

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ SISTEMŲ KATEDRA

Tadas Vysockis

DINAMINIŲ VERSLO PROCESŲ IŠTEKLIŲ MODELIAVIMAS IR SIMULIAVIMAS
DYNAMIC BUSINESS PROCESSES RESOURCES MODELING AND SIMULATION

Baigiamasis magistro darbas

Informacinių sistemų programų inžinerijos programa, valstybinis kodas 621E15002

Informacinių sistemų specializacija

Informatikos inžinerijos studijų kryptis

Vilnius, 2016

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ SISTEMŲ KATEDRA

TVIRTINU
Katedros vedėjas

_____
(Parašas)

_____
(Vardas, pavardė)

_____
(Data)

Tadas Vysockis

DINAMINIŲ VERSLO PROCESŲ IŠTEKLIŲ MODELIAVIMAS IR SIMULIAVIMAS
DYNAMIC BUSINESS PROCESSES RESOURCES MODELING AND SIMULATION

Baigiamasis magistro darbas

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

Vadovas	prof. dr. Olegas Vasilecas (Moksl. laipsnis/pedag. vardas, vardas, pavardė)	_____ (Parašas)	_____ (Data)
Konsultantas	dr. Kęstutis Normantas (Moksl. laipsnis/pedag. vardas, vardas, pavardė)	_____ (Parašas)	_____ (Data)
Lietuvių kalbos konsultantas	dr. Vaida Buivydienė (Moksl. laipsnis/pedag. vardas, vardas, pavardė)	_____ (Parašas)	_____ (Data)

Vilnius, 2016

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

Informatikos inžinerijos studijų kryptis

Informacinių sistemų programų inžinerijos studijų programa, valstybinis kodas 621E15002

Informacinių sistemų specializacija

TVIRTINU
Katedros vedėjas
_____
(Parašas)
_____
(Vardas, pavardė)
_____
(Data)

BAIGIAMOJO MAGISTRO DARBO
UŽDUOTIS

…….....Nr. .....
Vilnius

Studentui Tadui Vysockiui

Baigiamojo darbo tema: Dinaminių verslo procesų išteklių modeliavimas ir simuliavimas

patvirtinta 2015 m. spalio 22 d. dekano potvarkiu Nr. 376fm

Baigiamojo darbo užbaigimo terminas 2016 m. birželio 9 d.

BAIGIAMOJO DARBO UŽDUOTIS:

1. Atlikti su dinaminiais verslo procesai susijusių mokslinių darbų analizę. Išanalizuoti dinaminių verslo procesų modeliavimo ir simuliavimo būdus.
2. Atlikti dinaminių verslo procesų modeliavimui ir simuliacijai skirtų priemonių analizę.
3. Remiantis analize pasiūlyti dinaminių verslo procesų išteklių modeliavimo ir simuliavimo metodą.
4. Realizuoti pasiūlytą metodą ir atlikti eksperimentinį tyrimą, suformuoti galutines išvadas ir pateikti tolimesnes mokslinių tyrimų kryptis.

Baigiamojo darbo rengimo konsultantai: dr. Kęstutis Normantas
(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

dr. Vaida Buivydienė
(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

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

Užduotį gavau
……..
(Parašas)
Tadas Vysockis
(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 magistro baigiamasis darbas

Pavadinimas	**Dinaminių verslo procesų išteklių modeliavimas ir simuliavimas**
Autorius	**Tadas Vysockis**
Vadovas	**Olegas Vasilecas**

Kalba: lietuvių

Anotacija

Šiame darbe nagrinėjamas dinaminių verslo procesų išteklių modeliavimas ir simuliacija. Simuliacija – vienas iš būdų, leidžiančių atlikti įvairių verslo procesų vykdymo scenarijų analizę priklausančią nuo įvairių kintamųjų. Daugybė veiksnių gali lemti verslo procesų simuliacijos rezultatus, tačiau vienas svarbiausių veiksnių yra ištekliai, kurie reikalingi tam, kad įvykdytų veiklas. Neturint realistinio išteklių modelio yra sudėtinga gauti tikslius simuliacijos rezultatus. Todėl svarbu turėti detalų išteklių modelį, kuris leistų išvengti klaidingų simuliacijos rezultatų.

Darbe analizuojami dinaminiai verslo procesai, jų ištekliai ir verslo procesų modeliavimo bei simuliacijos įrankiai. Pasiūlytas išteklių modelis, kurio įgyvendinimas leidžia vykdyti dinaminę verslo procesų išteklių simuliaciją ir palaiko išteklių dalomumą. Remiantis pasiūlytu metodu buvo sukurta išteklių duomenų bazė ir pritaikyta pasirinktame įrankyje. Naudojantis įgyvendintu metodu buvo atliktas eksperimentinis tyrimas, parodantis dalomų išteklių svarbą gaunamiems simuliacijos rezultatams.

Darbo apimtis – 74 p. teksto be priedų, 19 pav., 10 lentelių, 41 bibliografinių šaltinių. Atskirai pateikiami darbo priedai.

Prasminiai žodžiai: Verslo procesai, dinaminiai verslo procesai, modeliavimas, simuliavimas, dinaminių verslo procesų išteklių modeliavimas, dinaminių verslo procesų 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 Graduation Thesis

Title	**Dynamic Business Processes Resource Modeling And Simulation**
Author	**Tadas Vysockis**
Academic supervisor	**Olegas Vasilecas**

Thesis language: Lithuanian

Annotation

This paper analyzes dynamic business process resources modelling and simulation. Simulation – one of the ways to perform a wide range analysis of business process execution scenario based on many variables. Numerous factors may cause simulation results of business processes, but one of the main factor are resources, which needed in order to perform activities. Whithout realistic resource model it is diffucult to obtain accurate simulation results. Therefore it is important to have a detailed resource model, which allows to avoid wrong simulation results.

This work analyzes dynamic business processes their resources and business modelling and simulation tools. A model is proposed which implementation allows to perform dynamic business process resource simulation and supports resources divisibility. Based on the proposed method a resource database was created and implemented in selected tool. A simulation experiment was carried out using implemented method, which shows the importance of divisible resources to obtained simulation results.

Thesis consists of: 74 pages of text without appendixes, 19 pictures, 10 tables and 41 references. Appendixes are supplied separately.

Keywords: Business processes, dynamic business processes, modelling, simulation, dynamic business processes resource modelling, dynamic business processes simulation

Vilniaus Gedimino technikos universiteto egzaminų sesijų ir baigiamųjų darbų rengimo bei gynimo organizavimo tvarkos aprašo
2 priedas

(Baigiamojo darbo sąžiningumo deklaracijos forma)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Tadas Vysockis, 20102069

(Studento vardas ir pavardė, studento pažymėjimo Nr.)

Fundamentinių mokslų fakultetas

(Fakultetas)

Informacinių sistemų programų inžinerija, ISIfm-14

(Studijų programa, akademinė grupė)

BAIGIAMOJO DARBO (PROJEKTO)

SĄŽININGUMO DEKLARACIJA

2016 m. birželio 09 d.

Patvirtinu, kad mano baigiamasis darbas tema „Dinaminių verslo procesų išteklių modeliavimas ir simuliavimas“ patvirtintas 2015 m. spalio 22 d. dekano potvarkiu Nr. 376fm, yra savarankiškai parašytas. Šiame darbe pateikta medžiaga nėra plagijuota. Tiesiogiai ar netiesiogiai panaudotos kitų šaltinių citatos pažymėtos literatūros nuorodose.

Parenkant ir įvertinant medžiagą bei rengiant baigiamąjį darbą, mane konsultavo mokslininkai ir specialistai: dr. Kęstutis Normantas. 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) Tadas Vysockis (Vardas ir pavardė)

TURINYS

PAVEIKSLŲ SĄRAŠAS 10
LENTELIŲ SĄRAŠAS 11
SANTRUMPOS 12
ĮVADAS 13
Tyrimo objektas 13
Darbo tikslas ir uždaviniai 13
Temos naujumas 13
Temos aktualumas 14
Tyrimo metodika 14
Mokslinė darbo vertė 14
Darbo rezultatai 14
Darbo struktūra 15
Darbo aprobacija 15
1. DINAMINIŲ VERSLO PROCESŲ, IŠTEKLIŲ, MODELIAVIMO IR SIMULIACIJOS LITERATŪROS ANALIZĖ 17
1.1. Verslo taisyklės 17
1.1.1. Verslo taisyklės apibrėžimas 17
1.1.2. Įvykis – sąlyga – veiksmas tipo verslo taisyklės 17
1.2. Verslo procesai 18
1.2.1. Verslo proceso apibrėžimas 18
1.2.2. Verslo procesų tipai 19
1.2.2.1. Statiniai verslo procesai 19
1.2.2.2. Lankstūs verslo procesai 20
1.2.2.3. Dinaminiai verslo procesai 21
1.2.3. Verslo procesų valdymas 23
1.2.3.1. Verslo procesų valdymas priklausomas nuo konteksto 24
1.2.3.2. Verslo procesų valdymas priklausomas nuo rizikų 25
1.2.3.3. Verslo procesų valdymas priklausomas nuo kainos 25
1.3. Verslo procesų modeliavimas 26
1.3.1. Verslo procesų modeliavimo kalbų apžvalga 27
1.3.1.1. Petri tinklai 28
1.3.1.2. BPMN 29

1.4. Verslo procesų simuliacija 30
1.4.1. Verslo procesų simuliacijos pliusai ir minusai 34
1.4.2. Verslo procesų simuliacijos ištekliai 35
1.5. Apibendrinimas ir rezultatai 37
2. SIMULIACIJOS ĮRANKIŲ IR IŠTEKLIŲ MODELIUI KELIAMŲ REIKALAVIMŲ ANALIZĖ 39
2.1. „ARIS Simulation“ 39
2.2. „IBM Websphere Business Modeler“ 39
2.3. „Simprocess“ 40
2.4. „Bizagi“ 41
2.5. „AnyLogic“ 42
2.6. „DBP Sim“ 42
2.7. Įrankių analizės apibendrinimas 43
2.8. Išteklių modelio reikalavimai 45
2.9. Apibendrinimas 47
3. DINAMINIŲ, TAISYKLĖMIS GRINDŽIAMŲ, VERSLO PROCESŲ IŠTEKLIŲ MODELIAVIMAS 48
3.1. Dabartinė situacija 48
3.2. Siūlomas metodas 49
3.2.1. Dalomų išteklių modeliavimas 49
3.2.2. Išteklių būsenų diagrama ir perėjimų ribojimai 50
3.2.3. Išteklių koncepcinis duomenų modelis 52
3.3. Įrankio architektūra 56
3.4. Apibendrinimas 57
4. DINAMINIŲ VERSLO PROCESŲ IŠTEKLIŲ SIMULIACIJOS PROTOTIPO REALIZACIJOS APRAŠYMAS 58
4.1. Kūrimo priemonės 58
4.2. Įgyvendintų komponentų aprašymas 58
4.3. Realizacijos aprašymas 59
4.3.1. Išteklių redaktorius 59
4.3.2. Išteklių valdytojas 61
4.3.3. Kitų komponentų aprašymas 63
4.4. Apibendrinimas 64

5. EKSPERIMENTINIS TYRIMAS 65
5.1. Nagrinėjamas uždavinys 65
5.2. Eksperimento eiga 66
5.3. Eksperimento rezultatai 66
5.4. Apibendrinimas 68
IŠVADOS 70
TOLIMESNIŲ TYRIMŲ KRYPTYS 71
LITERATŪRA 72
PRIEDAI 75

PAVEIKSLŲ SĄRAŠAS

1 pav. Įvykis – sąlyga – veiksmas taisyklės pavyzdys (Ziegler, 2015) 18
2 pav. Lankstaus verslo proceso pakitimų gyvavimo ciklas (Bhat *et al*., 2015) 21
3 pav. Verslo procesų valdymas (Interfacing, 2016) 23
4 pav. Petri tinklų pavyzdys (H. Mili *et al*., 2003) 28
5 pav. Verslo proceso simuliacijos gyvavimo ciklas (Aalst, V. D., Voorhoeve, M., 2016) 31
6 pav. „DBP Sim“ įrankio vartotojo sąsaja 43
7 pav. „DBP Sim“ įrankio architektūra (Remeika, M., 2014). 48
8 pav. Dalomų išteklių klasių diagrama 49
9 pav. Išteklių perėjimo taisyklių matematinės išraiškos 51
10 pav. Išteklių būsenų diagrama 51
11 pav. Išteklių koncepcinis duomenų modelis 53
12 pav. Išplėsta „DBP Sim“ įrankio architektūra 57
13 pav. Išteklių priskyrimo prie veiklos klasių struktūra 61
14 pav. Išteklių klasių diagrama 62
15 pav. Išteklių valdymo klasė 62
16 pav. Supaprastintas gaisro gesinimo procesas 65
17 pav. Simuliacijos rezultatai, kai hidrantas modeliuojamas kaip nedalomas išteklius (4 gaisrinės mašinos) 66
18 pav. Simuliacijos rezultatai, kai hidrantas modeliuojamas kaip dalomas išteklius (4 gaisrinės mašinos ir hidrantas palaiko 4 gaisrines mašinas vienu metu) 67
19 pav. Apibendrinti simuliacijos rezultatai 68

LENTELIŲ SĄRAŠAS

1 lentelė. Simuliacijos tipai ir jų aprašymai....33
2 lentelė. Išteklių charakteristikos....36
3 lentelė. Modeliavimo ir simuliacijos įrankių palyginimas....44
4 lentelė. Išteklių modelio reikalavimai ir jų įgyvendinimo modelyje paaiškinimai....53
5 lentelė. Išteklių charakteristikų pavyzdys....55
6 lentelė. Išteklių priskyrimo pavyzdys....55
7 lentelė. Sukurtų funkcijų aprašymas....59
8 lentelė. Patobulintų funkcijų aprašymas....61
9 lentelė. Išteklių valdymo klasės metodai....63
10 lentelė. Apibendrinti simuliacijos rezultatai....68

SANTRUMPOS

ADBVS – aktyvios duomenų bazių valdymo sistemos

DBVS – duomenų bazių valdymo sistemos

DVP – dinaminiai verslo procesai (angl. *Dynamic business processes*)

ECA – įvykis, sąlyga, veiksmas (angl. *Event-Condition-Action*)

OCL – objektų ribojimų kalba *(angl. Object Constraint Language*)

VPM – verslo procesų modeliavimas (angl. *Business Process Modelling*)

VPMN – verslo procesų modeliavimo notacija (angl. *Business Process Modelling Notation*)

VPV – verslo procesų valdųmas (angl. *Business Process Managment*)

VPVK – verslo procesų vykdymo kalba (angl. *Business Process Execution Language)*

VT – verslo taisyklė (angl. *Business Rule*)

ĮVADAS

Šiuolaikinis verslas veiklą vykdo konkurencingoje ir dinamiškoje aplinkoje, kuri verčia nuolat peržiūrėti, tobulinti savo verslo procesus ir užtikrinti gamybos lankstumą bei aukštą produktų individualizavimo lygį, o tai lemia gamybos procesų dinamiškumą. Tokiam verslui, kuris nuolat prisitaiko prie vartotojų poreikių ir konteksto pokyčių bei siekia nuolat augančio efektyvumo, reikia atsakymų į aktualų klausimą „kas bus, jeigu ..?“. Atsakymui svarbią įtaką turi tinkamas naudojamų išteklių įvertinimas. Todėl atsiranda poreikis ne tik modeliuoti dinaminių verslo procesų išteklius, bet ir juos simuliuoti. Išsprendus mokslines ir technologines dinaminių verslo procesų ir jų išteklių simuliacijos problemas, būtų įmanoma sukurti dinaminių gamybos procesų valdymo sprendimus, kurie užtikrintų optimalų gamyklos veikimą realiu laiku tiksliai reaguodami į verslo poreikius ir apribojimus. Šiuo metu tokių sprendimų nėra.

Šiame darbe nagrinėjama problema: nėra metodų leidžiančių tiksliai modeliuoti ir simuliuoti išteklius dinaminiuose verslo procesuose.

Tyrimo objektas

Tyrimo objektas – taisyklėmis grindžiamų dinaminių verslo procesų išteklių modeliavimas ir simuliacija.

Darbo tikslas ir uždaviniai

Darbo tikslas – patobulinti dinaminių, taisyklėmis grindžiamų, verslo procesų dalomų išteklių modeliavimo ir simuliavimo būdą.

Šiam tikslui pasiekti yra formuluojami tokie uždaviniai:

1. Atlikti su dinaminiais verslo procesai susijusių mokslinių darbų analizę. Išanalizuoti dinaminių verslo procesų išteklių modeliavimo ir simuliavimo būdus.
2. Atlikti dinaminių verslo procesų modeliavimui ir simuliacijai skirtų priemonių analizę.
3. Remiantis analize pasiūlyti dinaminių verslo procesų išteklių modeliavimo ir simuliavimo metodą.
4. Realizuoti pasiūlytą metodą ir atlikti eksperimentinį tyrimą, suformuoti galutines išvadas ir pateikti tolimesnes mokslinių tyrimų kryptis.

Temos naujumas

Dinaminių verslo procesų modeliavimas ir simuliavimas yra nauja ir dar iki galo neištirta sritis, kuri domina tiek mokslininkus, tiek ir verslo atstovus. Šioje srityje vis daugėja tyrimų ir susidomėjimas ja auga. Nors išteklių modeliavimas ir simuliavimas nagrinėjamas jau seniai ir yra

daugybė įrankių leidžiančių tą padaryti, tačiau šioje srityje vis dar yra neišspręstų problemų ir neatsakytų klausimų, kurie trukdo tiksliai modeliuoti ir simuliuoti verslo procesų išteklius.

Temos aktualumas

Rinkoje yra daug įrankių skirtų verslo procesų išteklių modeliavimui ir simuliacijai, tačiau šių įrankių rezultatai dažnai neatitinka realiame gyvenime gautų rezultatų (Aalst V. D., 2010). Todėl išteklių modeliavimo ir simuliavimo tema yra vis dar aktuali ir svarbi. Taip pat, ši problema yra aktuali dėl spartaus technologijų plitimo, konkurencijos didėjimo ir nuolat kintančių verslo poreikių. Dinaminiai verslo procesai kol kas nėra taikomi versle, todėl yra poreikis tobulinti ir plėsti įrankius galinčius realiu laiku optimizuoti gamybinę strategiją ir greitai įvertinti galimas alternatyvas. Šia tema yra nuolat skelbiami pranešimai aukščiausio lygio tarptautinėse moksliniuose leidiniuose ir konferencijose.

Tyrimo metodika

Analitinėje darbo dalyje, atliekant susijusios mokslinės literatūros analizę susijusią su dinaminių verslo procesų išteklių modeliavimu ir simuliacija, yra naudojama bibliotekinio tyrimo ir lyginamosios analizės metodai. Analizuojami rinkoje esantys įrankiai, jų galimybės ir išskiriami privalumai bei trūkumai. Rementis tyrimų rezultatais yra siūlomas dinaminių verslo procesų išteklių modeliavimo būdas, leisiantis tiksliau įvertinti verslo procesų išteklius simuliacijos metu. Pasiūlytas būdas yra įgyvendintas praktiškai ir atliekamas eksperimentinis tyrimas.

Mokslinė darbo vertė

Nagrinėjama tema yra nauja ir mažai išnagrinėta, todėl galima teigti, kad šio darbo sukurti įrankiai ir eksperimentai padės gilintis į problemą ir toliau plėsti darbo metu iškeltas idėjas. Remiantis šiuo darbu ateityje bus galima atlikti daugiau eksperimentų ir geriau ištyrinėti dinaminių verslo procesų išteklius.

Darbo rezultatai

1. Atlikus literatūros analizę buvo nustatyta tyrimų kryptis ir tiksliai apibrėžta problema, kurią sprendžia šis darbas.
2. Atlikus taisyklėmis grindžiamų verslo procesų modeliavimo ir simuliacijos įrankių analizę, nustatyta, rinkoje esančių įrankiu pliusai ir minusai. Pasirinktas įrankis ir apibrėžti 6 reikalavimai kuriamam išteklių modeliui.
3. Sukurtas dinaminių verslo procesų išteklių modelis.
4. Įgyvendintas išteklių modelis releacinėje duomenų bazių valdymo sistemoje.

5. Atliktas bandymas su dalomais ištekliais.

Darbo struktūra

Darbą sudaro įvadas, penki skyriai ir išvados:

Įvade iškeliama problema, suformuluojami darbo tikslai ir uždaviniai, aptariama aktuali darbo tematika ir pristatoma darbo struktūra.

Pirmame skyriuje yra aprašoma susijusių mokslinių darbų literatūros analizė, analizuojamos dalykinės srities sąvokos.

Antrame skyriuje nagrinėjamos rinkoje esantys modeliavimo ir simuliacijos įrankiai, pateikiami įrankių pliusai ir minusai. Taip pat iškeliami reikalavimai, kuriuos turi atitikti kuriamas išteklių modelis.

Trečiame skyriuje pateikiamas metodas, skirtas dinaminių verslo procesų išteklių simuliacijai. Taip pat pateikiama išplėsta įrankio architektūra.

Ketvirtame skyriuje aprašomas modelio įgyvendinimas pasirinktame simuliacijos įrankyje.

Penktame skyriuje yra aprašomas eksperimentinis tyrimas, kuris atliekamas naudojant išplėstą simuliacijos įrankio prototipą. Pateikiami gauti rezultatai.

Išvadose ir pasiūlymuose pateikiamos darbo išvados bei tolimesnių tyrimų kryptys, kurios gali būti tiliau vystomos remiantis šiuo darbu.

Darbo aprobacija

Svarbiausieji darbo rezultatai pagal tezių tematiką pateikti pranešimuose, skaitytuose:

1. 2015 m. balandžio 16 dieną, Vilniaus Gedimino technikos universitete vykusioje 18-oji jaunųjų mokslininkų konferencija „Mokslas – Lietuvos ateitis”.

Pagal tezių tematiką tarptautinėse duomenų bazėse esačiuose mokslo leidiniuose paskelbti straipsniai. Kitų tarptautinių duomenų bazių leidiniuose:

1. Kalibatiene, D., Vasilecas, O., Savickas, T., Vysockis, T., Bobrovs, V.: A New Approach on Rule and Context Based Dynamic Business Process Simulation. Baltic J. Modern Computing Vol. 4, 2016, No.3. (Priimtas)
2. Vasilecas, O., Normantas, K., Rusinaite, T., Savickas, T., Vysockis, T. Shared resource model for allocating resources to activities in BP simulation. Information and software technologies: 22th International Conference, ICIST 2016, Druskininkai, Lithuania, October 13-15, 2016. (Priimtas).

Straipsniai pakelbti kituose recenzuojamuose mokslo leidiniuose. Konferencijų pranešimų medžiagoje:

1. Rusinaitė, T., Vasilecas, O., Savickas, T., Normantas, K., Vysockis T. An approach for allocation of shared resources in the rule based business process simulation. Compsystech`16: proceedings of the 17th International conference on Computer Systems and Technologies, Palermo, Italy, June 23-24th, 2016. (Priimtas).
2. Vysockis, T.; Savickas, T.; Vasilecas O.. Dinaminiuose verslo procesuose naudojamų išteklių modeliavimas // IT sauga ir Informacinės sistemos 2015:18-oji jaunųjų mokslininkų konferencija „Mokslas – Lietuvos ateitis“. Vilnius: Vilniaus Gedimino technikos universitetas, 2015. ISSN 2029-7149, p. 24-28;
3. Rusinaitė, T.; Vasilecas, O.; Savickas, T.; Vysockis, T. Su dinaminiais verslo procesais susijusios literatūros analizė // Informacinės technologijos 2015 : XX tarpuniversitetinė magistrantų ir doktorantų konferencija : konferencijos pranešimų medžiaga / Vilniaus universiteto Kauno humanitarinis fakultetas, Kauno technologijos universitetas, Vytauto Didžiojo universitetas. Vilnius : Vilniaus universitetas, 2015. ISSN 2029-249X. P. 89-92.
4. Rusinaitė, T., Vasilecas, O., Savickas, T., Vysockis T. Selection of activities in dynamic business process simulation. :18-oji jaunųjų mokslininkų konferencija „Mokslas – Lietuvos ateitis“. Elektronika ir elektrotechnika. . Sekcija “Sistemų inžinerija, kompiuterių technologija”. Vilnius : Vilniaus Gedimino technikos universitetas, 2015. (Recenzuojamas).

1. DINAMINIŲ VERSLO PROCESŲ, IŠTEKLIŲ, MODELIAVIMO IR SIMULIACIJOS LITERATŪROS ANALIZĖ

Šiame skyriuje yra analizuojami susiję moksliniai darbai, sąvokos, metodų analizė ir pateiktos pagrindinės nagrinėjamos dalykinės srities sąvokos ir medžiaga, susijusi su nagrinėjama problema. Pateikiamos sąvokos ir apibrėžtys, kurios leis geriau suprasti analizuojamą sritį. Pateikta informacija toliau bus naudojama šiame darbe ir taip darbe bus siekiama išvengti dviprasmybių ar neteisingo terminų suvokimo.

1.1. Verslo taisyklės

1.1.1. Verslo taisyklės apibrėžimas

Verslo taisyklė (toliau VT) yra trumpas išdėstymas apie verslo aspektus. Taisyklė gali būti išreikšta terminais, kurie gali būti tiesiogiai susiję su verslu, naudojant elementarią, lengvai naudojamą ir nedviprasmišką kalbą, kuri yra prieinama visiems suinteresuotiems asmenims: verslo savininkui, verslo analitikui, programuotojui ir t. t.

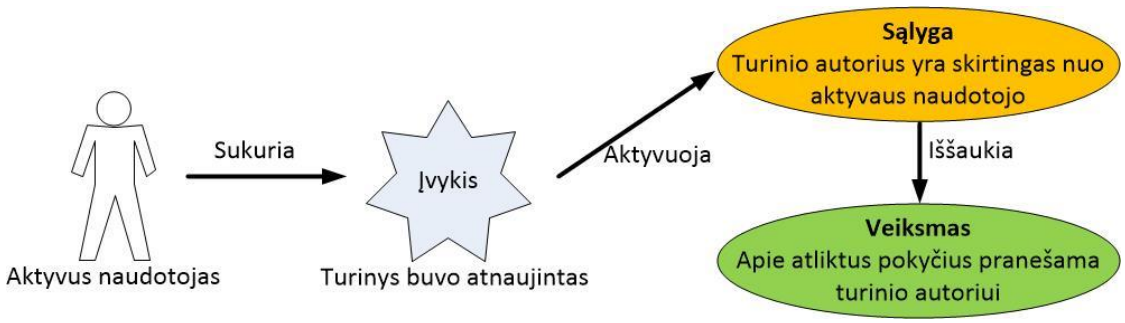
Paprastai suvokiama, kad VT yra užkoduota tai, ką galima atlikti. Bet kuriuo konkrečiu laiko mementu, turėtų būti apibrėžta galimybė nustatyti, ar sąlyga pagal numatytus apribojimus yra logiškai teisinga. Jeigu ne reikia atlikti pataisymą. Žiūrint iš programinės pusės, tai yra Būlio logika. Tai yra pagrindinė priežastis, kodėl kartu su VT yra dažnai vartojamas ir kitas apibrėžimas kaip verslo logika (Morgan *et al*., 2002).

1.1.2. Įvykis – sąlyga – veiksmas tipo verslo taisyklės

Įvykis – sąlyga – veiksmas tipo VT (angl. *Event – Condition – Action,* toliau – ECA*).* Trumpai tokią taisyklę galima užrašyti: „JEI įvykis IR sąlyga TAI veiksmas“. ECA taisyklės susideda iš trijų dalių (Ziegler, 2015).:

- Įvykis, kuris nurodo atsitikimą, kuriam esant aktyvuojama taisyklė.
- Sąlyga, tai įvykio atsitikimo metu vertinama užklausa.
- Veiksmas yra vykdomas, kai įvykis įvyksta ir sąlyga yra tenkinama.

Pateikiamas ECA tipo taisyklės pavyzdys (1 pav.), kuriame naudotojas atnaujindamas dokumento turinį sukuria įvykį. Šis įvykis aktyvuoja sąlygą, kuri patikrina, ar turinio autorius skiriasi nuo šiuo metu aktyvaus naudotojo. Jeigu skiriasi, tada atliekamas veiksmas, t. y. apie pokyčius yra informuojamas turinio autorius.



1 pav. Įvykis – sąlyga – veiksmas taisyklės pavyzdys (Ziegler, 2015)

Sąlyga specifikuoja papildomą patikrinimą, kurį būtina atlikti aktyvavus taisyklę, prieš vykdant veiksmą. ECA taisyklėse sąlyga yra nebūtina. Kai sąlyga nėra duota tuomet laikome, jog sąlyga yra visada tenkinama ir turime įvykis-veiksmas tipo taisyklę. Veiksmas ECA tipo taisyklėse aprašo užduotis, kurios turi būti įvykdytos, jeigu įvykis ir sąlyga tenkinama. Egzistuoja sistemos, kurios gali apibrėžti taisykles įvykis-veiksmas arba sąlyga-veiksmas t. y. praleidžiamas įvykis arba veiksmas. Tokios taisyklės vadinamos produkcinėmis (Vasilecas, O. *et al.*, 2008).

Verslo procesą (verslo procesų apibrėžimas pateikiamas 1.2.1 skyriuje) galima apibrėžti kaip baigtinį ECA tipo taisyklių rinkinį. Tokiu būdu sumodeliuotas verslo procesas yra vykdomas tokiu būdu: kai įvyksta įvykis, iš taisyklių rinkinio yra išrenkamas taisyklių poaibis, kuris reaguoja į taisyklėje aprašytą įvykį (Herbst, 1997). Kiekviena šio poaibio taisyklės sąlyga yra tikrinama (gali būti tikrinama tiek pagal įvykio parametrus, tiek pagal kitus) ir tokiu būdu yra sudaromas kitas, taisyklės sąlygą tenkinančių taisyklių, poaibis. Iš šio poaibio gali būti išrenkama viena taisyklė, kurios veiksmas vykdomas. Taip pat gali būti vykdomi visų sąlygas tenkinančių taisyklių veiksmai. Veiksmų vykdymo rezultatas gali būti sugeneruotas kitas įvykis į kurį gali reaguoti kitos taisyklės. Tokiu būdu gali būti konstruojama taisyklių seka, kuri atitinka verslo procesą.

Dažniausiai verslo reikalavimai yra užrašomi neformaliu taisyklių rinkiniu, todėl rinkinius transformavus į formalių ECA tipo taisyklių aibę galima suformuoti vykdomą verslo procesų modelį, kuris gali būti naudojamas simuliacijoje. Šio darbo kontekste toliau yra naudojamos ECA tipo taisyklės.

1.2. Verslo procesai

1.2.1. Verslo proceso apibrėžimas

Verslo procesas yra veiklų ir užduočių rinkinys, kuriam pasibaigus bus pasiektas organizacijos užsibrėžtas tikslas. Verslo procesas privalo turėti aiškiai apibrėžtą įvesties informaciją (angl. *input*) ir vieną išeigą (angl. *output*). Įvesties informacija yra sudaryta iš visų faktorių, kurie prisideda tiek tiesiogiai, tiek netiesiogiai prie produkto ar paslaugos pridėtinės

vertės (Appian, 2016). G. Pallas (1987) savo darbe verslo procesą suvokia, kaip loginę grandinę veiksmų, kuri yra susijusi su žmonių, medžiagų, energijos ir įrangos atliekamų veiklų modeliavimu tam, kad būtų pasiektas apibrėžtas pabaigos rezultatas. T. H. Devenportas ir bendraautoriai (1990) verslo procesą apibrėžia kaip loginių, susijusių veiksmų rinkinį, kuris yra atliekamas, tam kad pasiektume apibrėžtus verslo rezultatus.

Pagal Erikssoną ir Penkerį (1999), verslo procesas:

- Turi tikslą;
- Turi įvesties duomenis;
- Turi išeigą;
- Naudoja išteklius;
- Turi veiklas, kurios vykdomos numatyta tvarka;
- Gali įgyvendinti ne vieną organizacijos tikslą;
- Sukuria vertę klientui.

1.2.2. Verslo procesų tipai

W. Van Der Aalstas ir bendraautoriai (2003) teigia, jog kai tik pradėjo atsirasti verslo procesų valdymas (angl. *Business Process Managment*) ar darbų srautų valdymas (angl. *Workflow Managment*) labiausiai buvo susitelkta į statinius verslo procesus. Akivaizdu, jog pagrindinė to priežastis buvo ribojimai. Verslo procesų modeliavimas reikalauja daug laiko, nes tam reikia turėti paruoštą infrastruktūrą ir įgūdžius. Turimos priemonės lengvai susidoroja su statiniais verslo procesais ir tai nesudėtingai įgyvendinama, tačiau patiriami milžiniški nuostoliai tuomet, kai susiduriama su labiau dinamiška aplinka. Reaguojant į tokias situacijas buvo ir iki šiol tebėra atliekami tyrimai lanksčių bei dinaminių verslo procesų (toliau DVP) srityse. Šios sritys yra populiarios tiek tarp mokslininkų, tiek tarp verslininkų. Šiame poskyryje bus plačiau aptarti statiniai, lankstūs ir DVP bei nagrinėjama verslo procesų valdymo sritis.

1.2.2.1. Statiniai verslo procesai

Statiniai (struktūrizuoti) verslo procesai yra tokie procesai, kurie yra nekintamos formos arba, kurių pakitimai vyksta per ilgą laiką. Tokius procesus galima pagerinti naudojant valdymo priemones arba turimas žinias. Kadangi statinių procesų struktūra yra iš anksto žinoma, tokie procesai gali būti pilnai apibrėžiami struktūrizuota kalba (algoritmu). Todėl statinių procesų vykdymą, kuris nereikalauja sprendimų priėmimo, galima automatizuoti ir šias veiklas paskirti pramoniniams robotams arba kompiuteriams (Szelagowski, M., 2014).

Įgyvendinimo sritys:

- Verslo procesų gerinimo identifikavimas;
- Sprendimo priėmimo procesų identifikavimas ir standartizavimas;
- Verslo procesų modelių viešinimas visoje organizacijoje;
- Verslo procesų automatizavimas;
- Informacijos rinkimo ir analizės automatizavimas.

Nauda:

- Organizacijoje turimos žinios paaiškinamos ir suvienodinamos. Dažniausiai tai vyksta vieną kartą, verslo procesų indentifikavimo žingsnyje;
- Publikuotų ir naujausių verslo procesų modelių pasiekiamumas;
- Lengvas verslo procesų tobulinimo nustatymas;
- Visiškai kontroliuojamas realaus laiko procesų vykdymas (momentinis nukrypimų ir kylančių klaidų identifikavimas atsirandantis dėl verslo procesų vykdymo).

Rizikos ir pavojai:

- Nėra galimybės prisitaikyti prie pasikeitusių verslo sąlygų, taip pat nėra galimybės verslo procesų pritaikyti atskiram asmeniui;
- Galimas verslo procesų vykdymas standartiniu būdu, nors jis neatitinka visų verslo sąlygų (sėkmingai vykdyti įprastus veiksmus ir prisiimti sukeltus nuostolius);
- Kuriama verslo politika be atsakomybės.

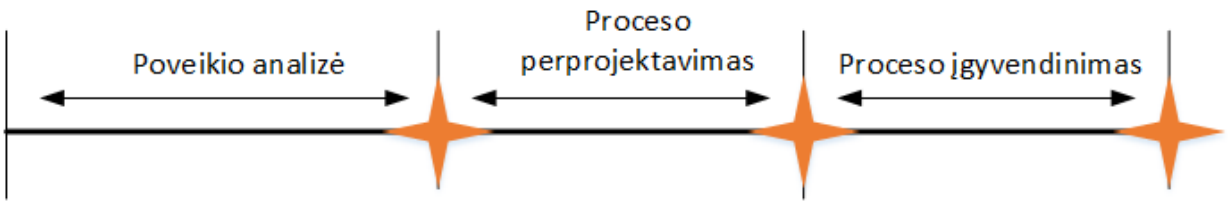
Deja, šiuolaikinėse organizacijose tik nedidelė dalis verslo procesų yra statiniai. Dažniausiai tai yra organizacijų vidiniai procesai, kurie yra nesusiję su klientų aptarnavimu ar teisiniais ribojimais.

1.2.2.2. Lankstūs verslo procesai

Lankstūs verslo procesai sugeba greitai (per priimtiną laiką) ir pigiai (sunaudojamų išteklių prasme) pasikeisti. Procesų lankstumo reikalavimai yra glaudžiai susiję su kaina. Lanksčiame verslo procese viskas yra numatyta iš anksto ir jo vykdymas kinta priklausomai nuo konteksto (pavyzdžiui: nuo aplinkos temperatūros, palūkanų normos ir t. t.). Tokie procesai, vykdymo metu, geba automatiškai prisitaikyti pagal esamą kontekstą ir sistema nusprendžia, kaip geriausia keisti vykdymo eigą, kad būtų pasiektas optimalus rezultatas (Bhat *et al*., 2005). Pavyzdžiui – konkurentams pasiūlius prekės nuolaidą, proceso vykdymas nukreipiamas kita, iš anksto apibrėžta šaka, kuri klientams taip pat pasiūlo nuolaidą. Verslo procesų pakitimų gyvavimo ciklas susideda iš 3 būsenų (2 pav.).

1) Poveikio analizės. Bandoma suprasti verslo elementų įtaką verslo procesui;
2) Proceso perprojektavimas;

3) Proceso įgyvendinimas. Pranešama visiems kam svarbūs atliekami pokyčiai.



2 pav. Lankstaus verslo proceso pakitimų gyvavimo ciklas (Bhat *et al.*, 2015)

Tinkamai įgyvendinti verslo procesai su pakankamu detalumu ir modalumu žymiai sumažina laiko sąnaudas ir reikiamų išteklių kiekį. Pakankamas detalumas užtikrina, jog poveikio analizė ir proceso įgyvendinimas atliekami greičiau ir iki galo. Modalumas palengvina proceso perprojektavimą ir įgyvendinimą taip, jog nebūtų sutrikdomas visos sistemos darbas.

Verslo procesų lankstumo klasifikavimas:

- Lankstumas verslo procesų sekoje. Pokyčių supaprastinimas veiklų vykdymo sekoje;
- Taikomų VT lankstumas. Pokyčiai VT ir nuostatose, prisitaikymas prie naujų verslo politikų;
- Lanksčios praktikos (angl. *Flexible practices*). Gebėjimas įdiegti naujas praktikas atsiradusias iš organizacijoje dirbančių darbuotojų sukauptų žinių;
- Lankstus verslo procesų išimčių apdorojimas.

1.2.2.3. Dinaminiai verslo procesai

Verslo procesai, kurie sudaryti iš veiklų ar subprocesų, kurie yra sunkiai aprašomi algoritmu arba procese yra tiek daug kintamųjų, jog jų modeliuoti yra neįmanoma. DVP vykdymo eiga priklauso nuo atskirų sąlygų vykdymo. Modeliavimo etape reikia atsižvelgti į tai, jog proceso vykdytojas gali priimti individualius sprendimus, kurių iš anksto numatyti negalime. Iš tiesų, modeliuojant ir tobulinant DVP reikia atsižvelgti ir į proceso vykdytojų žinias bei gebėjimus (Szelagowski, M., 2014).

Įgyvendinimo sritys yra tokios pačios, kaip statinių procesų, bet taip pat prisideda:

- Greitas pokyčių identifikavimas;
- Galimybė sukurti mechanizmus, kurie atranda verslo procesus iš turimų duomenų (procesų gavyba);
- Greito mokymosi mechanizmų įgyvendinimas.

Nauda yra tokia pati, kaip ir statinių procesų, bet taip pat prisideda:

- Nuolatinis verslo procesų tikrinimas ir naujų žinių kaupimas;
- Nuolatinis verslo procesų tobulinimas;
- Greitas naujų žinių panaudojimas.

Rizikos ir pavojai:

- Visada egzistuoja rizika, kad kodas nustos veikti dėl neįvertintos situacijos;
- Rizika sukurti chaosą dėl per daug nekontroliuojamų veiksnių.

Tuo metu kai yra identifikuojamas procesas, kuris yra iš prigimties dinaminis, bet bandoma jį įgyvendinti kaip statinį, toks projektas trunka ilgiau (didesnės išlaidos, atsiranda daugiau rizikų ir t. t.) ir negana to organizacijos efektyvumas krinta. Tai lemia, jog kai verslo procesas yra priklausomas nuo klientų ar verslo aplinkos yra būtina įvertinti ar proceso išskaidymas į mažesnes veiklas, kurias atlieka proceso vykdytojas, nesukels didesnių nuostolių dėl per daug sudėtingo verslo proceso. Tokiu atveju geriausia palikti bendrinį aprašymą, kuris suteiktų daugiau laisvės proceso vykdytojui. Pagrindinės rizikos atsirandančios modeliuojant DVP yra (Szelagowski, M., 2014):

1. Dėl per didelio specifikavimo lygio DVP modeliai tampa sunkiai suprantami ir keičiami. Taip atsitinka, kai bandoma specifikuoti visus galimus atvejus, nors jie ir nutinka tik tam tikromis labai specifinėmis situacijomis. Tokie verslo procesų modeliai tampa milžiniški ir vis tiek neapima visų galimų atvejų.
2. Sistemos naudotojų nepasitenkinimas. Visada egzistuoja rizika susijusi su tam tikrais organizacijos darbuotojais, kurie mato pakeitimus ne kaip galimybę dirbti geriau, bet kaip grėsmę jų privilegijomis ir darbo išsaugojimui. Informacijos, supratimo ir susitaikymo su pokyčiais stoka lemia, kad darbuotojai priešinasi procesų automatizacijai ir ne visada yra linkę atskleisti visas detalias apie organizacijoje vykdomus verslo procesus.

Zengas ir bendraautoriai (2002) pasiūlė pilnai automatizuotus verslo procesus, kuriuos automatizuojant programų sistemose yra sudaromos vykdomų darbų sekos (angl. *workflow*) ir iškėlė dinaminių darbų sekos sudarymo poreikį, kai apibrėžta darbų seka yra keičiama darbų sekos vykdymo metu. Tokio poreikio priežastys yra tai, kad keičiantis verslo poreikiams, programų sistemose taip pat reikia keisti vykdomų darbų eigos seką. Tam kad dinamiškai sugeneruotume darbų sekos modelį reikalingos VT. Taip modeliuojamų DVP privalumai:

- Automatizuojant DVP nebūtina numatyti iš anksto ir žinoti visų įmanomų variantų, kuriuos turi įgyvendinti procesas, pakanka koncentruotis į tikslą. Paaiškėjus

naujiems poreikiams ar pasikeitus reikalavimams, DVP gali būti keičiamas vykdymo metu;

- Pakeitus vykdomus DVP, juos nereikia paleisti iš naujo, todėl taupomas laikas.

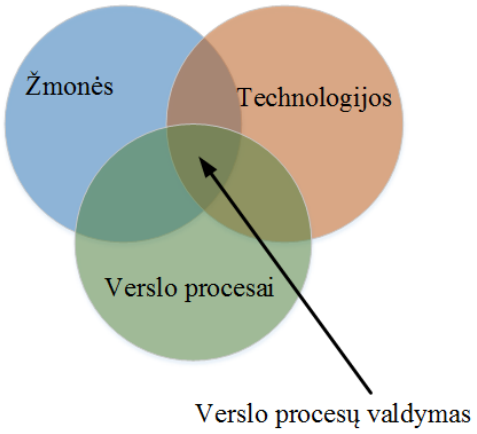
DVP jų vykdymo metu gali keisti išoriniai veiksniai. Tokį procesą gali keisti žmogus arba kita automatizuota sistema (taip pat gali būti ir kitas procesas). Proceso dinamiškumas gali būti tiek būsenos keitimas jos vykdymo metu, tiek jos struktūros keitimas. Proceso keitimas vykdymo metu gali pasireikšti:

- Įterpiamomis naujomis veiklomis arba jau esamų veiklų pašalinimu;
- Veiklos prasmės pasikeitimu;
- Loginių sąlygų keitimu.

Galimybė vykdymo metu keisti DVP ypač reikalinga nenumatytiems konteksto pokyčiams, nes vykdomų DVP nutraukimas kainuoja brangiai ir tai yra neefektyvu. Toliau šiame darbe bus nagrinėjami DVP.

1.2.3. Verslo procesų valdymas

Verslo procesų valdymas (toliau VPV) tai sisteminis požiūris į organizacijoje vykdomus darbų srautus tam, kad jie būtų atliekami efektyviau, veiksmingiau ir padaryti darbų srautus pajėgesnius prisitaikyti prie nuolatos besikeičiančios aplinkos. VPV tikslas yra sumažinti žmogiškųjų klaidų ir nesusikalbėjimo atvejų skaičių (M. Rouse, 2011). 3 paveiksle yra pavaizduota, jog norint teisingai suvaldyti verslo procesus reikia apjungti tiek ir žmones, tiek ir technologijas.



3 pav. Verslo procesų valdymas (Interfacing, 2016)

Galime išskirti penkis pagrindinius VPV gyvavimo ciklo etapus:

- Projektavimas (angl. *Design*) – ši veikla apima egzistuojančių verslo procesų identifikavimą įmonėje ir reikalingų naujų verslo procesų numatymą;

- Modeliavimas (angl. *Modeling*) – projektavimo metu apibrėžtų procesų modelių kūrimas, analizė, įvairių rizikų bei situacijų planavimas;
- Automatizavimas bei vykdymas (angl. *Execution*) – kuriama programinė įranga, automatizuojanti tam tikras verslo procų dalis bei optimizuojanti darbą;
- Stebėjimas (angl. *Monitoring*) – individualių verslo procesų vykdymo stebėjimas ir kontrolę, statistinių duomenų rezultatų kaupimas;
- Optimizavimas (angl. *Optimization*) – šios veiklos metu stebėjimo ir modeliavimo fazėse gauta informacija apdorojama ir ieškoma potencialių verslo procesų tobulinimo galimybių. Identifikavus tokias galimybes, ciklas vėl prasideda nuo projektavimo fazės ir taip nuosekliai ir nenutrūkstamai tobulinami bei keičiami verslo procesų modeliai ir jų realizacija.

Reckeris ir C. Janas (2011) išskiria pagrindines VPV kryptys t. y.: VPV priklausomas nuo konteksto (angl. *Context-aware*), VPV priklausomas nuo tam tikrų rizikų (angl. *Risk-aware*) ir VPV priklausomas nuo kainos (angl. *Cost-aware*). Toliau šios kryptys yra detalizuojamos.

1.2.3.1. Verslo procesų valdymas priklausomas nuo konteksto

Kontekstą galima apibūdinti kaip bet kokią informaciją, kuri gali paveikti verslo procesų vykdymą. Nagrinėjant verslo procesus, kuriems daro įtaką kontekstas, svarbu tinkamai apibrėžti, koks kontekstas yra aktualus verslo procese. Reckeris ir C. Janas (2011) savo darbe išskyrė tokias konteksto rūšis: tiesioginis, vidinis ir išorinis.

Daugelyje organizacijų verslo procesai priklauso nuo išorės konteksto kaip oro sąlygos, prekių kainos ar pramoniniai veiksmai. Vienas pagrindinių trūkumų yra, jog VPV priklausomas nuo konteksto nėra pritaikytas tipinės VPV sprendimams. Šiuo metu į tipinius procesų modeliavimo metodus nėra įtraukti konteksto kintamieji, tačiau vietoje to sutelkiamas dėmesys į įmonės viduje vykstančius procesus, tokius kaip transformacijos. Tuo tarpu VPV priklausomas nuo konteksto susitelkia į konteksto identifikavimą, koks kontekstas daro įtaką įmonės procesams.

Vidinis kontekstas apibūdina organizacijos aplinką, kuri verslo procesus paveikia netiesiogiai kaip išteklius, tikslus, siekius, vertybes, strategijas ir organizacijos struktūrą.

Tiesioginiam kontekstui priskiriama tokia informacija, kuri veikia verslo proceso vykdymo eigą. Daugiausiai tai yra ištekliai, oro sąlygos, informacija apie konkurentų darbus ir t. t.

Atsižvelgiant į konteksto daromą poveikį verslo procesams reikia sugebėti apdoroti ir prisitaikyti prie greitai besikeičiančių kintamųjų, todėl reikia sukurti dinamišką verslo procesų prisitaikymo prie kontekstinės informacijos mechanizmą.

1.2.3.2. Verslo procesų valdymas priklausomas nuo rizikų

Panašiai kaip konteksto įtakos nustatymas, kuris veikia verslo proceso vykdymą ar valdymą, buvo pradėta tirti, kaip rizikos, jų sušvelninimas ir valdymas lemia visą verslo proceso valdymą. VPV priklausomas nuo rizikų sujungia dvi standartiškai atskiras disciplinas t. y. procesų valdymas ir rizikų valdymas. Toks tradicinis atskyrimas lemia neveiksmingumą, nes sprendimai dažnai gali būti prieštaringi ir nesusiję tarpusavyje. Todėl VPV priklausomo nuo rizikų idėja yra, jog verslo procesų valdymas teiktų informaciją ne tik apie verslo procesus, bet ir apie saugumo problemas (Recker ir Jan, C., 2011).

Šiuo metu yra sukurta naujų metodų leidžiančių įvertinti rizikas verslo procesų modeliuose (Neiger, D. *et al.*, 2009). Šie metodai leidžia nustatyti iš kur kyla rizikos verslo procesuose, kokios gali būti pasekmės ir kokios kontroliavimo ir mažinimo strategijos turėtų būti įgyvendintos tam tikroje vietoje. Tuo metu kai verslo procesas yra specifikuojamas jutikliai gali identifikuoti tam tikras rizikos sąlygas, jeigu naudojama paskirstytoji, jutikliais paremta architektūra. Tiek istorinė, tiek ir dabartinė procesų vykdymo informacija gali būti panaudota tam, kad būtų surastos tokios rizikų sąlygos. Vykdymo metu kiekvienas jutiklis atskirai praneša apie savo būklę jutiklių valdikliui, tuo metu kai aptinka riziką. Tuo tarpu jutiklių valdiklis gali apsikeisti duomenimis su stebėjimo (angl. *Monitoring*) komponentu ir taip pranešti naudotojui apie galimas problemas.

1.2.3.3. Verslo procesų valdymas priklausomas nuo kainos

Verslo procesų kainos nustatymas yra viena iš priemonių susijusių su VPV priklausomu nuo kainos analize ir finansų valdymu. Šio VPV užduotis yra suvaldyti organizacijos išlaidas, planuojant jas iš anksto ir priimant pagrįstus strateginius sprendimus, paremtus kaina. Tačiau, tokia sąnaudų analizė gali būti ribota, jeigu neturėsime visų duomenų apie visas vykdomas veiklas organizacijoje ir jų pilną kainą. Tuo metu kai vykdomas detalus ataskaitų rengimas ir atliktas auditas dažniausiai jau būna per vėlu koreguoti sąnaudas tam laiko periodui. Todėl kiekvienas vadybininkas norėtų sugebėti atidžiai stebėti bendrą išlaidų procesą ir padaryti būtinus pakeitimus su susijusiais procesais. Tam, kad tai pavyktų reikalingi įrankiai kurie sugebėtų tą įvertinti.

VPV priklausomame nuo kainos bandoma pritaikyti VPV principus ir įrankius tam, kad būtų galima valdyti organizacijos išlaidas veiklų lygyje. Verslo procesų modelyje yra aprašomas veiklos vykdymas, laikas kada ji turi startuoti, kas turi ją startuoti ir kokios informacijos veiklai

reikia ir kokia informacija bus gauta. Taip pat VPV sistemos leidžia stebėti procesus ir patikrinti įvairius nustatymus. Tačiau, šiuo metu tokios sistemos tipiškai nėra priklausomos nuo kainos ir procesų vykdymas nėra sutelktas į kainą (angl. *cost-driven*). Gebėjimas suprasti verslo procesų veiklų sąnaudas ir informuoti apie kainų pakitimus realiu laiku leistų organizacijoms padidinti savo pelningumą ir anksti sureaguoti į netikėtus kainų šuolius.

1.3. Verslo procesų modeliavimas

Paprastai verslo procesų modeliavimas (angl. *Business Process Modellling* arba BPM) yra diagrama, kuri pavaizduoja veiklų seką. Dažniausiai tokios diagramos parodo įvykius, veiksmus ir nuorodas (angl. *links*) arba sujungimo taškus (angl. *connection points*) vienoje eilėje nuo pradžios iki galo. Verslo procesų modeliavimas apima tiek informacinių sistemų atliekamus procesus, tiek ir žmonių vykdomus procesus.

Verslo proceso modelio diagrama yra įrankis, kuris parodo pabaigą, o ne savaime veiklos rezultatą (Chapman, A., 2016). Galutinis rezultatas yra verslo proceso veikimo patobulinimas. Patobulinimai dažniausiai vykdomi pridėtinės vertės veiklose, kurios padaro klientų aptarnavimą arba jų patirtį malonesnę per trumpesnį laiką ir už mažesnę kainą. Išskiriami du pagrindiniai verslo procesų modeliavimo tipai:

- Toks koks yra (angl. *As is*) arba pradinis modelis (vaizduojama dabartinė situacija);
- Siekiamas (angl. *to be*) modelis (vaizduojama siekiama nauja situacija).

Šie tipai naudojami verslo procesų analizei, testavimui, įgyvendinimui ir tobulinimui. Modeliavimo tikslas yra pavaizduoti pilną verslo procesą, leidžiantį vadybininkams, konsultantams ir darbuotojams pagerinti proceso vykdymą. Verslo procesų modeliavimo rezultatai lemiantys didėjantį pelną:

- Vertės klientui pridėjimas;
- Kainos sumažinimas kompanijai.

Kitos antraeilės pasekmės atsirandančios dėl sėkmingo verslo procesų modeliavimo gali būti padidėjąs konkurencingumas, rinkos augimas ir geresnis organizacijoje vykdomų verslo procesų suvokimas tarp skirtingų darbuotojų (Chapman, A., 2016). Verslo procesų modeliavimas yra efektyvi metodika tuomet, kai yra nukreipta į verslo procesus. Verslo procesų modeliavimas suteikia galimybė:

- Grafiškai atvaizduoti verslo procesų seką;
- Kontroliuoti, analizuoti ir geriau suprasti verslo procesus ir juos sudarančius darbų srautus;

- Įvertinti nesėkmės poveikį verslo procesui;
- Pakeisti arba pagerinti verslo procesų vykdymą;
- Sukurti modelius vaizduojančius geresnį būdą, kaip atlikti konkrečius procesus;
- Ištestuoti alternatyvas ir nustatyti, kurie iš jų yra naudingiausi verslui;
- Vizualiai pademonstruoti rezultatus, daugeliui suprantama kalba.

Toliau darbe yra pateikiama detalesnė verslo procesų modeliavimo kalbų apžvalga.

1.3.1. Verslo procesų modeliavimo kalbų apžvalga

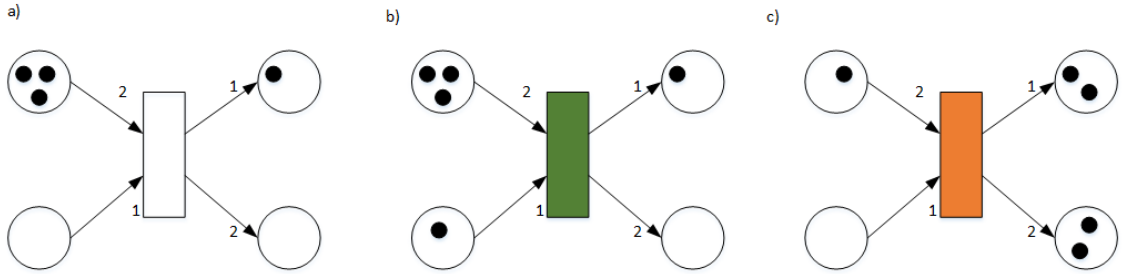
Kalbos kurias nagrinėsime šiame skyriuje apima skirtingus verslo procesų aspektus ir gali būti daugiau ar mažiau formalios priklausomai nuo paskirties ir auditorijos. H. Milis ir bendraautoriai (2003) išskiria 4 verslo procesų modeliavimo kalbų grupes:

1. Tradicinės modeliavimo kalbos. Dauguma šių kalbų atsirado iš informacijos inžinerijos (angl. *information engineering*) arba iš verslo procesų inžinerijos (angl. *business process engineering*). Su viena išimtimi (Petri tinklai) visos šios kalbos yra sunkiai suprantamos žmonėms. Dažniausiai šios kalbos nėra formalios, tačiau gali būti naudojamos įvairiuose neformaliuose arba euristinėse analizėse. Tradicinės modeliavimo kalbos yra: IDEF, Petri tinklai, Įvykiais paremtų procesų grandinė (angl. *Event Process Chains* arba EPC) ir kitos kalbos;
2. Darbų srautų modeliavimo kalbos. Darbų srautų valdymo sistemos yra kompiuterinės sistemos, kurios valdo verslo procesus priskirdamos prie veiklos reikiamus išteklius. Darbų srautų modeliavimo kalbos yra scenarijų (angl. *script*) kalbos, kurios aprašo darbų srautus taip, kad juos būtų galima panaudoti darbų srautų valdymo sistemose. Šios kalbos yra dauguma formalios ir jas galima tiesiogiai transformuoti į programinį kodą. Kalbos pavyzdys būtų darbų srautais paremtų procesų aprašymo kalba (angl. *Workflow Process Description Language* arba WPDL);
3. Procesų susiejimo kalbos. Elektroninis verslas (angl. *Business to Business* arba B2B) tarp skirtingų įmonių paskatino susidomėjimą verslo modeliavimo kalbomis, kurios būtų skirtos susieti du arba daugiau verslo partnerius. Šios kalbos daugiausia dėmesio skiria mechanizmams, kurių paskirtis yra abstrakčiai, nepriklausomai nuo naudojamų technologijų susieti skirtingus verslus. Kalbų pavyzdžiai: RosettaNet4, jau minėti Petri tinklai, BPMN, ebXML ir BPEL4WS.
4. Į objektus orientuotos kalbos. Į objektus orientuotos modeliavimo kalbos yra giriamos nuo pat jų atsiradimo pradžios dėl savo natūralaus pasaulio vaizdavimo būdo tokiu būdu, jog IT specialistai ir organizacijos darbuotojai gali suprasti vieni kitus. Riba tarp verslo procesų

modeliavimo ir programinės įrangos modeliavimo tapo aiškiai apibrėžta ir galima teigti, kad į objektus orientuotos modeliavimo kalbos yra skirtos daugiau programiniam kodui kurti, o ne modeliuoti verslo procesus. Į objektus orientuotos kalbos pavyzdys būtų UML2.

1.3.1.1. Petri tinklai

Petri tinklas yra specifinis grafas, kurio tikslas atvaizduoti dinaminės sistemos elgseną (Genrich *et al.*, 1981). Petri tinklai plačiai naudojami todėl, kad turint mažą konstruktorių skaičių (žetonai, sąlygos ir perėjimai arba įvykiai) galima atvaizduoti gana sudėtingą elgseną, gana glaustu būdu. Taip pat, modeliai sukurti naudojant Petri tinklus gali būti formaliai analizuojami ir patikrinami (Genrich *et al.*, 1981). Petri tinklas yra specifinis grafas, kuris turi dviejų rūšių mazgus, sąlygas ir perėjimus, ir tiesioginius lankus (su svoriu) tarp jų. Sąlygos, kurios yra prieš perėjimą yra vadinamos įeigos sąlygos, o sąlygos po perėjimo – išvesties sąlygomis. Perėjimas yra aktyvuotas tik tuo atveju, jeigu sąlyga turi bent jau tiek žetonų koks yra lanko svoris, kuris yra susietas su perėjimu. 4 paveikslo a dalyje matome perėjimą, kuris yra neaktyvuotas, nes nėra žetono dešinėje apatiniame apskritime. Atsiradus žetonui, perėjimas yra aktyvuojamas tai matome paveiksle 4 b. Kai perėjimas yra aktyvuojamas, tuomet žetonai yra perkeliami atitinkamai pagal lankų svorius į už perėjimo esančias sąlygas. 4 paveikslo c dalis rodo Petri tinklą po to kai perėjimas buvo įvykdytas.



4 pav. Petri tinklų pavyzdys (H. Mili *et al.*, 2003)

Galima įrodyti, jog baigtinės talpos Petri tinklą galima transformuoti į lygiavertį begalinės talpos tinklą (Genrich *et al.*, 1981). Petri tinklai yra tinkami atliekant formalę analizę. Formaliai patikrinti galime tokius tinklo nustatymus:

1. Pasiekiamumas, tai Petri tinklo būsena kai tinklas apibūdinamas, kaip skirtingų sąlygų kiekis (žymėjimas). Žymėjimas $\check{Z}_n$ yra pasiekiamas iš pradinio žymėjimo $\check{Z}_0$, jeigu egzistuoja perėjimų skaičius, kuris nuvestų iš $\check{Z}_0$ į $\check{Z}_n$;
2. Ribotumas. Petri tinklas yra ribotas, jeigu duotam pradiniam žymėjimui žetonų skaičius, bet kokioje sąlygoje niekada neviršija baigtinio skaičiaus k;

3. Gyvumas (angl. *Liveness*). Petri tinklas su pradiniu žymėjimu yra gyvas, jeigu kiekvienas perėjimas yra pasiekiamas iš pradinio žymėjimo;
4. Grįžtamumas ir pradinė būsena. Petri tinklas yra grįžtamas, jeigu kiekvienam iš žymėjimų $\check{Z}$, kuris yra pasiekiamas pro pradinę būseną $\check{Z}_0$, egzistuoja baigtinis skaičius perėjimų, kurie gražintų tinklą iš būsenos $\check{Z}$ į būseną $\check{Z}_0$ (ar į kurią kitą būseną, kuri yra nustatyta kaip pradinė).

Faktas jog šie nustatymai yra teoriškai patikrinami nereiškia, jog jie yra atsekami iš skaičiavimų. Dabartiniai tyrimai atliekami su Petri tinklais stengiasi juos padaryti atsekamus iš skaičiavimų.

Petri tinklai aiškiai yra skirti modeliuoti elgseną. Visų kitų perspektyvų šiam modeliavimo būdui trūksta, nors ir yra sukurti papildymai su turtingesniu žodynu, kuriuo galima modeliuoti ir struktūrizuotą informaciją. Petri tinklai nėra skirti komunikavimui su verslo žmonėmis, nors ir daug tyrėjų yra įvertinę šio modelio analizės ir tikrinimo galimybes. Petri tinklas yra žemo lygio kalba beveik kaip asemblerio lygio programavimo kalba modeliavime.

1.3.1.2. BPMN

Verslo procesų modeliavimo notacija (angl. *Business Process Modelling Notation* arba BPMN) tai standartas skirtas verslo procesų modeliavimui, kuris užtikrina grafinę notaciją specializuotiems verslo procesams. BPMN užduotis yra palaikyti VPV suprantamą tiek techniniams vartotojas, tiek verslo vartotojams, pateikdami notaciją, kuri yra intuityviai suprantama, bet kartu sugeba atvaizduoti sudėtingus procesus (Object Management Group, 2014). BPMN specifikacija taip pat užtikrina galimybę sujungti grafinį modelį su konstrukcinėmis vykdymo kalbomis, konkrečiai verslo procesų vykdymo kalba (angl. *Business Process Execution Language* arba BPEL).

Pirminis BPMN uždavinys yra pateikti notacijų standartą lengvai suprantamą visoms suinteresuotoms verslo dalims. Taip pat įskaitant ir verslo analitikus, kurie kuria ir tobulina procesus, programuotojus, kurie yra atsakingi už procesų įdiegimą ir verslo vadovą, kuris prižiūri ir palaiko juos. BPMN yra naudojama kaip bendra kalba, kuri padeda panaikinti klaidas kylančias tarp verslo proceso kūrimo ir vykdymo.

Įvykis tai kažkas kas nutinka proceso eigoje. Įvykiai veikia proceso vykdymo eigą ir paprastai turi priežastį arba poveikį, kas dažniausiai iššaukia reakcinius veiksmus. Termino „įvykis“ dažniausiai pakanka apimti daugelį dalykų procese. Veiklos pradžią, veiklos pabaigą, dokumento statuso pasikeitimą, gautą pranešimą ir t. t. visa tai gali būti laikoma įvykiais. BPMN

standarte naudojantis įvykiais galima aprašyti verslo procesus priklausomus nuo tam tikrų įvykių. Šiuose procesuose yra trys pagrindiniai įvykių tipai (Camunda, 2014):

1. Pradžios įvykiai, kurie parodo kur procesas startuos.
2. Pabaigos įvykiai, kurie parodo kur proceso kelias pasibaigs.
3. Tarpiniai įvykiai, kurie parodo kur kažkas įvyks, kažkur tarp proceso pradžios ir pabaigos.

Visi įvykiai skirstomi į dvi pagrindines rūšis (Object Management Group, 2014):

1. Kurie reaguoja į trigerį. Visi pradžios įvykiai ir kai kurie tarpiniai įvykiai yra reaguojantys įvykiai;
2. Kurie sukelia rezultatą. Visi pabaigos įvykiai ir kai kurie tarpiniai įvykiai yra sukeliantys rezultatą įvykiai, į kuriuos galiausiai gali reaguoti kitas įvykis. Trigerio iššaukimas taip pat gali būti numanomas.

1.4. Verslo procesų simuliacija

Simuliacija – realaus gyvenimo tikros arba tikėtinos situacijos, sąlygos arba įvykio imitavimas tam, kad surasti priežastį dėl ko praeityje įvyko tam tikras įvykis (nelaimės atveju) arba tam, kad prognozuoti būsimas pasekmes (BusinessDictionary, 2015). Simuliacija gali būti atlikta per:

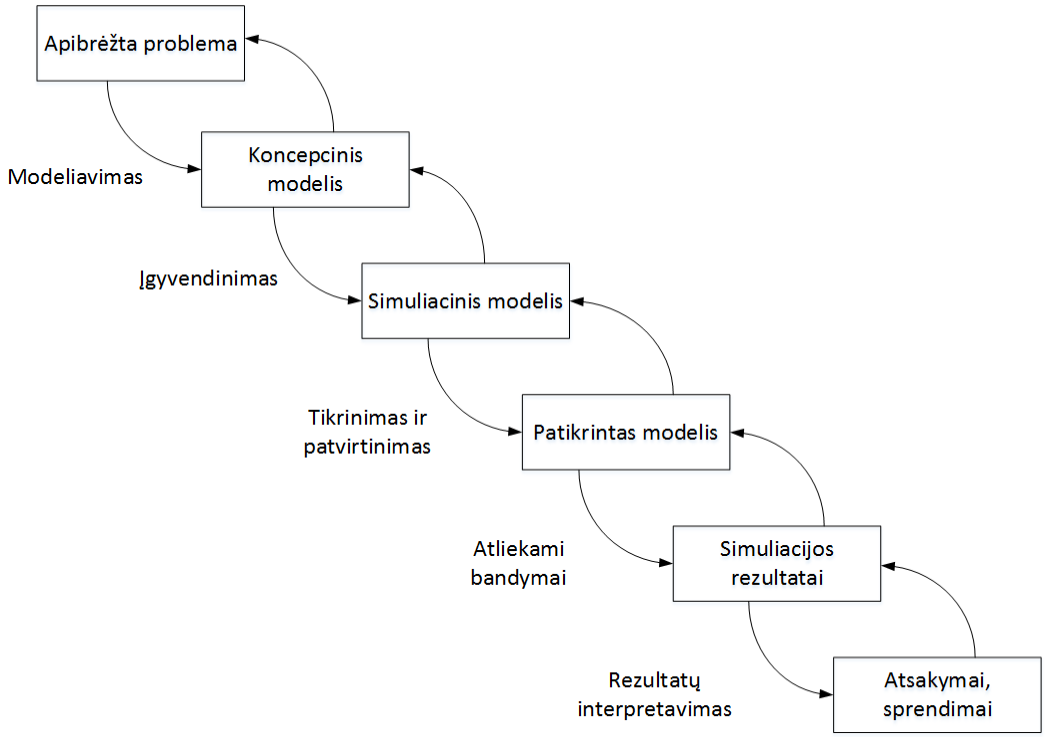
- Matematinį modelį, išsprendus aibę lygčių;
- Fizinį modelį, sukūrus nagrinėjamą problemą atitinkantį realų modelį;
- Suplanuotas repeticijas, kurių metu atkartojami visi žingsniai;
- Kompiuterį modelį.

Naudojant simuliaciją taip pat galima įvertinti ir sukurtą modelį, jo realios problemos atvaizdavimą. Maria ir kiti autoriai (1997) teigia, kad turint konkretų modelį dažnai prireikia nustatyti jo teisingumą ir efektyvumą. Tuo tikslu atliekami eksperimentai – modelių simuliacija. Prieš atliekant modelių simuliaciją, yra apibrėžiami modelių simuliacijos rezultatų vertinimo kriterijai, pavyzdžiui – kaina, laikas ir t. t. Modelio simuliavimas leidžia nustatyti modelio silpnąsias vietas ir jį patobulinti. Modelių simuliacija, kuri yra artima realiam pasauliui, gali pakankamai tiksliai suteikti reikiamą informaciją.

Modeliuojant verslo procesus ir siekiant ištirti jų efektyvumą būtina visų pirma juos išanalizuoti. Todėl yra atliekama verslo procesų simuliacija, įvairūs pakeitimai modeliuose ir taisyklėse, siekiant įvertinti jų įtaką ir pokyčių pasekmes. Toks pakeitimų įvertinimo būdas yra žymiai pigesnis negu juos vertinant atliekant realius sprendimus. Todėl buvo sukurtos įvairios

verslo procesų simuliacijos priemonės, kurios leidžia modeliuoti, atkartoti procesus ir padeda suvokti sprendimų poveikį. Verslo procesų simuliacija padeda atsakyti į šiuos klausimus:

1. Kaip galime pasiekti užsibrėžtus veiklos tikslus?
2. Kada turėtume išteklių skaičių padidinti, o kada sumažinti?
3. Koks bus poveikis, jei atliksime funkcinius pokyčius?



5 pav. Verslo proceso simuliacijos gyvavimo ciklas (Aalst, V. D., Voorhoeve, M., 2016)

Simuliacija dažnai naudojama priimant strateginius sprendimus. Todėl tai reikalauja tinkamos problemos analizės, modelio sukūrimo, atliktos simuliacijos ir rezultatų interpretavimo, kaip parodyta 5 paveiksle. Naudotojui draugiškos (angl. *User friendly*) priemonės leidžia dažną prototipavimą, tačiau pavojinga priimti strateginius sprendimus remiantis prototipais. Dažnai naudojamas krioklio (angl. *waterfall*) būdas, kuris iliustruoja simuliacinio projekto gyvavimo ciklą, kaip parodyta 5 paveiksle. Reikia pabrėžti, jog kitas žingsnis gali prasidėti ir nepasibaigus prieš jį buvusiam žingsniui bei atliekamą žingsnį gali įtakoti jo įpėdiniai (Van Der Aalst, W. ir Voorhoeve, M. 2016).

- Problemos apibrėžimas. Simuliacijos projektas prasideda nuo problemos apibrėžimo, kuris nusako tikslus ir nusako simuliacijos sritis. Taip pat yra apibrėžiama kas patenka į simuliacijos modelį, o kas ne. Problemos apibrėžime turi būti iškeltas klausimas į kurį simuliacija turi atsakyti. Klausimai turėtų būti kiekybiniai (suskaičiuojami) ir dėmesys turėtų būti sutelktas į parinkimą, o ne į optimizavimą. Vietoje to, jog klausimą formuluoti

„koks yra geriausias baro veikimo grafikas, kad klientai būtų patenkinti?“ reikėtų klausti „Kiek vidutiniškai klientai turi laukti eilėje, kai tvarkaraštis yra A, B arba C?“, kur A, B ir C yra baro darbuotojų tvarkaraščiai apibrėžti iš anksto;

- Po to kai yra apibrėžiama problema prasideda modeliavimo fazė. Šioje fazėje yra sukuriamas koncepcinis modelis. Šiame modelyje yra nurodomi objektų klasės ir sąryšiai tarp šių objektų. Taip pat svarbūs objektų nustatymai (angl. *properties*) yra išdėstomi koncepciniame modelyje. Modelyje reikia atskirti kokybinius ir kiekybinius nustatymus. Pavyzdžiui kokybinės charakteristikos galėtų būti kaip klientai yra aptarnaujami eilėje (pirmas atėjo pirmas išėjo (angl. *Fist In First Out*) arba atsitiktinai) taip pat nurodomi, kokie veiksmai atliekami atėjus klientui. Kiekybinių charakteristikų pavyzdžiai galėtų būti, kiek klientų yra aptarnaujami per minutę arba kiek klientų gali būti aptarnaujami tuo pačiu metu (lygiagretumas). Koncepcinio modelio sukūrimas greičiausiai atskleis neišsamius ir priešingus aspektus problemos apibrėžime. Taip pat, verslo procesų modeliavimas gali iššaukti naujus klausimus į kuriuos turėtų atsakyti simuliacija. Problemos apibrėžimas turėtų būti atitinkamai pritaikytas;
- Po koncepcinio modelio sukūrimo prasideda įgyvendinimo fazė. Šiame žingsnyje koncepcinis modelis yra susiejamas (angl. *mapped*) su simuliacijos modeliu. Simuliacijos modelis gali būti simuliuojamas kompiuteriu. Šio modelio sukūrimas priklauso nuo to, kokia simuliacijos priemonė yra pasirinkta;
- Simuliacijos modelis nebūtinai yra teisingas, todėl jį reikia patikrinti ir patvirtinti. Tikrinimo žingsnyje svarbu nustatyti ar visi nustatymai yra modelyje ir ar neliko programavimo klaidų. Tam, kad patikrintume modelį gali būti vykdomi testavimai nepalankiomis sąlygomis (angl. *stress test*). Nepalankių sąlygų testavimas patikrina modelį kraštinėse situacijose t. y. didesnis klientų atvykimas nei galima aptarnauti. Tokiu atveju laikui bėgant eilė turėtų nuolatos augti. Patvirtinimo metu modelį reikia palyginti su duomenimis, kurie leistų įsitikinti modelio teisingumu. Simuliacijos rezultatus galima patikrinti su sistemos istoriniais duomenimis ir įsitikinti, jog modelis yra teisingas. Patikrinimo ir patvirtinimo žingsnis gali lemti simuliacijos modelio korekcijas. Naujos įžvalgos gali lemti ir problemos apibrėžimo pokyčius. Patikrintas ir patvirtintas modelis tampa patvirtintu modeliu;
- Kitas žingsnis yra bandymų atlikimas. Šie bandymai turėtų gauti patikimus rezultatus. Simuliacijos eksperimentas yra paremtas patikrintu modeliu, kuriuo remiantis kompiuteris sugeneruoja simuliacijos metu atsitiktinius pavyzdžius iš nurodytų tikimybinių skirstinių.

Standartinė praktika yra simuliacijas leisti kelis kartus su skirtingais atsitiktiniais duomenimis tam, kad patikrintume ar rezultatai panašūs.

- Gauti simuliacijos rezultatai turi būti interpretuojami, tam kad būtų galima atsakyti į problemos apibrėžimo metu iškeltus klausimus. Turi būti apskaičiuoti patikimumo intervalai įvairiems parametrams gautiems per simuliaciją. Taip pat kokybiniai rezultatai turi būti interpretuojami ir nustatomi ar atsako į iškeltus klausimus. 1 lentelėje yra pateikiami simuliacijos tipai ir jų aprašymai.

1 lentelė. Simuliacijos tipai ir jų aprašymai

Simuliacijos tipas	**Aprašymas**
Stochastinė simuliacija	Tai kompiuterinio simuliavimo technika, kuri naudoja atsitiktinius pavyzdžius ir kitus statistinius metodus tam, kad surastų apytikslius sprendimus matematinėms arba fizikinėms problemoms. Stochastinė simuliacija sukuria projekcija, kuri yra paremta atsitiktinėmis reikšmėmis. Simuliacijos rezultatai yra įrašomi ir simuliacija yra pakartojama su nauja aibe atsitiktinių duomenų. Šie žingsniai yra kartojami kol pakankamas duomenų kiekis yra surenkamas. Po to rezultatų pasiskirstymas parodo labiausiai tikėtinus įverčius taip pat rezultatų rėžius į kuriuos patenka simuliacijos rezultatai (Simulace, 2011).
Diskrečiųjų įvykių simuliacija	Diskrečiųjų įvykių simuliacija tai procesas, kurio metu užkoduojama sudėtinga sistema, kaip išrikiuotų gerai apibrėžtų įvykių seka. Įvykis apima specifinį pokytį sistemos būsenoje, tam tikru specifiniu metu. Diskrečiųjų įvykių simuliacijos programos dažniausiai apima testavimą nepalankiomis sąlygomis, galimų finansinių investicijų įvertinimą ir procedūrų ir verslo procesų modeliavimą įvairiuose organizacijose. Diskrečiųjų įvykių simuliacijos pavyzdys galėtų būti kompanijos pastato suvartojamas elektros kiekis. Suvartojamas elektros kiekis būtų funkcija, kuri priklauso nuo laiko. Įvykiai kurie veiktų šią funkciją būtų, bet kurio elektrinio įrenginio įjungimas arba išjungimas. Bet koks prietaiso elektros sunaudojimo kiekio pakitimas būtų taip pat diskretus įvykis (Whatls, 2016).

1 lentelės tęsinys

Dinaminė simuliacija	Dinaminė simuliacija reaguoja (dažniausiai keičiasi) sistemoje priklausomai nuo įvedimo signalų. Signalus gali keisti žmogus arba kita automatizuota sistema (taip pat gali būti ir kitas procesas). Proceso dinamiškumas gali būti tiek būsenos keitimas jos vykdymo metu, tiek jos struktūros keitimas. Proceso keitimas vykdymo metu gali pasireikšti tuo, kad joje gali būti įterpiamos naujos veiklos, veiklos gali būti panaikinamos, pasikeisti veiklos prasmė arba išsišakojimų sąlygos.
Paskirstyta (angl. *Distributed*) simuliacija	Paskirstyta simuliacija yra apibrėžiama, kaip skirtingas modelių rinkinys ir/arba simuliacijos, veikiančios įprastoje dirbtinėje aplinkoje, bendraujančios per tinklą. Papildomai prie fizinių jungčių, egzistuoja standartinės architektūros ir susiję protokolai yra naudojami tam, kad perduoti informaciją tarp komponentų modelių ir simuliacijos (RPG Core Document, 2011).

1.4.1. Verslo procesų simuliacijos pliusai ir minusai

Dažnai simuliacija nėra vertinama dėl dažnai gaunamų klaidingų rezultatų. Tai dažnai atsitinka dėl nepakankamai detaliai sudaryto simuliacijos modelio. Detaliai ir tiksliai sudaryti simuliacijos modelį yra sudėtingas uždavinys ir dažnas simuliacijos įrankis nepalaiko realaus pasaulio vaizdavimo matematinių modelių. Todėl šiame skyriuje pateikiame simuliacijos teikiamus pliusus ir minusus.

Pliusai:

Simuliacija, palyginus su realiu mėginimu įgyvendinti tam tikrą problemą, turi du pagrindinius pliusus. Pirmas pliusas yra sutaupomos lėšos. Informacinių sistemų kūrimo procesas yra ilgas ir labai brangus procesas. Dažniausiai atlikti simuliaciją yra pigiau ir greičiau negu realiai sukurti ir ištestuoti informacinę sistemą. Antras simuliacijos pliusas yra didelis detalių skaičius, kurį galima gauti iš simuliacijos. Simuliacija gali gauti rezultatus, kurių negali gauti eksperimentiniai matavimai dabartinėje situacijoje. Pavyzdžiui paviršių sąveika atomo lygyje šiuo metu mes negalime pamatuoti, bet galime simuliuoti ir gauti rezultatus. Taip pat simuliaciją galima vykdyti kiek norima kartų ir su norimu detalumu, vieninteliai ribojimai yra vaizduotė, programavimo įgūdžiai ir kompiuterio skaičiavimo galimybės (CPU). V. D. Aalstas ir M. Voorhoeve (2016) įvardina šiuos simuliacijos pliusus:

- Simuliacija yra lanksti. Bet kuri situacija, nesvarbu kiek sudėtinga, gali būti ištirta pasinaudojus simuliacija;
- Simuliacija gali būti naudojama atsakant į įvairiausius klausimus. Įmanoma įvertinti laukimo laiką, rezervavimo greitį ir gedimų procentą iš vieno ir to pačio modelio;
- Simuliaciją lengva suprasti. Išsėmės simuliacija yra modelio kartojimas. Palyginus su analizei skirtų modelių gausa, reikia nedaug žinių tam, kad suprasti simuliacijos modelį.

Minusai

Vienas pagrindinių simuliacijos minusų yra klaidos. Bet koks neteisingas parametras potencialiai gali lemti neteisingus rezultatus. Taip pat dažnai programuojant naudojamos teorijos, kaip veikia tam tikras dalykas, o ne dėsniai, o teorijos nėra 100% teisingos. Taip pat dažnas neteisingų rezultatų šaltinis yra per daug supaprastintas simuliacijos modelis. Todėl dažnai simuliacijos nėra vertinamos, jeigu nėra duomenų su kuriais galima palyginti simuliacijos rezultatus ir parodyti, jog simuliacija veikia. Kitas didelis simuliacijos trūkumas yra pats faktas, jog tai simuliacija. Dažnai žmonės nesupranta to ko negali paliesti, pamatyti, išgirsti ar paragauti. V. D. Aalstas ir M. Voorhoeve (2016) įvardina šiuos simuliacijos trūkumus:

- Simuliacijos projektas gali pareikalauti daug laiko. Kartais reikia ilgai simuliuoti tam, kad gauti patikimus rezultatus;
- Simuliacijos rezultatus reikia interpretuoti atsargiai. Nustatant rezultatų patikimumą galima padaryti esmines klaidas, kurios lems viso projekto sėkmę arba nesėkmę;
- Simuliacija nepasiūlo įrodymų. Kas gali atsitikti teisingame simuliacijos modelyje gali atsitikti ir realybėje, tačiau atvirkščiai būti negali. Tam tikri atsitikimai gali nutikti realybėje, tačiau jie gali būti neįvertinti simuliacijos modelyje.

1.4.2. Verslo procesų simuliacijos ištekliai

Paprastai ištekliai yra medžiagos, paslaugos, personalas, žinios ir kitas turtas, kuris naudojamas tam, kad verslui atneštų naudą (Turner, M., 2015). Ištekliai gali būti sunaudojami (išteklius kurio kiekis nuolatos mažėja) arba panaudojami (atlikus darbą tokie ištekliai grįžta į laisvų išteklių sąrašą).

Nors verslo procesų simuliacija yra gana paplitusi, tačiau dauguma atveju simuliacijos rezultatai yra netikslūs lyginant su realiais rezultatais. Netikslumai gali kilti ir dėl per daug supaprastinto išteklių modelio ir dėl to, kad modeliuose yra neįvertinta (Aalst, V. D., 2010):

1. Ištekliai įtraukti į simuliaciją gali būti paskirstyti per daugelį procesų, nors simuliacijoje dažniausiai modeliuojamas yra vienas procesas;

2. Ištekliai nėra pasiekiami visą laiką;
3. Ištekliai (pavyzdžiui žmonės) nėra naudojami pastoviu greičiu.

Dėl šių išvardintų problemų yra reikalingas išplėstas išteklių modelis, kuris palaikytų išteklių aprašymą su sudėtingomis išteklių charakteristikomis, kurios gali lemti simuliacijos rezultatus. F. G. Fadelis ir bendraautoriai išskiria išteklių charakteristikas, kurios yra pateiktos 2 lentelėje.

2 lentelė. Išteklių charakteristikos

Pavadinimas	**Aprašymas**
Ištekliaus padalinimas	Charakteristika parodanti, jog išteklius gali būti padalinamas. Yra du padalinimo tipai fizinis ir funkcinis. Fizinis padalinimas yra susietas su objekto fizinio kūno padalinimu. Funkcinis padalinimas nurodo padalinimą, kuris daro įtaką funkcijai, o ne struktūrai.
Ištekliaus dalumas	Šis terminas apibūdina ištekliaus dalumo nustatymą atsižvelgiant į vykdomą veiklą, nepakeičiant ištekliaus vaidmens veikloje. Yra trys dalumo tipai: fizinis, funkcinis ir laikinas dalumas. Išteklius yra fiziškai dalijamas, jeigu kiekvienas padalinimas yra panaudojamas arba sunaudojamas tos pačios veiklos. Tai yra naudinga analitikui, kai sprendžiama ar ištekliaus dalis gali palaikyti veiklą. Funkcinis ištekliaus dalumas nusako, jog kiekvienas ištekliaus padalinimas daro įtaką funkcionalumui, o ne ištekliaus struktūrai. Tokio tipo dalomumo pavyzdys būtų stovėjimo aikštelė, kur stovėjimo vietų skaičius yra funkcinis padalinimas automobilių pastatymo veikloje. Laikinas ištekliaus dalumas parodo, jog laikui bėgant ištekliaus naudojimas nedaro įtakos to pačio ištekliaus naudojimui ateityje, toje pačioje veikloje.
Matavimo vienetas	Ši charakteristika aprašo standartinį ištekliaus matavimo vienetą, kai yra priskiriamas prie veiklos. Išteklių kiekis arba talpa yra matuojami naudojant nurodytą matavimo vienetą. Ši charakteristika naudojama nurodant tiek kokybinius, tiek kiekybinius matavimo vienetus. Kokybiniai matavimo vienetai galėtu būti: didelis, vidutinis, mažas, ir t. t. Kiekybiniai matavimo vienetai būtų: ilgis, svoris, tūris ir t. t.

2 lentelės tęsinys

Komponentas	Aprašo išteklių, kuris yra kito ištekliaus dalis, tai reiškia, kad išteklius susideda iš vieno ar daugiau smulkesnių išteklių. Išteklius gali būti fizinis arba funkcinis kito ištekliaus komponentas.
Nuolatiniai ir diskretiniai ištekliai	Nuolatiniai ištekliai nurodo išteklius, kurie yra nesuskaičiuojami. Diskretiniai ištekliai nurodo suskaičiuojamus išteklius.
Naudojimo būdas	Ši charakteristika grąžina ar išteklius veikloje yra palaikomas, kaip nuolatinis ar diskretinis. Ši charakteristika neparodo ar veikla yra diskretinė ar nuolatinė. Naudojimo būdas priklauso nuo veiklos, kuri sunaudoja arba panaudoja išteklių.
Lygiagretaus naudojimo ribojimai	Lygiagretaus naudojimo ribojimai draudžia išteklių panaudojimą/sunaudojimą dviem vienu metu veikiančioms veikloms. Pavyzdžiui tai gali būti naudojama tuo atveju, kai dvi veiklos negali naudoti to pačio ištekliaus „orkaitė“, nes joms reikia skirtingos temperatūros.
Alternatyvus išteklius	Ši charakteristika nurodo ištekliaus alternatyvas, kurie gali būti panaudojami ar sunaudojami veikloje, vietoje kito ištekliaus. Tai yra naudojama tuomet, kai pagrindinis išteklius sugenda ar yra nepasiekiamas, tada galima panaudoti to ištekliaus alternatyvą.

Į 2 lentelėje išvardytas charakteristikas bus atsižvelgiama, kuriant dinaminių verslo procesų išteklių modelį. Taip pat, įvertinus literatūra buvo nuspręsta toliau darbe nagrinėti ir spręsti dalomų išteklių problemą. Ši problema daro įtaką simuliacijos rezultatams ir išsprendus ją leistų gauti tikslesnius rezultatus.

1.5. Apibendrinimas ir rezultatai

Tam, kad būtu galima įvertinti dinaminių verslo procesų išteklius, visų pirma, reikia gebėti išteklius modeliuoti ir simuliuoti. Todėl šiame skyriuje buvo pateikiamos pagrindinės, su tema susijusios, sąvokos, išskirti verslo procesų ir simuliacijos tipai, aprašytos išteklių charakteristikos ir išskirtos verslo procesų modeliavimo galimybės.

Iš pateiktos literatūros analizės galime teigti, jog toliau darbe nagrinėjant dinaminių verslo procesų išteklius bus sprendžiama išteklių dalumo problema. Ši problema pasirinkta todėl, kad

išteklių dalumas gali lemti simuliacijos rezultatus ir nesugebant to įvertinti gaunami klaidingi rezultatai. Šio darbo kontekste yra nagrinėjama tik dinaminė simuliacija.

2. SIMULIACIJOS ĮRANKIŲ IR IŠTEKLIŲ MODELIUI KELIAMŲ REIKALAVIMŲ ANALIZĖ

Šiame skyriuje yra analizuojami rinkoje egzistuojantys modeliavimo ir simuliacijos įrankiai ir galimybes simuliuoti DVP ir išteklius. Atlikta analizė leis susipažinti su verslo simuliacija, kaip ji atliekama ir leis įvertinti rinkoje esančių produktų privalumus ir trūkumus.

2.1. „ARIS Simulation“

„ARIS Simulation“ yra įrankis skirtas dinaminiai verslo procesų analizei. Tai yra vienas paketas iš daugelio, kurie sudaro visą „ARIS“ įrankį, leidžiantį modeliuoti ir analizuoti įvairius verslo procesus. Verslo procesai sukurti naudojantis ARIS įrankiu yra transformuojami į simuliacijos modelį. Šiame įrankyje verslo procesai yra vaizduojami įvykiais valdomų procesų grandinėmis. Simuliacijos įrankis parodo ar sumodeliuotas verslo procesas gali būti įvykdytas ir atsako į klausimus apie pralaidumo laikus ir išteklių panaudojimo lygius bei kitus klausimus. Simuliacijos rezultatus galima eksportuoti į „Microsoft Excel“ ir pridėti statistinės analizės metodus įvykiams, funkcijoms, ištekliams ir kitiems parametrams (Jansen-Vullers, M. ir Netjes, M., 2006).

Prieš pradedant simuliaciją reikia nurodyti pradinius duomenis, kaip: vidutinis veiklos vykdymo laikas ir norimas skirstinys, sugeneruojamų atvejų skaičius, veiklų pasirodymo skirstinys ir išsišakojimų tikimybės (Aris, 2016).

Įrankio pliusai:

- Galima dinaminių verslo procesų ir išteklių simuliacija;
- Platus verslo procesų modeliavimo metodų pasirinkimas;
- Duomenų analizės ir ataskaitų generavimo galimybės.

Minusai:

- Simuliacijos paketas yra mokamas;
- Įrankio kodas nėra laisvai prieinamas.

2.2. „IBM Websphere Business Modeler“

„IBM Websphere Business Modeler“ yra pažangus verslo procesų modeliavimo ir analizavimo įrankis. Naudotojas, su šiuo įrankiu, gali sukurti verslo proceso modelius, juos simuliuoti ir analizuoti tam, kad geriau suprastų vykdomus verslo procesus ir juos dokumentuotų.

Šis įrankis pagreitiną verso procesų įgyvendinimą (IBM, 2016.). Tam, kad modelį būtų galima simuliuoti reikia nurodyti:

- Išteklius atsakingus už veiklų vykdymą;
- Laiką kuris reikalingas atliktį veiklą;
- Kiekvienam išsišakojimui suteikti tikimybę.

Šiame įrankyje galima pasirinkti procesų vykdymo šaką vykdymo metu, tačiau galimybės pilnai simuliuoti dinaminius verslo procesus nėra. Taip pat šis įrankis pasižymi šiomis savybėmis (IBM, 2016):

- Įtraukta pažangi simuliavimo galimybė, kuri yra paremta modeliuotais ir realiais duomenimis, tai leidžia naudotojams priimti sprendimus paremtus tikslia informacija;
- Leidžia atvaizduoti ir pilnai suprasti visus verslo proceso modelio aspektus prieš jį pradedant kurti;
- Visi „IBM BPM“ įrankiai yra integruoti tarpusavyje;
- Ištekliams galima nurodyti papildomas charakteristikas, kurios svarbios simuliacijos vykdymo metu;
- Palaikomas simuliacijos rezultatų ataskaitų generavimas ir analizavimas;

  Įrankio pliusai:
  - Galimas dalinis dinaminių išteklių simuliavimas;
  - Pažangus verslo procesų modelių kūrimas;
  - Skirtingų simuliacijos scenarijų palaikymas
  - Detalus išteklių modeliavimas;

  Įrankio minusas yra, jog įrankio kodas nėra laisvai prieinamas;

2.3. „Simprocess“

„Simprocess“ leidžia sukurti abstraktų proceso modelį, šiame įrankyje reikia sukurti kompiuterinį verslo proceso modelį ir dokumentaciją (diagramas ir aprašymus). Esybes yra vykdomos iš vieno proceso žingsnio į kitą ir kiekviename žingsnyje yra atliekama tam tikra veikla. Tokie ištekliai kaip: žmonės, įrenginiai ar informacija reikalinga užbaigti užduotį, taip pat yra įtraukiami į verslo proceso modelį (Simprocess, 2016).

„Simprocess“ surenka vertingą informaciją apie modelio veiklas, esybes ir išteklius. Ši informacija yra naudojama verslo proceso tikrinimui. Sukaupti statistiniai duomenys ir ataskaitos

yra naudojamos tam, kad nustatytų ar verslo procesą galima patobulinti. Šis įrankis leidžia įvertinti vykdymo alternatyvas ir pamatuoti galimus rezultatus.

Ši priemonė aprašant išteklius leidžia apibrėžti papildomus parametrus, pagal kuriuos galima pasirinkti išteklius. Taip pat šioje priemonėje aprašant išteklius galima nurodyti ne tik kainą ir trukmę, kada išteklius yra užimtas, bet taip pat leidžiama nurodyti papildomus parametrus, kurie aprašo ištekliaus charakteristikas. Nurodant ištekliaus užimtumą, nurodomas pradžios ir pabaigos datos arba galima sudaryti išteklių užimtumo grafiką, kas leidžia tiksliau sumodeliuoti išteklius.

Įrankio pliusai:

- Detalus išteklių modeliavimas;
- Duomenų statistinė analizė ir ataskaitų rengimo galimybė;

Įrankio minusai:

- Nėra galimybės simuliuoti dinaminių verslo procesų;
- Įrankio kodas nėra laisvai prieinamas.

2.4. „Bizagi“

„Bizagi Modeler“ yra verslo procesų modeliavimo ir dokumentavimo įrankis. Šis įrankis yra paremtas BPMN standartu. „Bizagi Simulation“ yra paremtas BPSim (angl. *Business Process Simulation*) standartu. Įrankyje yra palaikoma „kas jeigu...“ tipo simuliacija. Šio tipo simuliacijoje yra pasirenkami skirtingi scenarijai ir gaunami rezultatai. Tai leidžia įvertinti, kaip parametrų pakeitimai gali lemti verslo procesų simuliacijos rezultatus ir rasti optimaliausią variantą. Įrankis palygina skirtingų scenarijų gautus rezultatus ir suformuoja ataskaitą (Bizagi, 2016).

Simuliacija vykdoma 4 skirtingai lygiais. Pirmame lygyje yra patikrinamas verslo procesas, ar turima visa reikalinga informacija ir ar modelis yra tinkamai sukurtas. Antrame lygyje yra atliekama laikinė analizė. Trečiame lygyje atliekama išteklių analizė ir ketvirtame lygyje yra atliekama kalendorinė analizė, kuri įtraukia kalendorinę informaciją.

Įrankio pliusai:

- Palaiko BPMN standartą;
- Galimybė eksportuoti modelį į XML formatą;
- 4 lygių simuliacija;
- Skirtingų simuliacijos scenarijų palaikymas.

Įrankio minusai:

- Nevisi BPMN elementai yra palaikomi simuliacijoje;
- Nevisos BPMN diagramos yra palaikomos simuliacijoje;
- Nepalaiko dinaminės simuliacijos;

2.5. „AnyLogic“

„AnyLogic“ simuliacijos modeliavimo programinė įranga yra sprendimų priėmimo įrankis, kuris palaiko visus egzistuojančius modeliavimo metodus (sistemos dinamika, diskrečius įvykius ir agentais paremtą modeliavimą (angl. *agent based modelling*)). Simuliacijos modelis kuriamas ir simuliuojamas tame pačiame įrankyje, todėl nereikia išmokti kelių skirtingų naujų priemonių. Įrankis pasižymi šiomis savybėmis (AnyLogic, 2016):

- Įrankyje esanti kūrimo aplinka pagreitina kūrimo procesą;
- Įrankyje esančio objektų bibliotekos leidžia greitai įtraukti iš anksto paruoštus simuliacijos elementus. Taip pat šiuos elementus galima prisitaikyti pagal savo poreikius;
- Jau sukurtus modelius galima panaudoti ir kituose projektuose;

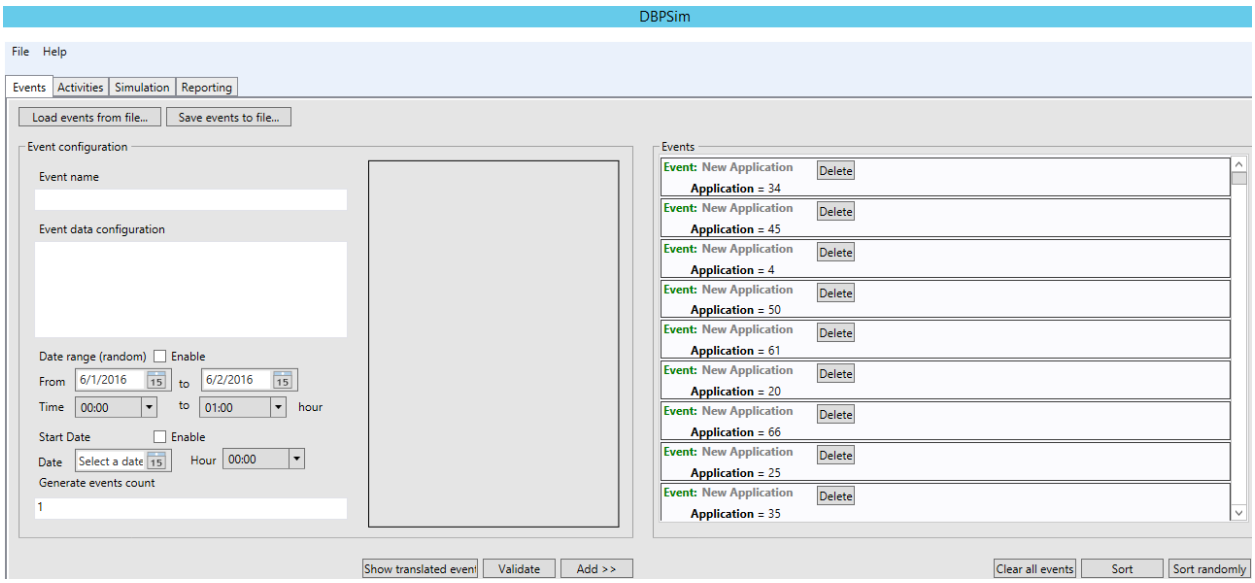
  Įrankio pliusai:

  - Nemokamas įrankis;
  - Palaikomas su Windows, Mac ir Linux operacinėmis sistemomis;
  - Dalinai galima simuliuoti DVP;
  - Detalus išteklių modeliavimas.

  Įrankio minusas yra, jog įrankio kodas nėra laisvai prieinamas;

2.6. „DBP Sim“

„DBP Sim“ yra dinaminių, taisyklėmis grindžiamų, verslo procesų simuliavimo įrankis, kuris leidžia apsirašius ECA taisykles ir nekuriant modelio simuliuoti dinaminius verslo procesus. Šiame įrankyje taisyklės yra atskirtos nuo programinio kodo, todėl tai leidžia bet kuriuo simuliacijos metu pakeisti taisykles ir stebėti, kaip proceso vykdymą keičia toks taisyklių pakeitimas. Tačiau, dabartinėje įrankio versijoje nėra galimybės įvertinti išteklių poveikį DVP. 6 paveiksle yra parodyta „DBP Sim“ įrankio vartotojo sąsaja.



6 pav. „DBP Sim“ įrankio vartotojo sąsaja

Įrankio pliusai:

- Įrankis atviro kodo, nemokamas;
- Skirtas simuliuoti DVP;
- Nėra apibrėžto pirminio modelio, viskas kuriama ECA taisyklėmis;

Įrankio minusai:

- Nėra galimybės modeliuoti išteklių;
- Nemokant programuoti įrankiu naudotis yra sudėtinga;

2.7. Įrankių analizės apibendrinimas

Atliekant šiuo metu rinkoje esačių verslo procesų modeliavimo ir simuliacijos įrankių analizę buvo nuspręsta kriterijus suskirstyti į 4 pagrindines grupes ir kiekvienai grupei identifikuoti kriterijus į kuriuos atsižvelgiama pasirenkant įrankį:

- Verslo procesų modelis;
  - Apibrėžiamas pradinis verslo procesų modelis;
  - Galimybė apibrėžti VT;
  - Modelio arba taisyklių keitimas vykdymo metu.
- Išteklių modelis;
  - Galimas išteklių simuliavimas;
  - Galimybė nurodyti skirtingas išteklių charakteristikas;
  - Išteklių klasifikavimas pagal parametrus;

- Išteklių konkuravimas;
- Galimybė padalinti išteklius per vienu metu vykstančias veiklas;

- Istorinė informacija
  - Istorinių duomenų išsaugojimas;
  - Simuliacijos rezultatų analizė;
  - Ataskaitų generavimas;
- Simuliacija;
  - Verslo procesų simuliacija;
  - Dinaminės simuliacijos palaikymas.

Pateikti programinės įrangos kriterijai yra įvertinami pagal 3 vertinimo lygius:

- (+) – kriterijus yra pilnai palaikomas programinėje įrangoje;
- (+/-) – kriterijus yra dalinai palaikomas programinėje įrangoje;
- (-) – kriterijus nepalaikomas programinėje įrangoje

Gauti rezultatai yra pateikiami 3 lentelėje.

3 lentelė. Modeliavimo ir simuliacijos įrankių palyginimas

Įrankis / **Palyginimo kriterijus**	**ARIS Simulation**	**IBM Websphere Business Modeler**	**Simprocess**	**Bizagi**	**Any Logic**	**DBP Sim**
1. Verslo procesų modelis						
1.1. Apibrėžiamas pradinis verslo proceso modelis	+	+	+	+	+	-
1.2. Galimybė apibrėžti VT	+	+	+/-	+	+/-	+
1.3. Modelio arba taisyklių keitimas vykdymo metu	-	+	-	-	-	+
2. Išteklių modelis						
2.1. Galimas išteklių simuliavimas	+	+	+	+	+	-
2.2. Galimybė nurodyti skirtingas išteklių charakteristikas	-	+	+	+	+	-
2.3. Išteklių klasifikavimas pagal parametrus	-	+	+	-	+	-
2.4. Išteklių konkuravimas	-	-	-	-	-	-

3 lentelės tęsinys

2.5. Galimybė padalinti išteklius per vienu metu vykstančias veiklas	-	-	-	-	-	-
3. **Istorinė informacija**						
3.1. Istorinių duomenų išsaugojimas	+	+	+	+	+	-
3.2. Istorinių duomenų analizė	+	+	+	+	+	+/-
3.3. Ataskaitų generavimas	+	+	+	+	+	-
4. Simuliacija						
4.1. Verslo procesų simuliacija	+	+	+	+	+	+
4.2. Dinaminės simuliacijos palaikymas	+	+/-	-	-	+/-	+

2.8. Išteklių modelio reikalavimai

Daugelis įmonių savo veiklai optimizuoti naudoja verslo procesų modelius, kurie leidžia analizuoti juos tam tikrose būsenose (Chinosi *et al*., 2012, Bosilj-Vuksic, V., *et al.* 2007). Verslo procesų modeliavime išteklių modeliavimas yra svarbus uždavinys. Kiekvienas iš išteklių yra susijęs su vykdoma veikla, kuri savo ruožtu yra nurodoma verslo procese (Dijkman, M. *et al*. 2008). Būtina užtikrinti, kad veiklos panaudotų būtent jai vykdyti reikalingus išteklius. Pernelyg didelis verslo procesų automatizavimas ir menkas darbų planavimas yra svarbiausias projektų, kurie remiasi darbų srautais, nesėkmės kriterijus (Stroppi, L. J. R., *et al*., 2011). Todėl labai svarbu verslo procesuose tinkamai modeliuoti turimus išteklius.

Šiame skyriuje yra apžvelgiama literatūra ir suformuluojami reikalavimai, kuriuos turi atitikti išteklių modelis. Kiekvienas verslo procesas susideda iš veiklų. Tam, kad įgyvendintume veiklas reikalingi ištekliai ir vien tik darbo srauto logikos nepakanka (Dijkman, M. *et al*., 2008). Dažnai veikloms vykdyti reikalingi konkretaus tipo ištekliai. Keli ištekliai gali turėti tokį patį tipą ir kelioms veikloms gali prireikti konkretaus tipo išteklių (Van Der Aalst, W. *et al*., 2015). Tam, kad įgyvendintume šį reikalavimą, turime užtikrinti, jog kuriamame modelyje būtų galimybė konkretiems ištekliams nurodyti neribotą skaičių charakteristikų. Pagal charakteristikas bus atrenkami ištekliai ir jie priskiriami prie vykdomos veiklos. Pavyzdžiui, klientui reikalinga

„mediena“, konkrečiai: „Ąžuolas“, „100 m^3“. Pagal šias charakteristikas bus atlikta užklausa ir parinktas būtent visas charakteristikas atitinkantis išteklius.

Kiekvienam ištekliui turi būti aprašyta panaudojimo kaina. Kaina gali būti už laiko vienetą arba už panaudojimo veikloje kartą (Vasilecas, O. *et al*., 2014). Tai leis įvertinti visos veiklos kainą, o tuo pačiu ir vykdomo proceso kainą. Ištekliai dažniausiai yra sunaudojami, todėl kiekvienam ištekliui turi būti nurodytas turimas kiekis. Sunaudojus išteklių šis kiekis turi būti sumažinamas.

Taip pat reikia nustatyti veiklos vykdymo trukmę. Daugelyje verslo procesų vidutinė veiklos vykdymo trukmė yra ilgesnė negu numatyta trukmė tai veiklai. Taip įvyksta, nes reikalingi ištekliai yra užimti ir tenka laukti kol jie atsilaisvins (Van Der Aalst, W. *et al*., 2015). Tam, kad svarbi veikla nebūtų atidėliojama dėl nepakankamo išteklių skaičiaus reikia, kad veiklai būtų galimybė nurodyti vykdymo prioritetą, o išteklius nesunkiai perskirstyti tarp veiklų. Taip pat svarbu nustatyti, kaip išteklių padidinimas arba sumažinimas lemia veiklos vykdymo trukmę. Išteklių priskyrimo datos ir išteklių atlaisvinimo datos skirtumas leis išsiaiškinti užduoties vykdymo trukmę.

Bet kuriuo metu pareikalavus turi būti užtikrinta galimybė gauti informaciją apie išteklių dabartinę būseną. Tai leis nustatyti laisvus išteklius ir juos priskirti prie veiklos arba leis pamatyti pasibaigusius išteklius bei numatyti jų papildymą. Panaudojamiems ištekliams turi būti užtikrinta, jog pasibaigus naudojimo laikui ištekliai grįžtų į laisvų išteklių sąrašą.

Akivaizdu, jog verslo procesai dažniausiai turi ribotą išteklių skaičių, dėl šios priežasties tarp procesų gali kilti konkurencija. Dažniausiai konkurencija yra pastebima kai du ar daugiau verslo procesų startuoja vienu metu ir naudoja tuos pačius išteklius (Vasilecas, O. *et al*., 2015). Todėl turi būti užtikrinta galimybė analizuoti du ar daugiau lygiagrečiai vykdomus, skirtingus procesus, kurie naudotų tuos pačius išteklius arba ištekliai būtų padalinti tarp skirtingų veiklų, jeigu ištekliai gali būti padalinami. Tai leistų įvertinti nuostolius atsirandančius dėl skirtingų verslo procesų konkurencijos ir pateikti sprendimą, kaip šiuos nuostolius sumažinti iki minimumo.

Išanalizavus literatūrą, toliau yra pateikiami apibendrinti dinaminiuose verslo procesuose naudojamų išteklių reikalavimai:

1. Ištekliai turi būti priskiriami prie veiklos. Taip pat keli ištekliai gali būti priskiriami vienai veiklai;
2. Ištekliams gali būti nurodytas neribotas skaičius charakteristikų, kurios apibūdina įvairias jų savybes;

3. Ištekliui būtina nurodyti jo panaudojimo kainą ir turimą kiekį;
4. Panaudojus išteklius turi būti užtikrinamas jų grąžinimas į saugyklą;
5. Ištekliai turi gebėti pateikti atsakymus apie jų būseną bet kuriuo laiko momentu;
6. Išteklių modelis turi palaikyti dalomumą.

Remiantis šiais reikalavimais buvo sukurtas išteklių valdymo duomenų bazės modelis, kuris detaliau aptariamas 3.2.3 skyriuje.

2.9. Apibendrinimas

Atlikus verslo procesų simuliacijos įrankių analizę buvo įvertinti įrankiai pagal nurodytus kriterijus bei atsižvelgta į jų pliusus ir minusus. Kaip matome iš 3 lentelės, dinaminę verslo procesų simuliaciją palaiko „ARIS Simulation“ ir „DBP Sim“, tačiau „ARIS Simulation“ yra mokamas įrankis ir prie šio įrankio programinio kodo prieiti neįmanoma. Taip pat nors ir dauguma įrankių sugeba simuliuoti verslo procesų išteklius, tačiau jų konkuravimo ir dalomumo problemos nėra sprendžiamos arba palieka tuo pasirūpinti analitikui, kuris kuria modelį. Toliau darbe buvo pasirinkta dirbti su „DBP Sim“ įrankiu, nors jis turi trūkumų šis įrankis buvo pasirinktas nes:

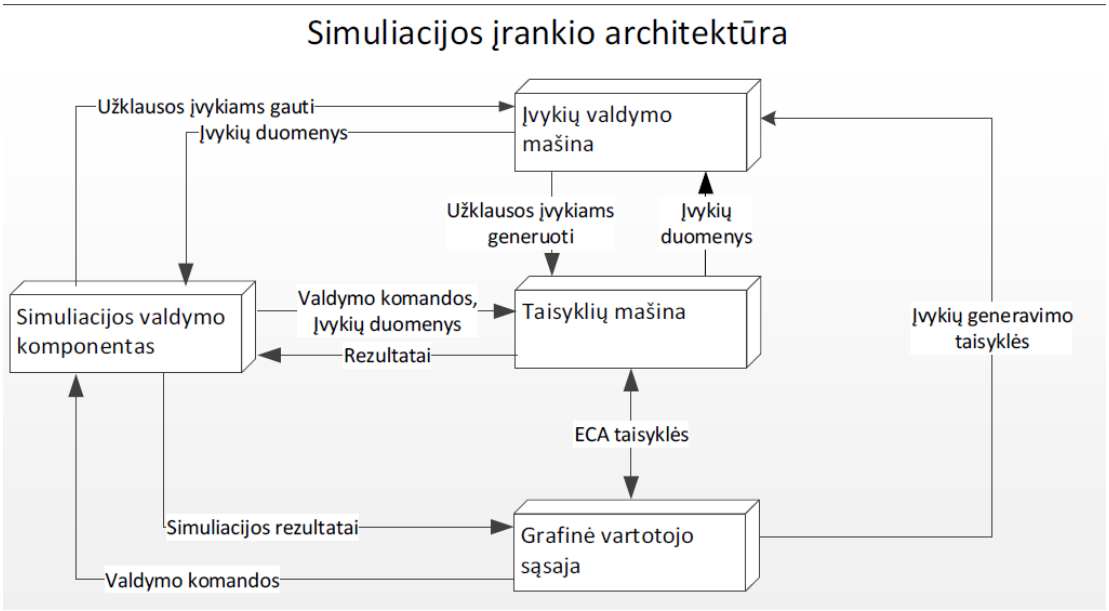
- Įrankis nemokamas;
- Atviro kodo;
- Skirtas dinaminių verslo procesų simuliacijai.

Kadangi šiame įrankyje nėra įgyvendinta išteklių modeliavimo galimybės buvo išskirti 6 reikalavimai, kuriuos turi tenkinti kuriamas išteklių modelis.

3. DINAMINIŲ, TAISYKLĖMIS GRINDŽIAMŲ, VERSLO PROCESŲ IŠTEKLIŲ MODELIAVIMAS

3.1. Dabartinė situacija

M. Remeika (2014) pateikia „DBP Sim“ įrankio architektūra (7 pav.). Šiame paveiksle yra pavaizduoti 4 pagrindiniai įrankio komponentai, kurie leidžia simuliuoti DVP.



7 pav. „DBP Sim“ įrankio architektūra (Remeika, M., 2014).

- Įvykių valdymo mašina. Suteikia galimybę generuoti ir valdyti įvykius. Šie įvykiai leidžia pradėti simuliacijos procesą. Įvykiai yra generuojami naudotojo;
- Simuliacijos valdymo komponentas. Atsakingas už simuliacijos kontroliavimą, pradėjimą, stabdymą ir simuliacijos greičio pasirinkimą;
- Taisyklių mašina. Atsakinga už simuliacijoje vykdomas taisykles, jų parinkimą ir įvykdymą. Taisykles generuoja naudotojas;
- Grafinė vartotojo sąsaja. Leidžia naudotojui valdyti visą simuliacijos procesą.

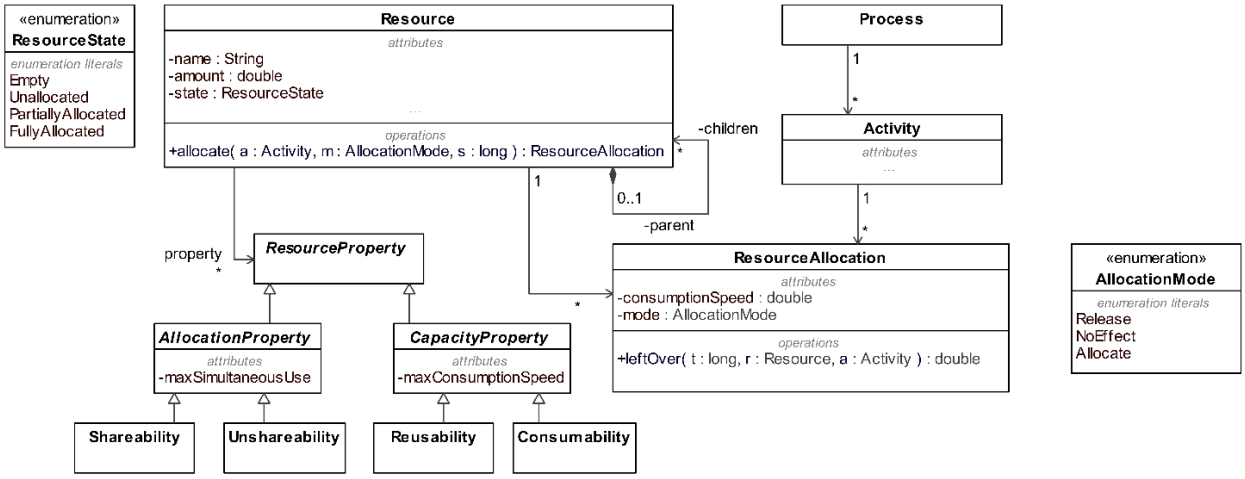
Šiame skyriuje yra siūloma šią architektūra išplėsti papildomais komponentais t. y. išteklių valdymo komponentu ir duomenų bazės serveriu. Išteklių valdymo komponentas bus atsakingas už visus veiksmus atliekamus su ištekliais, o duomenų bazės serveris bus atsakingas už išteklių

saugojimą ir duomenų pateikimą simuliacijos įrankiui. Plačiau apie šiuos komponentus bus aprašoma 3.2 skyriuje.

3.2. Siūlomas metodas

3.2.1. Dalomų išteklių modeliavimas

8 paveiksle yra pateiktas bendras išteklių modelis, kuris aprašo išteklių dalomumą ir tokių išteklių vartojimą. Kaip matome iš modelio išteklius gali būti 4 būsenų (modelyje angl. *ResourceState*), tai: tuščias (angl. *Empty*), nepriskirtas (angl. *Unallocated*), dalinai užimtas (angl. *PartiallyAllocated*) ir visiškai užimtas (angl. *FullyAllocated*). Būsenų perėjimo diagrama ir jos ribojimai pateikiami 3.2.2 skyriuje.



8 pav. Dalomų išteklių klasių diagrama

Ištekliai yra priskiriami prie veiklos (modelyje angl. *Activity*) pasinaudojus tarpine klase ištekliaus priskyrimas (modelyje angl. *ResourceAllocation*), kuri leidžia tai pačiai veiklai priskirti kelis skirtingus išteklius. Ši klasė atsakinga už išteklių atlaisvinimą (modelyje angl. *Release*), priskyrimą (modelyje angl. *Allocation*) ir tuo atveju, kai ištekliaus būsena nesikeičia priskyrimo režimas (modelyje angl. *AllocationMode*) tampa jokio poveikio (angl. *NoEffect*).

Šiame modelyje ištekliai yra aprašomi naudojantis išteklių nustatymais (modelyje angl. *ResourceProperty*), kurie nusako išteklių galimybes. Priskyrimo nustatymas (modelyje angl. *AllocationProperty*) aprašo galimybę padalinti išteklių per kelias veiklas, bet kuriuo laiko momentu. Jeigu išteklius yra dalomas, tuomet jis turi dalomumo (modelyje angl. *Shareability*) nustatymą ir priešingai, nedalomi ištekliai yra charakterizuojami ne dalomumo (modelyje angl. *Unshareability*) nustatymu.

Taip pat ištekliai gali būti charakterizuojami pagal jų panaudojimo (modelyje angl. *CapacityProperty*) nustatymą. Ištekliai gali turėti panaudojimo (modelyje angl. *Reusability*) nustatymą, kuris apibūdina išteklius kurie yra pakartotinai panaudojami per skirtingas veiklai. Sunaudojimo (modelyje angl. *Consumability*) nustatymas apibūdina išteklius kurie yra sunaudojami per veiklos vykdymą. Tuo metu kai veikloje yra atlaisvinami ištekliai, kurie turi panaudojimo nustatymą, ištekliai yra gražinami atgal į išteklių konteinerį, kaip nepriskirti ištekliai. Jeigu ištekliai turi sunaudojimo nustatymą, tada išteklių atlaisvinimo metu atitinkamas kiekis išteklių yra pašalinamas iš išteklių konteinerio.

Ištekliai tuo pačiu metu gali būti apibūdinami tiek priskyrimo, tiek panaudojimo nustatymais.

3.2.2. Išteklių būsenų diagrama ir perėjimų ribojimai

Visų pirma, apibrėžiant išteklių priskyrimo taisykles yra būtina formaliai įvardinti išteklius, jų būsenas ir būsenų perėjimo funkciją. Visų pirma apibrėžiama kaip ištekliai specifikuojami simuliacijoje.

1 apibrėžimas. *Išteklius yra apibrėžiamas kaip kortežas* $\mathrm{r} = (\mathrm{p}, \mathrm{v}, \mathrm{l}, \mathrm{s}, \mathrm{ty}, \mathrm{b}, \mathrm{k})$, *kur* p *- unikalus ištekliaus pavadinimas,* v *- maksimalus veiklų skaičius galintis vienu metu naudoti išteklių,* l *– priskyrimų skaičius,* s*- maksimalus ištekliaus suvartojimo greitis,* ty *- ištekliaus tipas,* b *- dabartinė ištekliaus būsena ir* k *- galimas ištekliaus kiekis*.

Visų antra, ištekliai simuliacijos metu gali būti priskirti naudojant išteklių priskyrimą, kaip nurodyta 2 apibrėžime.

2 apibrėžimas. *Ištekliaus priskyrimas yra apibrėžiamas kaip kortežas* $m = (a, r, cs, n)$, *kur* a *- dabartinė veikla,* r *- dabartinis išteklius,* cs *- tai greitis, kuriuo šiuo metu yra sunaudojamas išteklius veikloje,* n *- ištekliaus kiekis, kuris buvo sunaudotas.*

Galiausiai, išteklių būsenos pasikeitimas yra aktyvuojamas panaudojus išteklių priskyrimo funkciją.

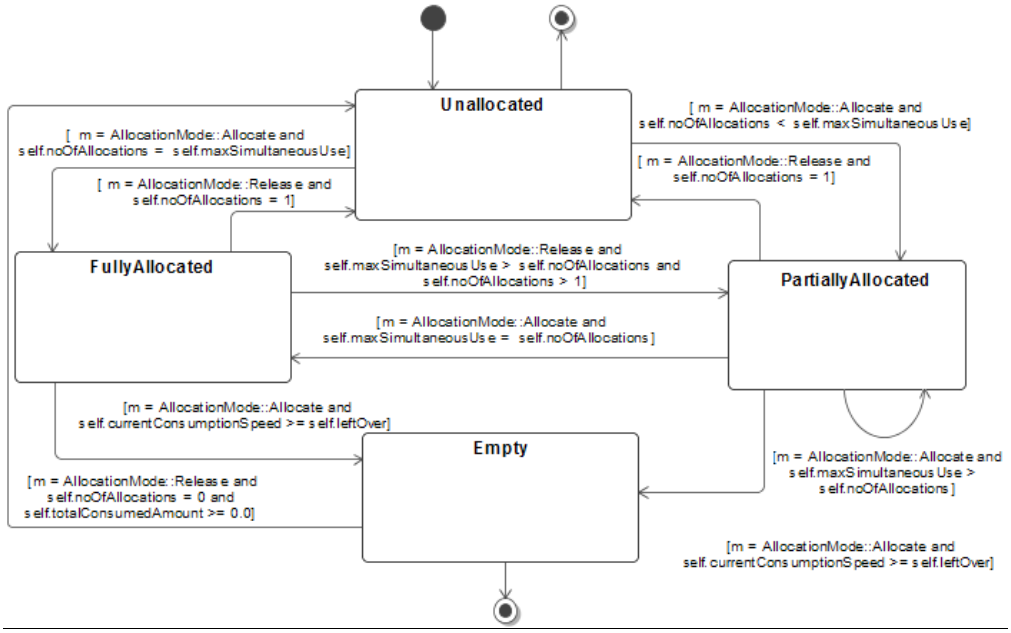
3 apibrėžimas. *Prisirymo funkcija yra apibūdinama kaip* $A(a, r, g, sp, L(t, r))$, *kur* a *– dabartinė veikla,* r *– dabartinis išteklius,* $g \in \{Atlaisvinti = 1, JokioPoveikio = 0, Priskirti = -1\}$ *– priskyrimo reikšmė, kuri apibūdina priskyrimo būdą,* sp *– greitis, kuris nusako kaip greitai veikla priskiria išteklių ir* $L(t, r)$*- yra funkcija, kuri prima parametrus laiką ir išteklių ir grąžina likusį ištekliaus kiekį.*

Apibrėžus pagrindines išteklių priskyrimo charakteristikas galima aprašyti išteklių perėjimo taisykles. 9 paveiksle pavaizduotos išteklių perėjimo taisyklės matematine išraiška.

$$\begin{cases} b' = In\ Use \wedge\ cs' = cs + sp \wedge\ l' = l + g\ , when\ b = Unallocated \wedge g = 1 \wedge v > l'\ ; \\ b' = PartiallyAllocated \wedge\ cs' = cs + sp \wedge l' = l + g, when\ b = PartiallyAllocated \wedge\ g = 1 \wedge v > l'\ ; \\ b' = PartiallyAllocated \wedge\ cs' = cs + sp, when\ b = FullyAllocated \wedge l' = l + g \wedge g = -1 \wedge v > l' \wedge v > 1; \\ b' = FullyAllocated \wedge\ cs' = cs + sp \wedge l' = l + g, when\ b\ \neq Empty \wedge g = 1 \wedge l' = v; \\ b' = Unallocated\ cs' = 0 \wedge sp = 0, l = 0, when\ b = Empty \wedge L(t+1, r) > 0; \\ b' = Unallocated \wedge cs' = 0 \wedge sp = 0, l' = l + g, when \wedge g = -1 \wedge l' = 0 \end{cases}$$

9 pav. Išteklių perėjimo taisyklių matematinės išraiškos

10 paveiksle yra pavaizduota išteklių būsenų diagrama su ribojimais. Ribojimai gauti matematines išraiškas pavaizduotas 8 paveiksle transformavus į objektų ribojimų kalbą (angl. *Object Constraint Language* arba OCL).



10 pav. Išteklių būsenų diagrama

Išteklius gali pasiekti kelias pabaigos būsenas: Nepriskirtas (paveiksle angl. *Unallocated*) ir Tuščias (paveiksle angl. *Empty*). Būsena Nepriskirtas reiškia, kad išteklius nėra priskirtas prie veiklos, tačiau išteklių konteineryje yra tam tikras šio ištekliaus kiekis. Būsena Tuščias reiškia, kad ištekliaus kiekis konteineryje lygus nuliui ir tuo pačiu tai reiškia, kad jokia veikla šio ištekliaus nenaudoja. Jeigu $v = 1$ tuoment būsena iš Neužimtas (paveiksle angl. *Unallocated*) gali būti pakeista tik į PilnaiUžimtas (paveiksle angl. *FullyAllocated*). Būsena PilnaiUžimtas apibūdina išteklius, kuriuos veiklos visiškai užima ir jų naudoti daugiau negalima, kol neatsilaisvins. Iš šios būsenos, jeigu išteklius turi panaudojimo nustatymą tuomet po atlaisvinimo, kurį atlieka veikla, išteklius yra grąžinamas atgal į konteinerį su būsena neužimtas. Kitu atveju, būsena patampa

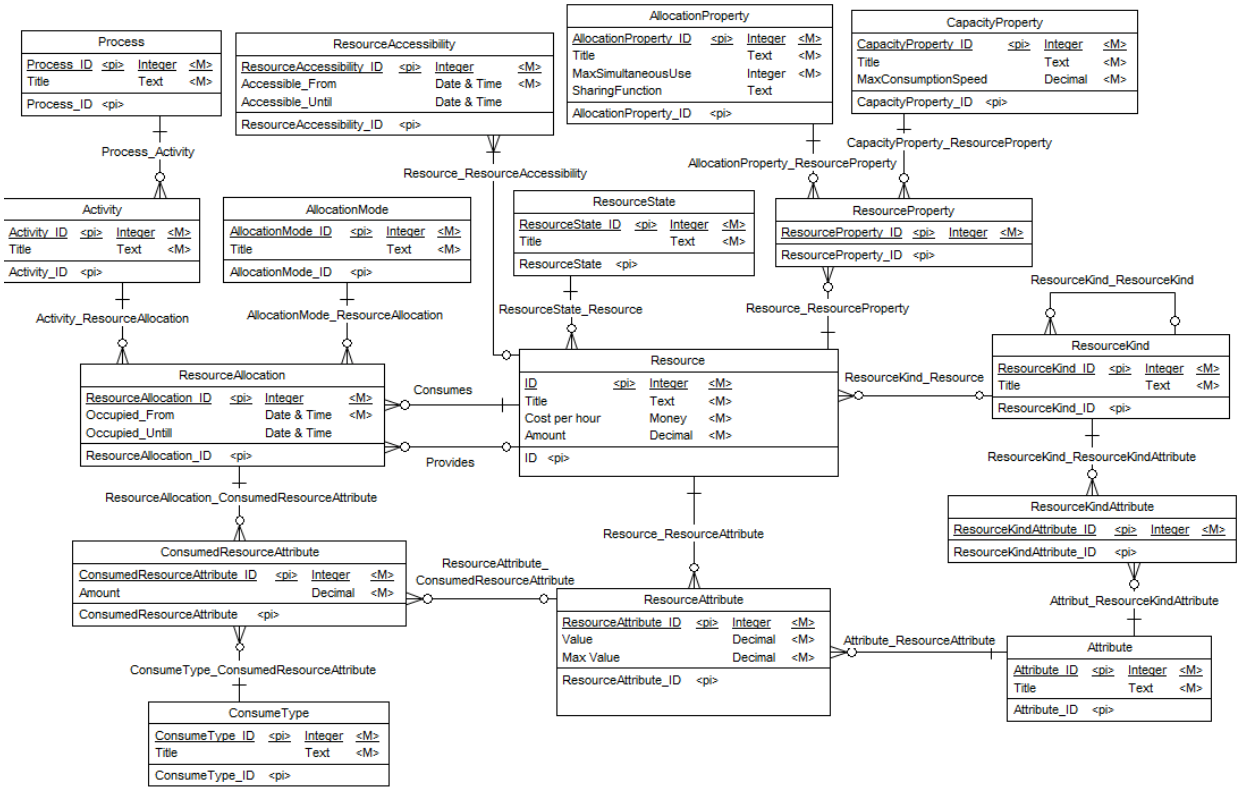
Tuščias. Ištekliaus būsena gali būti DalinaiUžimtas (paveiksle angl. *PartiallyAllocated*) tuo metu, kai $v > 1$ ir $l \neq v$ tai reiškia, kad tik dalomi ištekliai, kurie nėra pilnai priskirti prie veiklų, gali būti būsenoje DalinaiUžimtas. Tuo atveju kai išteklius yra pilnai paskirstytas tarp veiklų, jo būsena tampa PilnaiUžimtas, o kai bet kuri veikla šį išteklių atlaisvina, būsena grįžta atgal į DalinaiUžimtas. Būsenos perėjimas iš DalinaiUžimtas į Tuščias galimas, jeigu likusio kiekio (modelyje angl. *leftOver*) funkcija grąžina nulį, kitais atvejais ištekliaus būsena tampa Nepriskirtas. Jeigu ištekliaus būsena yra Tuščias tuomet išteklių galima papildyti ir jo būsena pasikeičia į Nepriskirtas.

Iš pateikto paveikslo ir paaiškinimų viršuje galima pastebti, kad Nedalomi ištekliai niekada neatsidurs būsenoje DalinaiUžimtas. Be to, išteklius gali būti būsenoje PilnaiUžimtas arba DalinaiUžimtas tik tada, jeigu jo buvusi būsena buvo Nepriskirtas arba DalinaiPriskirtas. Galiausiai, būsena Tuščias negali sekti iškarto po būsenos Nepriskirtas.

Išteklių būsenų perėjimas yra glaudžiai susijęs su veiklų vykdymu. Veiklos gali būti įvairiuose būsenose simuliacijos metu. Tačiau, šiame tyrime mes svarstome tik tris pagrindines būsenas, kurios yra svarbios išteklių modeliui t. y.: Aktyvi, Pasyvi ir Sustabdyta. Veiklos su būsena Aktyvi yra vykdomos šiuo metu, Pasyvi tai veiklos, kuriuos šiuo metu yra nevykdomos, ir Sustabdyta tai veiklos kurios negali būti vykdomos dėl trūkstamų išteklių.

3.2.3. Išteklių koncepcinis duomenų modelis

Tam kad simuliacijos išteklius saugoti duomenų bazėje visų pirma yra reikalinga išteklių duomenų bazės modelis. 11 paveiksle yra pateikiamas koncepcinis išteklių duomenų bazės modelis. Šis modelis buvo sukurtas remiantis 2.7 skyriuje pateiktais reikalavimais.



11 pav. Išteklių koncepcinis duomenų modelis

4 lentelėje yra pateikiamas reikalavimas ir jo paaiškinimas, kaip įgyvendinta modelyje.

4 lentelė. Išteklių modelio reikalavimai ir jų įgyvendinimo modelyje paaiškinimai

Nr.	Reikalavimas	Paaiškinimas
1.	Ištekliai turi būti priskiriami prie veiklos. Taip pat keli ištekliai gali būti priskiriami vienai veiklai	Šis reikalavimas buvo įgyvendintas sujungus išteklių (modelyje angl. *Resource*) su išteklių priskyrimų (modelyje angl. *ResourceAllocation*) ir veiklos (modelyje angl. *Activity*) lentelėmis. Kiekvienam ištekliui ir veiklai galim priskirti daug išteklių priskyrimų.
2.	Ištekliui būtina nurodyti jo panaudojimo kainą ir turimą kiekį	Reikalavimas buvo įgyvendintas sukūrus išteklių lentelėje du būtinus atributus t. y. kaina per valandą (modelyje angl. *Cot per hour*) ir kiekis (modelyje angl. *Amount*)
3.	Ištekliai turi gebėti pateikti atsakymus apie jų būseną bet kuriuo laiko momentu	Reikalavimas buvo įgyvendintas sujungus išteklių lentelę su išteklių būsenų lentele, kurioje yra aprašomos visos galimos išteklių būsenos.

4 lentelės tęsinys

4.	Ištekliams gali būti nurodytas neribotas skaičius charakteristikų, kurios apibūdina įvairias jų savybes	Reikalavimas buvo įgyvendintas dvejopai. Pirma neribotą charakteristikų skaičių galima nurodyti išteklių tipui (modelyje angl. *ResourceKind*). Išteklius gali turėti tipą pavyzdžiui „Peugeot Boxer“ tipas yra automobilis, o tuo tarpu automobilio tipas yra transporto priemonė. Visi ištekliai su vienodu išteklių tipu pasižymės tam tikromis charakteristikomis. Antra neribotas charakteristikas galima nurodyti tiesiogiai ištekliui per ištekliaus atributų (modelyje angl. *ResourceAttribute*) lentelę.
5.	Panaudojus išteklius turi būti užtikrinamas jų grąžinimas į saugyklą	Šį reikalavimą įgyvendins trigeris, kuris kiekvieną kartą pasikeitus ištekliaus būsenai (modelyje angl. *ResourceState*) į Nepriskirtas bus patikrinama ar išteklius yra panaudojamas ir jeigu taip gražinamas visas naudojamas išteklių kiekis.
6.	Išteklių modelis turi palaikyti dalomumą	Reikalavimas buvo įgyvendintas pridėjus priskyrimų nustatymo (modelyje angl. *AllocationProperty*) lentelė ir pridėjus atributus maksimalus naudojimas vienu metu (modelyje angl. *MaxSimultaneousUse*), kuris parodo maksimalų skaičių veiklų galinčių vienu metu naudoti išteklių, ir dalinimosi funkciją (modelyje angl. *SharingFunction*), kuri nurodo kaip ištekliai yra padalinami.

5 lentelėje pateikiamas rodinys, kuris vaizduoja išteklių charakteristikų pavyzdį, kuris gali būti aprašytas įgyvendinus pademonstruotą išteklių modelį. Visos išteklių rūšys turi standartinį charakteristikų skaičių, kurio yra užpildomos įvedus naują išteklių, tačiau jeigu prireikia naujos charakteristikos apibūdinančios tik tą vienintelį išteklių visada galima pridėti naują. Veikloje

nurodžius, kad reikalinga transporto priemonė, kurios kategorija B ir spalva yra juoda bus parenkamas išteklius atitinkantis šias savybes.

5 lentelė. Išteklių charakteristikų pavyzdys

Išteklius pavadinimas	**Rūšis**	**Charakteristikos pavadinimas**	**Reikšmė**	**Matavimo vienetas**	**Ištekliaus kiekis**
Peugeot Boxer	Transporto priemonė	Tūris	12	m^3	2
Peugeot Boxer	Transporto priemonė	Kategorija	B	Null	2
Peugeot Boxer	Transporto priemonė	Degalai	Dyzelis	Null	2
Peugeot Boxer	Transporto priemonė	Variklis	1.6	Null	2
Peugeot Boxer	Transporto priemonė	Metai	2002	Null	2
Peugeot Boxer	Transporto priemonė	Spalva	Juoda	Null	2

6 lentelėje yra pateikiamas pavyzdys, kuriame vaizduojama, kaip duomenų bazėje aprašomas išteklių priskyrimas veiklai. Kaip matome iš modelio išteklius yra priskiriamas veiklai, o veikla savo ruožtu yra priskiriama prie proceso. Tai leidžia sužinoti procese vykdomas veiklas ir kokie ištekliai tose veiklose yra naudojami. Taip pat veiklai galima priskirti išteklių, kuris yra naudojamas kito ištekliaus, pavyzdžiui: „Antanas Antanaitis“ naudoja išteklių „*Peugeot Boxer*“ tam, kad pristatytų prekes.

6 lentelė. Išteklių priskyrimo pavyzdys

Proceso pavadinimas	**Veiklos pavadinimas**	**Priskiriama nuo**	**Priskiriama iki**	**Ištekliaus pavadinimas**	**Naudojantis išteklius**
Prekių pristatymas	Paruošti prekes	2015-04-16 12:23	2015-04-16 16.12	Jonas Jonaitis	Null
Prekių pristatymas	Pakrauti prekes	2015-04-16 16.12	2015-05-17 09.00	Petras Petraitis	Null
Prekių pristatymas	Pristatyti prekes	2015-05-17 09.00	2015-05-17 12.00	Peugeot Boxer	Antanas Antanaitis

3.3. Įrankio architektūra

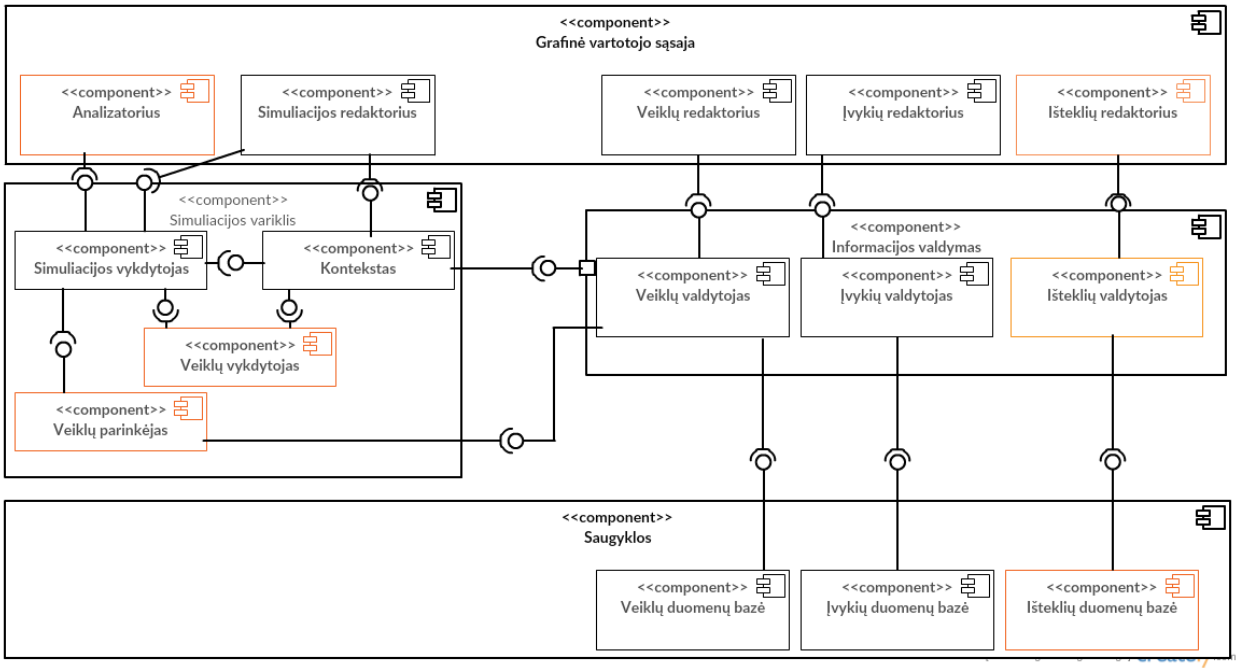
Aukšto lygio požiūriu, į tarnybų pagrindu kurtą programinę įrangą galime žiūrėti, kaip į daugelį tarnybų, kurios bendrauja viena su kita per pranešimus. Kitaip sakant į tarnybas galime pažiūrėti, kaip į bendrus sprendimus. Tačiau, kiekviena tarnyba susideda iš programinių komponentų, lygiai taip pat kaip ir kitos programos, ir šie komponentai logiškai gali būti sugrupuoti į pateikimo, verslo ir informacijos sluoksnius. Kitos programos gali naudotis tarnybomis nežinodamos, kaip jos buvo įgyvendintos. Tokiu pat pagrindu, buvo sukurta „DBP Sim“ įrankio architektūra (12 pav.). Ši architektūra aiškiai išdėstyta į tris sluoksnius, tai yra: Grafinė vartotojo sąsaja (angl. *Graphical User Interface* arba GUI), loginiai komponentai ir informacijos saugyklos. Kiekvieną iš šių sluoksnių panagrinėkime detaliau.

Grafinė vartotojo sąsaja. Šioje dalyje yra išvardinti komponentai atsakingi už naudotojo bendravimą su simuliaciją. Analizavimo komponentas atsakingas už simuliacijos rezultatų pateikimą naudotojui. Šiame komponente galime valdyti norimą informaciją apie simuliaciją. Galime pakeisti stebimus kintamuosius, stebėti jų pasikeitimą laike ir vertinti jų pokytį. Simuliacijos redaktorius yra atsakingas už simuliacijos paleidimą, stabdymą bei nustatymų keitimą, kaip: simuliacijos greitis, stebimi kintamieji, naujų taisyklių pridėjimas ir t. t. Veiklų redaktorius atsakingas už veiklų pridėjimą ir redagavimą. Veiklų įvedimo metu yra patikrinama ar įvesta veikla yra teisinga ir ar ją galima teisingai sukompiliuoti.

Simuliacijos variklis: Komponentas atsakingas už simuliacijos vykdymą ir rezultatų gavimą. Šis komponentas nusprendžia kuri veikla yra vykdoma. Šis komponentas reaguoja į sistemoje pasirodžiusius įvykius ir iš aibės veiklų yra išrenkama vykdoma veikla, patikrinama sąlyga ir įvykdoma veikloje aprašytos užduotys.

Informacijos valdymas: tai komponentas atsakingas už informacijos perdavimą tarp naudotojų ir duomenų bazių. Šis komponentas atlieka veiksmus su duomenimis ir juos transformuoja iš duomenų bazių į programai suprantamus duomenis ir atvaizduoja juos vartotojui.

Saugyklos: įrankyje yra trys atskiros saugyklos tai veiklų duomenų bazė ir įvykių duomenų bazė, kuriuose yra saugomi įvykiai ir veiklos aprašytos programoje (XML formatu) ir yra išteklių modelis skirtas aprašyti simuliacijoje naudojamiems ištekliams.



12 pav. Išplėsta „DBP Sim“ įrankio architektūra

3.4. Apibendrinimas

Pasiūlytas metodas skirtas DVP dalomų išteklių simuliacijai, kuris leidžia įvertinti išteklius verslo procesų simuliacijoje. Apibrėžtos galimos išteklių būsenos ir būsenų perėjimų ribojimai. Pateiktas koncepcinis išteklių duomenų bazės modelis. Pasiūlytas būdas, kaip reikia išplėsti esamą įrankį, pateikta įrankio išplėsta architektūra. Remiantis šia architektūra bus realizuojamas simuliacijos įrankio prototipas.

4. DINAMINIŲ VERSLO PROCESŲ IŠTEKLIŲ SIMULIACIJOS PROTOTIPO REALIZACIJOS APRAŠYMAS

4.1. Kūrimo priemonės

Dinaminių verslo procesų išteklių simuliavimo prototipas buvo realizuojamas naudojantis Microsoft™ kompanijos sukurtais produktais: integruota kūrimo aplinka (angl. *Integrated Development Environment*) *Visual Studio Community 2016* ir reliacinių duomenų bazių valdymo sistema *SQL Server 2014 Express*. Realizacijos metu buvo naudojamas *Windows Server 2012* serveris. *Visual Studio Community 2016* yra naudojama *C#* programavimo kalba ir .NET 4.5 versijos programavimo karkasas. Tam, kad reliacinėje duomenų bazių valdymo sistemoje įgyvendintą išteklių modelį susieti su programavimo kalba buvo panaudota *.NET LINQ to SQL* karkasas. Taip pat kuriant duomenų bazę buvo naudojamas *Sybase PowerDesigner 16.1*. Šie produktai skirti Windows operacinei sistemai. Šie produktai buvo pasirinkti nes:

- Pradinė prototipo versija buvo sukurta naudojantis *C#* programavimo kalbą ir *.NET* karksą;
- Visi šie įrankiai, nevykdant komercinės veiklos, yra nemokami;
- *C#* yra aukšto lygio programavimo kalba, taip pat šiai programavimo kalbai yra sukurta daug papildomų bibliotekų, kurios palengvina programavimą;
- Visual Studio ir SQL Server įrankiai yra vieni iš plačiausiai naudojamų ir geriausiai išvystytų technologijų kuriant programinę įrangą.

4.2. Įgyvendintų komponentų aprašymas

Remiantis 10 paveikslu buvo įgyvendinti arba patobulinti šie komponentai:

- Išteklių redaktorius – komponentas atsakingas už išteklių pridėjimą, nuskaitymą, atnaujinimą ir pašalinimą iš išteklių duomenų bazės. Šis komponentas yra valdomas naudotojo, pasinaudojus grafinę naudotojo sąsają. Naudotojas taip pat atsakingas už išteklių valdymą t. y. jų pridėjimą prie veiklos, papildymą ir pašalinimą;
- Išteklių valdytojas – komponentas atsakingas už išteklių priskyrimą (duomenų bazės lygyje) ir priskirtų išteklių atlaisvinimą. Šiame komponente yra atliekami patikrinimai (ar turimas pakankamas išteklių kiekis? Ar išteklius gali būti priskirtas? Ir kiti patikrinimai).
- Išteklių duomenų bazė – komponentas atsakingas už simuliacijoje naudojamų išteklių saugojimą ir ataskaitų generavimą;
- Simuliacijos rezultatų analizatorius – Komponentas atsakingas už simuliacijos rezultatų atvaizdavimą naudotojui. Šis komponentas buvo patobulintas ir šiuo metu vaizduoja ne tik

rezultatus, bet ir jų vidurkius. Taip pat vaizduojama veiklų vykdymo istorija, sudaromas įvykdytų veiklų grafas;

- Veiklų vykdytojas – komponentas atsakingas už parinktų simuliacijos veiklų įvertinimą ir vykdymą. Šis komponentas buvo patobulintas pridedant funkcijas skirtas darbui su ištekliais.
- Veiklų parinkėjas – komponentas atsakingas už vykdomos veiklos parinkimą. Buvo atsisakytą veiklos parinkimo pagal prioritetą, šiuo metu yra atrenkamos visos veiklos ir jos įvykdomos.

4.3. Realizacijos aprašymas

Šiame skyriuje yra pateiktas išteklių realizacijos aprašymas „DBP Sim“ įrankyje. Šiame skyriuje pateiksime esminių komponentų vidinę struktūrą ir veikimo principą.

4.3.1. Išteklių redaktorius

„DBP Sim“ įrankyje veikla susideda iš šių parametrų (M. Remeika, 2014):

- Identifikatoriaus – unikalus raktas kiekvienai veiklai;
- Pavadinimas – veiklos pavadinimas;
- Sąlyga – esant teigiamam sąlygos rezultatui, veikla yra parenkama vykdymui;
- Veiksmas – užduočių seka, kurios yra vykdomos esant teisingam sąlygos rezultatui.

Norint prie veiklos vykdymo priskirti išteklius reikia pasinaudoti išteklių priskyrimo funkcijomis. 7 lentelėje yra pateikiamas detalus šių funkcijų aprašymas.

7 lentelė. Sukurtų funkcijų aprašymas

Funkcijos pavadinimas	**Funkcijos parametrai**	**Funkcijos aprašymas**
RES_CHECKALLOCATION	Ištekliaus pavadinimas	Funkcija skirta sužinoti ar norimas išteklius yra pilnai užimtas ar ne. Funkcija gražina *true*, jeigu išteklius gali būti priskirtas, arba *false*, jeigu išteklius negali būti daugiau priskirtas prie veiklos. Tikrinimo metu yra atsižvelgiama į ribojimus aprašytus 3.2.2 skyriuje.

7 lentelės tęsinys

RES_ALLOCATE	Ištekliaus pavadinimas, kiekis, santykinis laiko vienetas, veiklos pavadinimas	Funkcija skirta priskirti norimą išteklių prie veiklos. Funkcija visų pirma patikrina nurodytą kiekį, jei sąlyga tenkinama išteklius priskiriamas veiklai. Veiklos pavadinimas nėra būtinas kintamasis, jeigu šis kintamasis nėra nurodomas išteklius bus priskiriamas prie dabartinės veiklos. Priskyrimo metu veiklai yra suteikiamas unikalus identifikatorius.
RES_CANCEL	Ištekliaus pavadinimas, priskyrimo identifikatorius, kiekis, santykinis laiko vienetas	Funkcija skirta atšaukti išteklių priskyrimą ir grąžinti tam tikrą ištekliaus kiekį atgal į laisvo ištekliaus būseną.
RES_FILL	Ištekliaus pavadinimas, kiekis, santykinis laiko vienetas	Funkcija skirta išteklių papildymui. Po tam tikro laiko nurodytas kiekis yra pridedamas prie ištekliaus.
RES_FINDAMOUNT	Ištekliaus pavadinimas, ištekliaus nustatymas, santykinis laiko vienetas	Funkcija, kuri pagal nurodytą ištekliaus nustatymą suranda turimą ištekliaus kiekį.
RES_RELEASE	Ištekliaus pavadinimas, santykinis laiko vienetas	Funkcija, kuri skirta atlaisvinti priskirtus išteklius (jeigu išteklius yra panaudojamas) po tam tikro laiko arba pašalinti (jeigu išteklius yra sunaudojamas) tam tikrą kiekį.

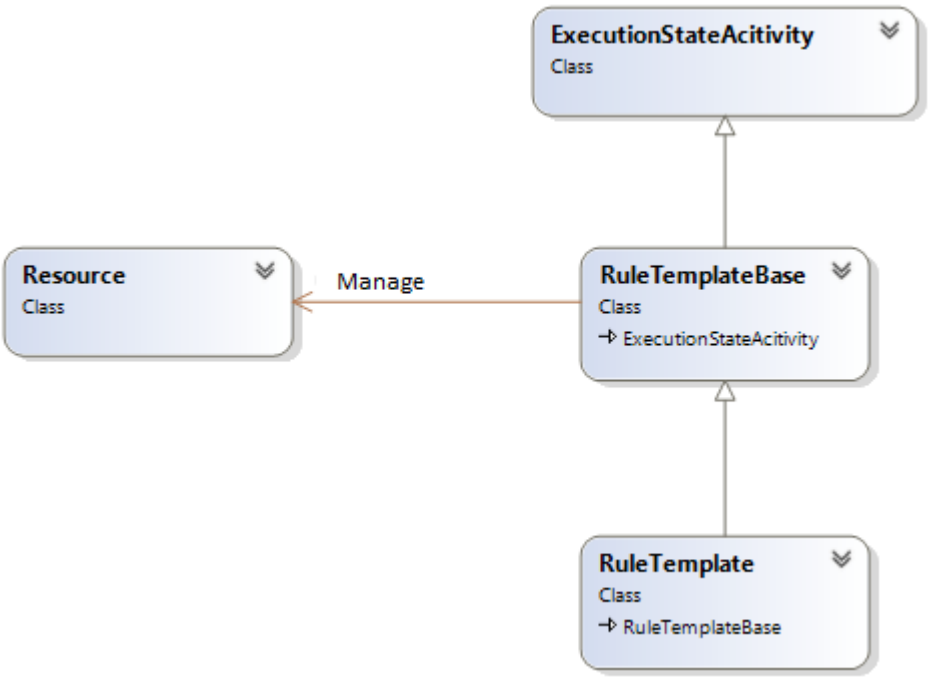
Kaip matome iš 7 lentelės daugumai funkcijų reikia nurodyti santykinį laiko vienetą. Tai parametras reikalingas tam, kad būtų galima įvertinti kiek laiko truko veikla ir sudaryti simuliacijos laiko tėkmę. Pagal šį būdą buvo pakoreguotos veiklų funkcijos. Lentelėje 8 detalei aprašytos šios funkcijos.

8 lentelė. Patobulintų funkcijų aprašymas

Funkcijos pavadinimas	**Funkcijos parametrai**	**Funkcijos aprašymas**
CREATE	Pavadinimas, santykinis laiko vienetas	Funkcija, kuri per tam tikrą laiką sukuria faktą.
RETRACT	Pavadinimas, santykinis laiko vienetas	Funkcija, kuri per tam tikrą laiką pašaliną sukurtą faktą.

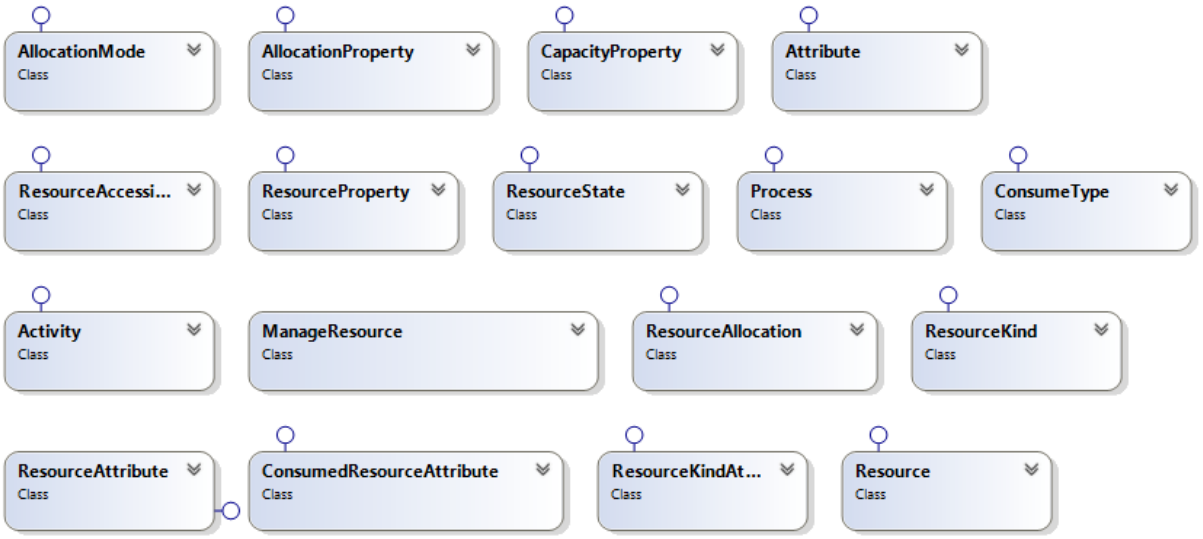
4.3.2. Išteklių valdytojas

Sukurta išteklių klasė (angl. *Resource*) yra susieta su taisyklių šablono baze (angl. *RuleTemplateBase*), kuri yra skirta veiklų aprašymui ir nuajų veiklų kurimui (13 pav.). Tai leidžia kuriant naujas veiklas prie jų priskirti tai veiklai atsakingus išteklius. Po to aukšto lygio kalba užrašyta veikla yra transformuojama į vykdomą programinį kodą. Veiklos aprašomos sukurta, „Microsoft Visual Basic“ paremta programavimo kalba.



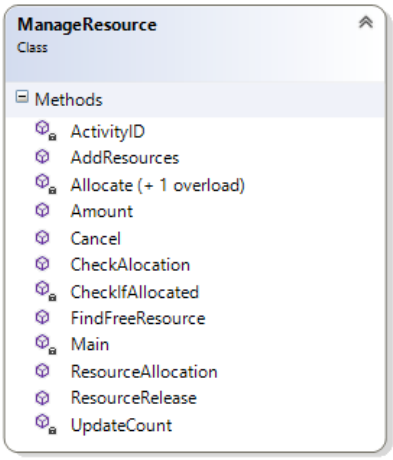
13 pav. Išteklių priskyrimo prie veiklos klasių struktūra

14 pav. yra pateikiamos sukurtos išteklius aprašančios klasės. Jos susietus su duomenų baze. Išteklių valdymo klasė (angl. *ManageResource*) yra klasė skirta susieti programinį kodą su duomenų bazių valdymo sistema. Ši klasė, naudodama *Linq to SQL* paketą, užtikrina tinkamą komunikavimą tarp kodo ir duomenų bazės.



14 pav. Išteklių klasių diagrama

15 pav. yra pavaizduoti išteklių valdymo klasei sukurti metodai. Šie metodai yra iškviečiami iš klasės taisyklių šablono bazė. Šioje klasėje esantys metodai (žiūrėti 4.3.1 skyriaus 7 lentelę) iškviečia išteklių valdymo klasės metodus. Naudotojas priskirdamas išteklius turi užtikrinti, kad pasibaigus veiklos vykdymui priskirti ištekliai bus atlaisvinti. Esant nepakankamam išteklių kiekiui funkcijos grąžins pranešimą, jog išteklių kiekis nepakankamas, jų papildymų turi pasirūpinti vartotojas sukurdamas atskirą veiklą, kaip išteklių papildymas.



15 pav. Išteklių valdymo klasė

9 lentelėje pateikiamas detalus sukurtų metodų aprašymas.

9 lentelė. Išteklių valdymo klasės metodai

Metodo pavadinimas	**Aprašymas**
AddResource	Prideda naują išteklių į išteklių duomenų bazę. Šio išteklio statusas tampa nepriskirtas ir po pridėjimo jį galima priskirti veiklai.
Cancel	Funkcija, kuri atšaukia priskirtą išteklių ir dalį arba visus nesunaudotus išteklius grąžina atgal į saugyklą.
CheckAllocation	Funkcija skirta patikrinti ar išteklius priskirtas ir ar galima jam priskirti dar papildomų veiklų. Ši funkcija leidžia įgyvendinti išteklių dalomumą.
FindFreeResource	Grąžina pirmą nepriskirtą išteklių atitinkantį tam tikras nurodytas išteklių charakteristikas
ResourceRelease	Duomenų bazėje atlaisvina nurodyta išteklių ir nustato jo statusą į nepriskirtas arba tuščias.
UpdateCount	Padidina nurodyto ištekliaus kiekį duomenų bazėje.
ResourceAllocation	Funkcija skirta duomenų bazėje pakeisti išteklio statusą iš nepriskirto arba dalinai priskirto į dalinai arba pilnai priskirtą. Taip pat ši funkcija pagal išteklių dalinimo funkciją nustato, kaip išteklius turėtų būti padalinamas per skirtingas veiklas. Ši funkcija taip pat leidžia įgyvendinti išteklių dalomumą. Priskirdama išteklių ši funkcija taip pat atsižvelgia į išteklio pasiekiamumą, kurį galima nurodyti duomenų bazėje.

4.3.3. Kitų komponentų aprašymas

Šiame skyriuje trumpai aprašysime kiekvieną iš likusių komponentų ir jų įgyvendinimą.

Išteklių duomenų bazė – buvo sukurta transformavus koncepcinį duomenų bazės modelį (11 pav.) į fizinį duomenų bazės modelį pasinaudojus *Sybase PowerDesigner* programos

funkcionalumu. Fizinis duomenų bazės modelis tuomet buvo transformuotas į *SQL Server 2014 Express* kodą, kuris automatiškai sugeneruoja duomenų bazės schemą.

Simuliacijos rezultatų analizatorius – šis komponentas buvo papildytas funkcionalumu, kuris suskaičiuoja gautų simuliacijos rezultatų vidurikius. Šis funkcionalumas palengvina gautų duomenų analizę ir vykdant daug to pačio verslo proceso iteracijų lengviau nustatyti prie kokio rezultato artėja simuliacija (17-18 pav.). Tai buvo įgyvendinta kiekviename žingsnyje, įvykdžius veiklą, surandamas rezultatas ir suskaičiuojamas gautų rezultatų vidurkis. Tokiu atveju galima stebėti ne tik kaip kinta rezultatai, bet ir vidurkis.

Veiklų parinkėjas – šiame komponente buvo panaikintas kolizijų sprendimo mechanizmas, kuris veiklą atrinkdavo pagal nurodytą prioritetą. Šiuo metu kiekviename žingsnyje yra atrenkamos visos veiklos, kurios tenkina sąlygas ir yra įgyvendinamos, tokių būdu vienu metu gali būti įvykdytos kelios veiklos. Tai buvo įgyvendinta veiklų vykdytojo komponentui paduodant ne vieną veiklą, o veiklų objektą.

Veiklų vykdytojas – šis komponentas buvo papildytas galimybe vykdyti ne vieną veiklą, tačiau aibę taisykles atitinkančių veiklų.

4.4. Apibendrinimas

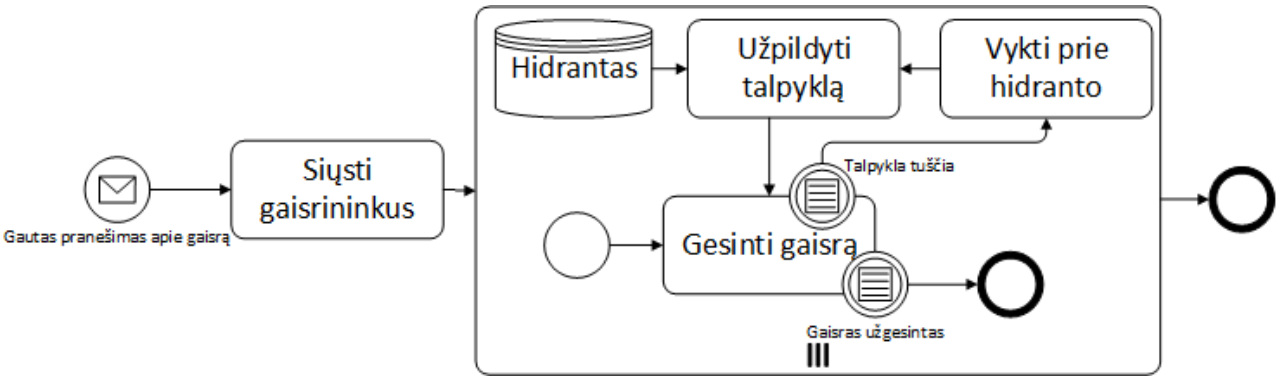
Siūlomo metodo realizacija parodė, kad naudojant šiuolaikines programavimo priemones galima realizuoti dalomų išteklių simuliavimo būdą. Iš to galima padaryti prielaidą, jog galima simuliacijos įrankiuose įgyvendinti sudėtingų išteklių modelį, kuris leistų gauti tikslesnius simuliacijos rezultatus.

5. EKSPERIMENTINIS TYRIMAS

Šiuo eksperimentiniu tyrimu yra siekiama išsiaiškinti, ar pasiūlytas metodas tinka simuliuoti išteklius ir įrodyti išteklių dalomumo svarbą gaunamiems simuliacijos rezultatams. Todėl buvo sukurtas sintetinis verslo proceso modelis, jis transformuotas į simuliacijos modelį ir įvykdyta simuliacija. Toliau šiame skyriuje aprašomi įvykdytas eksperimentas.

5.1. Nagrinėjamas uždavinys

Kaip pavyzdys buvo panaudotas supaprastintas gaisro gesinimo procesas, kuris pavaizduotas 16 paveiksle. Šiame verslo procese hidrantas yra dalomas išteklius, kuris yra naudojamas tam, kad pripildytų gaisrinių mašinų vandens rezervuarus, kai jie tampa tušti. Hidrantu vienu metu gali naudotis kelios gaisrinės mašinos. Tarkime, kad hidrantu vienu metu gali naudotis 2 gaisrinės mašinos, o tam, kad užgesintų gaisrą reikalingas vandens kiekis yra lygus 8 gaisrinėse mašinose telpančiam vandens kiekiui. Gaisrinės stotis turi 6 vienodos talpos gairines mašinas. Taip pat, yra žinoma, jog vidutiniškai nuvykti iki nelaimės vietos užtrunka 10 minučių, ištuštinti vandens talpyklą užtrunka 15 minučių ir pripildyti iš naujo talpyklą užtrunka 30 minučių. Simuliacijos tikslas būtų nustatyti optimalų siunčiamų gaisrinių mašinų kiekį tam, kad gaisrą užgesintų per trumpiausią laiką ir atlikti analizę, kaip išteklių dalomumas veikia rezultatus, t. y., kas atsitinka, jeigu maksimalų veiklų skaičių, galintį vienu metu naudoti išteklių, padidintume.



16 pav. Supaprastintas gaisro gesinimo procesas

Tipiškai simuliacijos įrankiai negali palaikyti dalomų išteklių dėl per daug paprastų išteklių modelių naudojamų įrankiuose. Pateiktame pavyzdyje, jeigu modeliuotume hidrantą, kaip nedalomą išteklių reikštų, kad visos siųstos mašinos gesinti gaisrą lauktų eilėje tam, kad prisipildytų vandens talpyklas (jeigu pildytųsi vienu metu).

5.2. Eksperimento eiga

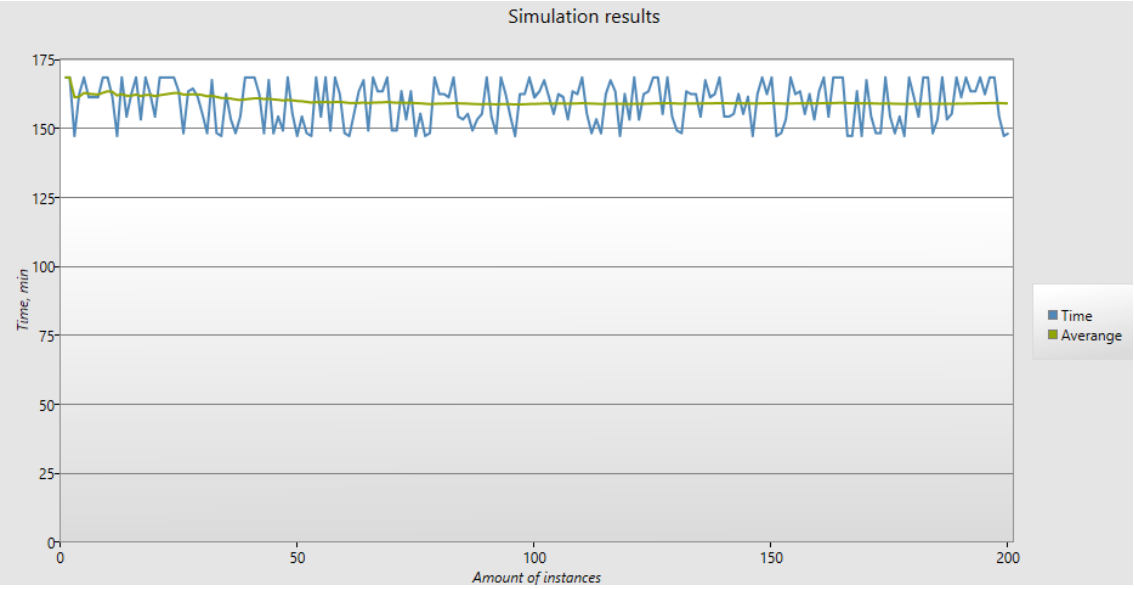
Verslo procesas pavaizduotas 16 paveikse buvo transformuotas į simuliacijos modelį, kuris yra aprašomas taisyklėmis.

Tam, kad išryškėtų skirtumai, buvo atlikta keturios skirtingos simuliacijos: pirma, kurioje hidrantas modeliuojamas, kaip nedalomas išteklius, o kitose trijose simuliacijose išteklius modeliuojamas, kaip dalomas su skirtingais gaisrinių mašinų kiekiais vienu metu galinčiais pildytis vandenį. Kiekviena simuliacija susideda iš 6 vykdymų, kiekvienas iš jų yra vykdomas su skirtingu gaisrinių mašinų skaičiumi, kuris kinta nuo 1 iki 6. Tam, kad pasiektume pastovų rezultatą kiekvienos simuliacijos metu buvo vykdoma 200 iteracijų. Šis skaičius buvo nustatytas stebint proceso vidutinį veiklų vykdymo laiką. Buvo nustatyta, jog po 200 iteracijų simuliacijos rezultatai nusistovi ir kinta neženkliai.

Simuliacijos pradiniai duomenys yra gaisrinių mašinų kiekis ir kiek vienu metu hidrantas gali pildyti skirtingų gaisrinių automobilių. Pasibaigus simuliacijai įvertinamas gaisro gesinimo laikas į kitus kriterijus (kaip kaina) nėra atsižvelgiama.

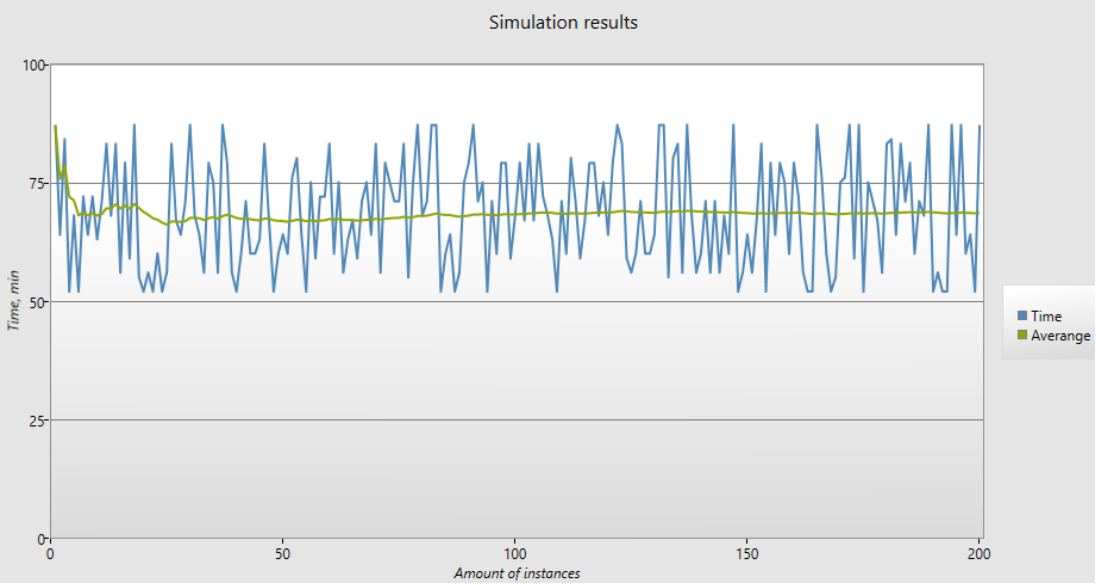
5.3. Eksperimento rezultatai

Simuliacijos rezultatai pavaizduoti 17 ir 18 paveiksuose. 17 paveiksle yra pavaizduoti 4 gaisrinių automobilių gaisro gesinimo rezultatai, kai hidrantas modeliuojamas kaip nedalomas išteklius, ir atitinkamai kai hidrantas modeliuojamas kaip dalomas išteklius ir juo gali dalintis 4 gaisrinės mašinos vienu metu, yra pavaizduotas paveiksle 18.



17 pav. Simuliacijos rezultatai, kai hidrantas modeliuojamas kaip nedalomas išteklius (4 gaisrinės mašinos)

Tada kai hidrantas modeliuojamas kaip nedalomas išteklius, gesinimo laikas kinta tarp 148 ir 170 minučių. Palyginimui, kai hidrantas modeliuojamas kaip dalomas išteklius, laikas kinta tarp 52 ir 90 minučių. Galima pastebėti, jog atvejyje su dalomais ištekliais procesas yra žymiai efektyvesnis ir gesinimo laikas ženkliai sumažėja. Taip pat, didesnis laiko pokytis yra dėl hidranto dalinimosi, nes gaisrinės automobiliai mažiau laiko stovi eilėje ir gesinimo rezultatai pasiskirsto per didesnę aibę reikšmių.



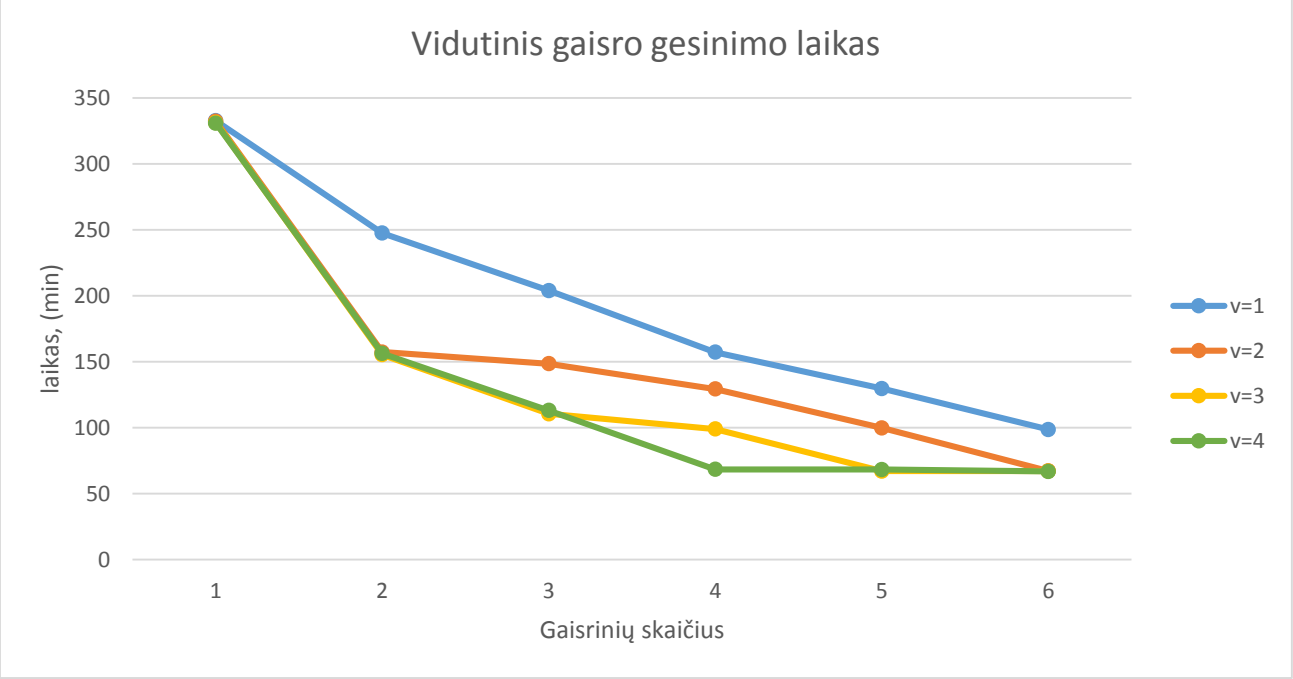
18 pav. Simuliacijos rezultatai, kai hidrantas modeliuojamas kaip dalomas išteklius (4 gaisrinės mašinos ir hidrantas palaiko 4 gaisrines mašinas vienu metu)

10 lentelė apibendrina gautus rezultatus visais 4 galimais variantais. Galima pastebėti, kad vidutinis laikas reikalingas užgesinti gaisrą yra beveik vienodas visais atvejais, kai yra siunčiamas vienas automobilis. Toks rezultatas yra gaunamas, nes tokiu atveju hidrantas nėra dalomas, kai juo naudojasi tik viena gaisrinės mašina ir dalomumas rezultatams neturi jokios įtakos. Tuo atveju kai yra daugiau negu viena gaisrinės mašina, optimalus skaičius mašinų mažėja, kai hidranto dalomumo skaičius didėja. Kiekvienu dalomumo atveju optimalus gaisrinių mašinų kiekis yra skirtingas. Po to kai pasiekiamas optimalus gaisrinių mašinų skaičius rezultatai kinta paklaidų rėžyje.

10 lentelė. Apibendrinti simuliacijos rezultatai

#	Gaisrinių skaičius / Rezultatai	1	2	3	4	5	6
1	Vidutinis laikas minutėmis be išteklių dalomumo	332,67	247,43	203,88	157,22	129,74	98,68
2	Vidutinis laikas minutėmis su ištekliu dalomumu $v = 2$	332,23	157,31	148,49	129,27	99,84	67,21
3	Vidutinis laikas minutėmis su ištekliu dalomumu $v = 3$	331,32	155,26	110,48	98,95	67,12	67,04
4	Vidutinis laikas minutėmis su ištekliu dalomumu $v = 4$	330,85	156,25	113,11	68,48	68,32	66,89

19 paveiksle matome gautų rezultatų grafiką. Iš šio grafiko galime nustatyti optimalias reikšmes. Iš šio paveikslo matome, jeigu hidranto dalinimo koeficientas $v = 2$ optimalus gaisrinių skaičius būtų 6, kai $v = 3$ optimalus gaisrinių skaičius būtų 5 ir kai $v = 4$ tuomet optimumas būtų 4 gaisrinės.



19 pav. Apibendrinti simuliacijos rezultatai

5.4. Apibendrinimas

Apibendrinant, šis eksperimentas parodė pasiūlyto būdo tinkamumą ir pabrėžė šio įgyvendinimo naudą dinaminių verslo procesų simuliacijoje. Buvo įrodyta, kad išteklių

dalomumas yra svarbi problema, kuri lemia simuliacijos rezultatų tikslumą. Nagrinėjant pakankamai nesudėtingą verslo proceso pavyzdį buvo gauti rezultatai, kurie gerokai skyrėsi nuo standartinių simuliacijos rezultatų. Pasiūlytas būdas, kaip modeliuoti išteklius dinaminiuose verslo procesuose, leidžia gauti tikslesnius rezultatus ir simuliacijos modelį labiau priartino prie tikrovės.

IŠVADOS

1. Atlikus susijusių mokslinių darbų literatūros analizę, nustatyta, kad išteklių dalomumas, dinaminių verslo procesų kontekste, nėra plačiai tyrinėjama tema.
2. Atlikus verslo procesų modeliavimo ir simuliacijos įrankių analizę, nustatyta, jog įrankių gamintojai pradeda kurti įrankius skirtus dinaminių verslo procesų simuliacijai. Nors dauguma įrankių turi galimybę simuliuoti išteklius, tačiau galimybės simuliuoti dalomus išteklius nėra.
3. Remiantis išteklių modelio reikalavimais buvo pasiūlytas dinaminių verslo procesų išteklių modelis, kuris leidžia simuliuoti dalomus išteklius.
4. Eksperimentiniai tyrimai parodė, kad įgyvendintas metodas yra tinkamas norint pasiekti tikslesnius rezultatus naudojant dalomus išteklius.

TOLIMESNIŲ TYRIMŲ KRYPTYS

Tyrimo metu ir įgyvendinant pasiūlyta metodą buvo pastebėtos kitos tyrimų kryptys ir galimybės išplėsti įrankio funkcionalumą:

1. Pasiūlyto išteklių modelio išpletimas sprendžiant kitas su ištekliais susijusias problemas.
2. Išteklių rezervacijos įgyvendinimas.
3. Vartotojo sąsajos patobulinimas, pritaikant darbui su ištekliais.
4. Pridėti ataskaitų generavimo ir rezultatų palyginimo mechanizmus.
5. Ištirti dalomų išteklių įtaka simuliacijos rezultatams realiuose verslo procesuose.
6. Išteklių konkuravimo tyrimas tarp skirtingų veiklų.

LITERATŪRA

1. AnyLogic, 2016. Why AnyLogic? [*žiūrėta 2016-03-12*]. Prieiga per internetą <http://www.anylogic.com/features>.
2. Appian, 2016. Business Process Definition [*žiūrėta 2016-03-12*]. Prieiga per internetą: <http://www.appian.com/about-bpm/definition-of-a-business-process/>.
3. Aris, 2016. Aris express details. Aris Comunity [*žiūrėta 2016-03-12*], Prieiga per internetą: < https://www.ariscommunity.com/aris-express/details>.
4. Bhat, Jyoti M., Nivedita Deshmukh, 2005. Methods for modeling flexibility in business processes. *Workshop on Business Process Modeling, Design and Support (BPMDS05), Proceedings of CAiSE05 Workshops*, 1-8.
5. Bizagi, 2016. Bizagi simulation [*žiūrėta 2016-03-12*]. Prieiga per internetą < http://help.bizagi.com/processmodeler/en/index.html?simulation.htm>.
6. Bosilj-Vuksic, V., Ceric, V., Hlupic, V., 2007. Criteria for the evaluation of business process simulation tools. Interdisciplinary Journal of Information, Knowledge, andd Management, 2, 73-88.
7. BusinessDictionary, 2015. Simulation [*žiūrėta 2015-03-01*]. Prieiga per internetą: < http://www.businessdictionary.com/definition/simulation.html>.
8. Camunda, 2014. BPMN Modeling Reference [*žiūrėta 2015-03-01*]. Prieiga per internetą: <http://camunda.org/bpmn/reference/#events-parallel>.
9. Chapman, A., 2016. Business process modelling. Businessballs [*žiūrėta 2016-01-19*]. Prieiga per internetą <http://www.businessballs.com/business-process-modelling.htm>.
10. Chinosi, M., Trombetta, A., 2012. BPMN: An introduction to the standard. Computer Standards & Interfaces, 34(1): 124-134.
11. Davenport, Thomas, H., Short, James, E., 1990. The new industrial engineering: information technology and business process redesign. Sloan management review, 31(4).
12. Dijkman, M., Dumas, M., Ouyang, C., 2008. Semantics and analysis of business process modelis in BPMN. Information and Software Technology, 50.12: 1281-1294.
13. Eriksson, H., Penker, M., 1999. Business Modeling with UML: Business Patterns at work. Canada: John Wiley & Sons.

14. Fadel, F. G., Fox, M. S., Gruninger. M., 1994. A generic enterprise resource ontology. *Enabling Technologies: Infrastructure for Collaborative Enterprises, 1994. Proceedings., Third Workshop on*. IEEE, 117-128.

15. Genrich, Hartmann J., Lautenbach K., 1981. System modelling with high-level Petri nets. *Theoretical computer science* 13.1: 109-135.

16. Herbst, H. 1997. A Business Rule-Oriented Conceptual Modeling Approach. Heidelberg: Physica-Verlag, 92-95.

17. IBM, 2016. WebSphere Business Modeler [*žiūrėta 2016-03-12*]. Prieiga per internetą: < http://www-03.ibm.com/software/products/en/modeler-advanced>.

18. Interfacing, 2016. What is BPM? [*žiūrėta 2016-02-24*]. Prieiga per internetą < http://www.interfacing.com/Literature/what-is-bpm>.

19. Jansen-Vullers, M., Netjes, M., 2006. Business process simulation–a tool survey. In *Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, Aarhus, Denmark* (Vol. 38), 1-20.

20. Mili, H., Jaoude, G. B., Lefebvre, E., Tremblay, G., Petrenko, A., 2003. "Business process modeling languages: Sorting through the alphabet soup. *Manuscript, 1-57.*

21. Morgan, T., 2002. Business rules and information systems: aligning IT with business goals. Addison-Wesley Professional.

22. Neiger, D., Rotaru, K., Churilov, L., 2009. Supply Chain Risk Identification With Value Focused Process Engineering, Journal of Operations Management 27(2), 154-168.

23. Object Management Group, 2014. Business Process Model and Notation [*žiūrėta 2015-03-01*]. Prieiga per internetą: < http://www.bpmn.org/ >.

24. Pall, G., 1987. Quality Process Management. Englewood Cliffs: Prentice-Hall.

25. Recker, Jan C., 2011 "X-aware business process management." *BP Trends* 8.12, 1-7.

26. Remeika, M., 2014. Magistro tezės. Taisyklėmis grindžiamų dinaminių verslo procesų modeliavimo ir simuliacijos būdas. Vilnius: Vilniaus Gedimino technikos universitetas.

27. RPG Core Document, 2011. Distributed simulation [*žiūrėta 2015-04-21*]. Prieiga per internetą: < http://www.msco.mil/documents/Core_Federation01_Federation.pdf>

28. Simprocess, 2016. About Simprocess [*žiūrėta 2016-03-12*]. Prieiga per internetą < http://simprocess.com/about-simprocess/>.

29. Simulace, 2011. Stochastic simulation [*žiūrėta 2015-04-21*]. Prieiga per internetą: < http://www.simulace.info/images/Stochastic_simulation.pdf>.

30. Stroppi, L., Chiotti, O., Villarreal, P., 2011. A BPMN 2.0 extensions to define the resource perspective of business process modelis. XIV Iberoamerican-Conference on Software Engineering, 25-38.

31. Szelagowski, M, 2014. Static and Dynamic Processes. Dynamic BPM [*žiūrėta 2016-03-05*]. Prieiga per internetą < http://www.bpmleader.com/2014/08/28/static-and-dynamic-processes/>.

32. Van Der Aalst, Voorhoeve, M., 2016. Business Process Simulation. Lecture notes 2II75:6-8.

33. Van Der Aalst, Wil MP, Hofstede, A., Weske, M., 2003 Business process management: A survey. *Business process management*. Springer Berlin Heidelberg, 1-12.

34. Van Der Aalst, Wil MP., 2015. Business process simulation survival guide. Handbook in Business Process Management, 337-370.

35. Vasilecas , O., Rima, A., 2015. Verslo procesų simuliacija: verslo procesų ir išteklių modelių reikalavimai. Mokslas-Lietuvos ateitis, 1-6.

36. Vasilecas, O., Laureckas, E., Rima, A., 2014. Analysis of using resources in business process modelling and simulation. Applied computer systems: scientific journal of RTU, 19-25.

37. Vasilecas, O.; Kalibatienė, D. 2008. *Šiuolaikinės duomenų bazės*, Klaipėdos universiteto leidykla, Klaipėdos universitetas, p. 180-190.

38. W. M. Van der Aalst, 2010. Business process simulation revisited. In: *Enterprise and Organizational Modeling and Simulation*. Springer Berlin Heidelberg, 1 - 14.

39. Whatls, 2016. Discrete event simulation (DES) [*žiūrėta 2015-04-21*]. Prieiga per internetą: <http://whatis.techtarget.com/definition/discrete-event-simulation-DES>.

40. Zeng, L., Flaxer, D., Chang, H., Jeng, J., 2002. PLM flow—Dynamic Business Process Composition and Execution by Rule Inference. Technologies for E-Services, 141-150.

41. Ziegler, W., 2015. d8rules [*žiūrėta 2015-04-05*]. Prieiga per internetą <http://d8rules.org/content/rules-success-story>.

PRIEDAI

1. Veiklos pavyzdys. Veikloje patikrinama ar egzistuoja faktas „Assess“. Jeigu ši sąlyga tenkinama tada vykdomas veiklos užduotys.

Title:

Assess Response

Condition:

```
Fact.Exists("Assess")
```

Body:

```
Retract("Assess",1)
$Duration.Time=$Duration.Time+ ActivityDuration
Dim Rand = New System.Random().Next(1,10)
IF(Rand<3) Then
  Create("Appoinment",11)
End IF
IF(Rand>=3 AND Rand<6) Then
  Create("ApplicationOK",8)
End IF
IF(Rand>=6 AND Rand<9) Then
  Create("Rejection",5)
End IF
IF(Rand>=9) Then
  Create("Cancel",5)
End IF
$Duration.Time=$Duration.Time+ ActivityDuration
```

2. Veiklos pavyzdys, kurioje yra priskiriami ištekliai. Tai atlieka funkcija RES_ALLOCATE.

Title:

Perform Order

Condition:

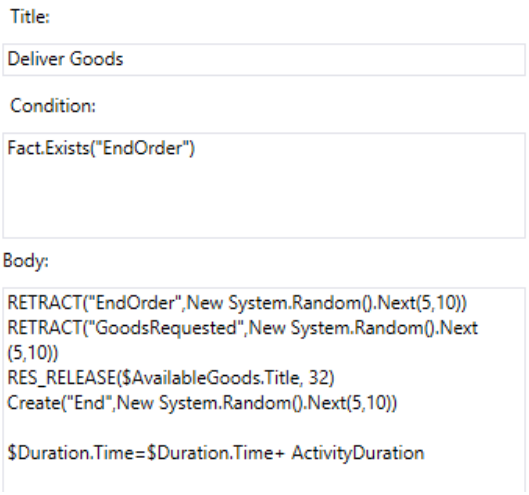
```
$AvailableGoods.Goods>=$GoodsRequested.Goods
```

Body:

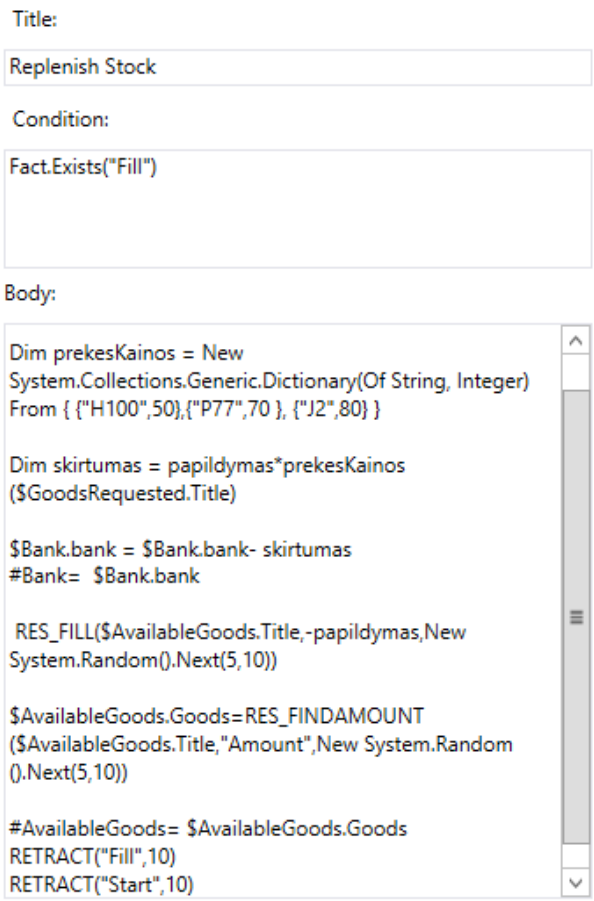
```
 RES_ALLOCATE($AvailableGoods.Title,
$GoodsRequested.Goods,New System.Random().Next(5,10))

 If Not Fact.Exists("ID") Then
  Create("ID",10)
End If
$ID.Value=ID
$AvailableGoods.Goods=RES_FindAmount
($AvailableGoods.Title,"Amount",New System.Random().Next
(5,10))
#AvailableGoods= $AvailableGoods.Goods
If  $AvailableGoods.Goods<=50 Then
  Create("Fill",New System.Random().Next(5,10))
End If
If $AvailableGoods.Goods>50 Then
  RETRACT("Start",New System.Random().Next(5,10))
End If
Create("Bill",New System.Random().Next(5,10))
$Duration.Time=$Duration.Time+ ActivityDuration
```

3. Veiklos pavyzdys, kurioje yra atlaisvinami ištekliai. Tai atlieka funkcija RES_RELEASE



4. Veiklos pavyzdys, kurioje ištekliai yra papildomi. Tai atlieka funkcija RES_FILL.



5. Grafikas sudarytas iš įvykusių veiklų.

