

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

INFORMACIJOS SISTEMŲ KATEDRA

Edminas Vrubliauskas

VEIKLOS PROCESO MODELIO, PRAPLĖSTO  
VEIKLOS TAISYKLĖMIS, TRANSFORMAVIMO Į  
VEIKLOS PASLAUGAS TYRIMAS

Magistro darbas

Darbo vadovas

prof. dr. R. Butleris

Konsultantė

dokt. L. Tutkutė

KAUNAS, 2012

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

INFORMACIJOS SISTEMŲ KATEDRA

Edminas Vrubliauskas

Veiklos proceso modelio, praplėsto veiklos taisyklėmis,  
transformavimo į veiklos paslaugas tyrimas

Magistro darbas

Recenzentė

lekt.dr. Audronė Janavičiūtė  
2012-05-28

Darbo vadovas

prof. dr. Rimantas Butleris  
2012-05-28

Konsultantė

dokt. Lina Tutkutė  
2012-05-28

Darbą atliko:

IFM-0/4 gr. stud.  
Edminas Vrubliauskas  
2012-05-28

KAUNAS, 2012

## Summary

In today's rapidly evolving world, one of the most important resource of organizations is the time. The time cost of the activities that determine the organization's competitiveness and vulnerability. Organizations increasingly use information systems and want to quickly and efficiently create. Most importantly, seek to improve the organization's activities and to meet the requirements and help achieve our goals. Organizations are able to manage themselves and model their own processes. Thus, they can shape the requirements for the information system. However, the IS developers taking over the organization consisting of process models do not provide the full match. Because creating an information system, they can not use models and concluded the information contained therein. This should eliminate the gap between business process model transformation of the operational services. Manual transformation requires specific knowledge and tangible time-consuming. Possible reasons for the success made Generation: a lack of knowledge of specific, time-consuming limitations of price and programming code errors; cases translation differences, other external factors. To reduce the risk, it is necessary to automate the business process model, extended business rules, transformation of business services. This will ensure consistency of performance rendering process in the form of program code, speeding up the transformation of (reduced time cost) and to ensure, conveyed by the process, transforming fullness of detail and increased security.

Process-oriented activity is to coordinate tasks and activities carried out, involving both business and IT professionals and facilities in order to implement a particular organization or unit of a systemic goal.

The analytical part of the activities carried out in the standards process modeling analysis showed that the optimal choice – the Business Process Modeling Notation (*BPMN*). This choice is based on this principle, that the standard focuses on business process modeling and has probably the richest arsenal of disposable items they define. Analysis will also examine the operating rules of the award and performance standards for service development. Choice of *SBVR* (rules of the contract) and *WSDL* (operational services) standards.

The evaluation results of the analysis, the proposed business process model, extended business rules, transformation of the business service method, and concluded the transformation algorithm. Presented transformation method and transformation of the basic steps required for the storage of information class diagram.

The paper describes the transformation process is algorithmically. Transformation is used to store data required for the *XML* document, ie entrant data flow diagram of the *BPMN* business process and read the result after the transformation (business services) to the user in

*XML* format. With the nomination method is typical of other measures to avoid the negative factors (availability of simplicity, flexibility, cost, operational complexity of the absorption) effect. The method supports a mixed transformation process, which involves transforming the user and control system.

The proposed method is realized in an online environment, thus transforming the user to use the system enough to control Internet browser and Internet access. Transformation management system is designed according to the created transformation algorithm, implemented using *PHP*, *HTML* and *CSS* programming languages and *MySQL* database management system.

The object of research: business process model, the fixed business rules, transformation of business services.

Effective and efficient business process is possible only when the proper use of resources. The upgrade in technology fail to optimize and speed up the process, looking for new methods to effectively convey the process - business process model. The paper analyzes the study areas - business process model, limited to business rules, and its transformation into operational service.

The aim: to extend the transformation of business process diagrams into executable code (business services), thereby enhancing the computerized processes in compliance with real-world processes.

Objectives:

1. Analyze business process modeling standards.
2. To find out the business process model transformation services to the business opportunities.
3. Develop business process model, extended business rules, business transformation services to the algorithm.
4. To create a prototype based on the method comprising transformation.
5. Test a prototype and determine the usefulness of the method.

## Terminų ir santrumpų žodynis

BMPN – veiklos procesų modeliavimo notacija (*angl. The Business Process Modeling Notation*);

UML – unifikauta modeliavimo kalba (*angl. Unified Modeling Language*);

Workflow – darbo sekos modelis (*angl. Workflow*);

SOAP – paprastas objekto ryšių protokolas (*angl. Simple Object Access Protocol*);

XML – *Extensible Markup Language*;

BPEL4WS – *Business Process Execution Language for Web Services*;

BPML – *Business Process Modeling Language*;

WSDL – veiklos paslaugų aprašymo kalba (*angl. The Web Services Description Language*);

UDDI – *Universal Description Discovery and Integration*;

ATL – aktyvi šablonų biblioteka (*angl. The Active Template Library*);

QVT – *Query/View/Transformation*;

CORBA – *The Common Object Request Broker Architecture*.

# Turinys

SUMMARY .....	3
TERMINŲ IR SANTRUMPŲ ŽODYNĖLIS .....	5
PAVEIKSLŲ SĄRAŠAS .....	8
LENTELIŲ SĄRAŠAS .....	10
1. ĮVADAS .....	11
2. VEIKLOS PROCESO MODELIO TRANSFORMACIJOS Į VEIKLOS PASLAUGAS TYRIMO ANALIZĖ .....	13
2.1. ANALIZĖS TIKSLAS .....	13
2.2. MDA .....	14
2.3. VEIKLOS PROCESŲ KONCEPCIJA .....	15
2.3.1. VEIKLOS PROCESŲ MODELIAVIMO METODŲ ANALIZĖ .....	15
2.3.2. <i>Veiklos procesų modeliavimo notacija (BPMN)</i> .....	16
2.3.3. <i>Unifikuota modeliavimo kalba (UML)</i> .....	17
2.3.4. <i>Darby sekos modelis</i> .....	18
2.3.5. <i>Veiklos procesų modeliavimo metodų palyginimas</i> .....	19
2.4. VEIKLOS PASLAUGŲ KŪRIMO METODAI .....	20
2.4.1. <i>CORBA standartas</i> .....	20
2.4.2. <i>Veiklos paslaugų aprašymo kalba (WSDL)</i> .....	21
2.4.3. <i>Paprastas objekto ryšių protokolas (SOAP)</i> .....	23
2.4.4. <i>Veiklos paslaugų kūrimo metodų palyginimas</i> .....	24
2.5. TRANSFORMACIJOS METODAI .....	24
2.5.1. <i>QVT transformacijos standartas</i> .....	25
2.5.2. <i>Aktyvi šablonų biblioteka (ATL)</i> .....	25
2.6. ESAMŲ SPRENDIMŲ ANALIZĖ .....	26
2.6.1. <i>Borland Together sistema</i> .....	26
2.6.2. <i>Visual Paradigm for UML sistema</i> .....	27
2.6.3. <i>UModel sistema</i> .....	28
2.6.4. <i>Transformacijos iš veiklos proceso modelio į veiklos paslaugas esamų sprendimų palyginimas</i> .....	29
2.7. ANALIZĖS IŠVADOS .....	30
3. VEIKLOS PROCESO MODELIO TRANSFORMAVIMO Į VEIKLOS PASLAUGAS ALGORITMO SUDARYMAS... 31	
3.1. VEIKLOS PROCESO MODELIO TRANSFORMACIJOS Į VEIKLOS PASLAUGAS ALGORITMAS .....	31
3.2. BPMN ELEMENTŲ HIERARCHIJA .....	33
3.3. 1 ŽINGSNIS: „PARUOŠTI VEIKLOS PROCESO MODELĮ TRANSFORMAVIMUI“ .....	35
3.4. 2 ŽINGSNIS: TRANSFORMUOTI VEIKLOS PROCESO MODELĮ Į VEIKLOS PASLAUGAS .....	36
3.4.1. <i>2.1-2.3 žingsniai: Modelio paruošimas transformavimui</i> .....	37
3.4.2. <i>2.4 žingsnis: Suformuoti veiklos paslaugų karkasą</i> .....	37
3.4.3. <i>2.5 žingsnis: Suformuoti modelyje naudojamus elementus į eigos seką, pagal proceso vykdymo srauto eigą</i> .....	40
3.4.4. <i>2.6 žingsnis: Identifikuoti egzistuojančias, modelyje panaudotas, elementų struktūras (elementų struktūros apibrėžtos iš anksto)</i> .....	40
3.4.5. <i>2.7 žingsnis: Suformuoti likusių netransformuotų elementų seką</i> .....	43
3.4.6. <i>2.8 žingsnis: Transformuoti likusius elementus į veiklos paslaugas</i> .....	43
3.4.7. <i>2.9 žingsnis: Pateikti vientisą transformacijos metu sudarytą vykdomąjį kodą</i> .....	49
3.5. 3 ŽINGSNIS: „PATVIRTINTI VEIKLOS PASLAUGAS“ .....	49
4. VEIKLOS PROCESO MODELIO TRANSFORMAVIMO Į VEIKLOS PASLAUGAS SPRENDIMO REALIZACIJA.... 52	
4.1. SPRENDIMO REIKALAVIMŲ SPECIFIKACIJA IR ANALIZĖ .....	52
4.1.1. <i>Dalykinės srities analizė</i> .....	52
4.1.2. <i>Funkciniai reikalavimai</i> .....	54
4.1.3. <i>Nefunkciniai reikalavimai</i> .....	56
4.1.4. <i>Dalykinės srities esybių modelis</i> .....	57
4.1.5. <i>Reikalavimų analizės apibendrinimas</i> .....	57
4.2. SISTEMOS ARCHITEKTŪRA .....	58

4.3	DETALUS PROJEKTAS .....	60
4.4	SISTEMOS ELGSENOS MODELIS .....	61
4.5	DUOMENŲ BAZĖS MODELIS .....	61
4.6	REALIZACIJOS MODELIS.....	64
4.7	VARTOTOJŲ ANALIZĖ.....	66
4.7.1	VARTOTOJŲ AIBĖ, TIPAI IR SAVYBĖS .....	66
4.7.2	VARTOTOJŲ TIKSLAI IR PROBLEMOS .....	67
4.7.3	VERTINIMO KRITERIJAI .....	67
4.8	TESTAVIMO PLANAS IR REZULTATAI.....	68
<b>5.</b>	<b>EKSPERIMENTINIS SISTEMOS TYRIMAS .....</b>	<b>72</b>
5.1.	EKSPERIMENTOS PLANAS .....	72
5.2.	EKSPERIMENTO EIGA.....	74
5.3.	SISTEMOS VEIKIMO IR SAVYBIŲ ANALIZĖ, KOKYBĖS KRITERIJŲ ĮVERTINIMAS .....	79
<b>5.</b>	<b>IŠVADOS.....</b>	<b>81</b>
<b>6.</b>	<b>LITERATŪRA .....</b>	<b>83</b>
<b>7.</b>	<b>PRIEDAI .....</b>	<b>85</b>
7.1.	1 PRIEDAS. BPMN ELEMENTŲ SPECIFIKACIJA .....	85
7.2.	2 PRIEDAS. UML ELEMENTŲ SPECIFIKACIJA .....	87
7.3.	3 PRIEDAS. WORKFLOW ELEMENTŲ SPECIFIKACIJA.....	90
7.4.	4 PRIEDAS. BPMN PAVYZDINIS FAILAS. ....	90
7.5.	5 PRIEDAS. WSDL PAVYZDINIS FAILAS. ....	94
7.6.	6 PRIEDAS. STRAIPSNIS 17-OJE TARPTAUTINĖJE MAGISTRANTŲ IR DOKTORANTŲ MOKSLINĖJE KONFERENCIJOJE „INFORMACINĖS TECHNOLOGIJOS“ .....	95

## Paveikslų sąrašas

1 PAV. MDA TECHNOLOGIJA.....	14
2 PAV. VEIKLOS PROCESO GYVAVIMO CIKLAS.....	15
3 PAV. MAGICDRAW PAKETU SUDARYTA BPMN PROCESS DIAGRAMOS PAVYZDYS.....	17
4 PAV. MAGICDRAW PAKETU SUDARYTAS ACTIVITY DIAGRAMOS PAVYZDYS.....	18
5 PAV. PROVISION PAKETU SUDARYTAS WORKFLOW DIAGRAMOS PAVYZDYS.....	19
6 PAV. CORBA ĮRANKIO INFRASTRUKTŪROS KODO GENERAVIMAS.....	20
7 PAV. JAVA IDL SERVERIO KOMPILIAVIMO PAVYZDYS NAUDOJANT CORBA.....	21
8 PAV. WSDL ARCHITEKTŪRA.....	22
9 PAV. WSDL STRUKTŪRA.....	22
10 PAV. SOAP STRUKTŪRA.....	23
11 PAV. SOAP ARCHITEKTŪRA.....	24
12 PAV. QUERY/VIEW/TRANSFORMATION ARCHITEKTŪRA.....	25
13 PAV. ATL STRUKTŪRA.....	26
14 PAV. ATL TRANSFORMACIJA.....	26
15 PAV. TRANSFORMACIJOS RINKINIŲ APRAŠAS.....	31
16 PAV. TRANSFORMACIJOS ALGORITMO PAGRINDINIAI ŽINGSNIAI.....	31
17 PAV. APIBENDRINTAS METODO METAMODELIS.....	32
18 PAV. PRINCIPINĖ METODO METAMODELIO SCHEMA.....	32
19 PAV. APIBENDRINTAS TRANSFORMACIJOS ALGORITMAS.....	33
20 PAV. BPMN APIBENDRINTAS ELEMENTŲ METAMODELIS.....	34
21 PAV. BPMN PROCESO DIAGRAMA - „EGZAMINŲ SESIJOS PROCESAS“.....	35
22 PAV. TRANSFORMACIJOS ALGORITMO ŽINGSNIO „PARUOŠTI VEIKLOS PROCESO MODELĮ TRANSFORMAVIMUI“ DETALIZAVIMAS.....	36
23 PAV. TRANSFORMACIJOS ALGORITMO ŽINGSNIO „TRANSFORMUOTI VEIKLOS PROCESO MODELĮ Į VEIKLOS PASLAUGAS“ DETALIZAVIMAS.....	36
24 PAV. TRANSFORMACIJOS ALGORITMO ŽINGSNIAI - MODELIO PARUOŠIMAS TRANSFORMAVIMUI.....	37
25 PAV. TRANSFORMACIJOS ALGORITMO ŽINGSNIO „SUFORMUOTI VEIKLOS PASLAUGŲ KARKASĄ“ DETALIZAVIMAS.....	37
26 PAV. „ACTIVITY“ METAMODELIS.....	38
27 PAV. TRANSFORMACIJOS ALGORITMO ŽINGSNIO „SUFORMUOTI MODELyje NAUDOJAMUS ELEMENTUS Į EIGOS SEKĄ, PAGAL PROCESO VYKDYMO SRAUTO EIGĄ“ DETALIZAVIMAS.....	40
28 PAV. TRANSFORMACIJOS ALGORITMO ŽINGSNIO „IDENTIFIKUOTI EGZISTUOJANČIAS, MODELyje PANAUDOTAS, ELEMENTŲ STRUKTŪRAS (ELEMENTŲ STRUKTŪROS APIBRĖŽTOS IŠ ANKSTO)“ DETALIZAVIMAS.....	41
29 PAV. TRANSFORMACIJOS ALGORITMO ŽINGSNIO „SUFORMUOTI LIKUSIŲ NETRANSFORMUOTŲ ELEMENTŲ SEKĄ“ DETALIZAVIMAS.....	43
30 PAV. TRANSFORMACIJOS ALGORITMO ŽINGSNIO „TRANSFORMUOTI LIKUSIUS ELEMENTUS Į VEIKLOS PASLAUGAS“ DETALIZAVIMAS.....	44
31 PAV. „GATEWAY“ METAMODELIS.....	44
32 PAV. „EVENTS“ METAMODELIS.....	47
33 PAV. „ITEMS AND DATA“ METAMODELIS.....	48
34 PAV. TRANSFORMACIJOS ALGORITMO ŽINGSNIS „PATEIKTI VIENTISĄ TRANSFORMACIJOS METU SUDARYTĄ VYDOMĄJĮ KODĄ“.....	49
35 PAV. TRANSFORMACIJOS ALGORITMO ŽINGSNIO „VALIDUOTI VEIKLOS PASLAUGAS“ DETALIZAVIMAS.....	50
36 PAV. VEIKLOS TIKSLŲ MODELIS.....	52
37 PAV. VEIKLOS KONCEPTŲ (OBJEKTŲ) MODELIS.....	53
38 PAV. VEIKLOS PROCESŲ MODELIS.....	54
39 PAV. KOMPIUTERIZUOJAMŲ PANAUDOJIMO ATVEJŲ MODELIS.....	54
40 PAV. PA „TRANSFORMUOTI VEIKLOS PROCESO MODELĮ Į VEIKLOS PASLAUGAS“ SEKŲ DIAGRAMA.....	55
41 PAV. DALYKINĖS SRITIES ESYBIŲ KLASIŲ MODELIS.....	57
42 PAV. KONTEKSTINĖ DIAGRAMA.....	59
43 PAV. PANAUDOJIMO ATVEJŲ DIAGRAMA.....	59
44 PAV. PA „TRANSFORMUOTI VEIKLOS PROCESO MODELĮ Į VEIKLOS PASLAUGAS“ REALIZACIJOS SEKŲ DAIGRAMA.....	60
45 PAV. VEIKLOS PROCESO MODELIO BŪSENŲ DIAGRAMA ATLIEKANT TRANSFORMACIJĄ Į VEIKLOS PASLAUGAS.....	61
46 PAV. KLASIŲ DIAGRAMA.....	61
47 PAV. DUOMENŲ BAZĖS MODELIS.....	62
48 PAV. SISTEMOS KOMPONENTŲ DIAGRAMA.....	64
49 PAV. SISTEMOS DIEGIMO MODELIS.....	65
50 PAV. NAVIGAVIMO PLANAS.....	66
51 PAV. PRISIJUNGIMAS PRIE SISTEMOS – NEJVESTAS SLAPTAŽODIS.....	70
52 PAV. PRISIJUNGIMAS PRIE SISTEMOS – KLAIDA (NEJVESTAS PRISIJUNGIMO VARDAS IR/ARBA SLAPTAŽODIS).....	70
53 PAV. PRISIJUNGIMAS PRIE SISTEMOS – KLAIDA (NETEISINGI DUOMENYS).....	70
54 PAV. PRISIJUNGIMAS PRIE SISTEMOS – KOREKTIŠKI DUOMENYS (PRISIJUNGIAMA PRIE SISTEMOS).....	70



55 PAV. NAUJO PROJEKTO SUKŪRIMAS .....	70
56 PAV. DIAGRAMOS ĮKĖLIMAS.....	71
57 PAV. TRANSFORMAVIMO PASIRINKIMAS.....	71
58 PAV. VEIKLOS PROCESAS „SU KELETU VEIKLŲ“ SUMODELIUOTAS SU BPMN STANDARTU .....	72
59 PAV. VEIKLOS PROCESAS „SU VIENA VEIKLA“ SUMODELIUOTAS SU BPMN STANDARTU .....	72
60 PAV. VEIKLOS PROCESO „SU KELETU VEIKLŲ“ XML FAILO FRAGMENTAS.....	73
61 PAV. VEIKLOS PROCESO „SU VIENA VEIKLA“ XML FAILO FRAGMENTAS.....	73
62 PAV. DIAGRAMOS PATALPINIMO Į SISTEMĄ VARTOTOJO SĄSAJOS LANGAS.....	74
63 PAV. VEIKLOS PROCESO „SU KELETU VEIKLŲ“ NUSKAITYTŲ ELEMENTŲ SEKA .....	75
64 PAV. VEIKLOS PROCESO „SU VIENA VEIKLA“ NUSKAITYTŲ ELEMENTŲ SEKA .....	75
65 PAV. VEIKLOS PROCESO „SU KELETU VEIKLŲ“ IDENTIFIKUOTŲ BPMN ELEMENTŲ INFORMACIJA DUOMENŲ BAZĖJE .....	75
66 PAV. VEIKLOS PROCESO „SU VIENA VEIKLA“ IDENTIFIKUOTŲ BPMN ELEMENTŲ INFORMACIJA DUOMENŲ BAZĖJE .....	76
67 PAV. VEIKLOS PROCESO „SU KELETU VEIKLŲ“ IDENTIFIKUOTOS ELEMENTŲ GRUPĖS PROGRAMINIO KODO FRAGMENTAS .....	76
68 PAV. VEIKLOS PROCESO „SU KELETU VEIKLŲ“ REZULTATO PATEIKIMAS VARTOTOJUI .....	77
69 PAV. VEIKLOS PROCESO „SU VIENA VEIKLA“ REZULTATO PATEIKIMAS VARTOTOJUI .....	77
70 PAV. BPMN 2.0 DIAGRAMOS ELEMENTAI .....	86
71 PAV. SRAUTO OBJEKTAI (ĮVYKIAI) NAUDOJAMI BPMN 2.0 .....	86
72 PAV. SRAUTO OBJEKTAI (VEIKLOS) NAUDOJAMI BPMN 2.0 .....	86
73 PAV. SRAUTO OBJEKTAI (TINKLŲ SĄSAJOS) NAUDOJAMI BPMN 2.0.....	86
74 PAV. SUJUNGIMO OBJEKTAI NAUDOJAMI BPMN 2.0 .....	87
75 PAV. PLAUKIMO TAKELIAI NAUDOJAMI BPMN 2.0 .....	87
76 PAV. ARTEFAKTAI NAUDOJAMI BPMN 2.0 .....	87
77 PAV. UML DIAGRAMŲ ELEMENTŲ PAVYZDŽIAI MAGICDRAW MODELIAVIMO ĮRANKYJE (ATITINKAMAI IŠ KAIRĖS PUSĖS: ACTIVITY, CLASS, USE CASE IR SEQUENCE DIAGRAMOS) .....	88
78 PAV. WORKFLOW MODELIO ELEMENTAI (PROVISION PAKETE) .....	90
79 PAV. PRINCIPINĖ TRANSFORMAVIMO METODO METAMODELIO DIAGRAMA .....	97
80 PAV. TRANSFORMAVIMO METODO APIBENDRINTA UML VEIKLOS DIAGRAMA .....	97
81 PAV. TRANSFORMAVIMO METODO KLASIŲ DIAGRAMA .....	98

## Lentelių sąrašas

1 LENTELĖ. VEIKLOS PROCESU MODELIAVIMO METODŲ LYGINAMOJI ANALIZĖ .....	19
2 LENTELĖ. VEIKLOS PASLAUGŲ KŪRIMO METODŲ LYGINAMOJI ANALIZĖ .....	24
3 LENTELĖ. ESAMŲ SPRENDIMŲ LYGINAMOJI ANALIZĖ.....	29
4 LENTELĖ. BPMN ACTIVITY (SUB-PROCESS IR TASK) ELEMENTŲ TRANSFORMAVIMAS Į VEIKLOS PASLAUGAS .....	38
5 LENTELĖ. ELEMENTŲ STRUKTŪRŲ TRANSFORMAVIMO GALIMYBIŲ VAIZDAVIMAS.....	41
6 LENTELĖ. BPMN GATEWAY ELEMENTŲ TRANSFORMAVIMO GALIMYBIŲ VAIZDAVIMAS .....	45
7 LENTELĖ. BPMN EVENT ELEMENTŲ TRANSFORMAVIMO GALIMYBIŲ VAIZDAVIMAS .....	47
8 LENTELĖ. BPMN ITEMS AND DATA ELEMENTŲ TRANSFORMAVIMO GALIMYBIŲ VAIZDAVIMAS.....	49
9 LENTELĖ. VEIKLOS TAISYKLIŲ TRANSFORMAVIMO Į VEIKLOS PASLAUGAS APRAŠAS .....	50
10 LENTELĖ. PANAUDOJIMO ATVEJO „TRANSFORMUOTI VEIKLOS PROCESO MODELĮ Į VEIKLOS PASLAUGAS“ SPECIFIKACIJA .....	55
11 LENTELĖ. PANAUDOJIMO ATVEJO „SUDARYTI VEIKLOS PROCESO MODELĮ“ SPECIFIKACIJA .....	56
12 LENTELĖ. PANAUDOJIMO ATVEJO „SUKURTI VYKDOMĄJĮ KODĄ“ SPECIFIKACIJA .....	56
13 LENTELĖ. „VARTOTOJAS“ ATRIBUTŲ SPECIFIKACIJA .....	62
14 LENTELĖ. „PROJEKTAS“ ATRIBUTŲ SPECIFIKACIJA.....	62
15 LENTELĖ. „BPMN_DIAGRAMA“ ATRIBUTŲ SPECIFIKACIJA.....	63
16 LENTELĖ. „BPMN_ELEMENTO_TIPAS“ ATRIBUTŲ SPECIFIKACIJA.....	63
17 LENTELĖ. „BPMN_STRUKTūros_ELEMENTAS“ ATRIBUTŲ SPECIFIKACIJA .....	63
18 LENTELĖ. „BPMN_STRUKTURA“ ATRIBUTŲ SPECIFIKACIJA .....	63
19 LENTELĖ. „BPMN_DIAGRAMOS_ELEMENTAS“ ATRIBUTŲ SPECIFIKACIJA.....	63
20 LENTELĖ. „WSDL_MODELIS“ ATRIBUTŲ SPECIFIKACIJA .....	64
21 LENTELĖ. „VEIKLOS_PASLAUGŲ_STRUKTURA“ ATRIBUTŲ SPECIFIKACIJA .....	64
22 LENTELĖ. KOMPONENTŲ SPECIFIKACIJA .....	65
23 LENTELĖ. VARTOTOJŲ TIKSLAI.....	67
24 LENTELĖ. VARTOTOJAMS IŠSKYLANČIOS PROBLEMAS .....	67
25 LENTELĖ. VERTINIMO KRITERIJŲ LENTELĖ .....	68
26 LENTELĖ. TESTAVIMO PLANAS .....	68
27 LENTELĖ. GAUTI ELEMENTAI PANAUDOTI NUSKAITYTOSE DIAGRAMOSE .....	74
28 LENTELĖ. TRANSFORMAVIMO METODO VERTINIMO LENTELĖ (NAGRINĖJAMAS VEIKLOS PROCESAS - "SU KELETU VEIKLŲ").....	74

## 1. Įvadas

Šiuolaikiniame, sparčiai besivystančiame pasaulyje, vienas iš svarbiausių organizacijų resursų yra laikas. Laiko sąnaudos vykdant veiklas, nulemia organizacijos konkurencingumą ir pažeidžiamumą. Organizacijos vis daugiau naudoja informacines sistemas, nori jas greitai ir efektyviai sukurti. Svarbiausia, siekia, kad jos gerintų organizacijos veiklą ir atitiktų keliamus reikalavimus bei padėtų pasiekti užsibrėžtus tikslus. Organizacijoms yra sudarytos sąlygos pačioms valdyti ir modeliuoti savo procesus. Tokiu būdu jos gali suformuoti reikalavimus kuriamoms informacinėms sistemoms. Vis dėlto, IS kūrėjai, perimdami organizacijos sudarytus procesų modelius, neužtikrina visapusiško atitikimo. Kadangi kurdami informacinę sistemą, jie negali pasinaudoti sudarytais modeliais ir juose esančia informacija. Šį atotrūkį turėtų eliminuoti veiklos proceso modelio transformavimas į veiklos paslaugas. Rankinis transformavimas reikalauja specifinių žinių bei apčiuopiamai didelių laiko sąnaudų. Galimos priežastys, kodėl nesėkmingai vykdomas generavimas: specifinių žinių trūkumas; laiko sąnaudų ribotumas; kaina; programinio kodo klaidos; atvejų interpretuojamumo skirtumai; kiti išoriniai veiksniai. Norint sumažinti riziką, būtina automatizuoti veiklos proceso modelio, praplėsto veiklos taisyklėmis, transformavimą į veiklos paslaugas. Taip bus užtikrintas nuoseklesnis veiklos proceso perteikimas programinio kodo pavidalu, pagreitinamas transformavimas (sumažintos laiko sąnaudos) bei užtikrintas, perteikiamo proceso, transformavimo pilnumas ir padidintas išsamumo užtikrinimas.

Veiklos procesas yra orientuotas koordinuoti užduotis ir atliekamas veiklas, įtraukiant tiek verslo bei IT srities specialistus bei technines priemones, tam kad įgyvendintų konkretų organizacijos ar sisteminio vieneto tikslą.

Analitinėje dalyje atlikta veiklos procesų modeliavimo standartų analizė parodė, kad optimalus pasirinkimas – veiklos proceso modeliavimo notacija (*Business Process Modeling Notation, BPMN*). Šis pasirinkimas grindžiamas tuom, kad standartas orientuotas į veiklos procesų modeliavimą bei turi bene gausiausią disponuojamų elementų arsenalą jiems apibrėžti. Analizės metu taip pat išnagrinėti veiklos taisyklių sudarymo bei veiklos paslaugų kūrimo standartai. Pasirinkta SBVR (taisyklių sudarymo) ir WSDL (veiklos paslaugų kūrimo) standartai.

Įvertinus analizės rezultatus, pasiūlytas veiklos proceso modelio, praplėsto veiklos taisyklėmis, transformavimo į veiklos paslaugas metodas bei sudarytas transformavimo algoritmas. Pateikiama transformavimo metodo pagrindiniai žingsniai bei transformavimo informacijos saugojimui reikalinga klasių diagrama.

Darbe algoritmiškai aprašoma transformavimo procesas. Transformavimui reikalingų duomenų saugojimui naudojamas *XML* dokumentas, t.y. įeinantysis duomenų srautas *BPMN*

veiklos proceso diagrama nuskaitoma bei po transformavimo gautas rezultatas (veiklos paslaugas) vartotojui pateikiama *XML* formatu. Naudojant pasiūlytąjį metodą, yra išvengiama kitoms priemonėms būdingų neigiamų veiksnių (prieinamumo paprastumo, lankstumo, kainos, veikimo įsisavinimo sudėtingumo) įtakos. Metodas palaiko mišrų transformavimo procesą, kuriame dalyvauja vartotojas ir transformavimo valdymo sistema.

Pasiūlytasis metodas realizuotas internetinėje aplinkoje, taip vartotojui norint naudotis transformavimo valdymo sistema užtenka internetinės naršyklės bei prieigos prie interneto. Transformavimo valdymo sistema suprojektuota pagal sukurtąjį transformavimo algoritmą, realizuota naudojant *PHP*, *HTML* bei *CSS* programavimo kalbas bei *MySQL* duomenų bazių valdymo sistemą.

Tyrimo objektas: veiklos proceso modelio, apriboto veiklos taisyklėmis, transformacija į veiklos paslaugas.

Efektyvus ir produktyvus veiklos procesas galimas tik tada, kai tinkamai naudojami ištekliai. Kadangi technologijų atnaujinimas nebepadeda optimizuoti ir sparčiau atlikti procesų, ieškoma naujų metodų efektyviau perteikti procesą – veiklos proceso modelyje. Darbe nagrinėjama tyrimo sritis – veiklos proceso modelis, apribotas veiklos taisyklėmis, ir jo transformavimas į veiklos paslaugas.

Tyrimo tikslas: praplėsti transformaciją iš veiklos procesų diagramos į vykdomą kodą (veiklos paslaugas), tokiu būdu padidinant kompiuterizuojamų procesų atitikimą realaus pasaulio procesams.

Uždaviniai:

1. Išanalizuoti veiklos proceso modeliavimo standartus.
2. Išsiaiškinti veiklos proceso modelio transformavimo į veiklos paslaugas galimybes.
3. Sudaryti veiklos proceso modelio, praplėsto veiklos taisyklėmis, transformavimo į veiklos paslaugas algoritmą.
4. Sudaryti prototipą, pagrįstą sudarytu transformavimo metodu.
5. Ištestuoti prototipą ir nustatyti metodo naudingumą.

## 2. Veiklos proceso modelio transformacijos į veiklos paslaugas tyrimo analizė

### 2.1. Analizės tikslas

Kasmet vis daugėjant įmonių, kurios atlieka eilę darbų, kurių apimtys ir sudėtingumas vis didėja. Norint optimizuoti procesą pasitelkiamas veiklos proceso modelis, tačiau ne visuomet yra tinkamų priemonių ar metodų jį pritaikant. Veiklos paslaugas ta sritis, kurioje yra mažai išvystyta proceso optimizavimo galimybė. Nėra pakankamai aiškių metodų, kaip pasitelkti veiklos proceso modelį.

Analizės tikslas yra išanalizuoti veiklos proceso modeliavimo standartus bei išsiaiškinti veiklos proceso modelio transformavimo į veiklos paslaugas galimybes. Išnagrinėti jau egzistuojančius sprendimus bei identifikuoti jų privalumus bei trūkumus.

Siekiant įgyvendinti šį tikslą, buvo detaliai išnagrinėti ir palyginti (privalumai ir trūkumai) šiam perėjimui, nuo veiklos proceso modelio prie veiklos paslaugų, reikalingos tikslinės priemonės (*BPMN*, *SBVR*, *WebServices* ir kt.). Ištyrinėtos galimybės atlikti transformavimą, išnagrinėta jau egzistuojantys probleminės srities sprendimai bei išskirti ir detalizuoti galimi galutinio produkto vartotojai (vartotojų analizė).

Analitinėje dalyje buvo aiškinamasi ar nėra jau sukurtų analogiškų sprendimų. Analizės metu nustatyta, kad tiek *Borland Together* tiek *VP-UML* tiek *Umodel* licencija yra mokama, darbas su šiais įrankiais atsieis nemažai finansinių kaštų. Taip pat nustatyta, kad nors jie ir palaiko *BPMN* standartą, tačiau labiau skirti darbui su *UML* modeliavimo kalba ir jų suderinamumas su šiuo standartu yra labiau pažengęs į priekį. Išanalizuota ir palyginta tikslinės priemonės, kurių pagalba galima sukurti naują, pakoreguoti ar kitaip modifikuoti jau esamą sprendimą.

Šie sprendimai įvertinti pagal nustatytus kriterijus. Tam tikslui panaudota lyginamosios analizės metodas.

Lyginamoji analizė – tai procesas, kai tarpusavyje palyginami bent du analogiški sprendimai tikslui pasiekti. Jie įvertinami ir palyginami pasitelkiant kritinius sprendimų parametrus nagrinėjančius kriterijus, juos įvertinant atitinkamais įverčiais, atsižvelgiant į kriterijaus pobūdį ir vertinamo sprendimo santykį su galutiniu siektinu įgyvendinti tikslu.

Lyginamosios analizės procesas gali būti detalizuotas tokiais žingsniais:

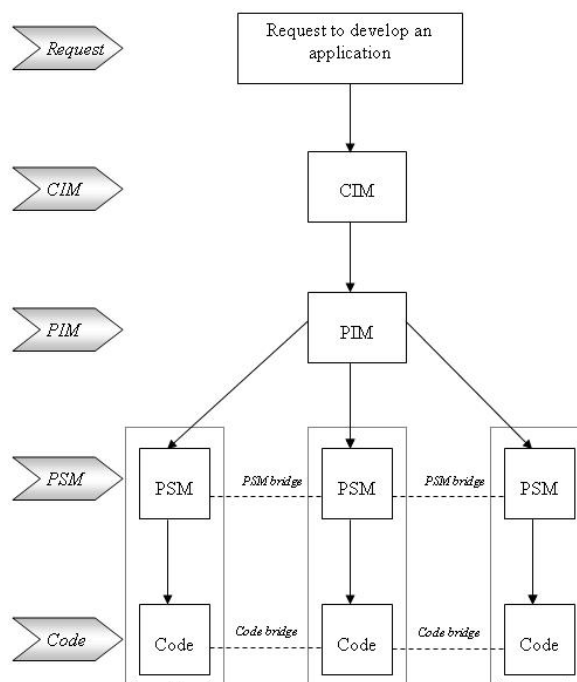
1. Apibrėžti sprendimų palyginimo kontekstą;
2. Atrinkti lygintinus sprendimus ir juos išanalizuoti;
3. Nustatyti esmines lyginamų sprendimų savybes;
4. Įvertinti ir palyginti sprendimus taikant pasirinktą schemą tekstui organizuoti.

Pasitelkiant lyginamosios analizės metodą atliekama:

- Veiklos procesų modeliavimo metodų (*BPMN*, *UML* ir *Workflow*) analizė.
- Veiklos paslaugų kūrimo metodų (*CORBA* ir *WSDL*) analizė.
- Transformacijos iš veiklos proceso modelio į veiklos paslaugas esamų sprendimų (*Together*, *VP-UML* ir *UModel*) analizė.

## 2.2. MDA

Modeliais pagrįsta architektūra (angl. *Model Driven Architecture*, *MDA*) yra *Object Management Group* (*OMG*) sukurtas produktas skirtas programinės įrangos architektūros kūrimui. *MDA* (1 pav.) sudaro keturi žingsniai. Pirmajame žingsnyje yra sudaromas *CIM* (*Computation Independent Model*), nuo skaičiavimų nepriklausomas, modelis. Antrajame žingsnyje *CIM* modelyje sukauptos žinios perduodamos *PIM* (*Platform Independent Model*) modeliui. *PIM* modelis aprašo sistemos funkcionalumą. Tai yra nuo platformos nepriklausantis modelis, kuris atvaizduoja visą veiklos procesą. Gautas *PIM* modelis yra transformuojamas (trečiasis *MDA* koncepcijos žingsnis) į vieną arba keletą *PSM* (*Platform Specific Models*) modelių. *PSM* kitaip nei *PIM* yra orientuotas į sistemos aprašymą bei kaip ji bus realizuota naudojant konkrečią technologiją. *PSM* modeliai skaidomi į tris grupes: reliaciniai modeliai; *EJB* (*Enterprise Java Beans*) modeliai bei *Web* modeliai. Ketvirtasis *MDA* žingsnis – transformuoti sudarytus *PSM* modelius į kodą.



1 pav. MDA technologija

## 2.3. Veiklos procesų koncepcija

Veiklos procesas (*angl. Business process*) yra tarpusavyje susijusių, struktūriškai apibrėžtų veiklų ar užduočių paletė. Jis orientuotas koordinuoti užduotis ir atliekamas veiklas, įtraukiant tiek verslo bei IT srities specialistus bei technines priemones, kad įgyvendintų konkretų organizacijos ar sisteminio vieneto tikslą. Veiklos procesas pagal savo panaudojimo sritis skaidomas į tris dalis: valdymo procesus (*angl. Management processes*); eksploatacinius procesus (*angl. Operational processes*) bei patvirtinamuosius procesus (*angl. Supporting processes*). Veiklos proceso gyvavimo ciklas yra įvairiai interpretuojamas, vienas iš pavyzdžių pateiktas 2 pav.



2 pav. Veiklos proceso gyvavimo ciklas

### 2.3.1. Veiklos procesų modeliavimo metodų analizė

Veiklos procesas [1] yra orientuotas koordinuoti užduotis ir atliekamas veiklas, įtraukiant tiek verslo bei IT srities specialistus bei technines priemones, tam kad įgyvendintų konkretų organizacijos ar sisteminio vieneto tikslą.

Veiklos procesų modeliavimas [2] sistemų ir programinės įrangos inžinerijoje yra veikla, atspindinti procesus vykstančius įmonėje. Tuo siekiama dabartinius procesus tinkamu aspektu išnagrinėti ir pagerinti. Veiklos procesų modeliavimą dažniausiai atlieka analitikai, siekdami pagerinti procesų efektyvumą ir kokybę. Proceso tobulinime dažniausiai dalyvauja IT srities specialistai, nors tai nėra pagrindinis veiksnys sudarant veiklos proceso modelį,

kuris atvaizduodamas sistemos ar organizacijos funkcionalumą, pateikia vizualinį jo proporcijų vaizdą.

Kitas aspektas yra verslo procesų valdymas [3, 23] (*Business Process Management, BPM*), tai yra valdymo metodas skirtas suderinti visus organizavimo aspektus su klientų poreikiais. Šis Holistinis valdymo metodas [4, 22] skatina verslo efektyvumo ir veiksmingumo proporcingą augimą. Taip pat naujovių siekimą, kas lemia integracijos su technologijomis augimą. Teigiama, kad veiklos proceso valdymas leidžia organizacijoms tikslingiau ir efektyviau optimizuoti funkcionalumą ten kur tikslinga.

Galiausiai pereinant prie konkrečių veiklos procesu modeliavimo metodų, analizei pasirinkti veiklos proceso modeliavimo notacija [3, 19, 20, 21] (*Business Process Modeling Notation, BPMN*), unifikauta modeliavimo kalba [5] (*Unified Modeling Language, UML*) ir *Workflow* [6] susidedančio iš susijusių veiklų sekų ir operacijų kompleksinio mechanizmo modelio.

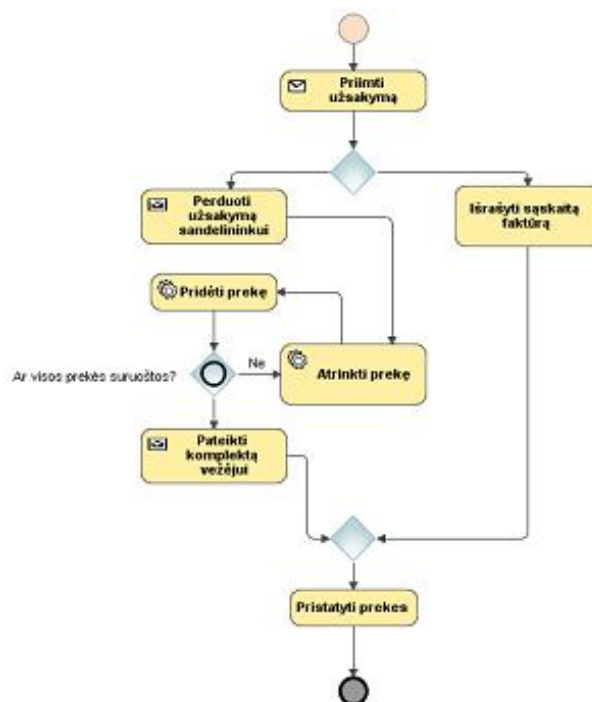
### **2.3.2. Veiklos procesų modeliavimo notacija (BPMN)**

*Veiklos procesų modeliavimo notacija (angl. The Business Process Modeling Notation; BPMN)* [3,19,20] yra *Object Management Group (OMG)* grupės *Business Process Management Initiative (BPMI)* sukurtas naujas standartas, skirtas modeliuoti verslo procesus ir veiklos paslaugų procesus. Pradinis *BPMN* tikslas – pateikti visiems veiklos dalyviams lengvai suprantamą aiškinamąją notaciją. Tai apima veiklos analitikus, kurie perteikia pirminius projekto reikalavimus ir siekiamą funkcionalumą vykdomojo kodo kūrėjams (programuotojams).

Kitas ne ką mažiau svarbus *BPMN* tikslas [21] yra užtikrinti, kad *Extensible Markup Language (XML)* kalbos skirtos veiklos procesų, tokių kaip *Business Process Execution Language for Web Services (BPEL4WS)* ir *Business Process Modeling Language (BPML)* vykdymas būtų vizualiai išreikštas bendra notacija (*BPMN* elementų specifikacija pateikta 1 priede).

*BPMN* procesų diagramos pavyzdys, kuriame pateikta užsakymo įvykdymo procesas nuo jo priėmimo iki pristatymo, pateiktas 3 pav.





3 pav. MagicDraw paketu sudaryta BPMN Process diagramos pavyzdys

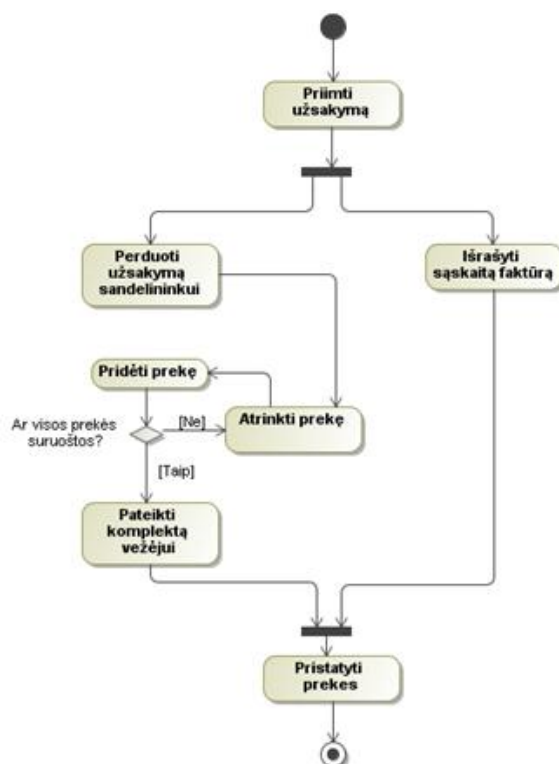
*BPMN* yra skirtas modeliuoti veiklos procesus, verslo elementų aibė labai plati, apimanti daugumą veiklos sričių. Diagramų tipai kuriuos palaiko *BPMN* yra *Choreography*, *Collaboration*, *Conversation* ir *Process*. Išskiriamos 4 pagrindinės elementų grupės (veiklos, įvykių, tinklų sąsajų ir sujungimų). Nėra priemonių perteikti duomenims ir dokumentams, šie elementai gali būti pateikiami tik kaip pastabos. Galima aprašyti veiklos taisykles, jų specifikavimui *BPMN* naudoja *XPath* išraiškas (jomis šiek tiek patogiau specifiuoti veiklos taisykles nei *OCL* išraiškų kalba). Trūksta plačiai naudojamų priemonių joms tikrinti ar transformuoti į vykdomąjį kodą. Yra galimybė transformuoti *BPMN* modelius į *BPEL* standartą.

### 2.3.3. Unifikuota modeliavimo kalba (UML)

Unifikuota modeliavimo kalba (*angl. Unified Modeling Language; UML*) [5, 24] yra *OMG (Object Management Group)* sukurta standartizuota bendrosios paskirties modeliavimo kalba, programinės įrangos inžinerijos srityje. Veiklos procesų modeliavimas yra tik vienas iš galimų šios kalbos panaudojimo variantų.

Naujausia *UML 2.0* versija suteikia galimybę specifiuoti vykdomuosius modelius, todėl ji gali būti kaip nuo platformos nepriklausoma vaizdinio programavimo kalba (UML elementų specifikacija pateikta 2 priede).

*UML* modeliavimo kalbos *Activity* diagramos pavyzdys, kuriame pateikta užsakymo įvykdymo procesas nuo jo priėmimo iki pristatymo, pateiktas 4 pav.



4 pav. MagicDraw paketu sudarytas Activity diagramos pavyzdys

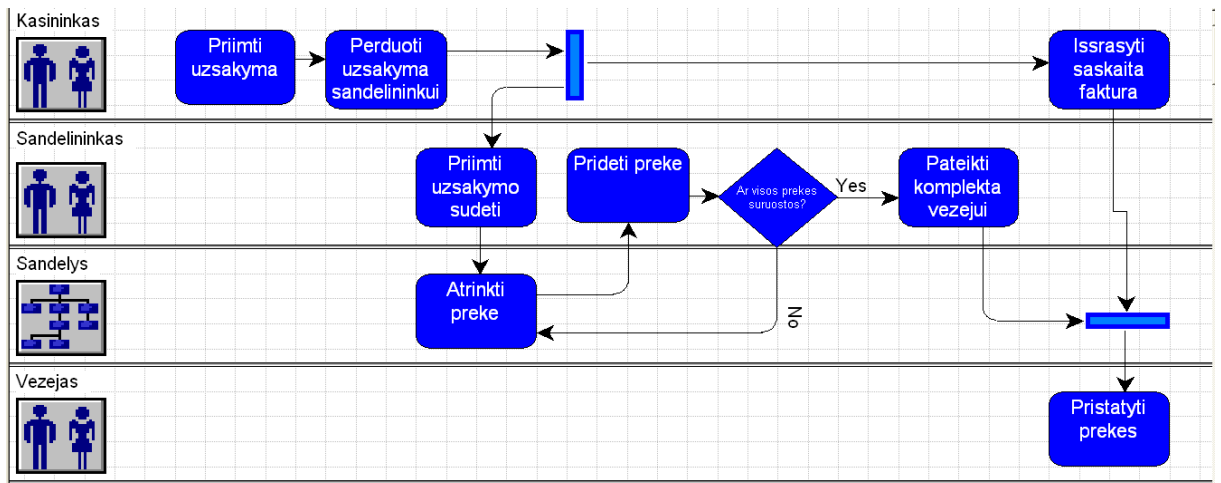
*UML* yra universali modeliavimo kalba, veiklos procesų modeliavimas yra tik vienas iš galimų šios kalbos panaudojimo variantų. Veiklos procesams atvaizduoti reikalingų elementų aibė mažesnė už *BPMN* pasiūlą. *UML* diagramos naudojamos veiklos procesams modeliuoti: *Class*, *Use Case*, *Activity* ir *Sequence*. Aktualiausia *Activity* diagrama, kurioje naudojami pagrindiniai elementai: pradinis mazgas, veiksmas, valdymo srautas ir pabaigos mazgas. *UML* veiklos diagramose yra išsamus priemonių rinkinys duomenims ir dokumentams atvaizduoti. Jie vaizduojami kaip objektai, kurie dalyvauja veiklos procese ir turi įtakos jo eigai. Galima aprašyti veiklos taisykles, joms specifikuoti *UML* turi *OCL* išraiškų kalbą. Trūksta nusistovėjusių rekomendacijų joms tikrinti ar transformuoti į vykdomąjį kodą. Yra galimybė sugeneruoti vykdomąjį kodą iš *UML Class* diagramos.

#### 2.3.4. Darbų sekos modelis

Darbų sekos (*angl. Workflow*) [6, 25] modelis yra vizualinio modelio specializuota forma, apibrėžiama programos įrangos inžinerijos srityje. Šiame modelyje konkrečiai nurodomos pasikartojančios problemos ir susiję sprendimai su kuriamos taikomosios programinės įrangos darbo eiga. Ją sudaro susijusių veiksmų sekos, darbo eiga gali būti vertinama kaip bet kokia reali darbo abstrakcija.

Taigi darbų sekos (*Workflow*) modelis tarnauja, kaip virtualus realaus darbo vaizdinys perteikiantis realaus darbo srautą (*Workflow* elementų specifikacija pateikta 3 priede).

*Workflow* diagramos pavyzdys, kuriame pateikta užsakymo įvykdymo procesas nuo jo priėmimo iki pristatymo, pateiktas 5 pav.



5 pav. ProVision paketu sudarytas Workflow diagramos pavyzdys

Workflow skirtas modeliuoti pačiai darbo eigai, todėl verslo elementų aibė gana siaura. Workflow diagramos naudojamos veiklos procesams modeliuoti: *Activity* ir *Business Process*. Nėra pakankamai priemonių atvaizduoti dalykinri sričiai. Galima aprašyti veiklos taisykles, joms specifikuoti naudojama *Disease Intervention (DIS works)* išraiškos. Nėra tinkamų būdų joms tikrinti ar transformuoti į vykdomąjį kodą. Yra galimybė Workflow susieti su vykdomuoju kodu. Tai atliekama pasitelkiant *Dynamic Petri Net (DPN)* ir *Orc* kodo generatorių.

### 2.3.5. Veiklos procesų modeliavimo metodų palyginimas

Atlikus veiklos procesų modeliavimo standartų *BPMN*, *UML* ir *Workflow* lyginamąją analizę (1 lentelė), nustatyta, kad *BPMN* standartas pranašesnis už kitus turimų elementų (kurie skirti veiklos procesams modeliuoti) gausa.

1 lentelė. Veiklos procesu modeliavimo metodų lyginamoji analizė

Nr.	Savybės\Metodai	BPMN	UML	Workflow
1	Veiklos diagramų tipai, elementų aibė ir kiekis.	+	+ / -	+
2	Duomenų ir dokumentų atvaizdavimo galimybės.	-	+	-
3	Galimybė aprašyti ir specifikuoti veiklos taisykles.	+	+	+
4	Galimybė veiklos taisykles transformuoti į vykdomąjį kodą.	+ / -	-	-
5	Transformacijos susiejimas su vykdomuoju kodu	+	+	+
6	Atitikimas realaus pasaulio procesams	+	+ / -	+ / -
7	Galimybė atvaizduoti sisteminius reikalavimus reikalingus IS kūrimui	+ / -	+	+ / -

## 2.4. Veiklos paslaugų kūrimo metodai

Veiklos paslaugos [9] (*angl. Web Services*) paprastai yra taikomojo programavimo sąsaja (*API*) arba *WebAPI*, kuri yra prieinama per *Hypertext Transfer Protocol (HTTP)* protokolą ir atlieka sistemos paskirtą užduotį. Veiklos paslaugos paprastai skirtos: didelės veiklos paslaugas ir nedideles (ramaus pobūdžio) veiklos paslaugas atliekamų užduočių ir funkcionalumo apimties prasme.

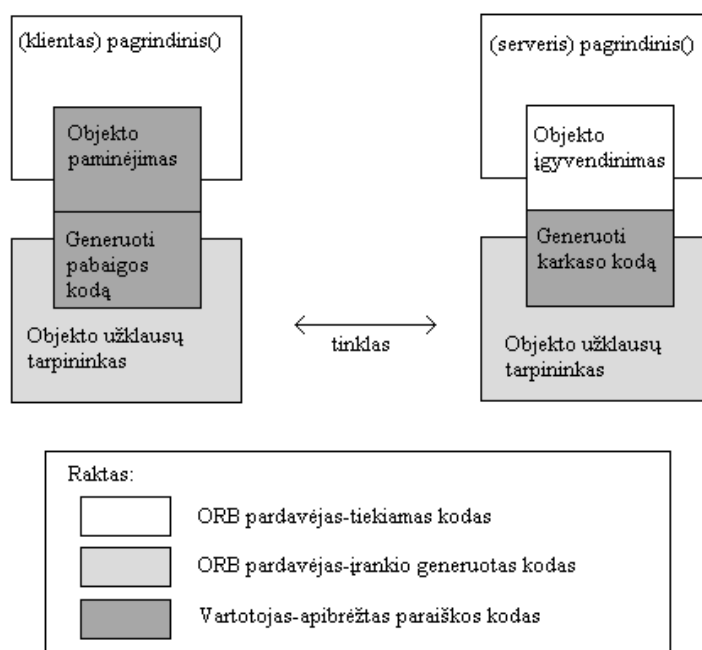
Veiklos paslaugų kūrimui yra sukurta nemažai priemonių, viena iš jų *Object Management Group (OMG)* kompanijos sukurtas *The Common Object Request Broker Architecture (CORBA)* standartas. Taip pat nagrinėjamas standartas veiklos paslaugų kūrimo sferoje yra *The Web Services Description Language (WSDL)*.

Standartas, skirtas struktūrizuotai perteikti interneto paslaugų įgyvendinimą kompiuterių tinklais, yra *SOAP Simple Object Access Protocol (SOAP)*.

### 2.4.1. CORBA standartas

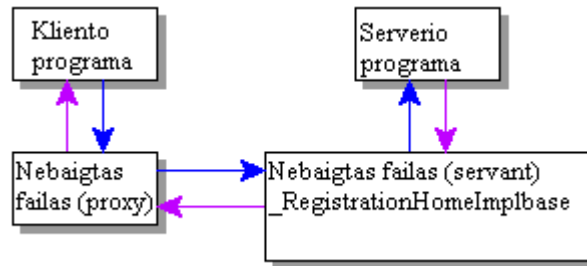
*The Common Object Request Broker Architecture (CORBA)* [9] yra *Object Management Group (OMG)* sukurtas standartas, kuris leidžia skirtingiems programinės įrangos paketams palaikomiems skirtinguose kompiuteriuose – dirbti kartu, kaip vienas didelis vientisas paslaugų rinkinys. *CORBA* [14] naudoja sąsajos apibrėžimo kalbą (*angl. Interface Definition Language, IDL*) tam, kad nustatyti kurie objektai sąveikauja su išoriniu pasauliu.

Įrankio infrastruktūros kodo generavimas *CORBA* įrankio pagalba pateiktas 6 pav. Jame pavaizduota kliento sąveika su serveriu, generuojamo kodo ir informacijos mainų aspektu.



6 pav. CORBA įrankio infrastruktūros kodo generavimas

*Java IDL* serverio kompiliavimo pavyzdys, naudojant *CORBA* standartą pateiktas 7 pav. Jame vaizduojama kompiliavimo principas, kai kliento programa sąveika su serverio programa per tarpinius elementus.



7 pav. *Java IDL* serverio kompiliavimo pavyzdys naudojant *CORBA*

*CORBA* privalumas – gali veikti skirtinguose kompiuteriuose ir sąveikauti tarpusavyje. *CORBA* yra naudinga nes palaiko daugelį skirtingų protokolų. Tačiau yra pagrindinis protokolas, kuris yra naudojamas – *General InterORB Protocol (GIOP)*. *CORBA* standartas buvo sukurtas atsižvelgiant į funkcijas, bet nekreipiant dėmesio į jų nuoseklumą, dėl to *CORBA* specifikacija yra sudėtinga. Naudoja *TCP/IP* protokolą siekiant perduoti duomenis. Procesas vykdomas sėkmingai, jei vartotojas nėra pasirinkęs aukšto apsaugos lygio ugniasienės ar *Proxy* serverio aplinkos.

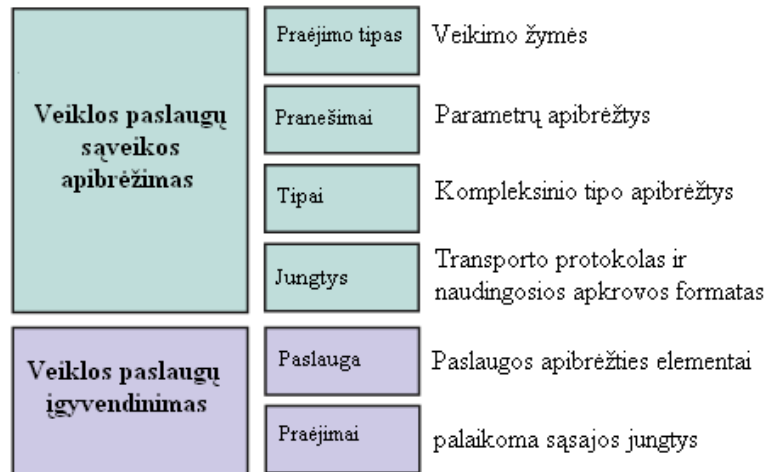
#### 2.4.2. Veiklos paslaugų aprašymo kalba (WSDL)

Veiklos paslaugų aprašymo kalba (*angl. The Web Services Description Language; WSDL*) [10] yra *Extensible Markup Language (XML)* pagrindu sukurta kalba, kuri teikia apibrėžtą veiklos paslaugų modelį. *WSDL* apibūdina tinklo paslaugas, kaip veikiančių pranešimų parametrų dokumentus. Šių dokumentų sudėtyje yra bet kuri iš dokumentui-orientuota (*angl. document-oriented*) arba procedūrai-orientuota (*angl. procedure-oriented*) nustatyta informacija.

Operacijos ir pranešimai yra aprašyti abstrakčiai, tada susijungia su konkrečiu tinklo protokolu ir pranešimo formatu perteikia įvykdytų veiksmų požymius. Susiję realūs požymiai yra sujungti į abstrakčius požymius (paslaugas).

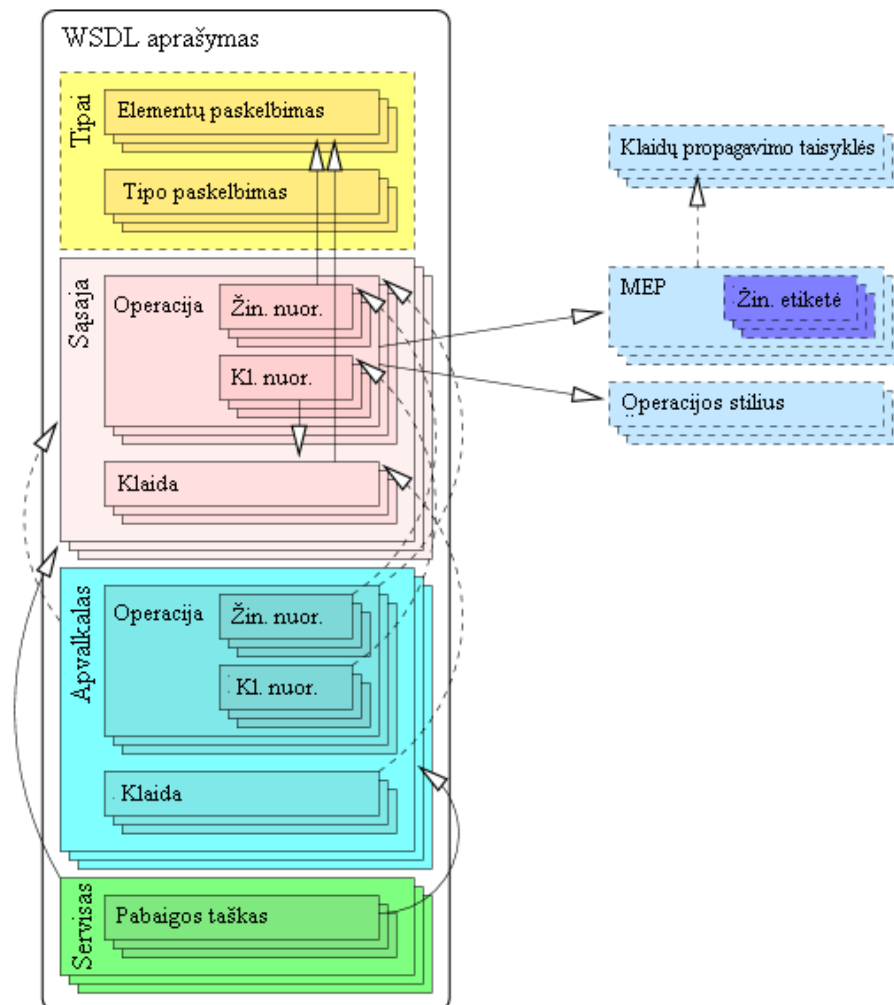
*WSDL* [15] yra išplėčiamas tam, kad leistų aprašyti požymius ir jų pranešimus. Aprašant požymius neatsižvelgiama į tai koks yra pranešimo formatas ar koks tinklo protokolas yra naudojamas komunikuoti tarpusavyje.

*WSDL* architektūra pateikta 8 pav., čia vaizduojama pagrindinės *WSDL* sudedamosios dalys: veiklos paslaugų sąveikos apibrėžimas bei jo dalys (praėjimo tipas, pranešimas, tipai ir jungtys) ir veiklos paslaugų įgyvendinimas, kuris apima apibrėžties elementus ir sąsajos jungtis.



8 pav. WSDL architektūra

WSDL struktūra (aprašymas) pateikta 9 pav. Vaizduojama pagrindiniai struktūriniai WSDL sudarantys elementai (tipai, sąsaja, apvalkalas ir servisas), jų sudėtis bei sąveikos.



9 pav. WSDL struktūra

WSDL naudojamas su SOAP ir XML Schema teikti interneto paslaugas per internetą skirtingiems vartotojams. WSDL yra orientuotas būtent WebServices palaikymui. WSDL

specifikacija standartizuota. Į paslaugas orientuota architektūra užtikrina lengvesnę specifikacijos suprantamumą, lyginant su *CORBA*. Naudojamas *HTTP* gauti/atsakyti modelis leidžia sėkmingai atlikti informacijos mainus, nekrepiant dėmesio į pasirinktus ugniasienės parametrus.

### 2.4.3. Paprastas objekto ryšių protokolas (SOAP)

Paprastas objekto ryšių protokolas (*angl. Simple Object Access Protocol; SOAP*) [12, 13] aprašo struktūrizuotą informacijos perteikimą veiklos paslaugų įgyvendinimo kompiuterių tinklais principą. Jis naudoja *Extensible Markup Language (XML)* savo pranešimų formatavimui, bei kitus aplikacijų valdymui skirtus protokolus, visų pirma tokius, kaip *Remote Procedure Call (RPC)* ir *Hypertext Transfer Protocol (HTTP)*.

Šis *XML* formatu pagrįstas veiklos paslaugų maketavimo protokolas susideda iš trijų dalių: apvalkalo (šis aspektas apibrėžia, kas yra pranešama), kodavimo taisyklių rinkinio bei atstovavimo tvarkos apibendrinimo.

*SOAP* (10 pav.) susideda:

- apvalkalas – suteikia informaciją, kokia pranešimo sudėtis;
- antraštė – tai kodavimo taisyklių rinkinys, skirtas išreikšti apibrėžtiems duomenų tipams;
- korpusas – faktiniai duomenys.



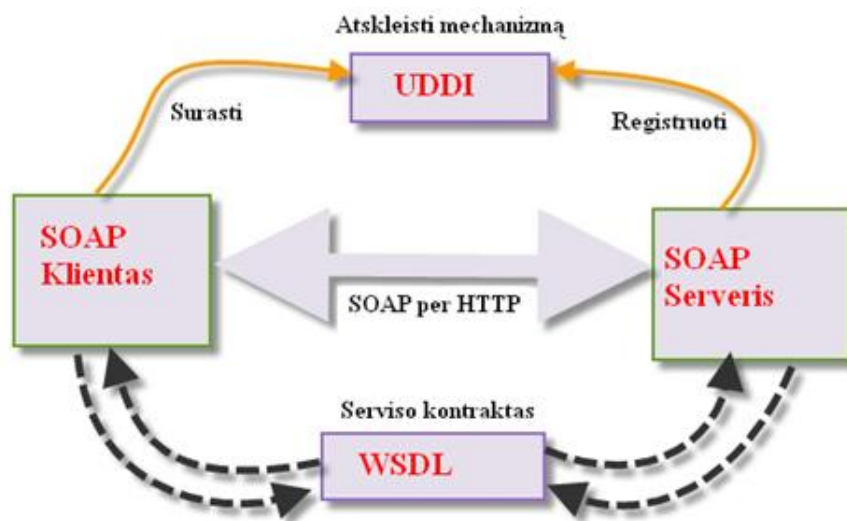
10 pav. *SOAP* struktūra

*SOAP* įrankio architektūra pateikta 11 pav. *SOAP* architektūros schemoje pateikti pagrindiniai elementai:

- *SOAP* Klientas – programa palaikanti *SOAP* protokolą ir vykdanči jos operacijas;
- *SOAP* Serveris – *SOAP* protokolą palaikantis serveris;
- *UDDI (Universal Description Discovery and Integration)* - yra nepriklausomas nuo platformos, *Extensible Markup Language (XML)* pagrįstas įmonių registras,

padedantis identifikuotis įmonėms viena kitai, nustatyti, kad programinės įrangos paketai bendradarbiautų per internetą;

- *WSDL* – veiklos paslaugų aprašymo kalba, kuri teikia apibrėžtą veiklos paslaugų modelį, apibūdina tinklo paslaugas, kaip veikiančių pranešimų parametrų dokumentus.



11 pav. SOAP architektūra

#### 2.4.4. Veiklos paslaugų kūrimo metodų palyginimas

Atlikus veiklos paslaugų kūrimo metodų *CORBA* ir *WSDL* lyginamąją analizę (2 lentelė), nustatyta, kad *WSDL* lenkia *CORBA* veikimo sparta ir tuom, kad *WSDL* specifikacija yra tinkamesnė transformacijai iš veiklos proceso modelio į veiklos paslaugas. *CORBA* privalumas tas, kad joje yra galimybė naudoti daug skirtingų platformų ir kalbų bei dinamiškus iškvietimus. *WSDL* suteikia galimybę vartotojui dirbti skirtinguose kompiuteriuose. *WSDL* specifikacija standartizuota. Į paslaugas orientuota architektūra užtikrina lengvesnį specifikacijos suprantamumą, lyginant su *CORBA*

2 lentelė. Veiklos paslaugų kūrimo metodų lyginamoji analizė

Nr	Savybės\Metodai	<i>CORBA</i>	<i>WSDL</i>
1	Darbas skirtinguose kompiuteriuose	+	+
2	Skirtingų protokolų palaikymas	+	-
3	Specifikacijos lengvas suprantamumas	-	+
4	Ugniasienė ir <i>Proxy</i> serverio įgaliojimai	-	-

#### 2.5. Transformacijos metodai

Veiklos proceso modelio transformavimą į veiklos paslaugas galima traktuoti, kaip modelio perteikimą, kai atliekant tam tikrą standartizuotų veiksmų seką sudaromas gamybos sisteminis programinis kodas.

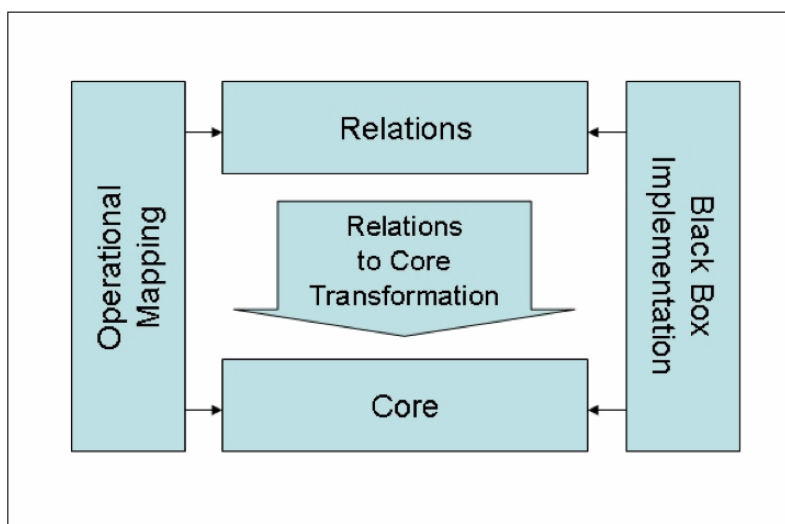


Egzistuoja keletas metodų (programinės įrangos paketų) skirtų atlikti šią transformaciją. Vienas jų *SmartQVT TM* atviro kodo įrankis, skirtas atlikti transformaciją pereinant nuo modelių prie realaus kodo generavimo, kas leidžia pagreitinti programinės įrangos kūrimo procesą.

Kitas metodas – *MDA™ / MDE*, tai metodas orientuotas į tiesioginio supratimo modelių naudojimą, juos perteikiant į *MDE* programinės įrangos kūrimo proceso vykdomąjį kodą.

### 2.5.1. QVT transformacijos standartas

*Query/View/Transformation (QVT)* [8] yra *OMG (Object Management Group)* sukurta standartizuota *MDA* modelio transformacija. Pagrindinė šios transformacijos esmė yra konvertuoti modelį *Ma* atitinkančio metamodelio *MMa* į modelį *Mb* atitinkantį metamodelį *MMb*. Jeigu  $MMa = MMb$ , tada transformacija yra endogeninė (vidinė), kitu atveju egzogeninė (išorinė). Modelio transformacija yra svarbus *MDA* modelio komponentas. *QVT* architektūra pateikta 12 pav.

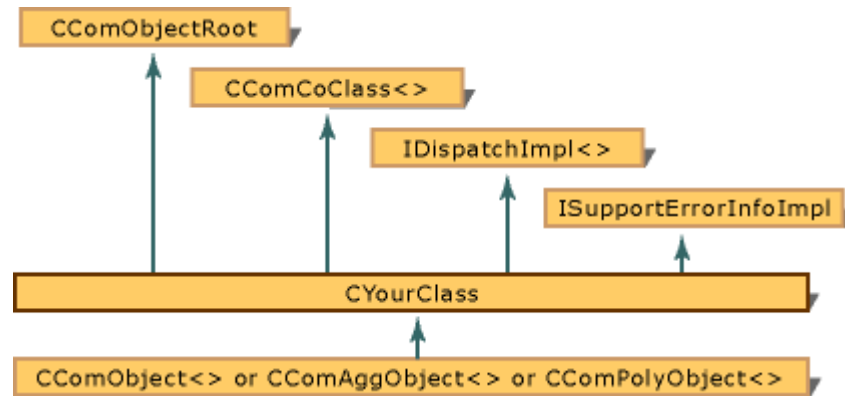


12 pav. *Query/View/Transformation* architektūra

### 2.5.2. Aktyvi šablonų biblioteka (ATL)

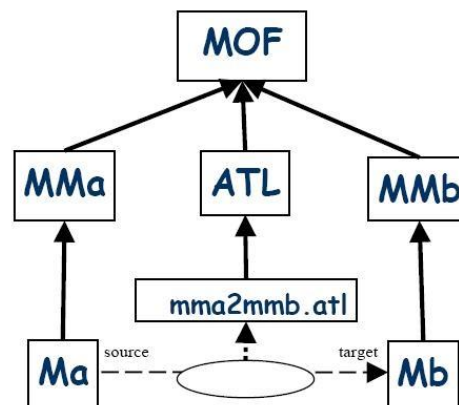
Aktyvi šablonų biblioteka (*angl. The Active Template Library; ATL*) [11] (13 pav.) yra *Microsoft* kompanijos šablonų pagrindu sukurta C++ klasės modelių transformavimo kalba ir įrankių rinkinys. Ši priemonė skirta supaprastinti *Component Object Model (COM)* objektus. *COM* palaikantis *Microsoft Visual C++* leidžia vartotojams kurti *COM* objektus.

*ATL* objektų vedlys suteikia galimybę ir tam reikalingas priemones sukuriant pirminius struktūrinius objektus labai greitai su minimaliais pataisymais rankiniu būdu.



13 pav. ATL struktūra

ATL yra modelių transformacijos kalba (14 pav.), kuri leidžia nurodyti, kaip vieną (arba daugiau) tikslinių modelių galima sukurti iš šaltinio modelių rinkinio. Kitaip tariant, ATL pristato sąvokas, kuriomis galima aprašyti modelio transformacijos rinkinius. ATL yra panaši į anksčiau paminėto *Query/View/Transformation (QVT)* transformacijos kalba su atviro kodo biblioteka.



14 pav. ATL transformacija

## 2.6. Esamų sprendimų analizė

Jau egzistuoja keletas skirtingų siūlomų transformacijos variantų, vienas jų yra *Borland* kompanijos sukurtas *Together* [7] produktas, kuris integruotas į *Java IDE*, šis produktas iš pradžių vienijo *JBuilder* su *UML* modeliavimo įrankiu.

*Visual Paradigm for UML (VP-UML)* yra *Visual Paradigm International Ltd* kompanijos sukurtas produktas skirtas veiklos procesų modeliavimui, įrankis palaiko ataskaitų generavimą bei kodo inžinerijos galimybes (įskaitant kodo generavimą).

### 2.6.1. Borland Together sistema

*Together* [7] yra *Borland* sukurtas produktas, kuris integruotas į *Java IDE*.

Produktas pradžioje buvo išskaidytas pagal funkcionalumą, atskiri paketai *Together Developer*, *Together Designer* ir *Together Architect*. Toks principas gyvavo iki 2007 metų

kai galiausiai juos apjungė į vieną vientisą produktą. Iki 2006 metų *Together* buvo savarankiški produktai, tačiau nuo duotųjų metų apjungtas vientisas produktas naudojamas kaip *Eclipse* programos įskiepis.

*Together Developer* [16] palaiko *UML 1.4* modeliavimą, daugiakalbiškumą, fizinių duomenų modeliavimą, vaizdinius modelius, kodo vaizdinio atpažinimą ir dokumentacijos generavimą.

*Together* praplėstas *UML 2.0* diagramų, veiklos procesų modeliavimo, loginių duomenų modeliavimo, loginių fizinio duomenų modelio transformacijos ir nestandartinių modelio plėtiniiu.

- Palaikoma *Eclipse 3.2*
- *BPMN* diagramos gali būti sukuriamos jas importuojant ir naudojamos sugeneruoti perėjimą į veiklos proceso vykdymo kalbą su veiklos paslaugų apibrėžimais (*BPEL4WS*).
- Glaudesnis bendradarbiavimas su *UML, Design Patterns, Java 6, C, CORBA*.
- *Together Architect* skirtas *Eclipse* yra vienas iš pirmųjų *UML* įrankių leidžiančių modelių transformavimo galimybes.

*Borland Together* sistemoje galima sudaryti bei importuoti *UML* modelius, tačiau *BPMN* kūrimas nėra integruotas, *BPMN* modelius galima tik importuoti. *Together* yra integruotas į *JavaIDE*, todėl kada galim generuoti (atlikti transformaciją) tik į keletą programavimo kalbų (*Java, C, C#*). *Together* įrankis nors ir palaiko *BPMN*, tačiau orientuotas į *UML* naudojimą.

### 2.6.2. Visual Paradigm for UML sistema

*Visual Paradigm for UML (VP-UML)* [17] yra *Visual Paradigm International Ltd* kompanijos sukurtas produktas skirtas veiklos procesų modeliavimui, paremtas *UML* karkaso įrankių naudojimu. Įrankis palaiko *UML 2.0, SysML* ir *BPMN*.

*VP-UML* taip pat suteikia galimybę generuoti numatytas ataskaitas bei kodo inžinerijos galmybes (įskaitant kodo generavimą). Šio įrankio pagalba galima atkurti inžinerinius modelius iš vykdomojo kodo, taip pat atlikti transformaciją iš modelio į vykdomąjį kodą (galimybė transformaciją atlikti į įvairias programavimo kalbas).

*VP-UML* palaiko tokius *UML* modelius:

- klasių diagrama (*angl. Class diagram*);
- panaudojimo atvejų diagrama (*angl. Use case diagram*);
- sekų diagrama (*angl. Sequence diagram*);
- ryšių diagrama (*angl. Communication diagram*);

- būsenų diagrama (*angl. State machine diagram*);
- veiklos diagrama (*angl. Activity diagram*);
- komponentų diagrama (*angl. Component diagram*);
- diegimo diagrama (*angl. Deployment diagram*);
- paketų diagrama (*angl. Package diagram*);
- objektų diagrama (*angl. Object diagram*);
- kompozicinė struktūrinė diagrama (*angl. Composite structure diagram*);
- chronometražo diagrama (*angl. Timing diagram*);
- sąveikų apžvalgos diagrama (*angl. Interaction overview diagram*).

*VP-UML* viena iš funkcijų – reikalavimų modeliavimas, įskaitant panaudojimo atvejų, *SysML* reikalavimų diagramų bei tekstinės analizės dokumentų sudarymą. *SysML* reikalavimų diagrama skirta specifiuoti ir analizuoti reikalavimus.

*VP-UML* suteikia galimybę iš modelio sugeneruoti kodą daugeliu programavimo kalbų, tokių kaip *Java*, *PHP*, *C#*, *VB.NET*, *C++*, *PERL* ir kt.

*VP-UML* sistemoje kodo generavimas vyksta sukūrus naują arba importavus jau sukurtą modelį, pasirenkama norima programavimo kalba, kuri bus naudojama generavimo metu bei pasiremiant pasirinktu modeliu sugeneruojamas vykdomasis kodas.

*VP-UML* sistema nors ir palaiko *BPMN*, tačiau kompanija nėra suinteresuota šios srities vystymu (apsiribojama *UML* modeliavimo kalba) ir labiau koncentruojamasi ties atvirkštine kodo generavimui eiga – iš vykdomojo kodo sudaryti modelius.

*VP-UML* sistema suteikiama galimybę sudaryti tiek *UML*, tiek *BPMN* modelius bei juos importuoti. *VP-UML* įrankyje yra galimybė sugeneruoti vykdomąjį kodą į daugelį programavimo kalbų, tokių kaip – *Java*, *PHP*, *C#*, *VB.NET*, *C++*, *PERL* ir kt. Suteikiama galimybė atlikti transformaciją remiantis *BPMN*, tačiau kaip ir *Together* skirtas *UML* naudojimui.

### **2.6.3. UModel sistema**

*UModel* [18] yra *UML* programinės įrangos modeliavimo priemonė, kurią sukūrė *Altova* kompanijos programuotojai. *UModel* palaiko visas 14 *UML 2.0* diagramų tipus ir vieną papildomą unikalią diagramą, skirtą *XML* schemų modeliavimui pasitelkiant *UML* technologiją. Ši programinė įranga taip pat palaiko *SysML* bei veiklos procesų modeliavimo notaciją (*BPMN*).

*UModel* apima kodo inžinerijos funkcijas, įskaitant kodo generavimą (*Java*, *C# VisualBasic* programavimo kalbomis).

*UModel* palaiko modelių apskaitimą su kitais *UML* įrankiais per *XMI* standartą, susijungia su peržiūros kontrolės sistemomis ir veikia kaip *Eclipse* ir *Visual Studio IDE* programų įskiepis.

Aukščiau išvardintos *UModel* savybės suteikia sistemai lankstumo. Įrangos modeliavimo procesas yra patogus ir gerai dokumentuotas.

*UModel* sistemoje galima sudaryti bei importuoti jau sukurtą modelį. *UModel* palaiko modelių apskaitimą su kitais *UML* įrankiais per *XMI* standartą, bei veikia, kaip *Eclipse* ir *Visual Studio IDE* programų įskiepis, tai suteikia sistemai lankstumo norint integruoti jau sukurtus modelius. Kodo generavimas galimas į *Java*, *C#* ir *Visual Basic* programavimo kalbas. Orientuota į *UML*. *BPMN* palaikomas, tačiau naudojimas mažiau išplėtotas nei kituose lyginamuose programinės įrangos paketuose.

#### 2.6.4. Transformacijos iš veiklos proceso modelio į veiklos paslaugas esamų sprendimų palyginimas

Atlikus transformacijos iš veiklos proceso modelio į veiklos paslaugas esamų sprendimų (*Together*, *VP-UML* ir *UModel*) analizę (3 lentelė) nustatyta, kad *UModel* įrankis pranašesnis už *Together* ir *VP-UML* galimybe modeliuoti ir importuoti veiklos proceso modelius (suderintas su kitais modeliavimo įrankiais per *XMI* standartą) bei kodo generavimo praplėtimo galimybe (veikia ir kaip *Eclipse* ir *Visual Studio IDE* programų įskiepis).

3 lentelė. Esamų sprendimų lyginamoji analizė

Nr.	Savybės\Metodai	<i>Together</i>	<i>VP-UML</i>	<i>UModel</i>	Siūlomas sprendimas
1	Modelio sudarymo ar jau egzistuojančio modelio importavimas.	+ / -	+	+	+
2	Kodo generavimas: daugiakalbiškumas.	+	+	+	+ / -
3	Veiklos proceso modelis.	+	+	+	+
4	Licencija.	Mokamas (30 dienų nemokama <i>Trial</i> versija)	Mokamas (30 dienų nemokama <i>Trial</i> versija)	Mokamas (30 dienų nemokama <i>Trial</i> versija)	Nemokamas
5	<i>WSDL</i> palaikymas.	-	-	-	+
6	Prieinamumas, diegimo erdvė.	Reikia įdiegti kompiuteryje.	Reikia įdiegti kompiuteryje.	Reikia įdiegti kompiuteryje.	Prieiga per internetą.
7	<i>BPMN</i> palaikymas.	-	+	+	+
8	<i>XML</i> formato palaikymas.	+	+	+	+
9	Versijavimas.	Nuolat atnaujinamas.	Nuolat atnaujinamas.	Kasmet išleidžiama nauja versija.	-

Nr.	Savybės\Metodai	<i>Together</i>	<i>VP-UML</i>	<i>UModel</i>	<i>Siūlomas sprendimas</i>
10	Reikalingi resursai palaikymui, <i>CPU % / RAM MB</i> .	2-15% / 250 MB	3-20% / 320 MB	2-16% / 14 MB	Tiek, kiek reikalauja internetinės naršyklės palaikymas (pvz. <i>Opera</i> palaikymas 2-8% / 40 MB)

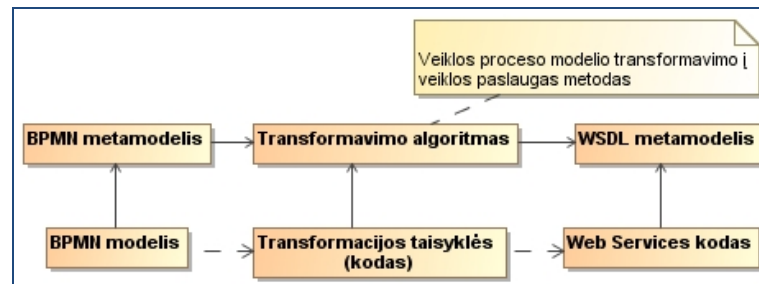
## 2.7. Analizės išvados

1. Atlikus veiklos procesų modeliavimo standartų *BPMN*, *UML* ir *Workflow* lyginamąją analizę, buvo atliekamas palyginimas pagal šiuos kriterijus: veiklos diagramos tipai, elementų aibė, jų kiekis; atitikimas realaus pasaulio procesams; transformacijos susiejimas su vykdomuoju kodu ir kt. Nustatyta, kad *BPMN* standartas tinkamiau ir detaliau atvaizduoja realaus pasaulio procesus, elementų naudojimas yra intuityvus IT išsilavinimo neturintiems žmonėms. *UML* yra dedikuotas informacinių sistemų kūrimui ir neatitinka vartotojų poreikių specifikuojant procesus. Analogiškai, *Workflow* nesuteikia galimybės vartotojui efektyviai atvaizduoti sudėtingus procesus ar išskirtines situacijas.
2. Atlikus veiklos paslaugų kūrimo metodų *CORBA* ir *WSDL* lyginamąją analizę, nustatyta, kad *WSDL* užtikrina geresnę veikimo spartą bei jo specifikacija yra tinkamesnė transformacijai iš veiklos proceso modelio į veiklos paslaugas. Kriterijai, pagal kuriuos buvo atliekama lyginamoji analizė: darbas skirtinguose kompiuteriuose; skirtingų protokolų palaikymas; specifikacijos įsisavinimo laikotarpis.
3. Atlikus transformacijos iš veiklos proceso modelio į veiklos paslaugas esamų sprendimų (*Together*, *VP-UML* ir *UModel*) analizę nustatyta, kad *UModel* įrankis pranašesnis už *Together* ir *VP-UML* savo galimybe modeliuoti ir importuoti veiklos proceso modelius (suderintas su kitais modeliavimo įrankiais per *XMI* standartą) bei kodo generavimo praplėtimo galimybe (veikia ir kaip *Eclipse*, ir kaip *Visual Studio IDE* programų įskiepis).

### 3. Veiklos proceso modelio transformavimo į veiklos paslaugas algoritmo sudarymas

Veiklos proceso modelio, praplėsto veiklos taisyklėmis, transformavimo į veiklos paslaugas algoritmas skirtas padėti vartotojui automatizuotu būdu gauti vykdomąjį kodą (iš veiklos proceso modelio).

Principinė schema su pagrindiniais duomenų šaltiniais bei naudojamomis technologijomis, paremta ATL transformacijos technologijos principu, pateikta 15 pav.



15 pav. Transformacijos rinkinių aprašas

Algoritmo pagrindiniai žingsniai (16 pav.):

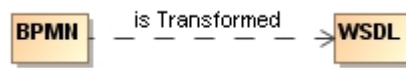
- „1. Paruošti veiklos proceso modelį transformavimui“ – šiame žingsnyje atliekama vartotojo pusėje, sumodeliuojant veiklos procesą, pasirenkant transformavimo nagrinėjimo detalumą ir inicijuojant transformavimą į veiklos paslaugas;
- „2. Transformuoti veiklos proceso modelį į veiklos paslaugas“ – šis žingsnis vykdomas sistemos. Jo metu pažingsniui atliekama transformavimo procedūra.
- „3. Patvirtinti veiklos paslaugas“ – šio žingsnio esmė, gautą vykdomąjį kodą peržiūrėti, ištaisyti klaidas ir patvirtinti transformaciją (jų nelikus). Ši dalis vėl gali būti atliekama vartotojo (programuotojo).



16 pav. Transformacijos algoritmo pagrindiniai žingsniai

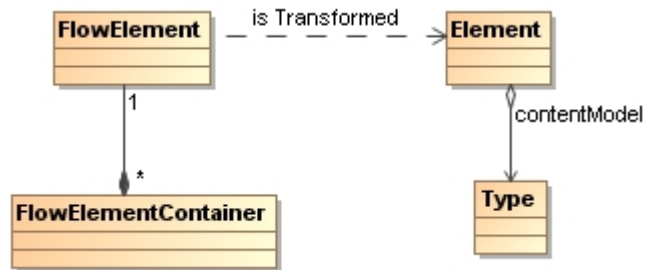
#### 3.1. Veiklos proceso modelio transformacijos į veiklos paslaugas algoritmas

Veiklos proceso modelio, praplėsto veiklos taisyklėmis, transformavimo į veiklos paslaugas algoritmas skirtas padėti vartotojui automatizuotu būdu gauti vykdomąjį kodą (iš veiklos proceso modelio). Pagrindinės technologijos naudojamos atliekant transformavimą yra *BPMN* ir *WSDL*, todėl 17 pav. Pateikta būtent šių technologijų tarpusavio sąsaja (kryptingu „is Transformed“ ryšiu apjungus *BPMN* ir *WSDL*), kuri įvardinama, kaip apibendrintas metodo metamodelis.



17 pav. Apibendrintas metodo metamodelis

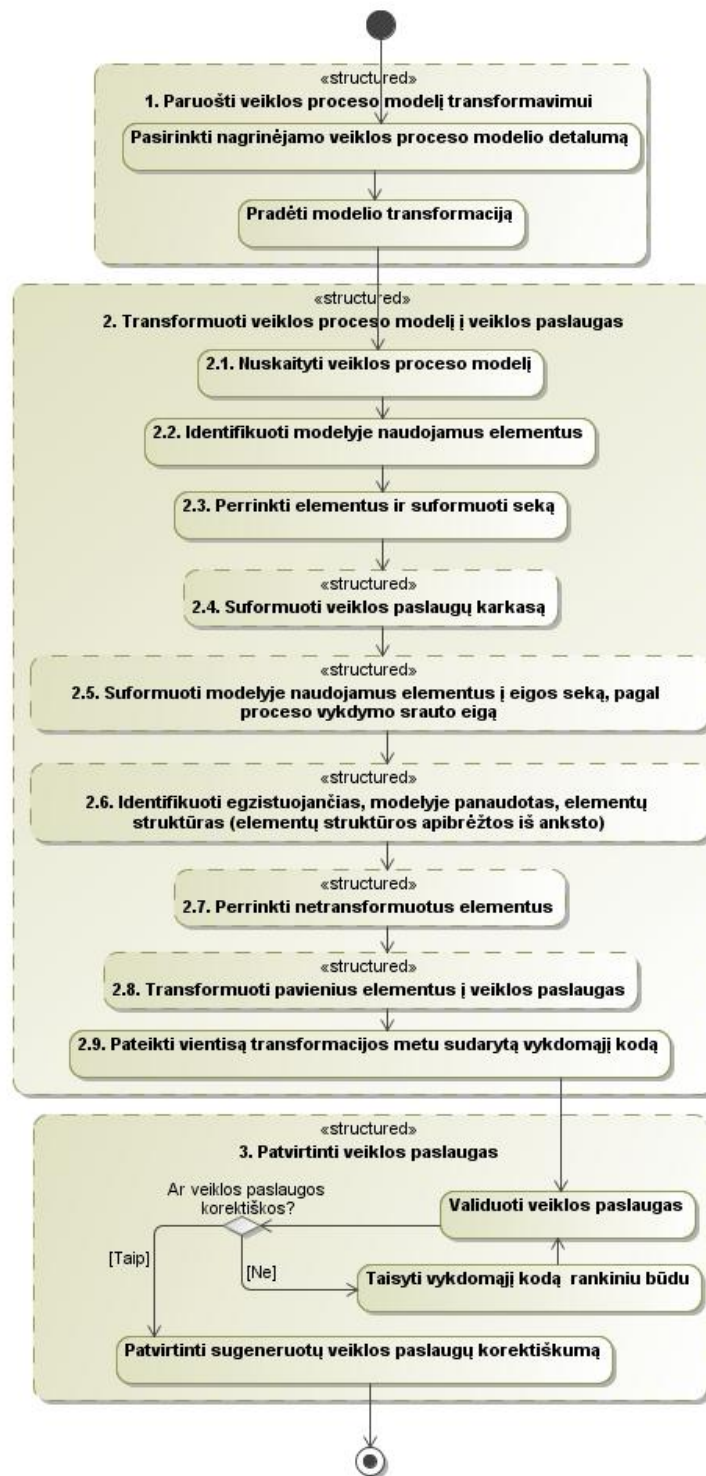
Principinė schema (18 pav.) su pagrindiniais *BPMN* („*FlowElement*“ ir „*FlowElementContainer*“) bei *WSDL* („*Element*“ ir „*Type*“), atspindinčiais bei detalizuojančias 17 pav. vaizduojantį transformacijos principą, transformacijos sąsajoje tarp *BPMN* ir *WSDL*.



18 pav. Principinė metodo metamodelio schema

Apibendrintas veiklos proceso modelio, praplėsto veiklos taisyklėmis, transformacijos į veiklos paslaugas algoritmas pateiktas 19 pav.





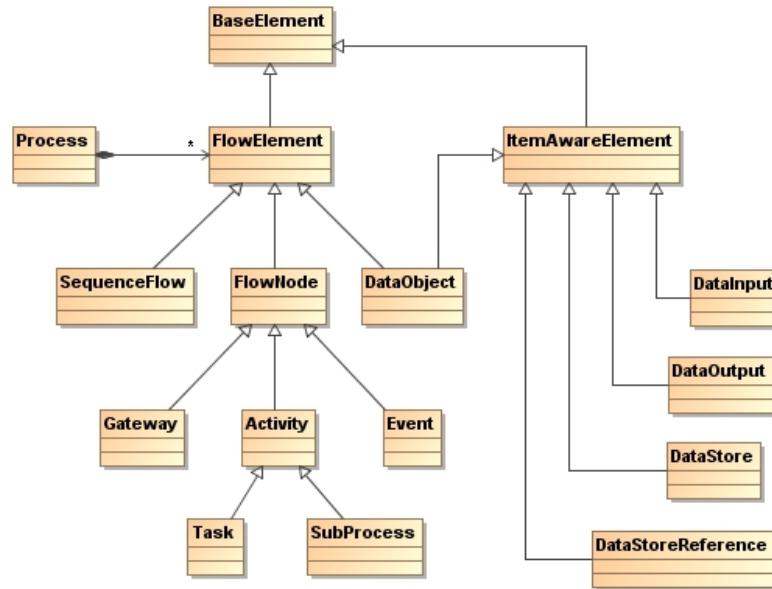
19 pav. Apibendrintas transformacijos algoritmas

### 3.2. BPMN elementų hierarchija

Veiklos proceso modeliui sudaryti naudojamas *BPMN*. Nagrinėjamos notacijos apibendrintas metamodelis pateiktas 20 pav. Jame pateikiama notacijoje naudojami elementai ir jų tarpusavio priklausomybės. Bendriniai notacijos elementai nagrinėjami pateikiant jų išsamesnius metamodelius.

Išskirti dažniau naudojami *BPMN* elementai pagal jų grupes:

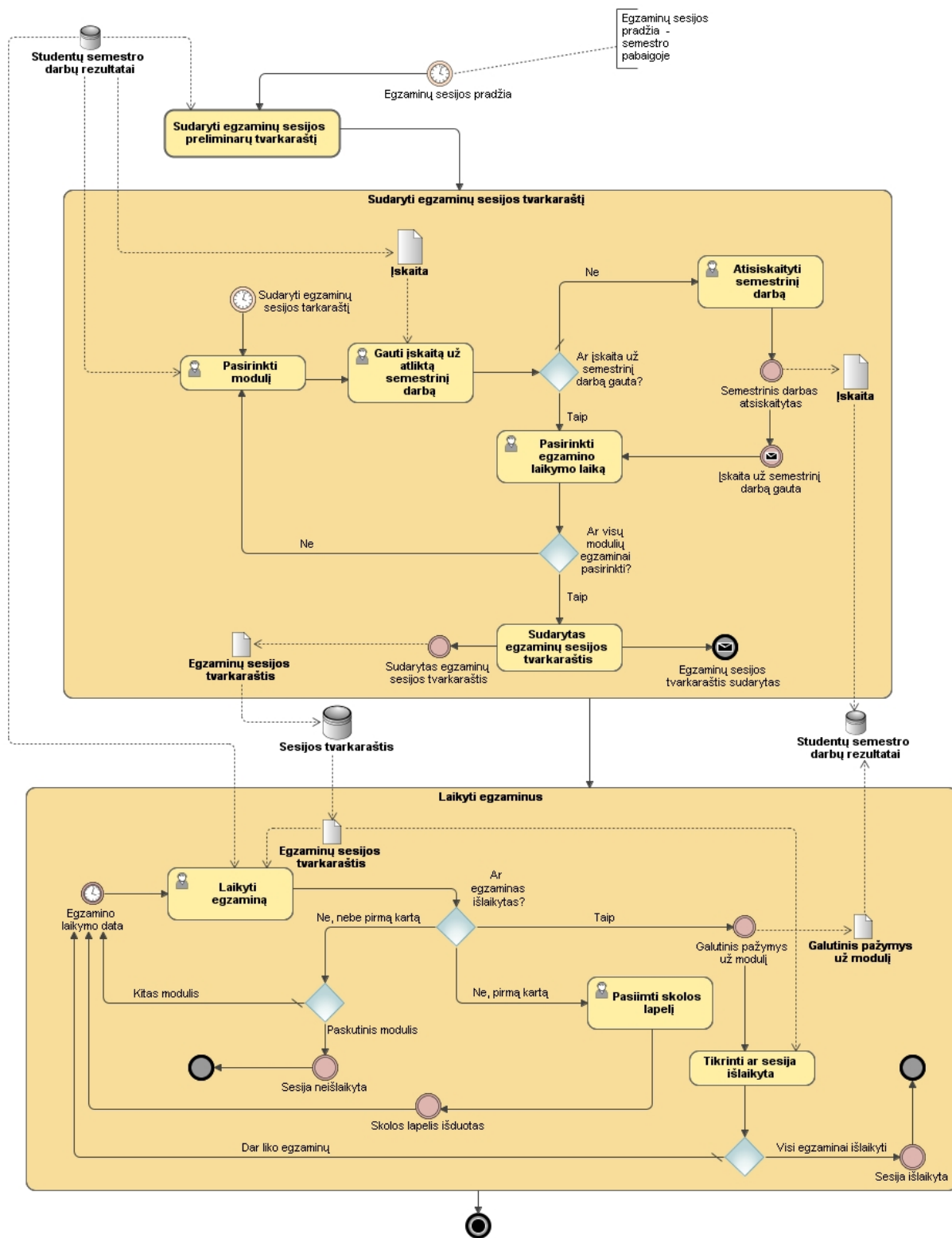
- *Activities;*
- *Events;*
- *Gateways;*
- *Items and Data;*
- *Resource assignment.*



20 pav. BPMN apibendrintas elementų metamodelis

Darbo metu nagrinėjamas pavyzdys yra egzaminų sesijos procesas. Šis procesas (21 pav.) pateiktas sudarytoje *BPMN* proceso diagrama (*angl. BPMN Process Diagram*).

Pagrindinis tikslas kodėl pasirinktas šis procesas – galimybė panaudoti platų *BPMN* elementų spektrą.

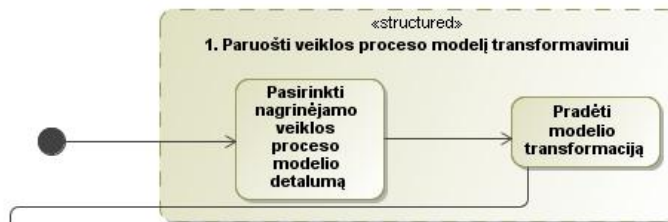


21 pav. BPMN proceso diagrama - „Egzaminų sesijos procesas“

### 3.3. 1 žingsnis: „Paruošti veiklos proceso modelį transformavimui“

Pirmasis transformacijos algoritmo žingsnis (22 pav.) – „Paruošti veiklos proceso modelį transformavimui“. Šis žingsnis susideda iš dviejų poveiksmių: „Pasirinkti nagrinėjamo veiklos proceso modelio detalumą“ ir „Pradėti modelio transformaciją“.

Pirmuoju atveju tikslas – vartotojui nurodyti kokių detalumo lygmeniu nagrinėti norimą transformuoti veiklos proceso modelį. Galimi pasirinkimo variantai: paviršutiniškas (elementai nėra struktūriniai) ir detalus (nagrinėjami elementai gali būti struktūriniai ir/arba sudėtiniai).

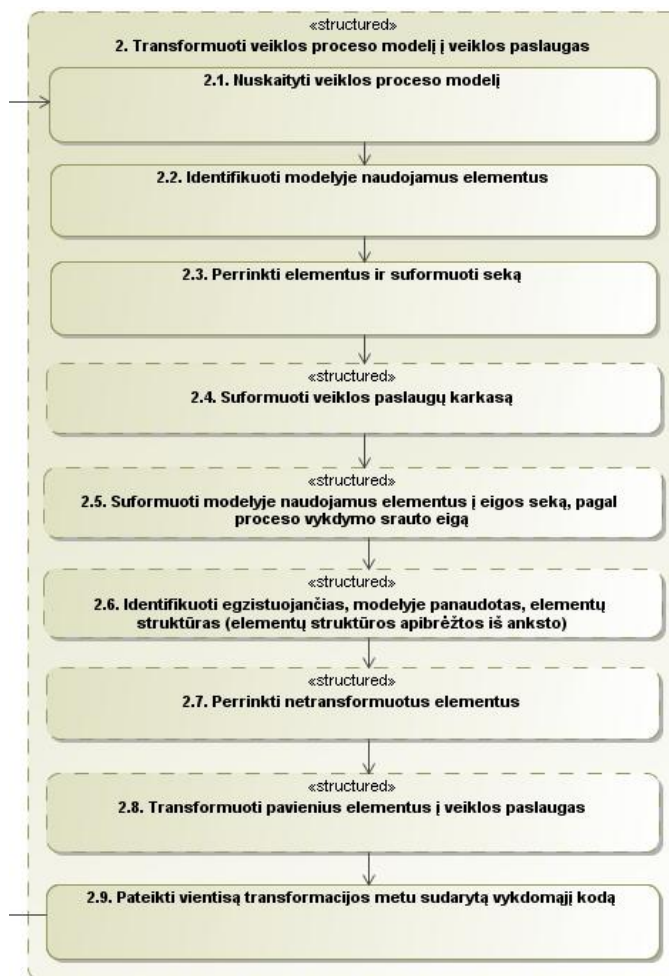


22 pav. Transformacijos algoritmo žingsnio „Paruošti veiklos proceso modelį transformavimui“ detalizavimas

### 3.4. 2 žingsnis: Transformuoti veiklos proceso modelį į veiklos paslaugas

Antrasis transformacijos žingsnis (23 pav.) – „Transformuoti veiklos proceso modelį į veiklos paslaugas“ yra esminis algoritmo žingsnis, kurio metu yra atliekama pagrindiniai transformavimo veiksmai.

Šio žingsnio elementai yra dviejų tipų – struktūriniai ir nestruktūriniai. Pastarieji yra vieno žingsnio sistemos atliekami veiksmai.

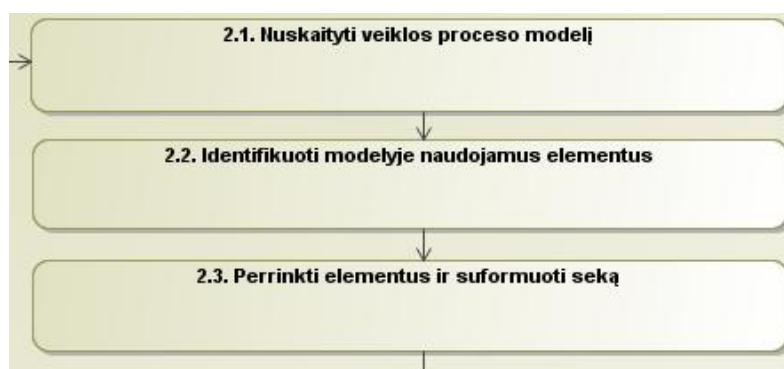


23 pav. Transformacijos algoritmo žingsnio „Transformuoti veiklos proceso modelį į veiklos paslaugas“ detalizavimas

### 3.4.1. 2.1-2.3 žingsniai: Modelio paruošimas transformavimui

2.1-2.3 žingsniai yra nestruktūriniai (24 pav.) antrajame transformacijos žingsnyje sistemos atliekami veiksmai:

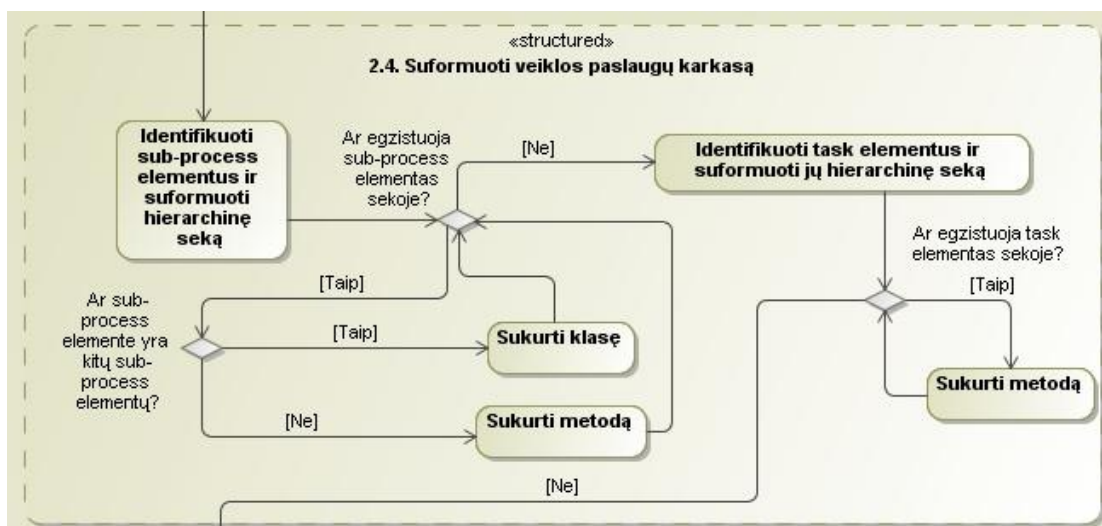
- „Nuskaityti veiklos proceso modelį“ – šio veiksmo tikslas, programos viduje nuskaityti vartotojo pateiktą veiklos proceso modelį;
- „Identifikuoti modelyje naudojamus elementus“ – identifikuojami visi pateiktajame modelyje naudojami elementai;
- „Perrinkti elementus ir suformuoti seką“ – identifikavus naudojamus elementus, sudaroma elementų seka (ši seka nėra vientisa, srautas gali išsiskirti ir/arba apsiungti);



24 pav. Transformacijos algoritmo žingsniai - modelio paruošimas transformavimui

### 3.4.2. 2.4 žingsnis: Suformuoti veiklos paslaugų karkasą

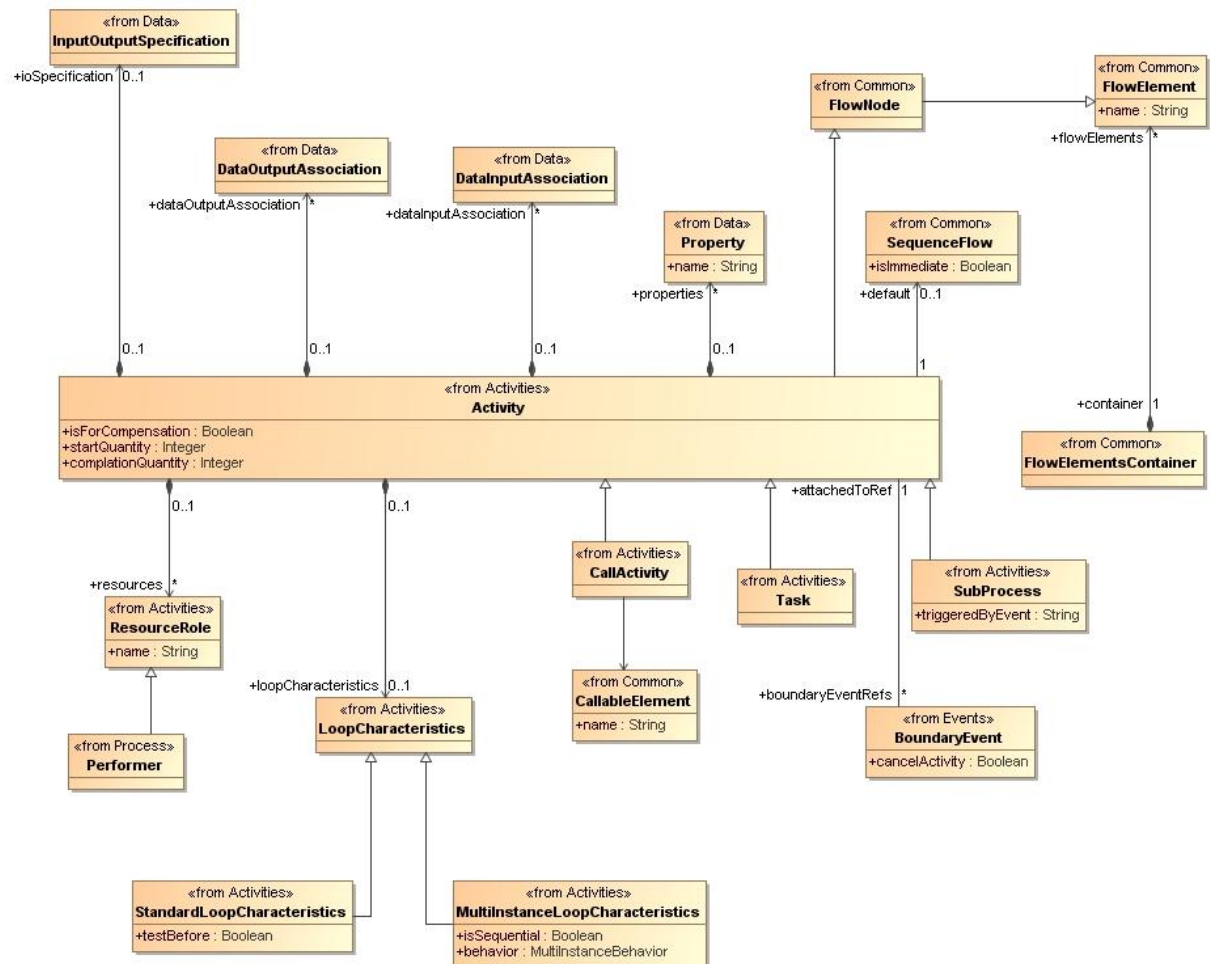
Sistema atlikdama „Suformuoti veiklos paslaugų karkasą“ punktą (25 pav.), identifikuoja BPMN notacijoje naudojamus *Activity* elementus (*Sub-process* ir *Task*), juos transformuoja atitinkamai pagal struktūrą į veiklos paslaugas (klases ir metodus) suformuodami karkasą.



25 pav. Transformacijos algoritmo žingsnio „Suformuoti veiklos paslaugų karkasą“ detalizavimas

Veikla (*angl. Activity*) yra darbas, kuris atliekamas pagal verslo procesą. Veikla gali būti atominė ar neatominė, priklausomai nuo jungčių. Dalies proceso veiklos tipai yra tokie: užduotis (*angl. Task*), dalinis procesas (*angl. Sub-Process*) ir iškvietimo veikla (*angl. Call Activity*), kuri leidžia pakartotinai naudoti užduotis ir procesus diagramose.


Veiklos elementų ir jų priklausomybių metamodelis pateiktas 26 pav.





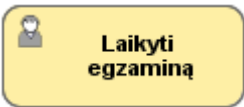


26 pav. „Activity“ metamodelis

21 pav. pateikto pavyzdyje panaudojamų veiklos elementų transformavimo į veiklos paslaugas principai pateikti 4 lentelėje.

4 lentelė. BPMN Activity (Sub-process ir Task) elementų transformavimas į veiklos paslaugas

Nr.	BPMN elementas	Veiklos paslaugų struktūra	Komentaras
1.	 BPMN Activity Task	<pre>task() { // metodo programinis tekstas } &lt;element name="task"&gt; &lt;/element&gt;</pre>	Bendruoju atveju sukuriamas metodas su pavadinimu task (BPMN Task elemento pavadinimas).

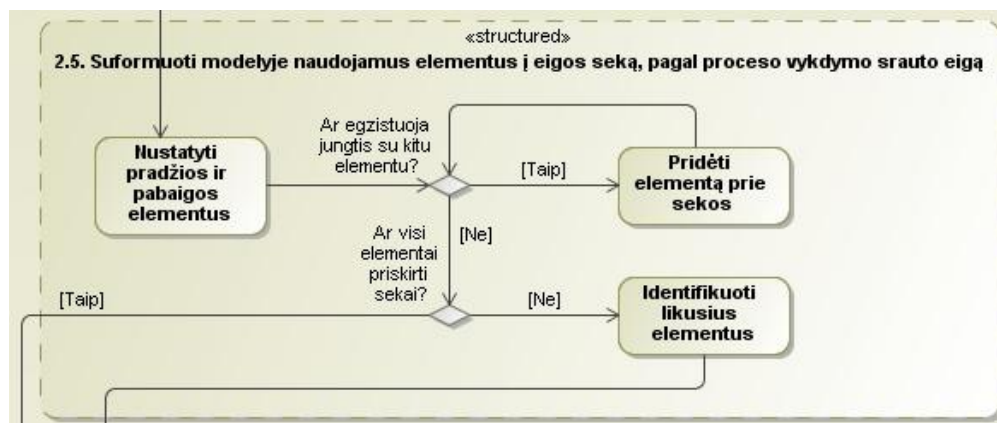
Nr.	BPMN elementas	Veiklos paslaugų struktūra	Komentaras
	Pavyzdys:  (Task – „Kitas egzaminas“)	<pre> kitas_egzaminas() { // metodo programinis tekstas } </pre>	Metodus pavadinimu kitas_egzaminas. Pavadinimas sudaromas modifikuojant elemento (modelyje) pavadinimą – visas didžiąsias ir lietuviškas reides (jas paverčiant mažosiomis ir/arba lotyniškoms raidėms) ir panaikinant visus tarpus (vietoje jų naudojant simbolį „_“).
		<pre> &lt;element name="kitas_egzaminas"&gt; &lt;/element&gt; </pre>	
2.	 BPMN Activity Sub-process	<pre> class laikyti_egzaminus() { // klasės programinis tekstas } </pre>	Bendruoju atveju, jeigu sub-process elementas savo viduje turi struktūrinių elementų, jis transformuojamas į klasę (laikyti_egzaminus).
		<pre> laikyti_egzaminus() { // metodo programinis tekstas } </pre>	Jeigu transformuojamas elementas viduje neturi struktūrinių elementų, jis transformuojamas į metodą (laikyti egzaminus).
		<pre> &lt;types&gt;   &lt;complextypename="subprocess"&gt;     ...   &lt;/complextypename&gt; &lt;/types&gt; </pre>	
	Pavyzdys:  (Sub-process – „Laikyti egzaminus“)	<pre> laikyti_egzaminus(\$kint) { \$kint = ...; // metodo programinis tekstas } </pre>	 (DataOutput) elemento, kuriame saugoma informacija apie sesijos metu laikomus egzaminus.
		<pre> &lt;types&gt;   &lt;complextypename="laikyti_egzaminus"&gt;     ...   &lt;/complextypename&gt; &lt;/types&gt; </pre>	
3.	 (User Task – „Laikyti egzaminą“)	<pre> laikyti_egzamina(\$kintamasis) { // metodo programinis tekstas \$kintamasis = ...; \$kintamasis2 = ...; return \$kintamasis2; } </pre>	Metodus pavadinimu laikyti_egzamina, kuris gražina apskaičiuotą reikšmę (duotuoju atveju laikyto egzamino įvertinimą) kintamojo pavidalu (kintamasis nėra apibrėžiamas, jis tik aprašomas). Įeinantis kintamasis (\$kintamasis) - laikomo egzamino identifikacinis numeris.

Nr.	BPMN elementas	Veiklos paslaugų struktūra	Komentaras
		<pre> laikyti_egzaminus(\$kint) { \$skint = ...; \$segz_pazymys = laikyti_egzamina(\$skint); // metodo programinis tekstas } </pre>	<p>Tas pats (User Task) elementas panaudojamas ir kitokiu pavidalu (nuoroda į metodą). Duotuoju atveju metodo laikyti_egzaminus viduje, paduodant kintamąjį \$kint (egzamino identifikacinis numeris) kreipiamasi į kitą metodą laikyti_egzamina. Kurio apskaičiuota reikšmė priskiriama kintamajam - \$segz_pazymys.</p> <p>* Naujai pridėtas programinis kodas išskirtas mėlyna spalva.</p>
		<pre> &lt;types&gt; &lt;complextypename="laikyti_egzamina"&gt; &lt;elementname=""&gt;&lt;/element&gt; &lt;/complextypename&gt; &lt;/types&gt; </pre>	

### 3.4.3. 2.5 žingsnis: Suformuoti modelyje naudojamus elementus į eigos seką, pagal proceso vykdymo srauto eigą

Šioje dalyje (27 pav.) suformuojama modelyje naudojamų elementų eigos seka (seka nėra vientisa, srautai gali išsiskirti ir/arba susijungti).

Pirmiausia identifikuojami pradžios ir pabaigos elementai. Vėliau (pradedant nuo pradžios elemento) tikrinama ar elementas turi sąsają su išore ir jeigu turi ar sekantis elementas nėra identifikuotasis pabaigos elementas. Tokiu būdu suformuojama elementų eigos seka.



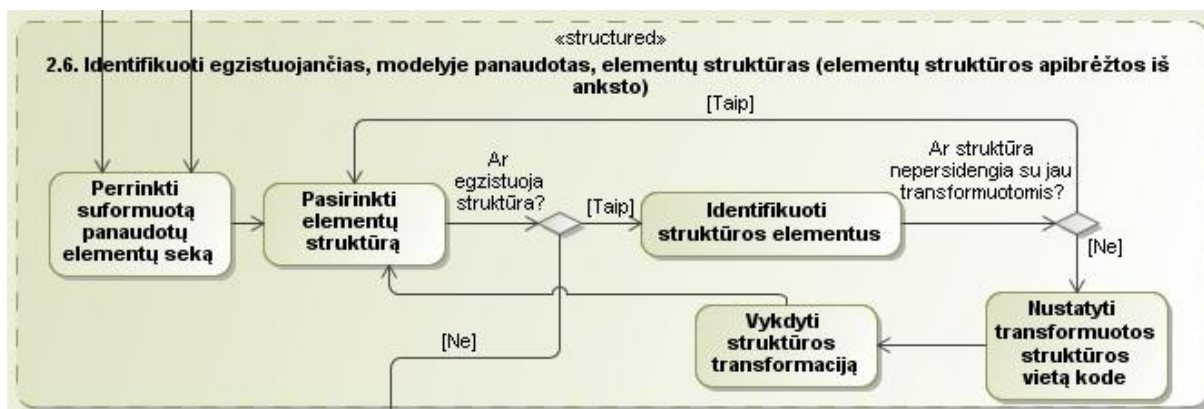
27 pav. Transformacijos algoritmo žingsnio „Suformuoti modelyje naudojamus elementus į eigos seką, pagal proceso vykdymo srauto eigą“ detalizavimas

### 3.4.4. 2.6 žingsnis: Identifikuoti egzistuojančias, modelyje panaudotas, elementų struktūras (elementų struktūros apibrėžtos iš anksto)

„Identifikuoti egzistuojančias, modelyje panaudotas, elementų struktūras (elementų struktūros apibrėžtos iš anksto)“ (28 pav.) proceso metu veiklos proceso modelyje ieškoma elementų struktūrų, kurias būtų galima tiesiogiai transformuoti į veiklos paslaugas.



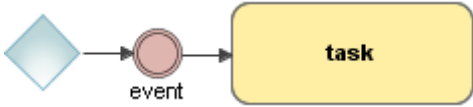
Pirmiausia suformuojama surastų elementų struktūrų seka, vėliau pasirinkus konkrečią struktūrą, identifikavus joje naudojamus elementus, tikrinama ar ši struktūra nepersidengia su jau transformuotomis elementų struktūromis. Nustačius, kad persidengimo nėra, nustatoma jos vieta kode bei atliekama transformacija.


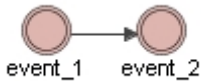



28 pav. Transformacijos algoritmo žingsnio „Identifikuoti egzistuojančias, modelyje panaudotas, elementų struktūras (elementų struktūros apibrėžtos iš anksto)“ detalizavimas

Veiklos proceso modelyje surastų elementų struktūrų transformavimas pateiktas 5 lentelėje (naudojamas veiklos proceso modelio pavyzdys pateiktas 21 pav.).

5 lentelė. Elementų struktūrų transformavimo galimybių vaizdavimas

Nr.	BPMN elementas	Veiklos paslaugų struktūra	Komentaras
1.	 <p>Struktūros elementai:</p> <ul style="list-style-type: none"> <li>• <i>Gateway</i> – šalutinis elementas;</li> <li>• <i>Event</i> – pagrindinis elementas;</li> <li>• <i>Task</i> – šalutinis elementas.</li> </ul>	<pre> if(){ task(); //jeigu išeinantis srautas yra priskirtas Default parametrui } else { task(); //jeigu išeinantis srautas nėra priskirtas Default parametrui } </pre>	<p>Pagrindinis elementas <i>Event</i> yra transformuojamas į nuorodą į metodą (metodas – <i>Task</i> elementas). Metodas pridamas if sąlygoje, jo vieta priklausomai ar išeinantis iš <i>Gateway</i> srautas yra pagrindinis ar ne (ar jis yra <i>Default</i>).</p>
		<pre> &lt;portType name="gateway_event_task"&gt;   &lt;operation name="event"&gt;     &lt;input name="gateway"&gt;       ...     &lt;/input&gt;     &lt;output name="task"&gt;       ...     &lt;/output&gt;   &lt;/operation&gt; &lt;/portType &gt; </pre>	

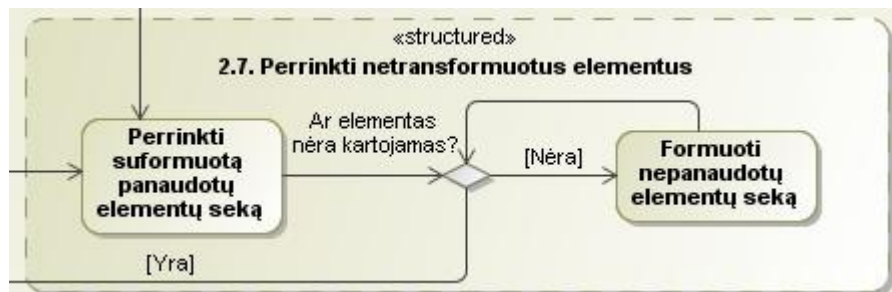
Nr.	BPMN elementas	Veiklos paslaugų struktūra	Komentaras
	 <p>(Task – „Pasirinkti modulį“ sujungta su Event – „Kitas modulis“)</p>	<pre> laikyti_egzaminus(\$kint) {   \$kint = ...;   if(\$like_egz != 0){     // metodo programinis     tekstas   } else {     pasirinkti_moduli();   }   // metodo programinis   tekstas } pasirinkti_moduli() {   // metodo programinis   tekstas } </pre> <pre> &lt;portType name="gateway_event_task"&gt;   &lt;operation name="kitas_modulis"&gt;     &lt;input name="gateway"&gt;       ...     &lt;/input&gt;     &lt;output name="pasirinkti_moduli"&gt;       ...     &lt;/output&gt;   &lt;/operation&gt; &lt;/portType &gt; </pre>	<p>Tinklų sąsajos vieno iš galimų atvejų metu sukuriamas įvykis „Kitas modulis“ privalomai turintis perėjimo sąsają su <i>User Task</i> elementu transformuojamas į naują metodą <i>pasirinkti_moduli()</i> bei šio metodo iškvietimo komandą, kuri patalpinama pagrindiniame metode <i>laikyti_egzaminus()</i>. Iškvietimo nuoroda įterpiama <i>if</i> (sąlygos) sakinyje atitinkamoje <i>Gateway</i> pasirinkimo vietoje.</p>
2.	 <p>Struktūros elementai:</p> <ul style="list-style-type: none"> <li>• <i>Event</i> – pagrindinis elementas (<i>event_1</i>);</li> <li>• <i>Event</i> – šalutinis elementas (<i>event_2</i>).</li> </ul>	<pre> event_1(){   // metodo programinis   kodas } </pre> <pre> &lt;message name="event_2"&gt;   &lt;part name="parameters"   element="event_1"&gt;     ...   &lt;/part&gt; &lt;/message&gt; </pre>	<p>Pagrindinis elementas šioje struktūroje <i>Event</i> elementas pavadinimu „event_1“. Jis transformuojamas į metodą.</p>

Nr.	BPMN elementas	Veiklos paslaugų struktūra	Komentaras
	 <p>Du <i>Event</i> elementai („Semestrinis darbas atsiskaitytas“ ir „Įskaita už semestrinį darbą gauta“)</p>	<pre>semestrinis_darbas_atsiskaitytas() { // metodo programinis kodas }  &lt;message name="Įskaita_uz_semestrini_darba_gauta"&gt;   &lt;part name="parameters"&gt;     element="Semestrinis_darbas_atsiskaitytas"&gt;       ...     &lt;/part&gt;   &lt;/message&gt;</pre>	<p>Struktūra transformuojama į metodą pavadinimu <code>semestrinis_darbas_atsiskaitytas</code>.</p>

### 3.4.5. 2.7 žingsnis: Suformuoti likusių netransformuotų elementų seką

Atlikus elementų struktūrų paiešką ir transformavimą, pereinama prie kitos dalies – „Suformuoti likusių netransformuotų elementų seką“ (29 pav.).

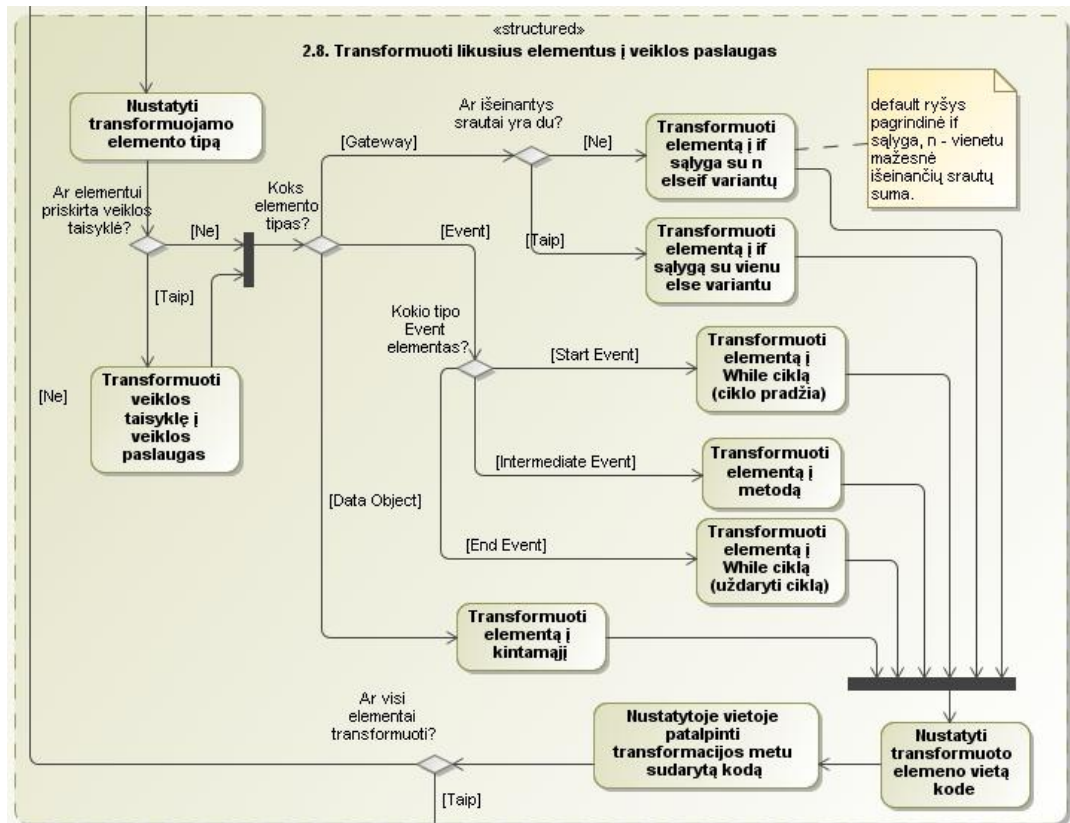
Šio proceso tikslas – perrinkti suformuotą panaudotų elementų seką, juos panaikinant iš turimos veiklos proceso modelyje panaudotų elementų sekos, suformuoti naują nepanaudotų (transformuojant į veiklos paslaugas) elementų sąrašą.



29 pav. Transformacijos algoritmo žingsnio „Suformuoti likusių netransformuotų elementų seką“ detalizavimas

### 3.4.6. 2.8 žingsnis: Transformuoti likusius elementus į veiklos paslaugas

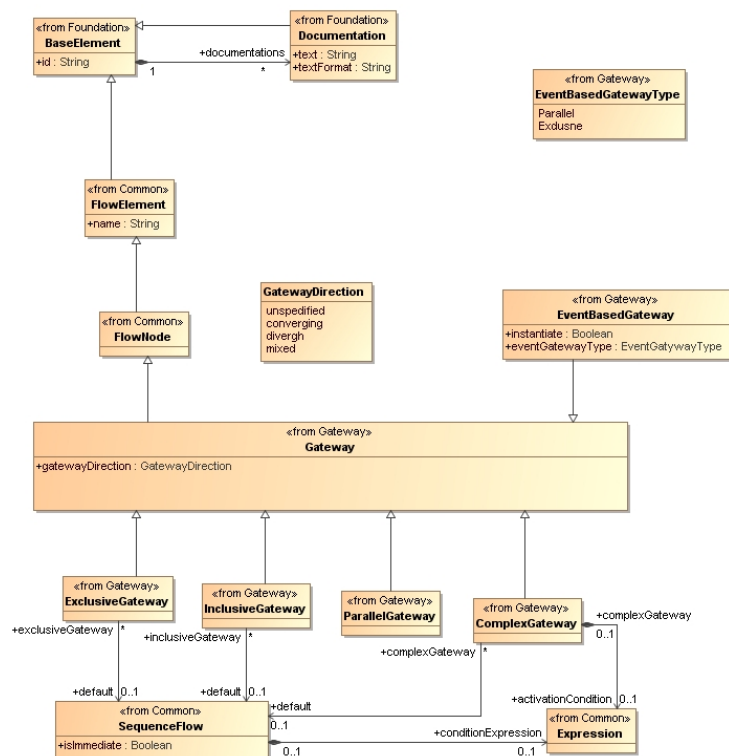
„Transformuoti likusius elementus į veiklos paslaugas“ (30 pav.) proceso metu identifikuojami ir transformuojami likusieji dar netransformuoti elementai.



30 pav. Transformacijos algoritmo žingsnio „Transformuoti likusius elementus į veiklos paslaugas“ detalizavimas

„Gateway“ elementas


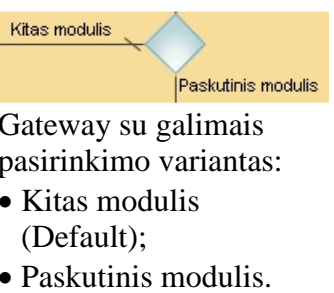
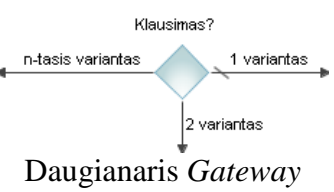
Tinklų sąsajos (angl. *Gateway*) naudojamos siekiant kontroliuoti, kaip srautai sąveikauja, kai jie susilieja ar išsiskiria proceso metu. Tačiau, jei srauto nereikia kontroliuoti, tada šis elementas yra nereikalingas. Tinklų sąsajos metamodelis pateiktas 31 pav.




31 pav. „Gateway“ metamodelis

Veiklos proceso modelyje surastų *Gateway* elementų transformavimas pateiktas 6 lentelėje.

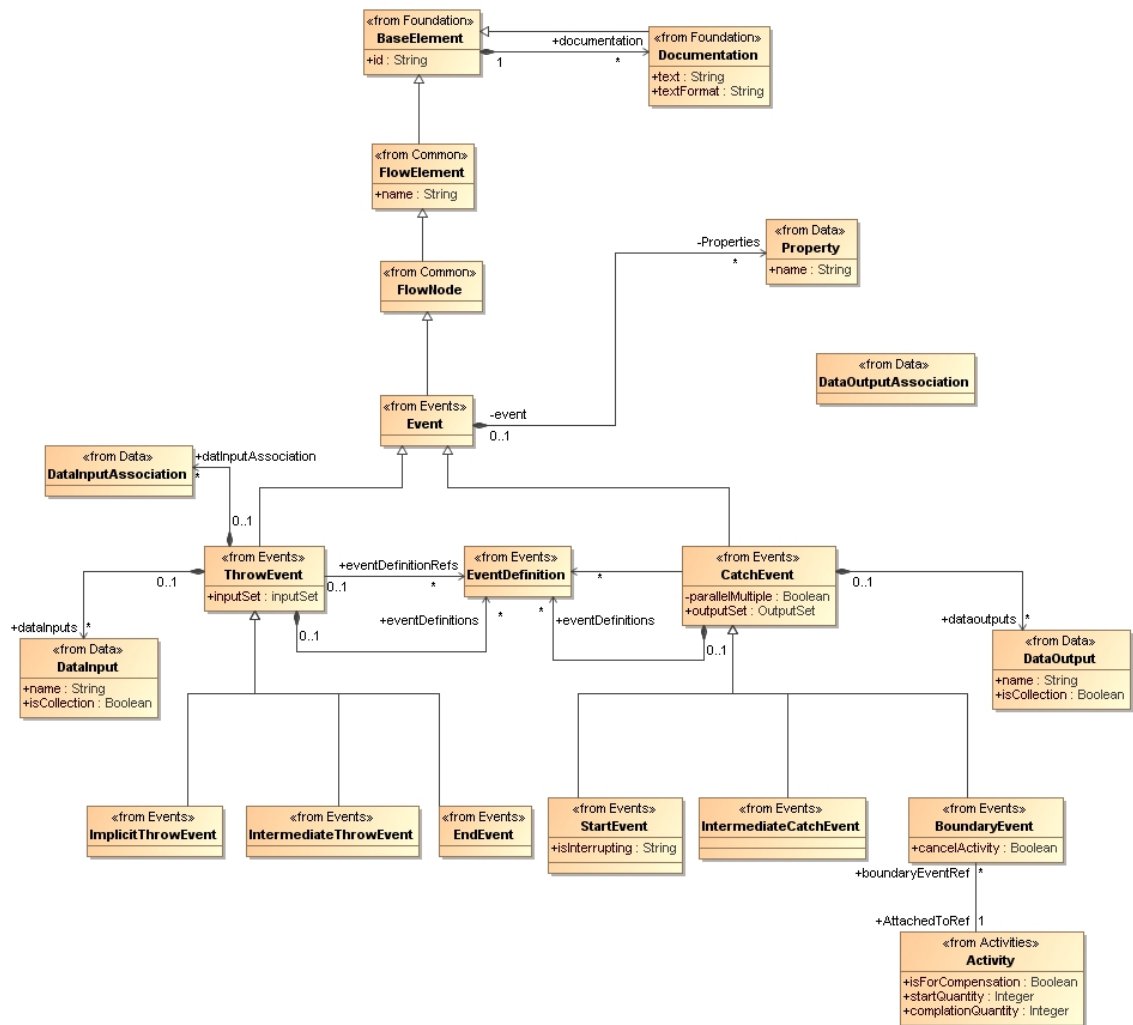
6 lentelė. BPMN Gateway elementų transformavimo galimybių vaizdavimas

Nr	BPMN elementas	Veiklos paslaugų struktūra	Komentaras
1.		<pre>if(){ // 1 variantas } else { // 2 variantas }</pre>	Bendruoju atveju dvi atšakas turintis <i>Gateway</i> elementas atsivaizduoja į <i>if</i> sąlygos sakinį, kurio pagrindinę išpildomąją sąlygą parodo <i>Gateway</i> parametras <i>Default</i> . Kita atšaka tampa <i>else</i> sąlygos vykdomąja funkcija.
		<pre>&lt;binding&gt; &lt;operation name="klausimas"&gt;   &lt;input&gt;   &lt;/input&gt;   &lt;output&gt;     &lt;element name="1_variantas"&gt;&lt;/element&gt;     &lt;element name="2_variantas"&gt;&lt;/element&gt;   &lt;/output&gt; &lt;/operation&gt; &lt;/binding&gt;</pre>	
		<pre>if(){ // <i>Kitas modulis (Default)</i> } else { // <i>Paskutinis modulis</i> }</pre>	Bendrasis <i>Gateway</i> transformavimo atvejis pritaikomas ir gaunamas vykdomasis kodas.
2.		<pre>if(){ // 1 variantas } elseif(){ // 2 variantas }... ...} elseif(){ // n-tasis variantas }</pre>	Bendruoju atveju <i>Gateway</i> elementas turintis daugiau nei dvi atšakas transformuojamas tai pat į <i>if</i> sąlygos sakinį, tačiau <i>else</i> dali kitna. Ji tampa <i>elseif</i> sąlygos sakiniiais (jų skaičius esamų atšakų suma mažesnė 1).

Nr	BPMN elementas	Veiklos paslaugų struktūra	Komentaras
		<pre> &lt;binding&gt;   &lt;operation name="klausimas"&gt;     &lt;input&gt;     &lt;/input&gt;     &lt;output&gt;       &lt;element name="1_variantas"&gt;&lt;/element&gt;       &lt;element name="2_variantas"&gt;&lt;/element&gt;       &lt;element name="n-tasis_variantas"&gt;&lt;/element&gt;     &lt;/output&gt;   &lt;/operation&gt; &lt;/binding&gt; </pre>	
	 <p data-bbox="379 987 663 1093">Daugianaris <i>Gateway</i> („Ar egzaminas išlaikytas?“)</p>	<pre> //ar egzaminas išlaikytas? if(){ // Taip } elseif() { // Ne, pirma kartą } elseif(){ // Ne, nebe pirmą kartą } </pre>	<p>Bendrasis <i>Gateway</i> transformavimo atvejis pritaikomas ir gaunamas vykdomasis kodas. Sukuriamas komentaras (jeigu egzistuoja klausiamasis sakiny) prieš <i>if</i> sąlygą, tada pati <i>if</i> sąlygos sakiniai.</p>
		<pre> &lt;binding&gt;   &lt;operation name=""&gt;     &lt;input&gt;     &lt;/input&gt;     &lt;output&gt;       &lt;element name="Taip"&gt;&lt;/element&gt;       &lt;element name="Ne_nebe_pirma_karta"&gt;&lt;/element&gt;       &lt;element name="Ne_pirma_karta"&gt;&lt;/element&gt;     &lt;/output&gt;   &lt;/operation&gt; &lt;/binding&gt; </pre>	

„Event“ elementas


Įvykis (*angl. Event*) (32 pav.) yra kažkas, kas atsitinka proceso metu. Šie įvykiai turi įtakos proceso srautui ir paprastai sukelia poveikį ir reikalauja reakcijos arba leidžia reakciją. Sąvoka įvykis apima daug dalykų procese. Veiklos pradžioje ir pabaigoje bei pakeičiant dokumento būklę.




32 pav. „Events“ metamodelis

Veiklos proceso modelyje surastų Event elementų transformavimas pateiktas 7 lentelėje.

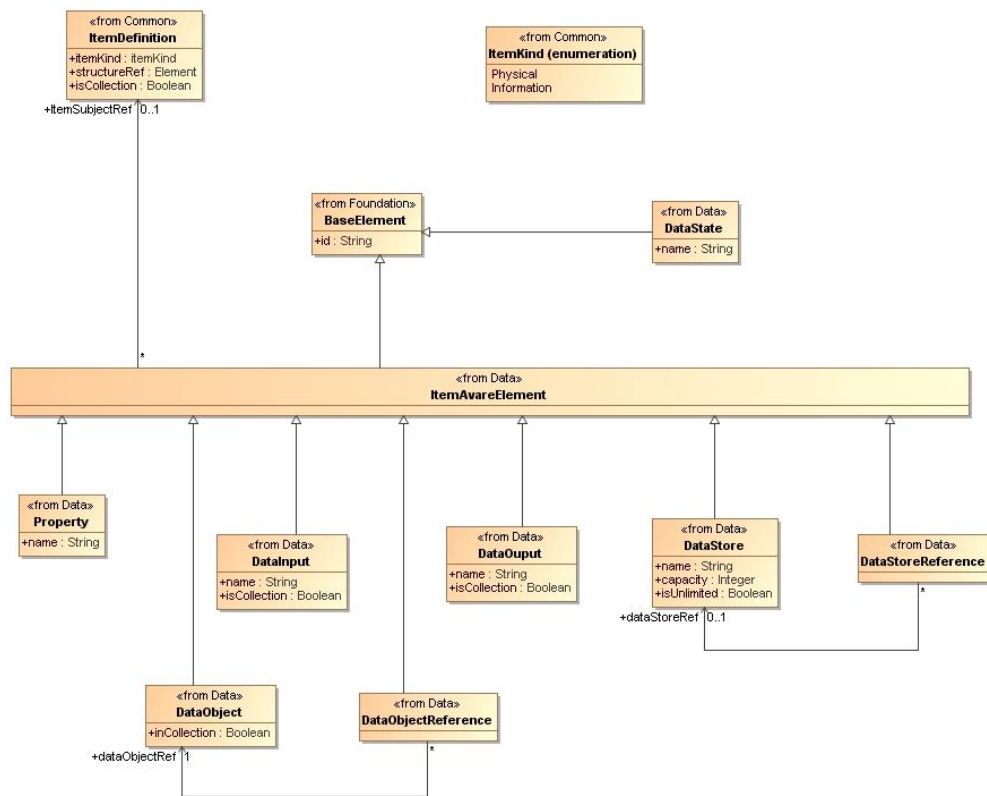
7 lentelė. BPMN Event elementų transformavimo galimybių vaizdavimas

Nr.	BPMN elementas	Veiklos paslaugų struktūra	Komentaras
1.	 event	<pre>event(){ // metodo programinis kodas }</pre>	Bedroju atveju Event elementas gali būti transformuojamas į: <ul style="list-style-type: none"> <li>• metodą;</li> </ul>
<pre>event();</pre>		<ul style="list-style-type: none"> <li>• nuorodą į metodą;</li> </ul>	
<pre>\$kintamasis = event();</pre>		<ul style="list-style-type: none"> <li>• nuorod1 į metodą, kuris grąžina reikšmę (return metodas);</li> </ul>	
<pre>\$event = "";</pre>		<ul style="list-style-type: none"> <li>• kintamąjį;</li> </ul>	
<pre>while(\$event){ // programinis kodas }</pre>		<ul style="list-style-type: none"> <li>• while ciklą.</li> </ul>	
<pre>&lt;element name="event"&gt; &lt;/element&gt;</pre>			

Nr.	BPMN elementas	Veiklos paslaugų struktūra	Komentaras
	 <p>Galutinis pažymys už modulį (Event – „Galutinis pažymys už modulį“)</p>	<pre>galutinis_pazymys_uz_moduli() { // metodo programinis tekstas \$skintamasis = ...; return \$skintamasis; }</pre>	Įvykis „Galutinis pažymys už modulį“ transformuojamas į metodą, kurio pagrindinė paskirtis gražinti kintamojo (apskaičiuotą) kintamojo reikšmę.
		<pre>&lt;element name="galutinis_pazymys_uz_moduli"&gt; &lt;/element&gt;</pre>	

„Items and Data“ elementai

Tradiciniai reikalavimai veiklos proceso modeliavime yra tai, kad būtų galimybė modeliuoti elementais (fiziniais ar informaciniais elementais), kurie sukuriama, manipuluojami ir naudojami gaminant proceso vykdymą. Elementų ir duomenų metamodelis pateiktas 33 pav.





33 pav. „Items and Data“ metamodelis

Veiklos proceso modelyje surastų *Items and Data* elementų transformavimas pateiktas 8 lentelėje.



8 lentelė. BPMN Items and Data elementų transformavimo galimybių vaizdavimas

Nr.	BPMN elementas	Veiklos paslaugų struktūra	Komentaras
1.	 data object Data Object	$\$data\_object = „“;$	Bendroju atveju <i>Data Object</i> elementas transformuojamas į kintamąjį. Pavadinimas sudaromas modifikuojant elemento (modelyje) pavadinimą – visas didžiąsias ir lietuviškas reides (jas paverčiant mažosiomis ir/arba lotyniškėmis raidėmis) ir panaikinant visus tarpus (vietoje jų naudojant simbolį „_“).
		$\langle element\ name=“\ data\_object”\rangle$ $\langle /element\rangle$	
	 Egzaminų sesijos tvarkaraštis Data Object („Egzaminų sesijos tvarkaraštis“)	$\$egzaminu\_sesijos\_tvarkaraštis = ““;$	Sukuriamas kintamasis egzaminu_sesijos_tvarkaraštis.
		$\langle element\ name=“\ egzaminu\_sesijos\_tvarkaraštis”\rangle$ $\langle /element\rangle$	

### 3.4.7. 2.9 žingsnis: Pateikti vientisą transformacijos metu sudarytą vykdomąjį kodą

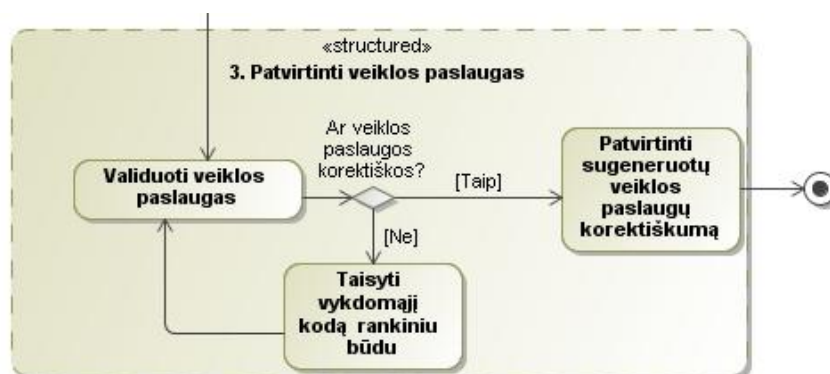
Atlikus visų elementų ir jų struktūrų transformaciją iš veiklos proceso modelio į veiklos paslaugas, jis atitinkama forma pateikiamas vartotojui (34 pav.). Šios veiklos pagrindinis tikslas ir yra vartotojui perteikti vykdomąjį kodą jam suprantamu pavidalu.



34 pav. Transformacijos algoritmo žingsnis „Pateikti vientisą transformacijos metu sudarytą vykdomąjį kodą“

### 3.5. 3 žingsnis: „Patvirtinti veiklos paslaugas“

Paskutinis transformacijos žingsnis – „Patvirtinti veiklos paslaugas“ (35 pav.). Šį veiksmą atlieka sistemos vartotojas (programuotojas). Esminis žingsnio akcentas – gautą vykdomąjį kodą patikrinti, jeigu yra klaidų jas pašalinti. Jei klaidų nėra aptinkama arba jos ištaisomos, transformacijos metu gautas vykdomasis kodas yra patvirtinamas, kaip tinkamas naudoti vykdant tolimesnius projekto etapus.



35 pav. Transformacijos algoritmo žingsnio „Validuoti veiklos paslaugas“ detalizavimas

*Semantics of Business Vocabulary and Business Rules (SBVR)* yra *Object Management Group (OMG)* priimtas standartas. Šis standartas skirtas formaliai ir kartu išsamiai, pasinaudojus natūralios kalbos išraiškomis, aprašyti veiklos apribojimus.

*SBVR* yra natūralios kalbos veiklos taisyklių modeliavimas. *SBVR* formuojamos remiantis kalbotyra ir formalia logika. Jos išreiškiamos kalbos loginėmis struktūromis, vadinamosiomis semantinėmis formulėmis.

*SBVR* yra skirtas išreikšti veiklos žodyną ir veiklos taisykles bei nurodyti verslo informacinių sistemų reikalavimus natūralia kalba.

Veiklos taisyklė yra veiklos sąlygų išraiška. Visos veiklos taisyklės turi tą pačią paskirtį, t.y. sąrašą sąlygų, kurias turi patenkinti prieš atliekant veiksmų sąrašą. Veiklos taisyklė yra sudaryta iš raktinių žodžių, terminų, veiksmazodžių bei faktų. Terminai ir faktai yra saugomi veiklos žodyne. Šie identifikuojami kertiniai veiklos taisyklės sakinio fragmentai panaudojami ją transformuojant į veiklos paslaugas (*WebServices*). Žemiau 9 lentelėje pateikiama galimo transformavimo aprašas.

9 lentelė. Veiklos taisyklių transformavimo į veiklos paslaugas aprašas

Nr.	SBVR išraiška	WebService išraiška	Pastaba
1	Šis <u>objektas</u> privalo <u>turėti</u> <u>kažką</u> , kad <u>galėtų</u> atlikti kokį veiksmą.	if ( <i>objektas turi kažką</i> ){ // gali atlikti veiksmą. }	Jeigu įvardintame veiklos taisyklės sakinyje apibrėžiamas objektas turi pasirinkimą atlikti, nuspręsti, pakeisti ar kitaip įtakoti kažką, šio sakinio atspindys veiklos paslaugose būtų <i>if</i> sąlygos sakiny.

Nr.	SBVR išraiška	WebService išraiška	Pastaba
		<pre> &lt;types&gt;   &lt;element name=" objektas"&gt;     &lt;complex type&gt;       &lt;sequence&gt;         &lt;element name="" /&gt;       &lt;/sequence&gt;     &lt;/complex type&gt;   &lt;/element&gt; &lt;/types&gt;  &lt;message name=""&gt;   &lt;part name="" element="objektas"&gt; &lt;/message&gt; </pre>	
2	<p>Šis <u>objektas</u> privalo <u>turėti kažką</u> (a,b ar c), kad <u>galėtų atlikti atitinkamai pirmą, antrą ar trečią veiksmus.</u></p>	<pre> switch (<i>objektas turi kažką</i>){   care <i>turi a</i>;   // gali atlikti pirmą veiksmą;   break;   care <i>turi b</i>;   // gali atlikti antrą veiksmą;   break;   care <i>turi c</i>;   // gali atlikti trečią veiksmą;   break; } </pre> <pre> &lt;types&gt;   &lt;element name=" objektas"&gt;     &lt;complex type&gt;       &lt;sequence&gt;         &lt;element name="" /&gt;       &lt;/sequence&gt;     &lt;/complex type&gt;   &lt;/element&gt; &lt;/types&gt;  &lt;message name="tureti_kazka"&gt;   &lt;part name="a" element="objektas"&gt;   &lt;part name="b" element="objektas"&gt;   &lt;part name="c" element="objektas"&gt; &lt;/message&gt; </pre>	<p>Jeigu veiklos taisyklės sakinyje apibrėžiamas objektas aukščiau paminėtus veiksmus gali atlikti pasirinkdaas iš daugiau nei dviejų alternatyvų, tada atlikus transformavimą į veiklos paslaugas bus sudaromas pasirinkimo (switch) sąlygos sakinyš.</p>

## 4. Veiklos proceso modelio transformavimo į veiklos paslaugas sprendimo realizacija

Veiklos proceso modelio praplėsto veiklos taisyklėmis transformavimo į veiklos paslaugas sprendimo realizacijos dalyje atlikta dalykinės srities analizė, nustatyti funkciniai ir nefunkciniai reikalavimai bei sudaryta dalykinės srities esybių klasių modelis.

### 4.1 Sprendimo reikalavimų specifikacija ir analizė

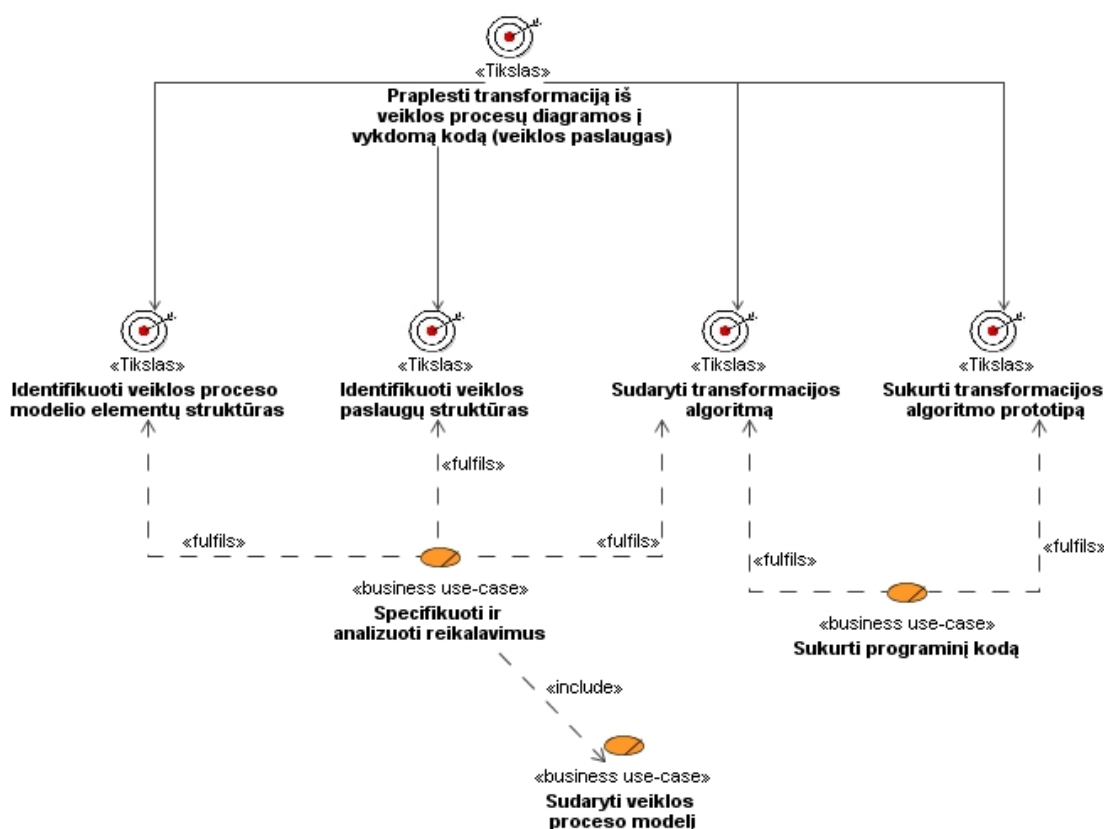
#### 4.1.1 Dalykinės srities analizė

Metodo realizacijos tikslų modelis pateiktas 36 pav.

Pagrindinis magistrinio darbo tikslas yra „Praplėsti transformaciją iš veiklos proceso modelio į vykdomą kodą (veiklos paslaugas)“.

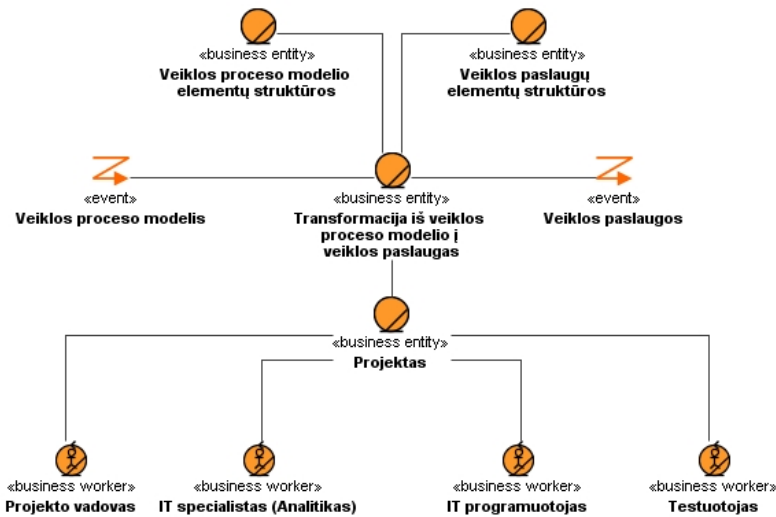
Šiam tikslui įgyvendinti išskiriami keturi tikslai:

1. „Identifikuoti veiklos proceso modelio elementų struktūras“;
2. „Identifikuoti veiklos paslaugų struktūras“;
3. „Sudaryti transformacijos algoritmą“;
4. „Sukurti transformacijos algoritmo prototipą“.



36 pav. Veiklos tikslų modelis

Veiklos objektų modelyje (37 pav.) matyti pagrindinės veiklos esybės. Viena svarbiausių veiklos esybių yra „Projektas“. Su ja susiję visi organizacijos veiklos dalyviai (darbuotojai). Iš kurių IT specialisto (analitiko) siekis – kuo tiksliau perteikti dalykinės srities (vartotojo reikalavimų) esminius akcentus IT programuotojui. Tam pasitelkiama kita šiame modelyje išskirta esybė – „Transformacija iš veiklos proceso modelio į veiklos paslaugas“.

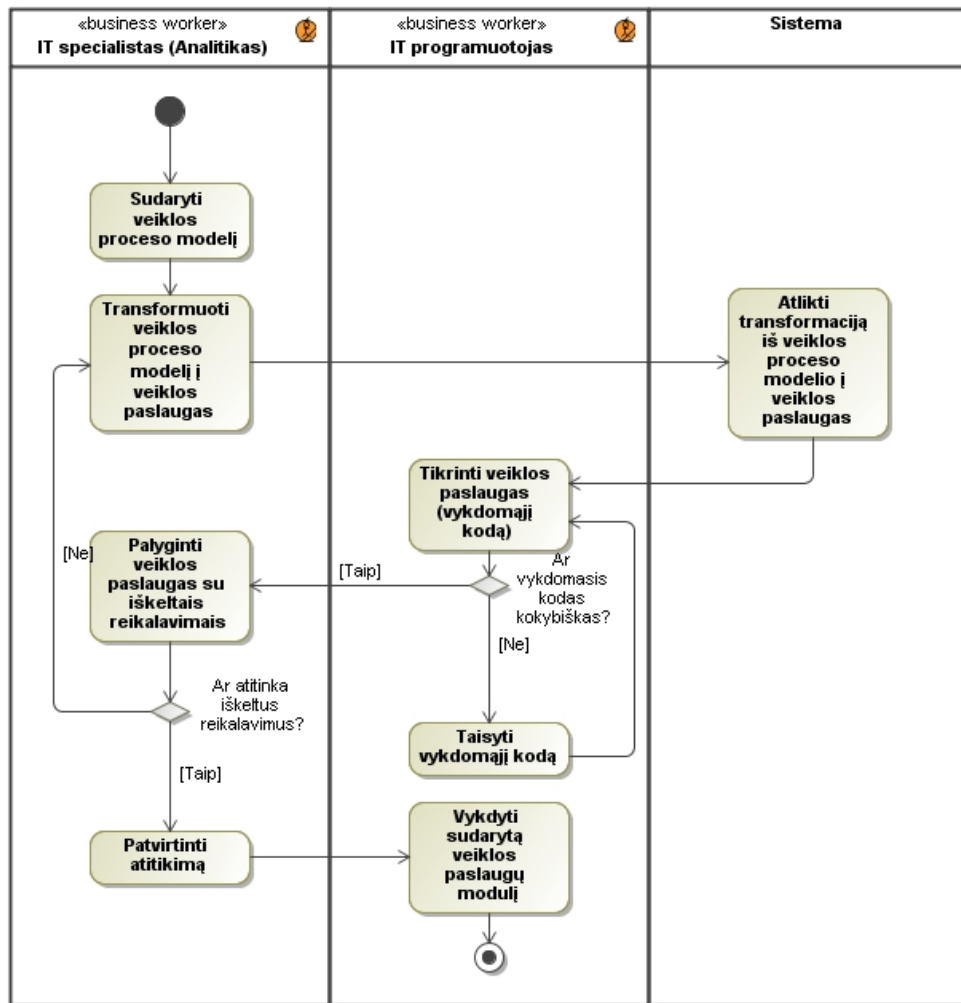


37 pav. Veiklos konceptų (objektų) modelis

Veiklos procesų modelyje (38 pav.) pateikta IT specialisto (analitiko) ir IT programuotojo komunikacija atliekant transformaciją iš veiklos proceso modelio į vykdomąjį kodą.

Pirmiausia IT specialistas (analitikas) sudaro veiklos proceso modelį. Failas su veiklos proceso modelio dokumentacija įvedamas į sistemą, kuri atlieka transformaciją į veiklos paslaugas. Atlikus transformaciją, gautas rezultatas failo pavidalu (pvz.: XML dokumentas) pateikiamas IT programuotojui, kuris tikrina sudaryto vykdomo kodo teisingumą. Jeigu kode nėra klaidų – gaunamas patvirtinimas, jei klaida rasta – IT programuotojas taiso vykdomąjį kodą, kol jis tampa teisingas.

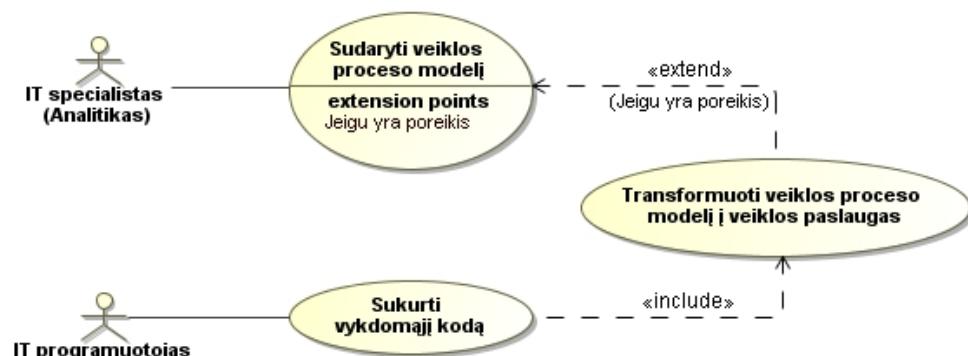
Toliau kodas tikrinamas ar atitinka pradinis vartotojo reikalavimus (tai atlieka IT specialistas). Gavus patvirtinimą, kad kodas atitinka pradinis vartotojo reikalavimus, jis grąžinamas IT programuotojui – tolesniam projekto eigos vykdymui.



38 pav. Veiklos procesų modelis

#### 4.1.2 Funkciniai reikalavimai

Kompiuterizuojamų panaudojimo atvejų (PA) modelis pateiktas (39 pav.). Iš veiklos PA sudaryti nauji kompiuterizuojami PA, jie yra: „Sudaryti veiklos proceso modelį“, „Transformuoti veiklos proceso modelį į veiklos paslaugas“ ir „Sukurti vykdomąjį kodą“.

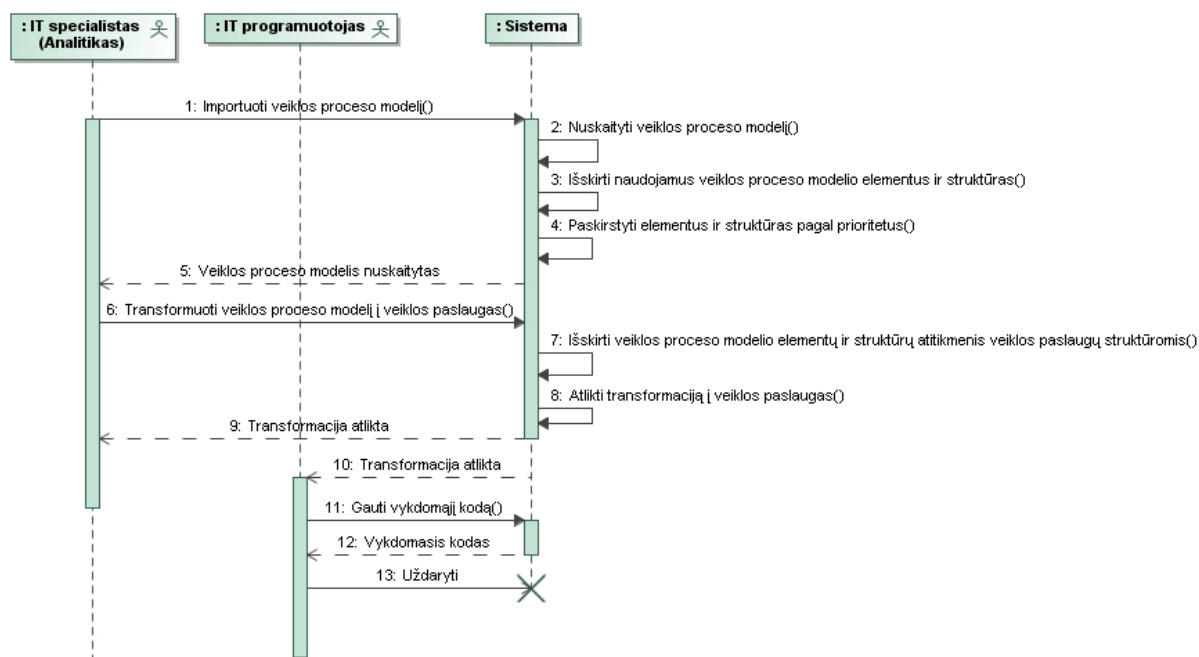


39 pav. Kompiuterizuojamų panaudojimo atvejų modelis

Pagrindinis iš kompiuterizuojamųjų PA yra „Transformuoti veiklos proceso modelį į veiklos paslaugas“ PA. Šį PA detalizuojanti sekų diagrama pateikta 40 pav. Šis PA „extend“ (Jei yra poreikis) ryšiu siejasi su „Sudaryti veiklos proceso modelį“ PA, kas reiškia, kad IT

specialistas (analitikas) nėra įpareigotas privalomai vykdyti šį PA. Tačiau norėdamas tiksliau perteikti dalykinę sritį ir vartotojo reikalavimus, turi galimybę pasinaudoti šiuo metodu. Tuo tarpu pagrindinis detalizuojamas PA su „Sukurti vykdomąjį kodą“ PA siejamas „include“ ryšiu – jei egzistuoja duotasis PA, IT programuotojas privalomai jį taiko.

Kadangi „Transformuoti veiklos proceso modelį į veiklos paslaugas“ PA yra pagrindinis, pateikiama šio PA sekų diagrama (40 pav.). Kiti PA nėra detalizuojami, tačiau pateikiama visų trijų PA specifikacijų lentelės, atitinkamai „Sudaryti veiklos proceso modelį“ (11 lentelė), „Transformuoti veiklos proceso modelį į veiklos paslaugas“ (10 lentelė) ir „Sukurti vykdomąjį kodą“ (12 lentelė) PA.



40 pav. PA „Transformuoti veiklos proceso modelį į veiklos paslaugas“ sekų diagrama

10 lentelė. Panaudojimo atvejo „Transformuoti veiklos proceso modelį į veiklos paslaugas“ specifikacija

PA	„Transformuoti veiklos proceso modelį į veiklos paslaugas“.	
Tikslas.	Atlikti transformaciją iš veiklos proceso modelio į veiklos paslaugas.	
Aprašymas.	Šis panaudojimo atvejis reikalingas norint atlikti transformaciją iš veiklos proceso modelio į veiklos paslaugas.	
Prieš sąlyga	Sudarytas veiklos proceso modelis.	
Aktorius	IT specialistas (analitikas) ir IT programuotojas.	
Sužadavimo sąlyga	Reikia atlikti transformaciją iš veiklos proceso modelio į veiklos paslaugas.	
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	
	Specializuoja PA	
Pagrindinis įvykių šrautas	Sistemos reakcija ir sprendimai	
1.1. Vartotojas nurodo kelią iki dokumento su sudarytu veiklos proceso modeliu.	1.1. Sistema nuskaito dokumentą. 1.2. Sistema identifikuoja naudojamus BPMN elementus modelyje. 1.3. Sistema patvirtina, kad modelis nuskaitytas.	

1.1. Vartotojas pasirenka transformuoti veiklos proceso modelį į veiklos paslaugas.	2.1. Sistema išskiria prioritetinius veiklos proceso modelyje naudojamus elementus. 2.2. Sistema atlieka transformaciją į vykdomąjį kodą. 2.3. Sistema praneša, kad transformacija atlikta. 2.4. Sistema pateikia vykdomąjį kodą.
Po sąlyga:	
1.1. Sistemos duomenų bazėje nėra elemento panaudoto sudarytame veiklos proceso modelyje.	1.1. Sistema identifikuoja ir sudaro netransformuotų elementų sąrašą. 1.2. Sistema pateikia netransformuotų elementų sąrašą.
Alternatyvūs scenarijai	Sistemos reakcija ir sprendimai

11 lentelė. Panaudojimo atvejo „Sudaryti veiklos proceso modelį“ specifikacija

PA	„Sudaryti veiklos proceso modelį“.
Tikslas.	Pagal užsakovo pateiktus kriterijus sudaryti veiklos proceso modelį.
Aprašymas.	Šis panaudojimo atvejis reikalingas norint sudaryti veiklos proceso modelį remiantis užsakovo pateiktais kriterijais.
Prieš sąlyga	Žinomi užsakovo reikalavimai.
Aktorius	IT specialistas (analitikas).
Sužadinimo sąlyga	IT specialistas (analitikas) nori perteikti dalykinę sritį IT programuotojui.
Susiję panaudojimo atvejai	Išplečia PA
	Apima PA
	Specializuoja PA

12 lentelė. Panaudojimo atvejo „Sukurti vykdomąjį kodą“ specifikacija

PA	„Sukurti vykdomąjį kodą“.
Tikslas.	Sukurti vartotojo reikalavimus atitinkantį vykdomąjį kodą.
Aprašymas.	Šis panaudojimo atvejis reikalingas norint sukurti vykdomąjį kodą.
Prieš sąlyga	Atlikta transformacija iš veiklos proceso modelio į veiklos paslaugas.
Aktorius	IT programuotojas.
Sužadinimo sąlyga	Reikia sukurti vykdomąjį kodą.
Susiję panaudojimo atvejai	Išplečia PA
	Apima PA
	Specializuoja PA

#### 4.1.3 Nefunkciniai reikalavimai

Pagrindiniai nefunkciniai sistemos reikalavimai:

##### 1. Išorinė ir vidinė kokybė

##### 1.1. Funkcionalumas

- a) Algoritmas turi tiksliai (be klaidų) transformuoti vartotojo sudarytą veiklos proceso modelį į veiklos paslaugas (vykdomąjį kodą).



- b) Skirtingi algoritmo moduliai turi sėkmingai sąveikauti tarpusavyje ir naudoti turimus išteklius (veiklos proceso modelio elementų aibę ir jų atspindžius veiklos struktūrų pavidalu).
- 1.2. Pilnumas – kuo daugiau *BPMN* elementų bei skirtingo sudėtingumo lygio elementų kombinacijų būtų transformuota į veiklos paslaugas.
  - 1.3. Veiksmingumas – turimi ištekliai naudojami tikslingai, t.y. atliekant transformaciją iš veiklos proceso modelio į veiklos paslaugas, veiklos proceso modelio fragmentai nesikartoja (išvengiama programinio kodo įrašų dubliavimo).
2. Naudojimo kokybė
    - 2.1. Efektyvumas – transformacija atliekama sąlyginai greitai.
    - 2.2. Našumas – didžioji dalis veiklos proceso modelio elementų ir jų skirtingo sudėtingumo lygio elementų kombinacijų perteikiama veiklos paslaugų struktūromis (vykdomuoju kodu).

#### 4.1.4 Dalykinės srities esybių modelis

Dalykinės srities esybių klasių modelis pateiktas 41 pav. Jame pateikta dvi pagrindinės esybių grupės – „Veiklos proceso modelio elementų struktūros“ ir „Veiklos paslaugų elementų struktūros“.



41 pav. Dalykinės srities esybių klasių modelis

#### 4.1.5 Reikalavimų analizės apibendrinimas

Reikalavimų analizės metu identifikuotas pagrindinis magistrinio darbo tikslas (36 pav.) – „Praplėsti transformaciją iš veiklos proceso modelio į vykdomą kodą (veiklos paslaugas)“. Jo įgyvendinimui išskiriami keturi tikslai: „Identifikuoti veiklos proceso modelio elementų struktūras“, „Identifikuoti veiklos paslaugų struktūras“ „Sudaryti transformacijos algoritmą“ ir „Sukurti transformacijos algoritmo prototipą“.

Veiklos objektų modelyje (37 pav.) matyti pagrindinės veiklos esybės. Viena svarbiausių veiklos esybių yra „Projektas“ bei su ja susiję visi organizacijos veiklos dalyviai (darbuotojai).

Veiklos procesų modelyje (38 pav.) pateikta IT specialisto (analitiko) ir IT programuotojo komunikacija atliekant transformaciją iš veiklos proceso modelio į vykdomąjį kodą.

Kompiuterizuojamų panaudojimo atvejų (PA) modelis pateiktas (39 pav.). Iš veiklos PA sudaryti nauji kompiuterizuojami PA, jie yra: „Sudaryti veiklos proceso modelį“, „Transformuoti veiklos proceso modelį į veiklos paslaugas“ ir „Sukurti vykdomąjį kodą“.

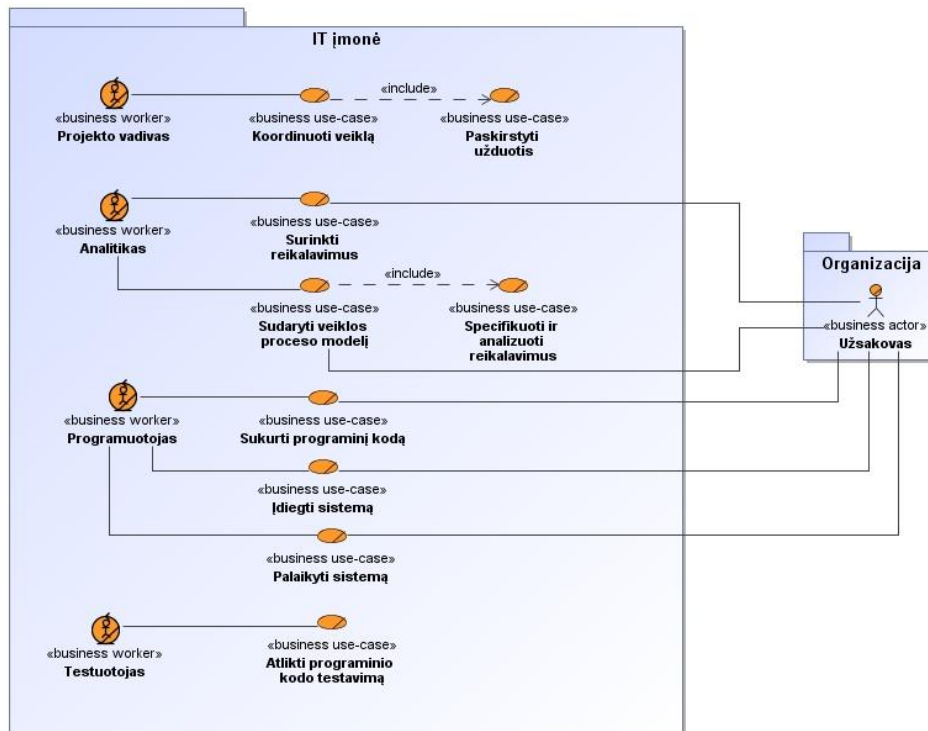
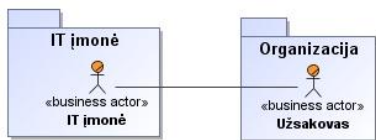
Kadangi „Transformuoti veiklos proceso modelį į veiklos paslaugas“ PA yra pagrindinis, pateikiama šio PA sekų diagrama (40 pav.). Kiti PA nėra detalizuojami, tačiau pateikiama visų trijų PA specifikacijų lentelės, atitinkamai „Sudaryti veiklos proceso modelį“ (5 lentelė), „Transformuoti veiklos proceso modelį į veiklos paslaugas“ (4 lentelė) ir „Sukurti vykdomąjį kodą“ (6 lentelė) PA.

Nustatyti pagrindiniai nefunkciniai sistemos reikalavimai: išorinės ir vidinės kokybės bei naudojimo kokybės.

Dalykinės srities esybių klasių modelis pateiktas 31 paveiksle. Jame pateikta dvi pagrindinės esybių grupės – „Veiklos proceso modelio elementų struktūros“ ir „Veiklos paslaugų elementų struktūros“.

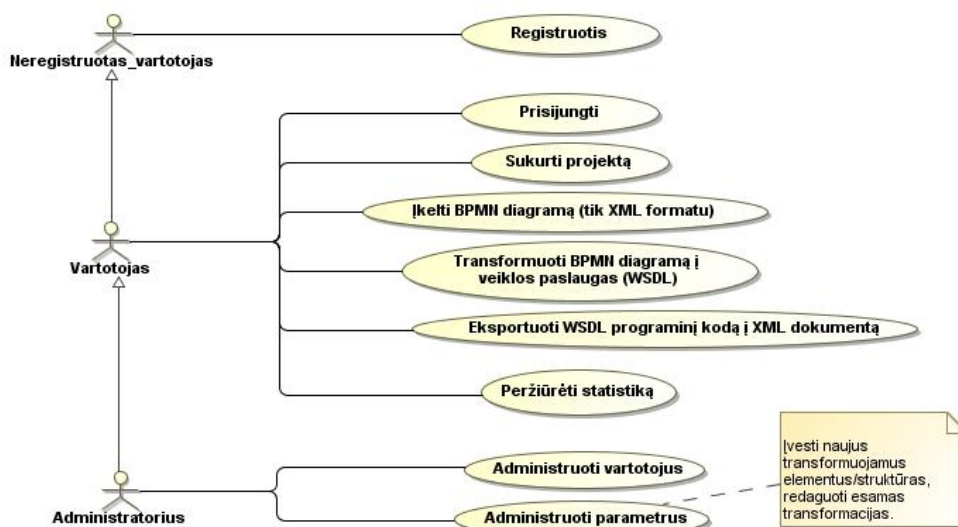
## **4.2 Sistemos architektūra**

Veiklos proceso modelio, praplėsto veiklos taisyklėmis, transformavimo į veiklos paslaugas kontekstinė diagrama pateikta 42 pav. Šioje diagramoje pateikta galima kuriamos sistemos panaudojimo sritis, t.y. tam tikra IT srities įmonė, kurios pagrindinė darbo sritis kurti programinį kodą, bendradarbiauja su įmonės klientais (42 pav. diagramoje: Užsakovai) ir vienoje iš programinio kūrimo stadijų siekiant pagreitinti darbo tempus panaudojama veiklos proceso modelio transformavimo į veiklos paslaugas galimybė (42 pav. diagramoje transformacijos metu dalinai aprėpiamos veiklos: „Sudaryti veiklos proceso modelį“, „Specifikuoti ir analizuoti reikalavimus“ ir „Sukurti programinį kodą“).



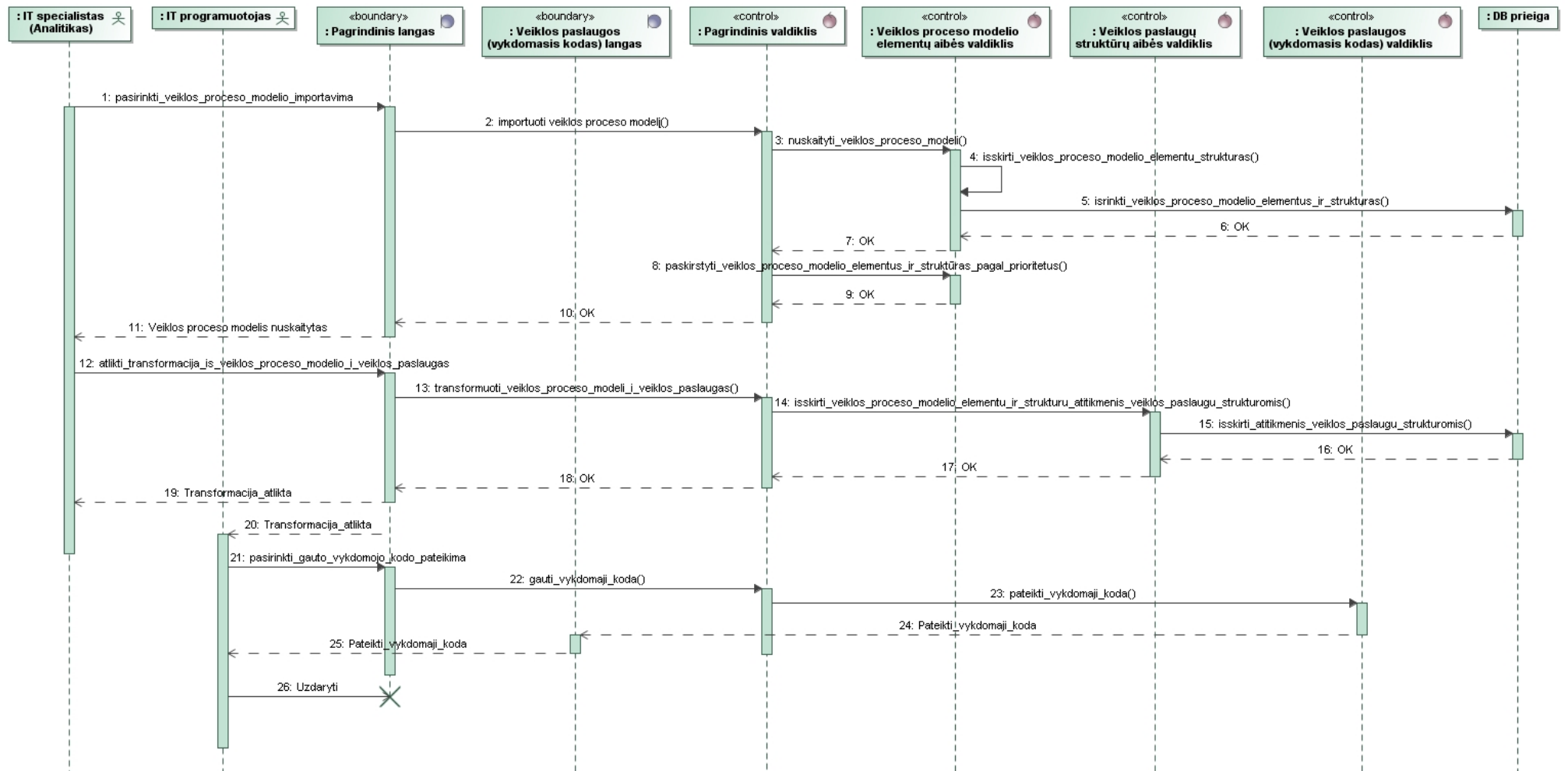
42 pav. Kontekstinė diagrama

Panaudojimų atvejų diagrama pateikta 43 pav. Joje identifikuojami sistemos vartotojai: neregistruotas vartotojas (kuris gali tik užsiregistruoti sistemoje), vartotojas (jam prieinamos funkcijos: prisijungimas, projekto sukūrimas, diagramos transformavimas ir rezultatų peržiūra) bei administratorius (šam vartotojų tipui papildomai prie anksčiau išvardintų funkcijų prieinama vartotojų bei parametrų administravimas).



43 pav. Panaudojimo atvejų diagrama

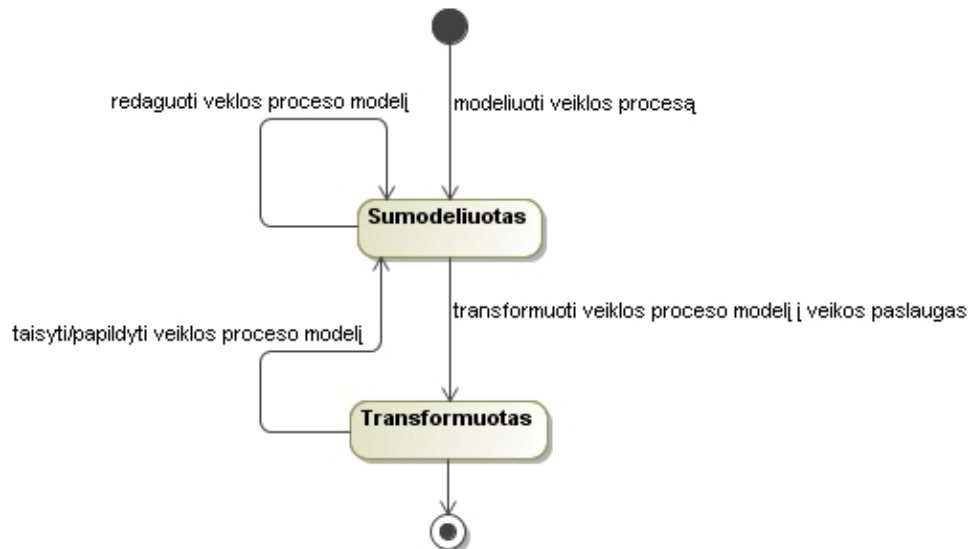
### 4.3 Detalus projektas



44 pav. PA, „Transformuoti veiklos proceso modelį į veiklos paslaugas“ realizacijos sekų daigrama

#### 4.4 Sistemos elgsenos modelis

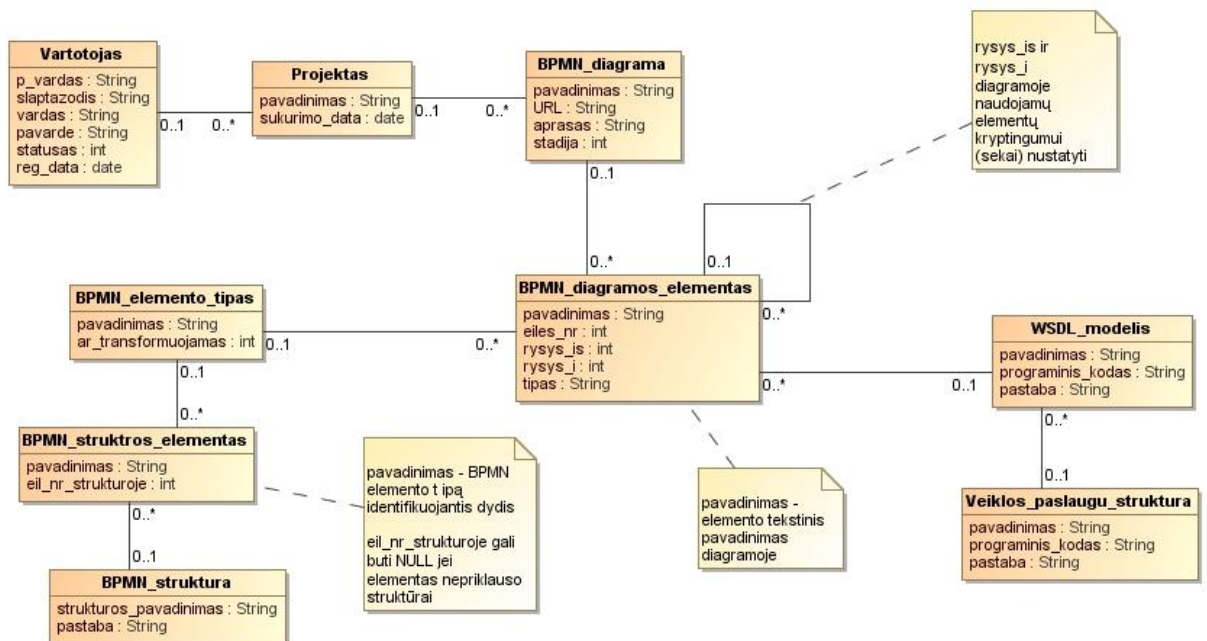
Veiklos proceso modelio būsena atliekant transformaciją (jos eigos metu) į veiklos paslaugas kinta ir šie pokyčiai pavaizduoti 45 pav.



45 pav. Veiklos proceso modelio būsenų diagrama atliekant transformaciją į veiklos paslaugas

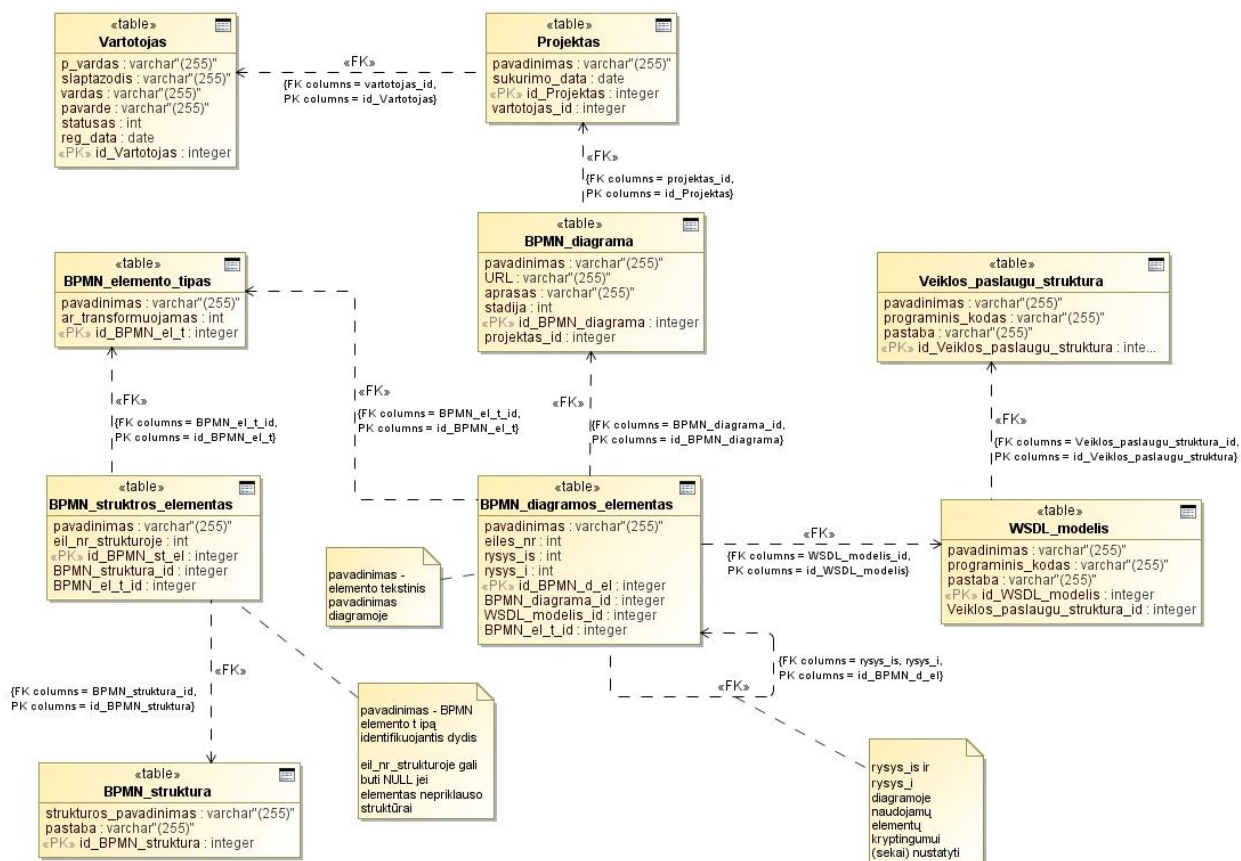
#### 4.5 Duomenų bazės modelis

Sistemos klasių diagrama pateikta 46 pav.



46 pav. Klasių diagrama

Duomeų bazės modelis pateiktas 47 pav.



47 pav. Duomenų bazės modelis

Duomenų bazės lentelių specifikacijos pateiktos 13 lentelėje - 21 lentelėje.

13 lentelė. „Vartotojas“ atributų specifikacija

Lauko pavadinimas	Tipas	Ar būtinas?	Raktas ar indeksas?	Aprašymas
id_Vartotojas	Integer	Taip	Pirminis raktas	Vartotojo identifikacinis numeris sistemoje.
p_vardas	String	Taip		Vartotojo prisijungimo prie sistemos vardas.
slaptazodis	String	Taip		Vartotojo slaptazodis.
vardas	String	Taip		Vartotojo vardas.
pavarde	String	Taip		Vartotojo pavardė.
statusas	Integer	Taip		Vartotojo statusas, nurodantis sistemai kokios funkcijos jam priskiriamos.
reg_data	Date	Taip		Vartotojo užsiregistravimo sistemoje data.

14 lentelė. „Projektas“ atributų specifikacija

Lauko pavadinimas	Tipas	Ar būtinas?	Raktas ar indeksas?	Aprašymas
id_Projektas	Integer	Taip	Pirminis raktas	Projekto identifikacinis numeris sistemoje.
pavadinimas	String	Taip		Projekto pavadinimas.
sukurimo_data	Date	Taip		Projekto sukūrimo data.
vartotojas_id	Integer	Taip	Išorinis raktas	Vartotojo, kuris sukūrė ir kuriam yra pasiekiamas šis projektas identifikacinis numeris.

15 lentelė. „BPMN diagrama“ atributų specifikacija

Lauko pavadinimas	Tipas	Ar būtinas?	Raktas ar indeksas?	Aprašymas
id_BPMN_diagrama	Integer	Taip	Pirminis raktas	BPMN diagramos identifikacinis numeris sistemoje.
pavadinimas	String	Taip		BPMN diagramos pavadinimas.
URL	String	Taip		Kelias iki BPMN diagramos serveryje.
aprasas	String	Ne		BPMN diagramos aprašas (jeigu reikia).
stadija	Integer	Taip		Stadija, kurioje duotuoju metu yra diagrama.
projektas_id	Integer	Taip	Išorinis raktas	Projekto, kuriam priklauso duotoji diagrama identifikacijos numeris.

16 lentelė. „BPMN elemento tipas“ atributų specifikacija

Lauko pavadinimas	Tipas	Ar būtinas?	Raktas ar indeksas?	Aprašymas
id_BPMN_el_t	Integer	Taip	Pirminis raktas	BPMN elemento tipo identifikacinis numeris sistemoje.
pavadinimas	String	Taip		BPMN elemento tipo pavadinimas.
ar_transformuojamas	Integer	Taip		Atributas parodantis ar šis BPMN elementas yra transformuojamas.

17 lentelė. „BPMN struktūros elementas“ atributų specifikacija

Lauko pavadinimas	Tipas	Ar būtinas?	Raktas ar indeksas?	Aprašymas
id_BPMN_st_el	Integer	Taip	Pirminis raktas	BPMN struktūros elemento identifikacinis numeris sistemoje.
pavadinimas	String	Taip		BPMN struktūros elemento pavadinimas.
eil_nr_strukturoje	Integer	Taip		BPMN struktūros elemento eilės numeris sekoje.
BPMN_struktura_id	Integer	Taip	Išorinis raktas	BPMN struktūros identifikacinis numeris sistemoje.
BPMN_el_t_id	Integer	Taip	Išorinis raktas	BPMN elemento tipo identifikacinis numeris sistemoje.

18 lentelė. „BPMN struktūra“ atributų specifikacija

Lauko pavadinimas	Tipas	Ar būtinas?	Raktas ar indeksas?	Aprašymas
id_BPMN_struktura	Integer	Taip	Pirminis raktas	BPMN struktūros identifikacinis numeris sistemoje.
strukturos_pavadinimas	String	Taip		BPMN struktūros pavadinimas.
pastaba	String	Taip		Pastaba/aprašas apie BPMN struktūrą.

19 lentelė. „BPMN diagramos elementas“ atributų specifikacija

Lauko pavadinimas	Tipas	Ar būtinas?	Raktas ar indeksas?	Aprašymas
id_BPMN_d_el	Integer	Taip	Pirminis raktas	BPMN diagramos elemento identifikacinis numeris sistemoje.
pavadinimas	String	Taip		BPMN diagramos elemento pavadinimas.
eiles_nr	Integer	Taip		BPMN diagramos elemento eilės numeris modelyje.
rysys_is	Integer	Ne	Išorinis raktas	BPMN diagramos elemento ryšys su kitu elementu (iš kurio pereinama į šį elementą).
rysys_i	Integer	Ne	Išorinis raktas	BPMN diagramos elemento ryšys su kitu elementu (į kurį pereinama iš šio elemento).
tipas	String	Taip		BPMN diagramos elemento tipas.
BPMN_diagrama_id	Integer	Taip	Išorinis raktas	BPMN diagramos, kurioje yra lokalizuojamas šis elementas, identifikacinis numeris.

Lauko pavadinimas	Tipas	Ar būtinas?	Raktas ar indeksas?	Aprašymas
WSDL_modelis_id	Integer	Taip	Išorinis raktas	WSDL modelio, į kurį transformuojamas šis elementas, identifikacinis numeris.
BPMN_el_t_id	Integer	Taip	Išorinis raktas	BPMN elemento tipo identifikacinis numeris.

20 lentelė. „WSDL\_modelis“ atributų specifikacija

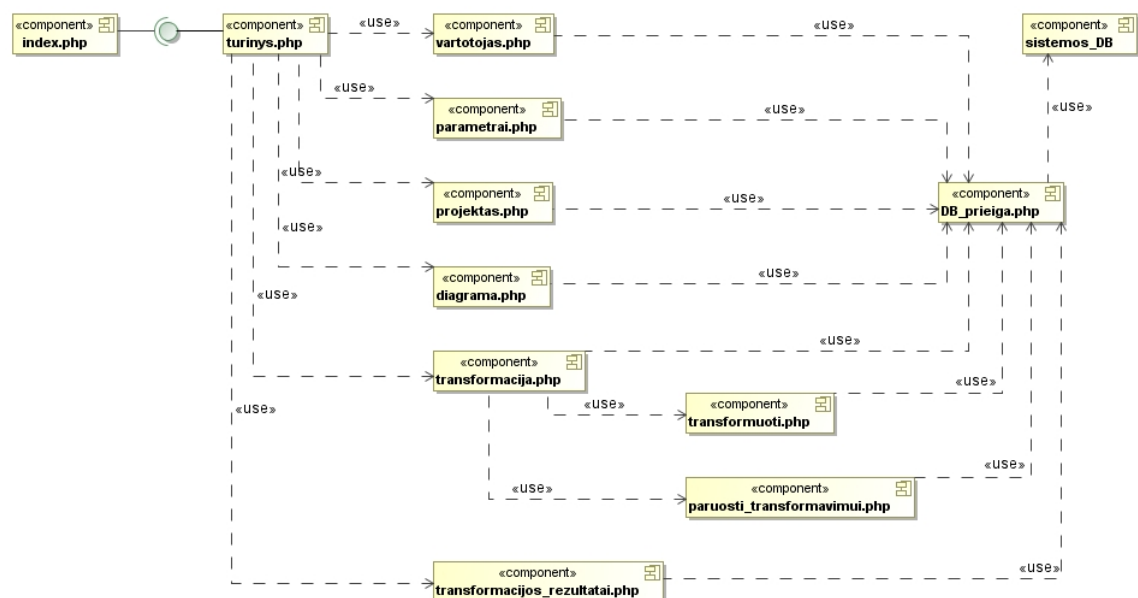
Lauko pavadinimas	Tipas	Ar būtinas?	Raktas ar indeksas?	Aprašymas
id_WSLD_modelis	Integer	Taip	Pirminis raktas	WSDL modelio identifikacinis numeris sistemoje.
pavadinimas	String	Taip		WSDL modelio pavadinimas.
programinis_kodas	String	Taip		WSDL modelio programinis kodas.
pastaba	String	Ne		Pastaba apie WSDL modelį.
veiklos_paslaugu_struktura_id	Integer	Taip	Išorinis raktas	Veiklos paslaugų struktūros identifikacinis numeris sistemoje.

21 lentelė. „Veiklos\_paslaugu\_struktura“ atributų specifikacija

Lauko pavadinimas	Tipas	Ar būtinas?	Raktas ar indeksas?	Aprašymas
id_veiklos_paslaugu_struktura	Integer	Taip	Pirminis raktas	Veiklos paslaugų struktūros identifikacinis numeris sistemoje.
pavadinimas	String	Taip		Veiklos paslaugų struktūros pavadinimas.
programinis_kodas	String	Taip		Veiklos paslaugų struktūros programinis kodas.
pastaba	String	Ne		Pastaba apie veiklos paslaugų struktūrą.

## 4.6 Realizacijos modelis

Sistemos komponentų diagrama pateikta 48 pav.



48 pav. Sistemos komponentų diagrama

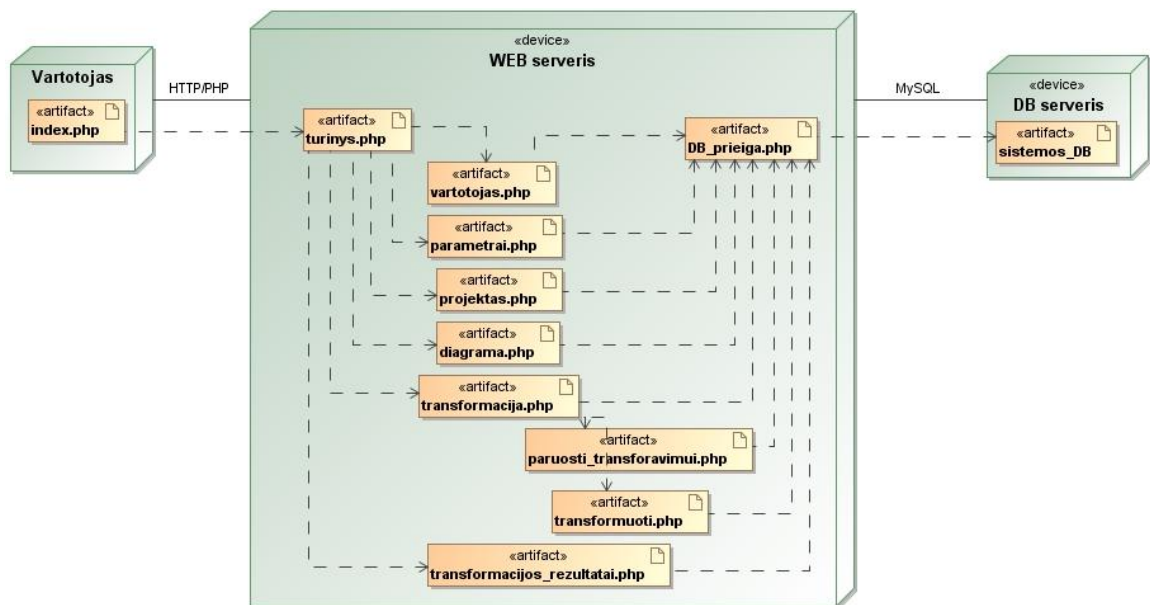


Komponentų specifikacijos pateiktos 22 lentelėje.

22 lentelė. Komponentų specifikacija

Pavadinimas	Paskirtis
index.php	Skirtas išvesti informaciją į ekraną.
turinys.php	Skirtas pagrindiniai informacijai išvesti į ekraną.
varotojas.php	Skirtas vartotojų administravimui.
parametrai.php	Skirtas parametrų administravimui.
projektas.php	Skirtas projektų valdymui.
diagrama.php	Skirtas diagramų valdymui.
transformacija.php	Skirtas transformacijos valdymui.
transformuoti.php	Skirtas atlikti transformavimą.
paruoti_transformavimui.php	Skirtas paruošti duomenis prieš atliekant transformaciją.
transformacijosRezultatai.php	Skirtas transformacijos rezultatų peržiūrai.
DB_prieiga.php	Skirtas užklausomis kreiptis į duomenų bazę.
sistemas_DB	Sistemos duomenų bazė.

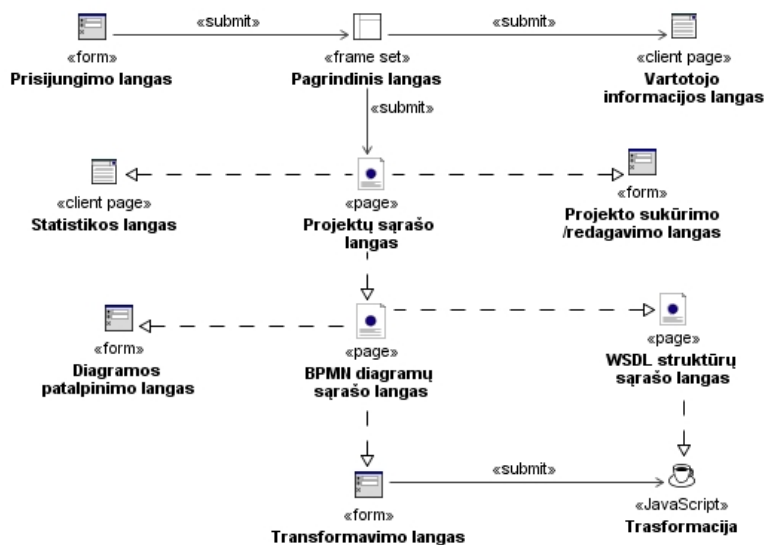
Sistemos diegimo modelis pateikta 49 pav.



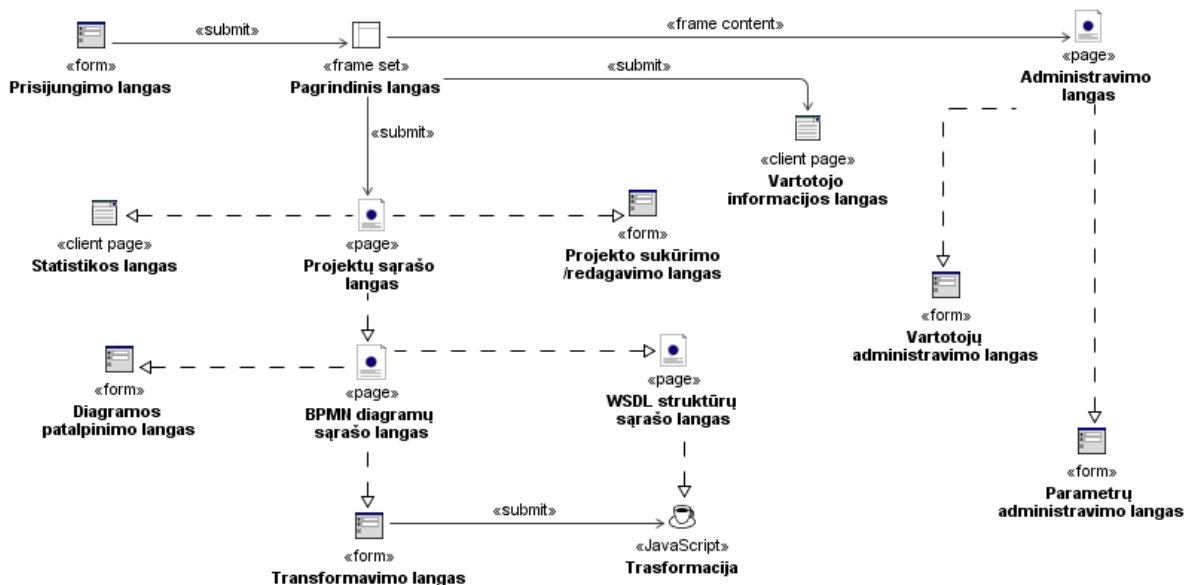
49 pav. Sistemos diegimo modelis

Sistemos navigavimo planas pateiktas 50 pav. Jame išskirtos dvi sritys (atskirtos punktyrine linija), pirmojoje pateikta registruoto vartotojo navigavimo planas, o žemiau punktyrinės linijos esančioje (antrojoje dalyje) pateikta administratoriaus navigavimo planas.

Registruoto vartotojo navigavimo planas



Administratoriaus navigavimo planas



50 pav. Navigavimo planas

## 4.7 Vartotojų analizė

### 4.7.1 Vartotojų aibė, tipai ir savybės

Atliekamo tyrimo, veiklos proceso modelio transformavimo į vykdomą kodą, išskiriamos dvi vartotojų grupės:

2. IT specialistai (analitikai) – atsakingi už veiklos proceso reikalavimų specifیکavimą, veiklos proceso modelio sudarymą ir jo praplėtimą veiklos taisyklėmis;
3. IT programuotojai – kuria vykdomą kodą, specifikuojasi ties veiklos paslaugų kūrimu.

#### 4.7.2 Vartotojų tikslai ir problemos

Visi vartotojai suinteresuoti operatyviai ir tinkamai įgyvendinti savo tikslus. Vartotojų tikslai pateikiami 23 lentelėje.

23 lentelė. Vartotojų tikslai

Vartotojai	Tikslai
IT specialistai (analitikai)	<ul style="list-style-type: none"><li>✓ Greitai ir tiksliai, atspindint probleminę sritį, sudaryti veiklos proceso modelį.</li><li>✓ Greitai ir detaliam praplėsti veiklos proceso modelį veiklos taisyklėmis.</li><li>✓ Suprantamai ir kokybiškai perteikti sukaupią informaciją programuotojams.</li></ul>
IT programuotojai	<ul style="list-style-type: none"><li>✓ Gauti informatyvų, lengvai suprantamą veiklos proceso modelį, praplėstą veiklos taisyklėmis.</li><li>✓ Sukurti, pilnai sudarytą modelį atitinkantį – vykdomą kodą (veiklos paslaugas)</li></ul>

Kaip ir kiekvienoje srityje, taip ir čia iškyla problemų įgyvendinant savo tikslus. Vartotojams kylančios problemos pateiktos 24 lentelėje.

24 lentelė. Vartotojams iškylančios problemos

Vartotojai	Problemos
IT specialistai (analitikai)	<ul style="list-style-type: none"><li>✓ Dviprasmiškumai probleminėje srityje, atspindint reikalavimus veiklos proceso modelyje.</li><li>✓ Neapibrėžtumai praplėčiant veiklos proceso modelį veiklos taisyklėmis.</li></ul>
IT programuotojai	<ul style="list-style-type: none"><li>✓ Neaiškiai perteikiami reikalavimai.</li><li>✓ Dviprasmiškas veiklos procesų modelis.</li></ul>

#### 4.7.3 Vertinimo kriterijai

Šio semestro metu kuriamas transformacijos algoritmo prototipas, prieš kūrimo proceso pradžią sudaryta sistemai keliami funkciniai (žiūrėti: 4.1.2 skirsnyje) ir nefunkciniai (žiūrėti: 4.1.3 skirsnyje) sistemos reikalavimai. Pastarieji identifikuoja pagrindinius parametrus, pagal kuriuos įvertinama sudaryto transformavimo metodo naudingumą.

Veiklos proceso modelio transformavimo į veiklos paslaugas metodo sudaryto prototipo išpildymo efektyvumo ir paties metodo naudingumo įvertinimo kriterijai (25 lentelė):

- sistemos funkcionalumo paprastumas – ar eilieniui vartotojui lengva naudotis vartotojo sąsaja ir sistemos teikiamomis galimybėmis (funkcijomis);
- pilnumas – ar viskas yra atliekama;
- darbo našumas – kaip greitai atliekama transformacija iš veiklos proceso modelio (BPMN diagramos) į veiklos paslaugas (WSDL modulius).

Esminis rezultato kokybės kriterijus yra kuo tiksliau perteikti veiklos proceso modelio kontekstą – veiklos paslaugų pavidalu. Tai atlikti aprėpiant kuo daugiau veiklos proceso modelyje naudojamų elementų ir/ar jų kombinacijų.

Nr.	Kriterijus	Rezultatas
1.	Sistemos funkcionalumo paprastumas	Sistemos navigacija yra elementari, visos sistemos vartotojui prieinamos funkcijos nesunkiai pasiekiamos.
2.	Pilnumas	Realizuota: galimybė transformuoti sistemos bedrinius elementus (suformuoti veiklos paslaugų karkasą).
3.	Darbo našumas	Sistema dirba greitai, transformacijai užtrunkama salyginai trumpą laiko tarpą.

#### 4.8 Testavimo planas ir rezultatai

Testavimo planas (26 lentelė):

1. Atliekamas atskirų sistemos dalių testavimas. Tam, kad įsitikinti, jog jos funkcionuoja tinkamai (taip, kaip reikia).
2. Atliekamas apjungtų (keletos dalių kaip vientiso objekto) testavimas. Šis veiksmas atliekamas tam, kad būtų įsitikinta, kad sujungti komponentai atlieka procesą tinkamai (komponentų apjungimas atliktas tinkamai).
3. Visos sistemos, kaip vientiso objekto (visų sujungtų komponentų) testavimas. Įsitikinti, kad visi sujungti komponentai į vieną vientisą sistemą veikia teisingai.

Planuojama panaudoti palaipsninį testavimą. Realizavus modulio dalį, ji bus testuojama. Jeigu testo rezultatai bus teigiami, modulis bus integruojamas į sistemą. Netinkamai suprogramavus, ar suklydus programuojant bus iš naujo atliekami programavimo darbai, arba esamame sprendime ieškoma klaidų.

26 lentelė. Testavimo planas

Nr.	Testavimo veiksmų seka	Laukiama sistemos reakcija į atliekamą veiksmą
<b>1. Prisijungimas prie sistemos</b>		
1.1	Neįvestas vartotojo vardas ir/arba slaptažodis. Bandoma prisijungti.	Pranešimas „Prisijungimo vardas ir/arba slaptažodis neįvesta.“.
1.2	Įvestas neteisingas vartotojo vardas ir/arba slaptažodis. Bandoma prisijungti.	Pranešimas „Prisijungimo duomenys neteisingi“.
1.3	Įvesti teisingi prisijungimo duomenys. Bandoma prisijungti.	Sistema išveda visų sistemoje esančių dokumentų sąrašą su detalios paieškos forma.
<b>2. Projekto sukūrimas</b>		
2.1	Neįvestas projekto pavadinimas. Bandoma sukurti.	Pranešimas „Projektas nebuvo sukurtas, nes pavadinimas nebuvo įvestas.“.
2.2	Įvestas pavadinimas panaudojant nekorektiškus simbolius. Bandoma sukurti naują projektą.	Pranešimas „Projektas nebuvo sukurtas, nes pavadinime naudojami nekorektiški simboliai.“.
2.3	Įvestas pavadinimas naudojant tik leistinus simbolius. Bandoma sukurti naują projektą.	Projektas sukuriamas, vartotojui išvedamas pranešimas, apie sėkmingą jo sukūrimą.
<b>3. Diagramos patalpinimas</b>		

Nr.	Testavimo veiksmų seka	Laukiama sistemos reakcija į atliekamą veiksmą
3.1	Neįvedama diagramos pavadinimas ir/arba jos aprašymas, nurodomas kelias kompiuteryje iki diagramos failo. Bandoma įkelti diagramą.	Pranešimas „Pateikta nekorektiška informacija. Neįvesta pavadinimas ir/arba aprašymas“.
3.2	Įvedama neteisingas diagramos pavadinimas ir/arba jos aprašymas, (ne)nurodomas neteisingas kelias kompiuteryje iki diagramos failo. Bandoma įkelti diagramą.	Pranešimas „Pateikta nekorektiška informacija. Įvestas pavadinimas ir/arba aprašymas naudojant nelesitinus simbolius. Nerastas diagramos failas.“.
3.3	Įvedama diagramos pavadinimas ir jos aprašymas, nurodomas kelias kompiuteryje iki diagramos failo. Bandoma įkelti diagramą.	Vartotojui pranešama, kad diagrama sėkmingai patalpinta.
4. Elementų identifikavimas		
4.1	Bandoma identifikuoti elementus. Netinkamas formatas, nerandama nei vieno elemento.	Pranešimas „Netinkamas diagramos failo formatas, nepavyko identifikuoti nei vieno elemento.“.
4.2	Bandoma identifikuoti elementus. Dokumento formatas tinkamas, nepanaudota nei vienas elementas.	Pranešimas „Nepavyko identifikuoti nei vieno elemento.“.
4.3	Bandoma identifikuoti elementus.	Vartotojui pranešama, kad transformacija įvykdyta.
5. Transformacijos rezultatų peržiūra		
5.1	Bandoma peržiūrėti transformacijos rezultatus. Nepavykus transformavimui.	Pranešimas „Nepavyko transformuoti diagramos“.
5.2	Bandoma peržiūrėti transformacijos rezultatus.	Vartotojui išvedama transformavimo rezultatai.

Kiekviena sistemos atliekama funkcija yra testuojama pagal aukščiau pateiktą lentelę, nuosekliai vykdant punktus ir žiūrint, ar sistemos reakcija į veiksmus atitinka lentelėje pateiktą.

Pirmiausia patikrinkime, sistemoje registruoto vartotojo prisijungimą prie sistemos. Vartotojas prisijungdamas neįveda prisijungimo vardo ir/arba slaptažodžio (51 pav.), išvedamas klaidos pranešimas (52 pav.). Vartotojui įvedus prisijungimo duomenis, tačiau tai padarius nekorektiškai išvedamas klaidos pranešimo langas (53 pav.), nepadarius klaidos (54 pav.) aktyvus vartotojas prisijungia prie sistemos.

**Prisijungimas prie sistemos**

Prisijungimo duomenys


Prisijungimo vardas	4adm
Slaptažodis	

[Prisijungti](#)

51 pav. Prisijungimas prie sistemos – neįvestas slaptažodis

**Prisijungimas**

Klaida

 Prisijungimo vardas ir/arba slaptažodis neįvesta.

[\[ Atgal \]](#)

52 pav. Prisijungimas prie sistemos – klaida (neįvestas prisijungimo vardas ir/arba slaptažodis)

**Prisijungimas**

Klaida

 Prisijungimo duomenys neteisingi.

[\[ Atgal \]](#)

53 pav. Prisijungimas prie sistemos – klaida (neteisingi duomenys)

**Prisijungimas prie sistemos**

Prisijungimo duomenys

Prisijungimo vardas	4adm
Slaptažodis	••••••••

[Prisijungti](#)

54 pav. Prisijungimas prie sistemos – korektiški duomenys (prisijungiama prie sistemos)

„Naujo projekto sukūrimas“ – tai terpė, kurioje vartotojas vėliau įkels norimą skaičių veiklos proceso modelio dokumentų. Paties projekto sukūrimas yra labai elementarus, tereikia parašyti pavadinimą ir sukurti projektą (55 pav.).

**Naujas projektas**

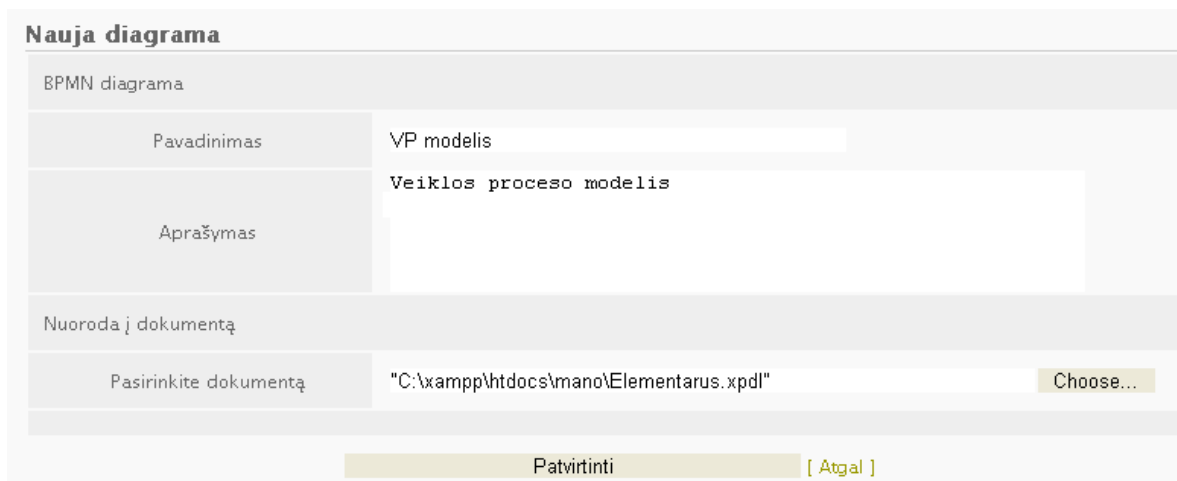
Projektas

Pavadinimas	Veiklos procesas
-------------	------------------

[Sukurti naują projektą](#) [\[ Atgal \]](#)

55 pav. Naujo projekto sukūrimas

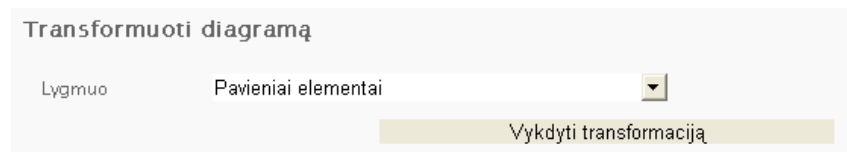
Kita funkcija „Naujos veiklos proceso diagramos įkėlimas“ (56 pav.). Pirma vartotojas iš projekto sarašo pasirenka konkretų projektą, kuriam nori priskirti (įkelti į sistemą) naują diagramą. Tada užpildo privalomus laukus (pavadinimas, aprašymas ir nuoroda iki failo).



Nauja diagrama	
BPMN diagrama	
Pavadinimas	VP modelis
Aprašymas	Veiklos proceso modelis
Nuoroda į dokumentą	
Pasirinkite dokumentą	"C:\xampp\htdocs\mano\Elementarus.xpdl" <span>Choose...</span>
<span>Patvirtinti</span> <span>[ Atgal ]</span>	

56 pav. Diagramos įkėlimas

„Transformavimo pobūdžio pasirinkimas“ (57 pav.) pasirenkama, koku principu nagrinėti veiklos proceso modelį (galimi pasirinkimai: pavieniai elementai bei pilna transformacija (ir struktūros)).



Transformuoti diagramą	
Lygmuo	Pavieniai elementai
<span>Vykdyti transformaciją</span>	

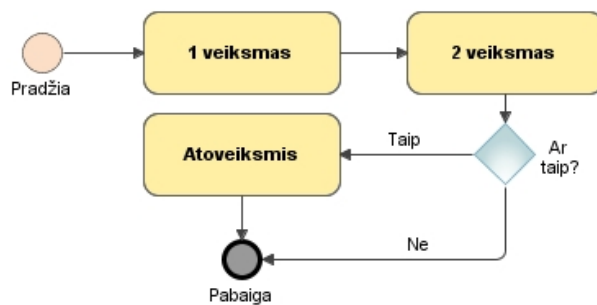
57 pav. Transformavimo pasirinkimas

## 5. Eksperimentinis sistemos tyrimas

Šiame skyriuje paeikiama sudarytojo veiklos proceso modelio transformavimo metodo įvertinimas. Ekspermentinis įvertinimas atliktas autoriaus sukurtu programiniu prototipu. Ekspermentui atlikti panaudoti du elementarus keletos elementų veiklos procesas sumodeliuotas *BPMN* diagramoje ir išsaugotas *XML* dokumente. Atlikus ekspermentinį tyrimą, pateikiamas gautų rezultatų apibendrinimas.

### 5.1. Eksperimentos planas

Norint įvertinti pasiūlyto veiklos proceso modelio, praplėsto veiklos taisyklėmis, transformavimo į veiklos paslaugas metodo tinkamumą buvo sudarytos dvi veiklos procesų diagramos pagal *BPMN* notaciją. Jos panaudojant sukurtą prototipą transformuotos į veiklos paslaugas. Sudarytosios diagramos pateiktos atitinkamai 58 pav. ir 59 pav.



58 pav. Veiklos procesas „su keletu veiklų“ sumodeliuotas su *BPMN* standartu



59 pav. Veiklos procesas „su viena veikla“ sumodeliuotas su *BPMN* standartu

*BPMN* modeliai aprašomi *XML* kalba, todėl juos galima išeksportuoti kaip *XML* failus. Taigi norint ištirti metodą, pirmiausia reikia išeksportuoti *BPMN* modelius į *XML* kalbos pagrindu sudarytus failus (*XML* failo pavyzdys pateiktas – 4 priede). Šių diagramas aprašančių failų turinių fragmentai pateikti atitinkamai 60 pav. ir 61 pav.



```

<?xml version="1.0" encoding="utf8" standalone="yes"?>
<Package Name="Untitled1" Id="_17_0_1_a8602c2_1327140870149_755556_8103" xmlns:ns2="http://www.wfmc.org/2002/
  <PackageHeader>
    <XPDLVersion>2.2</XPDLVersion>
    <Vendor>Nomagic</Vendor>
    <Created>12-00-22T00:14:00</Created>
    <ModificationDate>12-00-21T12:16:42</ModificationDate>
    <LayoutInfo PixelsPerMillimeter="2.8346457"/>
  </PackageHeader>
  <RedefinableHeader>
    <Author>Edminas</Author>
    <Version>-1</Version>
    <Codepage>Cp1257</Codepage>
    <Countrykey>LT</Countrykey>
  </RedefinableHeader>
  <ConformanceClass BPMNModelPortabilityConformance="SIMPLE" GraphConformance="NON_BLOCKED"/>
  <Pages>
    <Page Width="1193.0" Height="824.0" Id="_17_0_1_a8602c2_1327140870149_983048_8102" Name="Untitled1"/>
  </Pages>
  <Pools>
    <Pool MainPool="true" BoundaryVisible="false" Process="_17_0_1_a8602c2_1327141606538_168905_8308" Or
      <NodeGraphicsInfos>
        <NodeGraphicsInfo FillColor="#FFCC33" Width="1193.0" Height="824.0" PageId="_17_0_1_a8602c2_
          <Coordinates YCoordinate="5.0" XCoordinate="5.0"/>
        </NodeGraphicsInfo>
      </NodeGraphicsInfos>
    </Pool>
  </Pools>
  <WorkflowProcesses>
    <WorkflowProcess Name="_17_0_1_a8602c2_1327141606538_168905_8308_wf" Id="_17_0_1_a8602c2_13271416065
      <ProcessHeader>
        <Created>12-00-22T00:14:00</Created>
      </ProcessHeader>
      <Activities>
        <Activity Name="Pradžia" Id="_17_0_1_a8602c2_1327140874495_320826_8132">
          <Event>
            <StartEvent Implementation="Unspecified" Trigger="None"/>
          </Event>
          <NodeGraphicsInfos>
            <NodeGraphicsInfo FillColor="#FFCC33" Width="25.0" Height="25.0" PageId="_17_0_1_a86
              <Coordinates YCoordinate="182.0" XCoordinate="93.0"/>
            </NodeGraphicsInfo>
          </NodeGraphicsInfos>
        </Activity>
      </Activities>
    </WorkflowProcess>
  </WorkflowProcesses>

```

60 pav. Veiklos proceso „su keletu veikly“ XML failo fragmentas

```

<?xml version="1.0" encoding="utf8" standalone="yes"?>
<Package Name="Elementarus" Id="_17_0_1_a8602c2_1327141867037_231357_8313" xmlns:ns2="http://www.wfmc.org/2002/XPDL1.0" x
  <PackageHeader>
    <XPDLVersion>2.2</XPDLVersion>
    <Vendor>Nomagic</Vendor>
    <Created>12-00-22T00:31:00</Created>
    <ModificationDate>12-00-21T12:31:57</ModificationDate>
    <LayoutInfo PixelsPerMillimeter="2.8346457"/>
  </PackageHeader>
  <RedefinableHeader>
    <Author>Edminas</Author>
    <Version>-1</Version>
    <Codepage>Cp1257</Codepage>
    <Countrykey>LT</Countrykey>
  </RedefinableHeader>
  <ConformanceClass BPMNModelPortabilityConformance="SIMPLE" GraphConformance="NON_BLOCKED"/>
  <Pages>
    <Page Width="1193.0" Height="824.0" Id="_17_0_1_a8602c2_1327141867037_417121_8312" Name="Elementarus"/>
  </Pages>
  <Pools>
    <Pool MainPool="true" BoundaryVisible="false" Process="_17_0_1_a8602c2_1327141978354_184137_8402" Orientation="HC
      <NodeGraphicsInfos>
        <NodeGraphicsInfo FillColor="#FFCC33" Width="1193.0" Height="824.0" PageId="_17_0_1_a8602c2_1327141867037
          <Coordinates YCoordinate="5.0" XCoordinate="5.0"/>
        </NodeGraphicsInfo>
      </NodeGraphicsInfos>
    </Pool>
  </Pools>
  <WorkflowProcesses>
    <WorkflowProcess Name="_17_0_1_a8602c2_1327141978354_184137_8402_wf" Id="_17_0_1_a8602c2_1327141978354_184137_840
      <ProcessHeader>
        <Created>12-00-22T00:31:00</Created>
      </ProcessHeader>
      <Activities>
        <Activity Name="Pradžia" Id="_17_0_1_a8602c2_1327141871017_246279_8342">
          <Event>
            <StartEvent Implementation="Unspecified" Trigger="None"/>
          </Event>
          <NodeGraphicsInfos>
            <NodeGraphicsInfo FillColor="#FFCC33" Width="25.0" Height="25.0" PageId="_17_0_1_a8602c2_13271418
              <Coordinates YCoordinate="189.0" XCoordinate="177.0"/>
            </NodeGraphicsInfo>
          </NodeGraphicsInfos>
        </Activity>
      </Activities>
    </WorkflowProcess>
  </WorkflowProcesses>

```

61 pav. Veiklos proceso „su viena veikla“ XML failo fragmentas

Taigi eksperimento eiga vyks pagal veiklos proceso modelio transformavimo į veiklos paslaugas metodo pagrindinius procesus tokia seka:

1. Proceso „Transformuoti veiklos proceso modelį į veiklos paslaugas“ įvertinimas;
2. Proceso „Patvirtinti veiklos paslaugas“ įvertinimas.

## 5.2. Eksperimento eiga

Pirmiausia yra sukuriamas projektas, prie jo prikabinama *BPMN* diagrama (patalpinama *BPMN* diagrama *XML* failo pavidalu – 62 pav.). Tinkamai paruošus diagramą transformavimui, toliau pereinama prie metodo „Transformuoti veiklos proceso modelį į veiklos paslaugas“ žingsnio.

62 pav. Diagramos patalpinimo į sistemą vartotojo sąsajos langas

Pirmiausia „Transformuoti veiklos proceso modelį į veiklos paslaugas“ žingsnio metu yra vykdomas veiklos proceso modelio *XML* failo nuskaitymas. Toliau nuskaitytas turinys naudojamas diagramoje panaudotų elementų identifikavimui.

Nuskaičius duotąsias (58 pav. ir 59 pav.) diagramas gauti rezultatai pateikiami 27 lentelėje.

27 lentelė. Gauti elementai panaudoti nuskaitytose diagramose

Elementų skaičius	„su keletu veiklų“	„su viena veikla“
„Event“	2	2
„Activity“	3	1
„Gateway“	1	0
„Transmission“	6	2

Tęsiant eksperimentą metodo „Transformuoti veiklos proceso modelį į veiklos paslaugas“ žingsnio metu toliau vykdomas identifikuotųjų elementų perrinkimas, siekiant

suformuoti seką (nustatyti elementų išdėstymo diagramoje eiliškumą). Gautas rezultatas pateikiamas 63 pav. ir 64 pav.

Diagramos elementai			
Nr.	Pavadinimas	Eilės nr.	Tipas
1	Pradžia	1	StartEvent
2		2	Transition
3	1 veiksmas	3	Activity
4		4	Transition
5	2 veiksmas	5	Activity
6		6	Transition
7	Ar taip?	7	Gateway
8	Taip	8	Transition
9	Ne	8	Transition
10	Atoveiksmis	9	Activity
11		10	Transition
12	Pabaiga	11	EndEvent

63 pav. Veiklos proceso „su keletu veiklų“ nuskaitytų elementų seka

Diagramos elementai			
Nr.	Pavadinimas	Eilės nr.	Tipas
1	Pradžia	1	StartEvent
2		2	Transition
3	Veiksmas	3	Activity
4		4	Transition
5	Pabaiga	5	EndEvent

64 pav. Veiklos proceso „su viena veikla“ nuskaitytų elementų seka

63 pav. ir 64 pav. pateikti nuskaitytų diagramų duomenys, nurodoma išgryninta informacija, kuri yra įrašoma į duomenų bazės lenteles skirtas BPMN elemento aprašui saugoti, nagrinėjamųjų atvejų (diagramų: veiklos procesas „su keletu veiklų“ – 58 pav. ir veiklos procesas „su viena veikla“ – 59 pav.) diagramose identifikuotų elementų duomenų patalpinimas duomenų bazėje pateiktas atitinkamai 65 pav. ir 66 pav.

d_e_pavadinimas	d_e_eiles_nr	d_e_rsys_is	d_e_rsys_i	BPMN_diagrama_id	WSDL_modelis_id	BPMN_el_t_id
	2	_17_0_1_a8602c2_1327140874495_320826_8132	_17_0_1_a8602c2_1327140881967_716806_8156	16	0	5
Ar taip?	7	_17_0_1_a8602c2_1327140901762_65512_8262	_17_0_1_a8602c2_1327140928335_347886_8284	16	0	4
1 veiksmas	3	_17_0_1_a8602c2_1327140898557_207367_8244	_17_0_1_a8602c2_1327140900040_232252_8253	16	0	3
2 veiksmas	5	_17_0_1_a8602c2_1327140900040_232252_8253	_17_0_1_a8602c2_1327140901762_65512_8262	16	0	3
Pabaiga	11	_17_0_1_a8602c2_1327140928335_347886_8284		16	0	1
Atoveiksmis	9	_17_0_1_a8602c2_1327140918865_233252_8274	_17_0_1_a8602c2_1327140933438_540215_8293	16	0	3
	10	_17_0_1_a8602c2_1327140891447_725617_8221	_17_0_1_a8602c2_1327140879760_917827_8141	16	0	5
	4	_17_0_1_a8602c2_1327140881967_716806_8156	_17_0_1_a8602c2_1327140885535_175948_8182	16	0	5
Taip	8	_17_0_1_a8602c2_1327140889397_508286_8204	_17_0_1_a8602c2_1327140891447_725617_8221	16	0	5
	6	_17_0_1_a8602c2_1327140885535_175948_8182	_17_0_1_a8602c2_1327140889397_508286_8204	16	0	5
Ne	8	_17_0_1_a8602c2_1327140889397_508286_8204	_17_0_1_a8602c2_1327140879760_917827_8141	16	0	5
Pradžia	1	_17_0_1_a8602c2_1327140898557_207367_8244		16	0	2
	4	_17_0_1_a8602c2_1327141872670_265968_8357	_17_0_1_a8602c2_1327141876357_30035_8377	20	0	5
Pradžia	1		_17_0_1_a8602c2_1327141880163_854765_8387	20	0	2
	2	_17_0_1_a8602c2_1327141871017_246279_8342	_17_0_1_a8602c2_1327141872670_265968_8357	20	0	5
Veiksmas	3	_17_0_1_a8602c2_1327141880163_854765_8387	_17_0_1_a8602c2_1327141881838_428460_8396	20	0	3
Pabaiga	5	_17_0_1_a8602c2_1327141881838_428460_8396		20	0	1
	0	_17_0_1_a8602c2_1327560067919_525125_7892	_17_0_1_a8602c2_1327560050008_326484_7797	23	0	5
Taip	0	_17_0_1_a8602c2_1327560077071_120296_7963	_17_0_1_a8602c2_1327560050008_326484_7797	23	0	5
Ne	0	_17_0_1_a8602c2_1327560077071_120296_7963	_17_0_1_a8602c2_1327560067349_225611_7885	23	0	5

65 pav. Veiklos proceso „su keletu veiklų“ identifikuotų BPMN elementų informacija duomenų bazėje

d_e_pavadinimas	d_e_eiles_nr	d_e_rsys_is	d_e_rsys_i	BPMN_diagrama_id	WSDL_modelis_id	BPMN_el_t_id
	2	_17_0_1_a8602c2_1327140874495_320826_8132	_17_0_1_a8602c2_1327140881967_716806_8156	16	0	5
Ar taip?	7	_17_0_1_a8602c2_1327140901762_65512_8262	_17_0_1_a8602c2_1327140928335_347886_8284	16	0	4
1 veiksmas	3	_17_0_1_a8602c2_1327140898557_207367_8244	_17_0_1_a8602c2_1327140900040_232252_8253	16	0	3
2 veiksmas	5	_17_0_1_a8602c2_1327140900040_232252_8253	_17_0_1_a8602c2_1327140901762_65512_8262	16	0	3
Pabaiga	11	_17_0_1_a8602c2_1327140928335_347886_8284		16	0	1
Atoveiksmis	9	_17_0_1_a8602c2_1327140918865_233252_8274	_17_0_1_a8602c2_1327140933438_540215_8293	16	0	3
	10	_17_0_1_a8602c2_1327140891447_725617_8221	_17_0_1_a8602c2_1327140879760_917827_8141	16	0	5
	4	_17_0_1_a8602c2_1327140881967_716806_8156	_17_0_1_a8602c2_1327140885535_175948_8182	16	0	5
Taip	8	_17_0_1_a8602c2_1327140889397_508266_8204	_17_0_1_a8602c2_1327140891447_725617_8221	16	0	5
	6	_17_0_1_a8602c2_1327140885535_175948_8182	_17_0_1_a8602c2_1327140889397_508266_8204	16	0	5
Ne	8	_17_0_1_a8602c2_1327140889397_508266_8204	_17_0_1_a8602c2_1327140879760_917827_8141	16	0	5
Pradžia	1		_17_0_1_a8602c2_1327140898557_207367_8244	16	0	2
	4	_17_0_1_a8602c2_1327141872670_265968_8357	_17_0_1_a8602c2_1327141876357_30035_8377	20	0	5
Pradžia	1		_17_0_1_a8602c2_1327141880163_854765_8387	20	0	2
	2	_17_0_1_a8602c2_1327141871017_246279_8342	_17_0_1_a8602c2_1327141872670_265968_8357	20	0	5
Veiksmas	3	_17_0_1_a8602c2_1327141880163_854765_8387	_17_0_1_a8602c2_1327141881838_428460_8396	20	0	3
Pabaiga	5	_17_0_1_a8602c2_1327141881838_428460_8396		20	0	1
	0	_17_0_1_a8602c2_1327560067919_525125_7892	_17_0_1_a8602c2_1327560050008_326484_7797	23	0	5
Taip	0	_17_0_1_a8602c2_1327560077071_120296_7963	_17_0_1_a8602c2_1327560050008_326484_7797	23	0	5
Ne	0	_17_0_1_a8602c2_1327560077071_120296_7963	_17_0_1_a8602c2_1327560067349_225611_7885	23	0	5

66 pav. Veiklos proceso „su viena veikla“ identifikuotų BPMN elementų informacija duomenų bazėje

Kitas metodo „Transformuoti veiklos proceso modelį į veiklos paslaugas“ žingsnio sudedamasis elementas yra identifikuoti elementų grupes panaudotas modelyje. Pirmojoje veiklos proceso diagramoje „Su keletu veiklų“ identifikuojama viena elementų grupė, kurią sudaro – BPMN elementai: „Activity“ pavadinimu „Ar taip?“ su atšakomis „Taip“ ir „Ne“. Vykdomas šios elementų struktūros transformavimas į veiklos paslaugas, ir po transformacijos gautas vykdomojo kodo fragmentas (apibrauktas) pateiktas 67 pav.

```

</sequence>
</complexType>
</schema>
</types>

<binding>
  <operation name='ar_taip'>
    <input>
      <element name='' />
    </input>
    <output>
      <element name='Taip' />
      <element name='Ne' />
    </output>
  </operation>
</binding>
</description>

```

67 pav. Veiklos proceso „Su keletu veiklų“ identifikuotos elementų grupės programinio kodo fragmentas

Galiausiai perrenkami elementai, nustatoma kurie dar nėra transformuoti į veiklos paslaugas. Atliekamas jų transformavimas į veiklos paslaugas. Šio etapo metu taip pat tikrinama ar egzistuoja prisegta panaudotajam BPMN elementui veiklos taisyklė, jeigu taip ji taip pat transformuojama į veiklos paslaugas.

Pačioje pabaigoje vartotojui pateikiamas galutinis transformacijos metu gautas rezultatas (68 pav. ir 69 pav.). Rezultatas taip pat išsaugomas XML dokumente (XML failo pavyzdys pateiktas – 5 priede).

```

<?xml version='1.0' encoding='UTF-8'?>
<description Name='VP projektas'>

  <documentation>
    VP projektas.wsdl
    Autorius: Edminas Vrubliauskas
    Projektas: VP projektas
    BPMN diagrama: veiklos procesas
    BPMN diagramos aprašas: veiklos procesas
  </documentation>

  <types>
    <schema Name='veiklos procesas'>
      <complexType Name='' Id='' Type=''>
        <sequence>
          <element Name='1 veiksmas' Type='Activity' />
          <element Name='2 veiksmas' Type='Activity' />
          <element Name='Atoveiksmis' Type='Activity' />
        </sequence>
      </complexType>
    </schema>
  </types>

  <binding>
    <operation name='ar_taip'>
      <input>
        <element name='' />
      </input>
      <output>
        <element name='Taip' />
        <element name='Ne' />
      </output>
    </operation>
  </binding>
</description>

```

68 pav. Veiklos proceso „su keletu veiklų“ rezultato pateikimas vartotojui

```

<?xml version='1.0' encoding='UTF-8'?>
<description Name='VP projektas'>

  <documentation>
    VP projektas.wsdl
    Autorius: Edminas Vrubliauskas
    Projektas: VP projektas
    BPMN diagrama: Elementarus
    BPMN diagramos aprašas: elementarus srautas
  </documentation>

  <types>
    <schema Name='Elementarus'>
      <complexType Name='' Id='' Type=''>
        <sequence>
          <element Name='Veiksmas' Type='Activity' />
        </sequence>
      </complexType>
    </schema>
  </types>
</description>

```

69 pav. Veiklos proceso „su viena veikla“ rezultato pateikimas vartotojui

Veiklos proceso modelio, praplėsto veiklos taisyklėmis, transformavimo į veiklos paslaugas metodo vertinimo rezultatai apibendrinti ir pateikti 28 lentelėje. Veiklos procesas „veikla su keletu veiklų“ transformavimo procesas aprašomas 28 lentelėje, identifikuojant teigiamus ir neigiamus metodo prototipo aspektus. Veiklos proceso „su viena veikla“

transformavimo eiga persidengia su anksčiau minėto proceso transformavimo, todėl jis lentelėje nedetalizuojamas.

28 lentelė. Transformavimo metodo vertinimo lentelė (nagrinėjamas veiklos procesas – „su keletu veiklų“)

Nr.	Kriterijus	Įvertis	Įgyvendinimas					
1.	Naujo projekto sukūrimas.	Naujas projekto sukūrimas nesudėtingas ir užima nedaug laiko, taip padedama lengviau susiklasifikuoti disponuojamus veiklos proceso modelius.	Projektas sėkmingai sukuriamas įvedus taisyklingą jo pavadinimą.					
2.	Naujos diagramos patalpinimas.	Naujos diagramos patalpinimas galimas tik po to kai pasirenkamas konkretus projektas. Talpinimo metu surenkama informacija – diagramos pavadinimas ir jos aprašas. Diagramos dokumentas konkretus – XML failas.	Diagrama sėkmingai patalpinama įvedus taisyklingą pavadinimą ir jos aprašą bei jeigu tenkinama dokumento tipo sąlyga.					
3.	Diagramos nuskaitymas.	Diagramos nuskaitymas vyksta autonominiu būdu, inicijavus transformavimo pradžią. Atsidaromas XML failas ir nagrinėjamas jame esantis turinys, identifikuoto elemento informacija patalpinama duomenų bazėje.	Informacija apie diagramos elementus sėkmingai patalpinama duomenų bazėje.					
4.	Nuskaitytų, diagramoje naudojamų elementų, sudėjimas į seką.	Diagramoje panaudoti elementai surykiuojami eilės tvarka į seką, vykdomi pakeitima duomenų bazėje.	Elementai sėkmingai identifikuoti ir pakeičiama informacija duomenų bazėje – nurodomas elementų eiliškumas.					
5.	Identifikuoti pavieniai elementai.	<table border="1"> <tr> <td>„Event“</td> <td rowspan="4">Nustatoma identifikuotųjų elementų tipai pagal BPMN notaciją, vykdomi pakeitimai duomenų bazėje.</td> </tr> <tr> <td>„Activity“</td> </tr> <tr> <td>„Gateway“</td> </tr> <tr> <td>„Transmission“</td> </tr> </table>	„Event“	Nustatoma identifikuotųjų elementų tipai pagal BPMN notaciją, vykdomi pakeitimai duomenų bazėje.	„Activity“	„Gateway“	„Transmission“	Sėkmingai nustatyta elemento tipas ir pakeičiama informacija duomenų bazėje – nurodomas elemento tipas.
„Event“	Nustatoma identifikuotųjų elementų tipai pagal BPMN notaciją, vykdomi pakeitimai duomenų bazėje.							
„Activity“								
„Gateway“								
„Transmission“								
6.	Identifikuotos elementų struktūros.	Perrenkami elementai ir tikrinama ar egzistuoja sudėtinis elementų junginys, jeigu egzistuoja jis transformuojamas į veiklos paslaugas, o informacija patalpinama duomenų bazėje.	Sėkminga identifikuota elementų struktūra ir transformuota į veiklos paslaugas, duomenys įrašomi į duomenų bazę.					

<i>Nr.</i>	<i>Kriterijus</i>	<i>Įvertis</i>	<i>Įgyvendinimas</i>
7.	Identifikuotos veiklos taisyklės.	Perrankami elementai tikrinan ar nėra jam priskirta veiklos taisyklė, jeigu priskirta, ji nuskaitoma ir transformuojama į veiklos paslaugas.	Sėkmingai identifikuota veiklos taisyklė ir transformuota į veiklos paslaugas, duomenys įrašomi į duomenų bazę.
8.	Transformavimo sparta.	Transformavimas atliekamas sąlyginai greitai.	Diagramos transformavimas užtruko keliasdešimt sekundžių.
9.	Rezultatų išvedimas.	Atlikus transformavimą, pateikiami gauti rezultatai.	Vartotojui sistemos lange pateikiamas gautas transformavimo metu sudarytas programinis kodas.
10.	Sistemos funkcionalumo paprastumas.	Sistemos funkcionalumas yra nesudėtingas vartotoju sąlyginai nesudėtinga susiorientuoti sistemoje ir sėkmingai joje dirbti.	Vartotojas sėkmingai darbuojasi su transformavimo sistema.
11.	Veiklos proceso modelio padengimas transformavimo metu.	Veiklos proceso modelio padengimas – tai procentinė išraiška kiek panaudotų veiklos proceso modelyje elementų yra sėkmingia transformuota į veiklos paslaugas.	Sėkmingai transformuota 75% diagramoje panaudotų elementų.

### 5.3. Sistemos veikimo ir savybių analizė, kokybės kriterijų įvertinimas

Veiklos proceso modelio transformavimo į veiklos paslaugas metodo sudaryto prototipo išpildymo efektyvumas ir paties metodo naudingumas vertinamas pagal šiuos kriterijus:

1. Sistemos funkcionalumo paprastumas – ar eiliam vartotojui lengva naudotis vartotojo sąsaja ir sistemos teikiamomis galimybėmis (funkcijomis);
2. Pilnumas – ar viskas yra atliekama, ir ar atliekama teisingai;
3. Darbo našumas – kaip greitai atliekama transformacija iš veiklos proceso modelio (BPMN diagramos) į veiklos paslaugas (WSDL modulius).

Kiekybiniai sudarytojo metodo vertinimo kriterijai:

1. Transformacijos metu naudojamų BPMN elementų kiekis;
2. Sugeneruojamo kodo kiekis.

Esminis rezultato kokybės kriterijus yra kuo tiksliau perteikti veiklos proceso modeliui kontekstą – veiklos paslaugų pavidalu. tai atlikti aprėpiant kuo daugiau veiklos proceso modelyje naudojamų elementų ir/ar jų kombinacijų.

Realizuotas veiklos proceso modelio transformavimo į veiklos paslaugas metodo prototipas atitinka reikalavimų specifikacijoje identifikuotus kriterijus. Prototipas yra pritaikytas darbui su *BPMN* modeliavimo standarte apibrėžtu veiklos proceso modeliu perteiktu *XML* formatu bei *WSDL* standartu.

Svarbiausia sukurto prototipo savybė yra leisti sistemos vartotojui atlikti transformaciją iš veiklos proceso modelio, sudaryto pagal *BPMN* standartą atitinkančius kriterijus, į veiklos paslaugas (*WSDL* standartą). Atlikus transformaciją, gautą rezultatą išsaugoti *XML* formatu duomenų faile. Šis failas nėra galutinis (validuotas) veiklos paslaugų dokumentas, o tik vartotojui darbo palengvinimui skirtas, pradinio programinio kodo sugeneravimo įrankis. Jo tolimesniam panaudojimui reikalinga atlikti detalesni programavimo darbai.

Naudojantis prototipu vartotojas gali susikurti vieną ar keletą projektų, jam(-iems) priskirti vieną ar daugiau *BPMN* veiklos proceso modelių *XML* formatu. Vykdamas transformavimą į veiklos paslaugas nuskaitomas veiklos proceso modelis nustatant jų struktūrinį eiliškumą, kurį vėliau vartotojas gali peržiūrėti ir įsitikinti ar visi panaudoti modelyje *BPMN* notacijos elementai yra identifikuoti. Atlikus transformavimą vartotojui pateikiama galutinis transformavimo rezultatas.

Apibendrinant galima teigti, kad realizuotas prototipas yra gera pagalbiniė priemonė vartotojui, leidžianti sumažinti laiko sąnaudas kuriant programinį kodą (veiklos paslaugas). Atliekant veiklos proceso modelio transformavimą į veiklos paslaugas ir gautus rezultatus išsaugoti *XML* faile tolesniam darbui su programiniu kodu.



## 5. Išvados

1. Atlikus veiklos procesų modeliavimo standartų *BPMN*, *UML* ir *Workflow* lyginamąją analizę, buvo atliekamas palyginimas pagal šiuos kriterijus: veiklos diagramos tipai, elementų aibė, jų kiekis; atitikimas realaus pasaulio procesams; transformacijos susiejimas su vykdomuoju kodu ir kt. Nustatyta, kad *BPMN* standartas tinkamiau ir detaliau atvaizduoja realaus pasaulio procesus, elementų naudojimas yra intuityvus IT išsilavinimo neturintiems žmonėms. *UML* yra dedikuotas informacinių sistemų kūrimui ir neatitinka vartotojų poreikių specifikuojant procesus. Analogiškai, *Workflow* nesuteikia galimybės vartotojui efektyviai atvaizduoti sudėtingus procesus ar išskirtines situacijas.
2. Atlikus veiklos paslaugų kūrimo metodų *CORBA* ir *WSDL* lyginamąją analizę, nustatyta, kad *WSDL* užtikrina geresnę veikimo spartą bei jo specifikacija yra tinkamesnė transformacijai iš veiklos proceso modelio į veiklos paslaugas. Kriterijai, pagal kuriuos buvo atliekama lyginamoji analizė: darbas skirtinguose kompiuteriuose; skirtingų protokolų palaikymas; specifikacijos įsisavinimo laikotarpis.
3. Atlikus transformacijos iš veiklos proceso modelio į veiklos paslaugas esamų sprendimų (*Together*, *VP-UML* ir *UModel*) analizę nustatyta, kad *UModel* įrankis pranašesnis už *Together* ir *VP-UML* savo galimybe modeliuoti ir importuoti veiklos proceso modelius (suderintas su kitais modeliavimo įrankiais per *XMI* standartą) bei kodo generavimo praplėtimo galimybe (veikia ir kaip *Eclipse*, ir kaip *Visual Studio IDE* programų įskiepis).
4. Sudarytas veiklos proceso modelio, praplėsto veiklos taisyklėmis, transformavimo į veiklos paslaugas metodas bei algoritmas. Veiklos proceso modelio transformavimo metodas nagrinėja *BPMN* veiklos proceso modelį. Jam pagal algortimo žingsnius, atitinkamai identifikuojami metode naudojami elementai, nustatant jų tipą bei visų elementų eiliškumą. Nustatoma programinio kodo pagrindą (karkasą) sudarantys elementai, jie transformuojami į veiklos paslaugas. Suteikiama galimybė transformacijos vykdytojui peržvelgti gautus transformacijos rezultatus. Tai leidžia sumažinti laiko sąnaudas, nustatyti gauto programinio kodo atitikimą nagrinėtam veiklos proceso modeliui bei padidinti perteikiamo turinio pilnumą.
5. Išanalizavus potencialių vartotojų poreikius bei problemas, sudaryti sistemos modeliai (vartotojų panaudojimo atvejų diagramos, duomenų bazių modelis, sistemos loginės architektūros modelis bei sistemos navigavimo planai), kurie užtikrina tinkamą metodo veikimą bei realizaciją.

6. Pasiūlytasis metodas realizuotas internetinėje aplinkoje, taip vartotojui norint naudotis transformavimo valdymo sistema užtenka internetinės naršyklės bei prieigos prie interneto. Transformavimo valdymo sistema suprojektuota pagal sukurtą transformavimo algoritmą, realizuota naudojant *PHP*, *HTML* bei *CSS* programavimo kalbas bei *MySQL* duomenų bazių valdymo sistema.
7. Sudaryti vertinimo kriterijai (sistemos funkcionalumo paprastumas, pilnumas bei darbo našumas), leidžia užtikrinti, kad bus sumažintos laiko sąnaudos, nustatytas gauto programinio kodo atitikimas nagrinėtam veiklos proceso modeliui bei padidintas perteikiamo turinio pilnumą. Atlikus eksperimentą, nustatyta, kad realizuotasis prototipas yra efektyvi pagalbinė priemonė vartotojui, leidžianti atlikti veiklos proceso modelio transformimą į veiklos paslaugas ir gautus rezultatus išsaugoti *XML* faile tolesniam darbui su programiniu kodu.
8. Pasiūlytasis metodas pristatas 17-oje Tarptautinėje magistrantų ir doktorantų mokslinėje konferencijoje „Informacinės technologijos 2012“. Straipsnis („Veiklos proceso modelio praplėsto veiklos taisyklėmis transformavimas į veiklos paslaugas.“, 6 priedas) buvo publikuotas šios konferencijos pranešimų medžiagoje – „Informacinės technologijos 2012. XVII tarptautinė magistrantų ir doktorantų konferencija. Konferencijos pranešimų medžiaga.“.

## 6. Literatūra









1. R.Paiano, A.L.Guido and A.Pandurino. Evolution of Business Process Notation. Designing Complex Web Information Systems: Integrating Evolutionary Process Engineering. 2009. p. 1-23.
2. David S. Frankel. BPMN and OMG: The BPM Merger. MDA Journal, Business Process Trends. December 2005.
3. Martin Owen ir Jog Raj. BPMN and Business Process Management. Iš OMG ir Popkin Software [Interaktyvus]. 2003. Prieiga per internetą: [http://www.omg.org/bpmn/Documents/6AD5D16960.BPMN\\_and\\_BPM.pdf](http://www.omg.org/bpmn/Documents/6AD5D16960.BPMN_and_BPM.pdf).
4. J.vom Brocke, J.Rosemann. Handbook on Business Process Management: Strategic Alignment, Governance, People and Culture. International Handbooks on Information Systems. 2010.
5. I.Pašilskyte, L.Nemuraitė ir E.Ambrazevičius. UML 2.0 išplėtimas elektroninio verslo procesams projektuoti. Informatikos mokslai. 2006. p. 157-167.
6. Stephen A. White. Process Modeling Notations and Workflow Patterns. MDA Journal, Business Process Trends [Interaktyvus]. March 2004. Prieiga per internetą: <http://www.bptrends.com/publicationfiles/03-04%20WP%20Notations%20and%20Workflow%20Patterns%20-%20White.pdf>.
7. A.Rausch , Ch.Bartelt , Th.Ternité , M.Kuhrmann. The V-Modell XT Applied – Model-Driven and Document-Centric Development L-6 The V-Modell XT Applied – Model-Driven and Document-Centric Development. SEPADS'10 Proceedings of the 9th WSEAS international conference on Software engineering. 2006. p. 131-138.
8. M.Belaunde. Transformation Composition in QVT. First European Workshop on Composition of Model Transformations - CMT 2006. 2006. p. 39-45.
9. R.Schmidt. Web Services Based Execution of Business Rules. Proc. of the Intl. Workshop on Rule Markup Languages for Business Rules on the Semantic Web. 2002.
10. R.Chinnici, M.Gudgin, J.-J.Moreau ir S.Weerawarana. Web Services Description Language (WSDL) Version 1.2 Part 1: Core Language. W3C liability [Interaktyvus]. 2003. p. 78. Prieiga per internet: <http://www.w3.org/TR/2003/WD-wsdl12-20030611/wsdl12.pdf>.
11. Department of Information Systems, Creating UML&OCL Models from SBVR Business Vocabularies and Business Rules. VeTIS User Guide [Interaktyvus]. 2009. Prieiga per internet: <http://www.magicdraw.com/files/manuals/VeTISUserGuide.pdf>.
12. C.Ventures, A.Jaokar, N.Allott ir M.Meir-Huber. Exclusive .NET Developer's Journal "Indigo" Interview with Microsoft's Don Box. .NET developer's journal. 2004.
13. M.Gudgin, M.Hadley, N.Mendelsohn, J.-J. Moreau, H.F.Nielsen, A.Karmarkar ir Y.Lafon. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). W3C liability [Interaktyvus]. 2007. Prieiga per internet: <http://www.w3.org/TR/soap12-part1/#intro>.
14. M.Henning. The Rise and Fall of CORBA. Originally published in Queue vol. 4, no. 5. 2006. p. 29-34.
15. R.Chinnici, H.Haas, A.A.Lewis, J.-J.Moreau, D.Orchard ir S.Weerawarana. Web Services Description Language (WSDL) Version 2.0 Part 2: Adjuncts. W3C liability [Interaktyvus]. 2007. Prieiga per internet: [http://www.w3.org/TR/2007/REC-wsdl20-adjuncts-20070626/#\\_http\\_binding\\_default\\_rule\\_method](http://www.w3.org/TR/2007/REC-wsdl20-adjuncts-20070626/#_http_binding_default_rule_method).

16. D.McAmis. UML Modelling with Borland's Together for .NET. Builder. [Interaktyvus]. 2004. Prieiga per internet: <http://www.builderau.com.au/architect/dotnet/soa/UML-Modelling-with-Borland-s-Together-for-NET/0,339024710,339158522,00.htm>.
17. S.Šimonová, J.Lešák, M.Kalhous ir M.Vávra. Current approaches to data modeling. Scientific papers of the university of Pardubice, Series D.2007. p. 126-132.
18. C.E.Pereira, F.F.Lima ir J.P.Granja. Code Generation based on UML Model Transformation. Universidade de Minho. [Interaktyvus]. Prieiga per internet: <http://www3.di.uminho.pt/~prh/uce15-0809/g19.pdf>.
19. J.Recker, M.Indulska, M.Rosemann ir P.Green. Do Process Modelling Techniques Get Better? A Comparative Ontological Analysis of BPMN. In Proceedings of 16th Australasian Conference on Information Systems. 2005.
20. J.Recker. Opportunities and constrains: the current struggle with BPMN. Business Process Management Journal, 16(1). 2010. p. 181-201.
21. S.A.White. Using BPMN to Model a BPEL Process. MDA Journal, Business Process Trends [Interaktyvus]. March 2005. Prieiga per internetą: <http://www.businessprocesstrends.com/publicationfiles/03-05%20WP%20Mapping%20BPMN%20to%20BPEL-%20White.pdf>.
22. A.Raj, T.V.Prabhakar ir S.Hendryx. Transformation of SBVR Business Design to UML Models. Proceedings of 1<sup>st</sup> India Software Engineering Conference. 2008. p. 29-38.
23. V.Damjanovic. Semantic reengineering of business processes. Information Systems, Volume 35, Issue 4. June 2010. p. 496-504.
24. F.Rosenberg ir S.Dustdar, Towards a Distributed Service-Oriented Business Rules System. Proceedings of Web Services 2005, ECOWS 2005, Third IEEE European Conference. 2005.
25. V.Prezel, D.Gašević ir M.Milanović. Representational Analysis of Business Process and Business Rule Languages. Iš Ontorule – project [Interaktyvus]. 2009 [žiūrėta 2010-10-11]. Prieiga per internetą: [http://ontorule-project.eu/attachments/075\\_buro2010\\_paper\\_2.pdf](http://ontorule-project.eu/attachments/075_buro2010_paper_2.pdf).

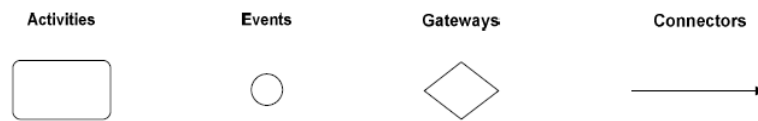
## 7. Priedai

### 7.1. 1 priedas. BPMN elementų specifikacija

Pagrindiniai BPMN elementai (pateikiami 70 pav.):

- Srauto objektai (*angl. Flow Objects*)
  - ✓ Įvykiai (*angl. Events*)
    - Pradžios įvykis ○ – naudojamas kaip triggeris procesui paskatinti, įvykio pradžios taškas;
    - Pabaigos įvykis ⊗ – nurodoma, kad įvykis užbaigtas;
    - Tarpinis įvykis ○ (*angl. Intermediate event*) – konstatuojamas tarpinis įvykis tarp įvykio pradžios ir pabaigos;
  - ✓ Veiklos (*angl. Activities*)
    - Užduotis  (*angl. Task*) – yra vienas veiklos uždavinys, kuris nėra arba negali būti detalizuojamas žemesnio lygmens elementais (tai nėra BPMN tikslas);
    - Sub-procesas  (*angl. Sub-process*) – naudojama paslėpti ar atskleisti žemesniojo lygio verslo proceso detalės;
    - Operacija  (*angl. Transaction*) – glaudžiai tarpusavyje susijusių veiksmų seka;
  - ✓ Tinklų sąsajos  (*angl. Gateways*) – išskaido arba apjungia srautus, priklausomai nuo sąlygos;
- Sujungimo objektai (*angl. Connecting Objects*)
  - ✓ Sekų srautas → (*angl. Sequence Flow*) – parodo kokia eilės tvarka bus atliekama veikla;
  - ✓ Pranešimų srautas ----> (*angl. Message Flow*) – pateikia, kokie pranešimų srautai egzistuoja organizacijos ribose (pvz. tarp plaukimo takelių);
  - ✓ Asociacija → (*angl. Association*) – naudojamas susieti artefaktą su srauto objektais;
- Plaukimo takeliai (*angl. Swimlines*)
  - ✓ Baseinas  (*angl. Pool*) – naudojama atskirti skirtingoms organizacijoms ar veiklos dalyviams;
  - ✓ Takelis  (*angl. Lane*) – naudojamas kategorizuoti veiklai pagal funkcijas ar roles;
- Artefaktai (*angl. Artefacts*)
  - ✓ Duomenų objektas  (*angl. Data Object*) – parodo, kokie duomenys yra naudojami ar sukuriami veiklos metu;
  - ✓ Grupė  (*angl. Group*) – skirta apjungti skirtingoms veikloms;

- ✓ Aiškinimas/Komentaras <sup>Annotation</sup> (*angl. Annotation*) – naudojama tam, kad bet koks asmuo, kuriam teks naudotis sukurtu modeliu lengviau suprastų pagrindinius jos elementus.



70 pav. BPMN 2.0 diagramos elementai



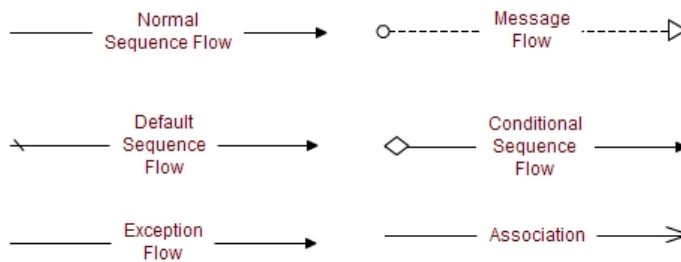
71 pav. Srauto objektai (įvykiai) naudojami BPMN 2.0



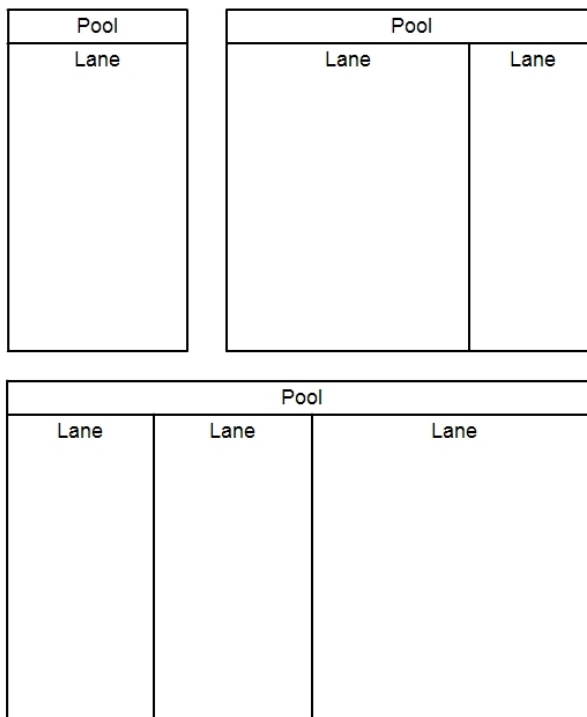
72 pav. Srauto objektai (veiklos) naudojami BPMN 2.0



73 pav. Srauto objektai (tinklų sąsajos) naudojami BPMN 2.0



74 pav. Sujungimo objektai naudojami BPMN 2.0



75 pav. Plaukimo takeliai naudojami BPMN 2.0



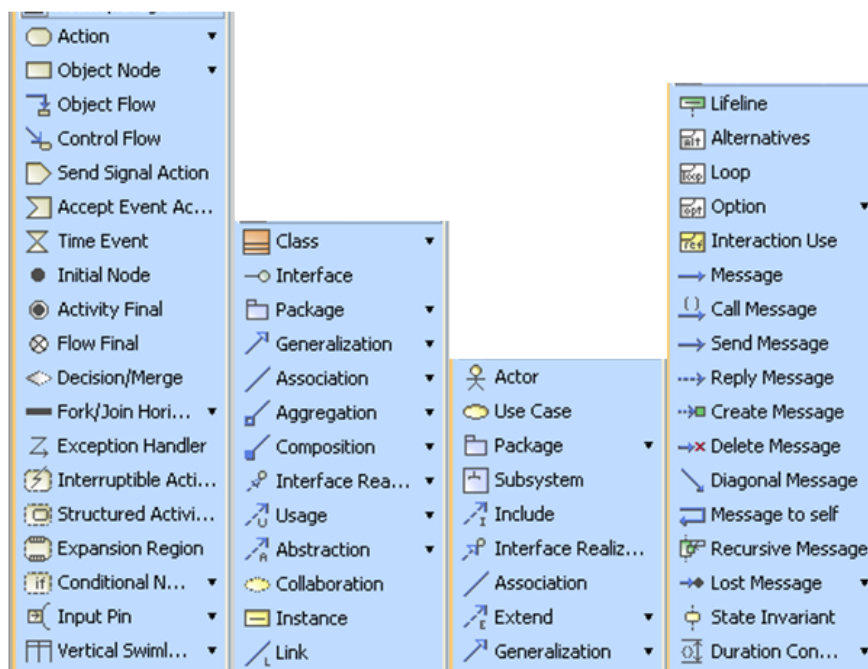
76 pav. Artefaktai naudojami BPMN 2.0

## 7.2. 2 priedas. UML elementų specifikacija

Pagrindinės *UML* diagramos skirtos veiklos proceso modeliavimui (77 pav.):

- Klasių (*angl. Class*) diagrama – aprašo sistemos objektus;
- Panaudojimo atvejų (*angl. Use Case*) diagrama – aprašo ką projektuojama sistema gali atlikti, kartu aprašydama ir išorinius sistemos veikėjus;
- Veiklos (*angl. Activity*) diagrama – naudojama veiklos procesams modeliuoti, taip pat skirtas vienam panaudojimo atvejui ar veiklos scenarijui detalizuoti;

- Sekų (*angl. Sequence*) diagrama – parodo objektų bendradarbiavimą remiantis laiko seka.






77 pav. UML diagramų elementų pavyzdžiai MagicDraw modeliavimo įrankyje (atitinkamai iš kairės pusės: Activity, Class, Use Case ir Sequence diagramos)






#### Veiklos diagramos pagrindiniai elementai

- Pradinis mazgas ● (*angl. Initial Node*) – diagramos pradžios taškas;
- Būsenos pabaigos mazgas ● (*angl. Activity Final*) – diagramos pabaigos taškas;
- Veiksmas □ (*angl. Action*) – apvalus stačiakampis reiškiantis vykdomą veiksmą;
- Valdymo srautas ↘ (*angl. Control Flow*) – parodo vykdymo eigos kryptį;
- Pasirinkimas ◊ (*angl. Decision Node*) – pasirenkama viena galimybė iš keletos variantų;
- Darbas/Veikla □ (*angl. Activity*) – diagramoje atstovauja veiklos proceso etapą arba visą veiklos procesą;
- Objektas □ (*angl. Object Node*) –
- Objekto srautas ↘ (*angl. Object Flow*) – yra veiklos srautas, aprašantis ir perteikiantis objektus ar informaciją;
- Signalų siuntimas ▷ (*angl. Send Signal Action*) – yra veiksmas, kai sukuriamas signalas veiksmo pradinėje stadijoje ir perduodamas į konkretų objektą;
- Srauto patvirtinimas ▷ (*angl. Accept Event Action*) – yra veiksmas, kuris atliekamas sulaukus atvejo, kuris nurodytas sąlygoje;







- Srauto pabaiga  (*angl. Flow Final*) – yra srauto pabaigą žymintis simbolis. Jis sunaikina visas žymes, kurios atkeliauja iki jo. Jis neturi įtakos kitiems srautams veikloje.
- Išskaidymas/apjungimas  (*angl. Fork/Join*) – tai yra kontrolės taškas, kuris srautą apjungia arba išskaido, atitinkamai pagal nagrinėjamo atvejo po būdį.
- „Plaukimo takeliai“  (*angl. Swimlines*) – naudojami atskirti veiklos dalyviams.





#### Klasių diagramos pagrindiniai elementai


- Klasė  (*angl. Class*) – apibūdina objektų, kurie turi tas pačias specifikacijas savybes;
- Abstrakcija  (*angl. Abstraction*) – yra ryšys tarp dviejų elementų arba jų rinkinių, kuriuos sudaro tos pačios sąvokos paimtos iš skirtingų lygmenų arba kylančios iš skirtingų požiūrių;
- Agregavimo ryškys  (*angl. Aggregation*) – asociacijos natūra, tai reiškia, kad vienas elementas, prie kurio yra žymė, rodo tai, kad tas elementas turi bendrąją agregaciją.
- Asociacija  (*angl. Association*) – nurodomas semantinis ryšys tarp dviejų elementų (atvejų);
- Apibendrinimas  (*angl. Generalization*) – yra ryšys, kuris parodo, kad vienas elementas paveldi kito elemento funkcijas.


#### Panaudojimo atvejų pagrindiniai elementai

- Aktorius  (*angl. Actor*) – su sistema sąveikaujantis vartotojas;
- Panaudojimo atvejis  (*angl. Use Case*) – veiksmų seka, kurią sistema gali atlikti sąveikaudama su sistemos aktoriais.
- Įtraukti  (*angl. Include*) – šis ryšys nurodo, kaip bazinio panaudos atvejo elgsena priklauso nuo įtraukiamo panaudos atvejo elgsenos;
- Praplėsti  (*angl. Extend*) – šis ryšys nurodo, kaip išplėsto panaudos atvejo elgsena priklauso nuo bazinio panaudos atvejo elgsenos.

#### Sekų diagramos pagrindiniai elementai

- Aktorius  (*angl. Actor*) – su sistema sąveikaujantis vartotojas;
- Gyvavimo tiltas  (*angl. Lifeline*) – vieno veiklos dalyvio atliekamų veiksmų eiliškumas laiko atžvilgiu;
- Pranešimas  (*angl. Message*) – komunikacinio pobūdžio ryšys tarp skirtingų veiklos dalyvių;
- Alternatyvos  (*angl. Alternatives*) – išraiškų sąveikos fragmentas, leidžiantis perteikti apibrėžtas sąveikas bei jų alternatyvą;

- Kontūras  (*angl. Loop*) – nurodo veiksmą bei apibrėžia darinio semantiką;








Atsakomasis pranešimas  (*angl. Reply Message*) – grįžtamojo pobūdžio ryšys perteikiantis vieno proceso dalyvio atsaką į atliekamas operacijas.

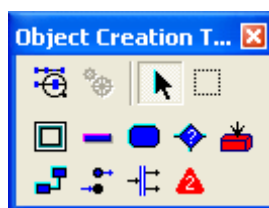
### 7.3. 3 priedas. Workflow elementų specifikacija

Pagrindinės *Workflow* [25] diagramos skirtos veiklos procesų modeliavimui:

- Veiklos (*angl. Activity*) diagrama – naudojama veiklos procesams modeliuoti, taip pat skirtas vienam panaudojimo atvejui ar veiklos scenarijui detalizuoti;
- Veiklos proceso (*angl. Business process*) diagrama

Pagrindiniai elementai naudojami *Workflow* diagramose (78 pav.):

- Darbų sekos takelis  (*angl. Workflow lane*) – naudojama išskirti veiklos dalyvius ir sugrupuoti jų atliekamas funkcijas;
- Veikla  (*angl. Activity*) – stačiakampis reiškiantis vykdomą veiklą;
- Pasirinkimo taškas  (*angl. Decision Point*) – suteikia pasirinkimo galimybę tarp vieno ar daugiau variantų;
- Darbo eiga  (*angl. Workflow*) – parodo darbo sekos srautą;
- Ištaka/įduba  (*angl. Source/Sink*) – atitinkamai veiklos proceso srauto tam tikru laiko momentu pradžia ir pabaiga;
- Susikryžiavimas  (*angl. Junction*) – galimybė apjungti ar išskaidyti veiklos proceso (darbo sekos) srautą.
- Tilto pradžia/ tilto pabaiga  (*angl. BridgeStart/BridgeEnd*) – atitinkamai veiklos proceso srauto pradžia ir pabaiga.



78 pav. *Workflow* modelio elementai (*ProVision* pakete)

### 7.4. 4 priedas. BPMN pavyzdinis failas.

```
<?xml version="1.0" encoding="utf8" standalone="yes"?>
<Package Name="Untitled1" Id="_17_0_1_a8602c2_1327140870149_755556_8103"
xmlns:ns2="http://www.wfmc.org/2002/XPDL1.0"
xmlns="http://www.wfmc.org/2009/XPDL2.2">
  <PackageHeader>
    <XPDLVersion>2.2</XPDLVersion>
    <Vendor>Nomagic</Vendor>
    <Created>12-00-22T00:14:00</Created>
    <ModificationDate>12-00-21T12:16:42</ModificationDate>
```

```

    <LayoutInfo PixelsPerMillimeter="2.8346457"/>
  </PackageHeader>
  <RedefinableHeader>
    <Author>Edminas</Author>
    <Version>-1</Version>
    <Codepage>Cp1257</Codepage>
    <Countrykey>LT</Countrykey>
  </RedefinableHeader>
  <ConformanceClass BPMNModelPortabilityConformance="SIMPLE"
GraphConformance="NON_BLOCKED"/>
  <Pages>
    <Page Width="1193.0" Height="824.0"
Id="_17_0_1_a8602c2_1327140870149_983048_8102" Name="Untitled1"/>
  </Pages>
  <Pools>
    <Pool MainPool="true" BoundaryVisible="false"
Process="_17_0_1_a8602c2_1327141606538_168905_8308" Orientation="HORIZONTAL"
Name="Untitled1" Id="_17_0_1_a8602c2_1327141606541_950839_8309">
      <NodeGraphicsInfos>
        <NodeGraphicsInfo FillColor="#FFCC33" Width="1193.0" Height="824.0"
PageId="_17_0_1_a8602c2_1327140870149_983048_8102" IsVisible="true"
ToolId="CBM">
          <Coordinates YCoordinate="5.0" XCoordinate="5.0"/>
        </NodeGraphicsInfo>
      </NodeGraphicsInfos>
    </Pool>
  </Pools>
  <WorkflowProcesses>
    <WorkflowProcess Name="_17_0_1_a8602c2_1327141606538_168905_8308_wf"
Id="_17_0_1_a8602c2_1327141606538_168905_8308">
      <ProcessHeader>
        <Created>12-00-22T00:14:00</Created>
      </ProcessHeader>
      <Activities>
        <Activity Name="Pradžia"
Id="_17_0_1_a8602c2_1327140874495_320826_8132">
          <Event>
            <StartEvent Implementation="Unspecified" Trigger="None"/>
          </Event>
          <NodeGraphicsInfos>
            <NodeGraphicsInfo FillColor="#FFCC33" Width="25.0" Height="25.0"
PageId="_17_0_1_a8602c2_1327140870149_983048_8102" IsVisible="true"
ToolId="CBM">
              <Coordinates YCoordinate="182.0" XCoordinate="93.0"/>
            </NodeGraphicsInfo>
          </NodeGraphicsInfos>
        </Activity>
        <Activity Name="Ar taip?"
Id="_17_0_1_a8602c2_1327140889397_508286_8204">
          <Route GatewayDirection="Unspecified" MarkerVisible="false"
ExclusiveType="Data" GatewayType="Exclusive"/>
          <NodeGraphicsInfos>

```

```

        <NodeGraphicsInfo FillColor="#D5D5D5" Width="40.0" Height="40.0"
PageId="_17_0_1_a8602c2_1327140870149_983048_8102" IsVisible="true"
ToolId="CBM">
        <Coordinates YCoordinate="288.0" XCoordinate="526.0"/>
        </NodeGraphicsInfo>
    </NodeGraphicsInfos>
</Activity>
    <Activity CompletionQuantity="1" StartQuantity="1" Name="1 veiksmas"
IsForCompensation="false" Id="_17_0_1_a8602c2_1327140881967_716806_8156">
    <Implementation>
        <No/>
    </Implementation>
    <NodeGraphicsInfos>
        <NodeGraphicsInfo FillColor="#FEEFA5" Width="120.0" Height="50.0"
PageId="_17_0_1_a8602c2_1327140870149_983048_8102" IsVisible="true"
ToolId="CBM">
        <Coordinates YCoordinate="169.0" XCoordinate="203.0"/>
        </NodeGraphicsInfo>
    </NodeGraphicsInfos>
</Activity>
    <Activity Name="Pabaiga"
Id="_17_0_1_a8602c2_1327140879760_917827_8141">
    <Event>
        <EndEvent Implementation="Unspecified" Result="None"/>
    </Event>
    <NodeGraphicsInfos>
        <NodeGraphicsInfo FillColor="#FFFFFF" Width="25.0" Height="25.0"
PageId="_17_0_1_a8602c2_1327140870149_983048_8102" IsVisible="true"
ToolId="CBM">
        <Coordinates YCoordinate="422.0" XCoordinate="251.0"/>
        </NodeGraphicsInfo>
    </NodeGraphicsInfos>
</Activity>
    <Activity CompletionQuantity="1" StartQuantity="1" Name="2 veiksmas"
IsForCompensation="false" Id="_17_0_1_a8602c2_1327140885535_175948_8182">
    <Implementation>
        <No/>
    </Implementation>
    <NodeGraphicsInfos>
        <NodeGraphicsInfo FillColor="#FEEFA5" Width="120.0" Height="50.0"
PageId="_17_0_1_a8602c2_1327140870149_983048_8102" IsVisible="true"
ToolId="CBM">
        <Coordinates YCoordinate="169.0" XCoordinate="476.0"/>
        </NodeGraphicsInfo>
    </NodeGraphicsInfos>
</Activity>
    <Activity CompletionQuantity="1" StartQuantity="1" Name="Atoveiksmis"
IsForCompensation="false" Id="_17_0_1_a8602c2_1327140891447_725617_8221">
    <Implementation>
        <No/>
    </Implementation>
    <NodeGraphicsInfos>

```

```

        <NodeGraphicsInfo FillColor="#FEEFA5" Width="120.0" Height="50.0"
PageId="_17_0_1_a8602c2_1327140870149_983048_8102" IsVisible="true"
ToolId="CBM">
        <Coordinates YCoordinate="283.0" XCoordinate="203.0"/>
    </NodeGraphicsInfo>
</NodeGraphicsInfos>
</Activity>
</Activities>
<Transitions>
    <Transition Name="" To="_17_0_1_a8602c2_1327140881967_716806_8156"
From="_17_0_1_a8602c2_1327140874495_320826_8132"
Id="_17_0_1_a8602c2_1327140898557_207367_8244">
        <ConnectorGraphicsInfos>
            <ConnectorGraphicsInfo
PageId="_17_0_1_a8602c2_1327140870149_983048_8102" IsVisible="true"
ToolId="CBM">
                <Coordinates YCoordinate="194.0" XCoordinate="118.0"/>
                <Coordinates YCoordinate="194.0" XCoordinate="203.0"/>
            </ConnectorGraphicsInfo>
        </ConnectorGraphicsInfos>
    </Transition>
    <Transition Name="" To="_17_0_1_a8602c2_1327140879760_917827_8141"
From="_17_0_1_a8602c2_1327140891447_725617_8221"
Id="_17_0_1_a8602c2_1327140933438_540215_8293">
        <ConnectorGraphicsInfos>
            <ConnectorGraphicsInfo
PageId="_17_0_1_a8602c2_1327140870149_983048_8102" IsVisible="true"
ToolId="CBM">
                <Coordinates YCoordinate="333.0" XCoordinate="263.0"/>
                <Coordinates YCoordinate="422.0" XCoordinate="263.0"/>
            </ConnectorGraphicsInfo>
        </ConnectorGraphicsInfos>
    </Transition>
    <Transition Name="" To="_17_0_1_a8602c2_1327140885535_175948_8182"
From="_17_0_1_a8602c2_1327140881967_716806_8156"
Id="_17_0_1_a8602c2_1327140900040_232252_8253">
        <ConnectorGraphicsInfos>
            <ConnectorGraphicsInfo
PageId="_17_0_1_a8602c2_1327140870149_983048_8102" IsVisible="true"
ToolId="CBM">
                <Coordinates YCoordinate="194.0" XCoordinate="323.0"/>
                <Coordinates YCoordinate="194.0" XCoordinate="476.0"/>
            </ConnectorGraphicsInfo>
        </ConnectorGraphicsInfos>
    </Transition>
    <Transition Name="" To="_17_0_1_a8602c2_1327140889397_508286_8204"
From="_17_0_1_a8602c2_1327140885535_175948_8182"
Id="_17_0_1_a8602c2_1327140901762_65512_8262">
        <ConnectorGraphicsInfos>
            <ConnectorGraphicsInfo
PageId="_17_0_1_a8602c2_1327140870149_983048_8102" IsVisible="true"
ToolId="CBM">
                <Coordinates YCoordinate="194.0" XCoordinate="323.0"/>
                <Coordinates YCoordinate="194.0" XCoordinate="476.0"/>
            </ConnectorGraphicsInfo>
        </ConnectorGraphicsInfos>
    </Transition>

```

```

        <Coordinates YCoordinate="219.0" XCoordinate="546.0"/>
        <Coordinates YCoordinate="288.0" XCoordinate="546.0"/>
    </ConnectorGraphicsInfo>
</ConnectorGraphicsInfos>
</Transition>
<Transition Name="Taip" To="_17_0_1_a8602c2_1327140891447_725617_8221"
From="_17_0_1_a8602c2_1327140889397_508286_8204"
Id="_17_0_1_a8602c2_1327140918865_233252_8274">
    <ConnectorGraphicsInfos>
        <ConnectorGraphicsInfo
PageId="_17_0_1_a8602c2_1327140870149_983048_8102" IsVisible="true"
ToolId="CBM">
            <Coordinates YCoordinate="308.0" XCoordinate="526.0"/>
            <Coordinates YCoordinate="308.0" XCoordinate="323.0"/>
        </ConnectorGraphicsInfo>
    </ConnectorGraphicsInfos>
</Transition>
<Transition Name="Ne" To="_17_0_1_a8602c2_1327140879760_917827_8141"
From="_17_0_1_a8602c2_1327140889397_508286_8204"
Id="_17_0_1_a8602c2_1327140928335_347886_8284">
    <ConnectorGraphicsInfos>
        <ConnectorGraphicsInfo
PageId="_17_0_1_a8602c2_1327140870149_983048_8102" IsVisible="true"
ToolId="CBM">
            <Coordinates YCoordinate="328.0" XCoordinate="546.0"/>
            <Coordinates YCoordinate="434.0" XCoordinate="546.0"/>
            <Coordinates YCoordinate="434.0" XCoordinate="276.0"/>
        </ConnectorGraphicsInfo>
    </ConnectorGraphicsInfos>
</Transition>
</Transitions>
</WorkflowProcess>
</WorkflowProcesses>
</Package>

```

### 7.5. 5 priedas. WSDL pavyzdinis failas.

```

<?xml version='1.0' encoding='UTF-8'?>
<description Name='VP projektas'>

    <documentation>
        VP projektas.wsdl
        Authorius: Edminas Vrubliauskas
        Projektas: VP projektas
        BPMN diagrama: veiklos procesas
        BPMN diagramos aprašas: veiklos procesas
    </documentation>

    <types>
        <schema Name='veiklos procesas'>
            <complexType Name="" Id="" Type="">
                <sequence>

```

```

        <element Name='1 veiksmas' Type='Activity' />
        <element Name='2 veiksmas' Type='Activity' />
        <element Name='Atoveiksmis' Type='Activity' />
    </sequence>
</complexType>
</schema>
</types>

<binding>
    <operation name='ar_taip'>
        <input>
            <element name="" />
        </input>
        <output>
            <element name='Taip' />
            <element name='Ne' />
        </output>
    </operation>
</binding>
</description>

```

## 7.6. 6 priedas. Straipsnis 17-oje Tarptautinėje magistrantų ir doktorantų mokslinėje konferencijoje „Informacinės technologijos“

### VEIKLOS PROCESO MODELIO PRAPLĖSTO VEIKLOS TAISYKLĖMIS TRANSFORMAVIMAS Į VEIKLOS PASLAUGAS

Edminas Vrubliauskas, Lina Tutkutė

*Kauno technologijos universitetas, Informacijos sistemų katedra,  
Studentų 50-309b, Kaunas, Lietuva, edminasv@gmail.com, lina.tutkute@ktu.lt*

Santrauka. Straipsnyje nagrinėjama veiklos proceso modelio transformavimas į veiklos paslaugas. Identifikuojama transformacijos automatizavimo svarba bei įtaka galutiniam rezultatui. Išanalizuota veiklos proceso modeliavimo standartai, veiklos apribojimo modeliavimo standartai bei veiklos paslaugų sudarymui naudojami standartai. Pateikta transformavimo metodo pagrindiniai žingsniai bei transformavimo informacijos saugojimui reikalinga klasių diagrama.

Raktiniai žodžiai: Veiklos procesas, veiklos paslaugas, *BPMN*, *SBVR*, *WSDL*.

#### 1 Įvadas

Šiuolaikiniame, sparčiai besivystančiame pasaulyje, vienas iš svarbiausių organizacijų resursų yra laikas. Laiko sąnaudos vykdant veiklas, nulemia organizacijos konkurencingumą ir pažeidžiamumą. Organizacijos vis daugiau naudoja informacines sistemas, nori jas greitai ir efektyviai sukurti. Svarbiausia, siekia, kad jos gerintų organizacijos veiklą ir atitiktų keliamus reikalavimus bei padėtų pasiekti užsibrėžtus tikslus. Organizacijoms yra sudarytos sąlygos pačioms valdyti ir modeliuoti savo procesus. Tokiu būdu jos gali suformuoti reikalavimus kuriamoms informacinėms sistemoms. Vis dėlto, IS kūrėjai, perimdami organizacijos sudarytus procesų modelius, neužtikrina visapusiško atitikimo. Kadangi kurdami informacinę sistemą, jie negali pasinaudoti sudarytais modeliais ir juose esančia informacija. Šį atotrūkį turėtų eliminuoti veiklos proceso modelio transformavimas į veiklos paslaugas. Rankinis transformavimas reikalauja specifinių žinių bei apčiuopiamai didelių laiko sąnaudų. Galimos priežastys, kodėl nesėkmingai vykdomas generavimas: specifinių žinių trūkumas; laiko sąnaudų ribotumas; kaina; programinio kodo klaidos; atvejų interpretuojamumo skirtumai; kiti išoriniai veiksniai. Norint sumažinti riziką, būtina automatizuoti veiklos proceso modelio, praplėsto veiklos taisyklėmis, transformavimą į veiklos paslaugas. Taip bus užtikrintas nuoseklesnis veiklos proceso perteikimas programinio kodo pavidalu, pagreitinamas transformavimas (sumažintos laiko sąnaudos) bei užtikrintas, perteikiamo proceso, transformavimo pilnumas ir padidintas išsamumo užtikrinimas.

## 2 Veiklos proceso modelio transformavimas į veiklos paslaugas

Veiklos procesas [1] yra orientuotas koordinuoti užduotis ir atliekamas veiklas, įtraukiant tiek verslo bei IT srities specialistus bei technines priemones, kad įgyvendintų konkretų organizacijos ar sisteminio vieneto tikslą. Šiuo metu labiausiai paplitę ir vystomi veiklos procesų modeliavimo metodai [2]: *Business Process Modeling Notation (BPMN)*; *Unified Modeling Language (UML)*; Darbų sekų modelis (*angl. Workflow*). Veiklos apribojimams modeliuoti yra naudojami keli standartai *SBVR*, *PPR*, *RuleML* ir pan. Veiklos paslaugoms sudaryti naudojami *The Web Services Description Language (WSDL)*, *The Common Object Request Broker Architecture (CORBA)* ir pan.

*BPMN* [3, 4, 5] yra naujas standartas, skirtas modeliuoti veiklos procesus ir veiklos paslaugų procesus. *BPMN* modeliavimo standartas, *Model Driven Architecture (MDA)* koncepcijoje, priklauso *Computation Independent Model (CIM)* lygmeniui. Pradinis *BPMN* tikslas – pateikti visiems veiklos dalyviams lengvai suprantamą aiškinamąją notaciją. Tai apima veiklos analitikus, kurie perteikia pirminius projekto reikalavimus ir siekiamą funkcionalumą vykdomojo kodo kūrėjams (programuotojams). *BPMN* standartas pranašesnis už kitus turimų elementų (kurie skirti veiklos procesams modeliuoti) gausa bei yra orientuotas į IT išsilavinimo neturinčius organizacijų atstovus, kurie patys gali sudaryti veiklos procesų modelius ir juos palaikyti.

*Semantics of Business Vocabulary and Business Rules (SBVR)* [6, 7, 8] yra *Object Management Group (OMG)* priimtas standartas. Šis standartas skirtas formaliai ir kartu išsamiai, pasinaudojus natūralios kalbos išraiškoms, aprašyti veiklos apribojimus. *SBVR* yra natūralios kalbos veiklos taisyklių modeliavimas. *MDA* kontekste *SBVR* naudojamas *Platform Independent Model (PIM)* lygmenyje pavaizduoti veiklos taisykles. *SBVR* formuojamos remiantis kalbotyra ir formalia logika. Jos išreiškiamos kalbos loginėmis struktūromis, vadinamosiomis semantinėmis formulėmis. *SBVR* yra skirtas išreikšti veiklos žodyną ir veiklos taisykles bei nurodyti verslo informacinių sistemų reikalavimus natūralia kalba. Veiklos taisyklė yra veiklos sąlygų išraiška.

Veiklos paslaugos [9, 10, 11] (*angl. Web Services*) paprastai yra taikomojo programavimo sąsaja (*API*) arba *WebAPI*, kuri yra prieinama per *Hypertext Transfer Protocol (HTTP)* protokolą ir atlieka sistemos paskirtą užduotį. Veiklos paslaugos paprastai skirstomos į: dideles (*Big Web Services*) veiklos paslaugas ir nedideles (*RESTful Web Services*) veiklos paslaugas, atliekamų užduočių ir funkcionalumo apimtį prasme.

Veiklos proceso modelio transformavimą į veiklos paslaugas galima traktuoti, kaip modelio perteikimą, kai atliekant tam tikrą standartizuotų veiksmų seką, sudaromas veiklos procesą atitinkantis sisteminis programinis kodas. Tiesioginis veiklos proceso modelio transformavimas į veiklos paslaugas, užtikrina nuoseklesnę veiklos proceso perteikimą programinio kodo pavidalu. Metodas reikalingas transformuojamo veiklos proceso perteikimo pilnumui ir informacijos išsamumo užtikrinimui padidinti, kas leistų mažesnėmis laiko ir resursų sąnaudomis gauti išsamesnę ir patikimesnę (atitinkantį realią situaciją) programinį kodą.

## 3 Veiklos proceso modelio transformavimo į veiklos paslaugas metodas

Veiklos proceso modelio, praplėsto veiklos taisyklėmis, transformavimo į veiklos paslaugas algoritmas skirtas padėti vartotojui automatizuotu būdu gauti vykdomąjį kodą.

Algoritmo pagrindiniai žingsniai:

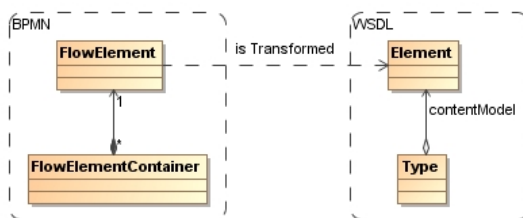
- I. „1. Paruošti veiklos proceso modelį transformavimui“ – šis žingsnis atliekamas vartotojo pusėje, sumodeliuojant veiklos procesą, pasirenkant transformavimo nagrinėjimo detalumą ir inicijuojant transformavimą į veiklos paslaugas;
- II. „2. Transformuoti veiklos proceso modelį į veiklos paslaugas“ – jo metu pažingsniui atliekama transformavimo procedūra (šis žingsnis vykdomas automatiškai);
- III. „3. Patvirtinti veiklos paslaugas“ – šis žingsnis reikalingas tam, kad patvirtinti vykdomąjį kodą ir užbaigti transformaciją.

### 3.1 Veiklos proceso modelio transformavimo į veiklos paslaugas metodo aprašas

Veiklos proceso modelio, praplėsto veiklos taisyklėmis, transformavimo į veiklos paslaugas algoritmas skirtas padėti vartotojui automatizuotu būdu gauti vykdomąjį kodą (iš veiklos proceso modelio). Metode naudojamas veiklos procesų modeliavimo standartas *BPMN*, praplėstas *SBVR* veiklos taisyklėmis ir veiklos paslaugų standartas *WSDL*.

Principinė schema su pagrindiniais *BPMN* („*FlowElement*“ ir „*FlowElementContainer*“) bei *WSDL* („*Element*“ ir „*Type*“) elementais, atspindinčiais transformacijos principą tarp *BPMN* ir *WSDL*, pateikta 18 pav.





79 pav. Principinė transformavimo metodo metamodelio diagrama

### 3.2 Veiklos proceso modelio transformavimo į veiklos paslaugas metodo aprašymas

Apibendrintas veiklos proceso modelio, praplėsto veiklos taisyklėmis, transformacijos į veiklos paslaugas algoritmas pateiktas 19 pav.

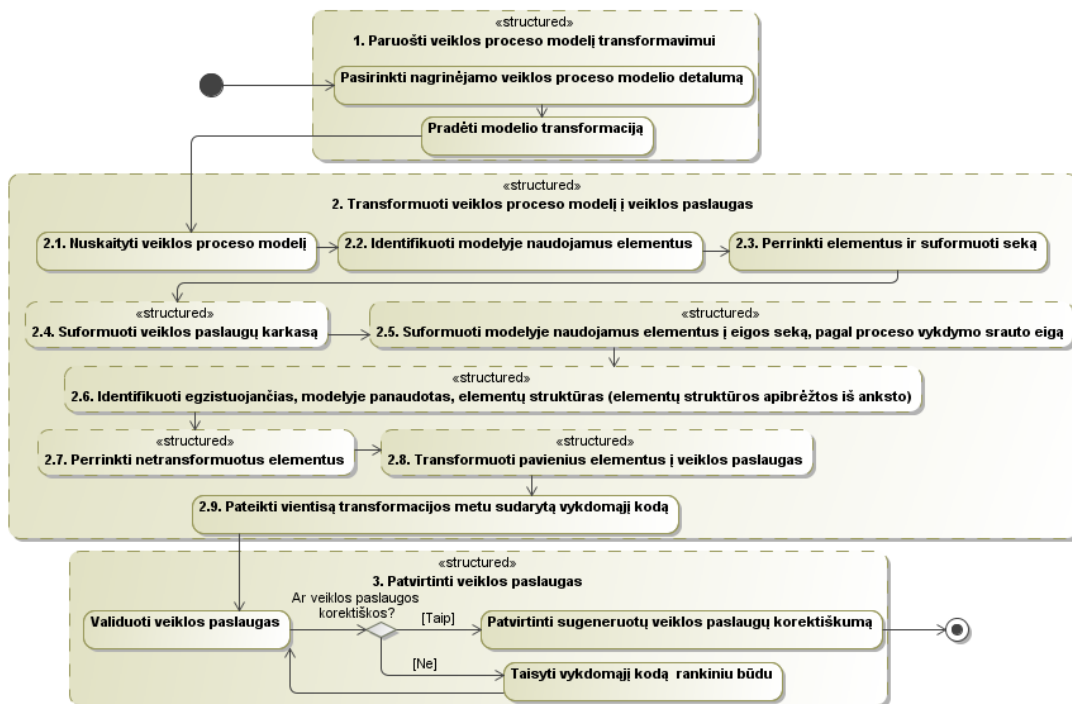
Pirmasis transformacijos algoritmo žingsnis – „1. Paruošti veiklos proceso modelį transformavimui“. Šis žingsnis susideda iš dviejų etapų: „Pasirinkti nagrinėjamo veiklos proceso modelio detalumą“ ir „Pradėti modelio transformaciją“. Pirmosios veiklos tikslas yra leisti vartotojui nurodyti, kokio detalumo lygmeniu nagrinėti norimą transformuoti veiklos proceso modelį. Galimi pasirinkimo variantai: paviršutiniškas (elementai nėra struktūriniai) ir detalus (nagrinėjami elementai gali būti struktūriniai ir/arba sudėtiniai).

Antrasis transformacijos žingsnis – „2. Transformuoti veiklos proceso modelį į veiklos paslaugas“ yra esminis algoritmo žingsnis, kurio metu yra atliekama pagrindiniai transformavimo veiksmai. Šio žingsnio elementai yra dviejų tipų – struktūriniai ir nestruktūriniai (vieno žingsnio sistemos atliekami veiksmai).

2.1-2.3 žingsniai yra nestruktūriniai, antrajame transformacijos žingsnyje, sistemos atliekami veiksmai: „Nuskaityti veiklos proceso modelį“ – šio veiksmo tikslas, programos viduje nuskaityti vartotojo pateiktą veiklos proceso modelį; „Identifikuoti modelyje naudojamus elementus“ – identifikuojami visi pateiktajame modelyje naudojami elementai; „Perrinkti elementus ir suformuoti seką“ – identifikuojamus naudojamus elementus, sudaroma elementų seka (ši seka nėra vientisa, srautas gali išsiskirti ir/arba apsijungti);

Sistema, atlikdama 2.4 „Suformuoti veiklos paslaugų karkasą“ punktą, identifikuoja *BPMN* notacijoje naudojamus *Activity* elementus („*Sub-Process*“ ir „*Task*“), juos transformuoja atitinkamai pagal struktūrą į veiklos paslaugas (klases ir metodus) suformuodami karkasą (pvz.: *BPMN* „*Task*“ elementas transformuojamas į *WSDL* „*<element>*“ struktūrą, o *BPMN* „*Sub-Process*“ elementas – į *WSDL* „*<service>*“).

2.5 žingsnio „Suformuoti modelyje naudojamus elementus į eigos seką, pagal proceso vykdymo srauto eigą“ metu suformuojama modelyje naudojamų elementų eigos seka (seka nėra vientisa, srautai gali išsiskirti ir/arba susijungti). Pirmiausia identifikuojami pradžios ir pabaigos elementai. Vėliau (pradedant nuo pradžios elemento), tikrinama ar elementas turi sąsają su išore, ir jeigu turi, ar sekantis elementas nėra identifikuotasis pabaigos elementas. Tokiu būdu suformuojama elementų eigos seka.



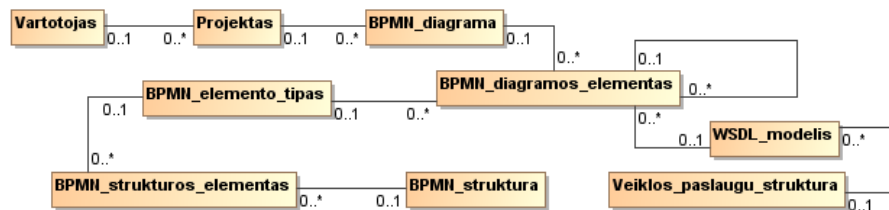
80 pav. Transformavimo metodo apibendrinta UML veiklos diagrama

2.6 žingsnio „Identifikuoti egzistuojančias, modelyje panaudotas, elementų struktūras“ metu, veiklos proceso modelyje, ieškoma elementų struktūrų, kurias būtų galima tiesiogiai transformuoti į veiklos paslaugas (pvz.: šio žingsnio metu transformuojami du sąryšiu tarpusavyje sujungti *BPMN* „*Event*“ elementai į *WSDL*

„<element>“ struktūrą). Pirmiausia suformuojama surastų elementų struktūrų seka, vėliau pasirinkus konkrečią struktūrą, identifikavus joje naudojamus elementus, tikrinama ar ši struktūra nepersidengia su jau transformuotomis elementų struktūromis. Nustačius, kad persidengimo nėra, nustatoma jos vieta kode bei atliekama transformacija.

2.7 žingsnio „Suformuoti likusių netransformuotų elementų seką“ tikslas – perrinkti suformuotą panaudotų elementų seką, juos panaikinant iš turimos veiklos proceso modelyje panaudotų elementų sekos, suformuoti naują nepanaudotų (transformuojant į veiklos paslaugas) elementų sąrašą. 2.8 žingsnio „Transformuoti likusius elementus į veiklos paslaugas“ metu, identifikuojami ir transformuojami likusieji – dar netransformuoti elementai. Šiame žingsnyje taip pat atliekama veiklos taisyklių (VT) transformacija. 2.9 žingsnio „Pateikti vientisą transformacijos metu sudarytą vykdomąjį kodą“ tikslas yra vartotojui perteikti vykdomąjį kodą jam suprantamu pavidalu.

Paskutinis transformacijos žingsnis yra „3. Patvirtinti veiklos paslaugas“. Šį veiksma atlieka sistemos vartotojas (programuotojas). Esminis žingsnio akcentas – gautą vykdomąjį kodą patikrinti, jeigu yra klaidų jas pašalinti. Jei klaidų nėra aptinkama, arba jos ištaisomos, transformacijos metu gautas vykdomasis kodas yra patvirtinamas, kaip tinkamas naudoti vykdant tolimesnius projekto etapus.



81 pav. Transformavimo metodo klasių diagrama

47 pav. pateiktas klasių modelis, užtikrinantis duomenų saugojimą. Pagrindiniai klasių modelyje naudojami elementai: „BPMN\_diagrama“ klasė skirta saugoti pagrindiniai informacijai apie transformuojamą veiklos proceso (BPMN pagrindu sudarytą) diagramą; „BPMN\_diagramos\_elementas“ – diagramoje naudojamiems, BPMN elementų spektrui priklausantiems, elementams apibrėžti; „BPMN\_elemento\_tipas“ – BPMN elementų tipui įvardinti; „BPMN\_struktūra“ – iš anksto nurodytų struktūrų, identifiкуotų transformuojamame modelyje, apibrėžimui; „Veiklos\_paslaugu\_struktūra“ – transformuotoms veiklos paslaugoms apibrėžti ir saugoti. „BPMN\_diagrama“ siejasi su „BPMN\_diagramos\_elementas“ elementu, pastarojoje saugoma informacija apie transformuojamoje diagramoje naudojamus elementus. Šis elementas („BPMN\_diagramos\_elementas“) siejasi su „BPMN\_elemento\_tipas“, šio sąryšio pagalba nurodoma, kokio tipo yra indentifiкуotasis BPMN elementas. „BPMN\_diagramos\_elementas“ taip pat susisieja su „Veiklos\_paslaugu\_struktūra“ (per elementą „WSDL\_modelis“), identifiкуojant į kokį vykdomąjį kodą duotasis BPMN elementas buvo transformuotas.

#### 4 Išvados

Pagrindiniai tyrimo objektai yra veiklos proceso modelis, apribotas veiklos taisyklėmis, bei veiklos paslaugos. Efektyvus ir produktyvus veiklos procesas galimas tik tada, kai tinkamai naudojami išteklių. Kadangi technologijų atnaujinimas nebepadeda optimizuoti ir sparčiau atlikti procesų, ieškoma naujų metodų efektyviau perteikti procesą – veiklos proceso modelyje. Norint optimizuoti procesą pasitelkiamas veiklos proceso modelis, tačiau ne visuomet yra tinkamų priemonių ar metodų jį pritaikant. Veiklos paslaugos ta sritis, kurioje yra mažai išvystyta proceso optimizavimo galimybė. Problemos sprendimas: sudarytas transformavimą dalinai automatizuojantis algoritmas, kurio pagalba veiklos proceso modelį, praplėstą veiklos taisyklėmis, galima transformuoti į veiklos paslaugas. Veiklos proceso modelio transformavimo metodas nagrinėja BPMN veiklos proceso modelį (Business Process model). Jam pagal metodo žingsnius, atitinkamai identifiкуojami metode naudojami elementai, nustatant jų tipą bei visų elementų eiliškumą. Nustatoma programinio kodo pagrindą (karkasą) sudarantys elementai, jie transformuojami į veiklos paslaugas. Suteikiama galimybė transformacijos vykdytojui peržvelgti gautus transformacijos rezultatus. Tai leidžia sumažinti laiko sąnaudas, nustatyti gauto programinio kodo atitikimą nagrinėtam veiklos proceso modeliui bei padidinti perteikiamo turinio pilnumą.

#### Literatūros sąrašas

- [1] R.Paiano, A.L.Guido, A.Pandurino. Evolution of Business Process Notation. Designing Complex Web Information Systems: Integrating Evolutionary Process Engineering. 2009. p. 1-23.
- [2] G.Mentzas, C.Halaris, S.Kavadias. Modelling business processes with workflow systems: an evaluation of alternative approaches. *International Journal of Information Management*, vol. 21, 2001, p. 123-135.
- [3] Martin Owen, Jog Raj. BPMN and Business Process Management. Iš OMG ir Popkin Software [Interaktyvus]. 2003. Prieiga per internetą: [http://www.omg.org/bpmn/Documents/6AD5D16960.BPMN\\_and\\_BPM.pdf](http://www.omg.org/bpmn/Documents/6AD5D16960.BPMN_and_BPM.pdf).
- [4] J.Recker, M.Indulska, M.Rosemann, P.Green. Do Process Modelling Techniques Get Better? A Comparative Ontological Analysis of BPMN. In Proceedings of 16th Australasian Conference on Information Systems. 2005, p. 36-45.

- [5] J.Recker. Opportunities and constrains: the current struggle with BPMN. *Business Process Management Journal*, 16(1). 2010. p. 181-201.
- [6] SBVR (Semantics of Business Vocabulary and Business Rules), v1.0 OMG Available Specification. 2008.
- [7] L. Nemuraitė, T.Skersys, A.Šukys, E.Šinkevičius, L.Ablonskis. VETIS tool for editing and transforming SBVR business vocabularies and business rules into UML&OCL models. *Information Technologies' 2010 : proceedings of the 16th international conference on Information and Software Technologies*, ISSN 2029-0020. 2010, p. 377-384.
- [8] A.Raj, T.V.Prabhakar, S.Hendryx. Transformation of SBVR Business Design to UML Models. *Proceedings of 1<sup>st</sup> India Software Engineering Conference*. 2008. p. 29-38.
- [9] R.Chinnici, H.Haas, A.A.Lewis, J.-J.Moreau, D.Orchard ir S.Weerawarana. Web Services Description Language (WSDL) Version 2.0 Part 2: Adjuncts. W3C liability [Interaktyvus]. 2007. Prieiga per internet: [http://www.w3.org/TR/2007/REC-wsd20-adjuncts-20070626/#\\_http\\_binding\\_default\\_rule\\_method](http://www.w3.org/TR/2007/REC-wsd20-adjuncts-20070626/#_http_binding_default_rule_method).
- [10] R.Schmidt. Web Services Based Execution of Business Rules. *Proc. of the Intl. Workshop on Rule Markup Languages for Business Rules on the Semantic Web [Interaktyvus]*. 2002. p. 158-166. Prieiga per internet: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.3.2910&rep=rep1&type=pdf>
- [11] R.Chinnici, M.Gudgin, J.-J.Moreau ir S.Weerawarana. Web Services Description Language (WSDL) Version 1.2 Part 1: Core Language. W3C liability [Interaktyvus]. 2003. p. 78. Prieiga per internet: <http://www.w3.org/TR/2003/WD-wsd12-20030611/wsd12.pdf>.

## TRANSFORMATION FROM BUSINESS PROCESS MODEL EXTENDED WITH BUSINESS RULES TO WEB SERVICES

The Article analyzes transformation from business process model to web services. Automation of the transformation is identified the importance and its influence the final result. Analyzed business process modeling standards, limitation of activity modeling standards and web services development standards used. Presented the basic steps of transformation method and class diagram which required preserve the information of transformation.