



**Kauno technologijos universitetas**

Informatikos fakultetas

# **Veiklos žodynų ir taisyklių neprieštaringumo tikrinimo metodas**

Baigiamasis magistro studijų projektas

---

**Deividas Smailys**

Projekto autorius

**Lekt. Algirdas Šukys**

Vadovas

---

**Kaunas, 2020**



**Kauno technologijos universitetas**

Informatikos fakultetas

# **Veiklos žodynų ir taisyklių neprieštarinimumo tikrinimo metodas**

Baigiamasis magistro studijų projektas

Informacinių sistemų inžinerija (6211BX009)

---

**Deividas Smailys**

Projekto autorius

**Lekt. Algirdas Šukys**

Vadovas

**Doc. dr. Vytautas Rudžionis**

Recenzentas

---

**Kaunas, 2020**



**Kauno technologijos universitetas**

Informatikos fakultetas

Deividas Smailys

## **Veiklos žodynų ir taisyklių neprieštaringumo tikrinimo metodas**

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Deivido Smailio, baigiamasis projektas tema „Veiklos žodynų ir taisyklių neprieštaringumo tikrinimo metodas“ 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)



**Kauno technologijos universitetas**

Informatikos fakultetas

## **Baigiamojo magistro projekto užduotis**

Projekto tema	<u>Veiklos žodynų ir taisyklių neprieštaringumo tikrinimo metodas</u>
Reikalavimai ir sąlygos (tikslinti pavadinimą pagal poreikį)	Tikrinami SBVR veiklos žodynų ir taisyklių neprieštaringumai ir pateikiamas metodas prieštarigumų aptikimui.
Vadovas / Vadovė	<u>Lekt. Algirdas Šukys</u> (vadovo pareigos, vardas, pavardė, parašas) (data)

Smailys, Deividas. Veiklos Žodynų ir Taisyklių Neprieštaringumo Tikrinimo Metodus. Magistro baigiamasis projektas / vadovas lekt. Algirdas Šukys; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informacijos sistemos, Informatikos mokslai.

Reikšminiai žodžiai: *SBVR*, *prieštaringumas*.

Kaunas, 2020. 100 p.

## **Santrauka**

Informacinės sistemos yra kuriamos siekiant palengvinti ir pagreitinti atliekamą darbą tam tikroje veiklos srityje. Kuriamos sistemos privalo tenkinti užsakovų keliamus reikalavimus, tačiau dažniausiai ne visi reikalavimai yra žinomi sistemos kūrimo proceso pradžioje. Taip nutinka dėl komunikacijos trūkumo tarp užsakovų ir informacinių sistemų kūrėjų. Siekiant užtikrinti sklandesnę bendravimo procesą tarp užsakovų ir sistemų kūrėjų yra naudojami veiklos žodynai ir taisyklės, aprašytos natūralia kalba. *SBVR* yra standartas, kuris leidžia veiklos žodyną ir taisykles aprašyti natūralia struktūrizuota kalba. Esamų sprendimų analizėje nustatyta, kad nėra realizuotų automatizuotų įrankių, kurie turėtų galimybę nustatyti aprašytų veiklos žodynų ir taisyklių prieštaringumus ir neapribotų *SBVR* standarto išraiškingumo. Siekiant išvengti žmogiškojo faktoriaus klaidų aprašinėjant veiklos žodyną ir taisykles, buvo sukurtas veiklos žodynų ir taisyklių neprieštaringumo tikrinimo metodas, kuris vėliau buvo realizuotas kaip neprieštaringumo tikrinimo metodo prototipas, leidžiantis aptikti tam tikras žodyno įrašų ir taisyklių aprašymo klaidas ir loginius prieštaringumus *SBVR* specifikacijose. Prototipas buvo realizuotas naudojantis šiame darbe aprašytais prieštaringumų aptikimo algoritmais. Sukurto metodo prototipo testavimo ir eksperimento rezultatai parodė, kad gana korektiškai yra atpažįstamos klaidos ir prieštaringumai *SBVR* specifikacijose. Tolimesniam sistemos vystymui rekomenduojama papildyti prieštaringumų aptikimą algoritmais, kurie sugeba atsekti konceptų hierarchijas ir jas panaudoti aptinkant konceptų sąryšius.

Smailys, Deividas. Method for Checking Consistency of Business Vocabularies and Rules. Master's Final Degree Project / supervisor lec. Algirdas Šukys; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Information Systems, Computing.

Keywords: *SBVR, consistency*.

Kaunas, 2020. 100 p.

### **Summary**

Information systems are designed to facilitate and speed up work within an area of activity. The developed systems must meet the requirements of the customer, but usually not all requirements are known at the beginning of the system development process. This happens due to the lack of communication between customers and information system developers. Business vocabularies and rules, expressed in natural language, are used to ensure a smoother communication process between customers and system developers. SBVR is a standard that allows describing business vocabulary and rules using structured natural language. The analysis of the existing solutions revealed that there are no automated tools that would be able to identify errors and contradictions in the business vocabulary and rules and would not limit the expressiveness of the SBVR standard. To avoid human factor errors in describing the business vocabulary and rules, a new method for contradiction detection was developed which later was implemented as a method prototype for error and contradiction checking within SBVR specifications. The prototype was implemented using contradiction detection algorithms, described in this work. The results of testing and experimentation of the prototype show that errors and contradictions in the SBVR specifications are mostly correctly identified. For future development of the system, it is recommended to supplement the detection of contradictions with algorithms that can trace the hierarchies of concepts and use them to detect the relationships of the concepts.

## Turinys

<b>Lentelių sąrašas .....</b>	<b>9</b>
<b>Paveikslų sąrašas .....</b>	<b>10</b>
<b>Santrumpų ir terminų sąrašas .....</b>	<b>12</b>
<b>Įvadas.....</b>	<b>13</b>
<b>1. Probleminės srities analizė.....</b>	<b>15</b>
<b>1.1. Analizės tikslas.....</b>	<b>15</b>
<b>1.2. Tyrimo objektas, sritis ir problema .....</b>	<b>15</b>
<b>1.3. Prieštaringumų ir klaidų tikrinimo veiklos žodynų ir taisyklių specifikacijose analizė.....</b>	<b>15</b>
1.3.1. Informacinių sistemų kūrimo procesas.....	15
1.3.2. SBVR standartas.....	16
1.3.3. SBVR veiklos žodynų ir taisyklių specifikacija .....	19
1.3.4. Prieštaringumas .....	32
1.3.5. Prieštaringumai SBVR specifikacijose.....	33
1.3.6. Išskirti žodyno įrašų prieštarigumų tipai .....	33
1.3.7. Išskirti SBVR taisyklių prieštarigumo tipai.....	37
<b>1.4. Veiklos žodynų ir taisyklių neprieštaringumo tikrinimo naudotojų analizė .....</b>	<b>40</b>
<b>1.5. Esamų SBVR prieštarigumo tikrinimo metodų analizė.....</b>	<b>41</b>
1.5.1. Nuo veiklos srities nepriklausantis metodas prieštarigumų aptikimui SBVR paremtose veiklos taisyklėse [18] .....	41
1.5.2. Eksperimentinis transformacijų iš SBVR veiklos žodynų ir taisyklių į OWL2 ontologijas tyrimas [22] .....	41
1.5.3. „Logika paremtas sprendimas“ SBVR neprieštaringumo užtikrinimui [27].....	42
1.5.4. Veiklos taisyklių neprieštaringumas panaudojant Alloy [29].....	42
1.5.5. SBVR prieštarigumo patikrinimo metodų palyginimas .....	42
<b>1.6. Darbo tikslas, uždaviniai ir siekiami privalumai.....</b>	<b>43</b>
<b>1.7. Siekiamo sprendimo apibrėžimas .....</b>	<b>43</b>
<b>1.8. Analizės išvados .....</b>	<b>44</b>
<b>2. Veiklos žodynų ir taisyklių neprieštaringumo tikrinimo sprendimo prototipo reikalavimų specifikacija ir projektas, formalus aprašas .....</b>	<b>45</b>
<b>2.1. Reikalavimų specifikacija .....</b>	<b>45</b>
2.1.1. Panaudojimo atvejai .....	45
<b>2.2. Dalykinės srities modelis.....</b>	<b>50</b>
<b>2.3. Naudotojų sąsajos modelis.....</b>	<b>51</b>
<b>2.4. Formalus veiklos žodynų ir taisyklių neprieštaringumo tikrinimo aprašas .....</b>	<b>52</b>
2.4.1. Prieštaringumų aptikimas žodyne.....	52
2.4.2. Prieštaringumų aptikimas veiklos taisyklėse.....	63
<b>2.5. Reikalavimų apibendrinimas .....</b>	<b>68</b>
<b>3. Veiklos žodynų ir taisyklių neprieštaringumo tikrinimo sprendimo prototipo projektas .</b>	<b>69</b>
<b>3.1. Sistemos loginė architektūra .....</b>	<b>69</b>
<b>3.2. Sistemos elgsenos modelis.....</b>	<b>69</b>
<b>3.3. Realizacijos modelis.....</b>	<b>70</b>
<b>3.4. Realizacijos projekto apibendrinimas .....</b>	<b>72</b>
<b>4. Sprendimo realizacija ir testavimas.....</b>	<b>73</b>

<b>4.1.Sprendimo realizacijos ir veikimo aprašas .....</b>	<b>73</b>
<b>4.2.Testavimo modelis, duomenys, rezultatai.....</b>	<b>75</b>
<b>5. Eksperimentinis veiklos žodynų ir taisyklių neprieštaringumo tikrinimo metodo prototipinio sprendimo tyrimas .....</b>	<b>79</b>
<b>5.1.Eksperimento planas.....</b>	<b>79</b>
<b>5.2.Eksperimento rezultatai.....</b>	<b>79</b>
5.2.1. Renginių organizavimo dalykinė sritis.....	79
5.2.2. Veiklos žodyno ir veiklos taisyklių aprašymas .....	80
5.2.3. Papildomų konceptų ir taisyklių aprašymas siekiant sukurti prieštaringumus.....	84
5.2.4. Prieštaringumų aptikimas pasinaudojant šiame darbe sukurtu metodo prototipu.....	86
5.2.5. Sprendimo veikimo ir savybių analizė, kokybės kriterijų įvertinimas.....	90
<b>Išvados .....</b>	<b>93</b>
<b>Literatūros sąrašas .....</b>	<b>94</b>
<b>Priedai.....</b>	<b>96</b>

## Lentelių sąrašas

<b>1.1 lentelė.</b> Veiklos žodyno dalinis pavyzdys .....	17
<b>1.2 lentelė.</b> Aletinio modalumo <i>SBVR</i> taisyklėse poliškumai .....	26
<b>1.3 lentelė.</b> Deontinio modalumo <i>SBVR</i> taisyklėse poliškumai .....	27
<b>1.4 lentelė.</b> Žodyno aprašo atributai su paaiškinimais .....	29
<b>1.5 lentelė.</b> Žodyno įrašo aprašo atributai su paaiškinimais.....	30
<b>1.6 lentelė.</b> <i>SBVR</i> kardinalumų ir intervalinių reikšmių sulyginimas .....	38
<b>1.7 lentelė.</b> Esamų <i>SBVR</i> prieštarigumų aptikimo sprendimų palyginimas .....	42
<b>2.1 lentelė.</b> PA 1. Aprašyti veiklos taisyklės specifikacija .....	46
<b>2.2 lentelė.</b> PA 2. Aprašyti veiklos žodyną specifikacija.....	46
<b>2.3 lentelė.</b> PA 3. Tikrinti <i>SBVR</i> žodyno prieštarigumus specifikacija.....	47
<b>2.4 lentelė.</b> PA 4. Tikrinti <i>SBVR</i> veiklos taisyklių prieštarigumus specifikacija .....	47
<b>4.1 lentelė.</b> Sukurto metodo prototipo testavimo rezultatai .....	75
<b>5.1 lentelė.</b> Renginių organizavimo veiklos žodynas.....	80
<b>5.2 lentelė.</b> Renginių organizavimo veiklos taisyklės.....	83
<b>5.3 lentelė.</b> Renginių organizavimo veiklos papildomi konceptai prieštarigumams sukurti .....	84
<b>5.4 lentelė.</b> Prieštarigumų aptikimo rezultatų palyginimas.....	87
<b>5.5 lentelė.</b> Klaidų matrica .....	90
<b>5.6 lentelė.</b> Eksperimento gautų rezultatų parametrai.....	91

## Paveikslų sąrašas

<b>1.1 pav.</b> Informacinės sistemos tipinis kūrimo procesas [3] .....	16
<b>1.2 pav.</b> Renginių organizavimo modelis aprašytas UML klasių diagrama [6].....	18
<b>1.3 pav.</b> Semantinis trikampis SBVR terminais [9] .....	20
<b>1.4 pav.</b> Renginių organizavimo veiklos žodyno fragmentas .....	21
<b>1.5 pav.</b> Individualių konceptų aprašymas.....	23
<b>1.6 pav.</b> <i>SBVR</i> kategorizacijos pavyzdys .....	25
<b>1.7 pav.</b> <i>SBVR</i> segmentacijos pavyzdys.....	25
<b>1.8 pav.</b> Leidimo atlikti veiksmą pavyzdys.....	28
<b>1.9 pav.</b> <i>SBVR</i> taisyklių kardinalumo pavyzdys.....	29
<b>1.10 pav.</b> Žodyno aprašas.....	30
<b>1.11 pav.</b> Žodyno įrašo aprašas .....	32
<b>1.12 pav.</b> Pasikartojančio atributo SBVR žodyno įrašė pavyzdys .....	34
<b>1.13 pav.</b> <i>Synonym</i> ir <i>See</i> atributų atitikimas .....	34
<b>1.14 pav.</b> <i>Synonym</i> ir <i>See</i> atributų neatitikimas .....	34
<b>1.15 pav.</b> Pasikartojančio įrašo sukeliančio dviprasmybę <i>SBVR</i> žodyne pavyzdys.....	34
<b>1.16 pav.</b> Pasikartojančio įrašo pasinaudojant <i>Subject</i> atributu pavyzdys .....	35
<b>1.17 pav.</b> Prieštarinių charakteristikų pavyzdys.....	35
<b>1.18 pav.</b> Statinio apribojimo prieštaravimo pavyzdys.....	35
<b>1.19 pav.</b> Rolės prieštaravimo pavyzdys .....	36
<b>1.20 pav.</b> Prieštaringas rolės priskyrimas.....	36
<b>1.21 pav.</b> <i>SBVR</i> kategorizacijos prieštaravimo pavyzdys .....	36
<b>1.22 pav.</b> <i>SBVR</i> žodyno įrašų cikliškumas .....	37
<b>1.23 pav.</b> Kardinalumo prieštaravimo pavyzdys.....	37
<b>1.24 pav.</b> Taisyklių veiksmožodinių konceptų prieštaravimas .....	39
<b>1.25 pav.</b> Taisyklių modalumo prieštaravimas .....	39
<b>1.26 pav.</b> Rolės prieštaravimas .....	39
<b>1.27 pav.</b> <i>SBVR</i> veiksmožodinio koncepto taisyklės prieštaravimo žodyno konceptų aprašymas .	40
<b>1.28 pav.</b> <i>SBVR</i> veiksmožodinio koncepto prieštaravimo taisyklės aprašymas.....	40
<b>1.29 pav.</b> <i>SBVR</i> veiksmožodinio koncepto taisyklės prieštaravimą sukeliantys faktai.....	40
<b>2.2 pav.</b> SBVR veiklos taisyklių ir žodyno neprieštaravimo tikrinimo panaudojimo atvejų diagrama .....	46
<b>2.3 pav.</b> PA 1. Aprašyti veiklos taisykles veiklos diagrama .....	48
<b>2.4 pav.</b> PA 2. Aprašyti veiklos žodyną veiklos diagrama.....	49
<b>2.5 pav.</b> PA 3. Tikrinti SBVR žodyno prieštaravimus veiklos diagrama .....	49
<b>2.6 pav.</b> PA 4. Tikrinti SBVR veiklos taisyklių prieštaravimus veiklos diagrama.....	50
<b>2.7 pav.</b> <i>SBVR</i> veiklos taisyklių ir žodyno įrašų prieštaravimo tikrinimo <i>UML</i> esybių klasių diagrama .....	51
<b>2.8 pav.</b> <i>SBVR</i> veiklos taisyklių ir žodyno įrašų prieštaravimo aptikimo naudotojo sąsajos modelis .....	52
<b>2.9 pav.</b> <i>SBVR</i> veiklos žodyno ir taisyklių neprieštaravimo tikrinimo proceso diagrama.....	52
<b>2.9 pav.</b> Aptikti pasikartojančius žodyno įrašų atributus .....	54
<b>2.10 pav.</b> Aptikti sinonimų aprašymų neatitikimus .....	55
<b>2.11 pav.</b> Aptikti pasikartojančius įrašus .....	56

<b>2.13 pav.</b> Aptikti prieštaringas charakteristikas, kurių bendriniai konceptai sutampa, o veiksmožodiniai konceptai yra priešingų poliškumų.....	57
<b>2.14 pav.</b> Aptikti aprašytų faktų neatitikimus statiniams apribojimams žodyne .....	58
<b>2.15 pav.</b> Aptikti netinkamus rolių aprašymus, kai rolės turi daugiau nei vieną bendrinį konceptą	59
<b>2.16 pav.</b> Aptikti netinkamus rolių priskyrimus, kai rolės neturi nurodyto tinkamo koncepto tipo .	60
<b>2.17 pav.</b> Aptikti konceptų aprašų prieštaringumus, kai konceptas priklauso daugiau negu vienai kategorijai iš vienos kategorizavimo schemos arba segmentacijos .....	61
<b>2.18 pav.</b> Aptikti cikliškumus konceptų aprašymuose, kai konceptui nurodytas bendrinis konceptas sukuria tipų cikliškumą.....	62
<b>2.19 pav.</b> Aptikti kardinalumų neatitikimus.....	64
<b>2.20 pav.</b> Aptikti veiksmožodinių konceptų prieštaringumus, kai veiksmožodinis konceptas taisyklėje sukuria priešingą poliškumą kitai taisyklei .....	65
<b>2.21 pav.</b> Aptikti modalumo prieštaringumus, kai modališkumai taisyklėse sukuria priešingus poliškumus.....	66
<b>2.22 pav.</b> Aptikti rolių prieštaringumus, kai taisyklės apriboja galimas rolių reikšmes ir reikšmių tipus .....	67
<b>2.23 pav.</b> Aptikti veiksmožodinių konceptų taisyklių prieštaringumus, kai žodyne aprašytų faktų kiekis sukelia prieštaringumą aprašytoms taisyklėms .....	68
<b>3.1 pav.</b> Detalizuota sistemos loginė architektūra.....	69
<b>3.2 pav.</b> Sistemos elgsenos modelis .....	70
<b>3.3 pav.</b> Sistemos komponentų diagrama.....	71
<b>3.4 pav.</b> Sistemos diegimo diagrama .....	71
<b>4.1 pav.</b> <i>SBVR</i> žodyno redagavimo langas.....	73
<b>4.2 pav.</b> <i>SBVR</i> taisyklių redagavimo langas.....	74
<b>4.3 pav.</b> Mygtukas veiklos taisyklių ir žodynų prieštaringumams aptikti .....	74
<b>4.4 pav.</b> Aptikto prieštaringumo klaidos pranešimas .....	75
<b>5.1 pav.</b> Renginių organizavimo dalykinės srities esybių ryšių modelis .....	80
<b>5.2 pav.</b> Renginių organizavimo veiklos žodyno ir taisyklių prieštaringumo aptikimo rezultatas ...	84
<b>5.3 pav.</b> Eksperimento metu gauto informacinio pranešimo langas .....	87
<b>5.4 pav.</b> Išsamumo ir tikslumo formulės [30] .....	91
<b>5.5 pav.</b> f-mato apskaičiavimo formulė [30].....	91

## Santrumpų ir terminų sąrašas

**SBVR** (ang. *Semantics of Business Vocabulary and Business Rules*) - tai *OMG* (ang. *Object Management Group*) standartas, skirtas detalia, formalia, struktūrizuota natūralia kalba aprašyti veiklos taisyklės, kurias gali suprasti kompiuterinės sistemos.

**SBVR specifikacija** – tai veiklos žodynas ir taisyklės, aprašytos natūralia struktūrizuota kalba pagal *SBVR* standartą.

**Prieštaringumas** – dviprasmybę sukeltantys įrašai arba *SBVR* taisyklės, kurios yra prieštaringos viena kitai ir vienu metu negali būti patenkinamos.

**OWL** (angl. *Web Ontology Language*) – tai žinių pavaizdavimo kalbų grupė, skirta aprašyti ontologijas.

**URI** (ang. *Uniform Resource Identifier*) – tai eilutė simbolių, kuri vienareikšmiškai identifikuoja resursą [1].

**DSL** (ang. *Domain Specific Language*) – tai kalba, skirta aprašyti tam tikrą veiklos sritį.

**JVM** (ang. *Java Virtual Machine*) – virtuali mašina, kuri leidžia kompiuteriui paleisti programinį kodą, sukompiliuotą *Java bytecode*.

**JDK** (ang. *Java Development Kit*) – programinis paketas, skirtas kurti programinę įrangą *Java* programavimo kalba. *JRE* yra *JDK* viena iš sudedamųjų dalių.

**JRE** (ang. *Java Runtime Environment*) – programinis paketas, kuris turi minimalius reikalavimus skirtus paleisti *Java* programavimo kalba parašytas programas.

**Ontologija** – kategorijų, savybių ir ryšių tarp konceptų, duomenų ir esybių formalus aprašas.

## Ivadas

Darbas priklauso “6211BX009 Informacinių sistemų inžinerija” studijų programai. Magistratūros kompetencija – vadovavimas. Šis darbas yra skirtas sukurti metodą, kuris leistų automatiškai patikrinti, ar *SBVR* specifikacijose nėra prieštaravimų. Šiame darbe bus analizuojama *SBVR* specifikacija, išskirti pasitaikančių teiginių prieštaravimų tipai, pasiūlyta, kaip aptikti prieštaravimus, ir eksperimentiškai ištirtas pasiūlytas prieštaravimų aptikimo metodas.

### Darbo problematika ir aktualumas

Informacinės sistemos yra kuriamos siekiant palengvinti ir pagreitinti atliekamą darbą tam tikroje veiklos srityje. Informacinių sistemų kūrimo procesas susideda iš reikalavimų analizės, projektavimo, realizacijos, testavimo ir, jeigu to reikalauja kuriama sistema, diegimo ir palaikymo procesų. Priklausomai nuo naudojamo metodo, tam tikri informacinių sistemų kūrimo proceso etapai gali būti atliekami iteraciniu būdu, nuosekliai arba naudojant kombinuotą kūrimo procesą. Kuriamos sistemos privalo tenkinti keliamus užsakovų reikalavimus ir neviršyti numatyto, iš anksto sutarto skiriamo biudžeto sistemos kūrimo procesui. Dažniausiai ne visi užsakovų keliami reikalavimai būna žinomi sistemos kūrimo proceso pradžioje. Tai nutinka dėl kuriamos sistemos užsakovų ir informacinės sistemos kūrėjų komunikacijos trūkumo arba gebėjimo suvokti reikiamą kuriamai sistemai keliamų reikalavimų detalumo lygį. Informacinės sistemos užsakovai turi viziją, kaip sukurta sistema turėtų veikti, tačiau nesupranta sistemos kūrimui naudojamų modelių ar schemų. Dėl šios priežasties užsakovams yra sudėtinga patvirtinti suprojektuotą informacinę sistemą. Norint užtikrinti sklandesnį bendravimą tarp užsakovų ir sistemų analitikų, architektų ir kūrėjų, yra naudojami veiklos žodynai ir taisyklės, kurios yra aprašomos natūralia struktūrizuota kalba. Natūralios struktūrizuotos kalbos privalumas yra tas, kad ji yra lengvai aprašoma ir suprantama veiklą vykdantiems asmenims bei suteikiama galimybę transformuoti aprašytus terminus ir taisykles į programinį kodą. Vienas iš būdų aprašyti veiklos žodynus ir taisykles yra *SBVR* standartas. Kintant veiklos procesams kinta ir naudojamas žodynas bei veiklos taisyklės. Informacinės sistemos turėtų gebėti kuo paprasčiau prisitaikyti prie veikloje atsiradusių pokyčių. Atsirandant pokyčiams yra neišvengiamos klaidos, kurios gali atsirasti dėl prieštaravimų esamoms aprašytoms veiklos taisyklėms arba žmogiškojo faktoriaus, kuris egzistuoja aprašinėjant veikloje naudojamą žodyną ir taisykles. Kuriant informacines sistemas veiklos žodynai ir taisyklės yra naudojamos kaip reikalavimai kuriamai sistemai, todėl esantys prieštaravimai gali nulemti netinkamą sukurtos sistemos veikimą. Siekiant užtikrinti veiklos žodyno ir taisyklių neprieštaravimą, jų aprašymo etape reikėtų realizuoti neprieštaravimo tikrinimo automatizavimą. Šiame darbe bus kuriamas metodas, kuris leis aptikti prieštaravimus aprašytuose *SBVR* veiklos žodynuose ir taisyklėse. Šiame darbe pasirenkamas *SBVR* standartas, nes jis leidžia aprašyti veiklos žodynus ir taisykles natūralia kalba.

**Problematika** – nėra tiesioginio būdo patikrinti *SBVR* žodyno ir taisyklių loginį neprieštaravimą.

### Darbo tikslas ir uždaviniai

**Tikslas** – sudaryti sąlygas prieštaravimams aptikti *SBVR* specifikacijoje.

### Uždaviniai

1. Išanalizuoti:
  - a. informacinių sistemų kūrimo procesą;

- b. *SBVR* standartą ir metamodelį;
  - c. *SBVR* veiklos žodynų ir taisyklių sudarymo principus;
  - d. esamus *SBVR* prieštaravimo aptikimo mechanizmus.
2. Nustatyti *SBVR* prieštaravimų tipus.
  3. Sukurti metodą, kuris būtų naudojamas prieštaravimų aptikimui *SBVR* specifikacijose.
  4. Sukurti prototipą, kuris pagal sukurtą metodą aptinka *SBVR* veiklos žodynų ir taisyklių prieštaravimus.
  5. Ištestuoti prieštaravimų aptikimo algoritmų veiksmingumą, naudojant nustatytus prieštaravimų tipus.
  6. Eksperimentiškai ištirti prieštaravimų aptikimą *SBVR* specifikacijose.
  7. Apibendrinti tyrimo rezultatus.

### **Darbo rezultatai ir jų svarba**

Darbo rezultatas yra sukurtas metodas, kuris leidžia aptikti *SBVR* specifikacijose užrašytų teiginių prieštaravimus jų užrašymo aplinkoje.

### **Darbo struktūra**

Darbą sudaro šios dalys:

- informacinių sistemų kūrimo proceso analizė;
- *SBVR* standarto analizė;
- prieštaravimų *SBVR* specifikacijose analizė;
- esamų sprendimų analizė;
- kuriamo metodo reikalavimų analizė ir projektavimas;
- sprendimo realizacija;
- eksperimentas ir rekomendacijos;
- išvados ir pastebėjimai.

## 1. Probleminės srities analizė

Šiame skyriuje pateikiama su darbo problematika susijusios informacijos analizė.

### 1.1. Analizės tikslas

Šiame skyriuje siekiama išanalizuoti *SBVR* specifikaciją, *SBVR* metamodelį, žodyną, taisyklių sudarymo principus ir esamus *SBVR* prieštaravimo tikrinimo metodus. Šiame skyriuje taip pat bus išskirti teiginių prieštaravimų tipai.

### 1.2. Tyrimo objektas, sritis ir problema

**Tyrimo objektas** – prieštaravimų ir klaidų tikrinimas veiklos žodyną ir taisyklių specifikacijose.

**Tyrimo sritis** – veiklos žodyną ir taisyklių specifikavimo procesas.

Pagrindinė šiuo metu sprendžiama problema yra ta, jog surašius *SBVR* teiginius nėra galimybės žinoti, ar teiginiai tarpusavyje logiškai nepriešaringi. Esami *SBVR* redaktoriai turi galimybes aptikti sintaksines klaidas, tačiau tai neapsaugoja nuo loginių klaidų, kurias yra sudėtinga aptikti neįsigilinus į visas aprašytas taisykles ir žodyną įrašus. Programinė įranga suteikia galimybę eksportuoti *SBVR* teiginius į *OWL* ontologiją, kurioje atsiranda galimybė pasinaudoti modelio patikrinimais, siekiant atrasti loginius prieštaravimus. Šio proceso trukumas yra tas, jog tai yra kelių veiksmų seka, kuri apima duomenų eksportavimą. Tai trukdo atlikti prieštaravimų aptikimo veiksmus realiu laiku ir suteikti naudotojui indikaciją grafinėje naudotojo sąsajoje. Prieštaravimų atradimas *SBVR* teiginiuose taip pat pagreitintų naudotojų darbą ir prisidėtų prie šio standarto adaptacijos dėl sumažėjusios teiginių užrašymo trukmės bei klaidų aptikimo ankstyvesniuose darbų etapuose.

### 1.3. Prieštaravimų ir klaidų tikrinimo veiklos žodyną ir taisyklių specifikacijose analizė

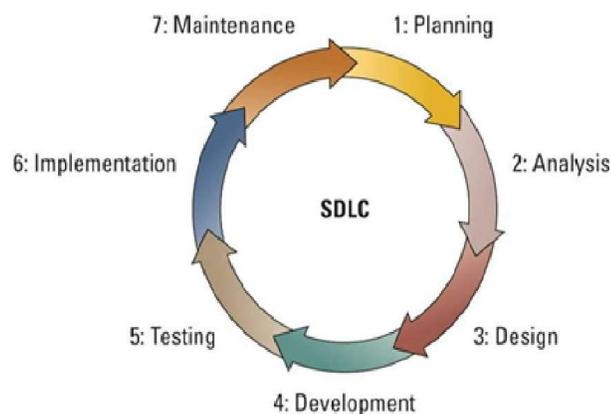
#### 1.3.1. Informacinių sistemų kūrimo procesas

Informacinių sistemų kūrimo procesas susideda iš sistemos planavimo, analizės, specifikavimo, kūrimo, testavimo, diegimo ir priežiūros [2]. 1.1 pav. pavaizduotas informacinės sistemos tipinis kūrimo procesas. Šis procesas padeda programinės įrangos inžinieriams sukurti informacines sistemas ir užtikrinti sukurtos sistemos korektišką veikimą, kurio pageidauja užsakovas. Toliau aprašyti šio proceso etapai:

- planavimo etape yra nagrinėjama veiklos problema, siūlomi alternatyvūs sprendimo būdai, analizuojami sistemos kūrimo kaštai ir atnešama nauda vykdomai veiklai;
- analizės etape yra apibrėžiami projekto tikslai, planuojamas sistemos funkcionalumas. Šiame etape taip pat yra surenkami faktai ir reikalavimai sistemai iš būsimų jos naudotojų. Reikalavimų rinkimas gali būti atliekamas surenkant faktus iš dokumentacijų, veiklos dalyvių apklausos metu, stebint veiklos dalyvius vykdant veiklą, pildant klausimynus, nagrinėjant esamas informacines sistemas, analizuojant siūlomą sistemą [2]. Šis etapas yra labai svarbus informacinių sistemų kūrimo procese, nes veiklos dalyviai dažniausiai turi galutinės sistemos veikimo įsivaizdavimą, tačiau negali to struktūrizuoti pateikti informacinių sistemų inžinieriams, kurie galėtų informaciją perkelti į kuriamos informacinės sistemos programinį kodą. *SBVR* standartas leidžia struktūrizuota natūralia kalba aprašyti veiklos žodyną ir taisykles, kurios gali būti panaudojamos išskiriant kuriamos sistemos reikalavimus ir dėl savo

savybių gali būti interpretuojamos kompiuterinių sistemų ir panaudojamos sistemos programinio kodo kūrime;

- specifikuojamo etape yra detaliam aprašomas planuojamas funkcionalumas:
  - sukuriama programinių langų atvaizdai;
  - kompiuterizuojamų procesų diagramos;
  - informacijos srautų diagramos;
  - komponentų diagramos;
  - duomenų bazių schemas ir visos kitos kuriamos sistemos aprašymui skirtos diagramos.
- kūrimo etape yra rašomas programinis kodas;
- testavimo etape yra sukuriama testavimo aplinka, kurioje sistema yra paleidžiama ir išbandomas sukurto sistemos funkcionalumas. Aptiktos klaidos ar nesuderinamumai yra fiksuojami ir planuojamas klaidų taisymas;
- diegimo etape sukurta informacinė sistema yra patalpina galutiniam naudotojui pasiekiamoje aplinkoje ir pradeda naudoti;
- priežiūros etape sistemos funkcionalumas yra vertinamas siekiant užtikrinti, kad informacinė sistema veikia korektiškai ir pagal iš anksto apibrėžtus reikalavimus. Šio etapo metu gali būti atliekami pakeitimai informacinei sistemai.



1.1 pav. Informacinės sistemos tipinis kūrimo procesas [3]

*SBVR* standartas leidžia aprašyti veiklos žodyną ir taisykles struktūrizuota natūralia kalba, todėl šiame darbe ir yra nagrinėjamas.

### 1.3.2. *SBVR* standartas

*SBVR* yra veiklos žodyno ir taisyklių aprašymo standartas. Veiklos turi apibrėžtas taisykles, pagal kurias yra vykdoma veikla bei siekiami užsibrėžti tikslai. Taisyklės veiklose dažnai būna sudėtingos ir pastoviai besikeičiančios [4]. Be terminų ir apribojimų, tam tikrų nusistovėjusių taisyklių, jokia veikla negalėtų tinkamai funkcionuoti. Organizacijos, norėdamos išlikti konkurencingos rinkoje siekia naudotis informacinėmis sistemomis, kurios palengvina, pagreitina ir sumažina veiklos vykdymo kaštus [5]. Informacinių sistemų kūrėjai, siekdami sukurti tinkamą naudojimui informacinę sistemą, turi suprasti vykdomą veiklą ir jos taisykles bei, remiantis surinktomis žiniomis, sukurti sistemą. Veikloje dalyvauja žmonės ir informacinės sistemos laikantis veiklos taisyklių, kurios

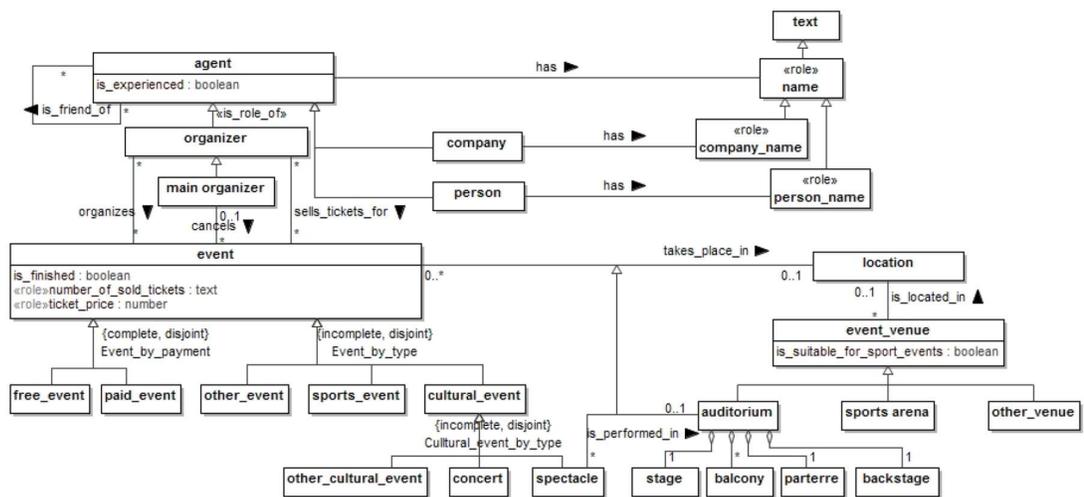
pastoviai keičiasi dėl besikeičiančios rinkos padėties arba reglamentų. Veiklą vykdančios asmenys turi sugebėti perteikti veikloje naudojamus terminus bei taisykles, o programuotojai suprogramuoti veiklos taisykles ir apribojimus programiniame kode. Suprogramavus informacines sistemas, veiklos dalyvis negali paimti ir patikrinti, ar programinis kodas yra tinkamai veikiantis. *SBVR* siekia šį atskyrimą sumažinti ir suteikia būdą veiklos taisykles užrašyti veiklą vykdančioms asmenims suprantama struktūrizuota natūralia kalba, kurią gali interpretuoti ir informacinės sistemos [6]. Tokiu būdu palengvina bendravimą tarp veiklos analitikų, modeliotojų, veiklos vykdytojų ir informacinių sistemų kūrėjų. Siekiant prisitaikyti prie besikeičiančių rinkų, turi būti keičiamos veiklos taisyklės, o tuo pačiu ir naudojamos informacinės sistemos. Pokyčiai veiklos taisyklėse sukuria prielaidas atsirasti taisyklių prieštaravimams [4], atsirandantiems dėl žmogiškųjų klaidų, kurios yra sunkiai išvengiamos, esant dideliame taisyklių kiekiui arba kai reikia keisti esamas veiklos taisykles.

*SBVR* specifikacijoje žodynai ir taisyklės aprašo veiklą natūralia kalba. Tai leidžia modeliuoti konceptus ir taisykles, kurios yra suprantamos veiklos dalyviams. Pagrindinė tokio požiūrio nauda yra komunikacija tarp veiklos dalyvių ir veiklos dalyvių su informacinių technologijų personalu [7]. Veiklos yra reglamentuojamos vidine tvarka patvirtintais dokumentais arba valstybiniais įsakymais. Šie dokumentai yra pagrindiniai veiklos terminų ir sąlygų šaltiniai, iš kurių išgaunama informacija ir sudaromos *SBVR* specifikacijos. Sudarinėjant *SBVR* žodyną, veiklos dalyviai turi suprasti veiklą supančius įstatymus, reglamentus ir sugebėti juos išreikšti pasinaudojant savo veiklos srities terminais. Dažnai išskirti terminai sukelia dviprasmiškumą, su kuriais veiklos dalyviai privalo susidoroti. *SBVR* specifikavimo proceso rezultatas yra formalus veiklos žodynas, kuris aprašo terminus ir jų sąsajas. Toks formalus žodynas padeda veikloms sumažinti dviprasmiškumus ir veiklos dalyviams suteikia aiškumo. Turint *SBVR* žodyną įmanoma atlikti transformacijas į *UML* modelius. 1.1 lentelėje pavaizduotas dalinis *SBVR* veiklos žodynas, skirtas renginių organizavimo veiklai. 1.2 pav. pavaizduotas į *UML* klasių diagramą transformuotas žodynas. *SBVR* bendriniai teiginiai yra išreikšti kaip *UML* klasės, o veiksmožodiniai konceptai yra išreikšti kaip ryšiai tarp klasių. Šis pavyzdys atspindi tam tikrus papildomus konstruktus, kuriuos galima aprašyti *SBVR* struktūrizuota kalba: klasių hierarchijos, apibendrinimai tarp rolių ir ryšių, dvikrypčiai ryšiai, agregacija, kardinalumo apribojimai.

**1.1 lentelė.** Veiklos žodyno dalinis pavyzdys

<b>Bendriniai konceptai</b>	<u>agent</u> <u>person</u> General concept: <u>agent</u> Synonym: <u>human</u> <u>company</u> General concept: <u>agent</u>
<b>Veiksmožodinių konceptų rolės</b>	<u>organizer</u> Concept_type: <u>verb_concept_role</u> General concept: <u>agent</u>
<b>Veiksmožodiniai konceptai</b>	<u>organizer organizes event</u> Synonymous_form: <u>event is_organized_by organizer</u>
<b>Savybių ryšiai</b>	<u>name</u> General_concept: <u>text</u> Concept_type: <u>role</u> <u>ticket_price</u>

	<u>General_concept: number</u> <u>Concept_type: role</u> <u>organizer has name</u> <u>Concept_type: property_association</u> <u>event has ticket_price</u> <u>Concept_type: property_association</u>
<b>Rolės</b>	<u>Company_name</u> <u>Concept_type: role</u> <u>General concept: text</u>
<b>Individualus konceptas</b>	<u>Jonas_Grinius</u> <u>General_concept: person</u> <u>Events_Ltd</u> <u>General_concept: company</u>



1.2 pav. Renginių organizavimo modelis aprašytas UML klasių diagrama [6]

Iš dokumentuotų veiklos reglamentų išskiriant terminus, tuo pačiu yra išskiriamos sąlygos, kurios leidžia arba draudžia vykdyti veiklą. Tuo pačiu yra išskiriami teiginiai, kurie nusako, kaip veikla turėtų būti vykdoma. *SBVR* specifikacijose veiklos taisyklės privalo naudotis terminais, aprašytais veiklos žodyne, todėl pirmiausia yra aprašomi naudojami terminai, rolės, sinonimai ir veiksmožodiniai konceptai, o tik tada pasinaudojant aprašytais konceptais – formuojamos *SBVR* veiklos taisyklės.

Taisyklė - tai deklaratyvus sakiny, kuris pagal duotą situaciją gali būti tenkinamas arba netenkinamas. Taisyklės yra formuluojamos paprastomis loginėmis formuluotėmis pasinaudojant veiksmožodiniu konceptu. Atominės formuluotės yra sudaromos susiejant (ang. *bind*) veiksmožodinių konceptų roles su individualiais konceptais, kintamaisiais (dažniausiai bendriniais konceptais) arba kitomis loginėmis formuluotėmis, pavyzdžiui kardinalumų arba atributų reikšmių rėžių taisyklėmis. Taisyklės gali būti sudarytos iš vieno veiksmožodinio koncepto, pavyzdžiui:

renginys yra\_organizuojamas organizatoriaus Events Ltd.

Taisyklės taip pat gali būti sudėtinės ir būti sudarytos iš kelių veiksmožodinių konceptų, sujungtų konjunkcija arba disjunkcija, pavyzdžiui:

renginys yra organizuojamas organizatoriaus Events Ltd ir turi parduotų bilietų kiekį didesnį negu 100

Deontinio modalumo taisyklė, kuri nurodo, kad bilietai parduodami tik tada, kai renginys yra mokamas:

It is obligatory that organizer sells tickets for event if event is paid event.

Aletinio modalumo taisyklė, kuri nurodo, kad renginys organizuojamas būtinai vienoje vietovėje:

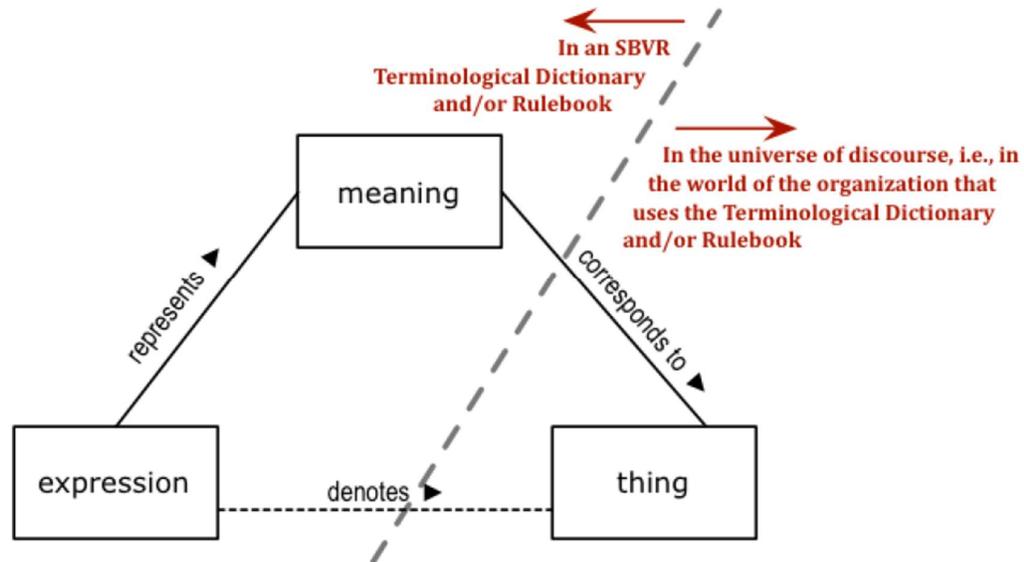
It is necessary that event takes place in exactly 1 location.

Aletinio modalumo taisyklė, kuri nurodo, kad organizatorius yra įmonė, kuri organizuoja bent 1 renginį:

It is necessary that organizer is company that organizes at least 1 event.

### 1.3.3. SBVR veiklos žodynų ir taisyklių specifikacija

*SBVR* (angl. *Semantics of Business Vocabulary and Business Rules*) – tai *OMG* (angl. *Object Management Group*) standartas, skirtas detalia, formalia, struktūrizuota natūralia kalba aprašyti veiklos žodynus ir taisykles, kurias gali suprasti kompiuterinės sistemos [8]. *SBVR* specifikacija - tai pagal *SBVR* standartą aprašytas veiklos žodynas ir veiklos taisyklės. *SBVR* specifikacijos lingvistika grindžiamos architektūros teorinis pagrindas yra *SBVR* semantinis trikampis, leidžiantis atskirti prasmę (ang. *meaning*) nuo atvaizdavimo (ang. *expression*). Prasmės nuo atvaizdavimo atskyrimas leidžia veiklos sričiai būdingų konceptų reikšmes atvaizduoti skirtingais būdais, pavyzdžiui skirtingomis kalbomis, sinonimais ar paveikslėliais. *SBVR* atvaizdavimas neprivalo būti tekstinė išraiška. 1.3 pav. pavaizduotas semantinis trikampis susietas su *SBVR* konceptais:



1.3 pav. Semantinis trikampis SBVR terminais [9]

- išraiška (ang. *expression*) – tai, kas išreiškia arba komunikuoja, bet yra traktuojama atskirai nuo jos interpretacijos, pavyzdžiui simbolių seka „automobilis“, diagrama, knygos tekstas. Išraiška *SBVR* semantiniame trikampyje pati iš savęs neturi jokios reikšmės ar interpretacijos. Išraiška yra atvaizdavimo konceptas, kuris tik nurodo į tam tikrą reikšmę. Veiklos žodynuose ir taisyklėse aprašomos išraiškos dažniausiai yra sakiniai. *SBVR* specifikacija aprašo nedidelę aibę anglų kalbos terminų, kurie yra susiejami su *SBVR* konceptais ir tokiu būdu leidžia aprašyti specifikacijas natūralia kalba. Šis anglų kalbos teiginių ir *SBVR* konceptų atitikimas yra vadinamas *SBVR Structured English*. *SBVR Structured English* atitinka išraiškos dalį 1.3 pav.;
- reikšmė (ang. *meaning*) – tai, ką norima pasakyti žodžiu, teiginiu arba ženklu. Tai gali būti apibūdinimas, kuris aprašo tai ką kažkas supranta. Reikšmė gali turėti daug išraiškų. *SBVR* specifikacijose reikšmės yra aprašomos žodynuose ir veiklos taisyklėse;
- dalykas (ang. *thing*) – viskas, kas yra suvokiama arba įsivaizduojama.

*SBVR* kontekste reikšmė atitinka dalyką, o išraiška atstovauja ir rodo reikšmę. Remiantis tokiu loginiu pagrindu yra formuluojami žodynai ir taisyklės.

Standarto specifikacija [9] apibrėžia žodyną ir taisykles, kurių pagalba yra užrašomi veikloje naudojami terminai, veiklos faktai bei veiklos taisyklės. Standartas tuo pačiu pateikia *XMI* schemą [10], kurios pagalba galima dalintis veiklos žodynų ir taisyklių informacija tarp skirtingų kompiuterizuotų sistemų. Standarte yra pateikiamas ir *SBVR* metamodelis [11], kuris aprašo naudojamus tipus. Vienas pagrindinių *SBVR* artefaktų yra *SBVR* veiklos žodynas.

### 1.3.3.1. *SBVR* veiklos žodynas

*SBVR* veiklos žodynas – tai veikloje naudojamų terminų ir jų apibrėžimų žodynas. Žodynai nenusako veiklos taisyklių struktūros, o patys žodyno elementai yra užrašomi kaip individualūs teiginiai, kurie

gali būti lengvai pridedami, panaikinami arba modifikuojami. Tai yra patogiu, nes naudojamos sąvokos gali pakankamai dažnai keistis, atsirasti sinonimų bei naujų terminų, skirtų apibūdinti esamus arba naujus procesus ir taisykles. 1.4 pav. pavaizduota keletas žodyno įrašų. Šiame žodyne aprašoma su renginiais susijusi veikla. Žodynas yra skirtas aprašyti naudojamus terminus, kaip jie tarpusavyje siejasi, kuriame kontekste gali būti naudojami, bei turi galimybę nurodyti papildomą informaciją, kuri yra naudinga veiklą vykdančioms asmenims. Toliau yra aptariamas *SBVR* struktūrizuotos kalbos žymėjimas.

```

event
event is_finished
⊖ payment_type
  Concept_type: categorization_type
  Necessity: is_for general_concept event
⊖ Events_by_payment
  Necessity: segmentation for general_concept event that subdivides event by payment_type
⊖ free_event
  General_concept: event
  Necessity: is_included_in Events_by_payment
⊖ paid_event
  General_concept: event
  Necessity: is_included_in Events_by_payment
⊖ sports_event
  General_concept: event
⊖ cultural_event
  General_concept: event
⊖ other_event
  General_concept: event
⊖ concert
  General_concept: cultural_event
⊖ spectacle
  General_concept: cultural_event
⊖ other_cultural_event
  General_concept: cultural_event
⊖ ticket_price
  General_concept: number

```

1.4 pav. Renginių organizavimo veiklos žodyno fragmentas

Siekiant pagerinti *SBVR* specifikacijų skaitomumą, yra naudojamas struktūrizuotas kalbos žymėjimas: „terminas“ stilius naudojamas aprašant bendrinius konceptus (žodis pradedamas mažąja raide). „Vardas“ stilius naudojamas aprašant individualius konceptus (žodis pradedamas didžiąja raide). „veiksmažodis“ stilius naudojamas aprašyti veiksmažodines sąvokas. „raktažodis“ stilius naudojamas rašant papildomus žodžius, kurie padeda formuluoti sakinius [6].

*SBVR* specifikacijoje raktažodžiai yra aprašomi struktūrizuota anglų kalba, kurie yra siūlomi naudoti kaip išraiškos elementai *SBVR* specifikacijose. Dalis *SBVR* siūlomų raktažodžių struktūrizuotoje anglų kalboje:

1. kiekybiniai identifikatoriai, pavyzdžiui *each, some, at least one*;

2. loginiai operatoriai, pavyzdžiui *p and q, p or q, if p then q*;
3. modališkumo operatoriai, pavyzdžiui: *it is impossible that p*;
4. kiti raktažodžiai, tokie kaip: *the, a, an, another, that, who, what*;

Toliau yra aptariami *SBVR* specifikacijų žodynuose naudojami pagrindiniai reikšmės (ang. *meaning*) elementai.

- ***SBVR* bendrinis konceptas** (ang. *general concept*) - tam tikra koncepcija, kuri klasifikuoja dalykus pagal jų bendrines savybes. Bendrinio koncepto pavyzdys:

renginys

- ***SBVR* individualus konceptas** (ang. *individual concept*) – bendrinio koncepto atskiras atvejis (ang. *instance*), kuris atitinka maksimaliai vieną dalyką aprašomoje veiklos srityje. Individualaus koncepto pavyzdys:

Mikutavičiaus koncertas

- ***SBVR* veiksmažodinis konceptas** (ang. *verb concept*) – tai konceptas, kuris specializuoja kitų konceptų būseną ir įtraukia vieną arba daugiau veiksmažodinio koncepto rolių. Veiksmažodinio koncepto pavyzdys:

renginys yra pasibaigęs.

renginys – veiksmažodinio koncepto rolė. yra pasibaigęs – veiksmažodinė sąvoka, kuri nusako įtrauktų rolių būseną.

- ***SBVR* veiksmažodinio koncepto rolė** - tai aprašas, kuris apibūdina esybes pagal jų įsitraukimą į veiksmažodinį konceptą. Veiksmažodinio koncepto rolė yra veiksmažodinio koncepto įsitraukimo taškas, kurį užpildant aprašoma būseną. Veiksmažodinio koncepto su viena veiksmažodinio koncepto role pavyzdys:

renginys yra pasibaigęs.

Šiame pavyzdyje renginys yra veiksmažodinio koncepto rolė. Aprašant būseną privaloma aprašyti veiksmažodinio koncepto rolę veiklos žodyne. Veiksmažodinio koncepto rolės aprašymo pavyzdys:

kalėdinis renginys

General\_concept: renginys

Concept\_type: verb concept role

kalėdinis renginys yra pasibaigęs

- ***SBVR* charakteristika** yra veiksmažodinis konceptas, kuris susidaro iš bendrinio koncepto ir veiksmažodinės sąvokos, apibūdinančios bendrinį konceptą. Charakteristikos yra aprašomos *SBVR* veiklos žodyne. Charakteristikos pavyzdys:

organizacija yra pelninga.

- **SBVR fakto** prasmė *SBVR* specifikacijose yra formuojama naudojant atomines formuluotes. Atominė formuluotė yra žemiausio lygio loginė formuluotė *SBVR* specifikacijose, kuri susidaro iš veiksmožodinio koncepto ir jo rolių, tačiau neturi loginių operacijų. Faktas yra sudaromas remiantis veiksmožodiniu konceptu [9] ir turi susietas veiksmožodinio koncepto roles, kurios formuoja reikšmę. Fakto pavyzdys:

Ryanair perka paslaugas iš Kauno oro uosto.

Šis faktas yra paremtas veiksmožodiniu konceptu:

organizacija perka paslaugas iš oro uosto.

Pirmoji veiksmožodinio koncepto rolė organizacija yra surišta su individualiu konceptu Ryanair. Antroji veiksmožodinio koncepto rolė oro uostas yra susieta su individualiu konceptu Kauno oro uostas. Aprašant faktą turi būti aprašytas statinis apribojimas [9], kuris nurodo fakto galimas reikšmių sritis. *SBVR* turi ir dinaminio apribojimo sąvoką, tačiau šiame darbe ši sritis nenagrinėjama. Kiekvienas aprašomas faktas turi turėti statinį apribojimą, kurį išpildo. Statinio apribojimo pavyzdys:

city is capital of country.

Šis teiginys apriboja įrašyti galimas individualių konceptų reikšmes. Įrašius individualius konceptus yra gaunamas faktas:

Vilnius is capital of Lithuania

Tačiau tokio aprašymo neužtenka. Kiekvienas fakto individualus konceptas turi turėti aprašyme bendrinio koncepto reikšmę, kuri atitinka aprašyto statinio apribojimo reikšmes. Tokio aprašo pavyzdys pavaizduotas 1.5 pav.

```

- Vilnius
  General_concept: city
- Lithuania
  General_concept: country

```

1.5 pav. Individualių konceptų aprašymas

Atominės formuluotės dažniausiai yra naudojamos veiklos taisyklėms išreikšti, nes rolės gali būti susiejamos su kintamaisiais, kurie gali būti apribojami kitų loginių formuluočių, tokių kaip: egzistencinių ar kardinalumo formuluočių.

- **SBVR rolės** *SBVR* specifikacijoje atitinka dalyko (ang. *thing*) atributus. Siekiant atskirti bendrinius konceptus nuo rolių, privalo būti nurodomas aprašomo koncepto tipas (ang. *Concept type*), kuris yra lygus rolei (ang. *role*). Daugiau informacijos apie žodyno įrašų laukus pateikta 1.5 lentelėje, 1.3.3.3 skyriuje. Rolės aprašymo pavyzdys:

amžius

Concept\_type: role

General\_concept: number

Aprašytos rolės pagrindinis aprašas (ang. *primary representation*) yra **amžius**, koncepto tipas nurodomas kaip **role**, o reikšmės tipas nurodomas pagrindinio koncepto (ang. *General\_concept*) atribute. Rolė gali turėti tik vieną pagrindinio koncepto reikšmę. *SBVR* specifikacijoje neaprašyta primityvių tipų aibė, tačiau išanalizavus vieną iš *SBVR* specifikacijų redaktorių [6] buvo nustatyta ne baigtinė, galimų primityvių rolių reikšmių tipų aibė, kurią galima papildyti:

1. Tekstas (ang. *text*).
2. Skaičius (ang. *number*).
3. Sveikasis skaičius (ang. *integer*).
4. Neneigiamas skaičius (ang. *nonnegative integer*).
5. Teigiamas skaičius (ang. *positive integer*).
6. Loginė reikšmė (ang. *boolean*).
7. Data (ang. *date time*).

Siekiant priskirti rolę bendriniam konceptui ir tokiu būdu paversti ją koncepto atributu, rolė yra priskiriama pasinaudojant standartiniu veiksmožodiniu konceptu „*turi*“:

asmuo turi amžių.

- ***SBVR* kategorizavimo schemas** *SBVR* specifikacijoje leidžia kategorizuoti konceptus pagal tam tikras savybes[9]. Kategorizacijos schema *SBVR* specifikacijoje yra naudojama priskirti skirtingas bendrinio koncepto kategorijas. Pavyzdžiui bendrinis konceptas asmuo gali būti skirstomas į kategorijas: berniukas ir mergaitė. 1.6 pav. yra pateikiamas kategorizacijos schemas pavyzdys. Pavyzdyje yra sukuriamas kategorizavimo tipas - asmens tipas, kuris yra skirtas bendriniam konceptui asmuo. Tuomet yra sukuriama kategorizavimo schema Asmenys pagal tipus. Toliau - sukuriamos dvi kategorijos: berniukas ir mergaitė. Toks kategorizacijos pavyzdys sukuria *incomplete disjoint*[12] kategorizavimą. *Incomplete disjoint* reiškia, kad kategorizacijos schema gali turėti daugiau kategorijų negu yra išvardinta ir aprašomas konceptas vienu metu gali turėti tik vieną iš aprašytų tipų.

```

asmuo
  › asmens_tipas
    Concept_type: categorization_type
    Necessity: is_for general_concept asmuo
  › Asmenys_pagal_tipus
    Necessity: categorization_scheme for general_concept asmuo that subdivides asmuo by asmens_tipas
  › berniukas
    General_concept: asmuo
    Necessity: is_included_in Asmenys_pagal_tipus
  › mergaitė
    General_concept: asmuo
    Necessity: is_included_in Asmenys_pagal_tipus

```

1.6 pav. SBVR kategorizacijos pavyzdys

- **SBVR segmentacija** SBVR specifikacijoje yra kategorizavimo schema, kurios visos kategorijos yra išvardintos ir yra nesusikertančios (ang. *disjoint*). Aprašant segmentaciją reikia aprašyti segmentacijos būtinybę (ang. *necessity*) ir kategorijas priskirti jai. 1.7 pav. pavyzdyje yra sukuriamas kategorizavimo tipas mokėjimo tipas, kuris yra skirtas bendriniam konceptui renginys. Tuomet sukuriama segmentacija Ivykiai pagal mokėjimo tipą, kuri suskirsto visus renginius pagal apibrėžtus mokėjimo tipus. Tokiu būdu yra apibrėžiama, kad tai yra pilna (ang. *complete*) nepersidengianti (ang. *disjoint*) kategorizacijos schema. Kategorijos nemokamas renginys ir mokamas renginys priskiriamos aprašytai segmentacijai.

```

renginys
  mokėjimo_tipas
    Concept_type: categorization_type
    Necessity: is_for general_concept renginys
  Ivykiai_pagal_mokėjimo_tipą
    Necessity: segmentation for general_concept renginys that subdivides renginys by mokėjimo_tipą
  nemokamas_renginys
    General_concept: renginys
    Necessity: is_included_in Ivykiai_pagal_mokėjimo_tipą
  mokamas_renginys
    General_concept: renginys
    Necessity: is_included_in Ivykiai_pagal_mokėjimo_tipą

```

1.7 pav. SBVR segmentacijos pavyzdys

### 1.3.3.2. SBVR veiklos taisyklės

SBVR veiklos taisyklės yra aprašomos pasinaudojant struktūrizuotos anglų kalbos (ang. *Structured English*) stiliumi, tačiau naudojami pagalbiniai terminai yra išversti į lietuvių kalbą. SBVR taisyklės yra skirstomos į elgesio/operacinės (ang. *behavioral*) ir struktūrinės (ang. *structural*) taisykles. Struktūrinės taisyklės nurodo, kad pritaikius apibrėžtą taisyklę, teiginys visuomet turėtų būti teisingas panaudojant individualius konceptus atitinkamo bendrinio koncepto vietoje [9]. Tai duomenų objektų apribojimus aprašančios taisyklės. Visi veiklos terminų apribojimai yra priskiriami struktūrinėms veiklos taisyklėms. Struktūrinės veiklos taisyklės pavyzdys būtų:

Privaloma, kad [asmuo išsinuomoja 1 automobilį](#).

Operacinės veiklos taisyklės indikuoja, kad žmonės arba organizacijos yra įpareigosos arba uždraudžiamos atlikti tam tikrus veiksmus [9]. Šios taisyklės aprašo dinامينius veiklos aspektus ir nurodo, kas gali būti atlikta su veiklai priskiriamais objektais esant nurodytoms sąlygoms. Operacinės veiklos taisyklės leidžia apibrėžti sąlygas, kurių laikantis veikla yra vykdoma. Taisyklės apibrėžia duomenų būsenas, kurios yra leidžiamos, ir užtikrina, kad vykdomi veiklos procesai pakeitę duomenų būsenas išlaiko jas teisingomis. Operacinės veiklos taisyklės pavyzdys būtų:

Draudžiama, kad [užsakymas apmokamas vekseliu](#).

Jeigu tokios taisyklės neegzistuoja – užsakymą būtų galima apmokėti visais apmokėjimo būdais. Ši elgesio taisyklė apriboja veiksmų laisvę ir neleidžia apmokėti užsakymus vekseliu. Šią taisyklę galima suprasti ir kaip:

Privaloma, kad [užsakymas apmokamas visais apmokėjimų būdais išskyrus vekseliu](#).

Pagal nutylėjimą yra priimta, kad taisyklės netiesiogiai nurodo kas yra leidžiama. Tai kas taisyklėmis nėra uždraudžiama – yra leidžiama.

#### 1.3.3.2.1. SBVR veiklos taisyklių modališkumai

SBVR yra lingvistika grindžiamas standartas, kuris palaiko lingvistinius modališkumus. Modališkumai leidžia išreikšti bendrinius autoriaus ketinimus [13]. SBVR kontekste aletinė modalinė logika atitinka struktūrines taisykles. Aletinio modalumo poliškumai pavaizduoti 1.2 lentelėje. Contingency modalumas yra naudojamas kaip patarimas, kuris siekia panaikinti išankstinį nusistatymą vykdomoje veikloje, kuris gali atsirasti dėl panašioje veiklos srityje esamų ir naudojamų taisyklių, tačiau esamoje veiklos srityje tokiai taisyklei nesant aprašyti. Pavyzdžiui

Įmanoma, bet nėra privaloma, kad [nuomotojo amžius būtų mažiau už 21 metus](#)

Ši taisyklė nurodo, kad vis dėlto yra įmanoma, kad nuomotojo amžius būtų mažesnis negu 21 metai, nors galbūt panašioje veikloje egzistuoja taisyklė, kuri draudžia tapti nuomininku, jeigu amžius yra mažesnis negu 21 metai. Šis modališkumas nėra nei teigiamo, nei neigiamo poliškumo.

1.2 lentelė. Aletinio modalumo SBVR taisyklėse poliškumai

Modališkumas	Teigiamas poliškumas	Neigiamas poliškumas	Neutralus poliškumas
<i>Necessity</i>	Privaloma, kad	Nėra privaloma, kad	
<i>Possibility</i>	Gali būti, kad Įmanoma, kad	Nėra įmanoma, kad Yra neįmanoma, kad Neįmanoma, kad	
<i>Contingency</i>			Įmanoma, bet nėra privaloma, kad

SBVR kontekste deontinė modalinė logika atitinka operacines taisykles. Deontinio modalumo poliškumai pavaizduoti 1.3 lentelėje. *Optionality* modališkumas yra naudojamas kaip patarimas, kuris siekia panaikinti išankstinį nusistatymą operacinėse veiklos taisyklėse, ir kuris gali būti numanomas, tačiau nėra aiškiai aprašytas. Dažnai numanomos operacinės taisyklės atsiranda dėl to, kad panašioje veikloje yra aprašytos ir taikomos tokios operacinės taisyklės. Pavyzdžiui

Leidžiama, bet nėra įpareigojama, kad nuomininkas parodytų tapatybės dokumentą prieš pasiimant automobilį

Ši taisyklė nurodo, kad patvirtinantį dokumentą galima parodyti, bet apibrėžtos taisyklės to nereikalauja norint vykdyti veiklą. Šis modališkumas nėra nei teigiamo nei neigiamo poliškumo.

**1.3 lentelė.** Deontinio modalumo SBVR taisyklėse poliškumai

Modališkumas	Teigiamas poliškumas	Neigiamas poliškumas	Neutralus poliškumas
<i>Obligation</i>	Įpareigojama, kad	Nėra įpareigojama, kad Neįpareigojama, kad	
<i>Permission</i>	Leidžiama, kad	Nėra leidžiama, kad Neleidžiama, kad Draudžiama, kad	
<i>Optionality</i>			Leidžiama, bet nėra įpareigojama, kad

Aletiška modalinė logika skiriasi nuo deontinės modalinės logikos tuo, kad aletiška logika yra susijusi su žmogišku įvertinimu apie įmanomą teiginio tiesą, o tuo tarpu deontinė modalinė logika apima žmonių grupės socialinio geidžiamumo įvertinimą, kad tam tikras teiginys yra teisingas [9]. Tai nurodo, kad deontinė logika reikalauja tam tikro konteksto, kuriame teiginys tampa teisingu arba neteisingu [14].

### 1.3.3.2.2. SBVR patarimai

SBVR turi priešingą taisyklėms konceptą, kuris yra vadinamą patarimais. Taisyklė visada potencialiai panaikina tam tikrą veiksmų laisvę, o patarimas primena, kad tam tikra veiksmų laisvė egzistuoja arba yra leidžiama. Patarimo pavyzdys galėtų būti:

Yra leidžiama užsakymą apmokėti kredito kortele

Šis patarimas indikuoja, kad už užsakymą galima apmokėti kredito kortele. Esant tokiam patarimui neturėtų egzistuoti taisyklė, kuri būtų prieštaringa tokiam patarimui.

Nors SBVR kontekste patarimai neturi didelės reikšmės, kadangi jie neapriboja galimų veiksmų aibės, o tiesiog primena apie tam tikrą galimą veiksmą – patarimus yra naudinga fiksuoti veiklos kontekste, nes taisykles/patarimus skaito ir taiko darbuotojai, kuriems draudimų gausa gali užgožti paliekamus galimus veiksmus. Patarimai šiuo atveju yra naudojami kaip priminimai darbuotojui.

### 1.3.3.2.3. SBVR išimtis ir leidimai

SBVR pagal nutylėjimą yra priimta, kad tai kas nėra uždrausta su taisykle – leidžiama. Veiklų praktikos patvirtina, kad šis pasirinkimas yra tinkamas ir panaudojamas daugelyje veiklų problemų sprendimams. Yra tam tikrų sričių ir veiklų zonų, kuriose tokia praktika nėra tinkama dėl labai jautrios veiklos aplinkos. Tai gali būti susiję su labai jautriais, pavojingais arba vertingais resursais. Dėl šios priežasties buvo reikalingas mechanizmas, leidžiantis suteikti prieigą tik aprašius tai taisykle. Tokios taisyklės yra vadinamos leidimais. SBVR kontekste leidimas turi būti tikslus ir aiškus, nes be jo nėra atvirkštinio leidimo veiksmų atlikimui. SBVR leidimai yra konkretesnis konceptas, kuris papuola po SBVR išimtimis. SBVR išimtis yra išskyrimo veiksmas iš apribojimų. Be išimties veiksmas būtų apribotas. Išimtis ir leidimai nurodo, kad tam tikri veiksmai uždraustoje veikloje vis dėlto gali įvykti. Kitaip sakant išimtis ir leidimai nėra prieštaravimai bendrinei taisyklei. Išimčių ir leidimų palaikymas atliekamas tokia tvarka:

1. yra aprašoma veiklos taisyklė, kuri apibūdina tam tikrą veiklos sritį kaip draudžiamą, išskyrus aprašytas išimtis arba leidimus. Kitaip sakant - aprašoma draudžiama veiklos sritis tam, kad panaikinti pagal nutylėjimą priimtą SBVR nuostatą: tai kas neuždrausta yra leidžiama;
2. tikslus leidimas arba išimtis yra aprašyta tam, kad deklaruoti išimtį iš bendrai uždraustos veiklos srities. Be tokio leidimo arba išimties nebūtų galima kažką daryti apibrėžtoje veiklos srityje.

Leidžiama aprašyti neribojamą skaičių išimčių, kurios nusveria bendrinę taisyklę nesukeliant prieštaravimų, tačiau bendrinė taisyklė privalo leisti aprašinėti išimtis. 1.8 pav. yra pavaizduotas pavyzdys, kuriame aprašomas veiksmo leidimas.

#### EXAMPLE

Two guidance statements, expressing a general rule and a more specific case for EU-Rent:

##### Vehicle Usage Rule

A vehicle may use an ice road only if the use is authorized by a Vehicle Usage Advice.

##### Arctic Circle Exemption

Any ice road that is north of the Arctic Circle may be used by any vehicle.

The Arctic Circle Exemption is a Vehicle Usage Advice.

1.8 pav. Leidimo atlikti veiksmą pavyzdys

Pirmoji taisyklė yra bendrinė, kuri veiklos erdvę apriboja ir neleidžia jokiai transporto priemonei naudotis ledo keliu, jeigu leidimas nėra suteiktas:

transporto priemonė gali naudotis ledo keliu tik tuomet, kai naudojimas yra leidžiamas transporto priemonės naudojimo patarime.

Tokiu būdu yra limituojami visi veiksmai veiklos srityje. Kita taisyklė aprašo, kad visi ledo keliai, kurie yra į šiaurę nuo poliarinio rato, gali būti naudojami visomis transporto priemonėmis:

##### Poliarinio rato išimtis

ledo kelias, kuris yra į šiaurę nuo poliarinio rato gali būti naudojamas visomis transporto priemonėmis.

Atrodytų, kad dviejų taisyklių pakanka norint aprašyti leidimą duotoje situacijoje, tačiau reikėtų atkreipti dėmesį į pirmos taisyklės įgaliojimo patarimą. Antroji taisyklė, kuri aprašo leidimą, nėra priskirta patarimui, kuris suteikia įgaliojimą. Šiam tikslui pasiekti yra aprašoma trečioji taisyklė, kuri nurodo, kad antroji taisyklė (išimtis) yra transporto priemonės naudojimo patarimas:

Poliarinio rato išimtis yra transporto priemonės naudojimo patarimas.

#### 1.3.3.2.4. SBVR taisyklių kardinalumai

SBVR taisyklėse galima nurodyti kardinalumus, kurie apibrėžia galimų pasikartojimų skaičių. Tai gali būti konkreti kiekybinė išraiška, intervalas arba intervalas su vienu begaliniu režiu (*min N*). Kardinalumo SBVR taisyklėje pavyzdys yra pavaizduotas 1.9 pav.

Privaloma, kad asmuo dirba bent 3 organizacijose

#### 1.9 pav. SBVR taisyklių kardinalumo pavyzdys

SBVR apibrėžti galimi kardinalumų tipai:

1. egzistencinis kardinalumas (*min 1*);
2. bent N kartų (*min N*);
3. maksimaliai 1 kartą (*max 1*);
4. maksimaliai N kartų (*max N*);
5. lygiai vieną kartą (1);
6. lygiai N kartų (N);
7. bent N kartų ir daugiausia M kartų (*min N, max M*);
8. daugiau negu (*min 2*);

#### 1.3.3.3. SBVR žodynų ir žodyno įrašų aprašai

Toliau pateikiami žodynų ir žodyno įrašų sudarymo principai ir pavyzdžiai. 1.10 pav. yra pavaizduoti žodyno galimi atributai. 1.4 lentelėje yra surašyti žodyno aprašo atributai su paaiškinimais kam kiekvienas atributas yra naudojamas.

#### 1.4 lentelė. Žodyno aprašo atributai su paaiškinimais

Atributas	Paaiškinimas
<i>Vocabulary Name</i>	Žodyno pavadinimas.
<i>Description</i>	Naudojamas užfiksuoti žodyno sritį ir paskirtį.
<i>Source</i>	Naudojamas, jeigu žodynas aprašomas remiantis formaliai aprašytu darbu, ir šioje vietoje nurodomas formalaus aprašo šaltinis.
<i>Speech Community</i>	Naudojamas nurodyti kalbos bendruomenę, kuriai aprašomas žodynas ir kuri yra už jį atsakinga.
<i>Language</i>	Naudojama nurodyti kalbą, kuria aprašoma žodyno įrašai. Kalbų pavadinimai yra naudojami iš ISO 639-2 [15]. Jeigu nenurodyta jokia kalba – anglų kalba yra naudojama pagal nutylėjimą.

<i>Included Vocabulary</i>	Naudojamas, kai kitas žodynas yra įtrauktas į aprašomą žodyną.
<i>Note</i>	Naudojamas paaiškinamiesiems tekstams, kurie yra netinkami po kitomis antraštėmis.

<Vocabulary Name>

Description:

Source:

Speech Community:

Language:

Included Vocabulary:

Note:

**1.10 pav.** Žodyno aprašas

Žodyno įrašai aprašo vieną konceptą. Įrašas pradedamas nuo pagrindinės sąvokos paminėjimo, kuri gali būti paskiriamoji arba veiksmožodinis konceptas. 1.11 pav. yra pavaizduotas žodyno įrašo formatas. 1.5 lentelėje yra surašyti žodyno įrašo galimi atributai su paaiškinimais kam kiekvienas atributas yra naudojamas.

Pagrindinis aprašas (ang. *primary representation*) yra žodyno įrašo išraiška. Išraiška gali būti formuojama kaip terminas, pavadinimas arba veiksmožodinis konceptas.

**1.5 lentelė.** Žodyno įrašo aprašo atributai su paaiškinimais

<b>Atributas</b>	<b>Paaiškinimas</b>
<i>Definition</i>	Formalus koncepto apibrėžimas
<i>Source</i>	Nurodo žodyną kaip šaltinį arba dokumentą nurodytam konceptui.
<i>Dictionary Basis</i>	Nurodo apibrėžimą iš bendro žodyno, kuris papildo žodyno įrašo naudojimą.
<i>General Concept</i>	Naudojamas, kai siekiama nurodyti bendrinį konceptą esamam įrašui. Nenaudojamas, jeigu <i>definition</i> atribute yra paminėtas bendrinis konceptas.
<i>Concept Type</i>	Naudojamas, kai norima nurodyti įrašo koncepto tipą. Nenaudojamas, jeigu tipas yra akivaizdus ir kyla iš pagrindinio aprašo, pavyzdžiui, jeigu pagrindinis aprašas yra iš vieno žodžio, tuomet automatiškai nustatomas <i>general concept</i> tipas. Jeigu pagrindinis aprašas yra iš trijų žodžių, tuomet nustatomas <i>verb concept</i> tipas.
<i>Necessity</i>	Naudojamas, norint išreikšti būtinas sąlygas esamam įrašui. Šiam tikslui gali būti panaudotas laukelis <i>Definition</i> .
<i>Possibility</i>	Naudojamas, norint išreikšti galimybes, galimas aplinkybes esamam įrašui.
<i>Reference scheme</i>	Naudojamas, siekiant nurodyti įrašo aplinką arba priskyrimą sričiai.

<i>Note</i>	Naudojamas, siekiant pateikti paaiškinamąjį tekstą, kuris netinkamas kituose atributuose.
<i>Examples</i>	Naudojamas pateikiant įrašo naudojimo pavyzdžius.
<i>Synonym</i>	Naudojamas, nurodant kitą įrašą, kuris gali būti lygiareikšmiai sukeistas su esamu įrašu nepakeičiant prasmės.
<i>Synonymous Form</i>	Naudojamas vietoje <i>Synonym</i> , kai įrašas yra veiksmožodinis konceptas.
<i>See</i>	Nurodomas aprašant sinoniminę formą ir rodo į pagrindinę žodžio formą. <i>SBVR</i> specifikacijoje vienas terminas visada būna pagrindinis, o kiti – sinonimai. Naudojamas kaip pakaitinis, kai esamas pagrindinis aprašas yra netinkamas atvaizdavimui.
<i>Subject Field</i>	Naudojamas, kai įrašas nėra unikalus esamame žodyne ir reikia papildomo įrašo apibūdinimo, kuris priskirtų tam tikram reikšmių laukui. Tokiu būdu vienodi bendriniai aprašai gali būti suprantami kaip skirtingi konceptai.
<i>Namespace URI</i>	Naudojamas, kai esamas įrašas yra skirtas apibūdinti reikšmių sritį arba žodyną. Tokiu atveju, šiame attribute atitinkamai yra nurodoma reikšmių srities arba žodyno URI.

Pagal *SBVR* specifikaciją, daugelis atributų yra nurodomi kaip nepasikartojantys kiekviename žodyno įrašė. Pasikartojantys atributai gali sukelti prieštaravimų interpretuojant konceptą. Kompiuterinės sistemos nežinotų, kuris atributas turėtų būti interpretuojamas ir naudojamas. Galimi pasikartojantys atributai *SBVR* žodyno įrašuose:

1. *general Concept* atributas gali kartotis. Jis logiškai susieja tipus. Įrašas, turintis kelis *general concept* atributus, įgauna visas susietų konceptų ypatybes;
2. *synonym* atributai gali kartotis žodyno įrašė. *Synonym* atributas yra naudojamas pagrindinėje įrašo formoje ir nurodo galimus įrašo sinonimus. Pagrindinės įrašo formos ir sinoniminės įrašo formos kardinalumas yra 1 : 0..\*;
3. *example* atributas gali kartotis, nurodant daugiau nei vieną įrašo pavyzdį;
4. *necessity* atributas gali kartotis, norint išreikšti visas būtinas sąlygas;
5. *possibility* atributas gali kartotis, norint išreikšti visas galimas sąlygas;

Žodyne pagrindinio aprašo pasikartojimai yra galimi, jeigu yra naudojamas *Subject* įrašo atributas. *Subject* atributas yra naudojamas, kai įrašas nėra unikalus esamame žodyne ir reikia papildomo įrašo apibūdinimo, kuris priskirtų tam tikram reikšmių laukui.

### <primary representation>

Definition:  
Source:  
Dictionary Basis:  
General Concept:  
Concept Type:  
Necessity:  
Possibility:  
Reference Scheme:  
Note:  
Example:  
Synonym:  
Synonymous Form:  
See:  
Subject Field:  
Namespace URI:

1.11 pav. Žodyno įrašo aprašas

#### 1.3.4. Prieštaringumas

Prieštaringumas (ang. *contradiction*) – tai situacija, kurioje du teiginiai logiškai prieštarauja arba paneigia kitą paskelbtą teiginį arba save patį. Tokia situacija nutinka, kai vienu metu paimti du teiginiai priveda prie dviejų išvadų, kurios formuoja inversiją viena kitai [16]. *SBVR* kontekste tai taisyklės arba žodyno įrašai, kurie egzistuojami tuo pačiu metu sukelia prieštarumą vienas kitam ir jų abiejų patenkinti vienu metu yra neįmanoma. Toliau yra aptariama, kaip semantiniai teiginiai gali būti skirstomi ir interpretuojami, kad būtų įmanoma aptikti prieštarumus. 2012 metais buvo publikuotas straipsnis [17], kuriame bandyta atrasti prieštarungus teiginius 600 milijonų internetinių puslapių. Teigiama, kad prieštarungų atradimo tikslumas buvo virš 70 procentų naudojant tik 46 pradines (ang. *seed*) reikšmes. Nors straipsnyje buvo nagrinėjama japonų kalba, tačiau jame panaudota logika gali būti pritaikoma anglų, lietuvių ir kitoms kalboms. Visų pirma yra apsibrėžiami šablonai, pagal kuriuos teiginiai yra skirstomi:

1. suaktyvinantis šablonas (ang. *excitatory*) – nurodo teiginius, kurie įjungia, aktyvuoja arba pagerina. Pavyzdžiui:
  - a. sukelti X;
  - b. nupirkti X;
  - c. gaminti X;
  - d. importuoti X;
  - e. padidinti X;
  - f. įjungti/įgalinti X;
2. slopinantis šablonas (ang. *inhibitory*) – nurodo teiginius, kurie išjungia, de-aktyvuoja arba slopina. Pavyzdžiui:
  - a. neleisti X;

- b. išmesti X;
  - c. sumažinti X;
  - d. išjungti X;
3. neutralus šablonas – nurodo teiginius, kurie nepriskiriami nei suaktyvinimo, nei slopinimo šablonui. Pavyzdžiui:
- a. apsvarstyti X;
  - b. proporcingas X;
  - c. susijęs su X;
  - d. įvertinti X;
  - e. artimas X.

X pavyzdžiuose nurodo veiksmą. Pateikti pavyzdžiai nenurodo visos įmanomos aibės teiginių ir jų formuluočių. Pavyzdžiai pateikiami siekiant parodyti šablonų esmę. Teiginiai, priskiriami neutraliam šablonui, negali būti lyginami ir naudojami darant išvadas apie teiginių prieštarumą, todėl yra nagrinėjami suaktyvinimo ir slopinimo šablonai, kurie bendrai yra vadinami sužadinaisiais (ang. *excitation*) šablonais. Suaktyvinantis ir slopinantis šablonai sukuria du priešingus poliškumus, kurie nepriklauso nuo teigiamų ar neigiamų formuluočių, pavyzdžiui rafinuotas susitikimas ir apsunkinti argumentą. Abu teiginiai yra suaktyvinimo teiginiai, nes žodis „rafinuoti“ reiškia įmantrumą, išsilavinimą (pagerinimas), o žodis „apsunkinti“ turi reikšmę padidinti sunkumą, kas taip pat nurodo suaktyvinimo šabloną, nors pati sakinio reikšmė yra negatyvi (pabloginti situaciją). Prieštarumų atradimui yra daroma prielaida, kad du prieštaringi teiginiai turės priešingo poliškumo šablonus, tą patį daiktavardį ir teiginių predikatinę kalbos dalis bus tokia pati [17]. Toliau yra apžvelgiama *SBVR* specifikacija ir kaip prieštarumų aptikimą galima pritaikyti struktūrizuotai *SBVR* kalbai taisyklėse bei žodynų įrašuose.

### 1.3.5. Prieštarumai *SBVR* specifikacijoje

*SBVR* specifikacijose prieštarumas gali būti suprantamas ne tik kaip semantinis taisyklių tarpusavio prieštarumas. Šio darbo kontekste *SBVR* specifikacijų prieštarumas yra suprantamas kaip dviprasmybę sukeliantys įrašai arba *SBVR* taisyklės, kurios yra prieštaringos viena kitai ir vienu metu negali būti patenkinamos. Dviprasmybės *SBVR* specifikacijoje aptinkamos *SBVR* žodynų ir žodynų įrašų aprašuose. Pavyzdžiui esant pasikartojantiems atributams, kurie kartotis negali viename žodyno įrašė, nėra aišku, kuris atributas turėtų būti naudojamas. Kitas pavyzdys būtų pasikartojantys įrašai žodyne su vienodomis pagrindinio aprašo reikšmėmis.

Toliau yra aprašomi prieštarumų tipai žodynuose ir veiklos taisyklėse, kurie buvo nustatyti šio darbo metu analizuojant *SBVR* specifikacijas.

### 1.3.6. Išskirti žodyno įrašų prieštarumų tipai

#### 1.3.6.1. Pasikartojantys atributai žodynų įrašuose

Pasikartojantys atributai yra aptinkami tikrinant visus žodyno įrašus ir jų įvestų atributų pavadinimus. 1.3.3.3 skyriuje buvo aptarti atributai, kurie gali kartotis žodynų įrašų atributuose, todėl tikrinimas turėtų juos ignoruoti. Jeigu to paties atributo pavadinimas aptinkamas daugiau negu vieną kartą atitinkamam žodyno įrašui – išvedamas klaidos pranešimas. Pasikartojančio atributo dviprasmybės pavyzdys pavaizduotas 1.12 pav.

```
Lithuania
  General_concept: country
  See: Lietuva
  See: Litva
```

1.12 pav. Pasikartojančio atributo SBVR žodyno įrašė pavyzdys

### 1.3.6.2. Žodynų įrašų *Synonym* ir *See* atributų neatitikimai

*Synonym* atributai žodyno įrašė gali kartotis ir nurodo sinonimines įrašo reikšmes. Sinoniminis įrašas turėtų turėti *See* atributą, kuris nurodo į pagrindinę įrašo formą. Teisingų įrašų kombinacija yra pavaizduota 1.13 pav. Neteisingų įrašų kombinacija yra pavaizduota 1.14 pav. Įrašų neatitikimas sukelia prieštarumą.

```
pastaba
  Synonym: pažyma
pažyma
  See: pastaba
```

1.13 pav. *Synonym* ir *See* atributų atitikimas

```
pastaba
  Synonym: pažyma
pažyma
  See: įrašas
```

1.14 pav. *Synonym* ir *See* atributų neatitikimas

### 1.3.6.3. Pasikartojantys įrašai žodyne

Pasikartojantys įrašai žodyne yra aptinkami tikrinant visus žodyno įrašus ir tikrinant, ar pagrindinis aprašas nėra pasikartojantis. Jeigu žodyno įrašas turi *Subject* laukelį, jo reikšmė yra išsaugojama kartu su pagrindiniu aprašu tam, kad aptikus pasikartojimą būtų galima palyginti atitinkamas *Subject* atributo reikšmes. Pasikartojančio įrašo, kuris sukelia netikslumą, pavyzdys yra pavaizduotas 1.15 pav. Pasikartojančio įrašo, kuris pasinaudoja *Subject* atributu, kad nesukelti dviprasmybių, pavyzdys yra pateikiamas 1.16 pav.

```
Tomas
  General_concept: asmuo
Tomas
  General_concept: pareigūnas
```

1.15 pav. Pasikartojančio įrašo sukeliančio dviprasmybę *SBVR* žodyne pavyzdys

```

Klientas
  Subject_field: "automobilių_nuoma"

Klientas
  Subject_field: "automobilių_prekyba"

```

1.16 pav. Pasikartojančio įrašo pasinaudojant *Subject* atributu pavyzdys

1.15 pav. pavaizduotas individualus konceptas Tomas gali būti aprašytas kaip asmuo ir kaip pareigūnas, tačiau jo aprašymas turėtų būti atliktas vieną kartą ir turėtų turėti dvi bendrinio koncepto reikšmes.

#### 1.3.6.4. Prieštaringų charakteristikų aptikimas

Dvi prieštaringos charakteristikos yra charakteristikos su tokiu pačiu bendriniu konceptu, o jų veiksmožadinės sąvokos yra skirtingo poliškumo. Iš 1.3.4 skyriaus buvo išskirta, kad viena veiksmožadinė sąvoka turėtų būti slopinanti, o kita – sužadinti. Prieštaringos charakteristikos pavyzdys yra pateikiamas 1.17 pav.

```

organizacija yra_pelninga
organizacija yra_nepelninga

```

1.17 pav. Prieštaringų charakteristikų pavyzdys

#### 1.3.6.5. Statinių apribojimų prieštarumai

Prieštaringumai fakte atsiranda tuomet, kai fakte aprašyti individualūs konceptai neatitinka statinio apribojimo. Statinio apribojimo prieštarungumo pavyzdys yra pateiktas 1.18 pav. Šiame pavyzdyje statinis apribojimas yra aprašytas su dviem bendriniais konceptais city ir country. Aprašytas faktas Kaunas is capital of Lithuania sukelia prieštarumą, nes individualus konceptas Kaunas neturi priskirto bendrinio koncepto, kuris atitiktų statinio apribojimo aprašą.

```

city is_capital_of country

Kaunas
Lithuania
  General_concept: country

Kaunas is_capital_of Lithuania

```

1.18 pav. Statinio apribojimo prieštarungumo pavyzdys

#### 1.3.6.6. Rolių neteisingas aprašymas

Rolės, kitaip negu bendriniai konceptai, *SBVR* specifikacijoje gali turėti tik vieną *General\_concept* reikšmę. Netinkamai aprašytos rolės pavyzdys yra pavaizduotas 1.19 pav.

```

amzius
  Concept_type: role
  General_concept: number
  General_concept: text

```

1.19 pav. Rolės prieštaravimo pavyzdys

### 1.3.6.7. Prieštaringas rolės priskyrimas

Rolių priskyrimas bendriniam konceptui yra atliekamas pasinaudojant specialiais raktažodžiais *turi* arba *has*. Tam, kad būtų galima priskirti rolę bendriniam konceptui, rolės žodyno įrašas privalo turėti aprašytą `Concept_type: role`. Prieštaringas rolės priskyrimas bendriniam konceptui yra pavaizduotas 1.20 pav.

```

data
  Concept_type: general_concept

asmuo turi data

```

1.20 pav. Prieštaringas rolės priskyrimas

### 1.3.6.8. Kategorizavimo schemų prieštaringumas

*SBVR* kategorizacijos yra nepersidengiančios (ang. *disjoint*), todėl nėra leidžiama aprašyti žodyno įrašo, kuris priklausytų dviem ar daugiau skirtingų kategorijų, aprašytų vienoje kategorizacijos schemoje. Šis prieštaringumas gali būti aptinkamas ir *SBVR* segmentacijos aprašuose, nes juose taip pat galioja kategorijų nepersidengiamumo taisyklė. *SBVR* kategorizacijos prieštaringumo pavyzdys yra pavaizduotas 1.21 pav.

```

renginys
  ⊖ mokėjimo_tipas
    Concept_type: categorization_type
    Necessity: is_for general_concept renginys

  ⊖ Įvykiai_pagal_mokėjimo_tipą
    Necessity: segmentation for general_concept renginys that subdivides renginys by mokėjimo_tipą

  ⊖ nemokamas_renginys
    General_concept: renginys
    Necessity: is_included_in Įvykiai_pagal_mokėjimo_tipą

  ⊖ mokamas_renginys
    General_concept: renginys
    Necessity: is_included_in Įvykiai_pagal_mokėjimo_tipą

  ⊖ Jazzu_renginys
    General_concept: mokamas_renginys
    General_concept: nemokamas_renginys

```

1.21 pav. *SBVR* kategorizacijos prieštaringumo pavyzdys

### 1.3.6.9. Cikliškumas konceptų aprašyme

Aprašomi konceptai *SBVR* žodyne gali turėti nurodytą *general concept*, kuris nurodo bendrinį konceptą esamam įrašui. Cikliškumo prieštarīgumas atsiranda aprašytuose konceptuose, jeigu atsekant visus esamo koncepto bendrinius konceptus yra sutinkamas esamas konceptas. Cikliškumas konceptų aprašyme yra pavaizduotas 1.22 pav.

```
asmuo
  General_concept: organizatorius

organizatorius
  General_concept: asmuo
```

1.22 pav. *SBVR* žodyno įrašų cikliškumas

### 1.3.7. Išskirti *SBVR* taisyklių prieštarīgumo tipai

#### 1.3.7.1. Kardinalumo neatitikimas

Jeigu dvi taisyklės aprašytos naudojant tokį patį veiksmažodinį konceptą, tačiau jų kardinalumų apibrėžiami daugialypiškumo režiai neatitinka, tai lemia prieštarīgumą. Kardinalumo prieštarīgumo pavyzdys yra pavaizduotas 1.23 pav.

```
Privaloma, kad asmuo dirba bent 3 organizacijose.
Privaloma, kad asmuo dirba daugiausiai 2 organizacijose.
```

1.23 pav. Kardinalumo prieštarīgumo pavyzdys

Kiekvieną *SBVR* kardinalumą galima išreikšti intervalu. Išraiškos yra pavaizduotos 1.6 lentelėje. Aštuonios skirtingos kardinalumo sąvokos sukuria galimybę 64 skirtingiems prieštarīgumų tipų aptikimams *SBVR* taisyklėse. Supaprastintai galima teigti, kad dvi taisyklės, kurios turi sutampančius bendrinius konceptus ir veiksmažodinius konceptus yra prieštarīgos taisyklės, jeigu jų apibrėžtų kardinalumų sąvokų intervalinės reikšmės apibrėžia skirtingus, nepersidengiančius, begalinius intervalus arba taisyklių apibrėžti baigtiniai intervalai nesutampa.

#### Nepersidengiančio intervalo prieštarīgumas

Taisyklės, kurios yra aprašytos pasinaudojant tokiu pačiu veiksmažodiniu konceptu ir kurių kardinalumų sąvokų reikšmės apibrėžia skirtingus nepersidengiančius intervalus. Taisyklė, kurios apibrėžtas intervalas yra  $[3; +\infty)$ :

```
Privaloma, kad asmuo dirba bent 3 organizacijose.
```

Taisyklė, kurios apibrėžtas intervalas yra  $[0; 2]$ :

```
Privaloma, kad asmuo dirba daugiausiai 2 organizacijose.
```

Šios taisyklės sudaro du nepersidengiančius intervalus:  $[0; 2]$  ir  $[3; +\infty)$ , todėl jos yra prieštarīgos viena kitai.

### Nesutampančio intervalo prieštarīgumas

Taisyklės, kurių kardinalumų sąvokų reikšmės apibrėžia baigtinius nesutampančius intervalus. Taisyklė, kurios apibrėžtas intervalas yra [2; 5]:

Privaloma, kad asmuo *dirba* mažiausia 2 ir daugiausia 5 organizacijose.

Taisyklė, kurios apibrėžtas intervalas yra [3; 6]:

Privaloma, kad asmuo *dirba* mažiausia 3 ir daugiausia 6 organizacijose.

Šios taisyklės sudaro du skirtingus baigtinius intervalus. Nors šie intervalai yra persidengiantys, tačiau sukuria dviprasmiškumą ir gali būti perrašytos vienu intervalu, todėl tokios taisyklės yra traktuojamos kaip prieštarīgotos.

### Persidengiančio intervalo pavyzdys

Taisyklės, kurių kardinalumų sąvokų reikšmės apibrėžia persidengiančius intervalus. Taisyklė, kurios apibrėžtas intervalas yra [2; +∞):

Privaloma, kad asmuo *dirba* bent 2 organizacijose.

Taisyklė, kurios apibrėžtas intervalas yra [0; 5]:

Privaloma, kad asmuo *dirba* daugiausiai 5 organizacijose.

Šios taisyklės sudaro persidengiantį intervalą [2; 5], kuris tenkina abi taisykles, todėl šios taisyklės yra neprieštarīgotos viena kitai.

1.6 lentelė. *SBVR* kardinalumų ir intervalinių reikšmių suliyginimas

<b><i>SBVR</i> kardinalumas</b>	<b>Intervalinė reikšmė</b>
Egizstencinis ( <i>min</i> 1)	[1; +∞)
Bent <i>N</i> kartų ( <i>min</i> <i>N</i> )	[ <i>N</i> ; +∞)
Maksimaliai 1 kartą. ( <i>max</i> 1)	[0; 1]
Maksimaliai <i>N</i> kartų. ( <i>max</i> <i>N</i> )	[0; <i>N</i> ]
Lygiai vieną kartą. ( <i>1</i> )	[1; 1]
Lygiai <i>N</i> kartų. ( <i>N</i> )	[ <i>N</i> ; <i>N</i> ]
Bent <i>N</i> kartų ir daugiausia <i>M</i> kartų. ( <i>min</i> <i>N</i> , <i>max</i> <i>M</i> )	[ <i>N</i> ; <i>M</i> ]
Daugiau negu ( <i>min</i> 2)	( <i>N</i> ; +∞)

### 1.3.7.2. Veiksmažodinių konceptų prieštarīgumai

Taisyklių modalumas, bendriniai ir individualūs konceptai sutampa, tačiau veiksmažodiniai konceptai sukuria priešingus poliškumus ir tokiu būdu sudaro prieštarīgumą. Tokios konfliktuojančios taisyklės pavyzdys yra nurodytas 1.24 pav.

Privaloma, kad renginys vyksta įvykio vietoje Kauno\_Dramos\_teatre.

Privaloma, kad renginys nevyksta įvykio vietoje Kauno\_Dramos\_teatre.

1.24 pav. Taisyklių veiksmožodinių konceptų prieštarumas

### 1.3.7.3. Modalumo prieštarumas

Taisyklių bendriniai konceptai, individualūs konceptai ir veiksmožodiniai konceptai sutampa, tačiau modalumas sukuria priešingą poliškumą tarp taisyklių. Tokios konfliktuojančios taisyklės pavyzdys yra nurodytas 1.25 pav.

Privaloma, kad renginys vyksta įvykio vietoje Kauno\_dramos\_teatras

Nėra privaloma, kad renginys vyksta įvykio vietoje Kauno\_dramos\_teatras

1.25 pav. Taisyklių modalumo prieštarumas

Ne visi modališkumą teigiami ir neigiami poliai tarpusavyje sudaro prieštarumą. Aletinės loginės formuluotės ir deontinės loginės formuluotės iš esmės aprašo skirtingus veiklos aspektus, todėl turi būti nagrinėjamos atskirai. Siekiant atrasti prieštarumus taisyklių modališkumuose yra tikrinami atitinkamų modališkumą poliškumai. Jeigu taisyklėje atitinka modališkumas, bendriniai konceptai, individualūs konceptai, veiksmožodinės formos, tačiau modališkumo poliai yra priešingi, tokios taisyklės sukuria prieštarumą.

### 1.3.7.4. Rolių prieštarumas

Rolės *SBVR* žodyne aprašomos pasinaudojant žodeliu „*turi*“, kuris priskiria aprašytą rolę bendriniam konceptui kaip atributą. Aprašyti rolei galima nurodyti taisykles, kurios apriboja galimas reikšmes. Nurodant tokias taisykles gali atsirasti prieštarumų dėl priskiriamų reikšmių tipų. Asmens viena iš rolių yra pavardė. Šis konceptas žodyne aprašomas taip:

pavardė

Concept\_type: role

General\_concept: text

asmuo

asmuo turi pavardę

Šiam žodyno konceptui prieštaraujanti veiklos taisyklė yra pavaizduota 1.26 pav.

Privaloma, kad asmuo turi pavardę lygų 1

1.26 pav. Rolės prieštarumas

Taisyklė bando prilyginti `pavardės` atributą skaitinei reikšmei. Žodyne aprašyta rolė nurodo, kad `pavardė` yra tekstinė reikšmė, todėl tai sukelia prieštarumą.

### 1.3.7.5. Veiksmažodinio koncepto taisyklės prieštarumas

Jeigu *SBVR* žodyne yra aprašytas veiksmažodinis konceptas ir konceptui yra aprašyta taisyklė, apribojanti galimų esybių skaičių, žodyne neturėtų būti aprašyti faktai, kurie prieštarautų taisyklei. Veiksmažodinio koncepto taisyklės prieštarumas yra pavaizduotas 1.27 pav. - 1.29 pav.

```
užsakymas
klientas

Užsakymas1
  General_concept: užsakymas

Klientas1
  General_concept: klientas

Klientas2
  General_concept: klientas
```

1.27 pav. *SBVR* veiksmažodinio koncepto taisyklės prieštaravimo žodyno konceptų aprašymas

```
Privaloma, kad užsakymas turi daugiausiai 1 klientą
```

1.28 pav. *SBVR* veiksmažodinio koncepto prieštaravimo taisyklės aprašymas

```
Užsakymas1 turi Klientas1
Užsakymas1 turi Klientas2
```

1.29 pav. *SBVR* veiksmažodinio koncepto taisyklės prieštarumą sukeliančys faktai

*SBVR* žodyne sukuriama bendriniai konceptai `užsakymas`, `klientas` ir individualūs konceptai `Klientas1`, `Klientas2`, `Užsakymas1`. *SBVR* taisyklių faile aprašoma taisyklė, kuri nurodo, kad konkrečiam užsakymui gali būti priskirtas daugiausiai vienas konkretus klientas. *SBVR* žodyne aprašomi faktai, kurie sukuria kardinalumo prieštarumą anksčiau aprašyti taisyklei.

## 1.4. Veiklos žodynų ir taisyklių neprieštaravimo tikrinimo naudotojų analizė

*SBVR* naudotojai yra veiklų bendruomenės, kuriose *SBVR* taisyklės įtakoja elgseną ir veiklos apribojimus, aprašo reagavimus į besikeičiančią aplinką ir įvykius. Taisyklės yra pagrindinis įrankis, kuriuo organizacijos naudojasi siekdamas valdyti veiklas. Aprašomos taisyklės yra aiškus ir struktūrizuotas būdas valdyti veiklą [8]. *SBVR* naudotojai yra ir informacinių technologijų profesionalai, kuriems *SBVR* taisyklės padeda suprasti veiklos taisykles ir prisideda prie lengvesnio modelių kūrimo. *SBVR* taisyklės taip pat padeda komunikuoti veiklos duomenų vadybininkams ir programinės įrangos kūrėjams.

Prieštarumų ir klaidų tikrinimo *SBVR* specifikacijose naudotojai potencialiai yra visi esami *SBVR* specifikacijos naudotojai, kurie aprašinėja veiklos žodynus ir taisykles. Didėjant *SBVR* žodyno ir taisyklių kiekiui, esamiems naudotojams darosi vis sunkiau užtikrinti žodynų ir taisyklių teisingumą bei neprieštarumą. Dažniausiai klaidos aptinkamos perkėlus taisykles į kompiuterizuotas sistemas

arba pradėjus vykdyti veiklą ir pastebėjus neatitikimus su norimais rezultatais. Prieštarinamumą ir klaidų tikrinimas *SBVR* specifikacijose visiems naudotojams suteiktą galimybę aptikti klaidas žodynu ir taisyklių aprašymo etape. Klaidų aptikimas potencialiai pritrauktų daugiau naujų *SBVR* naudotojų dėl sumažėjusio sudėtingumo korektiškumo užtikrinimui.

### **1.5. Esamų *SBVR* prieštarinamumo tikrinimo metodų analizė**

Šiame skyriuje analizuojami esami problemos sprendimo metodai, kurie tiesiogiai bando aptikti *SBVR* veiklos žodyno ir taisyklių prieštarinamumus arba galėtų būti pritaikyti prieštarinamumų aptikimui.

#### **1.5.1. Nuo veiklos srities nepriklausantis metodas prieštarinamumų aptikimui *SBVR* paremtose veiklos taisyklėse [18]**

Šiame darbe remiamasi pirmosios eilės logika [19] ir naudojamosi patenkinamumo teorijos [20] sprendėjais (ang. *solvers*), kad būtų surasti nesuderinamumai aprašytose veiklos taisyklėse. Darbe yra aprašoma 19 skirtingų sąsajų tarp *SBVR* ir *SMT-LIBv2* [21], tačiau teigiama, kad jų sukurtame prototipe *BuRRiTo* realizuota 40 sąsajų tarp *SBVR* ir *SMT-LIBv2* ir žadama realizuoti daugiau ateityje. Aprašytos sąsajos palaiko konceptų hierarchijas (specializacijas, sinonimus). Taip pat yra aprašomos sąsajos, kurios palaiko *SBVR* unarinius, binarinius ir daugialypius faktų aprašus su atitinkamu parametru skaičiumi. Darbe taip pat minimos sąsajos su loginiais operatoriais, tokiais kaip: ir, arba, ne, jeigu, tuomet, panaudojant logines sąsajas. Kiekybiniai įvertinimai, tokie kaip kiekvienas, egzistuoja yra realizuoti panaudojant *forall* ir *exists* komandas *SMT-LIBv2*. Ribojamieji kiekybiniai įvertinimai, tokie kaip: lygiai  $n$  kartų, bent  $n$  kartų, ne daugiau  $n$  kartų, yra palaikomi pasinaudojant predikatais ir funkciniais iškvietimais. Tai vienas iš nedaugelio darbų, kuris sudaro sąlygas *SBVR* veiklos taisyklių prieštarinamumo aptikimo automatizavimui.

#### **1.5.2. Eksperimentinis transformacijų iš *SBVR* veiklos žodyno ir taisyklių į *OWL2* ontologijas tyrimas [22]**

Tai naujausias darbas iš keturių darbų sekos, kurie analizuoja *SBVR* ir *OWL* metamodelius siekiant apibrėžti įmanomas transformacijas. Jau pirmajame darbe [23] yra aprašomos keturios *SBVR* konceptų transformacijos į *OWL2*, tačiau yra minima, kad daugialypiai faktų tipai reikalauja specialaus apdorojimo, nes nėra galimybės konvertuoti į *OWL2* ontologiją. Antrasis darbas [24] aprašo 23 papildomas transformacijas ir pateikia transformacijų pavyzdį, tačiau darbe taip pat minimi papildomi reikalavimai *SBVR* taisyklėms, kad būtų užtikrinti pastovūs transformacijų rezultatai, kas pradeda limituoti *SBVR* išraiškumą. Vienas iš tokių apribojimų pavyzdžių yra tai, kad norint turėti atvirkštinę savybę *OWL2* žodyne, privaloma aprašyti aletišką (ang. *alethic*) *SBVR* taisyklę. Trečiasis darbas [25] kaip ir antrasis aprašo papildomas transformacijas į *OWL2* su paaiškinimais, tačiau šis darbas fokusuojasi į galimybę konvertuotas *OWL2* ontologijas atversti atgal į *SBVR* be informacijos praradimo. Šiame darbe taip pat yra pateikiami apribojimai *OWL* aksiomų savybėms norint išvengti ciklinių priklausomybių. Nors darbuose tiesiogiai nekalbama apie prieštarinamumų aptikimą, tačiau *OWL2* redagavimo įrankiai tokie kaip *Protege* [26] turi argumentuotojus (ang. *reasoner*), kurie gali aptikti ontologijų prieštarinamumus.

### 1.5.3. „Logika paremtas sprendimas“ *SBVR* neprieštaringumo užtikrinimui [27]

Nors darbe minimas *SBVR* standartas, tačiau darbas naudoja *ORM2* [28] reprezentaciją vietoje *SBVR*. *SBVR* ir *ORM2* yra panašūs [4] iš pagrindų, todėl galima įtraukti darbą į lyginamąją analizę. Darbe yra pristatoma mintis atvaizduoti taisyklės pirmosios eilės deontinė-aletinė logika. Darbe taip pat yra pristatomas įrankis, kuris formalų *ORM2* aprašą transformuoja į *ALCQI*. *ALCQI* yra *OWL2* fragmentas. Siūlomas metodas aprašo nuoseklumo užtikrinimą panaudojant *ALCQI* pakankamumo patikrinimą, kuris savo ruožtu gali būti paverstas nepakankamumo patikrinimu *OWL2* ontologijose. Pirmosios eilės deontinė-atletinė logika turi labai stiprų teorinį pagrindą ir pateiktas metodas turi aiškiai apibrėžtą eiliškumą. *ORM2* naudojimas yra abejotinas, kadangi veiklos taisyklių supratimas gali būti paliktas interpretacijoms. *ALCQI* turi mažiau išraiškos galimybių negu *ORM2* ko pasėkoje daugybė *SBVR* konceptų lieka neištraukti į siūlomo įrankio realizaciją. Tokių konceptų pavyzdžiai yra: dažnumo apribojimai ir jų priskyrimai kelioms rolėms, daugialypių sąryšių atvaizdavimas. Šio metodo pranašumas yra tas, kad galima nustatyti ar prieštarinėjimai *SBVR* taisyklėse yra sukelti aletinių, deontinių ar kombinuotų apribojimų.

### 1.5.4. Veiklos taisyklių neprieštaringumas panaudojant *Alloy* [29]

*Alloy* yra kalba, skirta aprašyti struktūras ir jas nagrinėti pasinaudojant nagrinėjimo įrankiu. Darbo didelis trūkumas yra tas, kad nėra aprašytos transformacijos iš *SBVR* į *Alloy*. Darbe sukuriama *Alloy* modelis pasinaudojant veiklos konceptuali modeliu. Sekančiame žingsnyje darbo autoriai į sukurtą modelį įtraukia veiklos taisykles. Pridedant kiekvieną taisyklę nagrinėjimo įrankis bando rasti galiojantį pavyzdį, kuris atitiktų esamą modelį. Jeigu tokio pavyzdžio surasti nepavyksta, yra daroma išvada, kad modelis yra nenuoseklus. Nenuoseklumą sukurianti taisyklė yra randama sukuriant disjunkcijos tarp naujausiai pridėtos taisyklės ir tikrinamos taisyklės predikatą ir tuomet patikrinant, ar nuoseklumas yra išlaikytas. Jeigu nuoseklumas yra išlaikytas, tuomet taisyklės yra prieštaringos viena kitai. Šis procesas yra neautomatizuotas ir, jeigu modelyje pridėdama papildoma taisyklė, tai blogiausiu atveju prieštarinėjimo patikrinimą reikės atlikti tiek kartų, kiek modelyje yra taisyklių. Darbe taip pat neminimas atvejis kai keletas taisyklių sukuria prieštarinėjimą.

### 1.5.5. *SBVR* prieštarinėjimo patikrinimo metodų palyginimas

Buvo išanalizuoti keturi *SBVR* prieštarinėjimo patikrinimo metodai. Toliau pateikta 1.7 lentelė, kurioje visi aprašyti metodai apibendrinami ir palyginami pagal išskirtus kriterijus.

1.7 lentelė. Esamų *SBVR* prieštarinėjimų aptikimo sprendimų palyginimas

Palyginimo kriterijus	Domain Independent Method [18]	Transforming <i>SBVR</i> Business Semantics Into Web Ontology [22]	Logic based reasoning support for <i>SBVR</i> [27]	Method for Verifying the Consistency of Business Rules [29]
Naudojama tarpinė kalba	<i>SMT-LIBv2</i>	<i>OWL2</i>	<i>ORM2</i> ir <i>ALCQI</i>	<i>Alloy</i>
Automatizavimo lygis	Automatizuota	Pusiau automatizuota <sup>1</sup>	Rankinis <sup>2</sup>	Rankinis

Pateiktas veikiantis prototipas	Ne	Ne	Ne	Ne
Pateikti prieštarinamų tipai	Ne	Ne	Ne	Ne

1 - Automatizavimo lygis nurodytas kaip pusiau automatizuotas, nes atliktos transformacijos į *OWL2* ontologijas nėra automatiškai patikrinamos siekiant atrasti prieštarinamus.

2 - Automatizavimo lygis nurodytas kaip rankinis, nes įrankis priima *ORM2* informacinę formą vietoje *SBVR*.

Išanalizuoti esami *SBVR* prieštarinamų aptikimo sprendimai nebus naudojami kuriant šiame darbe aprašomą sprendimą, nes visi rasti prieštarinamų aptikimo *SBVR* specifikacijose sprendimai naudoja tarpines kalbas ir sprendimai nėra automatizuoti. Sprendimas, aprašytas *Domain Independent Method*[18], sudaro galimybes prieštarinamų aptikimo proceso automatizavimui. Šiame darbe aptikti *SBVR* specifikacijų aspektai bus panaudojami aptinkant prieštarinamus šiame darbe aprašytame sprendime.

## 1.6. Darbo tikslas, uždaviniai ir siekiami privalumai

Šiame skyriuje pateikiamas darbo tikslas, išskelti uždaviniai, kurie padės pasiekti tikslą ir kokie bus privalumai atlikus šį darbą.

**Tikslas** – sudaryti sąlygas prieštarinamams aptikti *SBVR* specifikacijoje.

### Uždaviniai

1. Išanalizuoti:
  - a. informacinių sistemų kūrimo procesą;
  - b. *SBVR* standartą ir metamodelį;
  - c. *SBVR* veiklos žodynų ir taisyklių sudarymo principus;
  - d. esamus *SBVR* prieštarinamo aptikimo mechanizmus.
2. Nustatyti *SBVR* prieštarinamų tipus.
3. Sukurti metodą, kuris būtų naudojamas prieštarinamų aptikimui *SBVR* specifikacijose.
4. Sukurti prototipą, kuris pagal sukurtą metodą aptinka *SBVR* veiklos žodynų ir taisyklių prieštarinamus.
5. Ištestuoti prieštarinamų aptikimo algoritmų veiksmingumą, naudojant nustatytus prieštarinamų tipus.
6. Eksperimentiškai iširti prieštarinamų aptikimą *SBVR* specifikacijose.
7. Apibendrinti tyrimo rezultatus.

**Darbo privalumai** - sukurtas metodas, kuris leidžia aptikti *SBVR* specifikacijose užrašytų teiginių prieštarinamus jų užrašymo aplinkoje.

## 1.7. Siekiamo sprendimo apibrėžimas

Siekiamas sprendimas turėtų nenaudojant transformacijų į kitas kalbas gebėti aptikti *SBVR* žodynuose ir taisyklių rinkiniuose išskirtus prieštarinamų tipus ir pranešti apie galimus prieštarinamus

naudotojui. Siekiamas sprendimas turėtų būti panaudojamas tame pačiame *SBVR* redaktoriaus lange paspaudus kelis mygtuko paspaudimus ir turėtų nereikalauti duomenų eksportavimo/importavimo.

### **1.8. Analizės išvados**

1. Nustatyta, kad *SBVR* yra standartas, kuris pateikia būdą aprašyti veiklos žodynus ir taisykles natūralia struktūrizuota kalba, kurią gali interpretuoti kompiuterinės sistemos. Standartas yra naudojamas kuriamos sistemos reikalavimų specifikavimo etape ir panaudojamas sistemos programinio kodo kūrime.
2. Išnagrinėjus *SBVR* žodynų ir taisyklių sudarymo principus nustatyta, kad prieštarigumai *SBVR* specifikacijose gali egzistuoti ir yra skirstomi į žodyno įrašų prieštarigumus ir veiklos taisyklių prieštarigumus. Pagrindinės prieštarigumų atsiradimo priežastys yra: *SBVR* standarto nesilaikymas, įrašų pasikartojimai, tipų neatitikimai, kardinalumų neatitikimai, modalumų prieštarigumai arba veiksmožodinių konceptų prieštarigumai. Išskirti ir apibrėžti *SBVR* žodynų ir taisyklių prieštarigumai gali būti automatizuotai aptinkami *SBVR* redaktoriuose.
3. Išanalizavus prieštarigumų ir klaidų tikrinimo *SBVR* žodynuose ir taisyklėse naudotojus buvo nustatyta, kad tai veiklos bendruomenės, kuriose *SBVR* taisyklės įtakoja elgseną, veiklos apribojimus, ir informacinių sistemų inžinieriai, kuriems *SBVR* padeda suprasti veiklos taisykles ir tokiu būdu palengvina komunikaciją tarp veiklos dalyvių ir informacinių sistemų inžinierių reikalavimų surinkimo etape.
4. Iš analizuotų, esamų *SBVR* žodynų ir taisyklių prieštarigumo aptikimo metodų tik vienas turėjo automatizavimo galimybę, tačiau nei vienas metodas nepateikė veikiančio prototipo ir konkrečių išskirtų prieštarigumų tipų. Išanalizuoti metodai pasižymėjo tarpinių kalbų naudojimu. Nors analizuotuose metoduose naudojamos tarpinės kalbos turi didesnę teorinę pagrindą ir išvystytus įrankius, tačiau jos sumažina arba uždeda apribojimus *SBVR* standarto išraiškingumui ir dėl to nuspręsta nenaudoti tarpinių kalbų transformacijų.

## **2. Veiklos žodynų ir taisyklių neprieštaringumo tikrinimo sprendimo prototipo reikalavimų specifikacija ir projektas, formalus aprašas**

### **2.1. Reikalavimų specifikacija**

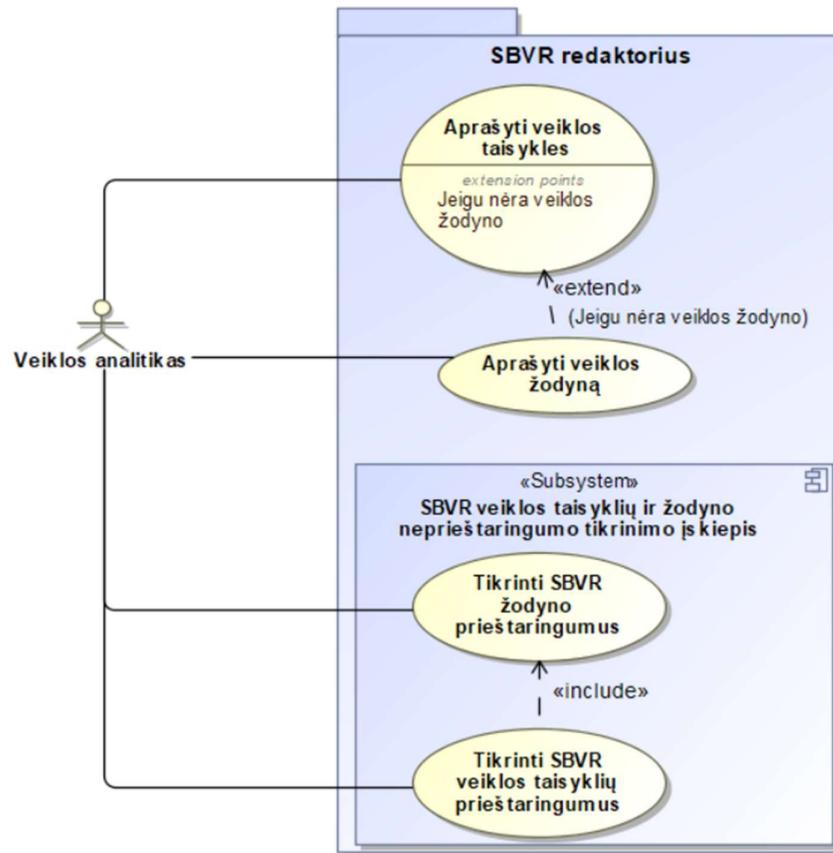
Reikalavimų specifikacijos skyriuje yra aprašomi keliami reikalavimai kuriamam *SBVR* veiklos žodyno ir taisyklių neprieštaringumo tikrinimo metodo prototipo realizacijai. Šiame skyriuje taip pat yra pateikiama kuriamos sistemos *UML* panaudojimo atvejų diagrama. Kiekvienam panaudojimo atvejui pateikiama panaudojimo atvejo specifikacija, naudotojo ir sistemos veiklos diagramos.

Keliami reikalavimai kuriamai sistemai:

1. sistema turi turėti galimybę aprašyti *SBVR* veiklos žodynus;
2. sistema turėtų turėti funkcionalumą, kuris leidžia *SBVR* žodynų įrašuose aptikti prieštarumus arba daugiaprasmiškumo situacijas;
3. sistema turi turėti galimybę aprašinėti *SBVR* veiklos taisykles;
4. sistema privalo turėti funkcionalumą, kuris sugeba aprašytose *SBVR* veiklos taisyklėse aptikti prieštarumus;
5. sistema neturėtų naudoti tarpinių duomenų formatų arba transformacijų iš *SBVR* specifikacija aprašytų žodyno įrašų ir taisyklių siekiant aptikti prieštarumus. Šis reikalavimas kyla iš analizės etapo, kuriame buvo nustatyta, kad daugelis esamų sprendimų siekia konvertuoti duomenis tam, kad būtų panaudoti esami prieštarumų aptikimo sprendimai. Tokie sprendimai limituoja *SBVR* žodynų ir taisyklių aprašymo galimybes arba dėl tam tikrų formatų neatitikimų nesugeba atvaizduoti visų *SBVR* išraiškų;
6. rezultatai turėtų būti pateikiami tame pačiame *SBVR* redaktoriaus lange, kad veiklos analitikas galėtų be konteksto keitimo patikrinti žodyno įrašų arba veiklos taisyklių prieštarumus.

#### **2.1.1. Panaudojimo atvejai**

Šiame skyriuje pateikiama *UML* panaudojimo atvejų diagrama yra pavaizduota 2.1 pav.



2.1 pav. SBVR veiklos taisyklių ir žodyno neprieštaravimo tikrinimo panaudojimo atvejų diagrama

2.1 lentelėje - 2.4 lentelėje yra pateiktos kiekvieno panaudojimo atvejo specifikacijos.

2.1 lentelė. PA 1. Aprašyti veiklos taisykles specifikacija

<b>PA 1. Aprašyti veiklos taisykles</b>	
<b>Tikslas/uždavinys.</b> Suteikti galimybę aprašyti <i>SBVR</i> veiklos taisykles	
<b>Aprašymas.</b> Aprašomos <i>SBVR</i> veiklos taisyklės	
<b>Prieš-sąlyga</b>	Veiklos analitikas atsidaręs <i>SBVR</i> redaktoriaus langą.
<b>Aktorius</b>	Veiklos analitikas
<b>Sužadinimo sąlyga</b>	Veiklos analitikas nori aprašyti veiklos taisykles
<b>Pagrindinis scenarijus</b>	
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija</b>
<ol style="list-style-type: none"> <li>1. Sukuriamas failas su .rules išplėtinium.</li> <li>2. Aprašomos <i>SBVR</i> veiklos taisyklės pasinaudojant .voc žodyno faile aprašytais konceptais, charakteristikomis ir rolėmis.</li> </ol>	<ol style="list-style-type: none"> <li>1. Sukurtas failas.</li> <li>2. Sistema išsaugoja aprašytas <i>SBVR</i> veiklos taisykles.</li> </ol>
<b>Po-sąlyga</b>	Sukurtos ir išsaugotos veiklos taisyklės

2.2 lentelė. PA 2. Aprašyti veiklos žodyną specifikacija

<b>PA 2. Aprašyti veiklos žodyną</b>
--------------------------------------

<b>Tikslas/uždavinys.</b> Suteikti galimybę aprašyti <i>SBVR</i> veiklos žodyną	
<b>Aprašymas.</b>	
<b>Prieš-sąlyga</b>	Veiklos analitikas atsidaręs <i>SBVR</i> redaktoriaus langą.
<b>Aktorius</b>	Veiklos analitikas
<b>Sužadinimo sąlyga</b>	Veiklos analitikas nori aprašyti veiklos žodyną
<b>Pagrindinis scenarijus</b>	
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija</b>
<ol style="list-style-type: none"> <li>1. Sukuriamas failas su .voc išplėtinio.</li> <li>2. Aprašomas <i>SBVR</i> veiklos žodynas su konceptais, charakteristikomis ir rolėmis.</li> </ol>	<ol style="list-style-type: none"> <li>1. Sukurtas failas.</li> <li>2. Sistema išsaugoja aprašytus <i>SBVR</i> žodyno įrašus.</li> </ol>
<b>Po-sąlyga</b>	Sukurti ir išsaugoti <i>SBVR</i> žodyno įrašai

2.3 lentelė. PA 3. Tikrinti *SBVR* žodyno prieštaravimus specifikacija

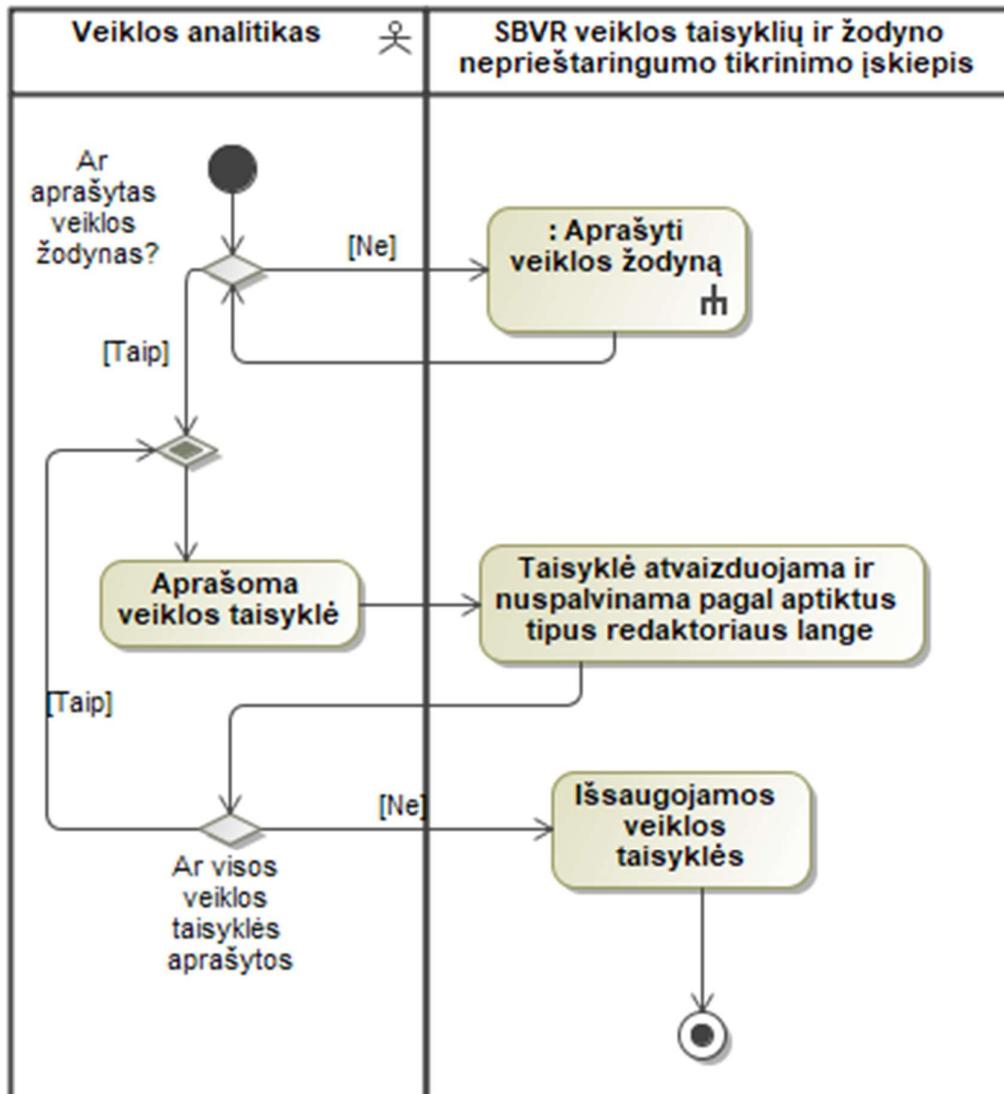
<b>PA 3.</b> Tikrinti <i>SBVR</i> žodyno prieštaravimus	
<b>Tikslas/uždavinys.</b> Patikrinti ar aprašyti konceptai, charakteristikos ir rolės <i>SBVR</i> žodyne nesukuria prieštaravimų.	
<b>Aprašymas.</b>	
<b>Prieš-sąlyga</b>	Aprašytas <i>SBVR</i> žodynas
<b>Aktorius</b>	Veiklos analitikas
<b>Sužadinimo sąlyga</b>	Veiklos analitikas nori patikrinti veiklos žodyno prieštaravimus ir paspaudžia atitinkamą naudotojo sąsajos mygtuką.
<b>Pagrindinis scenarijus</b>	
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija</b>
<ol style="list-style-type: none"> <li>1. Paspaudžiamas mygtukas, skirtas prieštaravimų aptikimui</li> </ol>	<ol style="list-style-type: none"> <li>1. Sistema patikrina <i>SBVR</i> žodyno įrašų prieštaravimus ir pateikia rezultatus ekrane.</li> </ol>
<b>Po-sąlyga</b>	Veiklos analitikui pateikiami <i>SBVR</i> žodyno įrašų prieštaravimo rezultatai.

2.4 lentelė. PA 4. Tikrinti *SBVR* veiklos taisyklių prieštaravimus specifikacija

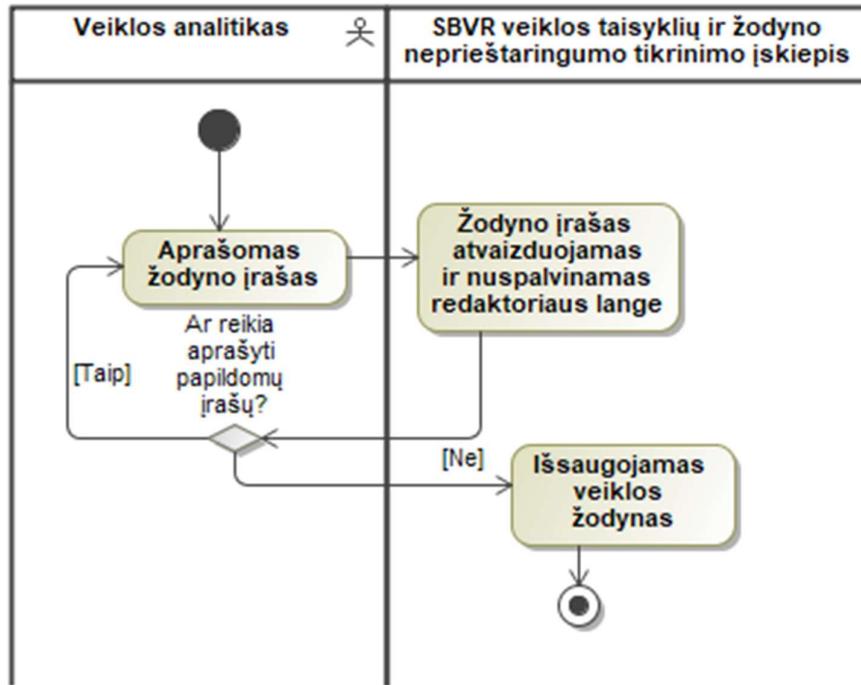
<b>PA 4.</b> Tikrinti <i>SBVR</i> veiklos taisyklių prieštaravimus	
<b>Tikslas/uždavinys.</b> Patikrinti ar aprašytos <i>SBVR</i> veiklos taisyklės nesukuria prieštaravimų tarpusavyje.	
<b>Aprašymas.</b>	
<b>Prieš-sąlyga</b>	Aprašytas <i>SBVR</i> žodynas ir veiklos taisyklės
<b>Aktorius</b>	Veiklos analitikas
<b>Sužadinimo sąlyga</b>	Veiklos analitikas nori patikrinti veiklos taisyklių prieštaravimus ir paspaudžia atitinkamą naudotojo sąsajos mygtuką.
<b>Pagrindinis scenarijus</b>	
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija</b>

1. Paspaudžiamas mygtukas, skirtas prieštarigumų aptikimui.	1. Sistema patikrina <i>SBVR</i> veiklos taisyklių prieštarigumus ir pateikia rezultatus ekrane.
<b>Po-sąlyga</b>	Veiklos analitikui pateikiami <i>SBVR</i> veiklos taisyklių prieštarigumo rezultatai.

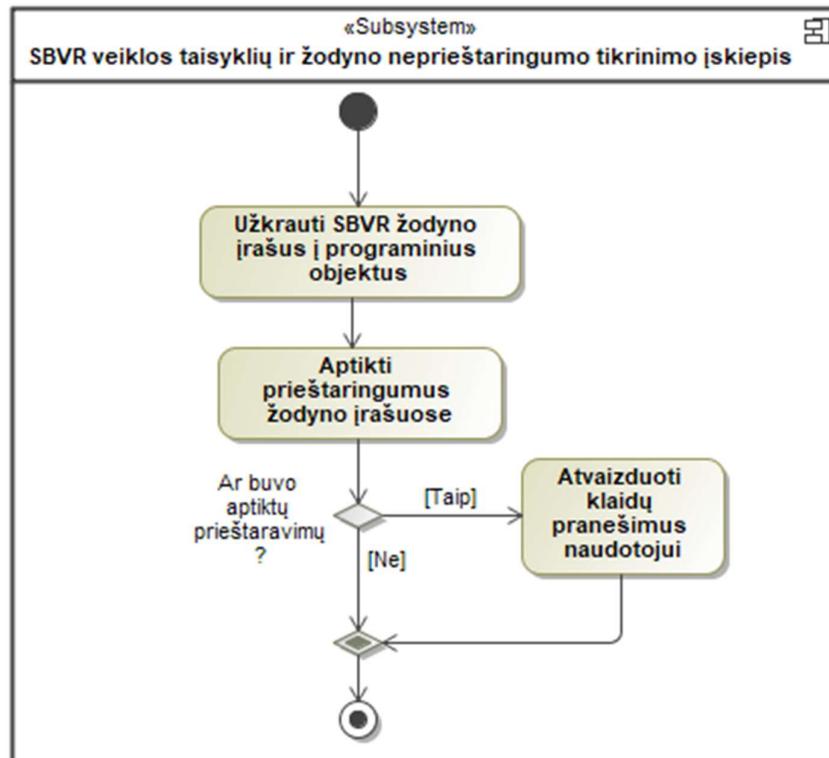
2.2 pav. - 2.5 pav. pateikiamos panaudojimo atvejų veiklos diagramos.



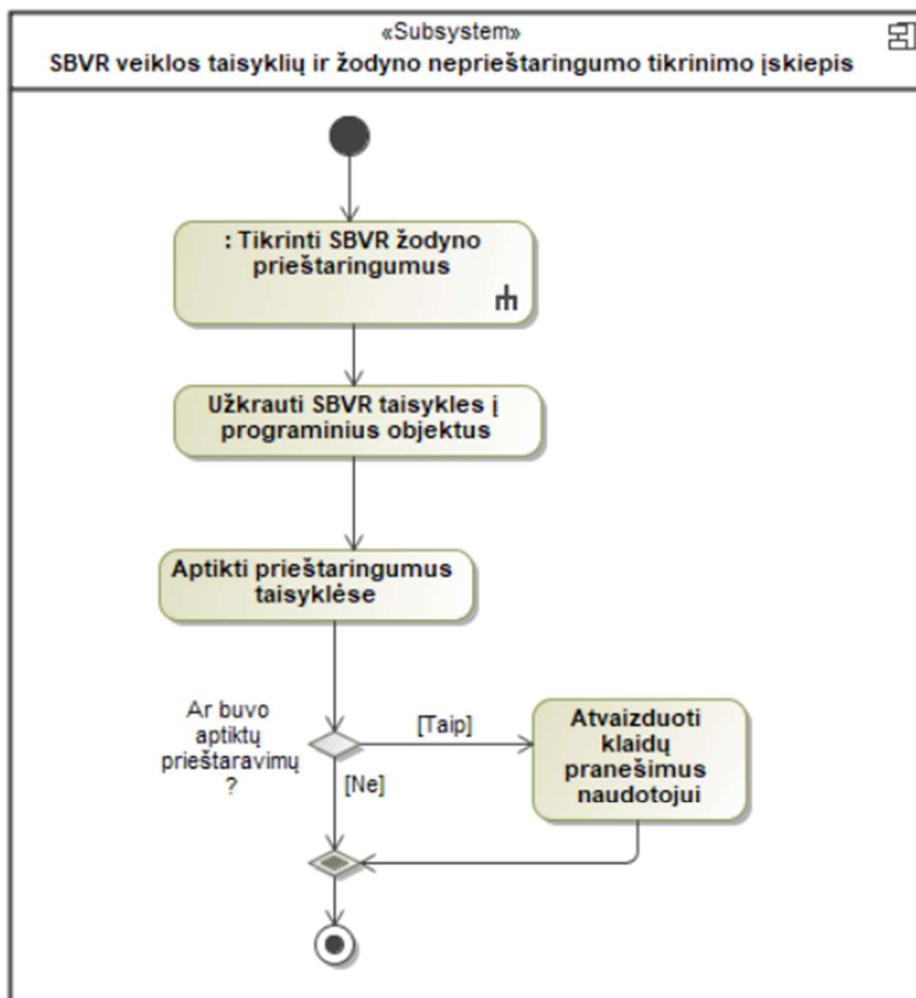
2.2 pav. PA 1. Aprašyti veiklos taisykles veiklos diagrama



2.3 pav. PA 2. Aprašyti veiklos žodyną veiklos diagrama



2.4 pav. PA 3. Tikrinti SBVR žodyno prieštaravimus veiklos diagrama

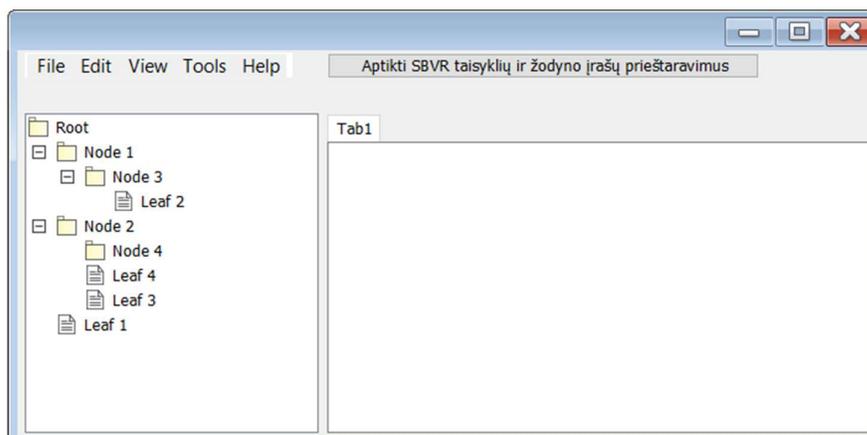


2.5 pav. PA 4. Tikrinti SBVR veiklos taisyklių prieštaravimus veiklos diagrama

## 2.2. Dalykinės srities modelis

Šiame skyriuje yra pateikiama *UML* esybių klasių diagrama, kuri detalizuoja analizės skyriuje nagrinėjamą dalykinę sritį. *SBVR* veiklos taisyklių ir žodyno įrašų prieštaravimo tikrinimo *UML* esybių diagrama yra pavaizduota 2.6 pav.





2.7 pav. *SBVR* veiklos taisyklių ir žodyno įrašų prieštaravimo aptikimo naudotojo sąsajos modelis

## 2.4. Formalus veiklos žodynų ir taisyklių neprieštaringumo tikrinimo aprašas

Neprieštaringumo tikrinimo pagrindinis tikslas yra aptikti prieštarigus arba dviprasmybes sukeliančius žodynų įrašus arba taisykles. Neprieštaringumo tikrinimas *SBVR* specifikacijos aprašytose taisyklėse ir žodynuose yra tikrinamas dviem etapais, kurie yra nepriklausomi vienas nuo kito. Pirmas etapas yra patikrinti prieštarigumus žodyne. Antras etapas yra patikrinti prieštarigumus aprašytose taisyklėse. Siūlomo metodo procesas pavaizduotas 2.8 pav.



2.8 pav. *SBVR* veiklos žodyno ir taisyklių neprieštaringumo tikrinimo proceso diagrama

„Aprašyti veiklos žodyną“ ir „Aprašyti veiklos taisykles“ proceso dalys yra jau realizuotos [6]. Šis darbas ruošiamas [6] darbo pagrindu pasinaudojant jau sukurto *SBVR* redaktoriaus galimybėmis, todėl šiame darbe visas dėmesys skiriamas prieštarigumų aptikimui *SBVR* veiklos žodynuose ir *SBVR* veiklos taisyklių aprašuose. Prieštaringumų tikrinimas ir aptikimas atliekamas dviem etapais.

### 2.4.1. Prieštaringumų aptikimas žodyne

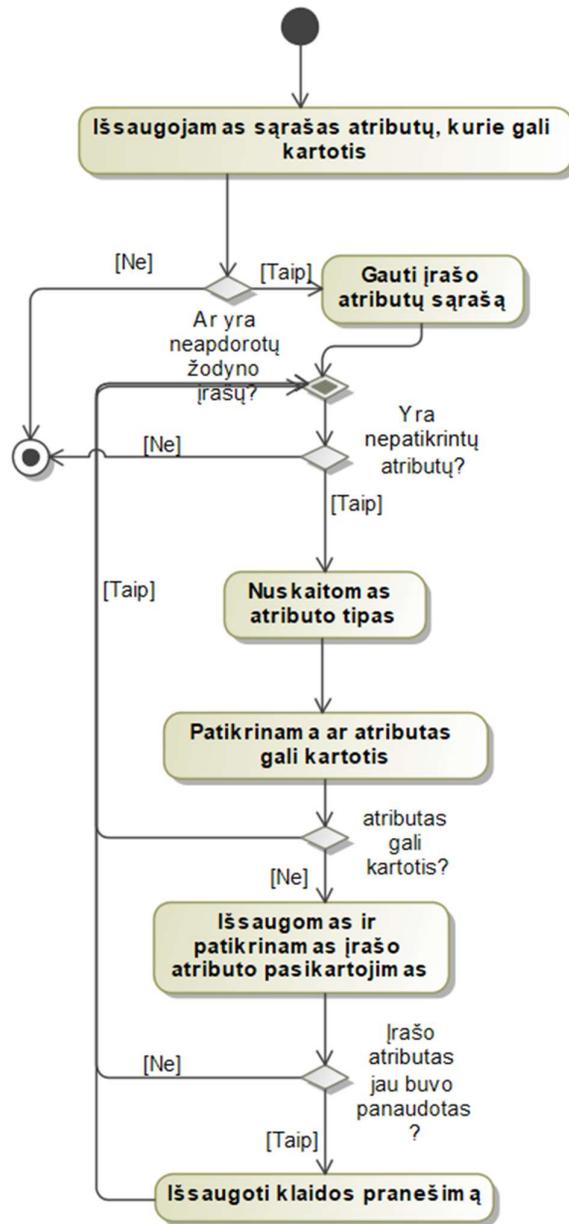
Pirmas etapas yra prieštarigumų aptikimas žodyne, kuris atliekamas paspaudus veiksmo iniciavimo mygtuką grafinėje naudotojo sąsajoje. Prieš paspaudžiant mygtuką veiklos analitikas privalo būti susikūręs veiklos žodyno ir taisyklių failus. Paspaudus mygtuką programinis kodas nuskaityto veiklos žodyno įrašus ir juos transformuoja į atitinkamus programinius objektus, kurie padeda išskirti informaciją iš kiekvieno žodyno įrašo. Nuskaičius žodyno įrašus programinio kodo veikimo eiga

perduodama prieštaringumų aptikimo žodynų įrašuose algoritmui, kuris turi skirtingus etapus. Kiekvienas algoritmo etapas yra skirtas aptikti vieną prieštaringumų tipą. Vykdamas iš eilės yra aptinkama:

1. pasikartojantys žodyno įrašų atributai, kurie negali kartotis;
2. sinonimų aprašymų neatitikimai;
3. pasikartojantys įrašai žodyne;
4. prieštaringos charakteristikos, kurių bendriniai konceptai sutampa, o veiksmožodiniai konceptai yra priešingų poliškumų;
5. aprašytų faktų neatitikimai statiniams apribojimams žodyne;
6. netinkami rolių aprašymai, kai rolės turi daugiau nei vieną bendrinį konceptą;
7. netinkami rolių priskyrimai, kai rolė neturi nurodyto tinkamo koncepto tipo;
8. konceptų aprašų prieštaringumai, kai vienas konceptas priklauso daugiau negu vienai kategorijai iš vienos kategorizacijos arba segmentacijos schemos;
9. cikliškumai konceptų aprašymuose, kai konceptui nurodytas bendrinis konceptas sukuria tipų cikliškumą.

Programinis kodas nestabdo vykdomo proceso aptikus prieštaringumą, o išsaugoja prieštaringumo klaidos pranešimą programiniuose objektuose, kuriuos galima atvaizduoti naudotojo sąsajoje po algoritmo įvykdymo. Toliau detalizuojami algoritmo žingsniai pavaizduoti 2.9 pav. - 2.17 pav.

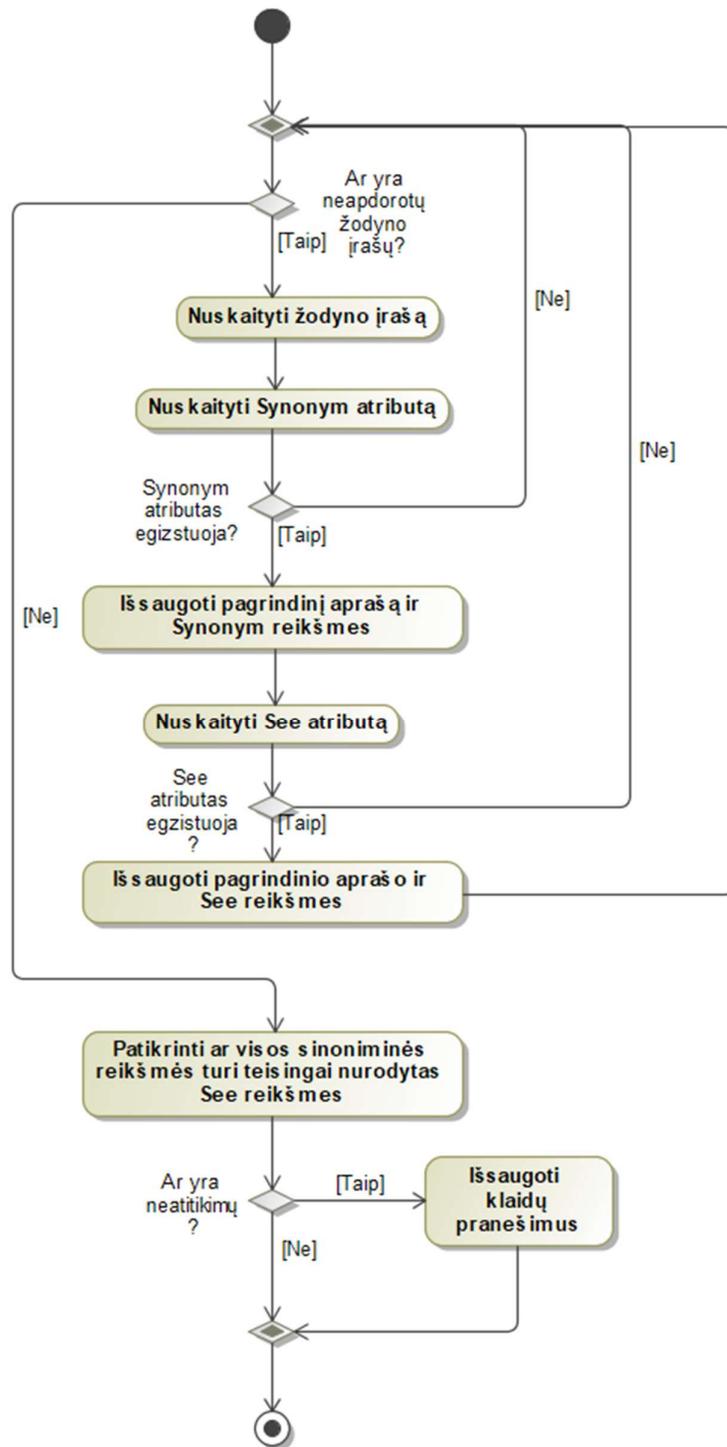
2.9 pav. pavaizduotas algoritmas skirtas aptikti pasikartojančius žodyno įrašų atributus, kurie negali kartotis. Algoritmo pradžioje yra išsaugojamas sąrašas atributų, kurie žodynų įrašuose gali kartotis, pavyzdžiui: *General concept*, *Synonym* ir kiti. Toliau algoritmas nuskaityti žodyno įrašus ir cikle imdamas po vieną žodyno įrašą vykdo veiksmus. Imant po vieną žodyno įrašą yra nuskaitytas žodyno įrašo atributų sąrašas ir kiekvienas atributas apdorojamas vidiniame cikle. Kiekvienam atributui yra išgaunamas atributo tipas ir patikrinama, ar atributas gali kartotis žodyno įrašuose. Jeigu atributas gali kartotis, pereinama prie sekančio atributo apdorojimo. Jeigu atributų daugiau nėra, pereinama prie sekančio žodyno įrašo apdorojimo. Jeigu atributas negali kartotis, patikrinama, ar atributas su tokiu pačiu tipu buvo nuskaitytas apdorojamam žodyno įrašui. Jeigu atributas su tokiu pačiu atributo tipu buvo aptiktas anksčiau, išsaugojamas klaidos pranešimas. Atributas išsaugojamas į aptiktų atributų sąrašą esamas žodyno įrašui ir algoritmas pereina prie kito atributo apdorojimo. Apdorojus visus žodyno įrašus, algoritmas baigia savo darbą.



2.9 pav. Aptikti pasikartojančius žodyno įrašų atributus

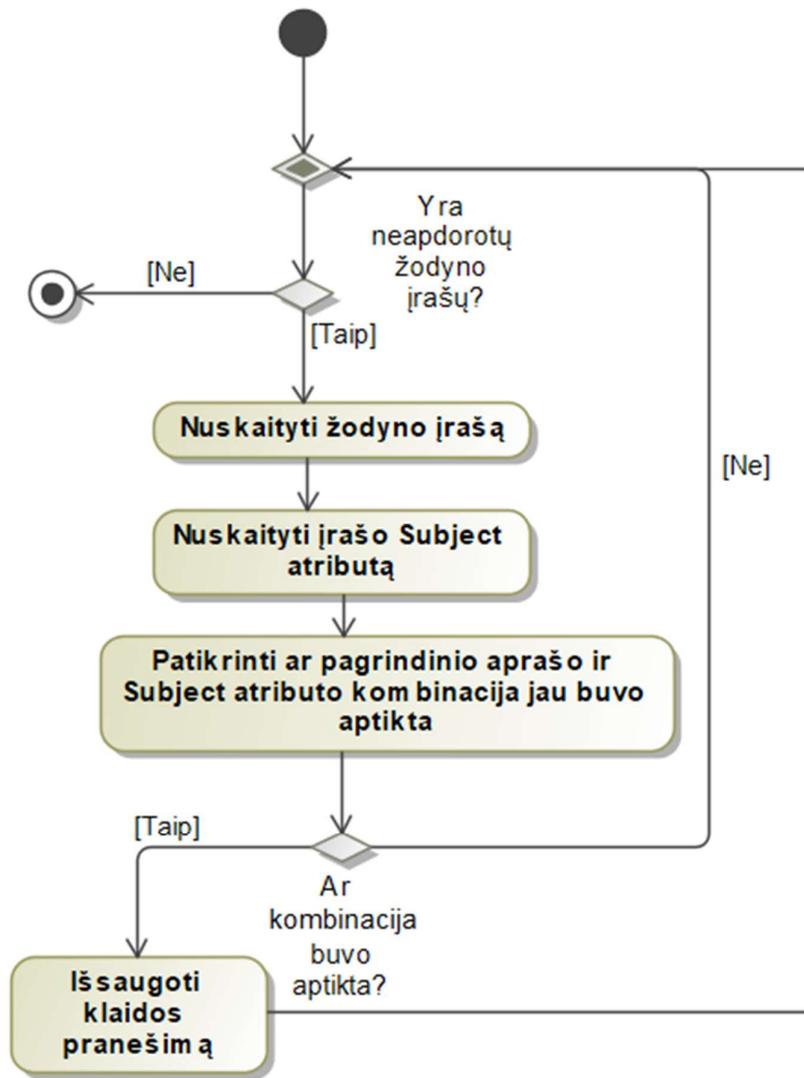
2.10 pav. pavaizduotas algoritmas skirtas aptikti sinonimų aprašymų neatitikimus. Algoritmas apdoroja kiekvieną žodyno įrašą atskirai cikle. Kiekvienam žodyno įrašui yra patikrinama, ar įrašas turi *Synonym* atributą. Jeigu atributas egzistuoja, nuskaitoma atributo reikšmė ir yra išsaugojama pagrindinio aprašo ir *Synonym* atributo reikšmės kombinacija pirmame sąrašė. Kiekvienam žodyno įrašui algoritmas taip pat patikrina, ar įrašas turi *See* atributą. Jeigu atributas egzistuoja, nuskaitoma atributo reikšmė ir išsaugojama pagrindinio aprašo ir *See* atributo reikšmės kombinacija antrame sąrašė. Algoritmui apdorojus visus žodyno įrašus, pereinama prie išsaugotų reikšmių apdorojimo. Kiekvienai išsaugotai pagrindinio aprašo ir *Synonym* atributo reikšmės kombinacijai iš pirmojo sąrašė yra bandoma rasti pagrindinio aprašo ir *See* atributo reikšmės kombinaciją iš antrojo sąrašė taip, kad

*Synonym* reikšmė atitiktų antrajame sąraše esantį pagrindinį aprašą, o antrojo sąrašo įrašė *See* atributo reikšmė atitiktų pirmojo sąrašo bendrinį aprašą. Algoritmui neaptikus atitikimo – išsaugojamas klaidos pranešimas kiekvienam neatitikimui.



2.10 pav. Aptikti sinonimų aprašymų neatitikimus

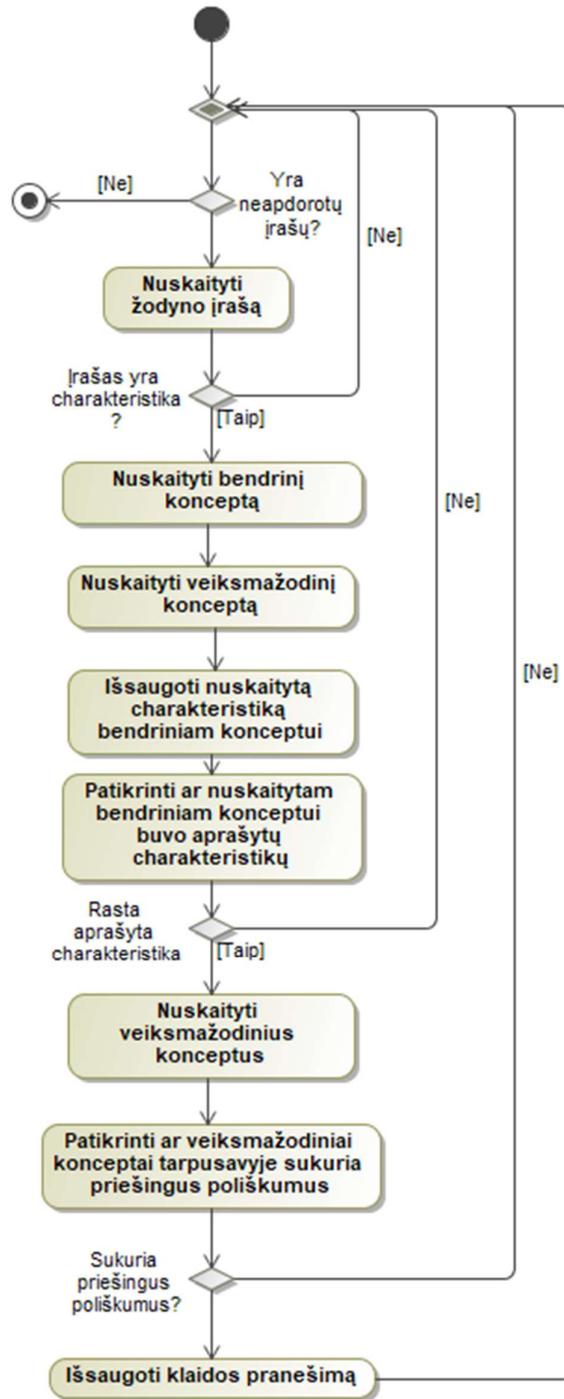
2.11 pav. pavaizduotas algoritmas, skirtas aptikti pasikartojančius įrašus žodyne. Algoritmas apdoroja kiekvieną žodyno įrašą atskirai cikle. Iš nuskaityto žodyno įrašo yra išgaunamas pagrindinis aprašas ir *Subject* atributo reikšmė. Jeigu įrašas neturi *Subject* atributo, algoritmas naudoja tuščią tekstinę eilutę vietoje *Subject* atributo reikšmės. Žodyno įrašo pagrindinio aprašo ir algoritmo nustatyta *Subject* atributo reikšmių kombinacija yra patikrinama, ar nėra jau prieš tai išsaugota sąrašė ir tuo pačiu išsaugojama. Jeigu tokia įrašo kombinacija anksčiau jau buvo aptikta – išsaugojamas klaidos pranešimas.



2.11 pav. Aptikti pasikartojančius įrašus

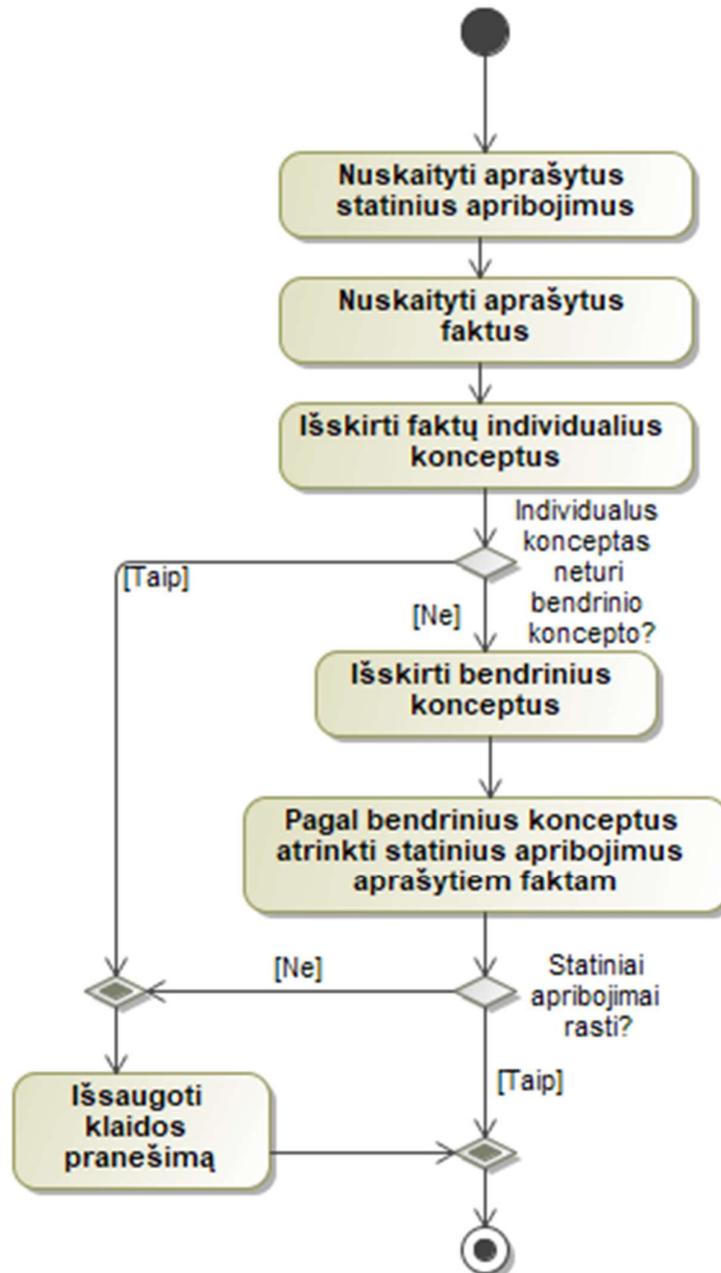
2.12 pav. pavaizduotas algoritmas, skirtas aptikti prieštaringas charakteristikas, kurių bendriniai konceptai sutampa, o veiksmazodiniai konceptai yra priešingų poliškumų. Algoritmas apdoroja kiekvieną žodyno įrašą atskirai cikle. Algoritmas patikrinama, ar žodyno įrašas yra charakteristika. Jeigu ne charakteristika – pereinama prie sekancio įrašo apdorojimo. Jeigu nuskaitytas žodyno įrašas yra charakteristika, nuskaitytas bendrinis konceptas, veiksmazodinis konceptas ir reikšmių pora išsaugojama. Sekantis algoritmo žingsnis yra patikrinti, ar charakteristikos bendriniam konceptui

buvo anksčiau išsaugotų charakteristikų su tokiu pačiu bendriniu konceptu. Jeigu buvo, nuskaitomi veiksmožodiniai konceptai. Veiksmožodiniai konceptai tarpusavyje patikrinami pasinaudojant priešdėliu „ne“. Jeigu randamas atitikmuo, algoritmas išsaugoja klaidos pranešimą, nes du veiksmožodiniai konceptai tarpusavyje sukuria priešingus poliškumus.



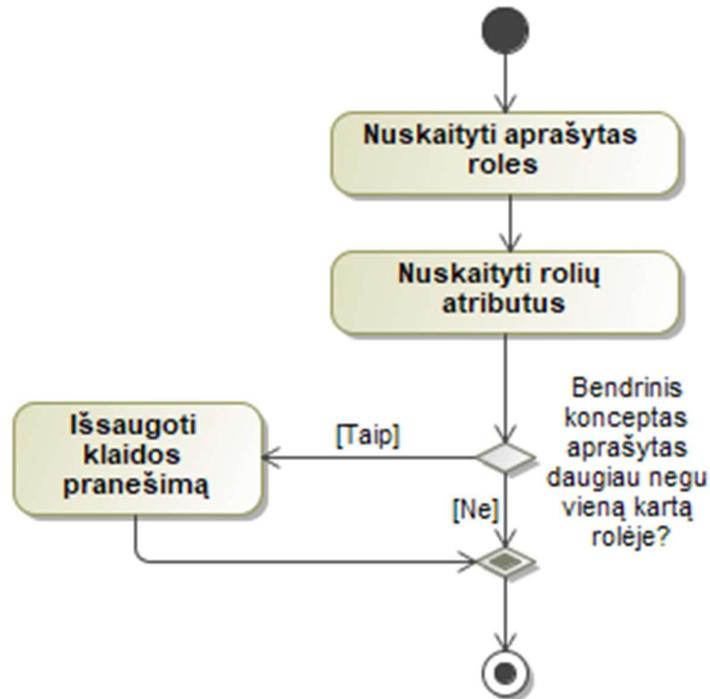
2.12 pav. Aptikti prieštaringas charakteristikas, kurių bendriniai konceptai sutampa, o veiksmožodiniai konceptai yra priešingų poliškumų

2.13 pav. pavaizduotas algoritmas, skirtas aptikti aprašytų faktų neatitikimus statiniams apribojimams žodyne. Algoritmas nuskaityto visus žodyno įrašus ir aptiktus statinius apribojimus išsaugoja į sąrašą. Algoritmui nuskaityti įrašus ir aptikus faktus, faktai išsaugojami į antrą sąrašą. Išsaugojus reikšmes algoritmas apdoroja visus išsaugotus faktus po vieną. Kiekvienam faktui yra išskiriami jo naudojami individualūs konceptai ir jų bendriniai konceptai. Jeigu individualiam konceptui nėra aprašyto bendrinio koncepto – išsaugojamas klaidos pranešimas. Išskirti bendriniai konceptai yra išsaugojami ir algoritmas bando rasti statinį apribojimą, kuris atitinka aprašytą faktą. Jeigu algoritmas neranda aprašyto statinio apribojimo, išsaugojamas klaidos pranešimas.



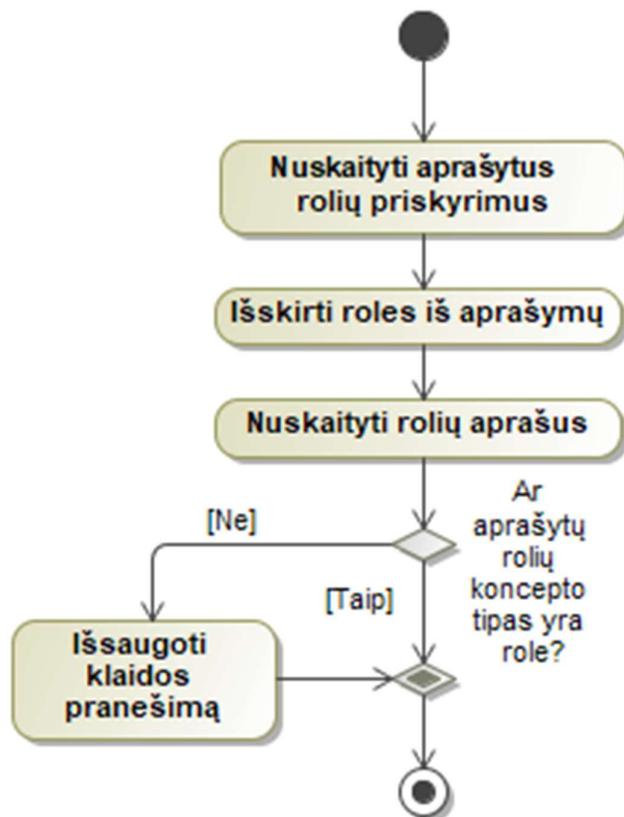
2.13 pav. Aptikti aprašytų faktų neatitikimus statiniams apribojimams žodyne

2.14 pav. pavaizduotas algoritmas, kuris aptinka netinkamą rolių aprašymą, kai rolė turi daugiau nei vieną bendrinį konceptą. Algoritmas apdoroja visus žodynų įrašus cikle. Kiekvienas įrašas yra patikrinamas ar jo *Concept\_type* atributas yra *role*. Jeigu nuskaitytas žodyno įrašas yra rolė, nuskaitymi įrašo atributai ir patikrinama ar bendrinis konceptas rolei yra aprašytas vieną kartą. Jeigu bendrinis konceptas aprašytas daugiau negu vieną kartą algoritmas išsaugoja klaidos pranešimą.



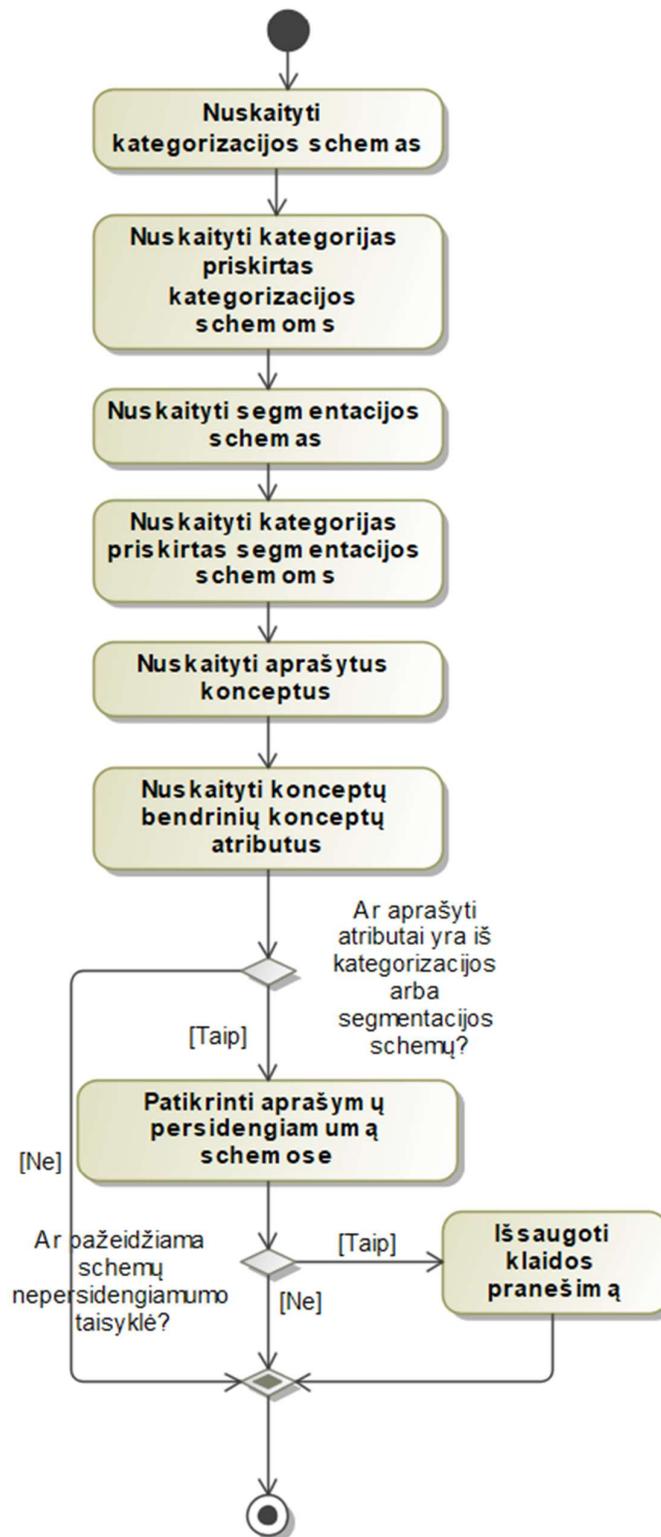
**2.14 pav.** Aptikti netinkamus rolių aprašymus, kai rolės turi daugiau nei vieną bendrinį konceptą

2.15 pav. pavaizduotas algoritmas, kuris aptinka netinkamus rolių priskyrimus, kai rolės neturi nurodyto tinkamo koncepto tipo. Algoritmas nuskaityti visus aprašytus rolių priskyrimus žodyne, išskiria priskiriamų rolių konceptus ir išsaugoja. Kiekvienai išsaugotai rolei algoritmas nuskaityti rolių aprašus ir jų atributus. Jeigu atributų sąrašas nėra nurodyto *Concept\_type* atributo su reikšme *role* – algoritmas išsaugoja klaidos pranešimą.



2.15 pav. Aptikti netinkamus rolių priskyrimus, kai rolės neturi nurodyto tinkamo koncepto tipo

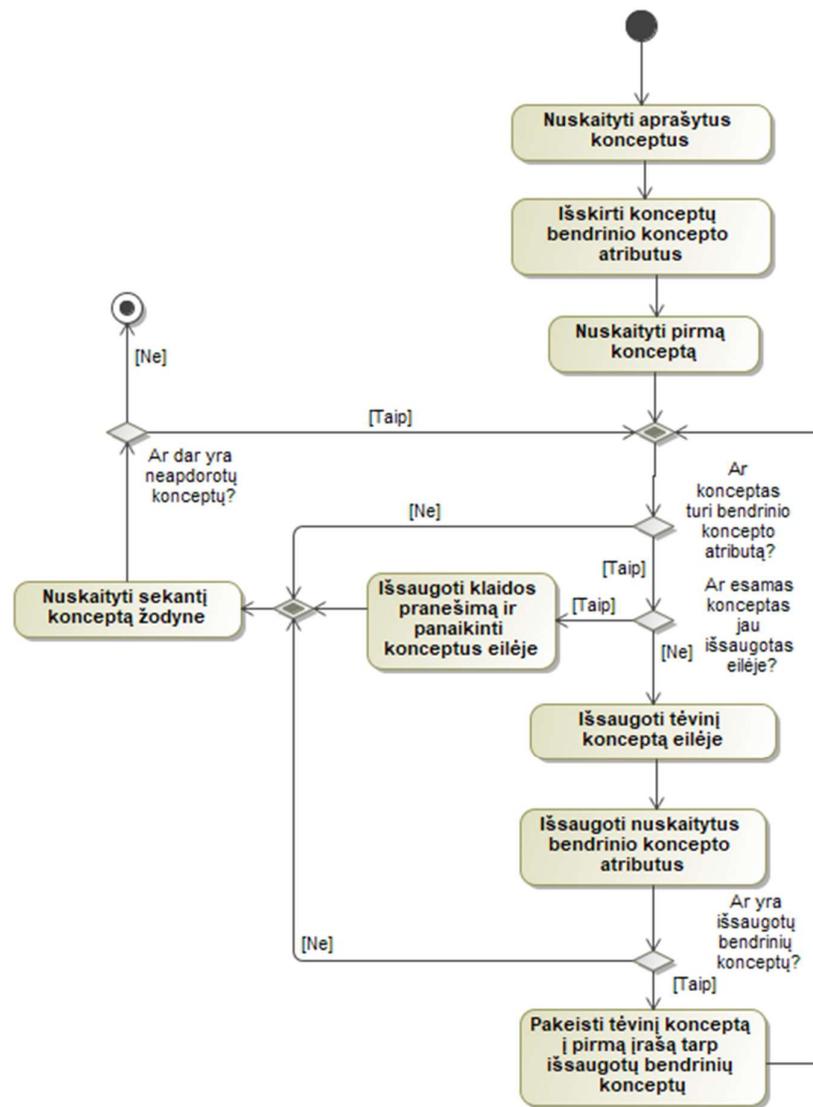
2.16 pav. pavaizduotas algoritmas, kuris aptinka konceptų aprašų prieštaravimus, kai konceptas priklauso daugiau negu vienai kategorijai iš vienos kategorizavimo schemos arba segmentacijos. Algoritmas nuskaitydamas ir apdorodamas žodyno įrašus nuskaityto jų *Concept\_type* atributo reikšmę ir bando ją prilyginti „*Categorization\_type*“ reikšmei. Tokiu būdu algoritmas suranda kategorizacijų tipus, išskiria jų pagrindinius aprašus ir išsaugoja. Antrą kartą nuskaitydamas žodyno įrašus, algoritmas ieško kategorizavimo įrašų ir pagal nurodytą *Concept\_type* atributo reikšmę priskiria jau išsaugotiems kategorizacijų tipam. Toliau algoritmas skenuoja žodyno įrašus ir tikrina, ar žodyno įrašas aprašo segmentaciją prieš tai išsaugotam kategorizacijos tipui. Rastam segmentacijos aprašui algoritmas skenuoja žodyno įrašus, ieškodamas kategorijas aprašytas segmentacijai. Rastos segmentacijos išsaugojamos aptiktiems kategorizacijų tipam, nes kategorizacijos tipai privalomai aprašomi tiek kategorizacijos, tiek segmentacijos schemoms. Nuo šios vietos algoritmas nebeturi segmentacijos ir kategorizacijos išskyrimo. Algoritmas turi išsaugojęs tipus ir tipų galimas reikšmes tiek kategorizacijai, tiek segmentacijai. Nors kategorizacija aprašo nepilną aibę galimų reikšmių, tačiau tiek kategorizacijai, tiek segmentacijai galioja nepersidengiamumo (ang. *disjoint*) sąlyga. Toliau algoritmas skenuoja žodyno įrašus tikrindamas jų bendrinių konceptų atributų reikšmes. Kiekvienam žodyno įrašui algoritmas nuskaityto aprašytus bendrinius konceptus ir patikrina, ar bendrinio koncepto atributo reikšmė yra kategorizacijos reikšmė. Jeigu aptinkama kategorizacijos reikšmė, panaudotas kategorizacijos tipas yra išsaugojamas ir patikrinama ar nagrinėjamas įrašas jau buvo panaudojęs kategorizacijos tipą. Jeigu aptinkama, kad antrą kartą panaudojamas tas pats kategorizacijos tipas, algoritmas išsaugoja klaidos pranešimą.



**2.16 pav.** Aptikti konceptų aprašų prieštaravimus, kai konceptas priklauso daugiau negu vienai kategorijai iš vienos kategorizavimo schemas arba segmentacijos

2.17 pav. pavaizduotas algoritmas, kuris aptinka cikliškumus konceptų aprašymuose, kai konceptui nurodytas bendrinis konceptas sukuria tipų cikliškumą. Algoritmas apdoroja kiekvieną žodyno įrašą

atskirai. Jeigu žodyno įrašas neturi bendrinio koncepto atributo, algoritmas pereina prie sekančio žodyno įrašo apdorojimo. Jeigu žodyno įrašas turi aprašytą bendrinį konceptą, jis yra išskiriamas ir išsaugojama jo reikšmė. Žodyno įrašo, kuriam atliekamas algoritmo apdorojimas, pagrindinis aprašas yra išsaugojamas sąrašė. Aptikta bendrinio koncepto reikšmė yra taip pat išsaugojama sąrašė. Algoritmo sekantis žingsnis yra aptikti žodyno įrašą, nurodytą bendrinio koncepto atribute. Rasto žodyno įrašo bendrinio koncepto atributas yra nuskaitymas, išskiriama reikšmė ir patikrinama, ar reikšmė jau yra išsaugota sąrašė. Jeigu reikšmė jau buvo išsaugota sąrašė - algoritmas išsaugoja klaidos pranešimą, kad buvo aptiktas cikliškumas. Jeigu reikšmės nėra sąrašė, ji yra išsaugojama į sąrašą ir algoritmo žingsniai kartojami bandant atsekti bendrinių konceptų tėvinį (ang. *parent*) elementą. Algoritmas nutraukia vieno įrašo apdorojimą aptikus cikliškumą arba aptikus bendrinį konceptą, kuris neturi bendrinio koncepto atributo.



2.17 pav. Aptikti cikliškumus konceptų aprašymuose, kai konceptui nurodytas bendrinis konceptas sukuria tipų cikliškumą

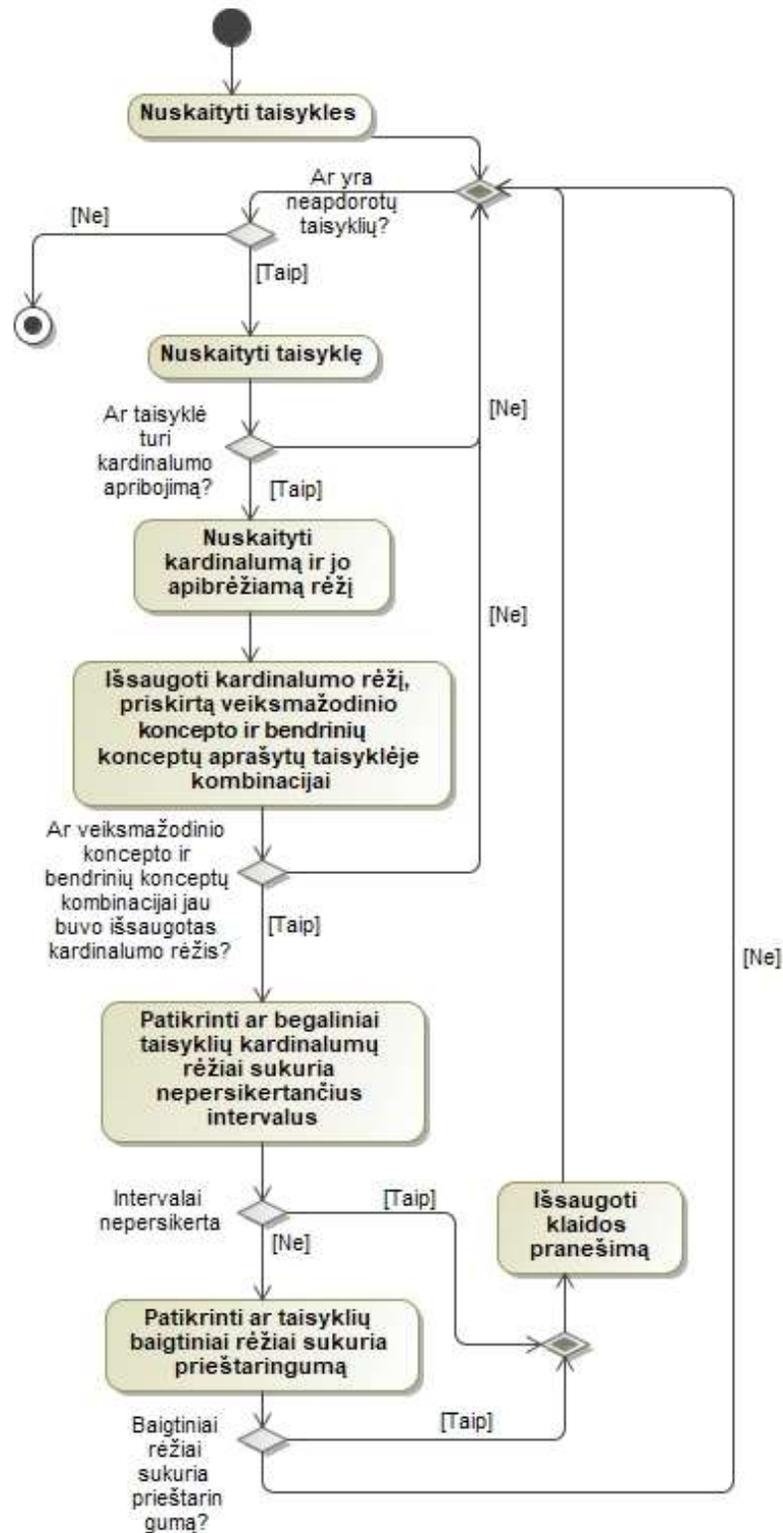
#### 2.4.2. Prieštaringumų aptikimas veiklos taisyklėse

Antras etapas yra prieštaringumų aptikimas veiklos taisyklėse, kuris atliekamas iškart po pirmojo etapo ir nereikalauja naudotojo įsikišimo. Programinis kodas nuskaityto veiklos taisyklių įrašus ir juos susieja su žodyno įrašais. Šioje vietoje verta paminėti, kad naudojamas grafinis *SBVR* taisyklių redaktorius automatiškai aptinka neteisingus žodyno panaudojimus taisyklių aprašymuose ir rodo klaidos pranešimus redaktoriaus lange. Nuskaityti programiniai objektai padeda išskirti informaciją apie kiekvieną veiklos taisyklę: modališkumus, veiksmožodinius konceptus taisyklėse, daiktavardines formas pagal jų pozicijas veiklos taisyklėse, kardinalumus ir kt. Nuskaitytus veiklos taisyklės kodo veikimo eiga perduodama prieštaringumų aptikimo veiklos taisyklėse algoritmui, kuris vykdo skirtingus etapus bandydamas aptikti taisyklių prieštaringumus. Veiklos taisyklių prieštaringumo aptikimo algoritmo etapai bando aptikti:

1. kardinalumų neatitikimus;
2. veiksmožodinių konceptų prieštaringumus, kai veiksmožodinis konceptas taisyklėje sukuria priešingą poliškumą kitai taisyklei;
3. modalumo prieštaringumus, kai modališkumai taisyklėse sukuria priešingus poliškumus;
4. rolių prieštaringumus, kai taisyklės apriboja galimas rolių reikšmes ir reikšmių tipus;
5. veiksmožodinio koncepto taisyklių prieštaringumus, kai žodyne aprašytų faktų kiekis sukelia prieštaringumą aprašytoms taisyklėms.

Kaip ir prieštaringumų aptikimas žodynuose taip ir veiklos taisyklėse, aptikus prieštaringumą, programinio kodo vykdymas nėra stabdomas aptikus prieštaringumą, o yra išsaugojami klaidų pranešimai programiniuose objektuose, kuriais vėliau galima pasinaudoti ir atvaizduoti pranešimus naudotojui grafinėje sąsajoje. Toliau detalizuojami algoritmo žingsniai 2.18 pav. - 2.22 pav.

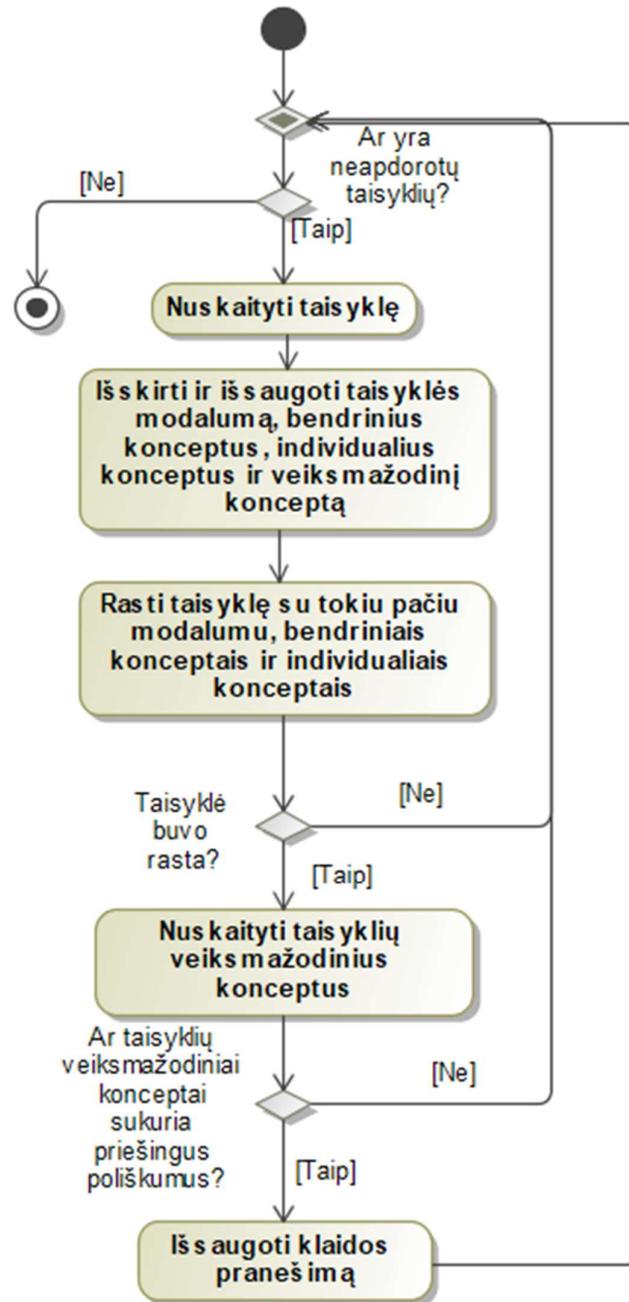
2.18 pav. pavaizduotas algoritmas, kuris yra skirtas aptikti kardinalumų neatitikimus taisyklėse. Algoritmas apdoroja taisyklės iš eilės pagal tai kaip yra aprašytos taisyklių faile. Algoritmas nuskaityto taisyklę ir patikrina, ar taisyklė turi aprašytą kardinalumą. Jeigu taisyklė neturi kardinalumo, algoritmas pereina prie sekančios taisyklės apdorojimo. Jeigu taisyklė turi kardinalumą, kardinalumo reikšmė yra išsaugojama režiu. Pavyzdžiui, jeigu kardinalumas apibrėžtas kaip **daugiausiai 3**, tai šis kardinalumas programos veikimo atmintyje yra išsaugojamas kaip intervalas tarp 0 ir 3. Išsaugotas taisyklės kardinalumas yra išsaugojamas kartu su veiksmožodiniu konceptu ir bendrinių konceptų reikšmėmis. Toliau algoritmas patikrina, ar buvo išsaugotas kardinalumo režis taisyklei su tokiu pačiu veiksmožodiniu konceptu ir bendriniais konceptais. Jeigu nebuvo, algoritmas pereina prie sekančios taisyklės apdorojimo. Jeigu kardinalumo režis jau buvo išsaugotas taisyklei su sutampančiu veiksmožodiniu konceptu ir bendriniais konceptais, algoritmas patikrina, ar išsaugoti intervalai yra persidengiantys. Jeigu intervalai nepersidengia, išsaugojamas klaidos pranešimas.



2.18 pav. Aptikti kardinalumų neatitikimus

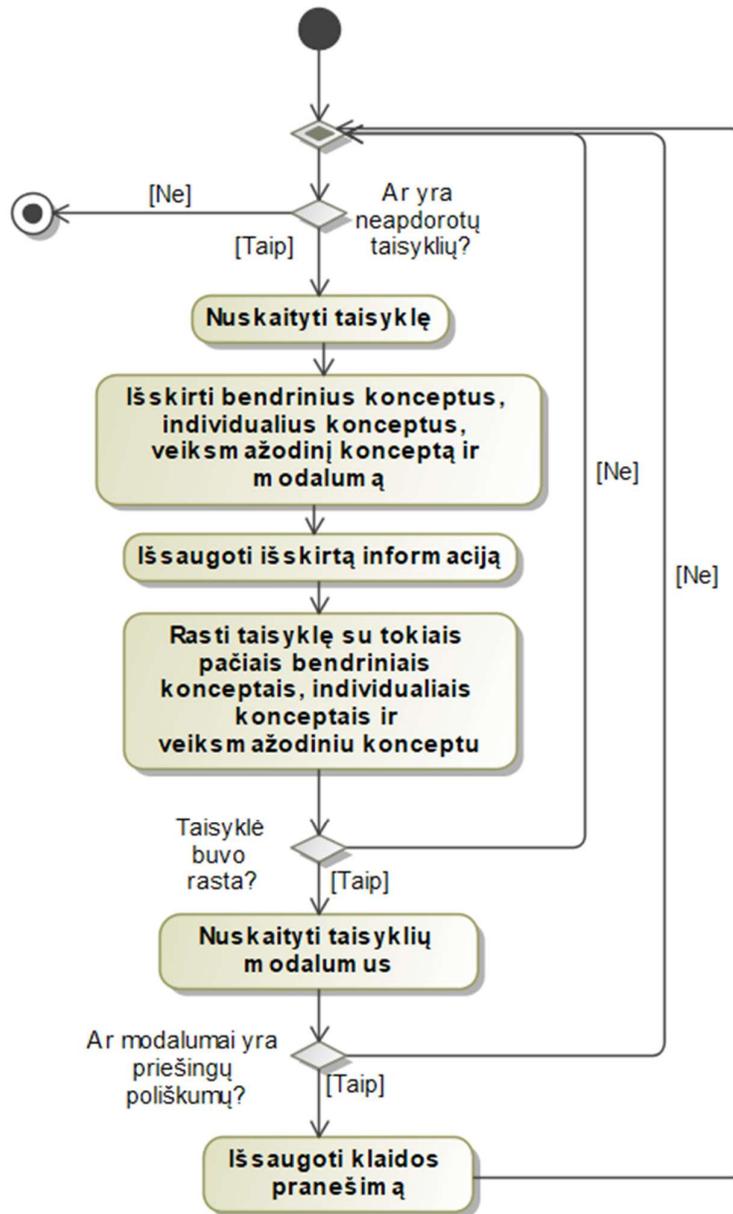
2.19 pav. pavaizduotas algoritmas, kuris yra skirtas aptikti veiksmožodinių konceptų prieštarinumus, kai veiksmožodinis konceptas taisyklėje sukuria priešingą poliškumą kitai taisykle. Algoritmas apdoroja taisykles iš eilės pagal tai, kaip yra aprašytos taisyklių faile. Algoritmas nuskaityt taisyklę ir

išsaugoja taisyklės modalumą, bendrinius konceptus, individualius konceptus ir veiksmažodinį konceptą. Toliau algoritmas bando rasti iš anksčiau išsaugotą taisyklę, kuri turėtų sutampantį modalumą, bendrinius konceptus ir individualius konceptus. Jeigu tokia taisyklė nerandama, algoritmas pereina prie sekančių taisyklių apdorojimo. Jeigu sutampanti taisyklė randama, nuskaitomi taisyklių veiksmažodiniai konceptai ir tarpusavyje patikrinami pasinaudojus priešdėliu „ne“. Jeigu randamas atitikmuo, algoritmas išsaugoja klaidos pranešimą.



2.19 pav. Aptikti veiksmažodinių konceptų prieštaravimus, kai veiksmažodinis konceptas taisyklėje sukuria priešingą poliškumą kitai taisyklei

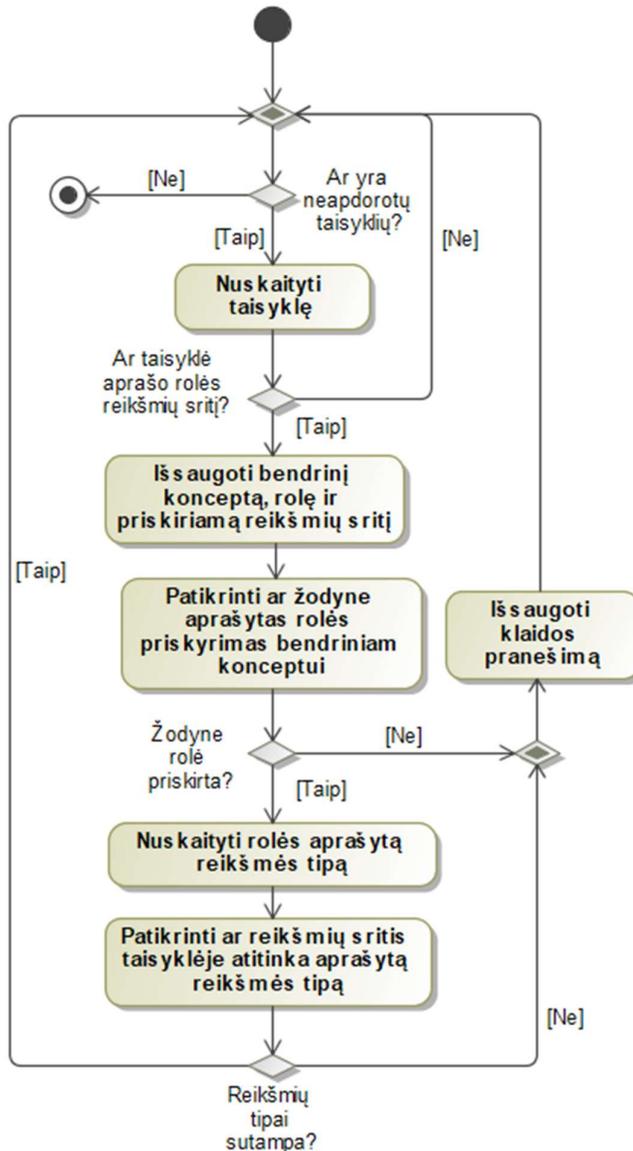
2.20 pav. pavaizduotas algoritmas, kuris aptinka modalumo prieštaringumus, kai modališkumai taisyklėse sukuria priešingus poliškumus. Algoritmas apdoroja taisykles iš eilės pagal tai kaip yra aprašytos taisyklių faile. Algoritmas nuskaitytą taisyklę išsaugoja taisyklės modalumą, bendrinius konceptus, individualius konceptus ir veiksmažodinių konceptų. Toliau algoritmas bando rasti prieš tai išsaugotą taisyklę, kurios bendriniai konceptai, individualūs konceptai ir veiksmažodiniai konceptai sutaptų. Jeigu tokia taisyklė nerandama, pereinama prie sekančios taisyklės apdorojimo. Jeigu taisyklė randama, nuskaityti taisyklių modalumai. Toliau algoritmas patikrina ar taisyklių modalumai tarpusavyje sukuria priešingus poliškumus. Modalumai su poliais yra pavaizduoti 1.2 lentelėje ir 1.3 lentelėje. Jeigu taisyklių modalumai tarpusavyje sukuria priešingus poliškumus, algoritmas išsaugoja klaidos pranešimą.



2.20 pav. Aptikti modalumo prieštaringumus, kai modališkumai taisyklėse sukuria priešingus poliškumus

2.21 pav. pavaizduotas algoritmas, kuris aptinka rolių prieštaringumus, kai taisyklės apriboja galimas rolių reikšmes ir reikšmių tipus. Algoritmas apdoroja taisykles iš eilės pagal tai kaip yra aprašytos

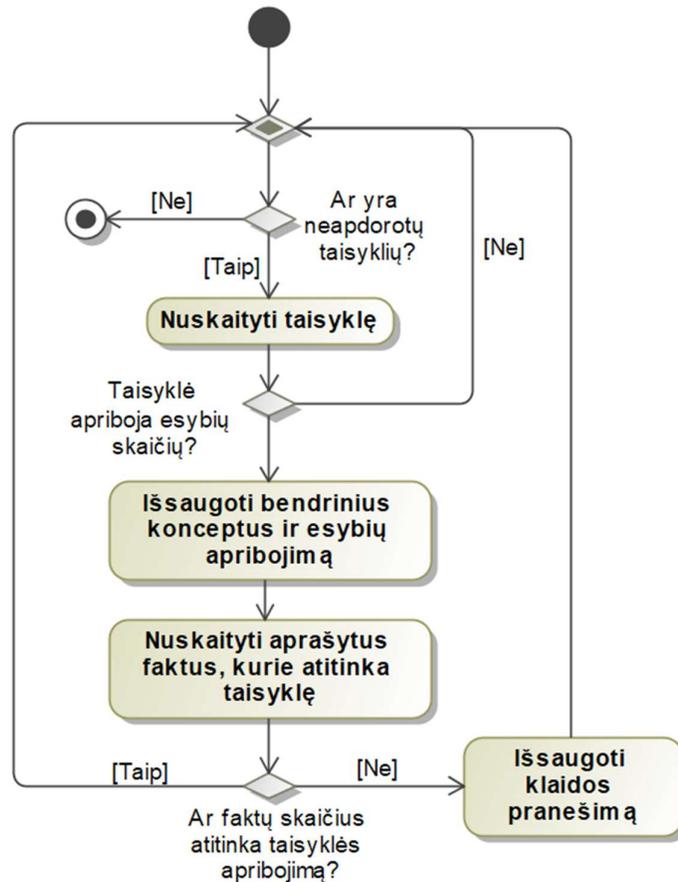
taisyklių faile. Visų pirma algoritmas patikrina, ar taisyklė aprašo apribojimą rolės priskyrimui patikrinant naudojamą veiksmožodinį konceptą. Jeigu veiksmožodinis konceptas lygus *turi* arba *has*, algoritmas nuskaityt taisyklės apibrėžtą reikšmės tipą rolei ir išsisaugoja rolę. Jeigu veiksmožodinis konceptas yra lygus kitoms reikšmės, algoritmas pereina prie sekančių taisyklių apdoravimo. Toliau algoritmas nuskaityt žodyno įrašus ir randa aprašytą rolės konceptą pagal pagrindinį aprašą. Radus konceptą, algoritmas patikrina, ar konceptas yra aprašytas kaip rolę, patikrinant koncepto *Concept\_type* atributo reikšmę. Jeigu reikšmė yra ne *role*, algoritmas išsaugoja klaidos pranešimą. Toliau algoritmas nuskaityt žodyno įrašo bendrinio koncepto atributo reikšmę, kuri nurodo rolės galimas reikšmes. Jeigu taisyklė apibrėžia skaitinę reikšmę, o žodyne rolės aprašymas nurodo tekstinę, loginę arba datos tipą – algoritmas išsaugoja klaidos pranešimą.



2.21 pav. Aptikti rolių prieštaravimus, kai taisyklės apriboja galimas rolių reikšmes ir reikšmių tipus

2.22 pav. pavaizduotas algoritmas, kuris aptinka veiksmožodinio koncepto taisyklių prieštaravimus, kai žodyne aprašytų faktų kiekis sukelia prieštaravimą aprašytoms taisyklėms. Algoritmas apdoroja taisykles iš eilės pagal tai kaip yra aprašytos taisyklių faile. Visų pirma algoritmas patikrina, ar

taisyklė apriboja esybių skaičių. Jeigu neapriboja, algoritmas pereina prie sekančios taisyklės apdorojimo. Jeigu taisyklė apriboja esybių skaičių, išsaugojami bendriniai konceptai aprašyti taisyklėje, veiksmazodinis konceptas ir esybių kiekio apribojimas. Toliau algoritmas nuskaityti žodyno konceptus ir surenka visus aprašytus faktus su taisyklėje sutampančiu veiksmazodiniu konceptu. Surinkti faktai toliau apdorojami ir jų individualiems konceptams išskiriami bendriniai konceptai. Išsaugoti faktai yra filtruojami pagal bendrinius konceptus iš apdorojamos taisyklės. Likę faktai yra grupuojami pagal pirmą individualų konceptą fakte ir yra skaičiuojamos grupių sumos (pasikartojimų skaičius tam pačiam individualiam konceptui). Toliau algoritmas kiekvieną rastą grupę ir joje esančių įrašų kiekį palygina su aprašytu skaitiniu apribojimu taisyklėje ir jeigu apribojimas yra pažeidžiamas – išsaugojamas klaidos pranešimas.



2.22 pav. Aptikti veiksmazodinių konceptų taisyklių prieštarumus, kai žodyne aprašytų faktų kiekis sukelia prieštarumą aprašytoms taisyklėms

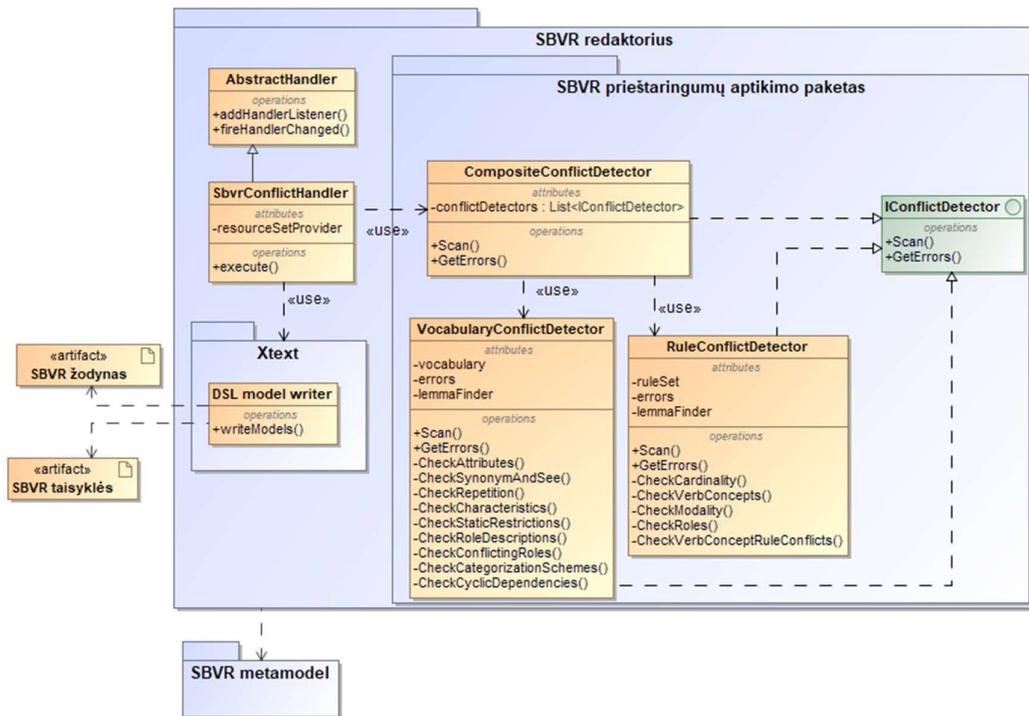
## 2.5. Reikalavimų apibendrinimas

1. Šiame darbe kuriamo metodo prototipo svarbiausi reikalavimai yra:
  - a. gebėjimas aptikti žodyno įrašų prieštarumus;
  - b. gebėjimas aptikti veiklos taisyklių prieštarumus;
  - c. rezultatų naudotojui pateikimas;
2. Sukurti algoritmai, skirti prieštarumų aptikimui veiklos žodynuose ir taisyklėse.
3. Sukurtos veiklos diagramos, kurios parodo metode naudojamus algoritmus.
4. Išskirti reikalavimai, algoritmų vizualizacijos ir jų detalūs aprašymai yra kuriamo metodo prototipo realizacinis pagrindas.

### 3. Veiklos žodynų ir taisyklių neprieštarinumo tikrinimo sprendimo prototipo projektas

#### 3.1. Sistemos loginė architektūra

Sistemos loginėje architektūroje pavaizduotas kuriamas esamo *SBVR* redaktoriaus įskiepis/paketas, kuris integruodamasis į esamą redaktoriaus funkcionalumą praplečia jį su prieštarinamų aptikimo galimybėmis *SBVR* specifikacijose. Detalizuota sistemos loginė architektūra pavaizduota 3.1 pav.



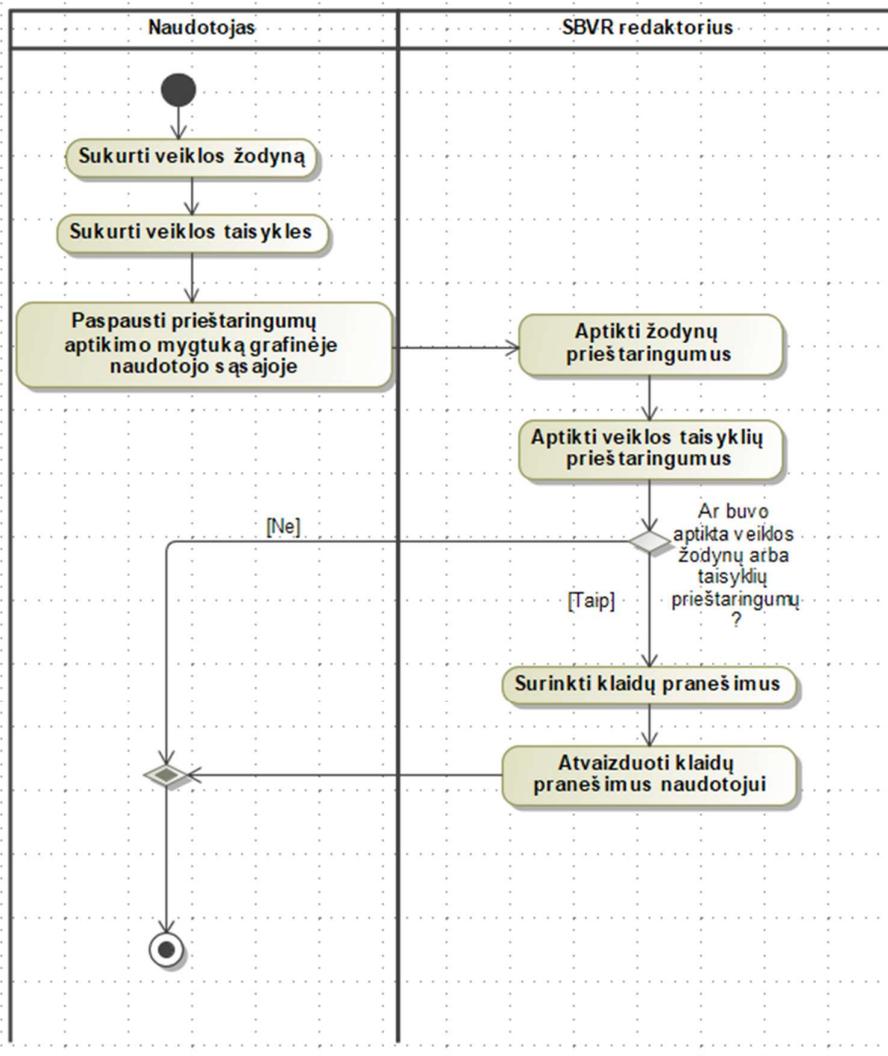
3.1 pav. Detalizuota sistemos loginė architektūra

Detalizuotoje sistemos loginėje architektūroje pateikiami pagrindiniai loginiai elementai, jų atributai, metodai ir artefaktai. Esamas *SBVR* redaktorius yra papildomas *SBVR* prieštarinamų aptikimo paketu (ang. *SBVR.Conflict.Detection*). Paketas patikrina, ar yra sukurtas žodyno ir taisyklių failas. Jeigu atitinkami failai yra sukurti, *SBVR* prieštarinamų aptikimo paketas nuskaityto žodyną ir veiklos taisykles pasinaudojant *Xtext DSL* transformacijomis į programinius objektus ir algoritmas perduoda nuskaitytą informaciją į atitinkamas prieštarinamų aptikimo klases. Žodyno nuskaityta informacija perduodama į *VocabularyConflictDetector* klasę, o veiklos taisyklių nuskaityta informacija perduodama į *RuleConflictDetector* klasę. Abi sukurtos klasės perduodamos į *CompositeConflictDetector* klasę, kuri agreguoja vaikinės klases ir perduoda atitinkamus metodus iškvietimus. Kiekvienos konkrečios klasės viduje iškvietus metodą *Scan*, yra paleidžiami prieštarinamų aptikimo algoritmai, kurie išvardinti 3.1 pav. kaip privatūs metodai. Aptikti prieštarinamumai yra saugomi klasių objektuose ir yra grąžinami pasinaudojant *GetErrors* metodu.

#### 3.2. Sistemos elgsenos modelis.

Sistemos elgsenos modelyje atvaizduojamas pagrindinis kuriamo sprendimo panaudojimo scenarijus. Sistemos elgsenos modelis pavaizduotas 3.2 pav. Naudotojas naudojantis *SBVR* redaktoriumi sukuria veiklos žodyno, taisyklių failus ir juos atitinkamai užpildo. Norėdamas patikrinti prieštarinamumus,

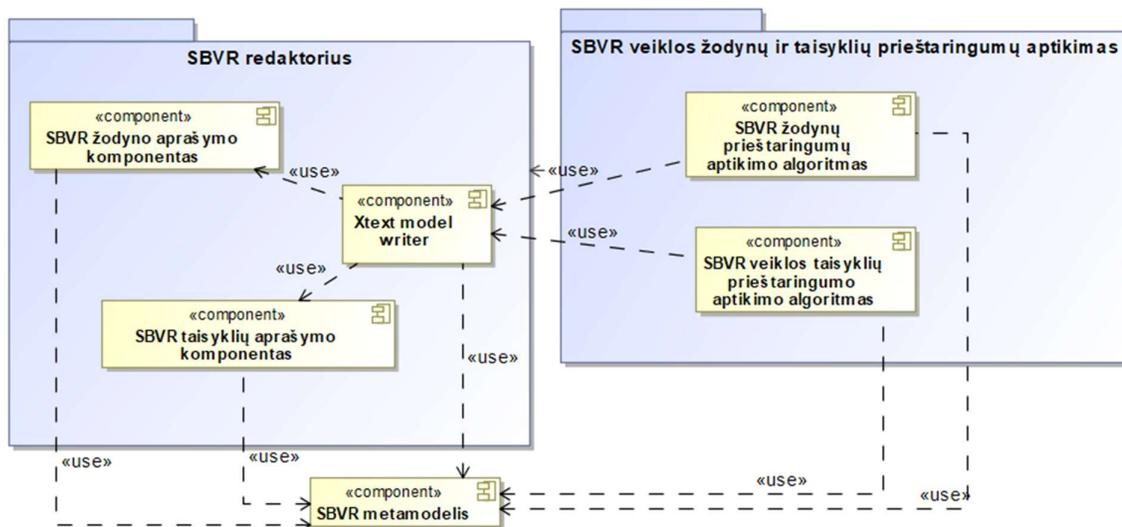
naudotojas grafinėje naudotojo sąsajoje paspaudžia prieštarigumų aptikimo mygtuką. Mygtuko paspaudimas yra prieštarigumų aptikimo algoritmo paleidimo būdas. Paspaudus mygtuką, programinė įranga nuskaito žodyno ir taisyklių failuose surašytą informaciją ir ją perduoda prieštarigumų aptikimo algoritmui. Jeigu algoritmas aptinka prieštarigumus veiklos žodyne arba taisyklėse, klaidos pranešimai yra surenkami ir atvaizduojami *SBVR* redaktoriaus lange.



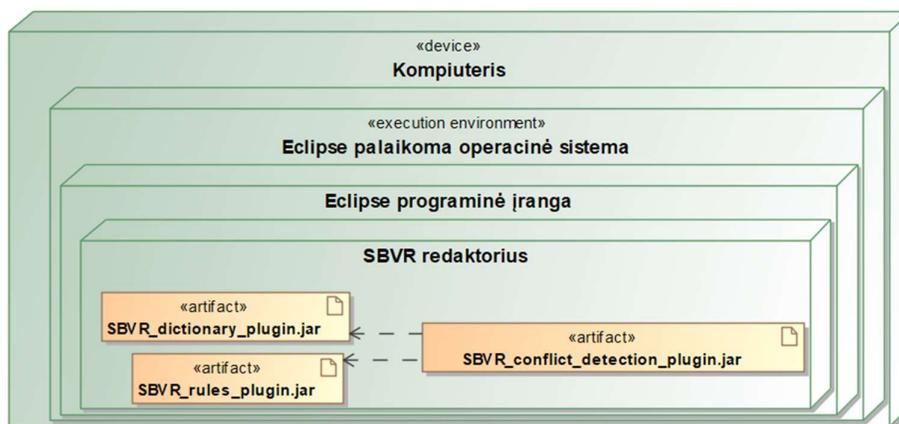
3.2 pav. Sistemos elgsenos modelis

### 3.3. Realizacijos modelis

Šiame skyriuje pateikiama sistemos komponentų diagrama pavaizduota 3.3 pav. ir diegimo diagrama pavaizduota 3.4 pav.



3.3 pav. Sistemos komponentų diagrama



3.4 pav. Sistemos diegimo diagrama

Norint įsidiesti *SBVR* veiklos žodynų ir taisyklių prieštarumų tikrinimo metodo prototipo įskiepi, privaloma turėti asmeninį kompiuterį, kuriame įdiegta *Eclipse* programinė įranga. Norint įsirašyti *Eclipse* programinę įranga, privaloma įsirašyti 64 bitų JDK (ang. *Java Development Kit*). Dokumentacijoje nurodytos palaikomos operacinės sistemos:

1. Windows;
2. Mac OS;
3. Linux.

Tikslių operacinių sistemų versijų nenurodoma. Svarbiausia, kad operacinė sistema palaikytų reikiamą *JDK* arba *JRE* versiją. *SBVR* redaktorius yra *Eclipse* programinės įrangos įskiepis, kuris turi galimybę integruotis į *Eclipse* grafinę naudotojo sąsają ir naudotis *Eclipse* programinio kodo bibliotekomis. Natūralios kalbos transformavimui į programiniu objektus yra naudojama *Xtext*

programinė įranga. Norint naudotis veiklos žodynų ir taisyklių neprieštaringumo tikrinimo metodo prototipo įskiepiu reikia prieš tai įsirašyti *SBVR* redaktoriaus įskiepius, skirtus *SBVR* žodynui ir *SBVR* veiklos taisyklėms.

### **3.4. Realizacijos projekto apibendrinimas**

1. Kuriamo *SBVR* veiklos žodynų ir taisyklių prieštaringumo tikrinimo metodo prototipo realizacijai buvo naudojamas esamas *SBVR* redaktorius.
2. Algoritmai prieštaringumų aptikimui veiklos žodynuose ir taisyklėse realizuoti naudojat *Java* programavimo kalbą.
3. Sukurtas metodo prototipas įdiegiamas kaip *Eclipse* įskiepis.
4. Programinis kodas gali būti praplečiamas pasinaudojant sukurtu duomenų nuskaitymo ir perdavimo funkcionalumu, realizuojant papildomus metodus atitinkamose žodyno arba taisyklių prieštaringumų aptikimo klasėse.

## 4. Sprendimo realizacija ir testavimas

### 4.1. Sprendimo realizacijos ir veikimo aprašas

Sukurtas sprendimas yra *SBVR* veiklos žodynų ir taisyklių neprieštarinimo tikrinimo paketas, kuris yra integruotas į jau sukurtą *SBVR* redaktorių. Sprendimas sukurtas pasinaudojant *Java* programavimo kalba, *EMF* (ang. *Eclipse Modeling Tools*) karkasu, kuris leidžia naudotis programinio kodo generavimo galimybėmis pasinaudojant struktūrizuotais duomenų modeliais. Taip pat sprendimas naudoja *Xtext* karkasą, kuris yra panaudotas *SBVR* specifikacijose naudojamos natūralios struktūrizuotos kalbos nagrinėjimui ir transformavimui į programinius objektus. *Xtext* suteikia ne tik analizatorius, kurie sugeba išgauti informaciją iš teksto, klasių modelius abstrakčiam sintaksiniam medžiui, bet ir sukuria lengvai modifikuojamą integruotą vystymo aplinką (ang. *Integrated Development Environment*) *Eclipse* įrankyje.

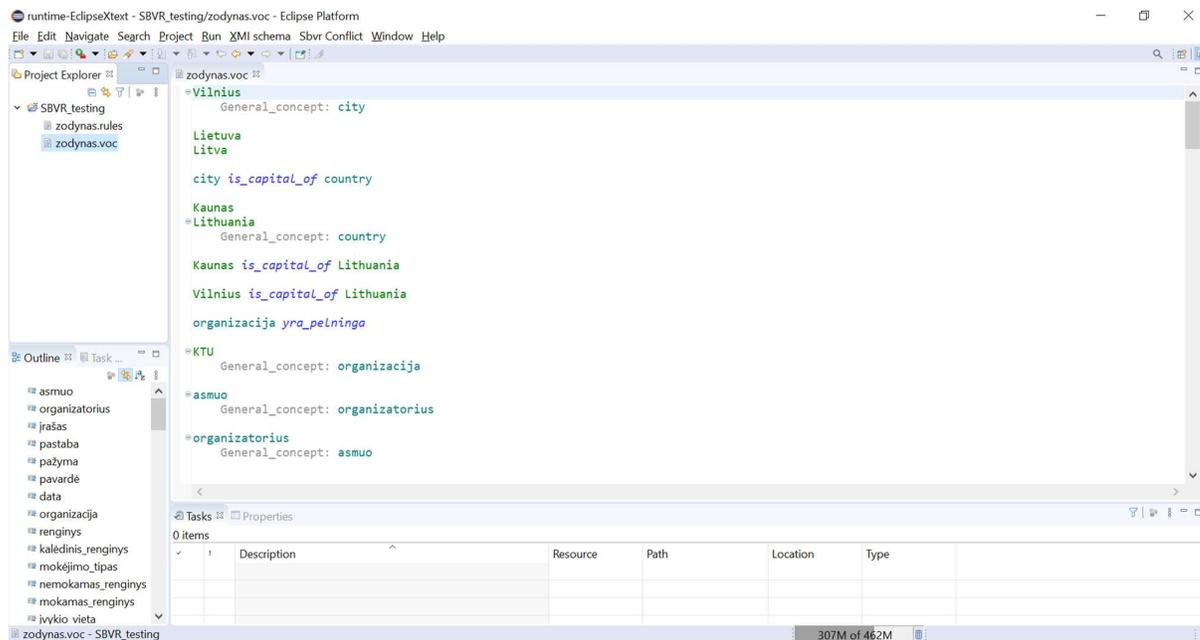
Sukurta sistema turi vieną naudotojo sąsajos langą, kuris yra naudojamas *SBVR* veiklos žodynų ir taisyklių aprašymams. Sistemos langas pavaizduotas 4.1 pav.

Norint aprašyti *SBVR* veiklos žodynus ir taisykles yra sukuriamas naujas bendros paskirties (ang. *general*) projektas ir du failai:

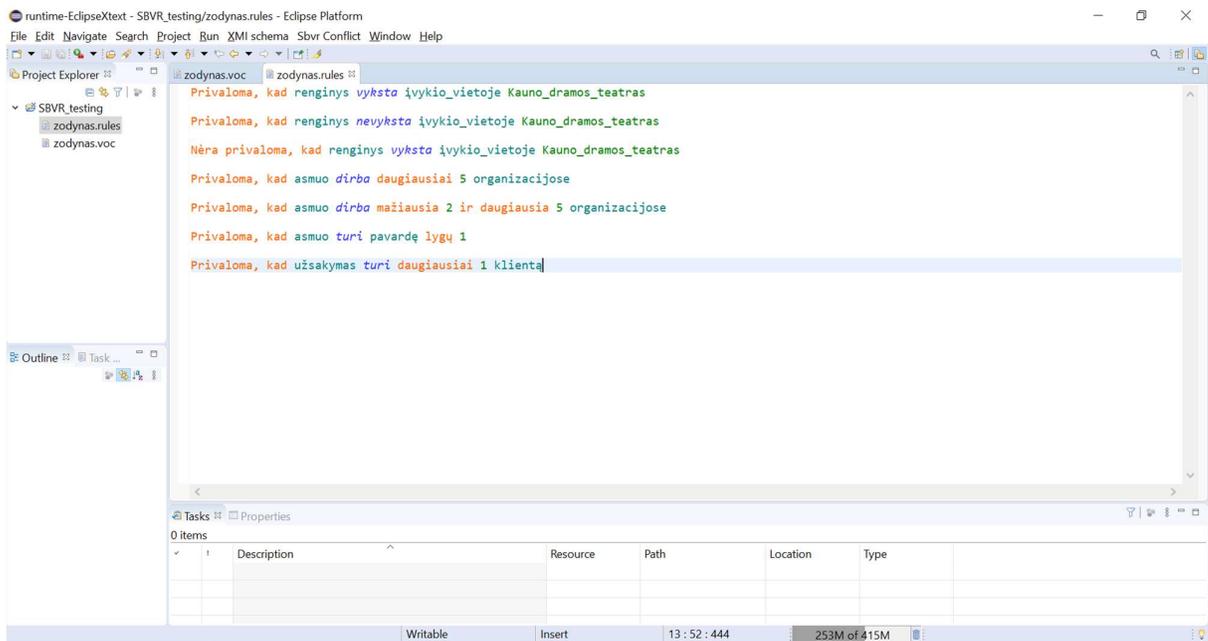
1. vienas failas yra skirtas veiklos žodynui aprašyti. Jis privalo turėti plėtinį „.voc“;
2. antras failas yra skirtas veiklos taisyklių aprašymams. Taisyklių aprašymų failas privalo turėti plėtinį „.rules“.

Abu failų pavadinimai be plėtinių privalo sutapti, kad programinė įranga galėtų aprašymus susieti.

4.1 pav. pavaizduotas programinis langas atsidarius aprašytą žodyną. 4.2 pav. pavaizduotas programinis langas, atsidarius aprašytas veiklos taisykles.

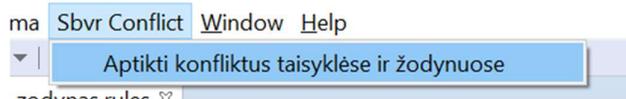


4.1 pav. *SBVR* žodyno redagavimo langas



4.2 pav. SBVR taisyklių redagavimo langas

*Xtext* programinė įranga leidžia struktūrizuotus natūralios kalbos žodžius spalvinti skirtingomis spalvomis pagal jų paskirtį arba reikšmę aprašytame *DSL*. 4.1 pav. ir 4.2 pav. pavyzdžiuose nuspalvinimai atitinka aptartą žymėjimą analizės skyriuje. Norint patikrinti veiklos žodynų ir taisyklių neprieštarinimus, yra paspaudžiamas meniu mygtukas „Aptikti konfliktus taisyklėse ir žodynuose“. Meniu mygtukas pavaizduotas 4.3 pav.



4.3 pav. Mygtukas veiklos taisyklių ir žodynų prieštarinėjimams aptikti

Veiklos žodynų ir taisyklių neprieštarinėjimo tikrinimo sprendimo prototipas leidžia:

1. susikurti žodyno failą;
2. susikurti taisyklių failą;
3. paspaudus meniu mygtuką „Aptikti konfliktus taisyklėse ir žodynuose“ nuskaityti aprašytą žodyną ir taisykles į programinius objektus ir paleisti *SBVR* veiklos žodynų ir taisyklių neprieštarinėjimo tikrinimo algoritmus;
4. aptiktus prieštarinėjimus juos atvaizduoti naudotojo sąsajoje.

Žemiau pateiktas žodyno su pasikartojančiu žodyno įrašu pavyzdys.

asmuo

pavardė

Concept\_type: role

General\_concept: text

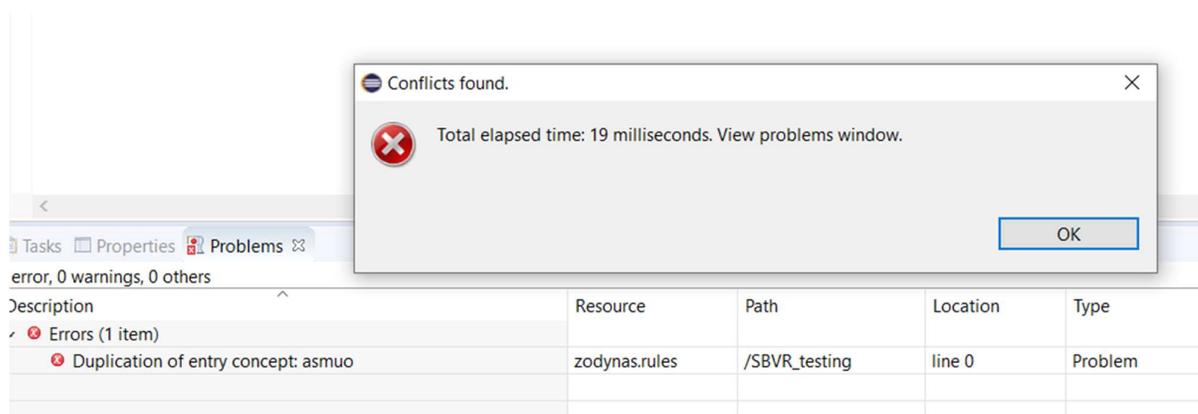
asmuo turi pavardę

organizacija

asmuo

asmuo *dirba* organizacijose

Turint aprašytą veiklos žodyną, paleidžiamas prieštarinamų tikrinimo algoritmas. Aptikto prieštarinamumo rezultatas pateikiamas naudotojo sąsajoje. Prieštarinamumo aptikimo pavyzdys pavaizduotas 4.4 pav.



4.4 pav. Aptikto prieštarinamumo klaidos pranešimas

## 4.2. Testavimo modelis, duomenys, rezultatai

Sukurto metodo prototipo realizacijos testavimui yra aprašyti minimalūs veiklos žodynų ir taisyklių fragmentai, kurie atspindi kiekvieną išskirtą prieštarinamų tipą. Veiklos žodynas ir taisyklės yra perkeliamos į realizuotą sistemą ir išbandomi prieštarinamų aptikimo algoritmai. Gauti prieštarinamų aptikimo algoritmų rezultatai yra užfiksuojami ir pateikiami 4.1 lentelėje.

4.1 lentelė. Sukurto metodo prototipo testavimo rezultatai

Prieštarinamumas	Duomenys	Rezultatas	Ar rezultatas teisingas?
1. Pasikartojantys atributai žodynų įrašuose	Šalis Lietuva Litva Lithuania General_concept: šalis See: Lietuva See: Litva	Rasta klaida individualiam konceptui Lithuania. Aptikti See atributo pasikartojimai.	Taip
2. Žodynų įrašų Synonym ir See atributų neatitikimai	įrašas pastaba Synonym: pažyma pažyma See: įrašas	Rasta klaida konceptui pažyma, nes See atributas nenurodo reikšmės pastaba.	Taip

3. Pasikartojantys įrašai žodyne	Tomas Tomas General_concept: klientas	Rasta klaida konceptui Tomas, nes konceptų pasikartojimai žodyne neleidžiami.	Taip
	Tomas Subject_field: "pažįstamų ratas" Tomas General_concept: klientas	Klaidų nerasta.	Taip
4. Prieštarinių charakteristikų aptikimas	organizacija organizacija yra pelninga organizacija yra nepelninga	Rasta klaida. Aptiktos dvi prieštaringos charakteristikos: pelninga ir nepelninga.	Taip
5. Statinių apribojimų prieštarigumai	miestas valstybė Vilnius Lietuva General_concept: valstybė miestas yra sostinė valstybės Vilnius yra sostinė Vilniaus	Rasta klaida. Individualus konceptas Vilnius neturi aprašyto bendrinio koncepto miestas.	Taip
6. Rolių neteisingas aprašymas	amžius Concept_type: role General_concept: text General_concept: number	Rasta klaida. Rolė amžius turi kelis General Concept atributus.	Taip
7. Prieštaringas rolės priskyrimas	data Concept_type: general_concept asmuo asmuo turi datą	Rasta klaida konceptui data. Koncepto tipas turėtų būti role.	Taip
8. Kategorizavimo schemų prieštarigumas	renginys mokėjimo_tipas Concept_type: categorization_type Necessity: is_for general_concept renginys Įvykiai pagal mokėjimo tipą Necessity: segmentation for general_concept renginys that subdivides renginys by mokėjimo tipą nemokamas_renginys General_concept: renginys Necessity: is_included_in Įvykiai pagal mokėjimo tipą mokamas_renginys General_concept: renginys Necessity: is_included_in Įvykiai pagal mokėjimo tipą Jazzu_renginys General_concept: mokamas_renginys General_concept: nemokamas_renginys	Rasta klaida individualiam konceptui Jazzu_renginys. Rastos kelios kategorijos priskirtos konceptui iš vieno kategorizacijos tipo.	Taip
9. Cikliškumas konceptų aprašyme	asmuo General_concept: organizatorius organizatorius General_concept: asmuo	Rastos dvi klaidos. Aptiktas cikliškumas konceptui asmuo ir	Taip

		aptiktas cikliškumas konceptui <i>organizatorius</i> .	
10. Kardinalumo neatitikimas	asmuo organizacija  asmuo <i>dirba</i> organizacijoje  Privaloma, kad asmuo <i>dirba</i> bent 3 organizacijose Privaloma, kad asmuo <i>dirba</i> daugiausiai 2 organizacijose	Rasta klaida. Taisyklių kardinalumai sukuria nepersidengiančius intervalus bendriniam konceptui: <i>organizacija</i> .	Taip
11. Veiksmažodinio koncepto prieštarumas taisyklėse	renginys įvykio_vieta  renginys <i>vyksta</i> įvykio_vietoje renginys <i>nevyksta</i> įvykio_vietoje  Kauno_dramos_teatras  Privaloma, kad renginys <i>vyksta</i> įvykio_vietoje Kauno_dramos_teatre Privaloma, kad renginys <i>nevyksta</i> įvykio_vietoje Kauno_dramos_teatre	Rasta klaida. Veiksmažodinių sąvokų prieštarumas tarp <i>vyksta</i> ir <i>nevyksta</i> .	Taip
12. Modalumo prieštarumas	renginys įvykio_vieta renginys <i>vyksta</i> įvykio_vietoje Kauno_dramos_teatras  Privaloma, kad renginys <i>vyksta</i> įvykio_vietoje Kauno_dramos_teatre Nėra privaloma, kad renginys <i>vyksta</i> įvykio_vietoje Kauno_dramos_teatre	Rasta klaida. Veiksmažodinė forma <i>vyksta</i> su bendrinio konceptu <i>įvykio_vieta</i> taisyklėje sukelia modalumo prieštarumą.	Taip
13. Rolių prieštarumas taisyklėse	pavardė Concept_type: role General_concept: text  asmuo asmuo <i>turi</i> pavardę  Privaloma, kad asmuo <i>turi</i> pavardę lygū 1	Rasta klaida. Rolė <i>pavardė</i> turi ne skaitinį tipą.	Taip
14. Veiksmažodinio koncepto taisyklės prieštarumas	užsakymas klientas Concept_type: role  užsakymas <i>turi</i> klientą  Užsakymas1 General_concept: užsakymas  Klientas1 General_concept: klientas  Klientas2 General_concept: klientas  Užsakymas1 <i>turi</i> Klientas1 Užsakymas1 <i>turi</i> Klientas2  Privaloma, kad užsakymas <i>turi</i> daugiausia 1 klientą	Rasta klaida. <i>Užsakymas1</i> turi daugiau priskirtų <i>klientas</i> esybių negu aprašytas taisyklėje maksimumas.	Taip

Atlikus testavimą išsiaiškinta, kad algoritmas geba aptikti individualius prieštaringumų tipus ir sėkmingai juos nustato pateikdamas klaidos pranešimus naudotojui.

## **5. Eksperimentinis veiklos žodynų ir taisyklių neprieštaringumo tikrinimo metodo prototipinio sprendimo tyrimas**

### **5.1. Eksperimento planas**

Vertinant sukurtą metodą, kuris yra skirtas aptikti veiklos žodynų ir taisyklių prieštarumus, yra atliekamas eksperimentas. Eksperimento eiga:

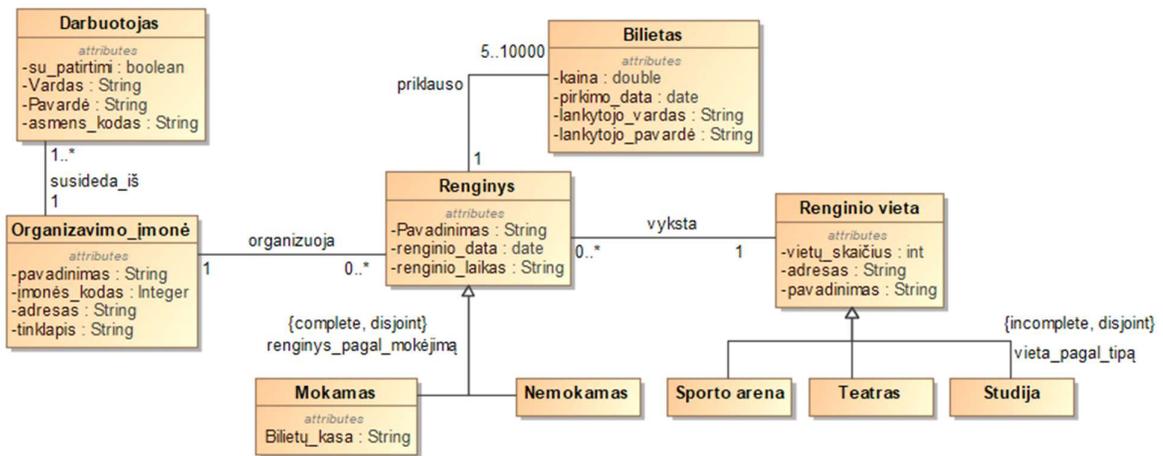
1. renginių organizavimo dalykinės srities aprašymas;
2. renginių organizavimo dalykinei sričiai skirtos veiklos žodyno ir veiklos taisyklių aprašymas;
3. papildomų konceptų ir taisyklių aprašymas esamuose renginių organizavimo dalykinės srities veiklos žodynuose ir taisyklėse siekiant sukurti prieštarumus;
4. prieštarumų aptikimas pasinaudojant šiame darbe sukurtu metodo prototipu;
5. gautų rezultatų analizė ir palyginimas su duomenimis. Išskiriami gautų rezultatų parametrai:
  - a. prieštarumų aptikimo skaičius;
  - b. aprašytų prieštarumų skaičius;
  - c. prieštarumų aptikimo tikslumas išreikštas procentu iš aptiktų prieštarumų ir aprašytų prieštarumų;
  - d. prieštarumų aptikimo tikslumas;
  - e. prieštarumų aptikimo išsamumas;
  - f. f-matas;
  - g. prieštarumų aptikimo sparta;
6. gautų rezultatų vertinimas;
7. sprendimo taikymo rekomendacijos.

### **5.2. Eksperimento rezultatai**

Eksperimento metu dalykinei sričiai sukurti veiklos žodynai ir taisyklės yra praplečiamos papildomais konceptais, siekiant apimti didesnę dalį sukurto prototipo funkcionalumo.

#### **5.2.1. Renginių organizavimo dalykinė sritis**

Siekiant eksperimentiškai ištirti sukurto metodo prototipą yra pasirinkta renginių organizavimo dalykinė sritis. Dalykinės srities esybių ryšių modelis pavaizduotas 5.1 pav. Renginių organizuoja organizavimo įmonės. Organizavimo įmonės susideda iš darbuotojų. Renginys turi pavadinimą ir renginio datą, laiką. Kiekvienam renginiui yra parduodami bilietai. Kiekvienas renginys turi priskirtą renginio vietą. Renginiai gali būti mokami arba nemokami. Renginių organizavimo dalykinei sričiai aprašomas *SBVR* veiklos žodynas pavaizduotas 5.1 lentelėje.



5.1 pav. Renginių organizavimo dalykinės srities esybių ryšių modelis

### 5.2.2. Veiklos žodyno ir veiklos taisyklių aprašymas

5.1 lentelė. Renginių organizavimo veiklos žodynas

Bendriniai konceptai
<p>organizavimo_įmonė</p> <p>darbuotojas</p> <p>bilietas</p> <p>reginys</p> <p>reginio_vieta</p>
Rolės
<p>pavadinimas</p> <p>Concept_type: role</p> <p>General_concept: text</p> <p>vardas</p> <p>Concept_type: role</p> <p>General_concept: text</p> <p>pavardė</p> <p>Concept_type: role</p> <p>General_concept: text</p> <p>asmens_kodas</p> <p>Concept_type: role</p> <p>General_concept: text</p> <p>reginio_data</p> <p>Concept_type: role</p> <p>General_concept: datetime</p> <p>pirkimo_data</p> <p>Concept type: role</p>

<p>General_concept: <b>datetime</b></p> <p><b>kaina</b>  Concept_type: <b>role</b>  General_concept: <b>number</b></p> <p><b>vietų_skaičius</b>  Concept_type: <b>role</b>  General_concept: <b>integer</b></p> <p><b>įmonės_kodas</b>  Concept_type: <b>role</b>  General_concept: <b>integer</b></p> <p><b>adresas</b>  Concept_type: <b>role</b>  General_concept: <b>text</b></p> <p><b>tinklapis</b>  Concept_type: <b>role</b>  General_concept: <b>text</b></p> <p><b>lankytojo_vardas</b>  Concept_type: <b>role</b>  General_concept: <b>text</b></p> <p><b>lankytojo_pavardė</b>  Concept_type: <b>role</b>  General_concept: <b>text</b></p> <p><b>renginio_laikas</b>  Concept_type: <b>role</b>  General_concept: <b>text</b></p> <p><b>bilietu_kasa</b>  Concept_type: <b>role</b>  General_concept: <b>text</b></p>
<b>Charakteristikos</b>
<b>darbuotojas <i>su patirtimi</i></b>
<b>Veiksmažodiniai konceptai</b>
<p>darbuotojas <i>turi</i> vardą  darbuotojas <i>turi</i> pavardę  darbuotojas <i>turi</i> asmens_kodą</p> <p>darbuotojas <i>dirba</i> organizavimo_įmonei</p> <p>organizavimo_įmonė <i>turi</i> pavadinimą  organizavimo_įmonė <i>turi</i> įmonės_kodą  organizavimo_įmonė <i>turi</i> adresą  organizavimo_įmonė <i>turi</i> tinklapį</p>

organizavimo\_įmonėje *dirba* darbuotojai  
organizavimo\_įmonė *organizuoja* renginius

renginys  *turi* pavadinimą  
renginys  *turi* renginio\_data  
renginys  *turi* renginio\_laiką

renginys *organizuojamas* organizavimo\_įmonės  
renginiui *parduodami* bilietai  
renginys *vyksta* renginio\_vietoje

bilietas  *turi* kainą  
bilietas  *turi* pirkimo\_data  
bilietas  *turi* lankytojo\_vardą  
bilietas  *turi* lankytojo\_pavarde

bilietas *priklauso* renginiui

renginio\_vieta  *turi* vietų\_skaičių  
renginio\_vieta  *turi* adresą  
renginio\_vieta  *turi* pavadinimą

renginio\_vietoje *vyksta* renginys  
mokamas\_renginys  *turi* bilietų\_kasą

#### Segmentacijos schema

mokėjimo\_tipas  
Concept\_type: categorization\_type  
Necessity: *is\_for* general\_concept renginys

Renginiai\_pagal\_mokėjimo\_tipą  
Necessity: segmentation *for* general\_concept renginys *that subdivides* renginys *by*  
mokėjimo\_tipą

nemokamas\_renginys  
General\_concept: renginys  
Necessity: *is\_included\_in* Renginiai\_pagal\_mokėjimo\_tipą

mokamas\_renginys  
General\_concept: renginys  
Necessity: *is\_included\_in* Renginiai\_pagal\_mokėjimo\_tipą

#### Kategorizavimo schema

vietos\_tipas  
Concept\_type: categorization\_type  
Necessity: *is\_for* general\_concept renginio\_vieta

sporto\_arena  
Concept\_type: vietos\_tipas

teatras

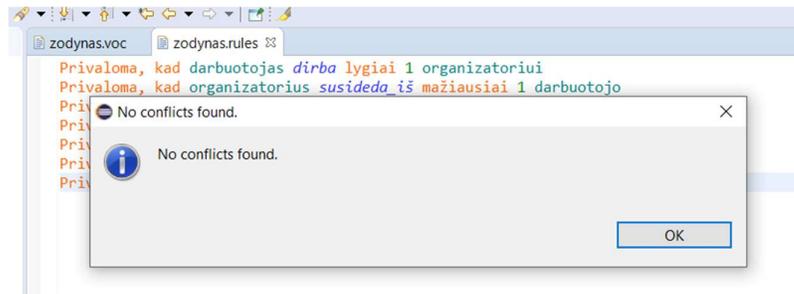
Concept_type: vietos_tipas studija Concept_type: vietos_tipas
<b>Individualūs konceptai</b>
<b>Eventus</b> General_concept: organizavimo_įmonė
<b>Tomas</b> General_concept: darbuotojas
<b>Bilietas_Nr101</b> General_concept: bilietas
<b>Kitokie_pasikalbėjimai</b> General_concept: renginys General_concept: mokamas_reginys
<b>Miesto_sodas</b> General_concept: renginio_vieta
<b>Operos_ir_baletu_teatras</b> General_concept: renginio_vieta General_concept: teatras

Renginių organizavimo dalykinei sričiai aprašomos SBVR veiklos taisyklės pavaizduotos 5.2 lentelėje.

**5.2 lentelė.** Renginių organizavimo veiklos taisyklės

<b>Veiklos taisyklės</b>
Privaloma, kad darbuotojas <i>dirba lygiai 1</i> organizavimo_įmonei Privaloma, kad organizavimo_įmonėje <i>dirba mažiausiai 1</i> darbuotojas Privaloma, kad renginys <i>organizuojamas lygiai 1</i> organizavimo_įmonės Privaloma, kad organizavimo_įmonė <i>organizuoja mažiausiai 0</i> renginių Privaloma, kad renginiui <i>parduodami mažiausiai 5 ir daugiausiai 10000</i> bilietų Privaloma, kad bilietas <i>priklauso daugiausiai 1</i> renginiui Privaloma, kad renginys <i>vyksta daugiausiai 1</i> renginio_vietoje

Perkėlus aprašytą veiklos žodyną ir veiklos taisykles į sukurto metodo prototipą ir paleidus prieštarų aptikimo algoritmą gaunamas rezultatas pavaizduotas 5.2 pav.



5.2 pav. Renginių organizavimo veiklos žodyno ir taisyklių prieštaravimo aptikimo rezultatas

### 5.2.3. Papildomų konceptų ir taisyklių aprašymas siekiant sukurti prieštaravimus

Eksperimentu metu siekiant aptinkant prieštaravimus aprašytoje veikloje, *SBVR* veiklos žodynas ir taisyklės papildomi konceptais, kurie sukuria įvairius prieštaravimus. Pridedami konceptai pavaizduoti 5.3 lentelėje.

5.3 lentelė. Renginių organizavimo veiklos papildomi konceptai prieštaravumams sukurti

Nr	Pridedamas įrašas	Paaiškinimas
1	<b>pavadinimas</b> Concept_type: <b>role</b> General_concept: <b>text</b>	Pavadinimo rolė jau buvo aprašyta žodyne, tačiau esybių diagramoje pavadinimą turi organizatorius ir renginys.
2	<b>data</b> <b>pardavimo_data</b> Concept_type: <b>role</b> General_concept: <b>datetime</b> General_concept: <b>data</b>	Aprašoma rolė su keliais bendriniais konceptais. Rolė veiklos žodynų aprašymuose gali turėti vieną tipą.
3	<b>darbuotojas</b> Synonym: <b>darbininkas</b> Synonym: <b>savanoris</b> <b>darbininkas</b> See: <b>darbuotojas</b> <b>savanoris</b> See: <b>darbuotojas</b> <b>renginio_vieta</b> Synonym: <b>vieta</b> <b>vieta</b>	Veiklos žodynas praplečiamas sinonimais. Pridedami korektiški sinonimų aprašymai ir nekorektiški sinonimų aprašymai.
4	<b>vieta</b> Concept_type: <b>role</b> Concept_type: <b>general_concept</b>	Aprašomi priešaringi koncepto atributai. Konceptas negali vienu metu būti role ir bendrinium konceptu.
5	<b>bilietas kainuoja kaina</b> <b>Bilietas_Nr102</b> <b>Bilietas_Nr103</b>	Aprašomas veiksmožodinis konceptas, kuris atitinka statinį apribojimą aprašomiems faktams. Bilietų individualūs

	<p>General_concept: kaina</p> <p>100_eur</p> <p>General_concept: kaina</p> <p>Bilietas_Nr102 kainuoja 100_eur</p> <p>Bilietas_Nr103 kainuoja 100_eur</p>	<p>konceptai aprašomi netinkamai. Bilietas_Nr102 neturi nurodyto bendrinio koncepto, o Bilietas_Nr103 turi netinkamą bendrinį konceptą.</p>
6	<p>vip_bilietas</p> <p>General_concept: bilietas</p> <p>Lietuvos_talentai</p> <p>General_concept: renginys</p> <p>Bilietas_Nr105</p> <p>General_concept: vip_bilietas</p> <p>Lietuvos_talentai parduodamas Bilietas_Nr105</p>	<p>Aprašomas konkretesnis bilietų tipas vip_bilietas, kuris priskiriamas individualiam konceptui Bilietas_105. Aprašomas faktas atitinkantis statinį apribojimą.</p>
7	<p>data</p> <p>renginys turi data</p>	<p>Aprašomas rolės priskyrimas bendriniam konceptui, tačiau rolė aprašoma kaip bendrinis konceptas.</p>
8	<p>spalva</p> <p>Concept_type: categorization_type</p> <p>bilietas turi spalva</p>	<p>Aprašomas rolės priskyrimas, tačiau priskiriama ne rolė.</p>
9	<p>Karmen_spektaklis</p> <p>General_concept: mokamas_reginys</p> <p>General_concept: nemokamas_reginys</p> <p>General_concept: renginys</p>	<p>Aprašomas individualus konceptas su priskirtais kategorijų tipais iš tos pačios segmentacijos schemas.</p>
10	<p>Žalgirio_arena</p> <p>General_concept: sporto_arena</p> <p>General_concept: studija</p>	<p>Aprašomas individualus konceptas su priskirtais kategorijų tipais iš tos pačios kategorizacijos schemas.</p>
11	<p>Renginys_ir_vieta_viename</p> <p>General_concept: mokamas_reginys</p> <p>General_concept: sporto_arena</p>	<p>Aprašomas individualus konceptas, kuris turi du kategorijų tipus iš skirtingų kategorizacijos schemų</p>
12	<p>organizatorius</p> <p>General_concept: darbuotojas</p> <p>darbuotojas</p> <p>General_concept: organizatorius</p> <p>Synonym: darbininkas</p> <p>Synonym: savanoris</p>	<p>Aprašomi bendriniai konceptai, kurių General_Concept atributai rodo vienas į kitą taip sukuriant cikliškumą konceptų aprašymuose.</p>
13	<p>plotis</p> <p>General_concept: spalva</p> <p>aukštis</p> <p>General_concept: plotis</p> <p>spalva</p> <p>Concept type: categorization type</p>	<p>Aprašomas didesnio gylio konceptų cikliškumas.</p>

	General concept: aukstis	
14	Privaloma, kad darbuotojas <i>dirba lygiai 1</i> organizatoriui  Privaloma, kad darbuotojas <i>dirba mažiausiai 2</i> organizatoriams	Aprašoma taisyklė sukuria prieštarumą esamai taisyklei.
15	Privaloma, kad renginys <i>vyksta mažiausiai 5 ir daugiausiai 0</i> renginio_vietoje	Aprašoma taisyklė, kurios kardinalumai aprašyti prieštarinai.
16	Privaloma, kad bilietas <i>nepriklauso daugiausiai 1</i> renginiui	Aprašoma taisyklė su priešingo poliškumo veiksmožodiniu konceptu.
17	Nėra privaloma, kad renginys <i>vyksta daugiausiai 1</i> renginio_vietoje	Aprašoma taisyklė su modalumo prieštarinumu esamai taisyklei
18	Privaloma, kad organizatorius <i>turi pavadinimą lygų 1</i>	Aprašoma taisyklė apribojanti rolės reikšmių sritį, tačiau apribojimo tipas neatitinka rolės galimas reikšmes.
19	Bilietas_Nr104 General_concept: bilietas  Kitokie_pasikalbėjimai <i>parduodamas Bilietas_Nr104</i>  Privaloma, kad renginiui <i>parduodami mažiausiai 5 ir daugiausiai 10000</i> bilietų	Aprašoma taisyklė, kuri apriboja veiksmožodinio koncepto esybių skaičių priskirtų pirmajai veiksmožodinio koncepto rolei, tačiau aprašomų faktų kiekis neatitinka taisyklės apribojimo.
20	Nėra įmanoma, kad bilietas <i>priklauso daugiausiai 1</i> renginiui	Aprašoma taisyklė su modalumo prieštarinumu esamai taisyklei.

Pilnas *SBVR* renginių organizavimo veiklos žodynas ir taisyklių rinkinys su papildytais papildomais įrašais siekiant sukurti prieštarinumus pateiktas 1 priede.

#### 5.2.4. Prieštarinimų aptikimas pasinaudojant šiame darbe sukurtu metodo prototipu

Paleidžiamas prieštarinimų aptikimo metodo prototipas ir fiksuojami gauti rezultatai. Informacinio pranešimo naudotojui langas pavaizduotas 5.3 pav.

The screenshot shows a 'Conflicts found' dialog box with a red 'X' icon and the text: 'Total elapsed time: 20 milliseconds. View problems window.' Below the dialog, a table lists 25 errors. The table has columns: Description, Resource, Path, Location, and Type. The errors include cycles for 'aukstis', 'darbuotojas', 'Eventus', 'organizatorius', 'plotis', and 'Tomas'; duplication of 'pavadinimas'; and various errors related to concept attributes and types.

Description	Resource	Path	Location	Type
✖ Cycle was detected for: aukstis	zodynas.rules	/SBVR_testing	line 0	Problem
✖ Cycle was detected for: darbuotojas	zodynas.rules	/SBVR_testing	line 0	Problem
✖ Cycle was detected for: Eventus	zodynas.rules	/SBVR_testing	line 0	Problem
✖ Cycle was detected for: organizatorius	zodynas.rules	/SBVR_testing	line 0	Problem
✖ Cycle was detected for: plotis	zodynas.rules	/SBVR_testing	line 0	Problem
✖ Cycle was detected for: spalva	zodynas.rules	/SBVR_testing	line 0	Problem
✖ Cycle was detected for: Tomas	zodynas.rules	/SBVR_testing	line 0	Problem
✖ Duplication of entry concept: pavadinimas	zodynas.rules	/SBVR_testing	line 0	Problem
✖ Entry concept vieta did not have attribute See with value: rengin	zodynas.rules	/SBVR_testing	line 0	Problem
✖ Error at concept: data. This concept should be of type: role.	zodynas.rules	/SBVR_testing	line 0	Problem
✖ Error at concept: spalva. This concept should be of type: role.	zodynas.rules	/SBVR_testing	line 0	Problem
✖ Error at GeneralConcept: It.ktu.isd.semantika.vocabulary.impl.De	zodynas.rules	/SBVR_testing	line 0	Problem
✖ Error at Individual concept: Karmen_spektaklis. Multiple categori	zodynas.rules	/SBVR_testing	line 0	Problem
✖ Error at Individual concept: Žalgirio_arena. Multiple categories c	zodynas.rules	/SBVR_testing	line 0	Problem
✖ Error at Role: It.ktu.isd.semantika.vocabulary.impl.DesignationOf	zodynas.rules	/SBVR_testing	line 0	Problem
✖ Faktas: Bilietas_Nr103 kainuoja 100_eur individualūs konceptai ne	zodynas.rules	/SBVR_testing	line 0	Problem
✖ Faktas: Lietuvos_talentai parduodami Bilietas_Nr105 individualūs	zodynas.rules	/SBVR_testing	line 0	Problem
✖ General concept: renginio_vieta su veiksmažodine forma: vyksta	zodynas.rules	/SBVR_testing	line 0	Problem
✖ General concept: renginio_vieta su veiksmažodine forma: vyksta	zodynas.rules	/SBVR_testing	line 0	Problem
✖ Individualus konceptas: Bilietas_Nr102 neturi bendrinio koncept	zodynas.rules	/SBVR_testing	line 0	Problem
✖ Role renginys - Kitokie_pasikalbėjimai has wrong number of inst	zodynas.rules	/SBVR_testing	line 0	Problem
✖ Role: pavadinimas has non numeric primary type.	zodynas.rules	/SBVR_testing	line 0	Problem
✖ Role: pavadinimas has non numeric primary type.	zodynas.rules	/SBVR_testing	line 0	Problem

5.3 pav. Eksperimento metu gauto informacinio pranešimo langas

Algoritmo rezultatas tekstone forma pateikiamas 2 priede. Iš viso algoritmas aptiko 25 prieštaravimus aprašytame veiklos žodyne ir taisyklėse. Visi aptikti prieštaravimai yra dėl 5.3 lentelėje aprašytų konceptų ir taisyklių pridėjimo. 5.4 lentelėje pateikiami prieštaravimų aptikimo rezultatai su išskirtais prieštaravimais, aptikimo įrašais iš prototipo rezultato ir aprašas, kuriame fiksuojamas veikimo korektiškumas.

5.4 lentelė. Prieštaravimų aptikimo rezultatų palyginimas

Nr.	Prieštaravimas	Aptikimo įrašas	Prieštaravumo atitikimas
1	<b>pavadinimas</b> Concept_type: <b>role</b> General_concept: <b>text</b>	10. Duplication of entry concept: pavadinimas	Prieštaravumas aptiktas
2	<b>data</b> <b>pardavimo_data</b> Concept_type: <b>role</b> General_concept: <b>datetime</b> General_concept: <b>data</b>	14. Error at Role: pardavimo_data. Multiple general_concept captions found: General_concept:	Prieštaravumas aptiktas
3	<b>darbuotojas</b> Synonym: <b>darbininkas</b> Synonym: <b>savanoris</b> <b>darbininkas</b> See: <b>darbuotojas</b>	9. Entry concept vieta did not have attribute See with value: renginio_vieta	Prieštaravumas aptiktas

	<p>savanoris See: darbuotojas</p> <p>renginio_vieta Synonym: vieta</p> <p>vieta</p>		
4	<p>vieta Concept_type: role Concept_type: general_concept</p>	8. Error at GeneralConcept: vieta. Multiple captions found: Concept_type:	Prieštaringumas aptiktas
5	<p>bilietas kainuoja kaina</p> <p>Bilietas_Nr102 Bilietas_Nr103 General_concept: kaina</p> <p>100_eur General_concept: kaina</p> <p>Bilietas_Nr102 kainuoja 100_eur Bilietas_Nr103 kainuoja 100_eur</p>	<p>11. Individualus konceptas: Bilietas_Nr102 neturi bendrinio koncepto</p> <p>12. Fakto: Bilietas_Nr103 kainuoja 100_eur individualūs konceptai neturi bendrinių konceptų, kurie atitiktų statinį apribojimą.</p>	Prieštaringumai aptikti
6	<p>vip_bilietas General_concept: bilietas</p> <p>Lietuvos_talentai General_concept: renginys</p> <p>Bilietas_Nr105 General_concept: vip_bilietas</p> <p>Lietuvos_talentai parduodamas Bilietas_Nr105</p>	13. Fakto: Lietuvos_talentai parduodami Bilietas_Nr105 individualūs konceptai neturi bendrinių konceptų, kurie atitiktų statinį apribojimą.	Prieštaringumas aptiktas, nors prieštaravimo nėra. Algoritmas nesugebėjo nustatyti bendrinio koncepto hierarchijos.
7	<p>data renginys turi data</p>	15. Error at concept: data. This concept should be of type: role.	Prieštaringumas aptiktas
8	<p>spalva Concept_type: categorization_type</p> <p>bilietas turi spalva</p>	16. Error at concept: spalva. This concept should be of type: role.	Prieštaringumas aptiktas
9	<p>Karmen_spektaklis General_concept: mokamas_renginys General_concept: nemokamas_renginys General_concept: renginys</p>	17. Error at Individual concept: Karmen_spektaklis. Multiple categories of one categorization scheme was assigned.	Prieštaringumas aptiktas
10	<p>Žalgirio_arena General_concept: sporto_arena General_concept: studija</p>	18. Error at Individual concept: Žalgirio_arena. Multiple categories of one categorization scheme was assigned.	Prieštaringumas aptiktas

11	Renginys_ir_vieta_viename General_concept: mokamas_renginys General_concept: sporto_arena	-	Prieštaringumas nerastas, nes prieštaravimo koncepto aprašyme nėra.
12	organizatorius General_concept: darbuotojas darbuotojas General_concept: organizatorius Synonym: darbininkas Synonym: savanoris	19. Cycle was detected for: organizatorius 20. Cycle was detected for: darbuotojas 21. Cycle was detected for: Eventus 22. Cycle was detected for: Tomas	Prieštaringumas aptiktas, tačiau vienam prieštaravimui išvesti 4 pranešimai. Po vieną kiekvienam cikliškume esantį konceptą.
13	plotis General_concept: spalva aukstis General_concept: plotis spalva Concept_type: categorization_type General_concept: aukstis	23. Cycle was detected for: plotis 24. Cycle was detected for: aukstis 25. Cycle was detected for: spalva	Prieštaringumas aptiktas, tačiau vienam prieštaravimui išvesti 3 pranešimai. Po vieną kiekvienam cikliškume esantį konceptą.
14	Privaloma, kad darbuotojas dirba lygiai 1 organizatoriui  Privaloma, kad darbuotojas dirba mažiausiai 2 organizatoriams	1. Taisyklė su kardinalumu bendriniam konceptui: organizatorius sukuria kardinalumų prieštaravimą.	Prieštaringumas aptiktas.
15	Privaloma, kad renginys vyksta mažiausiai 5 ir daugiausiai 0 renginio_vietoje	2. WARNING. Viena iš taisyklių su kardinalumu yra perteklinė bendriniam konceptui: renginio_vieta.	Algoritmas išveda pranešimą dėl perteklinės taisyklės dėl kitos taisyklės aprašančios kardinalumą renginio_vietai. Esamas prieštaravimas aprašytas taisyklėje neaptiktas.
16	Privaloma, kad bilietas nepriklauso daugiausiai 1 renginiui	-	Prieštaringumas neaptiktas
17	Nėra privaloma, kad renginys vyksta daugiausiai 1 renginio_vietoje	3. General concept: renginio_vieta su veiksmožodine forma: vyksta sukelia modalumo prieštaravimą. 4. General concept: renginio_vieta su veiksmožodine forma: vyksta	Prieštaringumas aptiktas ir išvesti du pranešimai dėl pasikartojančių taisyklių turinčių priešingo poliškumo modalumą.

		sukelia modalumo prieštaravimą.	
18	Privaloma, kad organizatorius turi pavadinimą lygų 1	5. Role: pavadinimas has non numeric primary type. 6. Role: pavadinimas has non numeric primary type.	Prieštaringumas aptiktas, tačiau išvestas perteklinis pranešimas.
19	Bilietas_Nr104 General_concept: bilietas  Kitokie_pasikalbėjimai parduodamas Bilietas_Nr104  Privaloma, kad renginiui parduodami mažiausiai 5 ir daugiausiai 10000 bilietų	7. Role renginys - Kitokie_pasikalbėjimai has wrong number of instances of bilietas. Minimum: 5, maximum: 10000	Prieštaringumas aptiktas.
20	Nėra įmanoma, kad bilietas priklauso daugiausiai 1 renginiui	-	Prieštaringumas neaptiktas.

### 5.2.5. Sprendimo veikimo ir savybių analizė, kokybės kriterijų įvertinimas

Siekiant įvertinti sukurto prieštaringumų aptikimo metodo prototipo veikimą, apskaičiuojamas tikslumas (ang. *precision*), išsamumas (ang. *recall*) ir f-matas. Klaidų matrica (ang. *confusion matrix*) pateikiama 5.5 lentelėje. Faktiniai įrašai nurodo esamus prieštaringumus aprašytame veiklos žodyne ir taisyklėse, o nustatyti įrašai nurodo algoritmu nustatytus prieštaringumus tame pačiame veiklos žodyne ir taisyklėse. Klaidų matricos reikšmių paaiškinimai:

- skaičius 16 nurodo teisingai nustatytų prieštaringumų skaičių;
- skaičius 2 nurodo neaptiktų prieštaringumų skaičių;
- skaičius 9 nurodo netinkamai aptiktus prieštaringumus kai prieštaringumo nėra arba prieštaringumas yra pasikartojantis;
- skaičius 66 nurodo žodyno ir taisyklių įrašų skaičių, kurie nesukėlė prieštaringumų.

5.5 lentelė. Klaidų matrica

		Faktinis	
		Teigiamas	Neigiamas
Nustatytas	Teigiamas	16	9
	Neigiamas	2	66

Tikslumas yra algoritmo gebėjimas nustatyti tik tinkamus prieštaringumus.

Išsamumas yra algoritmo gebėjimas nustatyti visus prieštaringumus.

F-matas yra tikslumą ir išsamumą apjungianti metrika.

Apskaičiuojamas tikslumas ir išsamumas pagal formules pavaizduotas 5.4 pav.

$$recall = \frac{true\ positives}{true\ positives + false\ negatives} \quad precision = \frac{true\ positives}{true\ positives + false\ positives}$$

5.4 pav. Išsamumo ir tikslumo formulės [30]

$$Tikslumas = \frac{16}{16 + 9} = 0.64$$

$$Išsamumas = \frac{16}{16 + 2} = 0.8889$$

Toliau apskaičiuojamas f-matas pagal formulę pavaizduotą 5.5 pav.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

5.5 pav. f-mato apskaičiavimo formulė [30]

$$F1 = 2 * \frac{0.64 * 0.8889}{0.64 + 0.8889} = 0.74419$$

Sukurtas metodo prototipas geba aptikti prieštarigumus veiklos žodynuose ir taisyklėse. Eksperimentiškai ištyrus prototipą buvo išskirti rezultatų parametrai ir pavaizduoti 5.6 lentelėje.

5.6 lentelė. Eksperimento gautų rezultatų parametrai

Parametras	Reikšmė
Aprašytų prieštarigumų skaičius	18
Teisingai aptiktų prieštarigumų skaičius	16
Teisingai aptiktų prieštarigumų išraiška procentais	89%
Neaptiktų prieštarigumų skaičius	2
Neteisingai aptiktų prieštarigumų skaičius	9
Prieštarigumų aptikimo tikslumas (ang. <i>precision</i> )	64%
Išsamumas (ang. <i>recall</i> )	89%
f-matas	74%
Prieštarigumų aptikimo sparta	30ms - 100ms

## 6. Savybių analizė ir kriterijų įvertinimas

Sukurtas metodo prototipas išveda didesnę kiekį aptiktų prieštarigumų negu prieštarigumų iš tikrųjų yra tam tikriems algoritmo žingsniams. Iš eksperimento rezultatų galima teigti, kad didesni

kiekiai prieštaringumų išvedami aptinkant konceptų cikliškumus ir taisyklių modalumo prieštaringumus. Apskaičiavus tikslumo, išsamumo ir f-mato metrikas paaiškėjo, kad prieštaringumų aptikimo algoritmai bent vieną klaidos pranešimą išvedė 89% visų aprašytų prieštaringumų, tačiau išveda perteklinių prieštaringumų aptikimų, todėl algoritmų tikslumas yra tik 64%. F-matas (ang. *F1 measure*) lygus 74%. Algoritmai nesusidoroja su konceptų hierarchijomis tikrinant atributus individualiems ir bendriniam konceptams ir bandant atrasti sąryšį su statiniais apribojimais ir/arba taisyklėmis. Atlikus eksperimento rezultatų parametrų analizę galima teigti, kad sukurtas algoritmas pakankamai tiksliai nustato nesudėtingų konceptų aprašymų prieštaringumus ir padėtų aptikti prieštaringumus *SBVR* veiklos žodynų ir taisyklių specifikacijose.

## **7. Sprendimo taikymo rekomendacijos**

1. Tolimesniam sistemos vystymui rekomenduojama papildyti prieštaringumų aptikimą algoritmais, kurie sugeba atsekti konceptų hierarchijas ir jas panaudoti aptinkant konceptų sąryšius.
2. Patogiausia aprašinėjant veiklos žodyną ir taisykles periodiškai patikrinti, ar neatsirado prieštaringumų, siekiant sumažinti vienu metu gautų pranešimų skaičių.
3. Sukurto sprendimo taikymas yra limituotas aprašytiems prieštaringumų aptikimams, kurie nepadengia visų įmanomų prieštaringumų *SBVR* specifikacijose.

## Išvados

1. Kuriant programinę įrangą, veiklos žodyno ir taisyklių aprašymas struktūrizuota natūralia kalba palengvina komunikaciją tarp kuriamos programinės įrangos užsakovų ir informacinių sistemų kūrėjų. Išnagrinėtas *SBVR* standartas leidžia aprašyti veiklos žodyną ir taisykles struktūrizuota natūralia kalba, kuri gali būti interpretuojama kompiuterinių sistemų ir transformuojama į programinį kodą.
2. Aprašant veiklos žodynus ir taisykles, yra neišvengiamos klaidos, kurios atsiranda dėl standarto nesilaikymo arba loginio taisyklių prieštaravimo. Kadangi kuriant informacines sistemas, veiklos žodynai ir taisyklės naudojamos kaip reikalavimai kuriamai sistemai, neištaisyti prieštaravimai gali nulemti netinkamą sukurtos sistemos veikimą.
3. Išanalizavus esamus *SBVR* prieštaravimų aptikimo metodus nustatyta, kad daugelis esamų prieštaravimų aptikimo sprendimų neturi automatizavimo galimybių ir naudojami konvertavimu į kitą kalbą, kuri turi geriau ištirtus metodus ir įrankius, tačiau sumažina *SBVR* standarto išraiškumo galimybes. Išanalizuoti esami metodai aprašo sąsajas/transformacijas iš *SBVR* į kitą kalbą ir naudoja jau sukurtus įrankius bandant aptikti prieštaravimus.
4. Šiame darbe sukurtas metodas apima 14 skirtingų algoritmų, kurie leidžia automatiškai aptikti prieštaravimus veiklos žodynuose ir taisyklėse išskirtus analizuojant *SBVR* standartą ir specifikacijas.
5. Metodas buvo realizuotas kaip esamo *SBVR* redaktoriaus papildymas, kuris leidžia aptikti prieštaravimus *SBVR* specifikacijose nenaudojant jokių tarpinių atvaizdavimo kalbų.
6. Eksperimento metu nustatyta, kad sukurtas prototipas sugeba nustatyti analizės etape išskirtus prieštaravimus. Algoritmai nustato 89% visų dalykinei sričiai sukurtų veiklos žodyno ir taisyklių prieštaravimų. Algoritmų tikslumas siekia 64% ir jų f-matas lygus 74%. Algoritmai nesugeba aptikti ryšių tarp konceptų jeigu panaudojami konceptai iš konceptų hierarchijų.
7. Eksperimento rezultatai leidžia teigti, kad sukurtas prototipas yra praktiškai pritaikomas ir sudaro sąlygas prieštaravimų aptikimui *SBVR* specifikacijose. Plėtojant metodą būtų galima praplėsti prieštaravimų aptikimo algoritmus su konceptų hierarchijų panaudojimais ir sąryšių tarp tokių konceptų aptikimu.

## Literatūros sąrašas

- [1] „Uniform Resource Identifier,“ 22 01 2020. [Tinkle]. Available: [https://en.wikipedia.org/wiki/Uniform\\_Resource\\_Identifier](https://en.wikipedia.org/wiki/Uniform_Resource_Identifier). [Kreiptasi 26 01 2020].
- [2] „Systems development life cycle,“ Wikipedia, 21 04 2020. [Tinkle]. Available: [https://en.wikipedia.org/wiki/Systems\\_development\\_life\\_cycle](https://en.wikipedia.org/wiki/Systems_development_life_cycle). [Kreiptasi 01 05 2020].
- [3] „santoshdhital,“ [Tinkle]. Available: <https://www.santoshdhital.com.np/system-development-phase/>. [Kreiptasi 01 05 2020].
- [4] P. K. C. Sayandeep Mitra, „A Systematic Review of Methods for Consistency Checking in SBVR-based Business Rules,“ 2017.
- [5] C. S. S. D. F. G. Keri E. Pearlson, *Managing and Using Information Systems: A Strategic Approach*, John Wiley & Sons, 2019.
- [6] A. Šukys, L. Ablonskis, L. Nemuraitė ir B. Paradauskas, „A Grammar for ADVANCED SBVR Editor,“ *INFORMATION TECHNOLOGY AND CONTROL*, t. 45, nr. 1, pp. ISSN 1392–124X (print), ISSN 2335–884X (online), 2016.
- [7] M. H. Linehan, „SBVR Use Cases,“ įtraukta *Rule Representation, Interchange and Reasoning on the Web: International Symposium, RuleML*, Orlando, 2008.
- [8] „Semantics of Business Vocabulary and Business Rules,“ Wikipedia, 25 06 2019. [Tinkle]. Available: [https://en.wikipedia.org/wiki/Semantics\\_of\\_Business\\_Vocabulary\\_and\\_Business\\_Rules](https://en.wikipedia.org/wiki/Semantics_of_Business_Vocabulary_and_Business_Rules). [Kreiptasi 11 01 2020].
- [9] OMG, „Semantics of Business Vocabulary and Business Rules,“ 19 10 2019. [Tinkle]. Available: <https://www.omg.org/spec/SBVR/1.5/PDF>. [Kreiptasi 11 01 2020].
- [10] OMG, „SBVR XML Schema in terms of XMI,“ [Tinkle]. Available: <https://www.omg.org/spec/SBVR/20190601/SBVR-XML-Schema.xsd>. [Kreiptasi 11 01 2020].
- [11] OMG, „SBVR Metamodel in terms of XMI,“ [Tinkle]. Available: <https://www.omg.org/spec/SBVR/20190601/SBVR-XMI-Metamodel.xml>. [Kreiptasi 11 01 2020].
- [12] No Magic, „Overlapping and incomplete subclasses,“ No Magic, 2020. [Tinkle]. Available: <https://docs.nomagic.com/display/CCMP190/Overlapping+and+incomplete+subclasses>. [Kreiptasi 21 04 2020].
- [13] „Wikipedia,“ 28 02 2020. [Tinkle]. Available: [https://en.wikipedia.org/wiki/Linguistic\\_modality](https://en.wikipedia.org/wiki/Linguistic_modality). [Kreiptasi 20 04 2020].
- [14] R. Dury, „A BRIEF GLOSSARY OF MODALITY,“ [Tinkle]. Available: <https://dinamico2.unibg.it/anglistica/slin/modgloss.htm>. [Kreiptasi 05 04 2020].
- [15] The Library of Congress Network Development & MARC Standards Office, „ISO 639-2,“ 18 10 2010. [Tinkle]. Available: <https://www.loc.gov/standards/iso639-2/>. [Kreiptasi 26 01 2020].

- [16] „Wikipedia,“ 10 12 2019. [Tinkle]. Available: <https://en.wikipedia.org/wiki/Contradiction>. [Kreiptasi 23 01 2020].
- [17] C. Hashimoto, K. Torisawa, S. D. Saeger, J. Oh ir J. Kazama, „Excitatory or inhibitory: a new semantic orientation extracts contradiction and causality from the web,“ įtraukta *Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012.
- [18] K. A. Pavan Kumar Chittimalli, „Domain-Independent Method of Detecting Inconsistencies in SBVR-Based Business Rules,“ *Proceedings of the International Workshop on Formal Methods for Analysis of Business Systems*, pp. 9-16, 2016.
- [19] „First Order Logic,“ Wikipedia, 16 02 2020. [Tinkle]. Available: [https://en.wikipedia.org/wiki/First-order\\_logic](https://en.wikipedia.org/wiki/First-order_logic). [Kreiptasi 29 02 2020].
- [20] „Satisfiability Modulo Theories,“ Wikipedia, 24 01 2020. [Tinkle]. Available: [https://en.wikipedia.org/wiki/Satisfiability\\_modulo\\_theories](https://en.wikipedia.org/wiki/Satisfiability_modulo_theories). [Kreiptasi 29 02 2020].
- [21] D. R. Cok, „The SMT-LIBv2 Language and Tools,“ 23 11 2013. [Tinkle]. Available: <https://smtlib.github.io/jSMTLIB/SMTLIBTutorial.pdf>. [Kreiptasi 29 02 2020].
- [22] L. A. L. N. B. P. Jaroslav Karpovič, „Experimental Investigation of Transformations from SBVR Business Vocabularies and Business Rules to Owl 2 Ontologies,“ *INFORMATION TECHNOLOGY AND CONTROL*, t. 45, nr. 2, pp. 195-207, 2016.
- [23] L. N. Jaroslav Karpovic, „TRANSFORMING SBVR BUSINESS SEMANTICS INTO WEB ONTOLOGY LANGUAGE OWL2: MAIN CONCEPTS,“ *Proceedings of the 17th international conference on Information and Software Technologies*, pp. 231-238, 2011.
- [24] L. N. M. S. Jaroslav Karpovic, „Requirements for Semantic Business Vocabularies and Rules for Transforming Them into Consistent OWL2 Ontologies,“ *International Conference on Information and Software Technologies*, pp. 420-435, 2012.
- [25] G. K. L. A. L. N. Jaroslav Karpovič, „The Comprehensive Mapping of Semantics of Business Vocabulary and Business Rules (SBVR) to OWL 2 Ontologies,“ *INFORMATION TECHNOLOGY AND CONTROL*, t. 43, nr. 3, pp. 289-302, 2014.
- [26] „Protege,“ Protege, [Tinkle]. Available: <https://protege.stanford.edu/>. [Kreiptasi 01 03 2020].
- [27] E. F. A. M. Dmitry Solomakhin, „Logic-based reasoning support for SBVR,“ *Fundamenta Informaticae*, pp. 543-560, 2013.
- [28] A. M. D. S. Enrico Franconi, „ORM2: Syntax and semantics,“ *Internal report, KRDB Research Centre for Knowledge and Data*, 2011.
- [29] P. M. A. L. C. E. A. S. A. J. A. Denilson dos Santos Guimarães, „A Method for Verifying the Consistency of Business Rules Using Alloy,“ *Proceedings of the Twenty-Sixth International Conference on Software Engineering & Knowledge Engineering*, pp. 381-386, 2014.
- [30] W. Koehrsen, „Beyond Accuracy: Precision and Recall,“ towardsdatascience, 03 03 2018. [Tinkle]. Available: <https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c>. [Kreiptasi 24 05 2020].

## Priedai

### 1. Renginių organizavimo *SBVR* veiklos žodynas ir taisyklės.

Renginių organizavimo *SBVR* veiklos žodynas prieštarigumams aptikti

organizatorius

General\_concept: darbuotojas

darbuotojas

General\_concept: organizatorius

Synonym: darbininkas

Synonym: savanoris

darbininkas

See: darbuotojas

savanoris

See: darbuotojas

bilietas

renginys

renginio\_vieta

Synonym: vieta

vieta

Concept\_type: role

Concept\_type: general\_concept

pavadinimas

Concept\_type: role

General\_concept: text

pavadinimas

Concept\_type: role

General\_concept: text

renginio\_data

Concept\_type: role

General\_concept: datetime

pirkimo\_data

Concept\_type: role

General\_concept: datetime

savininkas

Concept\_type: role

General\_concept: text

data

renginys turi data

pardavimo\_data

Concept\_type: role

General\_concept: datetime

General\_concept: data

kaina

Concept\_type: role

General\_concept: number

talpa

Concept\_type: role

General\_concept: integer

darbuotojas *su\_patirtimi*  
darbuotojas *dirba* organizatoriui

organizatorius *turi* pavadinimą  
organizatorius *susideda iš* darbuotojų  
organizatorius *organizuoja* renginius

renginys *turi* pavadinimą  
renginys *turi* renginio\_data  
renginys *organizuojamas* organizatorių  
renginiui *parduodami* bilietai  
renginys *vyksta* renginio\_vietoje

bilietas *turi* kainą  
bilietas *turi* pirkimo\_data  
bilietas *turi* savininką  
bilietas *priklauso* renginiui  
bilietas *nepriklauso* renginiui

renginio\_vieta *turi* talpą  
renginio\_vietoje *vyksta* renginys

mokėjimo\_tipas  
Concept\_type: categorization\_type  
Necessity: *is\_for* general\_concept renginys

Renginiai\_pagal\_mokėjimo\_tipą  
Necessity: segmentation *for* general\_concept renginys *that subdivides* renginys *by* mokėjimo\_tipą

nemokamas\_renginys  
General\_concept: renginys  
Necessity: *is\_included\_in* Renginiai\_pagal\_mokėjimo\_tipą

mokamas\_renginys  
General\_concept: renginys  
Necessity: *is\_included\_in* Renginiai\_pagal\_mokėjimo\_tipą

vietos\_tipas  
Concept\_type: categorization\_type  
Necessity: *is\_for* general\_concept renginio\_vieta

sporto\_arena  
Concept\_type: vietos\_tipas

teatras  
Concept\_type: vietos\_tipas

studija  
Concept\_type: vietos\_tipas

Eventus  
General\_concept: organizatorius

Tomas  
General\_concept: darbuotojas

Bilietas\_Nr101  
General\_concept: bilietas

Kitokie\_pasikalbėjimai  
General\_concept: renginys

General\_concept: mokamas\_renginys

Miesto\_sodas  
General\_concept: renginio\_vieta

Operos\_ir\_baleto\_teatras  
General\_concept: renginio\_vieta  
General\_concept: teatras

bilietas *kainuoja* kaina

Bilietas\_Nr102  
Bilietas\_Nr103  
General\_concept: kaina

100\_eur  
General\_concept: kaina

Bilietas\_Nr102 *kainuoja* 100\_eur  
Bilietas\_Nr103 *kainuoja* 100\_eur

Karmen\_spektaklis  
General\_concept: mokamas\_renginys  
General\_concept: nemokamas\_renginys  
General\_concept: renginys

Žalgirio\_arena  
General\_concept: sporto\_arena  
General\_concept: studija

Bilietas\_Nr104  
General\_concept: bilietas

Kitokie\_pasikalbėjimai *parduodamas* Bilietas\_Nr104

vip\_bilietas  
General\_concept: bilietas

Lietuvos\_talentai  
General\_concept: renginys

Bilietas\_Nr105  
General\_concept: vip\_bilietas

Lietuvos\_talentai *parduodamas* Bilietas\_Nr105

plotis  
General\_concept: spalva

aukstis  
General\_concept: plotis

spalva  
Concept\_type: categorization\_type  
General\_concept: aukstis

bilietas *turi* spalva

Renginys\_ir\_vieta\_viename  
General\_concept: mokamas\_renginys  
  
General\_concept: sporto\_arena

Renginių organizavimo *SBVR* veiklos taisyklės

Privaloma, kad darbuotojas *dirba* lygiai 1 organizatoriui  
Privaloma, kad organizatorius *susideda iš* mažiausiai 1 darbuotojo  
Privaloma, kad renginys *organizuojamas* lygiai 1 organizatoriaus  
Privaloma, kad organizatorius *organizuoja* mažiausiai 0 renginių  
Privaloma, kad renginiui *parduodami* mažiausiai 5 ir daugiausiai 10000 bilietų  
Privaloma, kad bilietas *priklauso* daugiausiai 1 renginiui  
Privaloma, kad renginys *vyksta* daugiausiai 1 renginio vietoje  
Nėra privaloma, kad renginys *vyksta* daugiausiai 1 renginio vietoje  
Privaloma, kad darbuotojas *dirba* mažiausiai 2 organizatoriams  
Privaloma, kad bilietas *nepriklauso* daugiausiai 1 renginiui  
Privaloma, kad organizatorius *turi* pavadinimą lygų 1  
Privaloma, kad renginys *vyksta* mažiausiai 5 ir daugiausiai 0 renginio vietoje  
Nėra įmanoma, kad bilietas *priklauso* daugiausiai 1 renginiui

## 2. SBVR veiklos žodyno ir taisyklių prieštaravimo tikrinimo eksperimento rezultatai

1. Taisyklė su kardinalumu bendriniam konceptui: organizatorius sukuria kardinalumų prieštaravimą.
2. WARNING. Viena iš taisyklių su kardinalumu yra perteklinė bendriniam konceptui: renginio\_vieta.
3. General concept: renginio\_vieta su veiksmožodine forma: vyksta sukelia modalumo prieštaravimą.
4. General concept: renginio\_vieta su veiksmožodine forma: vyksta sukelia modalumo prieštaravimą.
5. Role: pavadinimas has non numeric primary type.
6. Role: pavadinimas has non numeric primary type.
7. Role renginys - Kitokie\_pasikalbėjimai has wrong number of instances of bilietas. Minimum: 5, maximum: 10000
8. Error at GeneralConcept:  
lt.ktu.isd.semantika.vocabulary.impl.DesignationOfGeneralConceptImpl@3c63d57e (name: vieta).  
Multiple captions found: Concept\_type:
9. Entry concept vieta did not have attribute See with value: renginio\_vieta
10. Duplication of entry concept: pavadinimas
11. Individualus konceptas: Bilietas\_Nr102 neturi bendrinio koncepto
12. Fakto: Bilietas\_Nr103 kainuoja 100\_eur individualūs konceptai neturi bendrinių konceptų, kurie atitiktų statinį apribojimą.
13. Fakto: Lietuvos\_talentai parduodami Bilietas\_Nr105 individualūs konceptai neturi bendrinių konceptų, kurie atitiktų statinį apribojimą.
14. Error at Role:  
lt.ktu.isd.semantika.vocabulary.impl.DesignationOfGeneralConceptImpl@49ecd74f (name: pardavimo\_data). Multiple general\_concept captions found: General\_concept:
15. Error at concept: data. This concept should be of type: role.
16. Error at concept: spalva. This concept should be of type: role.
17. Error at Individual concept: Karmen\_spektaklis. Multiple categories of one categorization scheme was assigned.
18. Error at Individual concept: Žalgirio\_arena. Multiple categories of one categorization scheme was assigned.
19. Cycle was detected for: organizatorius
20. Cycle was detected for: darbuotojas
21. Cycle was detected for: Eventus

22. Cycle was detected for: Tomas  
23. Cycle was detected for: plotis  
24. Cycle was detected for: aukstis  
25. Cycle was detected for: spalva  
Total elapsed time: 100 milliseconds.