

**VILNIAUS UNIVERSITETAS  
KAUNO HUMANITARINIS FAKULTETAS**

**INFORMATIKOS KATEDRA**

Verslo informatikos studijų programa  
Kodas 62109P101

**GIEDRIMĖ IVANČIKAITĖ**

**MAGISTRO BAIGIAMASIS DARBAS**

**ŠIUOLAIKINIŲ VEIKLOS TAISYKLIŲ STANDARTŲ TAIKYMAS KURIANT  
INFORMACINES SISTEMAS**

Kaunas 2009

VILNIAUS UNIVERSITETAS  
KAUNO HUMANITARINIS FAKULTETAS  
INFORMATIKOS KATEDRA

GIEDRIMĖ IVANČIKAITĖ

MAGISTRO BAIGIAMASIS DARBAS

**ŠIUOLAIKINIŲ VEIKLOS TAISYKLIŲ STANDARTŲ TAIKYMAS KURIANT  
INFORMACINES SISTEMAS**

Leidžiama ginti \_\_\_\_\_

Magistrantė \_\_\_\_\_  
(parašas)

Darbo vadovas \_\_\_\_\_  
(parašas)

doc.dr. Gintautas Garšva

Darbo konsultantas \_\_\_\_\_  
(parašas)

lekt., dr. Kęstutis Kapočius

Darbo įteikimo data \_\_\_\_\_

Registracijos Nr. \_\_\_\_\_

Kaunas 2009

## TURINYS

TURINYS.....	3
SANTRUMPŲ SĄRAŠAS .....	5
PAVEIKSLŲ SĄRAŠAS.....	6
LENTELIŲ SĄRAŠAS.....	8
SUMMARY .....	9
ĮVADINĖ DALIS .....	10
1. VEIKLOS TAISYKLIŲ KONCEPCIJOS IR AKTUALIŲ ŠIUOLAIKINIŲ STANDARTŲ TYRIMAS .....	12
1.1. Literatūros šaltinių analizė .....	12
1.2. Pagrindiniai veiklos taisyklių modeliavimo standartai .....	15
1.3. OMG standartai MDA architektūros kontekste .....	17
1.3.1. MDA architektūra .....	17
1.3.2. MDA modeliai.....	18
1.4. OMG standartai MDA architektūra.....	20
1.4.1. BMM .....	21
1.4.2. SBVR .....	23
1.4.3. BPDm.....	24
1.4.4. PRR .....	24
1.5. UML, OCL ir veiklos taisyklės .....	25
1.6. Standartų palyginimas .....	27
1.7. Analizės išvados ir tolimesnio tyrimo gairės .....	29
2. SBVR IR PRR VEIKLOS TAISYKLIŲ STANDARTŲ SUSIEJIMO PRIELAUDŲ TYRIMAS.....	31
2.1. SBVR ir PRR standartų naudojimo metodikos koncepcija.....	31
2.2. Veiklos taisyklių specifikuojimas SBVR standartu.....	33
2.3. SBVR formuluotės .....	36
2.4. Konceptų ir juos siejančių faktų grafinė atvaizdavimo notacija remiantis SBVR.....	39
2.4.1. SBVR atvaizdavimo notacija .....	39
2.4.2. SBVR atvaizdavimo notacija remiantis UML .....	42
2.5. Konceptų žodyno sudarymas pagal SBVR standartą.....	48
2.6. Veiklos taisyklių specifikuojimas pagal SBVR standartą .....	54
2.7. Produkcinį veiklos taisyklių specifikuojimas PRR standartu.....	55

2.8.	SBVR ir PRR standartų sugretinimas .....	57
2.9.	SBVR išraiškų transformavimo į PRR tyrimas.....	60
2.10.	SBVR ir PRR standartų metodikos praktinis panaudojimo apibendrinimas.....	62
3.	SBVR IR PRR STANDARTŲ PANAUDOJIMO KURIANTIS METODIKA.....	63
3.1.	SBVR žodyno klasių modelis .....	63
3.2.	SBVR taisyklių šablonų sudarymas ir jų vertimo į lietuvių kalbą įvertinimas .....	65
3.3.	Struktūrinių taisyklių šablonai ir ryšiai .....	66
3.4.	Operacinės taisyklės šablono transformavimas į PRR OCL.....	72
3.5.	SBVR ir PRR panaudojimo metodika: apibendrinimas .....	75
4.	METODIKOS PRAKTINIS ĮVERTINIMAS REALIOS DALYKINĖS SRITIES PAGRINDU .	77
4.1.	Dalykinės srities klasių diagrama.....	77
4.2.	Dalykinės srities konceptų žodyno ir veiklos taisyklių sudarymas.....	79
4.3.	Operatyvinių SBVR taisyklių transformavimas į PRR .....	80
4.4.	Dalykinės srities taisyklių užrašymas naudojant SBeaVeR.....	82
4.5.	Rekomendacijos metodikos realizacijai .....	85
4.6.	Metodikos išbandymo dalykinėje srityje įvertinimas.....	87
	IŠVADOS.....	90
	LITERATŪROS SĄRAŠAS.....	92
	1 PRIEDAS: SBVR standarto matematiniai ir loginiai operatoriai .....	97
	2 PRIEDAS: SBVR standarto meta-modelio fragmentas .....	98
	3 PRIEDAS: PRR standarto meta-modelis .....	99
	4 PRIEDAS: Veiklos taisyklių saugyklos lentelių ir atributų reikšmės.....	100
	5 PRIEDAS: Dalykinės srities konceptų žodynas.....	105
	6 PRIEDAS: Konferencijoje pristatytas straipsnis.....	107

## SANTRUMPŲ SĄRAŠAS

VT – veiklos taisyklės, tam tikrus veiklos aspektus apibrėžiantis arba apribojantis teiginys, kuris reikalingas norint įvertinti veiklos struktūrą arba valdyti/įtakoti veiklą.

IS – Informacinė sistema

DB – duomenų bazė

OMG – (*Object Management Group*) tarptautinė standartų organizacija, kurios tikslas sukurti modeliavimo standartus programoms, sistemoms ir verslo procesams

SBVR – (*Semantics of Business Vocabulary and Business Rules*) OMG sukurtas VT apibrėžimo standartas, orientuotas į semantinę natūraliosios kalbos sakinių, kuriais išreiškiamos VT, struktūrą

PRR – (*Production Rule Representation*) OMG sukurtas produkcinį VT apibrėžimo standartas, kuriame VT aprašomos loginiais teiginiais (jeigu <sąlyga> tada <veiksmai> kitu atveju <veiksmai>)

UML – (*Unified Modeling Language*) OMG sukurta informacinių sistemų modeliavimo kalba

BMM – (*Business Motivation Model*) standartas, orientuotas veiklos tikslus ir būdus (strategijas) tiems tikslams pasiekti bei į šiuos du elementus įtakojančius faktorius

BPDM – (*Business Process Definition Metamodel*) OMG sukurtas veiklos procesų apibrėžimo meta-modelis

OCL – (*Object Constraint Language*) modeliavimo kalba UML meta-modeliams kitiems OMG standartais apibrėžiamiems meta-modeliams apribojimams nusakyti.

MDA – (*Model Driven Architecture*) OMG sukurta IS kūrimo architektūra, kurios esmė – verslo ir pritaikymo logikos atskyrimas nuo konkrečios platformos.

CIM – (*Computation Independent Model*) MDA modelis, kuris nusako kuriamos sistemos koncepciją, neparodydamas skaičiavimų ir sistemos struktūros detalių, dar vadinamas domeno (angl. *domain*) modeliu.

PIM – (*Platform Independent Model*) MDA modelis, pateikiantis formalų sistemos vaizdą, nepriklausomai nuo sistemos realizavimo platformos.

PSM – (*Platform Specific Model*) MDA modelis, kuris sujungia PIM specifikacijas, su detalėmis nusakančiomis kaip sistemai naudotis konkrečia platforma.

## PAVEIKSLŲ SĄRAŠAS

Pav. 1. Veiklos taisyklių tipai.....	12
Pav. 2. Veiklos taisyklių vieta ir funkcijos trijų lygmenų IS architektūroje .....	13
Pav. 3. Reikalavimų specifikacijos detalizuoto modelio sudarymo procesas .....	14
Pav. 4 Atitinkami modelių ryšiai pagal MDA bei ryšiai tarp keleto sistemų.....	19
Pav. 5 Bendra BMM struktūra .....	21
Pav. 6 SBVR vieta MDA architektūroje .....	23
Pav. 7. Veiklos taisyklių modeliavimo standartų panaudojimas kuriant IS.....	28
Pav. 8. OMG standartų BMM, SBVR ir PRR vieta MDA architektūros kontekste .....	28
Pav. 9. Siūlomos SBVR ir PRR standartų taikymo metodikos schema.....	32
Pav. 10. Penki pagrindiniai SBVR aspektai .....	35
Pav. 11. Konceptų grafinis vaizdavimas .....	39
Pav. 12. Konceptų kategorizavimas.. .....	39
Pav. 13. Konceptų kategorizavimas. ....	40
Pav. 14. Grafiškai išreikštų faktų skaitymas .....	41
Pav. 15. N faktų skaičiaus išreiškimas. ....	41
Pav. 16. Faktas turintis tik vieną konceptą.....	42
Pav. 17. Konceptui b suteikiama rolė dalis .....	42
Pav. 18. Du bendriniai konceptai, iš kurių vienas papildytas žodeliu „taip pat“ (also) ir papildomu terminu .....	43
Pav. 19. Individualių konceptų vaizdavimas.....	43
Pav. 20. Trys individualūs konceptai .....	43
Pav. 21. Trys būdai dvejetainiams faktams atvaizduoti .....	44
Pav. 22. Faktų tipai su žodeliu <i>turi</i> .....	44
Pav. 23. Fakto sudaryto iš trijų elementų atvaizdavimas .....	45
Pav. 24. Faktų tipų sudarytų iš vieno elemento atvaizdavimas.....	45
Pav. 25. Rolės vaizdavimas ryšio pabaigoje. ....	46
Pav. 26. Rolė vaizduojama kaip klasė, naudojant UML stereotipus.....	46
Pav. 27. Rolės terminas yra faktų tipo formoje.....	46
Pav. 28. Generalizacijos pavyzdžiai.....	47
Pav. 29. Konceptų skirstymas į kategorijas.....	47

Pav. 30. Dalinių faktų grafinis ir žodinis žymėjimas. ....	48
Pav. 31. Raktažodžių skirstymas pagal loginę prasmę.....	50
Pav. 32. Raktažodžių skirstymas pagal taisyklių rūšį .....	51
Pav. 33. Žodyno sudarymas pagal SBVR .....	51
Pav. 34. Konceptų aprašymo schema.....	52
Pav. 35. Taisyklių rinkinio aprašymo schema.....	54
Pav. 36. Taisyklių sudarymo schema .....	55
Pav. 37. Žodyno klasių diagrama .....	64
Pav. 38. Ryšio tarp faktų grafinė notacija .....	67
Pav. 39. Ryšio realizavimas duomenų bazėje .....	68
Pav. 40. Ryšiai tarp koncepto potipių ir kitų konceptų .....	68
Pav. 41. Ryšys vienas su daug duomenų modelyje.....	70
Pav. 42. Veiklos taisyklių saugyklos schema.....	71
Pav. 43. PRR OCL meta-modelio fragmentas, nurodantis klasių sąryšius .....	73
Pav. 44. Dalykinės srities klasių diagrama.....	78
Pav. 45. SBeaVeR žodyno sudarymo langas .....	83
Pav. 46. SBeaVeR sugeneruota XSD schema.....	84
Pav. 47. SBeaVeR užrašytos veiklos taisyklės .....	85
Pav. 48. Metodikos realizavimo koncepcinė schema.....	86

## LENTELIŲ SĄRAŠAS

1 lentelė: Raktažodžiai kiekiui išreikšti .....	49
2 lentelė: Raktažodžiai loginėms operacijoms išreikšti .....	49
3 lentelė: Raktažodžiai modalinėms operacijoms išreikšti .....	50
4 lentelė: SBVR ir PRR standartų sugretinimas.....	59
5 lentelė: Struktūrinių taisyklių šablonai.....	65
6 lentelė: Operatyvinių taisyklių šablonas .....	66
7 lentelė: Metodikos palyginimas .....	88



IVANČIKAITĖ, Giedrimė (2009). *IS Development Applying Current Production Rules Standards*. MBA Graduation Paper. Kaunas: Vilnius University, Kaunas Faculty of Humanities, Department of Informatics. 91 p.

## SUMMARY

*The aim of this paper is the analysis of the new OMG (Object Management Group) business rules standards, the creation of methodology for their application and its practical evaluation.*

*The main tasks of the work are:*

- 1. Analysis of business rules conception and application.*
- 2. Analysis of the standards designed for requirement specification phase (emphasizing SBVR and PRR).*
- 3. Definition of the main characteristics of the selected standards.*
- 4. Creation of the standards application methodology.*
- 5. Investigation of the implementation possibilities of the created methodology.*
- 6. Practical evaluation of methodology.*

*From the analyzed OMG standards two have been distinguished - SBVR (Semantics of Business Vocabulary and Business Rules) and PRR (Production Rule Representation), which are relevant to the production rules. The study defined the selected standards fundamental properties and investigated the binding of PRR and SBVR expressions and the options of the generation of templates.*

*The main results of the work are:*

- Analytical evaluation of SBVR and PRR standards and their interaction;*
- The methodology of SBVR and PRR application during the specification of the concept dictionary and production rules;*
- Definition of the repository structures for the developed methodology;*
- The analysis of the open-source tool SBeaVeR and the recommendations for its modification to fit the created methodology;*

*The application of business rules approach for the creation of IS is very promising but the usage of the latest business rule standards has various difficulties, because their use is not sufficiently elaborated. This work describes a methodology for the application of two new OMG standards – SBVR and PRR. The methodology is tested practically and evaluated.*

*The volume of the paper is 91 pages. In the main chapters there are 7 tables and 48 pictures.*

## ĮVADINĖ DALIS

Tradiciskai, kuriant informacines sistemas (IS), veiklos taisyklės nėra analizuojamos ir modeliuojamos atskirai nuo duomenų ir procesų specifiką atspindinčių IS dedamųjų. Realizuojant IS jos įtraukiamos į programinį kodą kaip tam tikri reikalavimai ir apribojimai. Toks būdas nėra patogus, nes pakitus kuriai nors VT, reikia naujai gilintis į programinį kodą ir keisti visą sistemą. Įvertinant tai, kad šiuolaikinės veiklos reikalavimai pakankamai dažnai keičiasi, toks būdas sukelia daug sunkumų. Todėl veiklos taisyklės yra svarbu valdyti kaip atskirą fizinį ir logiškai savarankišką sistemos lygmenį [11].

Nors veiklos taisyklių koncepcijos taikymas kuriant IS yra labai perspektyvus, tačiau iki šiol jis buvo menkai išplėtotas ir beveik nestandartizuotas. Atsirandant atskiras problemas liečiantiems standartams, prielaidos VT koncepcijai plisti stiprėja, tačiau būtina atlikti išsamius tyrimus, susiejant naujus standartus su jau esamais ir sukuriant jų taikymo metodikas.

Viena iš didžiausių veiklos taisyklių ir modeliavimo standartų kūrėja yra OMG (*Object Management Group*). OMG – tai tarptautinė standartų organizacija, kurios tikslas – sukurti modeliavimo standartus programoms, sistemoms ir verslo procesams. Šio magistrinio **darbo objektas** - OMG veiklos taisyklių standartai ir jų taikymas kuriant informacines sistemas.

**Pagrindinis darbo tikslas** – įvertinti standartizuotų VT koncepcijos sprendimų reikšmę, ištiriant naujausius OMG (*Object Management Group*) veiklos taisyklių standartus ir sukuriant aktualių standartų taikymo kuriant informacines sistemas metodiką bei ją eksperimentiškai įvertinti.

### **Pagrindiniai darbe keliami uždaviniai yra:**

- 1) Išnagrinėti veiklos taisyklių koncepciją ir jos taikymo principus.
- 2) Ištirti standartus, skirtus reikalavimų specifikavimo stadijai (akcentuojant SBVR ir PRR).
- 3) Apibrėžti pasirinktų standartų charakteringąsias savybes ir jų ryšius.
- 4) Sukurti pasirinktų standartų taikymo metodiką.
- 5) Ištirti pasiūlytos metodikos realizavimo galimybes.
- 6) Išbandyti metodiką pasirinktos dalykinės srities pagrindu.

Pirmojoje darbo dalyje pateikiami esami problemos sprendimai, taip pat detali OMG standartų apžvalga, MDA architektūros koncepcija, tiriami standartai jos kontekste bei standartų palyginimas. Šios dalies pabaigoje išskirti esminiai standartai, aktualūs tiriamos problemos atžvilgiu ir suplanuotas tyrimas.

Antrojoje dalyje nagrinėjami veiklos taisyklių specifikavimo SBVR bei PRR standartais ypatumai. Tiriama, kaip standartai sugretinami, iširtos galimybės iš SBVR standarto taisyklių gauti PRR išraiškas, pateikiami nagrinėjamų standartų susiejimo metodikos etapai.

Trečiojoje dalyje aprašoma SBVR ir PRR standartų panaudojimo kuriant IS metodika. Aptariamos SBVR pritaikymo lietuvių kalbai galimybės, pateikiama SBVR konceptų žodyno klasių diagrama, veiklos taisyklių saugyklos schema bei SBVR išraiškų transformavimo į PRR gairės.

Ketvirtojoje dalyje aprašomas praktinis metodikos išbandymas, pritaikant ją pasirinktai dalykinei sričiai. Šios metodikos išbandymas open-source įrankiu SBeaVeR bei gauti rezultatai.

Atliekant probleminės srities tyrimą, buvo analizuojami tiek Lietuvos, tiek ir užsienio autorių darbai. Didžioji dalis literatūros šaltinių yra OMG standartų specifikacijos, kurios pateikiamos OMG tinklalapyje, bei Business Rules Community organizacijos narių straipsniai, pateikiami Business Rules Community tinklalapyje.

Problemos sprendimui buvo sudaryta metodika; tam, kad būtų patikrintas jos veiksmingumas buvo atlikta metodikos analizė, sprendimų modeliavimas, sudaryta metodika buvo palyginta su jau esančiomis.

Pagrindiniai sunkumai atliekant tyrimą iškilo praktinio metodikos išbandymo metu – sudarinėjant SBVR išraiškų transformatorių, kuris turėjo generuoti PRR OCL išraiškas. Praktiškai realizuoti šio transformatoriaus nepavyko. Buvo tik apibrėžtos pagrindinės jo gairės ir pagal jas transformuotos taisyklės. Taip pat problemų sukėlė atviro kodo įrankis SBeaVeR.

Pagrindiniai darbo metu gauti moksliniai/taikomieji rezultatai yra:

- Analitinis SBVR ir PRR standartų įvertinimas jų tarpusavio santykio atžvilgiu;
- SBVR ir PRR panaudojimo IS kūrimo metu specifikuojant veiklos žodyną ir produkcines taisykles metodika, standartų sulietuvavimo galimybių analizė;
- Sukurtąją metodiką palaikančių saugyklų struktūros apibrėžimas.
- Atviro kodo įrankio SBeaVeR analizė bei jo modifikavimo rekomendacijos, siekiant jį pritaikyti darbe siūlomai metodikai.

### **Rezultatų aprobavimas**

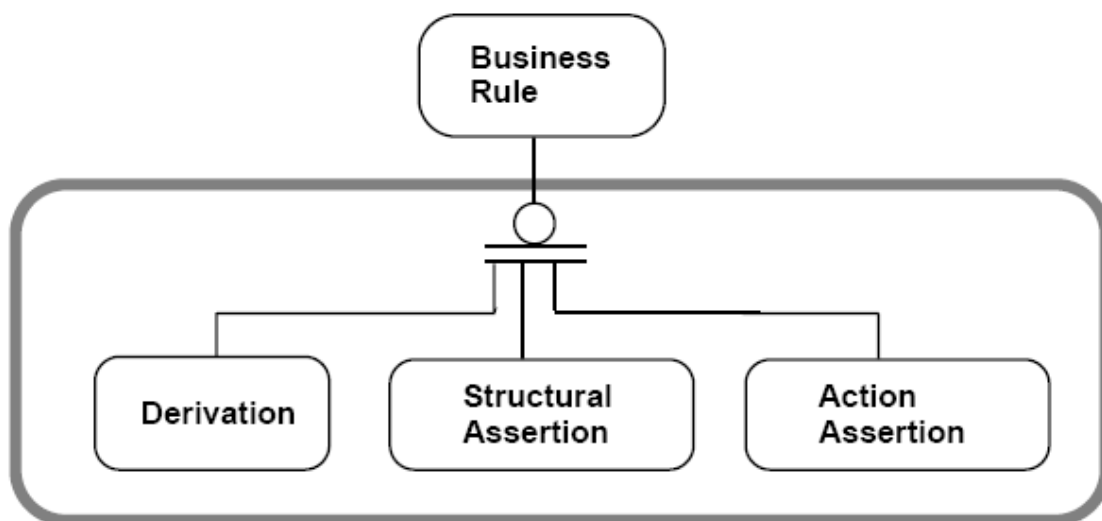
Darbo tema parašytas straipsnis „Šiuolaikinių veiklos taisyklių standartų taikymas kuriant informacines sistemas“ ir pristatytas 2009 metais vykusioje 14-toje tarpuniversitetinėje magistrantų ir doktorantų konferencijoje „Informacinės Technologijos“. Straipsnis išspausdintas konferencijos leidinyje.

Šis darbas sudarytas iš 91 puslapio, jame yra 48 paveikslai, 7 lentelės ir 6 priedai.

# 1. VEIKLOS TAISYKLIŲ KONCEPCIJOS IR AKTUALIŲ ŠIUOLAIKINIŲ STANDARTŲ TYRIMAS

Remiantis GUIDE [1], veiklos taisyklės gali būti trijų tipų:

- *Struktūrinis teiginys* (angl. *structural assertion*) – apibrėžtas teiginys arba koncepcija faktų, kurie išreiškia kurį nors verslo sistemos struktūros elementą.
- *Veiksmo teiginys* (angl. *action assertion*) – apribojimo arba sąlygos teiginys, kuris riboja arba valdo verslo sistemos veiksmus.
- *Išvestis* (angl. *derivation*) – teiginys išvestas iš žinių, turimų apie verslą.



Šaltinis: Business Rules Group. (2000). *Defining Business Rules ~ What Are They Really?* (3<sup>rd</sup> edition).

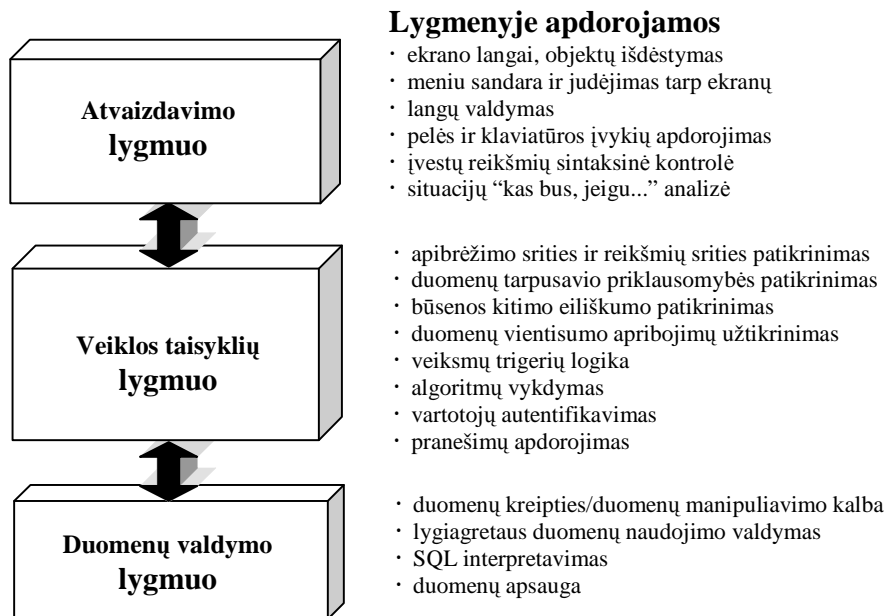
**Pav. 1. Veiklos taisyklių tipai.**

Informacinės sistemos kūrimo pradžioje veiklos taisyklės surenkamos ir apibrėžiamos natūraliąja kalba. Veiklos taisyklių šaltinis turėtų būti veiklos atstovai [3]. IS kūrimo metu projektuotojai palaiapsniui suformuluoja formalias išraiškas, kurios koordinuoja sistemos veiklą, t.y. nustato ribą tarp priimtinos ir nepriimtinos veiklos [3].

## 1.1. Literatūros šaltinių analizė

Galima teigti, jog tradicinės IS susideda iš dviejų pagrindinių lygmenų – duomenų valdymo ir vartotojo (programų) lygmens (pav. 2). Jeigu kuriant sistemą naudojama veiklos taisyklių koncepcija, sistemą galima vadinti trisluoksne [5]. Kitaip tariant, veiklos taisyklės sudaro naują sistemos lygmenį,

kuris yra nepriklausomas nuo procesų ir duomenų struktūros, - tai sukuria sąlygas jas keisti atskirai ir centralizuotai. Be to, sistema įgauna lankstumo. Šie faktoriai lemia VT koncepcija grindžiamų sprendimų tyrimo aktualumą.



Šaltinis: Butleris, R., Kapočius, K. (2001). *Struktūrizuotų veiklos taisyklių saugyklos architektūra*. Informacijos mokslai. Vilnius. Vilniaus universiteto leidykla, 17 tomas, 46-57.

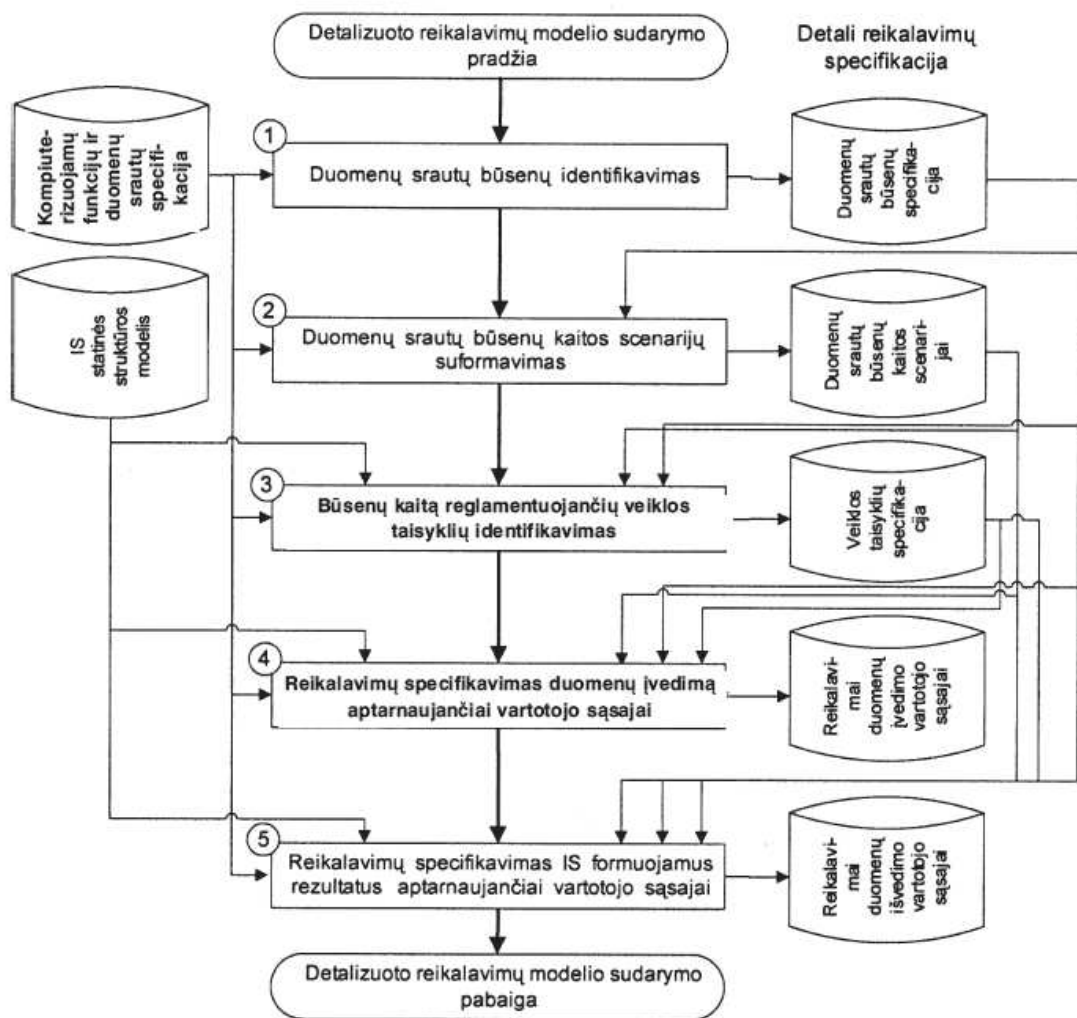
## Pav. 2. Veiklos taisyklių vieta ir funkcijos trijų lygmenų IS architektūroje

SBVR standarto taikymą MDA kontekste nagrinėjo S. Hendryx, kuris pabrėžė, kad SBVR galima traktuoti kaip žodyną, kuris suteikia galimybę interpretuoti veiklos taisykles, išreikštas natūraliąja kalba [15]. Taip pat šis mokslininkas atkreipia dėmesį į tai, kad iš natūraliosios kalbos šablonais užrašytų veiklos taisyklių galima sudaryti meta-modelius naudojantis PRR [14]. S. Nijssen savo straipsnyje pabrėžia, kad reikalinga sukurti SBVR standartui metodologiją t. y. formalų taisyklių apibrėžimo ir atnaujinimo procesą [21]. A. Kolber savo straipsnyje apžvelgia BMM, akcentuodamas, kad šis standartas patogus dėl to kad juo galima aiškiai apibrėžti įmonės tikslus ir strategijas [17]. Šiuos tris standartus nagrinėjo nemažai autorių, tačiau šių standartų taikymo metodologija – nepasiūlyta.

N. Haggerty savo straipsnyje [10] pateikia pagrindinius UML trūkumus modeliuojant VT. Pabrėždamas, kad UML neapibrėžia taisyklių, kaip atskiro modeliavimo elemento. Norint aprašyti veiklos taisykles UML, naudojama OCL kalba. OCL taikymą aprašo N. Fishman, atkreipdamas dėmesį į tai, kad OCL yra išraiškų kalba, ji ribota ir ja galima išreikšti tik dalį taisyklių [9]. Atsižvelgiant į tai galima teigti, kad tam tikra OCL taikymo metodologija egzistuoja, tačiau pats šis standartas nėra labai patogus VT išreikšti.

J. Montilva ir J. Barnios savo straipsnyje pristato BMM panaudojimą kuriant EIS (Enterprise Information Systems) [20]. Remiantis straipsniu galima teigti, kad BMM – tai labai patogus būdas aprašyti pačią IS veiklą. Kuriant IS labai aktualu pati VT realizacija, BMM modelis šiam tikslui nelabai tinka, nes yra orientuotas tik į pačią veiklą, vadinasi, norint pilnai realizuoti IS, gali prireikti sujungti BMM su kitu standartu.

R. Butleris savo straipsnyje apžvelgia reikalavimų specifikuojimo problemas ir pabrėžia, kad reikalavimus sistemai reikia rinkti viso IS kūrimo metu [4]. Straipsnyje minima, kad kuriami reikalavimų specifikuojimo modelių prototipai, pateikiamas reikalavimų specifikuojimo abstrahuoto modelio sudarymo procesas (pav. 3). Į modelį įtraukta ir VT specifikuojimas, tačiau VT kaip atskiras sistemos sluoksnis neišskirtos.



Šaltinis: R. Butleris. (2006) *Funkcinių reikalavimų specifikuojimas abstrahuoto ir detalizuoto reikalavimų modelių pagrindu*. Informacijos Mokslai, 168 - 177 psl. ISSN 1392-0561.

**Pav. 3. Reikalavimų specifikuojimo detalizuoto modelio sudarymo procesas**

Kitame straipsnyje R. Butleris ir K. Kapočius apibrėžia veiklos taisyklių saugyklos koncepciją [5]. Tačiau straipsnyje pabrėžta, kad praktiškai ši saugykla nėra realizuota. Vėlesniame straipsnyje R. Butleris ir L. Motiejūnas pristato veiklos taisyklių saugyklą su veiklos taisyklių iškvietimo mechanizmu [6]. Ši veiklos taisyklių saugykla realizuota naudojant OMG standartą OCL bei XML pagrįstą veiklos taisyklių kalbą RuleML. Veiklos taisyklių saugykla apibrėžiama kaip duomenų bazė, kurioje saugomi visi reikalingi duomenys apie veiklos taisykles.

I. Pašilskytė ir L. Nemuraitė savo straipsnyje išanalizuoja veiklos procesų modeliavimo kalbas UML 2.0, BPMN ir BPEL4WS bei siūlo UML papildyti BPMN elementais bei praplėsti grafine proceso apibrėžimo stadiją [33]. Taip pat šiame straipsnyje pabrėžiamas pagrindinis BPMN trūkumas – nėra pakankamai priemonių dalykinei sričiai atvaizduoti.

G. Vedrickas ir L. Nemuraitė savo straipsnyje aptaria RuleML standartą bei kaip juo apibrėžiamos veiklos taisyklės [39]. RuleML standartą taip pat apžvelgė R. Butleris ir L. Motiejūnas [6], paminėdami, kad šioje kalboje palaikomi tik keletas pagrindinių veiklos taisyklių tipų. Taip pat šiame straipsnyje minimas OCL kalbos sintaksės ribotumas.

O. Vasilecas ir D. Būgaitė savo straipsnyje siūlo kuriant veiklos taisyklių sistemą panaudoti dalykinės srities ontologiją [38]. Veiklos taisyklių ir dalykinės srities ontologijos sujungimas realizuotas panaudojant UML modeliavimo kalbą.

Išnagrinėjus literatūrą galima teigti, kad veiklos taisyklės yra labai aktualus klausimas kuriant IS. Šią problemą nagrinėjo daug užsienio bei Lietuvos mokslininkų. Tokios IS projektavimo stadijai skirtos kalbos kaip UML ar OCL yra plačiai išnagrinėtos. Taip pat nagrinėti standartai nepriklausantys OMG (BPEL4WS, RuleML, CASE). Tačiau naujiesiems OMG organizacijos standartams (ypač PRR, kurio specifikacijos beta versija priimtas tik 2007 m. gruodžio mėn. [22]), daugiausiai skirtiems reikalavimų specifikavimo ir veiklos taisyklių sudarymo stadijai, detali analizė nėra atlikta. Taip pat neaptikta tyrimų, kaip SBVR, BMM ir PRR galima suderinti tarpusavyje. Kiekvienas standartas turi savo privalumų ir trūkumų, tačiau šių standartų panaudojimas kuriant IS dėl jų sąlyginio naujumo yra labai ribotas. Iš tokią situaciją lemiančių priežasčių galima išskirti šių standartų taikymo metodikų nebuvimą.

## **1.2. Pagrindiniai veiklos taisyklių modeliavimo standartai**

Išnagrinėjus literatūrą galima išskirti pagrindinius jau patvirtintus arba kuriamus VT specifikavimo klausimus liečiančius standartus:

- SBVR - *Semantics of Business Vocabulary and Business Rules*. Tai VT apibrėžimo standartas, orientuotas į semantinę natūraliosios kalbos sakinių, kuriais išreiškiamos VT,

struktūrą [21]. SVBR suteikia priemones aprašyti semantines formuluotes, t.y. veiklos taisyklės nėra paverčiamos į formalią kalbą, o tiesiog taikoma konkreti sakinių struktūra, leidžianti formalizuoti VT [12, 15, 30]. SBVR veiklos reikšmių modelio struktūra nusakoma asocijuojant abstrakčius elementus su veiklos reikšme [7].

- PRR – *Production Rule Representation*. Tai produkcinių VT apibrėžimo standartas, suteikiantis galimybę taikyti VT komponentus kuriant IS. Šis standartas ypač patogus naudojant UML (*Unified Modeling Language*). Šiame standarte veiklos taisyklės aprašomos loginiais teiginiais (jeigu <sąlyga> tada <veiksmai> kitu atveju <veiksmai>) [8, 14, 37]. PRR numatyta vieta greta UML, šiuo standartu planuojama papildyti UML teikiamas modeliavimo galimybes, produkcinių VT modeliavimo galimybėmis [37].
- BMM – *Business Motivation Model*. Tai standartas, orientuotas veiklos tikslus ir būdus (strategijas) tiems tikslams pasiekti bei į šiuos du elementus įtakojančius faktorius [2, 23]. BMM standarte apibrėžta schema arba struktūra, kuria naudojantis galima vystyti ir valdyti verslo planus/strategijas. Taip sukuriamos sąlygos jiems sąveikauti tarpusavyje, siekiant organizuoto veiklos vystymo [17]. Pagrindinė BMM idėja yra sukurti veiklos modelį skirtą veiklos plano elementams, prieš suprojektuojant sistemą. Tokiu būdu veiklos planai gali tapti sistemos pagrindu ir orientuoti ją į konkrečią veiklą [7].
- OCL – *Object Constraint Language*. Tai modeliavimo kalba, skirta veiklos taisyklėms specifikuoti. Tai grynai formali specifikavimo kalba, todėl OCL išraiškos suprantamos tik vienareikšmiškai. [29]. Apdorojant OCL išraiškas, gaunama konkreti reikšmė, modelio veiklos tai neįtakoja, t.y. sistemos būseną niekada nepasikeis įvertinus OCL išraišką. Pradžioje ši kalba naudojama UML modelių apribojimams nusakyti, tačiau dabar ją galima taikyti ir kitiems OMG standartais apibrėžiamiems meta-modeliams.
- UML – *Unified Modeling Language*. tai OMG sukurta informacinių sistemų modeliavimo kalba. UML modeliavimo kalbai būdinga diagramų įvairovė, todėl ši modeliavimo kalba labai lanksti ir patogi įvairiam projektavimui. Tačiau UML neturi specializuotų VT modeliavimo priemonių [10, 25].
- BPDM – *Business Process Definition Metamodel*, tai OMG sukurtas veiklos procesų apibrėžimo meta-modelis, kurį galima naudoti kaip karkasas, bandant suprasti ir specifikuoti procesus vykstančius organizacijoje ar bendruomenėje [24].
- OSM – *Organizational Structure Metamodel*, tai veiklos lygio specifikacija, kuri suteikia galimybę sudaryti bendrą, nuolat pildomą duomenų rinkinį traktuojantį



organizacinius vienetus kaip resursus. Jo pagalba galima rinkti, klasifikuoti ir dokumentuoti informaciją apie organizacijos pozicijas, roles, nusakyti organizacinių vienetų galimybes ir priskirti jiems resursus [19]. Šis standartas dar yra kuriamas, oficiali jo specifikacija nėra paskelbta OMG puslapyje.

Minėti standartai priklauso OMG (*Object Management Group*) - tarptautinei standartų organizacijai, kurios tikslas sukurti modeliavimo standartus programoms, sistemoms ir verslo procesams [13, 22].

### **1.3. OMG standartai MDA architektūros kontekste**

Visi paminėti standartai patenka į OMG modeliais grindžiamos architektūros (*Model Driven Architecture* - MDA) koncepciją ir ją praturtina. Toliau bus bendrai apibūdinta MDA architektūra bei atlikta šių standartų analizė MDA architektūros kontekste.

#### **1.3.1. MDA architektūra**

Organizacijos OMG sukurta *Model Driven Architecture* (MDA) architektūra suteikia galimybę pažvelgti į verslo bei technologijų pokyčius nepriklausomai nuo programinės įrangos gamintojo. MDA yra paremta OMG standartais, šios architektūros esmė – verslo ir pritaikymo logikos atskyrimas nuo konkrečios platformos arba tiesiog IS kūrimo technologijos. MDA – tai naujas būdas kurti programinę įrangą bei kurti jai specifikacijas, remiantis nuo platformų nepriklausančiais modeliais – PIM (*Platform Independent Model*), nusakančiais verslo funkcijų specifika ar jų taikymą. Pilna MDA specifikacija susideda iš galutinio PIM bazinio modelio bei vieno ar daugiau nuo platformos priklausančio modelio – PSM (*Platform Specific Model*) bei aibės sąsajų apibrėžimų, nusakančių, kaip bazinis modelis yra diegiamas skirtingose tarpinėse platformose. Pilnas MDA taikymas susideda iš tikslaus PIM, vieno ar daugiau PSM ir pilnų pritaikymų kiekvienai platformai, kurią numatoma palaikyti [27].

MDA pirmiausia kreipia dėmesį į paskirstytos programinės įrangos ar sistemos funkcionalumą ir elgesį, neišskiriant išskirtinių platformos, kuriai bus taikoma, technologijos bruožų. Tokiu būdu, atskiriamos diegimo detalės nuo verslo funkcijų. Todėl, atsiradus naujai technologijai (pavyzdžiui tinklo paslaugoms – angl. *Web Services*), nereikia naujai apibrėžti programinės įrangos ar sistemos funkcionalumo ar elgesio. Kitos architektūros yra pririštos prie konkrečios technologijos. Remiantis MDA, sistemos funkcionalumas ir elgesys yra modeliuojamas tik viena kartą. Kelias nuo PIM per PSM į MDA palaikančias platformas, yra diegiamas naudojantis specializuotais įrankiais, - taip yra lengviau palaikyti naujas ar skirtingas technologijas [27].

### 1.3.2. MDA modeliai

MDA išskiria tam tikrus raktinius sistemų modelius ir pateikia nuoseklią tų modelių struktūrą. Visus modelius, nepriklausomai nuo sistemos, galima suskirstyti į CIM, PIM ir PSM:

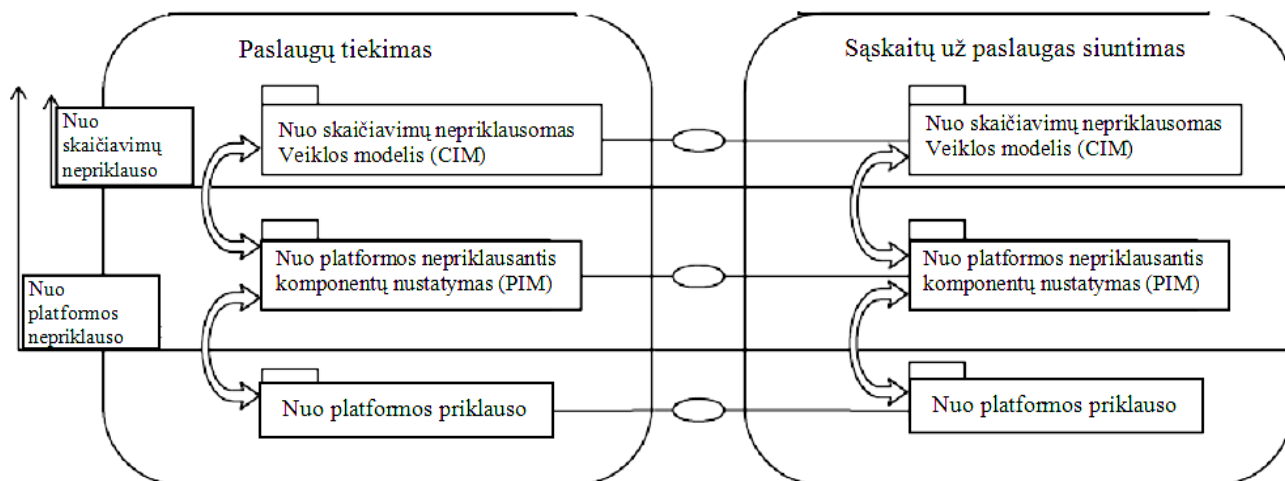
CIM – *Computation Independent Model*, modelis neparodo sistemos struktūros detalių. Šis modelis kartais vadinamas domeno (angl. *domain*) modeliu ir jo specifikacijoje naudojamas žodynas su terminais, kurie susiję su nagrinėjama sfera. CIM galima laikyti „tiltu“ tarp sferos ir jos reikalavimų ekspertų bei ekspertų projektuojančių ir kuriančių sistemos artefaktus [28].

PIM – *Platform Independent Model*, suteikia formalias sistemos struktūros ir funkcijų specifikacijas atmesdamas technines detales [26]. Jo komponentai ir sąsajos yra būdas gauti abstraktų informacinės programos ar sistemos vaizdą.

PSM – *Platform Specific Model*, sujungia PIM specifikacijas, su detalėmis nusakančiomis kaip sistemai naudotis konkrečia platforma.

MDA nusako atitinkamus ryšius tarp šių modelių. Kaip matome pav. 4, nagrinėjamoje sistemoje yra „papildančių“ ryšių (didelės dvipusės rodyklės) tarp veiklos modelio, nuo platformos nepriklausančių komponentų ir nuo platformos priklausančių komponentų. Taip pat tarp dviejų skirtingų sistemų, kurias norima sujungti, sąveikos gali būti nusakytos nuo platformos priklausančiu, nuo platformos nepriklausančiu ir net veiklos modelio abstrakcijos lygiu [26].

Kiekvienoje veikloje gali būti daugiau ir mažiau abstrakčių modelių, CIM, PIM ar PSM lygiuose. PIM (įskaitant ir CIM) ir PSM yra sistemos požiūrio specifikacijos naudojamos taikant kuri nors iš OMG standartų [26]. CIM kartais vadinamas – veiklos domeno modeliu (angl. *Business Domain*) arba tiesiog veiklos modeliu (angl. *Business Model*).



Šaltinis: Išversta iš Object Management Group (2001). *Model Driven Architecture (MDA)*

**Pav. 4 Atitinkami modelių ryšiai pagal MDA bei ryšiai tarp keleto sistemų**

Vienas iš MDA kūrėjų, John Siegel (2001), savo straipsnyje [36] teigia, jog modeliavimas yra vienintelis būdas užtikrinti, kad verslo informacinės sistemos įgytų tokį funkcionalumą, kurio reikalauja verslas – visapusišką, stabilų ir su galimybe tobulėti, kadangi laikui bėgant verslas bei jo pobūdis kinta. Vadinasi, sukurtas modelis gali nusakyti pačią verslo esmę, atmesdamas visus papildomus reikalavimus, kurie gali kilti kuriant vienai ar kitai platformai. Remiantis John Siegel (2001) – „Įmonės, su modelių saugykla, pasitikėjimas savo informacinėmis technologijomis, yra daug didesnis negu gali būti gaunamas iš kitų šaltinių“.

Prieš atsirandant MDA, modeliai ir programinis kodas būdavo kuriami atskirai. Programuotojai traktuodavo modelius kaip gaires ar apytikrius planus, bet ne kaip griežtus reikalavimus. Kildavo daug nepatogumų – palyginus su kūrimu be modeliavimo, kūrimas tokiame kontekste kainuodavo daugiau bei užtrukdavo ilgiau, o sukurta sistema turėdavo mažai panašumų į pradinį modelį, kuris atnešdavo nedaug naudos, nepaisant to, koks geras bebūdavo [36].

MDA tokią problemą pašalina susistemindama ir standartizuodama etapus, kuriuos turi pereiti modelis nuo kūrimo iki diegimo. MDA paremti įrankiai automatizuoja etapus, kuriuos apibrėžia pasirinktas OMG standartas, tokiu būdu sukurta programinė įranga atitinka ir verslo, ir ne tik (saugumas, detalumas ir pan.) reikalavimus. Modeliai tampa pagrindiniu įrankiu, apibrėždami ir saugodami ne tik verslo reikalavimus, bet ir būdami kaip pagrindas sistemos kūrimui, palaikymui ir evoliucijai [36].

Taigi, modeliavimą naudojant MDA galima apibendrinti kaip keturis pagrindinius etapus:

1. Pirmojo etapo metu, apibrėžiama kokiose sąlygose veiks kuriama sistema. Surenkami sistemos reikalavimai ir informacija apie taikymo sritį. CIM nedetalizuoja techninės informacijos apie tai, kaip sistema bus realizuota. Tiesiog surenkama informacija kas turės bus padaryta. Tai duoda naudos bandant suprasti problemų esmę bei suteikia bendrą žodyną, naudojama kuriant kitus modelius
2. Antrojo etapo metu sukuriamas aukšto abstrakcijos lygio modelis, kuris yra nepriklausomas nuo realizacijos platformos (PIM). Šis modelis nusako sistemos funkcionalumą ir elgesį, t.y. - esminius dalykus, kuriuos įtakoja tik pati verslo specifika. Tai kuriamos sistemos logika, konkrečios sąsajos tarp sistemos elementų.
3. Pagal PIM apibrėžiamas vienas ar daugiau modelių, kurie kuriami jau atsižvelgiant į konkrečią platformą (PSM). Šiame etape pagrindinis sistemos modelis surišamas su skirtingomis platformomis. Remiantis informacija, surinkta CIM, PSM modeliai realizuojami detaliau ir tiksliau.
4. Visi PSM modeliai realizuojami programiškai, t.y. viskas transformuojama į programinį kodą.

Nuo platformos nepriklausantis modelis (PIM) sudaromas remiantis veiklos taisyklėmis, kurios nusako verslo procesų logiką. OMG organizacija pastaraisiais metais pasiūlė keletą standartų, kurie galėtų naudojami norint apibrėžti veiklos taisykles.

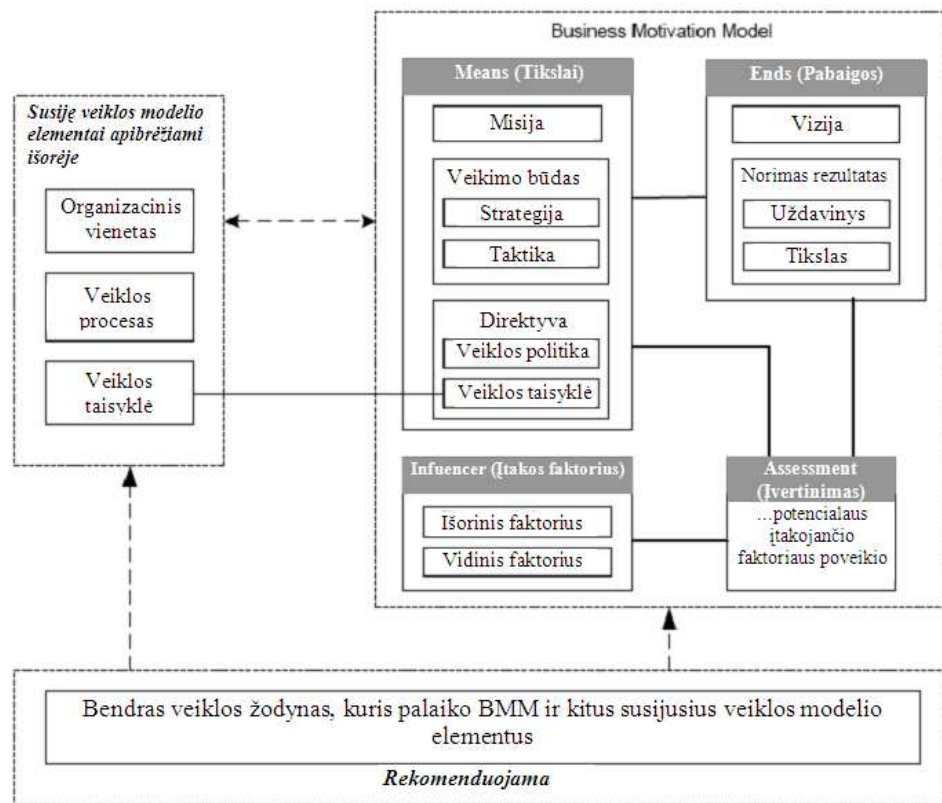
#### **1.4. OMG standartai MDA architektūra**

Kiekvienas standartas turi savo vietą MDA procese. Vieni standartai tinka veiklos modeliavimui ir CIM lygmeniui arba bendram sistemos modeliui aprašyti. Kitus standartus galima taikyti PIM lygyje. Toliau darbe bus nagrinėjami konkretūs OMG standartai bei jų pritaikymas, kuriant informacines sistemas pagal MDA koncepciją.

### 1.4.1. BMM

Veiklos motyvacijos modelis (angl. *Business Motivation Model*) sudarytas iš:

- Aibės konceptų, kurie nusako veiklos plano elementus ir yra sujungti į metodologijai neutralią struktūrą. Jie palaiko įvairius BMM modelio kūrimo ir palaikymo įmonėje būdus ir daugiausiai koncentruojasi į procesus, kuriuos nusako verslo pobūdis [23].
- Struktūros rolės turi tris esminius konceptus: Veiklos procesus (angl. *Business Process*), Veiklos taisykles (angl. *Business Rule*) ir Organizacini vienetą (angl. *Organization Unit*). Jie dalyvauja ryšiuose BMM viduje bei su ryšiais neįeinančiais į jo sferą – t.y. kitais OMG standartais – SBVR, BPDM ir OSM. Jie laikomi užuominomis į elementus, kurie bus apibrėžiami ir naudojami už BMM ribų [23]



Šaltinis: Išversta iš Object Management Group (2007). *Business Motivation Model (BMM) Specification*

**Pav. 5 Bendra BMM struktūra**

BMM modelis yra nesudėtingas, nes daugelis jo konceptų turi tik pagrindinius atributus – identifikatorių, tekstinį aprašą (plius suradimo galimybes įskaitant datas ir autoriaus pakeitimą). Daugelis jo ryšių yra neapriboti: laisvai pasirenkami ir daug su daug [23].

BMM modelis nėra:

- Specifikacija veiklos vystymo valdymo procesui ar įrankiui [23].
- Specifikacija projekto apibrėžimo ar valdymo procesui ar įrankiui [23].
- Pilno veiklos modelio specifikacija [23].

Jį galima įtraukti į tokias specifikacijas, tačiau jo vieno tam nepakanka.

BMM yra konceptualus įrankis pačios veiklos inžinerijai. Jis suteikia veiklos planų kūrėjams:

- Aibę konceptų, kuriuos galima naudoti kaip faktorių į kuriuos reikia atsižvelgti katalogą [23].
- Standartinį žodyną [23].
- Lankstų modelį palaikantį modeliuotojų kuriamus procesus [23].

Taip pat veiklos modeliuotojai gali naudotis įrankiais, kurie naudoja BMM veiklos planų saugojimui ir valdymui. Sukūrę detalius veiklos modelius, modeliuotojai ateityje gali naudotis kitais OMG standartais ir modeliais, tokiais kaip BPDM, SBVR ir OSM [23].

Iš esmės BMM taikomas dvejose srityse

- Veiklos planų Pabaigoms (angl. *Ends*) ir Priemonėms (angl. *Means*) nusakyti. Pabaigomis laikoma tai, ką įmonė nori pasiekti – tarkim Uždaviniai (angl. *Goals*) ir Tikslai (angl. *Objectives*). Priemonėmis laikoma dalykai, kuriuos įmonė naudos, kad pasiektų norimą Pabaigą – tarkim Strategijos, Taktikos, Veiklos politika ir Veiklos taisyklės [23].
- Įtakojančiams faktoriams (angl. *Influencers*), kurie formuoja veiklos planų elementus, ir Įvertinimams (angl. *Assesments*), kurie padaromi susidūrus su Įtakojančiais faktoriais siekiant Pabaigos ir naudojant Priemones (pvz. Stiprybės, Silpnybės, Galimybės ir Grėsmės angl. – *Strengths, Weaknesses, Opportunities, Threats*), nusakyti [23].

Visi BMM elementai yra sudaromi iš verslo perspektyvos. Esminė idėja yra sukurti veiklos modelį elementams ir veiklos planams, **prieš** prasidedant sistemos projektavimui ar techniniam vystymui. Tokiu atveju, veiklos planai gali tapti visos veiklos pagrindu, sujungdami sisteminius sprendimus su veiklos tikslais [23].

Taigi, MDA kontekste šis standartas eina pačioje pradžioje, verslo esmės apibrėžime. Sudarius tokį abstraktų modelį, galima juo remtis sudarinėjant veiklos taisykles ir veiklos taisyklių žodynus, iš kurių sudaromi PIM.

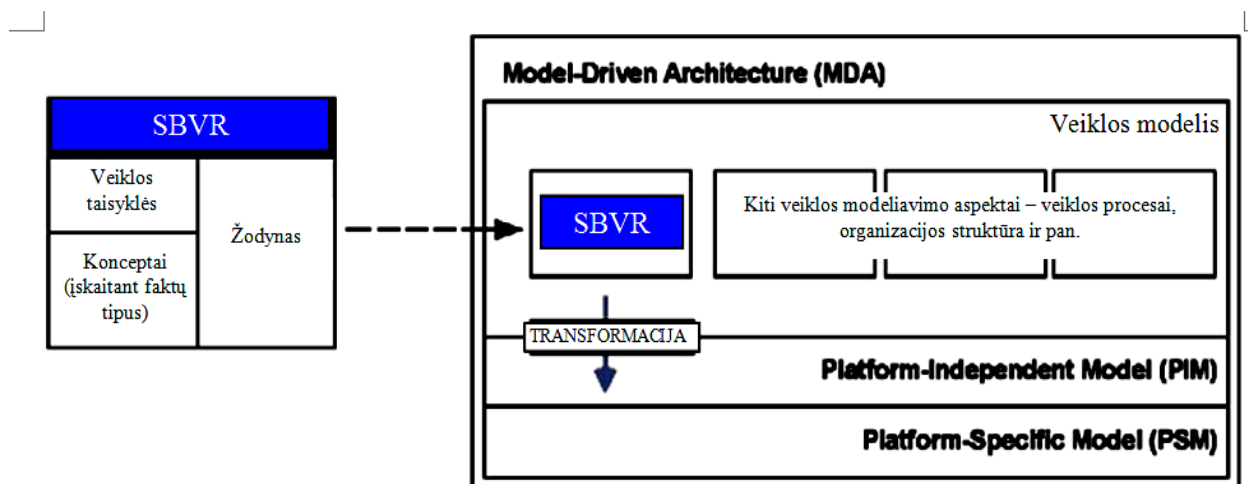
### 1.4.2. SBVR

Semantikos veiklos žodynui ir veiklos taisyklėms (*Semantics of Business Vocabulary and Business Rules* – SBVR) specifikacija iš esmės apima du meta-modelius, suteikdama jiems žodyną:

- SBVR „*Vocabulary for Describing Business Vocabularies*“ (Žodyną verslo žodynui apsaityti, šis meta-modelis dar vadinamas verslo žodyno meta-modeliu) [30]
- SBVR „*Vocabulary for Describing Business Rules*“ (Žodyną veiklos taisyklėms nusakyti, šis meta-modelis dar vadinamas veiklos taisyklių meta-modeliu), kuris sudaromas pagal „*Vocabulary for Describing Business Vocabularies*“ [30]

Verslo žodynas apibrėžiamas kaip vieta kur sukaupti visi specializuoti terminai ir koncepcijų apibrėžimai, kuriuos naudoja nagrinėjama organizacija ar bendruomenė kalbėdama ar rašydama apie savo veiklą. Veiklos taisyklė apibrėžiama kaip taisyklė veikianti pagal verslo jurisdikciją, tai reiškia, kad verslas gali priimti, koreguoti ir nutraukti veiklos taisykles, pagal savo poreikį [30].

SBVR vieta MDA architektūroje yra veiklos modelio t.y. CIM sluoksnis pav. 6.



Šaltinis: Išversta iš Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR)*, v1.0  
OMG Available Specification

**Pav. 6 SBVR vieta MDA architektūroje**

Taigi SBVR taikomas dvejopai :

- SBVR taikomas veiklos taisyklėms ir verslo žodynams, įskaitant ir tuos kurie susisiję su tais kurie naudojami su šiomis taisyklėmis. Kiti veiklos modelių aspektai taip pat turi būti kuriami, įskaitant ir verslo procesus bei organizacijos struktūrą, bet jie turi būti apdorojami naudojant kitas OMG iniciatyvas [30].
- Veiklos modeliams, įskaitant ir tuos modelius, kuriuos palaiko SBVR, tam, kad nusakytų veiklas, o ne informacines sistemas kurios tas veiklas palaiko [30].

SBVR surenka specifinę terminologiją ir reikšmes, visuose MDA lygiuose, taigi SBVR galima naudoti ir PIM ir PSM modelių žodynuose ir taisyklėse. Tačiau, specifikacijos sukaustos SBVR yra daugiau įrankis pačios veiklos, o ne ja skirtų informacinių sistemų nusakymui [30].

#### **1.4.3. BPDM**

Veiklos procesų apibrėžimo meta-modelis (*Business Process Definition Metamodel - BPDM*) yra karkasas, kurio pagalba galima suprasti ir specifikuot procesus vykstančius organizacijoje ar bendruomenėje. BPDM suteikia galimybę atvaizduoti ir modeliuoti veiklos procesus nepriklausomai nuo matavimo sistemos ar metodologijos, taigi sujungdamas šiuos skirtingus požiūrius. Tai atliekama naudojantis meta-modeliu – modeliu, skirtu nusakyti veiklos procesus – tai kaip bendras procesų žodynas su aiškiai išreikštais ryšiais tarp terminų ir konceptų. Šis meta-modelis sujungia reikšmes tarp matavimo sistemų ir technologijų tokiu būdu, kad pasidaro lengviau juos integruoti ir įtakuoti egzistuojančius ir naujus projektus [24].

SBVR ir BPDM vienas kitą papildo. SBVR specifikuoja veiklos žodyno ir taisyklių reikšmes bei atvaizdavimą. BPDM nusako veiklos žodyno ir taisyklių vartojimą įvairių BPDM modelio elemento. Pradinis ryšys tarp SBVR ir BPDM yra rolės kurias atlieka veiklos taisyklės BPDM. Taip pat SBVR gali suteikti formalius logika paremtus apibrėžimus visiems BPDM modelio elementams [24].

#### **1.4.4. PRR**

Produkcinių veiklos taisyklių vaizdavimas (*Production Rule Representation – PRR*) atitinka daugelį reikalavimų keliamų produkcinėms veiklos taisyklėms, programinės įrangos sistemoms, OMG standartams ir kitiems taisyklių standartams [32].



- PRR suteikia veiklos taisyklių atvaizdavimo standartą, kuris yra suderinamas su veiklos taisyklių variklio gamintojų veiklos taisyklių apibrėžimais. Taigi jis gali būti naudojamas veiklos taisyklių pasikeitimui tarp veiklos taisyklių modeliavimo įrankių (ir kitų įrankių, kurie palaiko veiklos taisyklių modeliavimą kaip funkciją ar kokią kitą užduotį) [32].
- Suteikia veiklos taisyklių atvaizdavimo standartą kuris lengvai susisieja su veiklos taisyklėmis, kurias nusako įvairūs veiklos taisyklių valdymo įrankių gamintojai [32].
- Suteikia veiklos taisyklių atvaizdavimo standartą, kuris suteikia galimybę sistemų gamintojams palaikyti veiklos taisyklių vykdymą [32].
- Suteikia galimybę sudaryti OMG MDA PIM modelį su didele tikimybe, kad PSM lygis bus palaikomas visuose bendradarbiaujančių gamintojų taisyklių varikliuose ir kad jie turės galimybę pridėti veiklos taisykles prie kitų OMG meta-modelių [32].
- Suteikia pavyzdžius kaip OMG UML gali būti naudojamas veiklos taisyklių palaikymui standartizuotu ir naudingu būdu [32].
- Suteikia standartinį veiklos taisyklių atvaizdavimą, kuris gali būti naudojamas kaip pagrindas kitiems taisyklių valymo variantams tokiems kaip W3C Rule Interchange Format (Taisyklių pasikeitimo formatas – RIF) ir RuleML veiklos taisyklių versijai [32].

MDA kontekste PRR - Production Rule Representation (PRR) – naudojamas kuriant PIM skirtą pavaizduoti veiklos taisykles UML. PRR tikslas veiklos taisyklių variklio nepriklausoma nuo klasių platforma, kurią galima plačiai naudoti nuo tiekėjo nepriklausančiame variklyje. PRR toliau apribotas reikalavimų specifikavimui dviejose populiariausiose taisyklių variklio formose – tiesioginio išvedimo/išvadų darymo (angl. *forward chaining/inferencing*) ir procedūrų su nuo klasių nepriklausančiomis platformoms. Šie du tipai apima visus sprendimus nuo sudėtingų priimančių sprendimą iki paprastų palaikančių verslo procesų valdymą (*Business Process Management* – BPM) [24].

### 1.5. UML, OCL ir veiklos taisyklės

Veiklos taisyklių modeliavimas UML yra plati tema, pavyzdžiui:

- Veiklos taisyklės yra nusakomos kaip formalūs tekstai dokumentacijai ar reikalavimams gali būti nusakomos SBVR standarto [32].
- Veiklos taisyklės, tokios kaip paprasti ryšiai ir apribojimai turi paprastus atitikmenis UML modelyje. Tarkime, veiklos taisyklė skambanti „užsakymai turi turėti bent vieną eilutės

elementą“ būtų paprastai pavaizduota kaip ryšio daugialypiškumo apribojimas. Kitos taisyklės lengvai išverčiamos į apribojimų išraiškas. Pavyzdžiui, taisyklės „gimtadienis turi būti anksčiau arba lygus esamai datai“ arba „sąskaita niekada negali turėti neigiamo balanso“ lengvai išreiškiamos kaip invariantai naudojant OCL kalbą [32].

- Į procesą orientuotos veiklos taisyklės nusako sąlyginius veiksmų/elgesio/būsenos pasikeitimus kaip veiklos taisykles, kurių OCL negali apdoroti [32].

Du UML mechanizmai, kurie apibūdina apribojimus ir elgesį yra OCL (*Object Constraint Language*) ir AS (*Action Semantics* – veiksmų semantika). Tačiau nei vienas iš jų nesuteikia galimybės atskirai apibūdinti veiklos taisykles [32]. PRR suteikia UML galimybę apimti veiklos taisykles.

OCL – labai plati išraiškų kalba, kuri apibrėžia modelio užklausų operacijas. Tačiau šios kalbos trūkumas, kad OCL negali aprašyti šalutinių poveikių, todėl nepalaiko tiesioginio modelio kreipimosi į metodus, kurie pakeičia sistemos būseną (to reikalauja veiklos taisyklės arba jų poaibis produkcinės taisyklės). Naujesnė OCL 2.0 versija leidžia sąsajas su operacijomis, kurios pakeičia sistemos būseną naudojant apribojimus, tačiau tokios sąsajos semantika paremta tuo, kad į operacija kreipiamasi tik tada kai patikrinama teisingumo konstanta. Tokia semantika, kuri leidžiama tik žemesnio lygio sąlygose, nepatenkina veiksmų sakinių (angl. *action clause*) reikalavimų keliamų produkcinėms taisyklėms, nes pastarosios negali būti naudojamos kaip operacijų žemesnio lygio sąlygos. Todėl OCL veiklos taisyklių valdymo įrankių kūrėjai nenaudoja kaip sintaksės veiklos taisyklėms aprašyti [31].

Veiksmų semantika (angl. *Action Semantics*) tai poreikis pavaizduoti elgesį su šalutiniais poveikiais, tokius kaip metodų iššaukimas su *action clause* produkcinėms taisyklėms sakiniams, duoda galimybę modeliuoti produkcinės taisyklės naudojant AS. AS palaiko *jeigu* <sąlyga>, *tada* <veiksmas> formos teiginius. Tačiau ne visa produkcinė veiklos taisyklių semantika atitinka AS [32].

- Vykdyto semantika (angl. *execution semantics*). AS leidžia du veiksmų semantikos vykdymo režimus: lygiagrečių vykdymą ir nuoseklią vykdymą paremtą aiškiai sumodeliuotų valdomų srautų arba duomenų srautų tarp veiksmų sakinių. Išvadų taisyklėms trūksta aiškaus sekos modeliavimo. Iš tikrųjų modeliuojant problemą, arba sprendimą naudojantis išvadų taisyklėmis, esminis poreikis yra nusakyti taisyklių vykdymo seką, nepaisant pačių taisyklių teiginių semantikos. Išvadų variklį galima laikyti kaip būdą apdoroti pačią išvadų seką paremtą veikimo momento sąlygomis [32].
- Daugialypės kvantifikatorių išraiškos (angl. *multiple quantified expressions*). AS suteikia išraiškoms kurios turi aibę atvejų klasifikatorių. Tačiau AS nepalaiko daugialypių

kvantifikatorių su ta pačia išraiška, ko reikalauja produkcinės veiklos taisyklės, t.y. jos nepalaiko išraiškų su aibėmis nekintančių reikšmių (angl. *tuples*) [32]. Kitaip tariant, su AS negalima apibūdinti išraiškų su keletu kintamųjų, turinčių keletą atributų, o tai labai svarbu kuriant taisykles algoritams, kurie pateikia išvadas.

## 1.6. Standartų palyginimas

Standartai UML ir OCL yra paprastai taikomi sistemos projektavimo stadijoje. UML suteikia galimybę kurti įvairius informacinės sistemos modelius, šis standartas daugiausiai skirtas pačios sistemos modeliavimui, veiklos taisyklėm aprašyti jis nėra patogus. OCL standarto pagalba galima aprašyti UML modelių apribojimus, tačiau šis standartas yra gana sunkiai išmokstamas ir jo pagalba galima formalizuoti tik dalį veiklos taisyklių.

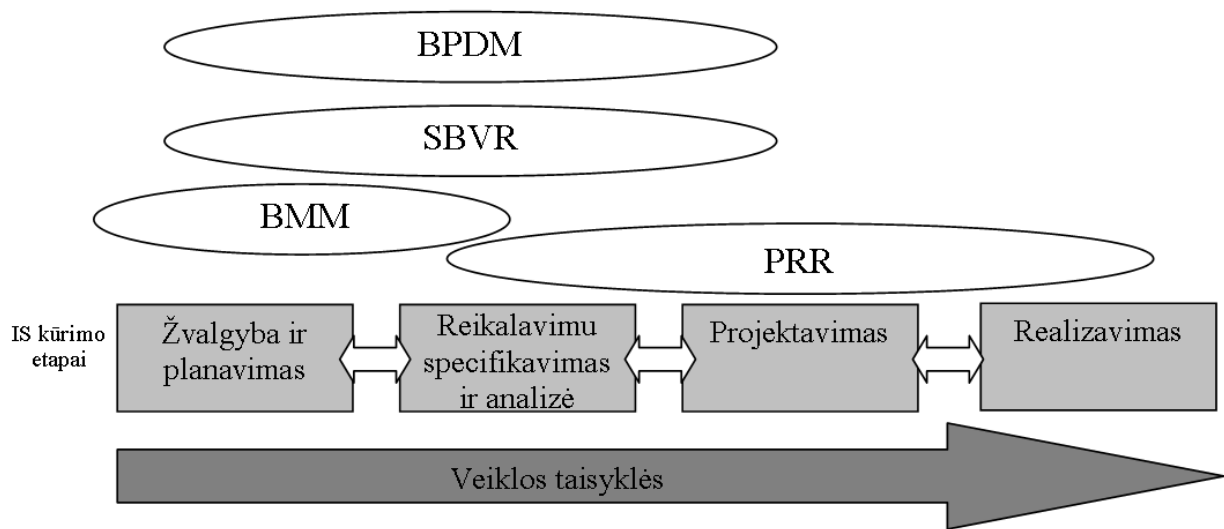
Informacinės sistemos reikalavimų specifikavimui naudojami standartai SBVR, PRR ir BMM. SBVR standartą galima palyginti su veiklos taisyklių žodynu. Jo pagalba galima formalizuoti taisykles, suteikti joms tam tikrą struktūrą ankstyvose reikalavimų specifikavimo stadijose. SBVR nagrinėja loginę sakinių, kuriais išreikšti reikalavimai veiklai, struktūrą. SBVR apibrėžimo reikšmė nustatoma pagal reikšmės semantinę formuluotę [30].

BPDM ir SBVR standartai vienas kitą papildo. BPDM daugiau skirtas veiklos procesu apibrėžimui, tuo tarpu SBVR yra daugiau skirtas apibrėžti veiklos taisykles. Nagrinėjant veiklos taisyklių specifikavimą MDA kontekste geriau tinka SBVR standartas, nes jis daugiau orientuotas į veiklos taisykles, abiejų standartų naudoti nebūtina.

PRR tai vienas iš naujausių OMG priimtų standartų [22]. Naudojantis PRR, veiklos taisykles galima užrašyti formaliai. PRR daugiausiai skirtas pagerinti veiklos taisyklių specifikavimą UML. PRR išreikštas veiklos taisykles jau galima naudoti sistemos realizacijoje.

BMM standartas apibrėžia veiklos planų kūrimo schemą [20]. Naudojantis BMM galima aprašyti kuriamos sistemos tikslus, strategijas ir visą tai įtakančius faktorius [7]. Tai daugiau veiklos modeliavimui skirtas sprendimas.

Lyginant reikalavimų specifikavimo standartus galima teigti, kad, BMM – tai standartas, kuris yra arčiausiai veiklos, bet toliausiai nuo realizacijos. SBVR jau arčiau realizacijos, juo jau galima formalizuotai apibrėžti apribojimus ir sistemos veiklą. PRR yra arčiausiai realizacijos, naudojantis šiuo standartu veiklos taisyklės pilnai formalizuojamos (pav. 4).



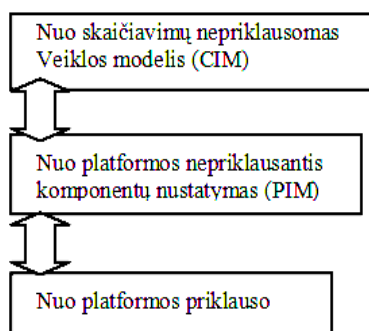
Šaltinis: Sudaryta autoriaus.

**Pav. 7. Veiklos taisyklių modeliavimo standartų panaudojimas kuriant IS.**

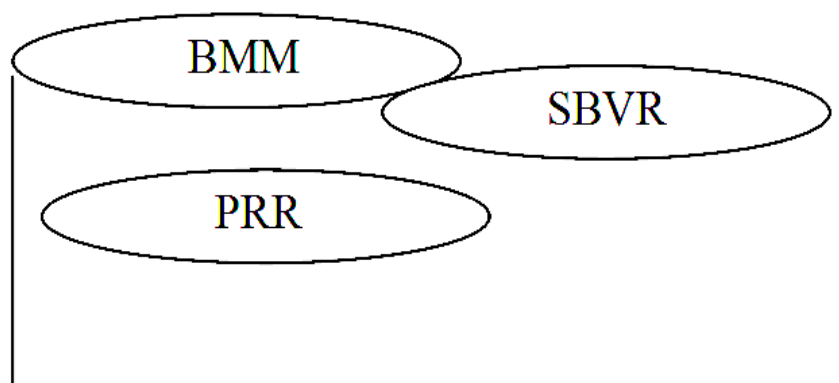
Kadangi darbo esmė yra šiuolaikinių veiklos taisyklių standartų taikymas kuriant informacines sistemas, BPDM naudoti nebūtina, nes jo funkcijas galima realizuoti naudojant BMM ir SBVR. Taip pat šis standartas daugiau skirtas sukurti procesų, vykstančių įmonėje, žodynui sudaryti, veiklos taisyklių modeliavimo funkciją geriau atlieka SBVR.

Taigi, nagrinėjant likusius tris standartus – BMM, SBVR ir PRR, galima apibrėžti jų vietą MDA architektūros kontekste (žr. pav. 8).

### MDA lygmenys



### Standartai



Šaltinis: Sudaryta autoriaus

**Pav. 8. OMG standartų BMM, SBVR ir PRR vieta MDA architektūros kontekste**

BMM standartas naudingas atliekant pačią įmonės ir veiklos analizę bei nustatant pagrindinius jos elementus. Šis standartas daugiau tinka veiklos modeliavimui. Darbo esmė – šiuolaikinių veiklos taisyklių standartų taikymas kuriant informacines sistemas. Standartas SBVR skirtas veiklos taisyklių žodynui sukurti, standartas PRR skirtas produkcinėms veiklos taisyklėms specifikuoti. SBVR kaupia terminus, kurie naudojami įmonėje ar kažkokioje konkrečioje veikloje, o su PRR suformuojami loginiai sakiniai kuriuos galima naudoti MDA architektūroje nuo platformos nepriklausančio modelio PIM kūrime.

Panagrinėjus ir palyginus esamus OMG standartus, skirtus veiklos modeliavimui bei veiklos taisyklių specifikuojimui, galima teigti, kad darbo temai geriausiai tinka SBVR ir PRR. Kiti nagrinėti standartai daugiau tinka pačios veiklos modeliavimui, o ne veiklos taisyklių specifikuojimui

### **1.7. Analizės išvados ir tolimesnio tyrimo gairės**

Apžvelgus esamus standartus, galima išskirti, kad specifikuojant veiklos taisykles reikšmingiausi būtų SBVR ir PRR. Šie standartai sąlyginai nauji, todėl detalių metodikų jiems nėra sukurta. Kiti standartai buvo atmesti, nes nagrinėjama gana specifinė veiklos modeliavimo sritis – veiklos taisyklės. BMM, BPDM daugiau yra skirti bendram veiklos modeliavimui, o standartas OSM – daugiau modeliuoti veiklos resursus, be to, jo specifikacija vis dar nėra išleista. Tolimesnėje tyrimo eigoje reikėtų išnagrinėti SBVR ir PRR standartų ypatumus bei suderinamumą.

PRR specifikacijoje teigiama, kad :

SBVR specifikacijoje nėra tiksliai apibūdinta, kaip gauti PIM lygio taisyklių išraiškas iš SBVR veiklos taisyklių teiginių. Iš esmės nėra aišku, ar kuri nors iš SBVR taisyklių meta-modelio koncepcinių charakteristikų gali būti naudojama PRR taisyklių modelyje [31].

Taigi, remiantis šiuo teiginiu, matome, kad dar nėra ištirta SBVR ir PRR sąsaja ir ar apskritai toks dalykas įmanomas. Tačiau toje pačioje specifikacijoje pabrėžiama, kad:

Nepaisant sunkumų, būdingų transformuojant SBVR į PRR, yra aiški šių standartų susiejimo nauda. Toks susiejimas leistų atlikti poveikio analizę („kurie PRR taisyklių rinkiniai yra paveikiami jei pakeičiama konkreti SBVR taisyklė“) ir sumažintų nuolatinio palaikymo kaštus. Apie tokį susiejimą yra nuolat diskutuojama [31].

Vadinasi, ryšys tarp SBVR žodynų ir PRR taisyklių formuluočių dar nėra ištirtas, tačiau tokia analizė būtų labai naudinga.

Tolimesnio tyrimo metu labiausiai bus koncentruojamasi į galimybę susieti SVBR ir PRR. Šie standartai turi gana sudėtingas angliškas formuluotes, taigi būtina išnagrinėti, kaip tos formuluotės gali būti taikomos lietuvių kalbos kontekste.

Atlikus analizę planuojama sukurti šių standartų taikymo metodiką, t. y. apibrėžti tam tikras pasirinktų standartų taikymo taisykles ir procesą. Kuriant informacinę sistemą reikia užfiksuoti ne tik taisykles, bet ir aprašyti duomenų struktūras, procesus, sistemos funkcijas. Taigi, kuriant metodiką, planuojama detalizuoti IS kūrimo stadijos eigos procesą, kuriame būtų panaudojami šie standartai. Atliekant sukurtosios metodikos bandymą, bus pasirinkta dalykinė sritis ir jai taikomi pasirinktieji standartai. Su SVBR bus sukurtas veiklos taisyklių žodynas, iš kurio bus bandoma suformuoti veiklos taisykles, kurios galėtų tikti MDA architektūros PIM lygmeniui, naudojantis PRR.

Planuojama tyrimo eiga:

1. SBVR ir PRR standartų panaudojimo metodikos koncepcijos apibrėžimas.
2. Veiklos taisyklių specifikuojimo SBVR standartu gairių apibrėžimas
3. Veiklos taisyklių specifikuojimo PRR standartu gairių apibrėžimas
4. SBVR išraiškų transformavimo į PRR galimybių tyrimas, metodikos detalizavimas

Atlikus tyrimą planuojama nustatyti, kiek galima susieti SBVR ir PRR standartus specifikuojant veiklos taisykles. Sukūrus standartų taikymo metodiką, jos veiksmingumą planuojama išbandyti realizuojant ją praktiškai, t.y. sudarant žodyno klasių diagramas, apibrėžiant veiklos taisyklių saugyklą bei aprašant tikslius SBVR standarto šablonus. Atlikus tai, planuojama įvertinti metodiką, pagal ją specifikuojant pasirinktos dalykinės srities veiklos taisykles bei apžvelgiant rezultatus.

## **2. SBVR IR PRR VEIKLOS TAISYKLIŲ STANDARTŲ SUSIEJIMO PRIELAIDŲ TYRIMAS**

Poreikis taikyti veiklos taisykles kuriant informacines sistemas nuolat didėja, tačiau naudojant naujausius VT standartus kyla įvairūs sunkumai, nes jų taikymo metodika nėra pakankamai plačiai aprašyta. Taigi šio darbo objektas yra šiuolaikinių veiklos taisyklių standartų taikymas kuriant informacines sistemas.

Atlikus OMG standartų apžvalgą, pasirinkti du standartai SBVR ir PRR, kurie atitinka MDA CIM ir PIM modelius. SBVR apibrėžia nagrinėjamoje veiklos srityje naudojamas sąvokas ir jų sąsajas, o PRR nusako produkcinės veiklos taisykles – t.y. sąlygas ir veiksmus, kurie atliekami šias sąlygas tenkinant arba ne.

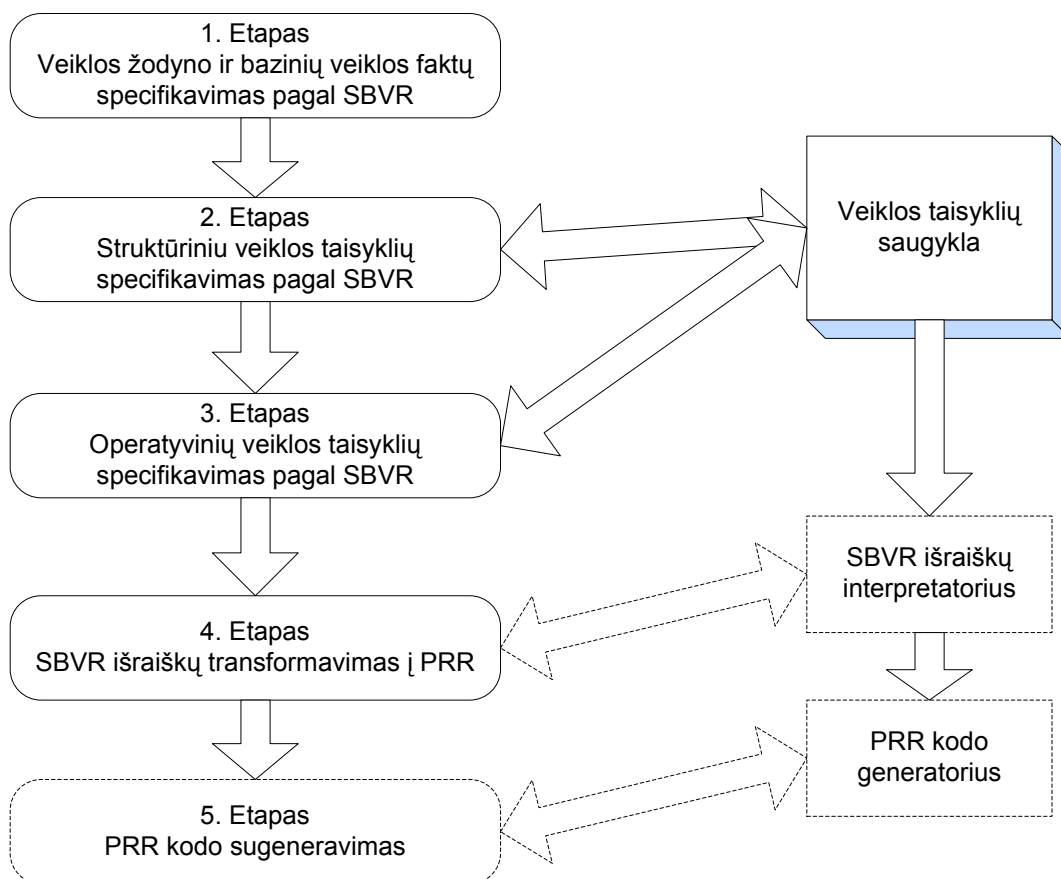
### **2.1. SBVR ir PRR standartų naudojimo metodikos koncepcija**

Išnagrinėjus SBVR ir PRR standartus, matome, kad pirmasis standartas yra labai platus ir juo galima apibrėžti labai įvairias taisykles ir veiklos koncepcijas. Tačiau šis standartas yra labai priklausomas nuo natūraliosios kalbos, todėl SBVR standarto automatizavimas yra sunkiai realizuojamas. PRR standartas yra labai konkretus ir lengvai realizuojamas. Tačiau juo užrašytos taisyklės yra per daug abstrakčios.

PRR specifikacijoje pateikiamas toks PRR panaudojimo atvejis [31]:

- Taisyklės nusakomos formalia kalba, pvz., OMG SBVR. Sukuriamas klasių modelis ir sujungiamas su atitinkamo MDA CIM lygio konstrukcijomis. Po taisyklių modeliavimo, galima atlikti įvairių tipų taisyklių transformaciją į PIM lygį [31].

Taigi, sudarius MDA CIM lygio konstrukcijas pagal SBVR, jos būtų kaip gairės, kuriant PIM lygį. Tačiau SBVR taisyklių šablonai turi būti griežtai apibrėžti, norint, kad iš jų būtų galima generuoti PRR išraiškas. Metodikos schema, pateikiama pav. 9.



Šaltinis: Sudaryta autoriaus

**Pav. 9. Siūlomos SBVR ir PRR standartų taikymo metodikos schema**

Šių standartų taikymo metodika sudaryta iš penkių etapų:

1. etapas. Apibrėžiami pagrindiniai sistemos konceptai ir baziniai faktai.
2. etapas. Iš jų sąryšių suformuojamos struktūrinės taisyklės.
3. etapas. Apibrėžti konceptai i struktūrinės taisyklės – tai pagrindas operatyvinių taisyklių kūrimui. Pagal juos veiklos ekspertas sudaro operatyvines taisykles. (Šablonuose struktūrinės taisyklės traktuojamos kaip apribojimai; operatoriai, kvantifikatoriai ir pan. – kaip sąlygos; konceptai – kaip kintamieji)
4. etapas. Pagal SBVR šablonus užrašytos VT transformuojamos į PRR išraiškas.
5. etapas. Sugeneruojamas PRR kodas (PRR OCL ar kokia kita nepriklausančia nuo platformos kalba).

PRR išraiškoms sudaryti naudojamos SBVR operatyvinės taisyklės, nes būtent šios taisyklės skirtos verslo procesų valdymui ir galima patikrinti, ar jų laikomasi, ar ne. Struktūrinės taisyklės apibrėžia tik konceptų sąsajas, papildo apibrėžimus.



## 2.2. Veiklos taisyklių specifikuojimas SBVR standartu

Remiantis SBVR standartu – taisyklė yra: Vadovavimo elementas, pateikiantis įsipareigojimus arba būtinybę [30].

Taisyklės skirstomos į dvi esmines kategorijas [30]:

- **Struktūrinės taisyklės:** Tai taisyklės nusakančios kaip sisteminami esminiai verslo elementai. Struktūrinės taisyklės papildo apibrėžimus, pavyzdžiui:

Būtinybė: *Klientas turi turėti bent vieną iš išvardintų dalykų:*

- Rezervuotą Nuomą
  - Nuomą esamu laiku
  - Turėjęs Nuomą per pastaruosius 5 metus
- **Operatyvinės taisyklės:** Tai taisyklės, kurios valdo verslo procesų elgesį. Priešingai negu Struktūrinės taisyklės, Operatyvinės taisyklės gali būti *tiesiogiai* pažeistos žmonių, susijusių su verslo reikalais, pavyzdžiui:

*Klientui, kuris atrodo apsvaigęs, negalima duoti Nuomojamos Mašinos.*

Visos veiklos taisyklės (ir patarimai) turi būti realios. Ar elementas, kuriuo vadovaujama, yra realus apsprendžiama pagal [30]:

- Operatyvioms veiklos taisyklėms tai reiškia, kad reikia išanalizuoti žmonių elgesį ir kuri jų elgesio forma laikoma teisinga. Kadangi operatyvi veiklos taisyklė yra pritaikoma praktiškai, žmogus kuris apie ją žino, gali iš karto nuspręsti ar jina nepažeidžiama.
- Struktūrinėms veiklos taisyklėms tai reiškia, kad įvertinamas kriterijus priklausantis taisyklei ir duodantis nutarimo ar skaičiavimų rezultatą (-us). Jeigu struktūrinę taisyklę galima pritaikyti praktiškai, žmogus žinantis apie ją, gali nuspręsti ar jos laikomasi, išanalizavęs gautus rezultatus.

Taisyklėse, kurias galima pritaikyti praktiškai, niekada nebūna neapibrėžtų nuorodų į žmones (pvz. „tu“, „aš“), vietas (pvz. „čia“) ir laiką (pvz. „dabar“). Tai reiškia, jeigu žmogus nežino kokia vieta ir/arba laikas, kurį omenyje turėjo taisyklės autorius(-iai), jis vis tiek tą taisyklę gali suprasti be (a) kokio nors įrenginio pagalbos (pvz. tam, kad „praneštų“ laiką), (b) pašalinio paaiškinimo. Remiantis

tu, išplaukia tai, kad esminiai verslo žodynai turi būti gerai suformuoti ir gerai tvarkomi. Tiksliau sakant, kiekvienas verslo konceptas turėtų būti [30]:

- Atskirai paėmus gerai apibrėžtas
- Logiškai pritapti prie bendros konceptų struktūros
- Būti prieinamas vartotojui tinkamu būdu

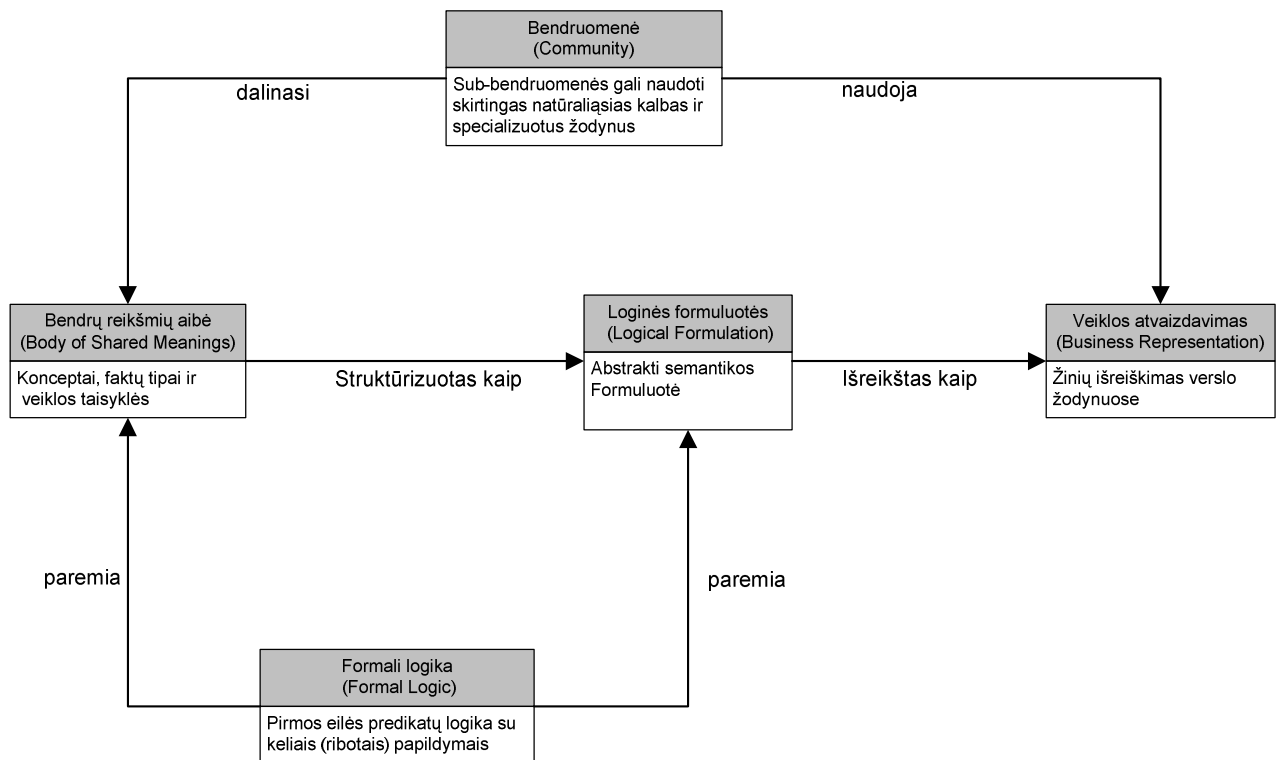
Visos operatyvinės taisyklės turi būti tiesiogiai įvykdomos. Tam, kad būtų įvykdoma, operatyvinė veiklos taisyklė turi būti išreikšta taip, jog būtų galima užfiksuoti jos pažeidimus.

Valdymo elementai tiesiogiai žmonių veiklą versle ir jie turi būti įvykdomi. Tiesioginis įvykdymas skiria veiklos politiką nuo operatyvinių veiklos taisyklių. Esme tame, kad kai žmonės specifikuoja verslo atvejų (ar turi apibrėžti) valdymą realiame pasaulyje, jiems reikia atsižvelgti į du faktorius [30]:

- Ar valdymo elementas yra tiesiogiai įvykdomas – t.y. ar galima stebėti žmonių veiklą ir išskirti ar jie tenkina reikalavimus be tolimesnio valdymo elemento plėtojimo ar aiškinimo. Taipogi reikia sudaryti operatyvines veiklos taisykles, išvestas iš veiklos politikos, kurios yra tiesiogiai įvykdomos.
- Jeigu valdymo elementas yra tiesiogiai įvykdomas, jis turi būti kilęs iš verslo politikos. Jeigu taip nėra, verslo projektuotojai turi į tai atsižvelgti (apsvarstyti ar taisyklė yra tinkama)

Nors veiklos taisyklės yra realios, tai nebūtinai reiškia, kad jos yra automatizuojamos. Daugelis veiklos taisyklių (ypač operatyvinių) yra ne automatizuojamos IS sistemose. Šis faktas nėra svarbus SBVR, kuris daugiausiai orientuotas į veiklos perspektyvą ir nepaiso to ar sukurtos taisyklės galima realizuoti IS. Tačiau, automatizavimas svarbus iš veiklos modelio kuriant PIM. Iš esmės, neautomatizuojamos veiklos taisyklės, turi būti realizuojamos, kaip vartotojo veiksmai ir paaiškinamos procedūrų instrukcijomis arba taisyklių knygomis.

SBVR sudaro penki pagrindiniai aspektai (pav. 9):



Šaltinis: Išversta iš Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR)*, v1.0  
OMG Available Specification

### Pav. 10. Penki pagrindiniai SBVR aspektai

- **Bendruomenė** - Verslo žodynėlio pagrindas yra bendruomenė. Verslo lygyje, bendruomenėmis laikomos įmonės, kurių veiklos taisyklės yra kuriamos ir išreiškiamos. Kitos bendruomenės yra pramonės šaka, kurioje veikia įmonė, partneriai, taikomi standartai, vadovybės – visa tai taip pat turi būti identifikuota. Svarbus bendruomenės aspektas yra tai, kad sub-bendruomenės, esančioms įmonėje, gali prireikti bendrų reikšmių (pradedant nuo fundamentalių konceptų) išreikštų įvairiuose žodynuose apimančiuose sąvokas nuo specializuoto žargono iki įvairių natūraliųjų kalbų. SBVR tokios bendruomenės vadinamos „kalbos bendruomenėmis“ [30].
- **Bendrų reikšmių aibė** - Bendruomenė turi bendrų reikšmių aibę, sudarytą iš konceptų (įskaitant ir faktų tipus) ir veiklos taisyklių. Dalinamasi reikšme, o ne faktų tipo forma. Savaime aišku, jog norint bendras reikšmes dalintis, aptarti ir patvirtinti – jos turi būti išreikštos. Tačiau SBVR atskira verslo reikšmę nuo bet kurios konkrečios fakto reikšmės. Bendrų reikšmių aibės struktūra, nusakoma asocijuojant abstrakčius konceptus, faktų tipus ir veiklos taisykles, o ne asocijuojant teiginius kokia nors kalba [30].

- **Loginės formuluotės** - Loginės formuluotės suteikia formalią, abstrakčią, nepriklausančią nuo kalbos sintaksę, tam, kad būtų galima užfiksuoti bendrų reikšmių aibės semantiką. Palaikoma daug atvaizdavimo formų, tokių kaip: daiktavardžių ir veiksmažodžių faktų tipo formos, asociacijų skaitymas abiem kryptimis. Loginės formuluotės palaiko du esminius SBVR bruožus. Pirmiausia sujungia bendrų reikšmių aibę su bendruomenėmis. Antra suteikia gaires XMI naudojimui, tokiu būdu įrankiai, palaikantys SBVR gali dalintis konceptais, faktais ir veiklos taisyklėmis [30].
- **Veiklos atvaizdavimas** - Konceptai ir veiklos taisyklės esantys bendrų reikšmių aibėje, turi būti atvaizduoti žodynuose priimtinuose kalbos bendruomenėmis ir jų naudojami reikšmėms pasidalinti. Šie žodynai gali būti įvairioms natūralioms kalboms, dirbtinėms kalboms (tokioms kaip UML) ar specializuotam natūraliųjų kalbų poaibiui (speciali terminija naudojama specialistų, tarkim inžinierių ar teisininkų ir pan.). SBVR sujungia verslo reikšmę su konkrečia kalba žymekliais atvaizduodamas bendrų reikšmių aibės elementus [30].
- **Formali logika** - SBVR turi tvirtą teorinį formalios logikos pagrindą, kuris pagrindžia logines formuluotes bei bendrų reikšmių aibės struktūras. Pagrindas yra pirmos eilės predikatų logika (su keliais uždraustais aukštesnio lygio logikos papildymais) su keletu ribotų modalinės logikos papildymų, tokių kaip deontic (deontinės) formos, reiškiančios pareigą ir draudimą, ir alethic formos būtinybei išreikšti [30].

### 2.3. SBVR formuluotės

Faktų tipas yra asociacija tarp dviejų ar daugiau konceptų, pvz.: „Nuomojama mašina yra laikoma Filiale“ (angl. „*Rental Car is located at Branch*“). SBVR taisyklės yra formuojamos prie faktų tipų pridėdant būtinybę arba išsipareigojimą. Pavyzdžiui, taisyklė „Nuomojama mašina negali turėti daugiau negu tris Papildomus Vairuotojus“ (angl. „*A Rental must not have more than three Additional Drivers*“) yra paremta fakto tipu – „Nuomojama mašina turi Papildomą Vairuotoją“ (angl. „*Rental has Additional Driver*“) [30].

Tokiu būdu SBVR realizuoja esminę Veiklos taisyklių traktuotę (angl. *Business Rules Approach*) verslo lygmenyje, teigiančią, kad „Veiklos taisyklės remiasi faktų tipais, o faktų tipai remiasi konceptais, kurie išreiškiami terminais“ [30].

Viena iš esminių SBVR savybių yra ta, jog konceptai (įskaitant ir faktų tipus) yra nepriklausomi nuo taisyklių. Tai suteikia galimybę SBVR naudoti konceptus (įskaitanti r faktų tipus) verslo žodynų kūrimui. Faktai, turintys tam tikrus faktų tipus, yra esminis SBVR kūrimo elementas. Žemiausio lygio vienetas SBVR – yra loginė formuluotė, paremta fakto tipu ir neturinti loginių operacijų [30].

SBVR nepaiso kaip faktai surenkami – ar jie patvirtinami kaip esminiai faktai ar gaunami išvadų darymo metu. Iš konkretaus faktų modelio išvados gali būti padaromos tik vieną kartą. SBVR neapibrėžia išvadų darymo, kurį būtų galima atlikti, tarp faktų modelių [30].

Taisyklių varikliams, išvadų darymo tvarkos valdymas, yra įprasta funkcija. SBVR apdoroja deklaratyvias taisykles, kurios išreiškiamos iš verslo perspektyvos. Perėjimai nuo faktų modelių ir taisyklių automatizavimas nėra SBVR užmojis [30].

Tikslus SBVR apibrėžimas, naudoja žodyną nurodytą SBVR – kitaip tariant, SBVR standartas nusakomas savo paties terminija [30].

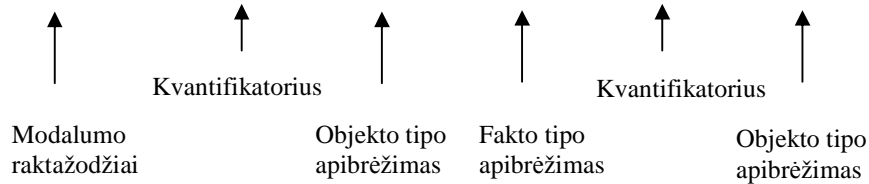
SBVR iniciatyva skirta verslo faktų bei veiklos taisyklių užfiksavimui, kurie gali būti išreikšti ir formaliai ir neformaliai. Veiklos taisyklių išraiškos laikomos formaliomis, jeigu jos yra išreikštos iš anksto apibrėžtoje schemoje tik faktų tipo terminais bei tam tikrais matematiniais/loginiais operatoriais, kvantifikatoriais (kurie pateikiami PRIEDE 1) ir pan. Formalūs taisyklių teiginiai gali būti transformuojami į logines formuluotes, kurias naudoja kita taisyklių pagrindu veikianti programinė įranga. Neformalūs teiginiai gali būti perduodami kaip neinterpretuoti komentarai [30].

Pagrindinių SBVR elementų atvaizdavimui naudojama „Struktūrinė anglų kalba“ (angl. „*Structured English*“) bei įvairūs teksto stiliai:

- terminas (angl. term) – toks teksto stilius skirtas atvaizduoti daiktavardžius, tokius kaip „Paraiška licencijai“ (angl. „Licence Application“).
- Vardas (angl. Name) – toks teksto stilius žymi individualius konceptus tokius kaip klerkas vardu „Jonas“
- veiksmažodis (angl. verb) – toks teksto stilius žymi faktų tipus, kurie nurodo ryšius tarp konceptų. Pavyzdžiui, „klerkas patvirtina paraišką“
- raktažodis (angl. keyword) – toks teksto stilius nurodo įvairius žodžius, skirtus žodyno apibrėžimų formavimui bei taisyklių teiginiams. Raktažodžiai nurodo standartines SBVR koncepcijas, tokias kaip „leidžiama, kad“ (angl. „it is permitted that“) arba „tiksliai vienas“ (angl. „exactly one“) [18].

Pradžioje SBVR apibrėžiami konceptų ir reikšmių žodynai, įvairių elementų sąsajos. Tada formuojamos taisyklės apibrėžiančios sistemos veiklą. Pavyzdžiui:

It is obligatory that each rental car is owned by exactly one branch.



Lietuviškai, ši taisyklė skamba taip:

Privaloma, kad kiekviena nuojama mašina *priklauso* tik vienam filialui.

Matome, kad anglišku taisyklių vertimas į lietuvių kalbą yra sudėtingas, nes dalies formuluočių neįmanoma tiksliai išversti. Labai skiriasi šių kalbų sakinių formulavimo logika. Tačiau SBVR specifikacijoje pabrėžiama, kad:

„SBVR žodynai pateikti specifikacijoje yra paremti anglų kalba, tačiau juos galima naudoti apibrėžiant žodynus įvairiomis kalbomis. Alternatyvūs SBVR žodynai, gali būti sudaromi naudojant kitos kalbos apibrėžimus konceptams, nusakytiems angliškuose SBVR žodynuose.“ [30]

Meta-modelių kūrimui SBVR remiasi Meta Object Family (MOF) principais. T.y. SBVR meta-modelis yra MOF pagrindo meta-modelis, atvaizduojantis SBVR konceptus pagal MOF reprezentaciją. SBVR meta-modelis yra sudarytos iš paketų serijos, kurių sąsajos pateikiamos XML dokumente. SBVR meta-modelio fragmentai ir nuorodos į jį, pateikiamos PRIEDE 2.

SBVR standartas kol kas panaudotas tik dviejuose įrankiuose:

- RuleExpress [34] – tai komercinis SBVR įrankis sukurtas kompanijos RuleArts, skirtas veiklos taisyklėms surinkti ir bendram verslo žodynui sukurti. Tačiau šios taisyklės neautomatizuojamos. Šis įrankis skirtas žinių apie veiklos sferą valdymui
- SBeaVer [35] – atviro kodo SBVR įrankis, sukurtas dviejų studentų iš Lecce universiteto Italijoje. Šis įrankis yra Eclipse „plug-inas“. Deja, instaliavus jį, jis pasimeta tarp visų Eclipse „plug-inų“, o tinklalapyje detalus paaiškinimas kaip tą įrankį išbandyti nepateiktas.

## 2.4. Konceptų ir juos siejančių faktų grafinė atvaizdavimo notacija remiantis SBVR

SBVR standartu sudarytą žodyną galima pavadinti koncepciniu duomenų modeliu. Žodyno elementus galima pavaizduoti grafiškai. Toliau pateikiama SBVR siūloma notacija, bei SBVR išreiškimas UML.

### 2.4.1. SBVR atvaizdavimo notacija

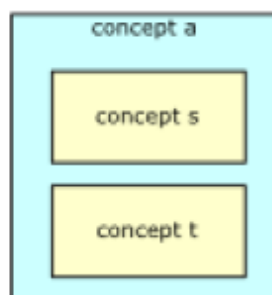
Pagrindiniai žodyno konceptai atvaizduojami stačiakampiu, su jame užrašytu koncepto pavadinimu: (pav. 11) [30]



Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

**Pav. 11. Konceptų grafinis vaizdavimas**

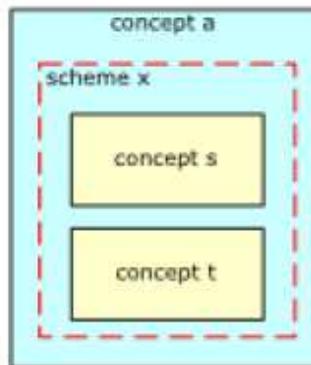
Jeigu turimus konceptus, galima apibrėžti bendresniu konceptu, t.y., jeigu turimi konceptai priklauso kažkokiai konceptų kategorijai, jie vaizduojami „dėžutė dėžutėje“ (pav. 12) [30]



Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

**Pav. 12. Konceptų kategorizavimas. Konceptas s ir konceptas t priklauso koncepto a kategorijai.**

Kartais kategorijos sudaro tam tikras kategorizavimo schemas, kurios žymimos punktyrine linija apibrėžiant konceptus priklausančius konkrečiai schemai (pav. 13) [30]



Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

**Pav. 13. Konceptų kategorizavimas. Konceptas s ir konceptas t yra privilegijuotos koncepto a kategorijos, priklausančios kategorizavimo schemai „scheme x“**

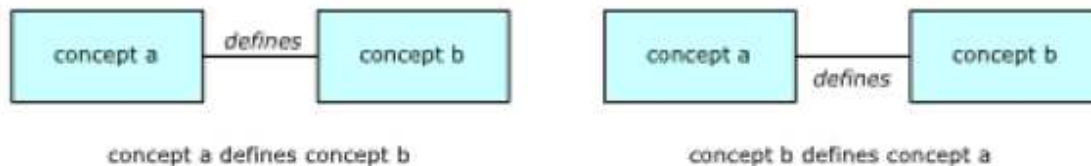
### **Konceptų ryšiai – faktų tipai**

Linija, jungianti du stačiakampius (arba tą patį stačiakampį ratu) nurodo esminių konceptų ryšį. Tokia linija nurodo faktą. Šalia linijos rašomos etiketės su veiksmožodžiais, tokiu būdu faktus esančius SBVR modelyje galima perskaityti kaip paprastus sakinius. Tokie sakiniai perteikia ryšių reikšmę SBVR modelio kontekste, žinoma daugiau paaiškinimų bei terminų apibrėžimus galima rasti pačiame žodyne [30].

Taisyklės, taikomos konceptams, taip pat yra SBVR modelio dalis, tačiau grafiškai jos neatvaizduojamos. Linijos, jungiančios konceptus, gali atvaizduoti paprastus, neapribotus faktus (pavyzdžiui: „daug-su-daug“ ir „laisvai pasirenkamas“ abiem kryptimis). Iš grafinio vaizdavimo galima matyti kai kurias taisykles, tačiau tiksli jų reikšmė nurodoma žodyne [30].

Kad būtų išvengta netvarkos, grafiškai vaizduojama tik po vieną faktų reikšmę. Faktai skaitomi laikrodžio rodyklės kryptimi aplink liniją, t.y. pradžioje skaitomas pirmasis konceptas, tada veiksmožodis (ar veiksmožodžio frazė) tada kitas konceptas (pav. 14), papildoma informacija pateikiama žodyne [30].



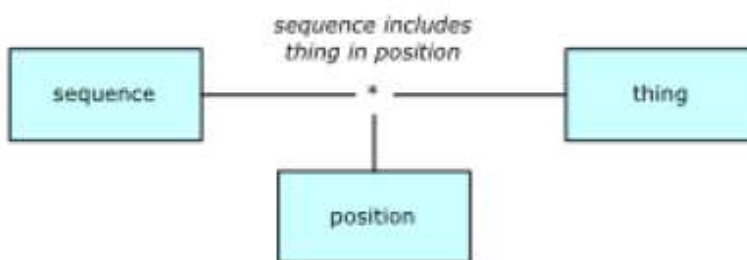


Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

**Pav. 14. Grafiškai išreikštų faktų skaitymas. Pirmuoju atveju konceptas a apibrėžia konceptą b, antruoju atveju konceptas b apibrėžia konceptą a**

### N faktų skaičiaus išreiškimas

Jeigu jungiama daugiau negu du konceptai, faktų tipo neįmanoma atvaizduoti viena linija. Tokiu atveju faktų tipas žymimas žvaigždute („\*“) iš kurios eina linijos į fakte dalyvaujančius konceptus. Fakto reikšmę pateikiama šalia žvaigždutės, o ant linijų nerašomi jokie veiksmažodžiai (pav. 15) [30]



Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

**Pav. 15. N faktų skaičiaus išreiškimas. Išreiškiamas faktas „seka turi daiktą tam tikroje pozicijoje“**

### Faktai susidedantys iš vieno elemento

Faktai, kuriuos sudaro tik vienas konceptas yra žymimi taip pat žvaigždute. Faktas vaizduojamas kaip linija išeinanti iš koncepto stačiakampio, fakto frazė užrašoma šalia jos (pav. 16) [30].



Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

**Pav. 16. Faktas turintis tik vieną konceptą**

### **Rolės**

Role konceptas vadinamas, jeigu jis dalyvauja fakto tipe. Tai atvaizduojama kaip terminas (rolės pavadinimas) esantis šalia stačiakampio, vaizduojančio konceptą, kuris dalyvauja fakto tipe (pav. 17). Kurioje linijos pusėje rašoma rolė nėra didelio skirtumo, tačiau reikia užrašyti taip, kad būtų išvengta painiavos tarp veiksmazodžio ir rolės pavadinimo [30].



Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

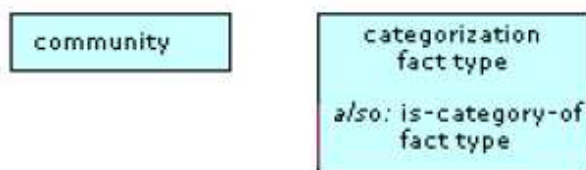
**Pav. 17. Konceptui b suteikiama rolė dalis (angl. *part*)**

### **2.4.2. SBVR atvaizdavimo notacija remiantis UML**

SBVR žodyną galima atvaizduoti naudojant UML, tokio tipo UML modeliai vadinami „Veiklos objekto modeliais“ (Business Object Model, BOM) [30].

#### **Bendras konceptas (Daiktavardžio konceptas)**

Pradinis koncepto terminas, kuris nėra rolė, individualus konceptas, arba faktų tipas nurodomas kaip klasė (stačiakampiu). Stačiakampiui uždedama etiketė su terminu įvardinančiu tą konceptą žodyne. Jeigu yra papildomų terminų apibūdinančių konceptą, jie įrašomi į stačiakampį (pav. 18) [30].

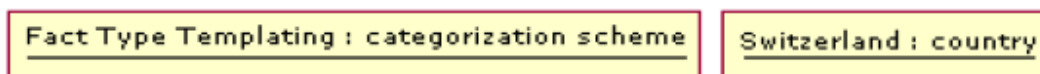


Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

**Pav. 18. Du bendriniai konceptai, iš kurių vienas papildytas žodeliu „taip pat“ (also) ir papildomu terminu**

### Individualus konceptas (Daiktavardinis konceptas)

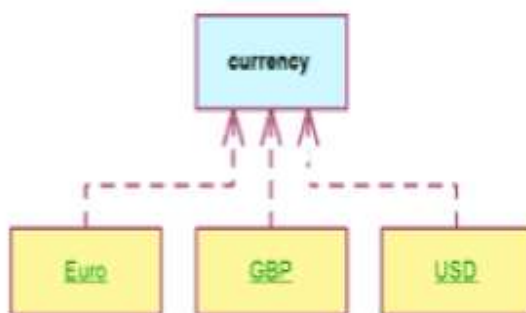
Pavadinimas, kurį turi individualus konceptas, yra rodomas kaip atvejis (stačiakampiu). Po pavadinimo rašomas dvitaškis ir bendrinis konceptas. Visa išraiška būna pabraukta (pav. 19) [30]



Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

**Pav. 19. Individualių konceptų vaizdavimas**

Taip pat individualius konceptus galima vaizduoti kaip bendrinio koncepto atvejus (pav. 20) [30]



Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

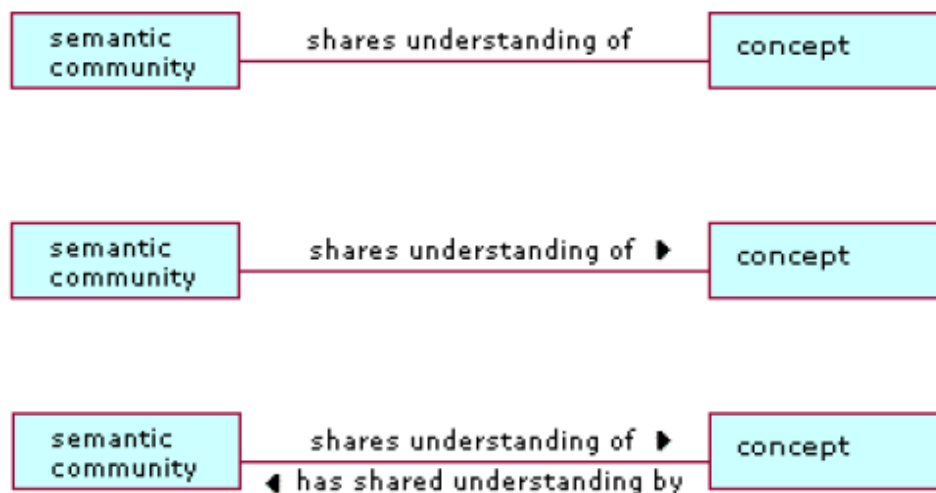
**Pav. 20. Trys individualūs konceptai (euras, svaras ir doleris), kurie yra bendro koncepto (valiuta) atvejai**

## Faktų tipai

UML ryšio tipas „asociacija“ naudojamas norint atvaizduoti ryšius tarp SBVR žodyno konceptų. Tačiau, ryšys tarp faktų nėra asociacijos tipo. Faktų tipas turi atskirą semantiką [30].

SBVR faktų tipai gali būti tokie :

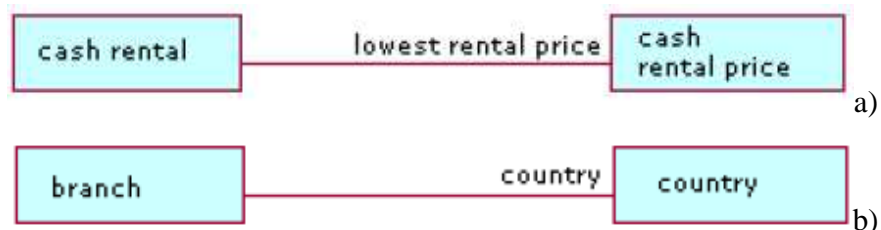
- *Dvejetainiai faktų tipai.* Tokio tipo faktai žymimi kaip asociacija (linija tarp dviejų stačiakampių). Faktas skaitomas pagal laikrodžio rodyklę arba kuria kryptim skaityti faktą nurodo trikampis (pav. 21) [30]



Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR)*, v1.0 OMG Available Specification

**Pav. 21. Trys būdai dvejetainiams faktams atvaizduoti**

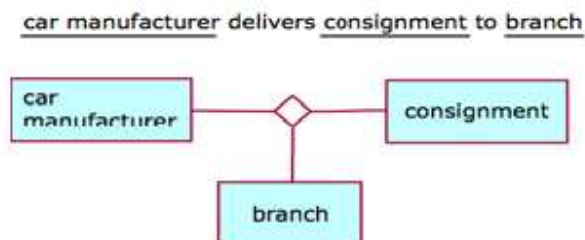
- *Dvejetainiai faktų tipai su žodeliu turi („has“).* Kiekvienai faktų tipo formai, kuri naudoja *turi*, rolės pavadinimas užrašomas ryšio gale. Veiksmažodis *turi* nerodomas diagramoje. Kiekvienas ryšio gale užrašytas pavadinimas nurodo arba priskyrimą, arba faktų tipo rolę (pav. 22. a)). Jeigu dvejetainis faktų tipas neturi specializuotos rolės, ryšio pabaigoje pakartojamas koncepto kuris sudaro rolę vardas (pav. 22. b)) [30].



Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR)*, v1.0 OMG Available Specification

**Pav. 22. Faktų tipai su žodeliu turi**

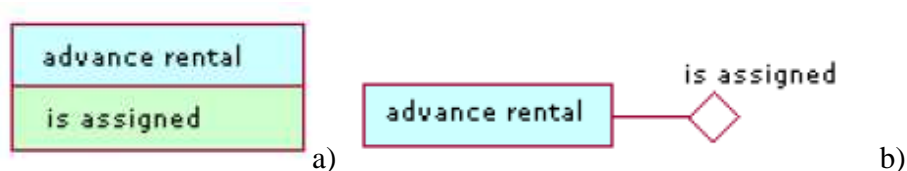
- *Faktų tipai sudaryti iš 3 ar daugiau elementų.* Faktų tipams turintiems daugiau rolių naudojamas toks pats žymėjimas kaip ir asociacija UML. Pradinio faktų tipo forma parodoma su pabrauktais ją sudarančiais konceptais (pav. 23) [30].



Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

**Pav. 23. Faktų sudaryto iš trijų elementų atvaizdavimas**

- *Faktų tipai sudaryti iš vieno elemento.* UML notacijoje asociacijos taikomos tik dvejetainiams ir iš dar daugiau elementų sudarytiems faktams. Faktų tipai sudaryti iš vieno elemento, remiantis UML notacija, paprastai pavirsta UML Boolean atributu (pav. 24. a)). Tačiau SBVR tokie faktai modeliuojami naudojant UML notaciją - faktas vaizduojamas kaip linija išeinanti iš koncepto stačiakampio, fakto frazė užrašoma šalia jos (pav. 24. b)) [30]



Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

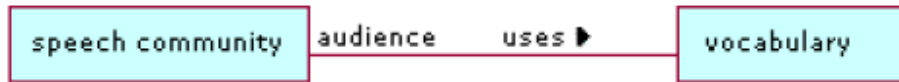
**Pav. 24. Faktų tipų sudarytų iš vieno elemento atvaizdavimas**

## Rolės

Rolės SBVR gali būti vaizduojamos tokiais būdais:

Rolė vaizduojama kaip asociacijos pabaigos pavadinimas. Jeigu rolės pavadinimas toje pačioje klasėje kartojasi, tada siūloma naudoti bendresnį rolės konceptą su nuodytom specifinėm rolėm (pav. 25) [30]

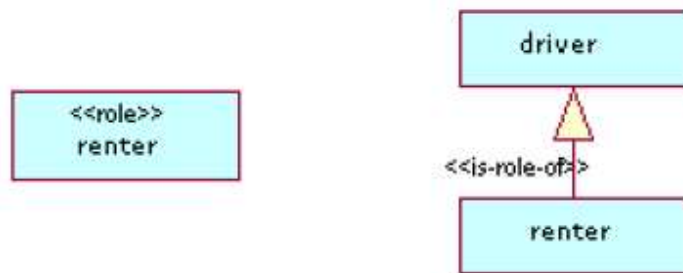
speech community uses vocabulary  
vocabulary has audience



Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

**Pav. 25. Rolės vaizdavimas ryšio pabaigoje.**

Matome, kad tokia grafinė notacija išreiškia du faktus: kalbos bendruomenė naudoja žodyną (angl. *speech community uses vocabulary*) ir žodynas turi vartotojus (angl. *vocabulary has audience*). Rolė vaizduojama naudojant UML stereotipus. Kadangi SBVR rolė taip pat yra konceptas, jis gali būti taip pat vaizduojamas kaip klasė (stačiakampiu), nurodant objekto tipą kurį jis apima (pav. 26) [30].



Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

**Pav. 26. Rolė vaizduojama kaip klasė, naudojant UML stereotipus**

Rolės terminas yra faktų tipo formoje. Jeigu rolės terminas yra faktų tipo formoje ir ta forma nenurodo atributų, tada reikia pavaizduoti rolės terminą. Tokiu atveju rolės negalima vaizduoti ryšio gale, nes tada būtų gaunama forma nurodanti atributus. Vietoje to, rolės terminas yra pabraukiamas ir rodomas kartu su faktų tipo formos veiksmažodžiu (pav. 27) [30]

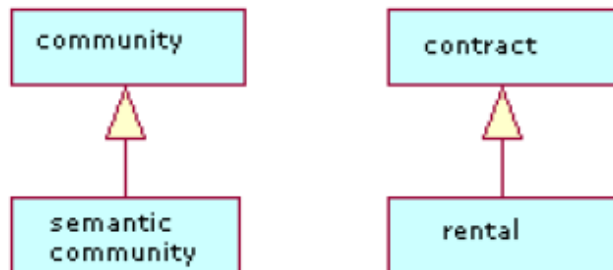


Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

**Pav. 27. Rolės terminas yra faktų tipo formoje.**

## Generalizacija

Generalizacija vaizduojama taip pat kaip UML (pav. 28) [30]

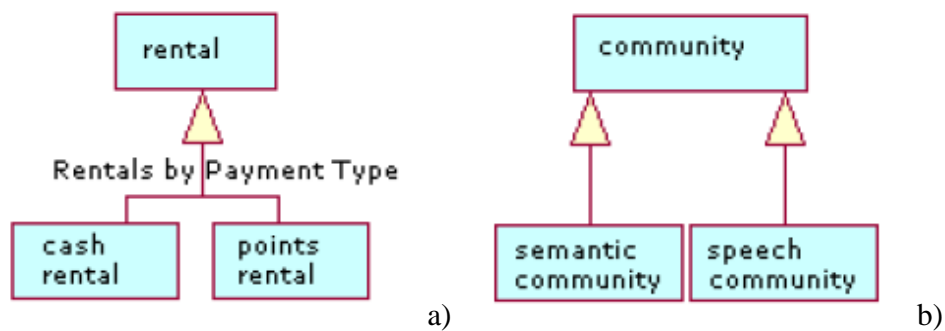


Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

**Pav. 28. Generalizacijos pavyzdžiai**

## Kategorizavimas

Konceptai skirstomi į kategorijas arba sujungiant generalizacijos linijas į vieną (pav. 29. a)) jeigu norime pavaizduoti du bendrai išskiriamus konceptus arba atskiromis linijomis, jeigu konceptai yra atskiri (pav. 29. b)) [30].

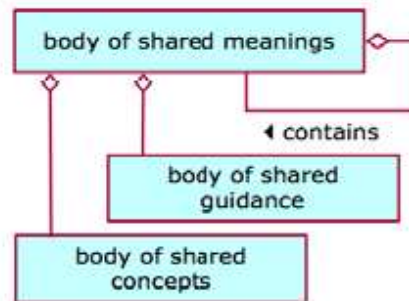


Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

**Pav. 29. Konceptų skirstymas į kategorijas: a) – konceptai kuriuos kažkas sieja, b) nepriklausomi konceptai**

## Dalinio fakto tipas

Dalinio fakto tipas nurodomas naudojant agregaciją (pav. 30) [30]



body of shared meanings *includes* body of shared concepts  
body of shared meanings *includes* body of shared guidance  
body of shared meanings<sub>1</sub> *contains* body of shared meanings<sub>2</sub>

Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR)*, v1.0 OMG Available Specification

**Pav. 30. Dalinių faktų grafinis ir žodinis žymėjimas.**

## 2.5. Konceptų žodyno sudarymas pagal SBVR standartą

Veiklos žodynas turi apibrėžti visą dalykinės srities terminologiją. Pradžioje apibrėžiami visi konceptai, iš kurių yra sudaromi baziniai faktai. Kadangi kuriamas veiklos žodynas yra taikomas konkrečiai auditorijai (tam tikros profesijos žmonėms, kurie susiduria su ta dalykine sritimi, t.y. tam tikrai kalbos bendruomenei), kai kurie terminai yra žinomi iš anksto. Jais apibrėžiama likusi terminologija.

Kaip minėta 2.3 skyriuje, SBVR žodynas ir taisyklės formuojami iš terminų (pvz. žemynas, valstybė), Vardų (pvz. Europa, Lietuva), veiksmažodžių (pvz. sudaro, turi) bei raktažodžių, kuriais išreiškiama taisyklių ir konceptų apibrėžimų logika.

Raktažodžiai gali būti dviejų rūšių – vieni jų naudojami kiekviui išreikšti (1 lentelė), kiti nusako logines operacijas (2 lentelė) tretį nurodo operacijų modalumą (3 lentelė). Visi raktažodžiai yra pateikti anglų kalboje (1 PRIEDAS), tačiau tyrimo metu jie išversti ir į lietuvių kalbą.



Frazės, skirtos loginėms formuluotėms sudaryti:

1 lentelė: Raktažodžiai kiekiui išreikšti

Frazė/raktažodis (Anglų kalba)	Tipas (Anglų kalba)	Frazė/raktažodis (Vertimas)	Tipas (Vertimas)
Each	<u>Universal quantification</u>	Kiekvienas	<u>Universali kiekio išraiška</u>
Some	<u>Existential quantification</u>	Keletas, keli, kelios	<u>Egzistencinė kiekio išraiška</u>
At least one	<u>Existential quantification</u>	Mažiausiai vienas arba Bent vienas	<u>Egzistencinė kiekio išraiška</u>
At least $n$	<u>At-least-ne quantification</u>	Mažiausiai $n$ arba Bent $n$	<u>Bent-<math>n</math> kiekio išraiška</u>
At most one	<u>At-most-one quantificaton</u>	Daugiausiai vienas	<u>Daugiausiai-vienas kiekio išraiška</u>
At most $n$	<u>At-most-n quantification</u>	Daugiausiai $n$	<u>Daugiausiai-<math>n</math> kiekio išraiška</u>
Exactly one	<u>Exactly-one quantification</u>	Tiksliai vienas	<u>Tiksliai-vienas kiekio išraiška</u>
Exactly $n$	<u>Excacly-n quantification</u>	Tiksliai $n$	<u>Tiksliai-<math>n</math> kiekio išraiška</u>
At least $n$ and at most $m$	<u>Numeric range quantification</u>	Mažiausiai $n$ ir daugiausiai $m$	<u>Skaičių intervalo kiekio išraiška</u>
More than one	<u>At-least-n quantification with <math>n = 2</math></u>	Daugiau negu vienas	<u>Mažiausiai-<math>n</math> kiekio išraiška kur <math>n = 2</math></u>

Šaltinis: Sudaryta autoriaus

2 lentelė: Raktažodžiai loginėms operacijoms išreikšti

Frazė/raktažodis (Anglų kalba)	Tipas (Anglų kalba)	Frazė/raktažodis (Vertimas)	Tipas (Vertimas)
It is not the case that $p$	<u>Logical negation</u>	Nėra tas atvejis, kad būtų $p$	<u>Loginis neigimas</u>
$p$ and $q$	<u>Conjunction</u>	$p$ ir $q$	<u>Konjunkcija</u>
$p$ or $q$	<u>Disjunction</u>	$p$ arba $q$	<u>Disjunkcija</u>
$p$ or $q$ but not both	<u>Exclusive disjunction</u>	$p$ arba $q$ , bet ne abu/abi	<u>Disjunkcija skirta tik vienam</u>
If $p$ then $q$	<u>Implication</u>	Jeigu $p$ tada $q$	<u>Itraukimas</u>
$q$ if $p$	<u>Implication</u>	$q$ jeigu $p$	<u>Itraukimas</u>
$p$ if and only if $q$	<u>Equivalence</u>	$p$ jeigu ir tik jeigu $q$	<u>Ekvivalentiškumas</u>
Not both $p$ and $q$	<u>Nand formulation</u>	Ne kartu $p$ ir $q$	<u>Ne-ir formulavimas</u>
Neither $p$ nor $q$	<u>Nor formulation</u>	Nei $p$ nei $q$	<u>Nei formulavimas</u>
$p$ whether or not $q$	<u>Whether-or-not formulaton</u>	$p$ bet kurio atveju $q$	<u>Bet-kuriuo-atveju formulavimas</u>

Šaltinis: Sudaryta autoriaus

3 lentelė: Raktažodžiai modalinėms operacijoms išreikšti

Frazė/raktažodis (Anglų kalba)	Tipas (Anglų kalba)	Frazė/raktažodis (Vertimas)	Tipas (Vertimas)
It is obligatory that $p$	<u>Obligation formulation</u>	Privaloma, kad $p$	<u>Privalumo formulavimas</u>
It is prohibited that $p$	<u>Obligation formulation</u> embedding a <u>logical negation</u>	Draudžiama, kad $p$	<u>Privalumo formulavimas patvirtinantis loginį neigimą</u>
It is necessary that $p$	<u>Necessity formulation</u>	Būtina, kad $p$	<u>Būtinumo formulavimas</u>
It is impossible that $p$	<u>Necessity formulation</u> embedding a <u>logical negation</u>	Neįmanoma, kad $p$	<u>Būtinumo formulavimas patvirtinantis loginį neigimą</u>
It is possible that $p$	<u>Possibility formulation</u>	Įmanoma, kad $p$	<u>Galimybės formulavimas</u>
It is permitted that $p$	<u>Permissibility formulation</u>	Leidžiama, kad $p$	<u>Leidimo formulavimas</u>

Šaltinis: Sudaryta autoriaus

Taip pat raktažodžiais laikomi ir žodeliai, kurie patikslina konceptą: kitas/kita (angl. *another*), duotasis/duotoji (angl. *a given*), kuris/kuri (angl. *that arba who*), sudarytas iš (angl. *is of*), kas (angl. *what*).

Vieni raktažodžiai priklauso alethic logikai, kiti deontic logikai (pav. 31). SBVR standarte, raktažodžiai priklausantys alethic logikai, naudojami sudarant struktūrinės taisyklės, o tie kurie priskiriami deontic logikai, naudojami sudarant operatyvines taisyklės (pav. 32)

<b>Alethic</b>	<b>Deontic</b>
It is necessary that	It is obligatory that
It is possible that	It is permitted that
It is impossible that	It is forbidden that

Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

**Pav. 31. Raktažodžių skirstymas pagal loginę prasmę**

Operative	Structural	Operative	Structural
It is obligatory that	It is necessary that	... must ...	... always ...
It is prohibited that	It is impossible that	... must not ...	... never ...
It is permitted that	It is possible that	... may ...	... sometimes ...

Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR)*, v1.0 OMG Available Specification

### Pav. 32. Raktažodžių skirstymas pagal taisyklių rūšį

Lentelėse pateikti ir lietuviški raktažodžių variantai, kaip matome lietuvių kalba perteikti tikslią jų reikšmę yra įmanoma, tačiau iškyla tam tikrų problemų. Pavyzdžiui, raktažodis: **At least one** verčiamas **Mažiausiai vienas** arba **Bent vienas**, kuri jo reikšmė bus naudojama, priklauso nuo lietuviškos veiklos taisyklės konteksto. Taip yra, todėl, kad lietuvių ir anglų kalbos labai skirtingos. Tačiau raktažodžių pritaikymas lietuvių kalbai yra įmanomas, didesnės problemos kyla sudarant lietuviškus veiklos taisyklių šablonus, tai bus aptarta tolimesniuose skyriuose.

Naudojantis raktažodžiais sudaromas konceptų žodynas. Kiekvienas SBVR žodynas turi antraštes esančias schemoje, pateiktoje pav. 33.

<u>&lt;Vocabulary Name&gt;</u>
Description:
Source:
Speech Community:
Language:
Included Vocabulary:
Note:

Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR)*, v1.0 OMG Available Specification

### Pav. 33. Žodyno sudarymas pagal SBVR

Aprašant žodyną, griežtų apribojimų nėra. Nebūtina užpildyti kiekvieną įrašą, pagrindinis reikalavimas tas, kad vartotojui būtų aiškų koks žodynas sudarytas ir kam jis skirtas. Toliau pateikiamos kiekvieno įrašo reikšmės :

Aprašymas (angl. Description) – aprašoma kokia žodyno paskirtis ir ką jis apima

Šaltinis (angl. Source) – šis įrašas, naudojamas, jeigu žodynas yra paremtas kitu dokumentu . Pavyzdžiui, jeigu žodynas paremtas specialių terminų žodynėliu ar kitu dokumentu, kuris buvo sukurtas ne pagal SBVR standartą, tai tas žodynėlis ar dokumentas nurodomas kaip šaltinis.

Kalbos bendruomenė (angl. Speech Community): kalbos bendruomenė, kuriai priklauso šis žodynas.

Kalba (angl. Language): įvardinama kokia kalba naudojama žodyne. Pagal nutylėjimą naudojama anglų kalba, tačiau galima apibrėžti atvaizdavimus ir į kitą kalbą.

Naudojamas žodynas (angl. Included vocabulary): nurodoma koks žodynas įtrauktas į sudaromą žodyną.

Pastabos (angl. Note): papildoma informacija

Pateikus informaciją apie žodyno paskirtį, aprašomi žodyne esantys konceptai. Kiekvienas žodyno terminas apibrėžia vieną konceptą, kuris aprašomas pagal schemą, pateiktą pav. 34.

#### <primary representation>

Definition:

Source:

Dictionary Basis:

General Concept:

Concept Type:

Necessity:

Possibility:

Reference Scheme:

Note:

Example:

Synonym:

Synonymous Form:

See:

Subject Field:

Namespace URI:

Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

#### **Pav. 34. Konceptų aprašymo schema**

Kaip ir aprašant žodyną, aprašant konceptus griežtų apribojimų nėra. Nebūtina užpildyti kiekvieną įrašą, pateikiama informacija reikalinga, kad vartotojas suprastų, ką terminas reiškia ir kam

jis skirtas, kai kurie konceptai būna aiškūs vartotojui iš bendro žodyno konteksto ir jų atskirai aiškinti nereikia (pvz., jeigu žodynas skirtas GIS ekspertams, jiems nereikia papildomai aiškinti tokių konceptų kaip sausuma, jūra, jie tiesiog naudojami aiškinant kitus konceptus). Toliau pateikiamos kiekvieno įrašo reikšmės:

Apibrėžimas (angl. Definition): Apibrėžimas yra išraiška, kuri logiškai atitinka pradinį atvaizdavimą. Tai nėra sakinyš, todėl neužbaigiamas tašku. Apibrėžimas gali būti pilnai formalus, dalinai formalus arba neformalus.

Šaltinis (angl. Source): Nurodo iš kokio žodyno ar dokumento paimtas konceptas.

Žodyno pagrindas (angl. Dictionary Basis): Pateikiamas paaiškinimas iš paprasto žodyno, kuris paaiškina konceptą. Žodyno pagrindas, netaikomas kaip apibrėžimas.

Bendras konceptas (angl. General Concept): Nurodomas konceptas, kuris apibendrina aprašomą konceptą, pavyzdžiui:

Switzerland

General Concept: country [30]

Koncepto tipas (angl. Concept Type): Nurodomas koncepto tipas.

Būtinybė ir galimybė (angl. Necessity and Possibility): Būtinybė išreiškia, tai kas yra būtinai tiesa. Galimybė išreiškia tai, kas gali būti, jeigu to neuždraudžia koncepto apibrėžimas.

Ryšių schema (angl. Reference scheme): Nurodo kaip dalykai, kuriuos žymi terminas, gali būti gauti vienas iš kitų, remiantis vienu ar daugiau faktų apie juos.

Pastabos (angl. Note): Papildomoms pastaboms, kurios netinka prie kitų antraščių.

Sinonimai (angl. Synonym): Kitas konceptas, kuris reiškia tą patį.

Sinonimo forma (angl. Synonymous Form): Tos pačios faktų tipo formos sinonimas.

Pavyzdžiui:

statement expresses proposition

Definition: the proposition is what is meant by the statement

Synonymous Form: proposition is expressed by statement

[30]

Žiūrėti (angl. See), Temos laukelis (angl. Subject Field) ir Pavadinimo URI (angl. Namespace URI): yra papildoma informacija.

Kartu į konceptų žodyną sukaupiami ir faktai apie konceptus (automobilis turi vairuotoją) bei charakteristikos (teiginiai, kurie būtini norint suprasti konceptą).

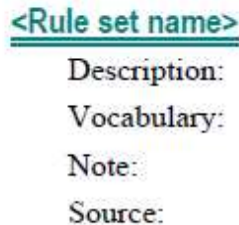
Toks žodynas sudaromas reikalavimų specifikuojimo pradžioje. Paskui, kai pereinama į kitą etapą, kai su vartotoju daugiau nebederinami reikalavimai, tada žodynas pavirsta į klasių modelį. Iš

esmės, sudaromas žodynas yra koncepcinis duomenų bei faktų modelis, juo apibrėžiami dalykinės srities objektai, apie kuriuos bus saugoma informacija, jų charakteristikos bei sąryšiai.

Kadangi siekiama taisyklių sudarymą automatizuoti arba dalinai automatizuoti, sudarytas žodynas negali būti perkrautas. Visi žodyno įrašai turi būti kiek galima tiksliau apibrėžti, vertėtų vengti detalių charakteristikų, kurios sunkiai išreiškiamos duomenų modelyje. Esminiai žodyno elementai įvardinti 3.3 poskyryje esančioje žodyno klasių diagramoje.

## 2.6. Veiklos taisyklių specifikavimas pagal SBVR standartą

SBVR standarte iš taisyklių sudaromi taisyklių rinkiniai. Kaip ir žodynas, pradžioje taisyklių rinkinys aprašomas pagal tokią schemą:



Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

### Pav. 35. Taisyklių rinkinio aprašymo schema

Aprašymas (angl. Description): nurodoma ką taisyklės apima ir kokia jų paskirtis

Žodynas (angl. Vocabulary): nurodoma koks žodynas (apibrėžtas SBVR standarto terminais), yra naudojamas sudarant taisyklių teiginius

Pastabos (angl. Note): papildomi paaiškinimai

Šaltinis (angl. Source): naudojama, jeigu taisyklės paremtos anksčiau apibrėžtu darbu, šiame laukelyje nurodoma nuoroda į jį

Taisyklių rinkinyje įvedami teiginiai (t.y. taisyklės) kurie apibrėžia nagrinėjamą veiklą. Teiginiai sudaromi pagal schemą pateiktą pav. 36.

### <Guidance Statement>

Name:  
Guidance Type:  
Description:  
Source:  
Synonymous Statement:  
Note:  
Example:  
Enforcement Level:

Šaltinis: Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*

#### **Pav. 36. Taisyklių sudarymo schema**

Pavadinimas (angl. Name): nurodomas teiginio pavadinimas, kuris naudojamas žodyne

Vadovavimo tipas (angl. Guidance type): nurodomas teiginio tipas, teiginys gali būti: operatyvinė veiklos taisyklė, struktūrinė veiklos taisyklė, leidimo patarimas (angl. *advice of permission*), galimybės patarimas (angl. *advice of possibility*), išimties patarimas (angl. *advice of optionality*) ir atsitiktinumo patarimas (angl. *advice of contingency*).

Aprašymas (angl. Description): neformali teiginio išraiška

Šaltinis (angl. Source): naudojama, jeigu taisyklė yra iš kažkokio konkretaus šaltinio

Pastaba (angl. Note): papildomi paaiškinimai

Pavyzdys angl. (Example): taisyklės taikymo pavyzdžiai

Reikalavimo lygis (angl. Enforcement level): naudojamas tik operatyvinėms veiklos taisyklėms, nurodo kiek reikalinga tą taisyklę vykdyti

## **2.7. Produkcinių veiklos taisyklių specifikavimas PRR standartu**

Remiantis PRR standartu [31]:

Produkcinė taisyklė yra programavimo logikos teiginys, nusakantis vieno ar daugiau veiksmų seką, jeigu tenkinama tų veiksmų atlikimo sąlyga.

Produkcinių veiklos taisyklių vykdymo poveikis gali priklausyti nuo taisyklių eilės tvarkos, nepriklausomai nuo to ar ta eilės tvarka aprašyta taisyklių vykdymo mechanizmo ar sutvarkyto taisyklių atvaizdavimo [31].

Produkciniė taisyklė paprastai apibrėžiama taip:

If [sąlyga] then [veiksmų sąrašas]

Kartais taikoma ir formuluotė su „else“:

If [sąlyga] then [veiksmų sąrašas] else [alternatyvus veiksmų sąrašas]

Tačiau tokia forma netaikoma PRR. Visos taisyklės turinčios „else“ teiginį, gali būti supaprastintos iki formuluotės be „else“, semantika „else“ veiksmų interpretavimui, gali būti per daug sudėtinga kai kurioms išvadų schemoms. Dėl šių faktų, verčiant PSM į PIM, transformacija kartais galima tik į vieną pusę. Taisyklė su „else“ teiginiais esanti PSM, PIM virsta į kelias iš kurių negalima grįžti į pradines taisykles. Tačiau naujosios taisyklės funkciniu požiūriu yra ekvivalenčios [31].

Produkcinės veiklos taisyklės, saugomos produkcinių taisyklių aibėje. Ši aibė suteikia:

- Priemonės taisyklių surinkimui, kurios susiję su koku nors verslo procesu ar veikla, kaip funkcinis vienetas [31].
- Taisyklių variklyje vykdymo vieneta (unit of execution) su interfeisu skirtu kreiptis į taisykles [31].

Taisyklės esančios taisyklių aibėje, veikia tam tikroje objektų terpėje, vadinamoje duomenų šaltiniu [31].

Taisyklių aibė suteikia tokius objektus:

- Parametrus [31]
- Kontekstą kreipimosi metu [31]

Pakitusios reikšmės, gautos po vykdymo, laikomos kreipimosi į taisyklių aibę rezultatu.

Sąlygos ir veiksmų sąrašai turi išraiškas (sąlygoms naudojamos Boolean), kurios remiasi dviem kintamųjų tipais (jie vadinami standartiniais kintamaisiais ir taisyklių kintamaisiais):

- Standartinis kintamasis turi tipą ir laisvai pasirenkamą pradinę išraišką. Standartiniai kintamieji apibūdinami taisyklių aibės lygyje [31].
- Taisyklių kintamasis turi tipą ir laisvai pasirenkamą domeną (domain), nurodomą duomenų šaltiniui pritaikius filtrą. Jeigu filtro nėra, sritimi laikomi visi duomenų šaltinyje esantys objektai sutinkantys su kintamojo tipu [31].



- Standartiniai kintamieji yra pririšti prie vienos reikšmės (kuri pati gali būti reikšmių rinkinys) esančios jų srityje. Reikšmę gali priskirti pradinė išraiška, arba ji priskiriama vykdant taisykles [31].
- Taisyklių kintamieji yra susieti/asocijuoti su aibe reikšmių esančių jų srityje ir specifikuotų jų tipu bei filtru. Kiekviena reikšmių kombinacija, kuri yra asocijuojama su kiekvienu taisyklės kintamuoju yra tuple, vadinama susiejimu (angl. *binding*). Tai suriša kiekvieną taisyklės kintamąjį su reikšme (objekto arba rinkinio) esančia duomenų šaltinyje. Šios sąsajos yra vykdymo konceptai: jie nemodeliuojami atskirai, o yra gaunami kaip taisyklių susiejimo ir taisyklių apibrėžimo rezultatai [31].

PRR standartas sudarytas iš dviejų dalių – PRR-CORE ir PRR OCL:

- PRR Core (PRR branduolys) yra klasių rinkinys, kurio dėka galima atvaizduoti produkcines veiklos taisykles ir taisyklių aibę, nuo platformos nepriklausančiu būdu (PIM), nenaudojant OCL išraiškų sąlygoms ir veiksams. Tokiu būdu sąlygos ir veiksmai yra „neskaidrūs“ ir laikomi tik „string“ tipo kintamaisiais. Nors tai riboja galimybę transformuoti taisyklės iš vienos veiklos aplinkos (PSM) į kitą, tai suteikia progą dalintis taisyklėmis tarp įrankių, kurie gali apdoroti bazinę produkinių veiklos taisyklių struktūrą [31].
- PRR OCL naudojami OCL meta-modeliu, atvaizduoti taisyklių kintamuosius, sąlygas ir veiksmus. PRR OCL sudaro: klasių rinkinys iš BasicOCL ir EssentialOCL paketų bei specifiniai apribojimai, apibrėžiantys OCL klasių naudojimą PRR OCL kontekste; PRRActionOCL paketas, papildantis BasicOCL paketą klasėmis skirtoms produkinių taisyklių veiksmų daliai atvaizduoti; PRR OCL Standartinė biblioteka, paremta OCL Standartine biblioteka, kuri suteikia apibrėžtus tipus ir operacijas, kurias turi palaikyti visos PRR OCL realizacijos [31].

PRR meta-modelis, parodantis kaip susiejami PRR elementai, pateikiamas PRIEDE 3.

## 2.8. SBVR ir PRR standartų sugretinimas

Išnagrinėjus abu standartus, galima teigti, kad SBVR yra daugiau skirtas pradiniam IS kūrimo etapams, o PRR pačios sistemos automatizavimui. SBVR yra labai platus standartas, suteikiantis galimybę surinkti terminus, veiklos specifiką, įvykių interpretacijas, specializuotą žargoną, faktus ir pan. – t.y. visus esminius verslo elementus. Šie elementai reikalingi kuriant informacines sistemas.

SBVR padeda atrinkti svarbią informaciją ir sukuria jai žodynus. Tų žodynų pagalbą žmonės gali aiškiau išreikšti savo žinias, komunikuoti. Kuriami žodynai ir žodynų žodynai padeda susieti skirtingas kalbos bendruomenes, lengviau integruojami nauji darbuotojai.

Naudojantis PRR, produkcinės veiklos taisyklės išreiškiamos labai tiksliai jas nesudėtinga automatizuoti. SBVR skirtas ne tik produkcinėms taisyklėms, bet ir bendroms taisyklėms apimančioms visą verslo specifiką. Šio standarto specifikacijoje akcentuojama, kad SBVR orientuotas į taisykles, kurios konstatuojamos iš verslo perspektyvos. Taisyklių mechanizavimas automatizuotose sistemose šiuo standartu neapibrėžiamas. Todėl SBVR yra standartas, skirtas MDA CIM atvaizduoti. PRR standartas specifikuoja pačią veiklos logiką, naudodamas sąlygos sakinius. PRR nusakomi veiksmai labai konkretūs, todėl šis standartas skirtas MDA PIM atvaizdavimui.

Detalesnis standartų palyginimas pateikiamas 4 lentelėje:

4 lentelė: SBVR ir PRR standartų sugretinimas

Pavadinimas	SBVR	PRR
Taisyklės apibrėžimas	Vadovavimo elementas, pateikiantis įsipareigojimus arba būtinybę.	Produkcinė taisyklė yra programavimo logikos teiginys, nusakantis vieno ar daugiau veiksmų seką, jeigu tenkinama tų veiksmų sąlyga
Pritaikymas IS sistemoms	Nors veiklos taisyklės yra realios, tai nebūtinai reiškia, kad jos yra automatizuojamos. Daugelis veiklos taisyklių (ypač operatyvinių) yra ne automatizuojamos IS sistemose	Galima vykdyti kompiuteriu. Taisyklė atvaizduoja programavimo logikos sąlygas.
MDA dalis	CIM	PIM
Apibrėžiamos	Semantiniais veiklos taisyklių žodynais	Taisyklių aibe (Ruleset)
Naudojama logika	First-order predicate logic, modal logic, higher logic	Boolean, sąlygos sakiniai
Kaip išreiškiamos taisyklės	SBVR taisyklės išreiškiamos pridant būtinybę arba pareigą faktų tipams. Veiklos taisyklės kuriamos iš faktų tipų, o faktų tipai kuriami iš konceptų išreikštų terminais.	Sąlygos sakiniai: If <sąlyga> then <veiksmų sąrašas>
Pagrindiniai formuluočių elementai	<u>Terminas</u> , <u>vardas</u> , <u>veiksmažodis</u> , raktažodis	Taisyklių aibė (Ruleset), taisyklių kintamieji (standartiniai ir taisyklių), sąlygos, veiksmų sąrašas
Semantika	Semantika yra reikšmės arba ryšiai su vienu simbolių arba jų aibės reikšmėmis. SBVR simboliai gali būti bet kokios formos : žodžiai, frazės, kodai, numeriai, ikonos, garsai ir pan.	Standartinių kintamieji yra pririšti prie vienos reikšmės. Taisyklių kintamieji susieti/asocijuoti su aibe reikšmių esančių jų srityje ir specifikuoti jų tipu bei filtru
Meta-modelis	SBVR meta-modelis yra MOF paremtas meta-modelis, palaikantis SBVR žodynais nusakytų konceptų atvaizdavimą, naudojant MOF.	Sudarytas iš PRR-Core ir PRR OCL klasių kurios padeda atvaizduoti produkcines veiklos taisykles ir taisyklių aibes.
Meta-modelio atvaizdavimas	SBVR meta-modelis yra sudarytos iš paketų serijos, kurių sąsajos pateikiamos XML dokumente	PRR meta-modelį sudaro detali klasių diagrama

Šaltinis: Sudaryta autoriaus

Taigi, kaip matome iš palyginimo, nors SBVR standartas tinka išgaunant veiklos taisykles iš dalykinės srities ekspertų, ne visas šiuo standartu specifikuotas taisykles įmanoma automatizuoti. PRR daugiau tinka veiklos taisyklių programiniam realizavimui, tačiau šiuo standartu užrašomos taisyklės yra išpaustos į griežtus sąlygos sakinius.

## 2.9. SBVR išraiškų transformavimo į PRR tyrimas

SBVR išraiškos nuo PRR turi labai daug skirtumų. PRR formatas yra griežtai apibrėžtas sąlygos sakiniams, o SBVR išraiškos labai įvairios. Ne visada viena SBVR taisyklė išreiškiama viena PRR taisykle. Kartais tenka SBVR formuluotę padalinti į kelias PRR. Pavyzdžiui, taisyklė:

**SBVR:** It is obligatory that rental duration of each rental is at most 90 rental days.

(Vertimas): Privaloma, kad nuomos trukmė kiekvienai nuomojamai mašinai yra ne didesnė negu 90 nuomos dienų.

Norint šią taisyklę pervesti į produkcinių veiklos taisyklių formatą:

For [rule variables] if [condition] then [action list]

Reikia iš SBVR taisyklės išskirti pagrindines produkcinių taisyklių dalis:

[rule variables]: **rental**, **rentalDays**, **result**

[condition]: **rental days** < 90

[action list]: **rent**= true

Kaip matome, pakeičiant formatą, sukonkretinamos ir kai kurios išraiškos ir gaunama tokia produkcinė veiklos taisyklė:

For [rental] if [rentalDays<90] then [result=true]

Kartais, iš SBVR darant produkcines veiklos taisykles, reikia patikslinti kai kuriuos taisyklės teiginius, pavyzdžiui:

**SBVR:** It is obligatory that each driver of a rental is qualified.

(Vertimas): Privaloma, kad kiekvienas nuomojamos mašinos vairuotojas yra kvalifikuotas.

Norint sužinoti ar nuomojamos mašinos vairuotojas yra kvalifikuotas, reikia patikrinti ar jis kvalifikuotas tuo metu, kai nuomojasi mašiną. Vadinasi taisyklėje atsiranda tokie kintamieji:

[rule variables]: **driver**, **rental**, **date/time**, **qualification**, **rentalStart**

[condition]: **driver**=true, **date/time**=**rentalStart**, **qualification**=true

[action list]: **driver**=true

Perrašius SBVR taisyklės PRR formatu, SBVR taisyklė skyla į kelias trumpesnes PRR taisykles:

For [driver] if [date/time=rentalStart] and if [qualification=true] then [driver=true]

For [driver] if [rental=true] then [driver=true]

Svarbu atkreipti dėmesį į tai, kad anglų kalba terminas „rental“ – „nuomojama mašina“ nepakito, t.y. abiejose veiklos taisyklėse naudojama ta pati jo forma. Išvertus į lietuvių kalbą buvo gauti du terminai: „nuomojamai mašinai“ ir „nuomojamos mašinos“, kurie reiškia ta patį, tačiau jų užrašymas skiriasi. Lietuvių kalba turi linksnius, todėl tas pats žodis turi daug formų. Kuriant įrankį, kuris SBVR taisykles transformuotų arba dalinai transformuotų į PRR išraiškas, šis faktas labai aktualus. Naudojant SBVR lietuvių kalbai, geriausia naudoti tik vieną vardininko linksnį. Į tai bus atsižvelgta kuriant programinę šio darbo realizaciją. Planuojama, kad vartotojas galės įvesti sąvoka pasirinktais linksniais, bet šablonas automatiškai ją transformuos į vardininką.

Veiksmažodžiai SBVR kontekste nėra tradiciškai interpretuojami. Jie SBVR paprastai atsako į klausimus „koks yra“, „kam yra“, „kam skirtas“, o ne į „ką veikia“ kaip įprasta. Taisyklėje: „Privaloma, kad kiekvienas nuomojamos mašinos vairuotojas yra kvalifikuotas“ veiksmažodžiu laikoma ne tik žodelis „yra“, bet visas junginys „yra kvalifikuotas“, nusakantis konkretų faktą.

Taisyklės kurios skirtos PRR standartui yra daug glaustesnės ir aiškiai nurodo ką reikia daryti konkrečiu atveju. PRR taisykles, galima analizuoti kaip atskirus elementus, nesigilinant į veiklos sritį.

Pavyzdžiui, taisyklė (nusakanti vieną iš elektroninės parduotuvės veiklos taisyklių) natūraliaja kalba skamba taip:

*If there is at least one garment item in the customer shopping cart and this garment is from a collection then add a hyperlink to the collections page in the customer web page.*

(Vertimas: Jeigu kliento krepšelyje yra bent vienas rūbas ir tas rūbas yra iš kolekcijos, tada į kliento tinklalapį reikia pridėti nuorodą į kolekcijų tinklalapį)

Taisyklėje atsispindi visos produkcinių veiklos taisyklių dalys:

[rule variables]: customer, shopping cart, garment

[condition]: garment > 0

[action list]: hyperlink to collections

Formali šios taisyklės PRR išraiška, užrašyta PRR OCL kalba, atrodytų taip:

**Rule** *atLeastOneGarment*

**ruleVariable:**

?customer: Customer = Customer->any()

?sCart: ShoppingCart = ShoppingCart->any(c:customer|c=?customer)

?garmentItems: Set = ?sCart.items->select(e:items|e.isCollection())

**Condition:**

?garmentItems.size() > 0

**Action:**

```
?customer hyperlinkToCollections = true
```

Pavertus šios taisyklės anglišką formuluotę į SBVR formą, ši taisyklė atrodytų taip:

It is obligatory that a hyperlink to the collections page *is added in the customer web page* if *there is at least one garment in the customer shopping cart* and the garment *is from a collection*.

(Vertimas: Privaloma, kad nuoroda į kolekcijų tinklalapį būtų pridėta kliento tinklalapyje, jeigu kliento krepšelyje yra bent vienas rūbas ir ta rūbas yra iš kolekcijos)

Kaip matome, naudojantis SBVR formuluotėmis galima išreikšti PRR taisyklių aprašymus. Tačiau formuluotė OCL kalba turi kintamąjį „customer“ (klientas), kuris nenusakytas veiklos taisyklėje. Jeigu taisyklė išreiškiama SBVR, tada galima patikrinti kokias sąsajas taisyklėje minimi konceptai, turi su kitais veiklos elementais. Vadinasi naudojantis SBVR, lengviau specifikuoti esminius produkcinį veiklos taisyklių elementus [16].

## **2.10. SBVR ir PRR standartų metodikos praktinis panaudojimo apibendrinimas**

SBVR ir PRR standartus galima laikyti UML meta-modelio plėtiniais, skirtais veiklos taisyklėms atvaizduoti. SBVR standartu galima specifikuoti labai įvairius veiklos aspektus. Juo naudojantis galima analizuoti reikalavimus sistemai, išreikštus natūraliąja kalba, išskirti pagrindinius veiklos konceptus. SBVR žodynais galima sujungti skirtingas kalbos bendruomenes, surinkti ir paaiškinti įvairių žargoną, kurį naudoja tam tikros veiklos sferos atstovai (mechanikai, informatikai, gydytojai ir pan.). PRR standartu detalizuojamos konkrečios taisyklės, išreikštos sąlyga-veiksmas sakiniais [16]. Šias taisykles patogiau automatizuoti.

Teoriškai SBVR būtų galima naudoti sudarant pasirinktos dalykinės srities žodyną, kuriuo būtų galima remtis sudarinėjant PRR taisykles. Tačiau SBVR žodynas sudaromas iš konceptų ir faktų, o PRR išraiškas sudaro kintamieji ir sąlygos. Todėl naudojant pagal SBVR sudarytą žodyną PRR išraiškoms sudarinėti iškiltų problemų dėl skirtingų standartų formatų, nebūtų aišku, kuriuos konceptus naudoti, nes SBVR žodyne jie būtų išreikšti pagal kitokią notaciją negu PRR..

Ištyrus SBVR ir PRR formuluotes, matome, kad SBVR elementais galima suformuluoti PRR taisyklių aprašymus. Tokiu būdu gaunamos ne tik veiklos taisyklės, bet ir jų dalis paaiškinantis žodynas. Taigi galima teigti, kad standartai gali vienas kitą papildyti.

### 3. SBVR IR PRR STANDARTŲ PANAUDOJIMO KURIANTIS METODIKA

Praeitame skyriuje apibrėžti metodikos etapai bei teorinės SBVR transformavimo į PRR išraiškas galimybės. Tačiau norint transformuoti SBVR išraiškas į PRR, SBVR standartas turi būti įdėtas į tam tikrus rėmus.

Siūlomos metodikos pirmame etape yra – Veiklos žodyno ir bazinių veiklos faktų specifikavimas pagal SBVR. Norint valdyti surinktus faktus ir konceptus, reikalinga žodyno klasių diagrama, kuri nusako kaip turėtų būti realizuoti SBVR žodyno valdymo variklis. Antrajame ir trečiajame etape reikia specifikuoti struktūrines ir operatyvines taisykles, SBVR nepateikia griežtų reikalavimų kaip jos turi būti apibrėžtos, todėl reikia sudaryti taisyklių šablonus.

Taigi, siekiant detalizuoti antrajame šio darbo skyriuje apibrėžtą metodikos koncepciją, buvo atliekamas toks tyrimas:

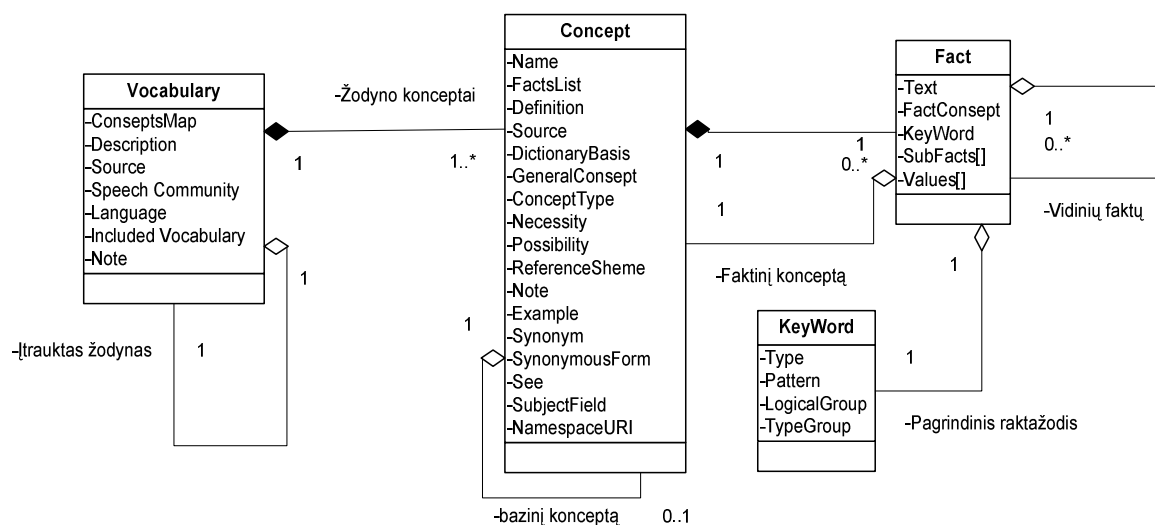
1. Žodyno klasių diagramos koncepcijos apibrėžimas
2. Šablonų struktūrinėms taisyklėms užrašyti sudarymas, remiantis SBVR standartu
3. Šablono operatyvinėms taisyklėms užrašyti, tinkančio ir SBVR ir PRR standartui, sudarymas
4. Veiklos taisyklių, užrašytų pagal šablonus, saugyklos schemos sudarymas
5. Operacinės taisyklės transformavimo į PRR OCL gairių apibrėžimas

Detalizavus metodiką, planuojama atlikti metodikos įvertinimą, pritaikant ją pasirinktai dalykinei sričiai (vertinimo eiga ir rezultatai aptarti šio darbo 4 skyriuje).

#### 3.1. SBVR žodyno klasių modelis

Paveikslėlyje 37 pateikiama veiklos žodyną atitinkanti klasių diagrama. Žodyno architektūra suprojektuota remiantis SBVR standartu ir praplėsta būtiniais atributais. Toliau plačiau aprašomi tik tie atributai ir jų ryšiai, kurie reikalingi siekiant SBVR standarto žodyną realizuoti programiniu būdu.

Klasė *vocabulary* suteikia priėjimą prie žodyno. Šioje klasėje talpinamas visų žodyną sudarančių koncertų žemėlapis. Siekiant išpildyti SBVR standarto sąlygas, žodynas gali turėti papildomą žodyną, kuris yra įtrauktas į pastarąjį, kitaip tariant kiekvienas žodynas gali paveldėti vieną žodyną, tokiu būdu įgyjant visus jo turimus konceptus. Šiame standarte nėra galimybės įtraukti dviejų žodynų vienu metu, t.y. žodynui A įtraukti žodžių iš žodynų B ir C. Šią užduotį reiktų įgyvendinti sekančiu būdu: į C įtraukti B ir tada į A įtraukti B.



Šaltinis: Sudaryta autoriaus

**Pav. 37. Žodyno klasių diagrama**

Kiekvienas žodynas talpina žodžių (konceptų) žemėlapi. Konceptą aprašo klasė *concept*. Kiekvienas konceptas gali turėti bazinį konceptą kuris nusakomas tokio pat tipo objektu. Konceptas turi turėti unikalų vardą visame žodyne. Kiekvienas konceptas gali turėti faktų rinkinį. Konkretus faktų rinkinys galioja tik konkrečiam konceptui. Fakto klasė *fact* apima tokius atributus:

- **Text** – tekstinė viso fakto išraiška
- **FactConcept** – konceptas kuris minimas fakte.
- **Keyword** – Raktažodis pagal kurio struktūrą sudarytas faktas.
- **SubFacts** – jei faktas sudėtinis, tai šis atributas talpina faktus iš kurių yra sudėtas pastarasis faktas.
- **Values** – jei faktas talpina paprasčiausias reikšmes (pvz. skaičius) tai tos reikšmės atitinkama paminėjimo tvarka saugomos šiame attribute.

Kiekvienas raktažodis yra sudėtinis atributas ir jis aprašomas klasėje *Keyword*. Raktažodis turi savo tipą bei šabloną (pvz.: „at least n“). Taip pat yra nurodoma kokiam grupei priklauso faktas pagal loginę prasmę ir pagal taisyklių rūšį.

Suprojektuota architektūra leidžia greitai ir efektyviai surasti norimus konceptus ir su jais susijusią informaciją. Taisyklių vykdymo metu, galima patikrinti ar kiekvienam konceptui yra išpildomos visos būtinos sąlygos ir ar nepažeidžiami kritiniai apribojimai.



### 3.2. SBVR taisyklių šablonų sudarymas ir jų vertimo į lietuvių kalbą įvertinimas

SBVR standarte pateiktos taisyklių aprašymo gairės bei jas sudarantys elementai, tačiau konkrečių šablonų nėra. Norint susieti šį standartą su PRR standartu, struktūrinės bei operatyvinės taisyklės turi būti apibrėžtos labai tiksliai, todėl buvo sudaryti struktūrinių ir operatyvinių taisyklių šablonai, pateikti 5 ir 6 lentelėse.

5 lentelė: Struktūrinių taisyklių šablonai

Angliška šablono versija	Lietuviška šablono versija
<b>It is necessary that/it is impossible that/it is possible that</b> <i>there is</i> <concept1>	<b>Būtina/Neįmanoma/Įmanoma</b> , kad <i>yra</i> <konceptas1>
<b>It is necessary that/it is impossible that/it is possible that</b> <concept1> <verb> <b>at least one/at least n/at most one/at most n/exactly one/exactly n/at least n and at most m/more than one</b> <concept2>	<b>Būtina/Neįmanoma/Įmanoma</b> , kad <konceptas1> <veiksmažodis> <b>bent vieną/bent n/daugiausiai vieną/daugiausiai n/tiksliai vieną/tiksliai n/mažiausiai n ir daugiausiai m</b> <konceptas2>
<b>It is necessary that/it is impossible that/it is possible that</b> <concept1> <i>has</i> <b>at least one/at least n/at most one/at most n/exactly one/exactly n/at least n and at most m/more than one</b> <concept2>	<b>Būtina/Neįmanoma/Įmanoma</b> , kad <konceptas1> <i>turi</i> <b>bent vieną/bent n/daugiausiai vieną/daugiausiai n/tiksliai vieną/tiksliai n/mažiausiai n ir daugiausiai m/</b> <konceptas2>
<b>It is necessary that/it is impossible that/it is possible that</b> <b>each</b> <concept> <i>that belongs to</i> <concept type> <i>has</i> <b>at least one/at least n/at most one/at most n/exactly one/exactly n/at least n and at most m/more than one</b> <concept1>	<b>Būtina/Neįmanoma/Įmanoma</b> , kad kiekvienas <konceptas> <i>priklausantis</i> <konceptų tipas> <i>turi</i> <b>bent vieną/bent n/daugiausiai vieną/daugiausiai n/tiksliai vieną/tiksliai n/mažiausiai n ir daugiausiai m/</b> <konceptas1>
<b>It is necessary that/it is impossible that/it is possible that</b> <i>there are</i> <concept1> or <concept2> <b>but not both</b>	<b>Būtina/Neįmanoma/Įmanoma</b> , kad <i>yra</i> <konceptas1> arba <konceptas2> bet ne abu
If there is <concept1> then <b>it is necessary that/it is impossible that/it is possible that</b> <i>there is</i> <concept2>	Jeigu yra <konceptas1> tada <b>būtina/Neįmanoma/Įmanoma</b> , kad <i>yra</i> <konceptas2>
If there is <concept1> then <b>it is necessary that/it is impossible that/it is possible that</b> <i>there is</i> <b>at least one/at least n/at most one/at most n/exactly one/exactly n/at least n and at most m/more than one</b> <concept2>	Jeigu yra <konceptas1> tada <b>būtina/Neįmanoma/Įmanoma</b> , kad <i>yra</i> <b>bent vieną/bent n/daugiausiai vieną/daugiausiai n/tiksliai vieną/tiksliai n/mažiausiai n ir daugiausiai m/</b> <konceptas2>
<b>It is necessary that/it is impossible that/it is possible that</b> <concept1> <b>consists of</b> <b>at least one/at least n/at most one/at most n/exactly one/exactly n/at least n and at most m/more than one</b> <concept2>	<b>Būtina/Neįmanoma/Įmanoma</b> , kad <i>yra</i> <konceptas1> sudarytas iš <b>bent vieną/bent n/daugiausiai vieną/daugiausiai n/tiksliai vieną/tiksliai n/mažiausiai n ir daugiausiai m/</b> <konceptas2>

Šaltinis: Sudaryta autoriaus

6 lentelė: Operatyvinių taisyklių šablonas

Angliška šablono versija	Lietuviška šablono versija
For < <b>at least one/at least <i>n</i>/at most one/at most <i>n</i>/exactly one/exactly <i>n</i>/at least <i>n</i> and at most <i>m</i>/more than one</b> <concept1>> if <condition> then <b>it is obligatory that/it is prohibited that/it is permitted that</b> <action>	<b>Bent vieną/bent <i>n</i>/daugiausiai vieną/daugiausiai <i>n</i>/tiksliai vieną/tiksliai <i>n</i>/mažiausiai <i>n</i> ir daugiausiai <i>m</i>/</b> <konceptas1> jeigu <sąlyga> tada <b>privaloma, kad/draudžiama, kad/leidžiama, kad</b> <veiksmai>

Šaltinis: Sudaryta autoriaus

Kaip buvo minėta verčiant raktažodžius į lietuvių kalbą – veiklos taisyklių šablonų užrašymas lietuviškai yra sudėtingas. Pirmiausia problemos iškyla su šablonuose esančiais konceptais, nes anglų kalba neturi linksnių, o lietuvių kalba yra linksniuojama. Realizuojant automatinę SBVR taisyklių transformavimą į PRR, automatini transformatorius du koncepto linksnius (pvz. mašina ir mašiniai) gali laikyti visiškai skirtingais konceptais. Taipogi, operatyvinių taisyklių šablone, kuris užrašytas For [variables] if [condition] then [action], žodelis FOR nesiverčia, nes lietuvių kalboje tai nurodoma pakeitus koncepto linksnį. Be to, PRR OCL yra pritaikyta angliškam išraiškom, apskritai visose programinėse kalbose sąlygos sakiniai realizuojami anglų kalba. Todėl taikant operatyvinių taisyklių šabloną, geriausia naudoti anglišką variantą.

Norint pilnai pritaikyti SBVR standartą lietuvių kalbai, reikėtų atsižvelgti į lietuviškų sakinių formulavimo logiką. Išspręsti linksniavimo problemą (vienas iš sprendimo variantų galėtų būti linksnių žodyno sudarymas). Taip pat, turėtų būti sudarytas visų angliškų terminų ir jų atitikmenų lietuvių kalboje žodynas. Žinoma, reikėtų įvertinti ar lietuvišką SBVR variantą galima sieti su PRR standartu. Toks darbas yra kitokios pakraipos, negu aprašomas šiame dokumente. Esamo darbo esmė SBVR ir PRR standartų susiejimas, o SBVR standarto pritaikymą turėtų atlikti lingvistai.

### 3.3. Struktūrinių taisyklių šablonai ir ryšiai

Remiantis 2.4.1 ir 2.4.2 pateikta notacija, grafiškai galima atvaizduoti beveik visus siūlomus struktūrinių taisyklių šablonus. Kiekvienas šablonas atvaizduoja tam tikrą duomenų sąryšį arba apribojimą. Tuo tarpu, operatyvinės taisyklės, nurodo sąlygas ir kokius veiksmus prie jų atlikti.

Vaizduojant grafiškai, raktažodžiai nurodo ryšio kardinalumą:

Raktažodžiai skirti loginėms operacijoms išreikšti:

- **Necessary** – nurodo, kad ryšys yra būtinas (ryšio tipas 1 arba daug arba 1 arba 1),
- **Impossible** – ryšio nėra tarp lentelių, niekaip nesivaizduoja, duomenų modelis nesikeičia (šis raktažodis daugiau tinka apribojimams)

- **Possible** – ryšys galimas, tačiau neprivalomas (ryšio tipas 0 arba daug, 0 arba 1)

Raktažodžiai, skirti kiekiui išreikšti:

- **at least one** – 0 ir daugiau
- **at least n** – n ir daugiau
- **at most one** – 0 arba 1
- **at most n** – nuo 0 iki n
- **exactly one** – 1
- **exactly n** – n
- **at least n and at most m** – mažiausiai n ir daugiausiai m (nusirodo ryšio ribos)
- **more than one** – 2 arba daug

Toliau pateikiamas detalus struktūrinių taisyklių šablonų nagrinėjimas.

Šablonas Nr. 1:

**It is necessary that/it is impossible that/it is possible that** *there is* <concept1>

Jeigu concept1 yra aukšto lygio konceptas, tai grafiškai neatsivaizduoja, tačiau tokią taisyklę galima laikyti tam tikru apribojimu. Jeigu concept1 yra kažkurio kito koncepto charakteristika, tai galima laikyti atributo arba duomenų privalomumu. Toks sąryšis grafiškai nevaizduojamas ir duomenų modelyje neatsispinti, o taip pat yra fiksuojamas kaip tam tikras apribojimas.

Šablonas Nr. 2:

**It is necessary that/it is impossible that/it is possible that** <concept1> <verb> **at least one/at least n/at most one/at most n/exactly one/exactly n/at least n and at most m/more than one** <concept2>

Ir šablonas Nr. 3:

**It is necessary that/it is impossible that/it is possible that** <concept1> *has* **at least one/at least n/at most one/at most n/exactly one/exactly n/at least n and at most m/more than one** <concept2>

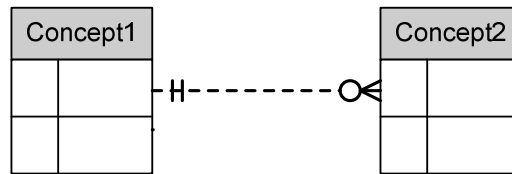
Šie šablonai atvaizduoja ryšį tarp dviejų konceptų. Remiantis SBVR siūloma UML paremta notacija, grafiškai toks ryšys atvaizduojamas taip:



Šaltinis: Sudaryta autoriaus

**Pav. 38. Ryšio tarp faktų grafinė notacija**

Duomenų bazėje, toks ryšys atsivaizduoja, taip:



Šaltinis: Sudaryta autoriaus

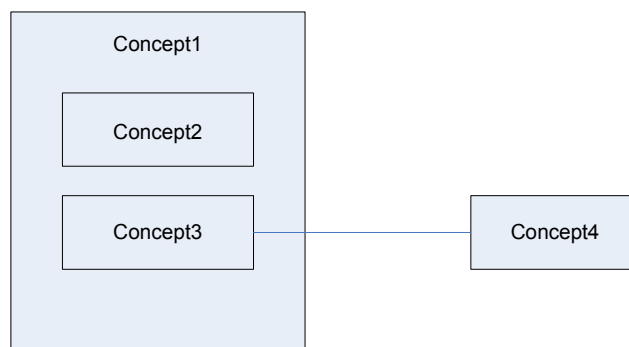
**Pav. 39. Ryšio realizavimas duomenų bazėje**

Šiais dviem šablonais specifikuojamos pagrindinės duomenų bazės lentelės bei jų sąryšiai su atributais.

Šablonas Nr. 4:

**It is necessary that/it is impossible that/it is possible that each <concept> that *belongs to* <concept type> has at least one/at least *n*/at most one/at most *n*/exactly one/exactly *n*/at least *n* and at most *m*/more than one <concept1>**

Šis šablonas nurodo ryšį tarp koncepto potipio ir kokio kito koncepto. Klasių diagramoje toks ryšys neatsivaizduoja, o koncepciniame duomenų modelyje tai vaizduotųsi taip:



Šaltinis: Sudaryta autoriaus

**Pav. 40. Ryšiai tarp koncepto potipių ir kitų konceptų**

Šablonas Nr. 5:

**It is necessary that/it is impossible that/it is possible that *there are* <concept1> or <concept2> but not both**

Šis konceptas nurodo, kad jeigu yra concept1, tai negali būti concept2. Toks ryšys grafiškai nevaizduojamas, veiklos taisyklės sudarytos pagal tokį šabloną, saugomos kaip apribojimai.

Šablonas Nr. 6:

If there is <concept1> then **it is necessary that/it is impossible that/it is possible that** *there is* <concept2>

Tokio ryšio klasių diagramoje nesimato, nes joje nenusirodo, kad jeigu vienas atsiranda tai ir kitas turi atsirasti arba atvirkščiai jo būti negali. Tai ryšys tarp dviejų duomenų struktūros lentelių esybių t.y., papildomi apribojimai, kurie gali būti realizuojami padarius tam tikrus atributus privalomus (pvz., jeigu lentelėje atsirado naujas pirkėjas, tai privaloma nurodyti jo sąskaitą, vadinasi atributas sąskaita yra privalomas).

Šablonas Nr. 7:

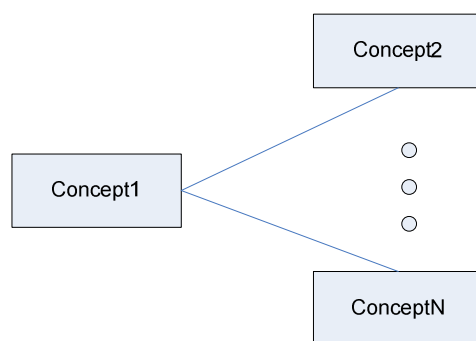
If there is <concept1> then **it is necessary that/it is impossible that/it is possible that** *there is at least one/at least n/at most one/at most n/exactly one/exactly n/at least n and at most m/more than one* <concept2>

Šis konceptas yra labai panašus į prieš tai aptartąjį. Teigiama, jeigu egzistuoja vienas konceptas, tai turi egzistuoti konkretus kiekis kito koncepto egzempliorių (pvz., jeigu atsirado automobilis tai turi egzistuoti bent vienas jo savininkas). Jeigu concept2 yra concept1 atributas, tai reiškia, kad turi egzistuoti konkretus kiekis atributo reikšmių.

Šablonas Nr. 8:

**It is necessary that/it is impossible that/it is possible that** <concept1> consists of **at least one/at least n/at most one/at most n/exactly one/exactly n/at least n and at most m/more than one** <concept2>

Tai „sudarytas iš“ tipo ryšys tarp dviejų esybių – t.y. ryšys vienas su daug. Grafiškai atsivaizduotų taip:

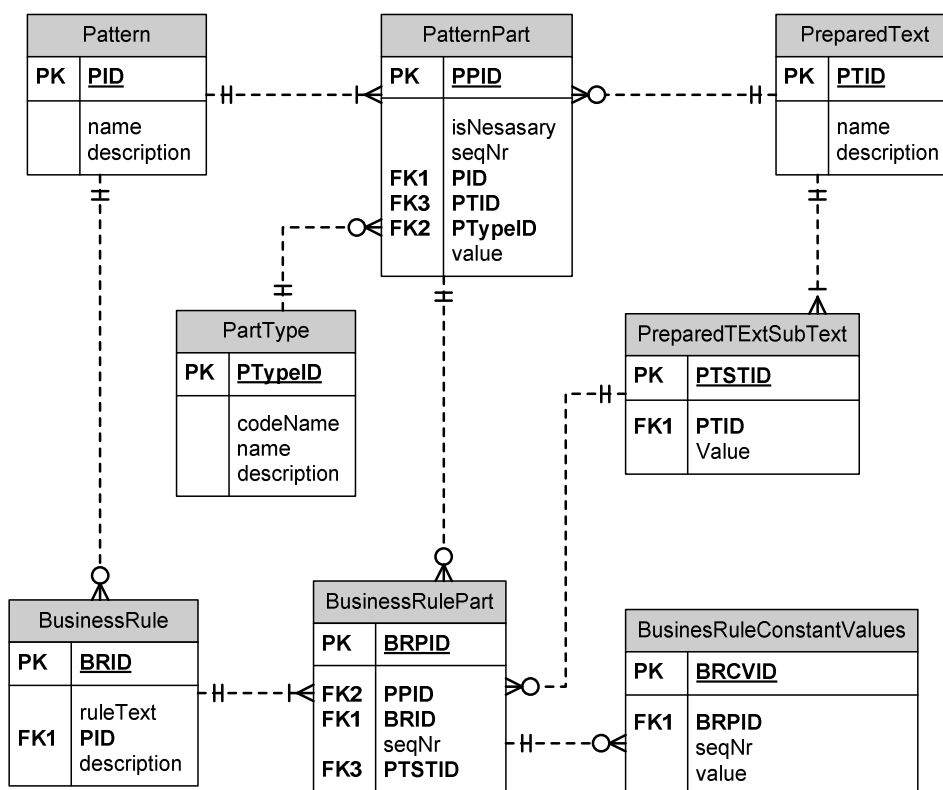


Šaltinis: Sudaryta autoriaus

#### **Pav. 41. Ryšys vienas su daug duomenų modelyje**

Kaip matome, sukuriant klasių diagramą, naudojami šablonai Nr. 2 ir Nr. 3. Likę šablonai aprašo papildomus duomenų sąryšius bei apribojimus, jais aprašytos taisyklės praplečia ir patikslina faktus. Raktažodis **is impossible that** visada suformuoja tam tikrus apribojimus.

Pagal aukščiau aptartus šablonus išreikštoms VT saugoti gali būti kuriamos saugyklos, kurių struktūra atitinka PRR standarte pateiktą meta-modelį. Tai ypač aktualu, jeigu sprendimas integruojamas į UML CASE įrankius. Vis tik kai kuriami autonomiški sprendimai, taisyklės gali būti saugomos labiau specializuotoje saugykloje, kurios struktūra iš esmės suderinama su PRR meta-modeliu. Šio darbo metu ir buvo sukurta tokios saugyklos struktūra, kurios modelis pateiktas 42 pav.



Šaltinis: Sudaryta autoriaus

**Pav. 42. Veiklos taisyklių saugyklos schema**

Šioje saugykloje saugomi SBVR veiklos taisyklių šablonai pateikti aukščiau bei pagal juos aprašytos veiklos taisyklės. Kaip matome, ši saugykla sudaryta iš 8 lentelių, trumpai jas aptarsime (detalus lentelių ir atributų aprašymas pateikiamas PRIEDE 4):

Lentelė – **BusinessRule (Veiklos taisyklė)** – Šioje lentelėje saugomi duomenys apie veiklos taisyklės, t.y. – veiklos taisyklės turinys. Taip pat ši lentelė siejasi su kitomis būtinomis lentelėmis, kuriose saugoma informacija apie veiklos taisyklės interpretavimą.

Lentelė – **BusinessRulePart (Veiklos taisyklės dalis)** – šioje lentelėje saugomi duomenys apie veiklos taisyklių dalis. T.y. saugoma struktūra kaip ir iš ko veiklos taisyklė yra sudaryta.

Lentelė – **BusinessRuleConstantValues (Veiklos taisyklių konstantų reikšmės)** – Paprastai rašant veiklos taisyklės vartotojas įveda savo terminus ir įvairias skaitines bei simbolines reikšmes. Ši lentelė skirta vartotojo įvestiems duomenims saugoti.

Lentelė – **Pattern (Šablonas)** – Saugomi galimi šablonai pagal kuriuos gali būti interpretuojamos veiklos taisyklės.

Lentelė – **PatternPart (Šablono dalis)** – Saugomi duomenys apie veiklos taisyklių šablonų dalis. T.y. saugoma struktūra kaip ir iš ko veiklos taisyklės šablonas yra sudarytas

Lentelė – **PreparedText (Paruoštas tekstas)** – Šioje lentelėje saugoma iš anksto parengtos tekstinių konstantų grupės (pvz.: must/should sudaro viena grupė iš kurios gali būti pavartotas tik vienas iš anksto parengtas žodis toje pat dalyje)

Lentelė – **PartType (Dalis tipas)** – Šioje lentelėje saugomi veiklos taisyklių dalių tipai kurie pagal savo prasmę nurodo ką ta dalis turi daryti. Pvz.: aprašyti konceptą, formuoti sąlygą ir t.t.

Lentelė – **PreparedTextSubText (Paruošto teksto reikšmės)** – Šioje lentelėje saugoma iš anksto parengtos tekstinių konstantų grupės žodžiai

### **3.4. Operacinės taisyklės šablono transformavimas į PRR OCL**

PRR OCL išraiškos yra sudarytos iš trijų elementų – taisyklės kintamųjų (angl. *Rule Variables*), taisyklės sąlygų (angl. *Rule Conditions*) ir taisyklės veiksmų (angl. *Rule Actions*). Remiantis PRR OCL specifikacija:

#### **Taisyklės kintamasis (RuleVariable):**

Taisyklių kintamieji yra naudojami nusakyti rinkiniui (angl. *collection*) ir/arba filtrui kurį atvaizduoja taisyklės kintamasis domene. Taisyklės kintamasis RuleVariable yra sujungtas asociacijos ryšiu su FilterExpression filtro išraiška [32].

#### **Taisyklės sąlyga (RuleCondition):**

Taisyklės sąlyga atvaizduoja Boolean išraiška, kuri sulyginama su esamais duomenimis tam, kad būtų nustatyta ar galima vykdyti taisyklės veiksmus (angl. *actions*) [32].

#### **Taisyklės veiksmai (RuleAction)**

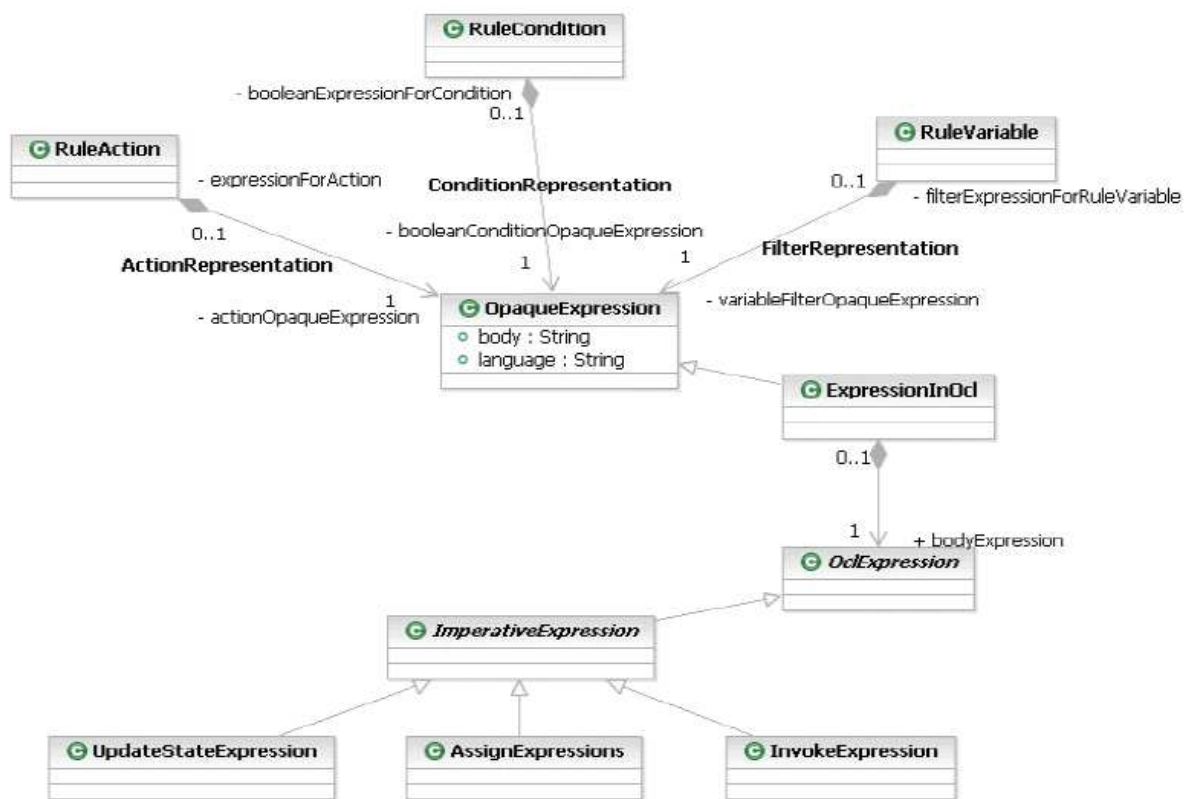
Taisyklės veiksmai apibrėžia sunumeruotą veiksmų sąrašą. Šie veiksmai gali atnaujinti kreipimosi į taisyklių rinkinio (angl. *Ruleset*) domeno objektus arba atlieka kokį nors išorinį kreipimąsi [32].

Galima atlikti tokius veiksmus [32]:

- Atnaujinti būsenos išraišką
- Įterpti (angl. *Assert*): pridėti objektą į veiklos taisyklių variklio sferą
- Atšaukti (angl. *Retract*): išmesti objektą iš veiklos taisyklių variklio sferos
- Atnaujinti (angl. *Update*): pranešti apie objekto pasikeitimą

Šių elementų sąryšius galima matyti PRR OCL meta-modelio fragmente (pav. 43)





Šaltinis: Object Management Group (2007). *Production Rule Representation (PRR), Beta 1.*

**Pav. 43. PRR OCL meta-modelio fragmentas, nurodantis klasių sąryšius**

Taigi, kaip matome PRR standarto išraišką **For** <variables> **if** <condition> **then** <action> galima laikyti tam tikru filtrų rinkiniu. Taigi naudojantis šiuo standartu išrenkami tam tikri duomenys, kurie prie tam tikrų sąlygų kinta.

Operatyvinės taisyklės sudaryto šablono forma yra artima PRR išraiškoms:

For < **at least one/at least n/at most one/at most n/exactly one/exactly n/at least n and at most m/more than one** <concept1>> if <condition> then **it is obligatory that/it is prohibited that/it is permitted that** <action>

Šiame šablone, įrašius konceptą, nurodomi duomenys kuriems bus taisyklė taikoma, o sąlyga nurodo kokius reikalavimus duomenys turi tenkinti. Tokiu būdu, operacinėje taisyklėje, sąlyga ir veiksmas susiejami su tam tikru duomenų šaltiniu.

Norint transformuoti SBVR išraišką užrašytas veiklos taisyklės į PRROCL kodą, rekomenduojame laikytis tokių transformacijos žingsnių. Pasinaudojant žemiau pateiktomis transformacijos gairėmis ateityje bus galima sukurti taikomas programas kurios automatizuotu būdu transformuos SBVR taisyklės pagal sudarytą žodyną į PRR OCL kodą. Toliau pateikiami siūlomi transformacijos žingsniai:

1. Tikrinant taisyklę aptinkame konceptus, kurie aprašyti SBVR žodyne. Turėdami konceptus galime sugeneruoti „*ruleVariable*“ dalį.
2. Aptinkame veiksmą, kuris turi būti atliktas, su konceptu paminėtu „*for*“ dalyje. Veiksmo pavadinimas gali būti pagrindas veiksmą atitinkančio metodo pavadinimo generavimui. Veiksmas gali būti išreikštas arba žodžiu arba matematinio operatoriumi. Jeigu veiksmas išreikštas žodžiu, tada turi būti sugeneruojamas metodo, kuris turi būti iškviestas tam veiksmui atlikti, pavadinimas.
3. Sąlygos „*if*“ dalyje turime turėti konceptą, kurį lygintume su surasta reikšme, esančia sąlygos dešinėje pusėje. Jei aptinkame keletą konceptų vienoje sąlygos pusėje, tai traktuojama, kad antrasis konceptas yra pirmojo atributas. Sąlygą į dvi dalis padalina tokie operatoriai kaip „< > =“ ir pan. Paprastai vienoje pusėje tikimasi rasti konceptus kitoje kažkokią reikšmę, tačiau natūralu, kad konceptai gali būti lyginami su konceptais, todėl jei operatoriaus perskirtiems konceptams galioja atskira hierarchija. Taip pat gali būti užrašomos sudėtinės sąlygos, atskirtos simboliais „&&“ („and“) arba „||“ (arba).
4. „*action*“ dalyje esančią veiksmo sąlygą, išraišką vėlgi galime padalinti į dvi dalis:
  - a. Pirmoji dalis yra kairėje veiksmo sąlygos pusėje. Ji nurodo konceptą kuris pakeičia būseną antroje dalyje esančio koncepto, t.y. šis konceptas turėtų būti paduodamas per parametrus.
  - b. Antroji dalis yra veiksmo sąlygos dešinėje pusėje. Antroje dalyje nurodytas konceptas turi pakeisti savo būseną iškvietus jo metodą kurio pavadinimas sugeneruojamas panaudojant veiksmo sąlygą. Jei kairėje pusėje yra kitas konceptas arba konceptai, tai jie paduodami kaip šio metodo parametrai.

Tarkim, jeigu turime tokią taisyklę, kuri natūraliaja kalba skambėtų taip:

*Jeigu pirkėjas perka bent 3 prekes, kurių kaina 50 LT arba daugiau, reikia taikyti 3% nuolaidą*

Pritaikius šiai taisyklei, siūlomą šabloną gauname:

For <at least one <pirkėjas> if <prekių kiekis>=3 && <kaina >= 50> then <nuolaida = 3>

Verta paminėti, laukelyje <sąlyga> galima įrašyti ne viena, o kelias sąlygas apibrėžiančias kintamuosius. Transformuojant šį šabloną į PRR OCL, pagal prieš tai apibrėžtas gaires – atliekame tokius žingsnius:

1. Nuskaičius taisyklę aptinkami tokie konceptai: pirkėjas, priekių kiekis, kaina, nuolaida. Atliekamas patikrinamas ar šie konceptai apibrėžti konceptų žodyne, jeigu viskas tvarkoje – turim

konceptus atitinkančias duomenų struktūras, transformaciją galima tęsti. Transformacijos metu būtų sugeneruotas toks kodas:

**ruleVariable:**

```
?pirkejas: Pirkejas = Pirkejas->any()  
?krepšelis: Krepšelis = Krepšelis->any(c: customer | c=?customer)
```

2. Aptinkame veiksmus, kurie turi būti atlikti su nurodytais kintamaisiais, jeigu veiksmas būtų išreikštas ne matematine išraiška, o žodžiu, būtų generuojamas tą žodį atitinkantis metodas.

3. Sąlygos „if“ dalyje, aptinkamos dvi sąlygos, atskirtos operatoriumi „&&“, jos sudaro tokį PRR OCL fragmentą:

**Condition**

```
(?krepšelis.items->collect(value))->count() >= 3)  
and  
(?krepšelis.items->collect(value))->sum() >= 50)
```

4. Iš „action“ dalyje nurodytų veiksmų, sugeneruojamas PRR OCL kodas:

**Action**

```
krepšelis.nuolaida = 3
```

Taigi išraiškų transformatorius sugeneruotų tokią taisyklę:

**Rule** atLeast3Prekes

**ruleVariable:**

```
?pirkejas: Pirkejas = Pirkejas->any()  
?krepšelis: Krepšelis = Krepšelis->any(c: customer | c=?customer)
```

**Condition**

```
(?krepšelis.items->collect(value))->count() >= 3)  
and  
(?krepšelis.items->collect(value))->sum() >= 50)
```

**Action**

```
krepšelis.nuolaida = 3
```

### 3.5. SBVR ir PRR panaudojimo metodika: apibendrinimas

Kaip matome, pilnai įgyvendinti pavyko pirmus tris metodikos etapus, o paskutiniai du buvo įgyvendinti tik dalinai. Taip pat buvo apibrėžta veiklos taisyklių saugykla, kurioje turėtų būti saugomos sudarytosios veiklos taisyklės ir aprašytos pagrindinės SBVR išraiškų interpretatoriaus bei PRR kodo generatoriaus kūrimo gairės.

Taikant šiame darbe apibrėžtą metodiką, reikalavimų specifikavimas įgyja vientisą struktūrą. T.y. sudarinėjant įvairias UML diagramas, turime iš anksto griežtai apibrėžtus konceptus, todėl nenutolstama nuo projekto esmės. Surinkti konceptai aiškiai nusako dalykinės srities specifiką, išvengiama perteklinių kintamųjų.

Tradiciniame reikalavimų specifikavimo procese sukurti žodynai atlieka tik pagalbinę – aiškinamąją funkciją. Šiame darbe sudaryta metodika, suteikia konceptų žodynams visai naują prasmę – iš jų sudaromas koncepcinis duomenų modelis (faktų modelis), kuris nurodo kokie yra objektai dalykinėje srityje, apie kuriuos bus saugojama informacija. Tolimesniame IS kūrimo etape, žodynas pavirsta į klasių modelį, kuriame aiškiai matosi kaip jo elementai vienas su kitu susiję.

Taip pat ši metodika siūlo dar vieną naujovę – iš šablonais užrašytų veiklos taisyklių galima sugeneruoti PRR OCL kodą. OCL kalbą gali interpretuoti kitos programavimo kalbos (tarkim „Magic Draw“ OCL išraiškas interpretuoja į JAVA kodą)>. PRR OCL kodo kol kas neinterpretuoja jokie įrankiai, tačiau OCL pavyzdys parodo, kad tai yra įmanoma ir tam yra prielaidos.

Naudojantis šia metodika, veiklos taisykles yra lengviau surinkti iš dalykinės srities ekspertų, be to, jas aprašius iš jų galima sugeneruoti XMI schemas pagal kurias, galima saugoti modelius, sudarytus iš veiklos taisyklių. Tai palengvina visą reikalavimų specifikavimo procesą, nes nereikia papildomai analizuoti sudarytų veiklos taisyklių ir iš jų išgauti tikslius apribojimus, viskas atliekama automatiškai ar pusiau automatiškai.

## 4. METODIKOS PRAKTINIS ĮVERTINIMAS REALIOS DALYKINĖS SRITIES PAGRINDU

Norint išsiaiškinti šiuolaikinių veiklos taisyklių standartų (PRR ir SBVR) naudojimo principus, nuspręsta juos pritaikyti praktiškai. Viena iš veiklos sričių, kurioje kyla problemos dėl veiklos taisyklių sudarymo ir naudojimo, yra geografinės informacinės sistemos arba GIS. GIS – tai informacinė sistema, skirta darbui su erdvine ir aprašomąja informacija. Sistema skirta skaitmeninių koordinuotų erdvėje duomenų kaupimui, saugojimui, vaizdavimui, redagavimui, integravimui bei analizei.

Metodikos praktinio įvertinimo eiga:

1. Konceptų žodyno specifikavimas pagal 2.5. poskyryje pateiktą schemą (žodynas pateiktas PRIEDE 5).
2. Dalykinės srities klasių diagramos sudarymas, remiantis sudarytu konceptu žodynu.
3. Struktūrinių veiklos taisyklių specifikavimas pagal 3.2. skyriuje pateiktus šablonus.
4. SBVR operatyvinių taisyklių transformavimo į PRR OCL imitavimas.
5. Dalykinės srities konceptų bei veiklos taisyklių, kurios specifikuotos SBVR standartu suvedimas į atviro kodo įrankį SBeaVeR bei išbandymas.
6. Apibendrinimas bei galimos realizacijos prototipo pateikimas.

### 4.1. Dalykinės srities klasių diagrama

GIS sistemos yra sparčiai integruojamos daugelio įmonių verslo procesuose. Efektyvus šių sistemų panaudojimas tampa vienu svarbiausių šiuolaikinio verslo uždavinių. Daugelis įmonių yra sukaukę didelius kiekius joms būdingų duomenų, kurie gali būti panaudoti daugelyje specifinių sričių (miestų planavimas, rinkodara, kartografija, kriminologija ir t.t.), todėl šią informaciją reikia greitai ir efektyviai susieti su erdvine informacija, bei pateikti vartotojui aiškioje ir patogioje formoje, t.y. skaitmeniniame žemėlapyje.

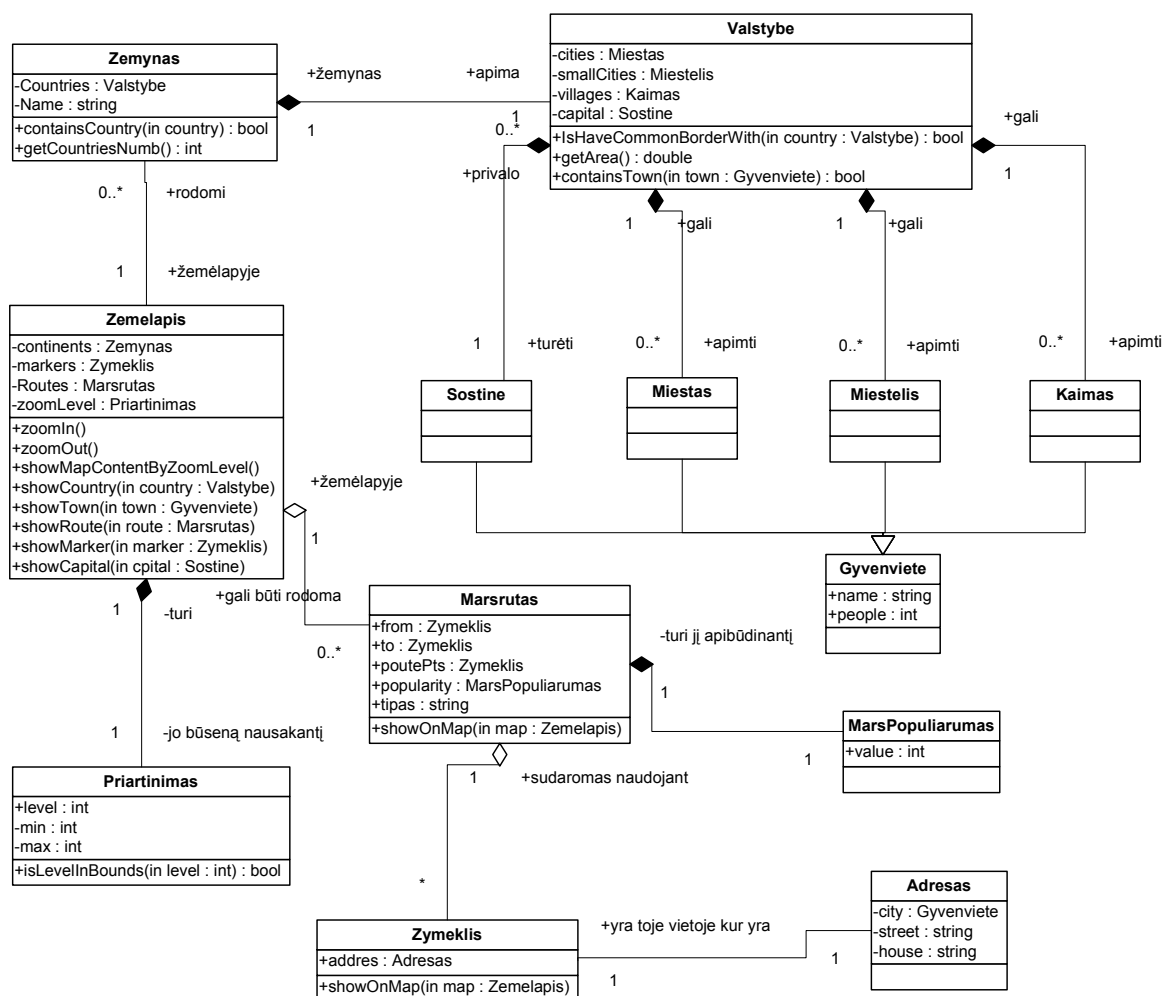
Taigi, dalykinė eksperimento sritis – geografinės informacinės sistemos. GIS sudaro programinę įrangą kuri jungia geografinę informaciją (kur daiktai yra) su aprašomąja informacija (į ką daiktai panašūs). Nuo įprastų popierinių žemėlapių, GIS skiriasi tuo, kad šios sistemos gali turėti daug sluoksnių informacijos. Todėl naudojantis jomis galima matyti verslo struktūrą ar tendencijas verslui aktualesiose srityse.

Erdvinės ir specifinės verslo sukauptos informacijos naudojimas pateikiant rezultatus žemėlapyje šiuo metu nėra dinamiškas ir pakartotinai panaudojamas procesas Norint sukurti optimalią sistemą, reikia tiksliai ir aiškiai apibrėžti konkretaus verslo veiklos taisykles. Veiklos taisyklės

sudaromos bendradarbiaujant informacinių sistemų specialistams bei verslo ekspertams. Tai sudėtingas darbas, kurio metu neišvengiama nesusipratimų, nes verslo ekspertai ne visada tiksliai gali perduoti savo žinias.

Atlikus eksperimentą planuojama patikrinti siūlomą metodiką bei kiek galima susieti SBVR ir PRR standartus specifikuojant veiklos taisykles. Iširti sukurtus šablonus ir jų panaudojimo galimybes.

Pagal 2.5. poskyryje pateiktą žodyno aprašymo schemą, sudaromas dalykinės srities konceptų žodynas (pateiktas PREDE 5). Jame apibrėžti konceptai sudaro klases, kurios bus naudojamos aprašant dalykinę sritį ir iš jų gaunama tokia klasių diagrama, pateikta 44 pav.



Šaltinis: Sudaryta autoriaus

**Pav. 44.** Dalykinės srities klasių diagrama

Paveikslėlyje pateikiama pasirinktos specifinės srities klasių diagrama. Diagramoje pavaizduotos klasės atitinka aprašytus konceptus. Klasėse pateikiamos bazinės funkcijos, kurias galima atlikti su konceptais bei parodomi ryšiai kurie nusako kaip vienas konceptas turi būti sukomponuotas iš kitų konceptų.

#### 4.2. Dalykinės srities konceptų žodyno ir veiklos taisyklių sudarymas

Remiantis metodika, išdėstyta ankstesniame skyrelyje, sudarytas GIS sistemų konceptų žodynas, kuris pateiktas PRIEDE 5. Remiantis sudarytais šablonais, galima apibrėžti dalykinės srities taisykles. Kadangi struktūrinės taisyklės nėra transformuojamos į PRR OCL, jas galima išreikšti ir lietuvių kalba:

Struktūrinės taisyklės:

##### **Taisyklių aibė – Žemėlapis sudarymo taisyklės**

**It is necessary that** *there is* žemėlapis.

(LT: **Būtina, kad yra** žemėlapis.)

Vadovavimo tipas: struktūrinė veiklos taisyklė

**It is necessary that** žemynas *has at least one* valstybe.

(LT: **Būtina, kad žemynas turi bent vieną** valstybe.)

Vadovavimo tipas: struktūrinė veiklos taisyklė

Paremiantys faktai: žemynas turi valstybe

**It is necessary that** valstybe *has only one* sostine.

(LT: **Būtina, kad valstybė turi daugiausiai vieną** sostinę.)

Vadovavimo tipas: struktūrinė veiklos taisyklė

Paremiantys faktai: valstybė turi sostinę

Kaip matome, nors taisyklės ir yra išreikštos pagal šablonus, jos labai artimos natūraliosios kalbos sakiniams. Tačiau kartu jos suformuoja tam tikras sąlygas arba ryšius. Šios taisyklės nurodo būtinus sistemos elementus, t.y. būtinus sistemos klases – žemynas, valstybe, miestas, .

Sukurtos struktūrinės veiklos taisyklės patikslina turimus faktus apie sistemą bei suteikia papildomų apribojimų sistemai. Tarkime, jeigu yra sudaromos tokios veiklos taisyklės:

### Taisyklių aibė – Maršrutų taisyklės

**It is possible that** marsrutas *consists of* **at least one** vietinis marsrutas.

(LT: **Įmanoma, kad** maršrutas sudaro bent vienas vietinis maršrutas )

Vadovavimo tipas: struktūrinė veiklos taisyklė

Aprašymas: Maršrutas turi tipą vietinis maršrutas.

**It is possible that** marsrutas *consists of* **at least one** tarptautinis marsrutas.

(LT: **Įmanoma, kad** maršrutas sudaro bent vienas tarptautinis maršrutas.)

Vadovavimo tipas: struktūrinė veiklos taisyklė

Aprašymas: Maršrutas turi tipą tarptautinis maršrutas.

**IT is necessary that** *there are* vietinis marsrutas or tarptautinis marsrutas **but not both..**

(LT: **Privaloma, kad yra** vietinis maršrutas arba tarptautinis maršrutas **bet ne abu**)

Vadovavimo tipas: struktūrinė veiklos taisyklė

Aprašymas: Maršrutas gali turėti tik vieną potipį.

### 4.3. Operatyvinių SBVR taisyklių transformavimas į PRR

Suformuluotoms veikloms taisyklėms buvo imituojama transformacija iš SBVR į PRR, remiantis sudaryta metodika. Žemiau pateikiami dviejų taisyklių transformavimo pavyzdžiai.

Sudarytos taisyklės nurodo, kad klasė Maršrutas skirstomas į tarptautinius arba vietinius pagal atributą – vietinis maršrutas ir tarptautinis maršrutas. Taigi, jeigu yra sudaroma tokia operatyvinė veiklos taisyklė:

For **<at least one** maršrutas> if **<valstybes sienos**  $\geq 2$ > then **it is obligatory that** **<maršrutas** priklauso potipiui tarptautinis marsrutas>

Naudojantis 3.4 poskyryje apibrėžtomis transformavimo į PRR OCL gairėmis – bus atliekami tokie žingsniai:

1. Tikrinant taisyklę aptinkami tokie konceptai: marsrutas, valstybes sienos, tarptautinis marsrutas. Atliekamas patikrinamas ar šie konceptai apibrėžti konceptų žodyne, jeigu viskas tvarkoje – turim konceptus atitinkančias duomenų struktūras, transformaciją galima tęsti. Transformacijos metu būtų sugeneruotas toks kodas:

`ruleVariable:`



```
?marsrutas: Marsrutas = Marsrutas->any()  
?valstybesSiena: valstybesSiena = valstybesSiena->any(c: maršrutas |  
C=?Marsrutas)
```

2. Aptinkamas veiksmas, kuris yra susietas su veiksmožodžiu “priklauso potipiui” ir išskviečiami atitinkami metodai.

3. Sąlygos if dalyje aptinkama matematinė išraiška: valstybės sienos = 2. Tai atitinka tokį PRR OCL fragmentą:

**Condition**

```
?valstybesSiena.size() >= 2
```

4. Aptinkamas priskyrimo veiksmą, kurį reikia atlikti prie nurodytų sąlygų, nagrinėjamas maršrutas priskiriamas tarptautinių maršrutų tipui. Sugeneruojamas toks PRR OCL kodas:

**Action**

```
tarpt.marsrutas.add(maršrutas)
```

Taigi rezultate, PRR OCL išraiškų transformatorius sugeneruotų tokią PRR OCL taisyklę:

**Rule** TarptMars

**ruleVariable:**

```
?marsrutas: Marsrutas = Marsrutas->any()  
?valstybesSiena: valstybesSiena = valstybesSiena->any(c: maršrutas |  
C=?Marsrutas)
```

**Condition**

```
?valstybesSiena.size() >= 2
```

**Action**

```
tarpt.marsrutas.add(maršrutas)
```

Pateikiame dar vieną operatyvinę veiklos taisyklę:

For <at least one <miestas> if <žemėlapiu priartinimas > 10 > **then it is obligatory that** <miestas rodomas ant žemėlapis>

Ši taisyklė teigia, kad jeigu priartiname žemėlapi daugiau negu per 10 žingsnių, reikia rodyti miestus. Atliekama tokia taisyklės transformacija:

1. Tikrinant taisyklę aptinkami šie konceptai: miestas, priartinimas, žemėlapis. Atliekamas patikrinamas ar šie konceptai apibrėžti konceptų žodyne, jeigu viskas tvarkoje – turim konceptus atitinkančias duomenų struktūras, transformaciją galima tęsti. Transformacijos metu būtų sugeneruotas toks kodas

**ruleVariable:**

```
?miestas1: Miestas->any()  
?zemelapis1: zemelapis();  
?priartinimas1: priartinimas->any(zemelapis1)
```

2. Aptinkamas metodas, kuris susietas su veiksmožodžiu: rodomas.

3. Salygos “if” dalyje aptinkami du konceptai todėl antrasis traktuojamas kaip pirmojo atributas, todėl žinoma, kad reikia palyginti žemėlapio priartinimą. Tai atitinka tokį OCL PRR fragmentą:

**Condition:**

```
(?priartinimas1 >10)
```

4. Aptinkama, kad konceptas žemėlapis yra po veiksmo sąlygos, tai reiškia, kad bus pakeičiama jo būseną pasinaudojant veiksmo pradžioje paminėtu konceptu miestas.

Tai atitinka PRR OCL išraišką, nurodančią, kad žemėlapis turi turėti miesto rodymo metodą:

**Action:**

```
?zemelapis1.showTown(town)
```

Taigi rezultate, PRR OCL išraiškų transformatorius sugeneruotų tokią PRR OCL taisyklę:

**Rule** MiestRod**ruleVariable:**

```
?miestas1: Miestas->any()  
?zemelapis1: zemelapis();  
?priartinimas1: priartinimas->any(zemelapis1)
```

**Condition:**

```
(?priartinimas1 >10)
```

**Action:**

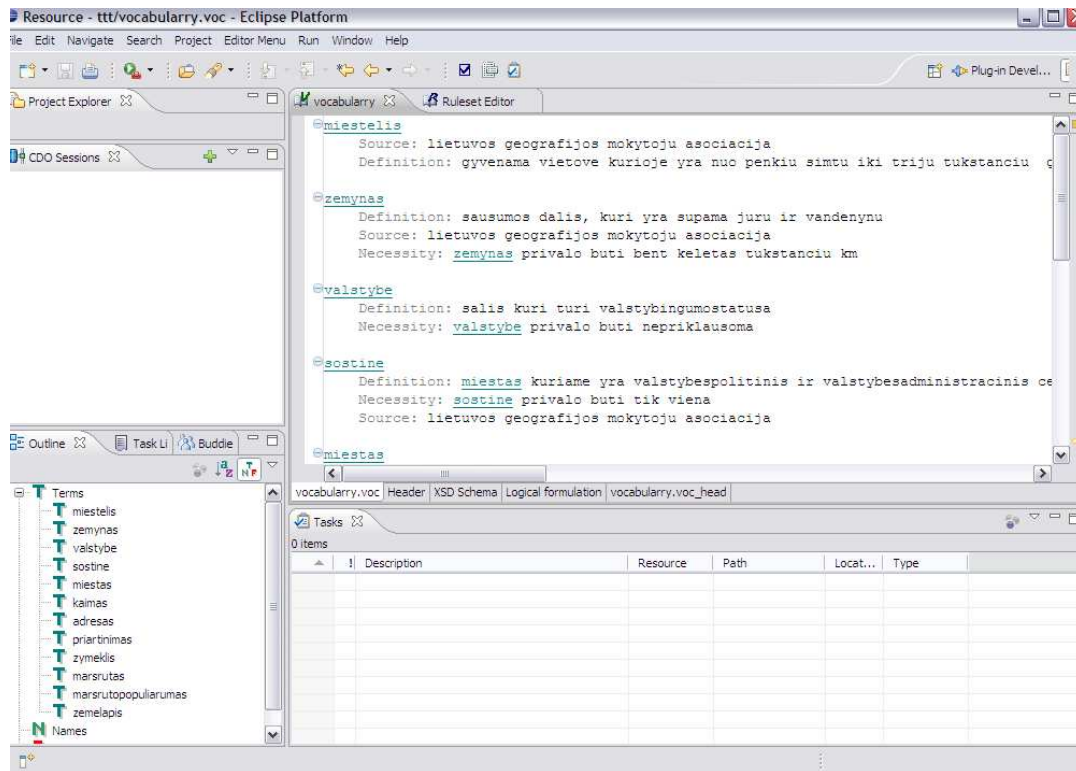
```
?zemelapis1.showTown(town)
```

#### 4.4. Dalykinės srities taisyklių užrašymas naudojant SBeaVeR

Eclipse įskiepis SBeaVeR yra skirtas SBVR žodyno sudarymui ir taisyklių rašymui. Įskiepio pagalba galima apibrėžti konceptus ir jų galimas reikšmes ir vėliau patogiai ir paprastai naudotis šiuo

žodynu rašant veiklos taisykles. Žodyno sudarymo pavyzdys pateikiamas žemiau esančiame paveiksle. Sudarymo principas yra pakankamai aiškus ir intuityvus.

Šiuo įrankiu buvo aprašyti dalykinės srities konceptai:

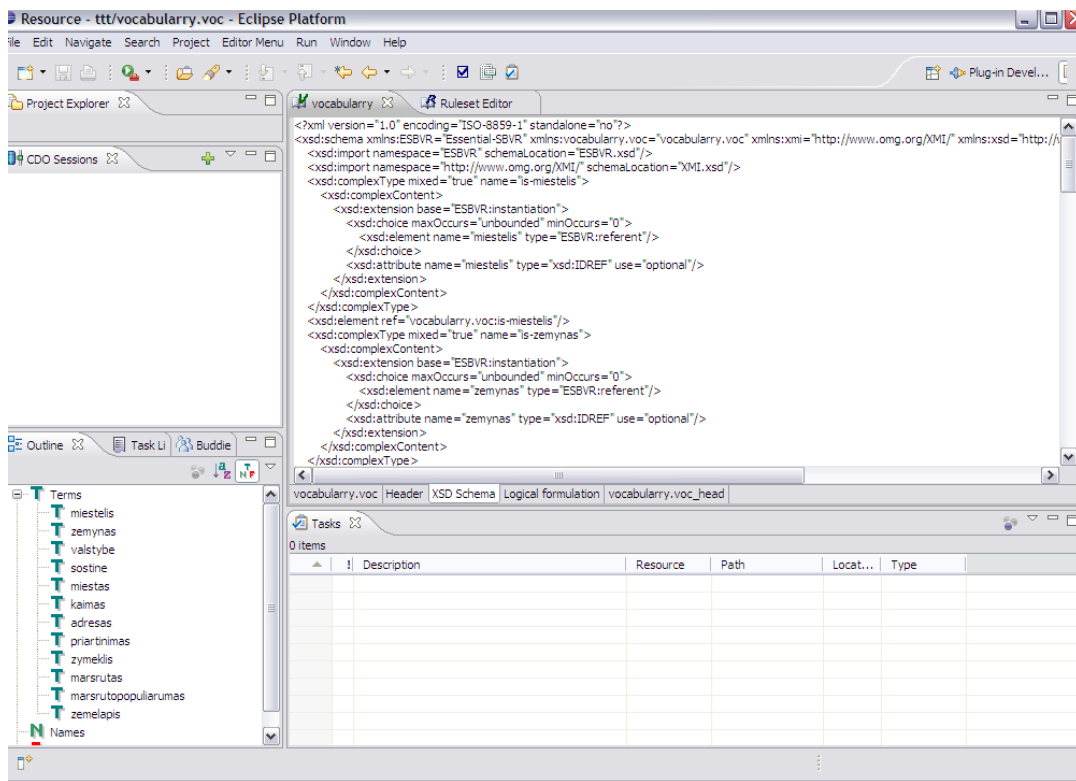


Šaltinis: SBeaVer įrankis

**Pav. 45. SBeaVer žodyno sudarymo langas**

Sudarant žodyną galima naudotis iš anksto parengtais atributų pavadinimais, kuriuos suteikia SBeaVer. Norint tai padaryti reikia paspausti ctrl+space ir tada iš sąrašo galėsime pasirinkti norimą atributą (pvz. *Definition*). Konceptas apibrėžiamas užrašant jį atskiroje eilutėje, o žemiau jo išvardinami papildomi atributai. Jei konceptas jau yra aprašytas, tai SBeaVer automatiškai jį atpažįsta ir pabraukia rašant atributų apibūdinimus.

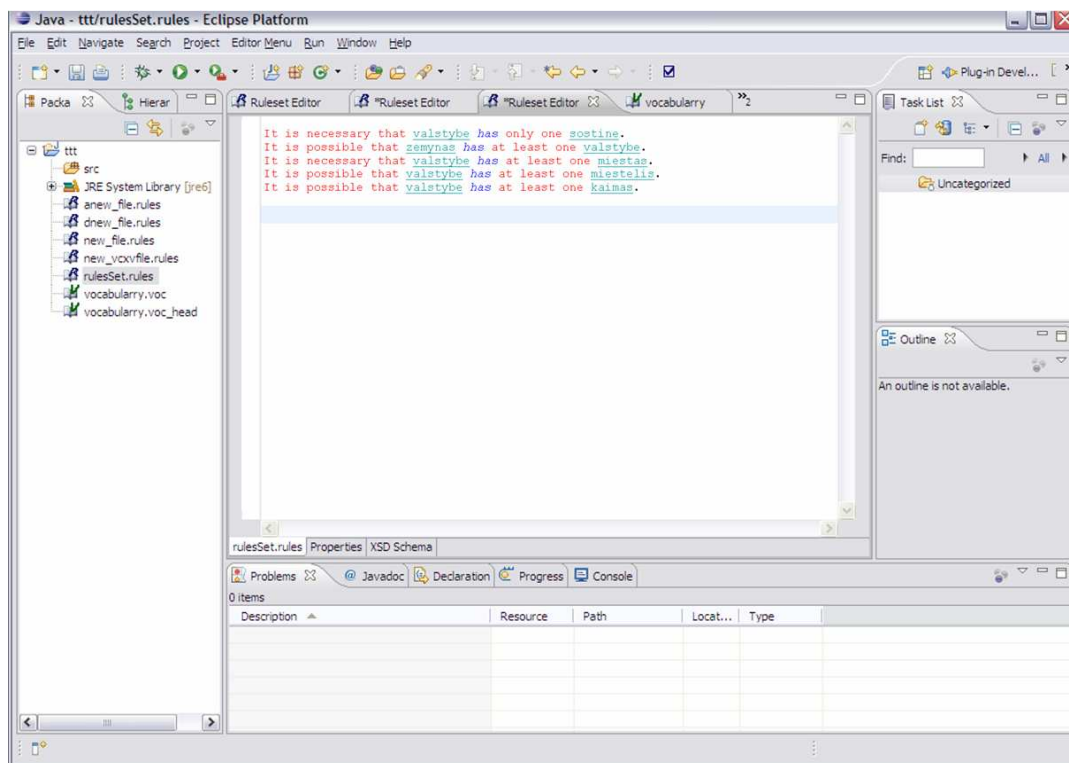
Sudarius žodyną SBeaVer leidžia automatiškai generuoti XSD schemą, sugeneruotos schemas pavyzdys pateikiamas žemiau esančiame paveiksle.



Šaltinis: SBeaVeR įrankis

**Pav. 46. SBeaVeR sugeneruota XSD schema**

Sudarius žodyną galima rašyti veiklos taisykles. Tam SBeaVeR turi specialią aplinką. Rašant taisyklę paspaudus ctrl+space galima gauti iš anksto parengtų žodžių sąrašą kuris atitinka SBVR standartą. Sąrašė taip pat galima rasti visus konceptus ir vardus kuriuos apibrėžėme sudarydami žodyną. Taip rašant veiklos taisykles beveik neįmanoma padaryti sintaksinių klaidų kadangi SBeaVeR mus priverčia rašyti taisykles rėmuose kuriuos patys apibrėžiame ir „nuklydus“ iš karto gauname klaidos pranešimus. Veiklos taisyklių užrašymo pavyzdys pateikiamas žemiau 47 pav.



Saltinis: SBeaVeR įrankis

**Pav. 47. SBeaVeR užrašytos veiklos taisyklės**

#### 4.5. Rekomendacijos metodikos realizacijai

Vienas iš galimų metodikos realizavimo būdų yra esamų sprendimų pritaikymas. Konkrečiu atveju, galima pritaikyti atviro kodo įrankį SBeaVeR. Tačiau norint, kad metodika būtų pilnai realizuota, šį įrankį būtina modifikuoti.

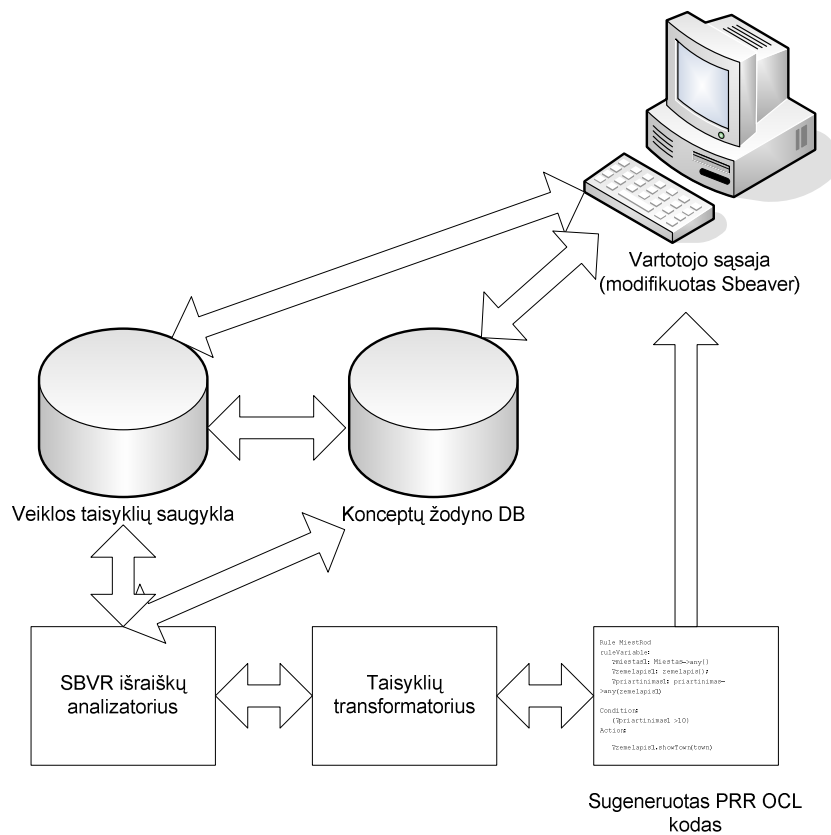
Nors įrankis pasileido ant naujesnės Eclipse versijos, su naujomis Eclipse aplinkomis jis korektiškai neveikia. Reikėtų detaliai peržvelgti jo programinį kodą ir pataisyti klaidas. Atitinkamai modifikavus įrankį, galbūt pavyktų įgyvendinti atvirkštinę inžineriją – t.y. padavus į SBeaVer jau realizuotas klases, įrankis galėtų pasiūlyti veiklos taisykles.

Nors SBeaVeR yra patogus ir vienintelis aptiktas atvirojo kodo įrankis skirtas SBVR veiklos taisyklėms rašyti, jis turi keletą trūkumų kurie priverčia žiūrėti į šį įskiepi kritiškai:

- Įrankis yra senas ir paskutiniai pataisymai buvo atlikti 2006 metais. Nuo to laiko įskiepis buvo tarsi pamirštas ir jokie patobulinimai nebuvo atlikti, todėl rasti informacijos apie naudojimąsi juo yra gan sudėtinga.
- Įrankis nesuderinamas su naujomis Eclipse versijomis, todėl norint jį paleisti, teko parsisiųsti jo kodą bei sukompiliuoti naudojant naujesnę Java versiją (1.6). Perkompiliavimo nepakako ir SBeaVer vis tiek darbo metu meta kritinius pranešimus ir ne visada veikia korektiškai.

- Sudarant terminų žodyną negalima naudoti daugelio simbolių (pvz.: / , skaičių ir t.t. ), tai yra tikrai rimtas nepatogumas kadangi ne visada galima parašyti tai ką norima.
- Žodyno sudarymas yra jautrus didžiosioms ir mažosioms raidėms, apskritai aprašinėjant niekur negalima naudoti didžiųjų raidžių, kadangi žodis prasidedantis didžiąja raide iškarto traktuojamas kaip koncepto pavadinimas, tai yra didelis trūkumas, kuris atima didelę dalį lankstumo.
- Kai padaroma klaida konceptų žodyne įskiepis nenurodo tiksliai kur ir kokia klaida yra padaryta, todėl kai žodynas yra didelis tampa sunku rasti klaidas ir atlikti pataisymus.
- SBeaVer yra atviro kodo Eclipse įskiepis, todėl jo pagrindinis privalumas yra tai, kad dauguma struktūrų realizuojančių SBVR žodyno programinio valdymo variklį jau yra sukurtos. Taip pat jau sukurtas taisyklių valdymo variklis, todėl būtų patogu ir tikslinga šį įrankį praplėsti SBVR transformavimu į PRR OCL išraiškas, naudojantis šiame darbe nurodytomis gairėmis.

Modifikavus SBeaVerį ir sujungus jį su SBVR išraiškų analizatorium ir transformatorium, vartotojas galėtų kurti veiklos taisykles, sudarinėti žodynus ir generuoti PRR OCL kodą. Prototipo koncepcija pateikiama pav. 48.:



Šaltinis: Sudaryta autoriaus

**Pav. 48. Metodikos realizavimo koncepcinė schema**

SBVR ir PRR standartus galima įdiegti į UML CASE įrankius, nes abu standartai ir buvo kuriami su tokia intencija. SBVR standartas yra gana neįprastas, nes visi UML modeliai (išskyrus OCL) yra grafiniai, o SBVR paremtas natūraliosios kalbos šablonais. Tačiau ir šis standartas orientuotas į UML CASE įrankius. Šiame darbe siūloma metodika yra orientuota į taisyklių transformavimą, todėl norint ją realizuoti nėra būtina sukurti CASE įrankio plėtinį. Tam pakanka pasinaudoti atviro kodo įrankiais, kurie yra daugiau specializuoti.

#### 4.6. Metodikos išbandymo dalykinėje srityje įvertinimas

Kaip matome, metodiką pritaikyti dalykinei sričiai pavyko, tačiau verta paminėti keletą pastabų:

- Pagal apibrėžtas gaires, konceptų žodyną sudaryti yra gana patogiu. Nėra būtina užpildyti visas antraštes nurodytas žodyno schemeje. Žodynas gaunasi aiškus ir detalus, dėl to atsiranda teorinė galimybė automatizuoti klasių diagramos sudarymą pagal šį žodyną.
- Sudarant veiklos taisykles pagal šablonus, matome, kad šablonams trūksta lankstumo. Kai kurias taisykles jais nesudėtinga išreikšti, tačiau kai kurios apsirašo sunkiau. Suprojektuota veiklos taisyklių saugykla leidžia saugoti ir patobulintus veiklos taisyklių šablonus – t.y. šablonus galima praplėsti, kad į juos tilptų daugiau konceptų bei raktažodžių.
- SBVR taisyklių aprašymui panaudotas įrankis SBeaVeR. Apibendrinant galima paminėti, kad norint naudoti SBeaVeR būtina ištaisyti klaidas, kurios trukdo dirbti sudarant žodyną bei veiklos taisykles. Taip pat verta pabrėžti, kad jis nėra patogus aprašant didelius žodynus bei sudarant didesnius kiekius veiklos taisyklių dėl anksčiau minėtų priežasčių, todėl yra būtina patobulinti vartotojo sąsajos patogumą siekiant padaryti šį įrankį lankstesniu ir plačiau naudojamu.
- Pagal apibrėžtas išraiškų transformavimo gaires transformuoti taisykles iš užrašytų pagal SBVR šablonus į PRR OCL išraiškas yra įmanoma. Tačiau norint metodiką pilnai realizuoti, reikia praktiškai realizuoti SBVR išraiškų interpretatorių ir PRR OCL generatorių.

Šiuo metu sparčiai populiarėja Agile procesai, siūlantys kuo mažiau dėmesio skirti dokumentavimui. Programinė įranga kuriama trumpomis iteracijomis, o vartotojo poreikiai išsiaiškinami bendraujant tiesiogiai [40]. Veiklos taisyklių koncepcija, kaip ir Agile koncepcija, yra

moderni, tradicinį požiūrį į IS sistemų kūrimą pakeisti bandanti kryptis. Abi koncepcijos remiasi tuo, kad sistemos esmė specifikuojama bendradarbiaujant su dalykinės srities ekspertu be to Agile aktualu tai, kad reikalavimus galima būtų keisti net vėlyvose projektavimo stadijose. Veiklos taisyklių sluoksniu įtraukimas suteikia tokias galimybes.

Toliau pateikiamas siūlomos metodikos palyginimas su AGILE reikalavimų specifikuojimo procesu:

7 lentelė: Metodikos palyginimas

Kriterijus	Siūloma metodika	AGILE reikalavimų specifikuojimas
<b>Vizijos dokumentas</b>	Specifikuojant dalykinę sritį remiantis SBVR notacija, iš karto išskiriami pagrindiniai konceptai ir jų sąryšiai. Jie tolimesniame etape nukeliamas į konceptų žodyną.	Vizijos dokumentas yra vienas svarbiausių reikalavimų dokumentų. Jis padeda užsibrėžti projekto tikslą, ir viso projekto vykdymo metu padeda nenukrypti į šoną. Vizijos dokumente apžvelgiami vartotojo poreikiai bei produkto funkcionalumas, aprašoma dalykinė sritis bei siūlomas sprendimas [40].
<b>Dalykinės srities sąvokų žodynas</b>	Šiame darbe sudaryta metodika, suteikia dalykinės srities sąvokų žodynams visai naują prasmę – iš jo sudaromas koncepcinis duomenų modelis (faktų modelis), kokie yra objektai dalykinėje srityje, apie kuriuos bus saugojama ir vaikišios informacija, kokias jie turi charakteristikas ir tie faktai visi, tai kaip jie susiję.	Reikalavimų daugiareikiškumo padeda išvengti dalykinės srities sąvokų žodynas. Dalykinės srities sąvokų žodyne reikia paaiškinti visus sistemai specifinius žodžius bei jais apibūdinamų sąvokų svarbiausias savybes [40].
<b>Dalykinės srities sąvokų žodyno panaudojimas</b>	Tolimesniame IS kūrimo etape, žodynas pavirsta į klasių modelį, kuriame aiškiai matosi kaip jo elementai vienas su kitu susiję.	Padeda užtikrinti, kad vartotojas bei sistemos kūrėjas naudotų tą pačią terminologiją, išvengti programinės įrangos kūrėjų nesusišnekėjimo su vartotoju bei įsigilinti į dalykinę sritį [40].
<b>Dalykinės srities analizės modelis</b>	Dalykinės srities grafinis modelis išplaukia iš sudarytų veiklos taisyklių bei konceptų žodyno.	Sudaromas dalykinės srities grafinis modelis, aprašytas statinėmis ir dinaminėmis UML diagramomis [40].
<b>Dalykinės srities terminų sąryšių specifikuojimas</b>	Dalykinės srities terminų tarpusavio sąryšiai jau būna pilnai (arba dalinai) specifikuoti sudarant konceptų žodyno.	Modeliuojami dalykinės srities terminai, jų tarpusavio ryšiai bei scenarijai, kaip vykdomos tikslinės veiklos [40].
<b>Funkcinių reikalavimų modelis pagrįstas panaudos atvejais</b>	Funkciniai reikalavimai daugiausiai nusakomi struktūrinėmis veiklos taisyklėmis bei operacinėmis veiklos taisyklėmis.	Aprašomi sistemos funkciniai reikalavimai: kokias funkcijas turi atlikti kuriama sistema, ir kaip šios funkcijos turi būti atliekamos [40].
<b>Nefuncinių reikalavimų specifikuojimas</b>	Šiuos reikalavimus galima specifikuoti taip pat pagal veiklos taisykles arba dalinai. Iš jų galima sudaryti papildomus apribojimus sistemai.	Šalia funkcinių reikalavimų aprašomi ir visi nefunciniai reikalavimai: kokybės atributai (patogumas, patikimumas, greitis, palaikomumas, saugumas), juridiniai bei kontrolės reikalavimai, palaikomos operacinės sistemos, suderinamumas ir kt. [40].

Šaltinis: Sudaryta autoriaus

Palyginus siūlomą metodiką su Agile, galima teigti, kad šiame darbe apibrėžtas reikalavimų specifikuojimo procesas suteikia papildomo funkcionalumo. Bendradarbiaujant su dalykinės srities ekspertais, naudojantis siūloma metodika, yra lengviau išgauti pagrindinius sistemos elementus. Taip



pat dalykinės srities žodynas įgyja visai kitokią paskirtį. Šiame darbe pateikta metodika yra pagrįste skirta veiklos taisyklėms specifikuoti. Todėl apibendrinant galima teigti, kad siūloma metodika, praplėstų Agile galimybes.

## IŠVADOS

1. Šiame darbe nagrinėjamos veiklos taisyklių specifikuojimo problemos, kurios yra labai aktualios kuriant informacines sistemas. Galimybė sistemos kūrimo metu (ir, potencialiai, realizuotoje sistemoje) centralizuotai keisti ir valdyti taisykles sutaupo daug laiko bei padeda IS prisitaikyti prie sudėtingų verslo sąlygų.
2. OMG (*Object Management Group*) – tai tarptautinė standartų organizacija, kurios tikslas sukurti modeliavimo standartus programoms, sistemoms ir verslo procesams. Apžvelgus OMG standartus, galima išskirti, kad specifikuojant veiklos taisykles reikšmingiausi būtų SBVR ir PRR. Šie standartai sąlyginai nauji, todėl detalių metodikų jiems tyrimo metu aptikta nebuvo.
3. SBVR standartas yra labai platus ir ganėtinai neapibrėžtas. Šį standartą patogiau naudoti, jeigu norima nusakyti verslo koncepciją detalia ir aiškiai suprantama kalba. SBVR suteikia šablonus veiklos taisyklių formavimui. Tačiau automatiniam apdorojimui šį standartą labai sunku pritaikyti.
4. PRR standartas yra aiškiai apibrėžtas, turi detalių meta-modelį ir aiškią taisyklių struktūrą. Tačiau specifikuojant veiklos taisykles šiuo standartu, bus užfiksuotos tik produkcinės veiklos taisyklės, t. y., taisyklės, turinčios konkrečias sąlygas ir veiksmų sąrašus. Nustatant sistemos reikalavimus ir pačią sistemos koncepciją, šis standartas nelabai tinka, nes iš esmės yra orientuotas į MDA architektūros PIM lygmenį. Be to, apdorojant reikalavimus, išreikštus natūraliąja kalba, sunku iš karto išskirti sąlygos sakinius, nes sudėtinga aprėpti sistemos konceptų sąsajas, kurios ir sudaro tas sąlygas.
5. Galima teigti, kad SBVR ir PRR standartai vienas kitą papildo. Atliktas tyrimas atskleidė, jog SBVR išraiškas įmanoma užrašyti taip, kad būtų išskirtos produkcinės taisyklių sąlygos, reikalingos PRR formuluotėms sudaryti. Tačiau SBVR standarto taisyklės turi būti įdėtos į tam tikrus rėmus. Detalizavus nagrinėjamą dalykinę sritį SBVR standartu, t.y. išskyrus pagrindinius sistemos konceptus ir jų sąsajas, PRR taisykles galima suformuluoti daug paprasčiau.
6. Pabandžius 2.1. poskyryje pateiktą metodiką pritaikyti praktikoje, pavyko įveikti tris pirmuosius metodikos etapus bei suprojektuoti veiklos taisyklių saugyklą. Paskutiniai du etapai įgyvendinti tik dalinai, t.y., buvo aprašytos pagrindinės SBVR išraiškų interpretatoriaus bei PRR kodo generatoriaus kūrimo gairės.
7. Siūloma metodika suteikia konceptų žodynams visai naują prasmę – iš jo sudaromas koncepcinis duomenų modelis (faktų modelis), atspindintis, kokie yra dalykinės srities objektai, apie kuriuos bus saugojama informacija, kokias jie turi charakteristikas ir kaip jie yra susiję. Taip pat ši

metodika siūlo dar vieną naujovę – iš šablonais užrašytų veiklos taisyklių galima sugeneruoti PRR OCL kodą.

8. Eksperimentinis metodikos pritaikymas taip pat pavyko, tačiau verta paminėti, kad sudarytus šablonus vertėtų patobulinti, tam, kad jais galima būtų aprašyti daugiau veiklos taisyklių. Taip pat, norint sukurti patogią vartotojo sąsają veiklos taisyklių specifikavimui, naudotas įrankis SBeaVeR turi būti pataisytas. Nes išbandant juo metodiką, jame pastebėta labai daug klaidų.
9. Palyginus siūlomą metodiką su esama, galima teigti, kad šiame darbe apibrėžtas reikalavimų specifikavimo procesas suteikia papildomo funkcionalumo. Kadangi ši metodika yra skirta pagrinde veiklos taisyklių specifikavimui, geriausi rezultatai būtų pasiekti, naudojant siūlomą metodiką, kartu su kitomis metodikomis.

## LITERATŪROS SĄRAŠAS

1. Business Rules Group. (2000). *Defining Business Rules ~ What Are They Really?* (3<sup>rd</sup> edition). Iš Business Rules Group interneto puslapio [interaktyvus], [žiūrėta 2008-01-16]. Prieiga per internetą: < [http://www.businessrulesgroup.org/first\\_paper/BRG-whatBR\\_3ed.pdf](http://www.businessrulesgroup.org/first_paper/BRG-whatBR_3ed.pdf) >.
2. Business Rules Group (2007). *The Business Motivation Model. Business Governance in a Volatile World*. Iš Business Rules Group internetinio puslapio [interaktyvus], [žiūrėta 2008-01-18]. Prieiga per internetą: < [http://www.businessrulesgroup.org/second\\_paper/BRG-BMM.pdf](http://www.businessrulesgroup.org/second_paper/BRG-BMM.pdf) >.
3. Business Rules Group. (2003). *Veiklos taisyklių manifestas. Taisyklių nepriklausomumo principai, ver. 2.0.* (Redaktorius Ronald G. Ross). Business Rules Group interneto puslapis [interaktyvus], [žiūrėta 2007-01-16]. Prieiga per internetą: < [http://www.businessrulesgroup.org/brmanifesto/BRManifestLithuanian\(v1.0\).pdf](http://www.businessrulesgroup.org/brmanifesto/BRManifestLithuanian(v1.0).pdf) >.
4. Butleris R. (2006). *Funkcinių reikalavimų specifikuojimas abstrahuoto ir detalizuoto reikalavimų modelių pagrindu*. Informacijos Mokslai, 168 – 177 psl. ISSN 1392-0561.
5. Butleris, R., Kapočius, K. (2001). *Struktūrizuotų veiklos taisyklių saugyklos architektūra*. Informacijos mokslai. Vilnius. Vilniaus universiteto leidykla, 17 tomas, 46-57.
6. Butleris R., Motiejūnas L. (2006) *Veiklos taisyklių saugykla, integruota su veiklos taisyklių išskvietimo mechanizmu*. Konferencijos pranešimų medžiaga. Kauno Technologijos Universitetas, Kaunas, Lietuva. Duomenų bazės ir modeliai, 2 Sekcija, 11 – 20 psl.
7. Editors of BRCommunity.com (2006). *A Brief History of the Business Rule Approach, 2nd ed.* Iš Business Rules Community internetinio puslapio [interaktyvus], [žiūrėta 2008-01-21]. Prieiga per internetą: < <http://www.brcommunity.com/a2006/b317.html> >.
8. Enterprice Decision Management (2007). *Production Rule Representation at OMG – a sneak peak*. Iš Enterprice Decision Management interneto puslapio [interaktyvus], [žiūrėta 2008-01-10]. Prieiga per internetą: < [http://www.edmblog.com/weblog/2007/06/production\\_rule.html](http://www.edmblog.com/weblog/2007/06/production_rule.html) >.
9. Fishman N. (2005). *Modeling Business Rules With The Object Constraint Language*. Iš Business Rules Community interneto puslapio [interaktyvus], [žiūrėta 2008-01-18]. Prieiga per internetą < <http://www.brcommunity.com/b021.php> >.
10. Haggerty N. (2005). *Modeling Business Rules Using the UML and CASE*. Iš Business Rules Community internetinio puslapio [interaktyvus], [žiūrėta 2008-01-18]. Prieiga per internetą: < <http://www.brcommunity.com/b016.php> >.

11. Hendryx S. (2005). *Are System Requirements Business Rules?* Iš Business Rules Community interneto puslapio [interaktyvus], [žiūrėta 2008-01-17]. Prieiga per internetą: < <http://www.BRCommunity.com/a2005/b249.html> >.
12. Hendryx S. (2004). *Initial OMG Business Rules Proposals Evolve.* Iš Business Rules Community interneto puslapio [interaktyvus], [žiūrėta 2008-01-17]. Prieiga per internetą: < <http://www.BRCommunity.com/a2004/b193.html> >.
13. Hendryx S. (2005). *OMG Business Modeling Pipeline.* Iš Business Rules Community internetinio puslapio [interaktyvus], [žiūrėta 2008-01-17]. Prieiga per internetą: < <http://www.BRCommunity.com/a2005/b219.html> >.
14. Hendryx S. (2003). *OMG Issues Business Rules RFPs.* Iš Business Rules Community interneto puslapio [interaktyvus], [žiūrėta 2008-01-17]. Prieiga per internetą: < <http://www.BRCommunity.com/a2003/b173.html> >.
15. Hendryx S. (2005). *SBVR and MDA: Architecture.* Iš Business Rules Community interneto puslapio [interaktyvus], [žiūrėta 2008-01-17]. Prieiga per internetą: < <http://www.BRCommunity.com/a2005/b258.html> >.
16. Ivančikaitė, G; Garšva, G, Kapočius, K. (2009) *Šiuolaikinių veiklos taisyklių standartų taikymas kuriant informacines sistemas.* 14-osios tarpuniversitetinės magistrantų ir doktorantų mokslinės konferencijos „Informacinės technologijos“ pranešimų medžiaga. Vilnius: UAB Biznio mašinų kompanija, 244 psl.
17. Kollber A. (2005). *Business Rule Motivation.* Iš Business Rules Community internetinio puslapio [interaktyvus], [žiūrėta 2008-01-19]. Prieiga per internetą: < <http://www.brcommunity.com/b041.php> >.
18. Linehan, M. H. (2006). *Semantics in Model-Driven Business Design.* Iš 2<sup>nd</sup> International Semantic Web Policy Workshop internetinio puslapio [interaktyvus], [žiūrėta 2009 01 19]. Prieiga per internetą: < [http://www.l3s.de/~olmedilla/events/2006/SWPW06/programme/paper\\_02.pdf](http://www.l3s.de/~olmedilla/events/2006/SWPW06/programme/paper_02.pdf) >.
19. Miers, D. (2007). *Overview of OMG BPM Standards.* Iš BPM Focus internetinio puslapio [interaktyvus], [žiūrėta 2008-06-18]. Prieiga per internetą: < <http://bpmfocus.pbwiki.com/General+Sessions> >.
20. Montilva J., Barnios J. (2004). *BMM: A Business Modeling Method For Information Systems Development.* CLEI Electronic Journal, Vol. 7, No. 2, Paper 3 [interaktyvus], [žiūrėta 2008-01-20]. Prieiga per internetą: < <http://www.clei.cl/cleiej/papers/v7i2p3.pdf> >.

21. Nijssen S. (2007). *SBVR: Semantics for Business*. Iš Business Rules Community interneto puslapio [interaktyvus], [žiūrėta 2008-01-16]. Prieiga per internetą < <http://www.BRCCommunity.com/a2007/b367.html> >.
22. Object Management Group (2007). *OMG Advances Standards for Finance, Government and Business Modeling at Technical Meeting in Burlingame, CA*. Iš Object Management Group internetinio puslapio [interaktyvus], [žiūrėta 2008-01-17]. Prieiga per internetą: < <http://www.omg.org/news/releases/pr2007/12-20-07.htm> >.
23. Object Management Group (2007). *Business Motivation Model (BMM) Specification*. Iš Object Management Group internetinio puslapio [interaktyvus], [žiūrėta 2008-01-18]. Prieiga per internetą: < <http://www.omg.org/docs/dtc/07-08-03.pdf> >.
24. Object Management Group (2007). *Business Process Definition MetaModel (BPDM), Beta 1*. Iš Object Management Group internetinio puslapio [interaktyvus], [žiūrėta 2008-06-15]. Prieiga per internetą: < <http://www.omg.org/docs/dtc/07-07-01.pdf> >.
25. Object Management Group (2005). *Introduction to OMG's Unified Modeling Language™ (UML®)*. Iš Object Management Group internetinio puslapio [interaktyvus], [žiūrėta 2008-01-17]. Prieiga per internetą < [http://www.omg.org/gettingstarted/what\\_is\\_uml.htm](http://www.omg.org/gettingstarted/what_is_uml.htm) >.
26. Object Management Group (2001). *Model Driven Architecture (MDA)*. Dokumentas nr. Ormsc/2001-07-01. Iš Object Management Group internetinio puslapio [interaktyvus], [žiūrėta 2008-06-17]. Prieiga per internetą: < <http://www.omg.org/docs/ormsc/01-07-01.pdf> >.
27. Object Management Group. *Model Driven Architecture (MDA) FAQ...* Iš Object Management Group internetinio puslapio [interaktyvus], [žiūrėta 2008-06-17]. Prieiga per internetą: < [http://www.omg.org/mda/faq\\_mda.htm](http://www.omg.org/mda/faq_mda.htm) >.
28. Object Management Group (2003). *MDA Guide Version 1.0.1*. Iš Object Management Group internetinio puslapio [interaktyvus], [žiūrėta 2008-06-17]. Prieiga per internetą: < <http://www.omg.org/docs/omg/03-06-01.pdf> >.
29. Object Management Group (2006). *Object Constraint Language OMG Available Specification Version 2.0*. Iš Chalmers University of Technology and Göteborg University internetinio puslapio [interaktyvus], [žiūrėta 2008-01-17]. Prieiga per internetą < [http://www.cs.chalmers.se/Cs/Grundutb/Kurser/form/Papers/OCL\\_2.0.pdf](http://www.cs.chalmers.se/Cs/Grundutb/Kurser/form/Papers/OCL_2.0.pdf) >.
30. Object Management Group (2008). *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification*. Iš Object Management Group internetinio puslapio [interaktyvus], [žiūrėta 2008-01-18]. Prieiga per internetą: < <http://www.omg.org/docs/formal/08-01-02.pdf> >.

31. Object Management Group (2007). *Production Rule Representation (PRR), Beta 1*. Iš Object Management Group internetinio puslapio [interaktyvus], [žiūrėta 2008-06-15]. Prieiga per internetą: < <http://www.omg.org/docs/dtc/07-11-04.pdf> >.
32. Object Management Group (2007). *Production Rule Representation Ver. 1.03*. Submission to Business Modeling and Integration Domain Taskforce. Iš Object Management Group internetinio puslapio [interaktyvus], [žiūrėta 2008-06-15]. Prieiga per internetą: < <http://www.omg.org/docs/bmi/07-08-01.pdf> >.
33. Pašilskytė I., Nemuraitė L. (2006). *Verslo procesų modeliavimo kalbų analizė ir specifikuojamų priemonių sukūrimas*. Konferencijos pranešimų medžiaga. Kauno Technologijos Universitetas, Kaunas, Lietuva. Duomenų bazės ir modeliai, 11 Sekcija, 525 – 533 psl.
34. Rule Arts internetinis puslapis [interaktyvus], [žiūrėta 2009 01 19]. Prieiga per internetą: < <http://www.rulexpress.com/index.php> >.
35. SBeaVer internetinis puslapis [interaktyvus], [žiūrėta 2009 01 19]. Prieiga per internetą: < <http://sbeaver.sourceforge.net/> >.
36. Siegel, J. (2002). *Making the Case: OMG's Model Driven Architecture*. Iš SDTimes internetinio puslapio [interaktyvus], [žiūrėta 2008-06-16]. Prieiga per internetą: < <http://www.sdtimes.com/content/article.aspx?ArticleID=26807> >.
37. Tabet S., Wagner G., Spreeuwenberg S., Vincent P., Jacques G., Pelnant J., Frank J., Durand J. (2004). *OMG Production Rule Representation – Context and Current Status*. Iš World Wide Web Consortium (W3C) internetinio puslapio [interaktyvus], [žiūrėta 2008-01-18]. Prieiga per internetą: < <http://www.w3.org/2004/12/rules-ws/paper/53/> >.
38. Vasilecas O., Būgaitė D. (2006). *Dalykinės srities ontologijos panaudojimas verslo taisyklių sistemai kurti*. Konferencijos pranešimų medžiaga. Kauno Technologijos Universitetas, Kaunas, Lietuva. Veiklos procesų ir informacinių poreikių analizė, 12 Sekcija, 624 – 635 psl.
39. Vedrickas G., Nemuraitė L. (2006). *Veiklos taisyklių hierarchijos modelis konfigūruojamų elektroninių paslaugų projektavimui*. Konferencijos pranešimų medžiaga. Kauno Technologijos Universitetas, Kaunas, Lietuva. Veiklos procesų ir informacinių poreikių analizė, 12 sekcija, 648 – 654 psl.
40. Šilingas D. *Reikalavimų valdymas. Programinės įrangos reikalavimų valdymo principai ir praktika*. Iš Baltijos programinės įrangos internetinio puslapio [interaktyvus]. [žiūrėta 2009 05 01]. Prieiga per internetą: < <http://www.bpi.lt/text.php?arg=119&item=161&lang=1> >.

41. Šarkiūnaitė, I; Simutis, R; Krikščiūnienė, D. (2007) *Magistro baigiamojo darbo rengimo tvarka. Metodiniai nurodymai. VU KHF informatikos katedros verslo informatikos ir verslo informacijos sistemų studijų programų studentams*. Vilnius: Vilniaus universiteto leidykla.



# 1 PRIEDAS: SBVR STANDARTO MATEMATINIAI IR LOGINIAI OPERATORIAI

## C.1.1.1 Quantification

each	<a href="#">universal quantification</a>
some	<a href="#">existential quantification</a>
at least one	<a href="#">existential quantification</a>
at least $n$	<a href="#">at-least-n quantification</a>
at most one	<a href="#">at-most-one quantification</a>
at most $n$	<a href="#">at-most-n quantification</a>
exactly one	<a href="#">exactly-one quantification</a>
exactly $n$	<a href="#">exactly-n quantification</a>
at least $n$ and at most $m$	<a href="#">numeric range quantification</a>
more than one	<a href="#">at-least-n quantification</a> with $n = 2$

## C.1.1.2 Logical Operations

it is not the case that $p$	<a href="#">logical negation</a>
$p$ and $q$	<a href="#">conjunction</a>
$p$ or $q$	<a href="#">disjunction</a>
$p$ or $q$ but not both	<a href="#">exclusive disjunction</a>
if $p$ then $q$	<a href="#">implication</a>
$q$ if $p$	<a href="#">implication</a>
$p$ if and only if $q$	<a href="#">equivalence</a> (see exception explained under Modal Operations below)
not both $p$ and $q$	<a href="#">nand formulation</a>
neither $p$ nor $q$	<a href="#">nor formulation</a>
$p$ whether or not $q$	<a href="#">whether-or-not formulation</a>

## C.1.1.3 Modal Operations

it is obligatory that $p$	<a href="#">obligation formulation</a>
it is prohibited that $p$	<a href="#">obligation formulation</a> embedding a <a href="#">logical negation</a>
it is necessary that $p$	<a href="#">necessity formulation</a>
it is impossible that $p$	<a href="#">necessity formulation</a> embedding a <a href="#">logical negation</a>
it is possible that $p$	<a href="#">possibility formulation</a>
it is permitted that $p$	<a href="#">permissibility formulation</a>

The following key words are used within expressions having a verb to form verb complexes that add a modal operation.

... must ...	<a href="#">obligation formulation</a>
... must not ...	<a href="#">obligation formulation</a> embedding a <a href="#">logical negation</a>
... always ...	<a href="#">necessity formulation</a>
... never ...	<a href="#">necessity formulation</a> embedding a <a href="#">logical negation</a>
... may ...	<a href="#">permissibility formulation</a>

## 2 PRIEDAS: SBVR STANDARTO META-MODELIO FRAGMENTAS

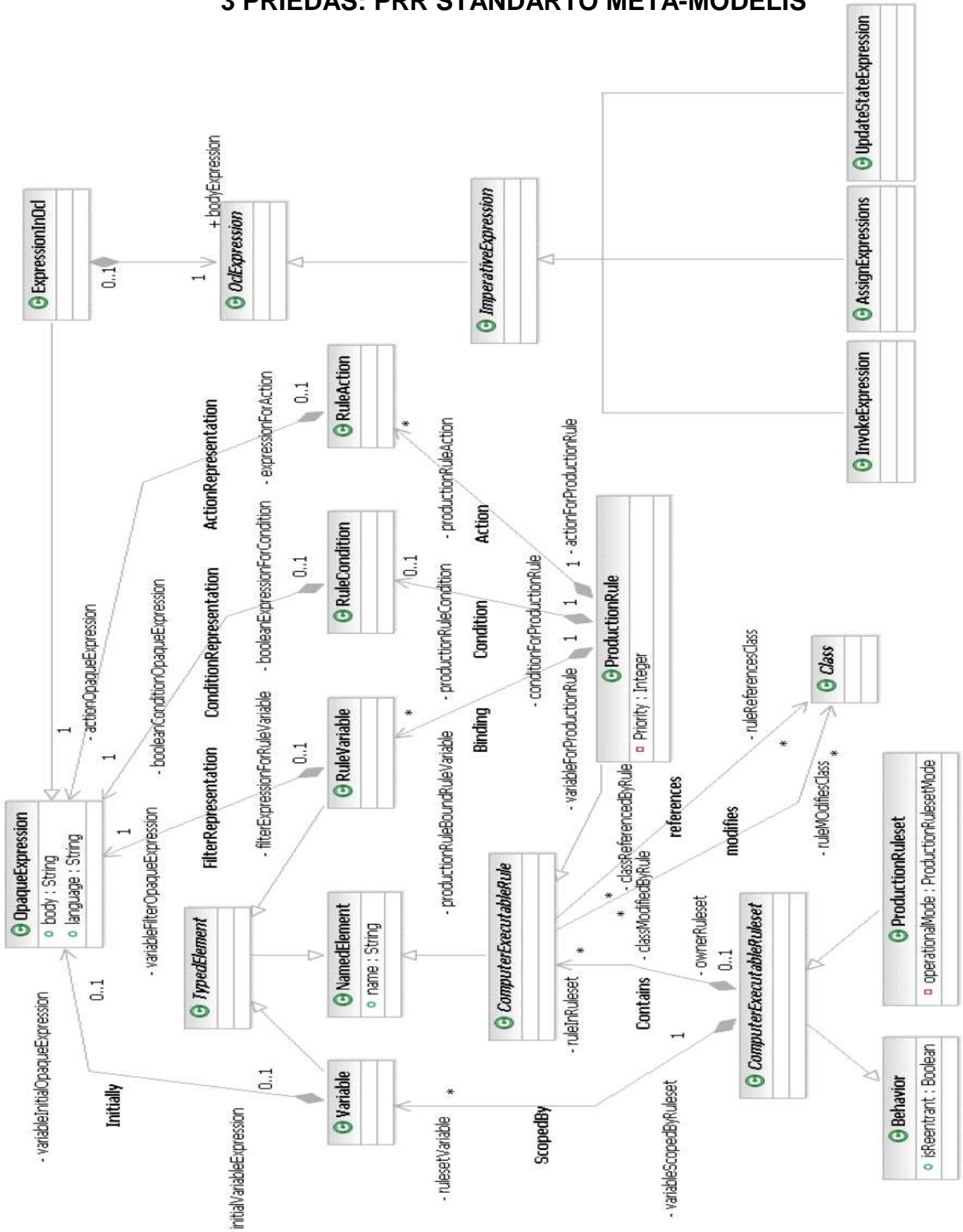
SBVR standarto meta-modelis yra padalintas į kelias dalis, kurios išreikštos kaip XML dokumentai, juos galima rasti tokiais adresais:

<http://www.omg.org/spec/SBVR/20070901/MeaningAndRepresentation-model.xml>  
<http://www.omg.org/spec/SBVR/20070901/LogicalFormulationOfSemantics-model.xml>  
<http://www.omg.org/spec/SBVR/20070901/DescribingBusinessVocabularies-model.xml>  
<http://www.omg.org/spec/SBVR/20070901/DescribingBusinessRules-model.xml>  
<http://www.omg.org/spec/SBVR/20070901/SBVR-model.xml>

Meta-modelio dalies SBVR-model fragmentas:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xmi:XMI xmi:version="2.1" xmlns:xmi="http://schema.omg.org/spec/XMI/2.1"
  xmlns:cmof="http://schema.omg.org/spec/MOF/2.0/cmof.xml">
<cmof:Package xmi:id="SBVR" name="SBVR" uri="http://www.omg.org/spec/SBVR/20070901/SBVR.xml">
  <ownedMember xmi:type="cmof:Class" name="actuality" xmi:id="actuality" superClass="stateOfAffairs" />
  <ownedMember xmi:type="cmof:Class" name="adopted definition" xmi:id="adoptedDefinition" superClass="definition"
  />
  <ownedMember xmi:type="cmof:Class" name="advice" xmi:id="advice" superClass="elementOfGuidance" />
  <elementImport xmi:type="cmof:ElementImport" alias="business advice of permission or possibility"
    xmi:id="businessAdviceOfPermissionOrPossibility" importedElement="advice" />
  <ownedMember xmi:type="cmof:Class" name="advice of contingency" xmi:id="adviceOfContingency"
    superClass="adviceOfPossibility" />
  <ownedMember xmi:type="cmof:Class" name="advice of optionality" xmi:id="adviceOfOptionality"
    superClass="adviceOfPermission" />
  <ownedMember xmi:type="cmof:Class" name="advice of permission" xmi:id="adviceOfPermission"
    superClass="advice" />
  <ownedMember xmi:type="cmof:Class" name="advice of possibility" xmi:id="adviceOfPossibility" superClass="advice"
  />
  <ownedMember xmi:type="cmof:Class" name="advice statement" xmi:id="adviceStatement"
    superClass="guidanceStatement" />
  <ownedMember xmi:type="cmof:Class" name="aggregation formulation" xmi:id="aggregationFormulation"
    superClass="projectingFormulation" />
  <ownedMember xmi:type="cmof:Class" name="answer nominalization" xmi:id="answerNominalization"
    superClass="projectingFormulation" />
  <ownedMember xmi:type="cmof:Class" name="associative fact type" xmi:id="associativeFactType"
    superClass="factType" />
  <ownedMember xmi:type="cmof:Class" name="assortment fact type" xmi:id="assortmentFactType"
    superClass="factType" />
  <ownedMember xmi:type="cmof:Class" name="at-least-n quantification" xmi:id="at-least-nQuantification"
    superClass="quantification">
    <ownedAttribute xmi:type="cmof:Property" name="minimum cardinality" xmi:id="at-least-
      nQuantification.minimumCardinality" type="nonnegativeInteger" lower="1" upper="1" />
  </ownedMember>
  <ownedMember xmi:type="cmof:Class" name="at-most-n quantification" xmi:id="at-most-nQuantification"
    superClass="quantification">
    <ownedAttribute xmi:type="cmof:Property" name="maximum cardinality" xmi:id="at-most-
      nQuantification.maximumCardinality" type="nonnegativeInteger" lower="1" upper="1" />
  </ownedMember>
  <ownedMember xmi:type="cmof:Class" name="at-most-one quantification" xmi:id="at-most-oneQuantification"
    superClass="at-most-nQuantification" />
  <ownedMember xmi:type="cmof:Class" name="atomic formulation" xmi:id="atomicFormulation"
    superClass="logicalFormulation">
    <ownedAttribute xmi:type="cmof:Property" name="role binding" xmi:id="atomicFormulation.roleBinding"
      type="roleBinding" lower="0" upper="*" />
  </ownedMember>
</cmof:Package>
```

### 3 PRIEDAS: PRR STANDARTO META-MODELIS



Šaltinis: Object Management Group (2007). *Production Rule Representation (PRR)*, Beta 1.

## 4 PRIEDAS: VEIKLOS TAISYKLIŲ SAUGYKLOS LENTELIŲ IR ATRIBUTŲ REIKŠMĖS

Priede pateikiamos suprojektuotos SBVR šablonų ir taisyklių saugyklos lentelė ir atributų aprašai..

### BusinessRule

Saugomi duomenys apie veiklos taisykles. Šioje lentelėje saugomas veiklos taisyklės turinys ir ji siejasi su kitomis būtinomis lentelėmis kuriose saugoma informacija kaip veiklos taisyklė turėtų būti interpretuojama.

Atributas	Aprašas	Galimos reikšmės	Būtinas	Tipas
<i>BRID</i>	Unikalus identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	bigint(20)
<i>ruleText</i>	Pilna veiklos taisyklė tekstiniame pavidale, kad kiekvieną kartą norint ją tik pamatyti nereikėtų sujunginėti atskirų dalių.	Bet koks tekstas.	✓	text
<i>PID</i>	Identifikacinis numeris šablono pagal kurį reikėtų interpretuoti šią veiklos taisyklę.	Sveikasis teigiamas skaičius	✓	bigint(20)
<i>description</i>	Laisvai suformuojamas veiklos taisyklės apibūdinimas kuris galėtų pateikti daugiau informacijos apie veiklos taisyklę.	Bet koks tekstas.		Text
<b>Vienas-su-daug ryšys su</b>		<b>Apibūdinimas</b>		
<BusinessRulePart>.<BRID>		Kiekviena veiklos taisyklė yra sudaryta bent iš vienos ar daugiau dalių.		
<b>Daug-su-vienas ryšys su</b>		<b>Apibūdinimas</b>		
<pattern>.<PID>		daug lentelėje saugomų veiklos taisyklių gali būti interpretuojamos pagal ta patį šabloną.		

## BusinessRulePart

Saugomi duomenys apie veiklos taisyklių dalis. T.y. saugoma struktūra kaip ir iš ko veiklos taisyklė yra sudaryta.

Atributas	Aprašas	Galimos reikšmės	Būtinas	Tipas
<i>BRPID</i>	Unikalus identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	bigint(20)
<i>BRID</i>	Veiklos taisyklės kuriai priklauso ši dalis identifikacinis numeris	Sveikasis teigiamas skaičius	✓	bigint(20)
<i>seqNr</i>	Sekos numeris kelinta veiklos taisyklėje ši dalis yra.	Sveikasis teigiamas skaičius. Kadangi nepatartina, kad veiklos taisyklė būtų ilga ir sudaryta iš didelio kiekio dalių, tai sekos saugojimui pakanka dviejų skaitmenų.	✓	smallint(2)
<i>PtypeID</i>	Laisvai suformuluojamas veiklos taisyklės apibūdinimas kuris galėtų pateikti daugiau informacijos apie veiklos taisyklę.	Iš anksto parengtų tipų pavadinimai	✓	bigint(20)
<i>PTSTID</i>	Tekstinio įrašo kuris taisyklės dalyje rezervuotas kaip konstanta identifikacinis numeris.	Sveikasis teigiamas skaičius		bigint(20)
<b>Vienas-su-daug ryšys su</b>		<b>Apibūdinimas</b>		
<BusinessRuleConstantValues>.<BRPID>		Kiekvienoje dalyje gali būti keletas vartotojo įvedamų konstantų.		
<b>Daug-su-vienas ryšys su</b>		<b>Apibūdinimas</b>		
<BusinessRule>.<BRID>		daug lentelėje saugomų veiklos taisyklių dalių gali priklausyti vienai veiklos taisyklei		
<PreparedTextSubText>.<PTSTID>		Daugelyje lentelėje saugomų dalių gali būti panaudotas tas pats iš anksto rezervuotas konstantinius tekstas.		
<PatternPart>.<PPID>		Daugelis lentelėje saugomų veiklos taisyklių dalių gali turėti tą patį šablono dalies tipą.		

## BusinessRuleConstantValues

Paprastai rašant veiklos taisykles vartotojas įveda savo terminus ir įvairias skaitines bei simbolines reikšmes. Ši lentelė skirta vartotojo įvestiems duomenims saugoti.

Atributas	Aprašas	Galimos reikšmės	Būtinai	Tipas
<i>BRCVID</i>	Unikalus identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	bigint(20)
<i>BRPID</i>	Veiklos taisyklės dalies kuriai priklauso ši konstanta identifikacinis numeris	Sveikasis teigiamas skaičius	✓	bigint(20)
<i>seqNr</i>	Eilės numeris, kelintu ši konstanta eina veiklos taisyklės dalyje.	Sveikasis teigiamas skaičius. Kadangi nepatartina, kad veiklos taisyklė būtų ilga ir sudaryta iš didelio kiekio dalių, tai sekos saugojimui pakanka dviejų skaitmenų.	✓	smallint(2)
<i>value</i>	Įvesta vartotojo reikšmė, ją patogiu saugoti tekstiniame pavidale	Bet koks tekstas	✓	varchar(50)
<b>daug-su-vienas ryšys su</b>		<b>Apibūdinimas</b>		
<BusinessRule>.<BRID>		daug lentelėje saugomų konstantų gali priklausyti tai pačiai veiklos taisyklės daliai.		

## Pattern

Saugomi galimi šablonai pagal kuriuos gali būti interpretuojamos veiklos taisyklės.

Atributas	Aprašas	Galimos reikšmės	Būtinai	Tipas
<i>PID</i>	Unikalus identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	bigint(20)
<i>name</i>	Šablono pavadinimas	Bet koks tekstas	✓	varchar(20)
<i>description</i>	Veiklos taisyklės šabloną apibūdinantis bet koks tekstas.	Bet koks tekstas		text
<b>Vienas-su-daug ryšys su</b>		<b>Apibūdinimas</b>		
<BusinessRule>.<BRID>		Pagal vieną šabloną gali būti užrašyta daug taisyklių		
<PatternPart>.<PID>		Veiklos taisyklės šablonas turi būti sudarytas iš vienos arba daugiau dalių.		

## PatternPart

Saugomi duomenys apie veiklos taisyklių šablonų dalis. T.y. saugoma struktūra kaip ir iš ko veiklos taisyklės šablonas yra sudarytas.

Atributas	Aprašas	Galimos reikšmės	Būtinas	Tipas
<i>PPID</i>	Unikalus identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	bigint(20)
<i>isNecessary</i>	Nurodo ar ši dalis veiklos taisyklėje yra būtina	Taip arba ne	✓	boolean
<i>seqNr</i>	Sekos numeris kelinta veiklos taisyklės šablone ši dalis yra.	Sveikasis teigiamas skaičius. Kadangi nepatartina, kad veiklos taisyklė būtų ilga ir sudaryta iš didelio kiekio dalių, tai sekos saugojimui pakanka dviejų skaitmenų.	✓	smallint(2)
<i>PID</i>	Identifikacinis numeris šablono kuriam priklauso ši dalis	Sveikasis teigiamas skaičius	✓	bigint(20)
<i>PTID</i>	Identifikacinis numeris iš anksto parengto teksto grupės kuri gali būti naudojama šioje dalyje.	Sveikasis teigiamas skaičius	✓	bigint(20)
<i>PtypeID</i>	Identifikacinis numeris šablono dalie tipo kuriam ji priskirt	Sveikasis teigiamas skaičius	✓	bigint(20)
<i>value</i>	Išraiška kaip šablono dalis turi būti išreikšta. Nurodoma kur turi būti vartotojo įvesta reikšmė, kur iš anksto parengta konstanta ir pan. Atitinkamos vietos išskiriamos simboliais „< >“		✓	bigint(20)
<b>Vienas-su-daug ryšys su</b>		<b>Apibūdinimas</b>		
<BusinessRulePart>.<PPID>		Pagal viena šablono dalį galima būt parašius daug taisyklių.		
<b>Daug-su-vienas ryšys su</b>		<b>Apibūdinimas</b>		
<Pattern>.<PID>		Vienas veiklos taisyklių šablonas turi būti sudarytas iš vienos ar daugiau dalių		
<PreparedText>.<PTID>		Tas pats iš anksto parengtas konstantinio teksto rinkinys gali būti panaudojamas daugelyje veiklos taisyklių dalių.		
<PartType>.<PTTypeID>		Daugelis šablonų dalių gali turėti tą patį tipą.		

## PartType

Šioje lentelėje saugomi veiklos taisyklių dalių tipai kurie pagal savo prasmę nurodo ką ta dalis turi daryti. Pvz: aprašyti konceptą, formuoti sąlygą ir t.t.

Atributas	Aprašas	Galimos reikšmės	Būtinas	Tipas
<i>PtypeID</i>	Unikalus identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	bigint(20)
<i>codeName</i>	Kodinis tipo vardas	Bet kokie simboliai	✓	varchar(20)
<i>name</i>	Laisva forma pasirinktas tipo vardas	Bet koks tekstas .		varchar(20)
<i>description</i>	Tipo aprašymas	Bet koks tekstas		text
<b>Vienas-su-daug ryšys su</b>		<b>Apibūdinimas</b>		
<BusinessRulePart>.<PPID>		Vienas šablono dalies tipas gali apibūdinti daug skirtingų šablonų dalių.		

## PreparedText

Šioje lentelėje saugoma iš anksto parengtos tekstinių konstantų grupės (pvz: must/should būtų viena grupė iš kurios gali būti pavartotas tik vienas iš anksto parengtas žodis toje pat dalyje)

Atributas	Aprašas	Galimos reikšmės	Būtinas	Tipas
<i>PTID</i>	Unikalus identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	bigint(20)
<i>name</i>	Teksto grupės vardas	Bet kokie simboliai	✓	varchar(20)
<i>description</i>	Laisva forma pasirinktas grupės aprašas	Bet koks tekstas .		varchar(20)
<b>Vienas-su-daug ryšys su</b>		<b>Apibūdinimas</b>		
<PatternPart >.<PTID>	Ta pati parengto teksto grupė gali būti panaudojama daugelyje šablonų dalių			
<PreparedTextSubText>.<PTID> >	Vienoje iš anksto parengto teksto grupėje gali būti vienas arba daug žodžių.			

## PreparedTextSubText

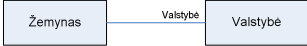
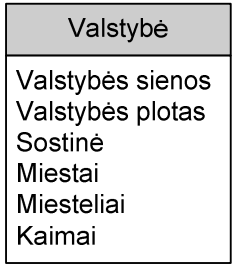
Šioje lentelėje saugoma iš anksto parengtos tekstinių konstantų grupės žodžiai

Atributas	Aprašas	Galimos reikšmės	Būtinas	Tipas
<i>PTSTID</i>	Unikalus identifikacinis numeris.	Automatiškai priskiriamas numeris.	✓	bigint(20)
<i>PTID</i>	Grupės kuriai priklauso šis žodis identifikacinis numeris	Sveikasis teigiamas skaičius	✓	bigint(20)
<i>Value</i>	Žodžio reikšmė	Bet koks tekstas .	✓	varchar(20)
<b>Vienas-su-daug ryšys su</b>		<b>Apibūdinimas</b>		
<BusinessRulePart >.<PTID>	Tas pats iš anksto parengtas žodis gali būti panaudotas daugelyje taisyklių			
<b>daug-su-vienas ryšys su</b>		<b>Apibūdinimas</b>		
<PreparedText>.<PTID>	Vienas arba daugelis žodžių gali sudaryti iš anksto parengtų žodžių grupę			



## 5 PRIEDAS: DALYKINĖS SRITIES KONCEPTŲ ŽODYNAS

Taigi, turime dalykinės srities konceptų žodyną. Veiklos taisyklės bus kuriamos iš šių konceptų bei raktažodžių. Remiantis siūloma metodologija, sudarius konceptų žodyną, sudaromos struktūrinės taisyklės, t.y. taisyklės, kurios nurodo sąlygas pagal kurias vykdoma veikla.

	Koncepto pavadinimas	Informacija apie konceptą	Faktai	Pastabos
1	Žemynas	<p>Apibrėžimas (Definition): <u>sausumos dalis, kuri yra supama jūrų ir vandenynų</u></p> <p>Šaltinis (Source): Lietuvos geografijos mokytojų asociacija</p> <p>Būtinybė (Necessity): <u>žemynas privalo būti bent keletas tūkstančių km<sup>2</sup></u></p> <p>Aprašymas (Description): didesnė kaip keleto tūkstančių km<sup>2</sup> ploto sausumos dalis (Žemės plutos masivas), supama jūrų ir vandenynų.</p>	<p><u>Žemynas turi valstybes</u></p> <p>Pastaba (Note): Žemynas Antarktida neturi valstybių</p>	<p>Suformuojama klasė „Žemynas“, kurioje gali būti talpinama 0 arba daugiau klasės „Valstybės“ objektų</p> <p>Grafiškai tai atrodytų taip:</p>  <p>Kaip matome suformuojamas fakto tipas su žodeliu <i>turi</i></p>
2	Valstybė	<p>Apibrėžimas (Definition): <u>šalis, kuri turi valstybingumo statusą</u></p> <p>Šaltinis (Source): Lietuvos geografijos mokytojų asociacija</p> <p>Būtinybė (Necessity): <u>valstybė privalo būti nepriklausoma</u></p> <p>Aprašymas (Description): šalis, turinti valstybingumo (nepriklausomybės) statusą.</p>	<p><u>Valstybė turi valstybės sienas</u></p> <p>Būtinybė (Necessity): <u>valstybė privalo turėti valstybės sienas</u></p> <p>Galimybė (Possibility): <u>valstybė gali turėti bendras valstybės sienas su kitomis valstybėmis</u></p> <p><u>Valstybė turi valstybės plotą</u></p> <p>Būtinybė (Necessity): <u>valstybė privalo turėti valstybės plotą</u></p> <p><u>Valstybė turi sostinę</u></p> <p>Būtinybė (Necessity): <u>valstybė privalo turėti tik vieną sostinę</u></p> <p><u>Valstybė turi miestus</u></p> <p>Būtinybė (Necessity): <u>valstybėje privalo būti bent vienas miestas</u></p>	<p>Suformuojama klasė „Valstybė“ kuri turi atributus: <u>valstybės sienos, valstybės plotas, sostinė, miestai, miesteliai, kaimai</u></p> 
3	Sostinė	<p>Apibrėžimas (Definition): <u>miestas kuriame yra valstybės politinis ir administracinis centras</u></p> <p>Būtinybė (Necessity): <u>sostinė privalo būti unikali</u></p> <p>Aprašymas (Description): valstybės administracinis bei politinis centras, dažniausiai esantis didžiausiam šalies mieste</p>		Tai klasės „Valstybė“ atributas
4	Miestas	<p>Apibrėžimas (Definition): <u>gyvenama vietovė kurioje yra bent 3000 gyventojų ir bent 2/3 jų dirba pramonėje ir/arba verslo srityje ir/arba gamybinėje srityje ir/arba socialinės infrastruktūros srityje</u></p> <p>Pastaba (Note): Mažiau kaip 3 tūkst. Gyventojų turintys Lietuvos Respublikos miestai, rajonų miestai bei miesto tipo gyvenvietės, turėję miesto statusą, kaip gyvenamosios vietovės yra laikomi miestais ir</p>		Tai klasės „Valstybė“ atributas

		įsigaliojus Lietuvos Respublikos teritorijos administracinių vienetų ir jų ribų įstatymui		
5	Miestelis	Apibrėžimas (Definition): <u>gyvenama vietovė kurioje yra mažiausiai 500 ir daugiausiai 3000 gyventojų ir bent 2/3 jų dirba pramonėje ir/arba verslo srityje ir/arba gamybinėje srityje ir/arba socialinės infrastruktūros srityje</u> Pastaba (Note): tradiciniai miesteliai ne visada turi visus miestelio požymius, tačiau vis tiek yra vadinami miesteliais		Tai klasės „Valstybė“ atributas
6	Kaimas	Apibrėžimas (Definition): <u>gyvenama vietovė kuri neturi miesto ir miestelio požymių</u>		Tai klasės „Valstybė“ atributas
7	Adresas	Apibrėžimas (Definition): <u>adresas nurodo kur yra konkretus objektas</u> Būtinybė (Necessity): <b>kiekvienas adresas privalo būti unikalus</b>	<u>Adresas turi miestą</u>  <u>Adresas turi gatvę</u>  <u>Adresas turi namo numerį</u>	Suformuojama klasė „Adresas“ turinti atributus: <u>miestas, gatvė, namo numeris</u>
8	Priartinimas	Apibrėžimas (Definition): <u>priartinimas nurodo žemėlapiu mastelį</u>	<u>Priartinimas turi dydį</u> Būtinybė (Necessity): <u>dydis privalo būti mažiausiai 1 ir daugiausiai 18</u> Pastaba (Note): Dydis yra skaičius nurodantis kiek kartų padidintas ar sumažintas žemėlapis	Tai kintamasis, naudojamas sudaryti sąlygoms
9	Žymeklis	Apibrėžimas (Definition): <u>žymeklis jungiasi su kitu žymekliu ir nurodo maršrutą</u>		Klasės „Maršrutas“ atributas
10	Maršrutas	Apibrėžimas (Definition): <u>maršrutas nurodo kelią nuo vieno adreso iki kito adreso</u>	<u>Maršrutas turi žymeklius</u>  <u>Maršrutas turi adresus</u>	Suformuojama klasė „Maršrutas“ turinti atributą: <u>žymeklis</u> ir kurioje galima talpinti 0 arba daugiau klasės „Adresas“ atributų
11	Mar_Populiarumas	Apibrėžimas (Definition): <u>Mar Populiarumas nurodo dažniausiai naudojamus maršrutus</u>		Tai kintamasis, naudojamas sudaryti sąlygoms
12	Žemėlapis	Apibrėžimas (Definition): <u>Žemėlapis yra vieta, kurioje vaizduojami visi kiti konceptai</u>	<u>Žemėlapis turi žemynus</u>	Suformuojama klasė „Žemėlapis“ turinti atributą: <u>žemynas</u> ir kurioje galima talpinti 0 arba daugiau klasės „Žemynas“ atributų

Atributus nuo klasių galima atskirti iš to, kad atributai neturi faktų ir yra naudojami faktams apie kitus konceptus sudaryti. Taip pat žodyne yra ir kintamųjų kurie yra naudojami sudarant “sąlygas” ir “veiksmus” PRR standarte.

## 6 PRIEDAS: KONFERENCIJOJE PRISTATYTAS STRAIPSNIS

### ŠIUOLAIKINIŲ VEIKLOS TAISYKLIŲ STANDARTŲ TAIKYMAS KURIANT INFORMACINES SISTEMAS

Giedrimė Ivančikaitė<sup>1</sup>, Gintautas Garšva<sup>2</sup>, Kęstutis Kapočius<sup>3</sup>

<sup>1,2</sup> Vilniaus universitetas, Kauno humanitarinis fakultetas, Muitinės 8, Kaunas

<sup>3</sup> Kauno technologijos universitetas, Informacijos sistemų katedra, Studentų 50-308c, Kaunas

<sup>1</sup>Giedrime.Ivancikaite@khf.stud.vu.lt, <sup>2</sup>gintautas.garsva@vukhf.lt, <sup>3</sup>kestutis.kapocius@ktu.lt

Straipsnyje apžvelgiami OMG (Object Management Group) veiklos taisyklių standartai ir jų taikymas kuriant informacines sistemas. Įvertinama standartizuotų veiklos taisyklių koncepcijos sprendimų reikšmė, analizuojant naujausius OMG veiklos taisyklių standartus. Iš siūlomų OMG standartų išskirti SBVR (Semantics of Business Vocabulary and Business Rules) ir PRR (Production Rule Representation), kurie aktualūs veiklos taisyklėms. Abu standartai patenka į OMG modeliais grindžiamos architektūros (Model Driven Architecture - MDA) koncepciją ir ją praturtina. Tyrimo metu apibrėžtos pasirinktų standartų charakteringosios savybės, sukurta jų naudojimo metodikos koncepcija, įvertintos standartų susiejimo bei PRR išraiškų generavimo iš SBVR šablonų galimybės.

**Raktiniai žodžiai:** modeliais grindžiama architektūra, informacijos sistemos, veiklos taisyklės, OMG standartai, SBVR, PRR

#### Įvadas

Informacijos srantai įmonėse yra vienas iš esminių dalykų. Informacijos srautus ir įvairias veiklos funkcijas valdo informacinės sistemos (IS). Tačiau norint tiksliai apibrėžti konkrečios įmonės veiklos funkcijas, reikia išsiaiškinti reikalavimus kuriamai sistemai, identifikuoti sritis, kuri bus kompiuterizuojama.

Veiklos taisyklių (VT) koncepcijos taikymas kuriant IS yra perspektyvus, tačiau iki šiol jis buvo menkai išplėtotas ir beveik nestandardizuotas. Atsirandant atskiras problemas liečiantiems standartams, prielaidos VT koncepcijai plisti stiprėja, tačiau būtina atlikti išsamius tyrimus, susiejant naujus standartus su jau esamais ir sukuriant jų taikymo metodikas. Verslo sistemos lygmenyje taisyklės nusako veiklos procesų logiką, o IS lygmenyje – informacijos apdorojimo taisykles. Tai leidžia jas naudoti daliai kuriamos IS reikalavimų išreikšti.

Šiame straipsnyje apžvelgta VT koncepcija, apibendrinta taisyklių svarba kuriant informacines sistemas. Pagrindinis dėmesys sutelkiamas ties tarptautinės standartų organizacijos OMG (*Object Management Group*) siūlomais standartais, pristatomas sukurto pasirinktų standartų taikymo kuriant IS metodikos karkasas.

#### Veiklos taisyklių koncepcijos principai

Remiantis GUIDE modeliu [1], veiklos taisyklės gali būti trijų tipų:

- *Struktūrinis teiginys* – apibrėžtas terminas arba konstatuojamas faktas, išreiškiantis kurį nors verslo sistemos struktūros elementą.
- *Veiksmo teiginys* – apribojimo arba sąlygos teiginys, ribojantis/valdantis verslo sistemos veiksmus.
- *Išvedimo taisyklė* – teiginys, išvestas iš žinių, turimų apie veiklą.

IS kūrimo pradžioje VT surenkamos ir aprėžiamos natūraliąja kalba. Veiklos taisyklių šaltinis turėtų būti veiklos ekspertai [2]. IS kūrimo metu projektuotojai palaipsniui formalizuoja VT išraiškas. Užfiksuotos taisyklės apibrėžia būsimosios sistemos veiklą, t.y. nustato ribą tarp priimtinos ir nepriimtinos veiklos [2].

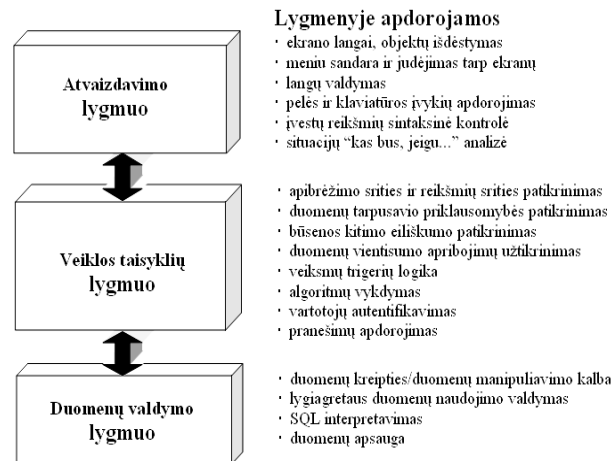
Nagrinėjant VT grindžiamos IS architektūros koncepciją pažymėtina tai, jog veiklos taisyklės sudaro atskirą sistemos lygmenį (žr. pav. 1.), kuris yra nepriklausomas nuo procesų ir duomenų lygmenų, - tai sukuria sąlygas taisykles keisti atskirai ir centralizuotai. Be to, sistema įgauna lankstumo. Šie faktoriai lemia VT koncepcija grindžiamų sprendimų tyrimo aktualumą.

#### Pagrindiniai veiklos taisyklių modeliavimo standartai

Veiklos taisyklėms ir veiklos modeliavimui OMG pateikia tokius jau patvirtintus arba kuriamus VT specifikuojančius klausimus liečiančius standartus:

- SBVR (*Semantics of Business Vocabulary and Business Rules*). Tai VT apibrėžimo standartas, orientuotas į semantinę natūraliosios kalbos sakinių, kuriais išreiškiamos VT, struktūrą [5]. Pagal šį standartą VT aprašomos naudojant šablonus, t.y. ne formalia kalba, o tam tikromis semantinėmis formuluotėmis.

- PRR (*Production Rule Representation*) skirtas produkcinėms veiklos taisyklėms apibrėžti. Produkcinė taisyklė yra programavimo logikos teiginys, nusakantis vieno ar daugiau veiksmų seką, jeigu tenkinama tų veiksmų sąlyga [10]. PRR standarte produkcinės veiklos taisyklės aprašomos loginiais teiginiais (jeigu <sąlyga> tada <veiksmai> kitu atveju <veiksmai>).



Pav. 1. Supaprastintas veiklos taisyklių lygmens vaidmuo VT grindžiamoje IS architektūroje [3]

- BMM (*Business Motivation Model*). Remiantis šiuo standartu galima sudaryti planą, pagrindžiantį įvairias veiklos metodologijas. Taikant BMM, veiklos planų elementai saugomi ir susiejami su informacija apie taikymo sritį nepriklausomai nuo to, kokia metodologija buvo naudojama jiems atrasti ir sukurti [6]. Pagrindinė BMM idėja – leisti sukurti veiklos modelį, skirtą veiklos plano elementams vaizduoti prieš pradėdant sistemos projektavimą. Tokiu būdu veiklos planai gali tapti sistemos pagrindu ir orientuoti ją į konkrečią veiklą [4].

Apžvelgus esamus standartus, galima teigti, kad specifikuojant VT reikšmingiausi būtų SBVR ir PRR, kurie ir buvo pasirinkti tolimesniems tyrimams. Šie standartai sąlyginai nauji, todėl detalių metodikų jiems nėra sukurta. BMM labiau orientuotas į veiklos modeliavimą t.y. skirtas pradinėi veiklos koncepcijai apibrėžti, o ne konkrečioms veiklos taisyklėms formuluoti.

Svarbu pabrėžti, jog minėti standartai priklauso OMG – tarptautinei sistemų ir veiklos modeliavimo standartų kūrimo organizacijai. Taigi, svarbu suprasti PRR ir SBVR vaidmenį MDA architektūros kontekste.

## MDA architektūra

OMG sukurta modeliais grindžiama architektūra (MDA) suteikia galimybę pažvelgti į verslo bei technologijų pokyčius nepriklausomai nuo programinės įrangos gamintojo. MDA yra paremta OMG standartais, o jos esmė – veiklos ir realizavimo logikos atskyrimas nuo konkrečios platformos arba IS kūrimo technologijos. Pagal MDA visus modelius, nepriklausomai nuo modeliujamos sistemos, galima suskirstyti į tris rūšis:

- CIM (*Computation Independent Model*) modelis neatskleidžia sistemos struktūros detalių. Šis modelis kartais vadinamas domeno (angl. *domain*) modeliu ir jo specifikacijoje naudojamas žodynas su terminais, kurie susiję su nagrinėjama sritimi. CIM galima laikyti „tiltu“ tarp dalykinės srities ir jos ekspertų bei ekspertų, projektuojančių ir kuriančių sistemos artefaktus [7].

- PIM (*Platform Independent Model*) modelyje pateikiamos formalios sistemos struktūros ir funkcijų specifikacijos, atmetant technines detales [8]. Jo komponentai ir sąsajos – tai būdas gauti abstraktų informacinės programos ar sistemos vaizdą.

- PSM (*Platform Specific Model*) sujungia PIM specifikacijas su detalėmis, nusakančiomis, kaip sistemai naudotis konkrečia platforma.

Pilnas MDA taikymas susideda iš tikslaus PIM modelio, vieno ar daugiau PSM modelių ir pilnų pritaikymų kiekvienai platformai, kurią numatoma palaikyti [9]. MDA apibrėžia etapus, kuriuos turi pereiti modelis nuo kūrimo pradžios iki įdiegimo. Todėl viso IS kūrimo metu išlaikoma pradinė sistemos idėja ir reikalavimai. Sudaryti sistemos modeliai detalizuoja ir saugo veiklos procesų logiką bei yra sistemos realizavimo, palaikymo ir evoliucijos pagrindas.

Prieš atsirandant MDA, modeliai ir programinis kodas būdavo kuriami atskirai. Programuotojai traktuodavo modelius kaip gaires ar apytikrius planus, bet ne kaip griežtus reikalavimus. Palyginus su kūrimu be modeliavimo, kūrimas

tokiame kontekste kainuodavo daugiau bei užtrukdavo ilgiau, o sukurta sistema turėdavo mažai panašumų su pradinių modelių, kuris atnešdavo nedaug naudos, nepaisant to, koks geras bebūdavo [12].

## SBVR ir PRR standartų susiejimo metodika

Prieš pradėdant metodikos apžvalgą svarbu pabrėžti, jog šiame straipsnyje pristatomo tyrimo tikslas nebuvo OMG standartų sulietvinimas. Todėl žemiau pateikiami VT pavyzdžiai SBVR formatu formuluojami pagal originalius, angliškus šablonus. Vis tik sudarytoji metodika yra universali naudojamos kalbos požiūriu.

SBVR ir PRR veiklos taisyklių išraiškos ženkliai skiriasi. PRR taisyklės yra griežtai apibrėžtos, iš jų yra įmanoma generuoti programinį kodą, o SBVR išraiškos labai įvairios. Dėl skirtingos specifikos SBVR taisyklė retai gali būti tiesiogiai išreikšta PRR taisykle, tačiau tam tikrais atvejais tai įmanoma, pavyzdžiui, turime VT:

**SBVR:** *It is obligatory that rental duration of each rental is at most 90 rental days [11].*

(Vertimas: Privaloma, kad nuomos trukmė kiekvienai nuomojamai mašinai būtų mažesnė nei 90 nuomos dienų)

Norint šią taisyklę pervesti į produkcinių veiklos taisyklių formatą <For [rule variables] if [condition] then [action list]>, reikia iš SBVR taisyklės išskirti pagrindines produkcinių taisyklių dalis:

```
[rule variables]: rental, rentalDays, result
[condition]: rental days < 90
[action list]: rent= true
```

Kaip matome, pakeičiant formatą, sukonkretinamos ir kai kurios išraiškos bei gaunama tokia produkcinė veiklos taisyklė:

```
For [rental] if [rentalDays<90] then [result=true]
```

Pagal PRR išreikštos VT yra glaustesnės ir aiškiau nurodo, ką reikia daryti konkrečiu atveju. PRR taisyklės galima analizuoti kaip atskirus elementus, nesigilinant į veiklos sritį. Tarkim, taisyklė (nusakanti vieną iš elektroninės parduotuvės veiklos taisyklių) natūraliąja kalba skamba taip:

*If there is at least one garment item in the customer shopping cart and this garment is from a collection then add a hyperlink to the collections page in the customer web page.*

(Vertimas: Jeigu kliento krepšelyje yra bent vienas rūbas ir tas rūbas yra iš kolekcijos, tada į kliento tinklalapį reikia pridėti nuorodą į kolekcijų tinklalapį)

Taisyklėje atsispindi visos produkcinių veiklos taisyklių dalys:

```
[rule variables]: customer, shopping cart, garment
[condition]: garment > 0
[action list]: hyperlink to collections
```

Formali šios taisyklės PRR išraiška, užrašyta PRR OCL kalba, atrodytų taip:

```
Rule atLeastOneGarment
ruleVariable:
    ?customer: Customer = Customer->any()
    ?sCart: ShoppingCart = ShoppingCart->any(c:customer|c=?customer)
    ?garmentItems: Set = ?sCart.items->select(e:items|e.isCollection())
Condition:
    ?garmentItems.size() > 0
Action:
    ?customer.hyperlinkToCollections = true
```

Pavertus šios taisyklės anglišką formuluotę į SBVR formą, ši taisyklė atrodytų taip:

It is obligatory that a hyperlink to the collections page is added in the customer web page if *there is* at least one garment in the customer shopping cart and the garment is from a collection.

(Vertimas: Privaloma, kad nuoroda į kolekcijų tinklalapį būtų pridėta kliento tinklalapyje, jeigu kliento krepšelyje yra bent vienas rūbas ir ta rūbas yra iš kolekcijos)

Kaip matome, naudojantis SBVR formuluotėmis, galima išreikšti PRR taisyklių aprašymus. Tačiau formuluotė PRR OCL kalba turi kintamąjį „customer“ (klientas), kuris nenusakytas veiklos taisyklėje. Jeigu taisyklė išreiškiama SBVR, tada galima patikrinti, kokias sąsajas taisyklėje minimi konceptai turi su kitais veiklos elementais. Vadinasi, naudojantis SBVR, lengviau specifiuoti esminius produkcinių VT elementus. Taigi, sudarius MDA CIM lygio konstrukcijas pagal SBVR, jos būtų kaip gairės, kuriant PIM lygio modelį. Tačiau SBVR taisyklių šablonai turi būti griežtai apibrėžti, norint, kad iš jų būtų galima generuoti PRR išraiškas. Apibendrinta sudarytos metodikos schema, pateikiama pav. 2.

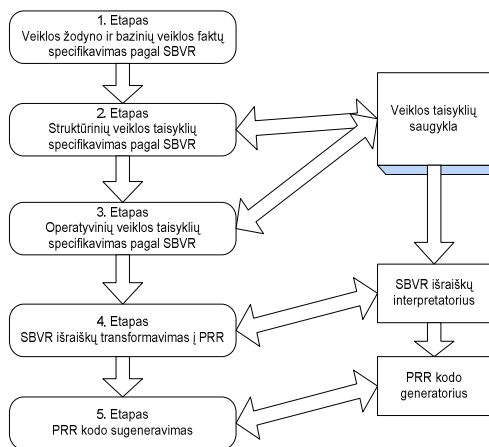
Galima teigti, kad SBVR ir PRR standartai vienas kitą papildė. Tačiau SBVR standarto taisyklės turi būti įdėtos į tam tikrus rėmus. Detalizavus nagrinėjamą dalykinę sritį SBVR standartu, t.y. išskyrus pagrindinius sistemos konceptus ir jų

šąsąjas, PRR taisykles galima suformuluoti daug paprasčiau. Remiantis tyrimo rezultatais, sudaryta SBVR ir PRR standartų taikymo metodika, kurioje sugretinami šie iš pirmo žvilgsnio gana skirtingos struktūros standartai bei pateikiamos jų nuoseklaus naudojimo gairės.

## Išvados

Apžvelgus IS kūrimui pagal VT koncepciją aktualius OMG standartus, pagrindinis dėmesys buvo sutelktas į SBVR ir PRR. Šie standartai sąlyginai nauji, todėl detalių metodikų jiems nėra sukurta.

SBVR standartas yra platus, jame apibrėžiami tik VT šablonų elementai, o ne patys šablonai. Šį standartą patogų naudoti, jeigu norima nusakyti veiklos koncepciją natūraliąja, tačiau aiškiai struktūrizuota kalba. Vis tik automatiniam VT apdorojimui šį standartą labai sunku pritaikyti. Tuo tarpu griežtesniame PRR standarte įtrauka formali kalba PRR OCL, VT turi aiškia struktūrą. Tačiau specifikuojant veiklos taisykles šiuo standartu, bus užfiksuotos tik produkcinės VT, t. y., taisyklės, turinčios konkrečias sąlygas ir veiksmų sąrašus. Ankstyvosiose IS kūrimo stadijose, nustatant sistemos reikalavimus, šis standartas nelabai tinka, nes iš esmės yra orientuotas į MDA architektūros PIM lygmenį. Be to, apdorojant natūraliąja kalba išreikštus reikalavimus sunku iš karto išskirti sąlygos sakinius, nes sudėtinga aprėpti sistemos konceptų, sudarančių šias sąlygas, sąsąjas.



Pav. 2. Koncepcinė siūlomos SBVR ir PRR standartų taikymo metodikos schema

SBVR ir PRR standartus galima laikyti UML meta-modelio plėtiniais, skirtais VT atvaizduoti. Naudojantis SBVR galima analizuoti reikalavimus sistemai, išreikštus natūraliąja kalba, išskirti pagrindinius verslo konceptus. SBVR žodynais galima sujungti skirtingas kalbos bendruomenes, surinkti ir paaiškinti skirtingų sričių atstovų naudojamą žargoną. PRR standartu detalizuojamos konkrečios taisyklės, išreikštos sąlygos sakiniiais. Šių taisyklių vykdymą patogų automatizuoti. Atsižvelgiant į visą tai, pasiūlyta originali nuoseklaus abiejų standartų panaudojimo metodikos koncepcija, kuri bus toliau tiriama ir detalizuojama.

## Literatūros šaltiniai

- [1] **Business Rules Group.** *Defining Business Rules ~ What Are They Really? (3<sup>rd</sup> edition)* 2000.
- [2] **Business Rules Group.** *Veiklos taisyklių manifestas. Taisyklių nepriklausomumo principai, ver. 2.0.* (Redaktorius Ronald G. Ross). 2003.
- [3] **Butleris, R., Kapočius, K.** *Struktūrizuotų veiklos taisyklių saugyklos architektūra.* Informacijos mokslai. Vilnius. Vilniaus universiteto leidykla, 17 tomas, 46-57. 2001.
- [4] **Editors of BRCommunity.com.** *A Brief History of the Business Rule Approach, 2nd ed.* 2006.
- [5] **Nijssen S.** *SBVR: Semantics for Business.* Business Rules Journal, Vol. 8, No. 10. 2007.
- [6] **Object Management Group.** *Business Motivation Model (BMM) Specification.* 2007.
- [7] **Object Management Group.** *MDA Guide Version 1.0.1.* 2003.
- [8] **Object Management Group.** *Model Driven Architecture (MDA).* Dokumentas nr. ormsc/2001-07-01. 2001.
- [9] **Object Management Group.** *Model Driven Architecture (MDA) FAQ...* 2007.
- [10] **Object Management Group.** *Production Rule Representation (PRR), Beta 1.* 2007.
- [11] **Object Management Group.** *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 OMG Available Specification.* 2008.
- [12] **Siegel J.** *Making the Case: OMG's Model Driven Architecture.* SDTimes On The Web. 2002.