

KAUNO TECHNOLOGIJOS UNIVERSITETAS

EGLĖ MICKEVIČIŪTĖ

VEIKLOS ŽODYNŲ IR VEIKLOS TAISYKLIŲ  
SEMANTIKA GRINDŽIAMAS VEIKLOS  
PROCESŲ VAIZDAVIMAS

Daktaro disertacija  
Technologijos mokslai, informatikos inžinerija (07T)

2017, Kaunas

Disertacija rengta 2012–2016 metais Kauno technologijos universitete, Informatikos fakultete, Informacijos sistemų katedroje. Mokslinius tyrimus rėmė Valstybinis studijų fondas.

**Mokslinis vadovas:**

Prof. dr. Rimantas BUTLERIS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija – 07T).

Interneto svetainės, kurioje skelbiama disertacija, adresas:

<http://ktu.edu>

**Redagavo:**

Rozita Znamenskaitė (leidykla „Technologija“)

© E. Mickevičiūtė, 2017

ISBN 978-609-02-1405-3

Leidinio bibliografinė informacija pateikiama Lietuvos nacionalinės Martyno Mažvydo bibliotekos Nacionalinės bibliografijos duomenų banke (NBDB)

# TURINYS

TERMINAI IR SANTRUMPOS.....	6
PAVEIKSLAI.....	9
LENTELĖS .....	12
1 ĮVADAS.....	13
1.1 Motyvacija.....	13
1.2 Tyrimo sritis ir objektas .....	14
1.3 Problema ir keliami klausimai.....	14
1.4 Tyrimo tikslas ir uždaviniai.....	14
1.5 Tyrimų metodika .....	15
1.6 Ginami teiginiai.....	15
1.7 Darbo mokslinis naujumas .....	16
1.8 Darbo rezultatų teorinė ir praktinė svarba.....	16
1.9 Darbo rezultatų aprobavimas .....	16
1.10 Darbo apimtis ir struktūra .....	16
1.11 Disertacijos rezultatų naudojimas projekte, autorinis indėlis.....	17
2 VEIKLOS PROCESŲ IR VEIKLOS TAISYKLIŲ MODELIAVIMAS .....	18
2.1 Veiklos žodyno, veiklos taisyklių, veiklos procesų bei veiklos procesų modelavimo konceptai ir jų samprata .....	18
2.2 Veiklos procesų modelavimo būdai .....	19
2.3 Modelavimo kalbų vertinimo metodikos .....	20
2.4 Veiklos procesų modelavimo gairės .....	21
2.5 OMG standartas BPMN2 .....	23
2.6 OMG standartas UML.....	26
2.7 Veiklos modelavimo kalbų vertinimas: BPMN ir UML .....	27
2.8 Kompleksinis modelavimo standartų vertinimas .....	30
2.9 OMG standartas SBVR .....	31
2.10 Lietuvos autorių susijusių darbų apžvalga .....	34
2.11 Veiklos procesų ir veiklos taisyklių modelavimo analizės apibendrinimas ir išvados.....	34
3 VEIKLOS PROCESŲ IR VEIKLOS TAISYKLIŲ INTEGRACIJOS IR TRANSFORMACIJOS ANALIZĖ.....	36
3.1 Modelių integravimo metodai .....	37
3.2 Esamų integravimo sprendimų tyrimas.....	39
3.3 Esamų BPMN2 ir SBVR transformacijų tyrimas.....	42
3.4 Atlikto tyrimo apibendrinimas ir išvados.....	44
4 BPMN2 IR SBVR INTEGRACIJA IR TRANSFORMACIJOS .....	47
4.1 Principinė sprendimo schema ir kontekstas .....	47
4.2 BPMN2 ir SBVR integravimas .....	48
4.2.1 BPMN2 modelio ir SBVR veiklos žodyno konceptų atitikimas.....	48
4.2.2 SBVR profilis.....	53

4.2.3	Iliustracinis BPMN2 ir SBVR pavyzdys .....	54
4.3	BPMN2 veiklos procesų modelio transformavimas į SBVR veiklos žodyną ir veiklos taisykles .....	58
4.3.1	Bendrieji modelių transformavimo specifikavimo principai .....	58
4.3.2	BPMN2 veiklos procesų modelio transformacijos į SBVR veiklos žodyną ir veiklos taisykles matrica .....	59
4.3.3	BPMN2 veiklos procesų modelio transformacijos į SBVR veiklos žodyną ir veiklos taisykles taisyklės ir algoritmai .....	62
4.3.4	Reikalavimai BPMN2 VPM, nuo kurių priklauso transformacijų rezultatų kokybė .....	68
4.3.5	Papildomi BPMN2 įvykių vardų sudarymo šablonai .....	70
4.3.6	Apibendrinta geroji BPMN2 VPM modeliavimo praktika: korektiško BPMN2 VPM kriterijai .....	73
4.4	BPMN2 veiklos procesų modelio transformavimas į BPMN2 procesų modelio žodyną ir taisykles .....	74
4.4.1	BPMN2 veiklos procesų modelio transformacijos į BPMN2 veiklos procesų modelio žodyną ir taisykles matrica .....	74
4.4.2	BPMN2 veiklos procesų modelio transformacijos į BPMN2 veiklos procesų modelio žodyną ir taisykles taisyklės ir algoritmai .....	76
4.5	SBVR veiklos žodyno ir veiklos taisyklių transformacija į BPMN2 veiklos procesų modelį .....	80
4.5.1	SBVR veiklos žodyno ir veiklos taisyklių transformacija į BPMN2 veiklos procesų modelį .....	80
4.5.2	BPMN2 veiklos procesų modelio žodyno ir taisyklių transformacija į BPMN2 veiklos procesų modelį .....	84
4.5.3	SBVR dalinio transformavimo į BPMN2 veiklos procesų modelį realizavimo galimybės .....	85
4.6	Sukurtų transformacijos taisyklių korektiškumo tikrinimas .....	86
4.7	BPMN2 veiklos procesų modelio modeliavimo metodas: sukurtų transformacijų taikymas .....	87
4.8	BPMN2 ir SBVR integracijos ir transformacijos sukurto sprendimo apibendrinimas ir išvados .....	89
5	SBVR VEIKLOS ŽODYNO IR VEIKLOS TAISYKLIŲ IŠGAVIMO IŠ BPMN2 VEIKLOS PROCESŲ MODELIO REALIZACIJA .....	90
5.1	Prototipo aprašymas .....	90
5.2	Prototipo trumpas naudojimo aprašas .....	94
5.3	Atvejai, kurių neįvertina sukurtas sprendimas .....	95
5.4	Realizacijos apibendrinimas ir išvados .....	96
6	SBVR VŽ&VT IŠGAVIMO IŠ BPMN2 VPM EKSPERIMENTAS .....	98
6.1	Eksperimentinio tyrimo aprašas ir jo įvertinimo kriterijai .....	98
6.2	Eksperimentinio tyrimo vykdymas .....	98
6.3	Eksperimento rezultatų apibendrinimas .....	100
6.4	Grėsmės rezultatų pagrįstumui .....	101

7	IŠVADOS.....	102
8	LITERATŪRA.....	103
9	MOKSLINIŲ PUBLIKACIJŲ DISERTACIJOS TEMA SĄRAŠAS .....	114
10	PRIEDAI .....	117
10.1	BPMN VPM transformacijos į SBVR VŽ&VT taisyklės.....	117
10.2	BPMN2 konceptų metažodynas .....	139
10.3	Ekspirimente naudoti BPMN2 VPM.....	141
10.3.1	BPMN2 VPM, kurti netaikant gerosios modeliavimo praktikos .....	141
10.3.2	BPMN2 VPM, kurti taikant gerąją modeliavimo praktiką .....	143
10.4	<i>MagicDraw</i> CASE įrankio langai .....	152
10.5	BPMN2 ir SBVR naudojamų elementų vertimai .....	154

## TERMINAI IR SANTRUMPOS

<b>Santrumpa</b>	<b>Aprašymas</b>
BPEL	Veiklos procesų vykdymo kalba, kurioje taikomos žiniatinklio paslaugos (angl. <i>web services</i> ) (angl. <i>Business Process Execution Language</i> )
BPMN	Veiklos procesų modeliavimo notacija (angl. <i>Business Process Modeling Notation</i> )
BRG	Veiklos taisyklių grupė (angl. <i>Business Rules Group</i> )
BWW	Bunge-Wand-Weber ontologija
CIM	Nuo skaičiavimų nepriklausomas modelis (angl. <i>Computation-independent Model</i> )
MDA	Modeliais paremta architektūra (angl. <i>Model Driven Architecture</i> )
MOF	Metaobjektų bazė (angl. <i>The Meta-Object Facility</i> )
OMG	Technologinių standartų konsorciumas (angl. <i>Object Management Group</i> )
QVT	Modelių transformavimo kalba (angl. <i>Object Query/View/Transformation</i> )
R2ML	Taisyklių aprašomoji kalba, kuri leidžia dalyintis taisyklėmis tarp skirtingų sistemų (angl. <i>Rewerse Rule Markup Language</i> )
rBPMN	Taisyklėmis grindžiama veiklos procesų modeliavimo kalba (angl. <i>Rule-based Business Process Modeling Language</i> )
SBPVR	Deklaratyvių procesų metamodelis, kurį galima integruoti su SBVR taisyklių metamodeliu (angl. <i>Semantics of Business Process Vocabulary and Process Rules</i> )
SBVR	Veiklos žodyno ir veiklos taisyklių standartas (angl. <i>The Semantics of Business Vocabulary and Business Rules</i> )
SRML	Supaprastinta taisyklių žymėjimo kalba (angl. <i>Simple Rule Markup Language</i> )
SWRL	Semantinė tinklo taisyklių kalba (angl. <i>Semantic Web Rule Language</i> )
UML	Unifikuota modeliavimo kalba (angl. <i>Unified Modeling Language</i> )
VD	Veiklos diagrama
VPM	Veiklos procesų modelis (pvz., BPMN VPM)
VPM T	Procesų modelio taisyklės (pvz., BPMN VPM T)
VPM Ž	Procesų modelio žodynas (pvz., BPMN VPM Ž)
VT	Veiklos taisyklė (pvz., SBVR VT)
VŽ	Veiklos žodynas (pvz., SBVR VŽ)
VŽ&VT	Veiklos žodynas ir veiklos taisyklės (pvz., SBVR VŽ&VT)
W3C	Konsorciumas, leidžiantis programinės įrangos standartus

	žiniatinkliui (angl. <i>World Wide Web Consortium</i> )
Workflow	Darbų (užduočių) seka
XMI	Apsikeitimo metaduomenimis per XML standartas (angl. <i>XML Metadata Interchange</i> )
XML	W3C rekomenduojama bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba (angl. <i>Extensible Markup Language</i> )
Ž&T	Veiklos procesų modelio žodynas ir veiklos procesų modelio taisyklės (pvz., BPMN VPM Ž&T)

<b>Terminas</b>	<b>Apibrėžimas</b>
BPMN2 konceptų metažodynas	Žinių struktūra, pagal kurią identifikuojami pagrindiniai BPMN2 konceptai, skirti procesų modelio žodynų bei taisyklių aprašymui (SBVR žinių modelis išplėstas BPMN2 konceptais nekeičiant metamodelių).
Integracija	Procesas, kurio metu keli atskiri komponentai yra sujungiami į vieną (elementų lygiu), taip užtikrinant jų tarpusavio suderinamumą.
Transformacija	Vieno modelio elementų ir/arba jo ryšių pervedimas į kito modelio elementus ir/arba jo ryšius. Abu modeliai išreiškia vieną ir tą pačią modeliuojamą veiklos sritį tik vartojant skirtingas notacijas ir/arba taikant skirtingus modeliavimo principus.
Vaizdavimas	Modelio pateikimas tam tikra forma (pvz., deklaratyviai/imperatyviai, vartojant grafinę notaciją, struktūrizuotą natūralią kalbą ir kt.).
Veiklos procesas	Rinkinys tarpusavyje susietų veiklų, kurios apima verslo, IT, technines priemones tam, kad įgyvendintų konkretų organizacijos ar jos vieneto tikslą.
Veiklos proceso taisyklė	Loginis teiginys, nusakantis tam tikrus veiklos aspektus, kurie egzistuoja tam tikroje veiklos situacijoje ir kurie susiję su duomenų srautais bei veiklos procesų modelio hierarchija.
Veiklos proceso žodynas	Žinių struktūra, pagal kurią identifikuojami pagrindiniai dalykinės srities terminai ir jų tarpusavio ryšiai veiklos procesų modeliavimo kalbos elementų atžvilgiu.
Veiklos procesų modeliavimas	Veiklos procesų vaizdavimas ir simuliacijos naudojant specialiai tam skirtas priemones, kad būtų galima modeliuojamą procesą analizuoti ar patobulinti.
Veiklos procesų modelis	Struktūrinė forma išreiškta tam tikros srities veiklų seka, kuri parodo organizacijos ar jos vieneto vykdomą veiklą.
Veiklos taisyklė	Loginis teiginys, nusakantis tam tikrus veiklos apribojimus, kurie egzistuoja tam tikroje veiklos situacijoje.
Veiklos žodynas	Žinių struktūra, pagal kurią identifikuojami pagrindiniai dalykinės srities terminai ir jų tarpusavio ryšiai.



## PAVEIKSLAI

2.1 pav. Pagrindiniai BPMN procesų diagramos elementai, remiantis (OMG, 2013)	24
2.2 pav. Pagrindiniai UML veiklos diagramos elementai, remiantis (OMG, 2015).	27
2.3 pav. SBVR dalių tarpusavio ryšiai, remiantis (OMG, 2008; OMG, 2013b, OMG, 2013c)	32
2.4 pav. SBVR prasmės metamodelis, remiantis (OMG, 2013c)	34
3.1 pav. SBVR ir SBPVR modeliai, remiantis (Agrawal, 2011)	41
3.2. pav. Metamodelių integravimas naudojant sąsają, remiantis (Skersys et al., 2012a)	42
4.1 pav. BPMN2 ir SBVR integravimo ir transformavimo kontekstas	47
4.2 pav. SBVR profilis (Mickeviciute et al., 2014b)	54
4.3 pav. Proceso „Rent Car“ diagrama (pirmasis hierarchijos lygis)	54
4.4 pav. Subproceso „Book Car“ diagrama (antrasis hierarchijos lygis)	55
4.5 pav. Subproceso „Approve Car Booking Request“ modelis (trečiasis hierarchijos lygis)	55
4.6 pav. <i>Event1</i> inicijuoja <i>Event2</i> , pavyzdys	63
4.7 pav. Pavyzdys su sprendimo priėmimo tašku ( <i>EventBasedGateway</i> )	64
4.8 pav. <i>Activity1</i> inicijuoja <i>activity2</i> , pavyzdys	65
4.9 pav. <i>Throw Event</i> inicijuoja <i>MessageFlow</i> , pavyzdys	65
4.10 pav. SBVR VŽ formavimo iš BPMN2 veiklos vardo pavyzdys	66
4.11 pav. SBVR VŽ formavimo iš BPMN2 įvykio vardo pavyzdys	66
4.12 pav. Algoritmas BPMN2 VPM į SBVR VŽ transformuoti	67
4.13 pav. Algoritmas BPMN2 VPM į SBVR VT transformuoti	68
4.14 pav. Algoritmas BPMN2 VPM į BPMN2 VPM Ž&T transformuoti	79
4.15 pav. Algoritmas SBVR VŽ&VT į BPMN2 VPM transformuoti	83
4.16 pav. Algoritmas BPMN2 VPM Ž&T į BPMN2 VPM transformuoti	84
4.17 pav. D&D šablonas SBVR bendrinis konceptas į BPMN2 <i>Message</i>	85
4.18 pav. D&D šablonas SBVR veiksmožodinis konceptas į BPMN2 <i>DataObject</i> ir <i>Lane</i>	86
4.19 pav. Sukurtų transformacijos taisyklių korektiškumo tikrinimas: 1 ir 2 etapai	87
4.20 pav. BPMN2 VPM kūrimo ir SBVR VŽ&VT bei BPMN2 VPM Ž&T išgavimo scenarijus	88
4.21 pav. BPMN2 VPM kūrimo pritaikant SBVR VŽ ir SBVR VŽ&VT bei BPMN2 VPM Ž&T išgavimo scenarijus	88
5.1 pav. BPMN2 transformacijų į SBVR taikymas	90
5.2 pav. BPMN2 veiklos procesas „Approve Car Booking Request“	92
5.3 pav. SBVR veiklos žodynas	93
5.4 pav. SBVR veiklos taisyklės	93
5.5 pav. Komponentų diagrama	94
5.6 pav. Įskiepio iškvietimas	94
5.7 pav. Šablono struktūra	94
5.8 pav. Įskiepio parinkčių langas	95
5.9 pav. BPMN2 VPM struktūros keitimas norint panaikinti vienas po kito einančius	

sprendimo priėmimo taškus .....	96
5.10 pav. BPMN2 VPM struktūros keitimas norint pakeisti veiklos parametraž nusakantį ciklą .....	96
6.1 pav. Eksperimento algoritmas .....	99
10.1 pav. BPMN2 procesas „Book car“ .....	117
10.2 pav. BPMN2 procesas „Approve car booking request“ .....	117
10.3 pav. BPMN2 konteinerių (angl. <i>Pool Lane</i> ) pavyzdys .....	117
10.4 pav. Pavyzdys, kaip vienas įvykis inicijuoja kitą .....	122
10.5 pav. Pavyzdys, kai ribinis įvykis inicijuoja įvykį .....	123
10.6 pav. Pavyzdys, kai veikla inicijuoja įvykį .....	123
10.7 pav. Pavyzdys, kai veikla su sąlyginiu sekos srautu inicijuoja įvykį .....	124
10.8 pav. Pavyzdys, kai pranešimo srautas inicijuoja įvykį (angl. <i>Catch Event</i> ) ...	124
10.9 pav. Pavyzdys, kai pranešimo srautas inicijuoja įvykį (angl. <i>Catch Event</i> ) ...	125
10.10 pav. Pavyzdys, kai veikla inicijuoja įvykiais pagrįstą sprendimų priėmimų tašką (angl. <i>Event Based Gateway</i> ), kuris turi išeinančius sekos srautus su įvykiais.....	126
10.11 pav. Pavyzdys, kai veikla inicijuoja kompensuojamą veiklą.....	126
10.12 pav. Pavyzdys, kai veikla inicijuoja kitą veiklą.....	127
10.13 pav. Pavyzdys, kai veikla su išeinančiais sekos srautais su sąlygomis inicijuoja veiklas .....	128
10.14 pav. Pavyzdys, kai įvykis inicijuoja veiklą.....	128
10.15 pav. Pavyzdys, kai ribinis įvykis susietas su veikla sekos srautu .....	129
10.16 pav. Sprendimų priėmimo taško (angl. <i>ExclusiveGateway</i> ) pavyzdys .....	130
10.17 pav. Kompleksinio sprendimų priėmimo taško (angl. <i>ComplexGateway</i> ) pavyzdys .....	131
10.18 pav. Pavyzdys su apimančiu sprendimų priėmimo tašku (angl. <i>InclusiveGateway</i> ) .....	132
10.19 pav. Pavyzdys su lygiagrečiuoju sprendimų priėmimo tašku (angl. <i>ParallelGateway</i> ).....	133
10.20 pav. Pavyzdys, kai įvykis (angl. <i>ThrowEvent</i> ) inicijuoja pranešimo srautą (angl. <i>MessageFlow</i> ).....	133
10.21 pav. Pavyzdys, kai įvykis (angl. <i>ThrowEvent</i> ) inicijuoja pranešimo srautą su pranešimu (angl. <i>MessageFlow (with ref. Message)</i> ) .....	134
10.22 pav. Pavyzdys, kai veikla (angl. <i>Activity</i> ) inicijuoja išeinantį pranešimo srautą (angl. <i>MessageFlow</i> ).....	134
10.23 pav. Pavyzdys, kai veikla (angl. <i>Activity</i> ) inicijuoja išeinantį pranešimo srautą su pranešimu (angl. <i>MessageFlow (with ref. Message)</i> ).....	135
10.24 pav. Pavyzdys, kai įeinantis pranešimų srautas (angl. <i>MessageFlow</i> ) pasirodo vykdant veiklą (angl. <i>Activity</i> ) .....	136
10.25 pav. Pavyzdys, kai įeinantis pranešimų srautas su pranešimu (angl. <i>MessageFlow (with ref. Message)</i> ) pasirodo vykdant veiklą (angl. <i>Activity</i> ).	136
10.26 pav. Pavyzdys, kai pranešimo srautas (angl. <i>MessageFlow</i> ) jungia du konteinerius (angl. <i>Pool Lane</i> ).....	137
10.27 pav. Pavyzdys, kai pranešimo srautas su pranešimu (angl. <i>MessageFlow (with     ref. Message)</i> ) jungia du konteinerius (angl. <i>Pool Lane</i> ).....	138

10.28 pav. Proceso „Provide Book Loan“ modelis .....	141
10.29 pav. Proceso „Loan a Book“ modelis .....	141
10.30 pav. Proceso „Order Handling“ modelis .....	142
10.31 pav. Subproceso „Retrieve Required Parts“ modelis .....	142
10.32 pav. Subproceso „Fulfill Order“ modelis .....	142
10.33 pav. Subproceso „Part Auction“ modelis .....	143
10.34 pav. Proceso „Procurement Process“ modelis .....	143
10.35 pav. Pagal gerąją praktiką pataisytas veiklos proceso „Loan a Book“ modelis .....	143
10.36 pav. Pagal gerąją praktiką pataisytas veiklos proceso „Provide Book Loan“ modelis.....	144
10.37 pav. Pagal gerąją praktiką pataisytas veiklos proceso „Handle Order“ modelis .....	144
10.38 pav. Pagal gerąją praktiką pataisytas subproceso „Fulfill Order“ modelis...	145
10.39 pav. Pagal gerąją praktiką pataisytas subproceso „Retrieve Required Parts“ modelis.....	145
10.40 pav. Pagal gerąją praktiką pataisytas subproceso „carry_out part auction“ modelis.....	145
10.41 pav. Pagal gerąją praktiką pataisytas veiklos proceso „Procure Order“ modelis .....	145
10.42 pav. EU-Rent proceso „rent car“ (išsinuomoti automobilį) BPMN2 modelis (pirmasis hierarchijos lygis).....	146
10.43 pav. EU-Rent proceso „calculate rental charge“ (apskaičiuoti nuomos kainą) BPMN2 modelis (pirmasis hierarchijos lygis).....	146
10.44 pav. EU-Rent subproceso „book car“ (rezervuoti automobilį) BPMN2 modelis (antrasis hierarchijos lygis).....	146
10.45 pav. EU-Rent subproceso „provide car“ (pateikti automobilį) BPMN2 modelis (antrasis hierarchijos lygis).....	147
10.46 pav. EU-Rent subproceso „return car“ (grąžinti automobilį) BPMN2 modelis (antrasis hierarchijos lygis).....	147
10.47 pav. EU-Rent subproceso „book car“ trečiojo lygio subprocesas „approve car booking request“ (patvirtinti automobilio rezervaciją).....	148
10.48 pav. EU-Rent subproceso „book car“ trečiojo lygio subprocesas „create rental contract“ (sukurti nuomos sutartį) .....	148
10.49 pav. EU-Rent subproceso „book car“ trečiojo lygio subprocesas „confirm estimated rental charge“ (patvirtinti apskaičiuotą nuomos mokestį) .....	149
10.50 pav. EU-Rent subproceso „book car“ trečiojo lygio subprocesas „cancel car booking request“ (atšaukti automobilio rezervaciją) .....	149
10.51 pav. EU-Rent subproceso „provide car“ trečiojo lygio subprocesas „manage rental change“ (valdyti nuomos pasikeitimus).....	150
10.52 pav. EU-Rent subproceso „provide car“ trečiojo lygio subprocesas „manage rental car for pick-up“ (paruošti išnuomotą automobilį (klientui)).....	150
10.53 pav. EU-Rent subproceso „provide car“ trečiojo lygio subprocesas „manage incident of rental car“ (tvarkyti išnuomoto automobilio incidentą).....	151
10.54 pav. SBVR žodyno diagrama .....	152

10.55 pav. SBVR veiklos taisyklės „Generic Table“ lentelėje.....	152
10.56 pav. BPMN2 veiklos proceso kūrimas naudojant SBVR veiklos žodyną ....	153
10.57 pav. SBVR veiklos žodynas CASE įrankyje <i>MagicDraw</i> .....	153

## LENTELĖS

2.1 lentelė. Septynios veiklos procesų modeliavimo gairės, remiantis (Mendling et al., 2010b) .....	22
2.2 lentelė. BPMN diagramų tipai ir jų aprašymai .....	25
2.3 lentelė. Veiklos procesų standartų analizės lentelė pagal pagrindinius vertinimo kriterijus .....	28
2.4 lentelė. Veiklos procesų standartų analizės lentelė pagal išskirtus kriterijus .....	29
2.5 lentelė. Modeliavimo kalbų vaizdavimo galimybių tyrimo rezultatai (zur Meuhlen et al., 2007) .....	30
2. 6 lentelė. SBVR taisyklių tipai .....	33
3.1 lentelė. Esamų sprendimų analizė .....	45
4.1 lentelė. BPMN2 modelio ir SBVR veiklos žodyno konceptų atitikimas.....	48
4.2 lentelė. BPMN2 modelio elementų ir SBVR konceptų atitikimas taikant BPMN2 konceptų metažodyną .....	52
4.3 lentelė. SBVR veiklos žodynas ir taisyklės pagal 4.5 pav. BPMN2 veiklos procesų diagramą .....	55
4.4 lentelė. BPMN2 procesų modelio žodynas ir taisyklės pagal 4.5 pav. BPMN2 veiklos procesų diagramą.....	57
4.5 lentelė. BPMN2 VPM transformacijų į SBVR VŽ&VT matrica .....	61
4.6 lentelė. BPMN2 VPM transformacijos į SBVR VŽ&VT taisyklių fragmentas. ....	62
4.7 lentelė. BPMN2 VPM į SBVR VŽ naudojami veiksmožodžiai .....	66
4.8 lentelė. Apibrėžtą gerąją praktiką atitinkančios BPMN2 elementų pavadinimų struktūros .....	70
4.9 lentelė. Įvykių priskyrimo atitinkamai grupei pagal vardus lentelė.....	70
4.10 lentelė. BPMN2 įvykių vardų šablonai.....	71
4.11 lentelė. Įvykių vardų pavyzdžiai.....	72
4.12 lentelė. BPMN2 VPM transformacijų į BPMN2 VPM Ž&T matrica .....	74
4.13 lentelė. BPMN2 VPM transformacijų į BPMN2 VPM Ž&T taisyklių sudarymo principai .....	76
4.14 lentelė. BPMN2 VPM į BPMN2 VPM Ž naudojami veiksmožodžiai .....	79
4.15 lentelė. SBVR VŽ&VT transformacijos į BPMN2 matrica .....	81
6.1 lentelė. Eksperimento rezultatų apibendrinimas (netaikant gerosios praktikos).....	100
6.2 lentelė. Eksperimento rezultatų apibendrinimas (taikant gerąją praktiką).....	100
10.1 lentelė. BPMN2 elementų grupės su vertimais į lietuvių kalbą.....	154
10.2 lentelė. BPMN2 elementai su vertimais į lietuvių kalbą .....	154
10.3 lentelė. SBVR elementai su vertimais į lietuvių kalbą .....	154
10.4 lentelė. BPMN2 elementai su vertimais į lietuvių kalbą .....	155
10.5 lentelė. BPMN2 įvykiai su vertimais į lietuvių kalbą.....	155
10.6 lentelė. BPMN2 laiko įvykių parametrai su vertimais į lietuvių kalbą.....	155

# 1 ĮVADAS

## 1.1 Motyvacija

Informacinių sistemų kūrimas prasideda nuo veiklos procesų modeliavimo ir veiklos žodyno bei taisyklių identifikavimo. Daugeliu atvejų sunkumai atliekant reikalavimų analizę ir modeliuojant veiklos procesus kyla dėl nevisiškai išsiaiškintos veiklos logikos, kuri apibrėžiama veiklos taisyklėmis ir veiklos procesais (Andrescu, Mircea, 2014). Norint turėti pilną ir išsamią veiklos procesų modelio specifikaciją, tikslinga taikyti kelis veiklos modeliavimo standartus, kuriuose nagrinėjami skirtingi veiklos aspektai ir kartu suteikiama galimybė bet kuriam veiklos dalyviui pasirinkti tokį veiklos proceso vaizdavimo būdą, kuris yra jam priimtinausias. Pasirinktas priimtinesnis proceso vaizdavimo būdas gali paspartinti bei pagerinti komunikaciją tarp veiklos dalyvių, padėti jiems išvengti nesusikalbėjimų dėl neteisingų interpretacijų ar terminų vartojimo; ypač tais atvejais, kai modelis yra didelis ir dėl to sunkiai skaitomas.

Veiklos procesų modeliavimas yra svarbus etapas informacinių sistemų kūrimo procese (Vasconcelos et al., 2001; Sinogas et al., 2001; Barjis, 2008; Kovalova, Turcok, 2014) ir būtinas norint suprasti, valdyti, optimizuoti veiklos procesus, įtraukti naujas arba atskirti esamas veiklos proceso veiklas ar jos dalis. Veiklos žodynas ir veiklos taisyklės yra kiekvieno veiklos procesų modelio dalis, kuri daro įtaką visoms jos sritims (Vasilecas, Lebedys, 2006), ir vargu ar egzistuoja nors viena veiklos sistema, kuri neturi tikslo, misijos bei veiklos taisyklių (Lovrencic et al., 2006). Kadangi veiklos procesų modeliavimas ir veiklos taisyklių modeliavimas yra laikomi vienas kitą papildančiais metodais – atsiranda poreikis šiuos du modelius taikyti kartu. Veiklos žodyno ir veiklos taisyklių taikymas kuriant veiklos procesų modelius naudingas ir tuo, kad organizacija gali gauti vieningą veiklos žodyną, t. y. terminus, kurie vartojami žodine ar rašytine forma organizacijos veikloje ir yra priimti kaip veiklos konceptai. Toks žodynas suteikia galimybę pagerinti veiklos suvokimą ir išvengti nesusipratimų tarp dalykinės srities atstovų bei IT specialistų. Be to, būtent veiklos žodynas suteikia galimybę sujungti dinامينius (proceso elgsena) ir statinius (apribojimai, informaciniai šrautai) proceso aspektus.

Palaiykite vientisumą tarp atskirai sumodeliuoto veiklos proceso ir atskirai aprašytų veiklos taisyklių nėra paprasta: nuolat kinta reikalavimai, atsiranda naujų kintamųjų, naujų sąvokų ar terminų – tokiu atveju papildant minėtus modelius naujomis žiniomis kyla rizika atsirasti prieštaravimams, nesutapimams ar kitoms spragoms. Norint veiklos procesų bei veiklos taisyklių modelius susieti tarpusavyje kyla sunkumų, kadangi veiklos procesams modeliuoti yra naudojamos grafinės notacijos, o žodynas ir veiklos taisyklės yra sudaromos naudojant ribotą natūralią kalbą.

Darbe yra siekiama integruoti veiklos procesų ir veiklos taisyklių modeliavimo standartus išsamiai bei darniai veiklos procesų specifikacijai gauti. Sprendimas pagrįstas BPMN2 ir SBVR modeliavimo standartais išsiaiškinus, kad kompleksinis jų taikymas leidžia geriausiai perteikti realaus pasaulio objektus bei procesus. Dviejų modelių suderinamumui užtikrinti realizuota galimybė transformuoti grafinius BPMN2 veiklos procesų modelius į SBVR veiklos žodynus ir veiklos taisykles su

galimybe susieti jų elementus vieno *MagicDraw* CASE įrankio aplinkoje.

## 1.2 Tyrimo sritis ir objektas

**Tyrimo sritis.** Veiklos procesų ir veiklos taisyklių modeliai, vaizdavimo metodai, kalbos, įrankiai.

**Tyrimo objektas.** Veiklos procesų ir veiklos taisyklių specifikuojimo, vaizdavimo ir susiejimo su taikomosios programinės įrangos modeliais procesas, apimantis visą veiklos procesų ir veiklos taisyklių gyvavimo ciklą.

## 1.3 Problema ir keliami klausimai

Darbe yra sprendžiama veiklos procesų, veiklos žodynų bei veiklos taisyklių susiejimo ir dalykinės srities ekspertams suprantamo veiklos aprašymo, kuris atitiktų modelių formalius aprašymus, problema. Veiklos procesų modeliavimas ir veiklos taisyklių modeliavimas yra laikomi vienas kitą papildančiais metodais, jų taikymas kartu modeliuojant sudėtingus veiklos procesus leidžia apimti skirtingus veiklos aspektus (dinaminius, kurie nusako proceso elgseną, ir statinius, kurie parodo apribojimus bei informacijos srautus) ir įtraukti visus su veikla susijusius dalyvius. Atsiranda suderinamumo tarp modelių problema, nes veiklos procesai paprastai išreiškiami grafiniu būdu, o veiklos taisyklės – deklaratyviu būdu, todėl šių dviejų modeliavimo metodų taikymas kartu, jų suderinamumas, veiklos žodyno bei veiklos taisyklių išgavimas iš grafinių veiklos procesų modelių, norint įtraukti dalykinės srities ekspertus į projektavimo etapą, yra plačiai nagrinėjamos temos. Atsiranda poreikis turėti darnias, aiškias veiklos procesų ir veiklos taisyklių specifikacijas, kurios suteiktų galimybę suprasti bei patikrinti veiklos procesus ir veiklos taisykles, pagerintų veiklos dalyvių tarpusavio susikalbėjimą organizacijos ar kelių organizacijų mastu.

Darbas turi atsakyti į šiuos klausimus:

1. Ar įmanoma grafinius veiklos procesų modelius išreikšti deklaratyviu būdu, naudojant ribotą natūralią kalbą?
2. Ko reikia norint gauti veiklos žodynus bei veiklos taisykles iš grafinių veiklos procesų modelių netaikant lingvistinių metodų?
3. Kokios grafinių veiklos procesų modelių savybės gali padėti pagerinti veiklos žodynų bei veiklos taisyklių korektišką išgavimą?
4. Ko reikia norint atlikti abipusę transformaciją tarp veiklos procesų ir veiklos žodynų bei veiklos taisyklių modelių?
5. Ar galima realizuoti veiklos žodynų bei veiklos taisyklių išgavimą iš grafinių veiklos procesų modelių su galimybe susieti šių modelių elementus vienoje modeliavimo aplinkoje?

## 1.4 Tyrimo tikslas ir uždaviniai

**Tyrimo tikslas.** Sudaryti sąlygas kurti integruotus veiklos procesų ir taisyklių modelius veiklos dalyviams suprantama kalba, sukuriant metodą ir priemones, kurios leistų suformuoti išsamią bei darnią veiklos procesų specifikaciją (grafinį VPM ir VŽ&VT) vieno įrankio aplinkoje.

### **Uždaviniai:**

1. Išanalizuoti literatūrą, susijusią su veiklos procesų, veiklos žodynų ir veiklos taisyklių modeliavimo būdais, kalbomis, metamodeliais, modeliavimo kalbų vertinimo metodikomis, integracija, transformacija bei esamais sprendimais.
2. Sukurti metodą, kuris leistų išreikšti grafinius procesų modelius veiklos žodyno konceptais bei veiklos taisyklėmis, ir tam aprašyti transformavimo taisykles bei algoritmus, taip pat įvertinant atvirkštinės transformacijos galimybes.
3. Realizuoti grafinių procesų modelių transformavimo į veiklos žodynus ir veiklos taisykles komponento prototipą CASE įrankio aplinkoje.
4. Atlikti eksperimentą tyrimo rezultatų tinkamumui įvertinti.

### **1.5 Tyrimų metodika**

Išanalizuoti esami veiklos procesų, veiklos žodynų ir taisyklių sudarymo bei vaizdavimo metodai, kalbos, metamodeliai, modeliavimo įrankiai. Analizei atlikti pritaikyti informacijos paieškos, analizės, sisteminimo, lyginamosios analizės, struktūrinės analizės bei apibendrinimo metodai.

Veiklos žodyno ir veiklos taisyklių suderinamumui bei integracijai su veiklos procesų modeliu taikyti konceptualaus modeliavimo, sisteminės procesų analizės, sistemų procesų inžinerijos bei modeliavimo metodai. Veiklos procesų modelio transformacijai į veiklos žodyną ir veiklos taisykles taikytas konceptualus taisyklių sudarymo metodas taikant EBNF notaciją.

Eksperimentiniams tyrimams atlikti taikytas disertacijos tyrimams sukurtas prototipinis įrankis, transformuojantis veiklos procesus į veiklos žodyną ir veiklos taisykles pagal sukurtas transformavimo taisykles. Eksperimentinio tyrimo rezultatams gauti buvo taikomi kiekybinio vertinimo, ekspertinio vertinimo, apibendrinimo bei lyginamosios analizės metodai.

Tyrimas buvo vykdomas pagal „Design Science“ metodą (Hevner et al., 2004): atlikta egzistuojančių sprendimų analizė, sukurti nauji artefaktai (nagrinėjamų modelių transformacijos taisyklės), atliktas eksperimentinis tyrimas su sukurtu prototipu, apibendrinti gauti rezultatai.

### **1.6 Ginami teiginiai**

Disertacijoje ginami teiginiai yra tokie.

1. Aprašytas išsamus BPMN2 ir SBVR metamodelių elementų atitikimas leidžia sukurti VPM transformacijos į VŽ&VT taisykles, leidžiančias automatiškai išgauti VŽ&VT iš VPM.
2. Sukurtas metodas leidžia suformuoti išsamią bei darnią specifikaciją, susidedančią iš VPM ir VŽ&VT modelių pagal du skirtingus scenarijus.
3. Apibrėžti reikalavimai VPM elementams leidžia automatinės transformacijos metu netaikant lingvistinių metodų gauti semantiškai korektiškus bei SBVR standartą atitinkančius veiklos žodyną ir veiklos taisykles.

4. Transformavimo algoritmas leidžia automatiškai iš VPM išgauti VŽ&VT, tačiau atvirkštinės transformacijos vykdymui vien tik VŽ&VT nepakanka, būtina naudoti du papildomus žodynus BPMN2 modelio informacijai išsaugoti.

### 1.7 Darbo mokslinis naujumas

Disertacijoje pasiūlytas metodas suteikia galimybę automatiškai gauti vientisą, išsamų, tarpusavyje suderintą veiklos žodyną, veiklos taisyklių ir veiklos procesų modelio taisyklių aprašymą, susietą su grafiniu veiklos procesų modeliu tam netaikant jokių lingvistinių metodų. Tai pasiekama naudojant aprašytą BPMN2 konceptų metažodyną, taikant sukurtas transformavimo taisykles bei algoritmus ir veiklos procesų modelių sudarymo reikalavimus.

### 1.8 Darbo rezultatų teorinė ir praktinė svarba

Sukurtas metodas, teikiantis galimybę kurti išsamesnius ir tikslesnius tarpusavyje suderintus veiklos procesų bei taisyklių modelius. Šiam metodui patikrinti sukurtas prototipinis įskiepis *MagicDraw* CASE įrankyje.

**Teorinė svarba:** ištirtos veiklos procesų, veiklos žodyno ir veiklos taisyklių integravimo bei transformavimo galimybės siekiant susieti veiklos procesų, veiklos žodyno ir taisyklių modelius vienoje modeliavimo aplinkoje; atskirtos veiklos taisyklės nuo veiklos procesų modelio taisyklių; apibrėžti reikalavimai veiklos procesų modelio elementų vardams; sukurtos transformavimo iš veiklos procesų modelio į veiklos žodyną bei veiklos taisykles taisyklės; teoriškai išnagrinėtos bei aprašytos atvirkštinės transformacijos iš SBVR veiklos žodyno ir veiklos taisyklių į BPMN2 veiklos procesų modelį, tam papildomai naudojant BPMN2 veiklos procesų modelio žodyną ir taisykles, kurios sudaromos naudojant BPMN2 konceptų metažodyną; sudarytos gerosios veiklos procesų modeliavimo gairės.

**Praktinė svarba:** galimybė automatinio būdu transformuoti veiklos procesų modelį į veiklos žodyną bei veiklos taisykles ir gautą modelį pateikti veiklos dalyviams validuoti; galimybė papildomai automatinio transformavimo metu iš veiklos procesų modelio gauti veiklos procesų modelio žodyną ir taisykles atvirkštinės transformacijos galimybei įgyvendinti, kad būtų išsaugota visa veiklos procesų modelio informacija; galimybė turėti du modelius, pagal kuriuos nagrinėjami skirtingi veiklos aspektai, vienoje aplinkoje, ir kurių elementai yra tarpusavyje susieti, taip sumažinant neatitikimų galimybę; veiklos žodyno elementus panaudoti keičiant/papildant veiklos procesų modelį; suformuoti vieningą organizacijos veiklos žodyną. Tokių sprendimų iki šiol nėra realizuotų CASE įrankiuose.

### 1.9 Darbo rezultatų aprobavimas

Darbo rezultatai buvo aprobuoti septyniose publikacijose. Mokslinių publikacijų sąrašas disertacijos tema pateiktas 9 skyriuje.

### 1.10 Darbo apimtis ir struktūra

Disertaciją sudaro įvadas, penki skyriai, išvados, literatūros sąrašas, mokslinių publikacijų disertacijos tema sąrašas, priedai. Disertacijos tekstą sudaro 104 puslapiai



aiškinamojo teksto ir 39 puslapiai priedų, pateikti 95 paveikslai, 30 lentelių. Literatūros sąrašą sudaro 154 literatūros šaltiniai, visi šaltiniai yra užsienio kalba.

### **1.11 Disertacijos rezultatų naudojimas projekte, autorinis indėlis**

Disertacijos tematika atlikti integravimo bei transformavimo galimybių tyrimai ir kiti susiję su šiomis temomis darbai buvo atlikti vykdant projektą „Veiklos procesų ir veiklos taisyklių integravimas veiklos semantikos pagrindu“ (VP131V008F) 2013–2015 metais (kitai dar vadinamas VEPSEM). SBVR profilis, transformacijos matricų, transformacijų taisyklių vaizdavimo formatai ir vartojamos notacijos bei atsekamumo ryšių kūrimo idėja yra perimta iš VEPSEM projekto.

Visų pateiktų transformacijų matricų turinys, aprašytos taisyklės, sudarytas BPMN2 konceptų metažodynas, modeliavimo metodas taikant šiame darbe sukurtą sprendimą, transformacijų iš BPMN2 veiklos procesų modelių į SBVR veiklos žodynus ir veiklos taisykles realizacija pritaikant aprašytą SBVR profilį bei sudaryta geroji VPM modeliavimo praktika yra laikomi autoriniais sprendimais.

Sukurtas sprendimas pritaikytas anglų kalbai, kadangi ji yra viena iš labiausiai paplitusių kalbų pasaulyje (įvertinant ir techninės literatūros paplitimą), be to, sprendimas realizuotas plačiai naudojamame *MagicDraw* CASE įrankyje, kadangi Lietuva panašių įrankių neturi. Sukurto sprendimo taikymas lietuvių kalbai susiaurintų jo pritaikomumą.

## 2 VEIKLOS PROCESŲ IR VEIKLOS TAISYKLIŲ MODELIAVIMAS

Skyriaus pradžioje pateikiami veiklos žodyno, veiklos taisyklių, veiklos procesų bei veiklos procesų modeliavimo konceptai ir jų samprata. Nagrinėjami veiklos procesų modeliavimo būdai bei jų kalbų vertinimas norint išsiaiškinti, kokia veiklos procesų ir veiklos taisyklių modeliavimo pora gali geriausiai pavaizduoti realaus pasaulio objektus bei procesus. Pasirinkta pora yra naudojama kuriant bei realizuojant sprendimą.

### 2.1 Veiklos žodyno, veiklos taisyklių, veiklos procesų bei veiklos procesų modeliavimo konceptai ir jų samprata

Veiklos žodynas yra organizacijos žinių struktūra (Ross, 2003), kuri apima bendrai vartojamus konceptų terminus ir jų apibrėžimus (Maldonado et al., 2013), pritaikomus kalbine ar rašytine forma tam tikroje organizacijos veiklos srityje.

Veiklos taisyklės yra teiginiai, kurie daro įtaką informacijos/sistemos elgsenai organizacijoje (BRG, 2002) bei reglamentuoja organizacijos veiklos proceso struktūrą (Joubert et al., 2013); jos išreiškia specialias žinias, procedūras, t. y. įvairius apribojimus, į kuriuos turi atsižvelgti kiekviena organizacija (Andreescu, Mircea, 2009); jos turi būti aiškiai suprantamos ir laikomos vertingu organizacijos turtu, nes nusako organizacijos veiklos esmę (Steinke, Nickolette, 2003).

Veiklos procesas yra tarpusavyje susietų veiklų rinkinys, kurio struktūra nurodo loginę atskirų veiklų vykdymo tvarką bei jų tarpusavio priklausomybes (Aguilar-Saven, 2004) ir kurio tikslas yra nukreiptas į norimą gauti rezultatą. Veiklos procesai yra kiekvienos organizacijos pagrindas ir tiesiogiai siejami su organizacijos sėkmingu funkcionavimu (Rajabi, Lee, 2009), todėl labai svarbu stebėti ir reaguoti į veiklos pasikeitimus. Daugelis procesų yra dinaminiai, ir būtina reaguoti į pasikeitimus netgi vykdymo metu. Informacinės technologijos (IT) yra glaudžiai susijusios su organizacijoje vykdomais veiklos procesais. Per pastaruosius du dešimtmečius rinkos pokyčiai lėmė nuolatinę organizacijų veiklos kaitą (Nurcan, 2008), kuri nuolat tobulinama stengiantis prisitaikyti prie klientų poreikių, patenkinant jų lūkesčius ir konkuruojant rinkoje.

Veiklos procesų modeliavimas (angl. *Business Process Modeling*) yra pirmasis ir svarbiausias žingsnis veiklos procesų valdymo gyvavimo (angl. *Business Process Management*) cikle (Rajabi, Lee, 2009), kuriuo atskiriama procesų logika nuo taikymo logikos, tokiu atveju pagrindiniai veiklos procesai gali būti automatizuoti (Sadiq, Orłowska, 1997). Paprastai procesų logika yra kuriama ir valdoma taikant veiklos procesų valdymo sistemas, o taikymo logika – naudojant pagrindinius taikymo komponentus (Lu, Sadiq, 2007). Veiklos procesų modeliavimo gyvavimo ciklas susideda iš keturių pagrindinių stadijų: procesų kūrimo (angl. *process design*), sistemos konfigūracijos (angl. *system configuration*), procesų vykdymo (angl. *process enactment*) ir įvertinimo (angl. *diagnosis*) (Van Der Aalst et al., 2003).

Veiklos procesų modeliavimas leidžia suformuoti bendrą nagrinėjamos veiklos procesų supratimą, taip pat jį analizuoti formuojant išsamų supratimą apie nagrinėjamus veiklos procesus. Veiklos procesų modeliai turi labai svarbią paskirtį – jie leidžia rasti bendrą kalbą srities ekspertams, analitikams bei programinės įrangos

kūrėjams (Figl, Laue, 2011), tačiau tyrimai (Reijers, Mendling, 2011; Rodrigues et al., 2015) rodo, kad grafinius veiklos procesus geriau supranta analitikai, kurie turi patirties su modeliavimo kalbomis.

## 2.2 Veiklos procesų modeliavimo būdai

Egzistuoja du veiklos procesų modeliavimo būdai: deklaratyvusis ir imperatyvusis (dar kitaip vadinamas procedūrinis). Deklaratyvusis modeliavimas apibūdina procesą kaip proceso elementų ir deklaratyvių teiginių rinkinį, kurie perteikia tam tikrus reikalavimus/apribojimus, o imperatyvusis modeliavimas apibrėžia procesą kaip tam tikrą proceso elementų seką (Agrawal, 2011). Imperatyvusis modeliavimas pasižymi tuo, kad jame yra pateikiami visi galimi veiklos proceso keliai (Fahland et al., 2009a; Pichler et al., 2012). Deklaratyvi kalba nusako, „kas“ yra atliekama, tačiau nepateikia „kaip“ (tai nusako imperatyvi kalba) (Pesic, van der Aalst, 2006; Fahland et al., 2009b). Kitaip tariant – imperatyvi veiklos procesų modeliavimo kalba akcentuoja nuolatinės procesų objektų kaitos aspektą: kiekvieno objekto, kuris dalyvauja procese, gyvavimas gali būti apibūdinamas pagal būsenas ir jų kaitą, o deklaratyvi veiklos procesų modeliavimo kalba orientuota į logiką, kuri išreiškiama objektais ir jų sąveikomis.

Deklaratyvios modeliavimo kalbos yra pagrįstos formalių taisyklių sudarymu (angl. *rule based formalism*), o imperatyviojo modeliavimo – grafais paremtu formalizmo taikymu (angl. *graph based formalism*) (Rajabi, Lee, 2009; Lu, Sadiq, 2007). Grafais paremtose veiklos modeliavimo kalbose naudojama grafinė notacija veiklos procesų modeliams pavaizduoti. Grafinių notacijų atsiradimą lėmė siekis procesų modeliavimą padaryti suprantamą tiek IT analitikams, tiek organizacijos atstovams (Ottensooser et al., 2012). Atsiradus grafiniam veiklos procesų modeliavimui organizacijų atstovai gavo galimybę patys identifikuoti ir apibrėžti savo veiklos procesus (Caetano et al., 2012).

Grafais pagrįstos veiklos modeliavimo kalbos yra patrauklios tuo, kad procesai vaizduojami grafiniu būdu, jos intuityvios ir aiškios netgi tiems, kas neturi ar turi labai mažai žinių IT srityje; o štai taisyklėmis pagrįstoms veiklos modeliavimo kalboms reikia gero pateiktos logikos bei loginių išraiškų sintaksės supratimo, taip pat jos mažiau patrauklios praktiškumo požiūriu (Rajabi, Lee, 2009). Imperatyviojo veiklos procesų modeliavimo kalbos gali būti Petri tinklai (tai grafais pagrįstas būdas modeliuoti procesams, iš kurio ir atsirado grafinės veiklos modeliavimo kalbos), spalvoti Petri tinklai (angl. *colored Petri nets*), srautu (angl. *flow*) pagrįstos modeliavimo kalbos, tokios kaip UML veiklos diagramos ar BPMN. Tačiau atsiradusiems tokiems standartams kaip SBVR nereikia sudėtingos loginių išraiškų sintaksės supratimo, todėl jų modeliai gali būti skaitomi ir neturint atitinkamų IT žinių.

Atlikti tyrimai parodė, kad deklaratyvusis veiklos procesų modeliavimas pasižymi didesniu lankstumu (Goedertier, Vanthienen, 2008), tačiau toks modeliavimo būdas yra sunkiau suprantamas, o tai gali sudaryti sąlygas sistemų palaikymo problemoms atsirasti (Haisjackl, Zugall, 2014). Atliktas tyrimas parodė, kad imperatyvios modeliavimo kalbos yra lengviau suprantamos negu deklaratyviojo modeliavimo kalbos (Pichler et al., 2012), o modelių suprantamumas yra vienas

pagrindinių faktorių norint sukurti korektiškai funkcionuojančią sistemą su kuo mažiau resursų. Taip pat galima pastebėti, kad atliktame tyrime (Rodrigues et al., 2015), kuriuo buvo siekiama išsiaiškinti, kaip pateikiamas veiklos procesas (grafiniu būdu ar tekstu) yra geriau suprantamas, buvo nustatyta, kad pateikimo būdas nėra svarbiausia – didžiausią įtaką darė asmens kompetencija.

Kadangi imperatyvusis modeliavimas yra labiau įprastas, toks grafinis veiklos procesų modeliavimas yra dažniau sutinkamas modeliuojant organizacijų veiklos procesus. Tačiau nereikia pamiršti, kad imperatyvusis modeliavimas turi daugiau ar mažiau ir deklaratyviojo modeliavimo savybių (Fahland et al., 2009a). Taigi modeliuojant grafiniu būdu visos procesų veiklos savyje turi talpinti visus įmanomus veiklos scenarijus (Lu, Sadiq, 2007; Koehler, 2011), todėl grafinė veiklos proceso prezentacija tampa labai didelė ir sunkiai suprantama. Sukurtas DMN (angl. *Decision Model and Notation*) standartas (OMG, 2016) suteikia galimybę atskirai modeliuoti sprendimų priėmimo modelius: BPMN standartas turi specialų veiklos tipą (angl. *Business Rule Task*) šiems modeliams išreikšti, kuris leidžia dalį veiklos proceso aprašyti taisyklėmis, sprendimų priėmimo lentelėmis. Toks būdas leidžia ne tik sumažinti paties veiklos procesų modelio apimtį, bet ir atskirti sudėtingas veiklos taisykles, kurios apibrėžia sudėtingus sprendimų priėmimus. Hibridinis procesų vaizdavimas (BPMN-D) buvo pasiūlytas (De Giacomo et al., 2015), kuris leidžia BPMN modelį praplėsti deklaratyviomis konstrukcijomis, tačiau toks vaizdavimo būdas nėra tinkamas, jei su modeliais dirba specialių IT bei matematinių žinių neturintys veiklos dalyviai dėl čia taikomų linijinio laiko logikos formulių (LTL (angl. *Linear Temporal Logic*)).

Taisyklėmis pagrįstos veiklos modeliavimo kalbos siejamos su veiklos taisyklių valdymo sistemomis, kurių bendras tikslas yra integruoti sudėtingą procesų logiką į procesų modelį kaip taisykles, kurios gali reaguoti į dinaminis pasikeitimus (Rajabi, Lee, 2009; Lu, Sadiq, 2007). Veiklos procesai, kurie vyksta organizacijose, yra dinaminiai, o ribojimai, kurie kartais gali turėti daug kintamųjų bei sudėtingą logiką, – statiniai. Sudėtinga procesų logika gali būti išreiškiama atskirai (pvz., naudojant DMN), už kurios vykdymą būtų atsakingi tam skirti mechanizmai.

### **2.3 Modeliavimo kalbų vertinimo metodikos**

Pastaraisiais dešimtmečiais buvo pasiūlytos keletas veiklos procesų modeliavimo kalbų. Veiklos procesų modelis yra svarbus norint žinoti, kas ir kaip vyksta organizacijoje, todėl veiklos procesų modeliavimo kalbos pasirinkimas yra viena iš sunkesnių ir atsakingesnių užduočių, kadangi vieną kartą priėmus sprendimą, jo keitimas gali labai padidinti išlaidas, kurios susijusios su organizacijos veiklos procesų kūrimu bei palaikymu.

Veiklos modeliavimo kalbų vertinimo metodai gali būti skirstomi į empirinius ir neempirinius (Recker, 2005). Empiriniai modeliavimo kalbų vertinimo metodai yra tokie.

- Apklausa. Tyrimo metu pasitelkiamos apklausos siekiant įvertinti vartotojų nuomonę, požiūrį dėl apklausoje vertinamų modeliavimo kalbų.

- Laboratorinis eksperimentas. Tyrimo metu kelios modeliavimo kalbos vertinamos atliekant modeliavimą, siekiant išmatuoti įvairias charakteristikas, kaip laikas, išlaidos ir pan.
- Srities eksperimentas. Tyrimas atliekamas konkrečioje organizacijoje siekiant įvertinti pagrindinius rodiklius bei galimybę išlaikyti kontrolę tarp jų.
- Atvejo analizė. Tyrimas atliekamas su konkrečia grupe arba subjektu, kuris naudoja tiriamą modeliavimo kalbą.
- Veiksmo tyrimas. Atliekamas simuliuojant bei testuojant sumodeliuotus procesus, kurie buvo sukurti akademinės bendruomenės modeliuojant realaus pasaulio procesus.

Neempiriniai modeliavimo kalbų metodai yra tokie.

- Savybių palyginimas. Tyrimas atliekamas modeliuojant tos pačios dalykinės srities procesus tiriamomis modeliavimo kalbomis, siekiant įvertinti, kaip skirtingos modeliavimo kalbos pagal įvairius kriterijus padeda išspręsti tas pačias problemas.
- Metamodeliavimas. Tyrimas atliekamas taikant modeliavimo kalbų metamodelius, siekiant įvertinti jų panašumus bei skirtumus.
- Metrikų taikymas. Tyrimas atliekamas taikant įvairias metrikas (matavimo sistemas), siekiant įvertinti modeliavimo kalbų sudėtingumą bei tinkamumą.
- Paradigminė analizė. Tyrimas atliekamas formuojant, patvirtinant arba paneigiant modeliavimo kalbų prielaidas.
- Nenumatytų atvejų identifikavimas. Tyrimas atliekamas norint identifikuoti projekto nenumatytus atvejus.
- Ontologinis įvertinimas. Tyrimas atliekamas sulyginant modeliavimo kalbos elementus su ontologijos elementais, siekiant įvertinti modeliavimo kalbų galimybes pavaizduoti realaus pasaulio objektus.
- Kognityviają (pažinimo) psichologija pagrįsti metodai. Tyrimas atliekamas taikant kognityviosios psichologijos teorijas modeliavimo kalboms įvertinti.

Jei norima kuo tiksliau įvertinti veiklos modeliavimo kalbas, geriausia jas nagrinėti taikant kelis vertinimo metodus (Kamandi, Habibi, 2008). Svarbiausia įvertinti, kaip veiklos modeliavimo kalba gali perteikti realaus pasaulio objektus (White, 2004; Wahl, Sindre, 2005; zur Meuhlen et al., 2007; zur Meuhlen, Recker, 2008; zur Muehlen, Indulska, 2010; Geambasu, 2012). Kai kuriais atvejais yra kombinuojamos kelios modeliavimo kalbos, kurios viena kitą papildydamos (taikomos kartu) geriausiai perteikia realaus pasaulio objektus (zur Meuhlen et al., 2007).

## 2.4 Veiklos procesų modeliavimo gairės

Pradėjus modeliuoti veiklos procesus atsirado poreikis turėti veiklos procesų modeliavimo gaires – taisykles, kurių laikantis būtų galima kurti korektiškus bei lengvai suprantamus veiklos procesų modelius. Vienas iš žinomiausių karkasų veiklos modelių kokybės vertinimui yra SEQUAL karkasas (dar gerai žinomas kaip

semiotinis kokybės karkasas) (Krogstie et al., 2006), sukurtas remiantis semiotine teorija, apibrėžia keletą kokybės aspektų, pagrįstų ryšiais tarp modelio, modeliuotojo, dalykinės srities, modeliavimo kalbos ir veiklų, susijusių su modeliavimu, mokymusi. Veiklos modelių kokybės gerinimui buvo sukurtas ir eksperimentiškai įvertintas modeliavimo gairių karkasas (angl. *Guidelines of Modeling (GoM)*) (Becker et al., 2000), kuris apėmė šešis aspektus: korektiškumo, aiškumo, tinkamumo, palyginamumo, ekonominio efektyvumo ir sisteminio projektavimo. Šių karkasų teikiamos žinios yra abstrakčios ir sunkiai suprantamos bei pritaikomos praktikoje karjerą pradendanties specialistams ar savamoksliams modeliuotojams, todėl kitų mokslininkų atliktas tyrimas leido apibrėžti septynias veiklos procesų modeliavimo gaires (Mendling et al., 2010b). Šiame empiriniame tyrime buvo išnagrinėti ryšiai tarp veiklos procesų modelio struktūros charakteristikų ir skirtingų supratimo faktorių, tokių kaip veiklos procesų suprantamumas, klaidų tikimybės ir elementų vardų dviprasmybės. Gauti rezultatai leido suformuoti septynias veiklos procesų modeliavimo gaires (angl. *7 Process Modelling Guidelines (7PMG)*)(2.1 lentelė).

**2.1 lentelė.** Septynios veiklos procesų modeliavimo gairės, remiantis (Mendling et al., 2010b)

Gairė	Gairės aprašymas
G1	Naudoti kuo mažiau elementų modelyje, kiek tik įmanoma. Modelio dydis daro įtaką modelio supratimui bei klaidų atsiradimui. Dideli modeliai dažniausiai būna sunkiau skaitomi ir turi didesnę klaidų tikimybę lyginant su mažesniais modeliais.
G2	Sumažinti srautų skaičių vienam elementui. Kuo didesnis elementų skaičius modelyje, pvz., įeinančių ir išeinančių srautų į vieną elementą, tuo modelį skaityti darosi sunkiau. Nustatyta didelė koreliacija tarp modeliavimo klaidų ir elementų skaičiaus modelyje.
G3	Naudoti vieną pradžios ir vieną pabaigos įvykį. Modeliai, kurie tenkina šį punktą, yra lengviau skaitomi bei analizuojami.
G4	Modeliuoti kaip įmanoma struktūriškiau. Modelis yra struktūrizuotas tada, kai kiekvienas išskyrimo elementas turi ir sujungimo elementą. Struktūrizuoti modeliai gali būti įsivaizduojami kaip matematinės formulės, kur kiekvienas atitinkamo tipo atsidarantis skliaustelis turi savo tipo uždaramą skliaustelį. Modeliuose, kurie netenkina šio punkto, pasitaiko daugiau klaidų. Pritaikius šį punktą, modeliai yra lengviau suprantami.
G5	Vengti OR elementų srautų. Modeliuose, kuriuose naudojami AND arba XOR elementai, klaidų pasitaiko rečiau.
G6	Veiklų pavadinimus formuoti naudojant „veiksmažodis–daiktavardis“ struktūrą. Atlikus išsamią veiklos procesų modeliuose vartojamų veiklų vardų analizę buvo nuspręsta, kad struktūra „veiksmažodis–objektas“ („atlikti tyrimą“) yra mažiau dviprasmiška negu struktūra „veikla–daiktavardis“ („tyrimo vykdymas“).
G7	Išskaidyti modelį, jei jį sudaro daugiau negu 50 elementų. Šis punktas susijęs su G1 punktu, kadangi yra ryšys tarp modelio dydžio ir klaidų atsiradimo. Modeliuose, kuriuose yra daugiau negu 50 elementų, klaidų atsiradimo tikimybė yra didesnė negu 50 %.

Įvertinti VPM kokybę labai svarbu, kadangi nuo modelio kokybės priklausys pačios sistemos funkcionavimas – ar jis bus sėkmingas, ar ne. VPM suprantamumas yra svarbus faktorius (Gruhn, Laue, 2009) norint sukurti korektiškai funkcionuojančią

sistema, kuri yra didelės apimties, ir ypač tada, kai jos kūrimo procese dalyvauja nemažas skaičius specialistų. Todėl tyrimai, kuriais siekiama išnagrinėti veiklos procesų modelius ir sumažinti jų sudėtingumą, yra aktualūs (Gruhn, Laue, 2009; la Rosa et al., 2011a; la Rosa et al., 2011b). Ryšys tarp modelio sudėtingumo ir jo suprantamumo įrodytas (Mendling, et al., 2007; Abdul et al., 2008; Recker et al., 2009; Lassen, van der Aalst, 2009). Šie sprendimai nurodo įvairius sudėtingesnius šios problemos sprendimus (tokius kaip šablonų taikymai), tačiau šiame darbe buvo nagrinėjamos bendrosios korektiško veiklos procesų modeliavimo gairės, apimančios tas pačias problemas, atsižvelgiant į sukurtos transformacijos reikalavimus.

Egzistuoja nemažai BPMN2 VPM kūrimo taisyklių (IBM KC, 2014; BPMN MG, 2014; Tristotech, 2016), kurios skiriasi tarpusavyje dėl to, kaip kiekviena organizacija ar ekspertas jas supranta ir taiko savo veikloje. Atlikti tyrimai (Mendling, et al., 2007; Pinggera et al., 2012; Pinggera et al., 2015) parodė, kad VPM supratimą lemia asmeniniai faktoriai, modelio supratimą apsunkina modelio dydis bei didelis ryšių tankis, t. y. kai vienas procesas savyje turi daug sekos srautų.

Daugelis aprašytų sprendimų nenurodo tikslų veiklos procesų modelių kūrimo gairių, daugelis jų yra abstrakčios, lengvai interpretuojamos pagal turimas žinias. Iš 10 aprašytų (Silver, 2008) modeliavimo patarimų vienintelis bei lengvai suprantamas ir pritaikomas yra veiklos vardo struktūros apibrėžimas („veiksmažodis–daiktavardis“), kurio naudojimas pagal atliktus tyrimus pagerina modelio kokybę (Mendling et al., 2010a) (vertinant modelio suprantamumą, lyginant su kitais veiklos pavadinimo formatais). Atlikti tyrimai patvirtina, kad klaidos, atsiradusios modeliuojant veiklą, sudaro nuo 10 % iki 20 % visų projektų (Mendling et al., 2008), todėl nesudėtingų ir lengvai suprantamų veiklos modeliavimo gairių apibrėžimo poreikis išlieka svarbus.

## 2.5 OMG standartas BPMN2

BPMN (angl. *Business Process Modeling Notation*) – nuo platformos nepriklausoma standartizuota veiklos procesų modeliavimo notacija (OMG, 2011; OMG, 2013), kuri buvo sukurta išanalizavus tokias modeliavimo notacijas kaip UML, IDEF, ebXML, *RosettaNet*, LOVeM ir įvykiais pagrįstas veiklos grandines (angl. *Event-driven Process Chains*) (Recker et al., 2006). Kadangi BPMN metametamodelis yra MOF (angl. *Meta-Object Facility*) (OMG, 2006), taip užtikrinamas pilnas suderinamumas su kitais OMG grupės standartais (pvz., UML, SBVR, OSM, BMM). Pirmoji versija pasirodė 2004 metais. Pirminis tikslas – pateikti notaciją, kuri būtų bendrai suprantama verslo vartotojams nuo verslo analitiko, kuris pateikia proceso pirminius eskizus, techninio kūrėjo, atsakingo už realizaciją, iki verslo personalo, atsakingo už diegimą ir priežiūrą (Koehler, 2011). BPMN2 (BPMN 2.0) – naujusia BPMN versija, kuri pastebimai skiriasi nuo pirmosios: buvo papildyti elementai, įtrauktos naujos diagramos. BPMN2 specifikacija praplėsta naujomis charakteristikomis: formalizuoja visiems BPMN elementams vykdymo semantiką, apibrėžia išplėtimo mechanizmą proceso modelio ir grafiniams plėtiniams, patobulina įvykių sudėtį ir koreliaciją, praplečia žmogaus sąveikos apibrėžimą, apibrėžia choreografijos (angl. *choreography*) ir bendravimo (angl. *conversation*) modelius (geresnei modeliavimo sąveikai) (Geambasu, 2012), taip pat išsprendžia ankstesnės

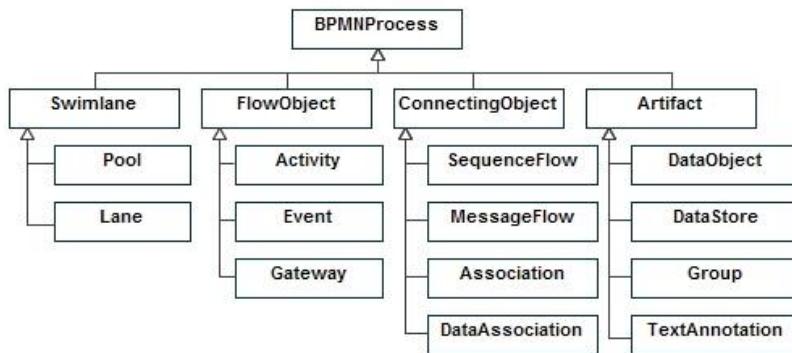
BPMN versijos nenuoseklumą ir dviprasmybes (Koehler, 2011). BPMN leidžia apibrėžti procesą, jo vykdymo aplinkybes, apribojimus, atsakingus asmenis, sekti procesų vykdymą ar atlikti jų simuliaciją.

Veiklos procesų modeliavimo standartas BPMN yra priskiriamas MDA (angl. *Model Driven Architecture*) architektūros CIM (angl. *Computation Independent Model*) lygmeniui (Asnina, 2009). Šiame lygmenyje yra apibrėžiamos organizacijos elementų sąvokos, jos tikslai, joje vykstantys procesai, nustatoma veiklos procesų vykdymo tvarka, t. y. sudaromas dalykinės srities veiklos modelis bei žodynas.

BPMN modeliavimo elementai yra skirstomi į penkias grupes pagal paskirtį:

- srauto objektai (angl. *flow objects*);
- jungiamieji objektai (angl. *connecting objects*);
- konteineriai (angl. *swimlanes*);
- duomenys (angl. *data*);
- artefaktai (angl. *artifacts*).

Pagrindiniai BPMN procesų diagramos elementai yra pateikti 2.1 paveiksle.



**2.1 pav.** Pagrindiniai BPMN procesų diagramos elementai, remiantis (OMG, 2013)

Pagrindiniai BPMN elementai gali būti išskirstomi smulkiau pagal elemento semantiką. Veiklos objektai išskirti į tris grupes: veiklos (angl. *activity*), subprocesas (angl. *subprocess*) ir išskleistas subprocesas (angl. *expanded subprocess*). Įvykiai suskirstyti į tris tipus: pradžios (angl. *start*), vidurio (angl. *intermediate*) ir pabaigos (angl. *end*) įvykius. Kiekvienas įvykio tipas turi savo potipius. Sprendimo priėmimo taškas turi 6 tipus: išskiriantis (angl. *exclusive*), išskiriantis su žymėjimu (angl. *exclusive with marker*), apimantis (angl. *inclusive*), kompleksinis (angl. *complex*), įvykiais pagrįstas (angl. *event based*), paralelinis (angl. *parallel*). Kiekvienas BPMN elementas turi savo grafinį atvaizdą bei semantinę reikšmę, kuri nusako funkcionalumą.

BPMN elementai ir diagramos. BPMN modeliavimo kalba turi grafinę notaciją, kuri susideda iš elementų, suskirstytų į dvi grupes: pagrindinius ir išplėstinius BPMN elementus. Vartojant BPMN notaciją galima sukurti struktūriškai skirtingus, bet savo elgsena ekvivalentus (Lam, 2009) procesų modelius, kurie apibrėžia tos pačios dalykinės srities procesą.

BPMN turi 4 tipų diagramas, kuriomis galima modeliuoti organizacijose vykstančius procesus; šių diagramų tipai ir aprašymai pateikti 2.2 lentelėje. Paskutinis lentelės stulpelis parodo, kaip BPMN diagramų naudojimas pasikeitė BPMN2



versijoje lyginant su ankstesne jos versija.

## 2.2 lentelė. BPMN diagramų tipai ir jų aprašymai

Diagramos tipas	Aprašymas	Nuo BPMN 2.0 versijos
Procesų (angl. <i>process</i> )	Diagrama skirta modeliuoti veiklos procesų sudėčiai bei logikai (orkestruoti).	ne (modeliuojama papildomai įtraukiant bendradarbiavimo diagramos logiką)
Choreografijos (angl. <i>choreography</i> )	Diagrama skirta parodyti sąveikoms tarp veiklos dalyvių kitokiu formatu, t. y. didžiausią dėmesį teikti pranešimų srautams, o ne atskiroms proceso veikloms. Pateikiama informacija kaip ir bendradarbiavimo diagramoje.	taip
Bendradarbiavimo (angl. <i>collaboration</i> )	Diagrama skirta parodyti sąveikoms tarp veiklos dalyvių, sekų srauto pasikeitimams tarp konteinerių (angl. <i>pools, lanes</i> ). Pateikia informaciją apie bendradarbiavimą tarp procesų diagramų. Anksčiau buvusi procesų diagramos dalis, tačiau atskirta perkeliant konteinerių ir pranešimo srautus į atskirą diagramą.	ne (modeliuojama vienoje diagramoje su procesų diagrama)
Bendravimo (angl. <i>conversation</i> )	Diagrama skirta modeliuoti bendravimui tarp atskirų veiklos dalyvių, neparodant atskirų pranešimo srautų. Šioje diagramoje pateikiami veiklos dalyviai susieti per veikloje naudojamus svarbiausius objektus. Pateikiama informacija kaip ir bendradarbiavimo diagramoje, tik čia nėra nurodyti atskiri konkretūs pranešimo srautai.	taip

BPMN procesų diagramomis yra pateikiama informacija apie proceso sudėtį bei logiką. Likusios BPMN diagramos naudojamos norint perteikti bendravimą tarp procesuose dalyvaujančių dalyvių arba jų veiklą. Dalis bendradarbiavimo logikos gali būti vaizduojama procesų diagramoje pagal tai, kas yra vaizduojama: privatus ar viešas procesas. Privatus procesas nuo viešo skiriasi tuo, kad viešame procese yra naudojami konteineriai bei pranešimo srautai norint parodyti sąveiką tarp privataus proceso ir kito proceso veiklos dalyvio, jo veiklą perteikiant pagal juodosios dėžės principą, dalyvio atstovaujantį konteinerį paliekant tuščią.

BPMN veiklos procesų diagrama palaiko daugiasluoksnius veiklos procesus. Tokiu atveju yra leidžiama veiklos procesą išskaidyti ir pavaizduoti skirtinguose hierarchijos lygiuose. BPMN modeliavimo notacija yra specialiai orientuota į veiklos procesų modeliavimą, todėl leidžia modeliuoti tiek procesų choreografiją, tiek orkestruotę (Radgui et al., 2012).

## 2.6 OMG standartas UML

UML (angl. *Unified Modeling Language*) – objektiškai orientuota modeliavimo kalba. Šios modeliavimo kalbos pagrindas yra trys modeliavimo technikos: *Booch*, objektinė modeliavimo technika (angl. *Object Modelling Technique*) ir objektiškai orientuotas programinės įrangos kūrimo procesas (angl. *Objectory (OOSE) Process*). UML modeliavimo kalba yra laikoma standartu; jos paskirtis – suteikti priemones programinės įrangos sistemų analizei, projektavimui bei įgyvendinimui, taip pat suteikti galimybes vaizduoti veiklos ar kitiems reikiamiems procesams (OMG, 2015). Šios galimybės skirtos sistemų architektams, programinės įrangos inžinieriams bei kūrėjams. UML yra priskiriamas MDA architektūros PIM (angl. *Platform Independent Model*) lygmeniui – tai dažniausiai reikalavimų programinei sistemai specifikacijos.

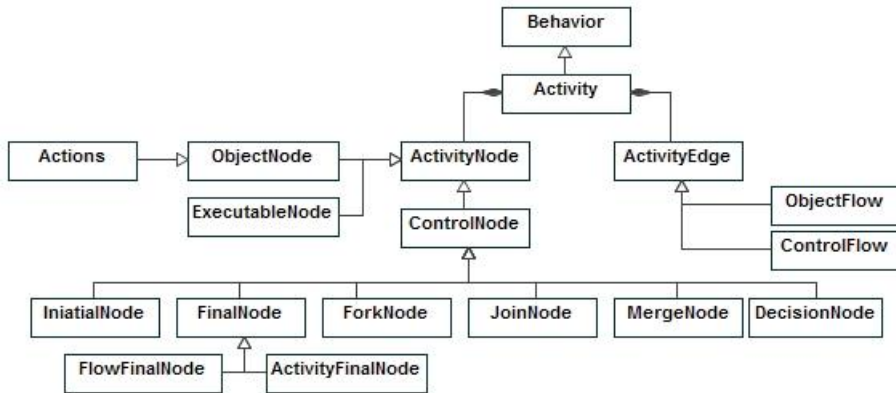
Naujausia UML 2.5 versija turi 14 skirtingų tipų modeliavimo diagramų, kurios yra suskirstytos į tokias grupes (OMG, 2015).

- Elgsenos diagramos (angl. *behaviour diagrams*). Šio tipo diagramos skirtos dinaminei sistemos objektų elgsenai vaizduoti. Nusako kuriamos sistemos funkcionalumą.
- Sąveikos diagramos (angl. *interaction diagrams*). Laikomos elgsenos diagramos dalimi, jų tikslas – pateikti išsamesnį veiklos vaizdą per objektų sąveikas.
- Struktūrinės diagramos (angl. *structure diagrams*). Skirtos sistemos struktūrai modeliuoti įvairiuose detalumo lygiuose, nuo žemiausių iki abstrakčiausių.

Veiklos procesų modeliavimui gali būti taikoma UML AD (angl. *activity diagram*) veiklos diagrama (toliau UML VD), kuri priklauso UML elgsenos (angl. *behavioral*) modeliams. Papildomos informacijos gavimui nagrinėjant veiklą bei jos objektus, be UML veiklos diagramų, gali būti papildomai įtraukiamos kitos elgsenos grupei priklausančios diagramos, tokios kaip panaudojimo atvejų, sąveikos, būsenos mašinių diagramos. UML VD diagramoms naudojami elementai yra suskirstyti į tokias kategorijas:

- veiksmas (angl. *actions*);
- kontrolės mazgai (angl. *control nodes*);
- objektai ir objektų srautai (angl. *objects and object flow*)

Pagrindinė UML veiklos diagramos struktūros ir srauto semantika pateikta 2.2 paveiksle.



2.2 pav. Pagrindiniai UML veiklos diagramos elementai, remiantis (OMG, 2015)

UML veiklos diagrama nepalaiko procesų hierarchiškumo, t. y. nesuteikia galimybių vaizduoti daugiasluoksniams veiklos procesams (Ko et al., 2009).

## 2.7 Veiklos modeliavimo kalbų vertinimas: BPMN ir UML

Srities specialistai gali pasirinkti iš daugelio skirtingų modeliavimo metodų, kiekvienas jų skiriasi tuo, ką norima akcentuoti (orientuoti): į duomenis orientuoti (angl. *data-oriented*), į objektus orientuoti (angl. *object-oriented*) ar į procesus orientuoti (angl. *process-oriented*) modeliavimo metodai. Dabartiniai tyrimai parodo, kad du modeliavimo požiūriai yra labiausiai paplitę; tai į objektus orientuotas modeliavimas programinei įrangai modeliuoti ir į procesus orientuotas modeliavimas procesų dokumentacijai tvarkyti bei tobulinti (Recker et al., 2009).

Tarp paplitusių ir naudojamų standartų galima išskirti UML ir darbų sekų modelį (angl. *workflow model*). Veiklos procesams modeliuoti plačiausiai naudojamos yra BPMN ir UML VD procesų modeliavimo notacijos (Geambasu, 2012). Modeliavimo kalbų vertinimams taikomi įvairūs metodai: empiriniai eksperimentai, literatūros analizė, įvairūs specializuoti karkasai.

Atliktame tyrime (van der Aalst et al., 2003) buvo išskirtas 21 šablonas, apibūdinantis veiklos procesų elgseną. Pagal šiuos šablonus buvo ištirtos dvi modeliavimo notacijos (White, 2004): BPMN procesų diagramos ir UML veiklos diagramos, kaip kiekviena iš jų gali pavaizduoti išskirtus šablonus. Buvo atliktas šių modeliavimo kalbų palyginimas šablonais, kurie yra suskirstyti į kelias grupes.

- Pagrindiniai valdymo šablonai. Pirmieji penki šablonai nusako paprastą proceso elgseną. Jie apibūdina pagrindinius veiklos procesų modeliavimo šablonus.
- Išplėsti šakojimosi ir sinchronizacijos šablonai. Kiti penki šablonai apibūdina sudėtingesnius veiklos proceso srauto išsišakojimo ir suliejimo būdus.
- Struktūriniai šablonai. Nusako tokį elgesį kaip veiklų cikliškas pakartojimas ir atskirų procesų kelių nepriklausomybė.
- Šablonai, apimantys kelis atvejus. Nusako, kaip atskiri atvejai ar veiklų pasikartojimai yra sukuriami.

- Būsenų grįžti šablonai. Nusako, kaip veiklos proceso elgsena kartais gali būti paveikta faktorių.
- Atšaukimo šablonai. Nusako veiklos ar veiklų atšaukimą.

Dviejų kalbų BPMN ir UML veiklos procesų modeliavimo palyginimas rodo, kad BPMN yra išraiškingesnis negu UML dėl to, kad UML veiklos diagramos ne visada atitikdavo išskirtų šablonų reikalavimus, o taikoma modeliavimo technika sudėtingesnė negu BPMN veiklos procesų diagramoje. Tyrime (zur Meuhlen, Recker, 2008) teigiama, kad UML (kalbant tik apie veiklos diagramas) sudėtingumas yra panašus į BPMN, tad BPMN sulaukia vis daugiau dėmesio (Chinosi, Trombetta, 2012) ne tik pramonėje kaip grafinis veiklos procesų vaizdavimo modeliavimo būdas, bet ir mokslinėje aplinkoje. Kitas empirinis tyrimas su naudotojais parodė, kad UML veiklos diagrama yra naudojama kaip ir BPMN: nebuvo didelio skirtumo tarp naudotojų išreikšto efektyvumo, pakankamumo ar pasitenkinimo naudojant vieną iš pasirinktų modeliavimo priemonių (Birkmeier et al, 2010).

BPMN veiklos procesų diagrama palaiko procesų hierarchiškumą (Recker, 2008), o UML VD procesų hierarchija yra vieno lygio, tik suteikianti galimybę atskiras veiklas detalizuoti kuriant atskiras veiklos diagramas (Ko et al., 2009).

Pirminis BPMN kūrimo tikslas buvo sukurti tokią modeliavimo kalbą, kurią suprastų visi veiklos dalyviai, todėl ji yra skirta visiems: nuo veiklos analitikų, programinės įrangos kūrėjų iki veiklos dalyvių (Geambasu, 2012). UML vartotojai yra programinės įrangos kūrėjai (Ko et al., 2009).

UML veiklos diagramos paskirtis labai skiriasi nuo BPMN veiklos diagramos paskirties (Flowers, Edeki, 2013). UML veiklos diagrama yra skirta detalizuoti aukštesnio lygio reikalavimams, pateiktiems panaudojimo atvejų diagrama, ir papildyti žemesnio lygio reikalavimams, pateiktiems sekų diagrama. Tuo tarpu pats veiklos procesų kūrimas nėra atliekamas vien tik informacinių sistemų kūrimui, bet ir optimizavimui, reinžinerijai, susijungimui/atskyrimui, naujo produkto įvedimui. Yra įprasta, kad UML skirta modeliuoti programinei įrangai, o BPMN – veiklos procesams (da Silva, 2015). Taip pat pastebėta, kad UML VD negali pilnai perteikti veiklos procesų, kurie susiję su resursų ar organizacinių aspektų vaizdavimu (Recker, 2010). Todėl galima daryti išvadą, kad jei pirminė taikymo sritis yra sistemų inžinerija, tada ji neturės pakankamo išraiškingumo lyginant su ta, kurios pirminė taikymo sritis yra procesų inžinerija. Veiklos procesų modeliavimo standartų (BPMN ir UML VD) lyginamoji analizė pagal pagrindinius vertinimo kriterijus yra pateikta 2.3 lentelėje.

**2.3 lentelė.** Veiklos procesų standartų analizės lentelė pagal pagrindinius vertinimo kriterijus

Kriterijus	BPMN	UML VD
ADM lygmuo	CIM	PIM
Metamodelis	MOF	MOF
Notacija	Yra	Yra
Elementų kiekis (įvertinant elementų tipus)	72 (veiklos diagramoje)	31
Taikymo sritis	Procesų inžinerija	Sistemų inžinerija
Modelio kūrimo tikslas	Analizė, aprašas,	Aprašas, vykdymas

Kriterijus	BPMN	UML VD
	vykdymas	
Vartotojų grupė	Analitikai, dalykinės srities atstovai	Programinės įrangos kūrėjai
Procesų hierarchijos palaikymas	Daugiasluoksnis	Vieno lygio
Vaidmenų hierarchijos palaikymas	Taip	Taip
Orkestruotės perspektyva	Taip	Taip
Choreografijos perspektyva	Taip	Ne
Bendradarbiavimo perspektyva	Taip	Ne
Bendravimo perspektyva	Taip	Ne
Panaudojimas veiklos procesų valdymo gyvavimo cikle	Modeliavimas, keitimas, patikrinimas, simuliacija, tobulinimas	Modeliavimas, keitimas

Pagal atliktą literatūros analizę (Geambasu, 2012) lyginant BPMN ir UML VD buvo išskirti tokie kriterijai: modeliavimo kalbos suprantamumas (t. y. kaip greitai notacija yra išmokstama ir suprantama); realaus pasaulio procesų pavaizdavimas (t. y. kaip modeliavimo kalbos sugeba perteikti realaus pasaulio procesus bei objektus); transformacija į BPEL (t. y. kaip modeliavimo kalbos gali būti transformuojamos į BPEL). Veiklos procesų modeliavimo standartų (BPMN ir UML VD) lyginamoji analizė pagal išvardytus vertinimo kriterijus yra pateikta 2.4 lentelėje.

**2.4 lentelė.** Veiklos procesų standartų analizės lentelė pagal išskirtus kriterijus

Kriterijus	BPMN	UML VD
Lengvas suprantamumas (analitikai, programinės įrangos kūrėjai, kiti dalykinės srities dalyviai)	Suprantamas panašiai kaip UML AD. Dalyviai, kurie neturėjo žinių apie notacijas, taip pat parodė, kad standartai suprantami panašiai (Peixoto et al., 2008).	Suprantamas panašiai kaip BPMN.
Galimybė pavaizduoti realaus pasaulio procesus bei objektus	Galimybė pavaizduoti tam tikras situacijas panaudojant mažiau elementų. Kiekvienas BPMN elementas turi tam tikrą specifinę paskirtį (Ko et al., 2009).	Kai kuriose situacijose būtina naudoti keletą elementų, o BPMN atveju jų užtektų mažiau.
Transformacija į BPEL	OMG specifikacijoje pateikta informacija apie elementų susiejimą norint atlikti transformaciją į BPEL4WS (angl. <i>Business Process Execution Language for Web Services</i> ).	OMG specifikacijoje informacija nepateikta. Transformacija į BPEL4WS metamodelio pagrindu (Korherr, List, 2006; Hlaoui, Benayed, 2011) yra ribota, elementų susiejimo rinkinys tarp UML VD ir BPEL nėra pilnas.

Veiklos procesų modeliavimui yra specializuota kalba – BPMN. Šios modeliavimo kalbos suprantamumas yra panašus lyginant su UML VD, tačiau UML pirminė taikymo sritis yra sistemų inžinerija. Be to, BPMN elementai turi siauresnę

semantinę prasmę; tas leidžia daryti išvadą, jog modelių interpretacijoms lieka mažiau vietos, ir tai pagerina jų suprantamumą. Galimybė procesus perteikti BPEL taip pat svarbi, kaip ir panaudojimo veiklos procesų valdymo gyvavimo cikle, šiuo atveju BPMN pasižymi platesnėmis galimybėmis.

## 2.8 Kompleksinis modeliavimo standartų vertinimas

Ontologija, arba vaizdavimo teorija, gali būti naudojama kaip etalonas darant spėjimus apie gramatikos galimybes pateikti realaus pasaulio srities išsamų ir aiškų vaizdavimą. Ontologijos taikymas tokiam tikslui yra vadinamas vaizdavimo analize (angl. *representational analysis*). BWW (Bunge-Wand-Weber) modelio pagrindas yra ontologija, kurios pagrindinis tikslas – nustatyti, ar modeliavimo kalba gali pilnai perteikti realaus pasaulio objektus. BWW modelis susideda iš 40 aukštesnio lygio abstrakčių konstrukčių, kurie gali būti sugrupuoti į 4 kategorijas (zur Meuhlen et al., 2007):

- daiktai ir jų savybės;
- daiktų būsenos;
- įvykiai ir transformacijos;
- sistemos ir jų sudėtis.

Panaudojant BWW modelį buvo atliktas tyrimas, kurio tikslas – ištirti dviejų taisyklių modeliavimo kalbų vaizdavimo galimybes – SRML (angl. *Simple Rule Markup Language*) ir SBVR lyginant su konceptualių modeliavimo kalbų vaizdavimo galimybių įvertinimais, šiuo atveju su BPMN. Gauti rezultatai pateikti 2.5 lentelėje. Žvaigždutės simboliu (\*) pažymėtas laukas nebuvo pažymėtas atliktame tyrime, tačiau būsenos gali būti gaunamos iš BPMN įvykių (jeigu jų vardai formuojami taikant gerąją veiklos procesų modeliavimo praktiką) bei iš BPMN duomenų objektų su būsenomis (*DataObject(withState)*).

**2.5 lentelė.** Modeliavimo kalbų vaizdavimo galimybių tyrimo rezultatai (zur Meuhlen et al., 2007)

	Kalba	SRML	SBVR	BPMN
	<b>BWW konstruktai</b>			
Daiktai (angl. <i>Things</i> )	Daiktas (angl. <i>Thing</i> )	✓	✓	✓
	Savybė (angl. <i>Property</i> )	✓	✓	✓
	Klasė (angl. <i>Class</i> )		✓	✓
	Tipas (angl. <i>Kind</i> )			✓
Būsenos (angl. <i>States</i> )	Būsena (angl. <i>State</i> )	✓		*
	Galima būsenos vieta (angl. <i>Conceivable State Space</i> )	✓		
	Būsenos taisyklė (angl. <i>State Law</i> )	✓	✓	
	Teisėta būsenos vieta (angl. <i>Lawful State Space</i> )	✓		
	Stabili būsena (angl. <i>Stable State</i> )			
	Nestabili būsena (angl. <i>Unstable State</i> )			
	Istorija (angl. <i>History</i> )			

	Kalba	SRML	SBVR	BPMN
	<b>BWW konstruktai</b>			
Įvykiai (angl. <i>Events</i> )	Įvykis (angl. <i>Event</i> )	√		√
	Galima įvykio vieta (angl. <i>Conceivable Event Space</i> )			
	Teisėta įvykio vieta (angl. <i>Lawful Event Space</i> )			
	Išorinis įvykis (angl. <i>External Event</i> )			√
	Vidinis įvykis (angl. <i>Internal Event</i> )			√
	Gerai apibrėžtas įvykis (angl. <i>Well-defined Event</i> )			√
	Silpnai apibrėžtas įvykis (angl. <i>Poorly-defined Event</i> )			√
	Transformacija (angl. <i>Transformation</i> )			√
	Poravimas (angl. <i>Coupling</i> )			√
	Veikimas (angl. <i>Acts on</i> )			√
Sistemos (angl. <i>Systems</i> )	Sistema (angl. <i>System</i> )		√	√
	Sistemos sudėtis (angl. <i>System Composition</i> )		√	√
	Sistemos aplinka (angl. <i>System Environment</i> )			√
	Sistemos struktūra (angl. <i>System Structure</i> )			
	Sistemos skaidymas (angl. <i>System Decomposition</i> )		√	√
	Lygių struktūra (angl. <i>Level Structure</i> )			√
	Posistemė (angl. <i>Sub-System</i> )			√
BWW rezultatai		10	7	19

Standartų porų analizė pagal Bunge-Wand-Weber ontologiją padėjo nustatyti, kad SRML ir SBVR standartų integracija su BPMN standartu turi geriausias išraiškingumo galimybes pavaizduojant realaus pasaulio procesus, įvertinant naudojamų elementų persidengimą bei reikiamos elementų aibės pilnumą.

## 2.9 OMG standartas SBVR

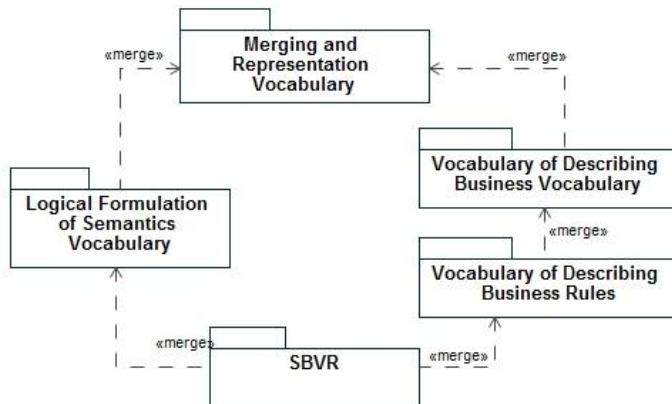
Veiklos žodyno ir veiklos taisyklių semantikos standartas SBVR (angl. *Semantics for Business Vocabularies and Rules*) yra OMG specifikacija (OMG, 2013b), kuria išreiškiamos veiklos žinios veiklos dalyviams suprantama kalba ir leidžiama užrašyti jas notacija, kuriai nereikia specialių IT žinių. SBVR veiklos konceptai ir veiklos taisyklės vaizduojamos terminais, vardais, veiksmažodžiais, apibrėžimais ir teiginiais, o jų semantika išreiškiama daiktavardiniais konceptais, veiksmažodiniais konceptais ir loginėmis formuluotėmis, pagrįstomis pirmos eilės logika ir kai kuriais modalinės logikos elementais. SBVR yra suderinamas su OMG modeliais grindžiama architektūra MDA (Hendryx, 2005; Ceravolo et al., 2007).

Veiklos žodyno ir veiklos taisyklių apskaitimui tarp skirtingų organizacijų ar sistemų yra naudojama XMI schema (OMG, 2008). MDA architektūroje SBVR yra priskiriamas CIM lygmeniui (Linehan, 2008; Asnina, 2009), šiame etape yra suteikiama galimybė formuoti veiklos žodyną, identifikuoti veiklos apribojimus ir juos specifiuoti ribota natūralia kalba, kuri yra suprantama tiek veiklos analitikams, tiek organizacijos atstovams (Miers, 2007; Nemuraite et al., 2010).

SBVR semantiniu lygiu realizuoja esminį veiklos taisyklių požiūrio principą, vadinamąją veiklos taisyklių mantrą, kurią apibrėžė veiklos taisyklių grupė (angl. *Business Rules Group*) 1995 m. (BRG, 2002). Šį principą galima išreikšti taip: „Veiklos taisyklės paremtos faktais, o faktai paremti terminais“. Kitaip tariant jei norima apibrėžti veiklos taisykles, reikia apibrėžti veiksmažodinius konceptus (faktus), o jei norima apibrėžti veiksmažodinius konceptus (faktus), reikia apibrėžti daiktavardinius konceptus (terminus).

SBVR yra sudarytas iš šių keturių dalių (2.3 pav.).

- *Meaning and Representation Vocabulary* (MRV) skirtas aprašyti pagrindinių veiklos konceptų prasmei, vaizdavimui, ryšiams.
- *Logical Formulation of Semantics Vocabulary* (LFSV) skirtas aprašyti veiklos taisyklių, išreikštų natūralia kalba, struktūrai semantinėmis formuluotėmis.
- *Vocabulary for Describing Business Vocabularies* (VDBV) – pagrindinis žodyno aprašymo metamodelis, skirtas aprašyti veiklos konceptams ir veiksmažodiniams konceptams, papildomai informacijai apie žodyną.
- *Vocabulary for Describing Business Rules* (VDBR) – veiklos taisyklių aprašymo metamodelis.



2.3 pav. SBVR dalių tarpusavio ryšiai, remiantis (OMG, 2008; OMG, 2013b, OMG, 2013c)

SBVR konceptai skirstomi taip.

- Veiklos žodyno konceptai:
  - Bendriniai konceptai (angl. *General concept*), išreiškiami bendriniais daiktavardžiais (pvz.: [car](#), [driver](#)).
  - Individai (angl. *Individual concept*), išreiškiami tikriniais daiktavardžiais (pvz.: [Lithuania](#), [100](#)).
  - Veiksmažodiniai konceptai (angl. *Verb concept*), išreiškiami veiksmažodžiais (pvz.: [driver drives car](#)).
  - Vaidmenys (angl. *Role*), išreiškiami daiktavardžiais. Tai daiktavardinis konceptas, kuris atitinka daiktus, priklausomus nuo jų atliekamos funkcijos, dalyvavimo ar panaudojimo tam tikroje situacijoje.



Veiksmožodiniai konceptai yra skirstomi į tokius tipus.

- Binarinis veiksmožodinis konceptas – tai bendriausias veiksmožodinis konceptas, kuris turi du ar daugiau vaidmenų (pvz.: [bank gives loan](#)).
- Unarinis veiksmožodinis konceptas – tai veiksmožodinis konceptas, kuris turi tik vieną vaidmenį (pvz.: [bank is reliable](#)).
- Dalies–visumos veiksmožodinis konceptas – tai binarinis veiksmožodinis konceptas, kuris nusako, kad vienas objekto tipas (visi jo egzemplioriai) yra kito objekto dalis (pvz.: [account is included in bank](#)).
- Savybės veiksmožodinis konceptas, kuris nusako svarbias objektų tipų savybes (pvz.: [loan has request date](#)).

SBVR apibrėžia dviejų tipų taisykles: struktūrinės ir vykdomąsias (operacines). Struktūrinės taisyklės yra laikomos nepažeidžiamomis, o operacinės priklauso nuo žmonių, todėl gali būti pažeistos. Veiklos taisyklių tipai ir jų pavyzdžiai (išreikšti taikant SBVR *Structured English*) pateikiami 2.6 lentelėje.

2. 6 lentelė. SBVR taisyklių tipai

Taisyklės tipas	Modalumas	SBVR taisyklių tipų pavyzdžiai taikant struktūrizuotos anglų kalbos stilių
Struktūrinė	būtinumas (angl. <i>necessity</i> )	It is necessary that <a href="#">order</a> always <i>has</i> exactly one <a href="#">customer</a> .
	galimybė (angl. <i>possibility</i> )	It is possible that <a href="#">order</a> <i>has</i> more than one <a href="#">line item</a> .
	negalimumas (angl. <i>impossibility</i> )	It is impossible that <a href="#">order</a> <i>never has</i> more than one <a href="#">customer</a> .
Vykdomoji	įsipareigojimas (angl. <i>obligation</i> )	It is obligatory that <a href="#">order</a> <i>is processed within</i> one <a href="#">business day</a> .
	leidimas (angl. <i>permission</i> )	It is permitted that <a href="#">customer representative</a> <i>approves</i> <a href="#">credit</a> .
	draudimas (angl. <i>prohibition</i> )	It is prohibited that <a href="#">clerk</a> <i>changes</i> <a href="#">term</a> and <a href="#">conditions</a> .

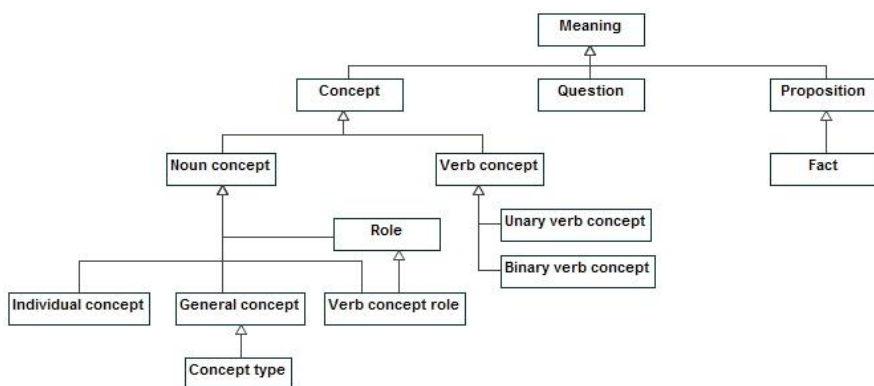
Veiklos taisyklės galima užrašyti dviem būdais.

- Naudojant specifinius raktinius žodžius (angl. *keywords*). Pagal veiklos taisyklės tipą yra naudojami raktiniai žodžiai, kaip „it is obligatory“, „it is possible“ ir kt., kurie yra veiklos taisyklės sakinio pradžioje (pvz.: [it is obligatory that each person on a bus has a ticket](#)).
- Taikant natūralios kalbos stilių. Šiuo atveju yra naudojami modaliniai veiksmožodžiai (pvz.: [each person must have a ticket on a bus](#)).

Veiklos taisyklės galima aprašyti:

- aktyvia sakinio forma (pvz.: [bank gives loan](#));
- pasyvia sakinio forma (pvz.: [loan is given by bank](#));
- daiktavardine forma (pvz.: [loan of the bank](#)).

SBVR prasmės metamodelio elementai. SBVR prasmės metamodelis pateiktas 2.4 paveiksle.



2.4 pav. SBVR prasmės metamodelis, remiantis (OMG, 2013c)

## 2.10 Lietuvos autorių susijusių darbų apžvalga

Lietuvoje yra atlikta nemažai įvairių tyrimų, susijusių su žinių inžinerija, modeliais pagrįsta inžinerija, informacinių sistemų kūrimu. Šie tyrimai yra atliekami ir iki šiol, kartais dalijantis patirtimi bei resursais su kitais universitetais. Vykdytas jungtinis projektas, kuriame dalyvavo Informacijos sistemų katedra iš Kauno technologijos universiteto ir Informacijos sistemų katedra iš Vilniaus Gedimino technikos universiteto (VETIS, 2009); projekto metu buvo sukurtas įrankis – SBVR tekstų redaktorių veiklos žodynams ir veiklos taisyklėms kurti bei validuoti.

Veiklos procesus bei jų metodus nagrinėja VGTU (ISD) (Vasilecas, Normantas, 2011; Normantas, Vasilecas, 2012; Normantas, Vasilecas, 2013; Vasilecas et al., 2014; Vasilecas et al., 2015), kurių darbai yra susiję su modeliais pagrįsta inžinerija.

Noras įtraukti visus dalykinės srities atstovus į kūrimo procesą buvo nagrinėjamas J. Karpovic (Karpovic et al., 2012; Karpovic et al., 2014; Karpovic et al., 2016) iš KTU (ISD). Darbe pasirinktas SBVR kaip priemonė neturintiems specifinių IT žinių dalykinės srities atstovams dalyvauti kūrimo bei validavimo procese kuriant ontologijas.

Artimiausių ryši su šiuo tyrimu turi L. Tutkutės tyrimas (Skersys et al., 2012a; Skersys et al., 2012b), atliktas KTU (ISD), kuriame yra nagrinėjamos integracijos problemos tarp veiklos procesų bei veiklos taisyklių (nagrinėjant BPMN ir SBVR). Pateiktas sprendimas apima veiklos žodyną išgavimą iš veiklos procesų ir pateiktą tekstų redaktoriuje.

Atliktas darbas yra susijęs su Informacijos sistemų katedros tyrimų veiklos kryptimis, tokiomis kaip conceptualusis modeliavimas, veiklos procesų bei veiklos taisyklių modeliavimas, modeliais pagrįsta inžinerija.

## 2.11 Veiklos procesų ir veiklos taisyklių modeliavimo analizės apibendrinimas ir išvados

Veiklos procesų modeliavimo būdų analizė parodė, kad veiklos procesus bei veiklos taisykles geriausia modeliuoti kartu, kadangi šie modeliavimo metodai yra laikomi vienas kitą papildančiais. Apžvelgus modeliavimo kalbų vertinimo metodikas išsiaiškinta, kad norint kuo tiksliau įvertinti veiklos modeliavimo kalbas svarbiausia

įvertinti, kaip veiklos modeliavimo kalba perteikia realaus pasaulio objektus. Kompleksinis modeliavimo standartų vertinimas (BWW ontologija) parodė, kad tinkamiausia sąsaja tarp veiklos taisyklių ir veiklos procesų modelių yra tarp BPMN ir SBVR arba BPMN ir SRML. Pasirinkta BPMN ir SBVR pora dėl šių priežasčių: OMG grupė plėtoja abi šias kalbas, ir susidomėjimas SBVR auga tiek mokslinių tyrimų, tiek praktinių taikymų bendruomenėse, o SRML praktiškai nebepalaikoma; SBVR išreiškiama natūraliai kalbai artima kalba ir suprantama veiklos dalyviams; BPMN yra veiklos procesų modeliavimui specializuota kalba (lyginant su UML, kuri skirta programinei įrangai modeliuoti).

Literatūros šaltinių apie veiklos modeliavimo būdus, jų metodikas bei modeliavimo gairių gausa rodo, kad veiklos procesų bei veiklos taisyklių modeliavimas yra plačiai nagrinėjama tema. Todėl tikslinga tobulinti veiklos procesų ir veiklos taisyklių modeliavimo kartu kūrimo metodus bei priemones.

### 3 VEIKLOS PROCESŲ IR VEIKLOS TAISYKLIŲ INTEGRACIJOS IR TRANSFORMACIJOS ANALIZĖ

Sudėtingų veiklos procesų modeliavimas susiduria su dideliais iššūkiais, kurie susiję su modelių išraiškingumu bei vientisumu. Dažnai neužtenka taikyti tik vieno tipo modelių norint gauti išsamią dalykinės srities specifikaciją, kuri būtų suprantama visiems veiklos dalyviams. Kelių skirtingų modelių kūrimas apimant vienos dalykinės srities procesus, naudojant skirtingas modeliavimo priemones, apsunkina patį kūrimo procesą bei palieka didelę tikimybę atsirasti klaidoms, kurios susijusios su skirtingų modelių suderinamumu: terminų neatitikimas, prieštaravimai (kai vieno modelio logika prieštarauja kito modelio logikai), kitos klaidos, pertekliškumas ir kt.

Modeliuojant veiklos procesų ir veiklos taisyklių modelius atskirai yra rizikuojama, kad abu modeliai neatitiks vienas kito, taip pat sumažės nauda norint veiklos procesą keisti ar tobulinti (Cheng et al., 2011). Veiklos procesų ir veiklos taisyklių modeliavimo būdai, nors ir išreiškiami skirtingai, laikomi vienas kitą papildančiais (Sinur, 2009; Koehler, 2011), tačiau kol kas nėra galimybės juos naudoti tame pačiame CIM lygmenyje bei aplinkoje. Tai, kad šie modeliavimo būdai būtų taikomi kartu, buvo diskutuojama jau seniai (zur Muehlen et al., 2008). Siekiant tinkamai sumodeliuoti procesą būtina įvertinti visas aplinkybes, sąlygojančias tinkamą proceso veikimą, tokias kaip veiklos apribojimai, t. y. veiklos taisyklės (Aghdasi, Malihi, 2010).

Veiklos procesų bei veiklos taisyklių modeliavimo kalbos yra skirtos tam pačiam tikslui – dokumentuoti organizacijos elgsenai ir procedūroms (Gasevic et al., 2010). Veiklos taisyklės egzistuoja kiekvienoje veikla pagrįstoje informacinėje sistemoje, dažnai jos būna „įsiūtos“ į kodą, jų negalima atskirti, be to – jos nuolat kinta (Vasilecas, Normantas, 2011) taikantis prie esamos situacijos. Dėl šių priežasčių atsiranda poreikis taisykles turėti greta veiklos procesų modelio.

Veiklos procesų bei veiklos taisyklių modeliavimo poreikis ir esami sprendimai buvo nagrinėjami (Mickeviciute, Butleris, 2013) straipsnyje. Literatūros analizė parodė, kad egzistuoja nemažai idėjų, kaip susieti BPMN2 VPM ir SBVR VŽ&VT, pvz.: (Agrawal, 2011; Cheng et al., 2011; Goedertier, Vanthienen, 2007; Graml et al., 2007; Habish et al., 2010; Hohwiler et al., 2011; Milanovic et al., 2011; Wu et al., 2011; Cheng et al., 2011). Tačiau pilno bei išbaigto metodo, kuris patenkintų visus integruoto modelio poreikius, nebuvo rasta. Kartu analizė parodė, kad buvo sukurti keletas SBVR redaktorių (Tommasi, Corallo, 2006; Musham et al., 2008; Nemuraite et al., 2010), tačiau nė vienas iš jų neturi galimybės palaikyti ryšius tarp dviejų modelių. Analizuojant veiklos procesų modeliavimo standartus pastebėta, kad vienas iš pagrindinių procesų modeliavimo kalbų trūkumų – esami standartai nesuteikia galimybės integruoti veiklos apribojimus (Recker et al., 2005; Green et al., 2005; zur Muehlen et al., 2007).

Atsirandantis pertekliškumas bei nesuderinamumas tarp veiklos procesų modelio ir veiklos žodyno bei taisyklių skatina tyrimus, kuriuose siekiama išsiaiškinti, kada veiklos taisyklė turėtų būti integruojama, kada ne, ir kokie faktoriai daro įtaką integruotiems ir neintegruotiems modeliams (Wang et al., 2016; Wang, 2016).

Šiame skyriuje aprašomi modelių integravimo metodai, pateikiama

egzistuojančių veiklos procesų ir veiklos taisyklių integracijos sprendimų apžvalga. Atsižvelgiant į esamų sprendimų trūkumus yra siūlomas naujas metodas veiklos procesų ir veiklos taisyklių integravimui dviem veiklos modeliavimo kalboms – BPMN2 ir SBVR.

### 3.1 Modelių integravimo metodai

Egzistuojant poreikiui sudėtingus veiklos procesus modeliuoti apimant skirtingus veikos aspektus ir įtraukiant visus su veikla susijusius dalyvius atsiranda poreikis taikyti kelis modeliavimo standartus kartu. Šią problemą galima išspręsti sukuriant į konkrečią dalykinę sritį orientuotą modeliavimo kalbą (angl. *Domain Specific Modeling Language (DSML)*): kuriamas naujas metamodelis, jis pasižymi visomis reikalingomis savybėmis, kurias turi keli metamodeliai. Tokio sprendimo trūkumai: kuriant DSML iš pagrindų, prarandama galimybė iš naujo pritaikyti standartizuotų modeliavimo kalbų metamodelius (UML, BPMN, SBVR ir kt.), kurie yra taikomi ir susilaukia vis daugiau dėmesio. Kitas problemos sprendimo būdas – integruoti skirtingų modeliavimo kalbų metamodelius tarpusavyje; tai leidžia kartu integruoti tais metamodeliais pagrįstus modelius. Yra išskiriami keli pagrindiniai metamodelių integravimo metodai (Emerson, Sztipanovits, 2006).

- Metamodelių suliejimas (angl. *Metamodel Merge*). Šis metodas leidžia sujungti kelis skirtingus metamodelius į vieną bendrą metamodelį. Dažniausiai šis metodas taikomas, kai metamodeliai turi vienodus elementų vardus, taip pat ir semantiką. Jeigu skirtingi metamodeliai turi bendrų elementų, t. y. tokių elementų, kurių semantinė reikšmė turi viena ir ta pati, tačiau jų vardai nėra vienodi, tada tokie elementai gali būti naudojami kaip jungimo taškai suliejant kelis metamodelius į vieną.
- Metamodelių sujungimas sukuriant sąsajas (angl. *Metamodel Interfacing*). Šis metodas pasižymi tarpinės sąsajos (struktūros) sukūrimu tarp kelių skirtingų metamodelių, nekeičiant nė vieno metamodelio struktūros. Tarpinėje sąsajoje talpinama informacija apie skirtingų metamodelių elementų tarpusavio ryšius. Toks metodas yra lengviau pritaikomas įvairiuose CASE įrankiuose, kur yra naudojami standartizuoti modeliavimo kalbų metamodeliai.
- Metamodelių klasių apibrėžimas (angl. *Class Refinement*). Šis metodas leidžia panaudojant vieną metamodelį praplėsti kito metamodelio elementus, kurie yra pateikti juodosios dėžės principu. Metodas taikomas panašiai kaip ir naudojant sąsajas, tačiau čia metamodelių ryšiai nusakomi detalizuojant vieno metamodelio konkretų konceptą kito metamodelio konceptais, tam panaudojant kompozicijos ryšius.
- Metamodelių susiejimas naudojant UML profilius (angl. *UML Profiles*). Šis metodas grindžiamas UML metamodelio naudojimu: UML yra plačiai paplitęs standartas, skirtas programinės įrangos modeliavimui. Nors UML laikoma bendrosios paskirties modeliavimo kalba, tačiau pasitaiko atvejų, kai neužtenka jos išraiškimumo perteikti specifinės tam tikros dalykinės srities aspektams. Šią problemą siūloma išspręsti naudojant UML profilius, kurie suteikia metamodelių praplėtimo

galimybę taikant tam sukurtą mechanizmą. UML profilis apibūdinamas kaip „profile” stereotipą turintis UML paketas (angl. *package*), kuris gali praplėsti tiek kitą metamodelį, tiek kitą jau sukurtą profilį. UML profilis yra aprašomas naudojant tris pagrindinius praplėtimo mechanizmo elementus (Fuentes-Fernandez, Vallecillo-Moreno, 2004).

- 1) Stereotipai (angl. *stereotypes*) – nusako, kurios metamodelio metaklasės yra praplečiamos.
- 2) Žymėtiosios reikšmės (angl. *tagged values*) – nusako praplečiamų metamodelio metaklasėjų papildomus atributus ir jų formatus.
- 3) Apribojimai (angl. *constraints*) – nusako praplėstų metamodelio metaklasėjų apribojimo taisykles (OCL (angl. *Object Constraint Language*) ar kt.). Atvejais, kai tam tikros savybės negali būti išreiškiamos per stereotipus ar žymėtąsias reikšmes, šios savybės yra išreiškiamos apibrėžiant apribojimo taisykles (Silingas, 2009).

Iš pradžių kuriamas dalykinės srities koncepcinis metamodelis, kuriame identifikuojami tos srities konceptai, jų savybės bei tarpusavio ryšiai. Vėliau šis koncepcinis modelis yra paverčiamas į UML metamodelio klases, kurios praplečiamos atitinkamomis savybėmis. Tam reikalingos UML žinios (Selic, 2007), kadangi kiekvienam dalykinės srities konceptui turi būti identifikuojama ir priskiriama labiausiai tinkama UML metamodelio klasė. Straipsnyje (Silingas, 2009) pateikiamas organizacijos struktūros modeliavimui skirtas metamodelis, vaizduojantis: konceptualų metamodelį, išreikštą naudojant UML klasių elementus ir UML profiliu paremtą metamodelį. Dauguma konceptų vaizduojami stereotipizuotomis UML metaklasėmis, kurios yra praplečiamos papildomomis savybėmis – atributais (žymėtosiomis reikšmėmis); jų nėra UML metamodelyje nurodant jų vardus ir duomenų tipus. Tas pats atliekama ir su ryšiais: jeigu neužtenka UML metamodelio sąryšių, apibrėžiami papildomi ryšiai kuriant stereotipus, kurie praplečia UML metamodelio sąryšio metaklases. Profilių kūrimą UML metamodelio pagrindu palaiko tokie gerai žinomi ir plačiai naudojami įrankiai kaip: IBM *Eclipse*, IBM *Rational Software Architect*, *NoMagic MagicDraw*.

Bet koks metamodelių keitimas (taikant metamodelių suliejimą) sumažina sprendimo efektyvumą bei pritaikomumą, kadangi keičiantis standartų metamodeliams (pvz.: UML, BPMN, SBVR) turi būti keičiamas ir pats sprendimas. Metamodelių sujungimas apibrėžiant klases galimas tik tais atvejais, kai vieno metamodelio elementai egzistuoja kitame metamodelyje, išreikšti juodosios dėžės principu. Sąsajų naudojimas sujungiant metamodelius reikalauja papildomos struktūros, kurioje būtų saugoma metamodelius jungianti informacija. UML profilio mechanizmo taikymas nepakeičia UML metamodelio elementų semantinės prasmės, bet papildoma juos pagal konkrečius dalykinės srities poreikius, ir tai leidžia atlikti

nekuriant papildomų sąsajų, tik naudojant tris aprašymo mechanizmo elementus: stereotipas, žymėtąsias reikšmes bei apribojimus. *MagicDraw* CASE įrankis suteikia galimybę taikyti hibridinį DSML kūrimo metodą (Silingas et al., 2009), paremtą naujų metamodelių konstravimu iš pagrindų, bet UML profilių aprašymu ir redagavimu specifinėms reikmėms. Be to, *MagicDraw* turi integruotą QVT operacinių transformacijų procesorių, taip pat turi gerai ištobulintą grafinio modeliavimo vartotojo sąsają (tuo nepasizymi *IBM Eclipse* ar *IBM Rational Software Architect*).

### 3.2 Esamų integravimo sprendimų tyrimas

Didžiausias dėmesys modeliuojant veiklos procesus yra kreipiamas į dinamiškus aspektus. Šiuo atveju svarbiausias tampa pats veiklos procesas, jo eiga ir siekiamas tikslas. Veiklos procesų modeliavime pasigendama formalaus veiklos žodyno ir veiklos taisyklių apibrėžimo, kurį būtų galima pritaikyti toje pačioje aplinkoje kaip ir modeliuojamas veiklos procesas. Todėl šis procesas turėtų būti aprašytas naujomis galimybėmis, kurios leistų apibrėžti veiklos terminus bei apribojimus, kurie daro įtaką modeliuojamam veiklos procesui. Veiklos procesų modeliai be veiklos taisyklių dažnai yra nepilni, dviprasmiški ar netgi klaidinantys. Literatūros analizė (Wang et al., 2014) parodė, kad integruotas (veiklos procesų ir veiklos taisyklių) modeliavimas turi didžiausią vertę lyginant su kitais metodais. Deja, metodai ir įrankiai šioje srityje nėra pakankamai ištobulinti, didžiausio indėlio šioje srityje reikalaujama konceptuali lygiu.

Modeliuojant veiklos taisykles yra taikomas deklaratyvusis modeliavimas. Veiklos procesas yra išreiškiamas procedūrinio grafiniu būdu, o veiklos taisyklės – deklaratyviu būdu, kai visi veiklos ribojimai yra išreiškiami SBVR struktūrizuota anglų kalba. Esant tokiems vienos srities skirtingo aprašomojo pobūdžio modeliavimo būdams, atsiranda neatitikimų tarp gautų rezultatų. Veiklos modeliavimo ir taisyklių modeliavimo rezultatai yra saugomi atskirai, juos sieja tik dalykinė sritis, todėl klaidų, tokių kaip nesuderinamumas, pasitaiko dažnai. Deklaratyvusis veiklos procesų modeliavimas naudojant SBVR buvo pasiūlytas Goedertier (Goedertier, Vanthienen, 2007), kur buvo aprašyta 16 veiklos taisyklių tipų. Daugelyje integravimo siūlymų tarp veiklos modeliavimo ir taisyklių modeliavimo metodų siūloma aprašyti taisyklių modeliavimo galimybes pritaikant jas procedūriniam veiklos modeliavimui.

Kai dalykinėje srityje yra daug taisyklių, kurios laikui bėgant kinta, veiklos modelis atrodo didelis ir painus, nes jame yra pavaizduojami visi galimi veiklos scenarijai. Tam, kad procesą būtų galima keisti keičiant tik taisykles, jas reikia aprašyti deklaratyviu būdu (Goedertier, Vanthienen, 2007). Buvo pasiūlytas ir kitas metodas – taisyklėmis praturtintas BPMN (rBPMN) (Milanovic et al., 2011). Šis veiklos modeliavimo metodas pagrįstas BPMN ir R2ML integracija, į veiklos proceso modelį per sprendimo priėmimo taškus įtraukiant taisyklių šablonus. Šiuo atveju taikomi tik apibrėžti taisyklių šablonai, o R2ML neturi taisyklių ir jų pagrindinių konstruktyvų aukštos semantinės reprezentacijos galimybių kaip SBVR. Taip pat buvo ir kitų siūlymų taikyti taisyklių šablonus veiklos procesams (Graml et al., 2007), čia naudojama išplėsta (pritaikyta šablonams) BPMN notacija, realizacijai taikant BPEL procesų automatizavimo technologiją ir paprastomis taisyklėmis pagrįstų paslaugų

komponentą.

Gartner pranešime yra pateikti 7 integravimo būdai – scenarijai, kurie nusako galimybes veiklos procesus integruoti su veiklos taisyklėmis nuo paties statiškiausio iki dinamiškiausio būdo. Šie 7 scenarijai nusako, kiek veiklos proceso yra nusprendžiama perkelti veiklos taisyklėms (Koehler, 2011). Vėliau šie 7 scenarijai buvo sutraukti į 5 scenarijus (Koehler, 2012).

- 1 scenarijus. Įterptos taisyklės (angl. *Embedded Rules*). Šiuo atveju visi galimi keliai (visi veiklos apribojimai) yra tiesiogiai pavaizduojami veiklos proceso modelyje. Modelis gali tapti didelis, sudėtingai skaitomas, nes jame pavaizduojami visi galimi veiklos proceso scenarijai; todėl šis scenarijus yra tinkamas stabiliems, retai keičiamiems procesams. Pirmo scenarijaus pavyzdys pateiktas 10 paveiksle, kuriame nurodomos kainų taisyklės tiesiogiai pačiame veiklos procese.
- 2 scenarijus. Aiškiai apibrėžtos navigacijos taisyklės (angl. *Explicit Navigation Rules*). Šiuo atveju yra aiškiai apibrėžtos taisyklės, kurios valdo ir nukreipia proceso eigą kiekvienu proceso atveju. Jos atskirtos nuo proceso modelio ir yra palaikomos taisyklių komponentuose bei įvertinamos taisyklių variklio (angl. *Rule Engine*), kuris išskviečia veiklos taisyklės pagal reikalavimus. Modelis yra supaprastinamas įvedant veiklos taisyklės veiklą (angl. *Business Rule Activity*). Antro scenarijaus pavyzdys pateiktas 11 paveiksle. Kainos apskaičiavimo (angl. *calculate price*) veikla – veiklos taisyklės veikla, deleguojanti kainos apskaičiavimą veiklos taisyklių varikliui/interpretatoriui.
- 3 scenarijus. Sudėtinga navigacija ir analizė (angl. *Complex Navigation and Analysis*). Šiuo atveju proceso naudojamos taisyklės tampa sudėtingesnės, yra naudojami keli taisyklių rinkiniai. Proceso eiga gali būti skirtinga atskirais atvejais. Taisyklių kiekis labai išauga, ir tampa vis sunkiau išlaikyti nuoseklų taisyklių rinkinį. Taisyklė turi daugiau veiklos objekto atributo reikšmių, tai leidžia daryti tikslesnes išvadas. Reikalingas dažnesnis atnaujinimas dėl atributų reikšmių kaitos. Trečio scenarijaus pavyzdys pateiktas 12 paveiksle.
- 4 scenarijus. Taisyklėmis grindžiama proceso eiga (angl. *Rule-Guided Behaviour*). Šiuo atveju papildomos taisyklės, kaip ir navigacijos taisyklės, yra aiškiai apibrėžtos. Jos grupuojamos siekiant daryti įtaką proceso rezultatams. Procesas ne tik turi savo sudėtingas valdymo taisykles, bet dar ir sąveikauja su taisyklėmis, kurios naudojamos kitų procesų. Ketvirtojo scenarijaus pavyzdys pateiktas 13 paveiksle.
- 5–7 scenarijai. Nuo taisyklėmis pagrįstos kompozicijos ir paslaugų iki visiškos taisyklių dinamikos. Šiuo atveju yra sujungti paskutiniai trys scenarijai.
  - Taisyklėmis pagrįsta proceso elgsena (angl. *Rule-Driven Process Composites*). Proceso sudėtis priklauso nuo individualaus proceso atvejo, ir šis scenarijus yra atliekamas vykdymo metu. Veiklos taisyklės nusprendžia, kaip procesas turi plėtotis, kad prisidėtų prie siekiamo tikslo.



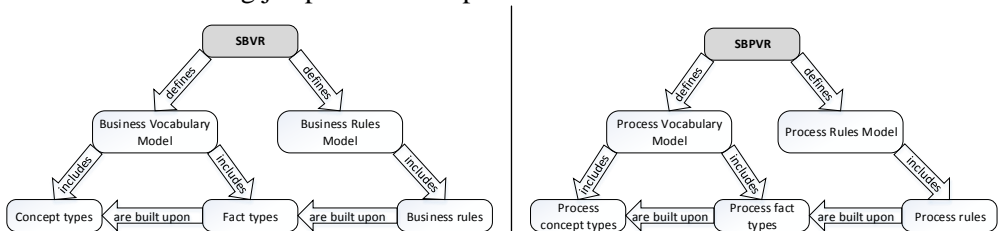
- Taisyklėmis pagrįstos paslaugos (angl. *Rule-Driven Services*). Proceso komponentai, labiau apimantys specifinę verslo situaciją. Paslaugų suderinamumas yra paremtas įvairių taisyklių paketais, kurie sugrupuojami tam, kad apimtų verslo sprendimą.
- Visiška taisyklių dinamika (angl. *Fully Rule-Dynamic*). Turintis mažiausią detalumo lygį, kur procesai dinamiškai konfigūruoja patys save kiekvienu atskiru proceso atveju ir yra labiau kontroliuojami apribojimais.

Šie scenarijai skiriasi vieni nuo kitų procesų valdymo galimybėmis – t. y. kaip išskiriami taisyklių rinkiniai, kurie valdo procesą sudarančių paslaugų vykdymą. Šias taisykles tikslinga atskirti nuo procesų/paslaugų komponentų. Autorių teigimu, parodyti šiuos taisyklių rinkinius ir procesų/paslaugų komponentus galima taikant BPMN 2 bendravimo diagramas.

Taikant deklaratyvųjį procesų aprašymą, galima atskirti veiklos apribojimų taisykles nuo procesų taisyklių. SBVR leidžia specifikuoti veiklos apribojimus – struktūrinius ir operacinius apribojimus, tačiau jis neapima procesų ir įvykių taisyklių. Procesų taisyklės apibrėžia veiklos procesų eigą, ir jos neturėtų persidengti su dalykinės srities ribojimų taisyklėmis. Agrawal (Agrawal, 2011) pateikė deklaratyvų procesų metamodelį, kurį galima integruoti su SBVR taisyklių metamodeliu (taikomas metamodelių suliejimo metodas). Bendras metamodelis buvo pavadintas SBPVR (angl. *Semantics of Business Process Vocabulary and Process Rules*). SBPVR turi konceptualius žodynus, kurie leidžia apibrėžti procesų elementus, jų semantiką ir procesų taisykles. SBPVR leidžia išskirti procesų žinias į tris atskiras dalis.

- Proceso koncepto tipus (angl. *Process concept type*), kurie apibrėžia dinaminę esybę proceso modelyje (pvz., užduotis, įvykis, sąveika).
- Proceso fakto tipus (nuo SBVR 1.1 versijos fakto tipo sąvoka pakeista į veiksmazodinio koncepto) (angl. *Process fact type*), kurie apibrėžia proceso koncepto tipo charakteristiką ar ryšį tarp jų.
- Procesų taisykles (angl. *Process rules*), kurios apibrėžia proceso struktūros ir eigos apribojimus.

SBPVR siūloma naudoti apibrėžiant/atskiriant proceso veiklos taisykles. SBVR ir SBPVR metodologijos pateiktos 3.1 paveiksle.

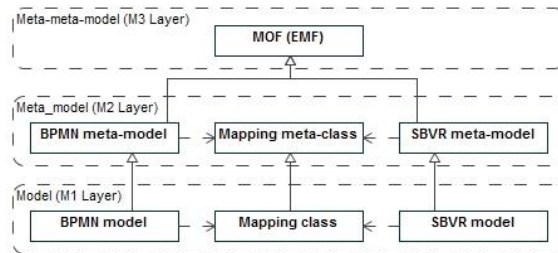


**3.1 pav.** SBVR ir SBPVR modeliai, remiantis (Agrawal, 2011)

Atlikta veiklos modeliavimo ir veiklos taisyklių modeliavimo kalbų analizė (zur Meuhlen et al., 2007) parodė, kad geriausias vaizdavimo galimybes atskleidžia BPMN ir SWRL arba BPMN ir SBVR kombinacijos. Kadangi SWRL apima tik dalį SBVR galimybių ir nėra tobulinama toliau, o SBVR nuolat tobulinama ir sulaukia vis

daugiau dėmesio, SBVR ir BPMN derinys yra tinkamesnis dalykinės srities veiklos procesams ir taisyklėms modeliuoti.

Skirtingi metamodelio integravimo metodai analizuojami (Skersys et al., 2012a; Skersys et al., 2012b). Autorių teigimu, geriausias būdas veiklos procesams ir taisyklėms integruoti – naudoti tarpinę duomenų struktūrą (sąsają), susiejančią skirtingų metamodelių elementus, nes jį taikant nereikia keisti standartinių metamodelių. M1 metalygmeniu (pagal OMG MOF metamodelių architektūros sluoksnius) realizuota metaklasė, susiejanti BPMN ir SBVR metamodelių elementus tarpusavyje (3.2 pav.). Metaklasę sudaro atributai, nusakantys, kokie BPMN metamodelio elementai atitinka (ar gali būti praplėsti) SBVR metamodelio elementais.



**3.2 pav.** Metamodelių integravimas naudojant sąsają, remiantis (Skersys et al., 2012a)

Vėlesniame darbe buvo atliktas veiklos žodynų ištraukimas iš veiklos procesų modelių (Skersys et al., 2013). Šiame etape taikytas elementų susiejimas žodyno lygmeniu, neapimant taisyklių. Tačiau šiuose darbuose nebuvo įvertinta UML profilio mechanizmo taikymo galimybė.

Binariniais veiksmožodiniai konceptais išreiškiami veiksmi, kuriuos aprašo operacinės taisyklės. Straipsnyje (Ceponiene et al., 2009) šie veiksmožodiniai konceptai sutapatunami su procesais ar įvykiais.

Veiklos taisyklių modeliavimas naudojant tam pritaikytą grafinę notaciją (Ross, 1997; Musham et al., 2008) parodė, kad toks modeliavimo būdas nėra patogus dėl didelio grafinių elementų kiekio notacijoje. Deklaratyvusis veiklos procesų aprašymas (Goedertier, Vanthienen, 2007; Roover et al., 2012) leidžia atskirti veiklos ribojimų ir veiklos procesų taisykles, tačiau jis nėra patogiausias būdas procesams vaizduoti. Veiklos dalyviai ir kompiuteriai geriau supranta procesus, pavaizduotus grafiniu būdu (Schacher, 2006; Ceponiene et al., 2009; Nemuraitė et al., 2010; Knowgravity, 2012, kt.). Šių autorių darbai grafiniam procesų vaizdavimui naudojo SBVR ir UML būsenų mašinas ar veiklą (angl. *activity*) modelius, tačiau BPMN yra labiau priimtina kaip veiklos procesų modeliavimo notacija. Be to, toks deklaratyvus sprendimas reikalautų modifikuoti SBVR standartą, o dėl to sumažėtų galimybes jį pritaikyti keičiantis UML ar BPMN metamodelių versijoms.

### 3.3 Esamų BPMN2 ir SBVR transformacijų tyrimas

Taikant du modeliavimo būdus, iškyla poreikis panaudoti tam tikrus elementus iš vieno modelio, kurie atspindi modeliuojamo objekto konceptus kitame modelyje. Dėl šios priežasties vieno modelio elementai turi būti transformuojami į kito modelio atitinkamus elementus pagal jų atliekamą funkciją tam, kad nebūtų prarasta

informacija ar iškreipta pirminio modelio prasmė.

Norint atlikti transformaciją tarp VPM ir VT visų pirma reikia nustatyti elementus, kurie iš vieno veiklos modeliavimo būdo gali būti transformuoti į kitą, kartu išlaikant tokią pačią modeliujamos veiklos prasmę. Tam galima pritaikyti integracijai skirtą elementų susiejimą – koks vienas tam tikras elementas šiuo atveju iš BPMN2 metamodelio atitinka kitą elementą iš SBVR metamodelio.

Automatinis metodas, pagal kurį galima BPMN VPM transformuoti į SBVR, pagrįstą natūralia kalba, yra aprašytas Malik ir Bajwa Sarwan (Malik, Bajwa, 2012; Malik, Bajwa, 2013). Pagal pateiktą transformacijos idėją buvo sukurtas įrankis *BR-Generator*. Šio metodo trūkumas yra tas, kad transformacijai naudojamas ribotas BPMN elementų skaičius. Pagal šį metodą yra naudojami tokie BPMN elementai: pradžios įvykis, veikla, sekos srautas, sąlyginis srautas, pranešimo srautas ir asociacija, plaukimo takeliai, duomenų objektai, grupės, anotacijos. Taip pat pateiktas VPM pavyzdys buvo sukurtas netaikant gerosios modeliavimo praktikos: veiklos pavadinimai yra pateikti blogu formatu – naudojamos daiktavardinės, o ne veiksmazodinės formos. Tokiu atveju, naudojant pateiktą įrankį, kiekvienam vartotojui reikia susipažinti su veiklos proceso elementų vardų užrašymo taisyklėmis, kitu atveju gauti natūralios kalbos sakiniai nebus korektiški.

Vėlesniame tų pačių autorių straipsnyje (Malik, Bajwa, 2013) pateikiama tikslesnė BPMN ir SBVR susiejimo informacija, taikoma geroji modeliavimo praktika, tačiau susiejamų elementų kiekis išlieka ribotas, nes neįvertinami visi galimi elementų tipai (pvz., gaunamieji ir siunčiamieji įvykiai, sprendimo priėmimo taškų tipai, užduočių tipai). BPMN modelio gavimui yra naudojamas *Enterprise Architecture* įrankis. BPMN transformacijai į SBVR naudojama anglų kalba, tačiau gramatikos klaidos (tokios, kaip 3-iojo asmens veiksmazodžio forma) išlieka, jos bus taisomos tolimesniuose autorių darbuose. Pristatytas pusiau automatinis veiklos žodyno išgavimas iš veiklos modelio (Skersys et al., 2013), tačiau šis metodas neapima taisyklių ir yra pusiau automatinis.

Norint gauti sklandų bei rišlų tekstą transformuojant iš VPM, reikia taikyti lingvistinius metodus tam, kad būtų galima išskirti sakinio dalis. Šie metodai priklauso nuo pasirinktos kalbos, nuo to priklauso ir metodo sudėtingumas (Leopold et al., 2012). Autoriai pristatė natūralios kalbos generatorių (angl. *Natural Language Generator* (NLG)), kuriam pritaikė natūralios kalbos analizę, grafų dekompozicijos metodus bei lingvistinį karkasą tekstams generuoti.

Tačiau netgi SBVR struktūrizuotai kalbai nepakanka praktinės lingvistinės analizės BPMN2 VPM, nes nėra atsižvelgiama į linksniavimą, laikus, daugiskaitą–vienaskaitą, būdingus natūraliai kalbai. Tai galioja ir anglų kalbai, nors jos gramatika nėra tokia sudėtinga (lyginant su lietuvių kalba) (Kleiner et al., 2009). Tačiau dalis natūralios kalbos apdoravimo yra taikoma daugelyje BPMN2 VPM transformacijų į SBVR VŽ&VT metodų, pvz.: (Skersys et al., 2012b; Wu et al., 2011). Semantika pagrįstas karkasas, kuriame taikomos semantinės žiniatinklio technologijos, buvo pristatytas (Fortineau et al., 2013), kuris iš natūralios kalbos teiginių formuoja formalias taisykles. Viena pagrindinių šio darbo priežasčių – veiklos ekspertų ir IT ekspertų nesusikalbėjimas, tačiau šiuo atveju VPM lieka atskirtas.

Aprašytas atvirkštinės transformacijos, transformuojant struktūrizuotos

natūralios kalbos tekstą į VPM, metodas (Friedrich et al., 2011), kuriame taikomi sudėtingi lingvistiniai metodai. Bandymai parodė, kad sugeneruojama vidutiniškai 77 % veiklos procesų modelio iš pateikiamo šaltinio. Sudėtingų lingvistinių metodų taikymas (Leopold et al., 2012) negarantuoja išėigos modelio išsamumo bei patikimumo.

Lingvistinių metodų alternatyva – taikyti šablonus taisyklėms aprašyti. Taikant tik šablonus teksto gavimas yra paprastesnis, taip pat reikia turėti šablonus visoms įmanomoms elementų kombinacijoms, kad gaunamas tekstas būtų korektiškas lyginant su veiklos diagrama.

Atvirkštinės transformacijos (iš SBVR VŽ&VT į BPMN2 VPM) buvo analizuojamos (Eder et al., 2008; Tantan, Akoka, 2014; Steen et al., 2010; Raj et al., 2008; Wu et al., 2011). Eden, Steen ir Raj metodai grindžiami „if condition then action“ veiklos taisyklių konstrukcija. Toks metodas nėra suderinamas su BPMN2, kadangi BPMN2 veiklos procesai yra inicijuojami ne sąlygų, o įvykių, kai konkrečios sąlygos yra tenkinamos ir konkretūs duomenų objektai yra prieinami. Tantan metodu siūloma padengti kai kurias taisykles, taikyti jas kaip prielaidas arba sąlygas po veiklos vykdymo ir naudoti veiklos taisyklių užduotis (angl. *Business Rule Tasks*) vietoj sprendimo priėmimo taškų (angl. *gateways*). Toks supaprastinimas tik paslepia dalį proceso logikos, kuri turėtų būti pavaizduojama BPMN2 VMP elementais. Siūlomame sprendime nėra sisteminio požiūrio į problemą.

Aprašyti siūlymai transformuoti SBVR VT į taisykles, kurios būtų inicijuojamos VT variklių (Steen et al., 2010; Eder et al., 2008; Raj et al., 2008), darbo eigos (angl. *workflow*) variklių (Habish et al., 2010, OPAALS, 2008), įvykių apdorojimo aplinkų (Roover et al., 2012) tam, kad nebūtų prarasta vykdymo semantika. Nerasta, kad būtų nagrinėjamos visų tipų SBVR deontinės taisyklės (įsipareigojimas, leidimas ir draudimas), kurios gali būti panaudojamos netgi tiksliau išreiškiant BPMN2 vykdymo semantiką. Šios VT gali būti transformuojamos atgal į BPMN2 modelį, taip pat gali būti naudojamos dirbant su BPEL vykdymo varikliais, tai siūloma kaip standartas dirbant su VPM (Recker, Mendling, 2006); įvykių apdorojimo aplinkomis; sutartimi paremtais programinės įrangos metodais (OWE et al., 2007; Wilke et al., 2010).

Norint atlikti dvikryptį transformavimą yra būtinybė turėti du atskirus žodynus, nes reikėtų atskirti VPT nuo VT. Tokiu būdu būtų sudaromos galimybės atvirkštiniam transformavimui neprarandant informacijos apie turėtą procesą (pvz., esant sudėtinėms veikloms, kurios detalizuojasi žemesniu hierarchijos lygiu).

### **3.4 Atlikto tyrimo apibendrinimas ir išvados**

BPMN2 ir SBVR modeliavimo kalbos modeliais grindžiamoje architektūroje (MDA) užima tą pačią vietą (abi yra CIM lygmenyje), be to, yra paremtos tuo pačiu meta-metamodeliu – MOF, todėl šių modelių integracija yra galima bei nagrinėjama įvairiuose tyrimuose. Atlikus esamų sprendimų literatūros analizę buvo išskirti pagrindiniai egzistuojantys sprendimai. Šių sprendimų privalumai bei trūkumai, kartu su nauju sprendimu, pateikti 3.1 lentelėje.

### 3.1 lentelė. Esamų sprendimų analizė

<b>Metodas</b>	<b>Privalumai</b>	<b>Trūkumai</b>
VT ir VPM integravimo scenarijai (Koehler, 2011; Koehler, 2012)	Keli modelių susiejimo variantai	Nėra informacijos, kaip įgyvendinti
VT grafinis pavaizdavimas (Ross, 1997)	Vizualus VT pavaizdavimas	Praktiškai neįgyvendinamas dėl per didelės apimties notacijos elementų; nėra ryšio su VPM
Integracija ir abipusė transformacija, pagrįsta SBVR ir BPMN metamodeliais (Malik, Bajwa, 2012; Malik, Bajwa, 2013)	Standartais pagrįstas sprendimas, pritaikomas CASE įrankiuose	Neišbaigtas sprendimas įvertinant ne visus VPM elementus; formuojamos VT, VŽ nepateikiamas atskirai
SBVR metamodelio praplėtimas (Agrawal, 2011)	Pilnas elementų susiejimas	Procesų Ž&T taikomas atskiras SBPVR metamodelis (taikant metamodelių suliejimą), dėl to palaikymas tampa sudėtingas dėl sprendime taikomų standartų (BPMN, SBVR) specifikacijų evoliucijos
Deklaratyvusis veiklos procesų modeliavimas (Goedertier, Vanthienen, 2007)	VT ir VPM formali integracija	Sunkiai įgyvendinamas praktiškai, nes veiklos procesai geriau suprantami, kai atvaizduojami grafiškai
Papildomos struktūros naudojimas elementų susiejimui; lingvistinių metodų taikymas (Skersys et al., 2012a; Skersys et al., 2012b)	Pritaikomas be papildomų reikalavimų modeliavimui; nereikia BPMN ar SBVR metamodelių pakeitimų	Nėra garantijos dėl išbaigtumo ir patikimumo; neįmanoma įvertinti visų vardų suteikimo variantų; formuojamas VŽ, VT gali būti kuriamos rankiniu būdu pagal pateiktus šablonus
Natūralios kalbos tekstai iš BPMN ir atvirkščiai (Friedrich et al., 2011; Leopold et al., 2012)	Nereikia jokių papildomų reikalavimų VPM	Reikia sudėtingų lingvistinių metodų; nėra garantijos dėl gaunamo rezultato užbaigtumo ir patikimumo; formuojamos VT, VŽ nepateikiamas atskirai; tik tekstinė VT reprezentacija
Siūlomas sprendimas	VPM ir VT integracija bei abipusės transformacijos nekeičiant metamodelių; formuoja VŽ ir VT apimant tiek apribojimo, tiek proceso taisykles; apima visą VPM elementų aibę; modelio	Reikia laikytis tam tikrų reikalavimų kuriant VPM

Metodas	Privalumai	Trūkumai
	elementų atsekamumo galimybė; netaikomi lingvistiniai metodai; VŽ pateikimas tiek tekstiniu, tiek grafiniu būdu; procesų modelio Ž&T formuojamos atskirame žodyne, nenaudojant atskiro metamodelio	

Vienuose aprašytuose sprendimuose nepateikiama išsami informacija, kaip juos būtų galima įgyvendinti, kiti yra sunkiai įgyvendinami praktiškai, sudėtingi dėl SBVR ir BPMN standartų evoliucijos, tretiems reikia sudėtingų lingvistinių metodų. Siūlomas naujas sprendimas leistų BPMN ir SBVR integraciją, transformaciją iš BPMN VPM į SBVR VŽ&VT bei suteiktų atvirkštinės transformacijos galimybę. Sprendimo privalumai: nekeičiami metamodeliai; formuojamas tiek VŽ, tiek ir VT apimant apribojimo ir procesų taisyklės (jų neatskiriant, kad būtų gaunama išsami veiklos proceso specifikacija ir tam naudojant vieną metamodelį); apimama pilna BPMN elementų aibė; suteikia galimybę užmegzti atsekamumo ryšius tarp modelių elementų; VŽ gali būti pateiktas grafiniu būdu; netaikomi jokie lingvistiniai metodai; procesų modelio Ž&T formuoti (papildomai VPM modelio informacijai išsaugoti) naudojamas atskiras žodynas, bet ne atskiras metamodelis. Kaip šio sprendimo trūkumas yra įvardijami reikalavimai BPMN VPM, kuriuos būtina taikyti norint gauti semantiškai korektišką SBVR VŽ&VT. Šie reikalavimai neprieštarauja gerajai veiklos procesų modeliavimo praktikai, priešingai – ją papildo, nes laikantis šių reikalavimų grafinio procesų modelio tam tikrų elementų tipų vardai turi vienodą formatą, pats modelis tampa tvarkingesnis bei lengviau skaitomas.

Sprendimas pagrįstas (Ceponiene et al., 2009) atskiriant veiklos taisyklės, kurios apibrėžia veiklos proceso vykdymą, nuo veiklos apribojimo taisyklių. Toks atskyrimas leidžia ne tik geriau suprasti modeliavimą, bet ir identifikuoti pasikeitimus, pvz.: keičiantis reikalavimams keičiant veiklos taisyklės pritaikyti tai veiklos procesų modeliui (Popp, Kaindl, 2014). Sudarant išsamią VPM specifikaciją abiejų tipų taisyklės yra talpinamos viename žodyne.

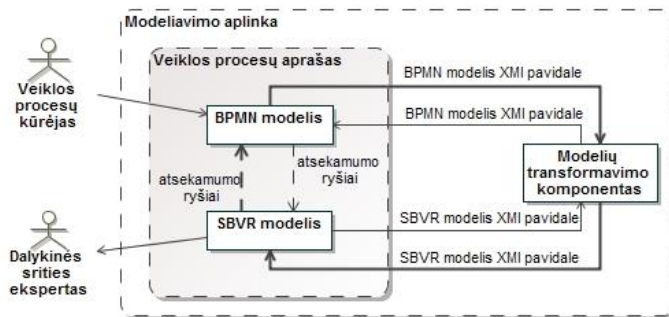
## 4 BPMN2 IR SBVR INTEGRACIJA IR TRANSFORMACIJOS

Šiame skyriuje pateikiama principinė sprendimo schema, BPMN2 ir SBVR integravimo ir transformavimo sprendimas: sudarytos BPMN2 ir SBVR elementų atitikimo lentelės, suformuluotos transformacijos taisyklės. Skyriuje nagrinėjamos BPMN VPM į SBVR VŽ&VT ir jai atvirkštinė transformacijos. Kiekviena iš jų dar skaidomos į dvi įvertinus antrosios transformacijos ribotumus panaudojant tik standartinį SBVR VŽ&VT. Skyriuje aprašomos transformacijos: 1) BPMN VPM į SBVR VŽ&VT; 2) BPMN VPM į BPMN VPM Ž&T; 3) SBVR VŽ&VT į BPMN VPM; 4) BPMN VPM Ž&T į BPMN VPM. Pagrindinis dėmesys yra skiriamas 1 transformacijai, kadangi ji yra dažniausiai reikalinga praktikoje norint modeliuoti dalykinę sritį skirtingais aspektais. Skyriaus pabaigoje pateikiami sukurto sprendimo taikymo scenarijai, apimantys 1 ir 2 transformacijas. Papildomai įtraukiama 2 transformacija, ji suteikia galimybę išsaugoti informaciją, kuri būtų prarandama atlikus pirmąją transformaciją. Šios informacijos nuostoliai buvo identifiukuoti nagrinėjant 3 transformaciją, o jos atkūrimui aprašyta 4 transformacija.

Skyriuje naudojami BPMN2 elementai, jų grupės bei SBVR elementai atitikimų lentelėse bei transformacijų matricose pateikti anglų kalba (kaip pateikiama standartų specifikacijose), visų elementų vertimai pateikti 9.5 priedų poskyryje 9.1–9.3 lentelėse.

### 4.1 Principinė sprendimo schema ir kontekstas

Abstraktus BPMN2 ir SBVR integravimo bei transformavimo vienoje modeliavimo aplinkoje kontekstas pateiktas 4.1 paveiksle.



4.1 pav. BPMN2 ir SBVR integravimo bei transformavimo kontekstas

Principinė sprendimo schema parodo BPMN2 ir SBVR integravimo bei transformavimo kontekstą: veiklos procesų kūrėjas atsakingas už BPMN2 modelio sukūrimą, SBVR modelis gali būti pateikiamas dalykinės srities ekspertui norint patikrinti sumodeliuotą veiklos procesą (validavimui). Darbo pagrindas – BPMN2 transformacija į SBVR (paveiksle pateikta ryškintomis linijomis), tačiau papildomai yra nagrinėjama ir jai atvirkštinė transformacija.

Išsamesnis BPMN2 transformavimo į SBVR kontekstas (sukurtų transformacijų taikymas) pateiktas 4.6 poskyryje, po to, kai pateikiama informacija apie priimtus sprendimus, susijusius su transformacijomis.

## 4.2 BPMN2 ir SBVR integravimas

Poskyryje pateikiamas BPMN2 ir SBVR modelių integracijos aprašas, kurio tikslas – išskirti BPMN2 modelio ir SBVR veiklos žodyno konceptų atitikimus, kurie leistų susieti šiuos du modelius, taip pat būtų naudojami kitame darbo etape – modelių transformacijų taisyklių sudarymui. Modelių integracijos įgyvendinimui pasirinktas ir aprašytas SBVR profilis, kuris suteikia galimybę BPMN2 ir SBVR modelius susieti vienoje *MagicDraw* CASE įrankio aplinkoje. Poskyrio pabaigoje pateikiamas iliustracinis pavyzdys, kuris buvo kuriamas remiantis SBVR standarto specifikacijoje (OMG, 2008) pateiktu „EU-Rent“ (automobilių nuomos) pavyzdžiu.

### 4.2.1 BPMN2 modelio ir SBVR veiklos žodyno konceptų atitikimas

Norint išsamiai aprašyti veiklos procesą buvo nuspręsta naudoti tris žodynus: nagrinėjamos dalykinės srities veiklos žodyną, kuriame dalykinė sritis aprašoma standartinėmis SBVR konstrukcijomis; BPMN2 konceptų metažodyną, kuriame aprašomi veiklos procesų diagramos metamodelio elementai SBVR konstrukcijomis, kurios naudojamos BPMN2 procesų žodynams sudaryti; ir BPMN2 procesų modelio žodyną, kuris aprašomas naudojant BPMN2 konceptų metažodyną. Paskutiniai du žodynai gali būti laikomi tekstone VPM reprezentacija.

Naudojant tik įprastą SBVR veiklos žodyną, ne visų BPMN2 modelio elementų atitikmenis galima nustatyti arba tie atitikmenys yra netiesioginiai (išvedami pagal taisykles, kurias nustato sprendimo kūrėjai), todėl veiklos procesų ir žodyno integravimas nepilnas, daug sunkiau pasiekiamas ir priklausomas nuo kūrėjų priimtų taisyklių. BPMN2 modelio ir SBVR veiklos žodyno konceptų atitikimas pateiktas 4.1 lentelėje. Atitikimas tarp konceptų buvo nagrinėjamas dviem laipsniais: pilno (PA) ir dalinio (DA) atitikimo. Tais atvejais, kai atitikimas nėra galimas, naudojamas „-“ simbolis. BPMN2 elementų pavadinimai, kurie pateikiami lentelėje, neverčiami į lietuvių kalbą, naudojami BPMN2 tokie elementų vardai, kaip jie yra įvardijami metamodelyje (ši taisyklė galioja visai darbo apimčiai).

4.1 lentelė. BPMN2 modelio ir SBVR veiklos žodyno konceptų atitikimas

BPMN2 elementas	SBVR elementas			Pavyzdys
	General Concept	Verb Concept	Business Rule	
<b>Flow Objects</b>				
Event	DA	PA/DA	DA	SBVR veiksmožodinis konceptas (unarinis), 2 SBVR bendriniai konceptai ir SBVR veiksmožodinis konceptas (binarinis) <a href="#">car booking request is_received</a> <a href="#">car booking request received</a> Concept type: <a href="#">state</a> <a href="#">car booking request has_state received</a>
Activity	DA	DA	DA	SBVR veiksmožodinio koncepto dalis <a href="#">branch create rental_contract</a>
<i>Message</i>	PA	DA	DA	SBVR bendrinis konceptas



BPMN2 elementas	SBVR elementas			Pavyzdys
	General Concept	Verb Concept	Business Rule	
				<a href="#">rental contract proposal</a>
<b>Connecting Objects</b>				
<i>SequenceFlow</i>	DA	DA	PA	SBVR deontinė veiklos taisyklė, nusakanti proceso eigą It is obligatory that <a href="#">car booking request is canceled</a> after <a href="#">rental contract is rejected</a> .
<i>SequenceFlow (with Condition from Gateway)</i>	DA	DA	PA	SBVR deontinė veiklos taisyklė, nusakanti apribojimą It is obligatory that <a href="#">branch accept car booking request</a> if <a href="#">car booking request is valid</a> .
<i>MessageFlow</i>	DA	PA	DA	Pranešimo srautas be pranešimo aprašomas SBVR veiksmožodiniu konceptu (binariniu) ir deontine taisykle <a href="#">renter receive message from branch</a> It is obligatory that <a href="#">renter receive message from branch</a> after <a href="#">car booking request is accepted</a> Pranešimo srautas su pranešimu aprašomas 2 SBVR veiksmožodiniais konceptais (binariniais) ir 2 deontinėmis taisyklėmis <a href="#">renter send car booking acceptance</a> <a href="#">branch receive car booking acceptance</a> It is obligatory that <a href="#">branch send car booking acceptance</a> after <a href="#">car booking request is accepted</a> It is obligatory that <a href="#">renter receive car booking acceptance</a> after <a href="#">branch send car booking acceptance</a>
<i>Association</i>	DA	DA	PA	Asociacija, sujungta su ribiniu kompensavimo įvykiu (CompensationBoundaryEvent), aprašoma deontine veiklos taisykle, sudaryta iš 2 SBVR veiksmožodinių konceptų (binarinių) ir 1 SBVR veiksmožodinio koncepto (unarinio) It is obligatory that <a href="#">branch cancel car booking request</a> if <a href="#">compensation is required</a> when <a href="#">branch approve car booking request</a> Asociacija, sujungta su įvykiu, kuris siunčia pranešimą, aprašoma SBVR veiksmožodiniu konceptu (binariniu)

BPMN2 elementas	SBVR elementas			Pavyzdys
	General Concept	Verb Concept	Business Rule	
				<a href="#">car booking acceptance include accepted car booking request</a>
<i>DataAssociation</i>	DA	DA	PA	SBVR veiksmožodiniu konceptu (binariniu) <a href="#">car booking acceptance include accepted car booking request</a>
<b>Swimlanes</b>				
<i>Lane</i>	PA	DA	-	<p>Jei konteineriye (<i>Lane</i>) nurodytas resursas (pvz., <i>branch</i>), ir vardas, pvz., <i>rental provider</i>, sukuriamas SBVR bendrinis konceptas, vaidmuo ir agregavimo ryšys su bendrinium konceptu, atitinkančiu konteineri (Pool) (pvz., <a href="#">rental company</a>): <a href="#">rental company</a> <a href="#">branch</a> <a href="#">rental provider</a> Concept type: <a href="#">role</a> General concept: <a href="#">branch</a> <a href="#">rental company include rental provider</a></p> <p>Jei <i>Lane</i> nurodytas tik resursas (pvz., <i>branch</i>), sukuriamas SBVR bendrinis konceptas ir agregavimo ryšys su bendrinium konceptu, atitinkančiu konteineri (Pool) (pvz., <a href="#">rental company</a>): <a href="#">rental company</a> <a href="#">branch</a> <a href="#">rental company include branch</a></p> <p>Jei konteineriye (<i>Lane</i>) nurodytas tik vardas, pvz., <i>rental provider</i>, sukuriamas SBVR bendrinis konceptas ir agregavimo ryšys su bendrinium konceptu, atitinkančiu konteineri (Pool) (pvz., <i>rental company</i>): <a href="#">rental company</a> <a href="#">rental provider</a> <a href="#">rental company include rental provider</a></p>
<i>Pool</i>	PA	DA	-	<p>Jei konteineriye (<i>Pool</i>) nurodytas resursas (pvz., <i>rental company</i>), ir vardas, pvz., <i>rental provider</i>, sukuriamas SBVR bendrinis konceptas ir vaidmuo:</p>

BPMN2 elementas	SBVR elementas			Pavyzdys
	General Concept	Verb Concept	Business Rule	
				<a href="#">rental company</a> <a href="#">rental provider</a> Concept type: <a href="#">role</a> General concept: <a href="#">rental company</a> Jei konteineryje ( <i>Pool</i> ) nurodytas tik resursas (pvz., <i>rental company</i> ), sukuriamas SBVR bendrinis konceptas <a href="#">rental company</a> Jei konteineryje ( <i>Pool</i> ) nurodytas tik vardas, pvz., <i>rental provider</i> , sukuriamas SBVR bendrinis konceptas <a href="#">rental provider</a>
<b>Artifacts</b>				
<i>DataObject</i> ( <i>withState</i> ), jei <i>DataObject</i> turi 1 būseną	PA	DA	DA	Išvedamas 1 SBVR veiksmožodinis konceptas (binarinis) ir 3 SBVR bendriniai konceptai <a href="#">car booking request received</a> Concept type: <a href="#">state</a> <a href="#">car booking request has_state received received car booking request</a> General concept: <a href="#">car booking request</a>
<i>DataObject</i> ( <i>withState</i> ), jei <i>DataObject</i> turi daugiau nei 1 būseną	PA	DA	DA	Išvedame 1 bendrinį konceptą (atitinkantį <i>DataObject</i> ), sukuriame kategorizavimo schemą pagal būsenų tipus ( <i>state type</i> ), kiekvienai būsenai sukuriame po 2 bendrinius konceptus su specialiomis savybėmis ir agregavimo ryšį: <a href="#">car booking request state type</a> Concept type: <a href="#">categorization type</a> <a href="#">Car booking request by state type</a> Necessity: <a href="#">categorization scheme for general concept car booking request that subdivides car booking request by state type received</a> Concept type: <a href="#">state</a> <a href="#">car booking request has_state received received car booking request</a> General concept: <a href="#">car booking request</a> Necessity: <a href="#">is_included_in categorization scheme Car booking request by state type</a>

BPMN2 elementas	SBVR elementas			Pavyzdys
	General Concept	Verb Concept	Business Rule	
				<a href="#">accepted</a> Concept type: <a href="#">state</a> <a href="#">car booking request has_state accepted</a> <a href="#">accepted car booking request</a> General concept: <a href="#">car booking request</a> Necessity: <a href="#">is included in categorization scheme</a> <a href="#">Car booking request by state type</a>
<i>DataStore</i>	PA	DA	-	SBVR bendrinis konceptas <a href="#">customer data store</a>
<i>Group</i>	PA	DA	-	SBVR bendrinis konceptas
<i>TextAnnotation</i>	-	-	-	SBVR Note Papildomos informacijos aprašymas laisva forma.

Veiklos taisyklių, kurios apibrėžia veiklos proceso vykdymą, ir veiklos apribojimo taisyklių atskyrimas atliekamas naudojant SBVR raktažodžius (jei taisyklėje naudojamas „if“ raktažodis – tada taisyklė yra ribojimo, jei „after“ – taisyklė nusako proceso eigą), visos taisyklės talpinamos viename žodyne.

Naudojant tik SBVR veiklos žodyną negalima aprašyti, iš ko susideda BPMN2 procesai, subprocesai ir grupės; kokių dalykinės srities esybių būsenas vaizduoja duomenų objektai; kokie pavaizduoti sprendimų taškai bei įvykių tipai ir t. t. Dėl šios priežasties ši informacijos dalis apie BPMN2 VPM yra prarandama. Tam, kad ši informacija išliktų, yra naudojami aprašyti du papildomi žodynai.

BPMN2 konceptų ir SBVR konceptų atitikimas naudojant BPMN2 konceptų metažodyną pateikiamas 4.2 lentelėje (konceptų tipai aprašomi BPMN2 konceptų metažodyne). Įtraukti tik tie elementai, kurie neaprašomi veiklos žodyne, nes atspindi tik procesams būdingas savybes ir taisykles.

**4.2 lentelė.** BPMN2 modelio elementų ir SBVR konceptų atitikimas naudojant BPMN2 konceptų metažodyną

BPMN2 elementas	Atitikimas	SBVR konceptas	Pavyzdys
<b>Flow Objects</b>			
<i>Event</i>	PA	<i>General Concept</i>	Aprašomi visi įvykių tipai, pvz.: <a href="#">car booking request is received</a> Concept type: <a href="#">message_start_event</a>
<i>Activity</i>	PA	<i>General Concept</i>	Aprašomi visi veiklos tipai, pvz.: <a href="#">create_rental_contract</a> Concept type: <a href="#">subprocess</a>
<i>Message</i>	PA	<i>General Concept</i>	<a href="#">rental_contract_proposal</a> Concept type: <a href="#">message</a>
<b>Connecting Objects</b>			
<i>Association</i>	PA	<i>Verb Concept, Business Rule</i>	<a href="#">branch get_data_output approved rental contract</a> It is obligatory that <a href="#">branch get_data_output approved rental contract</a> after <a href="#">branch approve rental contract</a>

BPMN2 elementas	Atitikimas	SBVR konceptas	Pavyzdys
<i>DataAssociation</i>	PA	<i>Verb Concept, Business Rule</i>	<a href="#">rental_provider</a> <a href="#">provide_data_input</a> <a href="#">received_car_booking_request</a> It is obligatory that <a href="#">branch</a> <a href="#">provide_data_input</a> <a href="#">received_car_booking_request</a> when <a href="#">branch</a> <a href="#">approve_car_booking_request</a>
<b>Swimlanes</b>			
<i>Lane</i>	PA	<i>General Concept</i>	<a href="#">branch</a> Concept type: <a href="#">participant</a>
<i>Pool</i>	PA	<i>General Concept</i>	<a href="#">branch_unit</a> Concept type: <a href="#">participant</a>
<b>Artifacts</b>			
<i>DataObject</i>	PA	<i>General Concept</i>	<a href="#">car_booking_request_received</a> Concept type: <a href="#">data_object</a>
<i>DataStore</i>	PA	<i>General Concept</i>	<a href="#">customer_data_store</a> Concept type: <a href="#">data_object</a>

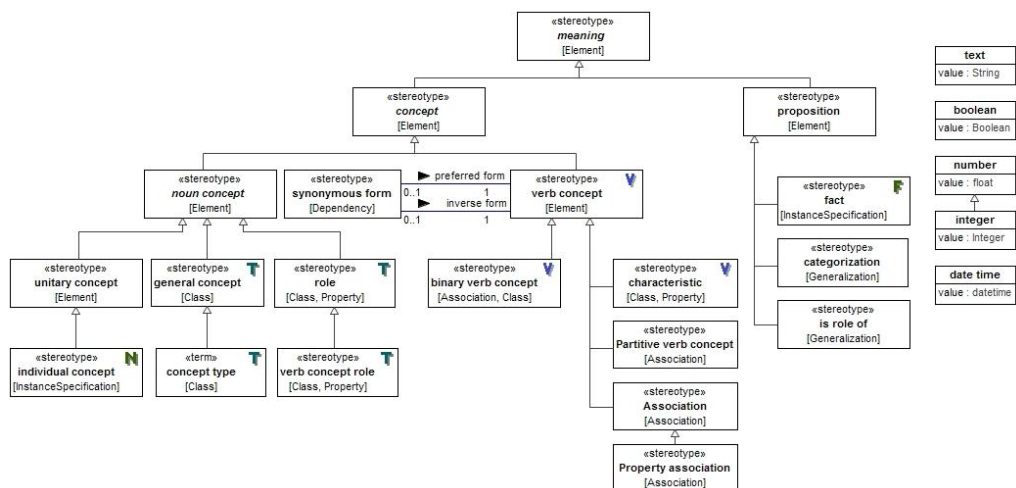
Papildomai įvedami veiksmožodiniai konceptai, kad nebūtų prarandama hierarchijos informacija bei informacija, kuri nusako, koks elementas priklauso kokiam proceso diagramai:

[process](#) [`approve\\_car\\_booking\\_request`](#) [include task](#) [`check renter driving licence`](#)  
[process](#) [`approve\\_car\\_booking\\_request`](#) [include task](#) [`check renter credit card`](#)

#### 4.2.2 SBVR profilis

Sprendimui įgyvendinti taikomas UML profilių mechanizmas *MagicDraw* įrankio aplinkoje. Toks pasirinkimas buvo padarytas todėl, kad naudojama DSL (angl. *Domain Specific Language*) leidžia neprogramuojant realizuoti siūlomą sprendimą greičiau, o transformacijoms galima naudoti QVT (angl. *Query/View/Transformation*) transformacijų kalbą, kuri yra taip pat tobulinama OMG grupės.

SBVR profilio (Mickeviciute et al., 2014b; Skersys et al., 2015) (4.2 pav.) pagrindas yra UML profilis. BPMN2 veiklos žodynui aprašyti sukurtas SBVR profilis išplečiantis BPMN2 profilis, kuriame realizuotas SBVR ir BPMN2 suderinamumas.

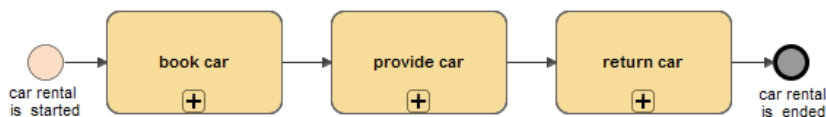


4. 2 pav. SBVR profilis (Mickeviciute et al., 2014b)

SBVR profilis yra realizuotas naudojant stereotipus atitinkamiems SBVR konceptams apibrėžti. Kiekvienas apibrėžtas SBVR profilio stereotipo atitinkmuo UML profilio klase ar atributu yra parodytas elemento (stačiakampio) apačioje, laužtiniuose skliaustuose.

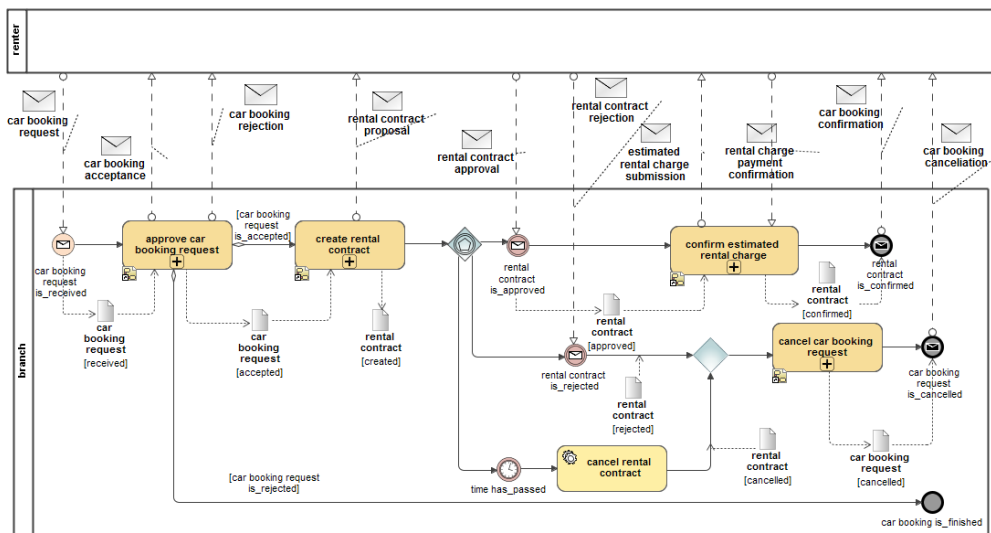
#### 4.2.3 Iliustracinis BPMN2 ir SBVR pavyzdys

Iliustracinis BPMN2 modelis buvo kuriamas pagal SBVR specifikacijoje esantį aprašą, kuris aprašo automobilių nuomos procesus. Šis modelis bus naudojamas tolimesniuose darbo etapuose, kaip pavyzdinis realios dalykinės srities modelis kuriamam sprendimui iliustruoti bei patikrinti. Modelis susideda iš trijų proceso hierarchijos lygių ir apima visą automobilių nuomos procesą nuo nuomos užklauskos pateikimo nuomos punktu iki automobilio grąžinimo nuomos paslaugą teikiančiai įmonei. Automobilio nuomos pirmas (aukščiausias) hierarchijos lygio procesas pateiktas 4.3 paveiksle.



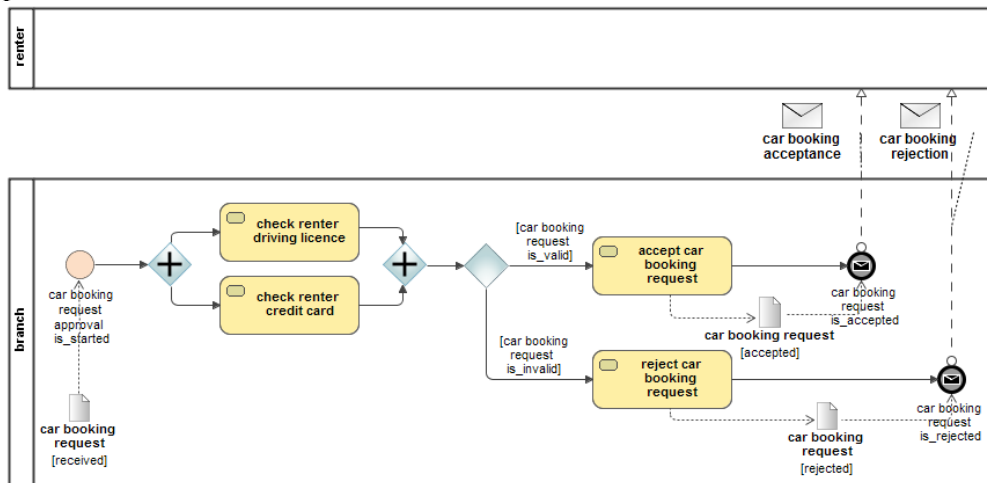
4.3 pav. Proceso „rent car“ diagrama (pirmasis hierarchijos lygis)

Antro hierarchijos lygio procesas pateiktas 4.4 paveiksle. Jame detalizuojamas pirmojo hierarchijos lygio subprocesas „book car“. Šioje proceso diagramoje pateikiami visi veiksmai, kurie atliekami nuo vartotojo užklauskos iki nuomos sutarties patvirtinimo arba užklauskos atmetimo.



4.4 pav. Subproceso „book car“ diagrama (antrasis hierarchijos lygis)

Trečiojo hierarchijos lygio subprocesas pateiktas 4.5 paveiksle. Jame detalizuojamas antrojo hierarchijos lygio subprocesas „approve car booking request“. Šioje proceso diagramoje atliekami veiksmai iki automobilio nuomos užklauso patvirtinimo arba atmetimo.



4.5 pav. Subproceso „approve car booking request“ modelis (trečiasis hierarchijos lygis)

Automobilių nuomos srities SBVR veiklos žodynas ir veiklos taisyklės pagal 4.5 pav. parodytą proceso diagramą pateiktos 4.3 lentelėje.

4.3 lentelė. SBVR veiklos žodynas ir taisyklės pagal 4.5 pav. BPMN2 veiklos procesų diagramą

Bendriniai konceptai
<a href="#">rental provider</a> <a href="#">role</a>

[branch](#)

Concept type: [role](#)

General concept: [rental provider](#)

[renter](#)

[car booking request approval](#)

[state](#)

[started](#)

Concept type: [state](#)

[car booking request](#)

[car booking request approval](#) *has\_state* [started](#)

[state type](#)

Concept type: [categorization type](#)

[Car booking request by state type](#)

Necessity: [categorization scheme](#) *for* [general concept car booking request](#) *that subdivides* [car booking request](#) *by state type*

[received](#)

Concept type: [state](#)

[car booking request](#) *has\_state* [received](#)

[received car booking request](#)

General concept: [car booking request](#)

Necessity: *is\_included\_in* [categorization scheme](#) [Car booking request by state type](#) [accepted](#)

Concept type: [state](#)

[car booking request](#) *has\_state* [accepted](#)

[accepted car booking request](#)

General concept: [car booking request](#)

Necessity: *is\_included\_in* [categorization scheme](#) [Car booking request by state type](#) [rejected](#)

Concept type: [state](#)

[car booking request](#) *has\_state* [rejected](#)

[rejected car booking request](#)

General concept: [car booking request](#)

Necessity: *is\_included\_in* [categorization scheme](#) [Car booking request by state type](#) [renter driving licence](#)

[renter credit card](#)

### Veiksmazodiniai konceptai

[car booking request](#) *is\_started*

[car booking request](#) *is\_accepted*

[car booking request](#) *is\_rejected*

[branch](#) *check* [renter driving licence](#)

[branch](#) *check* [renter credit card](#)

[branch](#) *accept* [car booking request](#)

[branch](#) *reject* [car booking request](#)

[branch](#) *send* [car booking acceptance](#)

[renter](#) *receive* [car booking acceptance](#)

[branch](#) *send* [car booking rejection](#)

[renter](#) *receive* [car booking rejection](#)

### Veiklos taisyklės

It is obligatory that [branch](#) *check* [renter driving licence](#) and [branch](#) *check* [renter credit card](#) after [car booking request](#) *is\_started*



[It is obligatory that branch accept car booking request after branch check renter driving licence and branch check renter credit card if car booking request is valid](#)  
[It is obligatory that branch reject car booking request after branch check renter driving licence and branch check renter credit card if car booking request is invalid](#)  
[It is obligatory that branch accept car booking request if car booking request is valid](#)  
[It is obligatory that branch reject car booking request if car booking request is invalid](#)  
[It is obligatory that car booking request is\\_accepted after branch accept car booking request](#)  
[It is obligatory that car booking request is\\_rejected after branch reject car booking request](#)  
[It is obligatory that branch send car booking acceptance after car booking request is\\_accepted](#)  
[It is obligatory that renter receive car booking acceptance after branch send car booking acceptance](#)  
[It is obligatory that branch send car booking rejection after car booking request is\\_rejected](#)  
[It is obligatory that renter receive car booking rejection after branch send car booking rejection](#)

BPMN2 procesų modelio žodynas (BPMN2 konceptų metažodynas pateiktas 4.3.2 poskyryje) pagal 4.5 pav. parodytą proceso diagramą pateiktas 4.4 lentelėje.

**4.4 lentelė.** BPMN2 procesų modelio žodynas ir taisyklės pagal 4.5 pav. BPMN2 veiklos procesų diagramą

<b>Papildomų žodynų įtraukimas</b>
<a href="#">Business process vocabulary for approve car booking request</a> Included vocabulary: <a href="#">Metavocabulary for BPMN</a> Included vocabulary: <a href="#">Business Vocabulary for approve car booking request</a>
<b>Bendriniai konceptai</b>
<a href="#">renter</a> Concept type: <a href="#">pool</a> <a href="#">branch</a> Concept type: <a href="#">pool</a> <a href="#">car booking request approval is started</a> Concept type: <a href="#">none start event</a> <a href="#">check renter driving licence</a> Concept type: <a href="#">task</a> <a href="#">check renter credit card</a> Concept type: <a href="#">task</a> <a href="#">accept car booking request</a> Concept type: <a href="#">task</a> <a href="#">reject car booking request</a> Concept type: <a href="#">task</a> <a href="#">car booking request is accepted</a> Concept type: <a href="#">message end event</a> <a href="#">car booking request is rejected</a> Concept type: <a href="#">message end event</a> <a href="#">car booking request</a> Concept type: <a href="#">data object</a> <a href="#">car booking acceptance</a> Concept type: <a href="#">message</a> <a href="#">car booking rejection</a> Concept type: <a href="#">message</a>

<b>Veiksmazodiniai konceptai</b>	
subprocess	' <a href="#">approve car booking request</a> ' <i>include</i> participant ' <a href="#">renter</a> '
subprocess	' <a href="#">approve car booking request</a> ' <i>include</i> participant ' <a href="#">branch</a> '
subprocess	' <a href="#">approve car booking request</a> ' <i>include</i> activity ' <a href="#">check renter driving licence</a> '
subprocess	' <a href="#">approve car booking request</a> ' <i>include</i> activity ' <a href="#">check renter credit card</a> '
subprocess	' <a href="#">approve car booking request</a> ' <i>include</i> activity ' <a href="#">accept car booking request</a> '
subprocess	' <a href="#">approve car booking request</a> ' <i>include</i> activity ' <a href="#">reject car booking request</a> '
subprocess	' <a href="#">approve car booking request</a> ' <i>include</i> event ' <a href="#">car booking request is started</a> '
subprocess	' <a href="#">approve car booking request</a> ' <i>include</i> event ' <a href="#">car booking request is accepted</a> '
subprocess	' <a href="#">approve car booking request</a> ' <i>include</i> event ' <a href="#">car booking request is rejected</a> '
subprocess	' <a href="#">approve car booking request</a> ' <i>include</i> data object ' <a href="#">car booking request</a> '
subprocess	' <a href="#">approve car booking request</a> ' <i>include</i> message ' <a href="#">car booking acceptance</a> '
subprocess	' <a href="#">approve car booking request</a> ' <i>include</i> message ' <a href="#">car booking rejection</a> '
branch	<a href="#">provide_data_input</a> received car booking request
branch	<a href="#">get_data_output</a> accepted car booking request
branch	<a href="#">provide_data_input</a> accepted car booking request
branch	<a href="#">get_data_output</a> rejected car booking request
branch	<a href="#">provide_data_input</a> rejected car booking request
<b>Veiklos taisyklės</b>	
	It is obligatory that <a href="#">branch provide_data_input</a> received car booking request <i>after</i> <a href="#">car booking request approval</a> <i>is started</i>
	It is obligatory that <a href="#">branch get_data_output</a> accepted car booking request <i>after</i> <a href="#">branch accept car booking request</a>
	It is obligatory that <a href="#">branch provide_data_input</a> accepted car booking request <i>when</i> <a href="#">branch send car booking acceptance</a>
	It is obligatory that <a href="#">branch get_data_output</a> rejected car booking request <i>after</i> <a href="#">branch reject car booking request</a>
	It is obligatory that <a href="#">branch provide_data_input</a> rejected car booking request <i>when</i> <a href="#">branch send car booking acceptance</a>

### 4.3 BPMN2 veiklos procesų modelio transformavimas į SBVR veiklos žodyną ir veiklos taisykles

Poskyryje pateikiami: bendrieji modelių transformavimo specifikavimo principai, sudaryta BPMN2 VPM transformavimo į SBVR VŽ&VT matrica, transformacijos taisyklės, algoritmas bei BPMN2 VPM modelio elementams taikytini reikalavimai, nuo kurių priklauso transformacijos rezultatų kokybė, papildomai išskiriant BPMN2 VPM įvykių elementų vardų šablonus. Šio poskyrio informacija buvo paskelbta (Mickeviciute et al., 2014a; Mickeviciute, Butleris, 2014; Mickeviciute et al., 2015; Mickeviciute et al., 2016; Mickeviciute et al., 2017) straipsniuose.

#### 4.3.1 Bendrieji modelių transformavimo specifikavimo principai

Šiame poskyryje pateikti bendri šio skyriaus tolimesniuose poskyriuose pristatomų tyrimo rezultatų modelių transformavimo specifikavimo principai, kurie buvo taikomi siekiant formaliai apibrėžti išeigos modelio išgavimo iš įeigos modelio sąlygas ir galimybes. Įeigos modelio transformavimas į išeigos modelį parodant, kokie įeigos elementai transformuojami į kokius išeigos elementus, yra pateikiamas matricos formatu; ten stulpelių ir eilučių antraštės įvardija įeigos ir išeigos modelių elementų metamodelių pavadinimus. Įeigos modelio elementai matricose

sunumeruojami, o jų indeksai naudojami specifikacijose. Matricos celė gali įgyti tokias reikšmes.

- $A^{-\{i\}}$  reiškia, kad galima visiškai automatinė transformacija, kurios rezultato kokybė nepriklauso nuo to, ar įeigos modelis tenkina gerąją praktiką, t. y. ar modeliui pritaikyti šiame darbe aprašyti reikalavimai. Tokiu atveju daroma išvada, kad jeigu tokios transformacijos rezultatas yra semantiškai klaidingas, tada transformacijos įeiga taip pat buvo semantiškai klaidinga.
- $A^{*\neg\{i\}}$  reiškia, jog galima automatinė transformacija, tačiau tik tuo atveju, jeigu įeigos modelio elementas(-ai) tenkina šiam modeliui apibrėžtą gerąją praktiką (ypač svarbios čia rekomendacijos elementų pavadinimų formavimui). Jeigu geroji praktika netenkinama, transformacijos rezultato semantinė prasmė gali pastebimai skirtis nuo įeigos elemento(-ų) prasmės.
- $M^{-\{i\}}$  reiškia, kad vienas ar daugiau įeigos modelio elementų gali būti transformuojami tik su vartotojo įsikišimu. Jeigu yra galima A arba A\* transformacija, reiškia, jog kartu galima ir M transformacija; todėl šioms transformacijoms papildomos M transformacijos matricose nevaizduojamos.

Visais trimis išvardytais atvejais indeksas  $i$  reiškia transformacijos įeigoje esančio modelio elementų tipų numerius (įeigoje gali būti vieno ar daugiau tipų elementai).

Transformacijai realizuoti suformuluotos taisyklės kodas nurodomas greta transformacijos simbolio. Pilna taisyklių aibė buvo suformuota BPMN2 VPM į SBVR VŽ&VT ir BPMN2 VPM į BPMN2 VPM Ž&T transformacijoms, kitais atvejais (nagrinėjant atvirkštines transformacijas) taisyklių aibės nepilnos. Atvirkštinės transformacijos galimybės buvo nagrinėtos tik teoriškai, norint išsiaiškinti informacijos nuostolius, kurie galimi transformuojant iš SBVR VŽ&VT į BPMN2 VPM, bei rasti sprendimą, kaip jų išvengti.

Taisyklės bei transformacijų pavyzdžiai (jų egzemplioriai) aprašomi naudojant formalizuotą sintaksę:

$$\text{transform}(\text{SourceModel}, \{P_1, \dots, P_j\}) \xrightarrow{a|a^*|m} \text{TargetConcept};$$

čia: SourceModel atitinka transformacijos šaltinio modelį;  $\{P_1, \dots, P_j\}$  yra parametrų aibė, atitinkanti šaltinio modelio konceptus, reikalingus išeigos modelio koncepto *TargetConcept* transformacijai. Ryšio  $P_i$  egzistavimo tarp elementų būtinumas apibrėžiamas kaip predikatas  $P_i(P_n, P_m)$ ; čia  $P_n$  ir  $P_m$  yra konceptai, susieti ryšiu  $P_i$ . Riestinių skliaustų notacijos  $\{\}$  naudojimas reiškia, kad galima taikyti daugiau nei vieno tipo parametrą ar suformuoti daugiau nei vieną konceptą; virš rodyklės rašomos aukščiau aptartą transformacijos pobūdį atspindinčios raidės  $a$ ,  $a^*$  arba  $m$ .

#### 4.3.2 BPMN2 veiklos procesų modelio transformacijos į SBVR veiklos žodyną ir veiklos taisyklės matrica

Vieno modelio transformacijos į kitą modelį aprašymui bei realizavimui buvo aprašytos transformacijos taisyklės. Prieš transformacijos taisyklių aprašymą buvo

apibrėžta, kokie vieno modelio elementai ar jų poros atitinka kito modelio elementus (4.1 lentelė). BPMN2 VPM transformacijų į SBVR VŽ&VT matrica pateikta 4.5 lentelėje. Šioje lentelėje naudojami simboliai: Ti – transformacijos *i*-toji taisyklė, A – automatinė transformacija, M – rankiniu būdu atliekama transformacija (smulkesnė informacija pateikiama 4.2.1 poskyryje).

Kairėje pusėje pateikti BPMN2 konceptai, dešinėje – SBVR konceptai. Sunumeruoti BPMN2 konceptai leidžia SBVR stulpelių celėse nurodyti, kokie BPMN2 elementai dalyvauja transformacijos taisyklėje (prie raidės A arba M), norint gauti atitinkamus SBVR konceptus. Simbolis „|“ tarp numerių taisyklėje nurodo loginę operaciją „OR“, t. y. taisyklė galioja pasirenkant vieną iš išvardytų skaičių tarp šio simbolio. Pavyzdžiui, norint gauti T12 taisyklėje (A9,13|14) aprašytą veiksmazodinį konceptą, reikia šių BPMN2 konceptų: *MessageFlow*, *Lane* arba *Pool*, kurie atitinkamai pažymėti skaičiais 9, 13 ir 14.

4. 5 lentelė. BPMN2 VPM į SBVR VŽ&VT transformacijų matrica

BPMN2 VPM			SBVR VŽ&VT				
Elementų grupė	Indek-sas	Elementas	General Concept	Individual Concept	Role	Verb Concept	Business Rule
<b>Flow Objects</b>	1	<i>Event</i>					$A_{1 2,7}$ (T <sub>20</sub> )
	2	<i>Timer Event</i>					$A_{1 2,3,7,13 14}$ (T <sub>21</sub> , T <sub>22</sub> , T <sub>30</sub> , T <sub>31</sub> )
	3	<i>Activity</i>					$A_{1 2,3,8,13 14}$ (T <sub>23</sub> )
	4	<i>Message</i>					$A_{1 2,9,13 14}$ (T <sub>24</sub> )
	5	<i>Gateway</i>					$A_{1 2,10,13 14}$ (T <sub>25</sub> )
	6	<i>ComplexGateway</i>					$A_{1 2,3,5,7,13 14}$ (T <sub>26</sub> )
<b>Connecting Objects</b>	7	<i>SequenceFlow</i>	$A^*_{1}$ (T <sub>1</sub> )			$A^*_{1 2}$ (T <sub>10</sub> )	$A_{1 2,3,11,13 14}$ (T <sub>27</sub> )
	8	<i>SequenceFlow (with Condition)</i>	$A^*_{2}$ (T <sub>2</sub> )			$A^*_{3,13 14}$ (T <sub>11</sub> )	$A_{3,7,13 14}$ (T <sub>28</sub> )
	9	<i>MessageFlow</i>	$A_{10}$ (T <sub>4</sub> )			$A_{9,13 14}$ (T <sub>12</sub> )	$A_{3,8,13 14}$ (T <sub>29</sub> )
	10	<i>MessageFlow (with ref. Message)</i>	$A_{14}$ (T <sub>5</sub> )	$A^*_{2}$ (T <sub>3</sub> )	$A_{13 14,15}$ (T <sub>6</sub> )	$A_{13 14}$ (T <sub>13</sub> )	$A_{17}$ (T <sub>15</sub> )
	11	<i>Association</i>	$A_{13}$ (T <sub>7</sub> )	$M_{19}$	$A_{13 14,15}$ (T <sub>6</sub> )	$A_{13 14}$ (T <sub>14</sub> )	$A_{3,7,5,13 14}$ (T <sub>32</sub> , T <sub>33</sub> , T <sub>35</sub> , T <sub>36</sub> , T <sub>37</sub> )
	12	<i>DataAssociation</i>	$A_{16 17}$ (T <sub>8</sub> )			$A_{17}$ (T <sub>15</sub> )	$A_{12,10,11,13 14,16 17}$ (T <sub>16</sub> )
<b>Swimlanes</b>	13	<i>Lane</i>	$A_{18}$ (T <sub>9</sub> )			$A_{12,10,12,13 14,16 17}$ (T <sub>17</sub> )	$A_{3,6,7,13 14}$ (T <sub>34</sub> )
	14	<i>Pool</i>	$M_{19}$			$A_{12,16 17,18}$ (T <sub>18</sub> , T <sub>19</sub> )	$A_{1,9,13 14}$ (T <sub>39</sub> )
<b>Resources</b>	15	<i>Resource</i>				$M_{19}$	$A_{1,10,13 14}$ (T <sub>40</sub> )
<b>Artifacts</b>	16	<i>DataObject</i>					$A_{3,9,13 14}$ (T <sub>41</sub> , T <sub>43</sub> )
	17	<i>DataObject (with state)</i>					$A_{3,10,13 14}$ (T <sub>42</sub> , T <sub>44</sub> )
	18	<i>DataStore</i>					$A_{9,13 14}$ (T <sub>45</sub> )
	19	<i>TextAnnotation</i>					$A_{10,13 14}$ (T <sub>46</sub> )

### 4.3.3 BPMN2 veiklos procesų modelio transformacijos į SBVR veiklos žodyną ir veiklos taisykles taisyklės ir algoritmai

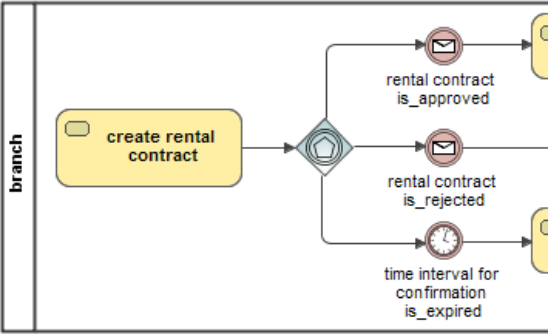
Transformacijos BPMN2 VPM į SBVR VŽ&VT taisyklių taikymo eiga yra suskirstyta į 5 žingsnius, kurie turi būti vykdomi nuosekliai, t. y. norint atlikti 2 žingsnį, turi būti atliktas 1 žingsnis. Toks suskirstymas yra paremtas veiklos taisyklių mantra (BRG, 2002); 2005 m. KTU tyrėjų parengtame Veiklos taisyklių manifesto vertime į lietuvių kalbą šis principas formuluojamas taip: „Taisyklės kuriamos faktų, o faktai – terminais išreikštų sąvokų pagrindu“ (BRG, 2003). Kitaip tariant, jei norima apibrėžti veiklos taisykles, reikia apibrėžti veiksmožodinius konceptus (faktus), o norint apibrėžti veiksmožodinius konceptus (faktus), reikia apibrėžti daiktavardinius konceptus (terminus). 1 žingsnis aprašo transformacijos taisykles iš BPMN2 elementų į bendrinius konceptus (daiktavardinius konceptus), 2 žingsnis – BPMN2 elementų ar jų porų transformacijos taisykles į veiksmožodinius konceptus ir 3–5 žingsniai – BPMN2 elementų ar jų porų transformacijos taisykles į veiklos taisykles. Transformacijos taisyklės, aprašytos 3–5 žingsniuose, išskirtos į atskirus žingsnius norint parodyti, kas yra taisyklės objektas ar iniciatorius.

Šiame skyriuje pateikiama tik dalis iš 46 transformacijos taisyklių (4.6 lentelė), kitas taisykles galima užrašyti analogiškai pagal transformacijų matricą, kuri pateikta 4.5 lentelėje. Visas taisyklių sąrašas pateikiamas kompaktiniame diske (CD), kuris pridedamas prie disertacijos (9.1 priedų poskyryje).

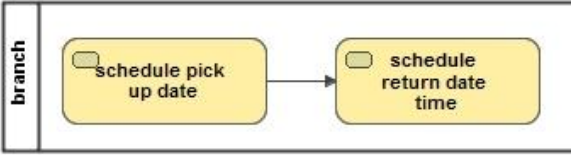
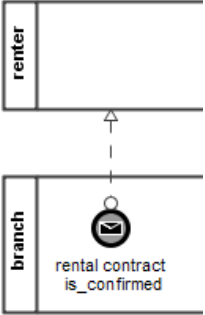
**4.6 lentelė.** BPMN2 VPM transformacijos į SBVR VŽ&VT taisyklių fragmentas

Žingsnis	Aprašymas
1.	<p>SBVR bendrinių konceptų išgavimas iš BPMN2 VPM elementų. Tikslas – suformuoti bendrinį konceptą iš atitinkamo BPMN2 VPM elemento.</p> <p><b>BPMN2 įvykis (angl. <i>Event</i>) → SBVR bendrinis konceptas (angl. <i>General Concept</i>)</b></p> <p><math>T_1: \text{transform}(\text{BPM}, \text{event: Event}) \rightarrow \text{SBVR General Concept}_1, \text{SBVR General Concept}_2</math> e.g.: <math>\text{transform}(\text{BPM}, \text{'car booking request is\_received'}) \rightarrow \text{car booking request, received}</math></p> <p><i>&lt; sąlyga &gt;</i> Laiko įvykis</p> <p><math>T_2: \text{transform}(\text{BPM}, \text{event: Event}) \xrightarrow{a^*} \text{SBVR General Concept}_1, \text{SBVR General Concept}_2</math> e.g.: <math>\text{transform}(\text{BPM}, \text{'week has\_passed'}) \xrightarrow{a^*}</math> <a href="#">time duration</a>, <a href="#">week</a> Concept type: <a href="#">time duration</a></p> <p><i>&lt; sąlyga &gt;</i> Laiko įvykio bendrinis konceptas (daiktavardinė frazė) prasideda didžiąja raide arba skaitmeniu</p> <p><math>T_3: \text{transform}(\text{BPM}, \text{event: Event}) \xrightarrow{a^*} \text{SBVR General Concept}, \text{SBVR Individual Concept}</math> e.g.: <math>\text{transform}(\text{BPM}, \text{'Week has\_passed'}) \xrightarrow{a^*}</math> <a href="#">time duration</a>, <a href="#">Week</a> General concept: <a href="#">time duration</a></p> <p><b>BPMN2 pranešimas (angl. <i>Message</i>) → SBVR bendrinis konceptas (angl. <i>General Concept</i>)</b></p>

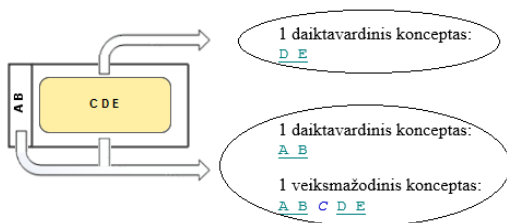
Žingsnis	Aprašymas
	<p>T<sub>4</sub>: transform(BPM, message: Message) <sup>a</sup> → SBVR General Concept  e.g.: transform(BPM, 'rental contract approval') <sup>a</sup> → <a href="#">rental contract approval</a></p>
2.	<p>SBVR veiksmožodinių konceptų išgavimas iš BPMN2 VPM elementų.  Tikslas – suformuoti veiksmožodinį konceptą iš atitinkamo BPMN2 VPM elemento.</p> <p><b>BPMN2 įvykis (angl. Event) → SBVR veiksmožodinis konceptas (angl. Verb Concept)</b></p> <p>T<sub>10</sub>: transform(BPM, event: Event) <sup>a*</sup> → SBVR Verb Concept<sub>1</sub>, SBVR Verb Concept<sub>2</sub>  e.g.: transform(BPM, 'rental contract is_rejected') <sup>a*</sup> → <a href="#">rental contract is_rejected</a>, <a href="#">rental contract has_state_rejected</a></p> <p><b>BPMN2 veikla (angl. Activity) → SBVR veiksmožodinis konceptas (angl. Verb Concept)</b></p> <p>T<sub>11</sub>: transform(BPM, pool lane: Pool Lane, activity: Activity) <sup>a*</sup> → SBVR Verb Concept  e.g.: transform(BPM, 'branch', 'create rental contract') <sup>a*</sup> → <a href="#">branch create rental contract</a></p>
3.	<p>SBVR veiklos taisyklių išgavimas iš BPMN2 VPM elementų, kai taisyklės objektas ar iniciatorius yra įvykis.</p>
3.1.	<p>SBVR veiklos taisyklių išgavimas iš BPMN2 VPM elementų, kai įvykis inicijuoja kito įvykio ar veiklos pasirodymą.  Tikslas – suformuluoti SBVR veiklos taisyklę, kuri atitinka atvejį, kai įvykis (<i>Event1</i>) inicijuoja kitą įvykį (<i>Event2</i>) arba veiklą (<i>Activity</i>), kuri įvyksta po įvykio (<i>Event2</i>) ir yra susieta sekos srautu (<i>SequenceFlow</i>).</p> <p><b>Vienas įvykis (angl. Event<sub>1</sub>) inicijuoja kitą įvykį (angl. Event<sub>2</sub>) (4.6 pav.)</b></p> <p>T<sub>20</sub>: transform(BPM, event<sub>2</sub>: Event<sub>2</sub>, sequence_flow(event<sub>2</sub>, event<sub>1</sub>): SequenceFlow, event<sub>1</sub>: Event<sub>1</sub>) <sup>a</sup> → SBVR Business Rule  e.g.: transform(BPM, 'rental contract is_rejected', sequence_flow('rental contract is_rejected', 'rental contract is_canceled'), 'rental contract is canceled') <sup>a</sup> → <a href="#">It is obligatory that car booking request is_canceled after rental contract is_rejected</a></p> <div data-bbox="577 1148 931 1334" style="text-align: center;"> </div> <p style="text-align: center;"><b>4.6 pav. Event1 inicijuoja Event2, pavyzdys</b></p>
3.2.	<p>SBVR veiklos taisyklių išgavimas iš BPMN2 VPM elementų, kai įvykis inicijuoja <i>EventBasedGateway</i> pasirodymą.  Tikslas – suformuluoti SBVR veiklos taisyklę, kuri atitinka atvejį, kai įvykis (<i>Event4</i>) su <i>EventBasedGateway</i> inicijuoja kitus įvykius, susietus su <i>EventBasedGateway</i> sekos srautais (<i>SequenceFlow</i>).</p> <p><b>Veikla (angl. Activity) inicijuoja įvykiais pagrįstą sprendimų priėmimų tašką (angl. EventBasedGateway) su išinančiais sekos srautais (angl. SequenceFlow) ir įvykiais (angl. Event) (4.7 pav.)</b></p>

Žingsnis	Aprašymas
	<p>T<sub>26</sub>: transform(BPM, pool lane: Pool Lane , activity: Activity, sequence_flow(activity, event_based_gateway): SequenceFlow, event_based_gateway: EventBasedGateway,((sequence_flow(event_based_gateway, event1):SequenceFlow, event1: Event1) + (sequence_flow(event_based_gateway, event2):SequenceFlow, event2: Event2) + (sequence_flow(event_based_gateway, event3):SequenceFlow, event3: Event3))) <sup>a</sup>→ SBVR Business Rule</p> <p>e.g. transform(BPM, 'branch','create rental contract', sequence_flow ('create rental_contract', EventBasedGateway), EventBasedGateway, (sequence_flow(EventBasedGateway, 'rental contract is approved'),'rental contract is_approved' + sequence_flow(EventBasedGateway, 'rental contract is_rejected'), 'rental contract is_rejected' + sequence_flow(EventBasedGateway, 'time interval for confirmation is_expired'),'time interval for confirmation is_expired'))<sup>a</sup>→</p> <p><u>It is obligatory that rental contract is approved or rental contract is_rejected or confirmation time interval is_expired after branch create rental contract</u></p>  <p><b>4.7 pav.</b> Pavyzdys su sprendimo priėmimo tašku (<i>EventBasedGateway</i>)</p>
4.	SBVR veiklos taisyklių išgavimas iš BPMN2 VPM elementų, kai taisyklės objektas ar iniciatorius yra veikla.
4.1.	<p>SBVR veiklos taisyklių išgavimas iš BPMN2 VPM elementų, kai veikla inicijuoja kitos veiklos ar įvykio pasirodymą.</p> <p>Tikslas – suformuoti SBVR veiklos taisyklę, kuri atitinka atvejį, kai veikla (<i>Activity1</i>) inicijuoja kitą veiklą (<i>Activity2</i>) arba įvykį (<i>Event</i>), kuris įvyksta po veiklos (<i>Activity2</i>) ir yra susietas sekos srautu (<i>SequenceFlow</i>).</p> <p><b>Viena veikla (angl. <i>Activity</i><sub>1</sub>) inicijuoja kitą veiklą (angl. <i>Activity</i><sub>2</sub>) (4.8 pav.)</b></p> <p>T<sub>28</sub>: transform(BPM, pool lane: Pool Lane, activity<sub>1</sub>: Activity<sub>1</sub>, sequence_flow(activity<sub>1</sub>, activity<sub>2</sub>): SequenceFlow, activity<sub>2</sub>: Activity<sub>2</sub>) <sup>a</sup>→ SBVR Business Rule</p> <p>e.g.: transform(BPM, 'branch', 'schedule pick up date time', sequence_flow('schedule pick up date time', 'schedule return date time'), 'schedule return date time')<sup>a</sup>→</p> <p><u>It is obligatory that branch schedule return date time after branch schedule pick up date time</u></p>



Žingsnis	Aprašymas
	 <p data-bbox="490 411 1005 438"><b>4.8 pav.</b> <i>Activity1</i> inicijuoja <i>activity2</i>, pavyzdys</p>
5.	SBVR veiklos taisyklių išgavimas iš BPMN2 VPM elementų, kur taisyklės objektas ar iniciatorius yra pranešimo srautas.
5.1.	<p data-bbox="358 535 1138 589">Tikslas – suformuluoti SBVR veiklos taisyklę, kuri atitinka atvejį, kai pranešimo srautą (<i>MessageFlow</i>) inicijuoja įvykis (<i>ThrowEvent</i>).</p> <p data-bbox="358 602 1138 657"><b>Įvykis (angl. <i>ThrowEvent</i>) inicijuoja pranešimo srautą (angl. <i>MessageFlow</i>) (4.9 pav.).</b></p> <p data-bbox="358 669 1138 778"> <math>T_{39}</math>: transform(BPM, pool lane<sub>1</sub>: Pool Lane<sub>1</sub>, event: Event, message_flow(event, pool lane<sub>2</sub>): MessageFlow, pool lane<sub>2</sub>: Pool Lane<sub>2</sub>) <math>\xrightarrow{a}</math> SBVR Business Rule  e.g.: transform(BPM, 'branch', 'rental contract is_confirmed', message_flow('branch', 'renter'), 'renter') <math>\xrightarrow{a}</math> </p> <p data-bbox="358 784 1138 802">It is obligatory that <u>renter receive_message_from_branch</u> after <u>rental contract is_confirmed</u></p>  <p data-bbox="443 1161 1053 1188"><b>4.9 pav.</b> <i>Throw Event</i> inicijuoja <i>MessageFlow</i>, pavyzdys</p>

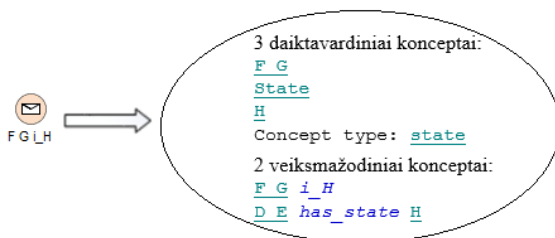
BPMN2 VPM elementų vardų apdorojimui lingvistiniai metodai nėra taikomi. SBVR VŽ&VT yra sudaromos nagrinėjant BPMN2 elemento vardo žodžius pagal jų vietą. Remiantis gerąja modeliavimo praktika sudarant BPMN2 elementų vardus buvo suformuluoti reikalavimai, kurių turi būti laikomasi norint gauti semantiškai teisingą SBVR VŽ&VT iš BPMN2 VPM (žiūrėti 4.3.4 poskyrį). Išskirti BPMN2 VPM elementai, kurie transformuojami neanalizuojant ir kurie transformuojami analizuojant BPMN2 elemento vardo sudėtį. Neanalizuojami: konteineriai (*Lane* ir *Pool*), duomenų objektai (*DataObject*), duomenų saugyklos (*DataStore*), pranešimai (*Message*). Analizuojami: veiklos (*Activity*), įvykiai (*Event*). Veiklos pavadinimo dalis be pirmo žodžio naudojama transformacijai į daiktavardinį konceptą, pilnas veiklos vardas kartu su konteineriu (*Lane*, *Pool*), kuriame yra veikla, yra vartojamas veiksmažodiniam konceptui formuoti. Pavyzdiniai SBVR žodyno konceptų formavimo iš BPMN2 elementų vardų principai yra pateikti 4.10 ir 4.11 paveiksluose.



kur A, B – žodžiai, sudarantys konteinerio vardą  
C, D, E - žodžiai, sudarantys veiklos vardą

#### 4.10 pav. SBVR VŽ formavimo pavyzdys iš BPMN2 veiklos vardo

Įvykio pavadinimo dalis be paskutinio žodžio naudojama transformacijai į daiktavardinį konceptą, visas įvykio vardas transformuojamas į unarinį veiksmazodinį konceptą, paskutinis žodis – į daiktavardinį konceptą, kuris nusako sukurto daiktavardinio koncepto būseną. Daiktavardinis konceptas „state“ kuriamas tik vieną kartą – pirmą kartą kuriant daiktavardinį konceptą iš įvykio (pavyzdyje H) arba duomenų objekto, vėliau šis konceptas naudojamas visiems kitiems būsenai atstovaujantiems daiktavardiniais konceptams, kurie gaunami iš įvykių ir duomenų objektų.



kur F, G, H – žodžiai, sudarantys įvykio vardą  
i – pagalbinis žodis is/are

#### 4.11 pav. SBVR VŽ formavimo pavyzdys iš BPMN2 įvykio vardo

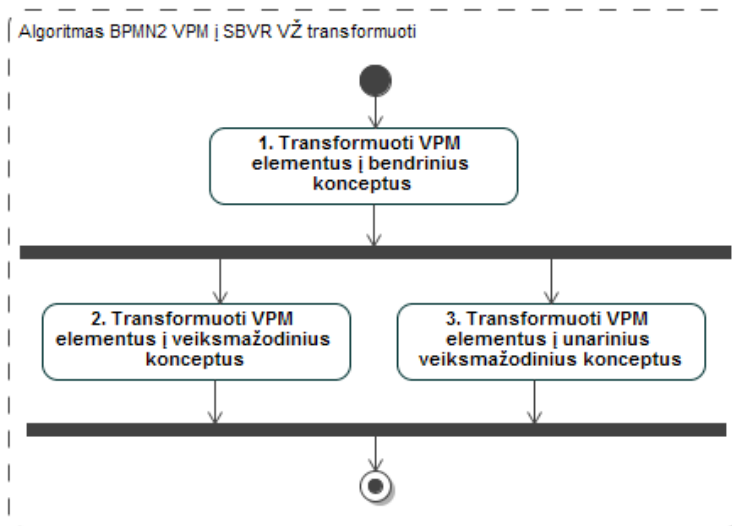
SBVR veiksmazodiniai konceptai ir veiklos taisyklės sudaromos papildomai naudojant papildomus veiksmazodžius, kurie nusako konceptų ryšį, pavyzdžiui, konteinerių (*Lane* ir *Pool*) priklausomybę.

Veiksmazodžiai, kurie yra naudojami aprašant veiklos žodyną transformacijų metu, pateikti 4.7 lentelėje. Šie veiksmazodžiai yra naudojami tam tikriems BPMN2 elementams.

4.7 lentelė. BPMN2 VPM į SBVR VŽ naudojami veiksmazodžiai

Veiksmazodis	Naudojamas BPMN2 elementui
<i>receive</i>	įeinantis <i>MessageFlow</i> (with ref. <i>Message</i> )
<i>send</i>	išeinantis <i>MessageFlow</i> (with ref. <i>Message</i> )
<i>include</i>	<i>Lane in Pool</i>
<i>receive_message_from</i>	įeinantis <i>MessageFlow</i>
<i>has_state</i>	Įvykiai ( <i>Event</i> ) ir duomenų objektai su būsenomis ( <i>DataObject</i> (with <i>state</i> ))

Transformacijos BPMN2 VPM į SBVR VŽ&VT algoritmas buvo išskirtas į dvi dalis. Atskiras algoritmų vaizdavimas pasirinktas norint papildomai pateikti veiklų išsamesnius aprašymus dalimis, be to, toks algoritmų atskyrimas neprieštaruoja SBVR VŽ&VT sudarymo principams – veiklos taisyklės turi būti kuriamos naudojant veiklos žodyno elementus. Be to, visi 4 skyriaus algoritmai, kurie aprašo veiklos žodyno bei veikos taisyklių išgavimą, yra paremti anksčiau šiame skyriuje aprašyta veiklos taisyklių mantra (ja remiasi ir SBVR), kuri griežtai nusako žodyno konceptų ir taisyklių formavimo eiliškumą. Pirmą algoritmo dalį nusako SBVR VŽ išgavimą iš BPMN2 VPM (4.12 pav.).

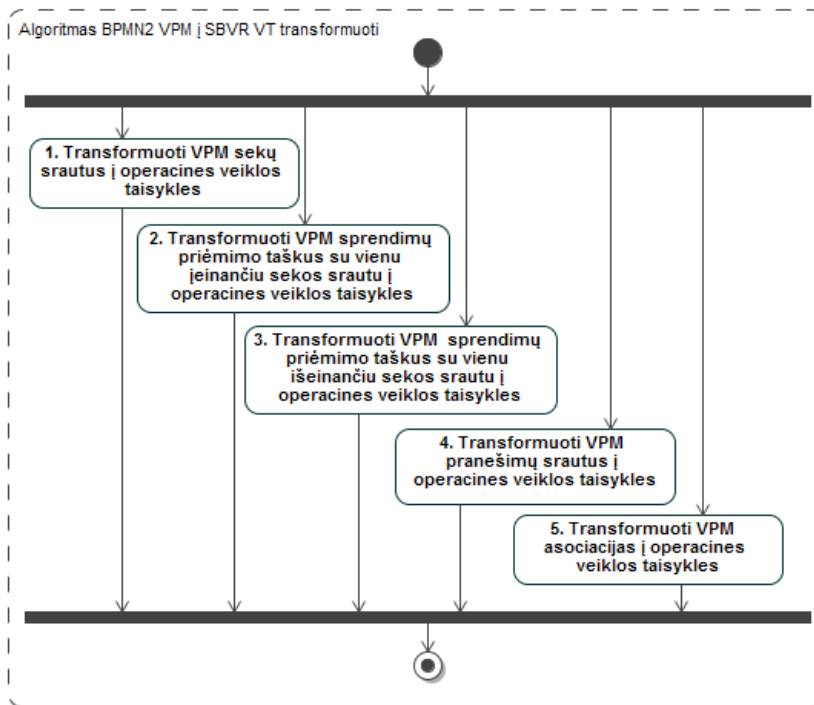


**4.12 pav.** Algoritmas BPMN2 VPM į SBVR VŽ transformuoti

Skaičiai veiklų pradžioje nurodo veiklos detalių paaiškinimą pagal toliau pateiktą numeraciją (veiklos skaičius sutampa su toliau einančiu numeriu).

1. Transformuoti BPMN2 VPM konteinerius (*Lane*, *Pool*), įvykių ir veiklų (*Event*, *Activity*) daiktavardines frazes, duomenų objektus (*DataObject*), duomenų saugyklas (*DataStore*), pranešimus (*Message*), duomenų objektus su būseną (*DataObject with State*) į SBVR bendrinius konceptus.
2. Transformuoti BPMN2 VPM sudėtinius konteinerius (*Pool with Lane*), veiklas (*Activity*) su konteineriais (*Lane*, *Pool*), pranešimų srautus (*MessageFlow*, *MessageFlow with ref. Message*), asociacijas (*Association*) ir duomenų asociacijas (*DataAssociation*), kurios susijusios su įvykiais, kurie savo ruožtu susiję su pranešimų srautais, į SBVR veiksmožodinius konceptus.
3. Transformuoti BPMN2 VPM įvykius (*Event*) į SBVR unarinius veiksmožodinius konceptus.

Antra algoritmo dalis nusako SBVR VT išgavimą iš BPMN2 VPM (4.13 pav.).



4.13 pav. Algoritmas BPMN2 VPM į SBVR VT transformuoti

Skaičiai veiklų pradžioje nurodo veiklos išsamų paaiškinimą pagal toliau pateiktą numeravimą (veiklos skaičius sutampa su toliau einančiu numeriu).

1. Transformuoti BPMN2 VPM sekų srautus (*SequenceFlow*) į SBVR operacines veiklos taisykles, įtraukiant srautų sąlygas, jei jos yra.
2. Transformuoti BPMN2 VPM sprendimų priėmimo taškus (*Gateway*) su vienu įeinančiu sekos srautu į SBVR operacines veiklos taisykles, įtraukiant sąlygas, jei jos yra.
3. Transformuoti BPMN2 VPM sprendimų priėmimo taškus (*Gateway*) su vienu išeinančiu sekos srautu į SBVR operacines veiklos taisykles.
4. Transformuoti BPMN2 VPM pranešimų srautus (*MessageFlow*, *MessageFlow with ref. Message*) į SBVR operacines veiklos taisykles.
5. Transformuoti BPMN2 VPM asociacijas (*Association*), išeinančias iš ribinio kompensavimo įvykio (*Boundary Compensation Event*), į SBVR operacines veiklos taisykles.

#### 4.3.4 Reikalavimai BPMN2 VPM, nuo kurių priklauso transformacijų rezultatų kokybė

BPMN2 VPM transformavimui į SBVR VŽ&VT įtakos turi du kriterijai: 1) VPM elementų vardai (koku formatu yra pavadintos veiklos, įvykiai ir kt. elementai) – dar kitaip vadinama geroji modeliavimo praktika ir 2) tam tikrų elementų naudojimas modeliuojant VPM.

- 1) Norint transformavimo metu gauti semantiškai korektišką SBVR VŽ&VT, neatliekant BPMN2 procesų modelio lingvistinės analizės, reikia vadovautis vadinamąja „gerąja modeliavimo praktika“, kuria remiantis BPMN2 elementų pavadinimai turėtų būti sudaromi laikantis tam tikrų taisyklių. Rekomenduojamų BPMN2 elementų pavadinimų struktūrų taikymo taisyklės pateikiamos 4.8 lentelėje. Atkreiptinas dėmesys į tai, kad tam tikros BPMN2 elementų vardų dalys turėtų būti vaizduojamos tarp žodžių rašant pabraukimo simboliu – tokiu būdu sukuriamos prielaidos automatiškai atskirti daiktavardinę elemento pavadinimo dalį nuo veiksmažodinės. 4.8 lentelėje taikomi šie pavadinimo struktūros EBNF notacija išreikšti elementų šablonai:
- *noun phrase* := *[[adjective]]noun[{{preposition}}][[adjective]][[preposition]][[noun]]* (Pvz., *quick order of customer, department of information systems, tea from green fields*)
  - *verb in past tense* – būtojo laiko veiksmažodžio forma, kurioje vietoj tarpų (jeigu jų yra) tarp žodžių naudojami pabraukimo simboliai ‘\_’ (Pvz., *confirmed, carried\_out*)
  - *verb in present tense* – esamojo laiko veiksmažodžio forma, kurioje vietoj tarpų (jeigu jų yra) tarp žodžių naudojami ‘\_’ simboliai (Pvz., *confirm, carry\_out*)
  - *verb phrase in past tense* := *[adverb], verb in past tense, [noun phrase], [adverb|preposition], vietoj tarpų tarp žodžių naudojant ‘\_’ simbolius* (Pvz., *accepted, took\_over, slowly\_took\_over, broke\_project\_down*)
  - *verb phrase in present tense* := *[adverb], verb in present tense, [noun phrase], [adverb|preposition], vietoj tarpų (jeigu jų yra) tarp žodžių naudojant ‘\_’ simbolius* (Pvz., *accept, take\_in, completely\_take\_in*)
  - *verb phrase for event* := “*is\_*“, *verb phrase in past tense* (Pvz., *is\_accepted*)
  - *verb phrase for activity* := *verb phrase in present tense, noun phrase, [preposition], [noun phrase]* (Pvz., *register customer request, quickly\_take\_over order of customer*)
- 2) Modeliuojant VPM reikia naudoti konteinerius (*Pool* arba *Lane*), kurie identifikuoja, kas yra atsakingas už konkrečios veiklos vykdymą (asmuo, organizacinis vienetas ar pan.). Transformavimo taisyklių sudarymui šie elementai yra būtini. Išimtis galima vaizduojant aukščiausio hierarchijos lygio veiklos procesus, kai aprašoma pagrindinių subprocesų vykdymo tvarka.

4.8 lentelė. Apibrėžta gerąją praktiką atitinkančios BMN2 elementų pavadinimų struktūros

Elementų grupė	Elementas	BPMN2 elemento pavadinimo struktūra	Pavyzdys
Flow Objects	Event	noun phrase, verb phrase for event	„rental contract is confirmed“
	Activity	verb phrase for activity	„confirm estimated rental charge“
	Message	noun phrase	„car booking request“
Swimlanes	Lane	noun phrase	„branch“
	Pool	noun phrase	„operating company“
Artifacts	DataObject	noun phrase	„rental contract“
	DataObject (with State)	noun phrase, verb phrase in past tense	„rental contract“ ( „ rejected“)
	DataStore	noun phrase	„renter“
	Group	noun phrase	„renter orders“
Property	Condition	Rekomenduojama taikyti SBVR sąlygų rašymo taisyklės. Nesilaikant šių taisyklių galimi įvairūs variantai, viena formavimo taisyklė negali būti apibrėžta.	„car booking request is accepted“ „price >5“

#### 4.3.5 Papildomi BPMN2 įvykių vardų sudarymo šablonai

BPMN2 turi gausią įvykių elementų aibę: joje yra 13 elementų tipų, kurie gali būti priskirti atskiroms įvykių grupėms; dėl šios priežasties jų skaičius yra dar didesnis. Tiesa, ne visos įvykių grupės gali turėti visus įvykių tipus. Įvykių grupės yra išskiriamos į pradžios, vidurio ir pabaigos grupes. Vidurio įvykių grupė dar išskiriama į priimančius (angl. *Catch*) ir inicijuojančius (angl. *Throw*). Kiekvienas įvykis turi savo prasmę VPM, todėl į tai reikia atsižvelgti transformuojant BPMN2 VPM į SBVR VŽ&VT. BPMN2 įvykiai buvo suklasifikuoti į dvi grupes pagal tai, kokius vardų formatus gali turėti: 1) vardas pagal šabloną; 2) vardas, kuris imamas kaip paprastas tekstas. Nuo to priklauso, kaip įvykiai (jų vardai) yra apdorojami BPMN2 modelio transformavimo metu. Visi BPMN2 įvykiai buvo priskirti vienai iš aprašytų grupių. Pateiktoje 4.9 lentelėje yra nurodoma, kuriai grupei priklauso atitinkamas įvykis: šviesiai pilka spalva žymi 1) grupę, tamsiai pilka – 2) grupę. Varnelės simbolis (✓) lentelėje parodo BPMN2 įvykių grupių galimus įvykių elementų tipus. Įvykių bei jų grupių vardų vertimai į lietuvių kalbą pateikti 9.5 priedų poskyryje, 9.4–9.6 lentelėse.

#### 4.9 lentelė. Įvykių priskyrimo atitinkamai grupei pagal vardus lentelė

Įvykis (angl. Event)	Įvykių grupės				
	Start	Intermediate Catch	Intermediate Throw	Boundary	End
None	✓	✓			✓
Message	✓	✓	✓	✓	✓
Timer	✓	✓		✓	
Error	✓			✓	✓
Escalation	✓		✓	✓	✓
Cancel				✓	✓
Compensation	✓		✓	✓	✓
Conditional	✓	✓		✓	
Link		✓	✓		
Signal	✓	✓	✓	✓	✓
Terminate					✓
Multiple	✓	✓	✓	✓	✓
Parallel Multiple	✓	✓		✓	

BPMN2 įvykio vardo sudarymo šablonas, pateiktas 4.8 lentelėje, buvo papildytas sukuriant tris šablonus (4.10 lentelė) BPMN2 įvykio vardui užrašyti. Šablonai aprašyti taikant EBNF notaciją: „first\_name“, „second\_name“ ir „third\_name“. Išskirti BPMN2 įvykių šablonai leidžia formuoti semantiškai turtingesnę SBVR veiklos žodyną, kuriame BPMN2 įvykiai aprašomi SBVR VŽ&VT pagal BPMN2 elemento prasmę. Darbe buvo išskirti atskiri šablonai BPMN2 laiko įvykiams ir sudėtiniais bei lygiagretiems sudėtiniais BPMN2 įvykiams.

#### 4.10 lentelė. BPMN2 įvykių vardų šablonai

Pavadinimas	BPMN2 įvykiai, kuriems taikoma	Šablonas
1 šablonas	Visiems BPMN2 įvykiams iš 1 grupės išskyrus <i>Timer</i> , <i>Multiple</i> ir <i>Parallel Multiple</i>	<b>first_name</b> = <i>noun phrase</i> , <i>verb phrase for event</i> , eg.: order is_received, when: <b>noun phrase</b> = <i>[[adjective]] noun [[preposition]] [[adjective]] [[preposition]] [[noun]]</i> (e.g.: quick order of customer, department of information systems, tea from green fields) <b>verb phrase for event</b> = “is” “are” “has” “have”, <i>verb phrase in past tense</i> (e.g.: is accepted)
2 šablonas	<i>Time Duration</i>	<b>second_name</b> = <i>noun phrase</i> , <i>verb phrase for timer time duration event</i> , when <b>verb phrase for timer time duration event</b> = “is” “are” “has” “have”, „_passed“ (e.g.: week has_passed)
	<i>Time Cycle</i>	<b>Time Cycle</b> : <b>second_name</b> = <i>noun phrase</i> , <i>verb phrase for timer time cycle event</i> , when <b>verb phrase for timer time cycle event</b> = “is” “are” “has” “have”, „_reoccured“ (e.g.: monday has_reoccured)

Pavadinimas	BPMN2 įvykiai, kuriems taikoma	Šablonas
	<i>Time Date</i>	<b>Time Date:</b> <i>second_name</i> = noun phrase, verb phrase for timer time date event, when <b>verb phrase for timer time date event</b> = “is“ “are“ “has“ “have“, „_occured“ (e.g.: 7 a.m. has_occured)
3 šablonas	<i>Multiple ir Parallel Multiple</i>	<b>third_name</b> = <i>first_name</i> , “and“ “or“, <i>first_name</i> , {“and“ “or“, <i>first_name</i> }

BPMN2 laiko (*Timer*) įvykiai užrašomi 3 skirtingais būdais pagal tai, kokiai situacijai yra taikomas šio tipo įvykis: nusakyti laiko trukmei, cikliškumui ar konkrečiai datai. Skirtumas nusakomas naudojamu veiksmažodžiu.

BPMN2 sudėtiniai (*Multiple*) ir lygiagrečiai sudėtiniai (*Parallel Multiple*) įvykiai yra naudojami iš karto keliems įvykiams aprašyti, todėl jų vardai aprašo vieno ar kelių objektų ir jų būsenų pasikeitimus. Šiuo atveju yra taikomas kitoks vardų apdorojimo algoritmas: skirtingus objekto ar objektų būsenas atskiriant „OR“ (*Multiple*) ir „AND“ skyrikliais (*Parallel Multiple*).

Išskirtos antrosios grupės (4.9 lentelė) elementų atveju jokie reikalavimai vardui nėra taikomi, vardai užrašomi laisva forma. BPMN2 ryšių (*Link*) įvykiai yra skirti BPMN2 diagramos vaizdavimui optimizuoti (pvz., kai proceso diagrama yra labai didelė (ilga), ji išskaidoma į atskiras dalis naudojant šio tipo įvykius), ir neturi veiklos procesui jokios įtakos, į SBVR veiklos žodyną ir taisykles nėra įtraukiami. Jų vardai naudojami norint sudaryti vientisą veiklos procesą (SBVR veiklos žodyno ir taisyklių atžvilgiu), kuris buvo išskaidytas panaudojant ryšio (*Link*) tipo įvykius. Kiekvieno BPMN2 įvykio pavyzdiniai vardai pateikti 4.11 lentelėje.

#### 4.11 lentelė. Įvykių vardų pavyzdžiai

Įvykis (angl. <i>Event</i> )	Įvykių grupės				
	<i>Start</i>	<i>Intermediate Catch</i>	<i>Intermediate Throw</i>	<i>Boundary</i>	<i>End</i>
<i>None</i>	rent car is_started	rent car is_started			rent car is_finished
<i>Message</i>	rental contract is_received	rental contract is_received	rental contract is_sent	error message is_sent	rental contract is_sent
<i>Timer</i> (laiko momentas, intervalas, ciklas)	10 a.m. has_occured	10 business days have_passed		week has_reoccured	
<i>Error</i>	order is_missing			order item is_missing	customer email is_invalid
<i>Escalation</i>	offer is_proposed		offer is_checked	offer is_rejected	offer is_confirmed
<i>Cancel</i>				order is_cancelled	offer is_cancelled
<i>Compensation</i>	booking	booking	booking	booking	booking



Įvykis (angl. Event)	Įvykių grupės				
	Start	Intermediate Catch	Intermediate Throw	Boundary	End
	is_compensated	is_compensated	is_compensated	is_compensated	is_compensated
Conditional	order is_packed	user information is_correct		order item is_broken	
Link		part1	part1		
Signal	account is_created	order is_rejected	order item is_shipped	offer is_rejected	order is_confirmed
Terminate					order is_cancelled
Multiple	order is_sent or order is_taken	order is_confirmed or order is_repeated	order is_confirmed or order is_repeated	order item is_broken or order item is_wrong	order is_canceled or order is_shipped
Parallel Multiple	credit card is_valid and driving licence is_valid	user information is_checked and offer is_prepared		order discount is_applied and order is_paid	

#### 4.3.6 Apibendrinta geroji BPMN2 VPM modeliavimo praktika: korektiško BPMN2 VPM kriterijai

Atsižvelgiant į siūlomas BPMN2 veiklos procesų modeliavimo gerąsias praktikas galima apibendrinti visus taikomus kriterijus, kuriuos turi tenkinti korektiškas BPMN2 VPM. Šie kriterijai buvo sugrupuoti į atitinkamas grupes.

##### 1. Struktūros

- 1.1. Diagramų vardai unikalūs.
- 1.2. Sprendimo priėmimo taškų naudojimas (ne daugiau kaip 1 įėjimas/išėjimas iš vienos veiklos ar įvykio).
- 1.3. Visi proceso srautai turi turėti pradžią bei pabaigą. Negali būti situacijų su akligatviais (angl. *deadend*) ar nesibaigiančių ciklų.
- 1.4. Proceso diagramose elementų kiekis turi būti proporcingai paskirstytas (išimtis gali būti daroma 0 hierarchijos lygyje).
- 1.5. Proceso diagramose turi būti naudojami konteineriai (angl. *lanes/pools*).
- 1.6. Geriausia, kai proceso hierarchijos gylis yra tarp 3 ir 5.
- 1.7. Nenaudoti daugiau negu 2 sprendimo priėmimo taškų, einančių vienas po kito.

##### 2. Išdėstymo

- 2.1. Diagrama neturi užimti didesnio ploto negu A4 formato lapas.
- 2.2. Elementai (išskyrus ryšius) negali persidengti.
- 2.3. Ryšiai negali prasidėti ar baigtis toje pačioje vietoje.

##### 3. Elementų vardų

- 3.1. Visi elementai (išskyrus ryšius) turi turėti pavadinimus.
- 3.2. Subproceso pavadinimas turi sutapti su jį aprašančiu diagramos vardu.
- 3.3. Iš sprendimų priėmimo taškų išeinantys srautai su alternatyviais veiksmiais turi turėti sąlygų vardus.
- 3.4. Elementų vardų struktūra turi būti sudaroma pagal reikalavimus 4.8

lentelėje (pateikta 4.2.4 poskyryje).

3.5. Globalios užduotys, kurios yra išskviečiamos iš kitų proceso diagramų, turi turėti pavadinimą.

Atsižvelgiant į šiame darbe suformuluotus apribojimus BPMN2 veiklos procesų modeliams tam, kad būtų gautas semantiškai korektiškas SBVR VŽ&VT iš BPMN2 VPM, būtina tenkinti 1.5, 3.4 ir 3.5 punktuose aprašytus kriterijus.

#### 4.4 BPMN2 veiklos procesų modelio transformavimas į BPMN2 procesų modelio žodyną ir taisykles

Atlikus BPMN2 VPM transformaciją į SBVR VŽ&VT būtų prarandama informacija apie BPMN2 VPM elementų tipus, procesų hierarchiją. Tam, kad ši informacija būtų išsaugota, buvo nuspręsta papildomai generuoti BPMN2 VPM Ž&T; jis kuriamas naudojant BPMN2 konceptų metažodyną, kuriame yra aprašyti BPMN2 konceptai. Šiame poskyryje pateikiama: BPMN2 VPM transformavimo į BPMN2 VPM Ž&T matrica, BPMN2 konceptų metažodynas, transformacijos taisyklės bei algoritmas.

##### 4.4.1 BPMN2 veiklos procesų modelio transformacijos į BPMN2 veiklos procesų modelio žodyną ir taisykles matrica

BPMN2 VPM elementų transformacijos galimybės į BPMN2 VPM Ž&T taikant BPMN2 konceptų metažodyną pateikiamos transformacijos matricoje, kuri aprašyta 4.12 lentelėje.

4.12 lentelė. BPMN2 VPM transformacijų į BPMN2 VPM Ž&T matrica

BPMN2 VPM		BPMN2 VPM Ž&T	
Elementų grupė	Elementas	Atitinka SBVR	Pavyzdys
Flow Objects	Process	General Concept	<a href="#">rent car</a> Concept type: <a href="#">process</a>
	Event	General Concept, Verb Concept	<a href="#">car booking request is accepted</a> Concept type: <a href="#">message end event</a> Jei įvykis yra proceso dalis: <a href="#">process</a> 'approve car booking request' <a href="#">include event</a> 'car booking request is accepted' Jei įvykis yra subprocesos dalis: <a href="#">subprocess</a> 'approve car booking request' <a href="#">include event</a> 'car booking request is accepted'
	Subprocess	General Concept, Verb Concept	<a href="#">approve car booking request</a> Concept type: <a href="#">subprocess</a> Jei subprocesas yra proceso dalis: <a href="#">process</a> 'book car' <a href="#">include subprocess</a> 'approve car booking request' Jei subprocesas yra kito subprocesos (aukštesnio hierarchijos lygio) dalis: <a href="#">subprocess</a> 'book car' <a href="#">include subprocess</a> 'approve car booking request'
	Task	General Concept, Verb	<a href="#">check renter credit card</a>

BPMN2 VPM		BPMN2 VPM Ž&T	
<i>Elementų grupė</i>	<i>Elementas</i>	<i>Atitinka SBVR</i>	<i>Pavyzdys</i>
		<i>Concept</i>	<p>Concept type: <a href="#">task</a></p> <p>Jei veikla yra proceso dalis:  <a href="#">process</a> `approve car booking request` <a href="#">include activity</a> `check renter credit card`</p> <p>Jei veikla yra subproceso dalis:  <a href="#">subprocess</a> `approve car booking request` <a href="#">include activity</a> `check renter credit card`</p>
	<i>Message</i>	<i>General Concept, Verb Concept</i>	<p><a href="#">car booking request rejection</a></p> <p>Concept type: <a href="#">message</a></p> <p>Jei pranešimas yra proceso dalis:  <a href="#">process</a> `approve car booking request` <a href="#">include message</a> `car booking request rejection`</p> <p>Jei pranešimas yra subproceso dalis:  <a href="#">subprocess</a> `approve car booking request` <a href="#">include message</a> `car booking request rejection`</p>
<i>Connecting Objects</i>	<i>Association</i>	<i>Verb Concept, Business Rule</i>	<p>Su duomenų objektu:  <a href="#">branch get_data_output approved rental contract</a></p> <p>It is obligatory that <a href="#">branch get_data_output approved rental contract</a> after <a href="#">branch approve rental contract</a></p> <p>Su duomenų saugykla analogiškai, tik naudojant „get_data_store_output”.</p>
	<i>Data Association</i>	<i>Verb Concept, Business Rule</i>	<p>Su duomenų objektu:  <a href="#">branch provide_data_input received car booking request</a></p> <p>It is obligatory that <a href="#">branch provide_data_input received car booking request</a> when <a href="#">branch approve car booking request</a></p> <p>Su duomenų saugykla analogiškai, tik naudojant „get_data_store_input”.</p>
<i>Swimlanes</i>	<i>Lane</i>	<i>General Concept, Verb Concept</i>	<p><a href="#">branch</a></p> <p>Concept type: <a href="#">lane</a></p> <p>Jei konteineris (<i>Lane</i>) yra proceso dalis:  <a href="#">process</a> `book car` <a href="#">include participant</a> `branch`</p> <p>Jei konteineris (<i>Lane</i>) yra subproceso dalis:  <a href="#">subprocess</a> `book car` <a href="#">include participant</a> `branch`</p>
	<i>Pool</i>	<i>General Concept, Verb Concept</i>	<p><a href="#">branch unit</a></p> <p>Concept type: <a href="#">pool</a></p> <p>Jei konteineris (<i>Pool</i>) yra proceso</p>

BPMN2 VPM		BPMN2 VPM Ž&T	
Elementų grupė	Elementas	Atitinka SBVR	Pavyzdys
			dalis: <a href="#">process</a> `book car` <i>include</i> <a href="#">participant</a> `branch unit` Jei konteineris ( <i>Pool</i> ) yra subproceso dalis: <a href="#">subprocess</a> `book car` <i>include</i> <a href="#">participant</a> `branch unit`
Artifacts	DataObject	General Concept, Verb Concept	<a href="#">car booking request</a> Concept type: <a href="#">data object</a> Jei duomenų objektas yra proceso dalis: <a href="#">process</a> `book car` <i>include</i> <a href="#">data object</a> `car booking request` Jei duomenų objektas yra subproceso dalis: <a href="#">subprocess</a> `book car` <i>include</i> <a href="#">data object</a> `car booking request`
	DataStore	General Concept, Verb Concept	<a href="#">customer data store</a> Concept type: <a href="#">data store</a> Jei duomenų saugykla yra proceso dalis: <a href="#">process</a> `book car` <i>include</i> <a href="#">data store</a> `customer data store` Jei duomenų saugykla yra subproceso dalis: <a href="#">subprocess</a> `book car` <i>include</i> <a href="#">data store</a> `customer data store`

#### 4.4.2 BPMN2 veiklos procesų modelio transformacijos į BPMN2 veiklos procesų modelio žodyną ir taisykles taisyklės ir algoritmai

Transformacijos BPMN2 VPM į BPMN2 VPM Ž&T metu tam tikri VPM elementai ar jų kombinacijos yra transformuojamos į veiklos procesų modelio žodyną bei taisykles. Visi reikalingi BPMN2 konceptai, kurie apibrėžia visus galimus elementus, yra įtraukti į metažodyną. BPMN2 konceptų metažodynas pateiktas 9.2 priedų poskyryje. Transformacijos taisyklių sudarymo principai, panaudojant BPMN2 konceptų metažodyną, pateikti 4.13 lentelėje.

**4.13 lentelė.** BPMN2 VPM transformacijų į BPMN2 VPM Ž&T taisyklių sudarymo principai

Žodyno sudedamoji dalis	Aprašymas
General Concept	Bendrinių konceptų išgavimas iš BPMN2 elementų: konteinerių, veiklų, įvykių, pranešimų, duomenų objektų, duomenų saugyklų. Tikslas – suformuoti bendrinį konceptą, kuris reikalingas informacijos apie BPMN2 elemento tipą išsaugojimui, iš atitinkamo BPMN2 VPM elemento. <b>BPMN2 elementas → bendrinis konceptas (angl. General Concept)</b>

Žodyno sudedamoji dalis	Aprašymas
	<p><i>transform(BPM, bpmn_element: BpmnElement) → SBVR General Concept</i>  e.g.: <i>transform(BPM, 'car booking request is received') →</i>  <a href="#">car booking request is received</a></p> <p>Concept type: <a href="#">message intermediate catch event</a></p>
Verb Concept	<p>Veiksmažodinių konceptų išgavimas iš BPMN2 elementų.</p> <p>Tikslas – suformuoti veiksmažodinį konceptą, kuris reikalingas informacijos apie BPMN2 VPM hierarchiją išsaugojimui, iš atitinkamo BPMN2 VPM elemento.</p> <p><b>BPMN2 elementas → SBVR veiksmažodinis konceptas (angl. Verb Concept)</b></p> <p><i>transform(BPM, bpmn_element_type: BpmnElementType, bpmn_element: BpmnElement, 'process', bpmn_process[BpmnProcess]) → SBVR Verb Concept</i>  e.g.: <i>transform(BPM, 'subprocess', 'approve car booking request', process 'book car') →</i>  <a href="#">process 'book car' include subprocess 'approve car booking request'</a></p> <p>Kiekvienas elementas vartojant veiksmažodį „include“ yra priskiriamas procesui, kuriam jis priklauso. Iš viso galimos 16 šios konstrukcijos variacijų: 7 su „process“ (nusako aukščiausio hierarchijos lygio procesą) ir analogiškos 7 su „subprocess“.</p> <p><a href="#">process 'name' include subprocess 'name'</a>  <a href="#">process 'name' include activity 'name'</a>  <a href="#">process 'name' include event 'name'</a>  <a href="#">process 'name' include participant 'name'</a>  <a href="#">process 'name' include message 'name'</a>  <a href="#">process 'name' include data object 'name'</a>  <a href="#">process 'name' include data store 'name'</a>  <a href="#">subprocess 'name' include subprocess 'name'</a>  <a href="#">subprocess 'name' include activity 'name'</a>  <a href="#">subprocess 'name' include event 'name'</a>  <a href="#">subprocess 'name' include participant 'name'</a>  <a href="#">subprocess 'name' include message 'name'</a>  <a href="#">subprocess 'name' include data object 'name'</a>  <a href="#">subprocess 'name' include data store 'name'</a></p> <p>BPMN2 elemento tipo naudojimas veiksmažodiniuose konceptuose (taip pat ir taisyklėse) leidžia suformuoti korektišką žodyno elementą: BPMN2 veiklų, įvykių pavadinimai formuojamame BPMN2 VPM žodyne yra aprašomi kaip daiktavardiniai konceptai. Veiksmažodiniuose konceptuose, taip pat ir veiklos taisyklėse tokių suformuotų konceptų vartojimas yra nenatūralus, todėl tampa sunkiai suprantamas. Šiai problemai išspręsti yra naudojamas BPMN2 elemento tipas.</p> <p>Veiksmažodinių konceptų išgavimas iš BPMN2 duomenų objektų, duomenų saugyklų.</p> <p>Tikslas – suformuoti veiksmažodinį konceptą, kuris reikalingas BPMN2 duomenų objektų ir duomenų saugyklų informacijai išsaugoti.</p>

Žodyno sudedamoji dalis	Aprašymas
	<p><b>BPMN2 duomenų objektas (angl. <i>DataObject</i>) (išeinantis) → veiksmožodinis konceptas (angl. <i>Verb Concept</i>)</b>  <i>transform(BPM, data_object:DataObject, lane pool:Lane Pool) → SBVR Verb Concept</i>  <i>e.g.: transform(BPM, 'approved rental contract', 'branch') →</i>  <a href="#">branch provide_data_input approved rental contract</a></p> <p>Jeigu duomenų objektas yra įeinantis, tada vartojamas veiksmožodis <i>get_data_output</i>.</p> <p>Veiksmožodiniai konceptai iš BPMN2 duomenų saugyklos BPMN2 elementų yra išgaunami analogiškai vietoj „provide_data_input“ ir „get_data_output“ naudojant „provide_data_store_input“ ir „get_data_store_output“.</p>
Process Rule	<p>Veiklos taisyklių išgavimas iš BPMN2 duomenų objektų, duomenų saugyklų.</p> <p>Tikslas – suformuoti veiklos taisyklę, kuri reikalinga BPMN2 duomenų objektų ir duomenų saugyklų informacijai išsaugoti.</p> <p><b>BPMN2 duomenų objektas (angl. <i>DataObject</i>) (išeinantis) → proceso taisyklė (angl. <i>Process Rule</i>)</b>  <i>transform(BPM, data_object:DataObject, sequence_flow(dataobject, activity: Activity):SequenceFlow, activity:Activity, lane pool:Lane Pool) → SBVR Process Rule</i>  <i>e.g.: transform(BPM, 'approved rental contract', sequence_flow('approved rental contract', 'confirm estimated rental charge'), 'confirm estimated rental charge', 'branch') →</i>  <a href="#">It is necessary that branch provide_data_input approved rental contract when branch confirm estimated rental charge</a></p> <p>Duomenų objektui, kuris yra įeinantis, vartojamas veiksmožodis „get_data_output“.</p> <p>Veiklos taisyklės iš BPMN2 duomenų saugyklos BPMN2 elementų yra išgaunamos analogiškai vietoj „provide_data_input“ ir „get_data_output“ naudojant „provide_data_store_input“ ir „get_data_store_output“.</p>

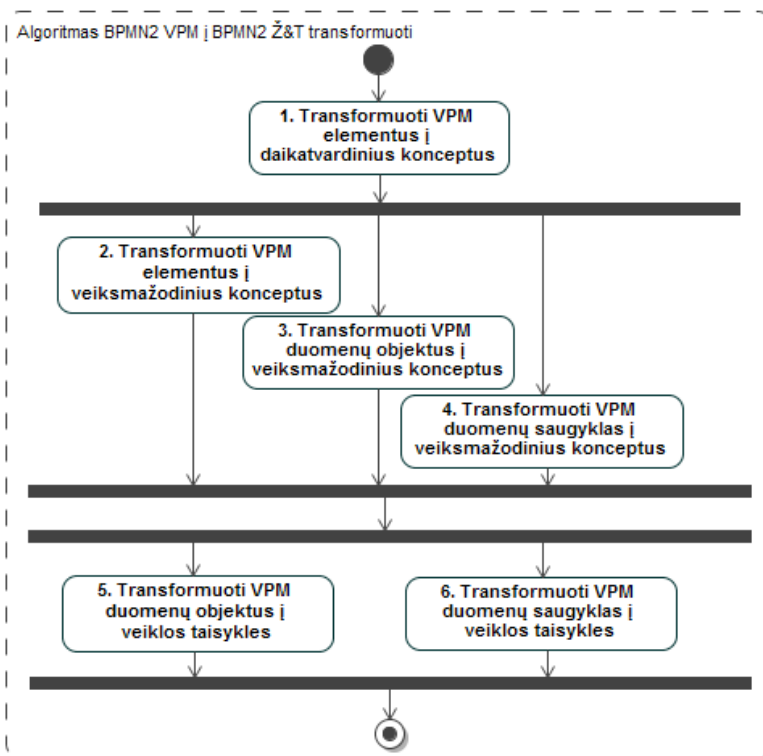
Priešingai negu sekos srautai (*SequenceFlow*), duomenų asociacijos (*DataAssociations*), kurios naudojamos duomenų objektų (*DataObject*) ir duomenų saugyklų (*DataStore*) susiejimui su proceso veiklomis, neturi tiesioginio efekto proceso eigai. Duomenų asociacijos paskirtis – nusakyti veiklų duomenų įeigas arba išeigas. Dėl to, kad šių elementų naudojimo semantika yra susijusi su proceso sudėtimi, o ne veikla, šie duomenų srautai buvo įtraukti į BPMN2 VPM Ž&T, o ne į SBVR VŽ&VT.

Veiksmožodžiai, kurie yra naudojami aprašant BPMN2 VPM Ž transformacijų metu pateikti 4.14 lentelėje. Šių veiksmožodžių yra naudojami tam tikriems BPMN2.

**4.14 lentelė.** BPMN2 VPM į BPMN2 VPM Ž naudojami veiksmazodžiai

Veiksmazodis	Naudojamas BPMN2 elementui
<i>get_data_output</i>	įeinantis <i>DataObject(with state)</i> , <i>DataObject</i>
<i>provide_data_input</i>	išeinantis <i>DataObject(with state)</i> , <i>DataObject</i>
<i>get_data_store_output</i>	įeinantis <i>DataStore</i>
<i>provide_data store_input</i>	Išeinantis <i>DataStore</i>
<i>include</i>	Modelio hierarchijai nurodyti

BPMN2 konceptų metažodynas, kuriame naudojant SBVR aprašomi veiklos procesų diagramos metamodelio elementai, yra naudojamas kuriant BPMN2 VPM Ž&T. Transformacijos BPMN2 VPM į BPMN2 VPM Ž&T algoritmas pateiktas paveiksle 4.14 pav.



**4.14 pav.** Algoritmas BPMN2 VPM į BPMN2 VPM Ž&T transformuoti

Skaičiai veiklų pradžioje nurodo veiklos išsamų paaiškinimą pagal toliau pateiktą numeravimą (veiklos skaičius sutampa su toliau einančiu numeriu).

1. Transformuoti BPMN2 VPM konteinerius (*Lane*, *Pool*), veiklas (*Activity*), įvykius (*Event*), pranešimus (*Message*), duomenų objektus (*DataObject*), duomenų saugyklas (*DataStore*) į SBVR bendrinius konceptus naudojant BPMN2 konceptų metažodyną.
2. Transformuoti BPMN2 VPM elementus į SBVR veiksmazodinius konceptus naudojant „include“ veiksmazodį. Gauti veiksmazodiniai

konceptai nurodo kiekvieno daiktavardinio koncepto priklausomybę konkrečiam procesui ar subprocesui.

3. Transformuoti BPMN2 VPM duomenų objektus (*DataObject*) į SBVR veiksmožodinius konceptus naudojant veiksmožodžius „get\_data\_output“ nurodant išeinantį duomenų objektą ir „provide\_data\_input“ nurodant įeinantį duomenų objektą pagal asociacijų ryšius (*DataAssociation*).
4. Transformuoti BPMN2 VPM duomenų saugyklas (*DataStore*) į SBVR veiksmožodinius konceptus naudojant veiksmožodžius „get\_data\_store\_output“ nurodant išeinančią duomenų saugyklą ir „provide\_data\_store\_input“ nurodant įeinančią duomenų saugyklą pagal asociacijų ryšius (*Association*, *DataAssociation*).
5. Transformuoti BPMN2 VPM duomenų objektus (*DataObject*) į SBVR veiklos taisykles pagal asociacijų ryšius (*DataAssociation*), naudojant tuos pačius veiksmožodžius, kaip 3 veiklos aprašyme.
6. Transformuoti BPMN2 VPM duomenų saugyklas (*DataStore*) į SBVR veiklos taisykles pagal asociacijų ryšius (*DataAssociation*), naudojant tuos pačius veiksmožodžius, kaip 4 veiklos aprašyme.

#### **4.5 SBVR veiklos žodyno ir veiklos taisyklių transformacija į BPMN2 veiklos procesų modelį**

Šiame darbe buvo papildomai nagrinėjama atvirkštinės transformacijos (SBVR VŽ&VT transformacija į BPMN2 VPM) galimybė. Tikslas – išsiaiškinti, ar galima atlikti atvirkštinę transformaciją iš SBVR VŽ&VT į BPMN2 VPM ir kokie šios transformacijos rezultatai. Šiame poskyryje pateikiama šitos transformacijos analizė su algoritmais bei trumpai aprašoma dalinės transformacijos galimybė, įgyvendinta kitame projekte, kurio rezultatai gali būti pritaikomi šiam sprendimui praplėsti.

##### **4.5.1 SBVR veiklos žodyno ir veiklos taisyklių transformacija į BPMN2 veiklos procesų modelį**

BPMN2 elementų identifikavimas gali būti atliekamas iš bendrinių konceptų, veiksmožodinių konceptų ir iš veiklos taisyklių. 4.15 lentelė parodo transformacijos galimybes iš SBVR VT&VŽ į BPMN2 VPM. Daugiausia BPMN2 elementų identifikuoti galima iš SBVR veiklos taisyklių, kadangi pagal jos struktūrą galima nustatyti tam tikrus BPMN2 elementus. Būtina sąlyga – nagrinėjamos taisyklės turi būti gautos atliekant transformaciją iš BPMN2 modelio arba parašytos taikant taisyklių aprašymo formatus, kurie taikomi BPMN2 VPM į SBVR VŽ&VT transformacijoms. Remiantis BPMN2 elementų pavadinimų formavimo taisyklėmis, kurios buvo apibrėžtos 4.8 lentelėje, iš SBVR taisyklių galima identifikuoti BPMN2 elementus. SBVR VŽ&VT suteikia galimybę identifikuoti tik bendrinius BPMN2 elementus, bet ne jų potipius (pvz., galima identifikuoti įvykį, bet ne jo potipį – ar jis pradžios, pabaigos, pranešimo ar kt. įvykis).

Lentelėje naudojamos raidės A, A\* ir M reiškia: A – automatinė transformacija; A\* – automatinė transformacija tuo atveju, jeigu veiklos taisyklės paremtos formatais, kurie taikomi BPMN2 VPM į SBVR VŽ&VT, kitu atveju – M; M – rankinė transformacija (smulkesnė informacija pateikiama 4.2.1 poskyryje).



4.15 lentelė. SBVR VŽ&VT transformacijos į BPMN2 matrica

BPMN2 VPM		SBVR VŽ&VT				
Indeksas		1	2	3	4	5
Elementų grupė	Elementas	<i>General Concept</i>	<i>Individual Concept</i>	<i>Role</i>	<i>Verb Concept</i>	<i>Business Rule</i>
<i>Flow Objects</i>	<i>Event</i>	M <sub>4</sub> , M <sub>5</sub>				
	<i>Activity</i>	M <sub>2</sub>				
	<i>Message</i>	M <sub>4</sub> , M <sub>5</sub>				
	<i>Gateway</i>	M <sub>1</sub> , A* <sub>4</sub> , A* <sub>5</sub>				
	<i>ComplexGateway</i>	A* <sub>5</sub>				
<i>Connecting Objects</i>	<i>SequenceFlow</i>	M <sub>5</sub>				
	<i>SequenceFlow (with Condition)</i>	A* <sub>5</sub>				
	<i>MessageFlow</i>	A* <sub>5</sub>				
	<i>MessageFlow (with ref. Message)</i>	A* <sub>5</sub>				
	<i>Association</i>	A* <sub>5</sub>				
	<i>DataAssociation</i>	M <sub>5</sub>				
<i>Swimlanes</i>	<i>Lane</i>	-				
	<i>Pool</i>	M <sub>1</sub> , A <sub>3</sub> , A* <sub>4</sub> , A* <sub>5</sub>				
<i>Artifacts</i>	<i>DataObject</i>	M <sub>1</sub> , A <sub>3</sub> , A* <sub>4</sub> , A* <sub>5</sub>				
	<i>DataObject (with State)</i>	M <sub>1</sub> , M <sub>3</sub>				
	<i>DataStore</i>	M <sub>1</sub>				
	<i>Group</i>	M <sub>1</sub>				
	<i>TextAnnotation</i>	M <sub>1</sub>				

BPMN2 VPM elementų identifikavimas iš bendrinių konceptų yra tik rankinis (M). Taip yra dėl to, kad neįmanoma nustatyti, kokio tipo BPMN2 VPM elementas. BPMN2 VPM elementų identifikavimas iš veiksmožodinių konceptų gali būti daugeliu atvejų automatinis, tačiau tik tada, kai taikomas specialus taisyklių aprašymo formatas (4.8 lentelė). Tik trys BPMN2 VPM elementai *Event*, *Lane*, *Pool* gali būti transformuojami automatinio būdu. Elementas *Event* identifikuojamas pagal vardo aprašymą (4.8 lentelė), kiti elementai – pagal vietą veiksmožodiniuose konceptuose (pirma daiktavardinė frazė veiksmožodiniame koncepte). BPMN2 VPM elementų identifikavimas iš veiklos taisyklės atliekamas keliais būdais, kai kuriais atvejais net tam pačiam elementui nustatyti. Šiuo atveju nagrinėjame tik tas taisykles, kurios užrašytos naudojantis 4.8 lentele, nes tokiu atveju galimas daugelio BPMN2 VPM elementų išgavimas. Veiklos taisyklių 4 variantai (BR1, BR2, BR3, BR4) (*Business Rule* (BR)), kurie nagrinėjami norint identifikuoti BPMN2 elementus, aprašyti EBNF notacija:

- *consequent1* := [Pool|Lane, Activity][Event]
- *condition1* := [Pool|Lane, Activity][Event][Pool|Lane, MessageFlow|MessageFlow(with ref. Message)]

- *condition2*:= [Gateway su SequenceFlow|SequenceFlow(with Condition)][ConditionalSequenceFlow]
- *BR1*:=consequent1, “after”, condition1, [“if”, condition2]
- *consequent3*:= [Pool|Lane, Activity][Event]
- *condition3*:= [Pool|Lane, Activity][Event]
- *condition4*:= [Gateway su SequenceFlow|SequenceFlow(with Condition)][ConditionalSequenceFlow]
- *BR2*:=consequent2, “if”, condition3, [“when”, condition4]
- *consequent4*:= [Event][Pool|Lane, MessageFlow|MessageFlow(with ref. Message)]
- *condition5*:= [Pool|Lane, Activity][Event]
- *BR3*:= [consequent4, “when”, condition5] [consequent4, “after”, condition5]
- *consequent5*:= [Pool|Lane, Activity][Event]
- *condition6*:= [BoundaryEvent]
- *condition7*:= [Pool|Lane, Activity]
- *BR4*:=consequent5, “if”, condition6, [“while”, condition7]

Aprašyti taisyklių variantai sudaromi iš pasekmių (angl. *consequent*) ir sąlygų (angl. *condition*).

BPMN2 VPM elementai, nagrinėjant SBVR VŽ&VT, gali būti identifikuojami taip.

1) Pagal naudojamus veiksmožodžius (4.7 lentelė). Šiuo būdu galima identifikuoti šiuos BPMN2 elementus:

- pranešimų srautą su pranešimu (*MessageFlow (with ref. Message)*) arba pranešimų srautą (*MessageFlow*) kartu su konteineriu (*Pool* arba *Lane*);
- konteinerius (*Pool* ir *Lane*), kai vienas konteineris (*Pool*) turi savyje kitą konteinerį (*Lane*).

Pvz.: It is obligatory that rental contract is approved if branch receive rental contract approval

It is obligatory that branch receive rental contract approval if renter send rental contract approval

2) Pagal BPMN2 elementų vardų sudarymo taisykles (4.8 lentelė). Šiuo būdu galima identifikuoti tokius BPMN2 elementus:

- veiklą (*Activity*) ir konteinerį (*Pool* arba *Lane*).

Pvz.: It is obligatory that rental contract is approved if branch receive rental contract approval

3) Pagal taisyklės struktūrą. Šiuo būdu galima identifikuoti tokius BPMN2 elementus.

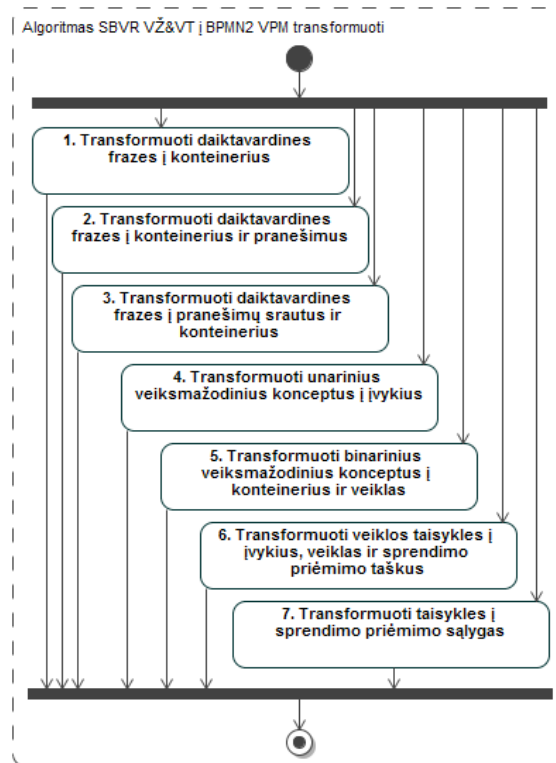
- Sprendimo priėmimo taškas (*Gateway*) – išskiriantis sprendimo priėmimo taškas (*ExclusiveGateway*) gali būti pakeistas su sąlyginiu sekos srautu (*ConditionalSequenceFlow*), kitais atvejais sprendimo priėmimo taško tipas priklauso nuo naudojamos AND/OR konstrukcijos. Reikia išskirti vieną atvejį, kai šios konstrukcijos gali reikšti ribinį įvykį (*BoundaryEvent*) – tokiu atveju, atvirkščiai negu paralelinio

sprendimo priėmimo taško (*ParallelGateway*) atveju, pirma eina jungtukas „if“, po to „and“.

- Sekos srautas su sąlyga (*SequenceFlow (with Condition)*) – identifikavus sprendimo priėmimo tašką (*Gateway*), sekančiomis taisyklėmis galima identifikuoti sąlygas, jei jos egzistuoja atsižvelgiant į sprendimo priėmimo taško tipą. Sąlygos randamos naudojantis surastais BPMN2 elementų vardais po jungtuko „if“.
- Sekos srautas (*SequenceFlow*) – aprašomas taisyklėse, taisyklės struktūra nusako, koks turi būti vykdomų veiklų ar esamų įvykių eiliškumas.

Pvz.: *It is obligatory that branch prepare offer or branch prepare discount or branch prepare offer and branch prepare discount or branch reject car booking request if branch check renter*  
*It is obligatory that branch prepare offer if renter is\_reliable*  
*It is obligatory that branch prepare discount if renter is\_regular*

Transformacijos SBVR VŽ&VT į BPMN2 VPM algoritmas pateiktas 4.15 paveiksle.



4.15 pav. Algoritmas SBVR VŽ&VT į BPMN2 VPM transformuoti

Skaičiai veiklų pradžioje nurodo veiklos išsamų paaiškinimą pagal toliau pateiktą numeravimą (veiklos skaičius sutampa su toliau einančiu numeriu).

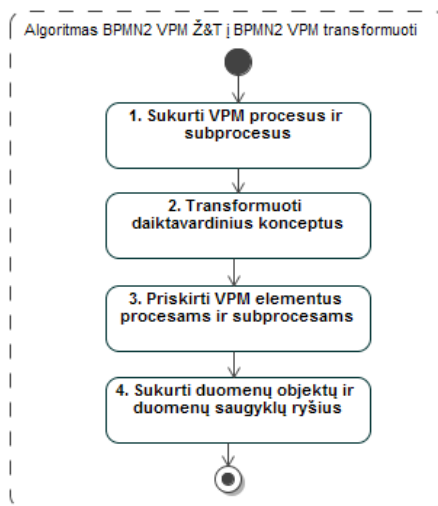
1. Transformuoti daiktavardines frazes iš SBVR veiksmožodinių konceptų su veiksmožodžiu „include“ į konteinerius (*Lane, Pool*).

2. Transformuoti daiktavardines frazes iš SBVR veiksmazodiniu konceptu su veiksmazodziais „receive“ ir „send“ į konteinerius (*Lane, Pool*) ir pranesimu srautus su pranesimais (*MessageFlow with ref. Message*).
3. Transformuoti daiktavardines frazes iš SBVR veiksmazodiniu konceptu su veiksmazodziu „receive\_message\_from“ į pranesimu srautus (*MessageFlow*) ir konteinerius (*Lane, Pool*).
4. Transformuoti SBVR unarinius veiksmazodinius konceptus į įvykius (*Event*).
5. Transformuoti SBVR binarinius veiksmazodinius konceptus į konteinerius (*Lane, Pool*) ir veiklas (*Activity*).
6. Transformuoti AND/OR konstrukcijas iš SBVR veiklos taisykliu į įvykius (*Event*), veiklas (*Activity*) ir sprendimo priemimo taskus (*Gateway*).
7. Transformuoti SBVR veiklos taisykles į sąlygas ant sekos srautu (*SequenceFlow*) prie sprendimo priemimo tasku (*Gateway*).

#### 4.5.2 BPMN2 veiklos procesu modeli žodyno ir taisykliu transformacija į BPMN2 veiklos procesu modeli

Siekiant gauti pilna BPMN2 VPM modeli atliekant atvirkstine transformacija (iš SBVR VŽ&VT į BPMN2 VPM) papildomai naudojamas BPMN2 VPM Ž&T, kuris yra sukurtas naudojant BPMN2 konceptu metažodyna. Bendriniai konceptai analizuojami pagal nagrinejamo žodyno aprasyma: visi žodyne esantys daiktavardiniai konceptai turi nurodyta tipa, pagal kuri identifikuojamas kiekvieno elemento tipas (4.3.2 poskyris). Veiksmazodiniai konceptai leidžia identifikuoti kiekvieno elemento priklausomybe atskiram procesui/subprocesui, taisykle nurodo BPMN2 VPM duomenu objektu ir duomenu saugyklu rysius su kitais transformuotais BPMN2 VPM elementais.

Transformacijos BPMN2 VPM Ž&T į BPMN2 VPM algoritmas pateiktas 4.16 paveiksle.



4.16 pav. Algoritmas BPMN2 VPM Ž&T į BPMN2 VPM transformuoti

Skaičiai veiklų pradžioje nurodo veiklos išsamų paaiškinimą pagal toliau pateiktą numeravimą (veiklos skaičius sutampa su toliau einančiu numeriu).

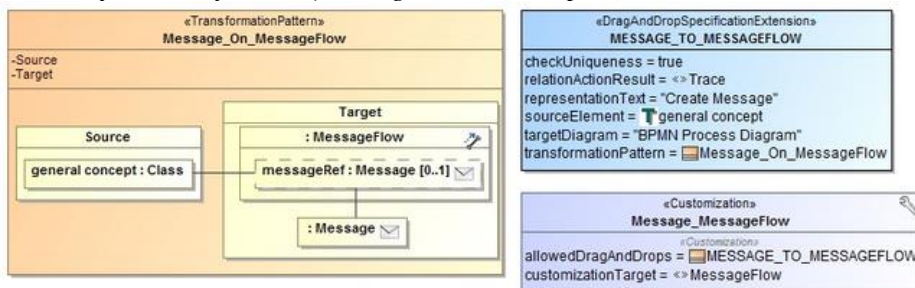
1. Sukurti BPMN2 VPM procesus ir subprocesus naudojant SBVR veiksmožodinius konceptus. Nagrinėjami SBVR veiksmožodiniai konceptai su veiksmožodžiu „include“.
2. Transformuoti SBVR daiktavardinius konceptus iš veiksmožodinių konceptų į BPMN2 VPM elementus.
3. Priskirti transformuotus BPMN2 VPM elementus procesui ar subprocesui pagal SBVR veiksmožodinius konceptus su veiksmožodžiais „include“.
4. Transformuoti SBVR veiklos taisykles į ryšius (*DataAssociation*, *Association*) tarp duomenų (*DataObject*, *DataObject with State*), kuriuose naudojami veiksmožodžiai „get\_data\_output“ ir „provide\_data\_intput“, objektų ir duomenų saugyklų (*DataStore*), kuriuose naudojami veiksmožodžiai „get\_data\_store\_output“ ir „provide\_data\_store\_intput“.

Siekiant gauti galutinę transformacijų rezultatą 4.8 ir 4.9 paveiksluose aprašytus algoritmus reikia taikyti kartu: iš pradžių vykdoma BPMN2 VPM Ž&T transformacija į BPMN2 VPM, vėliau – SBVR VŽ&VT transformacija į BPMN2 VPM papildomai kiekviename žingsnyje tikrinant kiekvieno papildomai sukuriama elemento priklausomybę konkrečiam procesui ar subprocesui pagal BPMN2VPM VŽ esančius veiksmožodinius konceptus, turinčius „include“ veiksmožodį.

#### 4.5.3 SBVR dalinio transformavimo į BPMN2 veiklos procesų modelį realizavimo galimybės

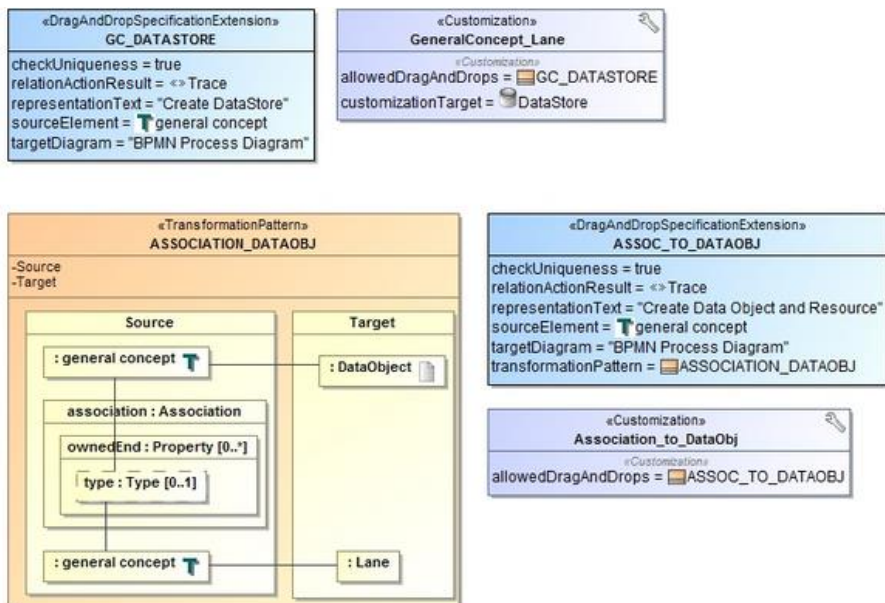
Įrankis *MagicDraw* suteikia galimybę pasinaudoti savybe „drag-and-drop“ (D&D): įrankio aplinkoje tempiant elementą iš veiklos žodyno į BPMN2 VPM diagramą sukurti atitinkamą elementą (kuris priklauso nuo aprašytos taisyklės). Toks sprendimas buvo sukurtas kitame projekte (Skersys et al., 2014), tačiau gali būti pritaikomas ir šiam sukurtam sprendimui norint realizuoti dalinio transformavimo galimybę (transformuojamas ne visas modelis, o atskiri jo elementai). Taikant šį mechanizmą, gali būti realizuotos ne tik automatinės (A, A\*), bet ir dalis rankinių (M) transformacijų. Šiame skyriuje pateikiami du iš sukurtų tokio pobūdžio rankinės transformacijos pavyzdžių.

1. SBVR bendrinis konceptas į BPMN2 pranešimą (*Message*) tempiant ant pranešimų srauto (*MessageFlow*) (4.17 pav.).



4.17 pav. D&D šablonas: SBVR bendrinis konceptas į BPMN2 Message

2. SBVR veiksmožodinis konceptas į BPMN2 duomenų objektą (*DataObject*) ir konteinerį (*Lane*) (4.18 pav.).



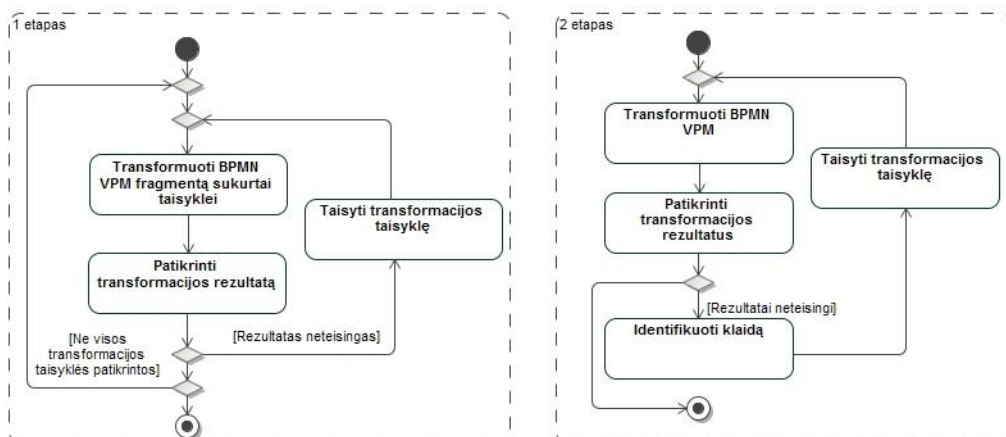
4.18 pav. D&D šablonas: SBVR veiksmožodinis konceptas į BPMN2 *DataObject* ir *Lane*

BPMN2 VPM transformacijų į SBVR VŽ&VT atveju (4.5 lentelė) D&D transformacijos šablonų realizavimas nėra būtinas, nes beveik visos transformacijos taisyklės atliekamos automatiškai. SBVR VŽ&VT transformacijų į BPMN2 VPM atveju (4.15 lentelė) būtų galima realizuoti D&D šablonus transformacijoms: BPMN2 VPM elementų gavimo iš bendrinių ar individualių konceptų (pažymėtas M raide).

#### 4.6 Sukurtų transformacijos taisyklių korektiškumo tikrinimas

BPMN2 VPM į SBVR VŽ&VT transformacijos taisyklių korektiškumo tikrinimas buvo atliekamas tikrinant po transformacijos gautą SBVR veiklos žodyną bei veiklos taisykles. Tikrinimui naudota *Eclipse* įrankio aplinka, kurioje buvo realizuotos transformacijos taisyklės naudojant QVT. Transformacijai panaudotas *EU-Rent* BPMN2 veiklos procesų modelis, kuris buvo kuriamas įtraukiant visus elementų tipus bei įvertinant įvairius jų sujungimo atvejus. Tikrinimas vyko dviem etapais: 1) tikrinamos atskiros transformacijos taisyklės transformuojant tai taisyklei sukurtą BPMN2 VPM fragmentą; 2) tikrinamos visos transformacijos taisyklės transformuojant *EU-Rent* BPMN2 VPM.

Šie etapai leido patikrinti transformacijos taisyklių veikimą kaip atskirų vienetų bei kaip visumos taikant visam BPMN2 VPM. Etapų algoritmai pateikti 4.20 paveiksle.



**4.19 pav.** Sukurtų transformacijos taisyklių korektiškumo tikrinimas: 1 ir 2 etapai

Pirmas tikrinimo etapas buvo atliekamas realizuojant transformacijos taisykles vieno asmens, antras – realizavus visas atskiras transformacijos taisykles visų tyrime dalyvaujančių asmenų. Toks tikrinimo būdas pasirinktas todėl, kad tik žmogus gali įvertinti, ar gautas rezultatas, išreikštas ribota natūralia kalba, yra semantiškai korektiškas ir atitinka VPM prasmę.

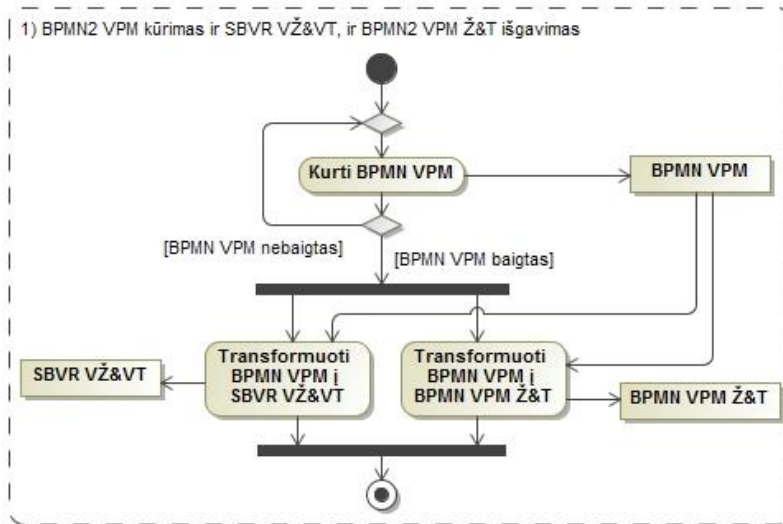
Nuodugnus transformacijos taisyklių tikrinimas, kuris apima šiuos du etapus, atliktas tik su transformacija, kuri buvo realizuota (BPMN VPM į SBVR VŽ&VT). Likusių transformacijų taisyklės nagrinėtos transformacijas atliekant rankiniu būdu pagal 1 etapą kelių projekte dalyvaujančių asmenų.

#### **4.7 BPMN2 veiklos procesų modelio modeliavimo metodas: sukurtų transformacijų taikymas**

Šiame poskyryje pateikiamas BPMN2 VPM modeliavimo metodas taikant sukurtą sprendimą. Buvo išskirti du scenarijai BPMN2 veiklos procesų modelių kūrimui.

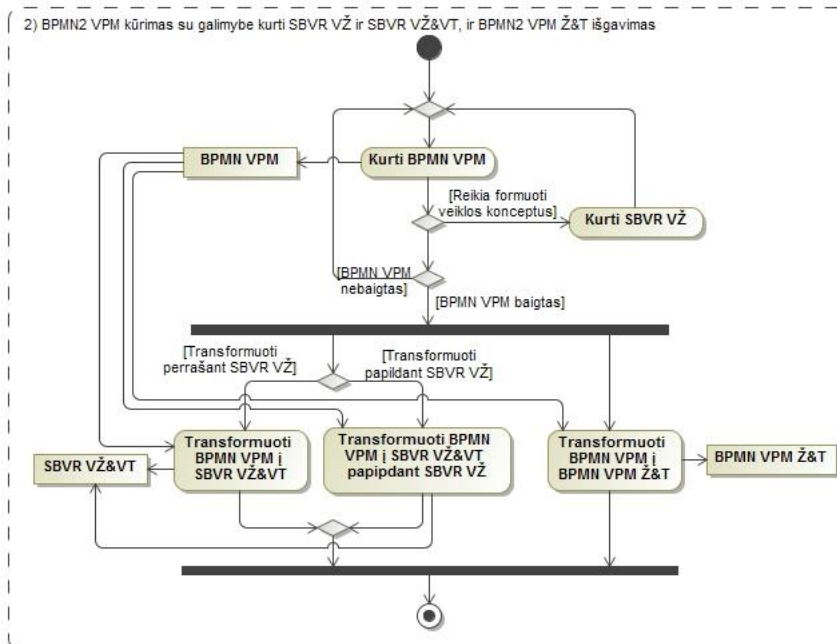
1) Pradžioje sukuriamas BPMN2 VPM, vėliau SBVR VŽ&VT ir BPMN2 VPM Ž&T išgaunami iš grafinio BPMN2 VPM atlikus transformacijas (4.20 pav.).





4.20 pav. BPMN2 VPM kūrimo ir SBVR VŽ&VT, ir BPMN2 VPM Ž&T išgavimo scenarijus

2) BPMN2 VPM kuriamas kartu pildant SBVR VŽ ir pritaikant iš anksto sukurtas D&D transformacijas (pristatyta 4.3.3 poskyryje), vėliau SBVR VŽ&VT yra papildomas (arba perrašomas pagal kūrėjo pasirinktą variantą) ir išgaunamas BPMN2 VPM Ž&T atliekant transformacijas (4.21 pav.).



4. 21 pav. BPMN2 VPM kūrimo panaudojant SBVR VŽ ir SBVR VŽ&VT, ir BPMN2 VPM Ž&T išgavimo scenarijus



#### **4.8 BPMN2 ir SBVR integracijos ir transformacijos sukurto sprendimo apibendrinimas ir išvados**

Išanalizavus veiklos procesų ir veiklos taisyklių integravimo bei transformavimo galimybes nuspręsta naudoti trijų tipų žodynus: įprastą veiklos žodyną, kuris leistų aprašyti veiklos konceptus; BPMN2 konceptų metažodyną, kuris leistų aprašyti BPMN2 elementų tipus, ir veiklos procesų modelio žodyną, kuris leistų aprašyti procesų modelio informaciją ir konstruoti procesų modelio taisykles. Paskutiniai du žodynai yra naudojami veiklos procesų modelio papildomai informacijai išsaugoti, kuri būtų prarasta atlikus BPMN2 VPM transformaciją į SBVR VŽ&VT.

Išsiaiškinti bei detalai apibrėžti BPMN2 ir SBVR standartų elementų atitikimai, kurie leido suformuoti transformavimo taisykles. Aprašytos keturios transformacijos: BPMN VPM į SBVR VŽ&VT; BPMN VPM į BPMN VPM Ž&T; SBVR VŽ&VT į BPMN VPM; BPMN VPM Ž&T į BPMN VPM. Pagrindinis dėmesys skiriamas pirmajai transformacijai, kadangi tokia transformacija yra dažniausiai sutinkama praktikoje modeliuojant dalykinę sritį skirtingais aspektais. Be to, gautas SBVR VŽ&VT gali būti pateikiamas dalykinės srities atstovams validuoti.

Norint suformuoti SBVR VŽ&VT darbe nagrinėjami BPMN2 VPM elementų vardai. Buvo apibrėžti reikalavimai BPMN2 VPM, kurių laikytis būtina norint gauti semantiškai korektišką SBVR VŽ&VT, kadangi sprendime nėra taikomi BPMN2 procesų modelio lingvistinės analizės metodai. Papildomai nagrinėti BPMN2 įvykio elementų parametrai leido išskirti papildomus šablonus su vardų sudarymo reikalavimais BPMN VPM įvykių elementams. Apibrėžtais reikalavimais papildyta darbe aprašyta apibendrinta geroji BPMN2 VPM modeliavimo praktika, kuri papildomai užtikrina vientiso, lengvai skaitomo BPMN VPM kūrimą.

Pateikti sukurto sprendimo taikymo scenarijai, kurie apima pirmas dvi transformacijas. Antroji transformacija yra reikalinga norint išsaugoti BPMN VPM informaciją: BPMN2 elementų tipams, procesų hierarchijai. Tam buvo apibrėžtas BPMN2 konceptų metažodynas, kuris yra papildomai suformuojamas atliekant antrąją transformaciją. Trečioji bei ketvirtoji transformacijos buvo nagrinėjamos papildomai, kaip atvirkštinės pirmajai ir antrajai transformacijoms, norint išsiaiškinti informacijos nuostolius transformuojant BPMN2 VPM į SBVR VŽ&VT. Šių transformacijų taisyklės bei algoritmai apibrėžti formaliai, tačiau nėra realizuoti.

Pasiūlytas sprendimas gali būti pritaikomas kitoms veiklos procesų bei veiklos taisyklių modeliavimo kalboms, papildomai išnagrinėjus jų integracijos galimybes bei metamodelių elementus ir pagal tai atitinkamai pritaikant šiame darbe aprašytas transformacijos taisykles.

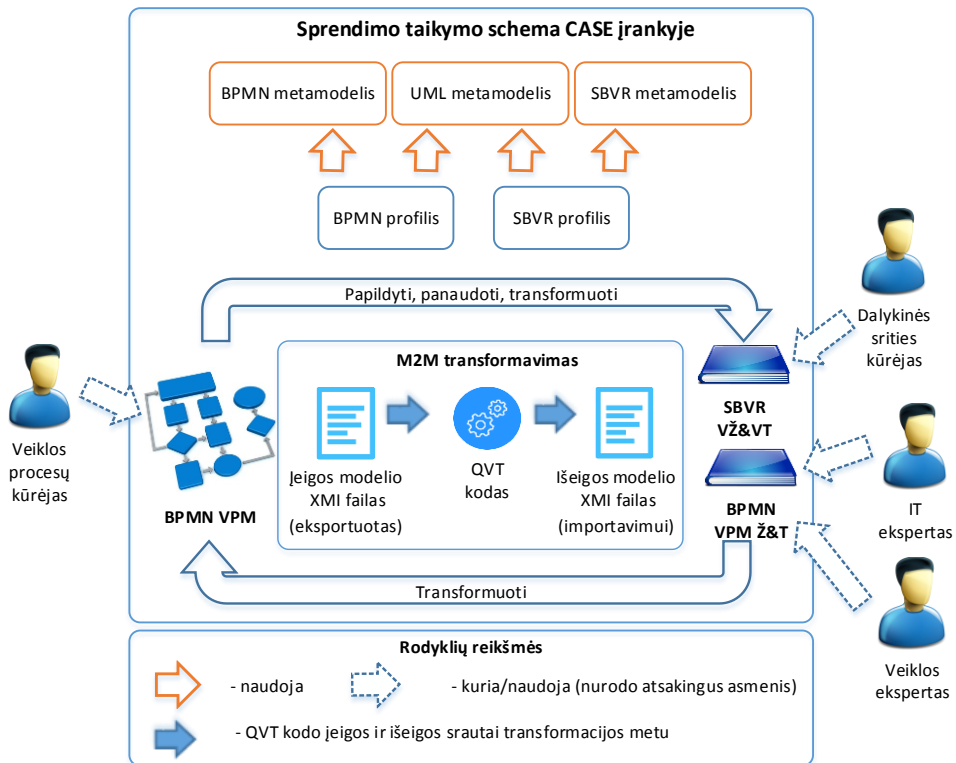
## 5 SBVR VEIKLOS ŽODYNO IR VEIKLOS TAISYKLIŲ IŠGAVIMO IŠ BPMN2 VEIKLOS PROCESŲ MODELIO REALIZACIJA

BPMN2 VPM į SBVR VŽ&VT transformavimo algoritmas realizuotas kaip įskiepis, kuris gali būti naudojamas *MagicDraw* CASE įrankyje. Transformavimo procesas yra automatinis, iškviečiamas per BPMN2 į SBVR konvertavimo sąsają, kurios įeiga – BPMN2 veiklos procesų modelis XMI formatu, išeiga – SBVR veiklos žodynas bei taisyklės XMI formatu. *MagicDraw* CASE įrankyje kartu naudojami BPMN ir SBVR profiliai suteikia galimybę šiuos du modelius atvaizduoti grafiškai, išskyrus SBVR taisykles, kurios yra pateikiamos tekstiniu formatu. Sąsajoje galima papildomai pasirinkti generuoti atsekamumo ryšius tarp dviejų modelių (BPMN2 ir SBVR) elementų.

Skyriuje aprašomas sukurtas SBVR VŽ&VT iš BPMN2 VPM prototipas bei supažindinama su pagrindiniais jo naudojimo principais, pateikiami iliustraciniai pavyzdžiai *MagicDraw* CASE įrankio aplinkoje. Skyriuje aprašoma realizacija buvo paskelbta (Mickeviciute, Butleris, 2014; Mickeviciute et al., 2015) straipsniuose.

### 5.1 Prototipo aprašymas

Sukurto sprendimo įvertinimui sukurtas įskiepis *MagicDraw* aplinkoje, kuris buvo kuriamas atsižvelgiant į darbe sudarytą BPMN2 į SBVR transformacijų principinę schemą (5.1 pav.).



5.1 pav. BPMN2 transformacijų į SBVR taikymas

Veiklos procesų modeliotojas kurdamas BPMN2 VPM gali panaudoti SBVR konceptus, kurie yra apibrėžti dalykinės srities eksperto arba kurti BPMN2 VPM įprastu būdu – nenaudojant SBVR konceptų. SBVR žodynas gali jau egzistuoti įmonėje, gali būti sukurtas anksčiau arba tuo pačiu metu, kai kuriamas VPM. Transformacija iš BPMN2 į SBVR gali būti atliekama CASE įrankiuose, kurie palaiko BPMN ir SBVR metamodelius.

Sukurtas prototipas apima BPMN2 VPM transformavimą į SBVR VŽ&VT. BPMN2 VPM transformavimo į BPMN2 VPM Ž&T galimybė nėra realizuota prototipe, tačiau išsamiai aprašyta šio darbo 4 skyriuje.

Transformacijai BPMN2 VPM į SBVR VŽ&VT realizuoti pasirinktas QVT kalbos operacinis poaibis (QVT *Operational*), kuris yra pilnai palaikomas praktiškai taikomuose įrankiuose (*Eclipse M2M*), taip pat vienintelis turi išsamiai aprašytą dokumentaciją (lyginant su QVT *Relations* ir QVT *Core*). Šios transformavimo kalbos pasirinkimą sąlygojo šie veiksniai: QVT (operacinio poaibio) transformacijų kalba yra OMG grupės standartas, kaip ir BPMN2 ir SBVR bei yra palaikoma projekte inžinerinei realizacijai pasirinkto CASE įrankio *MagicDraw*. *MagicDraw* CASE įrankis QVT (operacinio poaibio) įskiepi yra realizavęs per JSR 223 (JCP, 2014) skriptų procesorių, perpanaudojant *Eclipse M2M* projekto QVT (operacinio poaibio) implementaciją.

Transformacijos kodas realizuotas *Eclipse* (*Eclipse* wiki, 2014) įrankio aplinkoje, vėliau perkeltas į *MagicDraw* CASE įrankio aplinkoje kurtą transformacijų įskiepi (kuris taip pat apima ir kitų modelių transformacijas (5.5 pav.)). Duomenų apsikeitimas vyksta XMI schemų pagrindu. Inicijavus BPMN2 VPM į SBVR VŽ&VT transformaciją *MagicDraw* išeksportuoja BPMN2 veiklos procesų modelio XMI ir paduoda kaip įeigą transformacijos kodui, kurio rezultatas SBVR modelio XMI yra importuojamas ir pavaizduojamas naudojant SBVR profilį. Transformacijos pavyzdžiai QVT kalba yra pateikti toliau: (1) konteinerio transformavimas į bendrinį konceptą; ir (2) veiklos transformavimas į veiksmožodinį konceptą.

-- (1)

```
mapping ActivityPartition::containerToGeneralConcept()
    when { self.isStereotypedBy('Process', 'Lane') } -- mapping guard
    {
        var oneClass:Class :=
        self.returnGeneralConcept(getNameWithNoSpace(getLaneIfRepresents(self)));
        if(oneClass = null) then
        {
            oneClass:= self.map toGeneralConcept(getNameWithNoSpace(getLaneIfRepresents(self)));
            genConArray+=List{oneClass};
        }
        endif;
    }
}
```

-- (2)

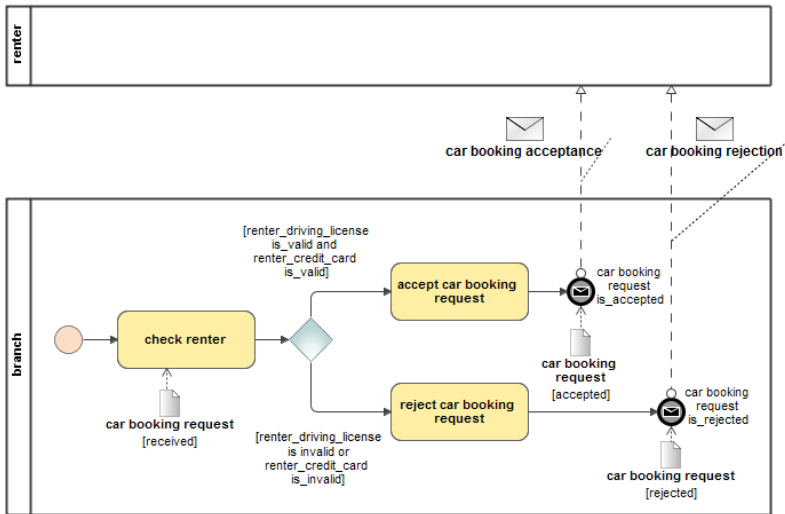
```
mapping ActivityNode::activityToVerbConcept() : Association
-- all types of activities
when { self.stereotypedAsTask() or self.stereotypedAsSubProcess() }
{
    this.modelis.ownedType += result;
    result.applyStereotype(ste_verbConcept);
    result.package:= paketas;
}
```

```

name:=getNameWithNoSpace(self.getActivityVerbPart()); -- verb
-- oneClass for class from Activity
var oneClass:Class :=
self.returnGeneralConcept(getNameWithNoSpace(self.getActivityNounPart()));
if(oneClass = null) then
{
oneClass:=self.maptoGeneralConcept(getNameWithNoSpace(self.getActivityNounPart()));
genConArray+=List{oneClass};
}
endif;
var prop1 : Property:= object Property
{
type:= oneClass;
};
oneClass:=self.returnGeneralConcept(getLaneIfRepresents(getActualContainer(self)));
if(oneClass = null) then
{
oneClass:= self.map toGeneralConcept(getLaneIfRepresents(getActualContainer(self)));
genConArray+=List{oneClass};
}
endif;
var prop2 : Property:= object Property
{
type:= oneClass;
};

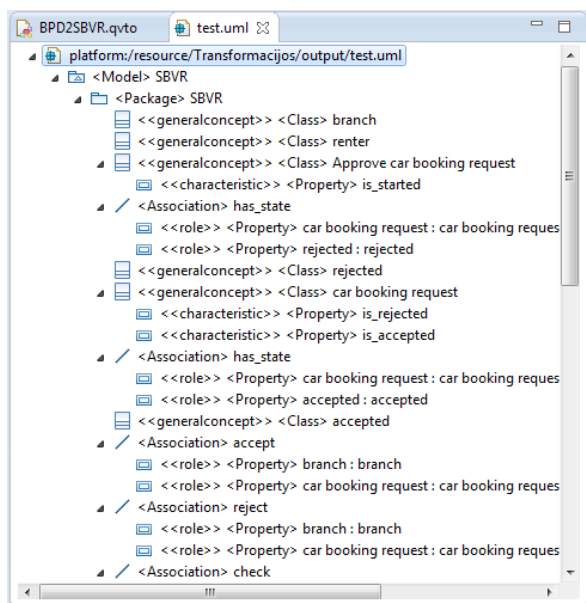
```

Pavyzdinis BPMN2 veiklos procesas pateiktas 5.2 paveiksle. BPMN2 VPM nagrinėjimas vyksta naudojant BPMN profilį, SBVR VŽ&VT kūrimui – SBVR profilis.



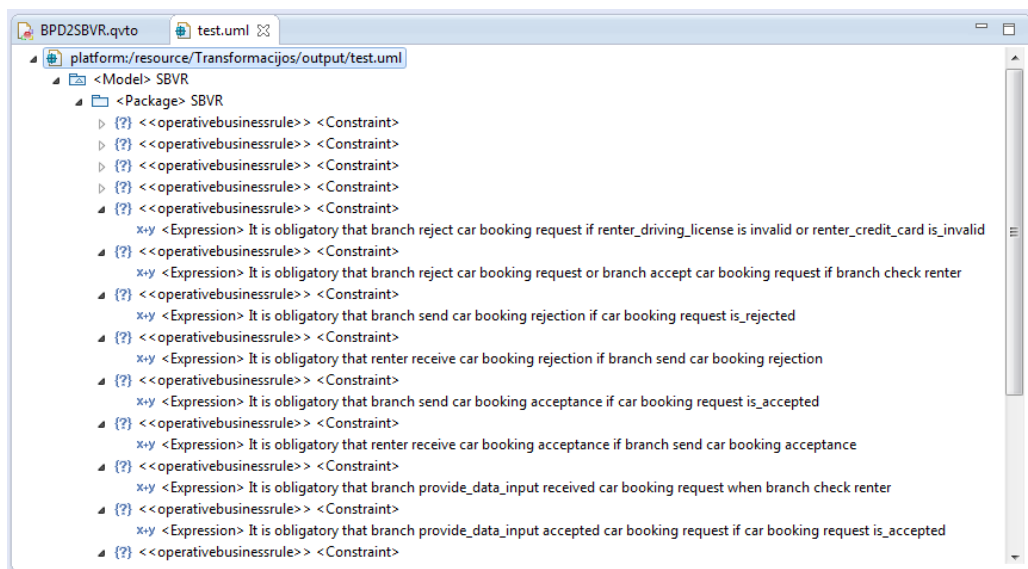
5.2 pav. BPMN2 veiklos procesas „Approve car booking request“

Ivykdžius BPMN2 VPM transformaciją į SBVR VŽ&VT suformuojamas SBVR VŽ (5.3 pav.) ir SBVR VT (5.4 pav.). SBVR VŽ elementams, kurie buvo gauti pagal šiame darbe aprašytas taisykles, priskiriami SBVR profilio stereotipai (matomi kiekvieno elemento, esančio SBVR paketo, pradžioje).



5.3 pav. SBVR veiklos žodynas

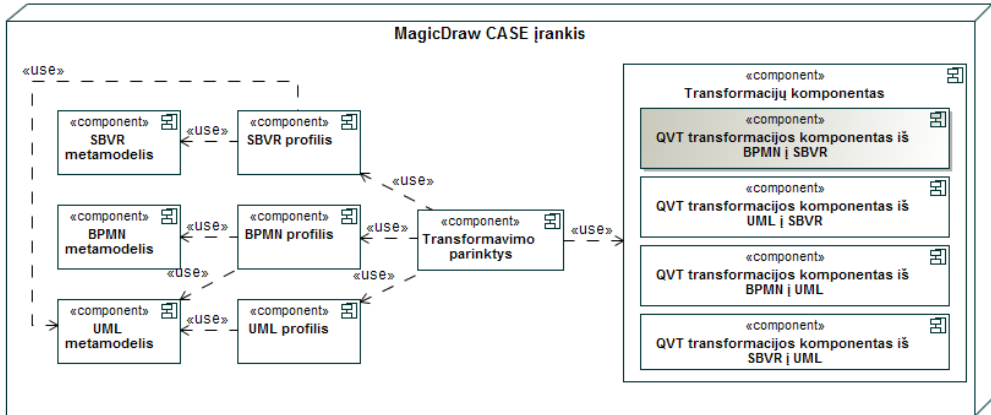
Kartu suformuojamos ir SBVR VT. Šiuo atveju gautoms taisyklėms yra priskiriamas vienas stereotipas, kuris nusako, kad veiklos taisyklės yra operacinės, o ne struktūrinės.



5.4 pav. SBVR veiklos taisyklės

Sukurtos transformacijos funkcionalumas buvo įtrauktas į VEPSEM įskiepi CASE įrankyje *MagicDraw*. Tokia galimybė atsirado, kai buvo sukurtas SBVR profilis. Pateikta komponentų diagrama parodo visas galimas transformacijos komponento QVT transformacijas ir sąsajas su *MagicDraw* CASE įrankio

komponentais (5.5 pav.).



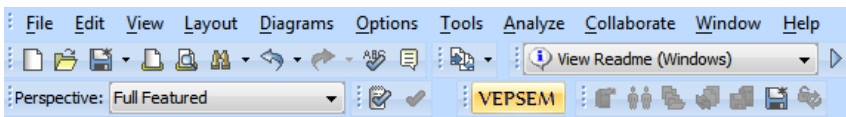
5.5 pav. Komponentų diagrama

Tamsiau pažymėtas transformacijų komponentas (QVT transformacijos komponentas iš BPMN į SBVR) yra sukurtas sprendimo, kuris aprašytas šiame darbe, rezultatas.

Sukurtas sprendimas gali būti pritaikomas ir kituose CASE įrankiuose, kurie palaiko modelių integravimą, UML profilius, DSL bei QVT ir turi jau sukurtą BPMN2 profilį, atsižvelgus į jų profilių realizacines savybes (vidinę profilių struktūrą).

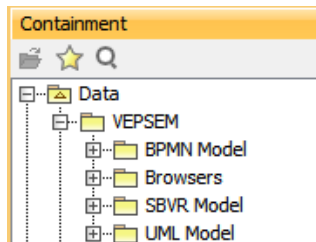
## 5.2 Prototipo trumpas naudojimo aprašas

Įskiepiu galima naudotis (5.6 pav.) tik pagal projekto šabloną kuriamuose projektuose. Kuriant projektą būtina naudoti šabloną „VEPSEM Template Project“. Pagal šį šabloną rengiamas projektas užtikrina būtinąją integruoto modelio medžio struktūrą bei reikiamų *MagicDraw* modulių įtraukimą.



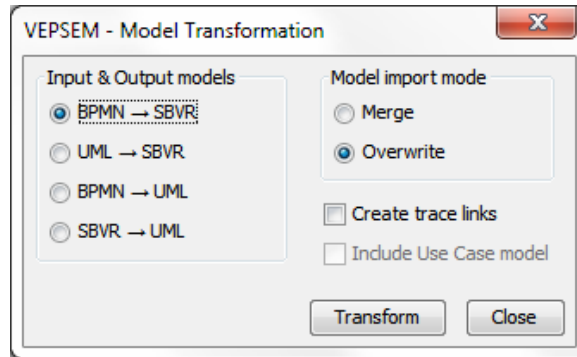
5.6 pav. Įskiepio iškvietimas

Sukurtaame projekte pagal šabloną (5.7 pav.) yra pateikiami jau sukurti visi būtinai projekto katalogai. Norint atlikti BPMN2 VPM transformavimą į SBVR VŽ&VT, BPMN2 VPM reikia kurti katalogo „BPMN Model“ viduje.



5.7 pav. Šablono struktūra

Išskiesto įskiepio transformacijų lange (5.8 pav.) pateikiamos keturios galimos transformacijos parinktys. Šio darbo rezultatas yra pirma parinktis iš transformacijų sąrašo.



5.8 pav. Įskiepio parinkčių langas

Įskiepis suteikia galimybę veiklos procesų modeliavimą derinti kartu su veiklos žodyno bei taisyklių kūrimu. SBVR žodyno diagramos fragmentas pateiktas 9.54 paveiksle (9.4 priedų poskyryje, kaip ir visi kiti iliustraciniai šio poskyrio paveikslai). Kairėje pusėje matomi visi galimi žodyno elementai. SBVR veiklos taisyklės yra vaizduojamos lentelės pavidalu, kaip parodyta 9.55 paveiksle.

Visoms modelių transformacijų kryptims galioja ir viena bendra parinktis – modelio atsekamumo ryšių kūrimo režimas. Pažymėjus varnelę ties „Create trace links“ papildomai sukuriama atsekamumo ryšiai tarp transformacijos įeigos ir išeigos modelių elementų. Šie ryšiai suteikia galimybę sekti atskirų modelių elementų pokyčius bei atitinkamai reaguoti į integralumo taisyklių pažeidimus. Atsekamumo ryšiams tarp skirtingų modelių elementų vizualizuoti naudojamas vienas iš *MagicDraw* funkcionalumų – priklausomybių matricos.

Veiklos žodyno konceptų kūrimas *MagicDraw* aplinkoje pateiktas 9.56 paveiksle. Veiklos žodyno konceptai, kurie gali būti kuriami, pateikti 9.56 (1) paveiksle. Veiklos žodynas gali būti pateiktas grafiniu būdu panaudojant pritaikytą UML klasių modelį (9.56 pav. (2)), kurio visi sukurti SBVR konceptai matomi medžio struktūros meniu 9.56 pav. (3). Vartotojo patogumui galima panaudoti specialią *MagicDraw* įrankio funkciją ir atsidaryti kelias diagramas viename lange, kaip šiuo atveju pateiktame paveiksle – apačioje turint BPMN2 VPM diagramą (9.56 pav. (4)).

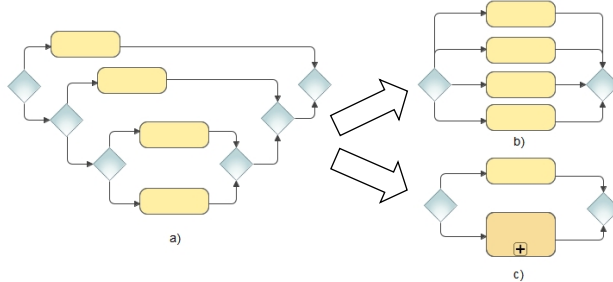
Transformacijos metu gautos SBVR veiklos taisyklės yra talpinamos į lentelę (9.57 pav. (1)). Taip pat yra suteikiama galimybė apibrėžti veiklos apribojimus (9.57 pav. (3)) naudojant SBVR SE (angl. *Structured English*) (9.57 pav. (2)). Tokiu būdu galima aiškiai išskirti veiklą ribojančias taisykles nuo tų taisyklių, kurios nurodo veiklos proceso seką.

### 5.3 Atvejai, kurių neįvertina sukurtas sprendimas

Šiame darbe aprašytas BPMN2 VPM transformavimas į SBVR VŽ&VT neapima šių atvejų.

1. Transformacijos algoritmas neįvertina sprendimo priėmimo taškų, einančių vienas po kito (jei yra daugiau negu 2 sprendimo priėmimo

taškai). Kelių vienas po kito einančių sprendimo priėmimo taškų naudojimas nėra geras sprendimas norint sukurti aiškų BPMN2 VPM. Tokias BPMN2 VPM dalis galima pakeisti joms ekvivalenčiomis naudojant mažiau sprendimo priėmimo taškų (5.13 pav.).



**5.9 pav.** BPMN2 VPM struktūros keitimas norint panaikinti vienas po kito einančius sprendimo priėmimo taškus

2. Transformacijos algoritmas neįvertina išplėstinių subprocesų, kurie naudojami norint matyti detalizuotą subprocesą žemesniame veiklos lygyje toje pačioje veiklos diagramoje. Dėl šios priežasties vietoj išplėstinių subprocesų reikia naudoti subprocesus, ir juose vykstančius procesus pateikti atskirose žemesnio lygio veiklos diagramose. Veiklos procesų modeliavimas naudojant išplėstinius subprocesus arba subprocesus veiklos proceso semantikos nekeičia – tai tik grafinio veiklos proceso vaizdavimo pasirinkimo galimybė.
3. Iškviečiamai veiklai priskirto proceso neužtenka, ji turi turėti vardą, kitu atveju ji ignoruojama.
4. Taisyklių kūrimo metu nėra įvertinama, ar visos taisyklės dedamosios egzistuoja, kad būtų gauta pilna taisyklė, dėl šios priežasties (kai netaikomos gerosios modeliavimo praktikos) yra gaunamos nepilnos ir semantiškai nekorektiškos taisyklės.
5. Transformacijos algoritmas dėl didelės apimties buvo kuriamas neįvertinant visų BPMN2 VPM elementų parametru, išskyrus BPMN2 elementų vardus, taip pat plačiau nagrinėtus ir anksčiau aprašytus BPMN2 laiko įvykių parametrus, paliekant tai tolimesniems tyrimams. Tai nesumenkina sukurto sprendimo, kadangi tas pačias situacijas, kurias nusako parametrai, galima pakeisti joms ekvivalenčiomis veiklos modelyje naudojant daugiau elementų (5.14 pav.).



**5.10 pav.** BPMN2 VPM struktūros keitimas norint pakeisti veiklos parametru nusakantį ciklą

## 5.4 Realizacijos apibendrinimas ir išvados

Realizuotas prototipas – įskiepis *MagicDraw* įrankio aplinkoje, kuris suteikia galimybę vykdyti BPMN2 VPM transformacijas į SBVR VŽ&VT. Toks pats



sprendimas gali būti pritaikomas kituose CASE įrankiuose, palaikančiuose BPMN ir SBVR metamodelius, kuriuose yra UML pagrindu sukurti BPMN ir SBVR profiliai.

Identifikuoti bei aprašyti atvejai, kurių neapima sukurtas sprendimas: sprendimo priėmimo taškų, einančių vienas po kito (jei yra daugiau negu 2 sprendimo priėmimo taškai); išplėstinių subprocesų; išskviečiamai veiklai priskirto proceso neužtenka, ji turi turėti vardą, kitu atveju ji ignoruojama; visų elementų parametrai; taisyklių kūrimo metu nėra įvertinama, ar visos taisyklės dedamosios egzistuoja, kad būtų gauta pilna taisyklė, dėl šios priežasties (kai netaikomos gerosios modeliavimo praktikos) yra gaunamos nepilnos ir semantiškai nekorektiškos taisyklės. Šie atvejai nėra kiek nesumenkina sukurto sprendimo, kadangi atlikti darbai yra tik pradžia; jie vėliau bus plėtojami tolimesniuose tyrimuose.

## 6 SBVR VŽ&VT IŠGAVIMO IŠ BPMN2 VPM EKSPERIMENTAS

Realizuotas prototipas buvo įvertintas atliekant eksperimentą su trimis dalykinės srities BPMN2 VPM norint įvertinti sukurto sprendimo kokybę. Skyriuje pateikiamas atlikto eksperimento aprašas, jo įvertinimo kriterijai, aprašoma eksperimentinio tyrimo eiga. Skyriaus pabaigoje įvertinti gauti eksperimento rezultatai bei identifikuotos grėsmės rezultatų pagrįstumui.

### 6.1 Eksperimentinio tyrimo aprašas ir jo įvertinimo kriterijai

Eksperimentinio tyrimo tikslas atliekant BPMN2 transformavimą į SBVR – įvertinti transformavimo kokybę tam tikrų vertinimo kriterijų atžvilgiu. Eksperimentui buvo naudojamas *EU-Rent* BPMN2 veiklos procesų modelio fragmentas ir dar dvi dalykinės sritys (bibliotekos veikla ir užsakymo vykdymo veikla). Pastarųjų dalykinių sričių modeliai buvo paimti iš *MagicDraw* CASE įrankio bibliotekos. Eksperimentas atliktas dviem etapais: 1) taikant gerąsias modeliavimo praktikas (su visomis paminėtomis dalykinėmis sritimis) ir 2) jų netaikant (dviejose paskutinėse dalykinėse srityse transformavimas buvo atliekamas su netaisytais (pirminiais) modeliais, tokiais, kokie jie buvo laikomi bibliotekoje).

Eksperimente buvo vertinami šie kokybės kriterijai:

- **tikslumas** (angl. *precision*)  $P$  (teisingų gautų SBVR elementų skaičiaus  $RE \cap AE$  santykiu su visais gautais SBVR elementais  $AE$ ):

$$P = \frac{|RE \cap AE|}{|AE|};$$

- **išsamumas** (angl. *recall*)  $R$  (visų gautų SBVR elementų skaičiaus  $RE \cap AE$  santykiu su teoriškai turimų gauti SBVR elementų skaičiumi  $RE$ ):

$$R = \frac{|RE \cap AE|}{|RE|};$$

- **vidurkis** (angl. *balanced F-score*)  $F$ :

$$F = 2 \times \frac{P \times R}{P + R}.$$

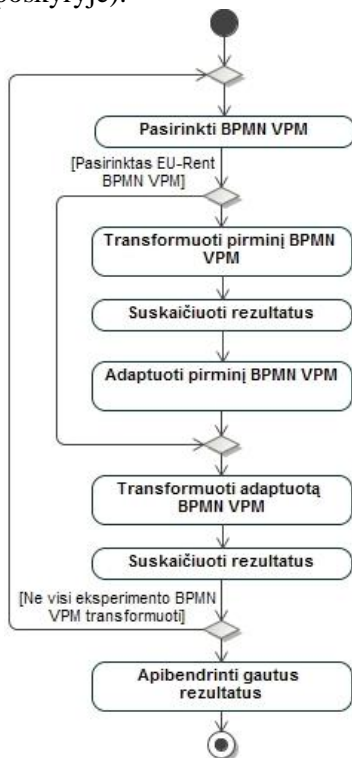
Čia  $RE$  reiškia teoriškai turimų gauti SBVR elementų skaičius;  $AE$  – visų gautų SBVR elementų skaičius;  $RE \cap AE$  – teisingų gautų SBVR elementų skaičius;  $F$  – vidurkis.

Šie kokybiniai vertinimo kriterijai buvo pasirinkti norint įvertinti sukurto sprendimo efektyvumą (išsamumas), gautų rezultatų kokybę (tikslumas) bei sukurto sprendimo atitikimą iškeltiems reikalavimams (vidurkis). Papildomai pateikti kiekybiniai vertinimo kriterijai (gautų elementų kiekis, reikalingų gauti elementų kiekis), kurie reikalingi išvestinių aprašytų kokybinių kriterijų skaičiavimui.

### 6.2 Eksperimentinio tyrimo vykdymas

Eksperimentas buvo vykdomas *MagicDraw* CASE įrankio aplinkoje. Pasirinkti BPMN2 veiklos procesų modeliai buvo transformuojami (taikant 4.11 ir 4.12 paveiksluose pateiktus algoritmus) pirminiai, vėliau – adaptuoti (6.1 pav.). Atlikus

BPMN2 VPM transformacijas į SBVR VŽ&VT gauti rezultatai įvertinti bei apibendrinti pagal šiame darbe aprašytus kriterijus. Visi eksperimente naudoti BPMN2 VPM pateikiami kompaktiniame diske (CD), kuris pridedamas prie disertacijos (9.3 priedų poskyryje).



6.1 pav. Eksperimento algoritmas

Eksperimentas atliekamas naudojant tuos pačius BPMN2 veiklos procesų modelius du kartus:

- 1) pirminį BPMN2 VPM;
- 2) adaptuotą BPMN2 VPM.

Pirminis BPMN2 VPM adaptuojamas pritaikant šiame darbe aprašytus reikalavimus (4.2.4 poskyryje), kurie yra reikalingi norint gauti teisingą bei semantiškai korektišką SBVR veiklos žodyną ir veiklos taisykles iš BPMN2 VPM transformacijos metu.

Papildomai buvo atliekamas eksperimentas ir su *EU-Rent* modelio fragmentu. Kadangi šis modelis buvo kuriamas šio darbo metu taikant gerąją modeliavimo praktiką, *EU-Rent* modelis buvo transformuotas tik vieną kartą.

Eksperimento rezultatų vertinimas (skaičiavimas) po kiekvienos atliktos transformacijos susideda iš 4 dalių:

- 1) veiklos žodyno daiktavardinių frazių išgavimo;
- 2) veiklos veiksmažodinių frazių išgavimo;
- 3) veiklos taisyklių išgavimo;
- 4) žodyno ir veiklos taisyklių suderinamumo sutikrinimo.

Gauti rezultatai įvertinti rankiniu būdu, t. y. kiekvienas veiklos žodyno elementas bei kiekviena veiklos taisyklė įvertinama kaip teisinga arba neteisinga. Toks vertinimo būdas pasirinktas dėl to, kad įvertinti, ar gauta veiklos taisyklė yra semantiškai teisinga, gali tik žmogus. Veiklos žodyno elementas ar veiklos taisyklė pagal realizuotus algoritmus gali būti formuojama teisingai, tačiau veiklos procesų modelio elementų vardai gali netenkinti vardų sudarymo reikalavimų, todėl tiek žodyno elementas gali būti neteisingas, tiek pati suformuota veiklos taisyklė gali būti nekorektiška. Taip pat buvo patikrinama, ar visos gautos taisyklės buvo suformuotos naudojant suformuoto žodyno elementus. Gauti rezultatai taip pat įvertinti ir pagal turimų gauti veiklos žodyno elementų bei veiklos taisyklių skaičius. Stengiantis išvengti klaidų, kurios gali atsirasti dėl žmogaus kaltės, rezultatai buvo vertinami ne mažiau kaip 3 kartus, t. y. tol, kol rezultatai sutampa 3 kartus iš eilės.

### 6.3 Eksperimento rezultatų apibendrinimas

Eksperimento rezultatų apibendrinimas pateikiamas 6.1 ir 6.2 lentelėse.

**6.1 lentelė.** Eksperimento rezultatų apibendrinimas (netaikant gerosios praktikos)

Veikla	$ RE \cap AE $	$ AE $	$ RE $	$P$	$R$	$F$
Biblioteka	55	131	127	0,42	0,43	0,42
Užsakymai	45	88	127	0,51	0,35	0,41
Vidurkis				0,47	0,39	0,42

**6.2 lentelė.** Eksperimento rezultatų apibendrinimas (taikant gerąją praktiką)

Veikla	$ RE \cap AE $	$ AE $	$ RE $	$P$	$R$	$F$
Biblioteka	127	127	127	1	1	1
Užsakymai	127	127	127	1	1	1
EU-Rent	430	430	430	1	1	1

Gauti eksperimento rezultatai rodo, kad gerosios veiklos procesų modeliavimo praktikos taikymas leidžia gauti 100 % transformacijų tikslumą ir išsamumą apibrėžtų transformavimo taisyklių atžvilgiu. Tačiau transformavimo taisyklės neapima sudėtingos procesų logikos, kai, pavyzdžiui, jungiami iš eilės keli sprendimo taškai, nes įvertinti visus galimus variantus sudėtinga. Todėl rekomenduojama taikyti kuo paprastesnius procesų modelius, nes dažnai sudėtingi modeliai atsiranda dėl gerai neištirtos proceso logikos. Jeigu tam tikras situacijas paprasčiau pavaizduoti veiklos procesų modelių nėra įmanoma – galima papildyti veiklos taisykles rankiniu būdu.

Gauti eksperimento rezultatai yra priklausomi nuo pasirinktų BPMN2 VPM. Atlikus eksperimentą su kitais BPMN2 VPM būtų gauti kitokie eksperimento rezultatai iš pirminių modelių transformacijų. Adaptuotiems modelių rezultatams tai neturėtų įtakos, todėl galima daryti išvadą, kad pirma dalis eksperimento priklauso nuo BPMN2 VPM kokybės ir reikalavimų atitikimo bei sprendimo efektyvumo nesumenkina, jei BPMN2 VPM tenkina visus reikalavimus.

## 6.4 Grėsmės rezultatų pagrįstumui

Atlikus eksperimentą buvo išskirtos ir aprašytos dvi grupės grėsmių, kurios gali daryti įtaką gautų rezultatų pagrįstumui.

- Vidinis pagrįstumas. Didžiausią įtaką rezultatams daro transformacijų komponento priklausomybė nuo naudojamo SBVR profilio. Šios grėsmės eliminavimui sukurtas transformacijų komponentas buvo suderintas su SBVR profiliu. Kitas labai svarbus veiksnys rezultatų pagrįstumui yra žmogaus klaidos faktorius atliekant transformacijos rezultatų vertinimą rankiniu būdu. Šiai grėsmei sumažinti kiekvienos transformacijos rezultatai buvo tikrinami iki trijų sutapimų, t. y. ne mažiau kaip tris kartus. Rezultatų reikšmėms įtaką darytų kiti pasirinkti BPMN2 VPM, tai būtų matyti iš rezultatų, gautų po pirminių BPMN2 veiklos procesų modelių transformacijų į SBVR veiklos žodynus ir veiklos taisykles. Šių reikšmių pasikeitimas rodytų, kad transformacijos rezultatai tiesiogiai priklauso nuo BPMN2 VPM kokybės, BPMN2 elementų vardų, modeliavimo stiliaus. Šiuo atveju rezultatai, gauti iš adaptuotų BPMN2 veiklos procesų modelių, liktų nepakitę.
- Išorinis pagrįstumas. Įtaką rezultatams gali daryti nesilaikymas šiame darbe aprašytų reikalavimų BPMN2 veiklos procesų modeliui. Aprašyti apribojimai gali daryti neigiamą įtaką sprendimo taikymui, šiuo atveju aprašyti reikalavimai gali būti laikomi gerosios BPMN2 veiklos procesų modeliavimo praktikos dalimi, kadangi jų laikantis BPMN2 VPM tampa tvarkingesni bei suderinti (vienodo formato veiklų, įvykių pavadinimai ir pan.). Adaptuotų modelių rezultatams gali daryti įtaką atvejai, kurių sukurtas sprendimas neįvertina, tačiau neapimtas situacijas galima pakeisti joms ekvivalenčiomis (smulkiau – 5.3 poskyryje). Taip pat didelę įtaką gali daryti neįvertinti BPMN2 veiklos procesų modeliai (turintys srauto trūkių ir pan.). Nagrinėjant grėsmes dėl nepanaudotų visų BPMN2 elementų savybių buvo nustatyta, kad išreiškus tam tikras BPMN2 elementų savybes (tokias kaip, tarkim, ciklas veikloje) kitais BPMN2 elementais, yra suformuojamas pilnas SBVR VŽ&VT rinkinys, kuris yra pilnai suderintas su BPMN2 VPM pagal transformavimo taisykles; todėl tai nedaro neigiamos įtakos sprendimo efektyvumui bei gaunamų rezultatų kokybei.

## 7 IŠVADOS

1. Literatūros apie veiklos procesų ir veiklos taisyklių modeliavimą gausa rodo, kad norint modeliuoti sudėtingus procesus vieno modeliavimo būdo neužtenka norint perteikti skirtingus veiklos aspektus. Kompleksinis modeliavimo standartų vertinimas parodė, kad tam tinkamiausias yra BPMN ir SBVR modeliavimo kalbos, be to, SBVR išreiškiama ribota natūralia kalba, suprantama visiems veiklos dalyviams bei gali būti apdorojama kompiuterinių sistemų.
2. Atlikta veiklos procesų ir veiklos taisyklių integravimo bei transformavimo sprendimų analizė leido įvertinti esamų sprendimų privalumus bei trūkumus ir apibrėžti išsamesnius BPMN2 ir SBVR metamodelių elementų tarpusavio atitikimus bei transformacijos taisykles, apimančias pilną VPM elementų aibę.
3. Sukurtas metodas suteikia galimybę turėti išsamią bei darnią veiklos procesų modelio specifikaciją, susidedančią iš VPM ir VŽ&VT. Atsižvelgiant į praktikoje pasitaikančius atvejus metodas įgyvendina šiuos specifikavimo scenarijus:
  - 3.1. formuoti grafinius VPM, remiantis VŽ&VT (papildant jei reikia), o vėliau VPT transformuoti į pilną VŽ&VT rinkinį;
  - 3.2. formuoti grafinius VPM ir vėliau VPT transformuoti į VŽ&VT.
4. Sukurtas algoritmas VPM į VŽ&VT transformuoti paremtas šiame darbe aprašytomis transformacijos taisyklėmis bei reikalavimais VPM, užtikrinančiais SBVR standartą atitinkančių VŽ&VT sudarymą. Sukurtas sprendimas užtikrina:
  - 4.1. automatinį VPM transformavimą į VŽ&VT netaikant lingvistinės teksto analizės;
  - 4.2. abipusį BPMN2 ir SBVR modelių suderinamumą bei susiejimą vienoje kūrimo aplinkoje su VŽ elementų atsekamumo galimybe;
  - 4.3. gauto transformacijos rezultato VŽ&VT atitikimą keliamiems reikalavimams pagal SBVR standarto specifikaciją.
5. Realizuotas VPM į VŽ&VT komponento prototipas CASE įrankio aplinkoje, kuris remiasi UML profilių mechanizmo panaudojimu, leidžiančiu išplėsti metamodelius jų nekeičiant, todėl realizacijai buvo panaudotas SBVR profilis. Įrankio prototipo, leidžiančio vykdyti VPM transformaciją į VŽ&VT, realizacija parodė, kad šios transformacijos yra įmanomos bei įgyvendinamos.
6. Atliktas VPM transformavimo į VŽ&VT eksperimentas patvirtino sukurto sprendimo efektyvumą su pasirinktais adaptuotais VPM bei parodė, kad:
  - 6.1. transformuojant pirminius VPM gauti rezultatai tiesiogiai priklauso nuo pačių VPM – kiek VPM tenkina šiame darbe apibrėžtus reikalavimus VPM;
  - 6.2. transformacijas galima taikyti vienam atskiram veiklos procesui, bet ne procesų hierarchijai, taip pat negalima atskirti visų BPMN2 elementų (užduočių, įvykių ir t. t.) tipų; todėl norint išsaugoti šią informaciją atvirkštinės transformacijos galimybei būtina papildomai naudoti VPM Ž&T ir BPMN2 konceptų metažodyną.

## 8 LITERATŪRA

1. (Abdul et al., 2008) Abdul, A. A., Wei, G. K. T., Muketha, G. M., Wen, W. P. Complexity Metrics for Measuring the Understandability and Maintainability of Business Process Models using Goal-Question-Metric (GQM), *International Journal of Computer Science and Network Security*, Vol. 8, No. 5, 2008, p. 219–225.
2. (Aghdasi, Malihi, 2010) Aghdasi, M., Malihi, S. E. Rule Based Business Process Optimization. In *International conference on Industrial Engineering and Engineering Management (IEEM)*, 2010, p. 305–309.
3. (Agrawal, 2011) Agrawal A. Semantics of Business Process Vocabulary and Process Rules. In *ISEC '11 proceedings of the 4th India Software Engineering Conference*, 2011, p. 61–68.
4. (Aguilar-Saven, 2004) Aguilar-Saven, R. S. Business process modelling: Review and framework. *International Journal of production economics*, 2004, 90(2), p. 129–149.
5. (Andreescu, Mircea, 2009) Andreescu, A.I., Mircea, M. Managing Knowledge as Business Rules, *Informatica Economica*, 13(4), 2009, p. 63–74.
6. (Andreescu, Mircea, 2014) Andreescu, A., Mircea, M.: Issues and Challenges of Business Rules Modeling in Software Systems for Business Management. *Informatica Economica* 18(1), 2014, p. 162–170.
7. (Asnina, 2009) Asnina, E. Use of Business Models within Model Driven Architecture. In *Scientific Journal of Riga Technical University. Computer Sciences*, 2009, p. 119–130.
8. (Barjis, 2008) Barjis, J. The Importance of Business Process Modeling in Software Systems Design. *Journal of The Science of Computer Programming*, Vol. 71, Issue 1, 2008, p. 73–87.
9. (Becker et al., 2000) Becker, J., Rosemann, M., Uthmann, C. Guidelines of Business Process Modeling, in: W. van der Aalst, J. Desel, A. Oberweis (Eds.), *Business Process Management. Models, Techniques, and Empirical Studies*, Springer, Berlin et al., 2000, p. 30–49.
10. (Birkmeier et al, 2010) Birkmeier, D., Overhage, S. Is BPMN Really First Choice in Joint Architecture Development? An empirical Study on the Usability of BPMN and UML Activity Diagrams for Business Users. In: *Research into Practice–Reality and Gaps*. Springer Berlin Heidelberg, 2010, p. 119–134.
11. (BPMN MG, 2014) BPMN Modeling Guidelines [interaktyvus], 2014 [žiūrėta 2015-01-22]. Prieiga per: <http://www.modeling-guidelines.org/>
12. (BRG, 2002) Business Rules Group. Defining Business Rules ~ What Are They Really? 4th ed., July 2002 [interaktyvus]. Originally published as GUIDE Business Rules Project Report [žiūrėta 2014-03-15]. Prieiga per: <http://www.BusinessRulesGroup.org>, 1995.
13. (BRG, 2003) Business Rules Group. Veiklos taisyklių manifestas. Taisyklių nepriklausomumo principai (oficialus lietuviškas vertimas, 2005) [inetraktyvus]. 2.0 versija, red. R. G Ross, 2003 [žiūrėta 2014-04-05]. Prieiga per: <http://www.businessrulesgroup.org/brmanifesto.htm>
14. (Caetano et al., 2012) Caetano, A., Pereira, C., Sousa, P. Generation of Business Process Model Views. In *Conference on ENTERprise Information Systems*, Vol. 5, 2012, p. 378–387.
15. (Ceponiene et al., 2009) Ceponiene, L., Nemuraite, L., Vedrickas, G. Semantic Business Rules in Service Oriented Development of Information Systems, *Information*

- Technologies' 2009 : proceedings of the 15th International Conference on Information and Software Technologies, IT 2009, Kaunas, Lithuania, April 23–24, 2009, Kaunas University of Technology. Kaunas : Technologija. ISSN 2029-0020. 2009, p. 404–416.
16. (Ceravolo et al., 2007) Ceravolo, P., Fugazza, C., Leida, M. Modeling Semantics of Business Rules. In proceedings of the Inaugural IEEE International Conference on Digital Ecosystems and Technologies, 2007, p. 171–176.
  17. (Cheng et al., 2011) Cheng, R., Sadiq, S., Indulska, M. Framework for Business Process and Rule Integration: A Case of BPMN and SBVR. *Business Information Systems, LNBIP*, Vol. 87, 2011, p. 13–24.
  18. (Chinosi, Trombetta, 2012) Chinosi, M., Trombetta, A. BPMN: An Introduction to the Standard. In *Computer Standards & Interfaces*, January 2012, Vol. 34, Issue 1, p. 124–134.
  19. (da Silva, 2015). Da Silva, A. R. Model-driven engineering: A survey supported by the unified conceptual model. *Computer Languages, Systems & Structures*, vol. 43, 2015, p. 139–155.
  20. (De Giacomo et al., 2015) De Giacomo, G., Dumas, M., Maggi, F. M., Montali, M. Declarative process modeling in BPMN. In: *International Conference on Advanced Information Systems Engineering*, Springer, Cham, 2015, p. 84–100.
  21. (Eclipse wiki, 2014) ATL/Tutorials - Create a simple ATL transformation, The Eclipse Foundation [interaktyvus], 2014 [žiūrėta 2015-11-17]. Prieiga per: [http://wiki.eclipse.org/ATL/Tutorials\\_-\\_Create\\_a\\_simple\\_ATL\\_transformation](http://wiki.eclipse.org/ATL/Tutorials_-_Create_a_simple_ATL_transformation)
  22. (Eder et al., 2008) Eder, R., Filieri, A., Kurz, T., Heistracher, T.J., Pezzuto, M. Model-Transformation-based Software Generation Utilizing Natural Language Notations. In: *2nd IEEE International Conference on Digital Ecosystems and Technologies, DEST 2008*, DOI: 10.1109/DEST.2008.4635155, 2008, p. 306–312.
  23. (Emerson, Sztipanovits, 2006) Emerson, M., Sztipanovits, J.: Techniques for Metamodel Composition, In *OOPSLA–6th Workshop on Domain Specific Modeling*, 2006, p. 123–139.
  24. (Fahland et al., 2009a) Fahland, D., Lubke, D., Mendling, J., Roijers, H., Weber, B., Weidlich, M., Zugal, S. Declarative versus Imperative Process Modeling Languages: The Issue of Understandability. In *Enterprise, Business-Process and Information Systems Modeling*. Springer Berlin Heidelberg, 2009, p. 353–366.
  25. (Fahland et al., 2009b) Fahland, D., Mendling, J., Reijers, H., Weber, B., Weidlich, M., Zugal, S. Declarative vs. Imperative Process Modeling Languages: the Issue of Maintainability. In *International Conference on Business Process Management*, Springer Berlin Heidelberg, 2009, p. 477–488.
  26. (Figl, Laue, 2011) Figl, K., Laue, R.: Cognitive Complexity in Business Process Modeling. In: Mouratidis, H., Rolland, C. (eds.) *CAiSE 2011*. LNCS, Vol. 6741, Springer, Heidelberg, 2011, p. 452–466.
  27. (Flowers, Edeki, 2013) Flowers, R., Edeki, C. Business process modeling notation. *International Journal of Computer Science and Mobile Computing*, 2(3), 2013, p. 35–40.
  28. (Fortineau et al., 2013) Fortineau, V., Fiorentini, X., Paviot, T., Louis-Sidney, L., Lamouri, S. Expressing Formal Rules Within Ontology-based Models using SWRL: an Application to the Nuclear Industry. *International Journal of Product Lifecycle Management*, 7(1), 2014, p. 75–93.



29. (Friedrich et al., 2011) Friedrich, F., Mendling, J., Puhlmann, F.: Process Model Generation from Natural Language Text. *Advanced Information Systems Engineering, Lecture Notes in Computer Science*, Vol. 6741, Springer, 2011, p. 482–496.
30. (Fuentes-Fernandez, Vallecillo-Moreno, 2004) Fuentes-Fernandez L., Vallecillo-Moreno A.: An Introduction to UML Profiles. *UPGRADE. The European Journal for the Informatics Professional*, vol. 2, 2004, p. 6–13.
31. (Gasevic et al., 2010) Gasevic, D., Guizzardi, G., Taveter, K. *Vocabularies, Ontologies, and Rules for Enterprise and Business Process Modeling and Management. Information Systems*, Vol. 35, Issue 4, 2010, p. 375–378.
32. (Geambasu, 2012) Geambasu, C. V. *BPMN vs. UML Activity Diagram for business process modeling. Accounting and Management Information Systems*, 11(4), 2012, p. 637–651.
33. (Goedertier, Vanthienen, 2007) Goedertier, S., Vanthienen, J. *Declarative Process Modeling with Business Vocabulary and Business Rules. In: OTM 2007 Ws, Part I, LNCS 4805, Springer-Verlag Berlin Heidelberg, 2007, p. 603–612.*
34. (Goedertier, Vanthienen, 2008) Goedertier, S., Vanthienen, J. *A Vocabulary and Execution Model for Declarative Service Orchestration. Business Process Management Workshops, LNCS Vol. 4928, 2008, p. 496–501.*
35. (Graml et al., 2007) Graml, T., Brach, R., Spies, M. *Patterns of Business Rules to Enable Agile Business Processes. In Proceedings of the 11 th IEEE International Enterprise Distributed Object Computing Conference, IEEE Computer Society, 2007, p. 365–375.*
36. (Green et al., 2005) Green, P.F., Rosemann, M., Indulska, M. *Ontological Evaluation of Enterprise Systems. Journal IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, Issue 5, 2005, p. 713–725.
37. (Gruhn, Laue, 2009) Gruhn, V., Laue, R. *Reducing the Cognitive Complexity of Business Process Models, in 2009 8th IEEE International Conference on Cognitive Informatics, ICCI 2009, 2009, p. 339–345.*
38. (Habish et al., 2010) Habish, D., Richly, S., Demuth, B., Gietl, F., Spilke, J., Lehner, W., Assman, U. *Joining Business Rules and Business Processes. In: 16th International Conference on Information and Software Technologies, 2010, p. 361–368.*
39. (Haisjackl, Zugal, 2014) Haisjackl, C., Zugal, S. *Investigating Differences between Graphical and Textual Declarative Process Models. In: CAISE'14 Workshops, LNBIP, vol. 178, Springer, 2014, p. 194–206.*
40. (Hendryx, 2005) Hendryx, S. *Model-Driven Architecture and the Semantics of Business Vocabulary and Business Rules [interaktyvus], 2005 [žiūrėta 2012-12-03]. Prieiga per: <http://www.semanticcore.org/Docs/MDA%20and%20SBVR%20v2.pdf>.*
41. (Hevner et al., 2004) Hevner, AR, March, ST, Park, J., Ram, S. *Design Science in Information Systems Research. MIS Quarterly*, 28(1), 2004, p. 75–105.
42. (Hlaoui, Benayed, 2011) Hlaoui, Y.B., Benayed, L.J. *A Model Transformation Approach Based on Homomorphic Mappings between UML Activity Diagrams and BPEL4WS Specifications of Grid Service Workflows. Computer Software and Applications Conference Workshops (COMPSACW) - 2011, p. 243–248.*
43. (Hohwiller et al., 2011) Hohwiller, J., Schlegel, D., Grieser, G., Hoekstra, Y. *Integration of BPM and BRM. In: Dijkman R, Hofstetter, J, Koehler J (eds.) LNBIP, Vol. 95, Springer, Heidelberg, 2011, p. 136–141.*

44. (IBM KC, 2014) IBM Knowledge Center. Best practices for Process Models [interaktyvus], 2014 [žiūrėta 2015-01-22]. Prieiga per: [http://www.ibm.com/support/knowledgecenter/SSCNZQ\\_7.0.0/com.ibm.ws.icp.devbp.doc/shared/dev\\_guide/devbp/r\\_bpprcsmdl\\_lk.html](http://www.ibm.com/support/knowledgecenter/SSCNZQ_7.0.0/com.ibm.ws.icp.devbp.doc/shared/dev_guide/devbp/r_bpprcsmdl_lk.html)
45. (JCP, 2014) JSR 223: Scripting for the Java™ Platform, Oracle Corporation [interaktyvus], 2014 [žiūrėta 2015-02-18] <http://www.jcp.org/en/jsr/detail?id=223>
46. (Joubert et al., 2013) Joubert, P., De Villiers, C., Kroeze, J. H. A grammar of business rules in Information Systems. TD: The Journal for Transdisciplinary Research in Southern Africa, 2013, 9(2), p. 241–276.
47. (Kamandi, Habibi, 2008) Kamandi, A., Habibi, J. Modeling Languages Study and Evaluation Techniques. In Second Asia International Conference: Modeling & Simulation, 2008, p. 553–558.
48. (Karpovic et al., 2012) Karpovic, J., Nemuraite, L., Stankeviciene, M. Requirements for semantic business vocabularies and rules for transforming them into consistent OWL2 ontologies. In Proceedings of 18th International Conference on Information and Software Technologies (ICIST 2012), Kaunas, Lithuania, September 13–14, 2012, p. 420–435.
49. (Karpovic et al., 2014) Karpovic, J., Kriksciuniene, G., Ablonskis L., Nemuraite, L. The Comprehensive Mapping of Semantics of Business Vocabulary and Business Rules (SBVR) to OWL 2 ontologies. Information Technology And Control, 43(3), 2014, p. 289–302.
50. (Karpovic et al., 2016) Karpovic, J., Ablonskis L., Nemuraite, L., Paradauskas, B. Experimental Investigation of Transformations from SBVR Business Vocabularies and Business Rules to Owl 2 Ontologies. Information Technology And Control, 45(2), 2016, p. 195–207.
51. (Kleiner et al., 2009) Kleiner, M., Albert, P., Bézivin, J. Parsing SBVR-Based Controlled Languages. In: Model Driven Engineering Languages and Systems, LNCS, Vol. 5795, 2009, p. 122–136.
52. (KnowGravity, 2012) KnowGravity. KnowEnterprise [inetraktyvus]. Version 1.7, 2012 [žiūrėta 2014-05-06]. Prieiga per: <http://www.knowgravity.com>
53. (Ko et al., 2009) Ko, R. K., Lee, S. S., Wah Lee, E. Business process management (BPM) standards: a survey. Business Process Management Journal 15(5), 2009, p. 744–791.
54. (Koehler, 2011) Koehler, J. The Process-Rule Continuum – Can BPMN & SBVR Cope With the Challenge? In Commerce and Enterprise Computing (CEC), 2011 IEEE 13th Conference, Luxembourg, 2011, p. 302–309.
55. (Koehler, 2012) Koehler, J. The Process-Rule Continuum – How can the BPMN and SBVR Standards interplay?, Lucerne University of Applied Sciences and Arts, Swicerland, 2012.
56. (Korherr, List, 2006) Korherr, B.; List, B. Extending the UML 2 activity diagram with business process goals and performance measures and the mapping to BPEL. In: International Conference on Conceptual Modeling. Springer Berlin Heidelberg, 2006, p. 7–18.
57. (Kovalova, Turcok, 2014) Kovalova, M., Turcok, L. The Importance Of Business Process Modelling In Terms Of University Education. International Journal of Scientific & Technology Reserch, ISSN 2277-8616, Vol. 3, Issue 12, 2014, p. 112–117.

58. (Krogstie et al., 2006) Krogstie, J., Sindre, G., Jørgensen, H. Process models representing knowledge foraction: a revised quality framework, *European Journal of Information Systems* 15 (1), 2006, p. 91–102.
59. (la Rosa et al., 2011a) la Rosa, M., Wohed, P., Mendling, J., ter Hofstede A. H. M., Reijers, H. A., van der Aalst, W. M. P. Managing Process Model Complexity via Abstract Syntax Modifications, *IEEE Transactions on Industrial Informatics*, Vol. 7, No. 4, 2011, p. 614–629.
60. (la Rosa et al., 2011b) la Rosa, M., ter Hofstede, A. H. M., Wohed, P., Reijers, H. A., Mendling, J., van der Aalst, W. M. P. Managing Process Model Complexity via Concrete Syntax Modifications, *IEEE Transactions on Industrial Informatics*, Vol. 7, No. 2, 2011, p. 255–265.
61. (Lam, 2009) Lam, V. S. W. Equivalences of BPMN processes. *Service Oriented Computing and Applications Journal*, Vol. 3, No.3, 2009, p. 189–204.
62. (Lassen, van der Aalst, 2009) Lassen, K. B., van der Aalst, W. M. P. Complexity Metrics for Workflow Nets, *Information and Software Technology*, Vol. 51, No. 3, 2009, pp. 610–626.
63. (Leopold et al., 2012) Leopold, H., Mendling, J., Polyvyanyy A.: Generating Natural Language Texts from Business Process Models. *Advanced Information Systems Engineering, Lecture Notes in Computer Science*, Vol. 7328, Springer, 2012, p. 64–79.
64. (Linehan, 2008) Linehan, Mark H. SBVR use cases. In *Rule Representation, Interchange and Reasoning on the Web*. Springer Berlin Heidelberg, Vol. 5321, 2008, p. 182–196.
65. (Lovrencic et al., 2006) Lovrencic, S., Rabuzin, K., Picek, R. Formal Modelling of Business Rules: What Kind of Tool to Use? *Journal of Information and Organizational Sciences*, Vol. 30, Issue 2, 2006, p. 225–239.
66. (Lu, Sadiq, 2007) Lu, R., Sadiq, S. A Survey of Comparative Business Process Modeling Approaches. In *International Conference on Business Information Systems (BIS 2007)*, 2007, p. 82–94.
67. (Maldonado et al., 2013) Maldonado, C. A., Gómez-López, M. T., Quintero, A. M. R., Ramos, I. An architecture to infer business rules from event condition action rules implemented in the persistence layer. *Uncovering Essential Software Artifacts through Business Process Archeology*, 2013, p. 201–221.
68. (Malik, Bajwa, 2012) Malik, S., Bajwa, Sarwan, I.: A Rule Based Approach for Business Rule Generation from Business Process Model. *Business Process Management Workshops Rules on the Web: Research and Applications*, LNCS 7438, Springer-Verlag Berlin Heidelberg, 2012, p. 92–99.
69. (Malik, Bajwa, 2013) Malik, S., Bajwa, Sarwan, I.: Back to Origin: Transformation of Business Process Models to Business Rules. In *Business Process Management Workshops*, LNBIP, 132, Springer-Verlag Berlin Heidelberg, 2013, p. 611–622.
70. (Mendling et al., 2008) Mendling, J., Verbeek, H. M. W., van Dongen, B. F., van der Aalst, W. M., Neumann, G. Detection and prediction of errors in EPCs of the SAP reference model. *Data & Knowledge Engineering* 64(1), 2008, p. 312–329.
71. (Mendling et al., 2010a) Mendling, J., Reijers, H. A., Recker, J. C. Activity labeling in process modeling: empirical insights and recommendations. *Information Systems*, 35(4), 2010, p. 467–482.

72. (Mendling et al., 2010b) Mendling, J., Reijers, H. A., van der Aalst, W. M. P. Seven Process Modeling Guidelines (7PMG), *Information and Software Technology*, Vol. 52, No. 2, 2010, p. 127–136.
73. (Mendling, et al., 2007) Mendling, J., Reijers, H.A., Cardoso, J. What Makes Process Models Understandable? *Business Process Management*, 5th International Conference, BPM 2007, Brisbane, Australia, September 24–28, 2007, Proceedings, Lecture Notes in Computer Science, Vol. 4714, Springer, Brisbane, Australia, 2007, p. 48–63.
74. (Mickeviciute et al., 2014a) Mickevičiūtė, Eglė; Nemuraitė, Lina; Butleris, Rimantas. Applying SBVR business vocabulary and business rules for creating BPMN process models // *Business information systems workshops : BIS 2014 international workshops*, Larnaca, Cyprus, May 22–23, 2014 : revised papers / Editors: Witold Abramowicz, Angelika Kokkinaki. Cham: Springer international publishing, 2014. (Lecture notes in business information processing, 183, ISSN 1865-1348), ISBN 9783319114590, p. 105–116.
75. (Mickeviciute et al., 2014b) Mickevičiūtė, Eglė; Pavalkis, Saulius; Nemuraitė, Lina; Butleris, Rimantas. Using SBVR profile for integrating business vocabulary with BPMN process models // *International journal of advances in computer science & its applications*. New York: IRED. ISSN 2250-3765. 2014, vol. 4, iss. 3, p. 108–113.
76. (Mickeviciute et al., 2015) Mickevičiūtė, Eglė; Skersys, Tomas; Nemuraitė, Lina; Butleris, Rimantas. SBVR business vocabulary and rules extraction from BPMN business process models // *Information systems 2015 : proceedings of the 8th IADIS international conference*, 14–16 March 2015, Madeira, Portugal / Edited by M.B. Nunes, P. Isaias, P. Powell. [S.l.]: IADIS Press, 2015, ISBN 9789898533333, p. 211–215.
77. (Mickeviciute et al., 2016) Mickevičiūtė, Eglė; Nemuraitė, Lina; Butleris, Rimantas. Improving BPMN2 Business Process Model to SBVR Business Vocabulary and Business Rules Transformation with BPMN2 Event Naming Patterns // *Information Technology and Control*, 45(4), IT&C 2016. ISSN: 1392-124X, p. 443–451.
78. (Mickeviciute et al., 2017) Mickevičiūtė, Eglė; Butleris, Rimantas; Gudas, Saulius; Karciauskas, Eimutis. Transforming BPMN 2.0 Business Process Model into SBVR Business Vocabulary and Rules // *Information Technology and Control*, 46(3), IT&C 2017. ISSN: 1392-124X, p. 360–371.
79. (Mickeviciute, Butleris, 2013) Mickevičiūtė, Eglė; Butleris, Rimantas. Towards the combination of BPMN process models with SBVR business vocabularies and rules // *Information and Software Technologies : 19th international conference, ICIST 2013*, Kaunas, Lithuania, October 10-11, 2013 : proceedings / [edited by] Tomas Skersys, Rimantas Butleris, Rita Butkiene. Berlin, Heidelberg: Springer, 2013. (Communications in computer and information science, Vol. 403, ISSN 1865-0929), ISBN 9783642419461, p. 114–121.
80. (Mickeviciute, Butleris, 2014) Mickevičiūtė, Eglė; Butleris, Rimantas. The comprehensive modelling of BPMN business processes and business rules using SBVR profile // *IC3K 2014 : 6th international joint conference on knowledge discovery, knowledge engineering and knowledge management : doctoral consortium*, Rome, Italy, 21–24 October, 2014. Lisbon: SciTePress, 2014, p. 57–63.
81. (Miers, 2007) Miers, D. The OMG Business Process Related Standards An emerging set of Standards that enable Model Driven Businesses. *BPM Focus*, 2007, p. 1–11.

82. (Milanovic et al., 2011) Milanovic, M., Gašević, D., Rocha, L. Modeling Flexible Business Process with Business Rule Patterns. In Proceedings of the 15th IEEE International Enterprise Distributed Object Computing Conference, 2011, p. 65–74.
83. (Musham et al., 2008) Musham, P., Singh, S., Bahal, R., Tv, P. Visual SBVR. In: 3rd International Conference, Digital Information Management, ICDIM 2008, 2008, p. 676–683.
84. (Nemuraite et al., 2010) Nemuraite, L., Skersys, T., Sukys, A., Sinkevicius, E., Ablonskis, L. VETIS Tool for Editing and Transforming SBVR Business Vocabularies and Business Rules into UML&OCL Models. In: Information Technologies' 2010 : proceedings of the 16th International Conference on Information and Software Technologies, IT 2010, Kaunas, Lithuania, April 21-23, 2010, p. 377–384.
85. (Normantas, Vasilecas, 2012) Normantas, K., Vasilecas, O. Business rules discovery from existing software Systems. International Journal of Scientific & Engineering Research, 3(10), 2012, p. 1–7.
86. (Normantas, Vasilecas, 2013) Normantas, K., Vasilecas, O. A systematic review of methods for business knowledge extraction from existing software Systems. Baltic Journal of Modern Computing (BJMC), Vol. 1, No. 1–2, 2013, p. 29–51.
87. (Nurcan, 2008) Nurcan, S. A Survey on the Flexibility Requirements Related to Business Processes and Modeling Artifacts. In Proceedings of the 41st Hawaii International Conference on System Sciences HICSS'2008, 2008, p. 378–387.
88. (OMG, 2006) MOF Core Specification v2.0 [inetarkyvus], 2006 [žiūrėta 2014-10-28], Object Management Group (OMG). Prieiga per: <http://www.omg.org>
89. (OMG, 2008) Semantics of Business Vocabulary and Business Rules (SBVR) specification.v1.0 [interaktyvus], OMG Document Number: formal/2008-01-02 [žiūrėta 2013-10-30], Object Management Group (OMG). Prieiga per: <http://www.omg.org/spec/SBVR/1.0/>
90. (OMG, 2011) Business Process Model and Notation (BPMN), v.2.0 [interaktyvus], OMG Document Number: formal/2011-01-03 [žiūrėta 2012-09-26], Object Management Group (OMG). Prieiga per: <http://www.omg.org/spec/BPMN/2.0>
91. (OMG, 2013) Business Process Model and Notation (BPMN), v.2.0.1 [interaktyvus], OMG Document Number: formal/2013-09-02 [žiūrėta 2013-09-10], Object Management Group (OMG), <http://www.omg.org/spec/BPMN/2.0.1>
92. (OMG, 2013b) Semantics of Business Vocabulary and Business Rules (SBVR) specification.v1.2 [interaktyvus], OMG Document Number: formal/2013-11-02 [žiūrėta 2013-11-29] Object Management Group (OMG). Prieiga per: <http://www.omg.org/spec/SBVR/1.2/>
93. (OMG, 2013c) Semantics of Business Vocabulary and Business Rules (SBVR) specification.v1.1 [interaktyvus]. OMG Document Number: formal/2013-09-02 [žiūrėta 2013-11-28] Object Management Group (OMG). Prieiga per: <http://www.omg.org/spec/SBVR/1.1/>
94. (OMG, 2015) OMG Unified Modeling Language (OMG UML) version 2.5 [interaktyvus]. OMG Document Number: formal/2015-03-01 [žiūrėta 2016-09-21] Object Management Group (OMG). Prieiga per: <http://www.omg.org/spec/UML/2.5/>
95. (OMG, 2016) Decision Model and Notation (DMN), v.1.1 [interaktyvus], OMG Document Number: formal/2016-06-01 [žiūrėta 2016-08-22], Object Management Group (OMG). Prieiga per: <http://www.omg.org/spec/DMN/1.1>

96. (OPAALS, 2008) Extended Vocabulary and Rule Set for an Existing ScenarioWP2: Automatic Code Structure and Workflow Generation from Models, Deliverable 2.3 of OPAALS Project (Contract n° IST-034824), 2008.
97. (Ottensooser et al., 2012) Ottensooser, A., Fekete, A., Reijers H. A. Making Sense of Business Process Descriptions: An Experimental Comparison of Graphical and Textual Notations. *Journal Of Systems And Software*, Vol. 85, Issue 3, 2012, p. 596–606.
98. (Owe et al., 2007) Owe, O., Schneider, G., Steffen, M. Components, Objects, and Contracts. Univ. of Oslo, Research Report No. 363, 2007, p. 1–18.
99. (Peixoto et al., 2008) Peixoto, D., Batista, V., Atayde, A., Borges, E., Resende, R., Padua, C. I. P. S. A comparison of BPMN and UML 2.0 activity diagrams.VII Simposio Brasileiro de Qualidade de Software, Vol. 56, 2008, p. 1–12.
100. (Pesic, van der Aalst, 2006) Pesic, M., van der Aalst, W. A Declarative Approach for Flexible Business Processes Management. In *Business Process Management Workshops*. Springer Berlin Heidelberg, 2006, p. 169–180.
101. (Pichler et al., 2012) Pichler, P., Weber, B., Zugal, S., Pinggera, J., Mendling, J., Reijers, H.A. Imperative versus Declarative Process Modeling Languages: An Empirical Investigation. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) *BPM Workshops 2011, Part I*. LNBP, Vol. 99, Springer, Heidelberg, 2012, p. 383–394.
102. (Pinggera et al., 2012) Pinggera, J., Soffer, P., Zugal, S., Weber, B., Weidlich, M., Fahland, D., Mendling, J. Modeling Styles in Business Process Modeling. In *Enterprise, Business-Process and Information Systems Modeling*, Springer Berlin Heidelberg, 2012, p. 151–166.
103. (Pinggera et al., 2015) Pinggera, J., Soffer, P., Fahland, D., Weidlich, M., Zugal, S., Weber, B., Reijers, H., Mendling, J.: Styles in Business Process Modeling: An Exploration and a Model. *Software & Systems Modeling* 14(3), 2015, p. 1055–1080.
104. (Popp, Kaindl, 2014) Popp, R., Kaindl, H. Automated Adaptation of Business Process Models Through Model Transformations Specifying Business Rules. In: *Joint Proceedings of the CAiSE 2014 Forum and CAiSE 2014 Doctoral Consortium*, Thessaloniki, Greece, June 18–20, 2014, p. 65–72.
105. (Radgui et al., 2012) Radgui, M., Saidi, R., Mouline, S. Extracting reusable fragments from business process using BPMN. In *Second International Conference on Digital Object*, 2012, p. 424–429.
106. (Raj et al., 2008) Raj, A., Prabhakar, TV, Hendryx, S. Transformation of SBVR Business Design to UML Models. In: *Proceedings of the 1st conference on India software engineering conference*, Hyderabad, India: ACM, DOI: 10.1145/1342211.1342221, 2008, p. 29–38.
107. (Rajabi, Lee, 2009) Rajabi, B. A., Lee, S. P. Change Management in Business Process Modeling Based on Object Oriented Petri Net. *International Journal of Human and Social Sciences*, Vol. 50, 2009, p. 12–17.
108. (Recker et al., 2005) Recker, J. C., Indulska, M., Rosemann, M., Green, P. Do Process Modelling Techniques Get Better? A Comparative Ontological Analysis of BPMN. In *16th Australasian Conference on Information Systems*, November 30–December 2, Sydney, Australia, 2005, p. 1–10.
109. (Recker et al., 2006) Recker, J. C., Indulska, M., Rosemann, M., Green, P. How Good is BPMN Really? Insights from Theory and Practice. In *14th European Conference on Information Systems*, June 12–14, 2006, Goeteborg, Sweden, pp. 1582–1593.

110. (Recker et al., 2009) Recker, J., zur Meuhlen, M., Siau, K., Ericson, J., Indulska, M. Measuring Method Complexity: UML versus BPMN. In 15th Americas Conference on Information Systems, 6–9 August, 2009, San Francisco, California, 2009, p. 1–9.
111. (Recker, 2005) Recker J. Conceptual Model Evaluation. Towards more Paradigmatic Rigor. In CAiSE'05 Workshops, FEUP, Porto, Portugal, 2005, vol. 1, p. 569–580.
112. (Recker, 2008) Recker, J. C. BPMN modeling—who, where, how and why. *BPTrends*, 5(3), 2008, p. 1–8.
113. (Recker, 2010) Recker, J. Opportunities and constraints: the current struggle with BPMN. *Business Process Management Journal*, 16(1), 2010, p. 181–201.
114. (Recker, Mendling, 2006) Recker, J. Mendling, J. On the Translation between BPMN and BPEL: Conceptual Mismatch between Process Modeling Languages. In Workshops and Doctoral Consortium for the 18th International Conference on Advanced Information Systems Engineering, Namur University Press, 2006, p. 521–532.
115. (Reijers, Mendling, 2011) Reijers, H.A., Mendling, J. A Study into the Factors that Influence the Understandability of Business Process Models *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 41(3), 2011, p. 449–462.
116. (Rodrigues et al., 2015) Rodrigues, R. de A., Barros, M. de O., Revoredo, K., Azevedo, L. G., Leopold, H. An Experiment on Process Model Understandability Using Textual Work Instructions and BPMN Models, *Software Engineering (SBES), 2015 29th Brazilian Symposium*, 2015, p. 41–50.
117. (Roover et al., 2012) Roover, WD, Caron, F., Vanthienen, J. A Prototype Tool for the Event-Driven Enforcement of SBVR Business Rules In: *BPM Workshops, Part I, LNBIP Vol. 99*, Springer-Verlag Berlin Heidelberg, 2012, p. 446–457.
118. (Ross, 1997) Ross, R. *The Business Rule Book*. Business Rule Solutions, Houston, 2nd ed., 1997.
119. (Ross, 2003) Ross R. *Principles Of The Business Rule Approach*. Addison Wesley, ISBN 0-201-78893-4, 2003.
120. (Sadiq, Orłowska, 1997) Sadiq, W., Orłowska, M. On Correctness Issues in Conceptual Modeling of Workflows. In proceedings of European Conference on Information Systems (ECIS '97), Cork, Ireland, 1997, p. 1–22.
121. (Schacher, 2006) Schacher, M. Business Rules from an SBVR and an xUML Perspective (Parts 1–3). *Business Rules Journal*, 2006, Vol. 7, p. 6–8.
122. (Selic, 2007) Selic, B. A systematic approach to domain-specific language design using UML, In Tenth IEEE International Symposium on ObjectOriented Real-Time Distributed Computing (ISORC 2007), 2007, p. 2–9.
123. (Silingas et al., 2009) Silingas D., Vitiutinas R., Armonas A., Nemuraite L.: Domain-Specific Modeling Environment Based on UML Profiles, *Information Technologies*, 2009, p. 167–177.
124. (Silver, 2008) Silver, B. Ten tips for effective process modeling, *Bpms watch*, BPMInstitute.org [interaktyvus], 2008. [žiūrėta 2016-08-15]. Prieiga per: <http://www.bpmnstitute.org/articles/articleasharticle/bpms-watch-ten-tips-foreffective-process-modeling.html>
125. (Sinogas et al., 2001) Sinogas, P., Vasconcelos, A., Caetano, A. Business Processes Extensions to UML Profile for Business Modeling. In proceedings of the 3rd International Conference on Enterprise Information Systems (ICEIS 2001), Setubal, Portugal, 2001, p. 673–678.

126. (Sinur, 2009) Sinur, J. The Art and Science of Rules vs. Process Flows. Research Report G00166408, Gartner, 2009.
127. (Skersys et al., 2012a) Skersys T., Tutkutė L., Butleris R.: The Enrichment of BPMN Business Process Model with SBVR. In Proceedings of the 34th International Conference, Information Technology Interfaces (ITI), 2012, p. 65–72.
128. (Skersys et al., 2012b) Skersys, T., Tutkutė L., Butleris, R., Butkienė, R. Extending BPMN Business Process Model with SBVR Business Vocabulary and Rules. ISSN 1392-124X, Information Technology and Control, Vol. 41, No. 4, 2012, pp. 356–367.
129. (Skersys et al., 2013) Skersys, T., Butleris R., Kapocius, K., Vileiniskis, T.: An Approach for Extracting Business Vocabularies from Business Process Models. Information Technology and Control, 42(2), 2013, p. 178–190.
130. (Skersys et al., 2014) Skersys, T., Pavalkis, S., Lagzdinytė-Budnikė, I. Model-Driven Approach and Implementation of Partial Model-to-Model Transformations in a CASE Tool. Proceedings of the 20th International Conference, International Conference on Information and Software Technologies (ICIST), 2014, p. 260–271.
131. (Skersys et al., 2015) Skersys, T., Pavalkis, S., Nemuraite, L. Implementing semantically rich business vocabularies in CASE tools\*. In T. E. Simos, & C. Tsitouras (Eds.), AIP Conference Proceedings, 1648(1), 2015, p. 310002-1–310002-4.
132. (Steen et al., 2010) Steen, B., Pires, LF, Iacob, M. Automatic Generation of Optimal Business Processes from Business Rules. In: 4th IEEE International Enterprise Distributed Object Computing Conference Workshops, IEEE, 2010, p. 117–126.
133. (Steinke, Nickolette, 2003) Steinke, G., Nickolette, C. Business Rules as the Basis of an Organization’s Information Systems. In Industrial Management & Data Systems, 2003, p. 52–63.
134. (Tantan, Akoka, 2014) Tantan, OC, Akoka, J. Automated Transformation of Business Rules into Business Processes. From SBVR to BPMN. In: International Conference on Software Engineering & Knowledge Engineering, July 1–3, Hyatt Regency, Vancouver, Canada, 2014, p. 684–687.
135. (Tommasi, Corallo, 2006) Tommasi, MD, Corallo, A. SBEAVER: A Tool for Modeling Business Vocabularies and Business Rules, in KES 2006, Part III, LNAI Vol. 4253, Springer-Verlag Berlin Heidelberg, 2006, p. 1083–1091.
136. (Tristotech, 2016) Tristotech. BPMN Modeling Best Practices [interaktyvus], 2016 [žiūrėta 2016-04-02]. Prieiga per: <http://www.trisotech.com/articles/bpmn-modeling-best-practices>
137. (van der Aalst et al., 2003) van der Aalst, W., ter Hofstede, A.H.M., Weske, M. Business Process Management: a Survey. In Proceedings of the International Conference on Business Process Management, BPM 2003, Eindhoven, The Netherlands, 26–27 June, 2003, p. 1–12.
138. (Vasconcelos et al., 2001) Vasconcelos, A., Caetano, A., Neves, J., Sinogas, P., Mendes, R., Tribolet, J. A Framework for Modeling Strategy, Business Processes and Information Systems. In proceedings of the 5th IEEE International Conference on Enterprise Distributed Object Computing, EDOC 2001, IEEE Press. Seattle, USA, 2001, p. 69–81.
139. (Vasilecas, Lebedys, 2006) Vasilecas, O., Lebedys, E. Moving Business Rules from System Models to Business Rules Repository. INFOCOMP Journal of Computer Science, Vol. 5, No 2, 2006, p. 11–17.



140. (Vasilecas et al., 2014) Vasilecas, O., Laureckas, E., Rima, A. Analysis of using resources in business process modeling and simulation. *Applied computer systems : scientific journal of RTU*. Riga: RTU (Riga Technical University) press, vol. 16, 2014, p. 19–25.
141. (Vasilecas et al., 2015) Vasilecas, O., Kalibatiene, D., Rima, A., Birzniece, I., Rudzajs, P. A resource model for the rule-based dynamic business process modelling and simulation. *The 2015 European Simulation and Modelling Conference - ESM'2015, 2015, Leicester, United Kingdom*. Ostend: EUROSIS-ETI Publication, 2015, p. 36–41.
142. (Vasilecas, Normantas, 2011) Vasilecas, O., Normantas, K. Deriving Business Rules from the Models of Existing Information Systems. *Proceedings of the 12th International Conference on Computer Systems and Technologies, ACM, 2011*, p. 95–100.
143. (VETIS, 2009) Business Rules Solutions for Information Systems Development, Research project No K B-04/2008, supported by Lithuanian State Science and Studies Foundation (2008–2009).
144. (Wahl, Sindre, 2005) Wahl, T., Sindre, G. An Analytical Evaluation of BPMN Using a Semiotic Quality Framework. In *CAiSE'05 Workshops, Faculdade de Engenharia da Universidade do Porto, Porto, 2005, Vol.1*, p. 533–544.
145. (Wang et al., 2014) Wang, W., Indulska, M., Sadiq, S. Integrated Modelling of Business Process Models and Business Rules: A Research Agenda. In: *25th Australasian Conference on Information Systems, 8th–10th Dec 2014, Auckland, New Zealand, 2014*, p. 1–10.
146. (Wang et al., 2016) Wang, W., Indulska, M., Sadiq, S. To Integrate or not to Integrate – The Business Rules Question. *Proceedings of the 28th International Conference on Advanced Information Systems Engineering, Springer International Publishing, 2016*, p. 51–66.
147. (Wang, 2016) Wang, W. Business Process Model and Business Rule Integration – Towards a Decision Framework [interaktyvus], 2016 [žiūrėta 2016-06-02]. Prieiga per: <http://ceur-ws.org/Vol-1603/10000062.pdf>
148. (White, 2004) White A. S. Process Modeling Notations and Workflow Patterns, *BPTrends, L. Fischer, ed., pp. 265-294, Future Strategies, Inc., 2004*, p. 1–25.
149. (Wilke et al., 2010) Wilke, C., Dietrich, J., Demuth, B. Event-Driven Verification in Dynamic Component Models. In: *Proceedings of Fifteenth International Workshop WCOP2010, 2010*, p. 79–86.
150. (Wu et al., 2011) Wu, Z., Yao, S., He, G., Xue, G. Rules Oriented Business Process Modeling. In: *Internet Technology and Applications (iTAP), 2011 International Conference, IEEE, DOI: 10.1109/ITAP.2011.6006338, 2011*, p. 1–4.
151. (zur Meuhlen et al., 2007) zur Meuhlen, M., Indulska, M., Kamp, G. Business Process and Business Rule Modeling: A Representation Analysis. *Eleventh International IEEE EDOC Conference Workshop (EDOCW'07), 2007*, p. 189–196.
152. (zur Meuhlen, Recker, 2008) zur Muehlen, M., Recker, J. How much Language is Enough? Theoretical and Practical Use of the Business Process Modeling Notation. In *Advanced information systems engineering. Springer Berlin Heidelberg, 2008*, p. 465–479.
153. (zur Muehlen et al., 2008) zur Muehlen, M., Indulska, M., Kittel, K. Towards Integrated Modeling of Business Processes and Business Rules. In *19th Australasian Conference on*

Information Systems (ACIS) - Creating the Future: Transforming Research into Practice, Christchurch, New Zealand, December 3–5, 2008, p. 690–697.

154. (zur Muehlen, Indulska, 2010) zur Muehlen, M., Indulska, M. Modeling languages for business processes and business rules: A representational analysis. *Information systems*, 35(4), 2010, p. 379–390.

## 9 MOKSLINIŲ PUBLIKACIJŲ DISERTACIJOS TEMA SĄRAŠAS

### **Mokslinės informacijos instituto duomenų bazės „ISI Web of Science“ leidiniuose, turinčiuose citavimo indeksą**

1. Mickevičiūtė, Eglė; Nemuraitė, Lina; Butleris, Rimantas. Improving BPMN2 Business Process Model to SBVR Business Vocabulary and Business Rules Transformation with BPMN2 Event Naming Patterns // *Information Technology and Control*, 45(4), IT&C 2016. ISSN: 1392-124X, pp. 443–451.
2. Mickevičiūtė, Eglė; Butleris, Rimantas; Gudas, Saulius; Karčiauskas, Eimutis. Transforming BPMN 2.0 Business Process Model into SBVR Business Vocabulary and Rules // *Information Technology and Control*, 46(3), IT&C 2017. ISSN: 1392-124X, pp. 360–371.

### **Mokslinės informacijos instituto duomenų bazės „ISI Web of Science“ leidiniuose, neturinčiuose citavimo indeksą**

1. Mickevičiūtė, Eglė; Nemuraitė, Lina; Butleris, Rimantas. Applying SBVR Business Vocabulary and Business Rules for Creating BPMN Process Models // *Business Information Systems Workshops: BIS 2014 International Workshops*, Larnaca, Cyprus, May 22–23, 2014: Revised Papers / Editors: Abramowicz, Witold and Kokkinaki, Angelika. Cham: Springer International Publishing, 2014. (Lecture Notes in Business Information Processing, 183, ISSN 1865-1348), ISBN 9783319114590. pp. 105–116.
2. Mickevičiūtė, Eglė; Butleris, Rimantas. Towards the Combination of BPMN Process Models with SBVR Business Vocabularies and Rules // *Information and Software Technologies: 19th International Conference, ICIST 2013, Kaunas, Lithuania, October 10–11, 2013: proceedings* / [edited by] Skersys, Tomas; Butleris, Rimantas and Butkiene, Rita. Berlin, Heidelberg: Springer, 2013. (Communications in Computer and Information Science, Vol. 403, ISSN 1865-0929), ISBN 9783642419461. pp. 114–121.

### **Kituose recenzuojamuose mokslo leidiniuose paskelbti straipsniai**

1. Mickevičiūtė, Eglė; Skersys, Tomas; Nemuraitė, Lina; Butleris, Rimantas. SBVR Business Vocabulary and Rules Extraction from BPMN Business Process Models // *Information Systems 2015: Proceedings of the 8<sup>th</sup> IADIS International Conference*, March 14–16, 2015, Madeira, Portugal / Edited by Nunes, M.B., Isaias, P., Powell, P. [S.l.]: IADIS Press, ISBN 9789898533333. pp. 211–215.
2. Mickevičiūtė, Eglė; Butleris, Rimantas. The Comprehensive Modelling of BPMN Business Processes and Business Rules Using SBVR Profile // *IC3K 2014: 6<sup>th</sup> International Joint Conference on Knowledge Discovery, Knowledge Engineering*

and Knowledge Management: Doctoral Consortium. Rome, Italy, October 21–24, 2014. Lisbon: SciTePress. pp. 57–63.

**Periodiniuose leidiniuose ir vienkartinuose straipsnių rinkiniuose ir kt.  
paskelbti straipsniai**

1. Mickevičiūtė, Eglė; Pavalkis, Saulius; Nemuraitė, Lina; Butleris, Rimantas. Using SBVR Profile for Integrating Business Vocabulary with BPMN Process Models // International Journal of Advances in Computer Science & Its Applications. New York: IRED. ISSN 2250-3765. 2014, vol. 4, iss. 3, pp. 108–113.

## PADEKOS ŽODIS

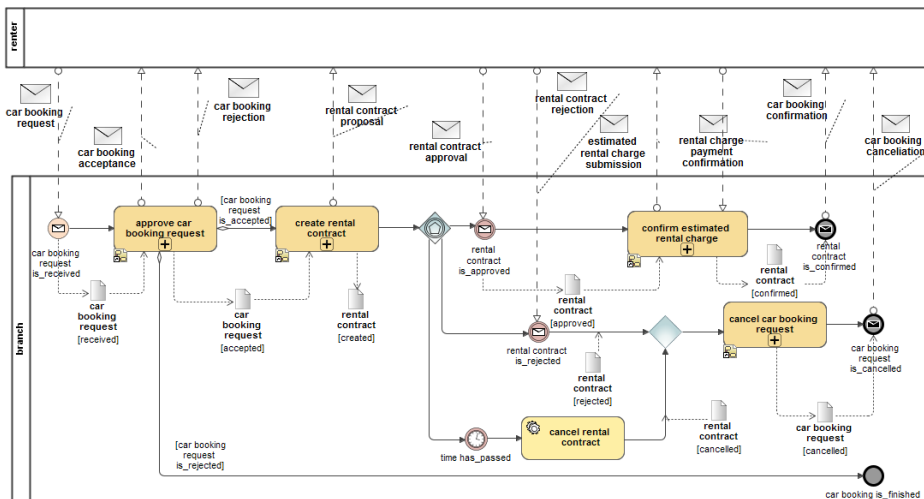
Noriu padėkoti savo mokslinio darbo vadovui prof. dr. Rimantui Butleriui už kantrybę ir pasitikėjimą sprendžiant mokslines užduotis. Taip pat dėkoju prof. dr. Linai Nemuraitei, buvusiai bakalauro bei magistro darbo vadovei, už jos kantrybę, optimizmą bei patarimus disertacijos tema, kurie niekada neleido išklysti iš kelio.

Dėkoju visiems Kauno technologijos universiteto kolegoms iš Informacinių sistemų katedros už pasidalytas žinias, patirtį, vertingas pastabas bei patarimus, kurie leido patobulinti daktaro disertacijos rašomąjį darbą. Dėkoju visiems kitiems, kurie prisidėjo savo naudingais patarimais bei įžvalgomis.

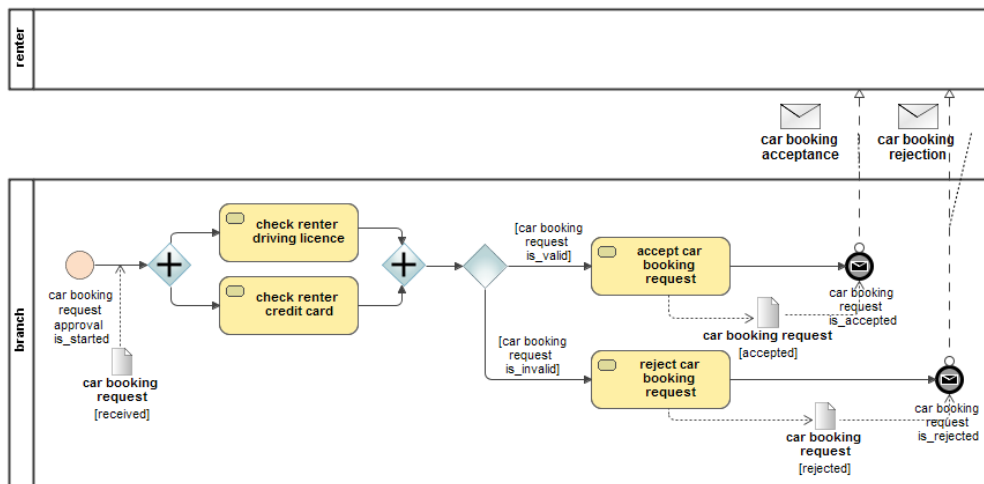
Ypatingą ačiū tariau savo šeimai bei draugams už nuolatinį palaikymą ir tikėjimą manimi šiame ilgame tyrimų kelyje.

## 10 PRIEDAI

### 10.1 BPMN VPM transformacijos į SBVR VŽ&VT taisyklės



10.1 pav. BPMN2 procesas „Book car“



10.2 pav. BPMN2 procesas „Approve car booking request“



10.3 pav. BPMN2 konteinerių (angl. Pool|Lane) pavyzdys

## Daiktavardiniai konceptai (angl. *General concepts*)

**Step 1.** SBVR daiktavardinių konceptų išgavimas iš BPMN2 veiklos procesų modelio elementų.

### BPMN įvykis (angl. *Event*) → SBVR *General Concept* (10.1 pav.)

$T_1: \text{transform}(\text{BPM}, \text{event: Event}) \xrightarrow{a^*} \text{SBVR General Concept}_1, \text{SBVR General Concept}_2$

pvz.:  $\text{transform}(\text{BPM}, \text{'car booking request is\_received'}) \xrightarrow{a^*} \text{car booking request, received}$

< sąlyga > Laiko įvykis (angl. *TimerEvent*) daiktavardinio koncepto dalis prasideda iš mažosios raidės.

$T_2: \text{transform}(\text{BPM}, \text{event: Event}) \xrightarrow{a^*} \text{SBVR General Concept}_1, \text{SBVR General Concept}_2$

pvz.:  $\text{transform}(\text{BPM}, \text{'week has\_passed'}) \xrightarrow{a^*} \text{time duration, week}$

Concept type: [time duration](#)

< sąlyga > Laiko įvykis (angl. *TimerEvent*) daiktavardinio koncepto dalis prasideda iš didžiosios raidės arba skaičiumi.

$T_3: \text{transform}(\text{BPM}, \text{event: Event}) \xrightarrow{a^*} \text{SBVR General Concept}, \text{SBVR Individual Concept}$

pvz.:  $\text{transform}(\text{BPM}, \text{'Week has\_passed'}) \xrightarrow{a^*} \text{time duration, Week}$

General concept: [time duration](#)

### BPMN pranešimas (angl. *Message*) → SBVR *General Concept* (10.1 pav.)

$T_4: \text{transform}(\text{BPM}, \text{message: Message}) \xrightarrow{a} \text{SBVR General Concept}$

pvz.:  $\text{transform}(\text{BPM}, \text{'rental contract proposal'}) \xrightarrow{a} \text{rental contract proposal}$

### BPMN konteineris (angl. *Pool*) → SBVR *General Concept* (10.3 pav.)

$T_5: \text{transform}(\text{BPM}, \text{pool: Pool}) \xrightarrow{a} \text{SBVR General Concept}$

pvz.:  $\text{transform}(\text{BPM}, \text{'operating company'}) \xrightarrow{a} \text{operating company}$

< sąlyga > Konteineris vaizduojamas kaip resursas, vardai nesutampa.

$T_6: \text{transform}(\text{BPM}, \text{lane: Lane}, \text{resource: Resource}) \xrightarrow{a} \text{SBVR General Concept}, \text{SBVR Role}$

pvz.:  $\text{transform}(\text{BPM}, \text{'branch'}, \text{'organization'}) \xrightarrow{a} \text{organization,}$

## branch

General concept: organization

**BPMN konteineris (angl. Lane) → SBVR General Concept (10.3 pav)**

$T_7: transform(BPM, lane: Lane) \xrightarrow{a} SBVR\ General\ Concept$

pvz.:  $transform(BPM, 'branch') \xrightarrow{a} \text{branch}$

*< sąlyga > Konteineris vaizduojamas kaip resursas, vardai nesutampa. Ta pati transformacijos taisyklė, kaip ir konteinerio (angl. Pool) atveju.*

**BPMN duomenų objektas (angl. DataObject) → SBVR General Concept (10.2 pav.)**

$T_8: transform(BPM, data\_object: DataObject/DataObject(withState)) \xrightarrow{a} SBVR\ General\ Concept_1, SBVR\ General\ Concept_2, SBVR\ General\ Concepts_3$

pvz.:  $transform(BPM, 'car\ booking\ request\ received') \xrightarrow{a} \text{car\_booking\_request, received car\_booking\_request, received}$

*Esant duomenų objektui (su būseną) (angl. DataObject(with State)) yra gaunami 3 konceptai, jei duomenų objektas neturi būsenos - tada 1.*

**BPMN duomenų saugykla (angl. DataStore) → SBVR General Concept (10.18 pav.)**

$T_9: transform(BPM, data\_store: DataStore) \xrightarrow{a} SBVR\ General\ Concept$

pvz.:  $transform(BPM, 'customer\ data\ store') \xrightarrow{a} \text{customer\_data\_store}$

**Veiksmažodiniai konceptai (angl. Verb concepts)**

**Step 2.** SBVR veiksmažodinių konceptų išgavimas iš BPMN2 veiklos procesų modelio elementų.

**BPMN įvykis (angl. Event) → SBVR Verb Concept (10.1 pav.)**

$T_{10}: transform(BPM, event: Event) \xrightarrow{a^*} SBVR\ Verb\ Concept_1, SBVR\ Verb\ Concept_2$

pvz.:  $transform(BPM, 'car\ booking\ request\ is\_received') \xrightarrow{a^*} \text{car\_booking\_request is\_received, car\_booking\_request has\_state received}$

**BPMN veikla (angl. Activity) → SBVR Verb Concept (10.1 pav.)**

$T_{11}: transform(BPM, pool/lane: Pool/Lane, activity: Activity) \xrightarrow{a^*} SBVR\ Verb\ Concept$

pvz.:  $transform(BPM, 'branch', 'approve\ car\ booking\ request') \xrightarrow{a^*} \text{branch approve car\_booking\_request}$

**BPMN pranešimo srautas (angl. MessageFlow) → SBVR Verb Concept**

$T_{12}: transform(BPM, pool/lane_1: Pool/Lane_1, message\_flow(pool/lane_1, pool/lane_2):$

*MessageFlow, pool|lane2: Pool|Lane*)  $\xrightarrow{a}$  SBVR Verb Concept

pvz.: *transform(BPM, 'branch', message\_flow('branch', 'renter', 'renter'))*  $\xrightarrow{a}$  [renter receive\\_message\\_from branch](#)

**BPMN pranešimo srautas su pranešimu (angl. *MessageFlow (with ref. Message)*)**  
→ **SBVR Verb Concept** (10.1 pav.)

$T_{13}$ : *transform(BPM, pool|lane1: Pool|Lane1, message\_flow(pool|lane1, pool|lane2): MessageFlow, message: Message, pool|lane2: Pool|Lane)*  $\xrightarrow{a}$  SBVR Verb Concept<sub>1</sub>, SBVR Verb Concept<sub>2</sub>

pvz.: *transform(BPM, 'branch', message\_flow('branch', 'renter'), 'rental contract proposal', 'renter')*  $\xrightarrow{a}$   
[branch send rental contract proposal](#)  
[renter receive rental contract proposal](#)

**BPMN konteineris (angl. *Lane*)** → **SBVR Verb Concept** (10.3 pav.)

$T_{14}$ : *transform(BPM, lane: Lane, pool: Pool)*  $\xrightarrow{a}$  SBVR Verb Concept

pvz.: *transform (BPM, 'branch', 'operating company')*  $\xrightarrow{a}$  [branch is\\_at operating company](#)

**BPMN duomenų objektas (su būseną) (angl. *DataObject (with State)*)** → **SBVR Verb Concept** (10.2 pav.) (Jeigu nėra susijusio įvykio, kuris būtų transformuotas į tokį patį veiksmąžodinį konceptą)

< sąlyga > nėra atitinkamo *catchEvent* prieš *DataObject* arba nėra *DataObject* prieš atitinkamą *throwEvent*.

$T_{15}$ : *transform(BPM, data\_object: DataObject)*  $\xrightarrow{a}$  SBVR Verb Concept

pvz.: *transform(BPM, 'car booking request received')*  $\xrightarrow{a}$  [car booking request has\\_state received](#)

< sąlyga > Duomenų objektas (angl. *DataObject*) turi įeinantį asociacijos ryšį iš įvykio ir įvykis turi įeinantį pranešimo srautą su pranešimu (angl. *Message flow(with ref. message)*)).

$T_{16}$ : *transform(BPM, data\_object: DataObject|DataObject(withState), event: Event, association(event, data\_object): Association, pool|lane: Pool|Lane, message\_flow(pool|lane, event): MessageFlow, message: Message)*  $\xrightarrow{a}$  SBVR Verb Concept

pvz.: *transform(BPM, 'car booking request', 'car booking request is\_received', association('car booking request is\_received', 'car booking request'), 'renter', message\_flow('renter', 'car booking request is\_received'), 'car booking request')*  $\xrightarrow{a}$   
[car booking request include received car booking request](#)

< sąlyga > Duomenų objektas (angl. *DataObject*) turi išeinantį asociacijos ryšį į įvykį ir įvykis turi išeinantį pranešimo srautą su pranešimu (angl. *Message flow(with ref. message)*)).



T<sub>17</sub>: *transform*(BPM, *data\_object*: *DataObject*/*DataObject*(withState), *event*: *Event*, *data\_association*(*event*, *data\_object*): *DataAssociation*, *pool/lane*: *Pool*/*Lane*, *message\_flow*(*pool/lane*, *event*): *MessageFlow*, *message*: *Message*) <sup>a</sup> → *SBVR Verb Concept*

pvz.: *transform*(BPM, 'rental contract', 'rental contract is\_confirmed', *data\_association*('rental contract is\_confirmed', 'rental contract'), 'renter', *message\_flow*('renter', 'rental contract is\_confirmed'), 'car booking confirmation')  
<sup>a</sup>  
→

[car\\_booking\\_confirmation include confirmed rental cintract](#)

**BPMN duomenų saugykla (angl. *DataStore*), duomenų objektas (angl. *DataObject*) → SBVR Verb Concept**

<*sąlyga*> Duomenų saugykla (angl. *DataStore*) turi išeinantį ryšį į duomenų objektą (angl. *DataObject*).

T<sub>18</sub>: *transform*(BPM, *data\_store*:*DataStore*, *data\_association*(*data\_store*, *data\_object*): *DataAssociation*, *data\_object*: *DataObject*/*DataObject*(withState)) <sup>a</sup> → *SBVR Verb Concept*

pvz.: *transform*(BPM, 'customer data store', 'dataAssociation', 'customer data') <sup>a</sup> →  
[customer\\_data\\_store provide\\_data\\_input customer\\_data](#)

Tokia pati konstrukcija yra naudojama duomenų objekto (angl. *DataObject*/*DataObject*(with state)) su kitais BPMN elementais. Jei naudojama duomenų saugykla (angl. *DataStore*), tada [provide\\_data\\_input](#) keičiamas į [provide\\_data\\_store\\_input](#)

**BPMN duomenų saugykla (angl. *DataStore*), duomenų objektas (angl. *DataObject*) → SBVR Verb Concept**

<*sąlyga*> Duomenų saugykla (angl. *DataStore*) turi įeinantį ryšį iš duomenų objekto (angl. *DataObject*).

T<sub>19</sub>: *transform*(BPM, *data\_object*: *DataObject*/*DataObject*(withState), *data\_association*(*data\_object*, *data\_store*):*DataAssociation*, *data\_store*:*DataStore*)  
<sup>a</sup>  
→ *SBVR Verb Concept*

pvz.: *transform*(BPM, 'customer data', 'dataAssociation', 'customer data store') <sup>a</sup> →  
[customer\\_data\\_store get\\_data\\_output customer\\_data](#)

Tokia pati konstrukcija yra naudojama duomenų objekto (angl. *DataObject*/*DataObject*(with state)) su kitais BPMN elementais. Jei naudojama duomenų saugykla (angl. *DataStore*), tada [get\\_data\\_output](#) keičiamas į [get\\_data\\_store\\_output](#)

**Veiklos taisyklės (angl. *Business Rules*)**

Veiklos taisyklės, kurios gaunamos iš BPMN2 veiklos procesų modelio, yra skirstomos į dvi grupes pagal BPMN2 elemento vaidmenį:

- 1) taisyklės objektas (pvz., BPMN2 įvykis *event*<sub>1</sub> taisyklėje T<sub>20</sub>);

2) taisyklės iniciatorius (pvz., BPMN2 įvykis  $event_2$  taisyklėje  $T_{20}$ ).

**Step 3.** SBVR veiklos taisyklių išgavimas iš BPMN2 veiklos procesų modelio, kai taisyklės objektas arba rezultatas yra įvykis.

**Step 3.1.** SBVR veiklos taisyklių išgavimas, kai įvykis inicijuoja kito įvykio ar veiklos pasirodymą.

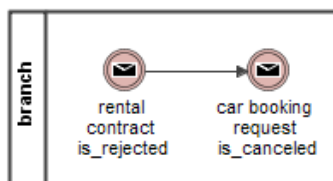
Šios transformacijos tikslas – sudaryti SBVR veiklos taisyklę tokiu atveju, kai įvykis ar veikla eina po kito įvykio (kuris gali turėti ir duomenų objektą (su būseną)) ir kai juos jungia srauto ryšys.

**Vienas įvykis (angl.  $Event_1$ ) inicijuoja kitą įvykį (angl.  $Event_2$ )** (10.4 pav.)

$T_{20}$ :  $transform(BPM, event_2: Event_2, sequence\_flow(event_2, event_1): SequenceFlow, event_1: Event_1) \xrightarrow{a} SBVR\ Business\ Rule$

pvz.:  $transform(BPM, 'rental\ contract\ is\_rejected', sequence\_flow('rental\ contract\ is\_rejected', 'rental\ contract\ is\_canceled'), 'rental\ contract\ is\_canceled') \xrightarrow{a}$

It is obligatory that car booking request is canceled after rental contract is rejected



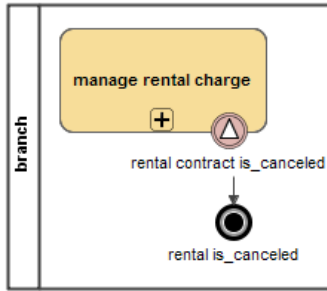
**10.4 pav.** Pavyzdys, kaip vienas įvykis inicijuoja kitą

< sąlyga > Ribinis įvykis (angl. Boundary event) inicijuoja įvykį (10.5 pav.)

$T_{21}$ :  $transform(BPM, pool/lane: Pool/Lane, subprocess: Subprocess, event_1: Event_1, sequence\_flow(event_1, event_2): SequenceFlow, event_2: Event_2) \xrightarrow{a} SBVR\ Business\ Rule$

pvz.:  $transform(BPM, 'branch', 'manage\ rental\ charge', 'rental\ contract\ is\_canceled', sequence\_flow('rental\ contract\ is\_canceled', 'rental\ is\_canceled'), 'rental\ is\_canceled') \xrightarrow{a} SBVR\ Business\ Rule$

It is obligatory that rental is canceled if rental contract is canceled while branch manage rental charge



10.5 pav. Pavyzdys, kai ribinis įvykis inicijuoja įvykį

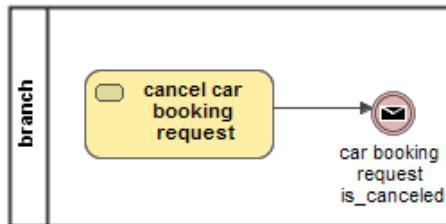
Šios transformacijos tikslas – sudaryti SBVR veiklos taisyklę tokiu atveju, kai įvykis eina po veiklos ir kai juos jungia srauto ryšys.

**Veikla (angl. Activity) inicijuoja įvykį (angl. Event) (10.6 pav.)**

$T_{22}: transform(BPM, pool/lane: Pool/Lane, activity: Activity, sequence\_flow(activity, event): SequenceFlow, event: Event) \xrightarrow{a} SBVR \text{ Business Rule}$

pvz.:  $transform(BPM, 'branch', 'cancel car booking request', sequence\_flow('cancel car booking request', 'car booking request is_canceled'), 'car booking request is_canceled') \xrightarrow{a}$

It is obligatory that car booking request is\_canceled after branch cancel car booking request



10.6 pav. Pavyzdys, kai veikla inicijuoja įvykį

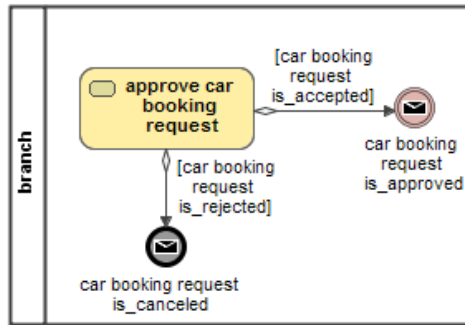
**< sąlyga >** Sekos srautas (angl. Sequence flow) turi sąlygą (3 taisyklės) (10.7 pav.)

$T_{23}: transform(BPM, pool/lane: Pool/Lane, activity: Activity, (conditional\_sequence\_flow(activity, condition_1: Condition_1, event_1): ConditionalSequenceFlow, event_1: Event_1)) \mid (conditional\_sequence\_flow(activity, condition_2: Condition_2, event_2): ConditionalSequenceFlow, event_2: Event_2)) \xrightarrow{a} SBVR \text{ Business Rule}$

pvz.:  $transform(BPM, 'branch', 'approve car booking request', (sequence\_flow('approve car booking request', 'car booking request is_accepted', 'car booking request is_approved'), 'car booking request is_approved') \mid (sequence\_flow('approve car booking request', 'car booking request is_rejected', 'car booking request is_canceled'), 'car booking request is_canceled')) \xrightarrow{a}$

It is obligatory that car booking request is\_approved or car booking request is\_canceled after branch approve car booking request

It is obligatory that car booking request is\_approved if car booking request is\_accepted  
 It is obligatory that car booking request is\_canceled if car booking request is\_rejected



10.7 pav. Pavyzdys, kai veikla su sąlyginiu sekos srautu inicijuoja įvyki

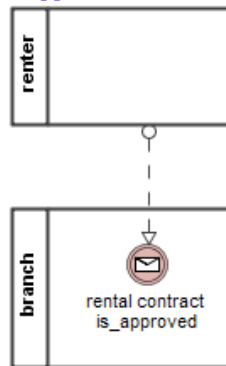
**Step 3.2.** SBVR veiklos taisyklės išgavimas, kai įvykį inicijuoja pranešimo srautas (angl. *Message Flow*).

Šios transformacijos tikslas – sudaryti SBVR veiklos taisyklę, kai įvykį inicijuoja pranešimo srautas.

**Pranešimo srautas (angl. *MessageFlow*) inicijuoja įvykį (angl. *Catch Event*)** (10.8 pav.)

$T_{24}: transform(BPM, pool/lane_1: Pool/Lane_1, message\_flow(pool/lane_1, event): MessageFlow, pool/lane_2: Pool/Lane_2, event: Event) \xrightarrow{a} SBVR \text{ Business Rule}$   
 pvz.:  $transform(BPM, 'renter', message\_flow('renter', 'branch', 'branch', 'rental contract is\_approved')) \xrightarrow{a}$

It is obligatory that rental contract is\_approved after branch receive\_message\_from renter



10.8 pav. Pavyzdys, kai pranešimo srautas inicijuoja įvykį (angl. *Catch Event*)

**Step 3.3.** SBVR veiklos taisyklės išgavimas, kai įvykį inicijuoja pranešimo srautas su pranešimu (*Message Flow with ref. Message*).

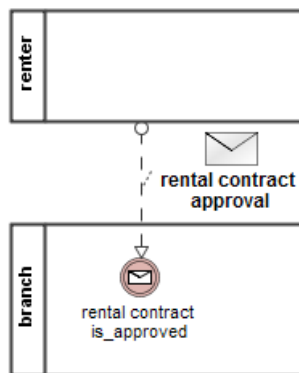
**Pranešimo srautas su pranešimu (angl. *MessageFlow (with ref. Message)*) inicijuoja įvykį (angl. *Catch Event*)** (2 taisyklės) (10.9 pav.)

$T_{25}: transform(BPM, pool/lane_1: Pool/Lane_1, message\_flow(pool/lane_1, message:Message, event): MessageFlow, pool/lane_2: Pool/Lane_2, event: Event) \xrightarrow{a} SBVR Business Rule$

pvz.:  $transform(BPM, 'renter', message\_flow('renter, 'rental contract approval', 'branch'), 'branch', 'rental contract is\_approved') \xrightarrow{a}$

It is obligatory that rental contract is\\_approved after branch receive rental contract approval

It is obligatory that branch receive rental contract approval after renter send rental contract approval



**10.9 pav.** Pavyzdys, kai pranešimo srautas inicijuoja įvykį (angl. *Catch Event*)

**Step 3.4.** SBVR veiklos taisyklės išgavimas, kai įvykį inicijuoja sprendimo priėmimo taškas, pagrįstas įvykiais (angl. *EventBasedGateway*).

Šios transformacijos tinklas – sudaryti SBVR veiklos taisyklę, kai įvykius inicijuoja įvykiais pagrįstas sprendimų priėmimų taškas, kuris eina po veiklos. Visi elementai sujungti sekos srautu.

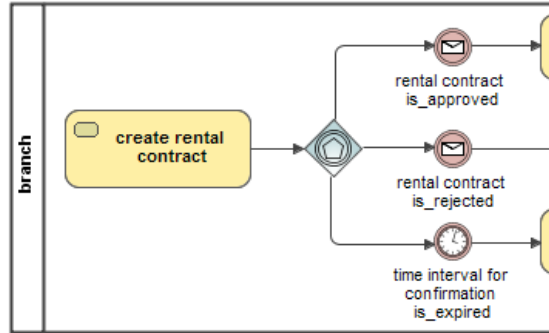
**Veikla (angl. *Activity*) inicijuoja įvykiais pagrįstą sprendimų priėmimų tašką (angl. *EventBasedGateway*) su išeinančiais sekos srautais (angl. *SequenceFlow*) ir įvykiais (angl. *Event*)** (10.10 pav.)

$T_{26}: transform(BPM, pool/lane: Pool/Lane, activity: Activity, sequence\_flow(activity, event\_based\_gateway): SequenceFlow, event\_based\_gateway: EventBasedGateway, ((sequence\_flow(event\_based\_gateway, event_1):SequenceFlow, event_1: Event_1) + (sequence\_flow(event\_based\_gateway, event_2):SequenceFlow, event_2: Event_2) + (sequence\_flow(event\_based\_gateway, event_3):SequenceFlow, event_3: Event_3))) \xrightarrow{a} SBVR Business Rule$

pvz.:  $transform(BPM, 'branch', 'create rental contract', sequence\_flow('create rental contract', EventBasedGateway), EventBasedGateway, (sequence\_flow(EventBasedGateway, 'rental contract is\_approved'), 'rental contract is\_approved' + sequence\_flow(EventBasedGateway, 'rental contract is\_rejected'), 'rental contract is\_rejected' + sequence\_flow(EventBasedGateway,$

'time interval for confirmation is\_expired'), 'time interval for confirmation is\_expired')<sup>a</sup> →

It is obligatory that rental contract is\_approved or rental contract is\_rejected or confirmation time interval is\_expired after branch create rental contract



**10.10 pav.** Pavyzdys, kai veikla inicijuoja įvykiais pagrįstą sprendimų priėmimų tašką (angl. *Event Based Gateway*), kuris turi išeinančius sekos srautus su įvykiais

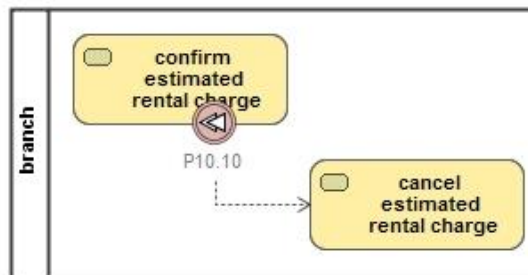
**Step 3.5.** Transformacijos tikslas – sudaryti SBVR veiklos taisyklę, kai yra taikoma kompensavimo veikla.

**Veikla (angl. *Activity*) inicijuoja kompensavimo įvykį (angl. *CompensationEvent*)** (10.11 pav.)

$T_{27}$ : *transform(BPM, pool/lane: Pool/Lane, activity<sub>1</sub>: Activity<sub>1</sub>, compensation\_event: CompensationEvent, sequence\_flow(activity<sub>1</sub>, activity<sub>2</sub>): SequenceFlow, activity<sub>2</sub>: Activity<sub>2</sub>, association(compensation\_event, activity<sub>3</sub>): Association, activity<sub>3</sub>: Activity<sub>3</sub>)*  
<sup>a</sup> → SBVR Business Rule

pvz.: *transform(BPM, 'branch', 'confirm estimated rental charge', CompensationEvent, sequence\_flow('confirm estimated rental charge', 'notify renter'), 'notify renter', association(CompensationEvent, 'cancel estimated rental charge'), 'cancel estimated rental charge')*<sup>a</sup> →

It is obligatory that branch cancel estimated rental charge if compensation is\_required while branch confirm estimated rental charge



**10.11 pav.** Pavyzdys, kai veikla inicijuoja kompensuojamą veiklą

**Step 4.** SBVR veiklos taisyklės išgavimas, kai taisyklės iniciatorius ar rezultatas yra veikla.

**Step 4.1.** SBVR veiklos taisyklės išgavimas, kai veikla eina po kitos veiklos ar įvykio.

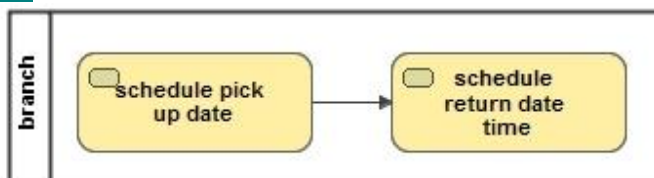
Šios transformacijos tikslas – SBVR veiklos taisyklės gavimas, kai viena veikla eina po kitos veiklos ir jas jungia sekos srautas.

**Viena veikla (angl. Activity<sub>1</sub>) inicijuoja kitą veiklą (angl. Activity<sub>2</sub>)** (10.12 pav.)

T<sub>28</sub>:  $transform(BPM, pool/lane: Pool/Lane, activity_1: Activity_1, sequence\_flow(activity_1, activity_2): SequenceFlow, activity_2: Activity_2) \xrightarrow{a} SBVR Business Rule$

pvz.:  $transform(BPM, 'branch', 'schedule pick up date time', sequence\_flow('schedule pick up date time', 'schedule return date time'), 'schedule return date time') \xrightarrow{a}$

It is obligatory that branch schedule return date time after branch schedule pick up date time



**10.12 pav.** Pavyzdys, kai veikla inicijuoja kitą veiklą

< sąlyga > Sekos srautas turi sąlygą (3 taisyklės) (10.13 pav.)

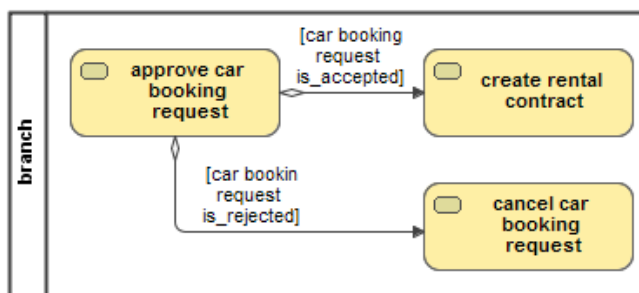
T<sub>29</sub>:  $transform(BPM, pool/lane: Pool/Lane, activity_1: Activity_1, (conditional\_sequence\_flow(activity_1, Condition_1:condition_1, activity_2): ConditionalSequenceFlow, activity_2: Activity_2)) / (conditional\_sequence\_flow(activity_1, Condition_2:condition_2, activity_3): ConditionalSequenceFlow, activity_3: Activity_3)) \xrightarrow{a} SBVR Business Rule$

pvz.:  $transform(BPM, 'branch', 'approve car booking request', (sequence\_flow('approve car booking request', 'car booking request is\_accepted', 'create rental contract'), 'create rental contract') | (sequence\_flow('approve car booking request', 'car booking request is\_rejected', 'cancel car booking request'), 'cancel car booking request')) \xrightarrow{a}$

It is obligatory that branch create rental contract or branch cancel car booking request after branch approve car booking request

It is obligatory that branch create rental contract if car booking request is\\_accepted

It is obligatory that branch cancel car booking request if car booking request is\\_rejected



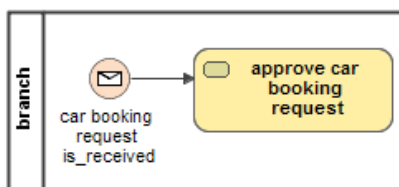
10.13 pav. Pavyzdys, kai veikla su išeinančiais sekos srautais su sąlygomis inicijuoja veiklas

**Įvykis (angl. Event) inicijuoja veiklą (angl. Activity)** (10.14 pav.)

$T_{30}: transform(BPM, pool/lane: Pool/Lane, event: Event, sequence\_flow(event, activity): SequenceFlow, activity: Activity) \xrightarrow{a} SBVR\ Business\ Rule$

pvz.:  $transform(BPM, 'branch', 'car\ booking\ request\ is\_received', sequence\_flow('car\ booking\ request\ is\_received', 'approve\ car\ booking\ request'), 'approve\ car\ booking\ request') \xrightarrow{a}$

It is obligatory that [branch approve car booking request](#) after [car booking request is received](#)



10.14 pav. Pavyzdys, kai įvykis inicijuoja veiklą

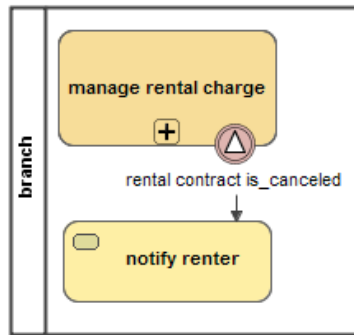
< sąlyga > Ribinis įvykis yra susietas su veikla sekos srautu (10.15 pav.)

$T_{31}: transform(BPM, pool/lane: Pool/Lane, subprocess: Subprocess, event: Event, sequence\_flow(event, activity): SequenceFlow, activity: Activity) \xrightarrow{a} SBVR\ Business\ Rule$

pvz.:  $transform(BPM, 'branch', 'manage\ rental\ charge', 'rental\ contract\ is\_canceled', sequence\_flow('rental\ contract\ is\_canceled', 'notify\ renter'), 'notify\ renter') \xrightarrow{a} SBVR\ Business\ Rule$

It is obligatory that [branch notify renter](#) if [rental contract is canceled](#) while [branch manage rental charge](#)





10.15 pav. Pavyzdys, kai ribinis įvykis susietas su veikla sekos srautu

**Step 4.2.** Transformacijos tikslas – sudaryti SBVR veiklos taisyklę, kai sprendimų priėmimo taškas (angl. *ExclusiveGateway*) turi įeinančią ir išeinančią sekos srautus su veiklomis.

**Sprendimų priėmimo taškas (*ExclusiveGateway*), kuris inicijuoja veiklą (angl. *Activity*) ir turi įeinančius sekos srautus (angl. *SequenceFlow*) su veiklomis (angl. *Activity*)** (10.16 pav.)

$T_{32}: transform(BPM, pool/lane: Pool/Lane, ((activity_2: Activity_2, sequence\_flow(activity_2, exclusive\_gateway): SequenceFlow)|( activity_3: Activity_3, sequence\_flow(activity_3, exclusive\_gateway): SequenceFlow)), exclusive\_gateway: ExclusiveGateway, sequence\_flow(exclusive\_gateway, activity_1): SequenceFlow, activity_1: Activity_1 \xrightarrow{a} SBVR Business Rule$

e.g.:  $transform(BPM, 'branch', (('accept car booking request', sequence\_flow('accept car booking request', ExclusiveGateway)|( 'reject car booking request', sequence\_flow('reject car booking request', ExclusiveGateway))), ExclusiveGateway, sequence\_flow(ExclusiveGateway, 'notify renter'), 'notify renter' \xrightarrow{a}$

It is obligatory that [branch notify renter](#) after [branch accept car booking request](#) or [branch reject car booking request](#)

**Veikla (angl. *Activity*) inicijuoja sprendimų priėmimo tašką (angl. *ExclusiveGateway*), kuris turi išeinančius sekos srautus su sąlygomis (angl. *SequenceFlow (with Condition)*) ir veiklomis (angl. *Activity*)** (3 taisyklės) (10.16 pav.)

$T_{33}: transform(BPM, pool/lane: Pool/Lane, activity_3: Activity_3, sequence\_flow(activity_3, exclusiveGateway): SequenceFlow, exclusive\_gateway: ExclusiveGateway, ((sequenceFlow(exclusiveGateway, condition_1: Condition_1, activity_1): SequenceFlow, activity_1: Activity_1) | ( sequenceFlow(exclusiveGateway, condition_2: Condition_2, activity_2): SequenceFlow, activity_2: Activity_2)) \xrightarrow{a} SBVR Business Rule$

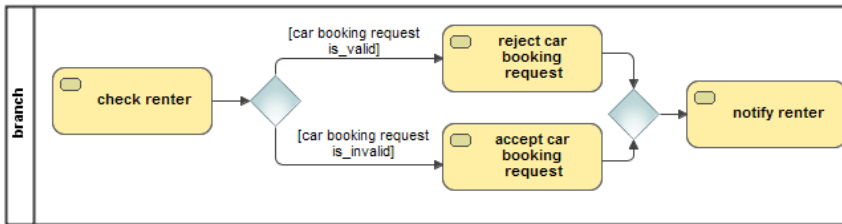
pvz.:  $transform(BPM, 'branch', 'check renter', sequence\_flow('check renter', ExclusiveGateway), ((ExclusiveGateway, (sequence\_flow(ExclusiveGateway, 'car booking request is\_valid', 'accept car booking request'), 'accept car booking$

request') | (sequence\_flow(ExclusiveGateway, 'car booking request is\_invalid', 'reject car booking request'))  $\xrightarrow{a}$

It is obligatory that [branch accept car booking request](#) or [branch reject car booking request](#) after [branch check renter](#)

It is obligatory that [branch accept car booking request](#) if [car booking request is valid](#)

It is obligatory that [branch reject car booking request](#) if [car booking request is invalid](#)



10.16 pav. Sprendimų priėmimo taško (angl. *ExclusiveGateway*) pavyzdys

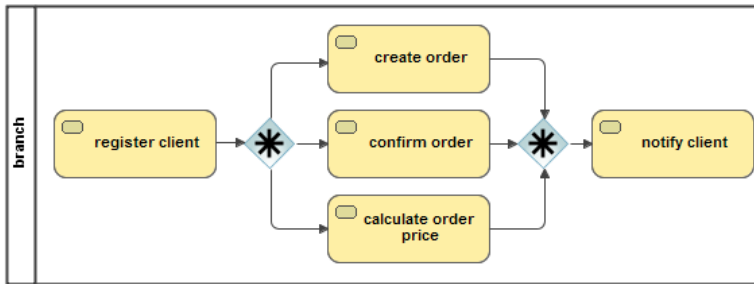
**Step 4.3.** Transformacijos tikslas – sudaryti SBVR veiklos taisyklę, kai kompleksinis sprendimų priėmimo taškas (angl. *ComplexGateway*) turi išeinančius sekos srautus su veiklomis.

**Veikla (angl. *Activity*) inicijuoja kompleksinį sprendimų priėmimo tašką (angl. *Complex Gateway*) su išeinančiais sekos srautais (angl. *SequenceFlow*) ir veiklomis (angl. *Activity*)** (10.17 pav.)

$T_{34}$ : transform(BPM, pool/lane: Pool/Lane, activity<sub>1</sub>: Activity<sub>1</sub>, sequence\_flow(activity<sub>1</sub>, complex\_gateway): SequenceFlow, complex\_gateway: ComplexGateway, ((sequence\_flow(complex\_gateway, activity<sub>2</sub>): SequenceFlow, activity<sub>2</sub>: Activity<sub>2</sub>) + (sequence\_flow(complex\_gateway, activity<sub>3</sub>): SequenceFlow, activity<sub>3</sub>: Activity<sub>3</sub>) + (sequence\_flow(complex\_gateway, activity<sub>4</sub>): SequenceFlow, activity<sub>4</sub>: Activity<sub>4</sub>)))  $\xrightarrow{a}$  SBVR Business Rule

pvz.: transform(BPM, 'branch', 'register client', sequence\_flow('register client', ComplexGateway), ComplexGateway, (sequence\_flow(ComplexGateway, 'confirm order'), 'confirm order') + (sequence\_flow(ComplexGateway, 'create order'), 'create order') + (sequence\_flow(ComplexGateway, 'calculate order price'), 'calculate order price'))  $\xrightarrow{a}$

It is obligatory that [branch create order](#) and [branch calculate order price](#) or [branch create order](#) and [branch confirm order](#) or [branch calculate order price](#) and [branch confirm order](#) if [branch register client](#)



10.17 pav. Kompleksinio sprendimų priėmimo taško (angl. *ComplexGateway*) pavyzdys

**Step 4.4.** Transformacijos tikslas – sudaryti SBVR veiklos taisyklę, kai apimantis sprendimų priėmimo taškas (angl. *InclusiveGateway*) turi įeinančius ir išeinančius sekos srautus su veiklomis.

**Apimantis sprendimų priėmimo taškas (angl. *InclusiveGateway*) su įeinančiais sekos srautais (angl. *SequenceFlow*) su veiklomis (angl. *Activity*)** (10.18 pav.)

$T_{35}$ :  $transform(BPM, pool/lane: Pool/Lane, ((activity_1: Activity_1, sequence\_flow(activity_1, inclusive\_gateway): SequenceFlow) + (activity_2: Activity_2, sequence\_flow(activity_2, inclusive\_gateway): SequenceFlow) + (activity_3: Activity_3, sequence\_flow(activity_3, inclusive\_gateway): SequenceFlow)), inclusive\_gateway: InclusiveGateway, sequence\_flow(inclusive\_gateway, activity_4): SequenceFlow, activity_4: Activity_4) \xrightarrow{a} SBVR\ Business\ Rule$

pvz.:  $transform(BPM, 'branch', (('prepare\ discount', sequence\_flow('prepare\ discount', InclusiveGateway)) + ('prepare\ offer', sequence\_flow('prepare\ offer', InclusiveGateway)) + ('prepare\ rejection', sequence\_flow('prepare\ rejection', InclusiveGateway))), InclusiveGateway, sequence\_flow(InclusiveGateway, 'notify\ renter'), 'notify\ renter') \xrightarrow{a}$

It is obligatory that [branch notify renter after branch prepare discount](#) or [branch prepare offer](#) or [branch prepare discount and branch prepare offer](#) or [branch prepare rejection](#)

**Veikla (angl. *Activity*) inicijuoja apimantį sprendimų priėmimo tašką (angl. *InclusiveGateway*) su išeinančiais sekos srautais su sąlygomis (angl. *SequenceFlow (with Condition)*) ir veiklomis (angl. *Activity*)** (3 taisyklės) (10.18 pav.)

$T_{36}$ :  $transform(BPM, pool/lane: Pool/Lane, activity_4: Activity_4, sequenceFlow(activity_4, inclusiveGateway): SequenceFlow, inclusiveGateway: InclusiveGateway, ((sequenceFlow(inclusiveGateway, activity_1): SequenceFlow, activity_1: Activity_1) + (sequenceFlow(inclusiveGateway, activity_2): SequenceFlow, activity_2: Activity_2))+ (defaultSequenceFlow(inclusiveGateway, activity_3): DefaultSequenceFlow, activity_3: Activity_3)) \rightarrow SBVR\ Business\ Rule$   
 $\xrightarrow{a} SBVR\ Business\ Rule$

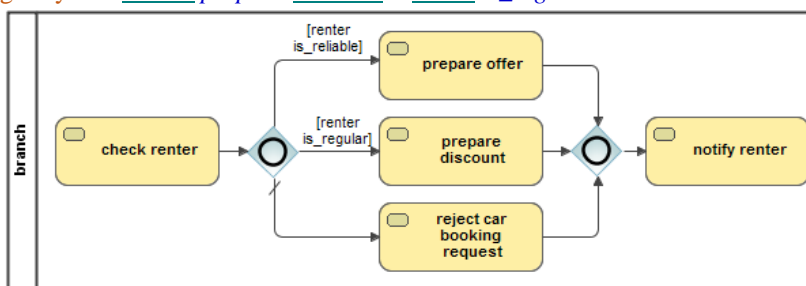
pvz.:  $transform(BPM, 'branch', 'notify\ renter', sequence\_flow('notify\ renter', InclusiveGateway), InclusiveGateway, ((sequence\_flow(InclusiveGateway, 'prepare\ offer'), 'prepare\ offer'))+( sequence\_flow(InclusiveGateway, 'prepare\ discount')$

'prepare discount'))  $\xrightarrow{a}$

It is obligatory that branch prepare offer or branch prepare discount or branch prepare offer and branch prepare discount or branch reject car booking request after branch check renter

It is obligatory that branch prepare offer if renter is\_reliable

It is obligatory that branch prepare discount if renter is\_regular



10.18 pav. Pavyzdys su apimančiu sprendimų priėmimo tašku (angl. *InclusiveGateway*)

**Step 4.5.** Transformacijos tikslas – sudaryti SBVR veiklos taisyklę, kai lygiagretusis sprendimų priėmimo taškas (angl. *ParallelGateway*) turi įeinančius ir išeinančius sekos srautus su veiklomis.

**Lygiagretusis sprendimų priėmimo taškas (angl. *ParallelGateway*) su įeinančiais sekų srautais (angl. *SequenceFlow*) ir veiklomis (angl. *Activity*) inicijuoja veiklą (angl. *Activity*)** (10.19 pav.)

$T_{37}: transform(BPM, pool/lane: Pool/Lane, ((activity_2: Activity_2, sequence\_flow(activity_2, parallel\_gateway): SequenceFlow) + (activity_3: Activity_3, sequence\_flow(activity_3, parallel\_gateway): SequenceFlow))), parallel\_gateway: ParallelGateway, sequence\_flow(parallel\_gateway, activity_1): SequenceFlow, activity_1: Activity_1) \xrightarrow{a} SBVR\ Business\ Rule$

pvz.:  $transform(BPM, 'branch', (('prepare\ offer', sequence\_flow('prepare\ offer', ParallelGateway)) + ('prepare\ discount', sequence\_flow('prepare\ discount', ParallelGateway))), ParallelGateway, sequence\_flow(ParallelGateway, 'notify\ renter'), 'notify\ renter') \xrightarrow{a}$

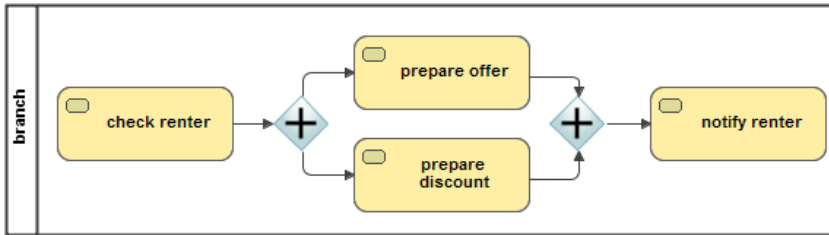
It is obligatory that branch notify renter after branch prepare offer and branch prepare discount

**Veikla (angl. *Activity*) inicijuoja lygiagretųjį sprendimų priėmimo tašką (angl. *ParallelGateway*) su išeinančiais sekų srautais (angl. *SequenceFlow*) ir veiklomis (angl. *Activity*)** (10.19 pav.)

$T_{38}: transform(BPM, pool/lane: Pool/Lane, activity_1: Activity_1, sequence\_flow(activity_1, parallel\_gateway): SequenceFlow, ((sequence\_flow(parallel\_gateway, activity_2): SequenceFlow, activity_2: Activity_2) + (sequence\_flow(parallel\_gateway, activity_3): SequenceFlow, activity_3: Activity_3))) \xrightarrow{a} SBVR\ Business\ Rule$

pvz.:  $transform(BPM, 'branch', 'check\ renter', sequence\_flow('check\ renter', ParallelGateway), ParallelGateway, ((sequence\_flow(ParallelGateway, 'prepare$

offer'), 'prepare offer') + (sequence\_flow(ParallelGateway, 'prepare discount'), 'prepare discount'))  $\xrightarrow{a}$   
 It is obligatory that branch prepare offer and branch prepare discount after branch check renter



10.19 pav. Pavyzdys su lygiagrečiuoju sprendimų priėmimo tašku (angl. *ParallelGateway*)

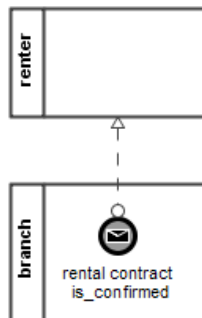
**Step 5.** SBVR veiklos taisyklių išgavimas, kai taisyklės iniciatorius ar rezultatas yra pranešimų srautas.

**Step 5.1.** Transformacijos tikslas – sudaryti SBVR veiklos taisyklę, kai pranešimų srautas (angl. *MessageFlow*) eina prieš įvykį (angl. *ThrowEvent*).

**Įvykis (angl. *Throw Event*) inicijuoja pranešimo srautą (angl. *MessageFlow*)** (10.20 pav.)

$T_{39}: transform(BPM, pool/lane_1: Pool/Lane_1, event: Event, message\_flow(event, pool/lane_2): MessageFlow, pool/lane_2: Pool/Lane_2) \xrightarrow{a} SBVR\ Business\ Rule$   
 pvz.:  $transform(BPM, 'branch', 'rental\ contract\ is\ confirmed', message\_flow('branch', 'renter'), 'renter') \xrightarrow{a}$

It is obligatory that renter receive message from branch after rental contract is confirmed



10.20 pav. Pavyzdys, kai įvykis (angl. *ThrowEvent*) inicijuoja pranešimo srautą (angl. *MessageFlow*)

**Step 5.2.** Transformacijos tikslas – sudaryti SBVR veiklos taisyklę, kai pranešimo srautas su pranešimu (angl. *MessageFlow (with ref. Message)*) eina prieš įvykį (angl. *ThrowEvent*).

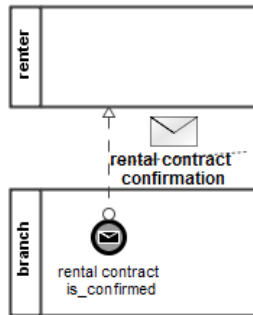
**Įvykis (angl. *Throw Event*) inicijuoja pranešimo srautą su pranešimu (angl. *MessageFlow(with ref. Message)*)** (2 taisyklės) (10.21 pav.)

T<sub>40</sub>:  $transform(BPM, pool/lane_1: Pool/Lane_1, event: Event, message\_flow(event, message: Message, pool/lane_2): MessageFlow, pool/lane_2: Pool/Lane_2) \xrightarrow{a} SBVR$   
*Business Rule*

pvz.:  $transform(BPM, 'branch', 'rental\ contract\ is\_confirmed', message\_flow('branch', 'rental\ contract\ confirmation', 'renter'), 'renter') \xrightarrow{a}$

It is obligatory that branch send rental contract confirmation after rental contract is\_confirmed

It is obligatory that renter receive rental contract confirmation after branch send rental contract confirmation



**10.21 pav.** Pavyzdys, kai įvykis (angl. *ThrowEvent*) inicijuoja pranešimo srautą su pranešimu (angl. *MessageFlow (with ref. Message)*)

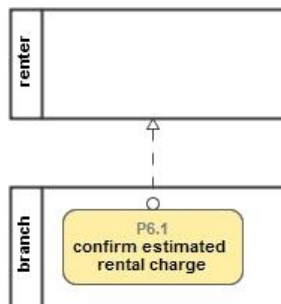
**Step 5.3.** Transformacijos tikslas – sudaryti SBVR veiklos taisyklę, kai pranešimo srautas (angl. *MessageFlow*) išeina iš veiklos (angl. *Activity*).

**Veikla (angl. *Activity*) inicijuoja išeinantį pranešimo srautą (angl. *MessageFlow*)** (10.22 pav.)

T<sub>41</sub>:  $transform(BPM, pool/lane_1: Pool/Lane_1, activity: Activity, message\_flow(activity, pool/lane_2): MessageFlow, pool/lane_2: Pool/Lane_2) \xrightarrow{a} SBVR$   
*Business Rule*

pvz.:  $transform(BPM, 'branch', message\_flow('confirm\ estimated\ rental\ charge', 'renter'), 'renter') \xrightarrow{a}$

It is obligatory that renter receive message from branch when branch confirm estimated rental charge



**10.22 pav.** Pavyzdys, kai veikla (angl. *Activity*) inicijuoja išeinantį pranešimo srautą (angl. *MessageFlow*)

**Step 5.4.** Transformacijos tikslas – sudaryti SBVR veiklos taisyklę, kai pranešimo srautas su pranešimu (angl. *MessageFlow (with ref. Message)*) išeina iš veiklos (angl. *Activity*).

**Veikla (angl. *Activity*) inicijuoja išeinantį pranešimo srautą su pranešimu (angl. *MessageFlow (with ref. Message)*)** (2 taisyklės) (10.23 pav.)

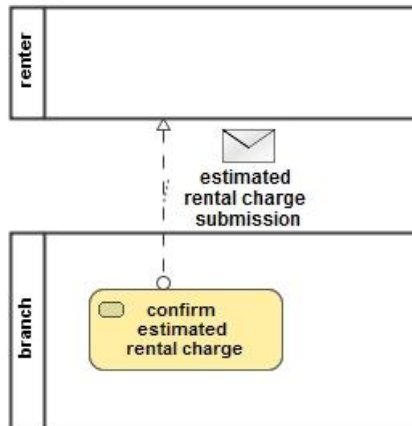
$T_{42}: \text{transform}(\text{BPM}, \text{pool}/\text{lane}_1: \text{Pool}/\text{Lane}_1, \text{activity}: \text{Activity}, \text{message\_flow}(\text{activity}, \text{message}: \text{Message}, \text{pool}/\text{lane}_2): \text{MessageFlow}, \text{pool}/\text{lane}_2: \text{Pool}/\text{Lane}_2) \xrightarrow{a} \text{SBVR Business Rule}$

pvz.:  $\text{transform}(\text{BPM}, \text{'branch'}, \text{message\_flow}(\text{'confirm estimated rental charge'}, \text{'estimated rental charge submission'}, \text{'renter'}, \text{'renter'}) \xrightarrow{a}$

$\text{'estimated rental charge submission'}, \text{'renter'}, \text{'renter'}) \xrightarrow{a}$

It is obligatory that branch send estimated rental charge submission when branch confirm estimated rental charge

It is obligatory that renter receive estimated rental charge submission after branch send estimated rental charge submission



**10.23 pav.** Pavyzdys, kai veikla (angl. *Activity*) inicijuoja išeinantį pranešimo srautą su pranešimu (angl. *MessageFlow (with ref. Message)*)

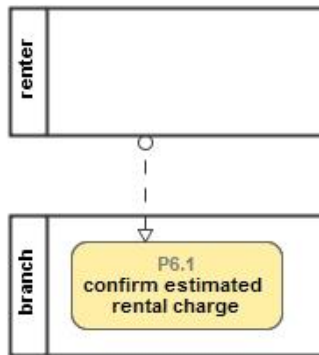
**Step 5.5.** Transformacijos tikslas – sudaryti SBVR veiklos taisyklę, kai veikla (angl. *Activity*) turi įeinantį pranešimo srautą (angl. *MessageFlow*).

**Įeinantis pranešimo srautas (angl. *MessageFlow*) pasirodo vykdant veiklą (angl. *Activity*)** (10.24 pav.)

$T_{43}: \text{transform}(\text{BPM}, \text{pool}/\text{lane}_1: \text{Pool}/\text{Lane}_1, \text{activity}: \text{Activity}, \text{message\_flow}(\text{pool}/\text{lane}_2, \text{activity}): \text{MessageFlow}, \text{pool}/\text{lane}_2: \text{Pool}/\text{Lane}_2) \xrightarrow{a} \text{SBVR Business Rule}$

pvz.:  $\text{transform}(\text{BPM}, \text{'renter'}, \text{message\_flow}(\text{'renter'}, \text{'confirm estimated rental charge'}, \text{'confirm estimated rental charge'}, \text{'branch'}) \xrightarrow{a}$

$\text{'confirm estimated rental charge'}, \text{'branch'}) \xrightarrow{a}$   
It is obligatory that branch receive message from renter when branch confirm estimated rental charge



10.24 pav. Pavyzdys, kai įeinantis pranešimų srautas (angl. *MessageFlow*) pasirodo vykdamt veiklą (angl. *Activity*)

**Step 5.6.** Transformacijos tikslas – sudaryti SBVR veiklos taisyklę, kai veikla (angl. *Activity*) turi įeinantį pranešimo srautą su pranešimu (angl. *MessageFlow (with ref. Message)*)).

Įeinantis pranešimo srautas su pranešimu (angl. *MessageFlow (with ref. Message)*) pasirodo vykdamt veiklą (angl. *Activity*) (3 taisyklės) (10.25 pav.)

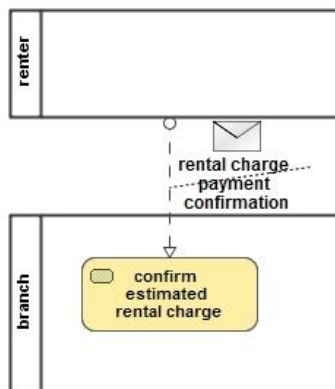
$T_{44}$ :  $transform(BPM, pool/lane_1: Pool/Lane_1, activity: Activity, message\_flow(pool/lane_2, message:Message, activity):MessageFlow, pool/lane_2: Pool/Lane_2) \xrightarrow{a} SBVR\ Business\ Rule$

pvz.:  $transform(BPM, 'renter', message\_flow('renter', 'rental\ charge\ payment\ confirmation', 'confirm\ estimated\ rental\ charge'), 'confirm\ estimated\ rental\ charge', 'branch') \xrightarrow{a}$

It is obligatory that renter send rental charge payment confirmation

It is obligatory that branch receive rental charge payment confirmation after renter send rental charge payment confirmation

It is obligatory that branch receive rental charge payment confirmation when branch confirm estimated rental charge



10.25 pav. Pavyzdys, kai įeinantis pranešimų srautas su pranešimu (angl. *MessageFlow (with ref. Message)*) pasirodo vykdamt veiklą (angl. *Activity*)

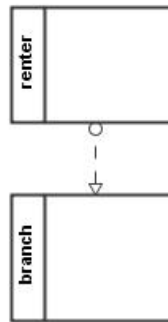


**Step 5.7.** Transformacijos tikslas – sudaryti SBVR veiklos taisyklę, kai du konteinerius (angl. *Pool|Lane*) jungia pranešimo srautas (angl. *MessageFlow*).

**Pranešimo srautas (angl. *MessageFlow*) tarp dviejų konteinerių (angl. *Pool|Lane*)** (10.26 pav.)

$T_{45}: transform(BPM, pool|lane_1: Pool|Lane_1, message\_flow(pool|lane_2, pool|lane_1): MessageFlow, pool|lane_2: Pool|Lane_2) \xrightarrow{a} SBVR \text{ Business Rule}$

pvz.:  $transform(BPM, 'renter', message\_flow('renter', 'branch'), 'branch') \xrightarrow{a}$   
 It is obligatory that [branch receive message from renter](#)



**10.26 pav.** Pavyzdys, kai pranešimo srautas (angl. *MessageFlow*) jungia du konteinerius (angl. *Pool|Lane*)

**Step 5.8.** Transformacijos tikslas – sudaryti SBVR veiklos taisyklę, kai du konteinerius (angl. *Pool|Lane*) jungia pranešimo srautas su pranešimu (angl. *MessageFlow (with ref. Message)*).

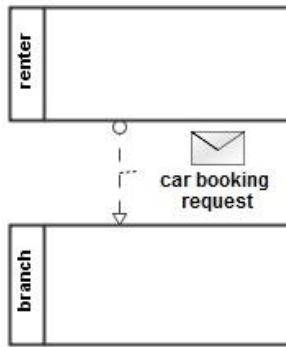
**Pranešimo srautas su pranešimu (angl. *MessageFlow (with ref. Message)*) tarp dviejų konteinerių (angl. *Pool|Lane*)** (2 taisyklės) (10.27 pav.)

$T_{46}: transform(BPM, pool|lane_1: Pool|Lane_1, message\_flow(pool|lane_2, message: Message, pool|lane_1): MessageFlow, pool|lane_2: Pool|Lane_2) \xrightarrow{a} SBVR \text{ Business Rule}$

pvz.:  $transform(BPM, 'renter', message\_flow('renter', 'car booking request', 'branch'), 'branch') \xrightarrow{a}$

It is obligatory that [renter send car booking request](#)

It is obligatory that [branch receive car booking request after renter send car booking request](#)



**10.27 pav.** Pavyzdys, kai pranešimo srautas su pranešimu (angl. *MessageFlow (with ref. Message)*) jungia du konteinerius (angl. *Pool/Lane*)

## 10.2 BPMN2 konceptų metažodynas

[process](#)

[subprocess](#)

[participant](#)

[event](#)

[activity](#)

[message](#)

[data object](#)

[data store](#)

[pool](#)

Concept Type: [participant](#)

[lane](#)

Concept Type: [participant](#)

[task](#)

Concept Type: [activity](#)

[user task](#)

Concept Type: [activity](#)

[service task](#)

Concept Type: [activity](#)

[send task](#)

Concept Type: [activity](#)

[manual task](#)

Concept Type: [activity](#)

[business rule task](#)

Concept Type: [activity](#)

[script task](#)

Concept Type: [activity](#)

[event subprocess](#)

Concept Type: [subprocess](#)

[transaction subprocess](#)

Concept Type: [subprocess](#)

[ad hoc subprocess](#)

Concept Type: [subprocess](#)

[call activity](#)

Concept Type: [activity](#)

[start event](#)

Concept Type: [event](#)

[none start event](#)

Concept Type: [start event](#)

[message start event](#)

Concept Type: [start event](#)

[timer start event](#)

Concept Type: [start event](#)

[error start event](#)

Concept Type: [start event](#)

[escalation start event](#)

Concept Type: [start event](#)

[compensation start event](#)

Concept Type: [start event](#)

[signal start event](#)

Concept Type: [start event](#)

[multiple start event](#)

Concept Type: [start event](#)

[Parallel multiple start event](#)

Concept Type: [start event](#)

[intermediate catch event](#)

Concept Type: [event](#)

[none intermediate event](#)

Concept Type: [intermediate catch event](#)

[message catch intermediate event](#)

Concept Type: [intermediate catch event](#)

[timer catch intermediate event](#)

Concept Type: [intermediate catch event](#)

[conditional catch intermediate event](#)

Concept Type: [intermediate catch event](#)

[link catch intermediate event](#)

Concept Type: [intermediate catch event](#)

[signal catch intermediate event](#)

Concept Type: [intermediate catch event](#)

[multiple catch intermediate event](#)

Concept Type: [intermediate catch event](#)

[parallel multiple catch intermediate event](#)

Concept Type: [intermediate catch event](#)

[intermediate throw event](#)

Concept Type: [event](#)

[message throw intermediate event](#)

Concept Type: [intermediate throw event](#)

[escalation throw intermediate event](#)

Concept Type: [intermediate throw event](#)

[compensation throw intermediate event](#)

Concept Type: [intermediate throw event](#)

[link throw intermediate event](#)

Concept Type: [intermediate throw event](#)

[signal throw intermediate event](#)

Concept Type: [intermediate throw event](#)

[multiple throw intermediate event](#)

Concept Type: [intermediate throw event](#)

[end event](#)

Concept Type: [event](#)

[none end event](#)

Concept Type: [end event](#)

[message end event](#)

Concept Type: [end event](#)

[error end event](#)

Concept Type: [end event](#)

[cancel end event](#)

Concept Type: [end event](#)

[compensation end event](#)

Concept Type: [end event](#)

[signal end event](#)

Concept Type: [end event](#)

[terminate end event](#)

Concept Type: [end event](#)

[multiple end event](#)

Concept Type: [end event](#)

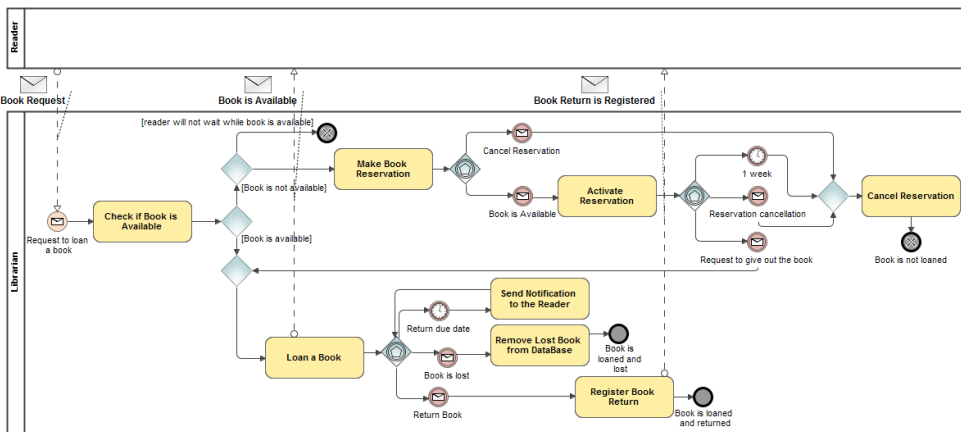
### 10.3 Eksperimente naudoti BPMN2 VPM

Šiame skyriuje pateikiami BPMN2 VPM, kurie buvo naudoti šio darbo eksperimente.

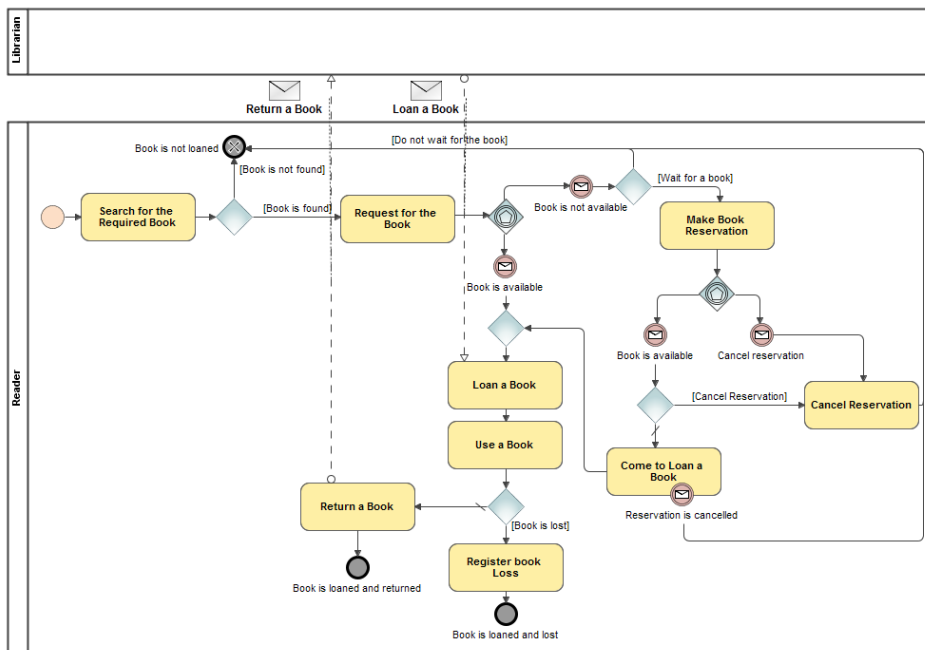
#### 10.3.1 BPMN2 VPM, kurti netaikant gerosios modeliavimo praktikos

Šiame skyriuje pateikiami dviejų skirtingų dalykinių sričių BPMN2 VPM, kuriuos kuriant nebuvo taikomos gerosios modeliavimo praktikos.

Bibliotekos BPMN2 VPM pateiktas 10.28 ir 10.29 paveiksluose.

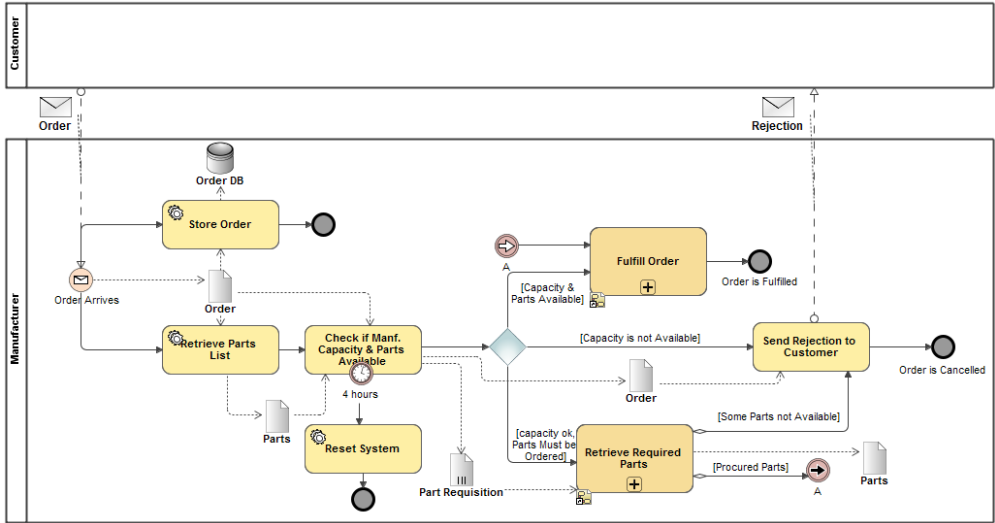


10.28 pav. Proceso „Provide Book Loan“ modelis

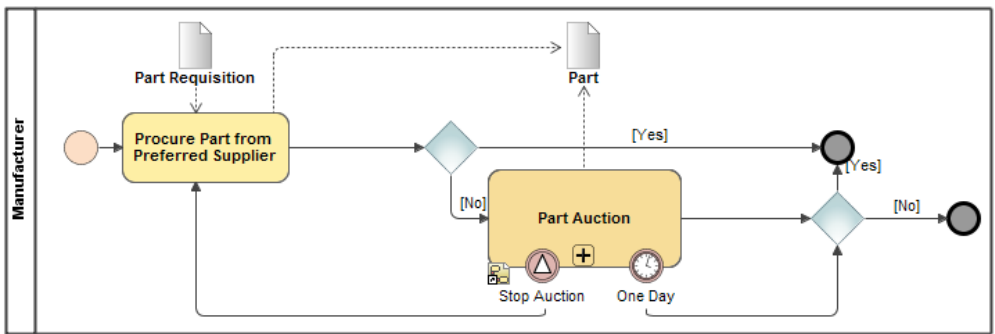


10.29 pav. Proceso „Loan a Book“ modelis

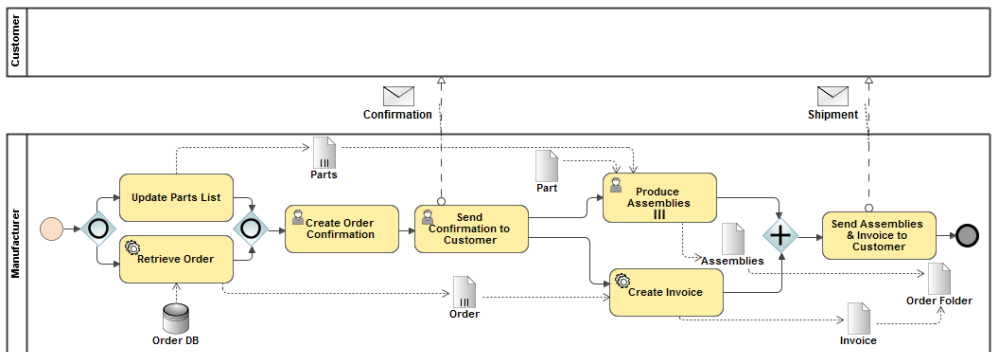
Užsakymų vykdymo BPMN2 VPM pateiktas 10.30–10.34 paveiksluose.



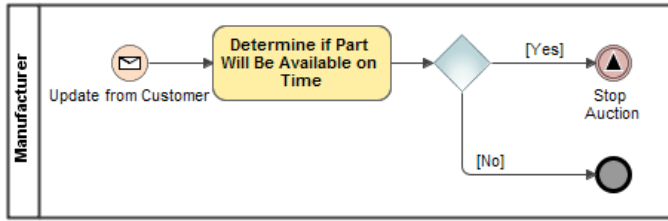
10.30 pav. Proceso „Order Handling“ modelis



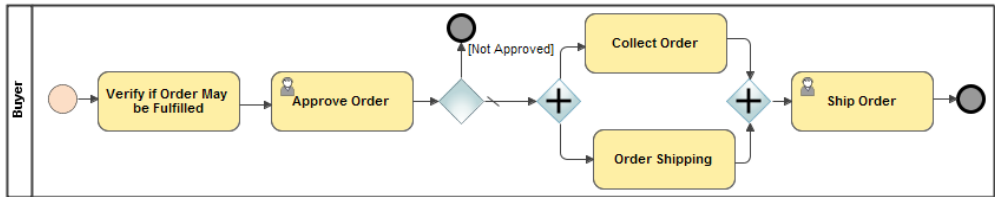
10.31 pav. Subproceso „Retrieve Required Parts“ modelis



10.32 pav. Subproceso „Fulfill Order“ modelis



10.33 pav. Subprocesas „Part Auction“ modelis

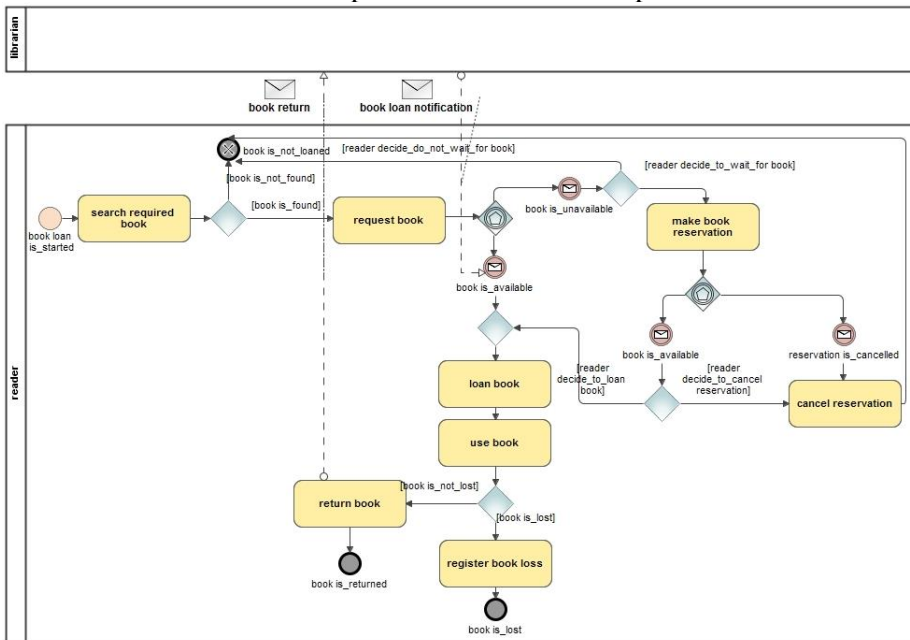


10.34 pav. Procesas „Procurement Process“ modelis

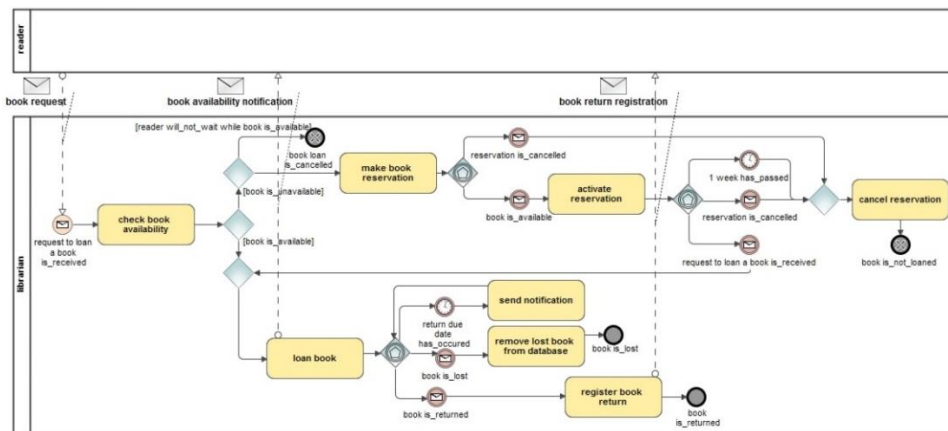
### 10.3.2 BPMN2 VPM, kuriai taikant gerąją modeliavimo praktiką

Šiame skyriuje pateikiami trijų skirtingų dalykinių sričių BPMN2 VPM, kuriuos kuriant nebuvo taikomos gerosios modeliavimo praktikos.

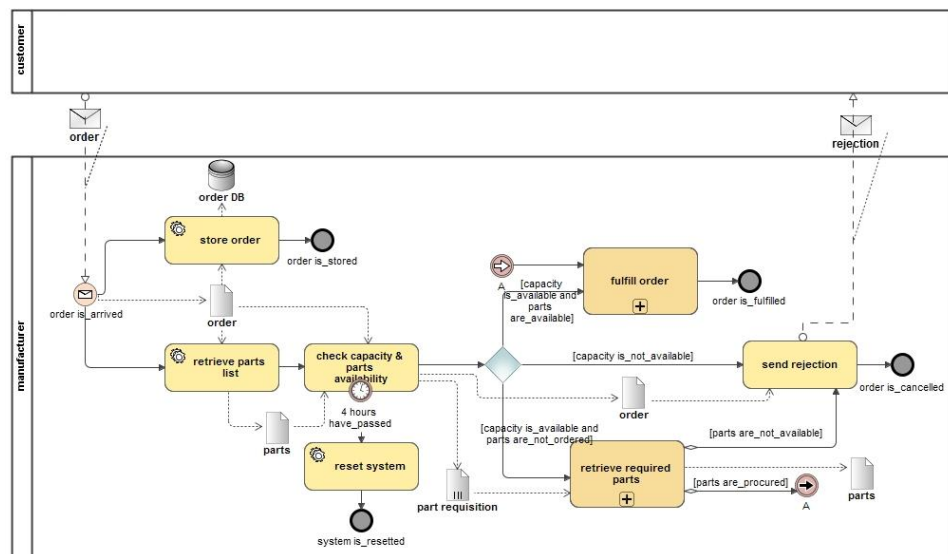
Bibliotekos BPMN2 VPM pateiktas 10.35 ir 10.36 paveiksluose.



10.35 pav. Pagal gerąją praktiką pataisytas veiklos proceso „Loan a Book“ modelis

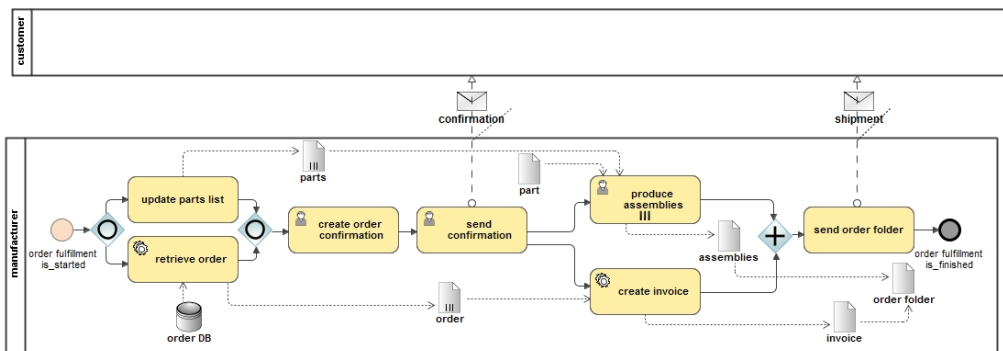


10.36 pav. Pagal gerąją praktiką pataisytas veiklos proceso „Provide Book Loan“ modelis  
Užsakymų vykdymo BPMN2 VPM pateiktas 10.37–10.41 paveiksluose.

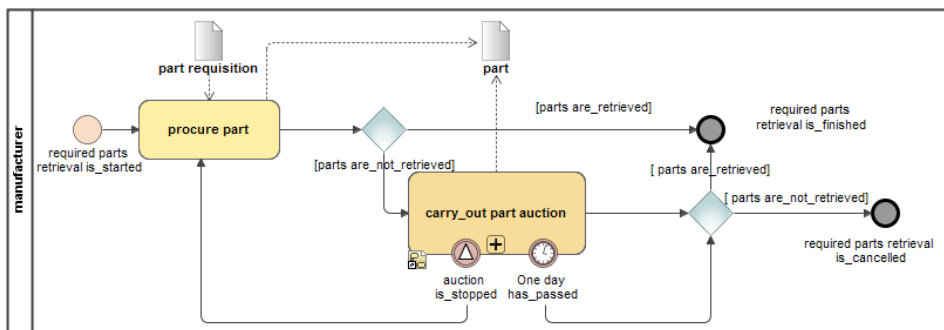


10.37 pav. Pagal gerąją praktiką pataisytas veiklos proceso „Handle Order“ modelis

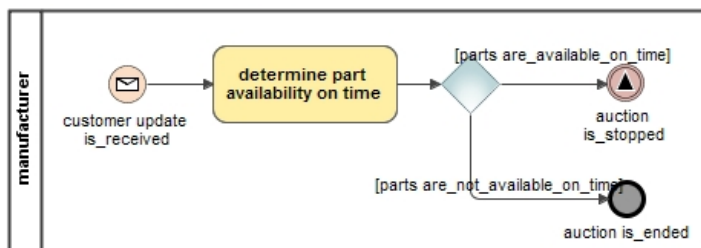




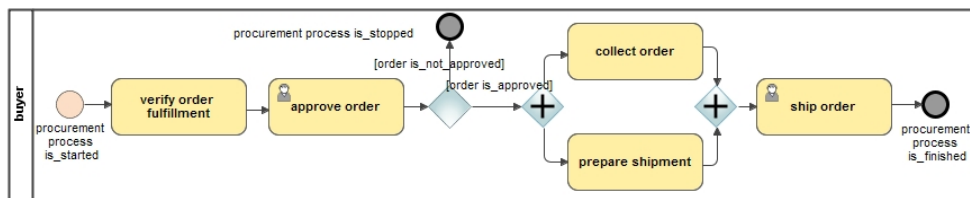
10.38 pav. Pagal gerąją praktiką pataisytas subprocesso „Fulfill Order“ modelis



10.39 pav. Pagal gerąją praktiką pataisytas subprocesso „Retrieve Required Parts“ modelis

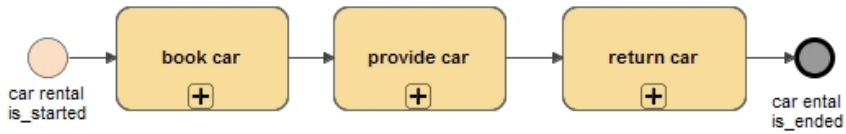


10.40 pav. Pagal gerąją praktiką pataisytas subprocesso „carry\_out part auction“ modelis

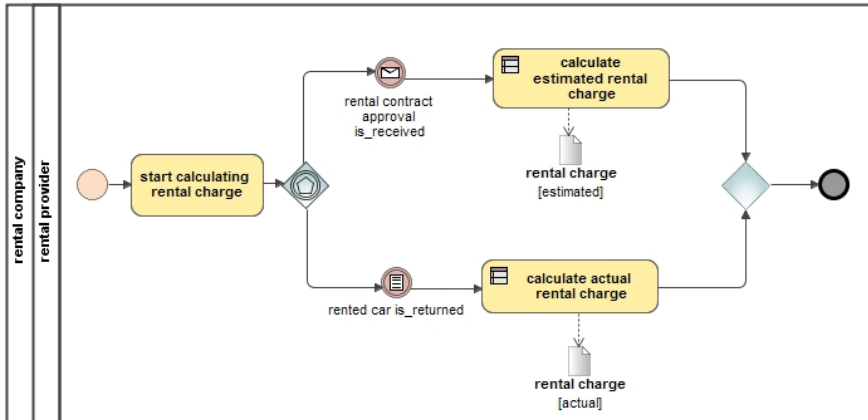


10.41 pav. Pagal gerąją praktiką pataisytas veiklos proceso „Procure Order“ modelis

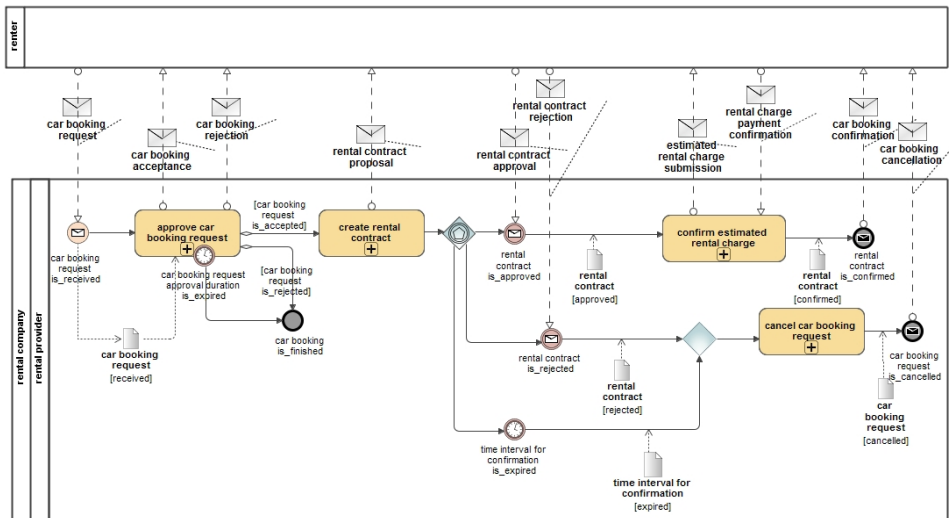
EU-Rent (automobilių nuomos) BPMN2 VPM pateiktas 10.42–10.53 paveiksluose.



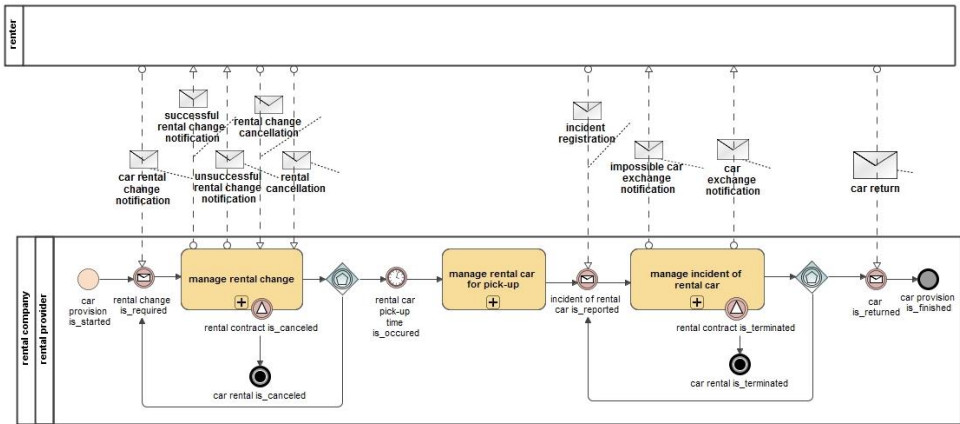
10.42 pav. EU-Rent proceso „rent car“ (iššinuomoti automobili) BPMN2 modelis (pirmasis hierarchijos lygis)



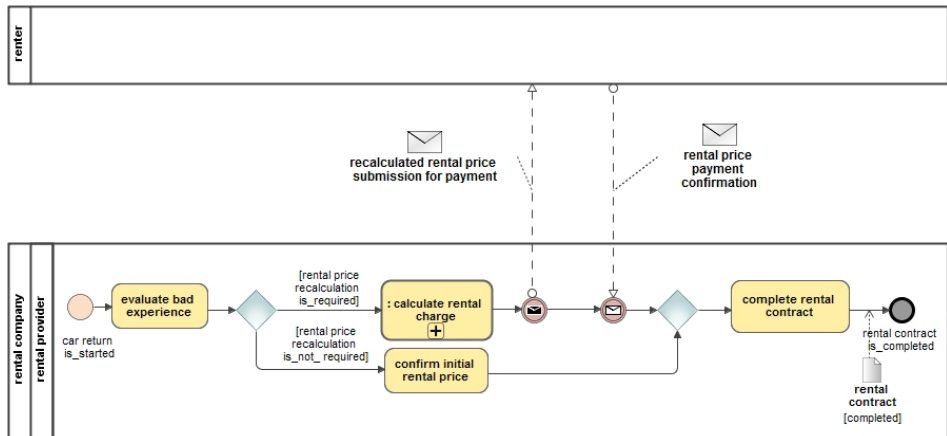
10.43 pav. EU-Rent proceso „calculate rental charge“ (apskaičiuoti nuomos kainą) BPMN2 modelis (pirmasis hierarchijos lygis)



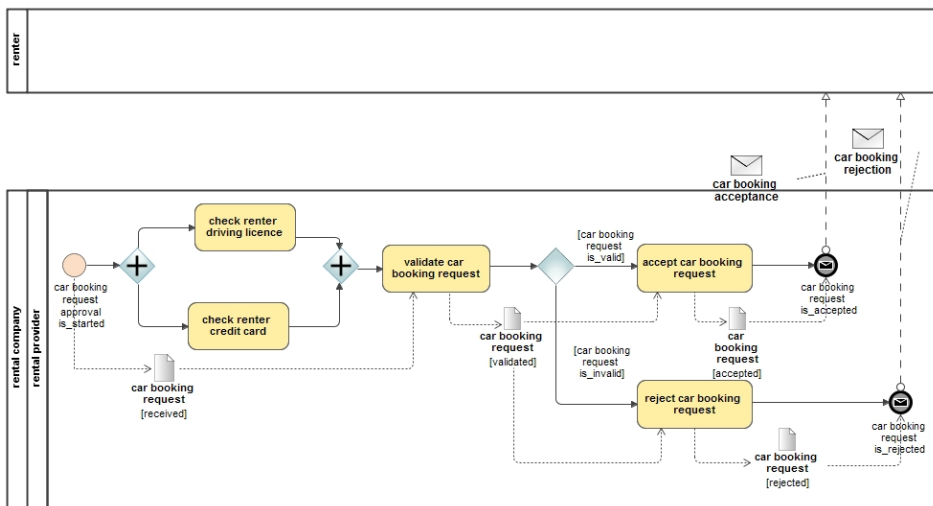
10.44 pav. EU-Rent subproceso „book car“ (rezervuoti automobili) BPMN2 modelis (antrasis hierarchijos lygis)



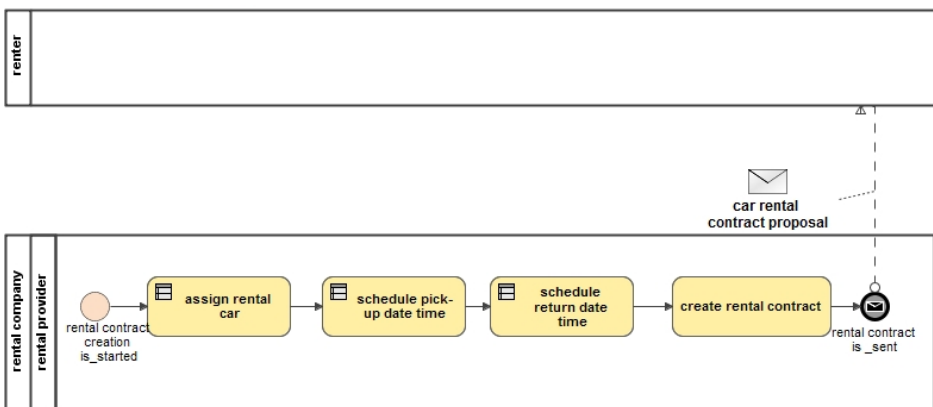
10.45 pav. EU-Rent subprocesso „provide car“ (pateikti automobilį) BPMN2 modelis (antrasis hierarchijos lygis)



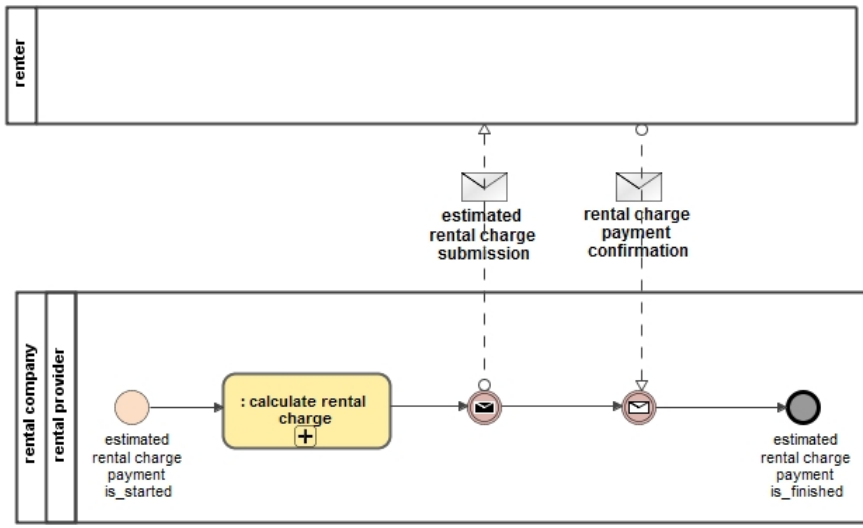
10.46 pav. EU-Rent subprocesso „return car“ (grąžinti automobilį) BPMN2 modelis (antrasis hierarchijos lygis)



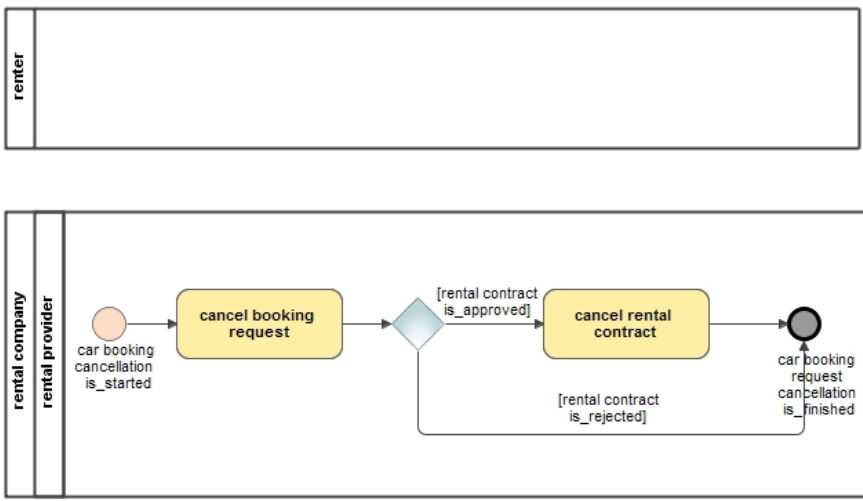
10.47 pav. EU-Rent subprocesso „book car“ trečiojo lygio subprocesas „approve car booking request“ (patvirtinti automobilio rezervaciją)



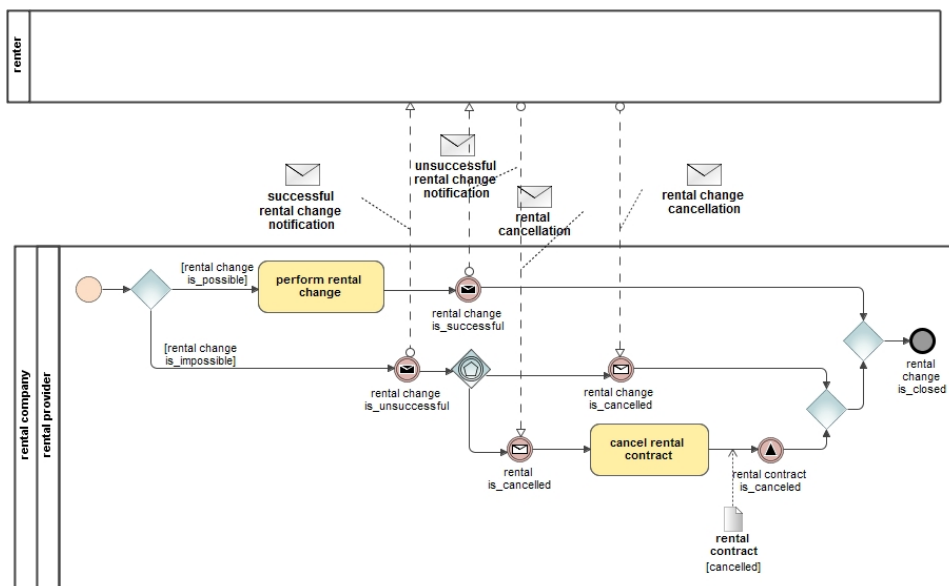
10.48 pav. EU-Rent subprocesso „book car“ trečiojo lygio subprocesas „create rental contract“ (sukurti nuomos sutartį)



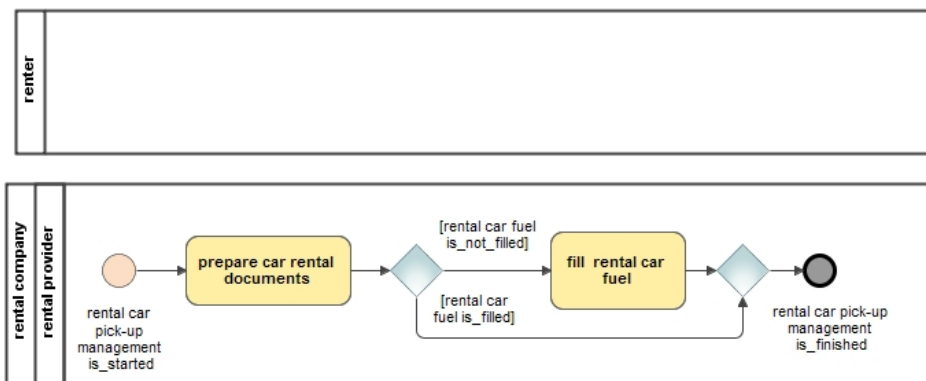
10.49 pav. EU-Rent subproceso „book car“ trečiojo lygio subprocesas „confirm estimated rental charge“ (patvirtinti apskaičiuotą nuomos mokesį)



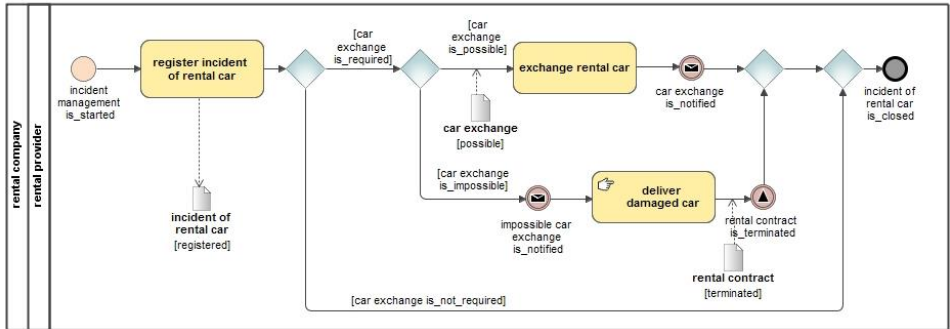
10.50 pav. EU-Rent subproceso „book car“ trečiojo lygio subprocesas „cancel car booking request“ (atšaukti automobilio rezervaciją)



10.51 pav. EU-Rent subprocesso „provide car“ trečiojo lygio subprocessas „manage rental change“ (valdyti nuomos pasikeitimus)



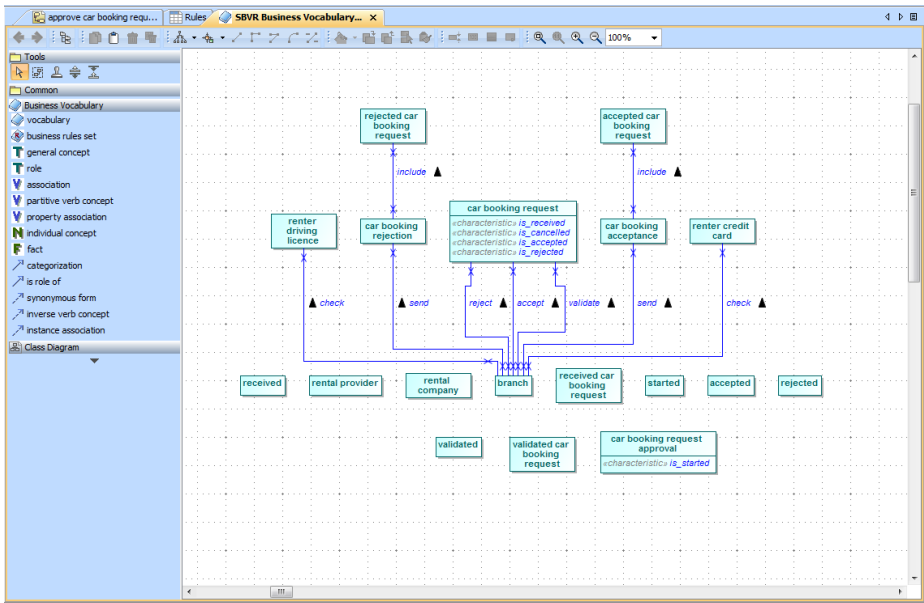
10.52 pav. EU-Rent subprocesso „provide car“ trečiojo lygio subprocessas „manage rental car for pick-up“ (paruošti išnuomotą automobilį (klientui))



10.53 pav. EU-Rent subproceso „provide car“ trečiojo lygio subprocesas „manage incident of rental car“ (tvarkyti išnuomoto automobilio incidentą)

## 10.4 MagicDraw CASE įrankio langai

Šiame skyriuje pateikiami *MagicDraw* CASE įrankio langai (10.54–10.57 pav.), kurie iliustruoja taikomą sukurtą sprendimą vienoje modeliavimo aplinkoje.

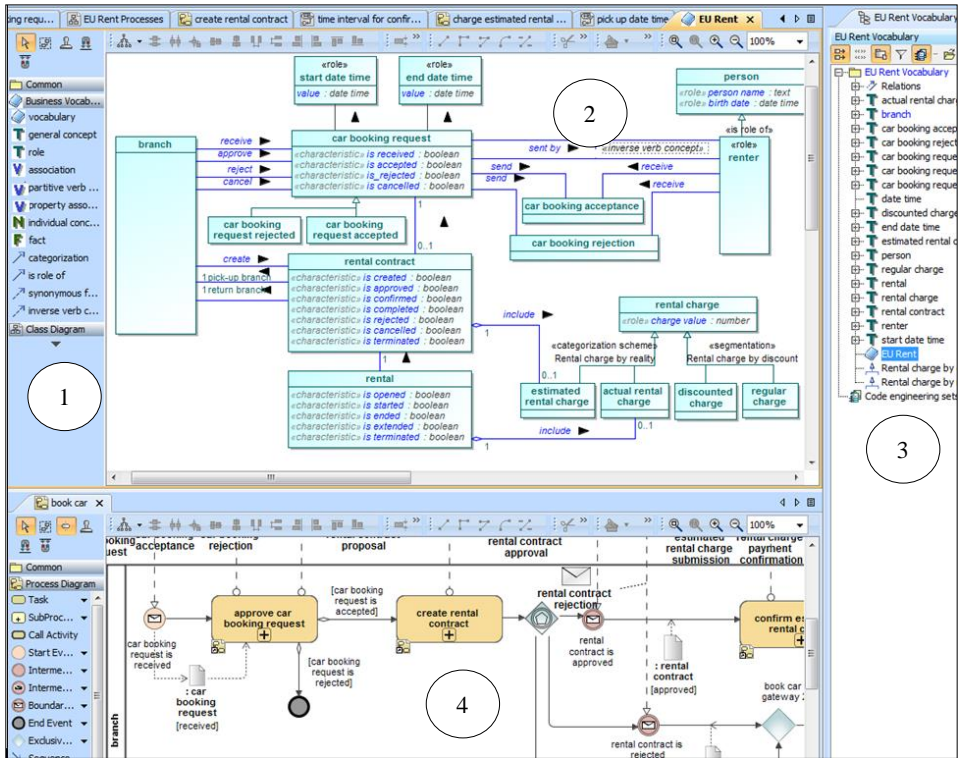


10.54 pav. SBVR žodyno diagrama

#	Name	Specification
1	R	It is obligatory that branch accept car booking request or branch reject car booking request after branch validate car booking request
2	R	It is obligatory that branch accept car booking request if car booking request is_vald
3	R	It is obligatory that branch accept car booking request or branch reject car booking request after branch validate car booking request
4	R	It is obligatory that branch check renter credit card and branch check renter driving licence after car booking request approval is_started
5	R	It is obligatory that branch send car booking rejection after car booking request is_rejected
6	R	It is obligatory that branch send car booking acceptance when branch approve car booking request
7	R	It is obligatory that car booking request is_rejected after branch reject car booking request
8	R	It is obligatory that car booking request is_accepted after branch accept car booking request
9	R	It is obligatory that client receive car booking rejection after branch send car booking rejection
10	R	It is obligatory that client receive car booking acceptance after branch send car booking acceptance

10.55 pav. SBVR veiklos taisyklės „Generic Table“ lentelėje





10.56 pav. BPMN2 veiklos proceso kūrimas naudojant SBVR veiklos žodyną

#	Business rule	Constrained Element	Condition
1	It is obligatory that branch get_data_output car_booking_request accepted if branch accept car_booking_request		
2	It is obligatory that branch create_rental_contract if car_booking_request is_accepted	Sequence Flow [BPMN models::EURent processes:	car booking request is accepted
3	It is obligatory that car_booking_request is_approved or car_booking_request is_canceled if branch approve car_booking_request		
4	It is obligatory that car_booking_request is_accepted if branch accept car_booking_request and branch provide_data_input car_booking_request_accepted		

**Condition Expression**

Edit Condition

Select language from the language list and specify body in a dedicated editor.

Language: SBVR SE

Body: car booking request is accepted

Expert

OK Cancel

10.57 pav. SBVR veiklos žodynas CASE įrankyje MagicDraw

## 10.5 BPMN2 ir SBVR naudojamų elementų vertimai

Šiame skyriuje pateikiami darbe naudojami BPMN2 elementų, jų grupių bei SBVR elementų vertimai į lietuvių kalbą iš anglų kalbos (kaip jie pateikiami standartų specifikacijose). BPMN2 elementų grupės ir jų vertimai pateikti 10.1 lentelėje.

### 10.1 lentelė. BPMN2 elementų grupės su vertimais į lietuvių kalbą

BPMN2 elementų grupė	Vertimas į lietuvių kalbą
<i>Flow Objects</i>	Srauto objektai
<i>Connecting Objects</i>	Jungiamieji objektai
<i>Swimlanes</i>	Konteineriai
<i>Resources</i>	Resursai (papildomai išskirta BPMN2 VPM transformacijų į SBVR VŽ&VT matricoje norint pažymėti SBVR rolę (angl. <i>Role</i> ))
<i>Artifacts</i>	Artefaktai

BPMN2 elementai ir jų vertimai pateikti 10.2 lentelėje.

### 10.2 lentelė. BPMN2 elementai su vertimais į lietuvių kalbą

BPMN2 elementas	Vertimas į lietuvių kalbą
<i>Event</i>	Įvykis
<i>Timer Event</i>	Laiko įvykis
<i>Activity</i>	Veikla
<i>Message</i>	Pranešimas
<i>Gateway</i>	Sprendimo priėmimo taškas
<i>ComplexGateway</i>	Kompleksinis sprendimo priėmimo taškas
<i>SequenceFlow</i>	Sekos srautas
<i>SequenceFlow (with Condition)</i>	Sekos srautas su sąlyga
<i>MessageFlow</i>	Pranešimo srautas
<i>MessageFlow (with ref. Message)</i>	Pranešimo srautas su pranešimu
<i>Association</i>	Asociacija
<i>DataAssociation</i>	Duomenų asociacija
<i>Lane</i>	Takelis
<i>Pool</i>	Baseinas
<i>Resource</i>	Resursas
<i>DataObject</i>	Duomenų objektas
<i>DataObject (with State)</i>	Duomenų objektas su būseną
<i>DataStore</i>	Duomenų saugykla
<i>TextAnnotation</i>	Teksto anotacija

SBVR elementai ir jų vertimai pateikti 10.3 lentelėje.

### 10.3 lentelė. SBVR elementai su vertimais į lietuvių kalbą

SBVR elementas	Vertimas į lietuvių kalbą
<i>General Concept</i>	Bendrinis konceptas
<i>Individual Concept</i>	Individualus konceptas
<i>Role</i>	Rolė
<i>Verb Concept</i>	Veiksmažodinis konceptas (gali būti unarinis arba binarinis)

<b>SBVR elementas</b>	<b>Vertimas į lietuvių kalbą</b>
<i>Business Rule</i>	Veiklos taisyklė

BPMN2 įvykių grupės ir jų vertimai pateikti 10.4 lentelėje.

**10.4 lentelė. BPMN2 elementai su vertimais į lietuvių kalbą**

<b>BPMN2 įvykių grupė</b>	<b>Vertimas į lietuvių kalbą</b>
<i>Start</i>	Pradžios įvykiai
<i>Intermediate Catch</i>	Vidurio laukiantys įvykiai
<i>Intermediate Throw</i>	Vidurio inicijuojantys įvykiai
<i>Boundary</i>	Ribiniai įvykiai
<i>End</i>	Pabaigos įvykiai

BPMN2 įvykiai ir jų vertimai pateikti 10.5 lentelėje.

**10.5 lentelė. BPMN2 įvykiai su vertimais į lietuvių kalbą**

<b>BPMN2 įvykis</b>	<b>Vertimas į lietuvių kalbą</b>
<i>None</i>	Pradžios įvykis
<i>Message</i>	Pranešimo įvykis
<i>Timer</i>	Laiko įvykis
<i>Error</i>	Klaidos įvykis
<i>Escalation</i>	Eskalavimo įvykis
<i>Cancel</i>	Atšaukimo įvykis
<i>Compensation</i>	Kompensavimo įvykis
<i>Conditional</i>	Sąlygos įvykis
<i>Link</i>	Jungimo įvykis
<i>Signal</i>	Signalų įvykis
<i>Terminate</i>	Nutraukimo įvykis
<i>Multiple</i>	Keli įvykiai
<i>Parallel Multiple</i>	Keli lygagretūs įvykiai

BPMN2 laiko įvykio parametrai, nusakantys laiko įvykio tipą, ir jų vertimai pateikti 10.6 lentelėje.

**10.6 lentelė. BPMN2 laiko įvykių parametrai su vertimais į lietuvių kalbą**

<b>BPMN2 laiko įvykio parametras</b>	<b>Vertimas į lietuvių kalbą</b>
<i>Time Duration</i>	Trukmė
<i>Time Cycle</i>	Ciklas
<i>Time Date</i>	Data