



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

Andrius Bumblauskas

TAISYKLĖMIS GRINDŽIAMŲ VERSLO PROCESŲ SIMULIAVIMAS
(RULES BASED BUSINESS PROCESS SIMULATION)

Baigiamasis magistro darbas

Informacinių sistemų programų inžinerijos studijų programa, valstybinis kodas 621E15002

Informatikos inžinerijos studijų kryptis

Vilnius, 2012

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

TVIRTINU
Katedros vedėjas

(Parašas)

doc. dr. D.Mažeika

(Vardas, Pavardė)

(Data)

Andrius Bumblauskas

TAISYKLĖMIS GRINDŽIAMŲ VERSLO PROCESŲ SIMULIAVIMAS
(RULES BASED BUSINESS PROCESS SIMULATION)

Baigiamasis magistro darbas

Informacinių sistemų programų inžinerijos studijų programa, valstybinis kodas 621E15002
Informatikos inžinerijos studijų kryptis

Vadovas prof. dr. Olegas Vasilecas

Lietuvių kalbos konsultantas doc. dr. Angelė Kaulakienė

Vilnius, 2012

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

TVIRTINU
Katedros vedėjas

Informatikos inžinerijos studijų kryptis

Informacinių sistemų programų inžinerijos studijų programa, valstybinis kodas 621E15002

.....specializacija

(Parašas)

(Vardas, pavardė)

(Data)

**BAIGIAMOJO MAGISTRO DARBO
UŽDUOTIS**

.....Nr.
Vilnius

Studentui (ei) ANDRIUI BUMBLAUSKUI.....
(Vardas, pavardė)

Baigiamojo darbo tema: Taisyklėmis grindžiamų verslo procesų simuliacijos.....

.....
.....

patvirtinta 201...m. d. dekanų potvarkiu Nr.

Baigiamojo darbo užbaigimo terminas 201...m. d.

BAIGIAMOJO DARBO UŽDUOTIS:

Ištirti egzistuojančių verslo procesų valdymo, verslo procesų ir taisyklių modeliavimo ir simuliacijos būdus, įvertinti integruoto verslo procesų ir taisyklių simuliacijos galimybes. Suprojektuoti ir įgyvendinti priemonę, realizuojančią taisyklėmis grindžiamų verslo procesų simuliacijos būdą

Baigiamojo darbo rengimo konsultantai:

.....
.....
.....
(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

Vadovas
(Paraša)

prof. dr. Olegas Vasilecas.....
(Moksl. laipsnis/pedag.vardas, vardas, pavardė)

Užduotį gavau

.....
(Parašas)

.....
(Vardas, pavardė)

.....
(Data)

Vilniaus Gedimino technikos universitetas
Fundamentinių mokslų fakultetas
Informacinių sistemų katedra

ISBN ISSN
Egz. sk.
Data-.....-.....

Antrosios pakopos studijų **Informacinių sistemų programų inžinerijos** programos baigiamasis darbas

Pavadinimas **Taisyklėmis grindžiamų verslo procesų simuliacija**

Autorius **Andrius Bumblauskas**

Vadovas **prof. dr. Olegas Vasilecas**

Kalba: lietuvių

Anotacija

Darbe yra nagrinėjama aktuali daugeliui šiuolaikinių įmonių verslo procesų ir taisyklių valdymo problema ir galimybė integruoti verslo taisyklių ir procesų modeliavimo ir simuliacijos metodus, laikantis mokslinės koncepcijos, teigiančios, kad verslo procesai ir verslo taisyklės yra savarankiškos sąvokos, o jų simuliacija leidžia padidinti verslo procesų modeliavimo kokybę.

Darbo tyrimo objektas yra taisyklėmis grindžiami verslo procesai.

Darbo tikslas- išplėtoti taisyklėmis grindžiamų verslo procesų simuliaciją.

Darbo uždaviniai: ištirti egzistuojančių verslo procesų valdymo, verslo procesų ir taisyklių modeliavimo ir simuliacijos būdus, įvertinti integruoto verslo procesų ir taisyklių simuliacijos galimybes; išplėtoti taisyklėmis grindžiamų verslo procesų simuliacijos būdą; suprojektuoti ir įgyvendinti priemonę, realizuojančią taisyklėmis grindžiamų verslo procesų simuliacijos būdą; atlikti taisyklėmis grindžiamo verslo proceso simuliaciją, pateikti rezultatus, bei tolimesnes plėtojimo galimybes.

Analizuojant probleminę sritį nustatyta, kad verslo procesų modeliavimas ir simuliacija yra dvi veiklos, priklausančios verslo procesų valdymo gyvavimo ciklo projektavimo fazei. Darbe pasiūlytas taisyklėmis grindžiamų verslo procesų modeliavimo būdas leidžia kurti paprastus, lengviau valdomus ir suprantamus verslo procesų modelius. Toks metodas siūlo naudoti plačiai paplitusią ir standartizuotą BPMN procesų modeliavimo kalbą bei ECA tipo taisykles, kas užtikrina lengvą metodo adaptavimą, nes nereikalauja specifinių žinių. Simuliacijos priemonių realizavimas parodė, kad panaudojant egzistuojančius verslo taisyklių ir verslo procesų vykdymo variklius bei diskrečiųjų įvykių simuliacijos karkasus ir juos sujungiant galima sukurti taisyklėmis grindžiamų verslo procesų simuliacijos priemones. Simuliuojant taisyklėmis grindžiamus verslo procesus įgyvendintos simuliacijos priemonės patvirtino jų tinkamumą ir panaudojamumą kasdieniams procesų optimizavimo klausimams spręsti.

Prasminiai žodžiai

verslo taisyklės, verslo procesai, modeliavimas, simuliacija

Vilnius Gediminas Technical University
Faculty of Fundamental Sciences
Department of Information Systems

ISBN ISSN
Copies No.
Date-.....-.....

Master Degree Studies **Information Systems Software Engineering** study programme Master's Thesis

Title **Rules based business process simulation**

Author **Andrius Bumblauskas**

Academic supervisor **Prof Dr Olegas Vasilecas**

Thesis
language: Lithuanian

Annotation

Author analyzes the actual business processes and rules management problem for many modern business companies and an opportunities to integrate business rules and process modeling and simulation methods, in accordance with the scientific concept, claiming that the business processes and business rules are independent concepts, and simulation of them allows to increase the quality of business processes modeling.

The object of finish papers is rules-based business processes.

The aim is to develop rules-based business process simulation.

The goals are: to analyze the existing modeling and simulation techniques of business process management, business processes and rules and to evaluate possibilities (capabilities) of the integrated business processes, rules simulation; to develop methods for rules-based business process simulation; to design and implement the measure realizing simulation methods for the rules-based business process; to make simulation of rules-based business processes, to present the results, and further development opportunities.

The analysis of the problematic area of the business process modeling and simulation are two activities belong to the business processes design's phase of business processes management's life-cycle. Author proposed the rules-based business processes' modeling method, that allows to create more simple, easier and more comprehensive business process' models. This approach proposes to use the widespread and standardized BPMN process modeling language and the ECA-type rules, which ensures easy adaptation of the method, because that measure does not require specific knowledge. Realization of simulation tools have shown that re usage of existing business rules and business process execution engines, discrete events' simulation frames and combining them allow to create a rules-based business process simulation tools. Implemented simulation measures during rule-based business processes simulation has confirmed their suitability and usability for everyday business process optimization issues.

Keywords

business processes, business rule, simulation, modeling

SUTRUMPINIMAI

VPV - verslo procesų valdymas

OMG - Open Management Group

BPMN - Business Process Modeling Notation

BMM - Business Motivation Metamodel

SBVR - Semantics Of Business Vocabulary And Business Rules

BPMS - Business Process Management System

BRMS - Business Rules Management System

ASL - Apache Software Licence

LGPL - Lesser General Public License

POJO - Plain Old Java Object

Paveikslėliai

1 pav. Verslo procesų valdymo gyvavimo ciklas.....	19
2 pav. Verslo procesų modeliavimas.....	21
3 pav. Verslo procesų valdymo metamodelis.....	22
4 pav. Verslo procesų valdymo sistemos bendrinė architektūra.....	34
5 pav. Verslo procesų simuliacinio sistemos architektūra.....	36
6 pav. Taisyklėmis grindžiamų verslo procesų simuliacinio priemonės architektūra.....	39
7 pav. DROOLS ir jBPM architektūra.....	41
8 pav. Supaprastinta DESMO-J klasių diagrama.....	44
9 pav. Simuliacijos konfigūracijos elementus atitinkančių klasių diagrama.....	47
10 pav. Simuliacijos konfigūracijos pavyzdys.....	48
11 pav. Simuliacijos mašinos klasių diagrama.....	50
12 pav. Procesų vykdymo simuliacijos klasės.....	50
13 pav. Veiklų vykdymo simuliacijos klasės.....	51
14 pav. Verslo logikos mašinos sąsajos.....	53
15 pav. Įvykių ir veiklų apdorojimo klasės.....	54
16 pav. Verslo logikos variklio klasių diagrama.....	54

Turinys

Įvadas.....	14
Tiriamoji problema.....	14
Darbo aktualumas.....	15
Tyrimo objektas.....	15
Darbo tikslas.....	15
Darbo uždaviniai.....	16
Darbo struktūra.....	16
1. Verslo procesų ir taisyklių modeliavimas ir simuliacijos.....	17
1.1. Verslo procesų valdymas.....	17
1.1.1. Apibrėžimas.....	17
1.1.2. Verslo procesų valdymo gyvavimo ciklas.....	18
1.1.3. Verslo modeliavimas.....	21
1.1.4. Verslo procesų valdymo metamodelis.....	22
1.1.5. Verslo proceso sąvoka.....	24
1.1.6. Verslo procesų modeliavimas.....	25
1.1.7. Verslo taisyklės sąvoka.....	26
1.1.8. Procesų ir taisyklių kontinuumas.....	27
1.2. Simuliacijos.....	28
1.2.1. Pagrindinės sąvokos.....	28
1.2.2. Diskrečių įvykių simuliacija.....	29
1.2.3. Statistika	32
1.3. Verslo procesų valdymo ir simuliacijos sistemos.....	32
1.3.1. Pagrindinės sąvokos.....	32
1.3.2. Verslo procesų valdymo sistemos architektūra.....	33
1.3.3. Verslo procesų simuliacijos sistemos architektūra.....	35
2. Taisyklėmis grindžiamų verslo procesų simuliacijos priemonė.....	37
2.1. Architektūra.....	37
2.1.1. Tikslai ir principai.....	37
2.1.2. Komponentės ir struktūra.....	38
2.2. Pasirinkta programinė įranga.....	41
2.2.1. DROOLS integruotos verslo logikos mašina.....	41
2.2.2. DESMO-J diskrečių įvykių simuliacijos variklis.....	43
2.2.3. Kūrimo priemonės.....	44
2.3. Realizacijos aprašymas.....	45

2.3.1. Projektas ir jo struktūra.....	45
2.3.2. Simuliacijos konfigūracija.....	47
2.3.3. Simuliacijos mašina.....	49
2.3.4. Verslo logikos mašina.....	52
3.Simuliacijos priemonės demonstracija.....	55
3.1. Modelis.....	55
3.1.1. Taisyklėmis grindžiamo verslo proceso modelis.....	55
3.1.2. Simuliacijos modelis.....	56
3.2. Simuliacijos rezultatai.....	56
Išvados.....	58
Priedai.....	63
Priedas Nr. 1 - Informacija apie RPSIM projektą.....	64
Priedas Nr. 2 - SimModel klasių diagrama.....	65
Priedas Nr. 3 - Veiklų simuliacijos algoritmas (JAVA kodo ištraukos).....	66
Priedas Nr. 4 - Demonstracijos proceso modelis.....	67
Priedas Nr. 5 - Demonstracijos proceso modelio taisyklės.....	68
Priedas Nr. 6 – Demonstracijos simuliacijos konfigūracija.....	69
Priedas Nr. 7 - RPSIM simuliacijos rezultatų failas	70

IVADAS

Tiriamoji problema

Neatsiejama šiuolaikinės verslo sistemos dalis yra informacinės ir programų sistemos. Šios sistemos dažniausiai yra naudojamos įmonės verslo procesų automatizavimui ir jų efektyvumo didinimui. Verslo procesai, priklausomai nuo dinamiškos verslo aplinkos pokyčių, nuolat turi būti peržiūrimi, atnaujinami ar keičiami. Tuo būdu verslo procesų valdymas tampa viena iš pagrindinių šiuolaikinės įmonės užduočių. Pastaraisiais metais vis labiau populiarėjančiomis ir adaptuojamomis paslaugomis grindžiamos architektūros (angl. SOA) idėjos, atskleidė naujas galimybes efektyviam verslo procesų valdymui. Verslo procesų analizei, modeliavimui, simuliacijai ir įgyvendinimui buvo sukurtos specialios verslo procesų modeliavimo kalbos (BPMN, BPEL) ir priemonės bei valdymo sistemos, leidžiančios procesus valdyti ir įgyvendinti atskirai nuo kitų programų sistemų.

Panašiai kaip ir su verslo procesais, pastaraisiais metais labai išaugo susidomėjimas verslo taisyklių valdymu. Įmonės, siekdamos greitai reaguoti į besikeičiančias verslo aplinkos sąlygas, pradėjo ieškoti efektyvių verslo taisyklių modeliavimo ir pritaikymo informacinėse ir programų sistemose būdų. Buvo suformuotas ir plėtotas ne vienas verslo taisyklių simuliacijos būdas (SBVR, SWRL, SRML). Kaip ir verslo procesų atveju, siekiant automatizuoti verslo taisyklių valdymą, buvo sukurtos specializuotos programų sistemos, automatizuojančios verslo taisyklių vykdymą. Tokios programų sistemos dažniausiai vadinamos verslo taisyklių mašinomis.

Lygiagrečiai tyrinėjant verslo procesų ir verslo taisyklių valdymo metodus ir jų pritaikymą informacinėse sistemose, susiformavo nuomonė, kad verslo taisyklių ir verslo procesų valdymas yra dvi priemonės, leidžiančios didinti įmonės informacinių sistemų efektyvumą ir lankstumą. Jų valdymo metodikos ir technologijos, kiekviena su savo privalumais ir trūkumais, buvo plėtojamos nepriklausomai ir palaipsniui susiformavo mokslinės idėjos, kaip išnaudoti visus verslo procesų ir verslo taisyklių modeliavimo ir realizavimo privalumus, tačiau šie metodai nebuvo integruoti tarpusavyje. Iškėlus verslo procesų ir verslo taisyklių integruotumo problemą, buvo pasiūlytos idėjos ir metodai, apibrėžiantys, kaip turi būti modeliuojamos verslo taisyklės ir verslo procesai kartu. Tačiau iki šiol nėra visiškai aišku, kaip reikėtų atlikti integruotą verslo procesų ir verslo taisyklių simuliaciją. Naudojant integruotą verslo procesų ir verslo taisyklių modeliavimą bei tokių modelių įgyvendinimą informacinėse sistemose iškyla integruoto simuliacijos problema.

Darbe nagrinėjami, verslo procesų ir taisyklių integruoto modeliavimo ir simuliacijos principai, metodai, technikos ir priemonės.

Darbo aktualumas

Darbe yra nagrinėjama verslo procesų ir taisyklių valdymo problema, kuri aktuali daugeliui šiuolaikinių įmonių. Verslo procesų tobulinimo, sąnaudų mažinimo ir analizės uždaviniams, „Gartner“ kompanijos rengiamose apklausose (Gartner 2010; Gartner 2009) IT vadovai jau keletą metų iš eilės suteikia aukščiausius prioritetus. T. Debevoise (Debevois 2007) ir M. Dumas (Dumas at al. 2005) teigia, kad verslo procesų ir verslo taisyklių modeliavimo, simuliacijos ir realizavimo priemonės yra laikomos vienu iš efektyviausių būdų didinti verslo procesų efektyvumą ir našumą. Verslo procesų ir verslo taisyklių mašinų naudojimas leidžia operatyviai reaguoti į nuolat besikeičiančią verslo aplinką ir adaptuoti įmonės informacines sistemas pagal nuolat besikeičiančius verslo poreikius.

Mokslinių tyrimų egzistavimas naujoje, bet sparčiai besiplėtojančioje verslo procesų ir taisyklių tyrinėjimo srityje taip pat išryškina šios problemos aktualumą. Kasmetinės verslo procesų ir taisyklių srities mokslinės konferencijos, knygos, moksliniai straipsniai patvirtina šios tyrinėjimų srities svarbą.

Per 15 verslo procesų ir verslo taisyklių tyrinėjimo metų, šios idėjos išgyveno keletą raidos etapų. Ankstyvuosiuose tyrinėjimų etapuose buvo formuojamos verslo procesų ir taisyklių koncepcijos, principai ir idėjos, galimybės procesus iškelti iš programų sistemų. Vėliau, kartu su paslaugomis grindžiama architektūra, tyrinėjimo objektu tapo verslo procesų ir taisyklių modeliavimo, įgyvendinimo ir automatizavimo galimybės. Šiuo metu vis daugiau dėmesio yra skiriama procesų gavybos (angl. *process mining*) ir simuliacijos metodams. Kompanija „Gartner“ savo pranešime (Gartner 2010) taip pat pabrėžia, kad artimiausiais metais bus daug dėmesio skiriama procesų lankstumo ir dinamiškumo didinimui.

Darbe nagrinėjama galimybė integruoti verslo taisyklių ir procesų modeliavimo ir simuliacijos metodus, laikantis mokslinės koncepcijos, teigiančios, kad verslo procesai ir verslo taisyklės yra savarakiškos sąvokos, o jų simuliacijos leidžia padidinti verslo procesų modeliavimo kokybę. Darbo metu bus įrodoma, kad integruotas verslo procesų ir verslo taisyklių modeliavimas ir simuliacijos yra patikimas mokslinis metodas automatizuojant verslo sistemas.

Tyrimo objektas

Darbo tyrimo objektas yra taisyklėmis grindžiami verslo procesai.

Darbo tikslas

Išplėtoti taisyklėmis grindžiamų verslo procesų simuliaciją.

Darbo uždaviniai

Siekiant iškelto tikslo yra sprendžiami šie uždaviniai:

1. Ištirti egzistuojančių verslo procesų valdymo, verslo procesų ir taisyklių modeliavimo ir simuliacijos būdus, įvertinti integruoto verslo procesų ir taisyklių simuliacijos galimybes.
2. Išplėtoti taisyklėmis grindžiamų verslo procesų simuliacijos būdą.
3. Suprojektuoti ir įgyvendinti priemonę, realizuojančią taisyklėmis grindžiamų verslo procesų simuliacijos būdą.
4. Atlikti taisyklėmis grindžiamo verslo proceso simuliaciją, pateikti rezultatus, bei tolimesnės plėtojimo galimybes.

Darbo struktūra

Darbą sudaro įvadas, trys skyriai ir išvados.

Įvade yra iškeliamą problema, parodomas darbo aktualumas, formuluojami darbo tikslas ir uždaviniai, pristatoma darbo struktūra.

Pirmame skyriuje apibrėžiamos bazinės verslo procesų valdymo, verslo procesų, verslo taisyklių, gyvavimo ciklo, modeliavimo ir simuliacijos sąvokos. Analizuojami verslo procesų ir verslo taisyklių modeliavimo ir simuliacijos metodai ir technikos. Nagrinėjama sąsaja tarp verslo taisyklių ir verslo procesų. Išgryninama taisyklėmis grindžiamų verslo procesų simuliacijos problema. Išpildant pirmąją darbo dalį naudojami literatūros šaltinių analizės, apibendrinimo, situacijų modeliavimo metodai.

Antrame skyriuje yra apibrėžiamas taisyklėmis grindžiamų verslo procesų modeliavimo ir simuliacijos metodas, jo pritaikymas uždavinių ir problemų sprendimui. Pateikiama integruoto verslo procesų, verslo taisyklių ir įvykių valdymo variklio pagrindu veikiančios simuliacijos aplinkos, kaip automatizuotos simuliacijos priemonės, realizavimo architektūra. Ruošiant šį skyrių remiamasi situacijų modeliavimo, eksperimento planavimo, analizės bei sintezės metodais, išlaikant mokslinį analitinį sisteminį požiūrį į analizuojamą problemą.

Trečiame skyriuje yra pateikiamas išsamus atlikto eksperimentinio taisyklėmis grindžiamo verslo proceso modeliavimo aprašas bei pateikiami konkretaus verslo proceso ir verslo taisyklių simuliacijos rezultatai. Šiame skyriuje yra naudojamas eksperimentinio modeliavimo, situacijos modeliavimo bei simuliacijos metodai, kurių pritaikymas išsamiai pateikiamas eksperimento apraše.

Išvadose pateikti apibendrinti darbo rezultatai ir pateiktos darbo išvados.

1. VERSLO PROCESŲ IR TAISYKLIŲ MODELIAVIMAS IR SIMULIAVIMAS

1.1. Verslo procesų valdymas

1.1.1. Apibrėžimas

Vienas pagrindinių šiuolaikinių įmonių tikslų yra veiklos efektyvumo didinimas. Tai ypač aktualu tapo ekonominės krizės laikotarpiu. Lėtėjant ekonomikai įmonės yra priverstos priimti svarbius veiklos optimizavimo sprendimus. Vienas iš galimų veiklos optimizavimo sprendimų yra tiesioginis veiklos sąnaudų mažinimas, kuris gali būti vykdomas mažinant darbuotojų atlyginimus, atleidžiant juos arba stabdant investicijas. Tokios priemonės leidžia pagerinti veiklos efektyvumo rodiklius, tačiau dažniausiai tai būna trumpalaikis teigiamas efektas, todėl ilgalaikėje perspektyvoje rezultatas tampa neigiamas ir vėl reikalinga priimti papildomus optimizavimo sprendimus. Siekiant ilgalaikio veiklos optimizavimo sprendimų efektyvumo, pasaulinėje įmonės veiklos optimizavimo praktikoje bandoma taikyti verslo vadybos metodologinę sistemą, kurios tyrimo objektu tampa verslo procesų valdymas.

Tarp IT specialistų, verslo procesų valdymas yra tapatinamas su programų sistemomis, leidžiančiomis modeliuoti ir automatizuoti verslo procesus bei kurti verslo procesais grindžiamas informacines sistemas. Analizuojant mokslinės literatūros šaltinius, nesudėtinga nustatyti, kad toks požiūris nėra visiškai teisingas. Naujų technologijų svarba valdant šiuolaikinius verslo procesus yra akivaizdi, tačiau technologijos yra tik verslo procesų valdymo dalis. Bendriausia prasme verslo procesų valdymas (toliau - VPV) - tai integruota verslo (įmonės) našumo vadybos sistema, paremta procesų visumos valdymu (Hammer 2009:4).

VPV mokslinės sampratos aiškinimų teorijos leidžia išskirti dvi pagrindines šiuolaikinio verslo procesų valdymo pagrindą sudarančias metodikas: nuolatinis verslo procesų tobulinimas (anlg. *continuous improvement*) ir verslo procesų reinžinerija (anlg. *business process reengineering*).

Nuolatinio procesų tobulinimo teorijos ištakomis yra laikomi Shewharto (Shewhart 1986) ir Demingo (Deming 1953) darbai, kurie laikui bėgant buvo išplėtoti iki tokių šiuolaikinių kokybės metodikų kaip „Six Sigma“ (šešių sigmų metodika). Kokybinio verslo procesų valdymo požiūrio pagrindinė metodologinė nuostata - nenutrūkstamas verslo procesų pagrindinių našumo parametrų nustatymas, jų stebėjimas, trūkumų identifikavimas ir šalinimas. Esminis šio metodo trūkumas yra verslo proceso sąvokos apibrėžimas ir jos naudojimas. Verslo procesų yra laikoma bet kokia veikla

ar jų seka, todėl įmonėje gali būti apibrėžti tūkstančiai pavienių procesų, tokių kaip paslaugos užsakymas ar dokumento siuntimas. Toks požiūris į procesą neleidžia matyti visumos, o atskirų procesų tobulinimas gali neduoti pageidaujamų rezultatų įmonės mastu.

Verslo procesų reinžinerijos metodologinis pagrindas yra Hammerio (Hammer 1990) bei Hammerio ir Champy (Hammer et al. 1993) darbai. Priešingai nei nuolatinio verslo procesų tobulinimo metodas, verslo procesų reinžinerijos metu procesai yra peržiūrimi iš pagrindų ir radikaliai keičiami, siekiant kardinalaus verslo procesų (efektyvumo) rodiklių pagerėjimo. Skirtingai yra apibrėžiamas ir pats verslo procesas. Verslo procesu yra laikoma įmonės veiklų visuma, sukurianti vertę klientui. Toks požiūris į procesą leidžia koncentruotis ties strateginiais įmonės tikslais.

Kaip rodo metodikų analizė, nors šiuolaikinis verslo procesų valdymas, integruojantis pažangiausias nuolatinio tobulinimo bei verslo procesų reinžinerijos savybes, sudaro įmonėms palankias veiklos efektyvumo didinimo galimybes, lygiagrečiai suformuodamos ir konkrečią probleminę sritį. Viena iš šios probleminės srities krypčių yra susijusi su tuo, kad radikaliai keičiant verslo procesus yra labai sudėtinga prognozuoti veiklos efektyvumo rodiklių pokyčius ir jų atitikimą keliamiems įmonės strateginiams tikslams. Viena iš esminių priežasčių, kodėl susidaro tokios probleminės sritys, yra tai, kad tobulinimo ir reinžinerijos verslo procesai yra sudėtingi ir dinamiški, o jų intuityvus suvokimas ir našumo vertinimas dažniausiai yra klaidingas. Būtent dėl tokių priežasčių optimalių verslo procesų pokyčių valdymo sprendimų priėmimo procese būtinaja komponente tampa verslo procesų simuliacija, sudaranti galimybes eliminuoti intuityvų pokyčių vertinimo veiksnį.

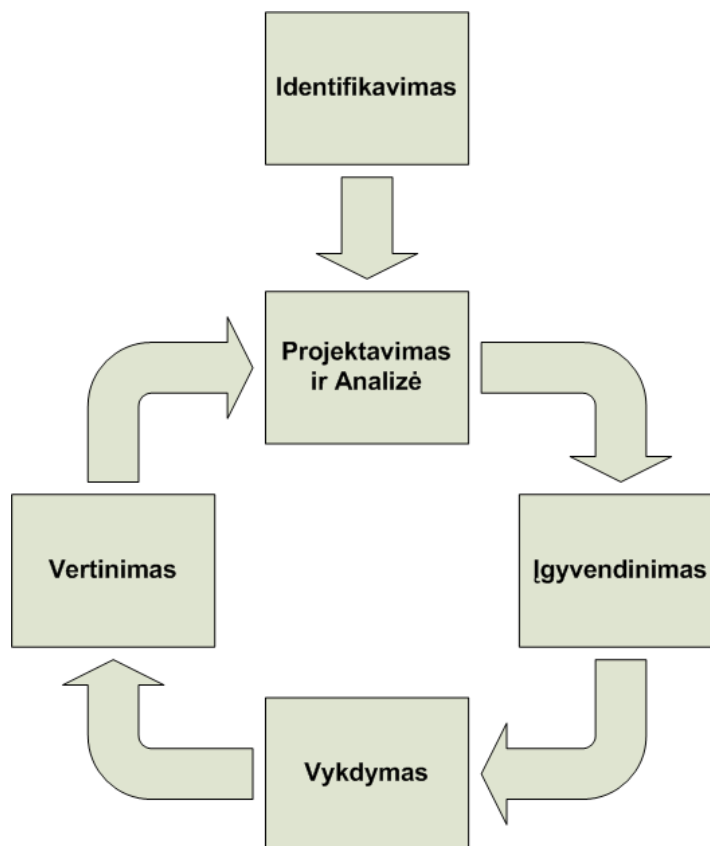
Analizuojant verslo procesų simuliaciją nuolatinio tobulinimo ir reinžinerijos kontekstuose, galima pastebėti pagrindinį skirtumą. Procesų tobulinimo metu gali būti naudojami istoriniai duomenys. Verslo procesų reinžinerijos atveju, istorinių duomenų dažniausiai nebūna.

Apibendrinant galima teigti, kad pagrindinis verslo procesų tobulinimo ir reinžinerijos tikslas yra veiklos efektyvumo rodiklių didinimas, siekiant optimizuoti įmonės veiklą ir priimant su tuo susijusius sprendimus. Tačiau verslo procesų dinamika yra stochastinė, todėl dažnai nėra galimybės intuityviai numatyti verslo proceso rezultatų ir tai gali įtakoti klaidų atsiradimą. Verslo procesų rezultatų prognozavimui yra siūloma pritaikyti VPV simuliacijos metodiką.

1.1.2. Verslo procesų valdymo gyvavimo ciklas

Šiuolaikinis VPV, kaip nuolatinio procesų tobulinimo ir verslo procesų reinžinerijos junginys, apibrėžia gyvavimo ciklo sąvoką. Nuolatinio strateginio planavimo procedūros, procesų

projektavimas, įgyvendinimas, stebėjimas, vertinimas ir tobulinimas yra procesų gyvavimo ciklo pagrindas. Nėra vieno standartizuoto VPV gyvavimo ciklo apibrėžimo. Įvairiuose literatūros



1 pav. Verslo procesų valdymo gyvavimo ciklas

šaltiniuose (Dumas at al. 2005; Weske 2007; Muehlen 2002; Hammer 2009) galima rasti skirtingų gyvavimo ciklų aprašymų, tačiau jų skirtumai nėra esminiai. Daugelis jų remiasi Demingo ciklu (Deming 1986) „Planuok, Vykdyk, Tikrink, Veik“ (Hammer 2009:6). Apibendrinus, galima išskirti kelias pagrindines verslo procesų gyvavimo ciklo fazes (1 pav.):

- **Identifikavimas** (angl. *identification*) - įmonėje vykstančių verslo procesų identifikavimas.
- **Projektavimas** (angl. *design*) - senų procesų pakeičių arba visiškai naujų procesų projektavimas. Šios fazės išeiga yra verslo procesų aprašas tam tikra modeliavimo kalba. Projektavimo metu atliekamos tokios veiklos kaip verslo procesų modeliavimas, validavimas, simuliacija ir verifikavimas.
- **Įgyvendinimas** (angl. *implementation*) - procesų modelių įgyvendinimas. Šios fazės išeiga yra įgyvendintas ir veikiantis procesas. Įgyvendinimo fazėje yra atliekamos tokios veiklos kaip verslo procesų modelių transformavimas į programų sistemų

komponentes ir komponentų testavimas.

- **Vykdymas** (angl. *enactment*) - proceso vykdymas įmonėje arba, kitaip tariant, kasdieninis proceso vyksmas. Vykdyto tarpsniu procesas yra nuolat matuojamas pamatuojamais rodikliais. Šios fazės išeiga - tai matavimo rezultatų ataskaita, kuri yra pagrindinis kitos vertinimo fazės pradinis duomuo. Vykdyto stadijoje yra atliekamos tokios veiklos kaip verslo procesų diegimas ir stebėjimas.
- **Vertinimas** (angl. *evaluation*) - proceso kokybinių ir kiekybinių parametrų vertinimas. Matavimų rezultatai rodikliai lyginami su normatyviniais rodikliais ir vadovaujantis palyginimo rezultatais yra priimamas sprendimas dėl proceso tobulinimo. Tais atvejais, kai palyginimo rezultatai yra prastesni už pageidaujamus ir yra reikalingas proceso optimizavimas, yra pereinama į verslo proceso projektavimo fazę.

Vadovaujantis nuolatinio verslo procesų tobulinimo principais, verslo procesų gyvavimo ciklo projektavimo ir analizės, įgyvendinimo, vykdymo ir valdymo fazės pasižymi ciklišku. Identifikavimo fazėje, analizuojant įmonės strateginius tikslus ir uždavinius, nuolat yra identifikuojami nauji verslo procesai, kurie turi būti optimaliai valdomi. Analizuojant mokslinės literatūros šaltinius, galima matyti, kad daugelio autorių darbuose (šaltiniai) identifikavimo fazė bendrame verslo proceso gyvavimo cikle nėra išskiriama, o verslo procesų identifikavimo tarpsnis yra priskiriamas projektavimo ir analizės fazėms.

Projektavimo ir analizės fazėje yra atliekamas naujų arba patobulintų verslo procesų projektavimas, o vėliau šie procesai yra įdiegiami, vykdomi, matuojami ir vertinami. Vertinimo fazėje atlikus matavimo rezultatų ir normatyvinių rodiklių lyginamąją analizę, yra įvertinamas verslo proceso efektyvumas. Vadinasi, šiame verslo proceso gyvavimo ciklo tarpsnyje gali būti nustatytos projektavimo metu padarytos klaidos ar netikslumai ir inicijuoti proceso pakeitimai. Tuo tarpu, jei klaidos ir netikslumai būtų identifikuoti dar projektavimo tarpsnyje, būtų gerokai sparčiau pasiektas reikiamas proceso kokybės lygis bei sumažinti įmonės kaštai. Klaidų ir netikslumų aptikimui verslo procesų projektavimo metu, rekomenduojama naudoti verslo procesų simuliacijos metodus. (Verslo procesų simuliacija leidžia iš anksto) Taip pat verslo procesų projektavimo kokybę lemia ir verslo procesų modeliavimo metodika, technikos bei priemonės.)

Apibendrinant verslo procesų gyvavimo ciklo struktūrą, galima teigti, kad verslo procesų valdymas - tai sudėtingas ciklinis procesas, todėl bet koks netikslumas pradinėje verslo proceso gyvavimo ciklo fazėje sudaro prielaidas nepageidaujamiems rezultatams susiformuoti, kas gali būti ne tik nesėkmingos verslo proceso įdiegimo patirties pavyzdžiu, bet ir sužlugdyti dalį ar visą verslo procesą. Siekiant sumažinti klaidingų sprendimų dėl verslo procesų diegimo ar reinžinerijos priėmimo tikimybę, siūloma preliminarai atlikti tokių procesų simuliaciją, kas sudarytų

pakankamas prielaidas iš anksto įvertinti proceso efektyvumą, jo neįgyvendinant ir nediegiant įmonės veikloje. Tačiau siekiant, kad simuliacijos procesas būtų patikimas bei kokybiškai atvaizduotų realią verslo aplinką, reikalinga laikytis pagrindinių projektavimo procesų kokybės reikalavimų, kurie yra numatyti atitinkamose modeliavimo metodikose bei pagal šias metodikas sukurtose techninėse priemonėse.

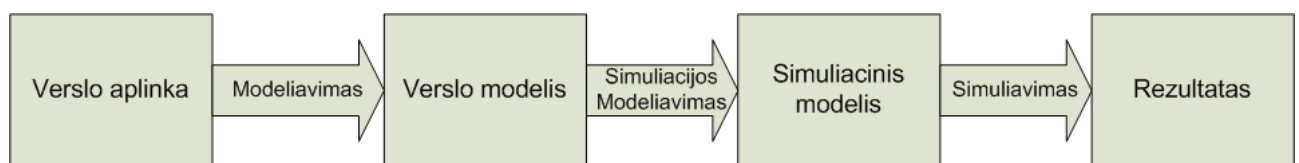
1.1.3. Verslo modeliavimas

Verslo procesai yra verslo sistemos dalis, todėl daugelis autorių verslo procesų modelius ir verslo modelius savo moksliniuose tyrimuose sutapatina arba verslo procesų modeliavimą laiko sudedamąja verslo modeliavimo dalimi. Verslo modelis (Bridgeland 2009) - tai supaprastintas sudėtingos verslo tikrovės formalus atvaizdavimas modelyje, siekiant užsibrėžto tikslo. Pagrindinis verslo modelio tikslas - kryptingai teikti žinias apie verslą ir konkrečias jo charakteristikas kitiems verslo procesų dalyviams: darbuotojams, klientams, partneriams, tiekėjams. Verslo modelis dar yra vadinamas koncepciniu modeliu (Vasilecas 2008), kurio uždavinys yra suformuoti alternatyvų požiūrį į probleminę situaciją. Verslo ar koncepcinio modelio kūrimo metu dėmesys yra sutelkiamas į nagrinėjamą verslo sritį arba verslo sistemą, jos struktūrą, elgseną, komponentes ir jų sąveiką.

Verslo modelio kūrimo procesas yra vadinamas verslo modeliavimu. Modeliavimo procesą sudaro tokios veiklos (Bridgeland 2009; Weske 2007):

1. **Projektavimas** (angl. *design*) – sukuriamas tam tikros verslo srities vienas ar keli modeliai.
2. **Tikrinimas** (angl. *validation*) – sukurtas modelis yra patikrinamas, ar atitinka visus reikalavimus.
3. **Simuliavimas** (angl. *simulation*) – naudojant modelį yra imituojama realybė ir stebimi gaunami rezultatai.
4. **Vertinimas** (angl. *verification*) – vertinama, ar sukurtas modelis atitinka verslo poreikius.

2 pav. pateikiama modeliavimo metu gaunama artefaktų seka.

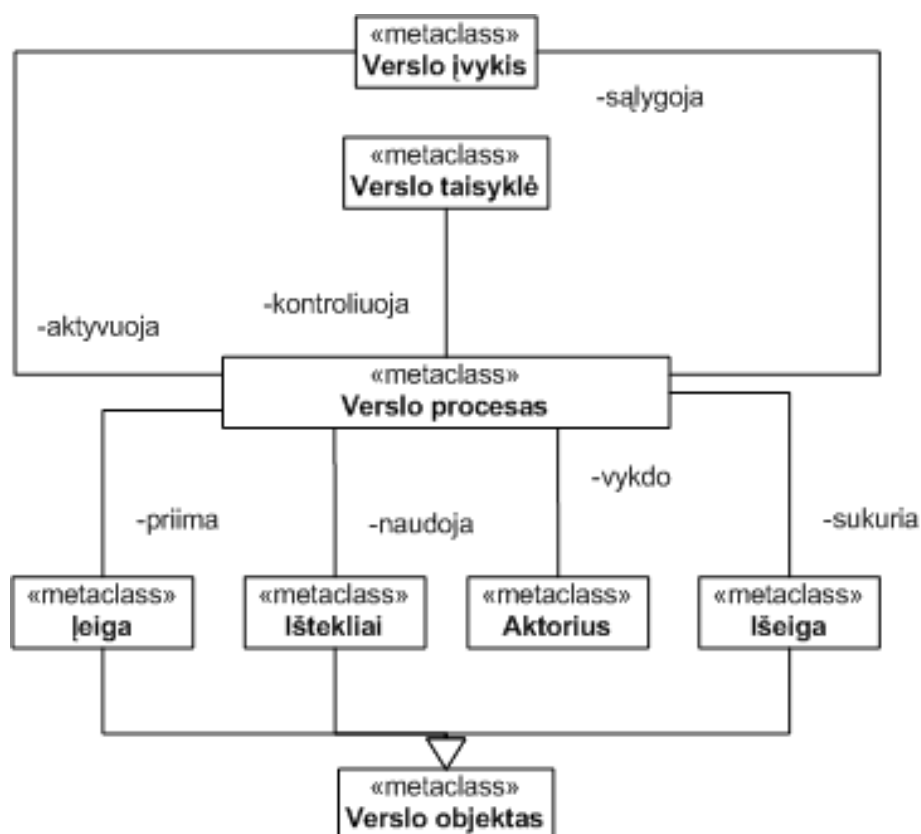


2 pav. Verslo procesų modeliavimas

Verslo modeliavimo metu, nagrinėjant tam tikrą dalykinę ir problemines sritis yra sudaromas supaprastintas realybės aprašymas - verslo modelis. Verslo modelis turi būti įvertintas ir patikrintas. Tikrinimo metu vienas iš svarbiausių uždavinių yra nustatyti ar aprašyto modelio būsimi rezultatai tenkins lūkesčius. Tam gali būti naudojama simuliacija. Verslo modelio pagrindu yra kuriamas verslo procesų simuliacinis modelis. Simuliacinis modelis gali būti atskiras nuo verslo modelio arba verslo proceso modelis gali būti papildomas informacija, reikalinga simuliacijai atlikti. Turint verslo simuliacijos modelį yra atliekama verslo procesų simuliacija ir gaunami rezultatai, kurie padeda įvertinti modelio teisingumą.

1.1.4. Verslo procesų valdymo metamodelis

Modeliuojant verslo procesus ar bendrai verslą, yra naudojamos tam tikros modeliavimo kalbos. Modeliavimo kalbos apibrėžimui yra naudojami metamodeliai. Kitaip tariant metamodelis yra modeliavimo kalbos modelis. Metamodelis turi būti tiksliai apibrėžtų sąvokų ir taisyklių darinys, kuris reikalingas dalykinės srities modelių kūrimui. Verslo procesų valdymo srityje yra apibrėžtas ne vienas metamodelis.



3 pav. Verslo procesų valdymo metamodelis

3 pav. yra pateikiamas apibendrintas VPV metamodelis (Vasilecas 2008; TOGAF9.1). Šis metamodelis apibrėžia pagrindines VPV sąvokas ir sąryšius tarp jų, reikalingas šio darbo problemai

spręsti. Lyginant su industrijos pripažintų standartų, tokių kaip BPMN pateikiamais metamodeliais, šis metamodelis yra supaprastintas ir pavaizduoja tik pagrindines esybes. Iš kitos pusės jis yra platesnis, nes savyje apjungia ir verslo taisykles, ir procesus.

Verslo proceso vykdymo metu įeiga yra verčiama išeiga, naudojant išteklius. Įeiga, išeiga ir ištekliai bendrai yra verslo objektai. Taigi, verslo procesai operuoja verslo objektais. Natūralu, kad verslo procesas savaime vykti negali, todėl verslo proceso metamodelyje yra išskiriamas tam tikras verslo proceso vykdytojas, kuris įvardijamas aktoriumi. Aktoriaus funkcijas verslo proceso vykdyme gali atlikti tiek žmogus, tiek padalinys, tiek ir programų sistema ar panašiai. Verslo procesus aktyvuoja įvykiais. Atitinkamai verslo proceso vykdymo metu gali būti sąlygojami nauji įvykiai, kurie savo ruožtu gali aktyvuoti kitus procesus.

Realioje verslo procesų vyksmo aplinkoje verslo procesai negali vykti be kontrolės mechanizmo, t.y. visi verslo procesai turi savo vykdymo taisykles ir kontrolė yra atliekama nustatant, ar proceso vykdymo procedūra atitiko taisykles. Analizuojamame apibendrintame verslo procesų valdymo metamodelyje numatytas verslo procesų kontrolės per verslo taisykles mechanizmas. Būtent todėl verslo taisyklės nėra verslo proceso dalis ir verslo procesai nevykdo verslo taisyklių.

Verslo taisyklės ir verslo procesai yra savarankiškos esybės (Ross 2008; BR Manifesto). Tačiau verslo modelio kontekste verslo procesai ir verslo taisyklės yra glaudžiai susiję ir yra neatsiejama verslo modelio dalis. Kuriant verslo modelį verslo procesai ir verslo taisyklės turi būti modeliuojamos kartu.

Verslo procesų valdymo raidos procese verslo proceso ir verslo taisyklės konceptai buvo tyrinėjami atskirai kaip du savarankiški ir tarpusavyje nepriklausomi mokslinių tyrimų objektai. Tokiomis technologinės raidos dinamikos sąlygomis sukurtos verslo procesų ir verslo taisyklių modeliavimo metodikos bei priemonės nepasižymi lanksčiomis verslo procesų ir verslo taisyklių integruoto modeliavimo charakteristikomis, nors paskutiniai moksliniai tyrimai jau įrodė, kad integruotas verslo procesų ir verslo taisyklių panaudojimas leidžia pasiekti realiai apčiuopiamų teigiamų veiklos efektyvumo rodiklių pokyčių (šaltiniai). Atsižvelgiant į tai, kad šių verslo esybių integruotas panaudojimas rodo teigiamą efektą, kuriant verslo modelį gali būti sėkmingai ir efektyviai naudojamas verslo procesų ir verslo taisyklių kontinuumas, tačiau itin didelės svarbos, tokiomis verslo procesų valdymo modeliavimo aplinkybėmis, reikalauja ir pačios verslo proceso bei verslo taisyklės sampratos.

1.1.5. Verslo proceso sąvoka

Įvairiuose literatūros šaltiniuose galima rasti ne vieną verslo proceso apibrėžimą. Ronald G. Ross (Ross 2005) verslo procesą apibrėžia kaip į verslo dalyvių vaidmenis, o ne į kasdienį jų darbą orientuotą užduotį, kuri turi būti atlikta siekiant įmonės tikslų, reaguojant į tam tikrus verslo aplinkos įvykius. M. Weske (Weske 2007) teigia, kad verslo procesas sudarytas iš veiklų rinkinio, kurį koordinuoja organizacinės ir techninės verslo aplinkos, o visas rinkinio veiklas sieja įmonės strateginės užduotys. Kiekvieno verslo proceso komunikacinė sistema yra apibrėžta vienos įmonės ribose, tačiau dažnai realioje verslo aplinkoje ši sistema išeina už vienos įmonės ribų ir įsilieja į keletą įmonių verslo procesų komunikacinę sistemą.

2011 metais OMG publikavo antrą BPMN specifikaciją (BPMN2.0), kurioje yra aprašomas verslo procesų meta-modelis, modeliavimo notacija bei apsikeitimo modeliais formatas. Vienas iš pagrindinių BPMN specifikacijos tikslų yra apibrėžti standartą, kuris būtų suprantamas visiems įmonės atstovams, nuo verslo analitikų, kurie kuria pradines procesų modelių versijas, iki technologinio lygio žinovų, kurie yra atsakingi už technologijų diegimą ir procesų automatizavimą, bei verslo žmonėms, kurie yra atsakingi už verslo procesų stebėjimą ir sprendimų priėmimą.

BPMN specifikacijoje verslo procesas apibrėžiamas kaip organizacijos veiklų (angl. *Activity*) seka, kurios tikslas yra atlikti tam tikrą darbą.

Viena iš pagrindinių užduočių, siekiant suvokti verslo procesų struktūrą ir atlikti jų analizę, yra verslo procesų klasifikavimas. Verslo procesai gali būti klasifikuojami pagal įvairias savybes (Dumas at al. 2005; Weske 2007; BPMN2.0): pagal proceso dalyvius, pagal proceso struktūrą ir nuspėjamumą, pagal proceso komunikacines ribas ir pan.

Pagal proceso dalyvius procesai gali būti klasifikuojami į (Dumas at al. 2005):

- P2P (angl. *Person-to-Person*) procesus, kuriuose visas veiklas atlieka žmogus. Žmonės dažnai pasitelkia įvairias informacines sistemas, tokias kaip el. pašto, projektų valdymo, grupinio darbo sistemas, bet tos sistemos dažniausiai tik palengvina žmogaus darbą arba bendravimą ir tiesiogiai neatlieka (neautomatizuoja) procesų veiklų.
- A2A (angl. *Application-to-Application*) procesai, kuriuose visos veiklos yra automatizuotos ir atliekamos programų sistemose.
- P2A (angl. *Person-to-Application*) procesai, kurie yra P2P ir A2A procesų mišinys. Dalį proceso veiklų atlieka žmogus, o kitą dalį automatizuotai atlieka programų sistemos.

Kitaip toks klasifikavimas dar vadinamas klasifikavimu pagal automatizacijos laipsnį (Weske 2007), o toks klasifikacijos pagal proceso dalyvius apibūdinimas yra sutinkamas ir BPMN

specifikacijoje, kurioje yra išskiriami ir apibrėžiami automatizuojami ir neautomatizuojami vidiniai įmonės procesai.

Pagal proceso struktūrą ir nuspėjamumą (angl. *structure and predictability*) verslo procesai gali būti skirstomi į atsitiktinius (angl. *ad-hoc*), administracinius (angl. *administrative*) ir kasdieninius (angl. *production*) (Dumas at al. 2005). Mokslinės literatūros šaltiniuose tokia klasifikacijos schema yra dar įvardijama kaip klasifikacija pagal pasikartojamumą (angl. *degree of repetition*), klasifikacija pagal struktūrą (angl. *degree of structuring*) (Weske 2007) ir pan.

Pagal proceso ribas verslo procesai gali būti klasifikuojami į griežtai vidinius ir išeinančius už įmonės ribų. (angl. *intraorganizational and interorganizational*) (Dumas at al. 2005). BPMN taip pat procesus klasifikuoja į vidinius (angl. *Private*) ir išorinius (angl. *External*).

1.1.6. Verslo procesų modeliavimas

Verslo modeliavimo įrankiai, notacijos ir standartai yra tik techninė verslo procesų modelių kūrimo dalis. Kitas labai svarbus verslo procesų modeliavimo aspektas yra metodas, leidžiantis sukurti aiškius ir lengvai valdomus verslo procesų modelius. Verslo procesų modeliavimas (specifikavimas) gali būti (Mulyar at al. 2007): imperatyvus (procedūrinis), kai aprašoma griežta veiksmų seka, arba deklaratyvus, kai sudaroma tik apribojimų sistema.

Šiuo metu dažniau yra naudojamas imperatyvus verslo procesų modeliavimo metodas. Tai patvirtina ir sparčiai populiarėjantis BPMN standartas. Siekiant užtikrinti verslo procesų modelių kokybę rekomenduojama laikytis septyniomis žemiau prioriteto tvarka išvardintomis taisyklėmis (Mendling 2010):

- Modelis turi būti kiek įmanoma labiau struktūrizuotas.
- Didesni nei 50 elementų modeliai turi būti dekomponuoti į kelis modelius.
- Modelyje reikia naudoti kiek įmanoma mažiau elementų.
- Veiklų pavadinimams naudoti veiksmažodžio-objekto (angl. verb-object) antraštes.
- Kiekvieno elemento maršrutų kiekis turi būti minimizuotas.
- Turi būti naudojamas tik vienas pradžios ir pabaigos įvykis.
- Reikia vengti ARBA proceso šakojimo elementų.

W.M.P. van der Aalst (Aalst 2009) tai pat teigia, kad įmonėje vykstančius procesus yra efektyviau modeliuoti naudojant daugiau smulkesnių ir nepriklausomų procesų, kurie tarpusavyje komunikuoja, nei bandyti kurti monolitinius procesų modelius. Kitas verslo procesų modeliavimo metodas (Koehler 2010) rekomenduoja kurti nedidelius procesų fragmentus, orientuotus į agentų veiksmus ir verslo objektų gyvavimo (būsenų) ciklą, kurie tarpusavyje bendrauja per įvykius (angl.

events). Tokių procesų fragmentų apjungimui rekomenduojama naudoti verslo taisykles, o tai reiškia, kad šis metodas apjungia tiek imperatyvų, tiek deklaratyvų verslo procesų modeliavimą.

1.1.7. Verslo taisyklės sąvoka

“Business rules group” verslo taisyklę apibrėžia verslo ir informacinės sistemos požiūriais (BR Manifesto). Verslo požiūriu verslo taisyklėmis yra laikomos gairės, kurios yra tam tikros veiklos ar sferos įpareigojimai susiję su elgesiu, veiksmais, taikoma praktika ar procedūromis. Verslo taisyklės turi dvi svarbias (išskirtines) savybes:

- Taisyklė turi turėti aiškią motyvaciją.
- Taisyklės turi turėti priemones ir tvarką, užtikrinančią, kad taisyklių bus laikomasi, bei apibrėžiančią, kokios yra pasekmės už taisyklių nesilaikymą.

Informacinės sistemos požiūriu verslo taisyklė yra teiginys, kuris apibrėžia arba apriboja tam tikrą verslo aspektą. Jos yra skirtos verslo struktūros apibrėžimui ar verslo elgsenos kontrolei ar įtakai. 2010 metais OMG publikavo verslo motyvacijos modelio (anlg. *Business motivation model – BMM*) pirmą specifikacijos versiją (BMM1.1). BMM verslo taisyklę apibrėžia kaip direktyvą, skirtą valdyti, nukreipti ar veikti verslo elgseną atitinkamai verslo politikoms, kurios yra formuluojamos atsižvelgiant į įmonės stiprybes, silpnybes, galimybes ir grėsmes (anlg. SWOT). “Business Rules Group” yra išleidusi manifestą (BR Manifesto), kuriame apibrėžia verslo taisyklių savybes ir panaudojamumą:

- Pirmaeiliai, ne antraeiliai reikalavimai (anlg. *Primary Requirements, Not Secondary*).
- Atsietos nuo procesų, ne jų dalis (anlg. *Separate From Processes, Not Contained In Them*).
- Sistemingos, ne atsitiktinės, žinios (anlg. *Deliberate Knowledge, Not A By-Product*).
- Deklaratyvios, ne procedūrinės (anlg. *Declarative, Not Procedural*).
- Taisyklingos, ne bet kaip sudarytos, išraiškos (anlg. *Well-Formed Expression, Not Ad Hoc*).
- Taisyklėmis grindžiama architektūra, ne jos imitacija (anlg. *Rule-Based Architecture, Not Indirect Implementation*).
- Taisyklių valdomi procesai, ne išimčių programavimas (anlg. *Rule-Guided Processes, Not Exception-Based Programming*).
- Veiklai, ne technologijai (anlg. *For the Sake of the Business, Not Technology*).
- Veiklos dalyviams priklausančios, jų sukurtos ir skirtos jiems, o ne informacinių technologijų specialistams (anlg. *Of, By, and For Business People, Not IT People*).

- Valdančios veiklos logiką, ne techninės/programinės įrangos aplinką (angl. *Managing Business Logic, Not Hardware/Software Platforms*).

Tęsiant „Business Rules Group“ darbą OMG 2008 metais publikavo verslo žodyno ir verslo taisyklių semantikos (angl. *Semantics of Business Vocabulary and Rules*) standartą SBVR1.0, kuris sutrumpintai žymimas SBVR. Šiuo metu tai vienas iš populiariausių verslo žodynų ir verslo taisyklių sudarymo standart. SBVR verslo taisyklės apibrėžia kaip tai, kas yra verslo jurisdikcijoje (angl. *rule that is under business jurisdiction*). Verslo jurisdikcijoje reiškia, kad verslas gali laikytis, peržiūrėti, keisti, atsisakyti verslo taisyklių taip, kaip tai yra suprantama verslo požiūriu. Jei taisyklės yra ne verslo jurisdikcijoje, tai ji nėra verslo taisyklė. Pavyzdžiui taisyklės, diktuojamos gamtos ar matematikos dėsniais, nėra verslo taisyklės.

SBVR standartas apibrėžia 2 verslo taisyklių klases:

- Struktūros taisyklė (angl. *Structural rules*) - struktūros taisyklė apibrėžia statinius verslo aspektus ar išreiškia sprendimo darymo kriterijus. Struktūros taisyklės pavyzdys: „vartotojas, įsigijęs prekių daugiau negu už 500 Lt, yra priskiriamas „auksinių“ pirkėjų kategorijai“.
- Veiklos taisyklė (angl. *Operative rules*) - veiklos taisyklė apibrėžia, kaip vyksta veikla laike. Šio tipo taisyklės pavyzdys: „namų darbai turi būti padaryti prieš einant į darlą“.

1.1.8. Procesų ir taisyklių kontinuumas

Galimi įvairūs verslo procesų ir verslo taisyklių bendro panaudojimo scenarijai. GARTNER kompanijos atlikto tyrimo ataskaitoje (Sinur 2009) yra pateikiami 7 verslo procesų ir verslo taisyklių bendro panaudojimo scenarijai, tačiau Joana Koehler (Koehler 2011) teigia, kad pakanka nagrinėti 4 skirtingus scenarijus:

- Taisyklės neišskiriamos - visa verslo taisyklių bei sprendimų priėmimo logika yra modeliuojama kaip vieno proceso šakos, atskiros taisyklės sąvoka nenaudojama.
- Išskirtos taisyklių veiklos - verslo procesai ir verslo taisyklės yra atskiros, jos modeliuojamos ir valdomos skirtingais būdais. Dažniausiai verslo procesai išskiria verslo taisyklės kaip veiklas. BPMN notacijoje tam tikslui yra apibrėžta specialus veiklų tipas - verslo taisyklių veiklos (angl. *Business Rules Task*). Verslo taisyklės gali būti valdomos ir vykdomos specializuotose verslo taisyklių mašinose. Pagrindinis klausimas, kuris kyla naudojant šį verslo taisyklių ir verslo procesų modeliavimo scenarijų yra, kas turi būti modeliuojama kaip procesai, o kas kaip taisyklės.
- Taisyklių koordinuojami procesai - taisyklės yra apibrėžiamos skirtinguose

kontekstuose ir taisyklių vykdymo rezultatai yra naudojami už vieno proceso ribų, kad nukreipti kitus procesus link užbrėžto tikslo.

- Aktyvios taisyklės - kiekvienas verslo proceso atvejis yra konstruojamas iš atskirų proceso komponentų ar atskirų veiklų dinamiškai, laikantis verslo taisyklėmis aprašytų apribojimų. Šiuo atveju yra kuriami arba proceso fragmentai arba yra aprašomos tik veiklos. Procesas yra dinamiškai sukonstruojamas remiantis verslo taisyklėmis. Taisyklės, siekiant optimizuoti verslo procesus, gali kisti laike.

Šiuo metu populiariausias yra antras verslo procesų ir verslo taisyklių integravimo scenarijus. BPMN notacija tokiai taisyklių ir procesų integracijai numato specialią verslo taisyklių veiklą.

Daugelis šiuolaikinių verslo procesų ir verslo taisyklių įrankių leidžia atskirai modeliuoti verslo taisykles ir verslo procesus, bei lengvai juos integruoti vykdymo metu, vadinamose verslo procesų (angl. *BPMS*) bei verslo taisyklių varikliuose (angl. *BRMS*). Kadangi verslo procesų ir verslo taisyklių modeliavimas yra atliekamas atskiruose įrankiuose, tai neleidžia sukurti vieningo verslo procesų ir verslo taisyklių modelio. Taip pat, atskiruose įrankiuose kuriamus verslo procesų ir verslo taisyklių modelius yra labai sudėtinga simuliuoti, o atitinkamai ir įvertinti modelio teisingumą.

1.2. Simuliavimas

1.2.1. Pagrindinės sąvokos

Simuliavimas - tai eksperimentinis metodas, kurio pagrindinis tikslas yra ištirti sumodeliuoto verslo proceso elgseną. Mokslinėje literatūroje egzistuoja daug simuliavimo apibrėžimų. Pagal Shannoną (Shannon 1975) simuliavimas - tai realiai veikiančios sistemos aprašymo (aiškinimo) procesas, kurio metu yra naudojamas eksperimentinio modeliavimo metodas, siekiant suprasti sistemos elgseną arba išnagrinėti egzistuojančias alternatyvias veiklos strategijas. Pagal (Smith 1998) simuliavimas - tai realios arba įsivaizduojamos sistemos modelio projektavimas ir eksperimento su tuo modeliu atlikimas, siekiant suprasti sistemos elgseną arba įvertinti sistemos operacinę strategiją. Pritaikant simuliavimo sampratą verslo procesų simuliavimui, galima teigti, kad simuliavimas - tai verslo proceso vertinimas, kurį dažniausiai atlieka analitikai.

Probleminė simuliavimo sritis yra susijusi su modelio sampratos specifika. Modelis - tai realios aplinkos abstrakcija, supaprastinimas, pritaikytas realaus objekto sudėtingumui valdyti. Tačiau net supaprastintas požiūris į verslo proceso aplinką gali būti pakankamai sudėtingas valdymo

požiūriu, kadangi egzistuoja didelis kiekis nepriklausomų komponentų ir įvykių, kurie tarpusavyje intensyviai sąveikauja. Sudėtingumą suformuoja todėl, kad kiekvienas verslo procesas pasižymi didele lygiagrečiai vykstančių įvykių, turinčių skirtingas būsenas ir reikalaujančių skirtingų išteklių, gausa. Vieni procesai turi būti pristabdyti, kol bus suformuoti pakankami pradiniai išteklių jiems vyksti, todėl atsiranda tam tikri trikdžiai.

Net matematinio optimizavimo metodai tik pavieniais atvejais gali padėti išspręsti susidariusią problemą, todėl tokie metodai negali būti pritaikomi daugeliui praktikoje susiformuojančių probleminių situacijų sprendimui. Analitiniai modeliai tuo ir skiriasi nuo simuliacinio, kad jie siūlo konkrečią lygčių sistemą, turinčią uždaros formos (angl. *closed - form*) apibendrintą sprendinį. Tačiau matematinio modeliavimo uždaviniuose yra suformuluojama pernelyg daug pradinių sąlygų ir apribojimų, dėl kurių sprendžiamo uždavinio atitikimas realiai aplinkai itin smarkiai sumažėja. O simuliacinis, šiuo atžvilgiu, nesudaro galimybės rasti optimalaus sprendinio, nes šio metodo pritaikymas leidžia apskaičiuoti konkrečių situacijų sprendimus ir “atspėti” sistemos elgesio specifiką. Būtent todėl simuliacinio rezultatus, pritaikius optimizavimo metodus, galima rasti labiausiai artimą pageidaujamą sistemos modelio elgseną, o toks optimizavimo metodas dar yra vadinamas modelio jautrumo analize (angl. *sensitivity analysis*).

Mokslinės literatūros šaltiniuose yra išskiriamos dvi simuliacinio rūšys diskretusis ir tolydusis simuliacinis. Tolydusis simuliacinis metu įvykių būsenos tolygiai kinta, o pats aprašomasis modelis pasižymi tuo, kad jo išvadoje yra sudaroma (angl. *fitting*) lygčių sistema. Diskretusis simuliacinis metu daroma prielaida, kad visos modelio pokyčių būsenos yra diskretūs įvykiai konkrečiu laiko momentu, todėl nieko neįvyksta tarp dviejų kaimyninių mazgų vienu ir tuo pačiu laiko momentu. Verslo procesams simuliuoti toks metodas yra pakankamas dėl toliau aptariamų priežasčių. Pirmiausia, diskretusis simuliacinis - tai natūralus verslo procesų simuliacinio metodo pasirinkimas, nes visi verslo procesai gali būti išreiškiami įvykiais, kurie įvyksta diskrečiais laiko momentais. Tačiau būtina atkreipti dėmesį į tai, kad simuliacinio rezultatų kokybė tiesiogiai priklauso nuo sukurto modelio teisingumo bei pradinių duomenų patikimumo. Ypatingai tai aktualu simuliuojant sudėtingus verslo procesų modelius, kuriuose net ir nedidelės klaidos ir jų santykis su nedidelėmis skaičiavimų paklaidomis gali itin akivaizdžiai pakeisti simuliacinio rezultatus.

1.2.2. Diskrečiųjų įvykių simuliacija

Diskrečiųjų įvykių modeliai yra sudaryti iš tokių vienetų, kurių būsenos pagal tam tikras taisykles ir dėsningumus kinta laikui bėgant. Minėti vienetai taip pat turi tam tikras vidines

būsenas, kurios specialių metodų pagalba gali būti keičiamos. Specialūs metodai yra taikomi būsenų pokyčiams suaktyvinti ir būsenų pokyčiai turi būti susiejami su tam tikru atskaitos tašku modelio laiko skalėje.

Modelio laikas yra visiškai nepriklausomas nuo realios aplinkos laiko, o tai apsaugo simuliaciją nuo realios skaičiavimų trukmės. Tai yra būtina simuliuojant sudėtingus (kompleksinius) procesus kurie simuliuojami yra ženkliai ilgiau, nei vyksta realioje aplinkoje.

Būsenos pokyčiai yra suaktyvinami per diskrečius įvykius, kurie atsiranda konkrečiuose simuliuojamo modelio laiko ašies taškuose. Simuliacijos laikas yra skaičiuojamas nuo įvykio iki įvykio, tačiau tarp įvykių vienetų būsenos negali keistis. Tam, kad tai galėtų būti įgyvendinta yra reikalingas specialus centrinis valdiklis, kurio pagalba gali būti išlaikomas įvykių sąrašo nuoseklumas, kuris užtikrina centralizuotą modelio laiko parametro valdymą. Tuomet nesudėtingo algoritmo pagalba pagal eiliškumą yra parenkamas sekantis įvykis, modelio laikas nustatomas atitinkamam laikui ir yra įvykdomas įvykis. Priklausomai nuo simuliacijos tikslų įvykiams konkrečiu laiko momentu gali būti suteikiamas prioritetas, jie gali būti įvykdomi bet kokia tvarka ir valdomi sudėtingesnio algoritmo pagalba.

Toks požiūris leidžia simuliaciją vykdyti leidžiant nuosekliai vyksti įvykiams. Gali būti generuojami tiek vidiniai, tiek ir išoriniai įvykiai, kas parodo, kad įvykiai sugeneruojami vieneto būsenos pokyčių išdavoje, o įvykių trukmė (laikas) gali būti tiek deterministinis, tiek ir stochastinis.

Diskrečių įvykių simuliacijoje sinchronizuojant laiko parametą įvykiai nėra tik konceptai. Be to, gali būti naudojamos tokios veiklos, kurioms aprašyti reikalingi keletas įvykių: pradžios įvykis, pabaigos įvykis, o kartais tarpiniai įvykiai.

Modelio būseną: kiekvienu laiko momentu modelio vienetai turi turėti apibrėžtą būseną, kuri vienokiu ar kitokiu būdu turi būti įgyvendinta sistemoje.

Simuliacijos laikrodis, kuris rodo einamąjį modelio laiką, neturintį jokių sąsajų su realios aplinkos laiku.

Įvykių sąrašas: tam tikra įvykių chronologinė seka, išdėstyta modelio laiko skalėje.

Centrinis valdiklis (angl. *sheduler*) privalo įvykdyti aprašytą pagrindinį algoritmą ir nutraukti simuliaciją konkrečiu (aprašytu) momentu.

Atsitiktinių skaičių generatorius, kuris yra reikalingas tam, kad sugeneruoti simuliacijos metu reikalingus duomenų srautus. Generatorius gali generuoti atkuriamus kai kurių stochastinių skirstinių reikšmių srautus, todėl kartais yra vadinamas pseudo atsitiktinių skaičių generatoriumi.

Statistiniai skaičiuotuvai (angl. *counter*)/ Duomenų kaupikliai (angl. *collector*) kaupia eksperimento statistinius duomenis ir iš dalies koku nors būdu juos gali interpretuoti.

Visi modeliavimo tipai dažniausiai susiduria su tipine problema, kaip sumodeliuoti laiko

tėkmę, nes simuliacinio laiko atskaita yra visiškai izoliuota nuo realios proceso vyksmo laiko atskaitos. Diskrečių įvykių simuliacinis yra vykdomas griežtai pagal suplanuotą grafiką ir turi įvykti konkrečiu modelio laiko momentu. Populiariausi tokių problemų sprendimai yra atliekami vadovaujantis dviem požiūriais: įvykiais pagrįstu ir procesais pagrįstu požiūriais, naudojamais daugelyje šiuolaikinių simuliacinio priemonių. Pagrindinis šių požiūrių skirtumas yra tas, kad vienu atveju yra modeliuojami įvykiai, o kitu - ryšiai tarp įvykių.

Procesais pagrįstu, modeliavimo požiūriu, visi įvykiai yra modeliuojami viso gyvavimo ciklo fazėse ir iš visų įmanomų vienetų tipų.

Stilistikos pasirinkimas priklauso nuo simuliacinio užduočių, tačiau pasirinkimas šiuo atveju yra labai kritinis. Dažniausiai stiliaus pasirinkimas priklauso nuo dviejų veiksnių:

- sukurti atskirą verslo procesų simuliacinio modelį pasirinktos simuliacinio priemonės kalba;
- pasinaudoti egzistuojančiu verslo proceso varikliu ir aprašyti verslo procesą atitinkama kalba (simuliacinio priemonė valdo tik variklį).

Naudojant verslo procesų variklį yra keletas esminių veiksnių, į kuriuose reikia atkreipti dėmesį pasirenkant modeliavimo tipą (stilių). Verslo proceso modelis. Jeigu egzistuojantys verslo proceso modeliai gali būti keletą kartų panaudojami, tai daug darbo elementų simuliacinio modelyje yra išsaugomi. Ir kiekvienas “nuosavas” simuliacinio modelis tampa neintegruotu į Verslo proceso modelį, todėl greitai “pasensta”, būtent todėl modelio laiko tėkmė laukimo būsenose yra jungiamoji verslo procesų ir jų simuliacinio grandis.

Ištekliai taip pat labai svarbi verslo proceso modelio sąvoka. Ištekliai gali būti žmogiškieji, įrangos, finansiniai ir pan. Modelyje juos pakanka aprašyti kaip ribotus išteklius, naudojamus pagal poreikį ir grąžinamus atgal likučių pavidalu. Tokie procesai, kuriems reikalingi konkrečiu laiko momentu neprieinami ištekliai, turi laukti eilėje.

Įvykiai: verslo procesų varikliai yra sukurti įvykių pagrindu, todėl veikia išorinius įvykius per specialius procesų srautus ir veiksmus.

Esybės. Iš pirmo žvilgsnio taip atrodo, kad verslo procesai iš esmės, o kartu ir ištekliai, taip pat gali būti laikomi esybėmis. Ir tai nėra klaidinga, tačiau vėlesniuose etapuose, kai bus atliekamas simuliacinis bus matyti kad formuluojant užduotis arba aprašant laukimo būsenas, ženkliai patogiau operuoti pagrindinėmis simuliacinio esybėmis.

Taigi, visa sistema atrodo, lyg būtų procesais grindžiama sistema, galbūt transakcijomis grindžiama sistema, tačiau eliminavus verslo procesų variklį sukurtas savybes, ir pažiūrėjus į modelį kaip į simuliacinio priemonės variklį, galima pastebėti, kad modelis pasižymi ir į įvykius orientuotu požiūriu. Taip yra dėl to, kad diskretus įvykių simuliacinis aprašo įvykius ir apdoroja

juos verslo proceso varikliais. Ir jeigu verslo procesas yra laukimo būsenoje, o modelio laikas turi būti sueikvotas, yra daug paprasčiau sukurti įvykį verslo proceso variklyje ir vėliau jį reaktyvuoti, o tada apdoroti jį diskretaus įvykių simuliacijos karkasu. Tai rodo, kad procesais grindžiami tyrimai tampa įvykiais grindžiamais tyrimais tais atvejais, kai yra laikoma, jog procesas gali būti reaktyvuojamas kaip įvykis.

1.2.3. Statistikos

Diskrečių įvykių simuliacijos pasižymi tam tikru atsitiktinumu ir tai yra pagrindinė priežastis, kodėl simuliacijai yra suteikiamas didesnis prioritetas nei matematiniam modeliavimui. Atsitiktinumas yra realios aplinkos veiksnys, todėl atsitiktinumo charakteristiką simuliacijos modelyje apibrėžia stochastinės komponentės, kurios dažniausiai priklauso stochastiniams skirstiniams.

Kompiuterių sistemose atsitiktinumo veiksnys yra tam tikro lygio problema, nes kompiuterinės technologijos pasižymi skaičiavimo charakteristika, o atsitiktinumas gali būti sukuriamas tik tam tikros techninės įrangos pagalba. Tačiau daugelių atvejų galima išsiversti ir be tokios įrangos, kadangi egzistuoja atsitiktinių stochastinių skirstinių reikšmių generatorius, kuriame atsitiktinumo veiksnys yra sukuriamas specialių algoritmų pagalba. Pagrindiniu reikalavimu atsitiktinėms reikšmių generacijoms tampa reikalavimas, kad reikšmės, vieno ciklo metu, neįgytų tokių pačių reikšmių.

Daugelis atsitiktinių skaičių generatorių reikšmės prasideda nuo tam tikros pradinės reikšmės, vadinamos išeigos tašku. Jeigu laikyti, kad jokie istoriniai duomenys egzistuoja, iškyla klausimas, kodėl jų negalima būtų panaudoti kaip pradinius simuliacijos duomenis? Problema yra ta, kad tie pradiniai duomenys yra pasiskirstę empiriniu skirstiniu ir paaiškina sistemos elgesį tik fiksuotu laikotarpiu ir šiame duomenų masyve gali nebūti visų įmanomų sistemos elgesį aprašančių reikšmių.

1.3. Verslo procesų valdymo ir simuliacijos sistemos

1.3.1. Pagrindinės sąvokos

Kaip ir daugelyje sričių, taip ir verslo procesų valdyme, informacinės technologijos yra vienas pagrindinių elementų, leidžiančių efektyviai atlikti visas veiklas nuo verslo procesų projektavimo iki vykdomų procesų matavimo ir vertinimo. Nors verslo procesų valdymas nėra vien tik technologinių priemonių, leidžiančių automatizuoti procesų gyvavimo ciklą, diegimas, tačiau

technologinės priemonės yra neatsiejama šiuolaikinio verslo procesų valdymo dalis. Tyrimai šioje srityje parodė, kad procesais grindžiamų informacinių sistemų kūrimas reikalauja mažiau pastangų nei sistemų kūrimas paremtais atvejais (angl. *case*) (Weber at al. 2010) bei detalus, vykdymui paruoštų procesų modeliavimas ir simuliacijos, o taip pat ir procesų vykdymo mašinų diegimas didina įmonės suvokimą apie joje vykstančius procesus bei procesų efektyvumą (Aalst 2005) bei didina sėkmingos verslo procesų reinžinerijos projektų baigties tikimybę. Šiuo metu nėra vieningo verslo procesų valdymo sistemos apibrėžimo. Tai gali būti ir modeliavimo bei simuliacijos įrankiai, procesų vykdymo mašinos, procesų našumo rodiklių stebėjimo bei analizės priemonės. Šiame darbe VPV sistema yra vadinama tarpusavyje integruotų technologinių priemonių aibė, leidžianti automatizuoti visas (ar dalį) verslo proceso gyvavimo ciklo veiklas. Kadangi VPV sistema automatizuoja gyvavimo ciklo veiklas, tai jomis remiantis galima apibrėžti ir bendrinius reikalavimus VPV sistemai (Havey 2005):

- Projektavimas - priemonės turi leisti modeliuoti verslo procesus, užtikrinti modelių teisingumą. Priemonės turėtų turėti simuliacijos galimybes, kad būtų galima įvertinti modeliuojamo proceso rezultatų prognozę.
- Įgyvendinimas - priemonės turi užtikrinti verslo procesų modelių transformavimą į vykdymui paruoštas komponentes (programinį kodą ar specialią vykdymo kalbą). Modelio transformavimas nėra būtinas reikalavimas. Verslo proceso modeliai gali būti projektuojami taip, kad juos iškart būtų galima vykdyti. Tokiu atveju ir projektavimo ir įgyvendinimo uždaviniai gali būti atliekami tomis pačiomis priemonėmis.
- Vykdydas - vykdymo priemonės turi užtikrinti verslo procesų modelių vykdymą, proceso parametrų matavimą, istorinių duomenų kaupimą, komunikaciją su išorės sistemomis bei procesų vykdytojais. Prie vykdymo priemonių kategorijos taip pat yra priskiriamos ir monitoringas, administravimo priemonės bei vartotojo sąsajos, leidžiančios valdyti neautomatizuotas proceso veiklas kaip užduotis vykdytojams.
- Vertinimas - vertinimo priemonės turi suteikti galimybę analizuoti verslo procesų veiklos rezultatų analizę, pagal vykdymo metu surinktą statistinę informaciją.

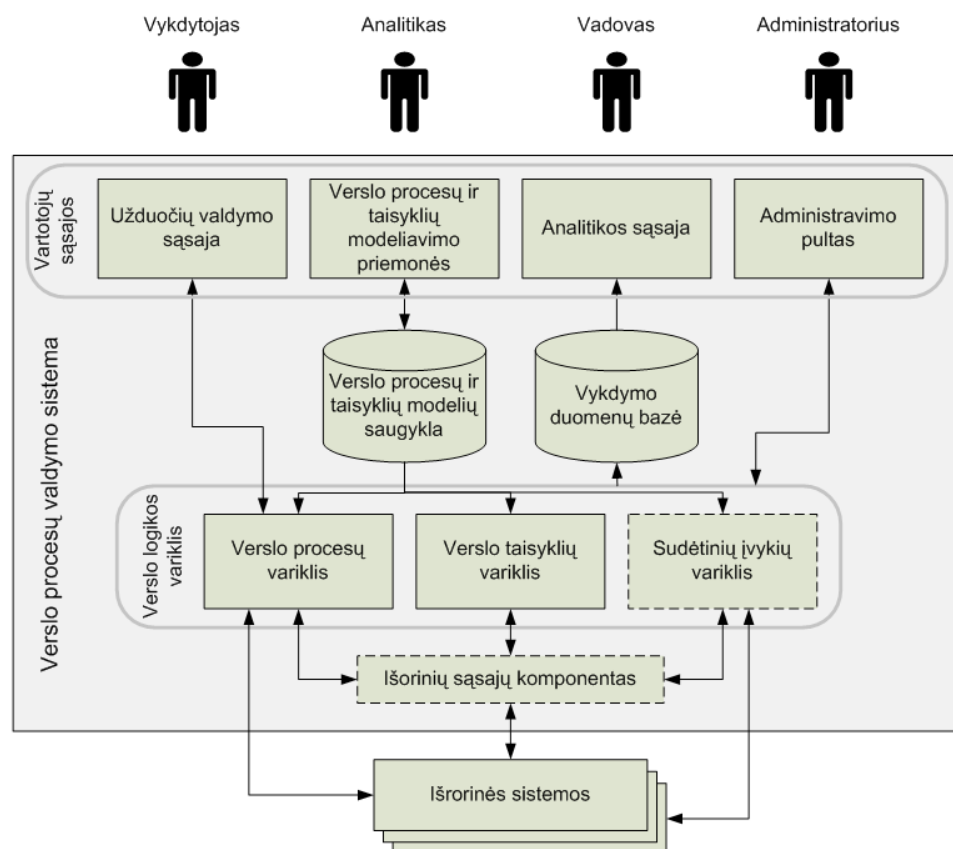
Taigi VPV sistemos automatizuoja verslo procesų gyvavimo ciklo veiklas ir tos veiklos apibrėžia reikalavimus sistemai. Pati VPV sistema gali būti sudaryta iš daugelio komponentų, tarpusavyje integruotų taip, kad atitiktų reikalavimus.

1.3.2. Verslo procesų valdymo sistemos architektūra

Verslo procesų valdymo sistema, tai tarpusavyje integruotų komponentų rinkinys.

Analizuojant literatūros šaltiniuose (Havey 2005; WPMC) aprašytus verslo procesų valdymo architektūrinius modelius, galima išskirti tokias pagrindines VPV sistemos komponentes: raginės modeliavimo priemonės, verslo procesų vykdymo variklis, rankinių užduočių sąrašo valdymo komponentas, procesų duomenų bazė, administravimo priemonės, komunikacijos su kitomis sistemomis priemonės. Reikia pastebėti, kad nagrinėjamuose architektūriniuose modeliuose yra minimas tik verslo procesų vykdymo variklis. Siekiant užtikrinti verslo taisyklėmis grindžiamą verslo procesų vykdymą VPV sistemoje turėtų būti įtrauktas verslo taisyklių variklis, bei sudėtinių įvykių variklis (Koehler 2010). Kadangi verslo taisyklių valdymo mokslinės idėjos vystėsi atskirai nuo verslo procesų, tai atitinkamai ir techninės priemonės buvo vystomos atskirai. Dėl savarankiškos verslo procesų ir verslo taisyklių automatizavimo priemonių raidos, jos yra prastai integruotos arba integruotos tik tam tikrose VPV gyvavimo ciklo fazėse.

Darbe (pav. 4) yra pateikiamas verslo VPV sistemos, apjungiančios verslo procesus, verslo taisykles, o taip pat ir sudėtinius įvykius, architektūrinis modelis. Šis modelis yra sudarytas praplečiant literatūroje (Havey 2005, WPMC) aprašytus architektūrinius modelius, verslo taisyklių ir sudėtinių įvykių variklių komponentams.



4 pav. Verslo procesų valdymo sistemos bendrinė architektūra

VPV sistemos architektūrą sudaro keturios komponentių grupės: vartotojų sąsajos, duomenų bazės, verslo logikos varikliai ir sąsajos su išorinėmis sistemomis.

Verslo procesų ir taisyklių modeliavimo priemonės leidžia grafiniu būdu projektuoti verslo modelius apjungiančius tiek verslo procesų, tiek verslo taisyklių modelius. Modeliavimo artefaktai yra patalpinami į centrinį repozitoriumą - verslo procesų ir taisyklių modelių saugyklą. Šioje architektūroje nėra išskiriamos procesų modelių įgyvendinimo komponentės, o jų funkciją atlieka modeliavimo priemonės, t.y. modeliavimo priemonėmis yra paruošiami tokie procesų modeliai, kurie gali būti vykdomi verslo logikos varikliuose.

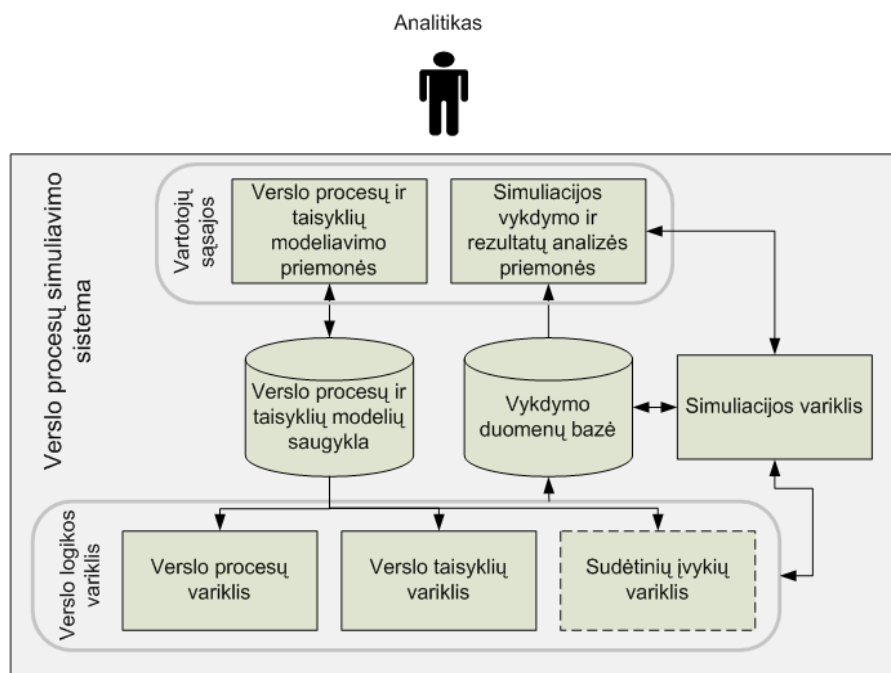
Verslo logikos variklis yra sudarytas iš trijų skirtingų sudedamųjų dalių. Geriausiu atveju šios trys dalys turėtų būti glaudžiai integruotos. Sudėtinių įvykių variklis analizuoja įvairius įvykius, tiek vidinius tiek išorinius ir identifikuoja prasmingus įvykius, kurie aktyvuoja taisykles arba verslo procesus. Taisyklių variklis vykdo verslo taisyklių rinkinius, o verslo procesų - verslo procesus. Verslo taisyklių ir verslo procesų modeliai bei sudėtinių įvykių šablonai yra užkraunami iš centrinės repozitorijos. Visos verslo logikos variklio komponentės, procesų ir taisyklių vykdymo metu, kaupia veiklos duomenis specialioje duomenų bazėje, kurie vėliau gali būti analizuojami siekiant įvertinti procesų našumo rodiklius.

Kadangi ne visus procesus pavyksta pilnai automatizuoti, tai VPV sistema turi turėti vartotojo sąsają leidžiančią valdyti neautomatizuotas užduotis. Taip pat yra išskiriama ir administravimo panelė, kuri leidžia stebėti sistemos būseną, klaidas ir našumą.

Apibendrinant galima teigti, kad VPV susideda iš tam tikrų komponentių, kurios, priklausomai nuo sprendžiamų uždavinių, gali būti neprivalomos. Reikia pastebėti, kad šiame architektūros modelyje nėra parodytų simuliacijos uždaviniams spręsti būdingų komponentių. VP simuliacijos sistemos architektūra pateikiama kitame skyriuje.

1.3.3. Verslo procesų simuliacijos sistemos architektūra

Verslo procesų simuliacija yra vienas iš efektyviausių būdų įvertinti naujai diegiamus arba keičiamus verslo procesus. 5 pav. yra pateikta verslo taisyklėmis grindžiamų verslo procesų simuliacijos priemonės architektūra. Ši architektūra yra paremta 1.3.2 skyriuje aprašyta taisyklėmis grindžiamų verslo procesų vykdymo sistemos architektūra.



5 pav. Verslo procesų simuliacijos sistemos architektūra

Simuliacijos sistemos, lyginant su vykdymu, vartotojai yra tik verslo procesų analitikai. Todėl architektūroje yra pavaizduotos tik dvi vartotojo sąsajos reikalingos procesų analitiko darbui su verslo procesais, t. y. verslo procesų ir taisyklių modeliavimo priemonės, bei simuliacijos vykdymo ir rezultatų analizės priemonės. Kadangi, vykdant verslo proceso simuliaciją, neautomatizuotos užduotys arba kreipimaisi į išorines sistemas nėra iš tikrųjų atliekami, o jų atlikimas yra imituojamas remiantis tam tikrais matematiniais-statistiniais dėsniais, tai užduočių valdymo vartotojo sąsajos bei sąsajos su išorinėmis sistemomis yra nereikalingos. Tam, kas būtų galima vykdyti procesų simuliaciją, reikalingas simuliacijos variklis. Simuliacijos variklis turi užtikrinti verslo proceso aplinkos imitavimą.

Pagrindinis šios architektūros privalumas yra tas, kad simuliacijai yra naudojami tie patys verslo procesų ir verslo taisyklių modeliai, kurie naudojami ir vykdymui. Tai gerokai supaprastina simuliacijos procesą. Žinoma, kad būtų galima vykdyti simuliaciją, yra būtini papildomi duomenys apibūdinantys aplinkos, kurioje veikia verslo procesas, elgseną.

2. TAISYKLĖMIS GRINDŽIAMŲ VERSLO PROCESŲ SIMULIACIJOS PRIEMONĖ

2.1. Architektūra

2.1.1. Tikslai ir principai

Įgyvendinant taisyklėmis grindžiamų verslo procesų simuliacijos priemonę buvo siekiama pagrindinio tikslo - atvirojo kodo programinės įrangos pagrindu sukurti taisyklėmis grindžiamų verslo procesų simuliacijos sprendimą, gebantį simuliuoti vykdymui suprojektuotus ir nepakeistus verslo procesų ir taisyklių modelius, papildant juos simuliacijai būtina informacija.

Atviro kodo programinės įrangos pasirinkimas simuliacijos sprendimui grindžiamas tuo, kad:

- atvirojo kodo programinė įranga yra prieinama visiems ir taip sudaromos prielaidos simuliacijos priemonės tolimesniam plėtojimui.
- atvirojo kodo programinė įranga gali būti keičiama, o tai palengvina jos panaudojimą sprendžiant taisyklėmis grindžiamų verslo procesų simuliacijos problemą.

Siekiant simuliuoti taisyklėmis grindžiamus verslo procesus, panaudojant atviro kodo programinę įrangą, yra naudojama tam tikra simuliacijos priemonių architektūra, todėl bet kuri simuliacijos priemonė turi tenkinti pagrindinius verslo taisyklėmis grindžiamų verslo procesų simuliacijos priemonės architektūros principus, kurių pagalba yra apibūdinamos pradinės verslo taisyklėmis grindžiamų verslo procesų simuliacijos sąlygos:

- kuriant taisyklėmis grindžiamų verslo procesų simuliacijos priemonę turi būti naudojamos verslo procesų ir verslo taisyklių vykdymo mašinos;
- vykdymui paruoštų simuliuojamų verslo procesų ir verslo taisyklių modeliai turi būti baigtiniai ir nebekeičiami;
- kadangi procesų ir taisyklių modeliai gali būti naudojami tiek vykdyme, tiek ir simuliacijoje, svarbu simuliacijos modelį (konfigūraciją), kuriame yra saugomi papildomi simuliacijos vykdymui reikalingi duomenys, atskirti nuo verslo taisyklių ir verslo procesų modelių;
- simuliacijos priemonės įgyvendinimui pagrįstai naudojama tik atvirojo kodo programinė įranga, kurios licencinės nuostatos neapriboja laisvės tolimesniam sprendimo plėtojimui, kitiems asmenims ar įmonėms, teikiant prioritetą liberalioms licencijoms, tokioms kaip ASL ir LGPL;

- taisyklėmis grindžiamų verslo procesų simuliacijos priemonė turi būti atvirojo kodo ir platinama su ASL licencija;
- JAVA technologijos turi būti naudojamos taisyklėmis grindžiamų verslo procesų simuliacijos priemonei kurti.

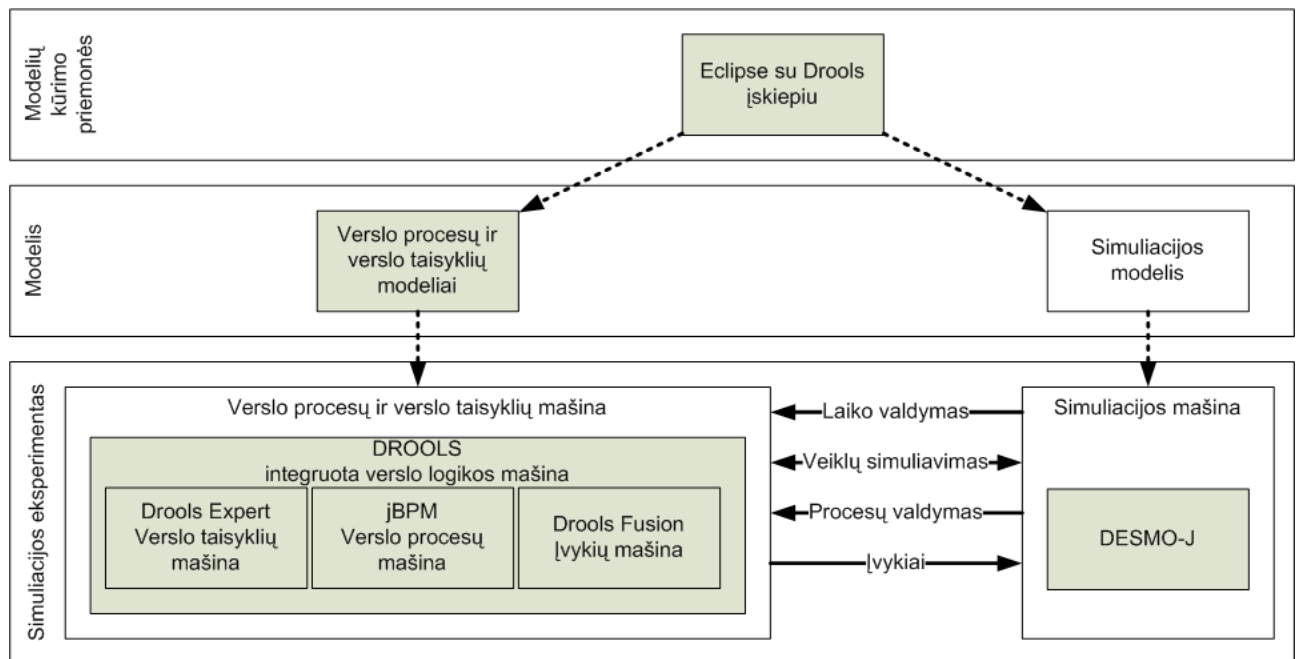
Verslo taisyklėmis grindžiamų verslo procesų simuliacijos priemonės architektūros principų išskyrimas sudaro pakankamas sąlygas sudaryti pagrindinę simuliacijos priemonės architektūrą, kurios pagalba programų sistemų požiūriu yra apibrėžiama verslo taisyklėmis grindžiamų verslo procesų simuliacijos priemonė, nustatomos sudedamosios dalys bei jų tarpusavio ryšiai.

2.1.2. Komponentės ir struktūra

Taisyklėmis grindžiamų verslo procesų simuliacijos priemonės architektūra yra sudaryta iš modelių kūrimo priemonių, sukurtų modelių (konfigūracijų) bei simuliacijos eksperimentų (6 pav). Pateikta simuliacijos priemonių architektūra visiškai atitinka 1.3.3 skyriuje išsamiai išanalizuotą ir aprašytą verslo procesų simuliacijos sistemos architektūrą.

Kaip parodyta 6 paveiksle, pasirinktos priemonės architektūros pagrindą sudaro programinė įranga DROOLS ir DESMO-J, detaliau aprašyta atitinkamai 2.2.1 ir 2.2.2 skyriuose. Svarbu paminėti, kad pasirinkta programinė įranga atskirai neturi taisyklėmis grindžiamų verslo procesų simuliacijos galimybių. DROOLS yra verslo logikos (taisyklių, procesų ir įvykių) vykdymo mašina, o DESMO-J diskrečiųjų įvykių simuliacijos mašina. Abu programinės įrangos paketai yra atvirojo kodo, todėl remiantis pagrindiniais architektūriniais principais gali būti pritaikyti darbe nagrinėjamai problemai spręsti. DROOLS tenkina būtinąsias simuliacijai apibrėžtas sąlygas, nes joje yra įdiegtas virtualus laikrodis, kuris gali būti naudojamas simuliacijos metu laiko kontrolei vykdyti. DESMO-J taip pat turi reikiamas priemones laiko sinchronizavimui. DESMO-J simuliacijos laikrodis turi galimybę pranešti kitoms sistemos dalims apie besikeičiantį simuliacijos laiką iškvičiant tam tikrą metodą.

Simuliacijos priemonės architektūroje (6 pav.), siekiant pademonstruoti, kurios programų sistemų komponentės panaudotos, o kurios sukurtos, komponentės tikslingai pažymėtos skirtingomis spalvomis: pilka spalva yra žymimos panaudotos DROOLS ir DESMO-J komponentės, o balta spalva žymimos tos architektūrinės komponentės, kurios buvo sukurtos atliekant darbą. Svarbu paminėti, kad pilkais keturkampiais pavaizduotos komponentės yra novatoriškai panaudotos laisvai prieinamos ir nemokamai platinamos atvirojo kodo bibliotekos.



6 pav. Taisyklėmis grindžiamų verslo procesų simuliacijos priemonės architektūra

Analizuojant verslo taisyklėmis grindžiamų verslo procesų simuliacijos priemonės architektūros simuliacijos eksperimento lygmeniui priskiriama integruota verslo logikos mašina yra sudaryta iš verslo taisyklių, verslo procesų ir įvykių mašinų. Pagrindinė DROOLS integruotos verslo logikos mašinos paskirtis yra verslo procesų, verslo taisyklių ir sudėtinių įvykių vykdymas (detalesnė aprašyta 2.2.1). Verslo procesų ir taisyklių simuliacijos funkcionalumo DROOLS standartiškai neturi, tačiau, kaip minėta, turi simuliacijai būtiną laiko kontrolę. DROOLS funkcionuoja virtualaus laiko valdymas, o tai reiškia, kad programinėmis priemonėmis nepriklausomai nuo realaus laiko gali būti kontroliuojamas verslo logikos mašinos laikas. Tokia DROOLS savybė leidžia panaudoti ją kaip taisyklėmis grindžiamų verslo procesų simuliacijos priemonės realizacijos komponentę, kurią simuliacijos metu valdo simuliacijos mašina.

DROOLS integruotos verslo logikos mašinos panaudojimas simuliacijos priemonės realizacijoje leidžia simuliacijos metu naudoti tuos pačius vykdymui paruoštus verslo procesų ir verslo taisyklių modelius, papildomai modelį papildant simuliacijai reikalingais duomenimis. Toks DROOLS integruotos verslo logikos mašinos panaudojimo būdas verslo taisyklėmis grindžiamų verslo procesų simuliacijai atitinka pagrindinius architektūrinius principus ir sumažina klaidingų simuliacijos rezultatų tikimybę, nes simuliacijai nereikia kurti visiškai atskiro simuliacijos modelio.

Verslo procesų ir verslo taisyklių mašina yra DROOLS integruotos verslo logikos mašinos plėtinys, kuriame yra realizuojama simuliacijai charakteringa logika. Verslo proceso vykdymo metu, verslo procesas turi nuolat sąveikauti su aplinka. Pavyzdžiui, vykdant verslo procesą gali būti identifikuotas poreikis jungtis prie išorinių sistemų, kuriose yra saugomi

duomenys, siųsti el. pašto žinutes, registruoti užduotis vartotojų užduočių valdymo sistemoj ar atlikti kitokias funkcijas. Verslo taisyklėmis grindžiamų verslo procesų simuliacijos metu simuliuojamo proceso aplinka atitinkamai taip pat turi būti simuliuojama. Vadinas, atsiranda nepageidautino prisijungimo prie išorinių sistemų simuliacijos metu tikimybė, o tai reiškia, kad simuliacijos metu reikalinga užtikrinti, jog simuliuojamas procesas nesusijungs prie išorinių sistemų, nesiuntinės pranešimų ir pan. Būtent dėl tokių priežasčių, verslo procesų ir verslo taisyklių mašina realizuota, kaip DROOLS integruotos verslo logikos mašinos plėtinys, užtikrina pakankamą komunikaciją su simuliacijos mašina ir apriboja komunikaciją su išore simuliacijos metu.

DESMO-J yra diskrečiųjų įvykių simuliacijos karkasas, realizuotas JAVA programavimo kalba. Detaliau DESMO-J yra aprašytas 2.2.2 skyriuje.

Simuliacijos mašina atlieka verslo procesų ir verslo taisyklių mašinos valdymo simuliacijos metu funkcijas. Simuliacijos mašinos pagrindą sudaro DESMO-J karkasas. DESMO-J yra universalus diskrečiųjų įvykių karkasas, leidžiantis JAVA kalba sukurti įvairius simuliacijos modelius ir atlikti jų simuliaciją. Simuliacijos mašina, naudodama DESMO-J, vykdo taisyklėmis grindžiamų verslo procesų simuliaciją, panaudodama verslo procesų ir taisyklių mašiną. Simuliacijos mašina nepriklauso nuo verslo procesų ir verslo taisyklių mašinos realizacijos tol, kol verslo procesų ir verslo taisyklių mašina realizuojama standartinė sąsaja. Tarp simuliacijos mašinos bei verslo procesų ir verslo taisyklių mašinos turi būti sinchronizuojamas laikas, procesų veiklų simuliacijos, procesų vykdymo inicijavimas bei apsikeitimas įvykiais. Taip pat simuliacijos mašina supaprastina simuliacijos modelio sukūrimą.

Verslo procesų ir verslo taisyklių modeliai, tai standartinėmis DROOLS palaikomomis priemonėmis sukurti modeliai. DROOLS palaiko BPMN2 standartą, bei galimybę aprašyti verslo taisykles naudojant intuityviai suprantamą taisyklių kūrimo vedlį.

Simuliacijos modelis, tai specialiai taisyklėmis grindžiamų verslo procesų simuliacijai aprašyti sukurtos JAVA klasės, kurios leidžia paprastai kurti simuliacijos modelius. JAVA klasių naudojimas simuliacijos modelių kūrime nėra statinis perspektyvos atžvilgiu, todėl perspektyvoje simuliacijos modeliams kurti analizuojama komponentė gali būti plečiama, kad būtų kitomis priemonėmis sukurti verslo taisyklėmis grindžiamų verslo procesų modeliai. Pavyzdžiui, kad būtų palaikomi XML standartu aprašyti simuliacijos modeliai.

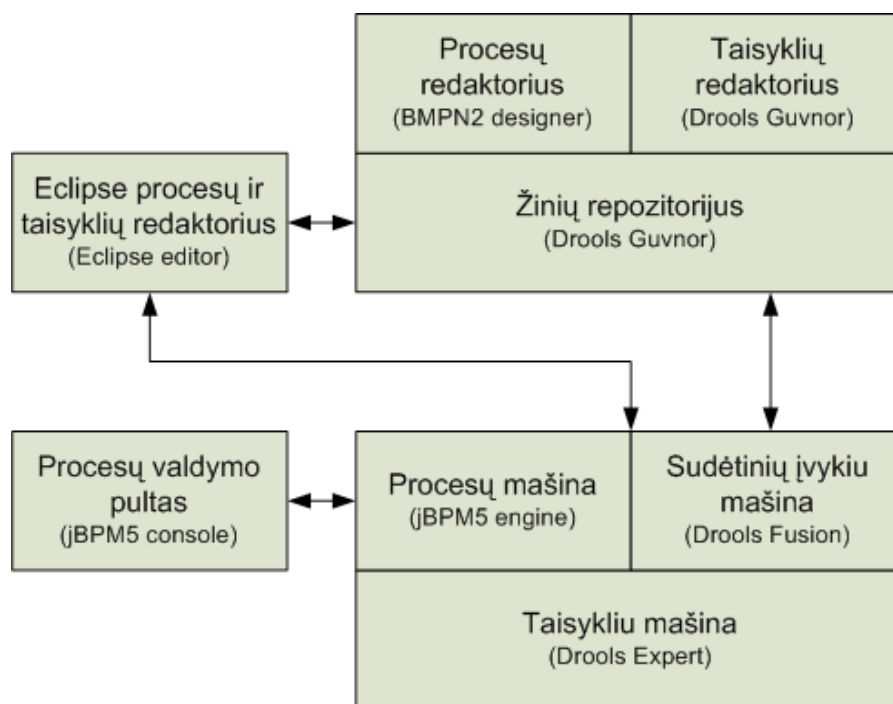
Vadovaujantis pagrindiniais simuliacijos priemonės architektūros principais bei struktūriniais reikalavimais, **modelių kūrimo priemone** pasirinkta ECLIPSE programinė priemonė su DROOLS įskiepiu, kurios pagalba yra sudaromi verslo taisyklėmis grindžiamų verslo procesų ir pačių verslo taisyklių modeliai bei visiškai atskirai nuo jų yra aprašoma modelio konfigūracija, turinti savyje visus papildomai reikalingus duomenis simuliacijai įvykdyti (6 pav). ECLIPSE su

DROOLS įskiepiu, tai standartinė simuliacijos kūrimo ir vykdymo priemonė.

2.2. Pasirinkta programinė įranga

2.2.1. DROOLS integruotos verslo logikos mašina

DROOLS - tai verslo logikos integravimo aplinka, kuri sukuria vieningą ir integruotą taisyklių, procesų ir įvykių apdorojimo platformą. DROOLS nuo pat pradžių buvo suprojektuota laikantis principo, kad taisyklės, procesai ir įvykiai be jokių kompromisų yra atskiros sąvokos (drools.org). DROOLS yra platinama su liberalia “Apache Software License” licencija, todėl nėra jokių esminių apribojimų tolimesniam sprendimų, veikiančių DROOLS pagrindu, plėtojimui ir panaudojimui.



7 pav. DROOLS ir jBPM architektūra

DROOLS logiškai gali būti padalinta į dvi pagrindines dalis. Viena dalis, kuri toliau sąlygiškai vadinama vykdomąja dalimi, yra atsakinga už verslo logikos išpildymą, kita, toliau vadinama kuriamąja dalimi, už logikos aprašymą arba jos kūrimą.

7 pav. yra pateikiama aukšto lygio Drools ir jBPM architektūra, kurioje pavaizduotos pagrindinės komponentės:

- vykdomosios: taisyklių mašina, procesų mašina, sudėtinių įvykių mašina, procesų

valdymo pultas;

- kuriamosios: žinių repozitorija, taisyklių redaktorius, procesų redaktorius, ECLIPSE procesų ir taisyklių redaktorius.

Taisyklių mašinoje („Drools Expert“ komponentė) yra vykdomos taisyklės. Priklausomai nuo to, kaip taisyklių mašina yra integruojama į programų sistemą, taisyklės gali būti užkraunamos iš centralizuotos žinių repozitorijos arba iš specialių laikmenų, saugomų kartu su programų sistema.

Sudėtinių įvykių mašina („Drools Fusion“ komponentė) apdoroja įvykių srautus ir pagal apibrėžtas taisykles identifikuoja svarbius verslo įvykius. Sudėtinių įvykių mašina veikia taisyklių mašinos pagrindu ir turi papildomą sudėtinių įvykių aptikimo funkcionalumą, kuris aprašomas taisyklėmis. Įvykių aptikimo taisyklės, analogiškai taisyklių variklio atvejui, gali būti užkraunamos tiek iš centrinės repozitorijos, tiek iš specialių laikmenų.

Procesų mašina („jBPM5 engine“ komponentė) yra verslo procesų vykdymo mašina. Ji taip pat veikia taisyklių variklio pagrindu ir išnaudoja taisyklių mašinoje realizuotą taisyklių sekų (angl. *rules flow*) funkcionalumą. Nors jBPM5 verslo procesų vykdymo mašina yra laikoma BPMN2 verslo procesų vykdymo mašina, tačiau egzistuoja esminis jBPM5 ir BPMN2 technologinis skirtumas. jBPM5 yra laikomas BPMN2 procesų vykdymo mašina, tačiau į procesų mašiną BPMN2 procesų modeliai yra užkraunami ne tiesiogiai. BPMN2 standarto modeliai iš pradžių yra konvertuojami į POJO (angl. *Plain Old Java Objects*) objektus ir tik tada užkraunami į procesų variklį. Tai leidžia, realizavus konverterius, panaudoti įvairias procesų kalbas, tokias kaip jPDL, BPEL ir kitas.

Procesų valdymo pultas („jBPM5 console“ komponentė) - tai papildoma jBPM5 komponentė, kuri leidžia kontroliuoti vykdomus procesus. Procesų valdymo pulte yra realizuotas vartotojo užduočių (angl. *Human Task*) palaikymo funkcionalumas.

Žinių repozitorija („Drools Guvnor“ komponentė) yra skirta verslo procesų modelių, verslo taisyklių, įvykių aptikimo taisyklių centralizuotam saugojimui. Repozitorijoje yra saugomi tiek pradiniai modeliai (BPMN2, DRL), tiek ir transformuoti į POJO modeliai. Tai leidžia į verslo procesų, verslo taisyklių ar įvykių mašinas užkrauti jau sukompilijuotas ir paruoštas žinias arba kitaip vadinamą verslo logiką.

Taisyklių redaktorius („Drools Guvnor“ komponentė) - tai taisyklių kūrimo aplinka, sukurta žiniatinklio technologijų pagrindu. Taisyklių redaktoriaus pagalba aprašytos taisyklės vėliau yra patalpinamos centrinėje repozitorijoje. Taisyklių redaktorius suteikia galimybę aprašyti verslo taisykles naudojant taisyklių kūrimo vedlį, kuris labai supaprastina taisyklių kūrimą ir jų keitimą bei yra intuityviai suprantamas. Toks taisyklių redaktoriaus funkcionalumas sukuria pakankamai paprastą taisyklių valdymo aplinką specialių techninių žinių neturintiems asmenims.

Procesų redaktorius („BPMN2 designer“ komponentė), kaip ir taisyklių redaktorius, yra procesų modelių kūrimo aplinka. Tai žiniatinklio paslaugomis grindžiama priemonė, kurios pagrindas yra „Oryx-editor“ projektas. Procesų redaktorius yra integruotas į „Drools Guvnor“ komponentę. Verslo procesų modeliai, tiek BPMN2 formatu, tiek transformuoti į POJO formatą, yra saugomi žinių repozitorijoje.

ECLIPSE procesų ir taisyklių redaktorius („Eclipse editor“ komponentė) - tai integruotos kūrimo aplinkos ECLIPSE plėtinys, leidžiantis kurti taisyklių ir procesų modelius, stebėti jų vykdymą ECLIPSE aplinkoje. Taisyklės ir procesų modeliai gali būti saugomi centrinėje žinių repozitorijoje arba tiesiogiai panaudojami programų sistemų kūrime.

Aprašytos sistemos dalys - tai tik pagrindinės komponentės, kurios gali būti skaidomos dar į smulkesnes komponentes. DROOLS ir jBPM yra modulinė (komponentizuota) sistema ir egzistuoja daug scenarijų kaip komponentės gali būti panaudotos, kuriant procesais, įvykiais ir taisyklėmis grindžiamas programų sistemas. Iš kitos pusės DROOLS ir jBPM yra glaudžiai integruotos ir suteikia galimybę išnaudoti tiek procesų ir taisyklių privalumus realizuojant reikiamą verslo logiką.

2.2.2. DESMO-J diskrečiųjų įvykių simuliacijos variklis

DESMO-J (*Discrete Event Simulation and Modeling in Java*) - tai atvirojo kodo diskrečiųjų įvykių simuliacijos objektinis karkasas. DESMO-J turi dvi pagrindines savybes [Desmo1]:

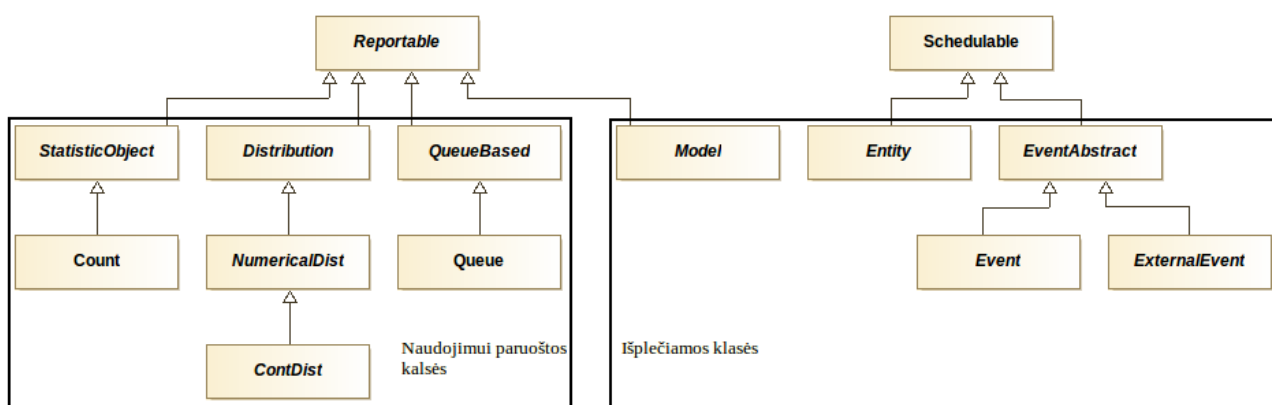
- DESMO-J palaiko diskrečiųjų įvykių simuliacijos paradigmą. Diskrečiųjų įvykių modeliuose, sistemos būsenų pokyčiai vyksta tam tikru (diskrečiu) laiko momentu. Tarpuose tarp įvykių sistemos būsena nesikeičia.
- DESMO-J yra realizuota JAVA programavimo kalba.

DESMO-J karkasas buvo sukurtas Hamburgo universitetui aktyviai dalyvaujant simuliacijos priemonių įvairiomis programavimo kalbomis kūrime. DESMO-J karkaso pirmtaku buvo 1989 m. sukurtas DESMO diskrečiųjų įvykių simuliacijos karkasas, parašytas naudojant „Modula-2“ programavimo kalbą. 1999 m. buvo išleista pirmoji DESMO-J versija, kuri iš esmės buvo JAVA kalba perrašyta DESMO karkaso versija. Iki šių dienų DESMO-J projektas išlieka aktyviu: nuolat yra taisomos klaidos, kuriamas papildomas funkcionalumas, atnaujinama susijusi dokumentacija. Per ilgesnį nei dešimties metų DESMO-J plėtojimo laikotarpį buvo sukurtas stabilus, kokybiškas ir gerai dokumentuotas diskrečiųjų įvykių karkasas.

DESMO-J yra platinamas su “Apache 2.0” licencija. Tai leidžia šį karkasą naudoti įvairiuose projektuose, tiek atviro kodo, tiek uždaro kodo komerciniuose projektuose. DESMO-J karkasas

JAVA praplečia konstruktais, kurie ženkliai supaprastina simuliacijos modelio kūrimą. DESMO-J modeliui suteikia [Desmo2]:

- paruoštas naudojimui JAVA klases, kurios realizuoja įvairius diskrečiųjų įvykių simuliacijos paradigmos konceptus, tokius kaip eilės, tikimybiniai skirstiniai, kurių pagrindą sudaro atsitiktinių skaičių generatoriai, duomenų kaupikliai (angl. *data collector*).
- abstrakčias JAVA klases, kurias pats modeliotojas turi praplėsti, realizuodamas specifinį modelio funkcionalumą. Tokiomis klasėmis yra realizuojami simuliacijos modelio (angl. *model*), esybės (angl. *entity*), įvykio (angl. *event*) konceptai.
- paruoštas naudoti JAVA klases, realizuojančias simuliacijos infrastruktūrą, sudarytą iš įvykių tvarkaraščio (angl. *scheduler*), įvykių sąrašų, simuliacijos laikrodžio. Visos šios klasės yra enkapsuliuotos į eksperimento klasę, kurioje taip pat yra realizuotas simuliacijos valdymo ir simuliacijos ataskaitų generavimo funkcionalumas.



8 pav. Supaprastinta DESMO-J klasių diagrama

8 pav. yra pavaizduota supaprastinta DESMO-J klasių diagrama, kurioje pademonstruotos kelios pagrindinės modeliotojo išplečiamos bei naudojimui paruoštos klasės, realizuojančios tam tikrus simuliacijos modelio konceptus. Pavyzdžiui apibrėžiant įvykius reikia specializuoti „Event“ arba „ExternalEvent“ klases, kuriose turi būti realizuotas esybių būsenos keitimo algoritmas. Kuriant simuliacijos modelį galima naudoti ir pilnai paruoštas klases, pavyzdžiui eiles, kuriose galima saugoti esybes.

2.2.3. Kūrimo priemonės

Taisyklėmis grindžiamų verslo procesų simuliacijos priemonės plėtojimui, remiantis

apibrėžtais pagrindiniais architektūriniais principais, aprašytais 2.1.1 skyriuje, siekiant sudaryti kuo didesnes priemonės plėtojimo galimybes, buvo pasirinkti atvirojo kodo ir nemokamai platinami įrankiai, veikiantys įvairiose platformose (Linux, Windows, OS-X) ir atitinkantys aprašytos architektūros (2.1.2 skyrius) reikalavimus.

Kūrimo priemonės išeities kodo kompiliavimui, testų vykdymui, priklausomybių valdymui pasirinktas „Apache Maven“¹ (toliau – MAVEN) įrankis. Tai vienas iš populiariausių įrankių JAVA taikymo srityje, leidžiantis automatizuoti programinės įrangos kūrimo ciklą. Jame išeities kodo (versijų) kontrolei panaudota paskirstytoji kodo kontrolės sistema GIT², o viešos GIT repozitorijos prieglobai pasirinktos „Google Code“ paslaugos.

Kodo rašymui gali būti pasirinkta bet kokia JAVA kalbą palaikanti integruota kūrimo aplinka. Darbo metu plėtojamos taisyklėmis grindžiamų verslo procesų simuliacijos priemonės kūrimui buvo naudojama „Eclipse IDE“³, o priemonės plėtojimui darbo metu buvo naudojami „Jenkins“⁴ nuolatinio integravimo (kompiliavimo automatizavimo) bei „Sonar“⁵ kodo kokybės užtikrinimo įrankiai.

Pasirinkti įrankiai palaiko įvairias platformas ir visi kartu sukuria efektyvią JAVA programinės įrangos kūrimo aplinką.

2.3. Realizacijos aprašymas

2.3.1. Projektas ir jo struktūra

Taisyklėmis grindžiamų verslo procesų simuliacijos priemonės įgyvendinimui buvo sukurtas atvirojo kodo projektas RPSIM (angl. **Rule based Process Simulation**). Darbo metu projektas buvo patalpintas „Google Code“ atvirojo kodo projektų prieglobos svetainėje⁶, kurioje gali būti randamas ir projekto išeities kodas. Projekto svetainė turi visas būtinas priemones tolimesniam priemonės plėtojimui. Toliau šiame darbe kalbant apie taisyklėmis grindžiamų verslo procesų simuliacijos priemonę, realizuotą šiame darbe, ar jos įgyvendinimo projektą, bus naudojamas trumpinys RPSIM. Šiame skyriuje yra aprašoma taisyklėmis grindžiamų verslo procesų simuliacijos priemonės realizacijos projektas.

Analizuojant taisyklėmis grindžiamų verslo procesų simuliacijos priemonių architektūrą (skyrius 2.1.2), buvo apibrėžtos dvi pagrindinės RPSIM komponentės: verslo logikos mašina ir

1 <http://maven.apache.org/>

2 <http://git-scm.com/>

3 <http://www.eclipse.org/>

4 <http://jenkins-ci.org/>

5 <http://www.sonarsource.org/>

6 RPSIM projekto svetainė: <http://code.google.com/p/rpsim-bumbis-lt/>

simuliacijos mašina. Projektuojant RPSIM buvo laikomasi idėjos, kad verslo logikos mašina gali būti realizuota įvairiomis priemonėmis, panaudojant įvairius sprendimus, nebūtinai DROOLS ir jBPM priemonių pagalba, todėl buvo apibrėžta verslo logikos vykdymo mašinos sąsaja (angl. *interface*), kurią turi realizuoti verslo logikos mašina, kad būtų įmanoma atlikti simuliaciją (2.3.4 skyrius). Atitinkamai darbe išanalizuotoms apibrėžtims, dekompozicijos būdu, RPSIM projektas buvo suskaidytas į du mažesnius sudėtinius projektus:

- RPSIM-CORE – projektas, kuriame apibrėžiamos visos reikiamos sąsajos, realizuojamas simuliacijos variklis bei visi simuliacijos elementai;
- RPSIM-DROOLSJBPM – projektas, kuriame realizuojamas taisyklėmis grindžiamų verslo procesų vykdymo variklis, kurio pagrindą sudaro DROOLS ir jBPM, bei yra pritaikytas simuliacijos vykdymui.

Kiekviename iš sudėtinių projektų visa realizacija yra papildomai suskaidyta į paketus. Pagal paskirtį ir funkcijas kiekvienas paketas savyje apjungia tam tikras klases., todėl paketai yra dar klasifikuojami pagal apjungiamų klasių funkcionalumą:

- *lt.bumbis.rpsim.core* – klasės realizuojančios simuliacijos variklį, bei sąsajas;
 - *lt.bumbis.rpsim.core.entities* – klasės realizuojančios diskrečiųjų įvykių simuliacijos esybes;
 - *lt.bumbis.rpsim.core.events* – klasės realizuojančios diskrečiųjų įvykių simuliacijos įvykius;
 - *lt.bumbis.rpsim.core.simconfig* – klasės, reikalingos simuliacijos konfigūracijos realizacijai;
 - *lt.bumbis.rpsim.core.exceptions* – išimčių klasių paketas;
- *lt.bumbis.rpsim.droolsjbpm* – DROOLS ir jBPM pagrindu realizuojančios verslo logikos variklį klasės.

Darbo metu, atliekant taisyklėmis grindžiamų verslo procesų simuliacijos priemonės realizaciją, t. y. RPSIM projektą, iš viso buvo sukurtos 49 klasės, 210 metodų, 110 klasių parametrų prieigos metodų, 43 testai (testai padengia 70,5% kodo). Daugiau informacijos apie projektą galima rasti 1 priede, kuriame yra pateikiamas SONAR kodo kokybės sistemos RPSIM projekto informacijos lango vaizdas.

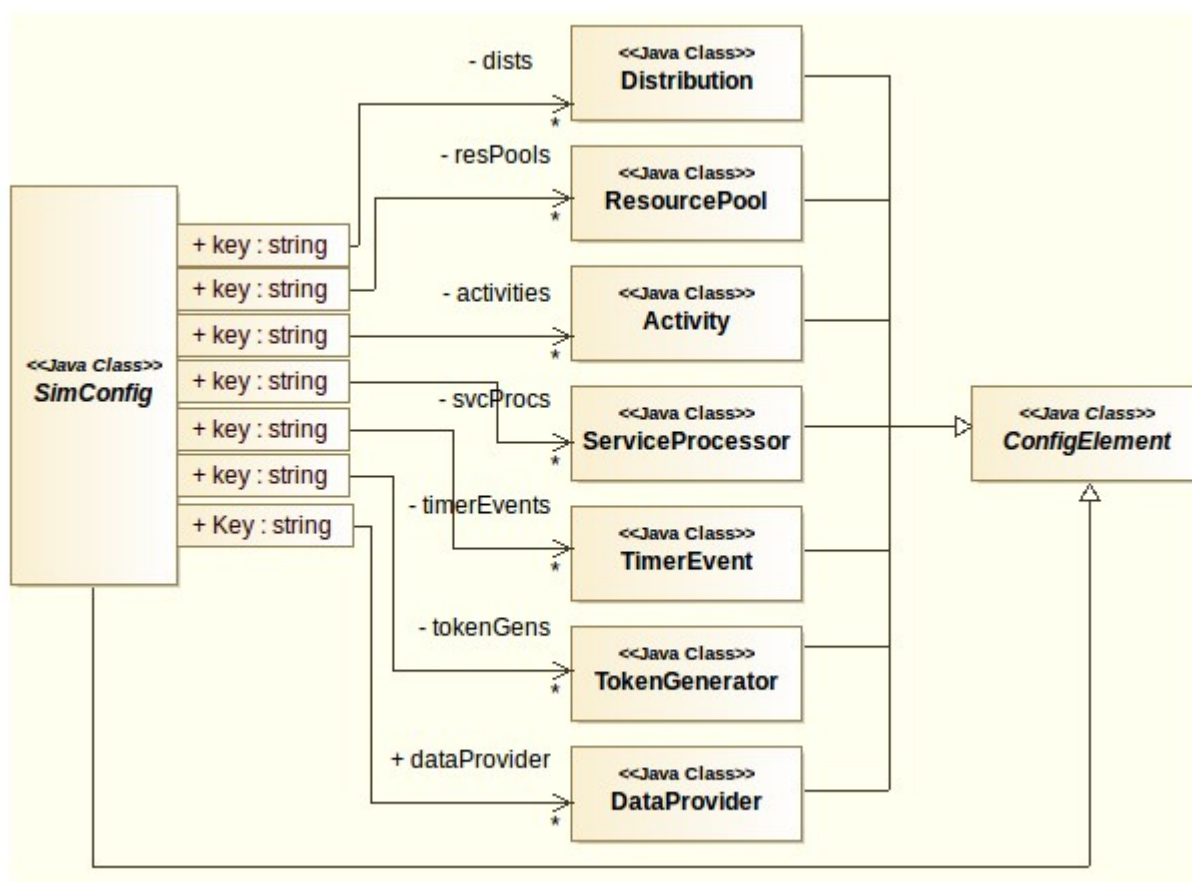
Pagal darbo metu suformuluotus uždavinius, visas RPSIM projektas gali būti sudarytas iš trijų pagrindinių sudėtinių dalių: simuliacijos konfigūracijos, simuliacijos variklio, verslo logikos variklio projektų. Remiantis šiomis trimis logiškais dalimis sekančiuose poskyriuose bus pateikiamas detalus RPSIM projekto realizacijos aprašymas.

2.3.2. Simuliacijos konfigūracija

Projektuojant RPSIM architektūrą, pradiniam projektavimo išeities taške apibrėžiant pradines sąlygas, nustatant apribojimus, buvo apibrėžta, kad:

- simuliacijai turi būti naudojami vykdymui paruošti verslo procesų modeliai bei verslo taisyklės;
- simuliacijai reikalingi duomenys turi būti aprašomi atskirai simuliacijos modelyje.

Realizuojant RPSIM buvo sukurtas atskiras paketas „*lt.bumbis.rpsim.core.simconfig*“ ir atitinkančios simuliacijos konfigūracijos elementus klasės.



9 pav. Simuliacijos konfigūracijos elementus atitinkančių klasių diagrama

Simuliacijos konfigūracijos klasių diagrama pateikiama 9 paveikslėlyje. Pakete „*lt.bumbis.rpsim.core.simconfig*“ iš viso yra 9 klasės:

- „*ConfigElement*“ – tai bazinė abstrakti konfigūracijos elemento klasė, kurią specializuoja, paveldi iš jos visus atributus ir metodus visos paketo klasės.
- „*SimConfig*“ – tai klasė, realizuojanti simuliacijos konfigūraciją. Ši klasė inkapsuliuoja

visus simuliacijos konfigūracijos elementus.

- „*Distribution*“ - tai klasė, atitinkanti tikimybinį skirstinį, kuris gali būti naudojamas procesų paleidimo ar veiklų įvykdymo dėsningumo apibrėžimui.
- „*ResourcePool*“ - tai klasė, atitinkanti išteklių grupes. Išteklių grupės yra priskiriamos veikloms. Simuliuojant procesų veiklų atlikimą yra naudojami ištekliai iš išteklių grupės.
- „*Activity*“ - tai klasė, atitinkanti proceso veiklą.
- „*ServiceProcessor*“ - tai klasė, atitinkanti loginio simuliacijos elemento, kuris yra naudojamas simuliuojant veiklų atlikimą, konfigūraciją. Kiekvienai proceso veiklai yra priskiriamas ją aptarnaujantis procesorius.
- „*TokenGenerator*“ - tai klasė, atitinkanti loginio simuliacijos elemento, kuris paleidžia verslo procesus simuliacijos metu, konfigūraciją.
- „*DataProvider*“ - tai klasė, apibrėžianti duomenų generatoriaus konfigūraciją. Duomenų generatoriai gali būti naudojami procesų kintamųjų reikšmių apibrėžimui, proceso startavimo metu.

Simuliacijos modelio kūrimo metu, jo konfigūracija yra aprašoma atskiroje klasėje, kuri specializuoja „*SimConfig*“ klasę ir realizuoja šios klasės abstraktų metodą „*configure*“. Simuliacijos konfigūracijos pavyzdys pavaizduotas 10 paveiksle.

```
public class SimulationConfiguration extends SimConfig {  
  
    @Override  
    public void configure() {  
        name("Demo simulation");  
        add(new TokenGenerator().name("TG")  
            .process("lt.bumbis.rpsim.demo.requestHandling")  
            .dist("DistArrival").timeUnit(TimeUnit.MINUTES)  
            .dataProvider("DP").useContextData(true));  
        // Distributions  
        add(new Distribution().name("DistArrival").distClass(ContDistExponential.class).distParams(20.0));  
        add(new Distribution().name("DistAuto").distClass(ContDistUniform.class).distParams(5.0, 2.0));  
        add(new Distribution().name("DistHT").distClass(ContDistUniform.class).distParams(10.0, 7.0));  
        add(new Distribution().name("DistHT1").distClass(ContDistUniform.class).distParams(15.0, 5.0));  
  
        // Data providers  
        add(new DataProvider().name("DP").providerClass(DataProviderImpl.class));  
  
        // Service processors  
        add(new ServiceProcessor().name("Auto-Process").dist("DistAuto").numExec(100).timeUnit(TimeUnit.SECONDS));  
        add(new ServiceProcessor().name("Backup").dist("DistHT").numExec(2).timeUnit(TimeUnit.MINUTES));  
    }  
}
```

10 pav. Simuliacijos konfigūracijos pavyzdys

Reikia pastebėti, kad konfigūracijos klasės yra naudojamos tik simuliacijos konfigūracijai apibrėžti JAVA programavimo kalba. Simuliacijos logika šiose klasėse nėra realizuojama.

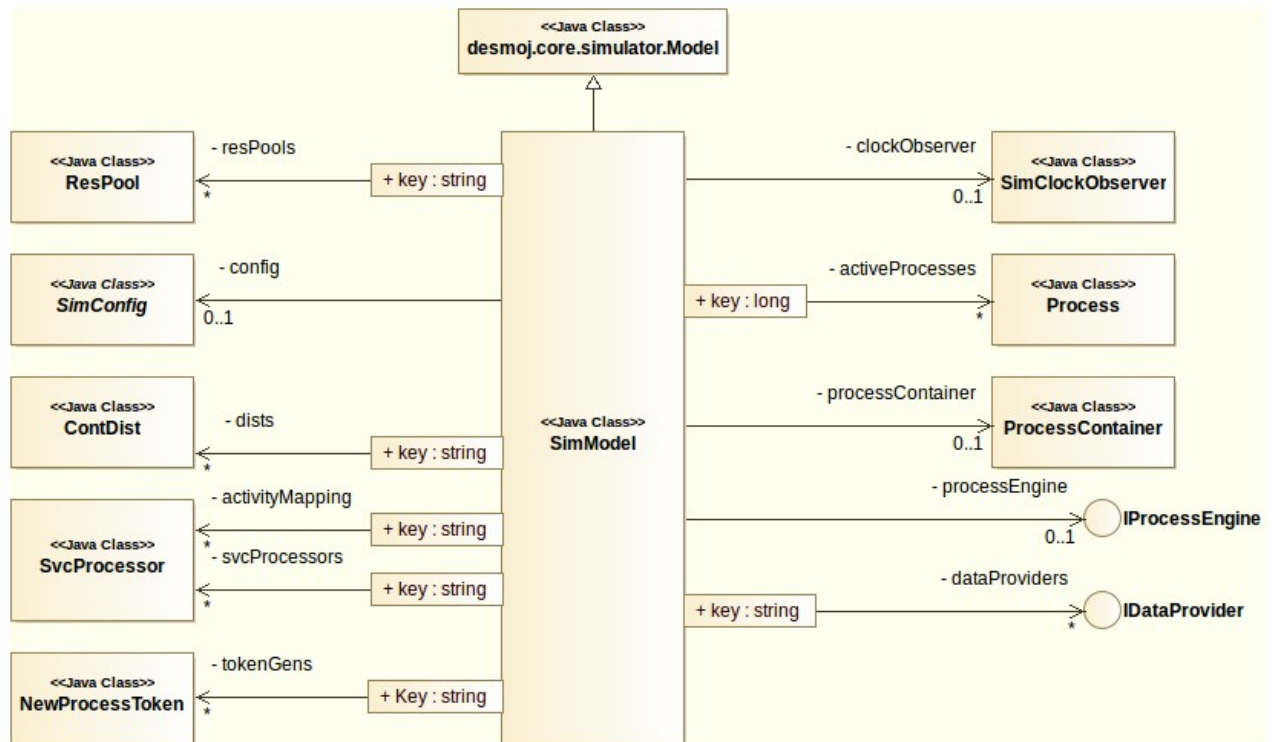
Inicializuojant simuliaciją, pagal šių klasių objektus yra dinamiškai sukuriami diskrečių įvykių simuliacijos modelio elementai ir pats simuliacijos modelis. Visas simuliacijos modelio generavimas yra atliekamas naudojant specialią klasę „ModelBuilder“, kuri naudodama konfigūracijos duomenis sukuria visus reikiamus simuliacijos modelio elementus.

Perspektyvoje, plėtojant sukurtą RPSIM modelį nesudėtingai galima sukurti specialų mechanizmą, kurio pagalba bus galima simuliacijos konfigūraciją užkrauti iš XML ar kitokio formato failo. Tai gali būti naudinga realizuojant specialias simuliacijos modelių kūrimo priemones.

2.3.3. Simuliacijos mašina

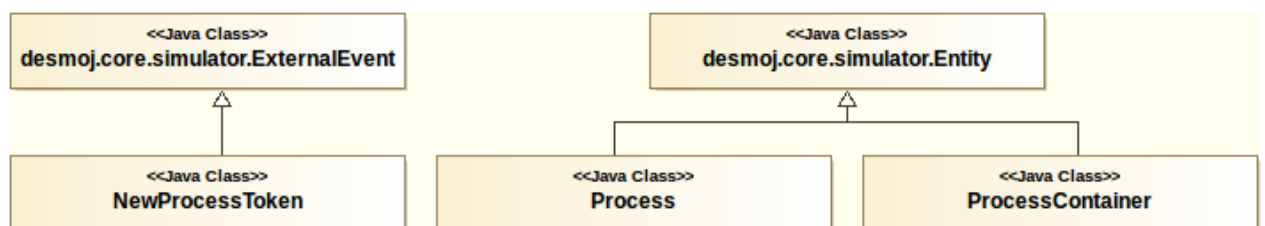
Suprojektavus simuliacijos konfigūraciją, sekančiame sudėtiname RPSIM projekte yra vykdomas simuliacijos variklio projektas. Jo metu simuliacijos mašinoje yra realizuojamas simuliacijos vykdymo bei verslo logikos mašinos valdymo algoritmas. RPSIM simuliacijos mašinos realizacijai yra naudojamas DESMO-J diskrečiųjų įvykių simuliacijos karkasas, kuriame simuliacija (simuliacijos modelis) yra aprašoma naudojant JAVA programavimo kalbą. DESMO-J simuliacijos modelis yra aprašomas specializuojant tam tikras klases, kuriose yra realizuojama simuliacijos logika (skyrius 2.2.2). RPSIM yra pritaikyta taisyklėmis grindžiamų verslo procesų simuliacijai panaudojant verslo procesų vykdymo variklius. Dėl šios priežasties, įgyvendinant RPSIM simuliacijos mašiną, buvo realizuotos verslo procesų simuliacijai reikalingos klasės, specializuojant DESMO-J klases. Aprašant RPSIM simuliacijos modelį taip pat buvo naudojama JAVA programavimo kalba, tačiau simuliacijos konfigūravimas buvo visiškai atskiras nuo paties modelio. Simuliacijos inicializavimo metu, naudojant simuliacijos konfigūracijos duomenis, dinamiškai buvo sukurti reikalingi simuliacijos modelio klasių egzemplioriai su apibrėžtais parametrais.

RPSIM simuliacijos modelio pagrindas yra „SimModel“ klasė, kuri savyje enkapsuliuoja kitas klases, atitinkančias diskrečiųjų įvykių simuliacijos esybes, bei užtikrina viso simuliacijos proceso vykdymą. Simuliacijos modelio klasių diagrama yra pateikiama 11 paveiksle. RPSIM simuliacijos modelis realizuoja simuliacijos mašinos sąsają (*ISimEngine*). Ši sąsaja, kartu su verslo logikos mašinos sąsaja (*IProcessEngien*) yra naudojama komunikacijai tarp simuliacijos ir verslo logikos mašinų, taip užtikrinant šių dviejų sudedamųjų dalių tarpusavio nepriklausomumą.



11 pav. Simuliacijos mašinos klasių diagrama

Diskrečių įvykių paradigmos pagrindą sudaro įvykiai (angl. *events*) ir esybės (angl. *entities*). Simuliacijos metu yra planuojamas įvykių aktyvavimas tam tikru laiko momentu, o įvykius aktyvavus yra keičiama su aktyvuotu įvykiu susijusių esybių būseną. Naudojant šiuos du konceptus yra aprašoma visa simuliacijos logika. RPSIM realizacijoje galima išskirti dvi įvykių ir esybių grupes. Viena grupė reikalinga proceso inicijavimui ir jo vykdymui simuliuoti, kita – veiklų vykdymui simuliuoti.

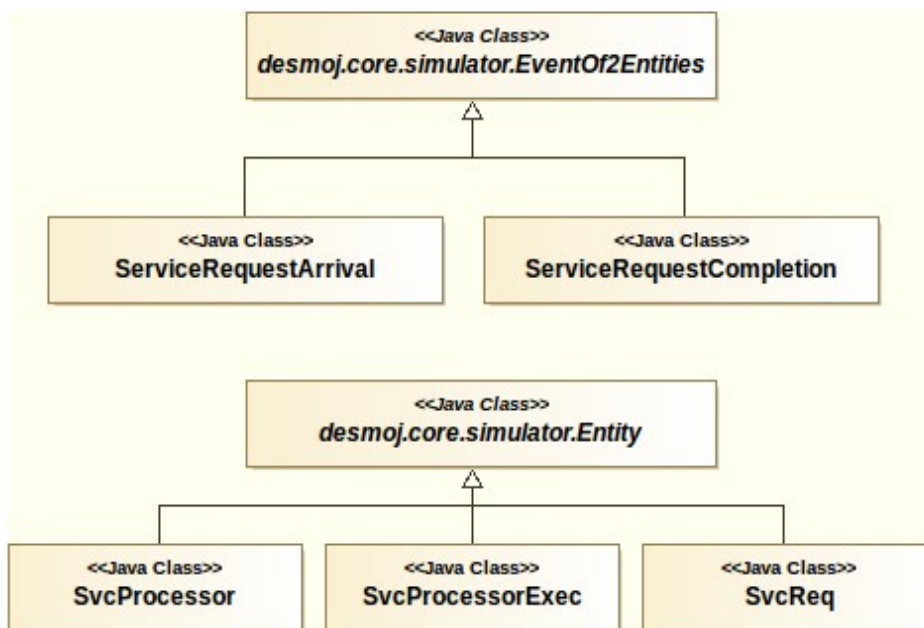


12 pav. Procesų vykdymo simuliacijos klasės

Proceso inicijavimui ir jo vykdymui simuliuoti naudojama viena įvykio ir dvi esybės klasės, realizuojančios procesų simuliacijos logiką (12 pav.). Įvykio klasė „NewProcessToken“ yra naudojama procesų paleidimui verslo logikos mašinoje simuliacijos metu. Inicializuojant simuliaciją, yra sukuriamas pirmas įvykis (klasės egzempliorius), kuris patalpinamas simuliacijos

įvykių tvarkaraštyje. Simuliacijos metu aktyvuotas įvykis paleidžia verslo procesą ir vėl perkelia save į simuliacijos įvykių tvarkaraštį. Tokiu būdu yra sukuriamas ciklinis įvykio aktyvavimas iki to momento, kai yra užbaigiamas simuliacijos procesas. „NewProcessToken“ klasė turi visus procesų paleidimui simuliuoti reikalingus atributus, kurių reikšmės yra nustatomos simuliacijos inicializavimo metu, naudojant simuliacijos konfigūraciją.

Esybių klasės „Process“ ir „ProcessContainer“ yra naudojamos proceso (jo pradžios ir pabaigos) vykdymui simuliuoti ir proceso simuliacijos statistiniams duomenims surinkti. Simuliacijos metu verslo logikos mašina informuoja simuliacijos mašiną apie paleistus ir užbaigtus procesus atitinkamai iškviesdama „newProcessArival()“ ir „newProcessCompletion()“ metodus. Pirmas metodas sukuria naują proceso esybę ir perkelia procesą į procesų konteinerį, o kitas metodas panaikina proceso esybę ir pašalina procesą iš konteinerio. Tokiu būdu simuliacijos mašina, išnaudodama DESMO-J funkcionalumą, gali surinkti statistinius duomenis apie įvykdytus procesus. Procesų vykdymo mechanizmas yra atskirtas nuo proceso paleidimo, nes verslo logikos mašinoje vykdomų procesų gali būti daugiau, nei simuliacijos mašina inicijuoja. Taip yra todėl, kad vykdant procesus ir reaguojant į tam tikrus įvykius gali būti paleidžiami kiti procesai.



13 pav. Veiklų vykdymo simuliacijos klasės

Veiklų simuliacijai realizuoti yra įvesta užduoties vykdytojo sąvoka (anlg. *Service Processor*). Kiekvienai proceso veiklai, kurios vykdymas turi būti simuliuojamas yra priskiriamas užduočių vykdytojas (klasių diagrama pateikiama 13 pav.). Simuliacijos metu verslo logikos mašina kiekvienai vykdomai veiklai siunčia užduoties apdorojimo užklausą, simuliacijos mašinai

iškviesdama „*newServiceRequest()*“ metodą. Kiekvienai užklausiai yra sukuriamas „*SvcReq*“ esybė, priklausomai nuo veiklos parenkamas užduoties vykdytojas - „*ServiceProcessor*“ esybė bei sukuriamas ir aktyvuojamas „*ServiceRequestArrival*“ įvykis. Įvykis yra susiejamas su užklausa bei jos vykdytoju ir įvykio aktyvavimo metu yra atliekamas veiklos vykdymo pradžios algoritmas (priedas Nr. 3). Algoritmo atlikimo metu yra sukuriamas užduoties atlikimo pabaigos įvykis (įvykis „*ServiceRequestCompletion*“), kurio aktyvavimas konkrečiu laiko momentu yra apibrėžiamas simuliacijos įvykių tvarkaraštyje. „*ServiceRequestCompletion*“ aktyvavimo metu be simuliacijos logikos, verslo logikos mašinai yra pranešama apie įvykdytą užklausą. Gavus tokį pranešimą, verslo logikos mašina gali toliau vykdyti verslo procesus ir taisykles.

Dar vienas labai svarbus simuliacijos aspektas yra laiko sinchronizavimas tarp simuliacijos ir verslo logikos mašinų. Simuliacijos eigą valdo simuliacijos mašina, todėl simuliacijos mašina, kai to reikia, turi atnaujinti verslo logikos mašinos laiką. DESMO-J turi priemones leidžiančias įvykdyti tam tikrą algoritmą pasikeitus simuliacijos laikui (naudojamas „Observer“ šablonas). Laiko sinchronizavimui realizuoti buvo sukurta „*SimClockObserver*“ klasė. Ši klasė simuliacijos inicializavimo metu yra registruojama simuliacijos laikrodyje, kuris kiekvieną kartą pasikeitus simuliacijos laikui iškviečia „*SimClockObserver*“ metodą „*update()*“. Šis metodas realizuoja verslo logikos mašinos laiko atnaujinimą. Taigi, simuliacijos metu kiekvieną kartą keičiantis simuliacijos laikui verslo logikos mašinoje yra nustatomas naujas laikas.

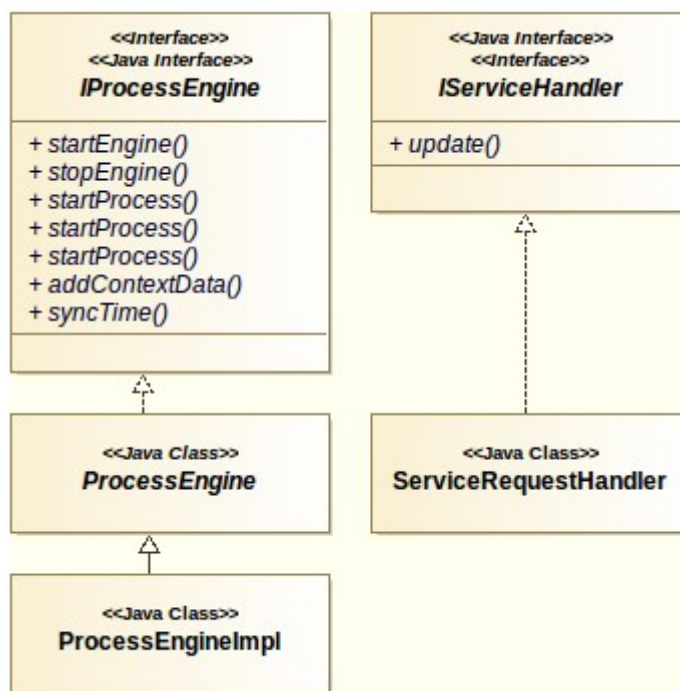
2.3.4. Verslo logikos mašina

Projektuojant RPSIM buvo suformuluotas uždavinys, atskirti verslo logiką realizuojančias komponentes nuo simuliaciją kontroliuojančių komponentių bei pakartotinai panaudoti egzistuojančias verslo procesų ir verslo taisyklių vykdymo mašinas. RPSIM realizacijai verslo logikos mašina buvo pasirinktos DROOLS ir jBPM verslo taisyklių ir verslo procesų mašinos (2.2.1 skyrius). Vykdymui pritaikytų procesų ir taisyklių mašinos panaudojimas leido įgyvendinti kitą architektūros principą – simuliacijai naudoti vykdymui paruoštus verslo procesų modelius ir verslo taisykles.

Verslo procesų ir verslo taisyklių varikliai be pakeitimų negali būti panaudoti simuliacijai, nes simuliacijos metu vykdomų procesų komunikacija su išorinėmis sistemomis ir vartotojais turėtų būti apribota. Vietoje to simuliacijos metu turi būti realizuota komunikacija su simuliacijos mašina. Komunikacija su simuliacijos mašina reikalinga tam, kad būtų galima simuliuoti procesų veiklas.

Siekiant RPSIM sudedamąsias dalis padaryti kiek įmanoma mažiau priklausomas viena nuo kitos, buvo apibrėžtos 2 sąsajos, kurias verslo logikos mašina turi realizuoti (14 pav.). Viena sąsaja

(„*IProcessEngine*“) skirta verslo logikos variklio valdymui bei verslo procesų paleidimui. Kita sąsaja („*IServiceHandler*“) naudojama veiklos vykdymo būsenai keisti iš simuliacijos mašinos. O tai reiškia, kad simuliacijos mašina simuliuoja veiklos vykdymą, o kai veikla yra įvykdyta siunčia pranešimą atgal į procesų mašiną ir pranešimui persiuntimui apie susimuliuotą veiklos vykdymą naudoja „*IServiceHandler*“ sąsają. Abi šios sąsajos yra realizuotos DROOLS pagrindu veikiančioje verslo logikos mašinoje.

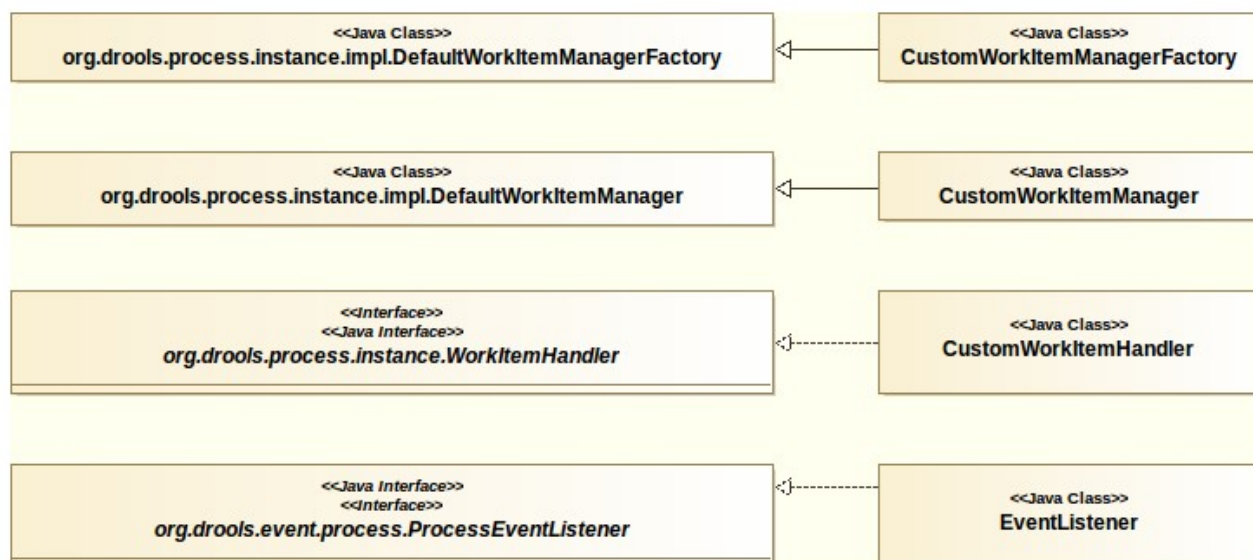


14 pav. Verslo logikos mašinos sąsajos

Kaip jau buvo minėta, norint DROOLS ir jBPM panaudoti simuliacijai yra reikalingi tam tikri pakeitimai. Realizuojant RPSIM, pats DROOLS ir jBPM programinis kodas nebuvo keičiamas, t. y. RPSIM realizacijoje yra naudojami standartiniai DROOLS ir jBPM paketai. Reikalingi pakeitimai buvo atlikti specializuojant DROOLS ir jBPM klases arba realizuojant reikalingas konkrečias sąsajas (15 pav). Visus pakeitimus pagal jų paskirtį galima skirstyti į dvi grupes:

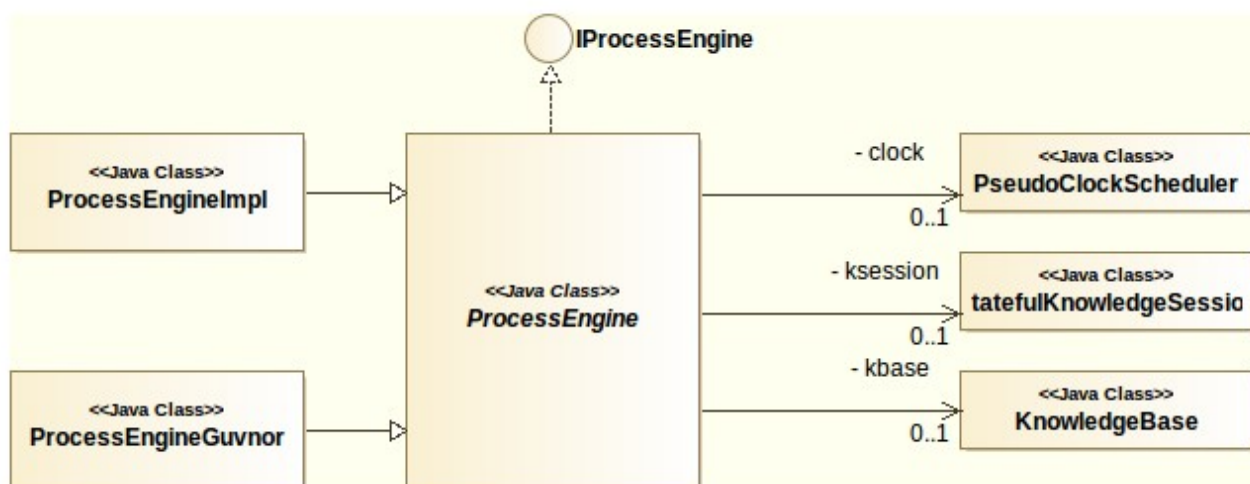
- su įvykių valdymu susiję pakeitimai;
- su veiklų valdymu susiję pakeitimai.

Pakeistas (arba praplėstas) įvykių valdymas yra reikalingas tam, kad galima būtų užfiksuoti verslo proceso pradžios ir pabaigos įvykius. Įvykių valdymo tikslais jBPM verslo procesų mašina leidžia registruoti įvykių apdorojimo klases, kurios turi realizuoti specialią sąsają „*EventHandler*“.



15 pav. Įvykių ir veiklų apdorojimo klasės

Proceso simuliacijos eigoje veiksmai, atliekami proceso veiklų metu, turi būti perimami ir nukreipiami į simuliacijos mašiną. Tam tikslui RPSIM priemonėje buvo realizuotos veiklų valdymo klasės (15 pav.), kurios pakeitė standartines DROOLS ir jBPM klases.



16 pav. Verslo logikos variklio klasių diagrama

Didžioji dalis verslo logikos variklio logikos yra realizuota „ProcessEngine“ klasėje (16 pav.), kuri realizuoja verslo logikos variklio sąsają „IPProcessEngine“, inicializuoja DROOLS ir jBPM žinias, sukuria taisyklų ir procesų vykdymo sesijas, užregistruoja įvykių apdorojimo bei veiklų vykdymo perėmimo mechanizmus.

3. SIMULIAVIMO PRIEMONĖS DEMONSTRACIJA

3.1. Modelis

3.1.1. Taisyklėmis grindžiamo verslo proceso modelis

Darbo metu buvo realizuota taisyklėmis grindžiamų verslo procesų simuliacijos priemonė, kurios veikimo demonstracija yra aprašoma šiame skyriuje. Šiame darbe buvo įvardinti keturi bendro verslo procesų ir verslo taisyklių panaudojimo būdai (1.1.8 skyrius). Realizuotos simuliacijos priemonės demonstracijai buvo pasirinkti du iš jų, t. y. taisyklės išskirtos į atskirą veiklą bei taisyklėmis koordinuojami verslo procesai. Abu paminėti taisyklių ir procesų integravimo būdai buvo panaudoti viename proceso modelyje. Demonstracijos proceso modelis yra pateikiamas priede Nr. 4, o taisyklės priede Nr. 5. Bendrai proceso modelis ir taisyklės gali būti laikomos taisyklėmis grindžiamo verslo proceso modeliu.

Demonstracijai buvo pasirinktas supaprastintas užklauskos apdorojimo procesas. Pirmu žingsniu (veikla „Validate“) yra tikrinami užklauskos duomenys. Tikrinimas yra realizuotas taisyklėmis, t. y. vykdomos taisyklės, kurios pagal pradinius duomenis visas užklauskas sukatégorizuoja į geras ir blogas. Šioje vietoje yra panaudotas pirmas taisyklių ir procesų integravimo būdas, kai taisyklės yra išskirtos į atskirą veiklą.

Toliau procesas išsišakoja. Jei užklausa yra bloga, tada yra išsiunčiamas pranešimas ir procesas užbaigiamas. Jei užklausa yra gera, tada pereiname prie dinaminio proceso elemento „Ad-Hoc Sub-Process“. Šioje proceso dalyje yra apibrėžtos veiklos, bet nėra nurodyta kurios iš šių veiklų turi būti atliktos, bei nėra apibrėžtas jų nuoseklumas. Kokios veiklos turi būti atliekamos yra apibrėžiama taisyklėmis. Šioje vietoje yra panaudotas antras procesų ir taisyklių integravimo būdas – taisyklėmis koordinuojami verslo procesai. Simuliacijos priemonės demonstracijai buvo apibrėžtos trys taisyklės (priedas Nr. 5), kurios priklausomai nuo užklauskos sumos aktyvuoja arba „Review“, arba „Auto-Process“, arba „Investigate“ veiklas. Reikia pastebėti, kad tuo atveju, kai yra aktyvuojama veikla „Investigate“, kartu su ja yra atliekama ir kita veikla „Approve“.

Verslo proceso modelis ir verslo taisyklės yra papildomos duomenų modeliu. Duomenų modelis yra apibrėžtas dviem JAVA klasėmis „Person“ ir „Request“, kurios reprezentuoja atitinkamu verslo objektus.

Taigi, taisyklės, proceso ir duomenų modeliai yra demonstracijoje naudojamo taisyklėmis grindžiamo verslo proceso modelio pagrindas. Tai tik viena simuliacijai reikalinga informacijos dalis. Kita dalis yra simuliacijos modelis.

3.1.2. Simuliacijos modelis

Norint atlikti taisyklėmis grindžiamą verslo procesų simuliaciją, reikia procesų modelį papildyti simuliacijos modeliu. Simuliacijos modelį sudaro simuliacijos konfigūracija ir duomenų generatorius.

RPSIM simuliacijos priemonėje, konfigūracija yra apibrėžiama JAVA kalba, naudojant simuliacijos modelio konfigūracijos klases (skyrius 2.3.2). Demonstracijos simuliacijos konfigūracija yra pateikiama priede Nr. 5. Simuliacijos konfigūracijoje yra apibrėžti:

- Vienas proceso paleidimo elementas („*TokenGenerator*“), kuris periodiškai paleidžia demonstracijoje simuliuojamą procesą.
- Visos (5) simuliuojamo proceso veiklos („*Activity*“), kurios yra susietos su atitinkamu užduoties atlikėju.
- 4 užduočių atlikėjai („*ServiceProcessor*“), kurių pagalba yra aprašoma veiklų atlikimo dinamika simuliacijos metu.
- 4 tikimybiniai skirstiniai arba atsitiktinių skaičių generatoriai („*Distribution*“), kurie naudojami veiklos atlikimo trukmei arba laiko tarpui tarp proceso paleidimų nustatyti.

Demonstracijos metu, simuliuojant verslo procesą, yra vykdomos taisyklės, todėl į simuliacijos modelį yra įtrauktas duomenų generatorius. Duomenų generatorius tai „*IdataProvider*“ sąsają realizuojanti klasė „*DataProviderImpl*“. Duomenų generatorius simuliacijos konfigūracijoje yra priskiriamas proceso paleidimo elementui.

3.2. Simuliacijos rezultatai

Kiekvienos simuliacijos metu RPSIM sugeneruoja ataskaitą su simuliacijos rezultatais. Aukščiau aprašyto taisyklėmis grindžiamo verslo proceso modelio simuliacijos rezultatas yra pateikiamas priede Nr. 7. Remiantis ataskaitos duomenimis, pavyzdžiui, galima nustatyti, kad simuliacijos metu:

- buvo įvykdyti 26178 procesai, vidutiniškai vienu metu buvo 7,8 aktyvių procesų;
- vidutinė proceso atlikimo trukmė buvo 45 min, maksimali – 4,3 h;
- veikla „*Finalize*“ buvo ilgiausiai atliekama (remiantis vidutiniu atlikimo laiku) veikla, o antroje vietoje pagal trukmę buvo „*Investigate&Aprove*“.

Siekiant pademonstruoti RPSIM galimybes buvo atliktas eksperimentas, kurio metu buvo vykdomos dvi aukščiau aprašyto proceso modelio simuliacijos su skirtingomis taisyklėmis. Eksperimento metu buvo keičiamos tik taisyklės. Visi kiti parametrai simuliacijos parametrai išliko pastovūs. 1 lentelėje yra pateikiamos apibendrintos taisyklės, kurios buvo naudotos abiejų

simuliacijų metu. Kaip matyti iš lentelių, buvo keičiami sumos režiai, nuo kurių priklauso, kokia veikla turi būti toliau vykdoma.

1 lentelė. Taisyklės naudotos simuliacijos metu

	Review	Auto-Process	Investigate&Approve
Simuliacija 1	500<= Amount < 1500	Amount < 500	Amount >= 1500
Simuliacija 2	500<= Amount < 1000	Amount < 500	Amount >= 1000

Eksperimento metu buvo siekiama nustatyti taisyklių pokyčio įtaka „Ad-Hoc Sub-process“ veikloms. Keletas abiejų simuliacijos apibendrintų rezultatai pateikiami 2 lentelėje.

2 lentelė. Eksperimento simuliacijų rezultatai

Parametras	Simuliacija 1	Simuliacija 2
Vidutinė proceso vykdymo trukmė, min.	41,3	50,8
„Investigate“ veiklos vykdymų skaičius, vnt.	6 995	13 707
„Auto-Process“ veiklos vykdymų skaičius, vnt.	5496	5565
„Review“ veiklos vykdymų skaičius, vnt.	13 685	6 903

Iš rezultatų lentelės matome, kad pakeitus taisykles išaugo vidutinis proceso atlikimo laikas, pasikeitė atliekamų užduočių kiekiai. Gautus rezultatus buvo galima nuspėti ir neatlikus simuliacijos, nes eksperimente simuliuotas taisyklėmis grindžiamas verslo procesas buvo pakankamai paprastas. Tačiau atliktas eksperimentas parodo, kad darbo metu sukurta RPSIM priemonės tinkamumą taisyklėmis grindžiamų verslo procesų simuliacijai.

IŠVADOS

1. Analizuojant verslo procesų valdymo probleminę sritį buvo nustatyta, kad verslo procesų modeliavimas ir simuliacijos yra dvi veiklos, priklausančios verslo procesų valdymo gyvavimo ciklo projektavimo fazei. Dėl stochastinės verslo proceso prigimties, intuityvus verslo procesų modelio vertinimas dažnai yra klaidingas ir turi būti taikomas verslo procesų simuliacijos, todėl simuliacija yra vienas iš efektyviausių būdų įvertinti projektuojamo verslo proceso našumo rodiklius. Simuliacija, kaip verslo proceso modelio vertinimo technika, užtikrina aukštesnę verslo procesų modeliavimo kokybę bei apskritai didina visos verslo procesų valdymo disciplinos efektyvumą.

2. Nagrinėjant verslo modeliavimo technikas buvo nustatyta, kad šiuo metu egzistuoja dvi mažai integruotos verslo procesų ir verslo taisyklių modeliavimo technikos, kurios istoriškai buvo plėtojamos nepriklausomai viena nuo kitos. Verslo taisyklių ir procesų kontinuumo problemos nagrinėjimas parodo, kad šiuo metu populiariausias šių dviejų technikų panaudojimo būdas neišnaudoja viso šių technikų potencialo, nes verslo procesai suaktyvina verslo taisykles kaip veiklą. Efektyvesnis veiklos modeliavimo būdas yra naudoti nedidelius verslo proceso ar jo fragmentų modelius, sujungiant juos verslo taisyklėmis, kaip kontrolės elementais, tačiau šis būdas yra nepakankamai palaikomas tiek modeliavimo standartuose, tiek ir modeliavimo bei simuliacijos priemonėse ir tai sąlygoja mažą tokio metodo populiarumą. Darbe pasiūlytas taisyklėmis grindžiamų verslo procesų modeliavimo būdas leidžia kurti paprastus, lengviau valdomus ir suprantamus verslo procesų modelius. Toks metodas verslo procesų modeliavimui siūlo naudoti plačiai paplitusią ir standartizuotą BPMN procesų modeliavimo kalbą bei ECA tipo taisykles, kas užtikrina lengvą metodo adaptavimą, nes nereikalauja specifinių žinių.

3. Taisyklėmis grindžiamų verslo procesų simuliacijos priemonių realizavimas parodė, kad panaudojant egzistuojančius verslo taisyklių ir verslo procesų vykdymo variklius bei diskrečiųjų įvykių simuliacijos karkasus ir juos sujungiant galima sukurti taisyklėmis grindžiamų verslo procesų simuliacijos priemones. Realizacija nepareikalavo didesnių verslo procesų ir taisyklių variklių pokyčių ir pakako virtualaus laikrodžio bei įvykių aptikimo funkcionalumo. Tai sudarė prielaidas kitų, pavyzdžiui, įmonėse jau naudojamų, verslo taisyklių ir verslo procesų vykdymo priemonių panaudojimo galimybes. Kadangi simuliacijos priemonės įgyvendinimui buvo naudojami verslo procesų ir verslo taisyklių vykdymo varikliai, tai galima teigti, kad taisyklėmis grindžiamų verslo procesų įgyvendinimas taip pat neturėtų sukelti didesnių problemų. Visai tai parodo metodo aktualumą ir naudingumą.

4. Taisyklėmis grindžiamų verslo procesų simuliacijos priemonių realizavimas parodė įgyvendintos simuliacijos

priemonės tinkamumą ir panaudojamumą kasdieniams procesų optimizavimo klausimams spręsti. Taisyklėmis grindžiamų verslo procesų simuliacijos yra efektyvi priemonė “kas jeigu” verslo procesų analizei atlikti. Simuliacijos metu yra naudojamas integruotas verslo procesų ir taisyklių modelis, kuris leidžia lengvai įvertinti taisyklių pokyčio įtaką proceso rezultatams.

LITERATŪRA

- 1 Gartner EXP Worldwide Survey of Nearly 1,600 CIOs Shows IT Budgets in 2010 to be at 2005 Levels [interaktyvus] 2010. [žiūrėta 2011-09-10]. Prieiga per internetą: <<http://www.gartner.com/it/page.jsp?id=1283413>>.
- 2 Gartner EXP Worldwide Survey of More than 1,500 CIOs Shows IT Spending to Be Flat in 2009 [interaktyvus] 2009. [žiūrėta 2011-03-01]. Prieiga per internetą: <<http://www.gartner.com/it/page.jsp?id=855612>>.
- 3 Debevoise, T. 2007. Business Process Management with a Business Rules Approach: Implementing the Service Oriented Architecture. Booksurge Llc.
- 4 Dumas, M.; Aalst, W. M. P. van der; Hofstede, A. H. M. ter 2005. Process-aware information systems: bridging people and software through process technology. John Wiley & Sons.
- 5 Hammer, M. 2009. What is Business Process Management?. Handbook on Business Process Management, volume 1, Springer, p. 3-16.
- 6 Shewhart, W. 1986. Statistical method from the viewpoint of quality control. Dover Publications, NY.
- 7 Deming, W.E. 1953. Statistical techniques in industry, Adv Manage, 18(11):8–12. .
- 8 Hammer, M. 1990. Reengineering work: Don't automate, obliterate. Harv Bus Rev, 68:104–112.
- 9 Hammer, M.; Champy, J. 1993. Reengineering the corporation: A manifesto for business revolution. Harper Collins Publishers.
- 10 Weske, M. 2007. Business Process Management: Concepts, Languages, Architectures. Springer.
- 11 Muehlen, M. zur 2002. Workflow-based Process Controlling: Foundation, Design, and Application of Workflow-driven Process Information Systems. Logos Verlag Berlin.
- 12 Deming, W.E. 1986. Out of the crisis. MIT Center for Advanced Engineering.
- 13 Bridgeland, D. M.; Zahavi, R. 2009. Business modeling: a practical guide to realizing business value. Morgan Kaufmann.
- 14 Vasilecas, O.; Saulis, A. 2008. Informacinių sistemų kūrimo metodai ir technikos. Klaipėda.
- 15 The Open Group Architecture Framework 9.1 [interaktyvus] 2011. [žiūrėta 2012-03-1]. Prieiga per internetą: <<http://pubs.opengroup.org/architecture/togaf9-doc/arch/>>.
- 16 Ross, R.G. 2008. Rules vs. Processes (Again) - Part 1,2: There's Simply No Need for Confusion. Business Rules Journal, Vol. 9, No. 6.
- 17 The Business Rules Manifesto [interaktyvus] 2003. [žiūrėta 2012-03-15]. Prieiga per internetą: <<http://www.businessrulesgroup.org/brmanifesto.htm>>.

- 18 Ross, R. G. 2005. How Rules and Processes Relate ~ Part 2. Business Process. Business Rules Journal, Vol. 6, No. 11.
- 19 Business Process Model and Notation (BPMN), version 2.0 [interaktyvus] 2011. [žiūrėta 2012-03-15]. Prieiga per internetą: <<http://www.omg.org/spec/BPMN/2.0/>>.
- 20 Mulyar, N.A.; Schonenberg, M.H.; Mans, R.S.; Russell, N.C.; Aalst, W.M.P. van der 2007. Towards a Taxonomy of Process Flexibility (Extended Version). BPM center Report BPM-07-11, BPMcenter.org.
- 21 Mendling, J.; Reijers, H.; Aalst, W.M.P. van der 2010. Seven Process Modeling Guidelines (7PMG). Information and Software Technology, 52(2), p. 127-136 .
- 22 Aalst, W.M.P. van der; Mans, R.S.; Russell, N.C. 2009. Workflow Support Using Proclets: Divide, Interact, and Conquer. IEEE Bulletin of the Technical Committee on Data Engineering, 32(3), p. 16-22.
- 23 Koehler, J. 2010. The Role of BPMN in a Modeling Methodology for Dynamic Process Solutions. 2nd Int. Workshop on BPMN, Springer LNBIP 67, p. 46-62.
- 24 Business Motivation Model (BMM), version 1.1 [interaktyvus] 2010. [žiūrėta 2012-03-15]. Prieiga per internetą: <<http://www.omg.org/spec/BMM/1.1/>>.
- 25 Sinur, J. 2009. The art and science of rules vs. process flows. Gartner Research Report G00166408.
- 26 Koehler, J. 2011. The Process-Rule Continuum — Can BPMN & SBVR Cope with the Challenge?..
- 27 Shannon, R.E. 1975. System Simulation: The Art and Science. Prentice Hall.
- 28 Simulation Article [interaktyvus] 1998. [žiūrėta 2012-03-15]. Prieiga per internetą: <<http://www.modelbenders.com/encyclopedia/encyclopedia.html>>.
- 29 Weber, B.; Mutschler, B.; Reichert, M. 2010. Investigating the Effort of Using Business Process Management Technology: Results from a Controlled Experiment. Science of Computer Programming.
- 30 Aalst, M.P. van der; Reijers, H. A. 2005. The effectiveness of workflow management systems: Predictions and lessons learned. International Journal of Information Management.
- 31 Havey, M. 2005. Essential Business Process Modeling. O'Reilly.
- 32 Workflow reference model [interaktyvus] 2005. [žiūrėta 2012-03-21]. Prieiga per internetą: <<http://www.wfmc.org/reference-model.html>>.
- 33 Drools - The Business Logic integration Platform [interaktyvus] . [žiūrėta 2012-03-02]. Prieiga per internetą: <<http://www.jboss.org/drools/>>.
- 34 DESMO-J Quick Overview [interaktyvus] . [žiūrėta 2012-03-22]. Prieiga per internetą:

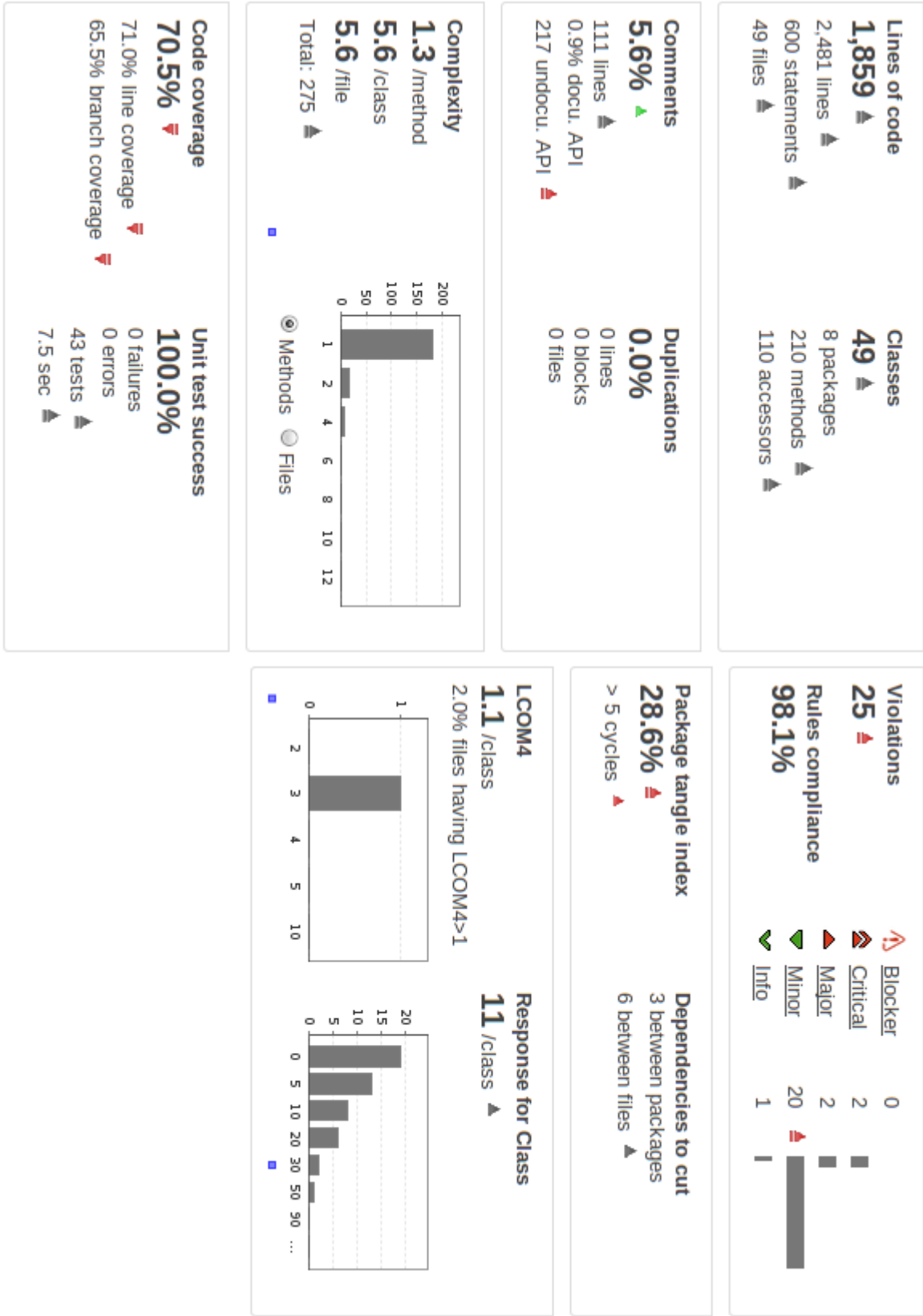
<<http://desmoj.sourceforge.net/overview.html>>.

35 DESMO-J Basic Features [interaktyvus] . [žiūrėta 2012-03-22]. Prieiga per internetą:

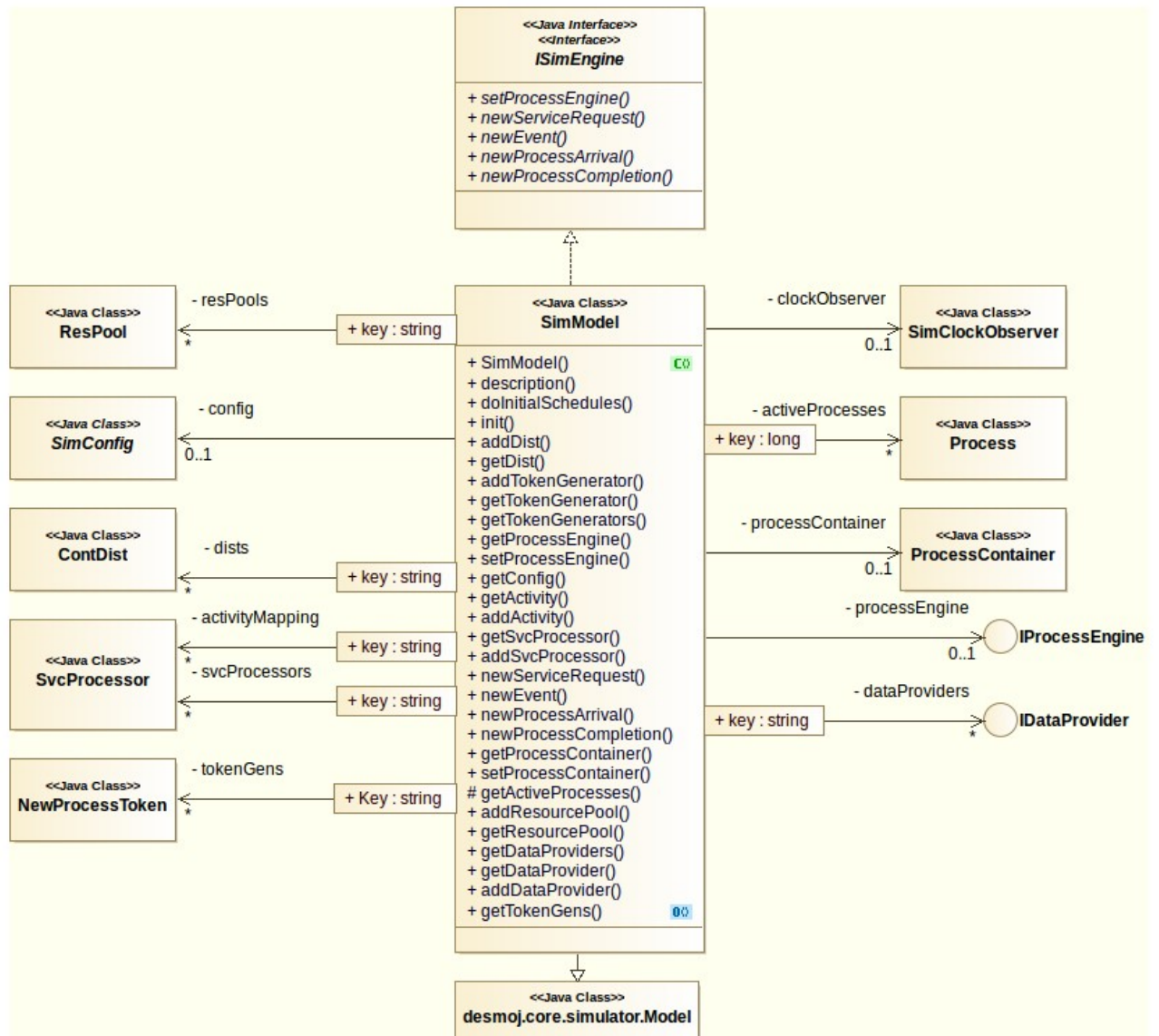
<http://desmoj.sourceforge.net/basic_features.html>.

PRIEDAI

Priedas Nr. 1 - Informacija apie RPSIM projektą



Priedas Nr. 2 - *SimModel* klasių diagrama



Priedas Nr. 3 - Veiklų simuliacijos algoritmas (JAVA kodo ištraukos)

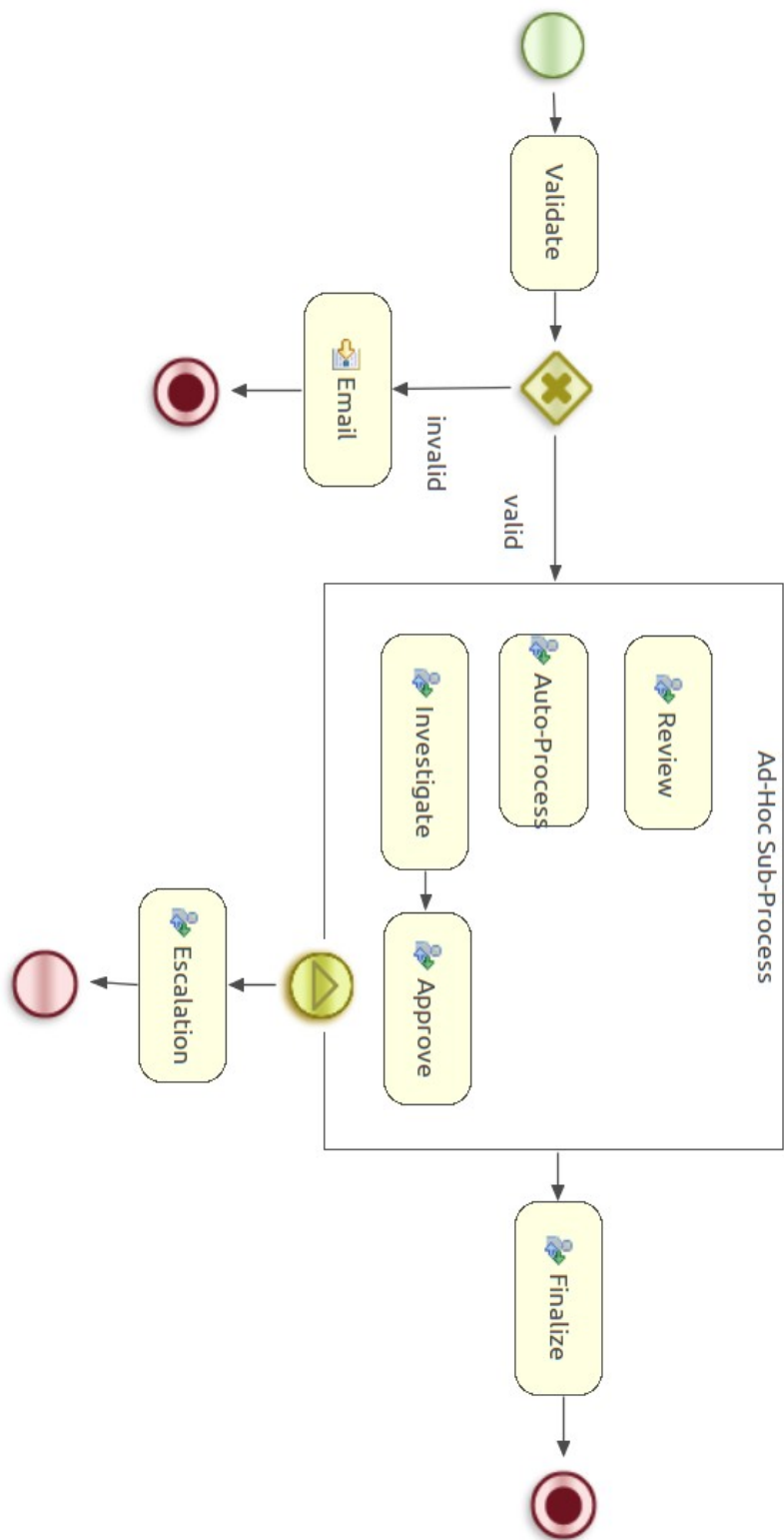
```
public class ServiceRequestArrival extends EventOf2Entities<SvcReq, SvcProcessor> {
...
    @Override
    public void eventRoutine(SvcReq request, SvcProcessor processor) {
        processor.add(request);
        if ( processor.isAvailable() ) {
            TimeSpan timeSpan = processor.start(request);
            ServiceRequestCompletion event = new ServiceRequestCompletion(getModel(),
"SR_Completion", false);
            event.schedule(request, processor, timeSpan);
        }
    }
}

public class ServiceRequestCompletion extends EventOf2Entities<SvcReq, SvcProcessor> {
...
    public void eventRoutine(SvcReq request, SvcProcessor processor) {
        processor.complete(request);
        request.getHandler().update();
        SvcReq nextRequest = processor.getNextRequest();
        if ( nextRequest != null ) {
            TimeSpan timeSpan = processor.start(nextRequest);
            ServiceRequestCompletion event = new ServiceRequestCompletion(getModel(),
"SR_Completion", false);
            event.schedule(nextRequest, processor, timeSpan);
        }
    }
}

public class SvcProcessor extends Entity {
...
    public TimeSpan start(SvcReq request) {
        request.setStartTime(presentTime().getTimeAsDouble());
        waitTimeStats.update(request.getStartTime() - request.getArrivalTime());
        SvcProcessorExec reqExec = idleQueue.first();
        idleQueue.remove(reqExec);
        waitQueue.remove(request);
        map.put(request, reqExec);
        return getServiceTime();
    }

...
    public void complete(SvcReq request) {
        idleQueue.insert(map.get(request));
        map.remove(request);
        request.setCompleteTime(presentTime().getTimeAsDouble());
        procTimeStats.update(request.getCompleteTime() - request.getStartTime());
        totalTimeStats.update(request.getCompleteTime() - request.getArrivalTime());
    }
...
}
```

Priedas Nr. 4 - Demonstracijos proceso modelis



Priedas Nr. 5 - Demonstracijos proceso modelio taisyklės

Veiklos „Validate“ taisyklės

```
rule "Unknown Customer" ruleflow-group "Validation"
when
    r: Request()
    not Person(id == r.personId)
then
    r.setInvalid("Unknown customer");
end

rule "Underage Customer" ruleflow-group "Validation"
when
    r: Request()
    Person(id == r.personId, age < 100)
then
    r.setInvalid("Underage customer");
end
```

„Ad-hoc“ proceso taisyklės

```
rule "Review average requests"
when
    r: Request( amount >= 500, amount < 1500 )
    p: WorkflowProcessInstance()
    n: DynamicNodeInstance( processInstance == p)
    String( this == r.id ) from p.getVariable("request")
then
    kcontext.getKnowledgeRuntime().signalEvent("Review", null, p.getId());
end

rule "Investigate large requests"
when
    r: Request( amount >= 1500 )
    p: WorkflowProcessInstance()
    n: DynamicNodeInstance( processInstance == p)
    String( this == r.id ) from p.getVariable("request")
then
    kcontext.getKnowledgeRuntime().signalEvent("Investigate", null,
p.getId());
end

rule "Automatically process small requests"
when
    r: Request( amount < 500 )
    p: WorkflowProcessInstance()
    n: DynamicNodeInstance( processInstance == p)
    String( this == r.id ) from p.getVariable("request")
then
    kcontext.getKnowledgeRuntime().signalEvent("Auto-Process", null,
p.getId());
end
```

Priedas Nr. 6 – Demonstrācijas simulācijas konfigurācija

```
@Override
public void configure() {
    name("Demo simulation");
    add(new TokenGenerator().name("TG")
        .process("lt.bumbis.rpsim.demo.requestHandling")
        .dist("DistArrival").timeUnit(TimeUnit.MINUTES)
        .dataProvider("DP").useContextData(true));

    // Distributions
    add(new Distribution().name("DistArrival")
        .distClass(ContDistExponential.class).distParams(20.0));
    add(new Distribution().name("DistAuto")
        .distClass(ContDistUniform.class).distParams(5.0, 2.0));
    add(new Distribution().name("DistHT")
        .distClass(ContDistUniform.class)
        .distParams(10.0, 7.0));
    add(new Distribution().name("DistHT1")
        .distClass(ContDistUniform.class)
        .distParams(15.0, 5.0));

    // Data providers
    add(new DataProvider().name("DP").providerClass(DataProviderImpl.class));

    // Service processors
    add(new ServiceProcessor().name("Auto-Process").dist("DistAuto")
        .numExec(100).timeUnit(TimeUnit.SECONDS));
    add(new ServiceProcessor().name("Review").dist("DistHT")
        .numExec(2).timeUnit(TimeUnit.MINUTES));
    add(new ServiceProcessor().name("Investigate&Approve").dist("DistHT1")
        .numExec(1).timeUnit(TimeUnit.MINUTES));
    add(new ServiceProcessor().name("Finalize").dist("DistHT1")
        .numExec(5).timeUnit(TimeUnit.MINUTES));

    // Activity mapping
    add(new Activity().name("Review").svcProcessor("Review"));
    add(new Activity().name("Auto-Process").svcProcessor("Auto-Process"));
    add(new Activity().name("Investigate").svcProcessor(
        "Investigate&Approve"));
    add(new Activity().name("Approve").svcProcessor("Investigate&Approve"));
    add(new Activity().name("Finalize").svcProcessor("Finalize"));
}
```

Priedas Nr. 7 - RPSIM simuliacijos rezultatų failas

Sim1 Report Model Demo simulation

Description

Description... Report drawn at 31536000.0000. Last reset at 0.0000.

Tallies

Title	(Re)set	Obs	Mean	Std.Dv	Min	Max
WaitTimeStats_Finalize	0.0000	26175	0.0	0.0	0.0	0.0
ProcTimeStats_Finalize	0.0000	26174	600.32132	173.56514	300.0002	899.96995
TotalTimeStats_Finalize	0.0000	26174	600.32132	173.56514	300.0002	899.96995
WaitTimeStats_Investigate&Approve	0.0000	13991	158.40033	334.56771	0.0	3054.73139
ProcTimeStats_Investigate&Approve	0.0000	13990	596.95192	174.66873	300.00483	899.91551
TotalTimeStats_Investigate&Approve	0.0000	13990	755.32488	377.94387	300.06304	3499.92893
WaitTimeStats_Auto-Process	0.0000	5496	0.0	0.0	0.0	0.0
ProcTimeStats_Auto-Process	0.0000	5496	3.4841	0.86341	2.00072	4.99882
TotalTimeStats_Auto-Process	0.0000	5496	3.4841	0.86341	2.00072	4.99882
WaitTimeStats_Review	0.0000	13685	3.64601	32.35923	0.0	521.58753
ProcTimeStats_Review	0.0000	13685	509.39738	52.22084	420.00453	599.98719
TotalTimeStats_Review	0.0000	13685	513.04338	61.27703	420.00453	1045.82983
ProcExecTime_ProcContainer	0.0000	26174	2477.67601	1332.54544	320.39395	14649.46952

Counts and Aggregates

Title	Type	(Re)set	Obs	Current Value	Min	Max
TokenCount_TG	Count	0.0000	26178	26178	0	26178

Queues

Title	Qorder	(Re)set	Obs	QLimit	Qmax	Qnow	Qavg.	Zeros	max.Wait	avg.Wait
Finalize_WQ	FIFO	0.0000	26175	unlimit.	1	0	0.0	26175	0.0000	0.0000
Finalize_IQ	FIFO	0.0000	26175	unlimit.	5	4	4.50174	0	20869.4482	5423.1851
Investigate&Approve_WQ	FIFO	0.0000	13991	unlimit.	5	1	0.07028	10349	3054.7313	158.4003
Investigate&Approve_IQ	FIFO	0.0000	13991	unlimit.	1	0	0.73518	3642	36118.7247	1657.1031
Auto-Process_WQ	FIFO	0.0000	5496	unlimit.	1	0	0.0	5496	0.0000	0.0000
Auto-Process_IQ	FIFO	0.0000	5496	unlimit.	100	100	99.99939	0	792860.8357	569020.3591
Review_WQ	FIFO	0.0000	13685	unlimit.	3	0	0.00158	13416	521.5875	3.6460
Review_IQ	FIFO	0.0000	13685	unlimit.	2	2	1.77895	269	29300.9413	4099.3123
ActiveProcessQueue_ProcContainer	FIFO	0.0000	26174	unlimit.	9	4	2.05655	0	14649.4695	2477.6760

Distributions

Title	(Re)set	Obs	Type	Parameter 1	Parameter 2	Parameter 3	Seed
DistHT	0.0000	13685	Cont Uniform	7.0	10.0		71529105
DistHT1	0.0000	40166	Cont Uniform	5.0	15.0		91080577
DistAuto	0.0000	5496	Cont Uniform	2.0	5.0		43946618
DistArrival	0.0000	26179	Cont Exponential	20.0			44944377

(Baigiamojo darbo sąžiningumo deklaracijos forma)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Andrius Bumblauskas, 20054673

(Studento vardas ir pavardė, studento pažymėjimo Nr.)

Fundamentinių mokslų fakultetas

(Fakultetas)

Informacinių sistemų programų inžinerija, ISIfm-10

(Studijų programa, akademinė grupė)

**BAIGIAMOJO DARBO (PROJEKTO)
SĄŽININGUMO DEKLARACIJA**

2012 m. birželio 10 d.

Patvirtinu, kad mano baigiamasis darbas tema „Taisyklėmis grindžiamų verslo procesų simuliacijos“ patvirtintas 2011 m. spalio 25 d. dekanų potvarkiu Nr. 371fm, yra savarankiškai parašytas. Šiame darbe pateikta medžiaga nėra plagijuota. Tiesiogiai ar netiesiogiai panaudotos kitų šaltinių citatos pažymėtos literatūros nuorodose.

Mano darbo vadovas prof. dr. Olegas Vasilecas.

Kitų asmenų indėlio į parengtą baigiamąjį darbą nėra. Jokių įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs (-usi).

(Parašas)

Andrius Bumblauskas

(Vardas ir pavardė)