiniais etapais, identifikuotais pagal prototipo funkcionalumą:
 - Reikalavimų modelio duomenų nuskaitymas ir atvaizdavimas;
 - Reikalavimų modelio elementų sąrašo redagavimas ir papildymas;
 - Veiklos žodyno elementų sudarymas ir redagavimas;
 - Veiklos žodyno faktų elementų sudarymas.
2. Prototipo testavimo metu vertinama realizuoto veiklos apribojimų automatinio išgavimo iš reikalavimų modelio kokybė parenkant skirtingus testavimo atvejų duomenis. Kadangi veiklos žodyno sudarymui svarbi reikalavimų sakinio struktūra, testavimui naudojami teisingi ir klaidingi įeigos duomenys.
3. Iš eksperimento rezultatų galima daryti išvadą, kad veiklos žodyno elementų teisingumas priklauso nuo sakinio struktūros ir papildomų kalbos dalių įtraukimo į sakinius. Taip pat, priklauso nuo apribojimų reikalavimų modeliui laikymosi ir vartotojo anglų kalbos gramatikos išmanymo.
4. Kokybiškas faktų formavimas iš veiklos žodyno elementų yra priklausomas nuo kokybiško veiklos žodyno elementų generavimo, ypač, nuo teisingai identifikuoto raktažodžio sakinyje.

4. IŠVADOS

1. Išanalizuotas reikalavimų modeliavimas, reikalavimų struktūros ir jų specifikavimo šablonai, palyginti *SRS* ir *Volere* reikalavimų specifikavimo šablonai. Išanalizuotos *SysML* modeliavimo kalbos galimybės bei *i** ir *KAOS* meta-modeliai. Atlikus analizę nustatyta, kad reikalavimų modelis geriausiai pateikiamas *SysML* modeliavimo kalba. Išanalizuotos ir palygintos veiklos modeliavimo *ArchiMate*, *UML* ir *BPMN* kalbos. Atlikus analizę nustatyta, kad konkrečiu atveju, siekiant išgauti veiklos apribojimus ir norint juos panaudoti veiklos modeliavimui, geriausia yra rinktis *UML* veiklos modeliavimo kalbą.

2. Sudarytas veiklos apribojimų automatizuoto formavimo iš reikalavimų modelio metodas. Metodo pagrindą sudaro detalios veiklos diagramos skirtingų elementų tekstų analizavimui, aprašančios, iš *SysML* reikalavimų modelių veiklos žodyno ir veiklos taisyklių sąrašo formavimą bei, naudojantis *SBVR* terminologija natūralia kalba leidžiančios jį pateikti vartotojui.

3. Realizuotas prototipas automatinio veiklos žodyno elementų formavimo algoritmas, kurio pagrindas yra reikalavimų modelio elementų tekstinės informacijos nuskaitymas, elementų pavadinimų ir tekstų nagrinėjimas pagal sudarytus algoritmus atsižvelgiant į anglų kalbos taisykles, reikalavimų modelio sudarymo apribojimus ir rekomendacijas. Išgauta veiklos apribojimų informacija pateikiama lentelių formoje ir sudaryta galimybė vartotojui rankiniu būdu koreguoti rezultatus.

4. Atlikus eksperimentinį prototipo tyrimą, nustatyta, kad automatinis veiklos žodyno išgavimas pagreitina veiklos žodyno elementų generavimą ir užtikrina klaidų eliminavimą. Pagrindinis realizuoto metodo rezultatas yra suformuoti veiklos žodyno elementai – pavadinimai, terminai, veiksmožodžiai, raktažodžiai, faktai ir nustatyti faktų tipai. Sudarytas veiklos žodynas gali būti panaudojamas organizacijos veiklos apribojimams specifikuoti sudarant organizacijos veiklos modelius.

5. LITERATŪRA

- [1] SysML specification, OMG group, 2007, [Kreiptasi 2015-01-17]. Prieiga per internetą: <http://www.omg.org/spec/SysML/1.0/>
- [2] Volere requirement specification, [Kreiptasi 2015-01-04]. Prieiga per internetą: <http://www.volere.co.uk/template.htm>
- [3] BPMN specification, OMG group, 2008, [Kreiptasi 2015-01-17]. Prieiga per internetą: <http://www.omg.org/spec/BPMN/1.1/>
- [4] OCEB certification guide: business process management, fundamental level Tim Weilkiens, Christian Weiss, Andrea Grass, 2011
- [5] Essentials of IBM Rational RequisitePro v7.0, 2006, Student Workbook.
- [6] Requirements Specification and Modeling through SysML, Michel dos Santos Soares, Jos Vrancken, 2007.
- [7] A goal-oriented requirements modeling language for enterprise architecture, Dick Quartel, Wilco Engelsman, Henk Jonkers, Marten van Sinderen, 2009
- [8] ArchiMate, The Open Group. Version 1. [Kreiptasi 2015-01-20]. Prieiga per internetą: <http://www.opengroup.org/archimate>
- [9] How Does Requirements Quality Relate to Project Success or Failure? Mayumi Itakura Kamata, Tetsuo Tamai. 2007
- [10] IEEE Recommended Practice for Software Requirements Specifications, IEEE Computer Society. 1998 October
- [11] Volere Requirements Specification Template, Edition 15. March, 2010
- [12] Exploring Intentional Modeling and Analysis for Enterprise Architecture. E. Yu, M. Strohmaier, and X. Deng. Proceedings of the EDOC 2006 Workshop on Trends in Enterprise Architecture Research (TEAR 2006). 2006.
- [13] Bach, J.S., "DOORS : A Tool Manage Requirements", [Kreiptasi 2015-01-17]. Prieiga per internetą: <http://www.springerlink.com/content/g763qw2t17366455/>
- [14] Semantics of Business Vocabulary and Business Rules (SBVR), Second SBVR Interim Specification without change bars, dtc/06-08-05 <http://www.omg.org/spec/SBVR/1.0/PDF/>
- [15] Visual SBVR Musham, P.; Singh, S.; Bahal, R.; Ty, P. Digital Information Management, 2008. ICDIM 2008. Third International Conference on DOI; 2008, p. 676 – 683
- [16] Using UML Part Two – Behavioral Modeling Diagrams. Sparx Systems, 2007, p. 5-15