

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Darius Pauliukevičius

**Veiklos procesų nuolatinio imitacinio optimizavimo
metodika ir jos taikymas seniūnijų viešosioms paslaugoms**

Magistro darbas

Darbo vadovas:

prof. dr. Lina Nemuraitė

Kaunas, 2009

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Darius Pauliukevičius

**Veiklos procesų nuolatinio imitacinio optimizavimo
metodika ir jos taikymas seniūnijų viešosioms paslaugoms**

Magistro darbas

Recenzentas

dr. Audronė Janavičiūtė

2009-01-

Vadovas

prof. dr. Lina Nemuraitė

2009-01-

Atliko

IFM-3/4 gr. stud.

Darius Pauliukevičius

2009-01-

Kaunas, 2009

Pauliukevičius D. **Methodology for Continuous Optimization of Business Processes via Simulation: Application for Public Services of Elderships**. Master of Information systems engineering / supervisor prof. PhD L. Nemuraite; Faculty of Informatics, Kaunas University of Technology – Kaunas, 2009 – 75 p. Keywords: continuous optimization, business process management, business modeling, eldership services

Summary

Nowadays there are many problems in business process development and management. Processes are not always arranged correctly or well organized. It is important to find out disadvantages and to eliminate them. This problem is solved using simulation. But this method does not always show optimal values. The better way to maintain, analyze and correct processes is to use continuous optimization methods. It makes regular business process observation and improvement. The continuous optimization helps to adopt the process to new environmental changes.

The field of this work is business process modeling and simulation.

The object of this work is business process continuous optimization. This method is quite new and gives a good opportunity to analyze and improve real world business processes. It adds a new functional ability to discrete-simulation and minimizes programming work.

The aim of the work is to create continuous optimization mechanism, to adjust it to Public Services of Elderships model and to arrange an experimental research. This research gives better knowing of critical points of real business process and those problematic areas can be fixed after all.

Model of Public Services of Elderships simulation showed the weak point of business process. One of the stages showed the extended time than it was expected. The problem was eliminated by simplifying the initiation of the service request and reducing the segment time. The results showed it is effective to use the continuous optimization for practical purposes. In this way the money and all the resources can be saved more efficiently.

Turinys

Turinys	4
Paveikslėlių sąrašas	6
Lentelių sąrašas	7
Santrumpų sąrašas	8
Įvadas	9
1. Veiklos procesų optimizavimo metodų analizė	11
1.1. Veiklos procesų imitacinio optimizavimo analizė	11
1.1.1. Imitacinio optimizavimo samprata.....	11
1.1.2. Klasikinis imitacinio optimizavimo aprašymas.....	12
1.1.3. Metaheuristinis imitacinio optimizavimo aprašymas	13
1.1.4. Diskretinio įvykio imitavimas (angl. discrete-event Simulation).....	15
1.1.5. Nuolatinis imitacinis optimizavimas.....	16
1.2. Veiklos procesų modeliavimo metodų analizė.....	17
1.3. Seniūnijos teikiamų viešųjų paslaugų veiklos procesas	18
1.3.1. Seniūnijos teikiamų viešųjų paslaugų organizacijos ir jos aplinkos analizė	20
1.3.2. Seniūnijos teikiamų viešųjų paslaugų vartotojų tikslai ir problemos	21
1.4. Veiklos procesų įgyvendinimo technologijų analizė.....	22
1.4.1. Dinaminio internetinio puslapio apžvalga.....	22
1.4.2. Dinaminio internetinio puslapio perkeltą į darbatalį apžvalga	24
1.4.3. Technologijų palyginimas	24
1.5. Esamų sprendimų analizė.....	25
1.6. Analizės išvados	28
2. Veiklos procesų nuolatinio imitacinio optimizavimo sistemos reikalavimų specifikacija ir analizė	29
2.1. Reikalavimų specifikacija	29
2.2. Nuolatinio imitacinio optimizavimo sistemos esybių diagrama	35
2.3. Reikalavimų analizės apibendrinimas	36
3. Nuolatinio imitacinio optimizavimo sistemos projektas	37
3.1. Veiklos procesų nuolatinio imitacinio optimizavimo modelis ir sistema.....	37
3.2. Komponentų modeliai.....	39
3.2.1. Loginė sistemos architektūra	39
3.2.2. Vartotojo paslaugos.....	40
3.2.3. Veiklos paslaugos	41

3.3. Esybių klasių diagrama	42
3.4. Detalus projektas	43
3.5. Elgsenos modelis	45
3.5.1. Integratoriaus posistemis	45
3.5.2. Stebėjimo sistemos posistemis.....	46
3.6. Duomenų bazės schema.....	48
3.7. Realizacijos modelis	49
4. Eksperimentinis nuolatinio imitacinio optimizavimo taikymas seniūnijos paslaugoms.....	51
4.1. Seniūnijų teikiamų viešųjų paslaugų modelio sudarymas	51
4.2. Seniūnijos teikiamų viešųjų paslaugų imitacinis optimizavimas	57
4.3. Optimizuoto veiklos proceso apibendrinimas	59
5. Nuolatinio imitacinio optimizavimo sistemos taikymo seniūnijos teikiamoms viešosioms paslaugoms realizacija	60
5.1. Sistemos įdiegimas	60
5.2. Veiklos procesų integravimas	60
5.3. Veiklos procesų vaizdavimo ir charakteristikų sąsaja	62
5.4. Seniūnijos teikiamų viešųjų paslaugų sistemos prisijungimas.....	64
5.5. Seniūnijos teikiamų viešųjų paslaugų sistemos sąsaja	65
5.6. Seniūnijos teikiamų viešųjų paslaugų sistemos asmens kortelių kūrimas	67
5.7. Seniūnijos teikiamų viešųjų paslaugų sistemos suformuoti dokumentai.....	67
5.8. Nuolatinio imitacinio optimizavimo taikymo seniūnijos teikiamų viešųjų paslaugų sistemai rezultatai.....	69
Išvados	70
Literatūros sąrašas.....	71
Priedai	73
1. Stebėjimo sistemos naudojama XML transformacija (1 dalis iš 2)	73
2. Stebėjimo sistemos naudojama XML transformacija (2 dalis iš 2)	74
3. Integratoriaus naudojama XML transformacija	75

Paveikslėlių sąrašas

1 pav.	Imitacinis optimizavimas metaeuristiniu optimizatoriumi	14
2 pav.	Metaeuristinis optimizatorius su metamodelio filtru.....	14
3 pav.	Sistemos analizavimo būdai.....	15
4 pav.	Modelių klasifikacija	15
5 pav.	Imitacinio optimizavimo integravimas su nuolatinio optimizavimu.....	16
6 pav.	Teikiamų viešųjų paslaugų perkėlimas į elektroninę erdvę.....	19
7 pav.	Žmonių procentas, kurie naudojami internetu	19
8 pav.	Organizacijos aplinkos modelis	21
9 pav.	Taikomoji programa kuriama „klientas – serveris“ principu.....	23
10 pav.	Interneto taikomųjų programų modelis	23
11 pav.	Dinaminio internetinio puslapio perkėlimo į darbatalį sąsaja tarp technologijų	24
12 pav.	Panaudojimo atvejų modelis	31
13 pav.	Analitiko ir nuolatinio optimizavimo sistemos sekų diagrama.....	34
14 pav.	Darbuotojo ir nuolatinio optimizavimo sistemos sekų diagrama.....	35
15 pav.	Nuolatinio imitacinio optimizavimo sistemos esybių diagrama	36
16 pav.	Nuolatinio imitacinio optimizavimo sistemos komponentai	37
17 pav.	Nuolatinio imitacinio optimizavimo sistemos procesas	38
18 pav.	Sistemos loginė architektūra	39
19 pav.	Stebėjimo posistemio vartotojo paslaugų klasių diagrama.....	40
20 pav.	Integratoriaus posistemio vartotojo paslaugų klasių diagrama	40
21 pav.	Stebėjimo sistemos posistemio veiklos paslaugų klasių diagrama	41
22 pav.	Integratoriaus posistemio veiklos paslaugų klasių diagrama.....	42
23 pav.	Esybių klasių diagrama.....	43
24 pav.	Panaudojimo atvejo „importuoti veiklos procesą į stebėjimo sistemą“ sekų diagrama	45
25 pav.	Panaudojimo atvejo „eksportuoti veiklos procesą“ sekų diagrama.....	46
26 pav.	Panaudojimo atvejo „priskirti sistemos daviklius“ sekų diagrama	46
27 pav.	Panaudojimo atvejo „stebėti optimizavimo parametrus“ sekų diagrama	47
28 pav.	Panaudojimo atvejo „keisti optimizavimo parametrus“ sekų diagrama.....	47
29 pav.	Duomenų bazės diagrama.....	48
30 pav.	Komponentų diagrama.....	49
31 pav.	Komponentų vaizdavimas fiziniemis duomenų laikmenomis	50
32 pav.	Sistemos diegimo diagrama	50
33 pav.	Seniūnijos teikiamų viešųjų paslaugų modelis (1 dalis iš 3)	54
34 pav.	Seniūnijos teikiamų viešųjų paslaugų modelis (2 dalis iš 3)	55
35 pav.	Seniūnijos teikiamų viešųjų paslaugų modelis (3 dalis iš 3)	56
36 pav.	Įėjimo taškų pasiskirstymas	57
37 pav.	Veiklos proceso pradinė užduotis „Užsakymo priėmimas“.....	58
38 pav.	Optimizuotas veiklos procesas „Prašymo spausdinimas“ ir „Užsakymo įrašo atnaujinimas“ ..	58
39 pav.	Veiklos procesų integravimo įrankio langas.....	61
40 pav.	Vidinis integratoriaus sistemos importavimo langas.....	62
41 pav.	Veiklos procesų vaizdavimo ir stebėjimo langas	62
42 pav.	Seniūnijos teikiamų viešųjų paslaugų sistemos prisijungimo langas	64
43 pav.	Seniūnijos teikiamų viešųjų paslaugų sistemos pagrindinis langas	65
44 pav.	Seniūnijos teikiamų viešųjų paslaugų sistemos asmens kortelių kūrimas.....	67
45 pav.	Seniūnijos teikiamų viešųjų paslaugų sistemos pažymos langas	68
46 pav.	Lentelės, jungiančios bendrus duomenis	69

Lentelių sąrašas

1 lentelė.	Internetinių puslapių kūrimo programavimo kalbų skirstymas	23
2 lentelė.	Įgyvendinimo technologijų palyginimas	24
3 lentelė.	Programinių sprendimų grupavimas pagal imitacinio optimizavimo tipus	25
4 lentelė.	Programinių sprendimų lyginimas pagal optimizatorių ir modeliavimo standartą	26
5 lentelė.	Programinių sprendimų lyginimas pagal nuolatinio imitacinio optimizavimo mechanizmo požymį.....	27
6 lentelė.	Programinių sistemų įvertinimų lentelė.....	27
7 lentelė.	Panaudojimo atvejo „Projektuoti veiklos procesus“ specifikacija	31
8 lentelė.	Panaudojimo atvejo „Integruoti veiklos procesus“ specifikacija	32
9 lentelė.	Panaudojimo atvejo „Analizuoti sistemos daviklius“ specifikacija	33
10 lentelė.	Panaudojimo atvejo „Administruoti realizuotą sistemą“ specifikacija	33
11 lentelė.	Panaudojimo atvejo „Naudoti realizuotą sistemą“ specifikacija.....	34
12 lentelė.	Seniūnijų teikiamų viešųjų paslaugų modelio vaidmenų ir resursų matrica.....	52
13 lentelė.	Seniūnijų teikiamų viešųjų paslaugų modelio trukmės matrica.....	52
14 lentelė.	Seniūnijų teikiamų viešųjų paslaugų modelio prieinamumo matrica.....	53
15 lentelė.	Seniūnijų teikiamų viešųjų paslaugų realaus modelio sprendinių tikimybės matrica	53
16 lentelė.	Procesų trukmė	57
17 lentelė.	Veiklos objektų sąrašas.....	63
18 lentelė.	Veiklos objektų būsenų sąrašas.....	63
19 lentelė.	Seniūnijos teikiamų viešųjų paslaugų sistemos pagrindinio meniu aprašymas	66
20 lentelė.	Seniūnijos teikiamų viešųjų paslaugų sistemos standartinių dokumentų sąrašas	68
21 lentelė.	Procesų trukmė	69

Santrumpų sąrašas

- AJAX** – svetainių programavimo technologija naudojanti HTTP, JavaScript, XML, XSL priemones (angl. Asynchronous JavaScript and XML)
- API** – taikomųjų programų kūrimo sąsaja (angl. Application Programming Interface)
- BPEL** – veiklos procesų vykdymo kalba (angl. Business Process Execution Language)
- BPMI** – grupė, orientuota į veiklos procesų modeliavimo standartų sudarymą (angl. Business Process Modeling Initiative)
- BPMN** – veiklos procesų modeliavimo notacija (angl. Business Process Modeling Notation)
- DB** – organizuotas duomenų rinkinys, duomenų bazė (angl. Database)
- ES** – Europos Sąjunga (angl. Europe Union)
- G2B** – valstybė verslui (angl. Government to Business)
- G2C** – valstybė klientui (angl. Government to Customer)
- G2G** – valstybė valstybei (angl. Government to Government)
- GUI** – grafinė vartotojo sąsaja (angl. Graphical User Interface)
- HTTP** – pagrindinis protokolas pasiekti informacijai pasauliniame tinkle. (angl. HyperText Transfer Protocol)
- IT** – informacinės technologijos (angl. Information Technologies)
- KPI** – veiklos rodiklius fiksuojantys indikatoriai (angl. Key Performance Indicator)
- LR** – Lietuvos Respublika (angl. Republic of Lithuania)
- SOA** – į paslaugas orientuota architektūra (angl. Service Oriented Architecture)
- SOAP** – protokolas, per kurį prieinama prie publikuojamų paslaugų. SOA architektūros dalis (angl. Simple Object Access Protocol)
- UDDI** – registrams skirtas protokolas, per kurį prieinama prie publikuojamų paslaugų. SOA architektūros dalis (angl. Universal Description, Discovery, and Integration)
- VPV, BPM** – veiklos procesų valdymas (angl. Business Process Management)
- WSDL** – speciali kalba XML pagrindu, skirta paslaugų apibūdinimui. SOA architektūros dalis (angl. WEB Services Description Language)
- WWW** – žiniatinklis, pasaulinis tinklas (angl. World Wide Web)
- XML** – duomenų struktūrų bei jų turinio aprašomoji kalba (angl. eXtensible Markup Language)
- XSL** – kalba, aprašanti XML dokumento transformaciją į HTML dokumentą arba į kitokios struktūros XML dokumentą (angl. eXtensible Stylesheet Language)

Ivadas

Šiandieniniame pasaulyje vyksta spartūs pokyčiai, kuriuos lemia informacinių ir komunikacinių technologijų raida, todėl atsiranda būtinybė taikyti naujoves įvairių sričių valdymo procese. Naujovių integravimą į kasdienių procesų veiklą galima atlikti kompiuterizuojant veiklos procesus, t.y. perkelti juos į elektroninę terpę.

Vykstant sparčiam veiklos procesų vystymuisi neišvengiama tam tikrų problemų. Procesai ne visada būna gerai apgalvoti, suderinti ar organizuoti. Dėl to reikia išsiaiškinti esamų procesų trūkumus ir juos pašalinti. Šiai problemai spręsti taikomas imitacinis optimizavimas. Jo esmė paremta tuo, kad sukuriamas procesą vaizduojantis modelis ir nustatomos procesų charakteristikos keičiant tam tikrus parametrus, tokius kaip etapų trukmė, aptarnaujančio personalo skaičius, programinės įrangos spartos parametrai, materialinių lėšų kiekis ir kiti.

Didėjant veiklos procesų valdymo programinės įrangos gamintojų skaičiui didėja ir procesų imitavimo galimybės. Plečiamas produktų funkcionalumas, kuris suteikia modelio kūrimui geresnes analitines savybes, tačiau šias sistemas dar sunku pritaikyti kuriamiems veiklos procesams. Vienos sistemos yra per daug sudėtingos ir reikalaujančios daug matematinių ar statistinių žinių, o kitos daugiau orientuotos į modelių sudarymą, bet ne optimalių veiklos procesų charakteristikų nustatymą.

Besiplečiant veiklos procesų imitavimo produktų funkcionalumui atsiranda galimybė naudoti nuolatinį imitacinį optimizavimą, kuris paremtas ciklišku proceso stebėjimo, analizavimo ir koregavimo mechanizmu. Nuolatinis imitacinis optimizavimas užtikrina pastovų veiklos procesų gerinimą.

Šio darbo **tyrimo sritis** – veiklos procesų modeliavimas ir imitacinis optimizavimas.

Tyrimo objektas – veiklos procesų nuolatinio imitacinio optimizavimo procesas. Šis metodas yra pakankamai naujas ir garantuoja puikias galimybes analizuoti ir tobulinti realaus pasaulio veiklos procesus. Taip pat jis suteikia funkcionalumo imitaciniam optimizavimui ir maksimaliai sumažina darbą programavimo srityje.

Darbo **tikslas** – sukurti nuolatinio imitacinio optimizavimo mechanizmą, jį pritaikyti realizuojant seniūnijos teikiamų viešųjų paslaugų modelį ir surengti eksperimentinį tyrimą. Tyrimo duomenys suteikia geresnį supratimą apie realaus veiklos proceso etapų trūkumus ir jie gali būti greitai pašalinti.

Sukurto modelio realizacija vykdoma naudojant naujausias internetines technologijas, kurios suteikia galimybę internetinę taikomąją programą perkelti į darbatalį. Ši technologija veikia įvairiose operacinėse sistemose, o taip pat suteikia galimybę kurti naujoviškas internetui skirtas taikomąsias programas naudojantis Ajax, HTML, Flash, Flex, ExtJS technologijomis. Šių programų pranašumas yra didesnė sparta visą sąsają apdorojant vietos kompiuteryje be siuntimo, bet kartu ir tiesioginė sąsaja su duomenimis internete.

Darbo uždaviniai:

1. nustatyti imitacinio optimizavimo reikšmę gerinant veiklos procesą;
2. išanalizuoti imitacinio optimizavimo metodus ir įrankius;
3. išsiaiškinti nuolatinio imitacinio optimizavimo sistemos reikalavimus;
4. išnagrinėti taikomųjų programų kūrimo technologijas atsižvelgiant į seniūnijų teikiamų viešųjų paslaugų specifiką;
5. pasirinkti vieną iš išnagrinėtų veiklos procesų valdymo įrankių;
6. sukurti nuolatinio imitacinio optimizavimo sistemos modelį pritaikant tinkamą technologinį sprendimą;
7. realizuoti šį sprendimą ir nustatyti trūkumus seniūnijos teikiamų viešųjų paslaugų sistemoje;
8. atlikti nuolatinio imitacinio optimizavimo eksperimentinį išbandymą ir įvertinti rezultatus;
9. pateikti pasiūlymus siekiant pagerinti seniūnijos teikiamų viešųjų paslaugų procesą.

Darbo struktūra:

1. Analizės dalis
 - ✓ Atlikti literatūros šaltinių analizę pagal savo darbo tikslus
 - ✓ Išanalizuoti imitacinio optimizavimo metodus
 - ✓ Išanalizuoti modeliavimo metodus
 - ✓ Atlikti galimų technologijų analizę modeliui sudaryti ir optimizuoti
2. Teorinė dalis
 - ✓ Sukurti nuolatinio imitacinio optimizavimo mechanizmą
3. Eksperimentinė dalis
 - ✓ Realizuoti seniūnijos teikiamų viešųjų paslaugų realų modelį
 - ✓ Integruoti nuolatinį imitacinio optimizavimo mechanizmą
4. Įvertinimas
 - ✓ Apibendrinti verslo modeliavimo kalbų privalumus ir trūkumus
 - ✓ Įvertinti modeliavimo metodų pritaikymą ir galimybes.
 - ✓ Apibrėžti technologijų teigiamas ir neigiamas savybes
 - ✓ Įvertinti verslo proceso planavimo ar pertvarkymo problemos sprendimą
 - ✓ Įvertinti nuolatinio imitacinio optimizavimo privalumus ir trūkumus

1. Veiklos procesų optimizavimo metodų analizė

Šios analizės tikslas – išnagrinėti naujas verslo modeliavimo kalbas ir įrankius, kuriuos būtų galima pritaikyti veiklos procesų optimizavimui. Apibrėžti modeliavimo įrankių imitacinio optimizavimo ypatumus. Taip pat išanalizuoti naujas internetines technologijas, reikalingas realizuoti optimizuotą modelį ir vartotojus, kurie dalyvauja kuriamame veiklos procese.

Nagrinėjant modeliavimo metodus ir įrankius bus taikomi analizės metodai:

- mokslinės literatūros analizė arba teorinė analizė ir apibendrinimas;
- lyginamoji analizė;
- apklausa (ekspertų apklausa). Šis analizės metodas bus taikomas darbo rezultatų įvertinimo etape.

1.1. Veiklos procesų imitacinio optimizavimo analizė

1.1.1. Imitacinio optimizavimo samprata

Didėjant veiklos procesų valdymo programinės įrangos gamintojų skaičiui, didėja ir procesų imitavimo galimybės. Plečiamas produktų funkcionalumas, kuris suteikia modelio kūrimui geresnes analitines savybes. Imitacinis optimizavimas grindžiamas „what-if“ scenarijumi. Šiuo metodu galima vertinti veiklos procesų pasikeitimą įvedant naujus objektus arba keičiant egzistuojančių objektų pradinis parametrus. Naudojantis imitacinio optimizavimo savybėmis, atsiranda galimybė egzaminuoti ir testuoti tam tikrus veiklos procesų sprendimus prieš įgyvendinant juos realioje aplinkoje. Pagrindinė šio optimizavimo savybė yra ta, kad skaičiavimo ir matematiniais metodais galima atvaizduoti realaus veiklos proceso eigą. Ši savybė leidžia koreguoti elektroninį procesą ir stebėti rezultatą fiktyvioje erdvėje. Tokiu būdu atsiranda galimybė nuspėti, surasti arba apskaičiuoti optimalų egzistuojantį arba naujai kuriamą veiklos procesą.

Mažiau nei trečdalis veiklos procesų programinės įrangos kūrimo organizacijų siūlo procesų modeliavimą papildyti imitacinio optimizavimo galimybe. Optimizavimo poreikis atsiranda tuomet, kai analitikams reikia nustatyti **optimalius** veiklos proceso parametrus (t.y. įėjimo taškus ir/arba struktūrinės dalis). Parametrų kiekis arba jų reikšmių asortimentas yra per didelis, kad analitikai imituotų visus galimus scenarijus, taigi, jiems reikia būdo surasti ar apskaičiuoti optimalų sprendimą. Be imitacijos dauguma realaus pasaulio problemų yra per sudėtingos modeliuoti matematinėmis formulotėmis, kurios optimizuojant veiklos procesą yra visko pagrindas. Dėl to tai yra pagrindinė problema - optimizavimo modeliai yra bejėgiai, kai susiduriama su veiklos procesų sudėtingumu ir dinamiškumu. Vien tik imitavimu taip pat nebūtų įmanoma surasti optimalaus sprendimo. Tačiau, sujungus šiuos du metodus - imitavimą ir optimizavimą (imitacinis optimizavimas) atsiranda galimybė išspręsti iškilusias problemas.

Pastarąjį dešimtmetį optimizacijos ir imitacijos jungimasis pastebimai didėjo. Pavyzdžiui, „Google.com” sistemoje atlikus paiešką „Imitacinis optimizavimas“, pateikiamos nuorodos į šešiasdešimt aštuonis tūkstančius internetinių svetainių. Šiuo metu egzistuoja labai daug algoritmų (matematinų, statistinių), kuriuos pritaikius įvairioms veiklos procesų imitacijoms, galima išgauti aukštos kokybės (optimalių) sprendimų. Lyginami gauti rezultatai maksimaliai didina sprendinių kokybę. Dauguma „abstraktaus atvejo“ arba „Monte Carlo“ imitacijos komercinių veiklos procesų kūrimo programinės įrangos paketų turi savyje optimizacijos modulius, kurie atlieka reikiamų optimalių parametrų paiešką.

Pagrindiniai žingsniai verslo procesų imitaciniam optimizavimui įvykdyti:

- apibrėžti resursus ir proceso galingumą, tai yra įmonės strategijos, žmonių resursų reikalavimai ir išlaidų matricos, užimtumo matricos, apsisprendimo matricos ir kiti;
- apibrėžti patį imitacinį optimizavimą ir jos savybes;
- įvykdyti ir išanalizuoti imitacinio optimizavimo rezultatus.

Pagrindinės imitacinio optimizavimo analizės matricos (rezultatai) yra:

- Resursų (angl. role resource) matrica – parodo kokie žmonių vaidmenys priskirti prie kiekvienos veiklos. Ši matrica taip pat nurodo kiekvieno vaidmens kainą (paprastai ji apibrėžta kaip žmogaus atlyginimas per laiko vienetą).
- Trukmės (angl. duration) matrica – parodo kiek laiko užima kiekvieno resursų vaidmens veikla.
- Prieinamumo (angl. availability) matrica – parodo žmonių rolėm priskirtą tvarkaraštį.
- Sprendimo tikimybės matrica – parodo koku būdu ar koku keliu procesas gali veikti.

Tam, kad įvykdytų imitacinį optimizavimo procesą, žmogus turi pasirinkti laiką, periodą (nuo kokios iki kokios dienos), kiek proceso įvykių turi būti įvykdytų.

1.1.2. Klasikinis imitacinio optimizavimo aprašymas

Šių dienų literatūroje egzistuoja keturi imitacinio optimizavimo metodai:

- tikimybinė aproksimacija - gradientinis metodas (angl. stochastic approximation - gradient-based approaches);
- nuoseklios atsako išorės analizės metodika (angl. sequential - response surface methodology);
- atsitiktinė paieška (angl. random search);
- šablono kelio optimizacija - taip pat žinomas kaip tikimybinis dublikatas (angl. sample path optimization - stochastic counterpart). [1]

Tikimybinės aproksimacijos algoritmai mėgdžioja gradientinį paieškos metodą, naudojamą deterministinėje optimizacijoje. Šia metodologija paremtos procedūros turi įvertinti funkcijos gradientą,

kad būtų galima nustatyti paieškos kryptį. Dėl artimo ryšio su gradientine paieška tikimybinė aproksimacija sprendžia tolydžių kintamųjų problemas. Ši metodologija naudojama pavienėms problemoms spręsti.

Nuoseklios atsako išorės analizės metodika paremta metamodelių kūrimo principais, tačiau kur kas siauresniais. Terminas „lokalaus paviršiaus atsakas“ naudojamas apibrėžti paieškos kelią (pavyzdžiui judėjimas nustatyta gradiento kryptimi) ir procesas kartojamas. Kitais žodžiais tariant, metamodeliai charakterizuojami ne viso sprendimo kontekste, bet greičiau konkretizuoja lokaliai, kad paieška yra nagrinėjama.

Atsitiktinės paieškos metodo sprendiniai juda atsitiktinai pažymėdami bet kurią vietą dabartinio taško kaimynystėje. Tai leidžia daryti prielaidą, kad kaimynystė turi būti apibrėžta kaip atsitiktinės paieškos algoritmo kūrimo dalis. Atsitiktinė paieška daugiausiai taikoma pavienėms problemoms spręsti, o jų patrauklumas paremtas teorinio konvergavimo įrodymų egzistavimu. Deja, šie teoriniai konvergavimo rezultatai praktikoje turi mažai reikšmės, kadangi čia kur kas svarbiau atrasti aukštos kokybės sprendimus per atitinkamą racionalų laiko tarpą, negu užtikrinti optimalų konvergavimą per neribotą skaičių žingsnių.

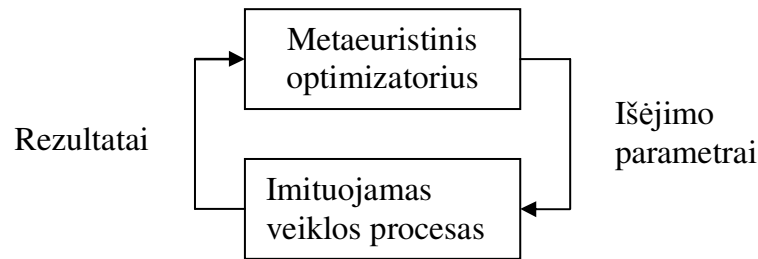
Šablono kelio optimizacija – tai metodas, kuris naudoja sukaupią informaciją, įgytą sprendžiant deterministinio tolydaus optimizavimo problemas. Pagrindinė šio metodo idėja yra deterministinių funkcijų, paremtų atsitiktiniais kintamaisiais, kur n yra kelio ilgis, optimizavimas. Modeliavimo kontekste įprastų atsitiktinių skaičių metodas naudojamas to paties šablono kelio numatymui, norint suskaičiuoti skirtingų įeities reikšmių atsaką. Šablono kelio optimizavimo pavadinimas perimtas iš įvertinto optimalaus sprendimo, kuris pasirodo yra paremtas deterministine funkcija, sukurta iš vieno šablono kelio, gauto imitaciniu modeliu. Apskritai, norint kad aproksimuojama optimizavimo problema būtų artima tikrajai optimizavimo problemai, n reikšmė turi būti didelė. [2]

Nei vienas iš šių metodų nenaudojamas komercinei programinei įrangai dėl to, kad šie metodai reikalauja žymių techninių vartotojų supratimo ir didelių laiko sąnaudų.

1.1.3. Metaeuristinis imitacinio optimizavimo aprašymas

Metaeuristiniu optimizatoriumi galima laikyti tokį optimizavimo uždavinių sprendimo metodą, kuriuo siekiama surasti aukštos kokybės, bet nebūtinai optimalų sprendinį per priimtina skaičiavimų laiką. Metaeuristiniai metodai negarantuoja gautų sprendinių optimalumo, o surasti sprendiniai paprastai yra tik lokaliai optimalūs duotos aplinkos atžvilgiu. Tuo šie metodai skiriasi tiek nuo tikslųjų algoritmų (angl. exact algorithms), kurie garantuoja optimalaus sprendinio suradimą, tiek nuo apytikslųjų (aproksimacinių) algoritmų (angl. approximate algorithms), kurie užtikrina, kad gauto sprendinio kokybė skirsis nuo optimalaus sprendinio kokybės ne daugiau, kaip iš anksto fiksuota paklaida $\varepsilon > 0$.

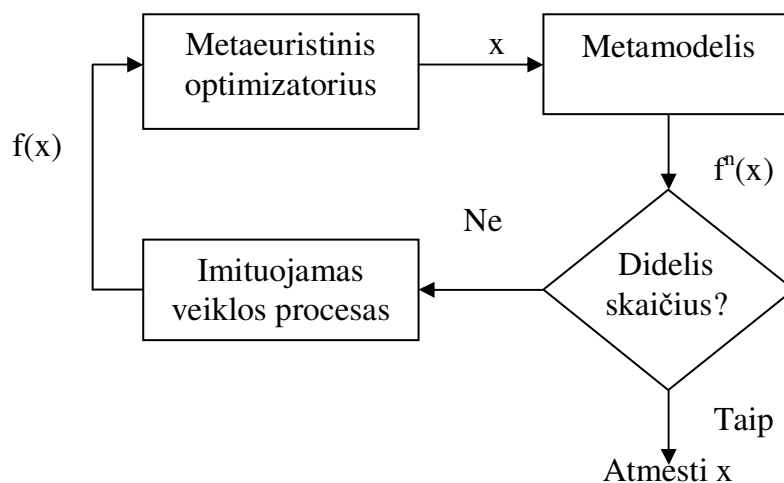
Metaeuristinį optimizatorių galima įsivaizduoti kaip „uždarą dėžę (angl. black-box)“, naudojančią metaeuristinį optimizavimo algoritmą. Ši dėžė turi įėjimo taškus – rezultatus, ir išėjimo taškus – išėjimo parametrus. Imitacinis optimizavimas metaeuristiniu optimizatoriumi pavaizduotas 1 paveiksle.



1 pav. Imitacinis optimizavimas metaeuristiniu optimizatoriumi

Dauguma optimizacinių algoritmų komerciniuose produktuose yra sukurti naudojant evoliucinį metodą. Evoliucinis metodas gamina sprendimų populiacijas. Evoliucionavimas pasiekiamas mechanizmu, kuris sukuria naujus sprendimus naudojant du arba daugiau esančius toje populiacijoje. Vieno sprendimo transformacija į naują sprendimą taip pat yra galimybė. Evoliucinį optimizavimo metodą naudoja „genetinis algoritmas (angl. genetic algorithms)“ arba „išsklaidyta paieška (angl. scatter search)“.

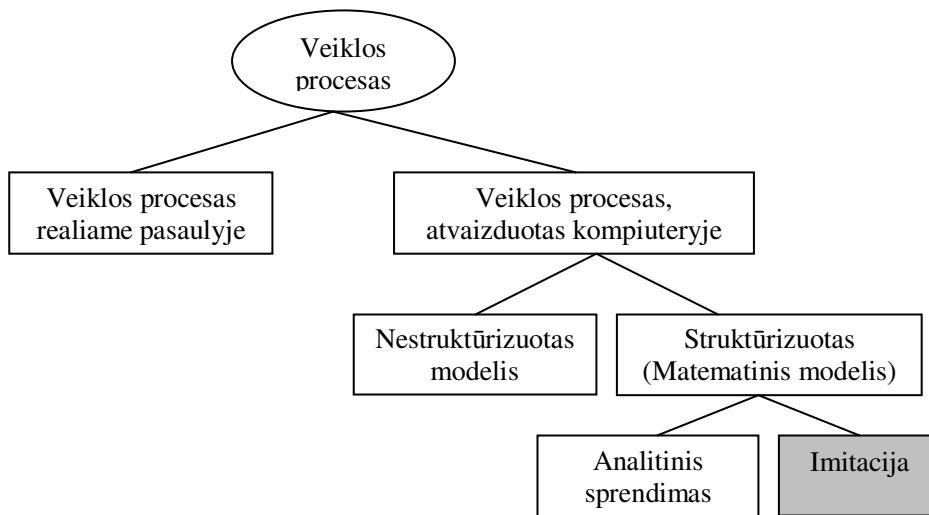
Imitacinio optimizavimo kontekste modelių imitavimas gali būti suprastas kaip „mechanizmas, kuris įeinančius parametrus paverčia į išeinančius našumo matavimus“. Kitais žodžiais tariant, imitacinio optimizavimo modelis yra funkcija, kurios detali forma nėra aiški. Šiai funkcija padavus tam tikrą kiekį parametrų gaunamas atitinkamas rezultatas. Imitacinio optimizavimo modelį žiūrint kaip į funkciją, pradėta naudoti paviršiaus atsako metodai (angl. response surface) ir metamodeliai. Paviršiaus atsako metodas išreiškiamas funkcijos skaitine reikšme, kurios yra saugojamos atliekant kiekvieną imitavimą su tam tikrais įeinančiais parametrais. Metamodelis yra algebrinis imitacinio optimizavimo atvaizdas. Metamodelis aproksimuoja paviršiaus atsako metodo rezultatus. 2 paveiksle pavaizduotas metaeuristinis optimizatorius su metamodelio filtru. [6]



2 pav. Metaeuristinis optimizatorius su metamodelio filtru

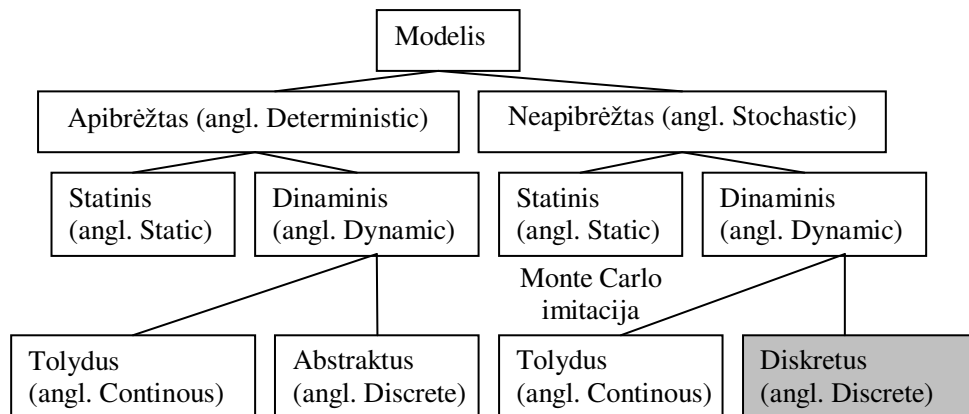
1.1.4. Diskretinio įvykio imitavimas (angl. discrete-event Simulation)

Imitacinį optimizavimą sudaro 3 pagrindinės sistemos dalys: modeliavimas (imitacija), imitavimas ir analizavimas (optimizavimas). Perkeliant realaus gyvenimo egzistuojantį veiklos procesą į fiktyvią erdvę pirmiausia reikia jį išanalizuoti. Yra keli sistemos analizavimo būdai. Nestruktūrizuotas modelis yra neapibrėžtas jokiais struktūromis, metodais ar rėmais. Sunku analizuoti sudėtingas sistemas ar jas optimizuoti. Struktūrizuotas – priešingai, sukurtas naudojant struktūrines dalis, metodus. Šiais principais sukurtiems modeliams galima taikyti matematinius optimizavimo metodus. Žemiau pateikti sistemos analizavimo metodų tipai:



3 pav. Sistemos analizavimo būdai

Modelis – koncepcinė sistemos struktūra. Veiklos procesas gali būti apibrėžtas – visos sistemos dalys yra nekintamos, arba stochastinis – gali kisti bet kuri sistemos dalis. Veiklos procesui turi įtakos laiko kintamasis, tačiau modeliuojant sistemą į šį parametą galima ir neatsižvelgti. Priklausomai nuo to, ar yra laiko kintamasis, modeliai skirstomi į statinius ir dinامينius. Statinėje sistemoje laikas nėra reikšmingas, pavyzdžiui, „Monte Carlo“ imitacija, kuri remiasi „labiausiai tikėtino spėjimo“ metodu. Dinaminėje sistemoje – priešingai, įvedamas laiko kintamasis. Jei sistemos būseną vystosi pastoviai, tai modelis vadinamas tolydžiu. Jei būseną kinta tik konkrečiame laiko intervale, tai modelis vadinamas abstrakčiu. Modelio tipai pavaizduoti 4 paveikslėlyje. [3]



4 pav. Modelių klasifikacija

1.1.5. Nuolatinis imitacinis optimizavimas

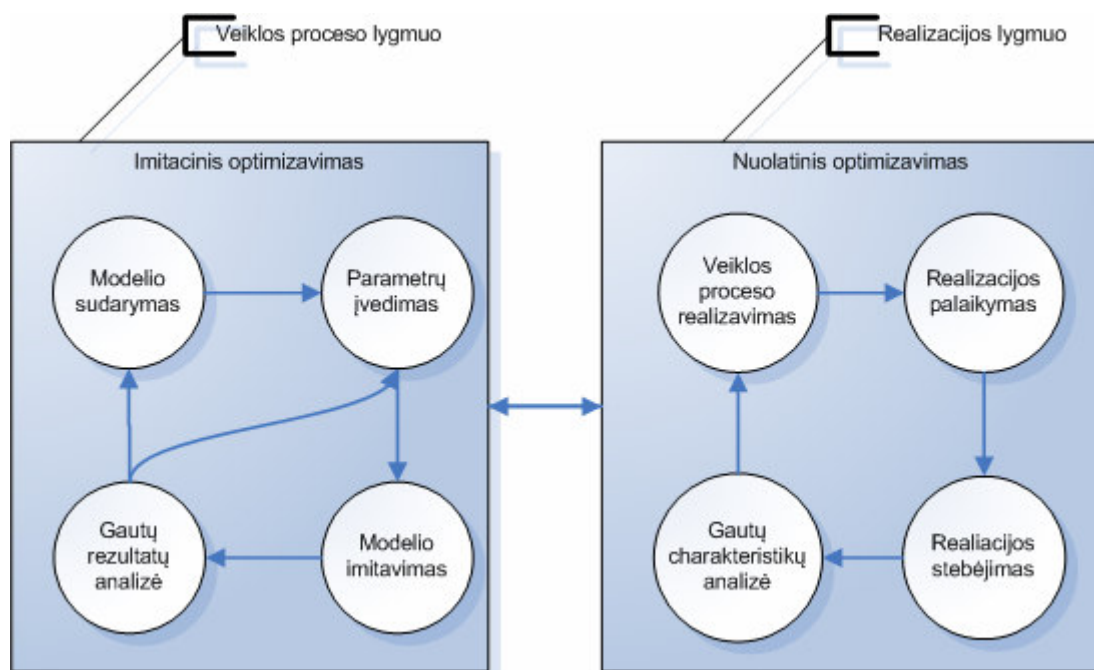
Imitacinis optimizavimas turi keletą trūkumų: problemų sprendimo neužtikrinimas ir optimizavimo neapibrėžtumas. Kad ir kaip būtų bandoma sukurti optimalų modelį, imitaciniu optimizavimo principu paremtas veiklos procesų gerinimas neužtikrina numatytų rezultatų. Tačiau šiuos trūkumus galima pašalinti nuolatinio optimizavimo mechanizmu, užtikrinant pastovų veiklos procesų optimizavimą.

Nuolatinis optimizavimas grindžiamas realizuoto veiklos proceso stebėjimu. Dauguma šiuolaikinių programinių paketų palaikančių veiklos procesų valdymą, turi savyje integruotą 4 komponentų sistemą. Šis programinis sprendimas apima veiklos procesų kūrimą, integravimą (realizavimą), realizuotos programos palaikymą serveryje ir realizacijos stebėjimą. Visi šie žingsniai dažniausiai būna susiję tarpusavyje ir integruoti apibusių ryšiu.

Realizuotos sistemos stebėjimas vyksta visą gyvavimo ciklą. Tokiame procese sistemos analitikas stebi KPI charakteristikų (angl. Key Performance Indicator) duomenis ir juos analizuoja. Veiklos procesų nuolatiniam optimizavimui naudojamos tokios pagrindinės charakteristikos:

- veiklos proceso trukmė (įvykdytų užduočių laikas);
- veiklos proceso įgyvendinimui naudojami resursai;
- veiklos proceso užduočių pralaidumas.

Pavadinime „nuolatinis imitacinis optimizavimas“ egzistuoja terminas imitacinis optimizavimas, kuris integruojamas su nuolatinio veiklos procesų optimizavimu. Tokiu principu sukurtas metodas suteikia galimybę pastoviai gerinti veiklos procesą. Kombinuotas metodas pateiktas 5 paveiksle.



5 pav. Imitacinio optimizavimo integravimas su nuolatinio optimizavimu

1.2. Veiklos procesų modeliavimo metodų analizė

Elektroninio verslo procesams modeliuoti turėtų būti naudojama verslo analitikui priimtina vaizdinė kalba, leidžianti aprašyti globalius veiklos procesus ir ilgas transakcijas; sąveikas tarp veiklos dalyvių (choreografiją); vidinius organizacijų procesus (orkestruotę); pranešimus ir veiklos dokumentus, susiejant juos su veiklos proceso žingsniais; kartu su vaizdine kalba turėtų būti ir išraiškų kalba, leidžianti aprašyti verslo taisykles ir apribojimus (angl. Business Transaction Protocol Primer; ebXML Business Process Specification Schema). Ši kalba turėtų būti realizuota įrankyje, kuris užtikrintų veiklos procesų modelių kūrimą, išsaugojimą ir vykdymą ar perdavimą vadinamiesiems veiklos procesų valdymo įrankiams. Yra daug veiklos modeliavimo kalbų, kurios gerai įgyvendina kai kuriuos iš šių reikalavimų, tačiau nėra tokios kalbos, kuri tenkintų juos visus ir užtikrintų elektroninio veiklos proceso sukūrimą nuo modelio lygmens reikalavimų iki programinės realizacijos.

BPMN (angl. Business Process Modeling Notation) notacija

BPMN yra vienas iš nedaugelio standartų, skirtų verslo procesų modeliams aprašyti. Veiklos procesai atvaizduojami virtualioje erdvėje jau senai, tačiau per pastarąjį penkmetį atsirado modeliavimo standartai, kaip tuos procesus aprašyti. BPMN yra naujas standartas, skirtas modeliuoti verslo procesų srautus, išleistas BPMI (angl. Business Process Management Initiative) organizacijos. Šios organizacijos vienas iš esminių tikslų - vykdomosios veiklos procesų kalbos XML pagrindu, pavyzdžiui, BPEL4WS (angl. Business Process Execution Language For Web Services) ir BPML (angl. Business Process Modeling Language), būtų lengvai atvaizduojamos bendra notacija.

BPMN notacijos vienas iš privalumų, kad ji lengvai suprantama visiems vartotojams: verslo analitikams, technikos ekspertams. BPMN turi pranašumų lyginant su universaliojiomis UML (angl. Unified Modeling Language) veiklos diagramomis dėl to, kad yra pritaikyta veiklai modeliuoti ir turi konkrečius elementus veiklos įvykiams, taisyklėms, jungimų tipams vaizduoti.

BPMN modelis gali būti trijų tipų:

- Privatūs veiklos procesai - vidiniai organizacijos procesai.
- Abstraktūs veiklos procesai vaizduoja sąveikavimą tarp vidinių veiklos procesų ir kitų procesų ar dalyvių.
- Bendradarbiavimo veiklos procesai vaizduoja sąveikavimą tarp dviejų ar daugiau veiklos objektų.

Visiems veiklos procesams atvaizduoti BPMN notacija specifikuoja keturias pagrindines objektų kategorijas: srauto objektus (angl. flow objects), sujungimo objektus (angl. connecting objects), sritis (angl. swimlane), žmogaus darbo produktus (angl. artifacts).

Srauto objektai yra pagrindiniai elementai, kurie apibrėžia veiklos procesus. Srauto objektai yra trijų rūšių: įvykiai (angl. events), veiklos (angl. activities), vartai (angl. gateways).

Sujungimo objektai jungia srauto objektus, arba kitus objektus. Sujungimo objektai skirstomi į: sekos srautas (angl. sequence flow), pranešimų srautas (angl. message flow), asociacija (angl. association).

Sritis – reiškia dalyvius, t.y. sritys turi vardus, kurie reiškia, kas ką daro.

Žmogaus darbo produktai naudojami pateikti papildomą informaciją apie procesus. Yra trys standartizuoti žmogaus darbo produktai: duomenų objektas (angl. data object), grupė (angl. group), anotacija (angl. annotation).

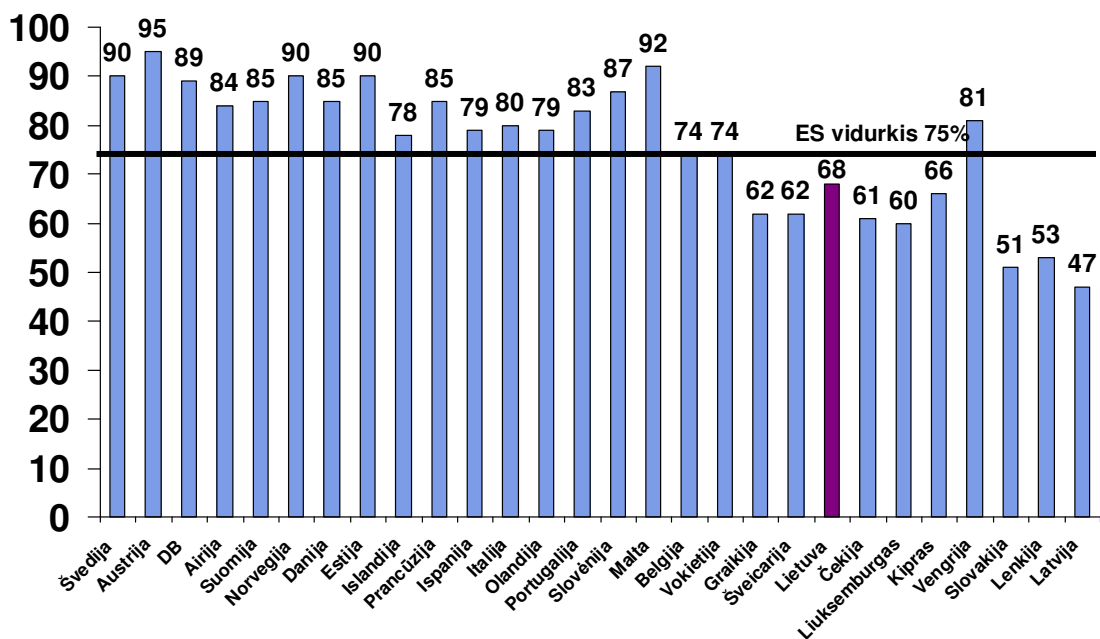
UML (angl. Unified Modeling Language) kalba

UML modeliavimo kalba gana paplitusi, lengvai išmokstama ir patogi realizuoti, specifikuoti ar dokumentuoti. Šiai kalbai būdinga diagramų įvairovė, todėl ji labai lanksti ir patogi projektuoti. UML modeliavimo kalboje diagramos skirstomos į tris kategorijas: statines, dinamines bei organizavimo, valdymo. Ši modeliavimo kalba yra nuolat tobulinama ir papildoma. Naujausia šiuo metu priinama modeliavimo kalbos UML versija - UML 2.0. Ji suteikia galimybę taip suprojektuoti modelį, kad pastarasis būtų labai artimas realizuojamai sistemai. Šis standartas papildytas elementais, skirtais projektuoti verslo modelius (duomenų saugykla, daugybė taškų tipų, veiklos suskaidymas, pertraukimas ir kt.). UML 2.0 yra universali kalba, ir veiklos procesų modeliavimas yra tik viena jos naudojimo krypčių.

Veiklos procesų modeliavimui BPEL kalba yra sukurta nemažai programinių IT paketų. Kiekvienas produktas skiriasi savo mechanizmu, tačiau visi jie laikosi pagrindinių SOA principų (SOAP, WSDL, UDDI).

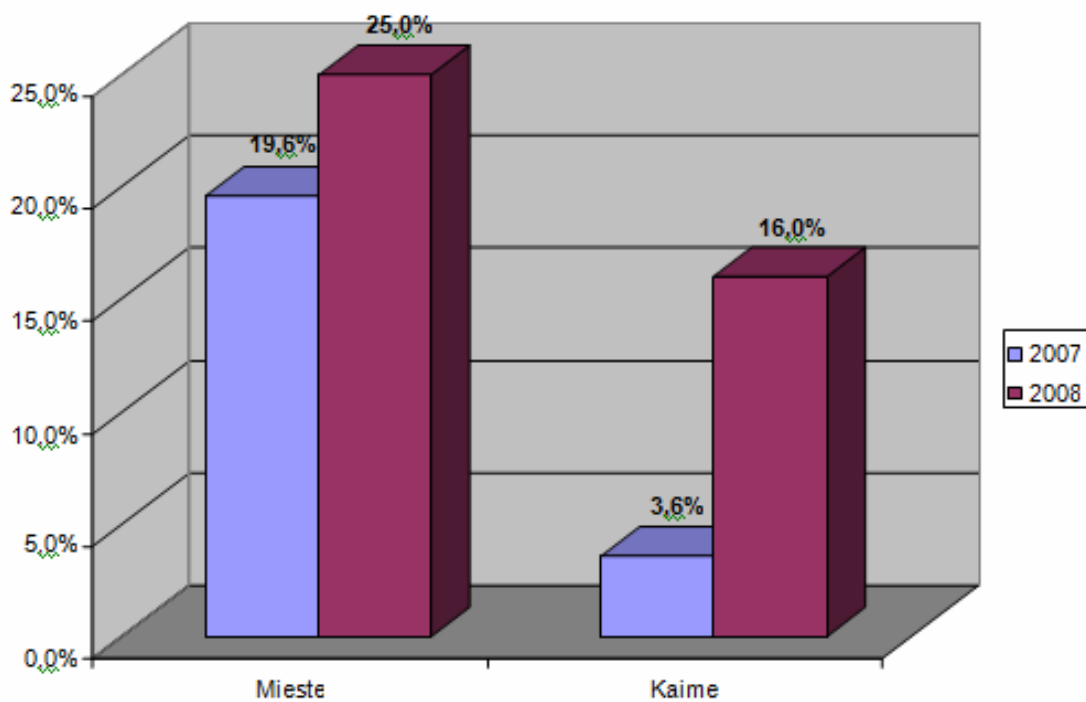
1.3. Seniūnijos teikiamų viešųjų paslaugų veiklos procesas

Informacinės visuomenės plėtra gyventojams suteikia naujas galimybes – atsiranda naujos veiklos formos, įprastinė, tradicinė veikla perkeliama į elektroninę erdvę. Pasaulyje ir mūsų šalyje vis daugiau dėmesio skiriama internetu teikiamų elektroninių paslaugų plėtrai. Tačiau, pažvelgę į paveikslėlį 6 galime įsitikinti, kad šiuo metu Lietuva atsilieka nuo daugelio Vakarų Europos šalių. Lietuvoje 68% teikiamų viešųjų paslaugų yra perkelta į elektroninę erdvę. Šis rodiklis net nesiekia Europos Sąjungos vidurkio, kuris yra 75%.



6 pav. Teikiamų viešųjų paslaugų perkėlimas į elektroninę erdvę

Naudojimasis internetu auga sparčiausiai nei naudojimasis bet kokia kita technologija. Tarp 1993 – 1997 m. prijungtų prie interneto kompiuterių skaičius augo nuo 1 milijono iki 20 milijonų, po 2000-ųjų šis skaičius viršijo 120 milijonų. Interneto naudojimas atspindi šalių ekonominio išsivystymo lygį. Lietuva taip pat sparčiai plečia savo interneto tinklus. Paveikslėlyje 7 galime palyginti 2007 ir 2008 metų rodiklius. Žmonių skaičius kaimuose, kurie naudojami internetu yra mažesnis negu mieste, tai logiška. Tačiau šis skaičius sparčiai didėja, todėl poreikis gauti viešąsias paslaugas kaimuose internetu įgauna pagreitį.



7 pav. Žmonių procentas, kurie naudojami internetu

Viešųjų paslaugų perkėlimo į elektroninę terpę branda yra skirstoma į penkis lygius:

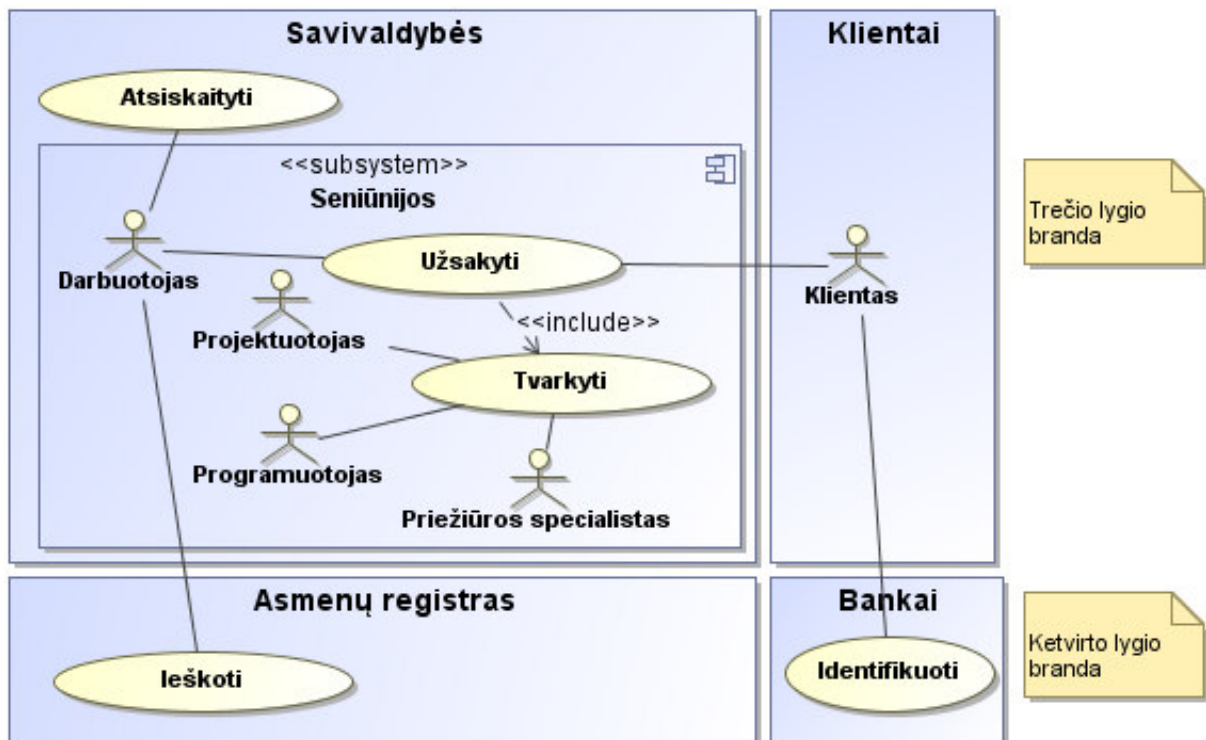
- Pirmasis lygis – informacinio pobūdžio viešosios paslaugos. Institucija internetu pateikia informaciją apie viešąją paslaugą.
- Antrasis lygis – dalinė transakcija. Institucija pateikia vartotojui savo tinklalapiuose iš dalies automatizuotas formas ir anketas, kurias užpildęs ir išspausdinęs vartotojas gali jomis naudotis (pavyzdžiui, pateikti institucijai duomenis).
- Trečiasis lygis – dalinis interaktyvumas. Vartotojo tapatybė nustatoma sistemoje. Tuo būdu vartotojas inicijuoja paslaugų teikimo procesą. Institucija elektroninės užklauskos pagrindu vykdo paslaugos teikimo procesą. Tačiau viešosios paslaugos rezultatas (pavyzdžiui, pažyma) pristatoma neelektronine forma.
- Ketvirtasis lygis – visiškas interaktyvumas, kai žingsniai nuo viešosios paslaugos inicijavimo iki rezultato pateikimo yra atliekami elektroninėje terpėje.
- Penktas lygis – proaktyvumas, kai paslaugos gavėjas, neinicijuodamas paslaugos gavimo, elektroninėje terpėje automatiškai gauna viešąją paslaugą, pagal numatytas sąlygas. [23]

1.3.1. Seniūnijos teikiamų viešųjų paslaugų organizacijos ir jos aplinkos analizė

Viešąsias paslaugas teikia LR struktūriniai valdymo padaliniai. Šiame darbe nagrinėjama būtent seniūnijų teikiamos paslaugos, kurios yra skirtos seniūnijų gyventojų, t.y. klientų (angl. customer), ir savivaldybių (angl. government) poreikiams tenkinti (G2C ir G2G). Šiuo metu seniūnijose yra naudojamos programos, skirtos apskaitai bei dokumentų spausdinimui. Tačiau tokiaime procese kliento ir savivaldybės laikas bei pinigai nėra taupomi, kadangi dokumentų užsakovas privalo atvykti į seniūniją, pateikti prašymą ir tik po tam tikro laiko gauna atitinkamus dokumentus:

1. Mirties aktus, liudijimus, įrašus – G2C (angl. Government to Customer).
2. Pažymas (skirtas seniūnijose gyvenantiems asmenims) - G2C (angl. Government to Customer).
3. Ataskaitas savivaldybėms – G2G (angl. Government to Government).

Paslaugų perkėlimo į elektroninę terpę branda šiuo metu yra trečio lygio. Šiame etape klientas parašo prašymą ir pateikia seniūnijos darbuotojams, kurie atspausdina reikalingus dokumentus. Toks sprendimas įgyvendintas beveik visose egzistuojančiose seniūnijose. Viešųjų paslaugų teikimą perkeliant į ketvirtąjį brandos lygį, atsiranda kelios papildomos organizacijos. Šiuo atveju, klientas jungiasi prie seniūnijos programinės įrangos per išorinę bankinę sistemą. Tuomet seniūnija gautus užsakymus inicijuoja. Jei reikalinga detalesnė informacija apie gyventoją, atliekama paieška gyventojų registre. Taip pat seniūnija teikia ataskaitas savivaldybei. Tiriamos organizacijos sąveika su aplinka pateikta 8 pav.



8 pav. Organizacijos aplinkos modelis

1.3.2. Seniūnijos teikiamų viešųjų paslaugų vartotojų tikslai ir problemos

Galima išskirti šiuos vartotojų tipus:

- Projektuotojai (vadybininko ir informacinių technologijų specializaciją turintys asmenys) – analizuoja ir perkelia verslo procesus į virtualią erdvę (modeliuoja). Imitacinio optimizavimo metodu analizuoja iškilusias klaidas ir jas sprendžia. Taip pat analizuoja kritines veiklos procesų vietas.
- Programuotojai (informacinių technologijų specialistai) – iš sukurtų modelių programuoja standartines internetines (paprastąsias, perkeltas į darbatalį) taikomąsias programas, jų praplėtimus.
- Priežiūros specialistai (informacinių technologijų specialistai) – prižiūri internetinių (paprastųjų, perkeltų į darbatalį) taikomųjų programų veikimą, suderinamumą su pakeitimais. Sprendžia iškilusias programų problemas.
- Darbuotojai – priima vartotojų užsakymus. Pildo duomenų bazės įrašus. (Tiesioginis sistemos vartotojas).
- Klientai
 - internetinių taikomųjų programų vartotojai.
 - paslaugų iniciatoriai.

Projektuotojų tikslas - sukurti veiklos proceso modelį atitinkantį realų pasaulį. Šie žmonės taip pat analizuoja (imitacinio optimizavimo pagalba) sukurtų procesų trūkumus ir šalina iškilusias problemas. Tokių žmonių, kurie turi vadybininko ir IT specializaciją yra pakankamai mažai, todėl pagrindinė problema - kvalifikacijos stoka.

Kai projektuotojai baigia modeliuoti veiklos procesus, programuotojai sieja ryšius su duomenų šaltiniais ir generuoja internetines taikomasias programas. Taigi, pagrindinis programuotojų tikslas – kurti internetines taikomasias programas. Šių žmonių žinios turi būti pakankamai plačios.

Priežiūros specialistų tikslas – sukurtos sistemos priežiūra. Šie asmenys analizuoja internetinių programų darbą. Pastebėti netikslumai (trūkumai) perduodami projektuotojams.

Darbuotojai – klientų aptarnavimo darbuotojai. Šie žmonės turi išmanyti naudojamą programinę įrangą, nes jie yra pagrindiniai vartotojai. Darbo efektyvumas priklauso nuo šių žmonių.

Klientai – tai fiziniai asmenys, kurie nori gauti tam tikrą paslaugą. Yra galimybė, kad vartotojas paslaugą gali užsakyti internetu.

1.4. Veiklos procesų įgyvendinimo technologijų analizė

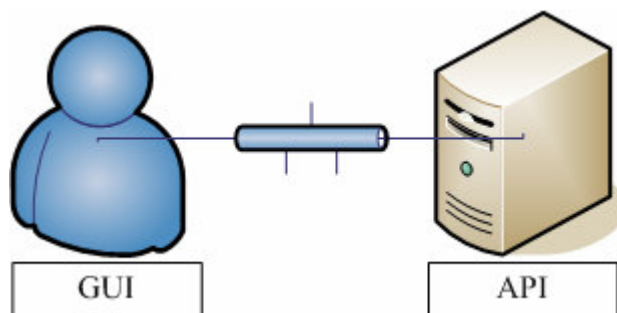
Realizuojant veiklos procesus susiduriama su technologijų atrankos problema. Šiuo metu rinkoje yra daug įvairių realizacijos metodų. Visi metodai turi savo trūkumų ir privalumų, kurie išryškėja atskirai kiekvienam veiklos procesui skirtingai. Dažniausiai technologija parenkama pagal naudojimo sritį. Įgyvendintoms realizacijoms egzistuoja tokie kokybės parametrai kaip greitis, patogumas, dinamiškumas. Technologijas galima skirstyti į tris pagrindines kategorijas:

- Dinaminis internetinis puslapis: ASP.NET, PHP, JavaScript (angl. Web Application).
- Kompiuterinė programa: C++, C#, VisualBasic, Delphi (angl. Application).
- Mišri programa (dinaminis internetinis puslapis perkeltas į darbatalį): Adobe Air, Mozilla Prism + ASP.NET, PHP, JavaScript. [8]

1.4.1. Dinaminio internetinio puslapio apžvalga

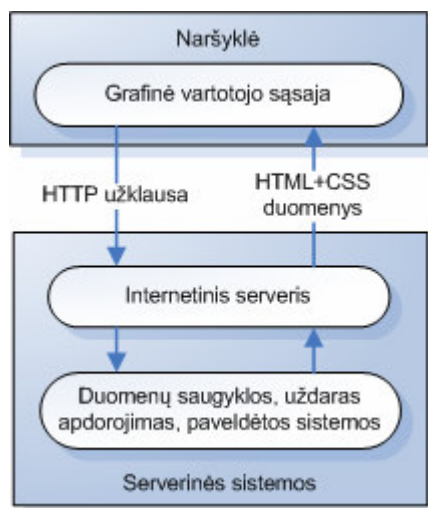
Dauguma šių dienų veiklos procesų realizuojami internetinių sprendimų pagalba, kuriant dinaminis puslapius. Technologija ypatinga tuo, kad internetinio puslapio principu įgyvendinti procesai yra prieinami iš bet kurio pasaulio taško.

Standartinėje internetinėje taikomoje programoje sistemos susideda iš dviejų pagrindinių dalių - klientinės (angl. Graphic User Interface - GUI) ir serverinės (angl. Application Programming Interface – API). 9 paveiksle pateiktas taikomosios programos kūrimo principas.



9 pav. Taikomoji programa kuriama „klientas – serveris“ principu

Kitais žodžiais tariant, ši sistema veikia dviejuose programinių sistemų lygmenyse. Pirmoji veikianti serveryje, o antroji kliento kompiuteryje. Bendravimas tarp šių sistemos dalių vyksta per SOAP protokolą (angl. Simple Object Access Protocol). Klientinę dalį sudaro vaizdavimo logika, šia dalimi naudojasi klientas. Grafinę vartotojo sąsają atvaizduoja naršyklė. Serverinė dalis susideda iš verslo logikos ir duomenų. Veikimo principas paremtas užklausomis. Vartotojas veikdamas grafinę vartotojo sąsają sudaro užklausą, kuri siunčiama į serverinę pusę HTTP protokolu. Atlikęs numatytus veiksmus, skaičiavimus, serveris grąžina rezultatus vartotojui. Tokia technologija grįstos internetinės taikomosios programos turi vieną labai svarbų trūkumą – puslapių perkrovimas. Pastaruoju metu puslapių perkrovimas jau nėra priimtinas šiuolaikinėms žiniatinklio taikomosioms programoms, nes susiformavo nauja technologija AJAX (angl. Asynchronous JavaScript and XML), kuri sugeba perkrauti tik dalį puslapio. Dinaminių internetinių puslapių sprendimų palyginimas pateiktas 10 paveiksle.



10 pav. Interneto taikomųjų programų modelis

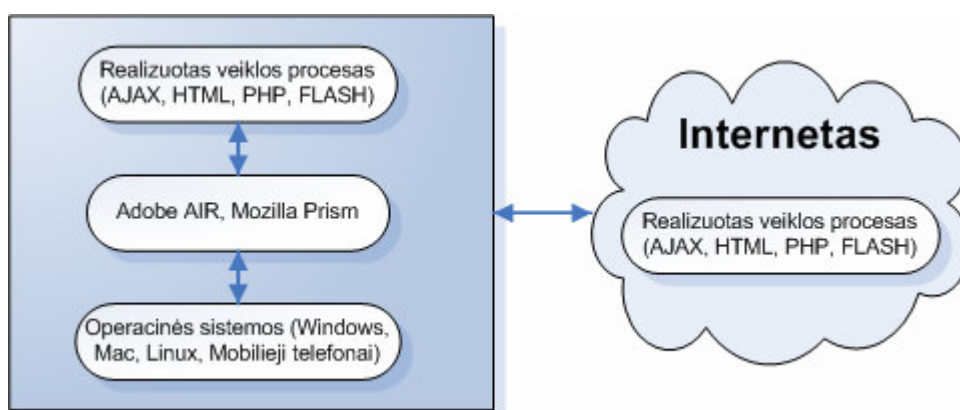
Internetinių puslapių kūrimo programavimo kalbas galima skirstyti į dvi kategorijas.

1 lentelė. Internetinių puslapių kūrimo programavimo kalbų skirstymas

Vaizdavimo technologijų grupė	Serverinės dalies technologijų grupė
HTML+CSS	ASP.NET (C#, VB)
XML+XSL	PHP
HTML+JavaScript (ExtJS)	ExtJS +
Adobe Flash	

1.4.2. Dinaminio internetinio puslapio perkeltas į darbstaalį apžvalga

Šiuo metu populiarėja nauja technologija, kai dinaminis internetinis puslapis perkeliamas į darbstaalį (Adobe AIR, Mozilla Prism). Tokios programos labiau integruotos vartotojų poreikiams tenkinti. Ši technologijų grupė veikia įvairiose operacinėse sistemose, o taip pat suteikia galimybę kurti naujoviškas internetui skirtas taikomasias programas naudojantis tokiomis technologijomis kaip Ajax, HTML, Flash, Flex, ExtJS. Teoriniai tokių programų privalumai: didesnė sparta visą sąsają apdorojat vietos kompiuteryje be siuntimo, bet kartu ir tiesioginė sąsaja su duomenimis internete. Galime laikyti, kad realizuotos programos yra mažos specializuotos naršyklės su originaliomis sąsajomis. Dinaminio internetinio puslapio perkeltas į darbstaalį sąsaja tarp technologijų pateikta 11 paveiksle.



11 pav. Dinaminio internetinio puslapio perkeltas į darbstaalį sąsaja tarp technologijų

Adobe Integrated Runtime (AIR), dar kitaip Apollo - atsirado kaip Adobe Labs bandomasis produktas, siekiantis Flah technologiją viena koja įkelti į vartotojų darbstaalius, paliekant kitą koja internete ir sukuriant vadinamąsias „žiniatinklio programas“. [9] [13]

1.4.3. Technologijų palyginimas

Technologijų palyginimas pateiktas 2 lentelėje.

2 lentelė. Įgyvendinimo technologijų palyginimas

Lyginimo kriterijus	Dinaminis internetinis puslapis	Dinaminis internetinis puslapis perkeltas į darbstaalį	Darbstaalio kompiuterinė programa
Prieinamumas internetu	Prieinamas	Prieinamas	Nėra galimybės naudotis programa internetu
Realizavimo sudėtingumas	Lengvai realizuojama internetinių programavimo kalbų pagalba	Lengvai realizuojama internetinių programavimo kalbų pagalba	Lengvai realizuojama kompiuterinių programų kūrimo programavimo kalbomis
Naujumas	Nauja	Labai nauja	Sena
Technologijų įvairumas	Labai didelė	Labai didelė	Labai didelė

Lyginimo kriterijus	Dinaminis internetinis puslapis	Dinaminis internetinis puslapis perkeltas į darbatalį	Darbastalio kompiuterinė programa
Funkcionalumas	Didelis	Didesnis, nei paprasto dinaminio internetinio puslapio. Galimybė prieiti prie kompiuteryje esančių duomenų.	Labai didelis
Integralumas	Nėra integruojama su pačiam kompiuteryje esančiais komponentais	Integruojama su internetinėmis technologijomis ir kompiuteryje esančiais komponentais.	Nėra integruojama su internetinėmis technologijomis

Lyginant šias technologijų grupes, galima teigti, kad didesnę funkcionalumą ir integralumą turi dinaminių internetinių puslapių perkeltų į darbatalį programos. Ši grupė iš dalies yra nesaugi, nes technologijos yra pakankamai naujos, todėl atsiranda informacijos trūkumas. Realizuojamas veiklos procesas reikalauja, kad būtų prieinamas internetu ir orientuotas į dirbančius žmones. Pasirenkama internetinio puslapio perkelta į darbatalį grupė, kurioje didžiausią funkcionalumą ir aprašymų įvairovę turi Adobe AIR. [17]

1.5. Esamų sprendimų analizė

Galima išskirti dvi imitaciniam optimizavimui skirtas įrankių grupes. Pirmoji kategorija būtų daugiau orientuota į imitacinį optimizavimą. Šiai grupei priklausanti programinė įranga reikalauja didelio matematinių žinių išmanymo. Antroji programų grupė orientuota daugiau į modelių sudarymą (imitaciją), tačiau palaikančių ir patį imitavimą. Optimizuojant veiklos procesus dažniausiai pastaruosiuose įrankiuose naudojami pačių kurti optimizatoriniai algoritmai, o rezultatai pateikiami ataskaitų, diagramų forma. Tokia orientacija sukurtos programos lengviau suprantamos vartotojų. Šios technologijos dažniausiai naudoja „what-if“ metodu paremtą optimizavimą, tačiau sunkėja galimybė surasti optimalius sprendinius. Suradus optimalų variantą negalima teigti, kad jis optimalus, neišanalizavus visų galimų sprendinių. Žemiau pateiktas sprendimų grupavimas pagal imitacinio optimizavimo programų tipus:

3 lentelė. Programinių sprendimų grupavimas pagal imitacinio optimizavimo tipus

Sistemos pavadinimas	Gamintojas	Internetinė svetainė
<i>Oriantacija į imitacinį optimizavimą</i>		
„AnyLogic“	xj technologies	www.xjtek.com
„Arena“	Rockwell Automation	www.rockwellautomation.com
„CSIM19“	Mesquite Software	www.mesquite.com
„Micro Saint Sharp“	Micro	www.maad.com
ProModel	ProModel	www.promodel.com
„SIMPROCESS“	CACI	www.simprocess.com
„SIMUL8“	SIMUL8 Corporation	www.simul8.com
„ExtendSim“	Imagine That	www.extendsim.com
„@Risk“	Palisade	www.palisade.com
<i>Oriantacija į modelių sudarymą</i>		
„Insight360“	Global 360	www.global360.com
„BusinessManager“	Savvion	www.savvion.com

Sistemos pavadinimas	Gamintojas	Internetinė svetainė
Oriantacija į modelių sudarymą		
„WebSphere Business Process Modeler“	IBM	www.ibm.com
„ProVision Business Process Modeling“	EduComp	www.educomp.com
„BPMlone“	Protos from Pallas Athena	www.pallas-athena.com
„Corporate Modeler Suite“	Casewise	www.casewise.com
„Business Process Management“	Singularity	www.singularity.co.uk
„Enterprise“	Metastorm	www.metastorm.com
„Enterprise Modeler“	iGrafx	www.igrafx.com

„X“ – Programoje ši savybė yra.

„-“ – Programoje šios savybės nėra

Visos 4 lentelėje pateiktos programos išsiskiria tam tikru funkcionalumu, kuris palengvina optimizacinį procesą. Pati optimizacija taip pat skiriasi, nes kiekviena programa turi savyje integruotą optimizatorių. Šis optimizatorius priklauso nuo veiklos proceso realizavimo tikslų ir principų. Pagrindinis efektyvumo kriterijus visoms šioms programoms būtų optimalių parametrų suradimas. Taip pat svarbus pats modelio sudarymas - modeliai turi atitikti BPMN notacijos standartus. Šią notaciją palaiko dauguma antrosios grupės programų. Tačiau, optimalių parametrų paieška geresnė pirmos grupės įrankiuose. [1]

4 lentelė. Programinių sprendimų lyginimas pagal optimizatorių ir modeliavimo standartą

Sistemos pavadinimas	Optimizatorius	Modeliavimo standartas
Oriantacija į imitacinį optimizavimą		
„AnyLogic“	OptQuest	-
„Arena“	OptQuest	-
„CSIM19“	OptQuest	-
„Micro Saint Charp“	OptQuest	-
ProModel	OptQuest	-
„SIMPROCESS“	OptQuest	-
„SIMUL8“	OptQuest	-
„ExtendSim“	Evolutionary Optimizer	-
„@Risk“	Evolver	-
Oriantacija į modelių sudarymą		
„Insight360“	„Built In“	BPMN
„BusinessManager“	„Built In“	BPMN
„WebSphere Business Process Modeler“	„Built In“	BPMN
„ProVision Business Process Modeling“	„Built In“	BPMN
„BPMlone“	„Built In“	BPMN
„Corporate Modeler Suite“	„Built In“	BPMN
„Business Process Management“	„Built In“	BPMN
„Enterprise“	„Built In“	BPMN
„Enterprise Modeler“	„Built In“	BPMN

„X“ – Programoje ši savybė yra.

„-“ – Programoje šios savybės nėra

Antrosios grupės programoms (orientacija į modelio sudarymą) pritaikius (sukūrus) nuolatinį imitacinio optimizavimo mechanizmą pranašumas didėja lyginant programas, kurios daugiau orientuotos į optimizavimą. Tolimesnei analizei pasirenkama antra programų grupė. Programų palyginimas, kuriuose įdiegtas nuolatinio optimizavimo mechanizmas pavaizduotas 5 lentelėje.

5 lentelė. Programinių sprendimų lyginimas pagal nuolatinio imitacinio optimizavimo mechanizmo požymį

Programinis paketas	Nuolatinis imitacinio optimizavimo mechanizmas
Global 360 „Insight360“	X
Savvion „BusinessManager“	X
IBM „WebSphere Business Process Modeler“	X
EduComp „ProVision Business Process Modeling“	X
Protos from Pallas Athena „BPMlone“	X
Casewise „Corporate Modeler Suite“	X
Singularity „Business Process Management“	X
Metastorm „Enterprise“	X
iGrafx „Enterprise Modeler“	X

„X“ – Programoje ši savybė yra.

Atlikus egzistuojančių programinių paketų analizę vertinant dešimties balų skalėje buvo sudaryta lyginimo lentelė (6 lentelė). Joje naudojami lyginimo kriterijai: sistemos integralumas – sistemos (rezultatu) integracija su kitais modeliavimo įrankiais; modelių sudarymas – modelių sudarymo patogumas, greitumas, lengvesnis supratimas; imitavimo parametrų kiekis – parametrų kiekis, pagal kuriuos atliekamas imitacinis optimizavimas (laikas, pinigai); imitavimo ataskaitų įvairovė – rezultatų pateikimo įvairovė; optimalių parametrų paieška – optimalių veiklos proceso parametrų suradimas.

6 lentelė. Programinių sistemų įvertinimų lentelė

Kriterijus Sistema	Sistemos integralumas	Modelių sudarymas	Imitavimo parametrų kiekis	Imitavimo ataskaitų įvairovė	Optimalių parametrų paieška	Vidurkis:
Global 360 „Insight360“	8	7	8	8	5	7.2
Savvion „BusinessManager“	7	9	6	7	5	6.8
IBM „WebSphere Business Process Modeler“	9	10	9	9	6	8.6
EduComp „ProVision Business Process Modeling“	8	6	7	8	5	6.8
Protos from Pallas Athena „BPMlone“	8	8	8	8	5	7.4
Casewise „Corporate Modeler Suite“	9	8	7	8	5	7.4

Kriterijus Sistema	Sistemos integralumas	Modelių sudarymas	Imitavimo parametrų kiekis	Imitavimo ataskaitų įvairovė	Optimalių parametrų paieška	Vidurkis:
Singularity „Business Process Management“	9	9	8	9	5	8.0
Metastorm „Enterprise“	10	8	9	9	6	8.4
iGrafx „Enterprise Modeler“	7	8	9	8	5	7.4

1.6. Analizės išvados

1. Atlikus verslo procesų modeliavimo kalbų ir metodų analizę nustatyta, kad verslo procesų modeliavimą, projektavimą ar rekonstrukciją naudinga išplėsti imitacinio optimizavimo etapu, kuris padėtų gauti veiklos proceso modelį, geriausiai atitinkantį verslo poreikius.

2. Verslo procesų modeliavimo įrankių analizė parodė, kad egzistuoja programiniai produktai, turintys imitacinio optimizavimo priemones. Panaudojus imitacinio optimizavimo funkcionalumą galima sukurtą veiklos procesą imituoti ir ieškoti optimalių parametrų.

3. Optimizuojant veiklos procesus imitacinio optimizavimo metodika sunku nustatyti, ar procesas yra optimalus, proceso savybės ir aplinka kinta, todėl į kompiuterizuojamą sistemą naudinga integruoti nuolatinį imitacinio optimizavimo mechanizmą.

4. Išanalizuotų nuolatinio optimizavimo įrankių analizė parodė, kad šių produktų kainos ir reikiami žmonių darbo resursai yra labai dideli. Siekiant to išvengti, yra galimybė integruoti modelių optimizatorių su pačia realizacija. Todėl nuspręsta pasinaudoti IBM programinės įrangos atskira dalimi – „IBM Business Process Modeler“.

5. Toliau numatyta sudaryti nuolatinio imitacinio optimizavimo metodiką ir pritaikyti realizuotą sistemą seniūnijos teikiamoms viešosioms paslaugoms optimizuoti.

2. Veiklos procesų nuolatinio imitacinio optimizavimo sistemos reikalavimų specifikacija ir analizė

2.1. Reikalavimų specifikacija

Išanalizavus imitacinio optimizavimo technologijas, nustatyti bendriausi tokių sistemų panaudojimo atvejai. Optimizavimo sistemas dažniausiai sudaro analitikai ir veiklos procesų kūrimo įrankiai, paremti imitacinio optimizavimo metodika. Tačiau toks imitacinio optimizavimo procesas nėra tinkamas kalbant apie optimalių parametrų paiešką. Norint užtikrinti veiklos proceso kokybę bei nenutrūkstamą veiklos procesų gerinimą sudaroma nuolatinio imitacinio optimizavimo metodika.

Kuriant nuolatinį imitacinį optimizavimo mechanizmą veiklos procesų kūrimo įrankis yra tik dalelė visos sistemos. Bendrą optimizavimo visumą sudaro komponentinės dalys, susietos tarpusavio sąryšiais. Kiekvienas komponentas turi savyje tam tikras veiklos taisykles, kurios atlieka specifines užduotis. Užtikrinant pastovų veiklos procesų gerinimą reikalingas bendravimas tarp veiklos procesų optimizavimo ir realizacinės sistemos. Tarp šių komponentų įsiterpia integravimas bei stebėjimo sąsajos. Bendra sistema susideda iš keturių pagrindinių komponentinių dalių:

- veiklos procesų kūrimo įrankio, imitacinio optimizatoriaus;
- integratoriaus;
- stebėjimo sistemos;
- realizacinės sistemos.

Integratorius skirtas sujungti dvi atskiras sistemos dalis: veiklos procesų modeliavimo įrankį ir stebėjimo sistemą, sukuriant duomenų struktūras. Šis komponentas turi vykdyti tokius veiksmus:

- importuoti duomenis iš veiklos procesų kūrimo įrankio į stebėjimo sistemą;
- eksportuoti duomenis iš stebėjimo sistemos į veiklos procesų kūrimo įrankį;
- sukurti duomenų struktūras.

Stebėjimo sistema sujungia realizaciją ir integruoja veiklos procesus. Šis komponentas turi vykdyti tokius veiksmus:

- vaizduoti veiklos procesus;
- priskirti realizacinės sistemos daviklius;
- keisti realizacinės sistemos daviklius;
- vaizduoti optimizacinius parametrus;
- keisti optimizacinius parametrus;
- vaizduoti veiklos procesų būseną.

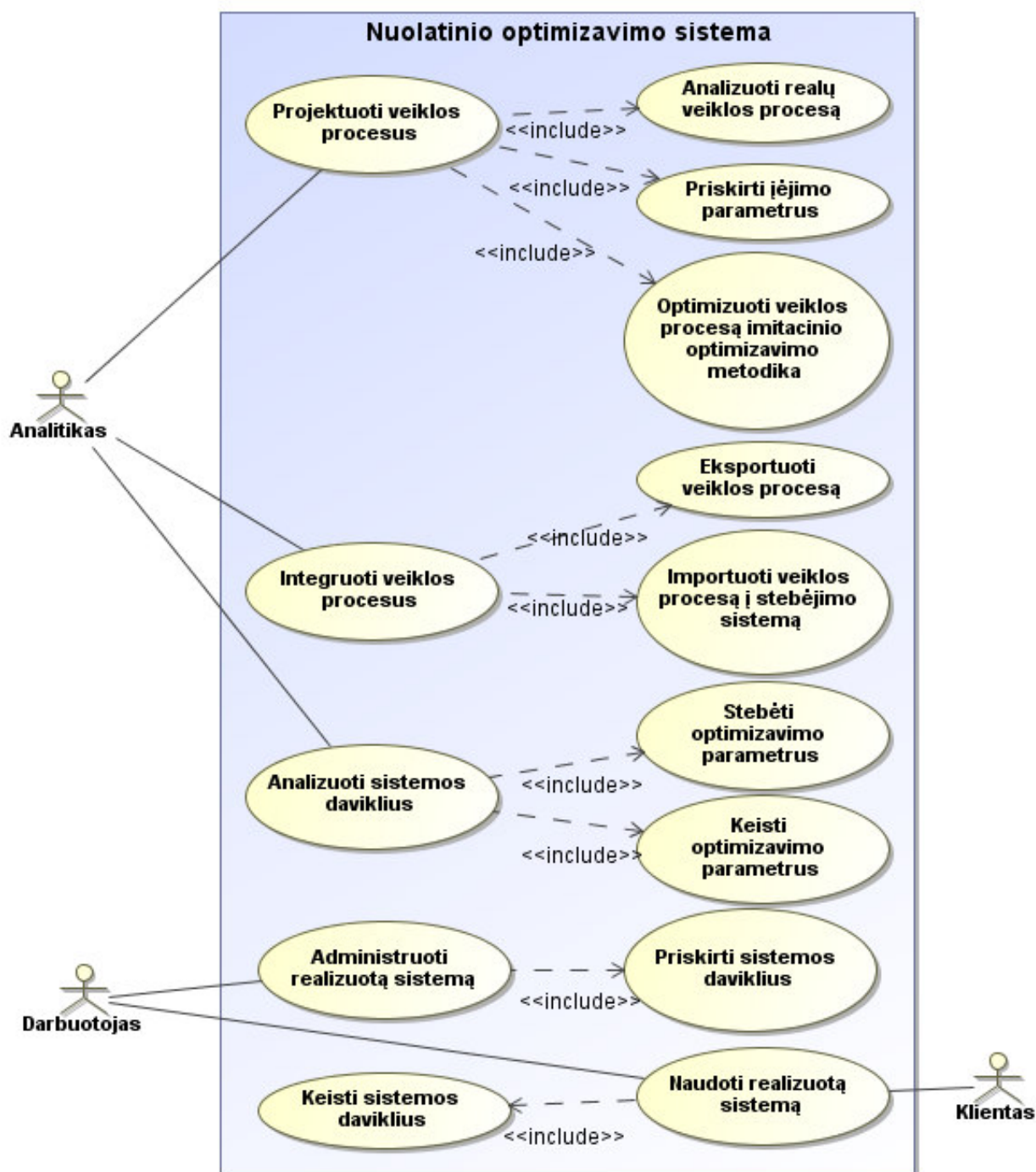
Realizacinė sistemos dalis fiksuoja realius kliento veiksmus, kurie yra apibrėžti individualiai kiekviename veiklos procese ir susieti su stebėjimo sąsaja. Fiksuojami pakitimai vadinami davikliais.

Bendroje sistemoje ši komponentinė dalis atlieka pagrindinę funkciją - keičia daviklių būsenas. Realizuota sistema taip pat turi integruotą veiklos procesų funkcionalumą.

Analitikas - pagrindinis nuolatinio imitacinio optimizavimo sistemos vykdymo asmuo. Veiklos procesų kūrimo priemonėmis jis sukuria realų modelį, kurį optimizuoja ir padaro artimą optimaliam. Naudojant integratorių, sukurtas veiklos procesas perkeliamas į naują informacinę sistemą. Analitiko funkcionalumo reikalavimai:

- projektuoti bei optimizuoti veiklos procesus;
- integruoti veiklos procesus naudojant integratorių;
- stebėjimo sistemos pagalba analizuoti sistemos daviklius;
- keisti stebėjimo sistemos daviklius.

Darbuotojas ir klientas yra susiję tik su realizuota sistema. Pagrindinė funkcija, kurią atlieka darbuotojas nuolatinio optimizavimo sistemoje, yra daviklių priskyrimas (sistemos administravimas). Tiek darbuotojas, tiek ir klientas gali naudotis šiuo komponentu. Naudojimo paskirtis priklauso nuo veiklos proceso. Nuolatinio optimizavimo sistemos panaudojimo atvejai ir „aktoriai“ pavaizduoti 12 paveiksle.



12 pav. Panaudojimo atvejų modelis

Kompiuterizuojamų panaudojimo atvejų specifikacijos pateikiamos lentelėse.

7 lentelė. Panaudojimo atvejo „Projektuoti veiklos procesus“ specifikacija

PA „Projektuoti veiklos procesus“	
Tikslas. Sukurti veiklos procesą bei surasti optimalius imitacinio optimizavimo parametrus	
Aprašymas. Šis PA apima veiklos procesų kūrimą ir optimizavimą	
Prieš sąlyga	Vartotojas turi įdiegtą veiklos procesų kūrimo įrankį Įrankis turi galimybę optimizuoti veiklos procesus imitacinio optimizavimo metodika
Sužadinimo sąlyga	Vartotojas nori surasti optimalius veiklos proceso parametrus

Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	Modeliuoti veiklos procesą Priskirti jėjimo parametrus Optimizuoti veiklos procesą imitacinio optimizavimo metodika
	Specializuoja PA	
Pagrindinis įvykių srautas:		Sistemos reakcija ir sprendimai
1. Analitikas nori kompiuterizuoti realų (egzistuojantį arba naujai kuriantį) veiklos procesą.		Sistema įvykdo PA „Analizuoti realų veiklos procesą“ ir pereina į antrą žingsnį
2. Analitikas nori priskirti jėjimo parametrus		Sistema įvykdo PA „Priskirti jėjimo parametrus“ ir pereina į trečią žingsnį
3. Analitikas nori optimizuoti veiklos procesą, naudojant imitacinio optimizavimo metodiką („what-if“)		Sistema įvykdo PA „Optimizuoti veiklos procesą imitacinio optimizavimo metodika“ ir pereina į ketvirtą žingsnį
4. Analitikas nori išsaugoti veiklos procesą		Sistema išsaugo veiklos procesą ir baigia PA
Po sąlyga		Gaunamas veiklos procesas artimas optimaliam
Alternatyvūs scenarijai		
Pastabos		

8 lentelė. Panaudojimo atvejo „Integruoti veiklos procesus“ specifikacija

PA „Integruoti veiklos procesus“		
Tikslas. Integruoti sukurtą veiklos procesą į specialų duomenų modelį		
Aprašymas. Šis PA integruoja veiklos procesus su stebėjimo sistema		
Prieš sąlyga		Vartotojas turi būti sukūręs veiklos procesą Veiklos procesas turi būti artimas optimaliam Daviklių būsenos rodo didelius nuokrypius
Sužadinimo sąlyga		Vartotojas nori stebėti veiklos procesų srautus
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	Eksportuoti veiklos procesą Importuoti veiklos procesą į stebėjimo sistemą Eksportuoti integruotus duomenis į veiklos procesų kūrimo įrankį
	Specializuoja PA	
Pagrindinis įvykių srautas:		Sistemos reakcija ir sprendimai
1. Analitikas eksportuoja sukurtą veiklos procesą į failų sistemą		Sistema įvykdo PA „Eksportuoti veiklos procesą“ ir pereina į antrą žingsnį
2. Analitikas importuoja veiklos procesą į stebėjimo sistemą		Sistema įvykdo PA „Importuoti veiklos procesą į stebėjimo sistemą“ ir baigia PA
Po sąlyga		Gaunama veiklos procesų integruota duomenų struktūra
Alternatyvūs scenarijai		
2a. Analitikas eksportuoja veiklos procesą į veiklos procesų kūrimo įrankį		Sistema įvykdo PA „Eksportuoti integruotus duomenis į veiklos procesų kūrimo įrankį“ ir baigia PA
Pastabos		

9 lentelė. Panaudojimo atvejo „Analizuoti sistemos daviklius“ specifikacija

PA „Analizuoti sistemos daviklius“		
Tikslas. Stebėti optimizavimo parametrus ir rodyti pakitimus		
Aprašymas. Šis PA fiksuoja stebėjimo sistemos daviklius		
Prieš sąlyga		Sukurtam veiklos procesui integratoriaus pagalba turi būti sudaryta duomenų struktūra
Sužadinimo sąlyga		Vartotojas sužadina veiklos procesų optimizavimą
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	Stebėti optimizavimo parametrus Keisti optimizavimo parametrus
	Specializuoja PA	
Pagrindinis įvykių srautas:		Sistemos reakcija ir sprendimai
1. Analitikas stebi fiksuojamus daviklius		Sistema įvykdo PA „Stebėti optimizavimo parametrus“
2. Analitikas pamato netinkamus daviklių parodymus ir keičia optimizavimo parametrus		Sistema įvykdo PA „Keisti optimizavimo parametrus“ ir baigia PA
Po sąlyga		Gaunasi pakitęs veiklos procesų integruotas duomenų modelis, kuris eksportuojamas į veiklos procesų kūrimo įrankį.
Alternatyvūs scenarijai		
Pastabos		

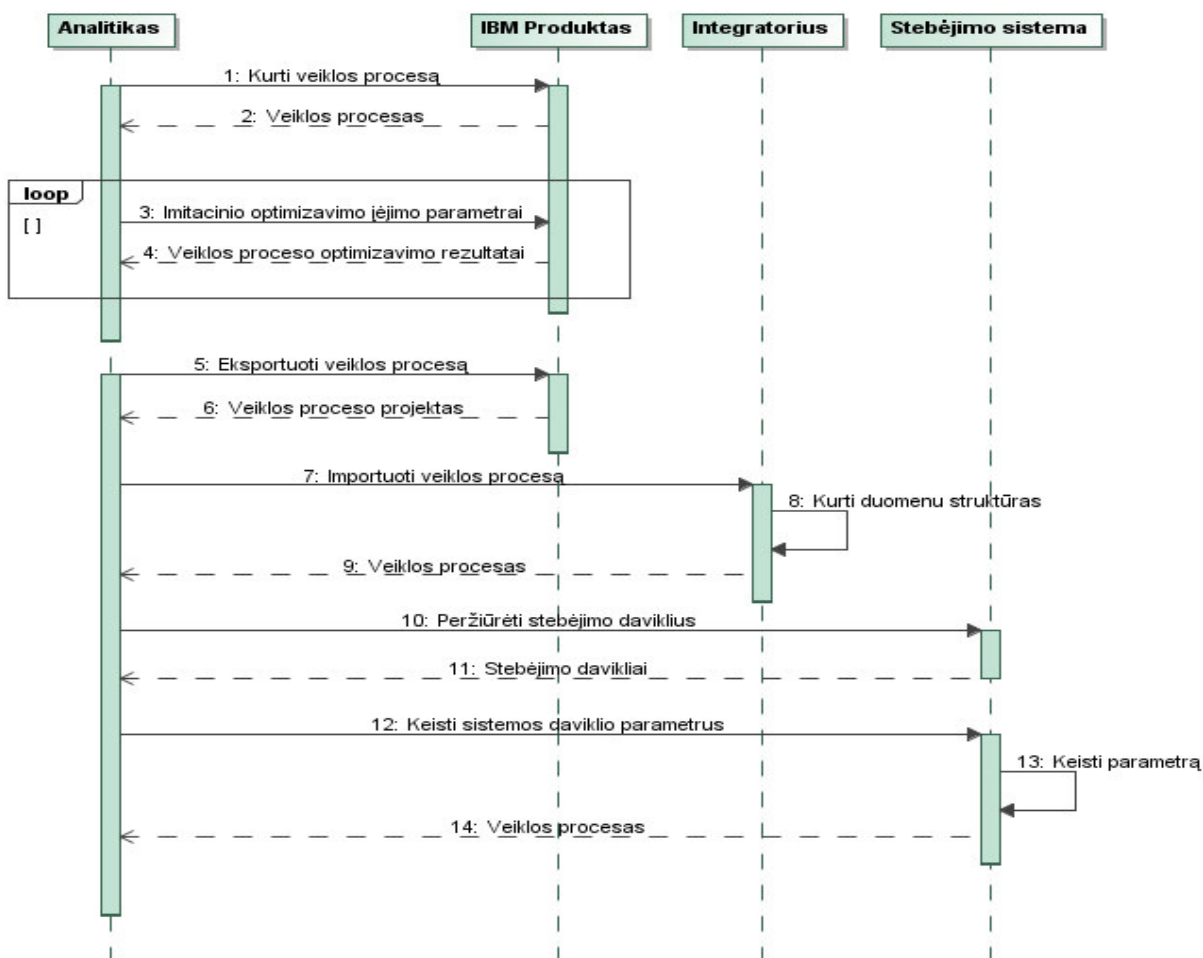
10 lentelė. Panaudojimo atvejo „Administruoti realizuotą sistemą“ specifikacija

PA „Administruoti realizuotą sistemą“		
Tikslas. Priskirti sistemos daviklius		
Aprašymas. Šis PA nurodo sąryšius tarp stebėjimo sistemos ir realizacinės sistemos daviklių		
Prieš sąlyga		Sukurtam veiklos procesui integratoriaus pagalba turi būti sudaryta duomenų struktūra
Sužadinimo sąlyga		Vartotojas sužadina veiklos procesų priskyrimą prie daviklių
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	Priskirti sistemos daviklius
	Specializuoja PA	
Pagrindinis įvykių srautas:		Sistemos reakcija ir sprendimai
1. Darbuotojas susieja realizacinės sistemos daviklius		Sistema įvykdo PA „Priskirti sistemos daviklius“ ir baigia PA
Po sąlyga		Gaunami sąryšiai tarp stebėjimo sistemos ir realizacijos
Alternatyvūs scenarijai		
2. Darbuotojas naudojami realizacine sistema		Sistema įvykdo PA „Naudoti realizuotą sistemą“ ir baigia darbą
Pastabos		

11 lentelė. Panaudojimo atvejo „Naudoti realizuotą sistemą“ specifikacija

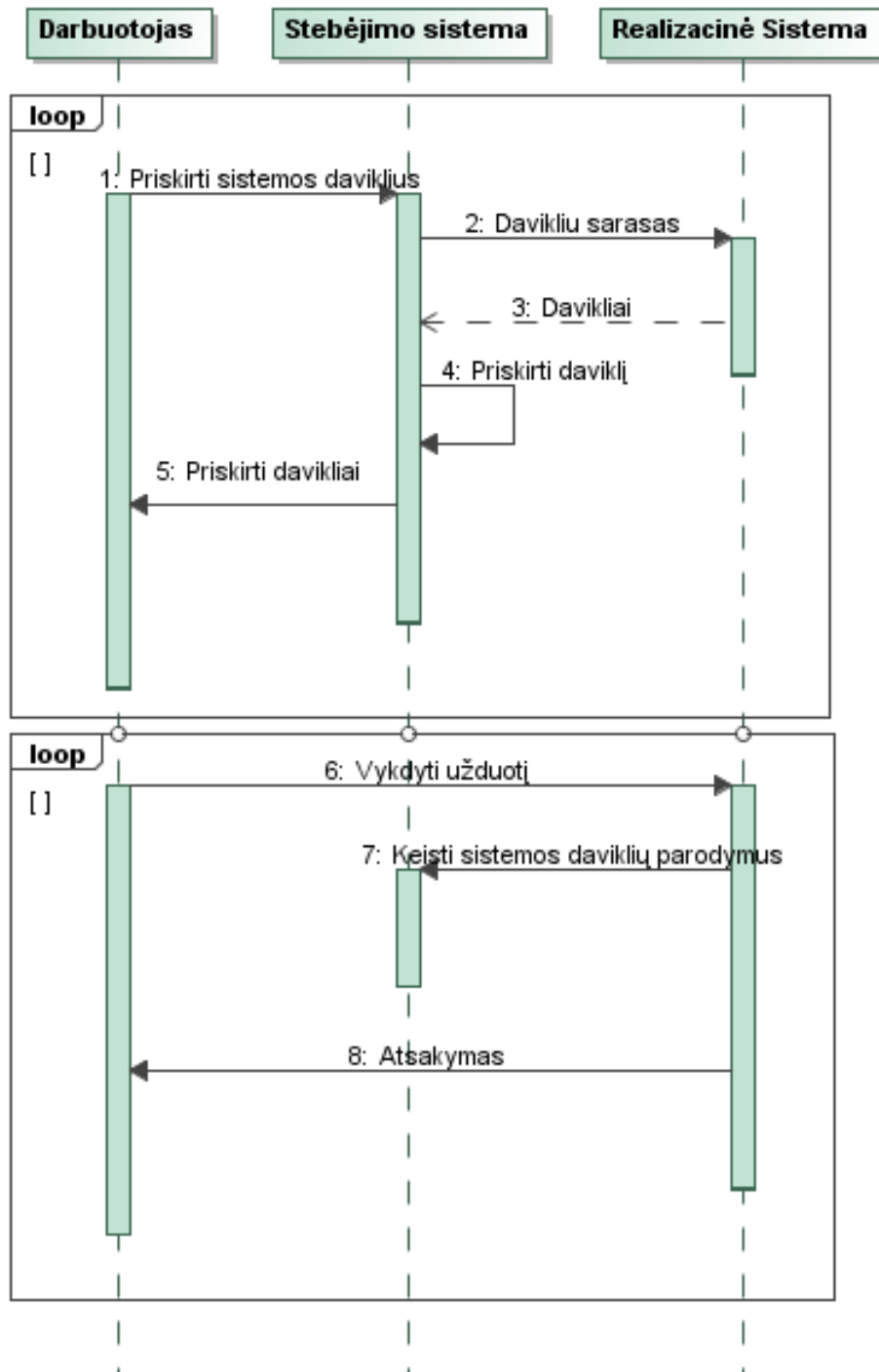
PA „Naudoti realizuotą sistemą“		
Tikslas. Naudoti realizacinę sistemą		
Aprašymas. Keisti realizacinės sistemos daviklius		
Prieš sąlyga	Veiklos procesas turi būti realizuotas	
Sužadinimo sąlyga	Darbuotojas arba klientas sužadina paslaugą	
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	Keisti sistemos daviklius
	Specializuoja PA	
Pagrindinis įvykių srautas:		Sistemos reakcija ir sprendimai
1. Klientas naudojami realizacine sistema		Sistema įvykdo PA „Keisti sistemos daviklius“
Po sąlyga		Užfiksuojami realizacinės sistemos davikliai
Alternatyvūs scenarijai		
Pastabos		

Analitikas ir pagalbini darbuotojas yra pagrindiniai nuolatinio imitacinio optimizavimo dalyviai. Visus veiklos procesų kūrimo ir optimizavimo žingsnius vykdo analitikas. Šis asmuo naudojami IBM programine įranga (modelio kūrimui ir optimizavimui), integratoriumi (integravimui į atskirą duomenų modelį) ir stebėjimo sistema (realių parametrų peržiūrai ir keitimui).



13 pav. Analitiko ir nuolatinio optimizavimo sistemos sekų diagrama

Darbuotojas atlieka dvi pagrindines užduotis: priskirti sistemos daviklius ir naudoti realizuotą sistemą. Šis asmuo naudojami stebėjimo ir realizacine sistemomis.

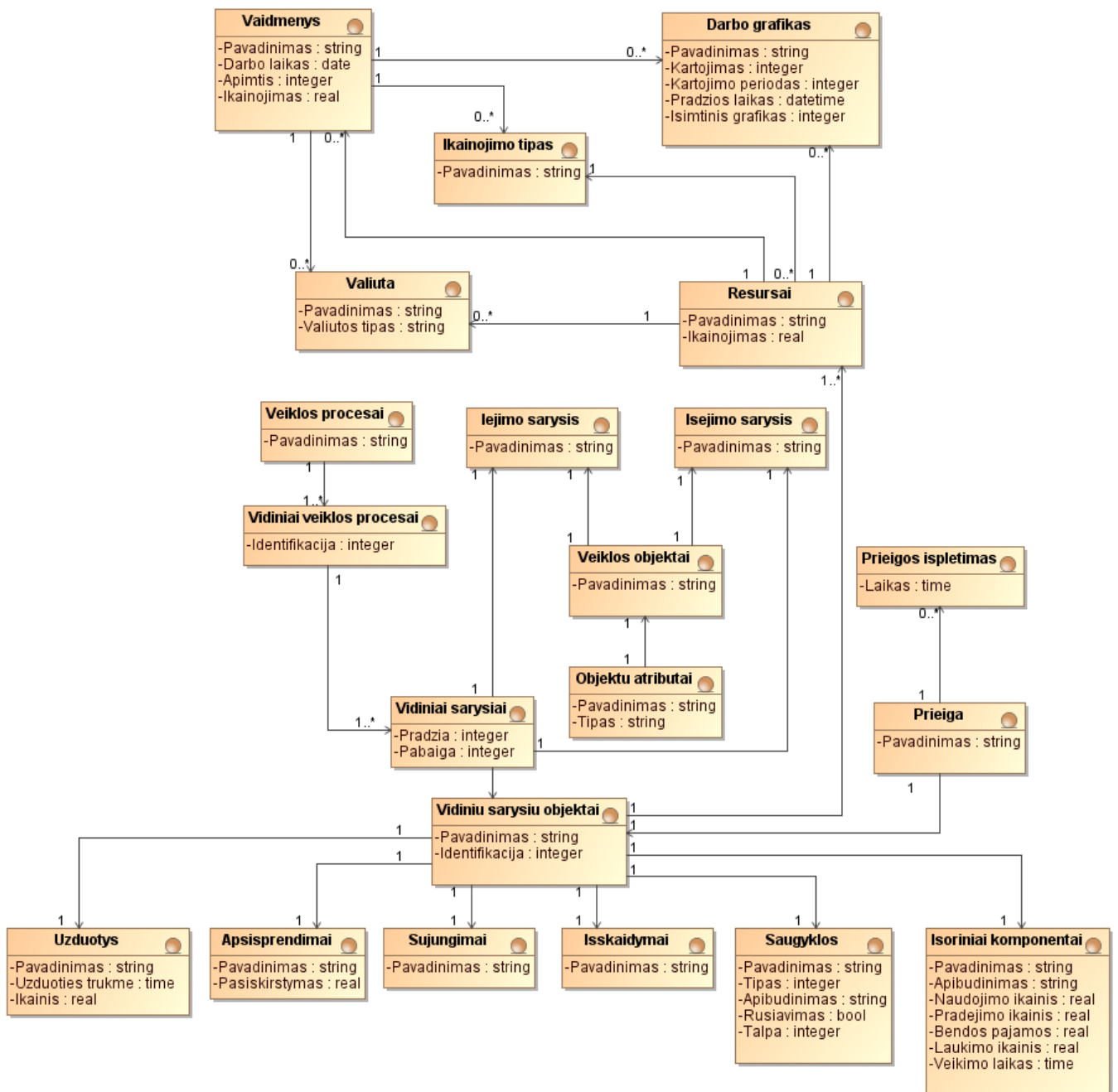


14 pav. Darbuotojo ir nuolatinio optimizavimo sistemos sekų diagrama

2.2. Nuolatinio imitacinio optimizavimo sistemos esybių diagrama

Apibrėžus reikalavimus buvo sudaryta nuolatinio imitacinio optimizavimo sistemos esybių diagrama. Ši diagrama susideda iš dviejų pagrindinių esybių: veiklos procesų ir prieigos. Pirmoji esybė skirstoma į tris detalesnius resursų, objektų ir procesų modelius.

Nuolatinio imitacinio optimizavimo sistemos esybių diagrama pavaizduota 15 paveiksle.



15 pav. Nuolatinio imitacinio optimizavimo sistemos esybių diagrama

2.3. Reikalavimų analizės apibendrinimas

Veiklos procesų nuolatinio imitacinio optimizavimo sistemos reikalavimų specifikacijos ir analizės metu buvo apibrėžti sistemos vartotojai bei jų reikalavimai. Specifikuotos komponentinės sistemos elementai: IBM modeliavimo paketas, integratorius, stebėjimo sistema ir realizacinė sistema. Taip pat apibrėžti šių komponentų sąryšiai su sistemos vartotojais. Toliau bus pateiktas sistemos projektas, kuriame visi apibrėžti elementai ir reikalaujas funkcionalumas yra suprojektuoti ir sukurta nuolatinio imitacinio optimizavimo metodika paremta sistema.

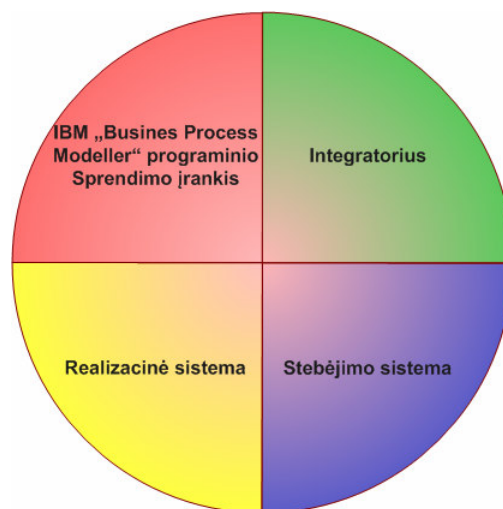
3. Nuolatinio imitacinio optimizavimo sistemos projektas

3.1. Veiklos procesų nuolatinio imitacinio optimizavimo modelis ir sistema

Optimizavimo sistemas dažniausiai sudaro analitikai ir veiklos procesų kūrimo įrankiai paremti imitacinio optimizavimo metodika. Tačiau, toks imitacinio optimizavimo procesas nėra tinkamas kalbant apie optimalių parametru paiešką. Norint užtikrinti veiklos proceso kokybę bei nenutrūkstamą veiklos procesų gerinimą sudaroma nuolatinio imitacinio optimizavimo metodika.

Kuriant nuolatinį imitacinį optimizavimo mechanizmą veiklos procesų kūrimo įrankis yra tik dalelė visos sistemos. Bendrą optimizavimo visumą sudaro komponentinės dalys, susietos sąryšiais. Kiekvienas komponentas turi savyje tam tikras veiklos taisykles, kurios atlieka specifines užduotis. Užtikrinant pastovų veiklos procesų gerinimą reikalingas bendravimas tarp veiklos procesų optimizavimo ir realizacinės sistemos. Tarp šių komponentų įsiterpia integravimas bei stebėjimo sąsajos. Bendra sistema susideda iš keturių pagrindinių komponentinių dalių:

- veiklos procesų kūrimo įrankio, imitacinio optimizatoriaus;
- integratoriaus skirtas apjungti dvi atskiras sistemas dalis (veiklos procesų kūrimo įrankį ir stebėjimo sistemą), sukuriant duomenų struktūras;
- stebėjimo sistemos pagalba yra sujungiama realizacija ir integruoti veiklos procesai;
- realizacinės sistemos dalis fiksuoja realius kliento veiksmus, kurie yra apibrėžti individualiai kiekviename veiklos procese ir susieti su stebėjimo sąsaja. Fiksuojami pakitimai vadinami davikliais. Bendroje sistemoje ši komponentinė dalis pagrinde keičia daviklių būsenas. Realizuota sistema taip pat turi integruotą veiklos procesų funkcionalumą.



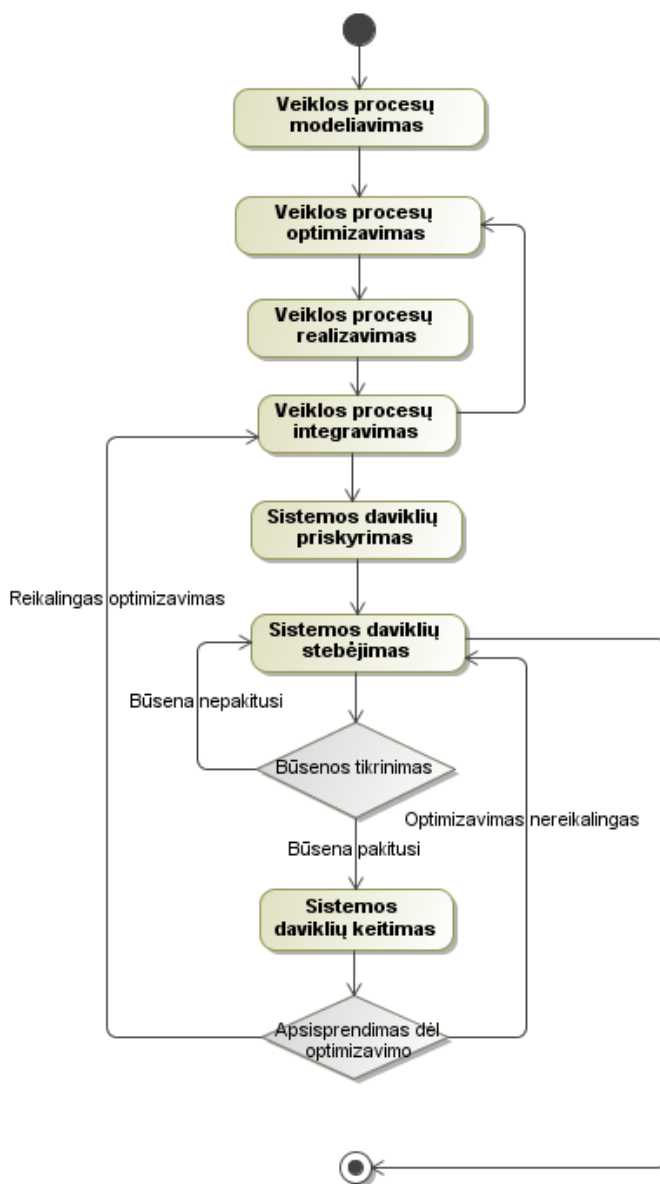
16 pav. Nuolatinio imitacinio optimizavimo sistemos komponentai

Nuolatinio imitacinio optimizavimo sistemos procesas pradedamas veiklos procesų modeliavimu bei optimizavimu – optimalių parametru paieška. Šie žingsniai yra svarbiausiai lyginant tolimesnius

etapus. Sistemos realizacija vykdoma tuomet, kai modeliuojama sistema atitinka visus keliamus reikalavimus ir įėjimo, išėjimo veiklos procesų parametrus. Prieš pradėdant realizuoti sistemą, apibrėžiami sistemos daviklių kiekybinės ir kokybinės charakteristikos – KPI (angl. Key Performance Indicator). Šios charakteristikos turėtų būti fiksuojamos vykdant realizacinę veiklos logiką.

Sukurta realizacija nėra susieta su IBM „Business Process Modeler“ programiniu paketu. Norint užtikrinti nuolatinį imitacinį optimizavimą naudojant integratoriaus veiklos logiką, sukuriama atskira duomenų struktūra. Ši struktūra susieja IBM programinį paketą su realizuota sistema. Integravimo galimybė dvipusė, todėl integratorius turi grįžtamąjį ryšį. Ši savybė užtikrina pastovų IBM „Business Process Modeler“ įrankio galimybių panaudojimą, šiuo atveju imitacinį optimizavimą.

Stebėjimo procesas prasideda susiejant realizuotą ir stebėjimo sistemas. Sistemos daviklių priskyrimas vyksta stebėjimo sistemoje. Šioje sistemoje taip pat peržiūrimos visos veiklos procesų būsenos, jų pakeitimai. Yra galimybė keisti sistemos charakteristikų parametrus.



17 pav. Nuolatinio imitacinio optimizavimo sistemos procesas

3.2. Komponentų modeliai

3.2.1. Loginė sistemos architektūra

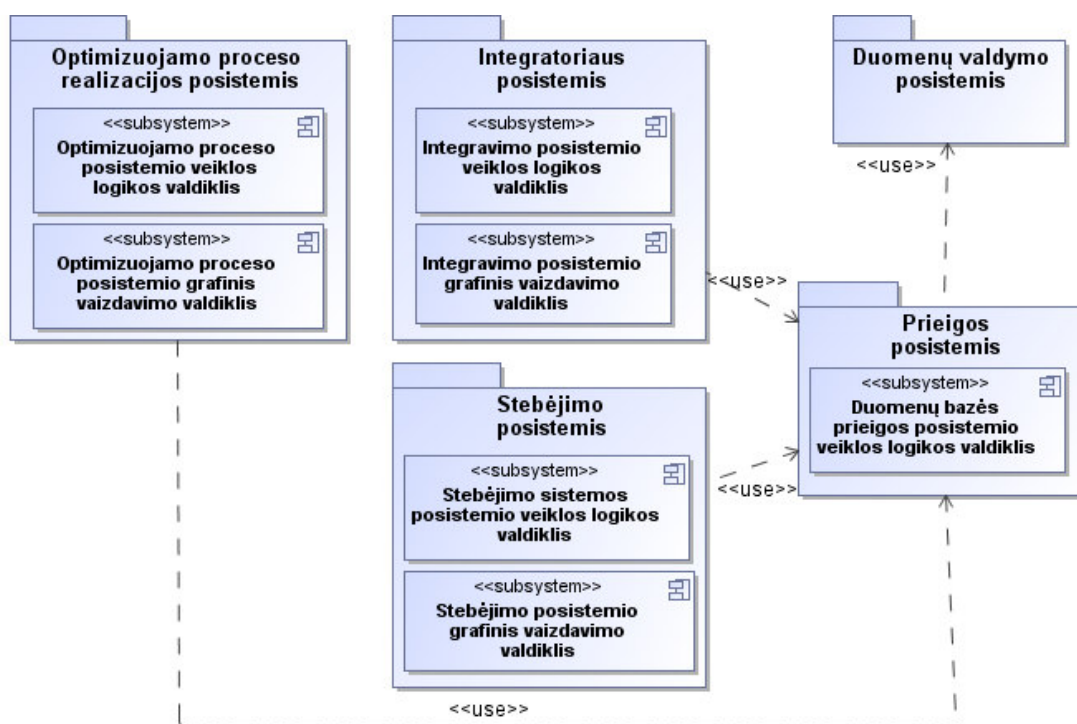
Nuolatinio imitacinio optimizavimo projektas suskirstytas į penkias pagrindines posistemas. Dauguma šių posistemų turi veiklos logikos ir grafinius vaizdavimo valdiklius. Išskiriami šie veiklos logikos valdikliai:

- Integravimo posistemio veiklos logikos valdiklis. Ši dalis atsakinga už duomenų integravimą į atskirą duomenų modelį. XML (angl. Extensible Markup Language) transformacija naudojant XSL (angl. Extensible Stylesheet language) technologiją.
- Stebėjimo sistemos posistemio veiklos logikos valdiklis. Ši dalis atsakinga už stebėjimo logikos vykdymą, valdiklių priskyrimą prie stebėjimo būsenų ir parametrų keitimą.
- Duomenų bazės posistemio veiklos logikos valdiklis. Ši dalis atsakinga už duomenų bazės valdymo funkcijas naudojamas integravimo, stebėjimo ir realizacinėje dalyje.
- Optimizuojamo proceso realizacijos posistemio veiklos logikos valdiklis. Ši dalis atsakinga už optimizuojamo proceso veiklos logiką.

Galima išskirti tokius grafinius vaizdavimo valdikius:

- Integravimo posistemio grafinis vaizdavimo valdiklis.
- Stebėjimo posistemio grafinis vaizdavimo valdiklis.
- Optimizuojamo proceso realizacijos posistemio grafinis vaizdavimo valdiklis.

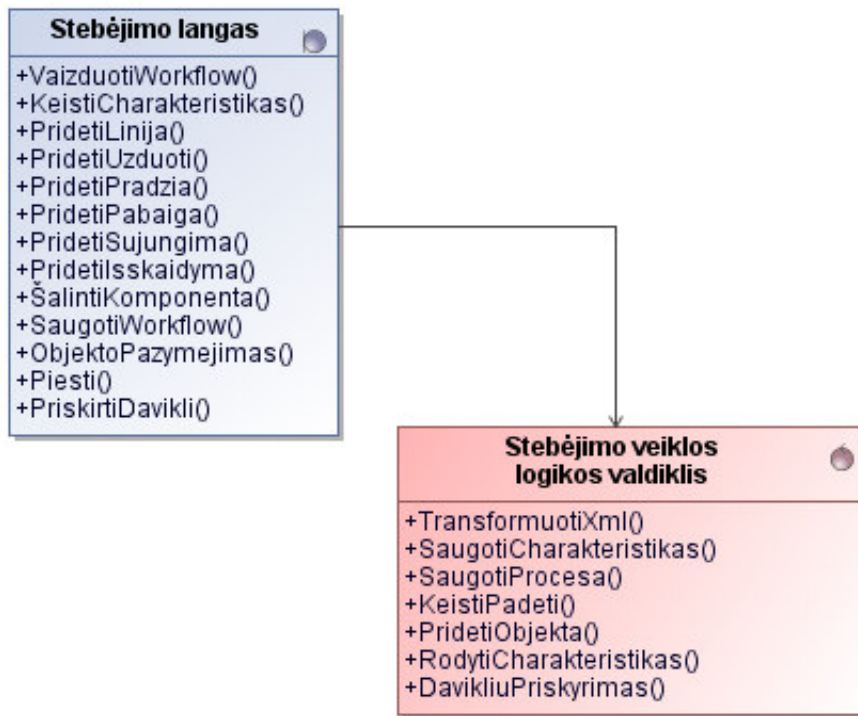
Loginė sistemos architektūra pateikta 18 paveiksle.



18 pav. Sistemos loginė architektūra

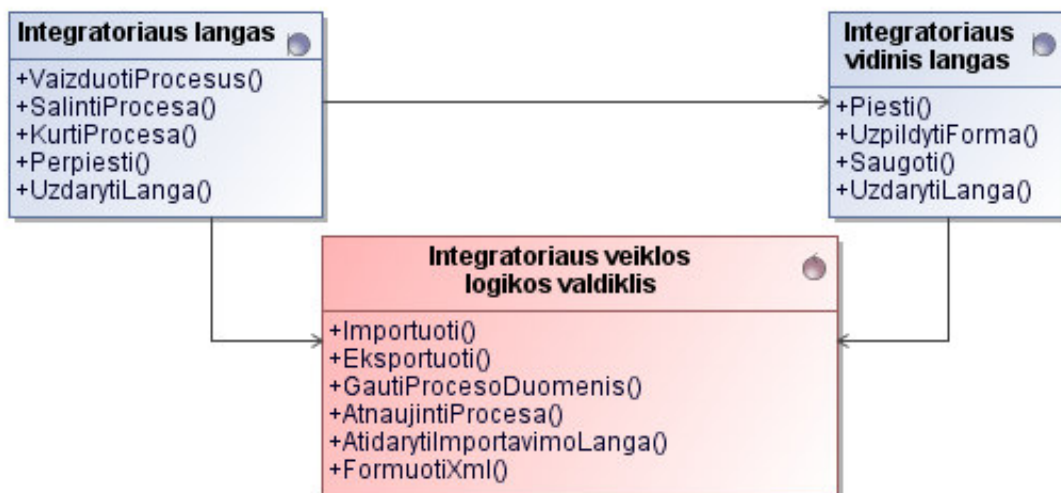
3.2.2. Vartotojo paslaugos

Nuolatinio imitacinio optimizavimo sistema turi du grafinius vaizdavimo valdiklius: integratoriaus posistemio langas ir stebėjimo posistemio. Šie grafiniai vaizdavimo valdikliai atvaizduoja daugelį kitų vidinių langų ar formų. Žemiau pateikta stebėjimo posistemio vartotojo paslaugų klasių diagrama. Ši valdiklį sudaro vienintelis stebėjimo langas, kuris atvaizduoja sukurtą integruotą veiklos procesą į atskirą duomenų struktūrą.



19 pav. Stebėjimo posistemio vartotojo paslaugų klasių diagrama

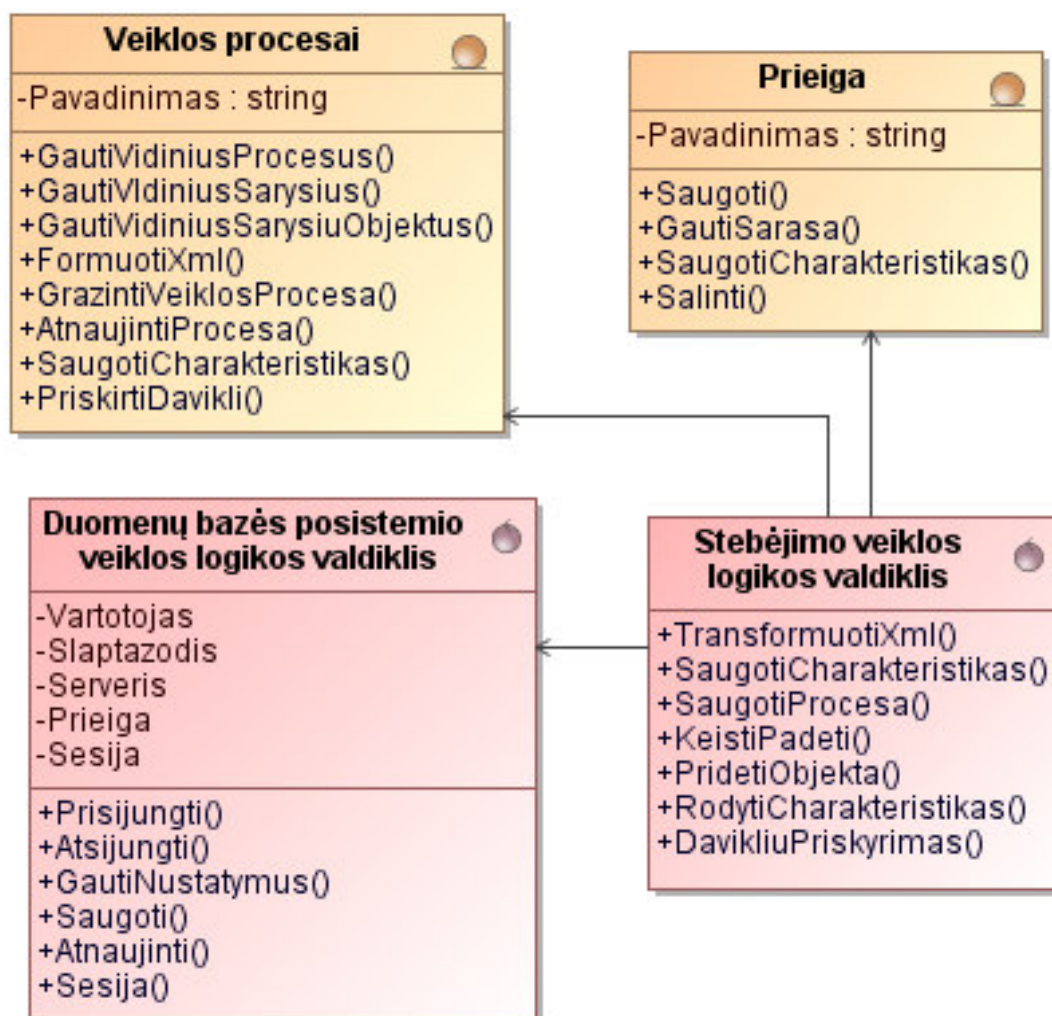
Integratoriaus posistemio grafinį vaizdavimo valdiklį sudaro taip pat vienintelis langas, tačiau integravimo procesų kūrimui naudojama atskira langų kūrimo biblioteka. 20 paveikslėlyje pateiktas integratoriaus posistemio vartotojo paslaugų klasių diagrama.



20 pav. Integratoriaus posistemio vartotojo paslaugų klasių diagrama

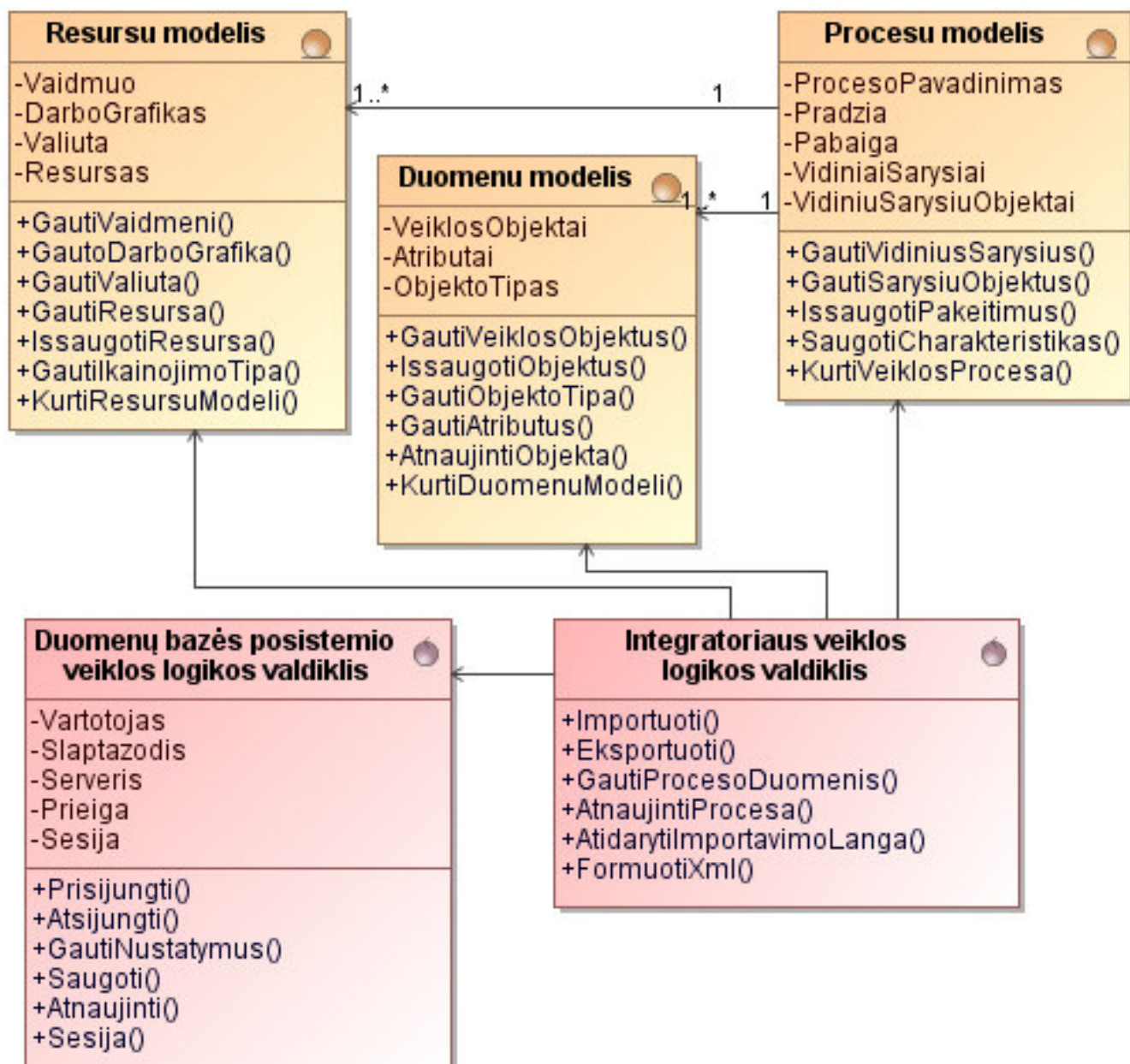
3.2.3. Veiklos paslaugos

Nuolatinio imitacinio optimizavimo sistema turi taip pat du veiklos logikos valdiklius: integratoriaus posistemio ir stebėjimo posistemio. Šie veiklos logikos valdikliai išplečiami duomenų bazės posistemiui, kuris yra atsakingas už duomenų bazės modelį. Žemiau pateikta stebėjimo sistemos veiklos paslaugų klasių diagrama.



21 pav. Stebėjimo sistemos posistemio veiklos paslaugų klasių diagrama

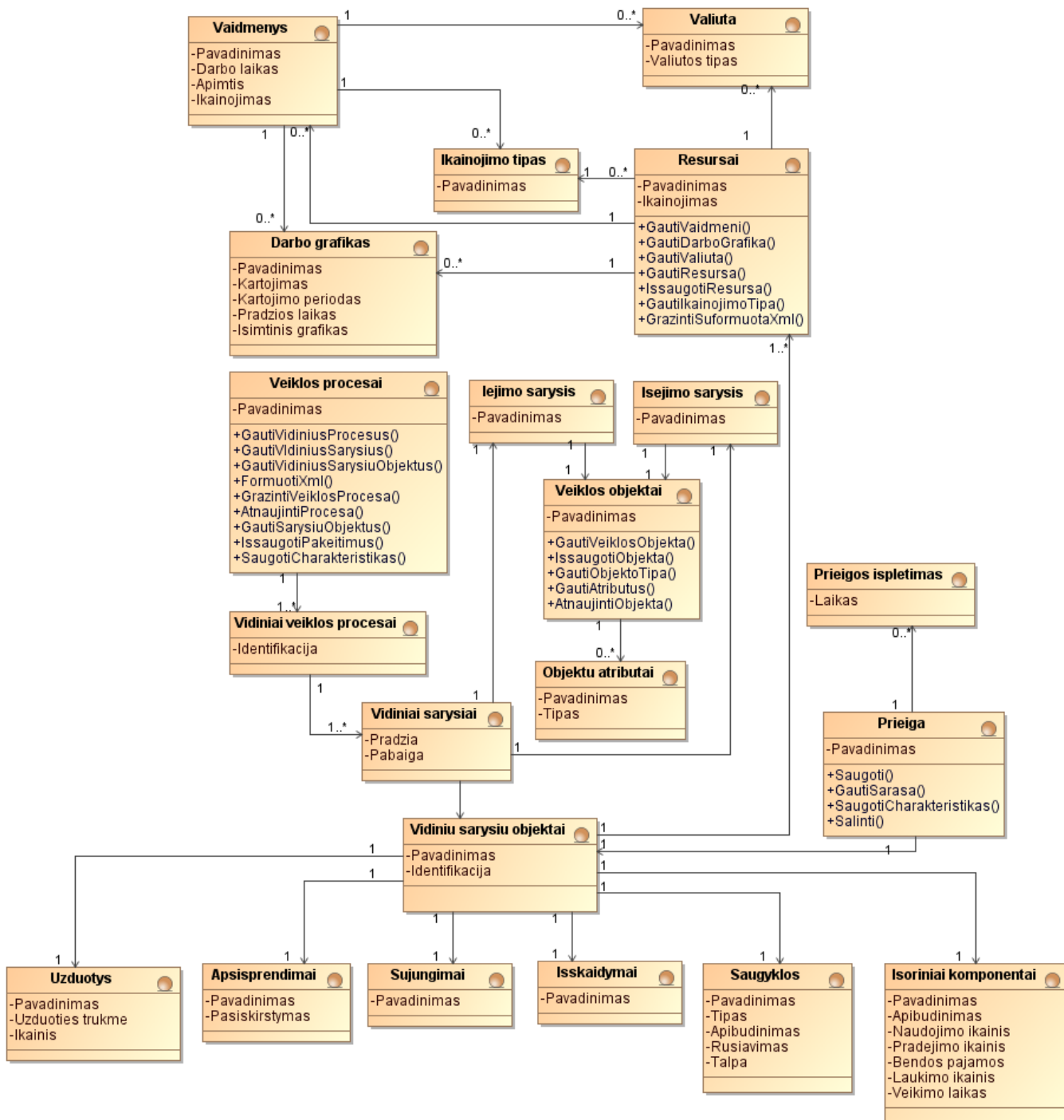
Integratoriaus posistemio veiklos logikos valdiklis taip pat išplečiamas duomenų bazės posistemiui. Integratoriaus posistemis naudoja tris pagrindines esybes: resursų, duomenų ir procesų. 22 paveikslėlyje pateiktas integratoriaus posistemio veiklos paslaugų klasių diagrama.



22 pav. Integroriaus posistemio veiklos paslaugų klasių diagrama

3.3. Esybių klasių diagrama

Nuolatinio imitacinio optimizavimo sistemos esybių klasių diagrama pateikta 23 paveiksle. Šioje diagramoje vaizduojami duomenų objektai, kurie bus reikalingi optimizavimo sistemai realizuoti. Duomenų objektai turi užtikrinti pilną duomenų išsaugojimą ir pateikimą.



23 pav. Esiybių klasių diagrama

3.4. Detalus projektas

Šiame skyriuje aprašomos dvi pagrindinės nuolatinio imitacinio optimizavimo sistemos valdiklių klasės: integratoriaus veiklos logikos valdiklis ir stebėjimo sistemos veiklos logikos valdiklis. Pateikiami detalūs aprašymai apie klasių ryšius su esybėmis ir naudojamais langais. Taip pat apibūdinamos egzistuojančios operacijos ir jų charakteristikos.

Klasė. Integratoriaus posistemis. Integratoriaus veiklos logikos valdiklis.

Aprašymas. Klasė skirta veiklos procesų integravimui – importuoja arba eksportuoja veiklos procesus. Susieja IBM „Business Process Modeler“ programinį paketą su stebėjimo sistema.

Klasės **ryšiai** su:

- integratoriaus langu (vidiniu integratoriaus langu). Integratoriaus langas inicijuoja kreipimąsi į šią klasę ir perduoda vartotojo išauktus įvykius.
- resursų modelio, duomenų modelio, procesų modelio esybe. Klasė yra atsakinga už modelių esybės duomenis. Sukuria esybės objektus, juos redaguoja arba šalina.
- duomenų bazės posistemio veiklos logikos valdikliu. Klasė sukuria duomenų bazės valdymo objektą.

Klasės **operacijos:**

- Importuoti() – IBM modeliavimo įrankio duomenų perkėlimas į atskirą duomenų struktūrą.
- Eksportuoti() – stebėjimo sistemos pakeitimų eksportavimas į IBM modeliavimo įrankį.
- GautiProcesus() – veiklos procesų gražinimo funkcija.
- SaugotiProcesa() – veiklos procesų pakeitimų saugojimo funkcija.
- AtidarytiImportavimoLanga() – importavimo lango atidarymo funkcija.

Klasė. Stebėjimo posistemis. Stebėjimo sistemos veiklos logikos valdiklis.

Aprašymas. Klasė skirta veiklos procesų stebėjimui – fiksuojamos charakteristikos kintančios realizacinėje sistemoje. Stebimi veiklos procesų parametrų pakitimai gali būti koreguojami ir išsaugomi.

Klasės **ryšiai** su:

- stebėjimo langu. Stebėjimo langas inicijuoja kreipimąsi į šią klasę ir perduoda vartotojo išauktus įvykius. Langa atvaizduoja veiklos procesą, bei jo charakteristikas.
- veiklos procesų esybe. Klasė yra atsakinga už veiklos procesų esybės duomenis. Sukuria veiklos procesų objektus, juos keičia arba šalina. Esysbė saugo veiklos procesų informaciją.
- prieigos esybe. Klasė yra atsakinga už prieigos esybės duomenis. Ši esybė saugo informaciją apie realizuotos sistemos charakteristikų būseną.

Klasės **operacijos:**

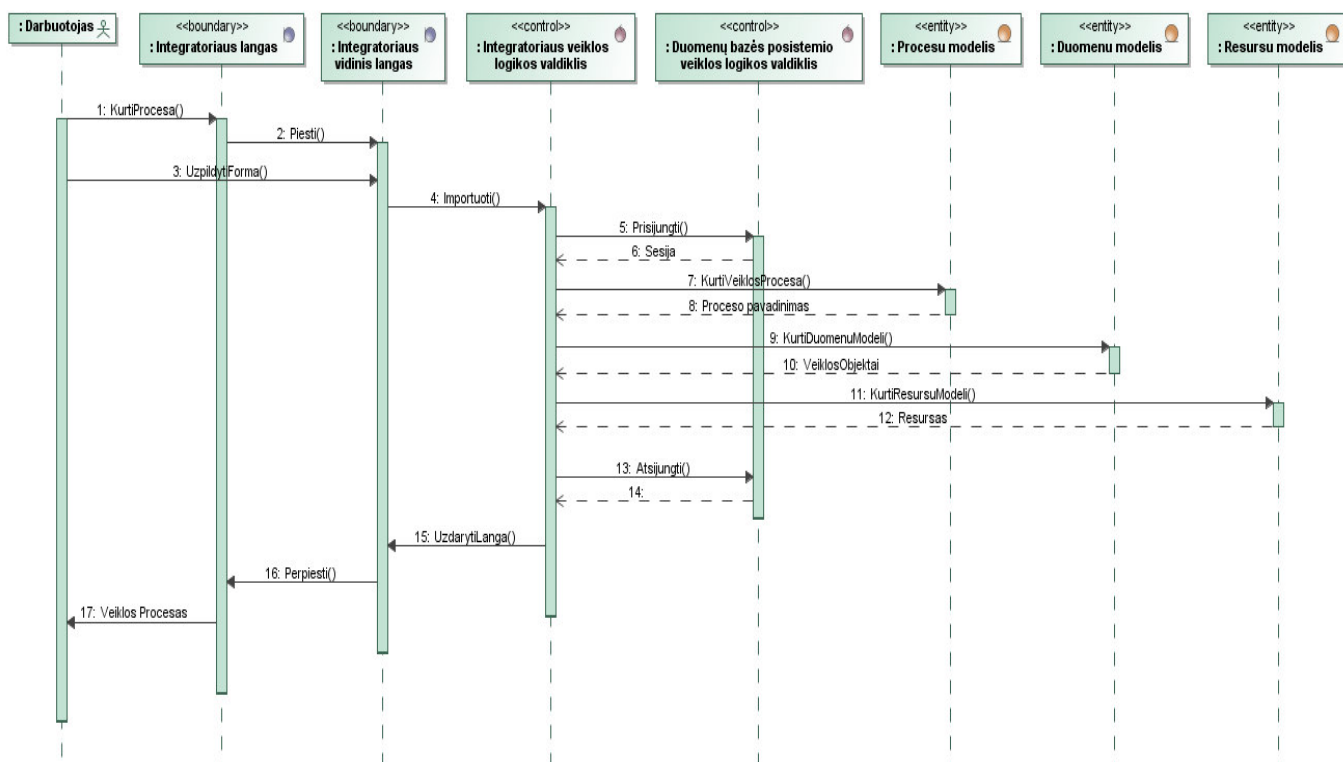
- TransformuotiXml() – XML duomenų struktūrų sukūrimas veiklos procesų modeliui atvaizduoti. Struktūra formuojama naudojant veiklos procesų esybės duomenis.
- SaugotiCharakteristikas() – veiklos procesų charakteristikų saugojimo funkcija.
- SaugotiProcesa() – veiklos procesų pakeitimų saugojimo funkcija.
- KeistiPadeti() – veiklos procesų objektų padėties keitimo funkcija.
- PridetiObjekta() – veiklos procesų objektų pridėjimo funkcija.

3.5. Elgsenos modelis

Šiame skyriuje aprašoma sistemų elgsena tarp naudojamų komponentų ir vartotojų. Išskiriamos dvi pagrindinės nuolatinės imitacinio optimizavimo sistemos posistemės: integratoriaus ir stebėjimo sistemos. Sistemų elgsenos modeliai vaizduojami sekų diagramomis.

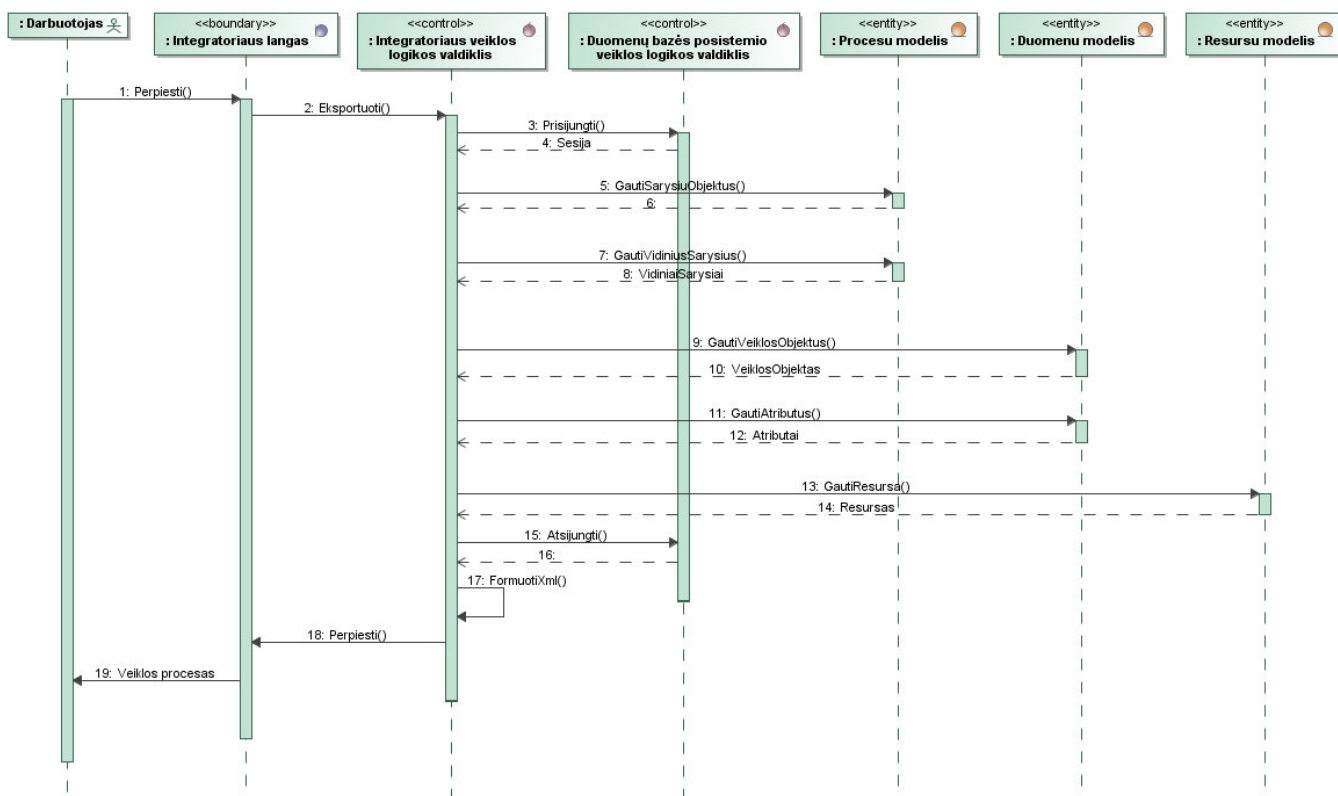
3.5.1. Integratoriaus posistemis

Panaudojimų atvejų diagramoje pavaizduoti du pagrindiniai integratoriaus panaudojimo atvejai: eksportuoti veiklos procesą, importuoti veiklos procesą į stebėjimo sistemą. Pagal šiuos atvejus sukuriamos dvi sekų diagramos. Sukurtų veiklos procesų IBM „Business Process Modeler“ programiniu modeliavimo įrankiu importavimas į stebėjimo sistemą pateiktas 24 paveiksle.



24 pav. Panaudojimo atvejo „importuoti veiklos procesą į stebėjimo sistemą“ sekų diagrama

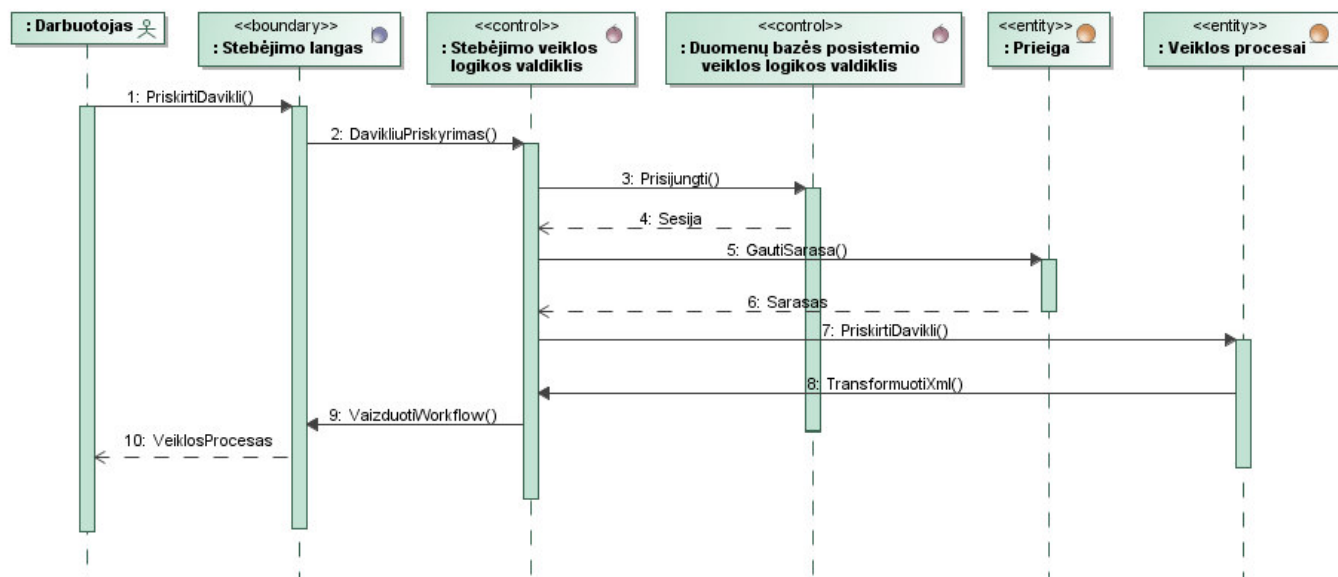
Pasikeitus veiklos procesų būsenai atsiranda būtinybė eksportuoti pakitusį veiklos procesą atgal į imitacinio optimizavimo įrankį. Šį procesą vaizduoja „eksportuoti veiklos procesą“ sekų diagrama, kuri pateikta 25 paveiksle.



25 pav. Panaudojimo atvejo „eksportuoti veiklos procesą“ sekų diagrama

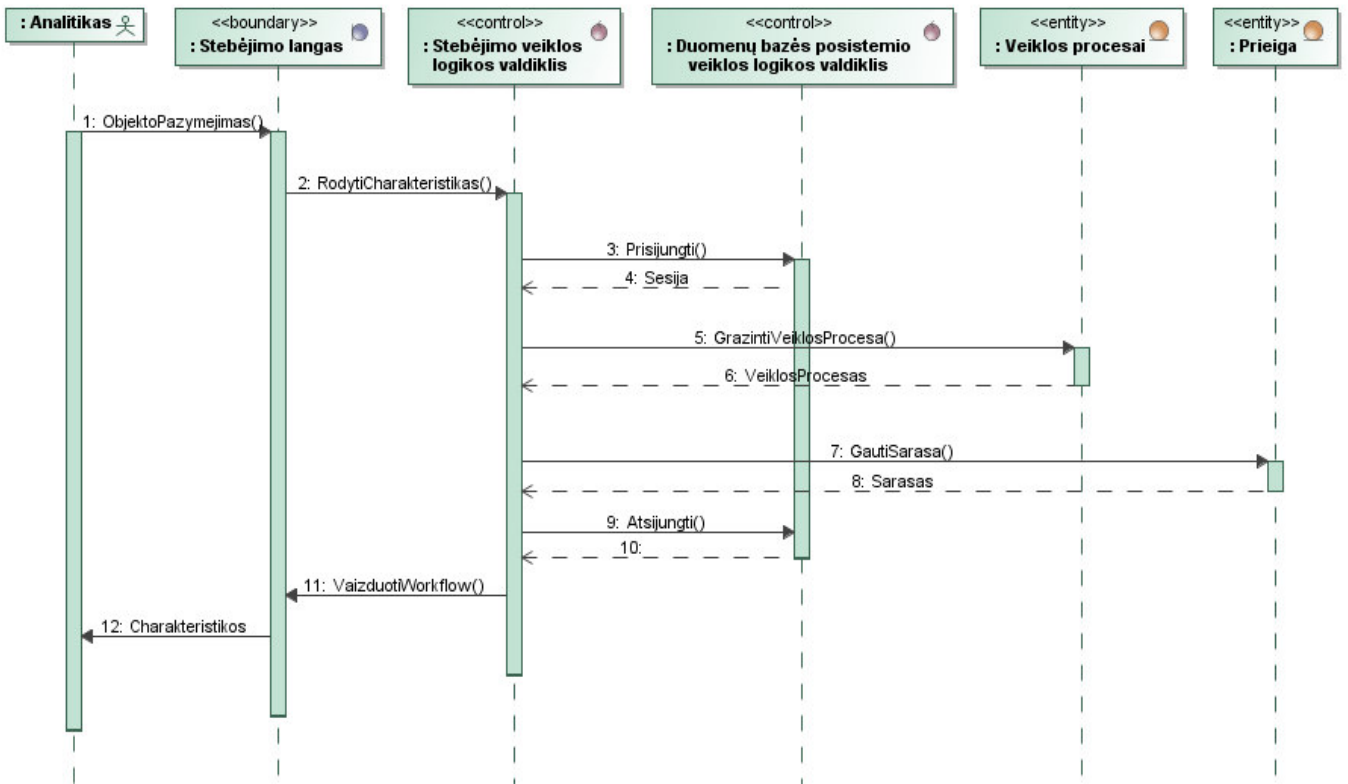
3.5.2. Stebėjimo sistemos posistemis

Stebėjimo sistemos posistemis atsakingas už importuotų veiklos procesų vaizdavimą. Šia sistemos dalimi naudojami darbuotojas ir analitikas. Vaizduojamiems objektams priskiriami optimizuojamos sistemos davikliai. 26 paveiksle pateikta panaudojimo atvejo „priskirti sistemos daviklius“ sekų diagrama.



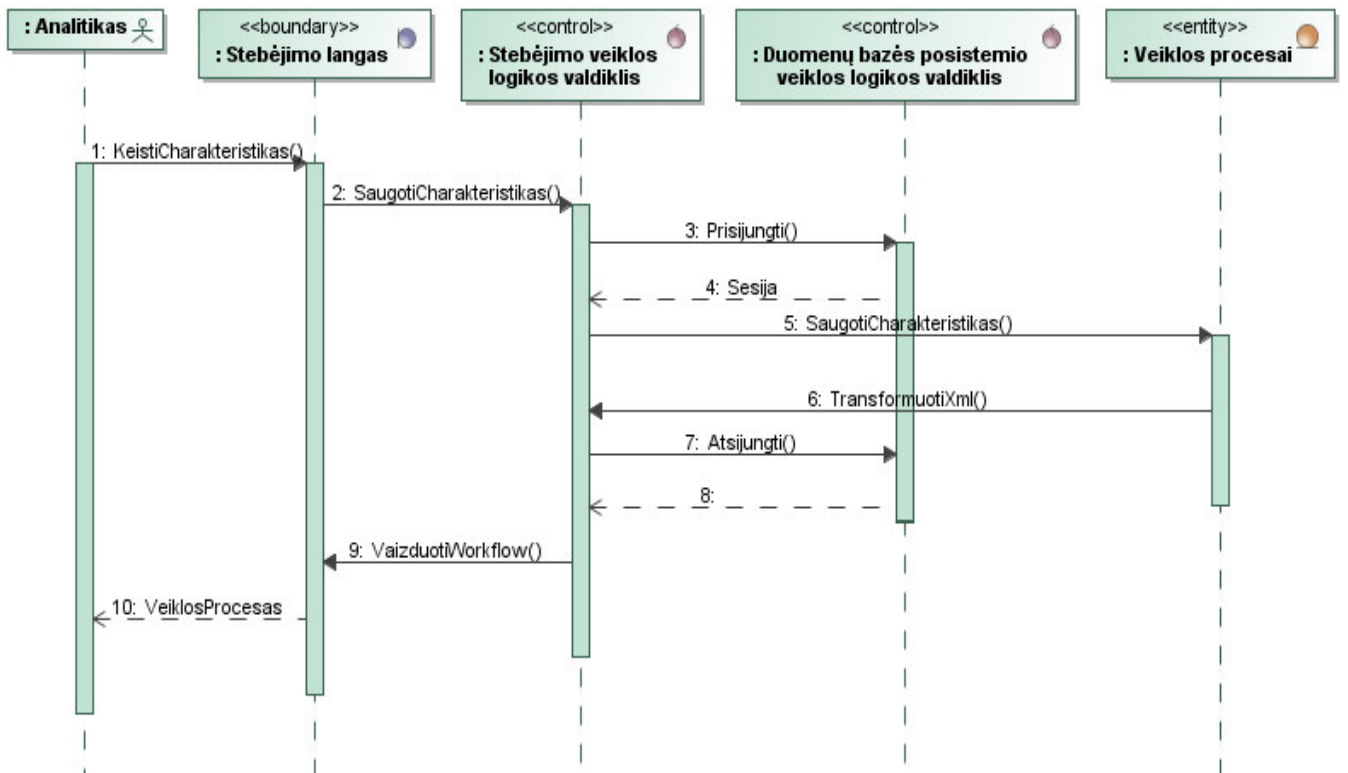
26 pav. Panaudojimo atvejo „priskirti sistemos daviklius“ sekų diagrama

Analitikas stebi veiklos procesų objektų charakteristikų pakitimus. Šiuos parametrus sistema atvaizduoja, kai pažymimas reikalingas objektas. 27 paveiksle pateikta panaudojimo atvejo „stebėti optimizavimo parametrus“ sekų diagrama.



27 pav. Panaudojimo atvejo „stebėti optimizavimo parametrus“ sekų diagrama

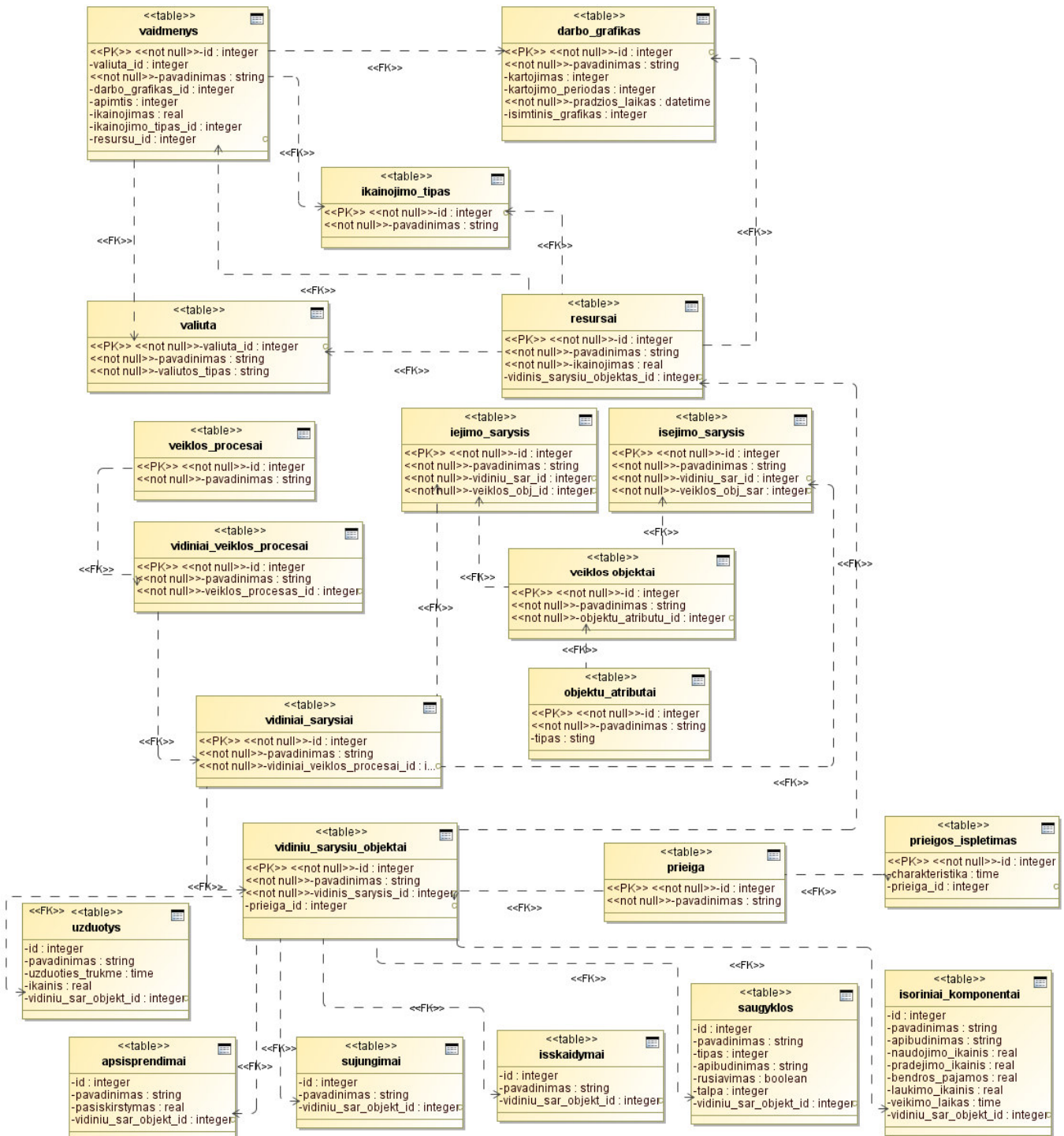
Stebimas charakteristikas keičia optimizuojamo veiklos proceso realizacija. Pastebėjus didesnius pokyčius atsiranda būtinybė koreguoti charakteristikas. 28 paveiksle pateikta panaudojimo atvejo „keisti optimizavimo parametrus“ sekų diagrama.



28 pav. Panaudojimo atvejo „keisti optimizavimo parametrus“ sekų diagrama

3.6. Duomenų bazės schema

Pasinaudojus esybių klasių diagrama, buvo sugeneruota duomenų bazės schema. Ši schema vaizduoja fizinius duomenų bazės objektus, kurių sudaro 21 lentelė. Sugeneruota schema pateikta 29 paveiksle.



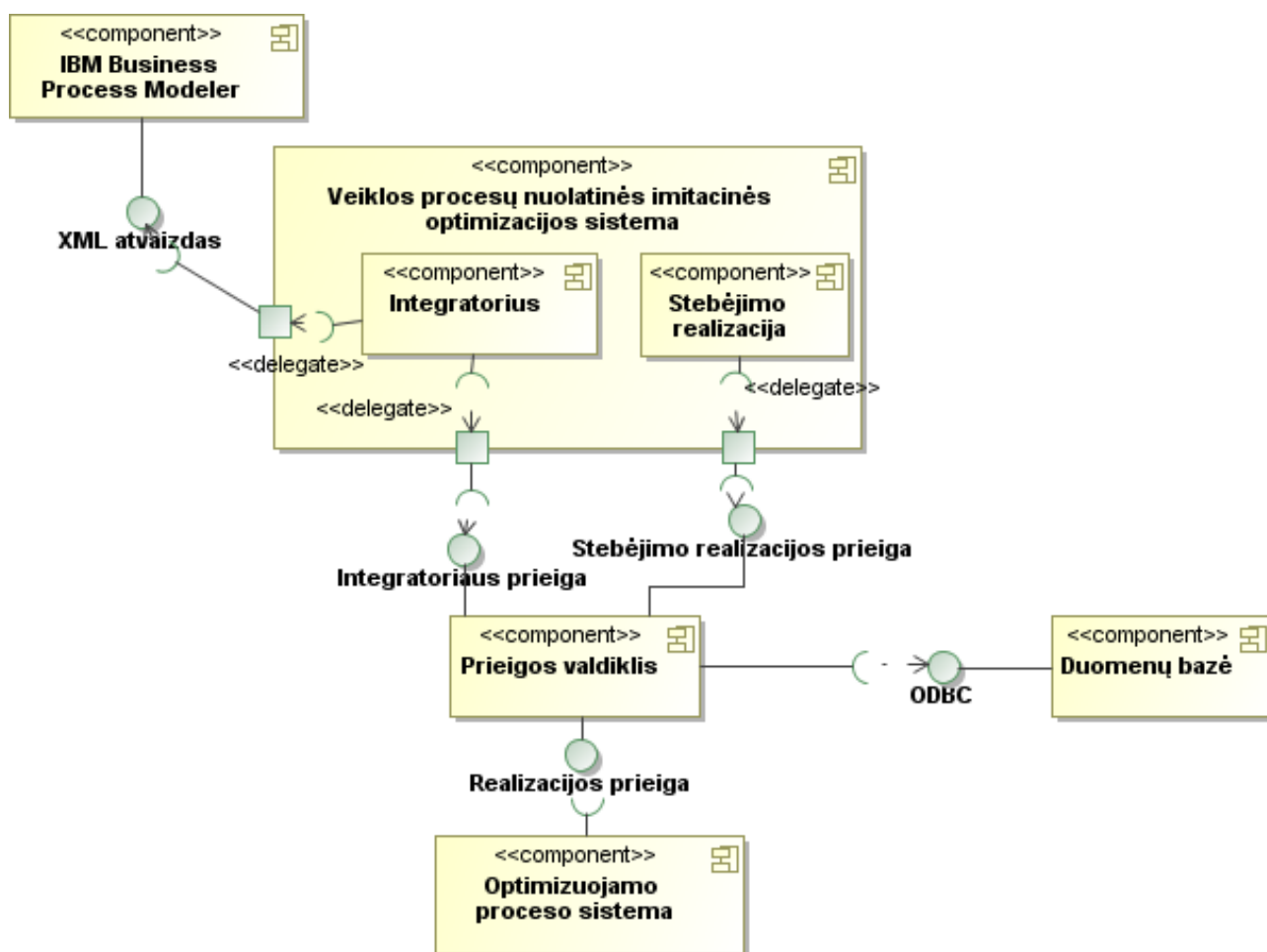
29 pav. Duomenų bazės diagrama

3.7. Realizacijos modelis

Nuolatinio imitacinio optimizavimo sistemą sudaro penkios komponentinės dalys:

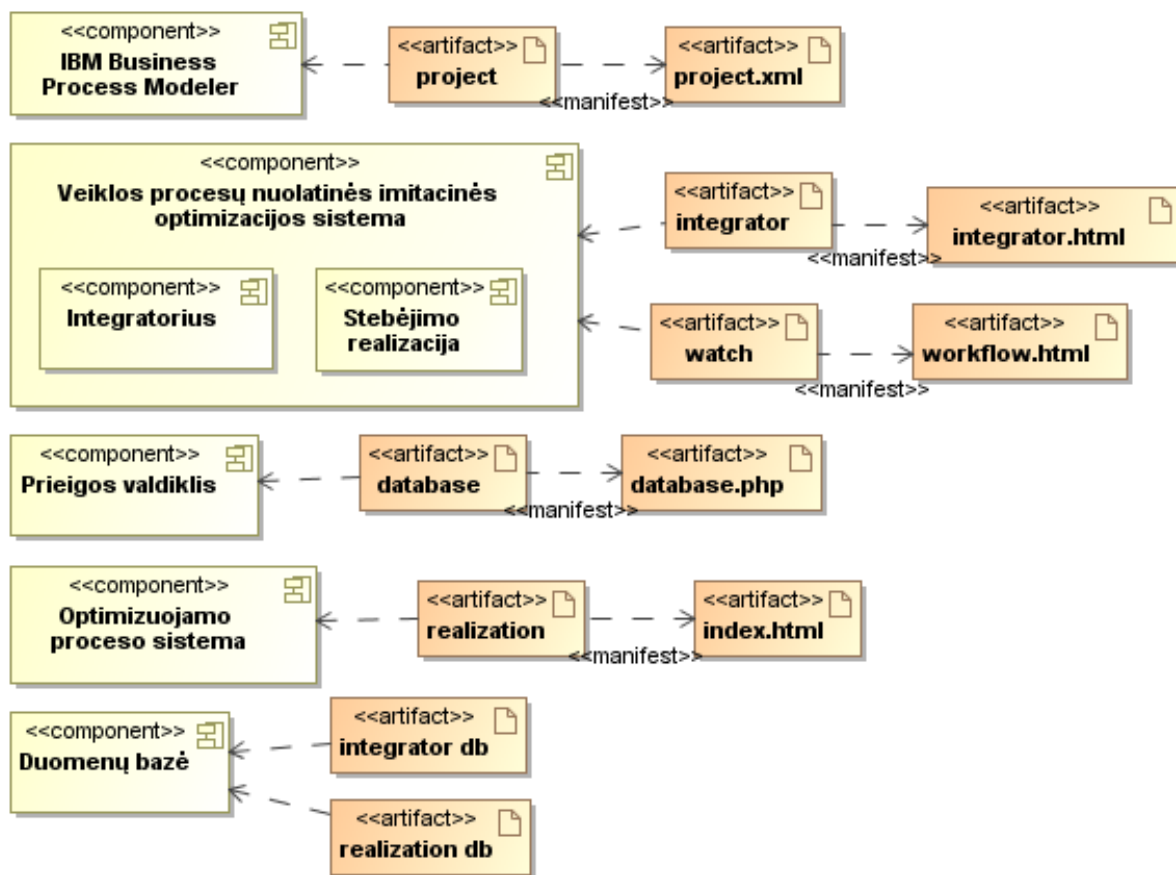
- **IBM „Business Process Modeler“** – veiklos procesų modeliavimo ir imitacinio optimizavimo įrankis. Šiuo įrankiu analitikai atlieka optimalių veiklos procesų paiešką.
- Veiklos procesų nuolatinės optimizacijos realizacija:
 - **Integratorius** – duomenų integravimo sistema naudojama sukurti duomenų struktūras gautas iš IBM įrankio.
 - **Stebėjimo sistema** – skirta fiksuoti seniūnijos teikiamų paslaugų sistemos darbą.
- **Optimizuojamo proceso sistema** – realizuotas veiklos procesas fiksuojantis daviklių parodymus.
- **Prieigos valdiklis** – komponentinių sistemos dalių duomenų bazės veiklos logikos realizacija. Šis valdiklis bendrauja su duomenų baze ir vykdo realizuotą logiką.
- **Duomenų bazė** – komponentinių sistemos dalių duomenų saugykla.

Sistemos komponentų diagrama pateikta 30 paveiksle.



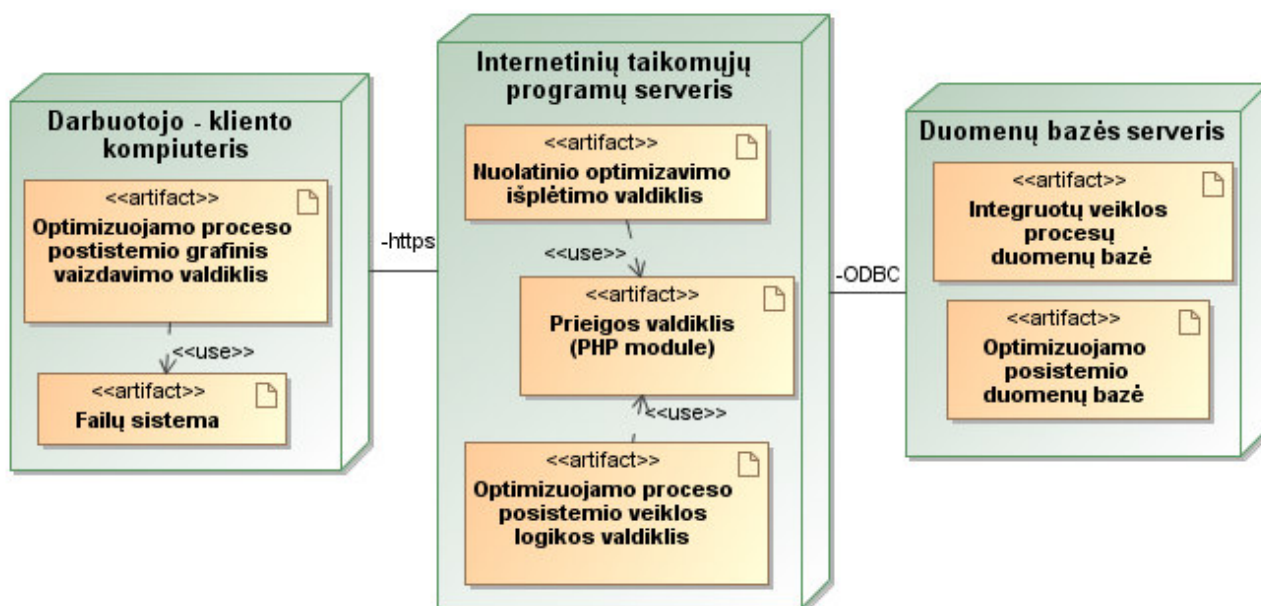
30 pav. Komponentų diagrama

Kiekvienas komponentas turi savo fizines duomenų laikmenas, kuriuose aprašyta veiklos logika. Šias laikmenas atvaizduoja 31 diagrama.



31 pav. Komponentų vaizdavimas fizinėmis duomenų laikmenomis

Fizinės duomenų laikmenos yra diegiamos į interneto taikomųjų programų serverį. Atsiradus poreikiui naudotis kliento failų sistema galima grafinį vaizdavimo valdiklį integruoti kliento kompiuteryje. Tai atliekama Adobe AIR programiniu paketu.



32 pav. Sistemos diegimo diagrama

4. Eksperimentinis nuolatinio imitacinio optimizavimo taikymas seniūnijos paslaugoms

4.1. Seniūnijų teikiamų viešųjų paslaugų modelio sudarymas

Realizuojant seniūnijų teikiamų viešųjų paslaugų modelį apibrėžti tokie etapai:

1. **Užsakymo priėmimas.** Seniūnijos gyventojas arba savivaldybės darbuotojas privalo atvykti į seniūniją ir pateikti dokumentų išdavimo prašymą.
2. **Prisijungimas prie sistemos.** Seniūnijos darbuotojas gavęs dokumentų išdavimo prašymą jungiasi prie naudojamos sistemos.
3. **Užsakymo įrašymas į duomenų bazę (sistema).** Seniūnijos darbuotojas įveda užsakymą į duomenų bazę.
4. **Dokumento atranka duomenų bazėje (sistema).** Seniūnijos darbuotojas tikrina, ar užsakyto dokumento šablonas egzistuoja programoje. Jei tokio dokumento nėra, tuomet specialiais įrankiais kuriamas naujas šablonas (**Dokumento kūrimas**).
5. **Asmens atranka duomenų bazėje (sistema).** Seniūnijos darbuotojas tikrina, ar asmuo yra duomenų bazėje. Jei tokio asmens nėra, tuomet specialiais įrankiais kuriama nauja asmens įrašymo kortelė.
6. **Dokumentų formavimas ir spausdinimas (sistema).** Programa suformuoja kliento užsakytą dokumentą ir nusiunčia į spausdinimo aparatą.
7. **Pristatymo užsakymas.** Jei paslauga teikiama valstybinei įstaigai, tuomet reikalingas dokumento pristatymas į atitinkamą vietą.
8. **Užsakymo įrašymo atnaujinimas duomenų bazėje (sistema).** Darbuotojas atnauжина užsakymo būklę duomenų bazėje.

Pagal šiuos etapus suprojektuotas seniūnijos teikiamų viešųjų paslaugų modelis. Ši veiklos procesą apibūdina matricos:

- vaidmenų ir resursų matrica;
- trukmės matrica;
- prieinamumo matrica;
- sprendimo tikimybės matrica.

Kuriamam seniūnijos teikiamų viešųjų paslaugų modeliui sudarytos šios matricos:

1. **Vaidmenų ir resursų matrica** (angl. Role resources matrix). Ši matrica parodo darbuotojų ir materialinių resursų tarpusavio santykį. Taip pat kiekvieno vaidmens kainą (paprastai ji apibrėžta kaip žmogaus atlyginimas per laiko vienetą, tačiau šiame modelyje egzistuoja resursai, kurie skaičiuojami kartais).

12 lentelė. Seniūnijų teikiamų viešųjų paslaugų modelio vaidmenų ir resursų matrica

Resursai Veiksmai	Kaina per val., Lt/h	Vieno karto kaina, Lt/k	Užsakymo priėmimas	Užsakymo įrašymas	Dokumento atranka	Asmens atranka	Dokumentų formavimas ir spausdinimas	Pristatymo užsakymas	Užsakymo įrašo atnaujinimas
Darbuotojas	9,38	-	✓		✓	✓		✓	
Klientas	5,00	-	✓						
Taikomoji programa	-	1,24		✓	✓	✓	✓		✓
Spausdintuvas	-	0,50						✓	
Telefonas	-	0,37						✓	
Siuntų Gabenimas	-	5,00						✓	

2. **Trukmės matrica** (angl. Duration matrix). Ši matrica parodo konkretaus veiksmo trukmę ir kiek laiko tam veiksmui realiai sugaištama.

13 lentelė. Seniūnijų teikiamų viešųjų paslaugų modelio trukmės matrica

Veiksmai Resursai	Kaina per val., Lt/h	Vieno karto kaina, Lt/k	Užsakymo priėmimas, min.	Užsakymo įrašymas, s	Dokumento atranka, s	Asmens atranka, s	Dokumentų formavimas ir spausdinimas, s	Pristatymo užsakymas, min.	Užsakymo įrašo atnaujinimas, s
Veiksmo trukmė	x	-	10	5	32	65	20	5	2
Darbuotojas	9,38	-	10		30	60		5	
Klientas	-	5,00	10						
Taikomoji programa	-	1,24		5	2	5	10		2
Spausdintuvas	-	0,50					10		
Telefonas	-	0,37						5	
Siuntų Gabenimas	-	5,00						5	

3. **Parengties matrica** (angl. Availability matrix). Ši matrica parodo žmonių vaidmenims priskirtą darbo grafiką. Pietų pertrauka yra įtraukta kaip nedarbo laikas.

14 lentelė. Seniūnijų teikiamų viešųjų paslaugų modelio prieinamumo matrica

Darbo grafikas	Resursas	Darbuotojas	Klientas	Siuntų gabenimas
Darbo diena (pirmadienis – penktadienis, 08:00-17:00 h)		✓		✓
Pietų pertrauka (pirmadienis – penktadienis, 12:00-13:00 h) – nedarbo laikas.		✓		✓
Pilnas (pirmadienis – sekmadienis, 01:00 - 24:00 h)			✓	

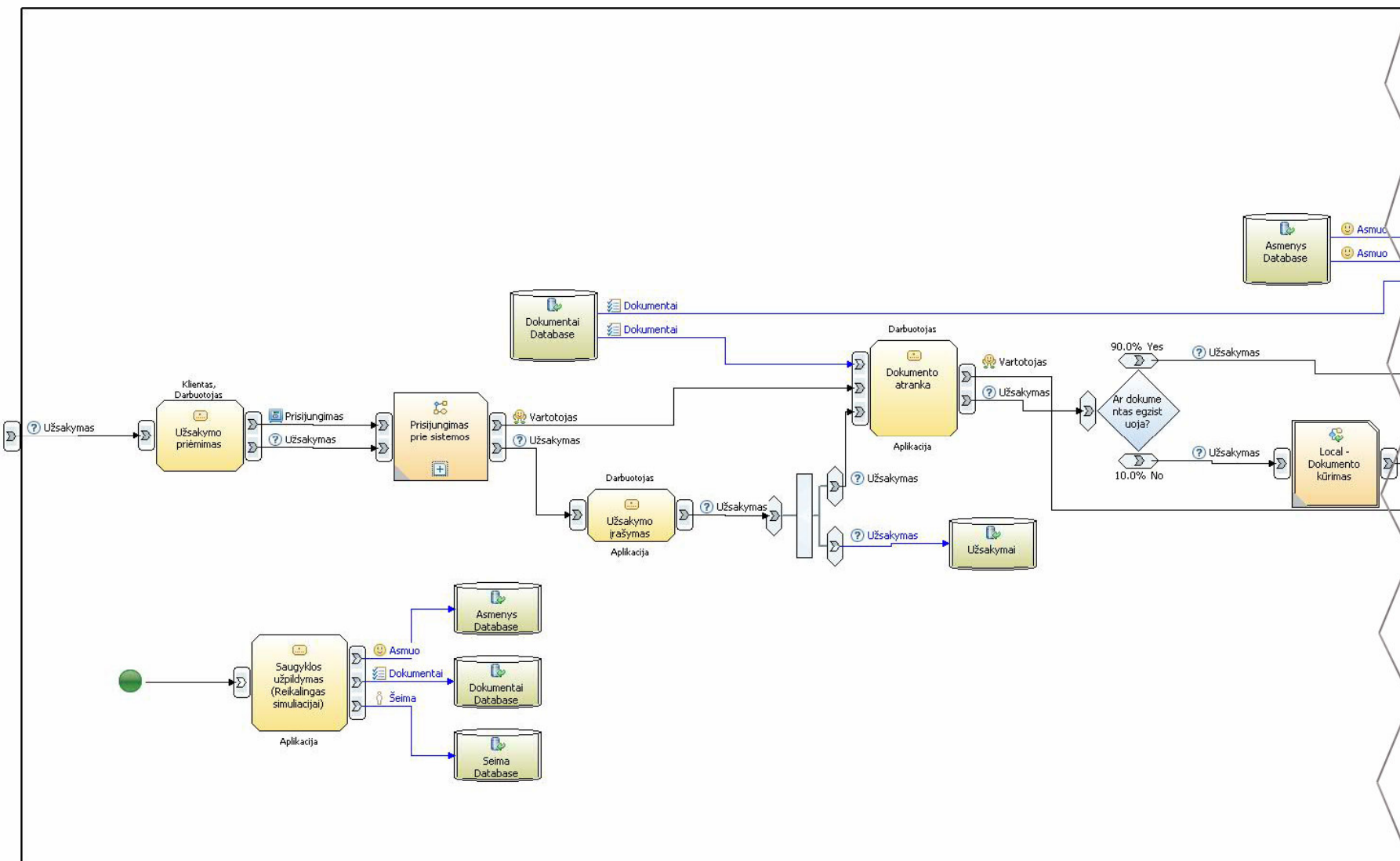
4. **Sprendimo tikimybės matrica** (angl. Decision probabilities matrix). Parodo galimus sprendimų variantus ir kokia tikimybė jiems įvykti.

15 lentelė. Seniūnijų teikiamų viešųjų paslaugų realaus modelio sprendinių tikimybės matrica

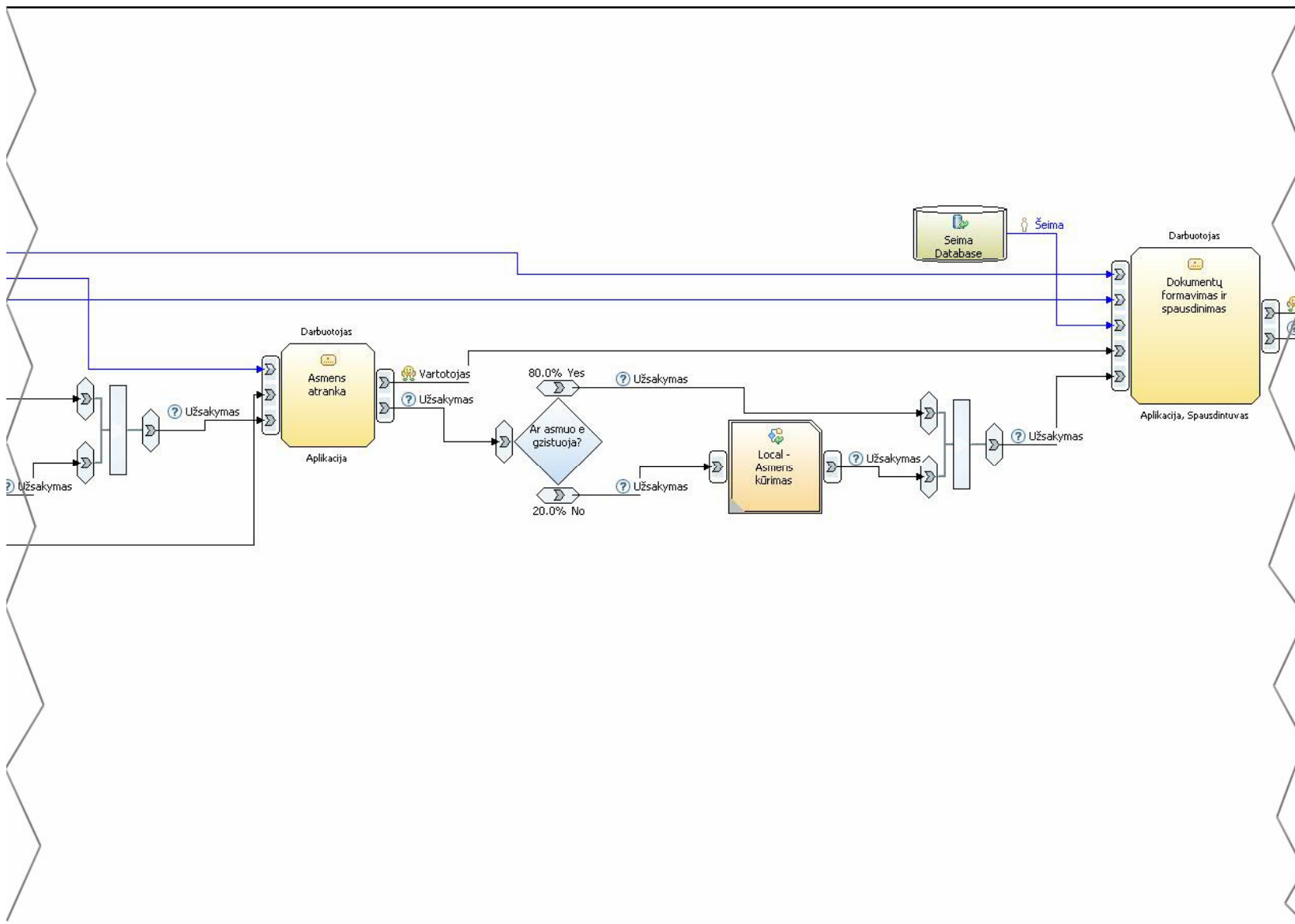
Sprendimas	Tikimybė	Taip	Ne
Ar dokumentas egzistuoja?		90%	10%
Ar asmuo egzistuoja?		80%	20%
Ar paslauga teikiama valstybinei įstaigai?		50%	50%

Pradedant paslaugos užsakymą, reikalingas prašymas. Prašymas formuojamas objekto pavidalu: raštiniu, spausdintiniu ir t.t. Šiame procese būtų neišvengiamas darbuotojų darbo krūvis, kliento stovėjimas eilėse. Tačiau, šis procesas yra kompiuterizuotas. Visi duomenys yra saugomi duomenų bazėse. Taip pat sukurta elektroninė sistema, per kurią darbuotojas įvedinėja duomenis, spausdina dokumentus.

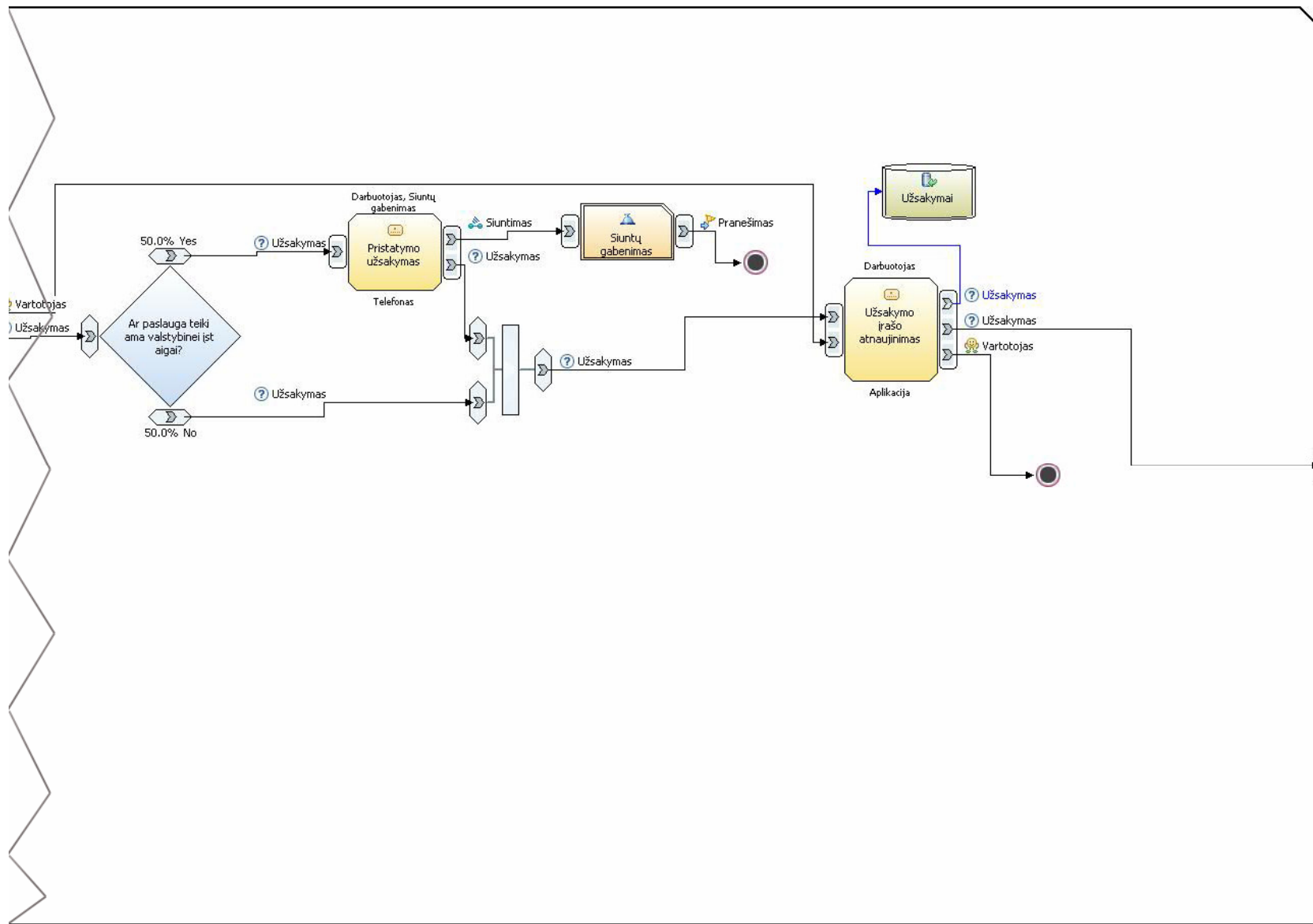
Žemiau pateikti seniūnijos teikiamų viešųjų paslaugų modeliai.



33 pav. Seniūnijos teikiamų viešųjų paslaugų modelis (1 dalis iš 3)



34 pav. Seniūnijos teikiamų viešųjų paslaugų modelis (2 dalis iš 3)



35 pav. Seniūnijos teikiamų viešųjų paslaugų modelis (3 dalis iš 3)

4.2. Seniūnijos teikiamų viešųjų paslaugų imitacinis optimizavimas

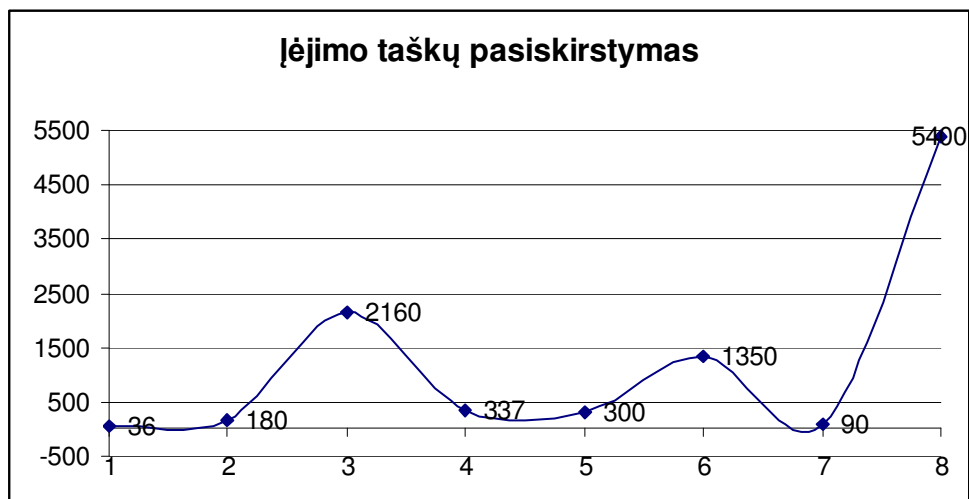
Vykiant imitacinį optimizavimą didžiausią reikšmę turi įėjimo taškų dažnumas (angl. spread). Įėjimo taškai, šiuo atveju yra klientai užsakantys paslaugą. Šiuo parametru optimizatorius gali parodyti ar pakankamas darbuotojų skaičius dirba įstaigoje. Taip pat apibrėžti maksimalų darbuotojų našumą. 16 lentelėje pavaizduota imitacinio optimizavimo rezultatų suvestinė.

16 lentelė. Procesų trukmė

Procesas	Darbuotojų skaičius	Įėjimo taškai, k/h	Laikas, s	Taškų dažnumas, k/h
Užsakymo priėmimas	3	50	5000	36
Prisijungimas prie sistemos	3	50	1000	180
Užsakymo įrašymas į duomenų bazę (sistema)	3	50	83	2160
Dokumento atranka duomenų bazėje (sistema)	3	50	833	337
Dokumento kūrimas	3	50	X	X
Asmens atranka duomenų bazėje (sistema)	3	50	1083	300
Asmens kūrimas	3	50	X	X
Dokumentų formavimas ir spausdinimas (sistema)	3	50	133	1350
Pristatymo užsakymas	3	50	2000	90
Užsakymo įrašymo atnaujinimas duomenų bazėje (sistema)	3	50	33	5400

X – klientas ir darbuotojas tiesiogiai nesusiję su procesu.

Žemiau pateiktas grafikas vaizduojantis pasiskirstymo monotoniškumą.



36 pav. Įėjimo taškų pasiskirstymas

Pastebime, kad egzistuoja dvi užduotys trikdančios monotonišką veiklos proceso veikimą. Pristatymo užsakymas vykdomas labai retai, todėl į šią užduotį galima nekreipti dėmesio. Tačiau „Užsakymo priėmimas“ aptarnauja tik 36 klientus per valandą. Reikia atkreipti dėmesį į tai, kad tuo metu dirba 3 darbuotojai. Galima daryti išvadą, kad naudojamas veiklos procesas netenkina miestų seniūnijos

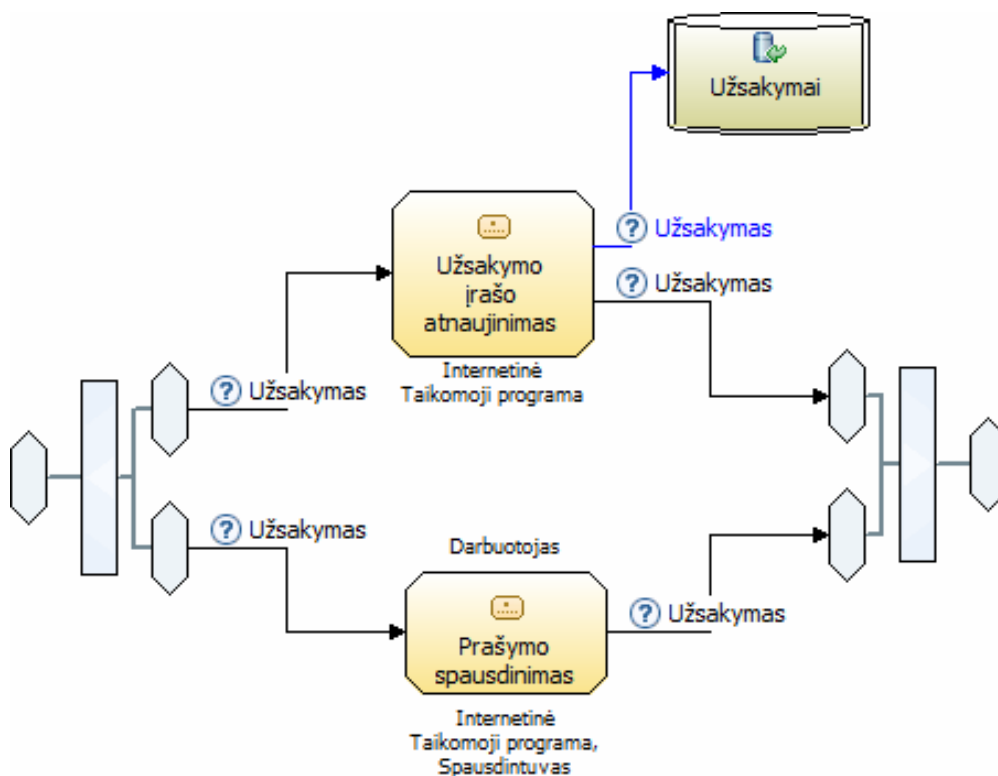
teikiamų viešųjų paslaugų veiklos. Taškų dažnumas turi būti dvigubai didesnis, nei šiuo metu egzistuojantis. Veiklos proceso pradinė užduotis pateikta 37 paveiksle.



37 pav. Veiklos proceso pradinė užduotis „Užsakymo priėmimas“.

Vienas iš sprendimo būdų - proceso „Užsakymo priėmimas“ šalinimas iš modelio. Tačiau, seniūnijos reikalauja užsakytos paslaugos prašymo. Šią veiksmą galime leisti suformuoti taikomajai programai, kai paskutiniame žingsnyje visa informacija yra žinoma. Klientui lieka tik pasirašyti atspausdintą prašymą.

Tokiu principu perorganizuota užduotis „Užsakymo priėmimas“ suvienodina paslaugos teikimo trukmės srautą padarydamas jį monotoniškesnį. Pagerintame modelyje įėjimo taškų dažnumas padidėja kelis kartus. Taigi, klientų aptarnavimas taip pat pagreitėja kelis kartus. 38 paveiksle pateikta pertvarkyta užduotis „Užsakymo priėmimas“.



38 pav. Optimizuotas veiklos procesas „Prašymo spausdinimas“ ir „Užsakymo įrašo atnaujinimas“.

Negalima teigti, kad šis veiklos procesas visiškai optimizuotas – sukurtas optimalus modelio variantas, nes dažniausiai, veiklos procesas priklausomas nuo labai daug kintamųjų, kurie keičiasi kiekvieną dieną. Svarbu pastoviai optimizuoti (stebėti) vykdomą įstaigos veiklą.

4.3. Optimizuoto veiklos proceso apibendrinimas

Naujo modelio įdiegimas yra perspektyvus ir netgi būtinas siekiant optimizuoti paslaugos tiekimo procesą. Sumodeliuotas esamas seniūnijos teikiamų viešųjų paslaugų (dokumentų išdavimo) procesas trunka kelis kartus ilgiau negu optimizuotas. Realiame modelyje atliktos paslaugos įkainuotos vidutiniškai 8,89 Lt, o optimizuoto modelio vidutinė kaina - tik 5,16 Lt. Tokiu būdu taupomas laikas ir piniginių lėšos.

Atsižvelgiant į optimizuoto modelio įvykių seką, klientams nebereikia rašyti prašymo. Dokumentą suformuoja internetinė taikomoji programa. Įdiegus optimizuotą veiklos procesą seniūnijos darbe mažėja klientų eilės. Tokiu būdu taupomi tiek mokesčių mokėtojų pinigai, tiek mažinamas institucijų darbuotojų apkrovimas.

Kintant veiklos sąlygoms keičiasi ir veiklos procesų savybės. Dėl to būtina diegti nuolatinį veiklos procesų optimizavimą. Periodinis procesų optimizavimas leistų pasiekti geriausių rezultatų, nes atsiradę netikslumai būtų pastebimi ir koreguojami kai tik jie atsiranda.

5. Nuolatinio imitacinio optimizavimo sistemos taikymo seniūnijos teikiamoms viešosioms paslaugoms realizacija

5.1. Sistemos įdiegimas

Nuolatinio imitacinio optimizavimo sistema buvo sukurta naudojantis naujausiomis internetinėmis technologijomis. Šios sistemos ypatingos tuo, kad joms nereikalinga jokia papildoma programinė įranga. Vartotojas sistemą pasiekti gali naudodamasis naršykle. Tačiau dažnai prireikus specifinių funkcijų (priėjimas prie failų sistemos) naudojamas Adobe įmonės AIR programinis paketas (angl. Adobe Integrated Runtime). Kaip buvo minėta, taip sukurta programinė įranga turi du sistemos veikimo principus: API ir GUI. Diegiant realizuotą sistemą serverinėje dalyje reikalinga tokia programinė įranga:

- Apache taikomųjų programų serveris. Šis HTTP (angl. HyperText Transfer Protocol) serveris naudojamas statinio ir dinaminio turinio svetainėms žiniatinklyje publikuoti.
- PHP 5 modulis naudojamas kartu su Apache taikomųjų programų serveriu. Šis modulis skirtas susieti duomenų bazę su duomenų bazės posistemės veiklos logikos valdikliu.
- Duomenų bazė (Microsoft SQL Server, MySQL server, Oracle Database). Organizuotas duomenų rinkinys, kuriuo galima individualiai naudotis elektroniniu ar kitu būdu.
- ExtJS JavaScript karkasas. JavaScript programavimo kalbos pagalbinis paketas.

Diegiant realizuotą sistemą į klientinę dalį (jei reikalingas priėjimas prie failų sistemos) reikalinga tokia programinė įranga:

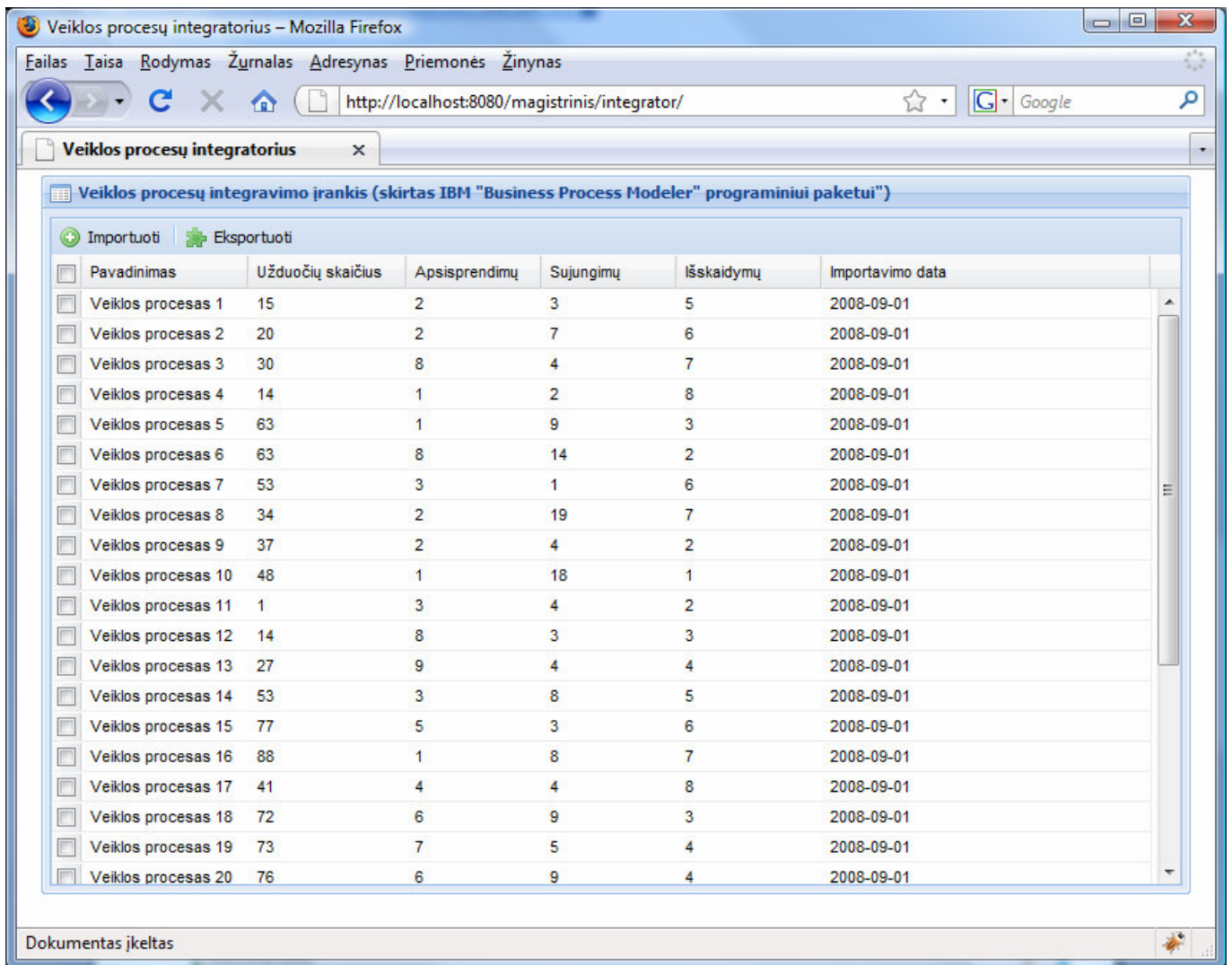
- Adobe įmonės AIR programinis paketas (angl. Adobe Integrated Runtime);
- ExtJS JavaScript karkasas. JavaScript programavimo kalbos pagalbinis paketas.

5.2. Veiklos procesų integravimas

Veiklos procesų nuolatinis imitacinis optimizavimas pradedamas duomenų integravimu iš IBM „Business Process Modeler“ modeliavimo įrankio. Už visą integravimo procesą atsakingas veiklos procesų integravimo įrankis. Šis programinis sprendimas pavaizduotas 39 paveiksle.


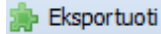
Atidarius integravimo įrankio pagrindinį langą matomas integruotų veiklos procesų sąrašas. Šis sąrašas papildytas vidinių objektų skaičiumi ir importavimo data. Sąrašo detalizavimas:

- veiklos proceso pavadinimas;
- veiklos proceso užduočių skaičius;
- veiklos proceso apsisprendimų skaičius;
- veiklos proceso sujungimų skaičius;
- veiklos proceso išskaidymų skaičius;
- veiklos proceso importavimo data.

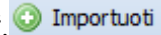


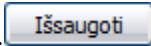


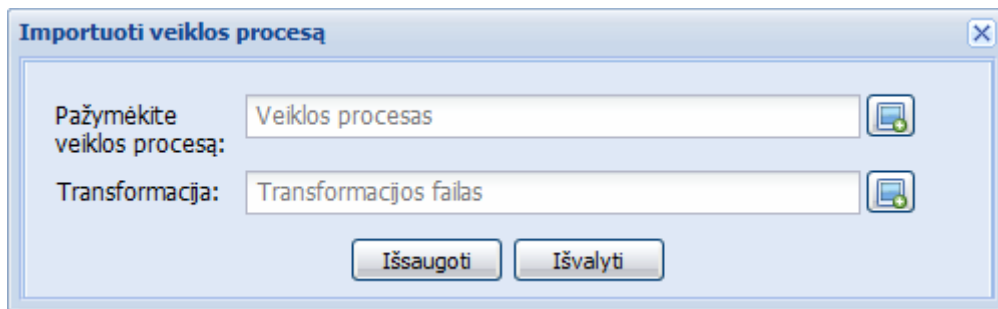
39 pav. Veiklos procesų integravimo įrankio langas

Pagrindinis integravimo langas gali atlikti dvi pagrindines funkcijas: importuoti arba eksportuoti veiklos procesus. Eksportavimas vykdomas šia seka:


- 1) Pažymimas žymeklis ties pasirinktu veiklos procesu. 
- 2) Spaudžiamas mygtukas „Eksportuoti“. 

Importavimas vykdomas šia seka:

- 1) Spaudžiamas mygtukas „Importuoti“. 
- 2) Atsidaro vidinis sistemos langas. Šis langas pavaizduotas 40 paveiksle.
- 3) Spaudžiamas laikmenos pasirinkimo mygtukas ir pasirenkama IBM “Business Process Modeler” programinės įrangos sugeneruota XML laikmena. 
- 4) Spaudžiamas transformacijos pasirinkimo mygtukas ir pasirenkama XSL laikmena, transformuojanti pasirinktą laikmeną trečiame punkte. 
- 5) Spaudžiamas mygtukas išsaugoti. 

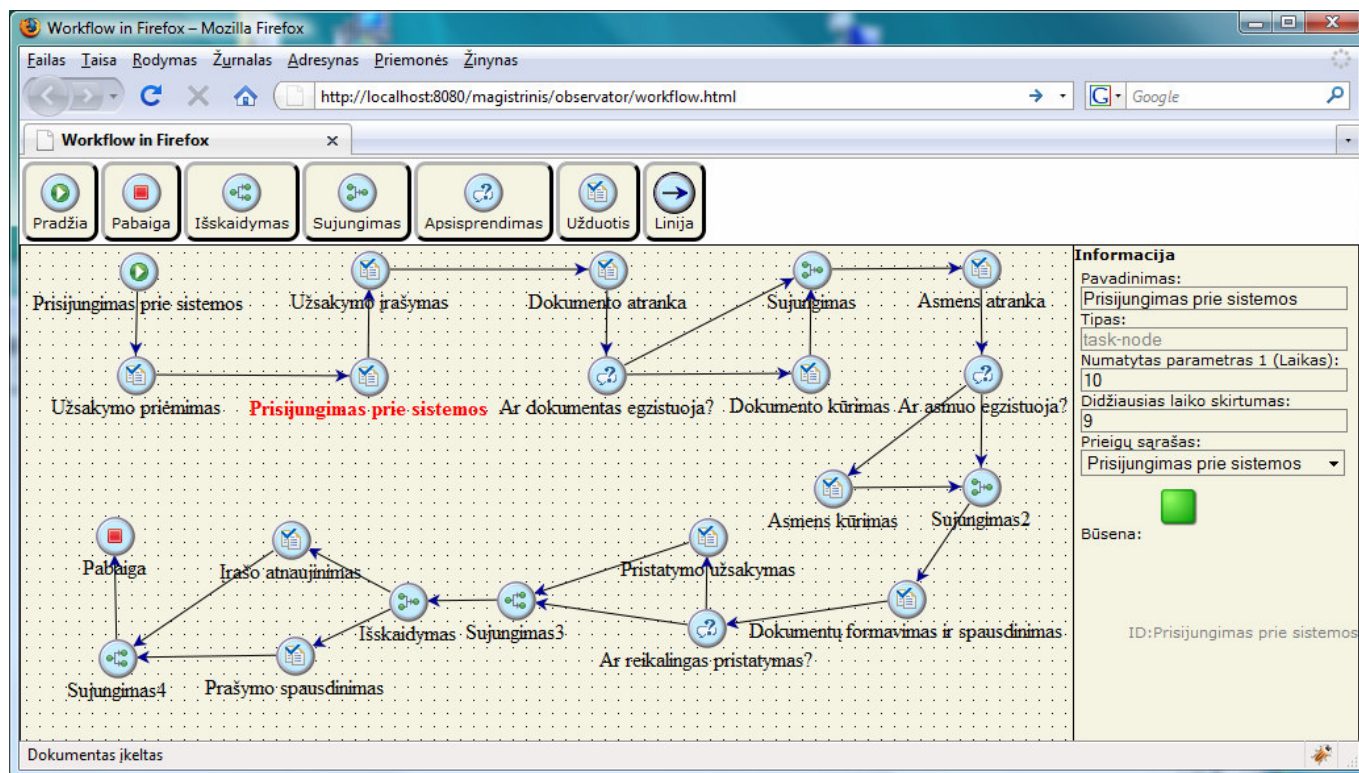


40 pav. Vidinis integratoriaus sistemos importavimo langas

Vidinis integratoriaus sistemos importavimo langas tikrina pasirinktas laikmenas ir iškilus netikslumams praneša informaciją vartotojui. 

5.3. Veiklos procesų vaizdavimo ir charakteristikų sąsaja








Integruotus veiklos objektus vaizduoja veiklos procesų vaizdavimo ir charakteristikų langas. Atidarius pagrindinį šios sistemos langą matomos trys pagrindinės sritys: vaizdavimo, charakteristikų ir objektų. Veiklos procesų vaizdavimo ir charakteristikų langas pateiktas 41 paveiksle.



41 pav. Veiklos procesų vaizdavimo ir stebėjimo langas




Vaizdavimo sritis atsakinga už egzistuojančių objektų ir sąryšių tarp jų išdėstymą. Šioje srityje galima taisyti objektus į visas puses. Taip pat šalinti sujungimą ar objektą. Pažymint pasirinktą vaizduojamą objektą yra atnaujinama charakteristikų sritis. Objektų sritį sudaro 7 pagrindiniai valdymo mygtukai. Paspaudus vieną iš šių mygtukų vaizdavimo srityje atsiranda objektas, kurį galima integruoti į bendrą veiklos procesą. Šių mygtukų charakteristikos pateiktos 17 lentelėje:

17 lentelė. Veiklos objektų sąrašas

Atvaizdas	Pavadinimas	Aprašymas
	Pradžia	Veiklos proceso pradžią apibūdinantis objektas. Turi tik išėjimo taškus (įėjimo taškų neturi)
	Pabaiga	Veiklos proceso pabaigą žymintis objektas. Turi tik įėjimo taškus (išėjimo taškų neturi)
	Išskaidymas	Jungties objektų išskaidymas į kelis ar daugiau punktų. Gali turėti vieną įėjimo jungtį, kuri skaidoma į nustatytą kiekį jungčių
	Sujungimas	Jungties objektų sujungimas į vieną jungtį. Gali turėti daug įėjimo tačiau vieną išėjimo jungtį.
	Apsisprendimas	Apsisprendimą nusakantis objektas. Turi vieną įėjimo jungtį ir kelis ar daugiau išėjimo jungčių.
	Užduotis	Veiklos proceso vidinė užduotis. Turi vieną ar daugiau įėjimo ir išėjimo jungčių.
	Jungtis	Jungties objektas. Jungia kitus veiklos proceso objektus.

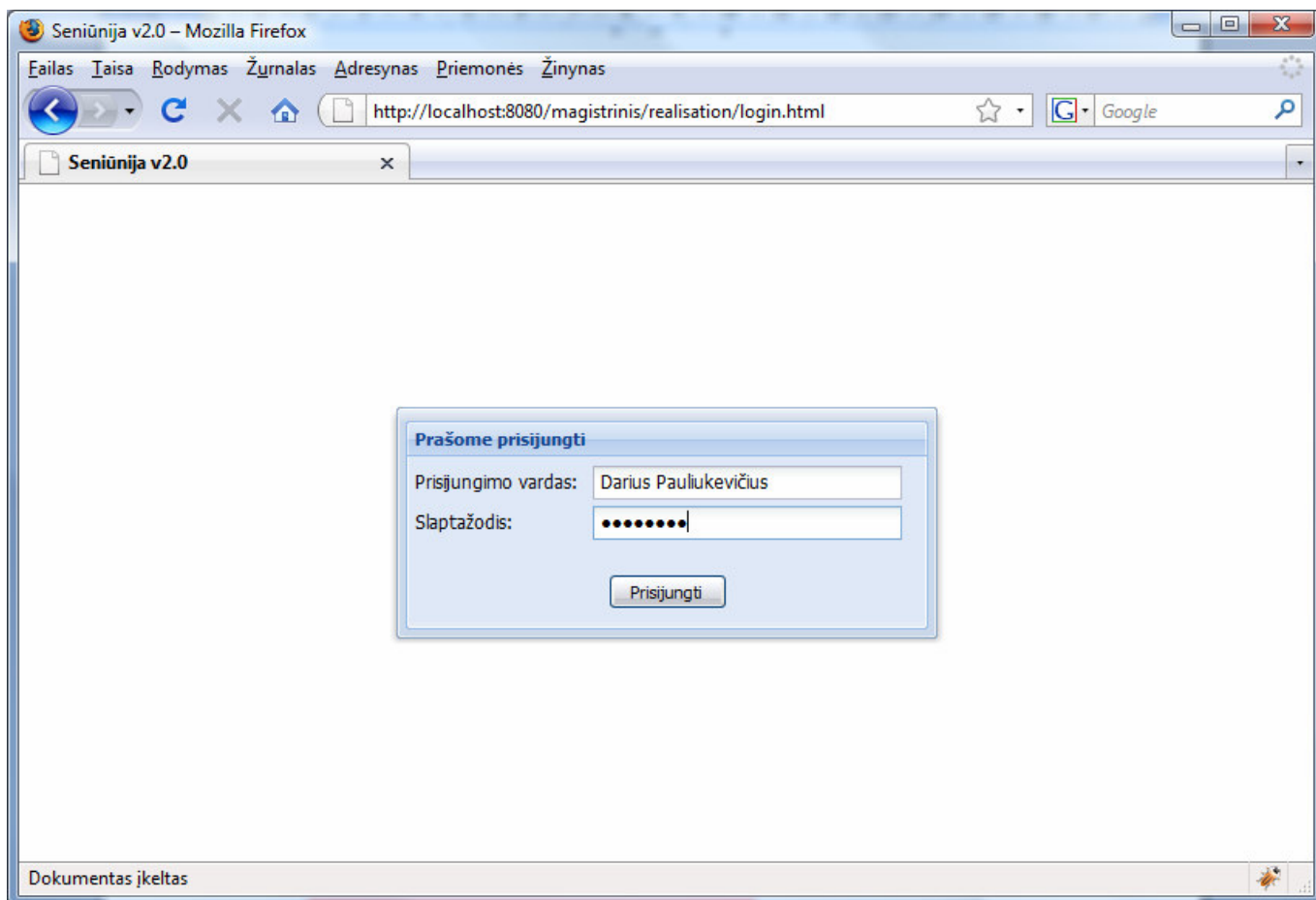
Vaizdavimo srityje pažymėjus vieną iš objektų atsinaujinama charakteristikų sritis. Šioje dalyje vaizduojama bendra objekto informacija: pavadinimas, tipas, charakteristinis parametras. Taip pat atnaujinami realizacijos charakteristiniai nuokrypiai ir prieigų sąrašas. Pažymėtas objektas turi savo būseną, kuri paskaičiuojama pagal gautus charakteristikų įvertinimus. Būsenų tipai detaliam aprašyti 18 lentelėje.

18 lentelė. Veiklos objektų būsenų sąrašas

Atvaizdas	Pavadinimas	Aprašymas
	Kritinė būsena	Kritinė būsena parodo, kad charakteristikų nuokrypis yra labai didelis. Todėl būtina keisti parametrus arba eksportuoti veiklos procesą į IBM optimizatorių ir atlikti imitacinį optimizavimą
	Patenkinama būsena	Patenkinama būsena parodo, kad yra nežymus nuokrypis nuo pateiktų charakteristikų ir būtų pravartu imituoti veiklos procesą panaudojant IBM optimizatorių. Tokiu būdu būtų galima nustatyti ar rizika yra didelė.
	Gera būsena	Gera būsena parodo, kad nėra žymių pakitimų, dėl kurių kiltų rimtesnių problemų.

5.4. Seniūnijos teikiamų viešųjų paslaugų sistemos prisijungimas

Paleidus seniūnijos teikiamų viešųjų paslaugų sistemos realizaciją pasileidžia vidinis prisijungimo langas. Šis langas pateiktas 42 paveiksle. Kiekvienas vartotojas prisijungęs ir išsaugojęs slaptažodį naršyklės nustatymuose paleidus programą šio lango nebematys. Sesija yra išsaugoma kiekvieno kompiuterio atmintyje.



42 pav. Seniūnijos teikiamų viešųjų paslaugų sistemos prisijungimo langas

Atidarytas programos langas jungiasi prie duomenų bazės ir tikrina, ar egzistuoja darbuotojas, bandantis prisijungti prie sistemos. Įvykus prisijungimo klaidai, išvedamas pranešimas apie įvykusių neatitikimus. Prisijungimo langas reikalauja dviejų parametrų: prisijungimo vardo ir slaptažodžio. Jei įvesti duomenys yra teisingi, šis langas užkrauna pagrindinę sistemos realizaciją.

5.5. Seniūnijos teikiamų viešųjų paslaugų sistemos sąsaja

Sėkmingai prisijungus prie sistemos atidaromas pagrindinis valdymo langas (43 paveikslas).

Seniūnijos teikiamų viešųjų paslaugų realizacinį langą sudaro trys sritys:

- valdymo medis (pagrindinis meniu);
- pagrindinis turinys;
- papildoma informacija.

Paspaudus norimą punktą iš valdymo medžio aktyvuojasi pagrindinio turinio vaizdavimas. Šis turinys yra integruotas į vidinį objektą. Tokiu principu sukurta sistema lengva naudotis, nes visi aktyvuoti langai yra pasiekiami akimirksniu. Pagrindinis meniu yra skirtingas kiekvienam vartotojui. Administratoriai mato pilną valdymo medžio struktūrą. Asmenys, turintys minimalias teises, gali tik spausdinti pažymas ir ataskaitas (ne visas, o nurodytas nustatymų lange). Redaktoriai turi papildomą funkciją - asmens kortelių valdymą ir namų ūkių kūrimą.

Papildomos informacijos sritis pateikia detalią informaciją apie kiekvieną aktyvų langą.

Nr.	Vardas	Pavardė	Asmens kodas	Kortelės būklė	Adresas	Veikla
1	Vardenis 1	Pavardenis 1	111111111111	Galioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Mokosi
2	Vardenis 2	Pavardenis 2	111111111111	Negalioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Žvejys" firmoje
3	Vardenis 3	Pavardenis 3	111111111111	Mirties įrašas	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Merkys" firmoje
4	Vardenis 4	Pavardenis 4	111111111111	Galioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Žvejys" firmoje
5	Vardenis 5	Pavardenis 5	111111111111	Negalioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Merkys" firmoje
6	Vardenis 6	Pavardenis 6	111111111111	Mirties įrašas	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Mokosi
7	Vardenis 7	Pavardenis 7	111111111111	Galioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Žvejys" firmoje
8	Vardenis 8	Pavardenis 8	111111111111	Negalioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Merkys" firmoje
9	Vardenis 9	Pavardenis 9	111111111111	Mirties įrašas	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Mokosi
10	Vardenis 10	Pavardenis 10	111111111111	Galioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Žvejys" firmoje
11	Vardenis 11	Pavardenis 11	111111111111	Negalioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Merkys" firmoje
12	Vardenis 12	Pavardenis 12	111111111111	Galioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Žvejys" firmoje
13	Vardenis 13	Pavardenis 13	111111111111	Mirties įrašas	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Merkys" firmoje
14	Vardenis 14	Pavardenis 14	111111111111	Negalioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Mokosi
15	Vardenis 15	Pavardenis 15	111111111111	Galioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Žvejys" firmoje
16	Vardenis 16	Pavardenis 16	111111111111	Negalioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Merkys" firmoje
17	Vardenis 17	Pavardenis 17	111111111111	Galioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Žvejys" firmoje
18	Vardenis 18	Pavardenis 18	111111111111	Galioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Merkys" firmoje
19	Vardenis 19	Pavardenis 19	111111111111	Mirties įrašas	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Merkys" firmoje
20	Vardenis 20	Pavardenis 20	111111111111	Negalioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Žvejys" firmoje
21	Vardenis 21	Pavardenis 21	111111111111	Galioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Mokosi
22	Vardenis 22	Pavardenis 22	111111111111	Galioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Žvejys" firmoje
23	Vardenis 23	Pavardenis 23	111111111111	Mirties įrašas	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Merkys" firmoje
24	Vardenis 24	Pavardenis 24	111111111111	Negalioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Mokosi
25	Vardenis 25	Pavardenis 25	111111111111	Galioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Žvejys" firmoje
26	Vardenis 26	Pavardenis 26	111111111111	Mirties įrašas	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Merkys" firmoje
27	Vardenis 27	Pavardenis 27	111111111111	Negalioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Žvejys" firmoje
28	Vardenis 28	Pavardenis 28	111111111111	Galioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Mokosi
29	Vardenis 29	Pavardenis 29	111111111111	Negalioja	Namo Nr. 1, Gatve 1, Gyvenvietė, Miestas	Dirba "UAB Žvejys" firmoje

43 pav. Seniūnijos teikiamų viešųjų paslaugų sistemos pagrindinis langas

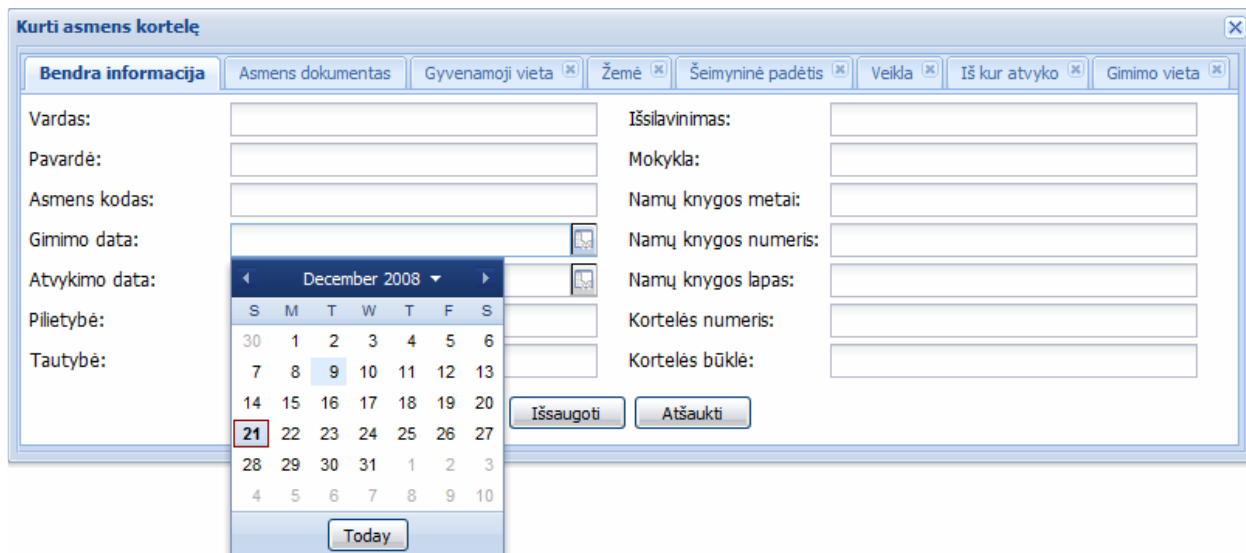
Seniūnijos teikiamų viešųjų paslaugų sistemos pagrindinis meniu detalizuotas 19 lentelėje.

19 lentelė. Seniūnijos teikiamų viešųjų paslaugų sistemos pagrindinio meniu aprašymas

Kategorija	Valdymo meniu	Aprašymas	Vartotojų teisės
Kortelių valdymas	Asmens kortelės	Asmens kortelių kūrimo, redagavimo, šalinimo langas. Mirties priskyrimo.	Redaktorius, Administratoriai
	Namų ūkis	Namų ūkių kūrimo, redagavimo, šalinimo langas.	Redaktorius, Administratoriai
Pažymos	Pažymos apie šeimos sudėtį	Pažymų apie šeimos sudėtį spausdinimo langas ir asmenų atranka.	Redaktorius, Peržiūrėtojas, Administratoriai
	Pažymos apie asmens turtą	Pažymų spausdinimo langas apie asmens turimą turtą. Spausdinamas pavadinimas, detali informacija.	Redaktorius, Peržiūrėtojas, Administratoriai
	Pažymos apie asmens žemę	Pažymų apie asmens žemę spausdinimo langas ir asmenų atranka.	Redaktorius, Peržiūrėtojas, Administratoriai
	Pažymos apie asmens gyvulius	Pažymų spausdinimo langas apie asmens turimus gyvulius. Spausdinamas gyvulių kiekis, pavadinimas.	Redaktorius, Peržiūrėtojas, Administratoriai
	Pažymos apie vienu adresu gyvenančius	Pažymų spausdinimo langas apie asmenis, kurie gyvenu vienu adresu. Pateikiama detali informacija apie kiekvieną asmenį.	Redaktorius, Peržiūrėtojas, Administratoriai
Ataskaitos	Gyventojų sąrašai	Ataskaitų lange pateikiami gyventojai ir jų detali informacija (vardas, pavardė, asmens kodas, gimimo data).	Redaktorius, Peržiūrėtojas, Administratoriai
	Namų ūkių sąrašai	Namų ūkių ataskaitų lange vaizduojami namų ūkiai ir jų detali informacija (socialiai remtina ir t.t.).	Redaktorius, Peržiūrėtojas, Administratoriai
	Įrašymo kortelių sąrašai	Įrašymo kortelių ataskaitų lange vaizduojamos filtruotos įrašymo kortelės ir duomenys apie kiekvieną asmenį.	Redaktorius, Peržiūrėtojas, Administratoriai
	Mirties aktų sąrašai	Mirties aktų ataskaitų lange vaizduojami išfiltruoti mirties aktai.	Redaktorius, Peržiūrėtojas, Administratoriai
Nustatymai	Rekvizitai	Seniūnijos rekvizitai (pavadinimas, kontaktiniai duomenys)	Administratoriai
	Automatinis užpildymas	Seniūnijos adreso užpildymo vieta.	Administratoriai
	Pažymų numeravimas	Pažymų numeravimas susideda iš trijų skaičių. Atskiros pažymos vidurinis skaičius skiriasi.	Administratoriai
	Administravimas – grupės	Programos administravimo langas skirtas grupių kūrimui, bei grupių teisių priskyrimui.	Administratoriai
	Administravimas – vartotojai	Programos administravimo langas skirtas vartotojų sukūrimui ir grupių priskyrimui.	Administratoriai
	Administravimas – DB kopija	Programos administravimo langas skirtas duomenų bazės kopijai sudaryti. Taip pat atstatyti prarastus duomenis.	Administratoriai

5.6. Seniūnijos teikiamų viešųjų paslaugų sistemos asmens kortelių kūrimas

Pagrindinio valdymo meniu punkte aktyvavus asmens kortelių langą valdymų juostoje susikuria 3 mygtukai: kurti kortelę, šalinti kortelę ir priskirti mirtį. Paspaudus „kurti kortelę“ mygtuką iššoka vidinis sistemos „kurti asmens kortelę“ langas. Šis langas pateiktas 44 paveiksle.



44 pav. Seniūnijos teikiamų viešųjų paslaugų sistemos asmens kortelių kūrimas

Kortelės kūrimo langą sudaro aštuonios kortelės:

- bendra informacija – detalesnė kuriamo asmens informacija (vardas, pavardė, asmens kodas, gimimo data, išsilavinimas, pilietybė ir t.t.);
- asmens dokumentas – pateikto dokumento duomenys (dokumento tipas, serija, numeris);
- gyvenamoji vieta – vieta, kur asmuo gyvena pastarąjį laikotarpį (teritorijų klasifikatoriaus langas);
- žemė – sąrašas turimų žemės sklypų (sklypo paskirtis, dydis);
- šeimyninė padėtis – vieta, kur pildoma apie šeimyninės padėties statusą (tipas, aprašymas);
- veikla – vieta, kur pildomas veiklos aprašymas (tipas, aprašymas);
- iš kur atvyko – vieta, iš kurios atvyko asmuo (teritorijų klasifikatoriaus langas);
- gimimo vieta – vieta, kurioje gimė asmuo (teritorijų klasifikatoriaus langas).

Užpildžius kortelės kūrimo langą spaudžiame „Išsaugoti mygtuką“. Tokiu būdu sukuriama asmens kortelė.

5.7. Seniūnijos teikiamų viešųjų paslaugų sistemos suformuoti dokumentai

Realizuota sistema turi integruotas 2 tipų diagramas: pažymos ir ataskaitos. Pagal šiuos diagramų tipus galima susikurti papildomus (filtruotus) dokumentus. Standartinėje programos versijoje sukurtos pažymų specifinės diagramos pavaizduotos 20 lentelėje.

20 lentelė. Seniūnijos teikiamų viešųjų paslaugų sistemos standartinių dokumentų sąrašas

Pažymos	Ataskaitos
pažymos apie šeimos sudėtį pažymos apie asmens turtą pažymos apie asmens žemę pažymos apie asmens gyvulius pažymos apie vienu adresu gyvenančius	gyventojų sąrašai namų ūkių sąrašai įrašymo kortelių sąrašai mirties aktų sąrašai

Sistemos vartotojas kurdamas ataskaitas nurodo specifinius filtrus, numatytus pačios programos. Šie filtrai priklauso nuo dokumento tipo. Seniūnijose dažniausiai prireikia ataskaitų, kuriuose būtų išvedamas nepilnamečių asmenų sąrašas. Kuriant tokią ataskaitą nurodomas amžiaus parametras, kuris yra mažesnis už 18.

Suformuotas dokumentas pateikiamas naršyklės lange. 45 paveiksle pavaizduotas seniūnijos teikiamų viešųjų paslaugų sistemos pažymos apie šeimos sudėtį langas.

SENTŪNIJOS PAVADINIMAS

Bendra informacija apie seniūnijos kontaktus

PAŽYMA
APIE ŠEIMOS SUDĖTĮ

	Frefiksas:	Numeris:	Nomenklatura:
2008.11.21 Nr.		1	

Pažymima, kad Vardenis 1 Pavardenis gimęs 1983.12.11, ak 1111111111, gyvena Namu Nr. 1, Gatve 1, Gyvenvietė 1, Miestas 1. Šeimą sudaro:

Lentele

PAGRINDAS: m. namų ūkio knyga Nr.,

45 pav. Seniūnijos teikiamų viešųjų paslaugų sistemos pažymos langas

5.8. Nuolatinio imitacinio optimizavimo taikymo seniūnijos teikiamų viešųjų paslaugų sistemai rezultatai

Eksperimentinis nuolatinio imitacinio optimizavimo tyrimas buvo atliktas penkiose seniūnijose, esančiose Šiaulių rajone. Šiose seniūnijose žmonių srautai nebuvo itin dideli, išskyrus vieną Šiaulių Kaimiškąją seniūniją. Programinė įranga buvo įdiegta jau prieš keletą metų, atnaujintas tik programos kodas, kuris suteikė nuolatinio optimizavimo funkciją.

Analizuojant sistemos rodiklius, pastebėti didžiausi nukrypimai nuo numatytų parametrų didelį klientų srautą aptarnaujančioje seniūnijoje. Asmens duomenų atranka duomenų bazėje trukdavo trigubai ilgiau nei numatyta, dėl to tai nulemdavo mažesnę suteikiamų paslaugų skaičių per laiko vienetą. Nutarta detaliau išanalizuoti aptarnavimo trukmės charakteristiką. Rezultatų suvestinė pateikta 21 lentelėje.

21 lentelė. Procesų trukmė

Procesas	Darbuotojų skaičius	Įėjimo taškai, k/h	Būsena	Pokytis, s
Užsakymo priėmimas	5	20	Gera	13
Prisijungimas prie sistemos	5	20	Gera	14
Užsakymo įrašymas į duomenų bazę	5	20	Gera	10
Dokumento atranka duomenų bazėje	5	20	Gera	32
Dokumento kūrimas	5	20	Gera	21
Asmens atranka duomenų bazėje	5	20	Kritinė būsena	646
Asmens kūrimas	5	20	Gera	10
Dokumentų formavimas ir spausdinimas	5	20	Gera	41
Pristatymo užsakymas	5	20	Gera	48
Užsakymo įrašymo atnaujinimas duomenų bazėje	5	20	Gera	23

Išanalizavus kritinės užduoties vykdymą nustatyta, kad duomenų atrankos rezultatų pateikimą lėtina duomenų bazė, net jei ji nėra apkrauta. Patikrinus duomenų bazės struktūrą pastebėta, kad įsivėlusį projektavimo klaidą. Lentelė „a_dokumentas“ nėra indeksuojama. Ištraukiant duomenis naudojamas lentelių apjungimas (angl. LEFT JOIN, RIGHT JOIN, INNER JOIN). Todėl duomenų bazė rezultatą gražina ilgiau nei numatyta. 46 paveiksle pateiktos lentelės, jungiančios bendrus duomenis.

asmenys	a_dokumentas
-zid	-id
-akodas	-zid
-pavarde	-lr
-vardas	-pavadinimas
-lytis	-serija
-gime	-numeris
-gvietas	-data
-tautybe	-vieta
-pilietybe	-pastaba

46 pav. Lentelės, jungiančios bendrus duomenis

Išvados

1. Išanalizavus šiuolaikinius imitavimo ir optimizavimo metodus ir įrankius nustatyta, kad vien imitaciniu modeliavimu ar optimizavimu yra sunku nustatyti, ar procesas yra optimalus. Be to, proceso aplinka nuolat kinta, dėl to naudinga taikyti nuolatinio imitacinio optimizavimo metodus, kad proceso optimizavimas vyktų visą laiką.

2. Nustatyta, kad norint vykdyti nuolatinį imitacinį optimizavimą reikalinga keturių komponentų sistema: modeliavimo įrankis, integratorius, stebėjimo ir realizacinė sistema.

3. Išnagrinėjus taikomųjų programų kūrimo technologijas nustatyta, kad yra perspektyvu perkelti funkcijų vykdymą į sistemos vartotojo kompiuterio darbalaukį. Tai suteikia nemažai privalumų, lyginant su kitomis šiam tikslui naudojamomis technologijomis.

4. Atlikus veiklos procesų valdymo produktų analizę nustatyta, kad darbui su šiais produktais reikalingos išlaidos ir žmonių išteklių yra labai dideli. Siekiant to išvengti, pasirinktas mažiausiai trūkumų turintis IBM “Business Process Modeler” įrankis.

5. Sudarytas nuolatinio imitacinio optimizavimo sistemos modelis, siejantis IBM “Business Process Modeler” optimizatorių su optimizuojamo proceso duomenų struktūromis. Tai leidžia efektyviau prisitaikyti prie nuolat kintančių sąlygų. Nuolatiniam optimizavimui realizuoti sukurti optimizatorių papildantys integravimo ir stebėjimo moduliai.

6. Sukurtas seniūnijos teikiamų viešųjų paslaugų modelis ir atliktas imitacinis optimizavimas. Nustatyta, kad užsakymo priėmimas trunka didžiąją dalį viso paslaugos vykdymo laiko. Nuspręsta supaprastinti tvarką priimant kliento paslaugos užsakymus automatizuojant prašymo rašymo etapą.

7. Pritaikius nuolatinio imitacinio optimizavimo metodiką seniūnijos teikiamų viešųjų paslaugų veiklai optimizuoti, nustatyta, kad optimizuotas procesas reikalauja ženkliai mažiau žmogiškųjų ir materialių resursų. Didėja atliekamų paslaugų greitis, mažėja eilės, darbuotojų apkrovimas. Atsiranda galimybė aptarnauti kokybiškiau, teikti papildomas paslaugas.

8. Pritaikius nuolatinio imitacinio optimizavimo metodiką, seniūnijos viešųjų paslaugų teikimo lygis pasikeičia iš trečio į ketvirtąjį, kai vykdoma veikla yra ne tik visiškai kompiuterizuota, bet ir jos rodikliai yra prognozuojami.

Literatūros sąrašas

- [1] April, J., J. Kelly, F. Glover and M. Laguna (2003) "Practical Introduction to Simulation Optimization" in Proceedings of the 2003 Winter Simulation Conference, S. Chick, T. Sanchez, D. Ferrin and D. Morrice, (eds.) pp. 71-78.
- [2] Andradóttir, S. (1998) "A Review of Simulation Optimization Techniques," in Proceedings of the 1998 Winter Simulation Conference, D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan (eds.), pp. 151-158.
- [3] Barnett, M. (2003) "Modeling & Simulation in Business Process Management," BP Trends Newsletter, White Paper & Technical Briefs, pp. 1-10.
- [4] DeFee, J. and P. Harmon (2004) "Business Activity Monitoring and Simulation," BP Trends Newsletter, White Paper and Technical Briefs, pp. 1-24.
- [5] Fu, M. (2002) "Optimization for Simulation: Theory and Practice," INFORMS Journal on Computing, vol. 14, no. 3, pp. 192-215.
- [6] Jay April, Marco Better, Fred Glover, James P. Kelly, Manuel Laguna OptTek Systems, Inc. „Enhancing Business Process Management With Simulation Optimization“. BPTrends, 2005 Sausis.
- [7] Ueli Wahli, Larissa Leybovich, Eric Prevost, Russell Scher, Andre Venancio, Sascha Wiederkom, Neil MacKinnon „*Business Process Management: Modeling through Monitoring Using WebSphere V6 Products*“. 1-asis leid., NC: IBM Corporation, International Technical Support Organization. Prieiga per internetą: www.redbooks.ibm.com. SG24 –7148-00
- [8] Andrew Muller „Building Your First Flex-based Adobe AIR Application“. Prieiga per internetą: http://labs.adobe.com/wiki/index.php/AIR:Articles:Building_Your_First_Flex-based_Adobe_AIR_Application
- [9] Ben Stucki „Working with Binary Data“. Prieiga per internetą: http://labs.adobe.com/wiki/index.php/AIR:Articles:Working_with_Binary_Data
- [10] Dainius Jurčikonis, Aurimas Paršonis, Egidijus Kazanavičius „Duomenų Integravimas. 2006“. Prieiga per internetą: www.ktu.lt
- [11] Deanna Bradshaw, Mark Kennedy, Craig West „Oracle BPEL Process Manager Developer’s Guide, 10g Release 2 (10.1.2)“. Prieiga per internetą: <http://download.oracle.com/otndocs/products/bpel/bpeldev.pdf>
- [12] Duane Nickull „Making Custom Chrome AIR Applications“. Prieiga per internetą: <http://technoracle.blogspot.com/2007/07/making-custom-chrome-air-applications.html>

- [13] Edward Mansouri „Using HTML in Flex-based Adobe AIR Applications“. Prieiga per internetą: http://labs.adobe.com/wiki/index.php/AIR:Articles:Using_HTML_in_Flex-based_Adobe_AIR_Applications
- [14] Jay April, Marco Better, Fred Glover, James P. Kelly, Manuel Laguna, OptTek Systems Inc. „Enhancing Business Process Management With Simulation Optimisation“. Prieiga per internetą: www.bptrends.com
- [15] John C. Bland II „Taking Adobe AIR Applications Offline“. Prieiga per internetą: http://labs.adobe.com/wiki/index.php/AIR:Articles:Taking_Apollo_Applications_Offline
- [16] Inga Pašilskytė, Lina Nemuraitė, Eugenijus Ambrazevičius „Extending uml 2.0 for design of e-business processes“. 2005, birželis.
- [17] Lucas Adamski „Introducing the Adobe AIR security model“. Prieiga per internetą: http://www.adobe.com/devnet/air/articles/introduction_to_air_security.html
- [18] Martin Keen, Jonathan Cavell, Sarah Hill, Chee Keong Kee, Wendy Neave, Bradley Rumph, Hoang Tran „BPEL4WS Business Processes with WebSphere Business Integration: Understanding, Modeling, Migrating“. Prieiga per internetą: www.redbooks.ibm.com. SG24-6381-00
- [19] Microsoft Corporation „BizTalk Server 2006 Tutorials. Application Deployment & Management. Business Activity Monitoring“. Prieiga per internetą: <http://www.microsoft.com/downloads/details.aspx?familyid=6a5f6ef4-aeb8-4d8d-a521-37333a875ce4&displaylang=en#Instructions>
- [20] Oracle® Business Process Architect. Quick Start Guide. Release 10.1.3.3. Prieiga per internetą: http://download.oracle.com/otn/nt/ias/10133/bpa/OracleBPASuite_10133_quick_start_guide_business_architect.pdf
- [21] Oracle BPA Suite. Business Process Architect. 10.1.3.3. Administration Guide. 2007 Kovas. Prieiga per internetą: http://download-llnw.oracle.com/otn/nt/ias/10133/bpa/OracleBPASuite_10133_administration_guide.pdf
- [22] Veiklos procesų kūrimo paplitimas. 2007. Prieiga per internetą: <http://www.biznews.lt/?psl=naujiena&id=6742>
- [23] Studija „Elektroninės valdžios Lietuvoje būklė ir perspektyvos“. Prieiga per Internetą: www.lrinka.lt/uploads/files/dir19/17_0.php

Priedai

1. Stebējimo sistemos naudojama XML transformacija (1 dalis iš 2)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:jbpm="urn:jbpm.org:jpd1-3.2"
  xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:svg="http://www.w3.org/2000/svg">
  <xsl:output omit-xml-declaration="yes" method="xml" indent='yes' />
  <xsl:template match="jbpm:process-definition">
    <svg:g id='drawing'>
      <xsl:attribute name="name"><xsl:value-of select="@name"/></xsl:attribute>
      <xsl:for-each select="//jbpm:transition">
        <svg:line type="line" x1="57" y1="57" x2="93" y2="93" fill-opacity="1"
          stroke="black" stroke-width="1px" style="stroke-width:1; stroke-linecap:butt;"
          onmousedown="LineClicked(evt)" marker-end="url(#endArrow)" >
          <xsl:variable name="Parent" select="parent::*"/>
          <xsl:variable name="ToId" select="@to"/>
          <xsl:variable name="ToObj" select="//jbpm:*[@name=$ToId]"/>
          <xsl:attribute name="from" ><xsl:value-of select="$Parent/@name"/></xsl:attribute>
          <xsl:attribute name="x1" ><xsl:value-of select="$Parent/@x"/></xsl:attribute>
          <xsl:attribute name="y1" ><xsl:value-of select="$Parent/@y"/></xsl:attribute>
          <xsl:attribute name="to" ><xsl:value-of select="$ToObj/@name"/></xsl:attribute>
          <xsl:attribute name="x2" ><xsl:value-of select="$ToObj/@x"/></xsl:attribute>
          <xsl:attribute name="y2" ><xsl:value-of select="$ToObj/@y"/></xsl:attribute>
          <xsl:attribute name="id" ><xsl:value-of select="$Parent/@name"/>_
          <xsl:value-of select="position()" /></xsl:attribute>
        </svg:line>
      </xsl:for-each>
      <xsl:for-each select=".*">
        <svg:use xlink:href="#bubbleTemplate" onmousedown="Clicked(evt)" onmouseup="UnClicked(evt)" >
          <xsl:attribute name="id"><xsl:value-of select="@name"/></xsl:attribute>
          <xsl:attribute name="type" >
            <xsl:value-of select="local-name(.)"/>
          </xsl:attribute>
          <xsl:attribute name="xlink:href" >
            <xsl:text>#</xsl:text>
            <xsl:value-of select="local-name(.)"/>
            <xsl:text>Template</xsl:text>
          </xsl:attribute>
          <xsl:attribute name="x" ><xsl:value-of select="@x"/></xsl:attribute>
          <xsl:attribute name="y" ><xsl:value-of select="@y"/></xsl:attribute>
          <svg:text font-size="16" style="text-anchor: middle" y="30">
            <xsl:value-of select="@name"/>
          </svg:text>
          <svg:rect x="-150" width="300" height="100" style="fill:none;"/>
          <xsl:for-each select=".*">
            <xsl:if test="local-name(.)!='transition'">
              <xsl:copy-of select="."/>
            </xsl:if>
          </xsl:for-each>
        </svg:use>
      </xsl:for-each>
      <svg:text id="textLineAddMode" font-size="20" fill="red" x="20" y="20">
        Linijos pridėjimo fazė
      </svg:text>
    </svg:g>
  </xsl:template>
  <xsl:template match="svg:svg">
    <jbpm:process-definition>
      <xsl:attribute name="name"><xsl:value-of select="svg:g[@id='drawing']/@name"/></xsl:attribute>
      <xsl:for-each select="//svg:use">
        <xsl:variable name="nodeName">
          <xsl:choose >
            <xsl:when test="./action">
              <xsl:text>node</xsl:text>
            </xsl:when>
            <xsl:otherwise>
              <xsl:value-of select="@type"/>
            </xsl:otherwise>
          </xsl:choose >
        </xsl:variable>
      </xsl:for-each>
    </jbpm:process-definition>
  </xsl:template>
</xsl:stylesheet>
```

2. Stebėjimo sistemos naudojama XML transformacija (2 dalis iš 2)

```
        <xsl:value-of select="@type"/>
    </xsl:otherwise>
</xsl:choose>
</xsl:variable>
<xsl:element namespace="urn:jbpm.org:jpd1-3.2" name="{ $nodeName }">
    <xsl:variable name="id" select="@id"/>
    <xsl:attribute name="name" ><xsl:value-of select="$id"/></xsl:attribute>
    <xsl:attribute name="x" ><xsl:value-of select="@x"/></xsl:attribute>
    <xsl:attribute name="y" ><xsl:value-of select="@y"/></xsl:attribute>
    <xsl:for-each select="./action">
        <xsl:copy-of select="."/>
    </xsl:for-each>
    <xsl:for-each select="//svg:line[@from=$id]">
        <jbpm:transition>
            <xsl:attribute name="to"><xsl:value-of select="@to"/></xsl:attribute>
        </jbpm:transition>
    </xsl:for-each>

</xsl:element>
</xsl:for-each>
<xsl:for-each select="//svg:line">
</xsl:for-each>
</jbpm:process-definition>
</xsl:template>
</xsl:stylesheet>
```

3. Integratoriaus naudojama XML transformacija

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:wbim="http://www.ibm.com/wbim/bomSchema1.0">
  <xsl:output omit-xml-declaration="no" method="xml"/>
  <xsl:template match="wbim:model">
    <xsl:element name="model">
      <xsl:call-template name="makeDom">
        <xsl:with-param name="parent" select="."/>
      </xsl:call-template>
    </xsl:element>
  </xsl:template>
  <xsl:template name="makeDom">
    <xsl:param name="parent"/>
    <xsl:for-each select="child::*">

      <xsl:if test="((local-name() != 'inputCriterion') and (local-name() != 'outputCriterion')
        and (local-name() != 'timeEstimation')) and
        ((local-name($parent) != 'flowContent') or (local-name($parent) = 'flowContent'))
        and ((local-name() = 'task') or (local-name() = 'decision') or (local-name() = 'join')) )">
        <xsl:element name="{local-name()}">
          <xsl:if test="local-name() != 'catalogs'">
            <xsl:for-each select="@*">
              <xsl:if test="(name()='name') or (name()='type') or (name()='currency') or (name()='timetable')
                or (name()='beginingOn') or (name()='repetitionPeriod') or (name()='associatedData'))">
                <xsl:attribute name="{name()}"><xsl:value-of select="."/></xsl:attribute>
              </xsl:if>
            </xsl:for-each>

            <xsl:value-of select="text()"/>
            <xsl:if test="((local-name() != 'task') and (local-name() != 'decision') and (local-name() != 'join')
              and (local-name() != 'timetable'))">
              <xsl:call-template name="makeDom">
                <xsl:with-param name="parent" select="."/>
              </xsl:call-template>
            </xsl:if>
          </xsl:if>
        </xsl:element>
      </xsl:if>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```