

**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA**

Sandra Budrevičienė

# **VEIKLOS ŽINIŲ KAUPIMO POSISTEMIS**

Magistro darbas

Darbo vadovas  
doc. dr. S. Gudas

**KAUNAS, 2004**

## Santrauka

Nagrinėjama viena iš IS inžinerijos problemų – kompiuterizuojamos srities analizė ir modeliavimas, siekiant apibrėžti vartotojo informacinius poreikius. Realizuotas darbų sekų modeliais grindžiamas veiklos žinių kaupimo metodas ir posistemis. Sukurtas sistemos prototipas skirtas praplėsti CASE aplinkas veiklos žinių baze. Posistemiui tiesiogiai naudosis analitikai, kurie modeliuoja kompiuterizuojamos organizacijos veiklą, siekdami surinkti ir analizuoti vartotojo informacinius poreikius. Šis darbas yra dalis ISK katedros atliekamo mokslinio tyrimo „Žiniomis grindžiamos IS inžinerijos metodas“.

Informacijos sistemų inžinerija pradedama veiklos srities analize – sudaromi analizuojamų procesų darbų sekų modeliai. Sukurtas darbų sekų modelių transformavimo metodas, kuris patikrina surinktos informacijos pilnumą, nuoseklumą.

Tokį sukurtu metodu patikrintą ir patalpintą CASE sistemos žinių bazėje veiklos modelį vadinsime veiklos žinių struktūra. CASE sistemos saugykloje kaupiamos kompiuterizuojamos dalykinės srities žinios, surenkamos taikant modifikuotus darbų sekų modelius. Sukauptos informacijos pilnumas tikrinamas pagal sukurtus dalykinės srities žinių surinkimo ir pilnumo tikrinimo algoritmus – informacijos perkėlimo, informacijos atskyrimo, trūkio taškų šalinimo, funkcijos sudėties nustatymo ir importavimo.

Suprogramuotas veiklos žinių kaupimo posistemis. Informacijos surinkimo ir pilnumo tikrinimo algoritmai realizuoti naudojant Provision WorkBench v. 4.0.2 Enterprise Pro, Microsoft Excel 2002 ir Microsoft Visual Studio 7.0 programinius paketus. Sukurta veiklos modelio žinių tvarkybos informacinė sistema. Taip informacijos sistemos kūrimo procesą papildant veiklos žinių baze gali būti sutaupomas IS projektavimo darbo laikas, gerinama IS projekto sprendimų kokybė – veiklos modelyje sukauptos jau patikrintos žinios apie probleminę sritį.

Veiklos žinių kaupimo posistemis sėkmingai patikrintas gamybinės įmonės „Linas“ duomenimis.

## Summary

One of IS engineering problems – are analyses and simulation, in order to define users informative needs. Activity knowledge accumulation method and part of the system has been realized grounded on workflow models. System prototype has been created to extend CASE environment activity knowledge basis. The part of the system will be directly used by analysts, who will simulate the activity of computerized organization, trying to collect and analyze users information requirements. This work is a part of the scientific research “IS engineering method grounded on knowledge” carried out in the department of IS.

Information systems engineering is started by the analyses of activity area – workflow model sequences of analyzed processes are made. Workflow model sequences transforming method has been created, that checks completeness and sequence of the data collected.

Such checked by created methods and located in CASE system knowledge basis activity model will be called activity knowledge structure. In the storage of CASE system professional knowledge in accumulated and collected using modified work sequence models. The completeness of the information collected is checked using professional knowledge and completeness checking algorithms – information shift, data separation elimination of the breakpoints, function structure setting and import.

The activity knowledge accumulation has been programmed. Information data accumulation and completeness checking algorithms are realized using Provision WorkBench v. 4.0.2 Enterprise Pro, Microsoft Excel 2002 and Microsoft Visual Studio 7.0 program packages. Activity model knowledge maintenance information system. In this way adding to the informative system process activity knowledge basis we can save IS designing time, improve IS project solution quality in the activity model we have accumulated already checked knowledge about a problematic field.

Activity knowledge accumulation system has been successfully checked using the data of manufacturing enterprise „Linas“.

# Turinys

<b>1.</b>	<b>ĮVADAS</b> .....	<b>8</b>
<b>2.</b>	<b>TERMINŲ IR SANTRUMPŲ ŽODYNAS</b> .....	<b>11</b>
<b>3.</b>	<b>ANALITINĖ DALIS</b> .....	<b>12</b>
3.1.	ORGANIZACIJOS VEIKLOS MODELIAVIMAS .....	12
3.1.1.	<i>Organizacija – kompiuterizuotos informacijos sistemos funkcionavimo aplinka ir objektas</i> .....	12
3.1.2.	<i>Organizacija kaip sistema</i> .....	12
3.1.3.	<i>Pagrindiniai terminai</i> .....	13
3.1.4.	<i>Sistema informatikos požiūriu</i> .....	13
3.1.5.	<i>Organizacijų veiklos modeliavimo problema</i> .....	14
3.2.	TARPTAUTINIŲ VEIKLOS MODELIAVIMO STANDARTŲ ANALIZĖ .....	15
3.3.	ORGANIZACIJOS VEIKLOS MODELIAVIMO PRIEMONIŲ ANALIZĖ .....	17
3.4.	DARBŲ SEKŲ MODELIŲ TAIKYMAS DALYKINĖS SRITIES ŽINIOMS SPECIFIKUOTI .....	19
3.4.1.	<i>Vartotojo poreikių surinkimas ir analizavimas naudojant veiklos metamodelį</i> .....	19
3.4.2.	<i>Veiklos darbų sekų modelio sudėtis</i> .....	20
3.4.3.	<i>Veiklos darbų sekų modelio išskyrimas į procesų ir funkcijų darbų sekų modelius</i> .....	21
3.4.4.	<i>Procesų darbų sekų modelis</i> .....	22
3.4.5.	<i>Funkcijų darbų sekų modelis</i> .....	23
<b>4.</b>	<b>TYRIMO DALIS</b> .....	<b>24</b>
4.1.	ARCHITEKTŪROS SPECIFIKACIJA .....	24
4.1.1.	<i>Loginis sistemos vaizdas</i> .....	24
4.1.2.	<i>Komponentų apžvalga</i> .....	26
4.2.	DARBO EIGOS ALGORITMAS .....	28
4.3.	VEIKLOS FUNKCIJŲ IR PROCESŲ INFORMACINIŲ POREIKIŲ SURINKIMAS.....	28
4.3.1.	<i>Grafinio darbų sekų modelio (G_WFM) sudarymas</i> .....	29
4.3.2.	<i>G_WFM eksportavimas į „Microsoft Excel 2002“ paketą</i> .....	32
4.3.3.	<i>„Microsoft Excel 2002“ darbo lapų tvarkymas</i> .....	34
4.3.4.	<i>Informacijos importavimas iš „Microsoft Excel 2002“ į „Microsoft Visual FoxPro 7.0“</i> .....	37
4.4.	INFORMACIJOS PERKĖLIMAS .....	42
4.4.1.	<i>Tarpinės duomenų lentelės</i> .....	42
4.4.2.	<i>Tarpinė (VP_WF) veiklos žinių bazė</i> .....	44
4.4.3.	<i>Informacijos perkėlimo algoritmo schemas</i> .....	45
4.4.4.	<i>Informacijos perkėlimo algoritmo aprašymas</i> .....	47
4.5.	INFORMACIJOS ATSKYRIMAS .....	51
4.5.1.	<i>Procesų (P_WFM) ir funkcijų (F_WFM) darbų sekų modelių atskyrimas</i> .....	51
4.5.2.	<i>Tarpinės (P_WF ir F_WF) veiklos žinių bazės</i> .....	54
4.5.3.	<i>Informacijos atskyrimo algoritmo schemas</i> .....	57
4.5.4.	<i>Informacijos atskyrimo algoritmo aprašymas</i> .....	59

4.6.	TRŪKIO TAŠKŲ ŠALINIMAS.....	62
	4.6.1. <i>Procesų ir funkcijų darbų sekų modelių trūkio taškų šalinimas</i> .....	62
	4.6.2. <i>Trūkio taškų šalinimo algoritmo schema</i> .....	63
4.7.	INFORMACIJOS IMPORTAVIMAS .....	64
	4.7.1. <i>Informacijos importavimas</i> .....	64
	4.7.2. <i>Veiklos žinių bazė</i> .....	64
	4.7.3. <i>Informacijos importavimo algoritmo schemas</i> .....	66
	4.7.4. <i>Informacijos importavimo algoritmo aprašymas</i> .....	68
4.8.	FUNKCIJOS SUDĖTIES NUSTATYMAS .....	70
	4.8.1. <i>Funkcijos sudėties nustatymas</i> .....	70
	4.8.2. <i>Funkcijos sudėties nustatymo algoritmo schema</i> .....	73
<b>5.</b>	<b>EKSPERIMENTINĖ DALIS .....</b>	<b>74</b>
5.1.	VARTOTOJO SAŠAJOS PROJEKTAS .....	74
5.2.	VARTOTOJO DOKUMENTACIJA .....	75
	5.2.1. <i>Sistemos funkcinis aprašymas</i> .....	75
	5.2.2. <i>Informacija</i> .....	76
	5.2.3. <i>Informacijos perkėlimas</i> .....	76
	5.2.4. <i>Informacijos atskyrimas</i> .....	77
	5.2.5. <i>Informacijos importavimas</i> .....	80
	5.2.6. <i>Pagalbinės funkcijos</i> .....	80
<b>6.</b>	<b>IŠVADOS.....</b>	<b>81</b>
<b>7.</b>	<b>LITERATŪRA .....</b>	<b>82</b>
	<b>PRIEDAS .....</b>	<b>84</b>

## Paveikslų sąrašas

1 PAV. <i>INFORMACIJOS SISTEMA ORGANIZACIJOJE</i> .....	12
2 PAV. <i>CEN ENV 4003 STANDARTE NAUDOJAMA PRINCIPINĖ VEIKLOS MODELIAVIMO SCHEMA</i> .....	15
3 PAV. <i>ORGANIZACIJOS MODELIAVIMUI BŪTINI PAGRINDINIAI KONSTRUKTAI, APIBRĖŽTI CEN ENV 12204 STANDARTE</i> .....	16
4 PAV. <i>UEML PRINCIPINĖ SCHEMA</i> .....	16
5 PAV. <i>VEIKLOS TAISYKLIŲ SAUGYKLOS VIETA CASE ĮRANKYJE</i> .....	17
6 PAV. <i>IS GC VARTOTOJO POREIKIŲ SURINKIMO IR ANALIZĖS ETAPAS, TAIKANT VEIKLOS METAMODELĮ [9]</i> .....	20
7 PAV. <i>DARBŲ SEKŲ MODELIO METAMODELIS (KLASIŲ DIAGRAMA)</i> .....	21
8 PAV. <i>VEIKLOS PROCESŲ DARBŲ SEKŲ MODELIO SKAIDYMO Į FUNKCIJŲ IR PROCESŲ DARBŲ SEKŲ MODELIOUS PRINCIPAS</i> .....	22
9 PAV. <i>BENDRAS SISTEMOS KOMPONENTŲ VAIZDAS</i> .....	25
10 PAV. <i>INFORMACIJOS TVARKYMO ALGORITMAS</i> .....	28
11 PAV. <i>VEIKLOS FUNKCIJŲ IR PROCESŲ SURINKIMO ALGORITMAS</i> .....	29
12 PAV. <i>DARBŲ SEKOS MODELIO PAVYZDYS</i> .....	29
13 PAV. <i>DARBŲ SEKOS MODELIS (VP_WFM), SPECIFIKUOJANTIS PROCESĄ „SIUVIMAS“</i> .....	30
14 PAV. <i>ŽEMESNIOJO LYGIO DARBŲ SEKOS MODELIS, SPECIFIKUOJANTIS PROCESĄ „PRODUKCIJOS GAMYBA“</i> .....	30
15 PAV. <i>ŽEMESNIOJO LYGIO DARBŲ SEKOS MODELIS, SPECIFIKUOJANTIS PROCESĄ „ŽALIAVŲ UŽSAKYMAS“</i> .....	31
16 PAV. <i>ŽEMESNIOJO LYGIO DARBŲ SEKOS MODELIS, SPECIFIKUOJANTIS PROCESĄ „DRABUŽIO SIUVIMAS“</i> .....	31
17 PAV. <i>„MODEL MAKER EXPORT WIZARD –STEP 1 OF 3“</i> .....	32
18 PAV. <i>„MODEL MAKER EXPORT WIZARD –STEP 2 OF 3“</i> .....	33
19 PAV. <i>„MODEL MAKER EXPORT WIZARD –STEP 3 OF 3“</i> .....	33
20 PAV. <i>FAILO „PIRMAS.XLS“ „WORKFLOW“ DARBO LAPAS</i> .....	34
21 PAV. <i>FAILO „PIRMAS.XLS“ „ACTIVITY“ DARBO LAPAS</i> .....	36
22 PAV. <i>„WORKFLOW IMPORT“ LANGAS</i> .....	37
23 PAV. <i>„IMPORT WIZARD 1-AS“ LANGAS</i> .....	38
24 PAV. <i>„IMPORT WIZARD 2-AS“ LANGAS</i> .....	38
25 PAV. <i>„IMPORT WIZARD 3-AS“ LANGAS</i> .....	39
26 PAV. <i>„IMPORT WIZARD 4.1-AS“ LANGAS</i> .....	39
27 PAV. <i>„IMPORT WIZARD 4.2-AS“ LANGAS</i> .....	40
28 PAV. <i>„IMPORT WIZARD 5-AS“ LANGAS</i> .....	40
29 PAV. <i>„IMPORT WIZARD 6-AS“ LANGAS</i> .....	41
30 PAV. <i>„ACTIVITY IMPORT“ LANGAS</i> .....	41
31 PAV. <i>DB KŪRIMO LANGAS</i> .....	42
32 PAV. <i>LENTELĖ „PROCESŲ_SEKAIMP“</i> .....	43
33 PAV. <i>LENTELĖ „PROCESAI_IMP“</i> .....	43
34 PAV. <i>VP_WFM DUOMENŲ BAZĖ</i> .....	44
35 PAV. <i>INFORMACIJOS PERKĖLIMO ALGORITMAS</i> .....	45
36 PAV. <i>DETALUS INFORMACIJOS PERKĖLIMO ALGORITMAS</i> .....	46
37 PAV. <i>DARBŲ SEKOS MODELIS (VP_WFM), APRAŠANTIS PROCESĄ „SIUVIMAS“</i> .....	51
38 PAV. <i>PROCESŲ DARBŲ SEKŲ MODELĮ – „SIUVIMAS-P“</i> .....	53
39 PAV. <i>FUNKCIJŲ DARBŲ SEKŲ MODELĮ – „SIUVIMAS-F“</i> .....	54
40 PAV. <i>P_WFM DUOMENŲ BAZĖ</i> .....	55

41 PAV. <i>F_WFM DUOMENŲ BAZĖ</i> .....	56
42 PAV. <i>INFORMACIJOS ATSKYRIMO ALGORITMAS</i> .....	57
43 PAV. <i>DETALUS INFORMACIJOS ATSKYRIMO ALGORITMAS</i> .....	58
44 PAV. <i>PROCESŲ TRŪKIO TAŠKŲ ŠALINIMO ALGORITMAS</i> .....	63
45 PAV. <i>VEIKLOS ŽINIŲ BAZĖ</i> .....	65
46 PAV. <i>INFORMACIJOS IMPORTAVIMO ALGORITMAS</i> .....	66
47 PAV. <i>DETALUS INFORMACIJOS IMPORTAVIMO ALGORITMAS</i> .....	67
48 PAV. <i>FUNKCIJOS SUDĖTIES NUSTATYMO ALGORITMAS</i> .....	73
49 PAV. <i>VARTOTOJO SAŠAJA</i> .....	74
50 PAV. <i>SISTEMOS LANGAS „SISTEMA“</i> .....	76
51 PAV. <i>SISTEMOS LANGAS „AUTORIUS“</i> .....	76
52 PAV. <i>SISTEMOS LANGAS „INFORMACIJOS PERKĖLIMAS“</i> .....	76
53 PAV. <i>SISTEMOS LANGAS „INFORMACIJOS ATSKYRIMAS“</i> .....	77
54 PAV. <i>SISTEMOS LANGAS „INFORMACIJOS ATSKYRIMO ETAPO ATASKAITOS“</i> .....	77
55 PAV. <i>SISTEMOS LANGAS „PROCESO VYKDYTOJO ATASKAITA“</i> .....	78
56 PAV. <i>SISTEMOS LANGAS „INFORMACIJOS ATSKYRIMAS, PERKELTA INFORMACIJA“</i> .....	79
57 PAV. <i>SISTEMOS LANGAS „INFORMACIJOS ATSKYRIMAS, ATSKIRTA INFORMACIJA“</i> .....	79
58 PAV. <i>SISTEMOS LANGAS „IMPORTAVIMAS Į VM“</i> .....	80
59 PAV. <i>PRANEŠIMAS „PRADINIAI DUOMENŲ FAILAI IŠVALYTI!“</i> .....	80

## Lentelių sąrašas

1 LENTELĖ. <i>PAGRINDINIAI ORGANIZACIJOS VEIKLOS MODELIAVIMO STANDARTAI</i> .....	15
2 LENTELĖ. <i>ŠIUO METU KURIAMI SU ORGANIZACIJOS MODELIAVIMU SUSIJĘ STANDARTAI</i> .....	16
3 LENTELĖ. <i>POSISTEMĖ – VARTOTOJO INTERFEISAS</i> .....	26
4 LENTELĖ. <i>POSISTEMĖ – VYKDOMIEJI MODULIAI</i> .....	27
5. LENTELĖ. <i>POSISTEMĖ – OBJEKTAI</i> .....	27
6 LENTELĖ. <i>FUNKCIJŲ DARBŲ SEKŲ MODELyje GALIMOS VEIKLOS ĮEIGOS IR IŠEIGOS KOMBINACIJOS</i> .....	71
7 LENTELĖ. <i>GALIMOS FUNKCIJOS SUDĖTINĖS DALYS, NUSTATOMOS PAGAL VEIKLOS ĮEIGOS IR IŠEIGOS SRAUTUS</i> . .....	72



# 1. Įvadas

Darbo tikslas - realizuoti darbų sekų modeliais grindžiamą veiklos žinių kaupimo metodą ir posistemį. Sistema skirta praplėsti CASE aplinkas veiklos žinių baze. Tiesiogiai skirta analitikams, kurie modeliuoja kompiuterizuojamos organizacijos veiklą, siekdami surinkti ir analizuoti vartotojo informacinius poreikius.

Darbo metu keliami uždaviniai - patikslinti veiklos analizės metodą, kurio pagrindas darbų sekų modelių transformavimas; darbų sekų modelių realizacija (*Provision WorkBench v. 4.0.2 Enterprise Pro*); modifikuotų darbų sekų modelių sudarymas (*Microsoft Visual FoxPro 7.0*); veiklos modelio užpildymo algoritmų projektavimas (*Microsoft Visio*) ir šių algoritmų realizacija (*Microsoft Visual FoxPro 7.0*).

Siekiant parinkti veiklos modelio sudėtį ir jo sudarymo inžinerines priemones (grafinius veiklos modeliavimo metodus), darbo metu atlikta tarptautinių veiklos modeliavimo standartų analizė (*ISO 14258, ISO 15704, ISO 10314, ISO/IEC 15288, CEN ENV 40003, CEN ENV 12 204, UEML, PSL, XBRL*) ir CASE sistemose taikomų veiklos modeliavimo priemonių analizė (*Oracle Designer, Provision Workbench, Rational Rose Enterprise Edition, Object Engineering Workbench (OEW), System Architect*).

Darbo metu sudaryta darbų sekų modelių taikymo technologija dalykinės srities žinioms specifikuoti. Taikant veiklos metamodelį surenkami ir analizuojami vartotojo poreikiai, analizuojama darbų sekų modelio sudėtis. Vykdomas šio modelio skaidymas į procesų ir funkcijų darbų sekų modelius ir atliekama jų sudėties analizė. Sukurtas ir realizuotas veiklos žinių kaupimo metodas, grindžiamas darbų sekų modelių transformavimo technologija, kuri leidžia aptikti nenuoseklumus (trūkius) vartotojo pateiktoje informacijoje. Sukurta ir realizuota veiklos modelio formavimo technologija - surinktų žinių perkėlimo algoritmas, procesų ir funkcijų atskyrimo algoritmas ir veiklos modelio formavimo algoritmas. Darbo metu įrodytas šių algoritmų naudingumas, nes realizacijos metu gauti teigiami rezultatai modeliuojant konkrečią veiklos sritį. Sukurta sistema leidžia aprašyti vartotojo pateiktą informaciją apie organizacijos veiklą ir struktūrizuoti tradicinio darbų sekų modelio pagrindą.

Darbų sekų modeliai realizuoti *Provision WorkBench v. 4.0.2 Enterprise Pro* paketu. Veiklos žinių posistemio projektavimas atliktas naudojant *Microsoft Visio* paketą. Veiklos modelio užpildymo algoritmai realizuoti naudojant *Microsoft Visual FoxPro 7.0* paketą.

Sistemos ypatumas ir naujumas – sukurtas ir realizuotas metodas, kuris leidžia tikrinti pirminę vartotojo pateiktą informaciją formalizuoto veiklos modelio atžvilgiu. Sukurtas ir praktiškai patikrintas metodas, kuri grindžiamas modifikuotais darbų sekų modeliais. Dalykinės srities žinios surenkamos, nuosekliai atliekant – informacijos perkėlimą, informacijos atskyrimą, trūkių šalinimą, informacijos

importavimą ir funkcijos sudėties nustatymą surandamos loginės spragos veiklos modeliuose. Patikrintą informaciją realizuota sistema įrašo į CASE sistemos veiklos žinių bazę.

Darbo rezultatai leidžia tvirtinti, kad CASE sistemos papildytos veiklos žinių bazė, turi privalumų kuriant IS projektus - veiklos žinių modelis leidžia papildomai tikrinti vartotojo pateiktą ir analitiko surinktą informaciją apie veiklos sritį ir projektinių sprendimų kokybę.

## 2. Terminų ir santrumpų žodynas

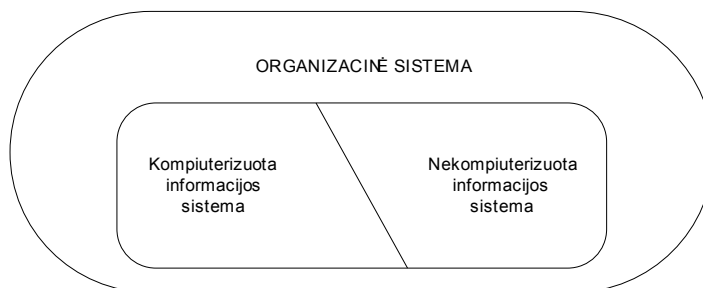
CASE	-	kompiuterizuota IS inžinerija
DB	-	duomenų bazė
DBVS	-	duomenų bazės valdymo sistema
DSD	-	duomenų srautų diagrama
F_WFM	-	funkcijų darbų sekų modelis
G_WFM	-	grafinio darbų sekų modelis
IS	-	informacinė sistema
P_WFM	-	procesų darbų sekų modelis
UEML	-	vieninga modeliavimo kalba
VP_WFM	-	veiklos procesų darbų sekų modelis
WFM	-	darbų sekų modelis

### 3. Analitinė dalis

#### 3.1. Organizacijos veiklos modeliavimas

##### 3.1.1. Organizacija – kompiuterizuotos informacijos sistemos funkcionavimo aplinka ir objektas

Organizacija – kompiuterizuotos informacijos sistemos ( IS ) funkcionavimo aplinka ir kartu kompiuterizuotas objektas. Tai iliustruoja 1 paveikslo schema [1].



1 pav. Informacijos sistema organizacijoje

Paprastai kompiuterizuojami pagrindiniai organizacijos valdymo procesai: duomenų apie gamybos procesus rinkimas, kaupimas ir apdorojimas; išteklių, reikalingų organizacijos vidinių procesų normaliai eigai užtikrinti, kiekių ir pristatymo terminų apskaičiavimas bei kiti. Šie duomenys analizuojami, formuojamos atitinkamos ataskaitos ir pateikiamos organizacijos valdymo padaliniais. Valdymo padaliniai analizuoja duomenis ir formuoja valdymo sprendimus, kurie geriausiai atitinka organizacijos keliamus tikslus. Valdymo sprendimų realizavimo kompiuterizavimas yra sudėtingas uždavinys.

Šiuolaikiniai IS kūrimo metodai realizuoti automatizuotuose IS kūrimo įrankiuose, vadinamuose CASE (*computer aided systems engineering*) priemonėmis. Šių programinių paketų funkcionavimo pagrindas yra organizacinės sistemos modelis, sudaromas ir saugomas CASE sistemos projektinių sprendimų saugykloje [2].

##### 3.1.2. Organizacija kaip sistema

Tyrimo objektas yra organizacija kaip sistema – organizacinė sistema. Organizacinėse sistemose cirkuliuoja medžiagų, energijos, informacijos srautai, jos aktyviai veikia, sąveikaudamos su aplinka. Kuo didesnė organizacija, tuo svarbesnis valdymo efektyvumo faktorius yra informacijos srautai. Organizacinės sistemos valdymo ypatumas tas, kad žmogus yra tiek valdančiosios sistemos, tiek valdomojo objekto elementas [3].

### **3.1.3. Pagrindiniai terminai**

Svarbiausiomis sistemų teorijos sąvokomis iš [4].

- ✓ Sistema – tarpusavyje susietų elementų aibė, kuri, panaudodama elementų tarpusavio ryšius, siekia įgyvendinti savo tikslus. Tikslu neturinti sistema yra tiesiog aibė susietų elementų. Sistema apibrėžiama išvardijant jos elementus, ryšius ir tikslus.
- ✓ Sistemos elementai – objektai arba įvykiai, kuriuos galima koku nors būdu aprašyti. Dažnai „elementai“ savo ruožtu yra sistemos.
- ✓ Ryšiai tarp elementų – sistemos elementai gali būti susiję įvairios prigimties ryšiais. Elementų ryšiai gali būti:
  - a) fiziniai arba technologiniai;
  - b) ryšiai laike;
  - c) loginiai;
  - d) informaciniai.
- ✓ Sistemos tikslai – tai sudėtinga problema. Neįmanoma apibūdinti sistemos tikslų, esant sistemos „viduje“. Vidiniai sistemos elementai gali tik iš dalies paaiškinti savo elgsenos apribojimo priežastis, sistemos tikslai jiems nėra iki galo aiškūs. Tik išoriniai objektai sistemos atžvilgiu gali stebėti sistemos sąveikas su aplinka ir suprasti sistemos elementų elgsenos priežastis t. y. apibūdinti sistemos tikslus. Tik organizacijos vadovas turi tiesioginį kontaktą su aplinka. Jis iš dalies yra „išorinis objektas“, todėl gali formuluoti sistemos tikslus.

### **3.1.4. Sistema informatikos požiūriu**

Informatikos požiūriu sistema yra tiriama ir modeliuojama, siekiant kompiuterizuoti jos valdymą. Informatikos požiūriu sistemą apibūdina tokios sąvokos:

- ✓ Sistema – objektų rinkinys, kur objektai turi tam tikrų savybių ir egzistuoja ryšiai tarp objektų bei tarp objektų savybių.
- ✓ Sistemos parametrai – įeiga, procesas, išeiga, grįžtamasis ryšys, apribojimai.
- ✓ Sistemos būseną nusako visuma kiekvieno parametro reikšmių konkrečiu laiko momentu.
- ✓ Savybės – objektų parametrų kokybė.
- ✓ Ryšiai – tai kas sujungia objektus ir savybes sisteminiame procese.
- ✓ Įeiga – jos funkcija yra teikti sistemai išteklius, reikalingus procesams vykdyti.

- ✓ Išeiga – tai paskirtis, kurios atžvilgiu sisteminiai objektai ir ryšiai susijungia į visumą, išeigos prasmė siejasi su tikslo supratimu.
- ✓ Procesas – paverčia įeigą išeiga. Išskirti sistemą realiame pasaulyje – vadinasi, nurodyti procesus, formuojančius konkrečią išeigą. Dirbtinėse sistemose, kurių elementus sukonstravo žmonės, išskiriami tokie procesų tipai:
  - a) pagrindinis procesas – tas, kuris įeigą paverčia išeiga;
  - b) grįžtamasis ryšys – sudėtingas procesas, kuris vyksta tokia tvarka:
    - 1) išmatuoja išeigą ir sulygina su modeline reikšme,
    - 2) nustato jų skirtumą,
    - 3) įvertina skirtumo turinį ir prasmę,
    - 4) priima sprendimą, kaip panaikinti išeigos ir modelinės reikšmės skirtumą,
    - 5) veikia pagrindinį procesą, mažindamas išeigos ir modelinės reikšmės skirtumą.
  - c) apribojimo procesą sukelia sistemos išeigos vartotojas ir aplinka. Vartotojas apriboja procesą, analizuodamas išeigą ir siekdamas, kad ši tenkintų jo tikslus.

### **3.1.5. Organizacijų veiklos modeliavimo problema**

Pagrindinis organizacijų veiklos modeliavimo tikslas yra suprasti ir dokumentuoti veiklos procesus, jų tikslus, problemas, terminologiją, veiklos dalyvius, kaip tai galima padaryti aprašant natūralia kalba. Veiklos modelis (*Enterprise model*) yra struktūrinis pagrindinių žinių, prielaidų, taikomųjų uždavinių ir duomenų aprašymas, kuris būtinas kuriant ir realizuojant kompiuterizuotas informacines sistemas.

Mūsų tyrimo objektas yra organizacinė sistema, jos veiklos procesai ir informaciniai veiklos valdymo procesai. Organizacijos tyrimo tikslas – parinkti arba sukurti tokį organizacijos dekomponavimo metodą, kuris leistų sudaryti organizacijos modelius, apimančius ir organizacijos veiklos procesus. Gautas modelis turi būti tinkamas kompiuterizuotoms informacijos sistemoms (IS) projektuoti. Nagrinėsime IS kūrimo priemones - CASE sistemas, kurių darbo pagrindas yra kompiuterizuojamos organizacijos (arba veiklos) modelis, sukaupiamas CASE sistemos projektinių sprendimų saugykloje.

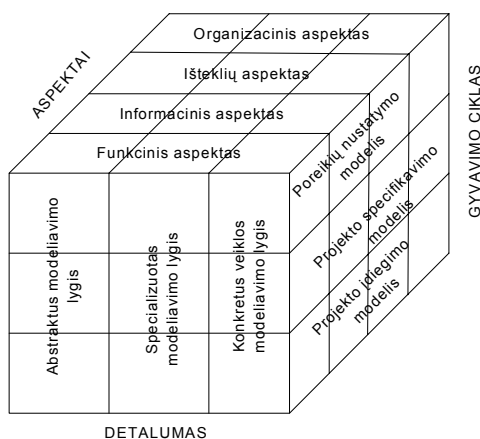
### 3.2. Tarptautinių veiklos modeliavimo standartų analizė

Šiuo metu yra keletas skirtingų organizacijos modeliavimo metodų ir kalbų: IDEF, OMT, UML, CIMOSA, ARIS ir kt. Tai sąlygojo tarpusavyje nesuderinamų programinių paketų, skirtų organizacijos veiklai modeliuoti, atsiradimą (*ARIS ToolSet, System Architect, FirstSTEP, CimTool* ir kt.). Pagrindinės organizacijos modeliavimo mokslinės grupės (ODP, OMG, PSL/NIST), didžiausi programinės įrangos gamintojai (*Oracle, Microsoft*) bei pagrindinės standartizacijos organizacijos (*ISO, CEN*) stengiasi sukurti bendrą standartą, kuriuo remiantis būtų kuriami nauji organizacijos veiklos modeliavimo metodai, kalbos bei su jais suderinta programinė įranga. Pagrindiniai organizacijos veiklos modeliavimo standartai pateikti 1 lentelėje [5].

1 lentelė. Pagrindiniai organizacijos veiklos modeliavimo standartai

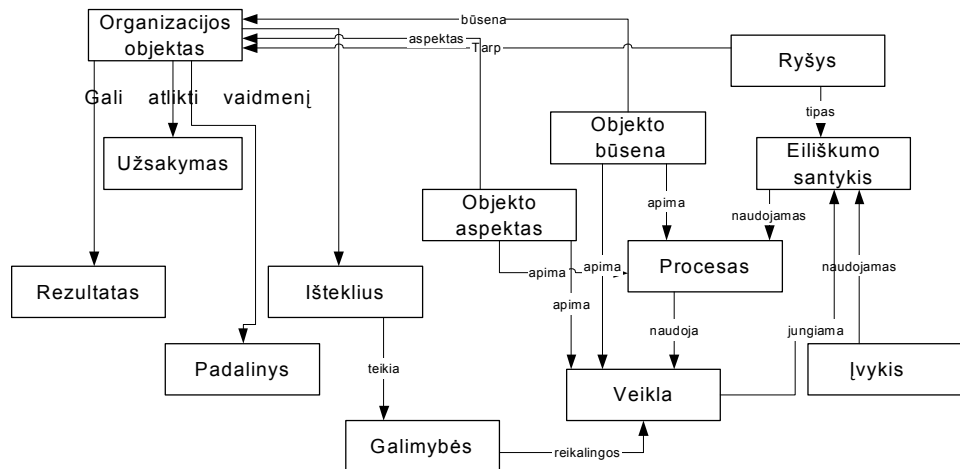
Standarto kodas	Standarto pavadinimas
ISO 14258	Concepts and Rules for Enterprise Models
ISO 15704	Requirements for Enterprise reference architectures and methodologies
ISO 10314	Shop floor production model
ISO/IEC 15288	System life cycle processes

Europos standartizacijos komitetas (CEN), bendradarbiaudamas su tarptautine standartizacijos organizacija (ISO), remdamasis 1 lentelėje pateiktais standartais sukūrė CEN ENV 40003 ir CEN ENV 12 204 standartus, kuriuose apibrėžti pagrindiniai organizacijos veiklos modeliavimo principai. CEN ENV 40003 standartas yra sukurtas CIMOSA [6] modeliavimo metodo pagrindu. CEN ENV 40003 standarte veiklos modelio projektavimo procesas pateikiamas kaip kubas, kurio ašys aprašo modeliavimo aspektus, projektavimo gyvavimo ciklo etapus bei modelio detalumo lygius. (2 pav.)



2 pav. CEN ENV 40003 standarte naudojama principinė veiklos modeliavimo schema

CEN ENV 40003 standarte veiklos modeliavimui būtinas sudėtinės dalis (konstruktus) apibrėžia CEN ENV 12204 standartas (3 pav.):

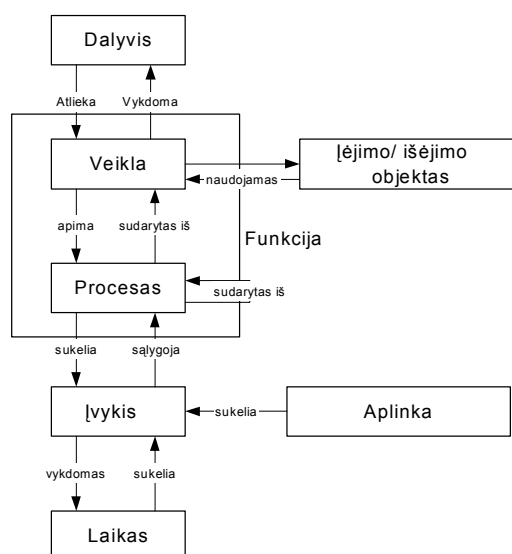


3 pav. Organizacijos modeliavimui būtini pagrindiniai konstruktai, apibrėžti CEN ENV 12204 standarte

Tačiau, kaip parodė praktika, CEN ENV 40003 ir CEN ENV 12204 standartai visiškai nepatenkino nei organizacijos veiklos projektuotojų, nei programinės įrangos gamintojų poreikių, todėl šiuo metu yra kuriamos naujos šių standartų versijos, taip pat kuriami nauji standartai (2 lentelėje) bei kalbos, pvz., UEML [7] (*Unified Enterprise Modeling Language*) (4 pav.).

2 lentelė. Šiuo metu kuriami su organizacijos modeliavimu susiję standartai

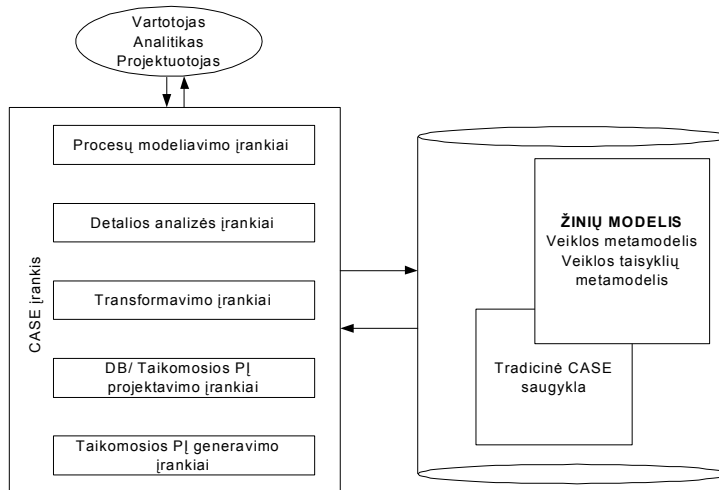
Standarto santrumpa	Standarto pavadinimas
UEML	Universal Enterprise Modeling Language
PSL	Process Specification Language
XBRL	Extensible Business Reporting Language



4 pav. UEML principinė schema



Projektuotojui, kuriančiam organizacijos informacinės sistemos projektą, būtų patogu naudotis ne tik projektavimui reikalinga informacija, esančia tradicinėje CASE įrankio saugykloje (*repository*), bet ir žinių modelio saugykla, kurioje saugomi organizacijos veikloje egzistuojančių veiklos taisyklių bei organizacijos veiklos metamodeliai. (5 pav.) [5].



5 pav. Veiklos taisyklių saugyklos vieta CASE įrankyje

Organizacijos funkcinių IS projektavimo kompiuterizuotai sistemai (CASE sistemai) teikia papildomą informaciją tokios veiklos modelio klasių sąsajos:

- ✓ Pagal funkcijos pavadinimą gali būti generuojamas susietų su šia funkcija procesų sąrašas, pateikiama kiekvieno proceso sudėtis,
- ✓ Pagal funkcijos pavadinimą arba proceso pavadinimą gali būti generuojamas susietų padalinių (vartotojų) sąrašas;
- ✓ Pagal funkcijos pavadinimą gali būti generuojama darbų sekų modelio, taip pat vartotojo taikomųjų uždavinių modelio (*use case model*) pradinė sudėtis;
- ✓ Pagal funkcijos pavadinimą gali būti generuojama esybių ryšių diagramos (arba klasių modelio) pradinė sudėtis.

Aptarėme tik dalį galimybių, kurias teikia organizacijos veiklos taisyklių bei CASE įrankio duomenų saugyklos informacinės sistemos kūrimo procese.

### 3.3. Organizacijos veiklos modeliavimo priemonių analizė

CASE (*Computer Aided Software Engineering*) – tai priemonė, kuri padeda programinės įrangos inžinieriams kurti, palaikyti, vystyti programinę įrangą. Kai kurios CASE priemonės yra skirtos tik diagramų braižymui. Kitos sutelktos į programinės įrangos realizavimą, kodo generavimą. Tačiau patogiausia yra dirbti su viena organizacijos veiklos modeliavimo priemone, kuri apimtų visą informacinės sistemos (IS) kūrimo gyvavimo ciklą.

Egzistuojančios modeliavimo priemonės:

- ✓ **Diagramų sudarymo priemonės** (*diagramming tools*). Šios priemonės skirtos tiksliai verslo diagramoms braižyti. Jos negali sukurtų diagramų praplėsti, transformuoti į kitus formatus. Diagramų sudarymo priemonės atlieka tik nedidelę BPR etapo dalį (*Visio, Micrografx Floxcharter, Corel Flow ir kt.*).
- ✓ **Procesų modeliavimo ir imitavimo priemonės** (*process modeling and simulation*). Tai priemonės su sudėtingais procesų imitatoriais. Kadangi šios priemonės yra per daug sudėtingos ir brangios verslo procesų modeliavimui, jos yra naudojamos statistikoje ir operacijų tyrinėjimuose (*BPwin, Optima, Promodel ir kt.*).
- ✓ **Tradicinės CASE priemonės** (*traditional CASE*), kurios apima informacinės sistemos modeliavimo, IS realizavimo etapus ir dalį verslo objektų modeliavimo etapo (*ADW/Cool and IEF/Composer from Sterling Software ir kt.*).
- ✓ **Duomenų bazių modeliavimo priemonės** (*database modeling*). Šios priemonės buvo kuriamos semantinių duomenų modeliavimui su griežtai pabrėžiamu duomenų bazių generavimu. Tačiau jos buvo kuriamos ne kaip organizacijos modeliavimo priemonės, todėl jas sunku perprojektuoti, kad būtų galima modeliuoti procesus (*Erwin, Power Designer, Silverrun ir kt.*).
- ✓ **Objektiškai orientuota analizė ir modeliavimas** (*OO A&D - object oriented analysis and design*). Šio tipo priemonės skirtos procesų dekompozicijai, sistemoms orientuotų taikomųjų uždavinių modeliavimui. Taip pat šiek tiek palaiko informacijos dokumentavimą, kuri reikalinga verslo modeliams kurti ir verslo sistemoms išvesti iš šių modelių (*Rational, Select, Paradigm Plus ir kt.*).

**ProVisionWorkbench** – tai organizacijos modeliavimo priemonė, kuri sujungia BPR ir OO A&D į vieną, integruotą modeliavimo priemonę. Paketas palaiko geresnes verslo modeliavimo galimybes. Palaikant reliacinių lentelių modelį (*relational model*), vartotojo sąsajos (*user interface model*) ir kitų UEML (*unified modeling language* – vieninga modeliavimo kalba) modelių kūrimą, paketas apima visą informacinės sistemos modeliavimo etapą. Dėl galimybės generuoti programinius kodus, paketas patenka ir į realizacijos etapo dalį.

Apie 78 procentų CASE priemonių yra objektiškai orientuotos, 20 procentų - struktūrinio pobūdžio (DSD pagrindu), ir likusius kelis procentus sudaro hibridinės sistemos (*System Architect (Popkin Software)*).

Labiausiai paplitę CASE paketai:

- ✓ ORACLE Designer2000;
- ✓ Provision Workbench 3.1;
- ✓ OEW - Object Engineering Workbench (Innovative Software);
- ✓ CASE 4/0 (microTOOL GmbH);
- ✓ Visible Analyst;
- ✓ System Architect (Popkin Software).

CASE paketų nuorodos internete:

- ✓ <http://www.qucis.queensu.ca/Software-Engineering/tools.html>
- ✓ <http://osiris.sunderland.ac.uk/sst/case2/tools.html>
- ✓ <http://sq.sqc.fi/~pyykkma/ctools.htm>
- ✓ <http://vanessa.diee.unica.it/TABOO/OO-CASE-tools.html>
- ✓ <http://www.pittarese.com/Auburn/cse625/case.htm>
- ✓ <http://www.proformacorp.com>

### **3.4. Darbų sekų modelių taikymas dalykinės srities žinioms specifikuoti**

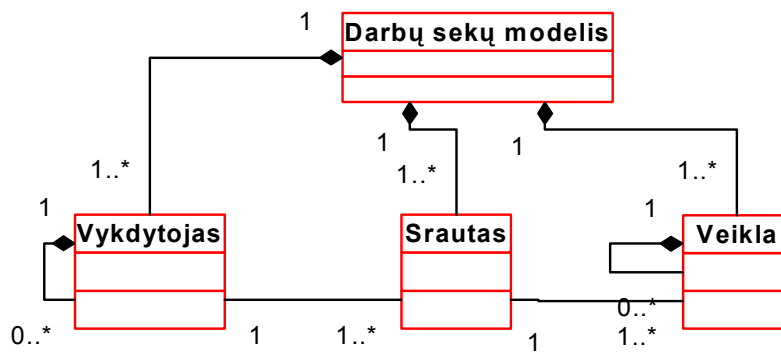
#### ***3.4.1. Vartotojo poreikių surinkimas ir analizavimas naudojant veiklos metamodelį***

IS kūrimo gyvavimo ciklo vartotojo poreikių surinkimo etapą (6 pav.) sudaro trys pagrindiniai poetaipai [8]:

- dalykinės srities žinių surinkimas, specifikuojimas ir validavimas;
- vartotojo poreikių surinkimas;
- surinktų vartotojo poreikių atitikimo dalykinės srities žinioms tikrinimas.

Probleminės srities žinių surinkimas, specifikuojimas ir validavimas vykdomas pagal probleminės srities žinių surinkimo ir pilnumo tikrinimo algoritmus. Probleminės srities žinios surenkamos taikant modifikuotus darbų sekų modelius, o jų pilnumas tikrinamas veiklos metamodelio pagrindu [9]. Surinktos žinios kaupiamos žinių bazėje, kuri sudaryta veiklos modelio ir veiklos metamodelio pagrindu [10]. Veiklos metamodelis apibrėžia veiklos modelio sudėtį.





7 pav. Darbų sekų modelio metamodelis (klasių diagrama)

Siekiant detalesnio informacijos surinkimo apie dalykinę sritį grindžiamo veiklos metamodeliu, veiklos darbų sekų modelį tikslinga modifikuoti įvedant dviejų tipų srautus (informacinius ir materialius). Veiklos procesų darbų sekų modelyje dviejų tipų srautai būtini jo skaidymui į procesų ir funkcijų darbų sekų modelius sekančiame probleminės srities žinių surinkimo ir analizės etape. Kiekvienas veiklos procesas, išskyrus pradinį ir paskutinį turi materialią ir (arba) informacinę įeigą ir išeigą.

Veiklos procesas veiklos darbų sekų modelyje apibrėžiamas kaip organizacijoje vykstančių veiksmų, transformuojančių įeigas į išeigas seka [13]. Veiklos darbų sekų modelio materialus srautas, tai materialus veiklos proceso įeiga ir (arba) išeiga, užtikrinanti veiklos proceso aprūpinimą materialiais ištekliais, būtiniais jo įvykdymui. Veiklos procesų darbų sekų modelyje materialus veiklos proceso įeiga (išeiga) nėra būtinas kiekvieno veiklos proceso elementas. Veiklos darbų sekų modelio informacinis srautas- tai informacinė veiklos proceso įeiga ir(arba) išeiga, skirta veiklos procesų valdymui. Veiklos procesų darbų sekų modelyje informacinė veiklos proceso įeiga (išeiga) nėra būtinas kiekvieno veiklos proceso elementas. Veiklos darbų sekų modelio vykdytojai (dalyviai) – asmuo, jų grupė ar organizacinis vienetas, vykdamas procesą bei atsakingas už jo sėkmingą įvykdymą.

### 3.4.3. Veiklos darbų sekų modelio išskyrimas į procesų ir funkcijų darbų sekų modelius

Procesų ir funkcijų darbų sekų modelių išskyrimas iš veiklos darbų sekų modelio yra pradinis dalykinės srities žinių pilnumo tikrinimo etapas.

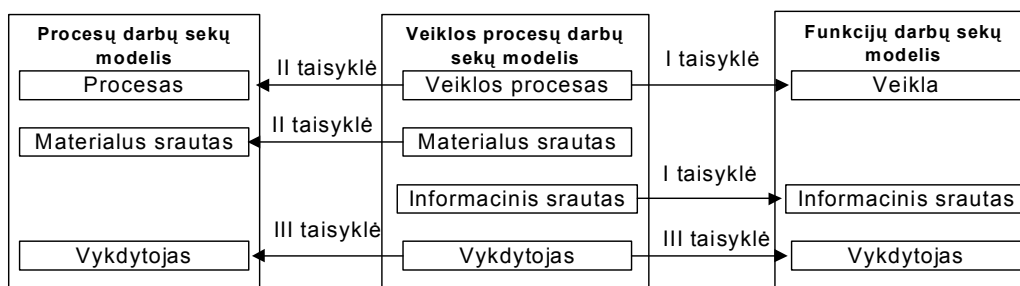
Veiklos darbų sekų modelį būtina išskirti į procesų ir funkcijų darbų sekų modelius, nes juose saugoma pradinė informacija, kurią naudoja žinių apie dalykinę sritį surinkimo ir pilnumo tikrinimo algoritmai. Šie algoritmai tikrina veiklos darbų sekų modeliu surinktų žinių pilnumą pagal veiklos metamodelio vidinę struktūrą. Veiklos metamodelis apibrėžia materialaus proceso ir jį valdančios funkcijos struktūrą ir jų tarpusavio sąveiką. Procesų darbų sekų modelyje iš veiklos darbų sekų modelio išskiriami dalykinėje srityje egzistuojantys materialūs srautai, procesai ir jų vykdytojai.

Funkcijų darbų sekų modelyje iš veiklos darbų sekų modelio išskiriami informaciniai srautai, veiklos ir vykdytojai.

Taikytinos trys pagrindinės veiklos darbų sekų modelio išskyrimo į procesų ir funkcijų darbų sekų modelius taisyklės.

- ✓ Pirma taisyklė- jei veiklos proceso įeiga ir (arba) išeiga yra informacinis srautas, šis veiklos procesas, su informaciniais įeigos ir (arba) išeigos srautais funkcijų darbų sekų modelyje atvaizduojamas kaip veikla (*activity*).
- ✓ Antra taisyklė- jei veiklos proceso įeiga ir (arba) išeiga yra materialus srautas, šis veiklos procesas su materialiais įeigos ir (arba) išeigos srautais procesų darbų sekų modelyje atvaizduojamas kaip procesas.
- ✓ Trečia taisyklė- veiklos proceso vykdytojas, funkcijų arba procesų darbų sekų modeliuose atvaizduojamas vykdančias tą veiklą ar procesą, kuris buvo išskirtas iš jo vykdomo veiklos proceso veiklos procesų darbų sekų modelyje.

Veiklos procesų darbų sekų modelio sudėtinių dalių atvaizdavimo funkcijų ir procesų darbų sekų modeliuose principas pavaizduotas 8 paveiksle [8].



8 pav. Veiklos procesų darbų sekų modelio skaidymo į funkcijų ir procesų darbų sekų modelius principas

### 3.4.4. Procesų darbų sekų modelis

Procesų darbų sekų modelyje pateikiami organizacijoje vykstantys procesai, turintys materialias įeigas ir išeigas. Šio modelio sudėtinės dalys yra procesai, materialūs srautai ir vykdytojai. Procesas- tai didžiausias veiklos vienetas, nurodantis organizacijos darbų seką nuo išorinių tiekėjų iki išorinių klientų. Įeigos keitimo į išeigą metu procesas naudoja materialius išteklius. Procesų darbų sekų modelyje įeigos ir išeigos – tik materialūs srautai. Procesai gali būti skaidomi į mažesnes dalis: subprocesus, užduotis, operacijas (sudaroma procesų hierarchija).

Materialaus srauto ir vykdytojų sąvokos apibrėžiamos analogiškai kaip ir veiklos darbų sekų modelyje. Pirmasis darbų sekų modelio procesas neturi įeigos, o paskutinis išeigos. Jeigu procesas nėra pirmas ar paskutinis, tačiau neturi įeigos arba išeigos srauto, vadinasi procesų darbų sekų modelyje egzistuoja trūkio taškas, kuris bus šalinamas probleminės srities žinių surinkimo ir pilnumo tikrinimo

algoritmu. Procesų darbų sekų modelio trūkio taškas, tai materialaus srauto, nutraukiančio nuoseklią procesų sekos vykdymą nebuvimas. Vykdamas probleminės srities žinių surinkimo ir pilnumo tikrinimo algoritmą, procesų darbų sekų modelis bus pildomas trūkstamais materialiais įeigos ir (arba) išeigos srautais (išeigos srautu nebus pildomas paskutinis procesų darbų sekų modelio elementas).

#### **3.4.5. Funkcijų darbų sekų modelis**

Funkcijų darbų sekų modelio sudėtinės dalys yra veiklos (*activity*), informaciniai srautai ir vykdytojai. Veikla- tai funkcija, arba jos sudėtinė dalis apdorojanti organizacijos informacinius srautus, keisdama jų informacinę įeigą į išeigą. Kiekvieną materialų procesą būtinai valdo bent viena funkcija. Funkcija- pilnai, o veikla, būdama funkcijos sudėtinė dalis dalinai valdo materialų procesą.

Pagal veiklos metamodelyje apibrėžtą funkcijos vidinę struktūrą, veiklos gali būti priskiriamos: Interpretavimui (*Interpretation*), Informacijos apdorojimui (*IP processing*) arba Realizavimui (*Realization*). Kiekviena funkcijų darbų sekų modelyje egzistuojanti veikla gali atitikti bet kurią iš aukščiau išvardintų funkcijos sudėtinių dalių. Probleminės srities žinių surinkimo ir pilnumo tikrinimo algoritmu nustatoma, kuriai funkcijos vidinei daliai priklauso ir kuri materialų procesą valdo funkcijų darbų sekų modelyje egzistuojančios veiklos. Funkcijos sudėties darbų sekų modelis sudarytas pagal veiklos metamodelyje apibrėžtą funkcijos sudėtį [16].

Informacinio srauto ir vykdytojų sąvokos apibrėžiamos analogiškai kaip ir veiklos darbų sekų modelyje.

Įvykdžius veiklos darbų sekų modelio išskyrimo į procesų ir funkcijų darbų sekų modelius algoritmą, funkcijų darbų sekų modelyje, kaip ir procesų darbų sekų modelyje gali atsirasti trūkio taškų. Funkcijų darbų sekų modelio trūkio taškas, tai informacinio srauto, nutraukiančio nuoseklią veiklų sekos vykdymą nebuvimas. Jeigu veikla neturi informacinės įeigos ir (arba išeigos) vadinasi funkcijų darbų sekų modelyje egzistuoja trūkio taškas (išskyrus tuos atvejus kai veikla pirmoji arba paskutinė).

## 4. Tyrimo dalis

### 4.1. Architektūros specifikacija

#### 4.1.1. Loginis sistemos vaizdas

Loginio architektūros vaizdo aprašymas apibūdina svarbiausias klases, jų pasiskirstymą į paketus ir į posistemas bei posistemų pasiskirstymą į sluoksnius. „Veiklos žinių kaupimo posistemio“ loginis vaizdas susideda iš trijų pagrindinių sluoksnių : vartotojo interfeiso, vykdomųjų modulių ir objektų. Realizuoto posistemio loginis vaizdas pateiktas 9 paveiksle.

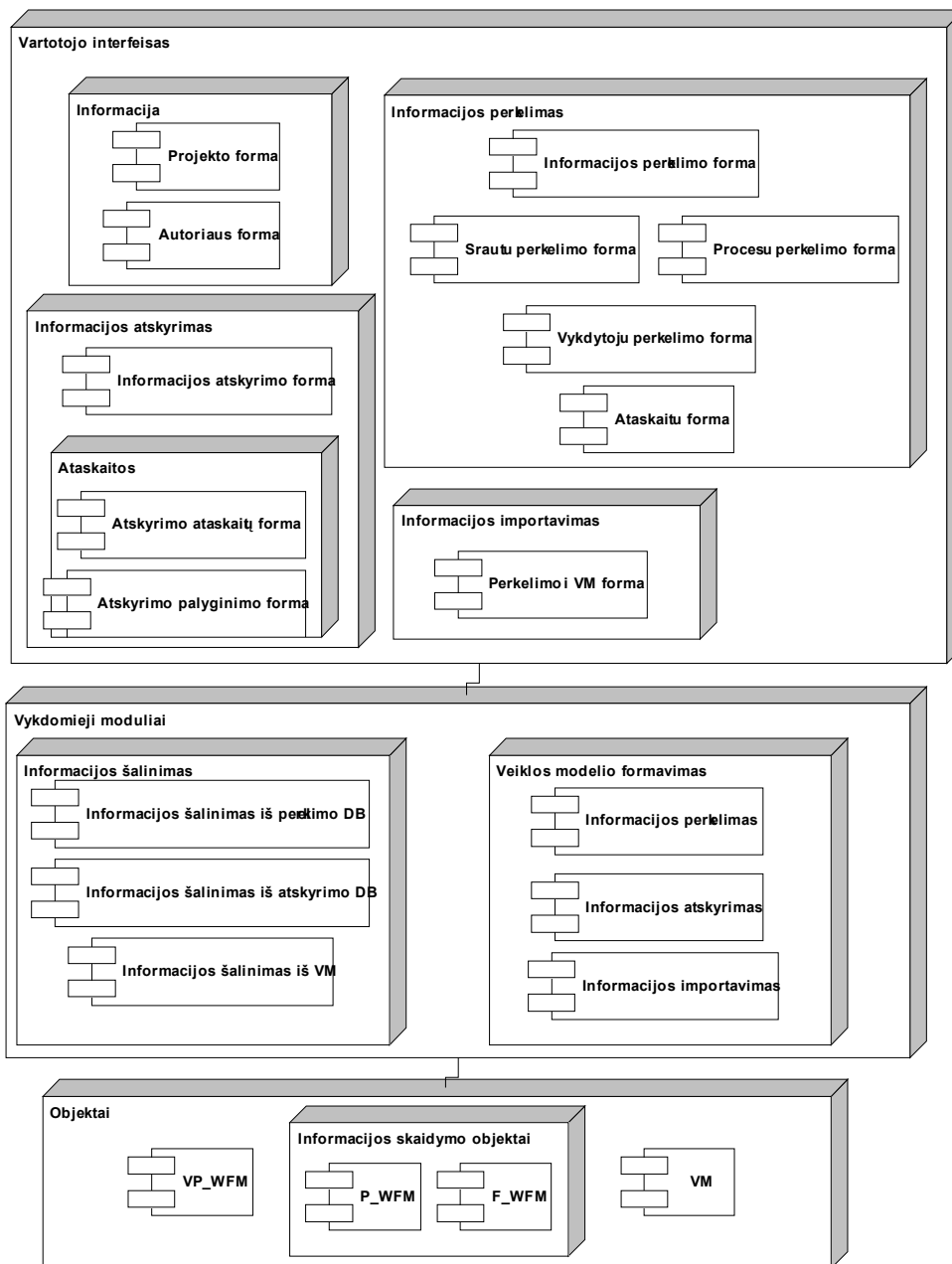
Vartotojo interfeisas susideda iš klasių kiekvienai formai, kurios pagalba vartotojas bendrauja su sistema. Juk pagrindinė sąsajos užduotis valdyti sąveiką tarp aktorių ir sistemos teikiamo funkcionalumo. Sąveiką tarp sistemos ir veikėjų modeliuoja ribinės klasės.

Vykdomųjų modulių sluoksnis tai sistemos modelio veiksmingumas – ką sistema turi daryti. Šio sluoksnio realizacijai naudojamos valdymo klasės, kurios naudojamos objektų koordinavimui, etapiškumo nustatymui ir t.t.

Objektų sluoksnis modeliuoja saugomą informaciją, kuri yra ilgalaikė ir išliekanti.

Vartotojo interfeisas yra aukščiausiam sluoksninės architektūros lygyje, tuo tarpu objektai – žemiausiam. Priežastis ta, kad nedaugelis sistemos komponentų priklauso nuo vartotojo interfeiso, o labiausiai komponentus įtakoja objektas.





9 pav. Bendras sistemos komponentų vaizdas

#### 4.1.2. Komponentų apžvalga

Komponentai, pateikti sistemos architektūros loginiame vaizde, turi būti detalai aprašyti [18]. Šiame skyriuje pateikiu loginių sistemos sluoksnių tai yra vartotojo interfeiso (3 lentelė), vykdomųjų modulių (4 lentelė) ir objektų (5 lentelė) apibūdinimą pagal B. Appleton'o pateiktus detalios architektūros komponentų apibūdinimo šablonus.

3 lentelė. Posistemė – Vartotojo interfeisas

<b>Vartotojo interfeisas</b>	
<i>Klasifikacija</i>	Posistemė
<i>Apibrėžimas</i>	Posistemė, kuri apima vartotojo sąsajos komponentus.
<i>Atsakomybės</i>	Posistemė realizuoja vartotojo sąsają, vartotojo interfeiso formų pasirinkimą ir duomenų atvaizdavimą vartotojui. Ši posistemė apjungia visas sistemos ribines klases.
<i>Apribojimai</i>	Nėra
<i>Struktūra</i>	Posistemę sudaro tokie komponentai: <ul style="list-style-type: none"> <li>✓ Projekto forma;</li> <li>✓ Autoriaus forma;</li> <li>✓ Informacijos perkėlimo forma;</li> <li>✓ Srautų perkėlimo forma;</li> <li>✓ Procesų perkėlimo forma;</li> <li>✓ Vykdytojų perkėlimo forma;</li> <li>✓ Ataskaitų forma;</li> <li>✓ Informacijos atskyrimo forma;</li> <li>✓ Atskyrimo ataskaitų forma;</li> <li>✓ Atskyrimo palyginimo forma;</li> <li>✓ Perkėlimo į VM forma.</li> </ul>
<i>Sąveikavimas</i>	Posistemė sąveikauja su vykdomųjų modulių posisteme.
<i>Resursai</i>	Šios posistemės komponentai naudoja procesorių, atmintį, monitorių, pelę ir klaviatūrą. <ul style="list-style-type: none"> <li>✓ Sistemos atmintis ir procesorius naudojami komponentų objektams sukurti, metodams iškviešti.</li> <li>✓ Klaviatūra, pelė naudojami informacijai formoje įvesti.</li> <li>✓ Monitorius naudojamas informacijos pateikimui vartotojui.</li> </ul>

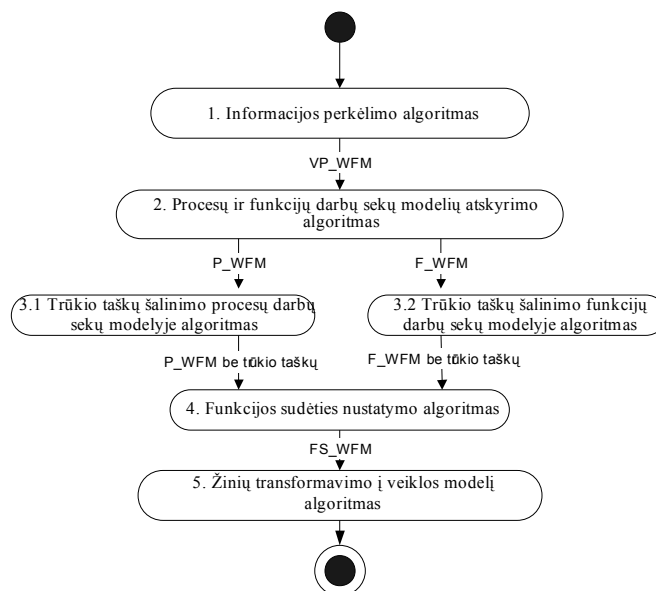
<b>Vykdomieji moduliai</b>	
<i>Klasifikacija</i>	Posistemė
<i>Apibrėžimas</i>	Posistemė, kuri apima sistemos vykdomųjų modulių komponentus.
<i>Atsakomybės</i>	Posistemė realizuoja informacijos šalinimą ir veiklos modelio formavimą. Ši posistemė apjungia visas valdymo klases.
<i>Apribojimai</i>	Informacijos šalinimas ir veiklos modelio formavimas galimas tik tada kai galima prieiga prie sistemos duomenų bazės.
<i>Struktūra</i>	Posistemę sudaro tokie komponentai: <ul style="list-style-type: none"> <li>✓ Informacijos šalinimas;</li> <li>✓ Veiklos modelio formavimas.</li> </ul>
<i>Sąveikavimas</i>	Posistemė sąveikauja su vartotojo interfeiso posisteme ir su objektų posisteme
<i>Resursai</i>	Posistemė naudojami procesoriai, atmintimi.

5. lentelė. Posistemė – Objektai

<b>Objektai</b>	
<i>Klasifikacija</i>	Posistemė
<i>Apibrėžimas</i>	Posistemė, kurioje saugomi informacijos šalinimo ir veiklos modelio formavimo duomenys.
<i>Atsakomybės</i>	Posistemė atsakinga už duomenų saugojimą, užklausų realizavimą, duomenų apdorojimą pagal užklausas.
<i>Apribojimai</i>	Nėra
<i>Struktūra</i>	Posistemę sudaro sekantys komponentai: <ul style="list-style-type: none"> <li>✓ VP_WFM;</li> <li>✓ P_WFM;</li> <li>✓ F_WFM;</li> <li>✓ VM.</li> </ul>
<i>Sąveikavimas</i>	Posistemė sąveikauja su sistemos vykdomųjų modulių posisteme.
<i>Resursai</i>	Šios posistemės komponentai naudoja procesorių ir atmintį.

## 4.2. Darbo eigos algoritmas

Dalykinės srities žinių surinkimo etape taikomi šešių tipų modifikuoti darbų sekų modeliai: veiklos procesų darbų sekų modelis (VP\_WFM), procesų darbų sekų modelis (P\_WFM), funkcijų darbų sekų modelis (F\_WFM), procesų darbų sekų modelis be trūkio taškų, funkcijų darbų sekų modelis be trūkio taškų ir funkcijos sudėties darbų sekų modelis (FS\_WFM). Šių modelių kūrimui ir žinių transformavimui į veiklos modelį taikomi penkių tipų algoritmai: informacijos perkėlimo algoritmas, procesų ir funkcijų darbų sekų modelių atskyrimo algoritmas, trūkio taškų šalinimo procesų darbų sekų modelyje algoritmas, trūkio taškų šalinimo funkcijų darbų sekų modelyje algoritmas, funkcijos sudėties nustatymo algoritmas ir žinių transformavimo į veiklos modelį algoritmas.

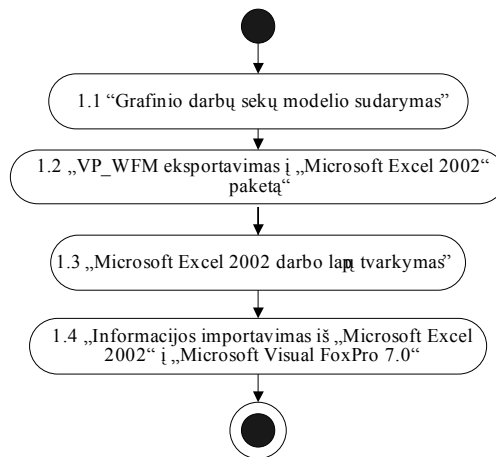


10 pav. Informacijos tvarkymo algoritmas

## 4.3. Veiklos funkcijų ir procesų informacinių poreikių surinkimas

Pirmasis darbo etapas susideda iš keturių žingsnių. 1.1 žingsniu paketu „ProVision Workbench v4.0.2 Enterprise Pro“ – „Grafinio darbų sekų modelio sudarymas“ (VP\_WFM), specifikuojantis kompiuterizuojamą dalykinę veiklos sritį. VP\_WFM skirtas nagrinėjamųjų veiklos procesų eigai aprašyti. Kai jau darbų sekų modelis - VP\_WFM - sudarytas, atliekamas sekantis pirmojo etapo žingsnis tai yra 1.2- „VP\_WFM eksportavimą į „Microsoft Excel 2002“ paketą“. Šis paketas naudojamas kaip tarpinis informacijos perkėlimui iš „ProVision Workbench v4.0“ į „Microsoft Visual FoxPro 7.0“. Šis paketas taikomas todėl, kad grafinio modeliavimo būdas leidžia informacijos eksportavimą į šį ir dar į keletą paketų. Apsispręsta pasirinkti „Microsoft Excel 2002“, nes būtent iš šio paketo įmanomas informacijos importavimas į „Microsoft Visual FoxPro 7.0“, kurio pagalba

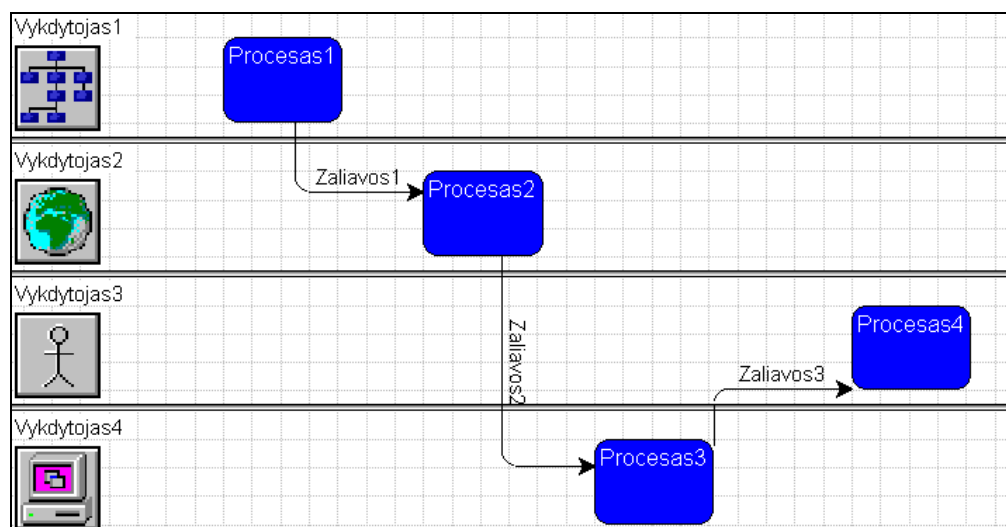
realizuojame VM valdymo informacinę sistemą. Taigi antru žingsniu atliekame - Informacijos eksportavimą iš VP\_WFM į „Microsoft Excel 2002“ darbo lapus. Atlikus minėtą žingsnį gautus „Microsoft Excel 2002“ darbo lapus reikia pasiruošti sekantiems žingsniams t.y. 1.3 pirmojo etapo žingsnis – „Microsoft Excel 2002 darbo lapų tvarkymas“ (panaikiname juose esančius nereikalingus laukus). Šis žingsnis atliekamas todėl, kad dalis laukų minėtuose darbų lapuose yra visiškai tušti. Atlikus trečiąjį žingsnį galime vykdyti 1.4 žingsnį – „Informacijos importavimą iš „Microsoft Excel 2002“ į „Microsoft Visual FoxPro 7.0“, nes visas darbas bus atliekamas būtent šiuo paketu.



11 pav. Veiklos funkcijų ir procesų surinkimo algoritmas

#### 4.3.1. Grafinio darbų sekų modelio (G\_WFM) sudarymas

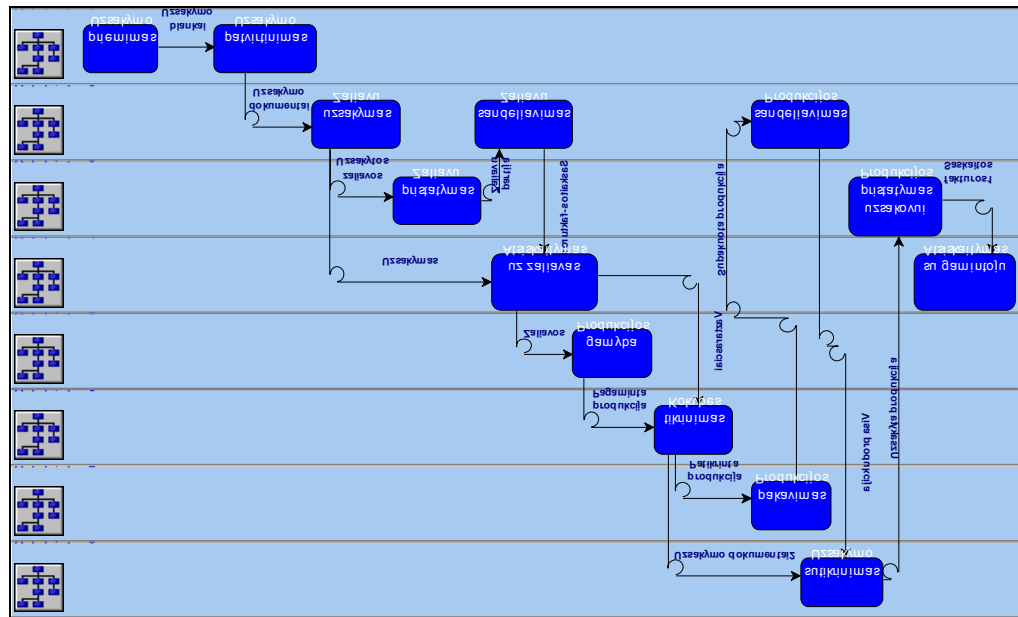
Grafinis darbų sekų modelis (VP\_WFM) skirtas nagrinėjamo veiklos proceso eigai aprašyti (12 pav.).



12 pav. Darbų sekos modelio pavyzdys

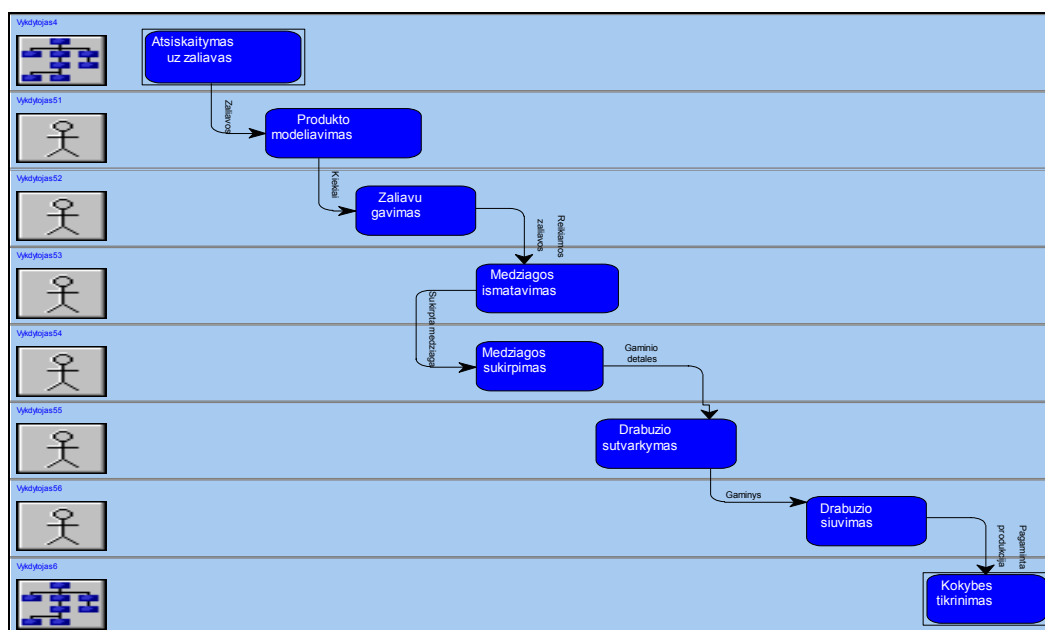
WFM vienas „takelis“ nurodo konkretaus vykdytojo atliekamus procesus. Ryšiai tarp procesų aprašo dokumentus arba objektus, perduodamus iš vieno proceso kitiems. Šis modelis įgalina tiksliai aprašyti organizacijos veiklos technologiją, atvaizduoti darbų rezultatų perdavimo tarp organizacijos padalinių tvarką. Kiekvienas WFM sudėtyje esantis procesas gali būti aprašytas detaliai atskiru darbų eigos modeliu, taip suformuojama VP\_WFM hierarchija.

Sudarome konkretų darbų sekų modelio (13 pav.) pavyzdį, kurį taikysime sekantiems darbo etapams.

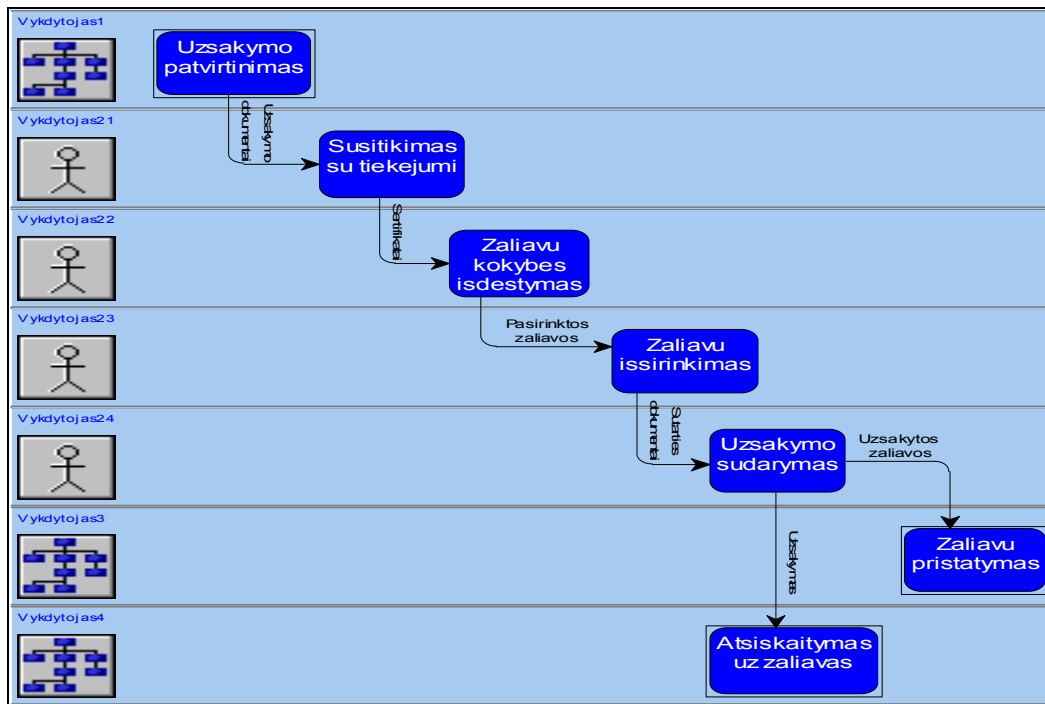


13 pav. Darbų sekos modelis (VP\_WFM), specifikuojantis procesą „Siuvimas“

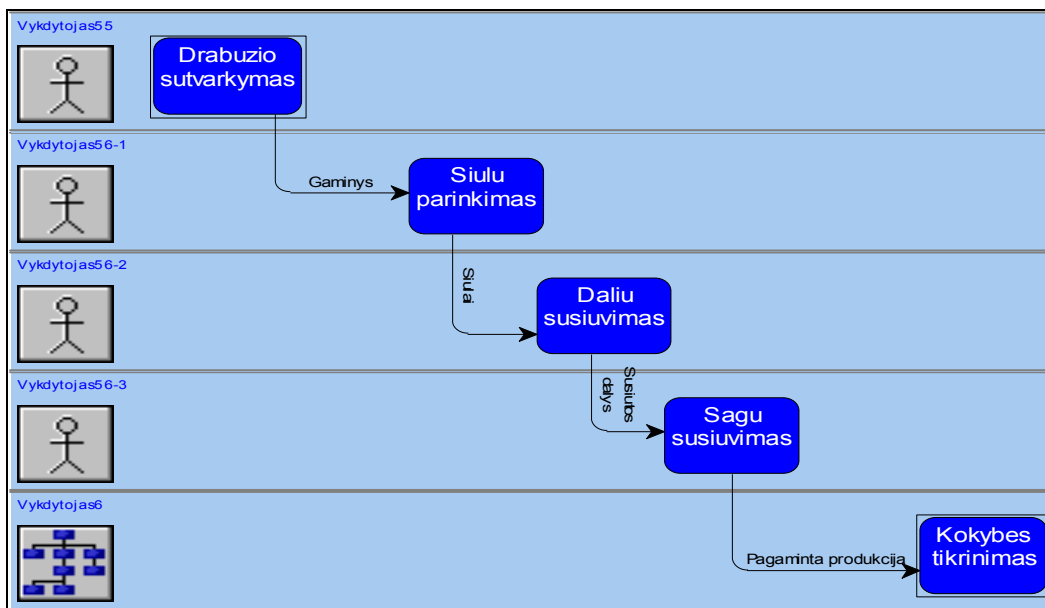
Dekomponavus 12 paveiksle pavaizduotą procesą „Siuvimas“, gaunami žemesnio lygio darbų sekų modeliai, tiksliai aprašantys „Produkcijos gamybos“ eigą (14 pav.), „Žaliavų užsakymo“ eigą (15 pav.) ir proceso „Drabužio siuvimo“ eigą (16 pav.).



14 pav. Žemesniojo lygio darbų sekos modelis, specifikuojantis procesą „Produkcijos gamyba“



15 pav. Žemesniojo lygio darbų sekos modelis, specifikuojantis procesą „Žaliavų užsakymas“

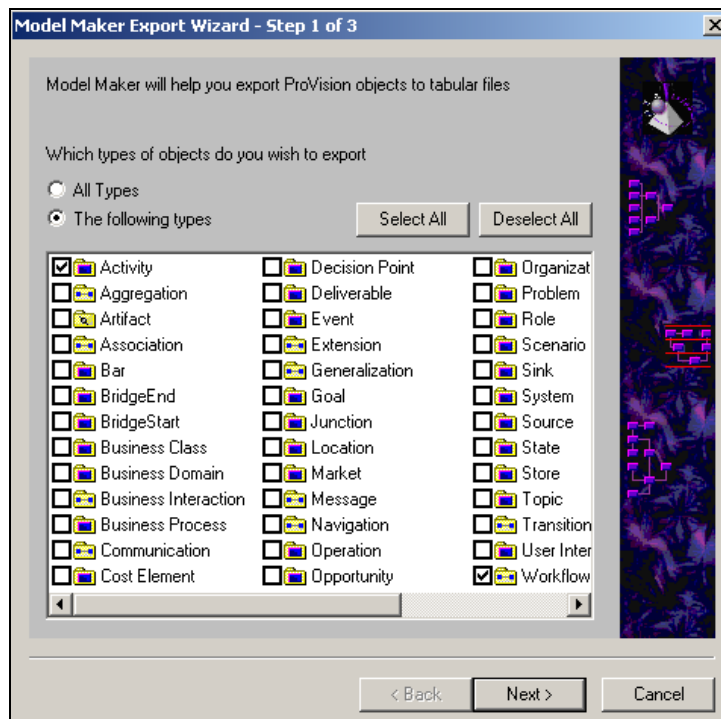


16 pav. Žemesniojo lygio darbų sekos modelis, specifikuojantis procesą „Drabužio siuvimas“

### 4.3.2. G\_WFM eksportavimas į „Microsoft Excel 2002“ paketą

Sekantis „Informacijos perkėlimo algoritmo“ žingsnis – informacijos eksportavimas į „Microsoft Excel 2002“ paketą. Šis žingsnis atliekamas tokia seka:

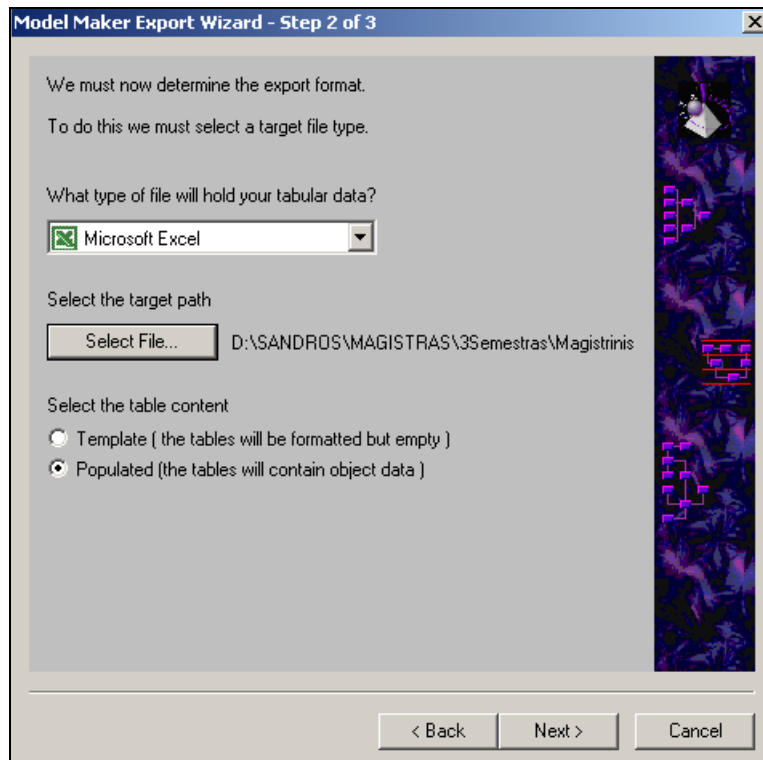
1. Pakete „ProVision Workbench v4.0.2 Enterprise Pro“ parenkame šiuos meniu punktus: *Interface -> Model Maker -> Export*;
2. Toliau pažymime pasirinkimo laukus: *Activity* ir *Workflow* langeliuose ir spaudžiame *Next*. Užžymime būtent šiuos minėtus objektus, kuriuose talpinama mums reikalinga informacija. Nereikia eksportuoti visos pateikiamos perteklinės informacijos iš sudaryto (VP\_WFM) darbų sekos modelio, kuri apsunkintų atliekama darbą.



17 pav. „Model Maker Export Wizard –Step 1 of 3“

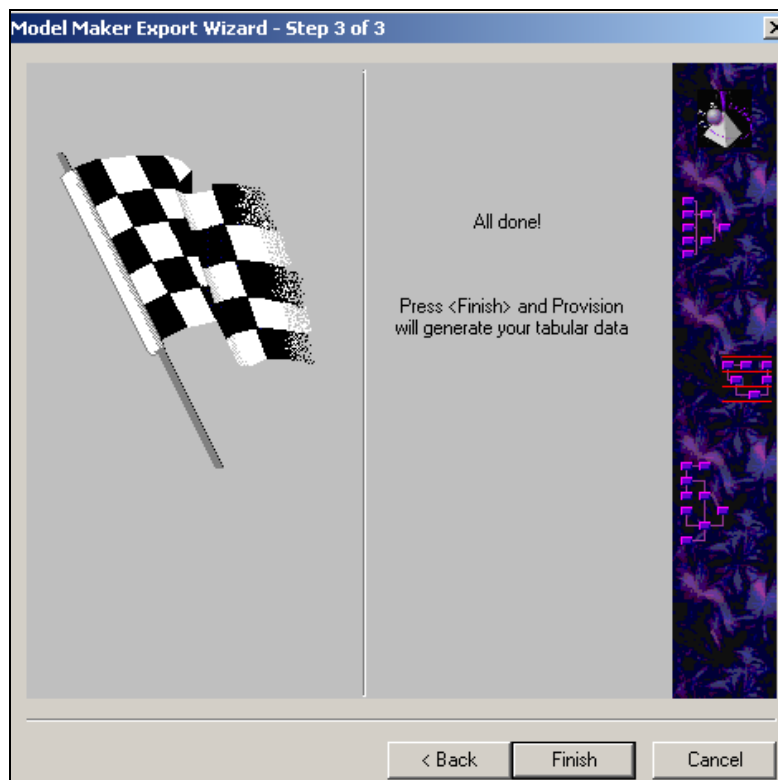
3. Sekančiame „Model Maker Export Wizard“ lange turime pasirinkti formatą ir nurodyti failą („Pirmas.xls“), kuriame bus pateikti iš VP\_WFM eksportuoti duomenys. Spaudžiame *Next*.





18 pav. „Model Maker Export Wizard –Step 2 of 3“

4. Paskutinis žingsnis – lange „Model Maker Export Wizard –Step 3 of 3“ spaudžiame *Finish*.



19 pav. „Model Maker Export Wizard –Step 3 of 3“

### 4.3.3. „Microsoft Excel 2002“ darbo lapų tvarkymas

1.3 „Informacijos perkėlimo algoritmo“ žingsnis – „Microsoft Excel 2002“ darbo lapų tvarkymas. Vykdydami informacijos eksportavimą iš VP\_WFM į „Microsoft Excel 2002“ paketo „Pirmas.xls“ failą, mes eksportavome tik dviejų (VP\_WFM) objektų - tai yra „Activity“ ir „Workflow“ - teikiamą informaciją. Atidarius minėtą („Pirmas.xls“) failą išvystame sukurtus du darbo lapus, pavadintus būtent šiais eksportuotų objektų („Activity“ ir „Workflow“) pavadinimais.

„Workflow“ darbo lapas susideda iš 28-ių laukų (*Activity End, Decision Point End, Source End, Sink End, Store End, Junction End, BridgeStart End, BridgeEnd End, Activity Start, Decision Point Start, Source Start, Sink Start, Store Start, Junction Start, BridgeStart Start, BridgeEnd Start, Capacity, Description, Discriminator, Flow Type, Transit Distribution, Transit Distribution Factor, Transit Distribution Factor2, Transit Distribution Seed, Transit Time, Transit Time Unit, Deliverable, Event*), tačiau tik 4-iuose iš jų (*Activity End, Activity Start, Flow Type ir Deliverable*) yra pateikiama informacija, importuota iš VP\_WFM. Tad visus likusius laukus reikia pašalinti, kad „Workflow“ darbo lapas būtų tvarkingas ir paruoštas tolimesniam darbui. Kad minėtų laukų nereikėtų trinti po vieną pasinaudojame sukurta makrokomanda „Tvarkyti“, kuri pateikta pagrindinėje meniu juostoje ( 20 pav.).

Atlikus minėtų laukų pašalinimą failo „Pirmas.xls“ „Workflow“ darbo lapas atrodo taip:

Activity End	Activity Start	Flow Type	Deliverable
Produkcijos gamyba	Atsiskaitymas uz zaliavas	512	Zaliavos
Kokybes tikrinimas	Produkcijos gamyba	512	Pagaminta produkcija
Produkcijos pakavimas	Kokybes tikrinimas	512	Patikrinta produkcija
Produkcijos sandeliavimas	Produkcijos pakavimas	512	Supakuota produkcija
Atsiskaitymas uz zaliavas	Zaliavu ussakymas	256	Uzsakymas
Uzsakymo sutikrinimas	Produkcijos sandeliavimas	512	Visa produkcija
Uzsakymo sutikrinimas	Kokybes tikrinimas	256	Uzsakymo dokumentai2
Produkcijos pristatymas ussakovui	Uzsakymo sutikrinimas	512	Uzsakyta produkcija
Kokybes tikrinimas	Atsiskaitymas uz zaliavas	256	Vaztarasciai
Atsiskaitymas su gamintoju	Produkcijos pristatymas ussakovui	256	Saskaitos fakturos1
Zaliavu kokybes isdestymas	Susitikimas su tiekejumi	256	Sertifikatai
Zaliavu pristatymas	Uzsakymo sudarymas	512	Uzsakytos zaliavos
Uzsakymo sudarymas	Zaliavu issirinkimas	256	Sutarties dokumentai
Zaliavu issirinkimas	Zaliavu kokybes isdestymas	512	Pasirinktos zaliavos
Atsiskaitymas uz zaliavas	Uzsakymo sudarymas	256	Uzsakymas
Uzsakymo patvirtinimas	Uzsakymo priemimas	256	Uzsakymo blankai
Produkto modeliavimas	Atsiskaitymas uz zaliavas	512	Zaliavos
Zaliavu ussakymas	Uzsakymo patvirtinimas	256	Uzsakymo dokumentai
Medziagos ismatavimas	Zaliavu gavimas	512	Reikiamos zaliavos
Zaliavu gavimas	Produkto modeliavimas	256	Kiekiai
Zaliavu pristatymas	Zaliavu ussakymas	256	Uzsakytos zaliavos
Drabuzio sutvarkymas	Medziagos sukirpimas	512	Gaminio detales
Medziagos sukirpimas	Medziagos ismatavimas	512	Sukirpta medziaga
Zaliavu sandeliavimas	Zaliavu pristatymas	512	Zaliavu partija
Kokybes tikrinimas	Drabuzio siuvinimas	512	Pagaminta produkcija
Drabuzio siuvinimas	Drabuzio sutvarkymas	512	Gaminys
Susitikimas su tiekejumi	Uzsakymo patvirtinimas	256	Uzsakymo dokumentai
Siulu parinkimas	Drabuzio sutvarkymas	512	Gaminys
Sagu susiuvimas	Daliu susiuvimas	512	Susiuotos dalys
Daliu susiuvimas	Siulu parinkimas	512	Siulai
Kokybes tikrinimas	Sagu susiuvimas	512	Pagaminta produkcija

20 pav. Failo „Pirmas.xls“ „Workflow“ darbo lapas

Šis lapas atvaizduoja procesus, srautus ir jų tipus, kurie yra susiję su tam tikrais procesais. Seka nustatoma taip: susirandamas pirmas procesas lapo lauko „*Activity Start*“ stulpelyje, kuris nėra pavaizduotas lapo lauko „*Activity End*“ stulpelyje. Šis procesas ir bus pirmas sekos elementas. Jei tokie elementai randami keli, tai bus vykdomos kelios sekos lygiagrečiai. Sekantis procesų sekos elementas bus procesas esantis lygiagrečiai pirmam surastam procesui tik jau esantis „*Activity End*“ stulpelyje. Norint rasti trečią sekos elementą reikia surasti antrą sekos procesą „*Activity Start*“ stulpelyje ir lygiagrečiai jam tik jau stulpelyje „*Activity End*“ bus pateiktas trečias ieškomos sekos procesas ir t.t. Stulpelyje „*Flow Type*“ esantys skaičiai nurodo, ar srauto tipas yra materialus ar informacinis. Pagal pasirinktą žymėjimą skaičius 256 parodo, kad srautas yra informacinis, o 512 nurodo, kad srautas yra materialus.

Sekantis failo „Pirmas.xls“ „*Activity*“ darbo lapas susideda iš 25-ių laukų (*Name, Delay Distribution, Delay Distribution Factor, Delay Distribution Factor2, Delay Distribution Seed, Delay Time, Delay Time Unit, Description, In-Queue Time, In-Queue Time Unit, Out-Queue Time, Out-Queue Time Unit, Work Distribution, Work Distribution Factor, Work Distribution Factor2, Work Distribution Seed, Work Time, Work Time Unit, Activity Parent, Business Process Parent, Market Performer, Organization Performer, Role Performer, System Performer, Operation*), tačiau kaip ir „*Workflow*“ darbo lape tik 5-iuose iš minėtų laukų (*Name, Activity Parent, Business Process Parent, Organization Performer ir Role Performer*) yra pateikiama informacija. Tad visus likusius laukus reikia pašalinti, kad darbo lapas būtų tvarkingas ir paruoštas tolimesniam darbui. Vėl gi pasinaudojame aukščiau paminėta makrokomanda „Tvarkyti“.

Atlikus minėtų laukų pašalinimą „Activity“ darbo lapas atrodo taip:

	A	B	C	D	E	F	G	H	I
1	Name	Activity Parent	Business Process Parent	Organization Performer	Role Performer				
2	Produkto modeliavimas	Produkcijos gamyba			Vykdytojas51				
3	Žaliavų gavimas	Produkcijos gamyba			Vykdytojas52				
4	Medžiagos sukirpimas	Produkcijos gamyba			Vykdytojas54				
5	Medžiagos išmatavimas	Produkcijos gamyba			Vykdytojas53				
6	Drabužio siuvimas	Produkcijos gamyba			Vykdytojas56				
7	Drabužio sutvarkymas	Produkcijos gamyba			Vykdytojas55				
8	Uzsakymo patvirtinimas		Siuvimas	Vykdytojas1					
9	Žaliavų pristatymas		Siuvimas	Vykdytojas3					
10	Susitikimas su tiekėjumi	Žaliavų užsakymas			Vykdytojas21				
11	Uzsakymo priėmimas		Siuvimas	Vykdytojas1					
12	Žaliavų išsirinkimas	Žaliavų užsakymas			Vykdytojas23				
13	Žaliavų užsakymas		Siuvimas	Vykdytojas2					
14	Uzsakymo sudarymas	Žaliavų užsakymas			Vykdytojas24				
15	Žaliavų kokybės išdėstymas	Žaliavų užsakymas			Vykdytojas22				
16	Žaliavų sandėliavimas		Siuvimas	Vykdytojas2					
17	Atsiskaitymas už žaliavas		Siuvimas	Vykdytojas4					
18	Produkcijos gamyba		Siuvimas	Vykdytojas5					
19	Kokybės tikrinimas		Siuvimas	Vykdytojas6					
20	Produkcijos pakavimas		Siuvimas	Vykdytojas7					
21	Produkcijos sandėliavimas		Siuvimas	Vykdytojas2					
22	Siūlų parinkimas	Drabužio siuvimas			Vykdytojas56-1				
23	Produkcijos pristatymas užsakovui		Siuvimas	Vykdytojas3					
24	Uzsakymo sutikrinimas		Siuvimas	Vykdytojas8					
25	Sagų susiuvimas	Drabužio siuvimas			Vykdytojas56-3				
26	Dalių susiuvimas	Drabužio siuvimas			Vykdytojas56-2				
27	Atsiskaitymas su gamintoju		Siuvimas	Vykdytojas4					
28									
29									
30									
31									
32									

21 pav. Failo „Pirmas.xls“ „Activity“ darbo lapas

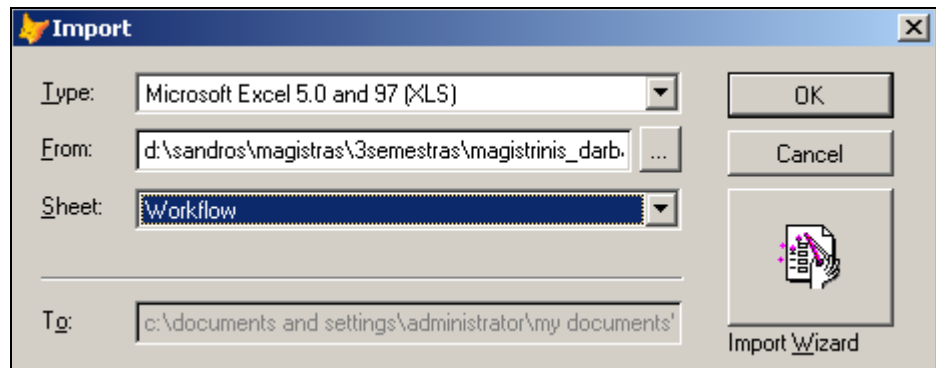
„Activity“ darbo lapas pateikia procesus, nuroydamas jų vykdytojus, tai yra koks padalinys, skyrius ar aktorius už kokį procesą yra atsakingas. Jis taip pat atvaizduoja procesų hierarchinę priklausomybę. Pagal stulpelių „Business Process Parent“ ir „Activity Parent“ procesų pavadinimus galime atrinkti procesus, kurie priklauso skirtingiems hierarchijos lygiams. Iš pateiktos informacijos darbo lape matome, kad aukščiausio hierarchijos lygio procesas yra „Siuvimas“, tačiau taip pat yra keli žemesnio hierarchijos lygio procesai tai yra – „Produkcijos gamyba“, „Žaliavų užsakymas“ ir „Drabužio siuvimas“. Pirmasis - „Produkcijos gamybos“ procesas susideda iš 6-ių subprocesų (Produkto modeliavimas, Žaliavų gavimas, Medžiagos sukirpimas, medžiagos išmatavimas, Drabužio siuvimas ir Drabužio sutvarkymas), sekantis procesas „Žaliavų užsakymas“ susideda iš 4-ių žemesnio lygio procesų (Susitikimas su tiekėjumi, Žaliavų išsirinkimas, Užsakymo sudarymas ir Žaliavų kokybės išdėstymas) ir paskutinį „Drabužio siuvimo“ procesą sudaro 3 subprocesai (Siūlų parinkimas, Sagų susiuvimas ir Dalių susiuvimas).

#### 4.3.4. Informacijos importavimas iš „Microsoft Excel 2002“ į „Microsoft Visual FoxPro 7.0“

Eksportavus iš VP\_WFM gautą informaciją į „Microsoft Excel 2002“ paketą ir jame paruošus „Workflow“ ir „Activity“ darbo lapus sekantiems etapams, sekantis žingsnis yra informacijos importavimas į „Microsoft Visual FoxPro 7.0“, nes informacija bus apdorojama būtent šiame pakete.

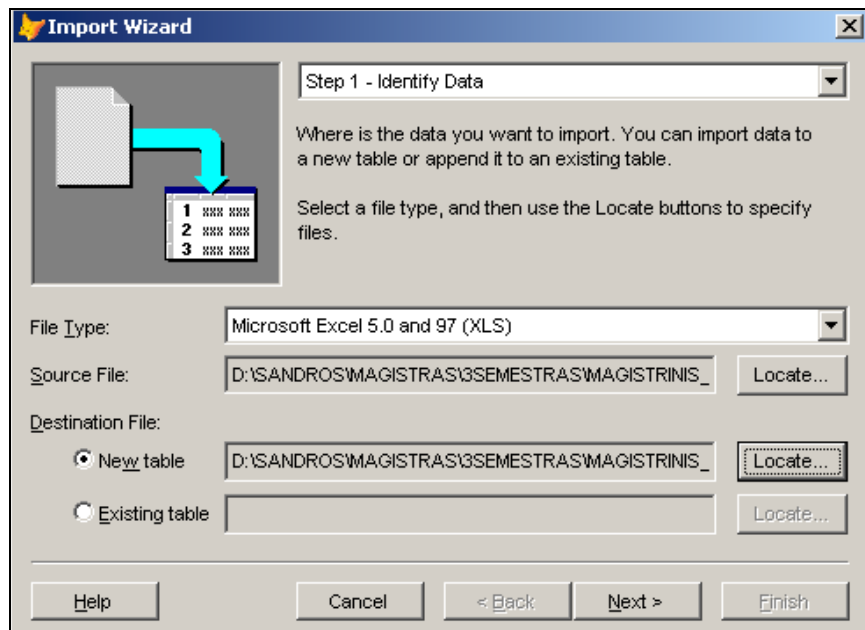
Šis „Informacijos perkėlimo algoritmo“ žingsnis atliekamas tokia seka:

1. Atsidaromas „Microsoft Visual Foxpro 7.0“ paketas, sukuriamas naujas projektas, duomenų bazė ir toliau vykdomas informacijos importavimas  
*File -> Import;*
2. Pasirenkame failą ir darbo lapą, šiuo atveju „Workflow“, iš kurio informaciją importuosime ir spaudžiame „Import Wizard“ (22 pav.);



22 pav. „Workflow Import“ langas

3. Sekančiame žingsnyje parenkame dokumento tipą, iš kurio importuosime informaciją – tai yra iš „Microsoft Excel 2002“ paketo, ir taip pat nurodome patį dokumentą, tai yra failą – „Pirmas.xls“. Toliau reikia nurodyti lentelės tipą – ar informacija bus importuojama į naują „Microsoft Visual Foxpro 7.0“ lentelę ar į jau egzistuojančią. Pasirenkame, kad informaciją kelsime į naują lentelę, tada nurodome būsimą jos pavadinimą „Procesu\_sekaimp“ ir spaudžiame *Next* (23 pav.);



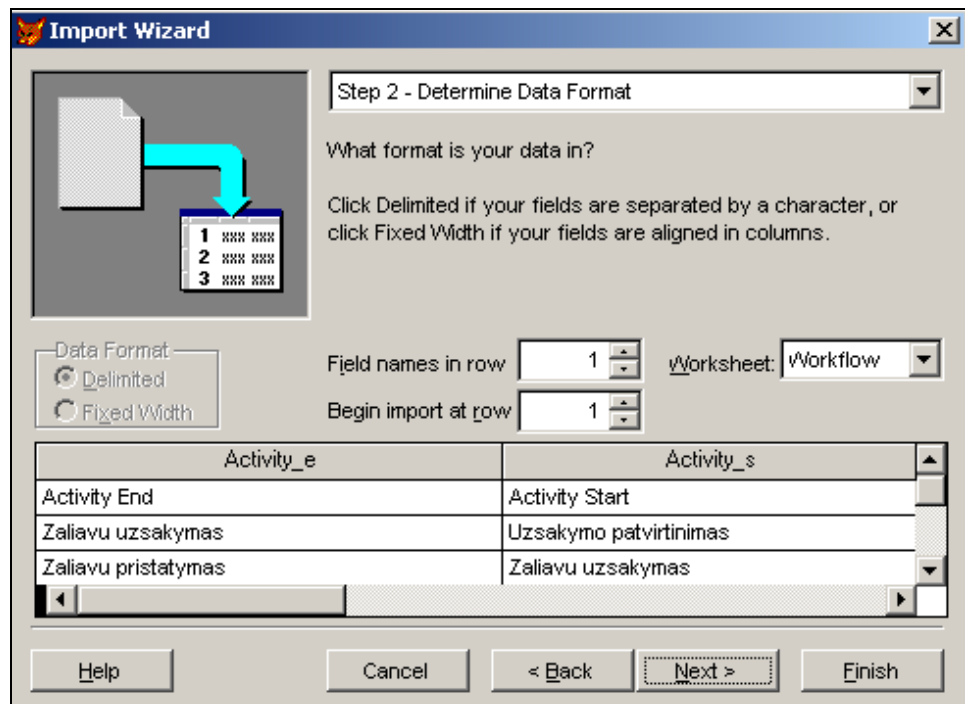
23 pav. „Import Wizard 1-as“ langas

4. Sekančiame lange galime pasirinkti ar importuojamą lentelę „Procesų\_sekaimp“ sukursime kaip nepriklausomą nuo konkrečios DB ar ne. Minėtą lentelę „Procesų\_sekaimp“ importuojama kaip nepriklausoma, nes reikalinga DB-zė sudaroma sekančiuose etapuose. Ir vėl spaudžiame *Next*;



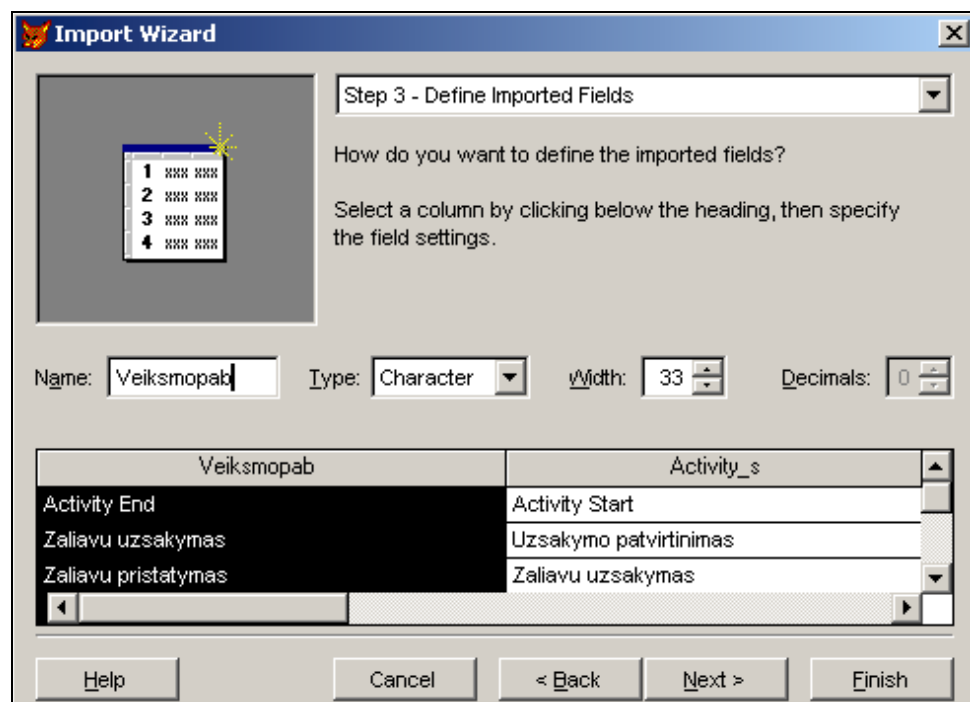
24 pav. „Import Wizard 2-as“ langas

5. 25 paveikslo „Import Wizard 3-iam“ lange nurodome, iš kurio darbo lauko, mūsų atveju „Workflow“, ir nuo kurios stulpelio eilutės bus importuojama informacija. Ir vėl spaudžiame *Next*;

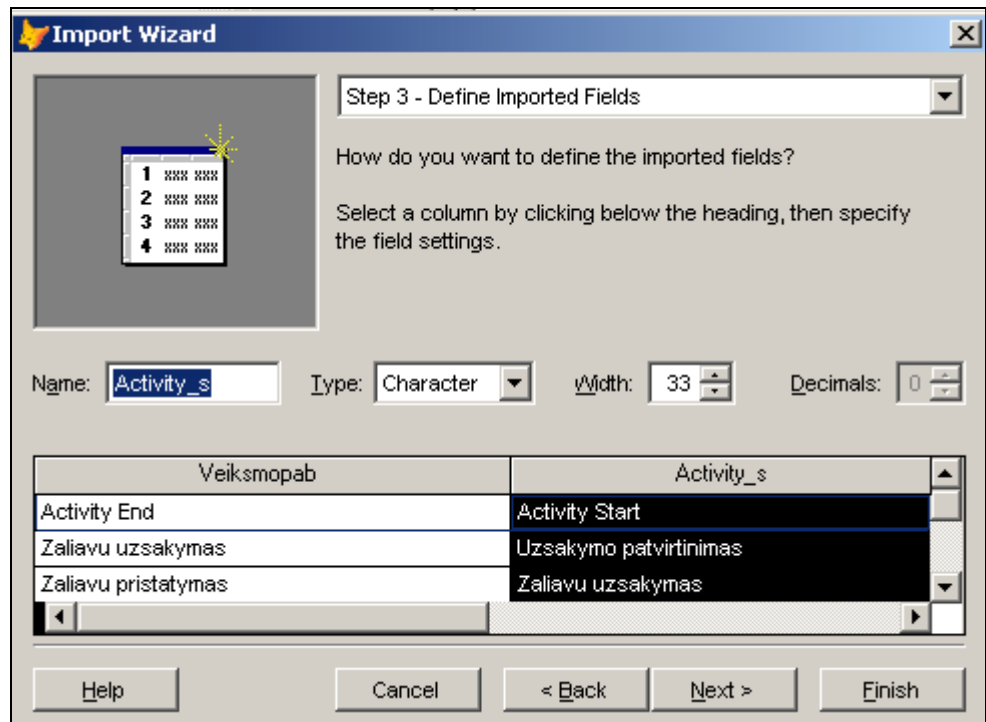


25 pav. „Import Wizard 3-ias“ langas

6. „Import Wizard 4.1-ame“ lange reikia atlikti importuojamos lentelės laukų pavadinimų koregavimą. Pateiktame langelyje „Name“ įrašome pirmo stulpelio „Activity End“ pavadinimą „Veiksmopab“ ir nurodome kitus duomenis t.y. duomenų tipą ir dydį. Toliau spragtelėdami pele užsižymime sekantį stulpelį („Import Wizard 4.2-as“ langas) „Activity Start“ ir atliekame analogiškus veiksmus. Tokią veiksmų seką vykdome tol, kol suteikiame pavadinimus visiems lentelės „Procesų sekaimp“ laukams. Sukūrus laukų pavadinimus vėl spaudžiame *Next*;



26 pav. „Import Wizard 4.1-as“ langas



27 pav. „Import Wizard 4.2-as“ langas

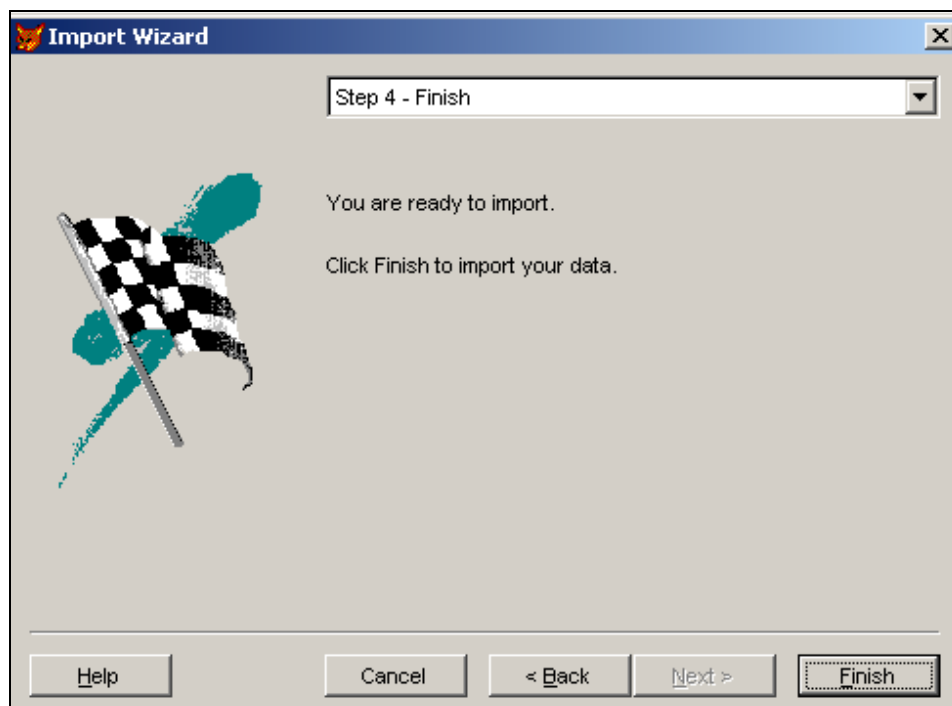
7. Sekančiame lange galime pasirinkti specialius nustatymus, tokius kaip valiutos simbolis, datos formatas ir t.t. ir spaudžiame *Next*.



28 pav. „Import Wizard 5-as“ langas

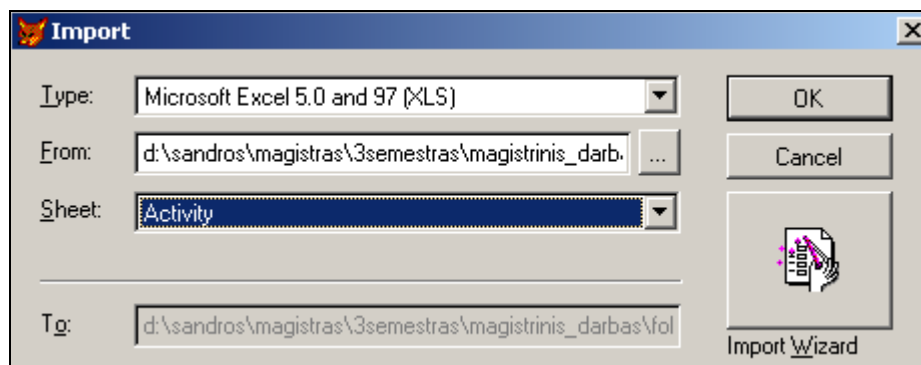
8. Paskutinis „Workflow“ informacijos importavimo žingsnis tai importavimo pabaiga. Spaudžiame *Finish*.





29 pav. „Import Wizard 6-as“ langas

9. Importavimo veiksmų seką reikia atlikti ir „Activity“ darbo lapui. Importavimas atliekamas analogiškai: pasirinkus failą ir darbo lapą, šiuo atveju „Activity“, vykdome jau žinomą aukščiau išdėstytą veiksmų procesą. Skirtumas tas, kad pasirinkus naują lentelę į kurią importuosime duomenis turime ją kitaip pavadinti tai yra „Procesai\_imp“.

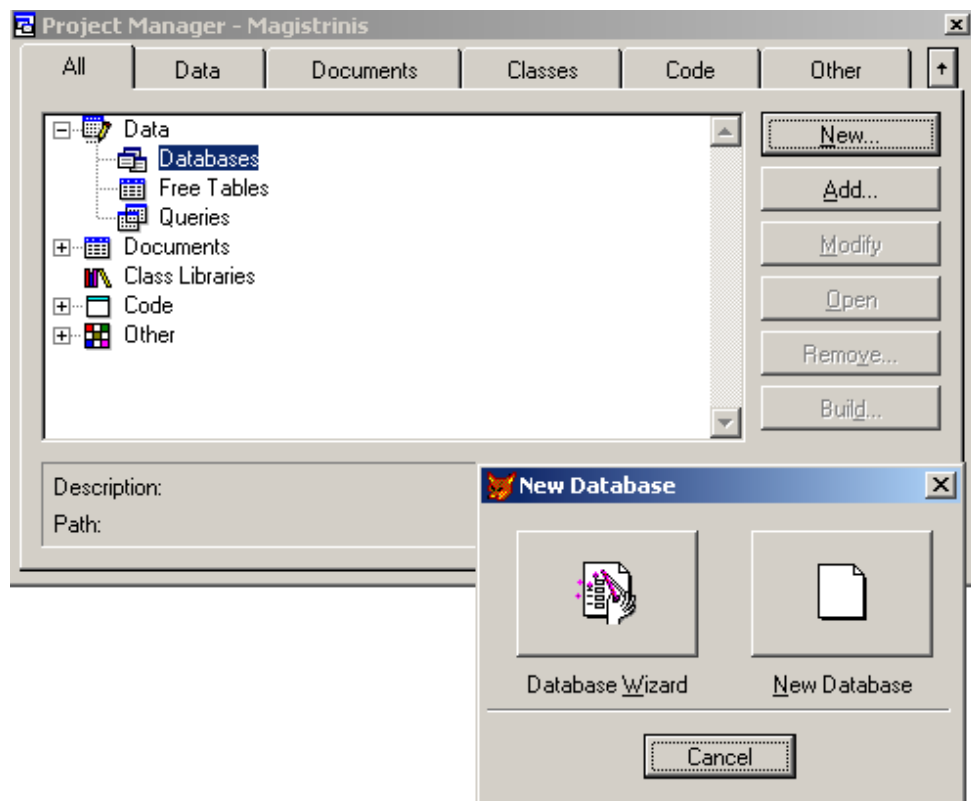


30 pav. „Activity Import“ langas

## 4.4. Informacijos perkėlimas

### 4.4.1. Tarpinės duomenų lentelės

„Informacijos importavimo“ žingsnyje sukurtas lentelės „Procesų\_sekaimp“ ir „Procesai\_imp“ reikia įkelti į DB, su kuria dirbsime sekančiuose žingsniuose. Atsidarome „Microsoft Visual FoxPro 7.0“ paketą, pairenkame *File -> New* ir pateiktame langelyje „Enter Project“ įvedame būsimo projekto pavadinimą – „Magistrinis.pjx“ (31 pav.). Toliau sukuriame DB, į kurią ir įkelsime minėtas lenteles. DB-ės sukūrimas atliekamas taip: *New Database* lange pasirenkame *New Database* ir parašome jos pavadinimą – DB. Pasižymėję „Tables“ spaudžiame *Add* mygtuką, susirandame jau sukurtą lentelę „Procesų\_sekaimp“ ir spaudžiame „OK“. Tą patį atliekame ir su lentele „Procesai\_imp“. Atlikus šiuos du veiksmus šias lenteles turime išvysti DB lentelių sąrašė.



31 pav. DB kūrimo langas

Po atliktų veiksmų DB-ėje gauname informacines lenteles, kuriose talpinama sekančiuose paveiksluose pateikta informacija.

Lentelėje „Procesų\_sėkaimp“ pateikiama tokia informacija:

Veiksmopab	Veiksmopra	Tipas	Srautai
Activity End	Activity Start	Flow Type	Deliverable
Produkcijos gamyba	Atsiskaitymas uz zalavas	512	Zalavos
Kokybes tikrinimas	Produkcijos gamyba	512	Pagaminta produkcija
Produkcijos pakavimas	Kokybes tikrinimas	512	Patikrinta produkcija
Produkcijos sandelivimas	Produkcijos pakavimas	512	Supakuota produkcija
Atsiskaitymas uz zalavas	Zalavu uzsakymas	256	Uzsakymas
Uzsakymo sutikrinimas	Produkcijos sandelivimas	512	Visa produkcija
Uzsakymo sutikrinimas	Kokybes tikrinimas	256	Uzsakymo dokumentai2
Produkcijos pristatymas uzsakovui	Uzsakymo sutikrinimas	512	Uzsakyta produkcija
Kokybes tikrinimas	Atsiskaitymas uz zalavas	256	Vaztarasciai
Atsiskaitymas su gamintoju	Produkcijos pristatymas uzsakovui	256	Saskaitos fakturos1
Zalavu kokybes isdestymas	Sustikimas su tekejumi	256	Sentikatai
Zalavu pristatymas	Uzsakymo sudarymas	512	Uzsakytos zalavos
Uzsakymo sudarymas	Zalavu issirinkimas	256	Sutarties dokumentai
Zalavu issirinkimas	Zalavu kokybes isdestymas	512	Pasirinktos zalavos
Atsiskaitymas uz zalavas	Uzsakymo sudarymas	256	Uzsakymas
Uzsakymo patvirtinimas	Uzsakymo priemimas	256	Uzsakymo blankai
Produkto modelivimas	Atsiskaitymas uz zalavas	512	Zalavos
Zalavu uzsakymas	Uzsakymo patvirtinimas	256	Uzsakymo dokumentai
Medziagos ismatavimas	Zalavu gavimas	512	Reikiamos zalavos
Zalavu gavimas	Produkto modelivimas	256	Kiekiai
Zalavu pristatymas	Zalavu uzsakymas	256	Uzsakytos zalavos
Drabuzio sutvarkymas	Medziagos sukirpimas	512	Gaminio detales
Medziagos sukirpimas	Medziagos ismatavimas	512	Sukipta medziaga
Zalavu sandelivimas	Zalavu pristatymas	512	Zalavu partija
Kokybes tikrinimas	Drabuzio siuvinimas	512	Pagaminta produkcija
Drabuzio siuvinimas	Drabuzio sutvarkymas	512	Gaminys
Sustikimas su tekejumi	Uzsakymo patvirtinimas	256	Uzsakymo dokumentai
Siulu parinkimas	Drabuzio sutvarkymas	512	Gaminys
Sagu susiuvimas	Daliu susiuvimas	512	Susiuotos dalys
Daliu susiuvimas	Siulu parinkimas	512	Siulai
Kokybes tikrinimas	Sagu susiuvimas	512	Pagaminta produkcija
Atsiskaitymas uz zalavas	Zalavu sandelivimas	256	Saskaitos fakturos

32 pav. Lentelė „Procesų\_sėkaimp“

Ši lentelė atvaizduoja procesų sekas ir srautus bei srautų tipus, kurie yra susiję su tam tikrais procesais. O lentelėje „Procesai\_imp“ pateikiama tokia mus dominanti informacija:

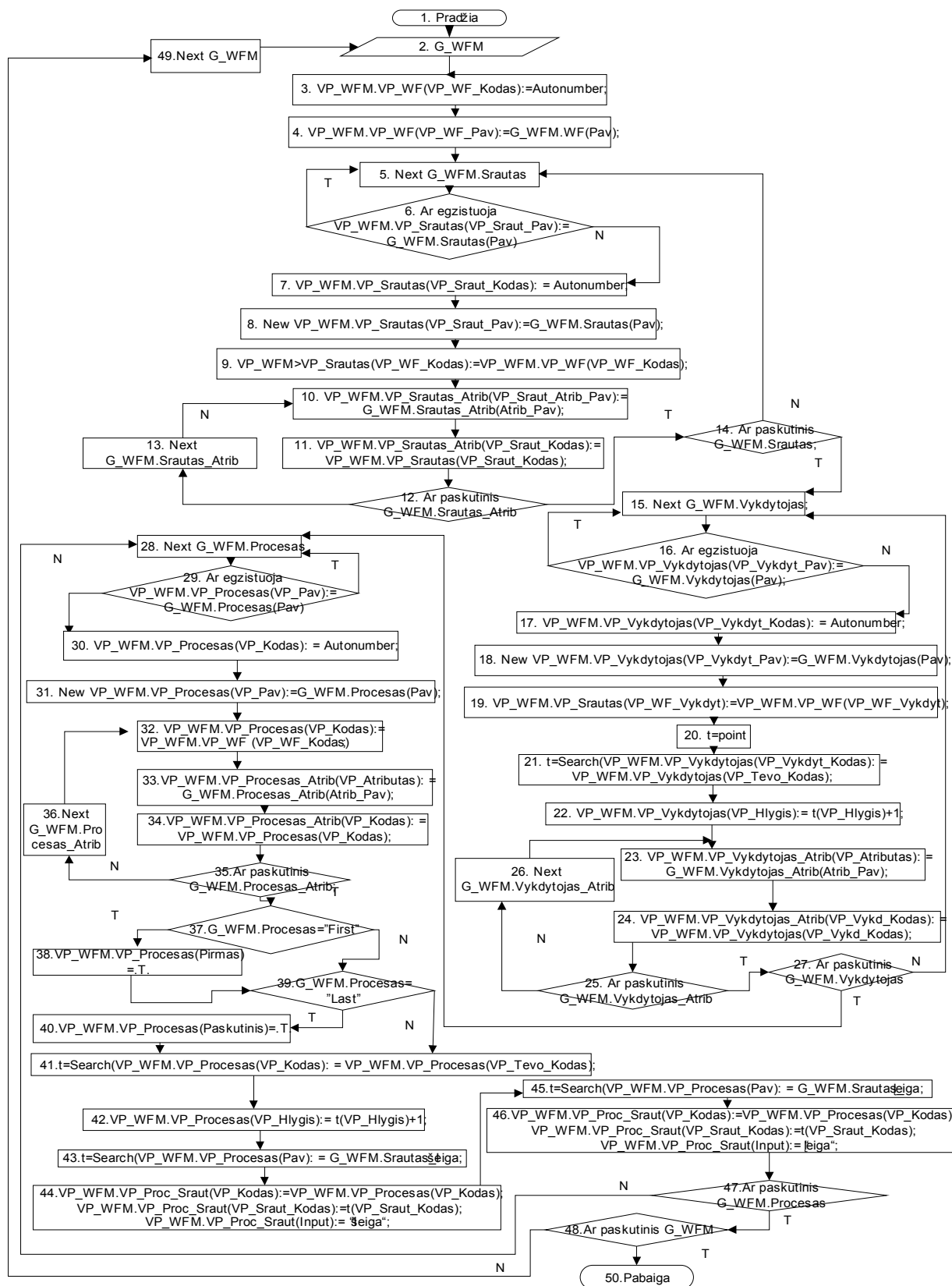
Name	Veiksmo_t	Proceso_t	Vykdytojas	V_vyk
Activity Parent	Business Process Parent	Organization Performer	Role Performer	
Produkto modelivimas	Produkcijos gamyba		Vykdytojas51	
Zalavu gavimas	Produkcijos gamyba		Vykdytojas52	
Medziagos sukirpimas	Produkcijos gamyba		Vykdytojas54	
Medziagos ismatavimas	Produkcijos gamyba		Vykdytojas53	
Drabuzio siuvinimas	Produkcijos gamyba		Vykdytojas56	
Drabuzio sutvarkymas	Produkcijos gamyba		Vykdytojas55	
Uzsakymo patvirtinimas	Siuvinimas	Vykdytojas1		
Zalavu pristatymas	Siuvinimas	Vykdytojas3		
Sustikimas su tekejumi	Zalavu uzsakymas		Vykdytojas21	
Uzsakymo priemimas	Siuvinimas	Vykdytojas1		
Zalavu issirinkimas	Zalavu uzsakymas		Vykdytojas23	
Zalavu uzsakymas	Siuvinimas	Vykdytojas2		
Uzsakymo sudarymas	Zalavu uzsakymas		Vykdytojas24	
Zalavu kokybes isdestymas	Zalavu uzsakymas		Vykdytojas22	
Zalavu sandelivimas	Siuvinimas	Vykdytojas2		
Atsiskaitymas uz zalavas	Siuvinimas	Vykdytojas4		
Produkcijos gamyba	Siuvinimas	Vykdytojas5		
Kokybes tikrinimas	Siuvinimas	Vykdytojas6		
Produkcijos pakavimas	Siuvinimas	Vykdytojas7		
Produkcijos sandelivimas	Siuvinimas	Vykdytojas2		
Siulu parinkimas	Drabuzio siuvinimas		Vykdytojas56-1	
Produkcijos pristatymas uzsakovui	Siuvinimas	Vykdytojas3		
Uzsakymo sutikrinimas	Siuvinimas	Vykdytojas8		
Sagu susiuvimas	Drabuzio siuvinimas		Vykdytojas56-3	
Daliu susiuvimas	Drabuzio siuvinimas		Vykdytojas56-2	
Atsiskaitymas su gamintoju	Siuvinimas	Vykdytojas4		

33 pav. Lentelė „Procesai\_imp“





### Detalus informacijos perkėlimo algoritmas:



36 pav. Detalus informacijos perkėlimo algoritmas

#### 4.4.4. Informacijos perkėlimo algoritmo aprašymas

36 –ame paveiksle pateikto informacijos perkėlimo algoritmo aprašymas:

- 1) Pradžia;
- 2) Imame grafinį darbų sekų modelį (**G\_WFM**);
- 3) Sugeneruojam įterpiamo Workflow modelio „unikalų“ kodą (negali būti dviejų vienodų kodų): **VP\_WFM.VP\_WF(VP\_WF\_Kodas)**;
- 4) Įterpiame pasirinktą Workflow modelį į VP\_WFM bazę:  
**VP\_WFM.VP\_Srautas(VP\_Sraut\_Pav):=G\_WFM.Srautas(Pav)**;
- 5) Imame šio G\_WFM srautą (sekantį srautą);
- 6) Tikriname ar VP\_WFM bazėje egzistuoja pasirinktas G\_WFM srautas:  
**VP\_WFM.VP\_Srautas(VP\_Sraut\_Pav):=G\_WFM.Srautas(Pav)**;
  - a. Jei pasirinktas srautas neegzistuoja VP\_WFM bazėje, vykdomas 5 žingsnis;
  - b. Jei pasirinktas srautas egzistuoja kartojame 4 žingsnį;
- 7) Sugeneruojam įterpiamo srauto „unikalų“ kodą (negali būti dviejų vienodų kodų):  
**VP\_WFM.VP\_Srautas(VP\_Sraut\_Kodas)**;
- 8) Įterpiame pasirinktą srautą į VP\_WFM bazę:  
**VP\_WFM.VP\_Srautas(VP\_Sraut\_Pav):=G\_WFM.Srautas(Pav)**;
- 9) Nustatome, kad įterpiamas srautas priklauso pasirinktam Workflow modeliui:  
**VP\_WFM.VP\_Srautas(VP\_Sraut\_Kodas):=VP\_WFM.VP\_WF (VP\_WF\_Kodas)**;
- 10) Įrašome pasirinkto G\_WFM srauto papildomą atributą į VP\_WFM bazę:  
**VP\_WFM.VP\_Srautas\_Atrib(VP\_Sraut\_Atrib\_Pav):=G\_WFM.Srautas\_Atrib(Atrib\_Pav)**;
- 11) Nustatome, kad įterpiamas srauto atributas priklauso pasirinktam srautui:  
**VP\_WFM.VP\_Srautas\_Atrib(VP\_Sraut\_Kodas):=VP\_WFM.VP\_Srautas(VP\_Sraut\_Kodas)**;
- 12) Tikriname ar paskutinis atributas susijęs su pasirinktu G\_WFM srautu (**G\_WFM.Srautas\_Atrib**):
  - a. Jei tai nepaskutinis pasirinkto G\_WFM srauto atributas, vykdomas 13 žingsnis;
  - b. Jei tai paskutinis atributas - vykdomas 14 žingsnis;
- 13) Parenkamas kitas G\_WFM srauto atributas (**G\_WFM.Srautas\_Atrib**) ir kartojamas 10 žingsnis;
- 14) Tikriname ar tai paskutinis G\_WFM srautas (**G\_WFM.Srautas**):
  - a. Jei tai nepaskutinis G\_WFM srautas, vykdomas 5 žingsnis;
  - b. Jei tai paskutinis G\_WFM srautas, vykdomas 15 žingsnis;

- 15) Imame šio G\_WFM vykdytoją (sekantį vykdytoją) (**G\_WFM.Vykdytojas**):
- 16) Tikriname ar VP\_WFM bazėje egzistuoja pasirinktas G\_WFM vykdytojas  
**VP\_WFM.Vykdytojas(VP\_Vykdyt\_Pav):=G\_WFM.Vykdytojas(Pav)**:
- a. Jei pasirinktas vykdytojas neegzistuoja VP\_WFM bazėje, vykdomas 17 žingsnis;
  - b. Jei pasirinktas vykdytojas egzistuoja - kartojamas 15 žingsnis;
- 17) Sugeneruojam įterpiamo Vykdytojo „unikalų“ kodą (negali būti dviejų vienodų kodų):  
**VP\_WFM.VP\_Vykdytojas(VP\_Vykdyt\_Kodas)**;
- 18) Įterpiame pasirinktą vykdytoją į VP\_WFM bazę :  
**VP\_WFM.VP\_Vykdytojas(VP\_Vykdyt\_Kodas):=G\_WFM.Vykdytojas(Pav)**;
- 19) Nustatome, kad įterpiamas vykdytojas priklauso pasirinktam Workflow modeliui:  
**VP\_WFM.VP\_Vykdytojas(VP\_WF\_Kodas):=VP\_WFM.VP\_WF (VP\_WF\_Kodas)**;
- 20) Inicijuojame rodyklę vykdytojo „tėvo“ kodo paieškai.
- 21) Ieškome pasirinkto vykdytojo „tėvo“ kodo:  
**t=Search(VP\_WFM.VP\_Vykdytojas(VP\_Vykdyt\_Kodas):=VP\_WFM.VP\_Vykdytojas(VP\_Tevo\_Kodas)**;
- 22) Nustatom vykdytojo hierarchijos lygį ir „tėvo“ (aukštesnio vykdytojo) kodą; Pradžioje šie parametrai lygūs atitinkamai 1 ir 0. Kadangi pradžioje vykdytojas yra pirmo hierarchijos lygio ir jis nepriklauso aukštesniojo lygio vykdytojams :  
**VP\_WFM.VP\_Vykdytojas(VP\_Hlygis):=t(VP\_Hlygis)+1**;
- 23) Įrašome pasirinkto G\_WFM vykdytojo papildomą atributą į VP\_WFM bazę:  
**VP\_WFM.VP\_Vykdytojas\_Atrib(VP\_Atributas):=G\_WFM.Vykdytojas\_Atrib(Atrib\_Pav)**;
- 24) Nustatome, kad įterpiamas vykdytojo atributas priklauso pasirinktam vykdytojui:  
**VP\_WFM.VP\_Vykdytojas\_Atrib(VP\_Vykd\_Kodas):=VP\_WFM.VP\_Vykdytojas(VP\_Vykd\_Kodas )**;
- 25) Tikriname ar paskutinis atributas susijęs su pasirinktu G\_WFM vykdytoju (**G\_WFM.Vykdytojas\_Atrib**)
- a. Jei tai nepaskutinis pasirinkto G\_WFM vykdytojo atributas, tada vykdomas 26 žingsnis;
  - b. Jei tai paskutinis atributas - vykdomas 28 žingsnis;
- 26) Parenkamas kitas G\_WFM vykdytojo atributas (**G\_WFM.Vykdytojas\_Atrib**) ir kartojamas 23 žingsnis;
- 27) Tikriname ar tai paskutinis G\_WFM vykdytojas (**G\_WFM.Vykdytojas**):
- a. Jei tai nepaskutinis G\_WFM vykdytojas, tai kartojamas 15 žingsnis;
  - b. Jei tai paskutinis G\_WFM srautas - vykdomas 28 žingsnis;



- 28) Imame šio G\_WFM procesą (sekantį procesą) (**G\_WFM.Procesas**);
- 29) Tikriname ar VP\_WFM bazėje egzistuoja pasirinktas G\_WFM procesas  
**VP\_WFM.VP\_Procesas(VP\_Sraut\_Pav):=G\_WFM.Procesas(Pav)**;
- Jei pasirinktas procesas neegzistuoja VP\_WFM bazėje, vykdomas 30 žingsnis;
  - Jei pasirinktas procesas egzistuoja kartojamas 28 žingsnis;
- 30) Sugeneruojam įterpiamo proceso „unikalų“ kodą (negali būti dviejų vienodų kodų):  
**VP\_WFM.VP\_Procesas(VP\_Kodas)**;
- 31) Įterpiame pasirinktą procesą į VP\_WFM bazę:  
**VP\_WFM.VP\_Procesas(VP\_Pav):=G\_WFM.Procesas(Pav)**;
- 32) Nustatome, kad įterpiamas procesas priklauso pasirinktam Workflow modeliui:  
**VP\_WFM.VP\_Procesas(VP\_Kodas):=VP\_WFM.VP\_WF (VP\_WF\_Kodas)**;
- 33) Įrašome pasirinkto G\_WFM proceso papildomą atributą į VP\_WFM bazę:  
**VP\_WFM.VP\_Procesas\_Atrib(VP\_Atributas):=G\_WFM.Srautas\_Atrib (Atrib\_Pav)**;
- 34) Nustatome, kad įterpiamas proceso atributas priklauso pasirinktam srautui:  
**VP\_WFM.VP\_Procesas\_Atrib(VP\_Kodas):=VP\_WFM.VP\_Procesas (VP\_Kodas)**;
- 35) Tikriname ar paskutinis atributas susijęs su pasirinktu G\_WFM procesu:  
**(G\_WFM.Procesas\_Atrib)**;
- Jei tai nepaskutinis pasirinkto G\_WFM proceso atributas, tada vykdomas 36 žingsnis;
  - Jei tai paskutinis atributas - vykdomas 38 žingsnis;
- 36) Parenkamas kitas G\_WFM proceso atributas (**G\_WFM.Procesas\_Atrib**) ir kartojamas 33 žingsnis;
- 37) Tikriname ar tai pirmas procesas (**G\_WFM.Procesas**);
- Jei tai pirmas procesas, tai vykdomas 38 žingsnis;
  - Jei tai nepirmas procesas, tai vykdomas 39 žingsnis;
- 38) Pažymime, VP\_WFM bazėje, kad tai pirmas procesas  
**(VP\_WFM.VP\_Procesas(Pirmas)=.T.)**;
- 39) Tikriname ar tai paskutinis procesas (**G\_WFM.Procesas**);
- Jei tai paskutinis procesas, tai vykdomas 40 žingsnis;
  - Jei tai nepaskutinis procesas, tai vykdomas 41 žingsnis;
- 40) Pažymime, VP\_WFM bazėje, kad tai paskutinis procesas  
**(VP\_WFM.VP\_Procesas(Paskutinis)=.T.)**;
- 41) Ieškome pasirinkto proceso „tėvo“ kodo:

**t=Search(VP\_WFM.VP\_Procesas(VP\_Kodas):=VP\_WFM.VP\_Procesas  
(VP\_Tevo\_Kodas);**

42) Nustatomas proceso hierarchijos lygis ir „tėvo“ (aukštesnio lygio proceso kodas). Proceso hierarchijos lygis bus lygus surasto aukštesnio lygio proceso hierarchijos lygis +1, o „tėvo“ kodas lygus surasto aukštesnio lygio proceso kodui):

**VP\_WFM.VP\_Procesas(VP\_Hlygis):=t(VP\_Hlygis)+1;**

43) Surandame proceso išeigos kodą ( Ieškoma VP\_WFM bazėje išeigos srauto kodo):**t=Search(VP\_WFM.VP\_Procesas(Pav):=G\_WFM.Srautas(Išeiga);**

44) Susiejam procesą su išeinančiu srautu (Įterpiam naują VP\_WFM. VP\_Proc\_Sraut lentelėje įrašą ir nurodome, kad tai yra proceso išeiga)

**P\_WFM.VP\_Proc\_Sraut(VP\_Kodas):=VP\_WFM.VP\_Procesas(VP\_Kodas);**

**VP\_WFM.VP\_Proc\_Sraut(VP\_Sraut\_Kodas):=t(VP\_Sraut\_Kodas);**

**VP\_WFM.VP\_Proc\_Sraut(Input):= “Išeiga“;**

45) Surandame proceso įeigos kodą ( Ieškoma VP\_WFM bazėje įeigos srauto kodo):**t=Search(VP\_WFM.VP\_Procesas(Pav):=G\_WFM.Srautas(Įeiga);**

46) Susiejam procesą su įeinančiu srautu (Įterpiam naują VP\_WFM. VP\_Proc\_Sraut lentelėje įrašą ir nurodome, kad tai yra proceso įeiga)

**VP\_WFM.VP\_Proc\_Sraut(VP\_Kodas):=VP\_WFM.VP\_Procesas(VP\_Kodas);**

**VP\_WFM.VP\_Proc\_Sraut(VP\_Sraut\_Kodas):=t(VP\_Sraut\_Kodas);**

**VP\_WFM.VP\_Proc\_Sraut(Input):= “Įeiga“;**

47) Tikriname ar tai paskutinis G\_WFM procesas (**G\_WFM.Procesas**);

a. Jei tai nepaskutinis G\_WFM procesas , kartojamas 28 žingsnis;

b. Jei tai paskutinis G\_WFM srautas - vykdomas 48 žingsnis;

48) Tikriname ar tai paskutinis G\_WFM (grafinis darbų sekų modelis);

a. Jei tai nepaskutinis G\_WFM kartojamas 49 žingsnis;

b. Jei tai paskutinis G\_WFM - vykdomas 51 žingsnis;

49) Imamas kitas G\_WFM modelis (**G\_WFM**) ir kartojamas 2 žingsnis;

50) Pabaiga.

## 4.5. Informacijos atskyrimas

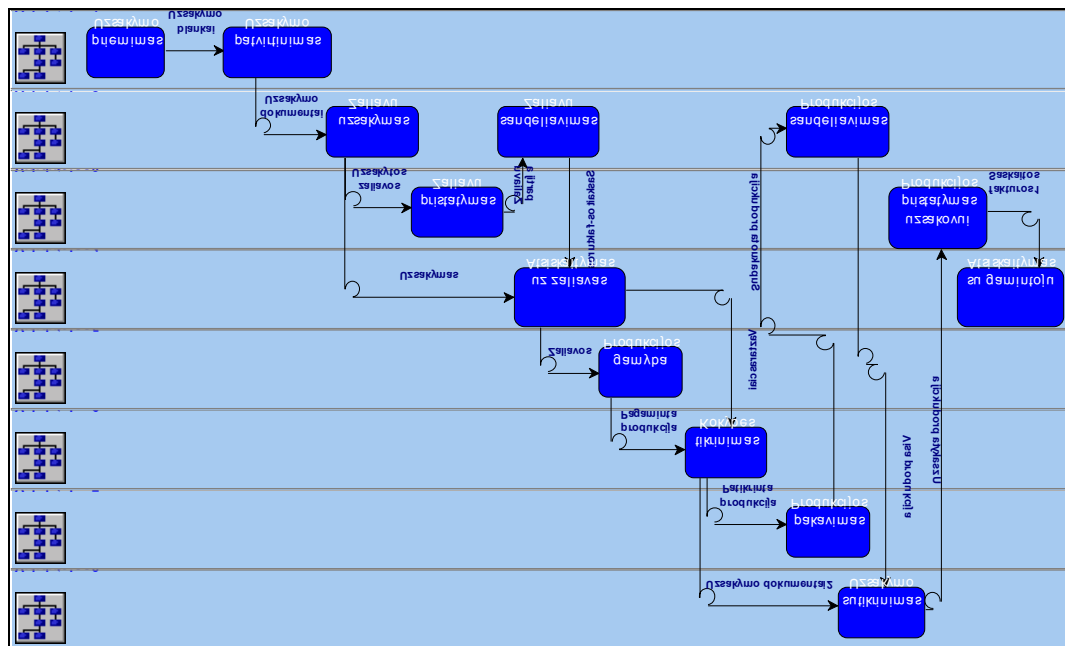
### 4.5.1. Procesų (P\_WFM) ir funkcijų (F\_WFM) darbų sekų modelių atskyrimas

Antruoju etapu VP\_WFM esanti informacija yra išskiriama į dvi dalis: tai yra į P\_WFM ir F\_WFM. Šis etapas realizuojamas analizuojant VP\_WFM veiklos procesų įeigos ir išeigos srautus. Darbų sekų modelio išskyrimą į procesų darbų sekų modelį ir funkcijų darbų sekų modelį atliekame pagal sekančias taisykles:

**Taisyklė Nr.1** - Jei veiklos proceso įeiga ir (arba) išeiga yra informacinis srautas, šis veiklos procesas, su informaciniais įeigos ir (arba) išeigos srautais funkcijų darbų sekų modelyje atvaizduojamas kaip informacinė veikla;

**Taisyklė Nr.2**- Jei veiklos proceso įeiga ir (arba) išeiga yra materialus srautas, šis veiklos procesas su materialiais įeigos ir (arba) išeigos srautais procesų darbų sekų modelyje atvaizduojamas kaip procesas;

**Taisyklė Nr.3** - Veiklos proceso vykdytojas atvaizduojamas vykdamas tą veiklą ar procesą, kuri buvo išskirta iš jo vykdomo veiklos proceso veiklos procesų darbų sekų modelyje.



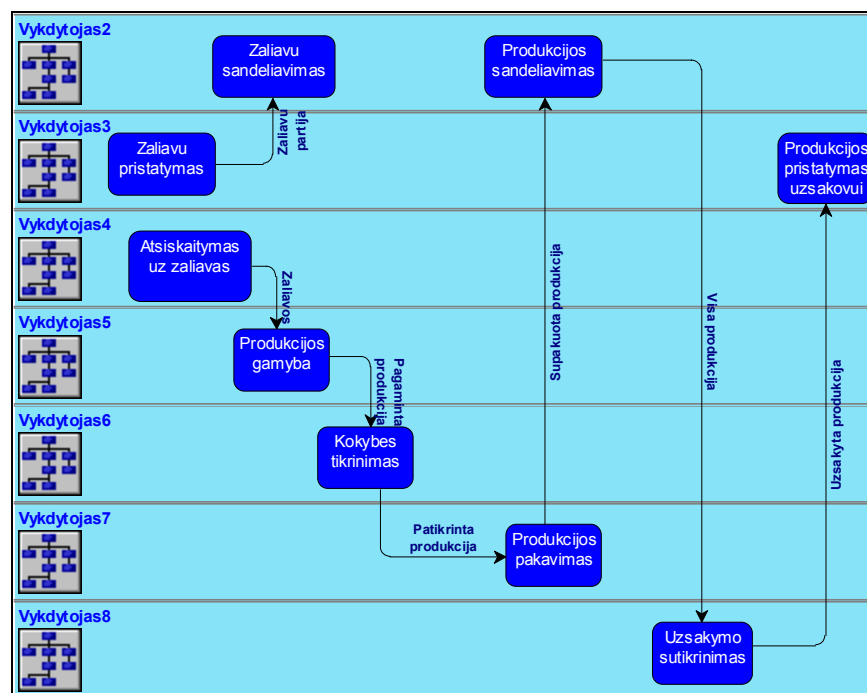
37 pav. Darbų sekos modelis (VP\_WFM), aprašantis procesą „Siuvimas“

Veiklos procesų darbų sekų modelį sudaro veiklos procesai – Užsakymo priėmimas, Užsakymo patvirtinimas, Žaliavų užsakymas, Žaliavų pristatymas, Žaliavų sandėliavimas, Atsiskaitymas už žaliavas, Produkcijos gamyba, Kokybės tikrinimas, Produkcijos pakavimas, Užsakymo sutikrinimas, Produkcijos sandėliavimas, Produkcijos pristatymas užsakovui ir Atsiskaitymas su gamintoju, informaciniai srautai – Užsakymo blankai, Užsakymo dokumentai, Užsakytos žaliavos, Užsakymas, Sąskaitos-faktūros, Važtaraščiai, Užsakymo dokumentai2 ir Sąskaitos- faktūros1 bei materialūs srautai – Žaliavų partija, Žaliavos, Pagaminta produkcija, Patikrinta produkcija, Supakuota produkcija, Visa

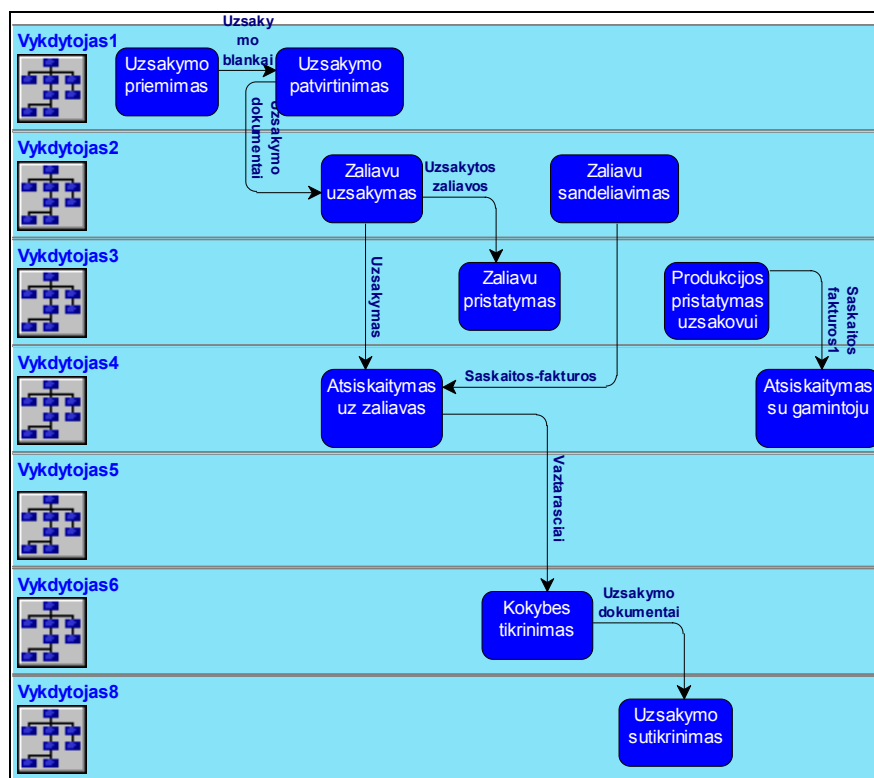
produkcija ir Užsakyta produkcija. Veiklos procesas Užsakymo priėmimas kartu su informacine išeiga Užsakymo blankai grindžiant pirma veiklos procesų darbų sekų modelio išskyrimo į procesų darbų sekų ir funkcijų darbų sekų modelius taisykle, atvaizduojamas funkcijų darbų sekų modelyje – „Siuvimas-F“ kaip ir veikla Užsakymo patvirtinimas, turinti informacinę įeigą Užsakymo blankai ir informacinę išeigą Užsakymo dokumentai. Veiklos procesų Užsakymo priėmimas ir Užsakymo patvirtinimas vykdytojas Vykdytojas1, pagal trečią atskyrimo taisyklę, funkcijos darbų sekų modelyje – „Siuvimas-F“ atvaizduojamas kaip vykdytojas Vykdytojas1. Veiklos procesas Žaliavų užsakymas kartu su informacine įeiga Užsakymo dokumentai ir informacinė išeiga Užsakymo žaliavos taip pat atvaizduojamas funkcijų darbų sekų modelyje – „Siuvimas-F“ pagal pirmą veiklos darbų sekų modelio išskyrimo į procesų darbų sekų ir funkcijų darbų sekų modelius taisyklę. Šios veiklos vykdytojas Vykdytojas2 remiantis trečia taisykle atskyrimo taisykle atvaizduojamas kaip vykdytojas Vykdytojas2 funkcijų darbų sekų modelyje – „Siuvimas-F“. Veiklos procesas Žaliavų pristatymas su informacine įeiga Užsakytos žaliavos remiantis pirma taisykle atvaizduojamas funkcijų darbų sekų modelyje – „Siuvimas-F“ ir taip pat veiklos procesas Žaliavų pristatymo su materialia išeiga Žaliavų partija atvaizduojamas procesų darbų sekų modelyje – „Siuvimas-P“ remiantis antra taisykle. Pagal trečią veiklos procesų atskyrimo taisyklę vykdytojas Vykdytojas3 atvaizduojamas ir funkcijų darbų sekų modelyje – „Siuvimas-F“, ir procesų darbų sekų modelyje – „Siuvimas-P“ kaip vykdytojas Vykdytojas3. Veiklos procesas – Žaliavų sandėliavimas - remiantis antra taisykle su materialia įeiga Žaliavų partija atvaizduojamas procesų darbų sekų modelyje – „Siuvimas-P“, o su informacine išeiga Sąskaitos – faktūros atvaizduojamas funkcijų darbų sekų modelyje – „Siuvimas-F“ pagal pirmąją taisyklę. Pagal trečią veiklos procesų atskyrimo taisyklę vykdytojas Vykdytojas2 atvaizduojamas ir funkcijų darbų sekų modelyje – „Siuvimas-F“, ir procesų darbų sekų modelyje – „Siuvimas-P“ kaip vykdytojas Vykdytojas2. Sekantis darbų sekų modelio veiklos procesas Atsiskaitymas už žaliavas su informacine įeiga atvaizduojamas funkcijų darbų sekų modelyje – „Siuvimas-F“ pagal pirmą taisyklę, o pagal antrąją taisyklę procesas Atsiskaitymas už žaliavas atvaizduojamas procesų darbų sekų modelyje – „Siuvimas-P“. Vykdytojas4 atvaizduojamas ir funkcijų darbų sekų modelyje – „Siuvimas-F“, ir procesų darbų sekų modelyje – „Siuvimas-P“ kaip Vykdytojas4 pagal trečiąją taisyklę. Procesas – Produkcijos gamyba – su materialia įeiga ir materialia išeiga pagal pirmą taisyklę atvaizduojamas procesų darbų sekų modelyje – „Siuvimas-P“. Vykdytojas5 atvaizduojamas procesų darbų sekų modelyje – „Siuvimas-P“ kaip vykdytojas Vykdytojas5. Sekantis veiklos procesas – Kokybės tikrinimas – turi ir materialią įeigą ir informacinę įeigą taip pat turi materialią bei informacinę išeigą tai remiantis pirma ir antra taisyklėmis veiklos procesas atvaizduojamas ir funkcijų darbų sekų modelyje – „Siuvimas-F“, ir procesų darbų sekų modelyje – „Siuvimas-P“. Pagal trečią veiklos procesų atskyrimo taisyklę vykdytojas Vykdytojas6 atvaizduojamas ir funkcijų darbų sekų modelyje – „Siuvimas-F“, ir procesų darbų sekų modelyje – „Siuvimas-P“ kaip vykdytojas Vykdytojas6. Procesas

Produkcijos pakavimas su materialia įeiga ir materialia išeiga pagal antrą taisyklę atvaizduojamas procesų darbų sekų modelyje – „Siuvimas-P“, o Vykdytojas7 pagal trečiąją taisyklę taip pat atvaizduojamas procesų darbų sekų modelyje – „Siuvimas-P“ kaip Vykdytojas7. Procesų darbų sekų modelyje – „Siuvimas-P“ taip pat atvaizduojamas procesas Produkcijos sandėliavimas, nes jis turi materialią įeigą ir materialią išeigą. Vykdytojas2 taip pat atvaizduojamas procesų darbų sekų modelyje – „Siuvimas-P“ pagal trečiąją taisyklę. Užsakymo sutikrinimas pagal pirmą taisyklę atvaizduojamas funkcijų darbų sekų modelyje – „Siuvimas-F“ su įeiga Užsakymo dokumentai2 nes ši įeiga informacinė, o pagal antrą taisyklę Užsakymo sutikrinimas su materialia įeiga Visa produkcija ir materialia išeiga Užsakyta produkcija atvaizduojamas procesų darbų sekų modelyje – „Siuvimas-P“. Vykdytojas8 atvaizduojamas ir procesų ir funkcijų darbų sekų modeliuose. Procesas Produkcijos pristatymas užsakovui su įeiga Užsakyta produkcija atvaizduojamas procesų darbų sekų modelyje – „Siuvimas-P“ pagal antrą taisyklę nes ši įeiga materialinė, o pagal pirmą taisyklę su informacine išeiga Produkcijos pristatymas užsakovui atvaizduojamas funkcijų darbų sekų modelyje – „Siuvimas-F“ nes išeiga informacinė. Vykdytojas3 atvaizduojamas ir procesų darbų sekų modelyje – „Siuvimas-P“, ir darbų sekų modelyje – „Siuvimas-F“ pagal trečią taisyklę kaip Vykdytojas3. Paskutinis veiklos procesas Atsiskaitymas su gamintoju kartu su įeiga Sąskaitos – faktūros1 atvaizduojamas funkcijų darbų sekų modelyje – „Siuvimas-F“ pagal pirmąją taisyklę nes jo įeiga informacinė. Vykdytojas4 atvaizduojamas pagal trečią taisyklę funkcijų darbų sekų modelyje – „Siuvimas-F“.

Atlikus tokį darbų sekų modelio „Siuvimas“ (VP\_WFM) informacijos išskyrimą į procesų darbų sekų modelį – „Siuvimas-P“ (P\_WFM) ir funkcijų darbų sekų modelį – „Siuvimas-F“ (F\_WFM) gaunamas toks rezultatas:



38 pav. Procesų darbų sekų modelį – „Siuvimas-P“



39 pav. Funkcijų darbų sekų modelį – „Siuvimas-F“

Grindžiant veiklos metamodeliu apibrėžiama, kad procesas turi tik materialią įeigą ir išeigą, o funkcija (arba viena iš jos dalių) gali turėti tik informacinę įeigą ir (arba) išeigą.

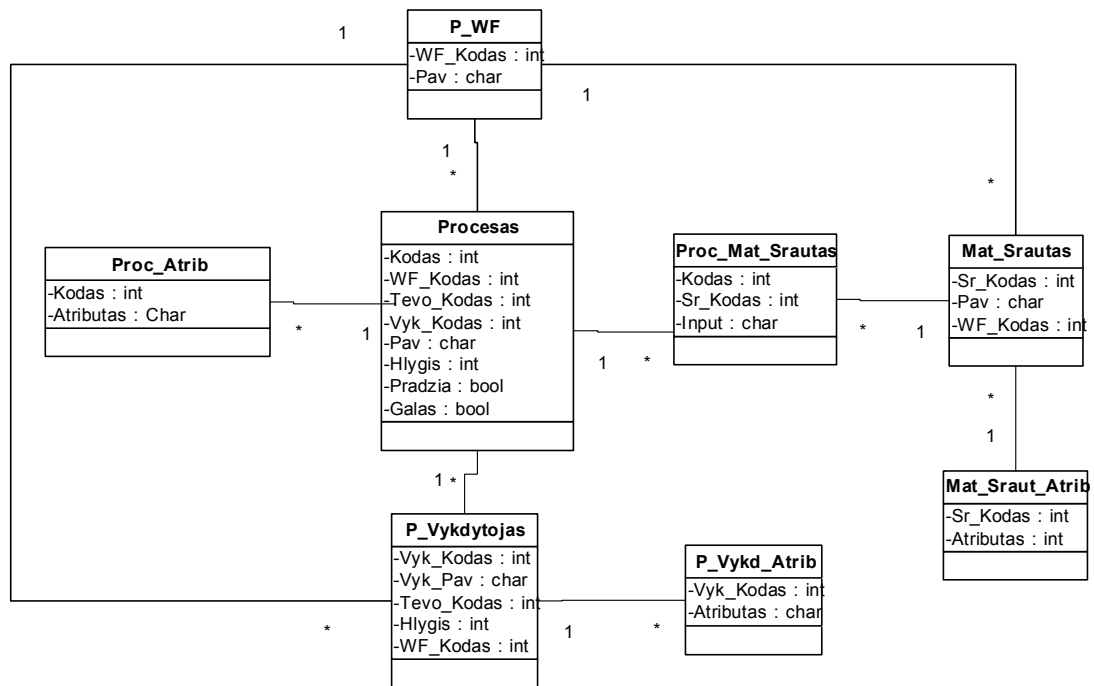
Kiekvienas procesas jei tik jis nėra pradinis arba paskutinis privalo turėti materialią įeigą ir išeigą. Analogiškas reikalavimas galioja ir veiklai, tik jos įeiga ir išeiga yra informacinė. Jeigu nėra tenkinama ši sąlyga, vadinasi funkcijų darbų sekų modelyje egzistuoja trūkio taškai, kurie pašalinami vykdant procesų arba funkcijų trūkio taškų šalinimo algoritmą.

Principinis VP\_WFM išskyrimo į F\_WFM ir P\_WFM algoritmas pateiktas 43 paveiksle.

#### 4.5.2. Tarpinės (P\_WF ir F\_WF) veiklos žinių bazės

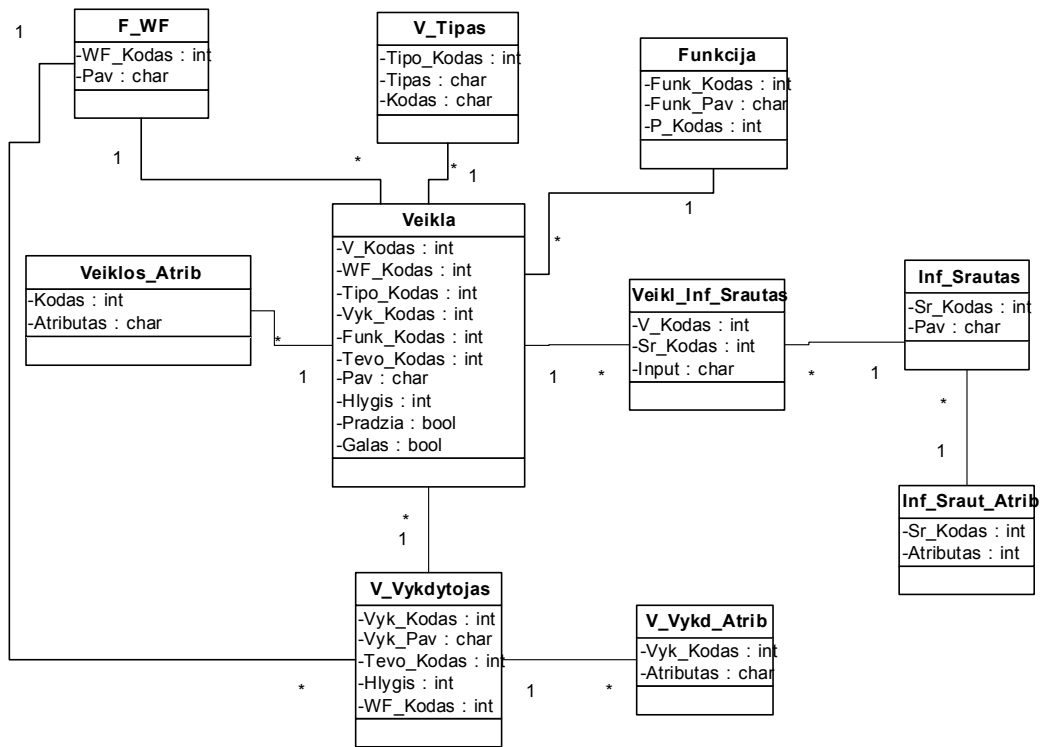
Antruoju etapu VP\_WFM esanti informacija yra išskiriama į dvi dalis: tai yra į P\_WFM ir F\_WFM. Atskirta informacija saugoma dvejose skirtingose duomenų bazėse tai yra procesų (P\_WFM) duomenų bazė (40 pav.) ir funkcijų (F\_WFM) duomenų bazė (41 pav.). Sekantis žingsnis - P\_WFM ir F\_WFM duomenų bazių sudarymas. P\_WFM bazę sudaro tokios lentelės: *P\_WF*, *Procesas*, *Proc\_Atrib*, *Proc\_Mat\_Srautas*, *Mat\_Srautas*, *Mat\_Sraut\_Atrib*, *P\_Vykdytojas* ir *P\_Vykd\_Atrib*. Lentelė *P\_WF* naudojama procesų modeliams saugoti. Lentelė *Procesas* nusako procesus. Su šia lentele susijusios dar trys lentelės: *Proc\_Atrib* - saugo procesų atributus, *Proc\_Mat\_Srautas* naudojama srautams kaupiti, susijusiems su procesais ir lentelė *P\_Vykdytojas* saugo procesų vykdytojus. *Mat\_Srautas* kaupia srautų duomenis, o lentelė *Mat\_Sraut\_Atrib* nusako srauto atributus.

Ir paskutinė lentelė *P\_Vykd\_Atrib* kaupia procesų vykdytojų atributus. Ši duomenų bazė naudojama procesų pradinei informacijai saugoti.



40 pav. *P\_WFM* duomenų bazė

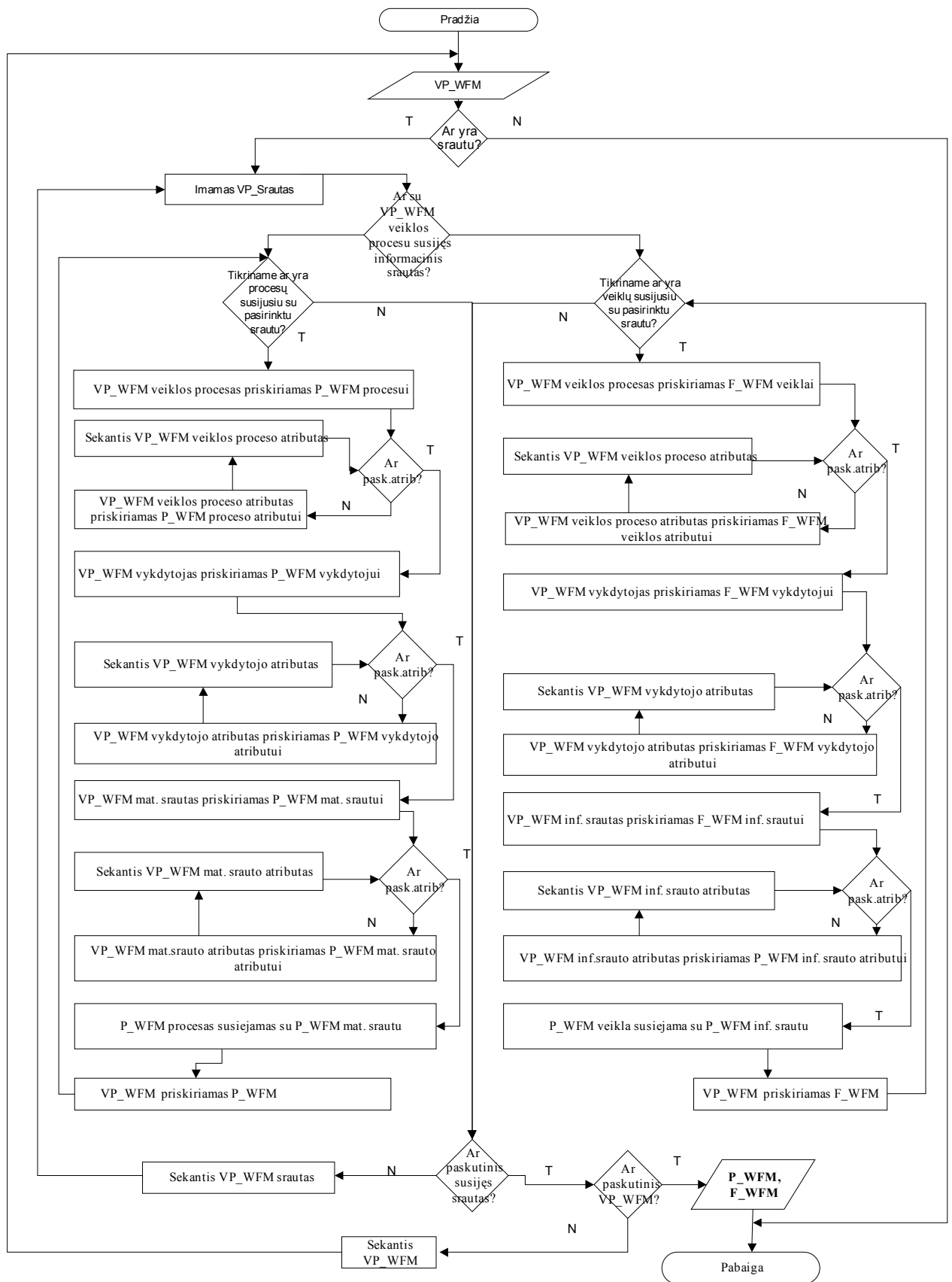
*F\_WFM* bazę sudaro tokios lentelės: *F\_WF*, *Veikla*, *V\_Tipai*, *Funkcija*, *Veiklos\_Atrib*, *Veikl\_Inf\_Srautas*, *Inf\_Srautas*, *Inf\_Sraut\_Atrib*, *V\_Vykdytojas* ir *V\_Vykd\_Atrib*. Lentelėje *F\_WF* saugomi veiklų modeliai. Lentelė *Veikla* nusako veiklas. Su šia lentele susijusios dar penkios lentelės: *V\_Tipai* – nurodo veiklos tipą (Interpretavimas, Informacijos apdorojimas ir sprendimų priėmimas (IP) ir Realizavimas), *Funkcija* – nurodo funkciją, kuriai priklauso veikla, *Veiklos\_Atrib* - saugo veiklų atributus, *Veikl\_Inf\_Srautas* saugo veiklų informacinius srautus, lentelė *V\_Vykdytojas* nusako veiklų vykdytojus. *Inf\_Srautas* saugo srauto duomenis, o lentelė *Inf\_Sraut\_Atrib* saugo srautų atributus. Ir paskutinė lentelė *V\_Vykd\_Atrib* saugo veiklų vykdytojų atributus. Ši duomenų bazė naudojama veiklų pradinei informacijai saugoti.



41 pav. F\_WFM duomenų bazė



### 4.5.3. Informacijos atskyrimo algoritmo schemas



42 pav. Informacijos atskyrimo algoritmas



#### 4.5.4. Informacijos atskyrimo algoritmo aprašymas

43 paveiksle pateikto algoritmo aprašymas:

- 1) Pradžia;
- 2) Imame veiklos procesų bazę (**VP\_WFM**);
- 3) Tikriname ar yra srautų (**VP\_WFM.VP\_Srautas**) veiklos procesų bazėje (**VP\_WFM**);
  - a. Jei srautų yra vykdomas 4 žingsnis;
  - b. Jei nėra vykdomas 41 žingsnis;
- 4) Pasirenkame srautą (**VP\_WFM.VP\_Srautas**);
- 5) Tikriname ar pasirinktas srautas yra materialus (**VP\_WFM.VP\_Srautas.VP\_Sraut\_Tipas="Materialus"**);
  - a. Jei pasirinktas srautas yra materialus, vykdomas 6 žingsnis;
  - b. Jei pasirinktas srautas nėra materialus vykdomas 21 žingsnis;
- 6) Tikriname ar yra naujų procesų (**VP\_WFM.VP\_Procesas**) susijusių su pasirinktu srautu (**VP\_WFM.VP\_Srautas**);
  - a. Jei egzistuoja naujas procesas, vykdomas 7 žingsnis;
  - b. Jei neegzistuoja vykdomas 36 žingsnis;
- 7) Įrašome procesą ir visus jo atributus į **P\_WFM** bazę (**P\_WFM.Procesas(P\_kodas, P\_WF\_Kodas, P\_Tevo\_Kodas, P\_Vykdyt\_Kodas, P\_Pav, P\_hlygis, Pradzia,Galas):= VP\_WFM.Procesas(VP\_kodas, VP\_WF\_Kodas, VP\_Tevo\_Kodas, VP\_Vykdyt\_Kodas, VP\_Pav, VP\_hlygis, Pradzia,Galas)**);
- 8) Tikriname ar įrašytas procesas (**P\_WFM.Procesas**) turi papildomų atributų;
  - a. Jei procesas turi atributų vykdomas 9 žingsnis;
  - b. Jei ne vykdomas 11 žingsnis;
- 9) Įrašome proceso papildomą atributą į **P\_WFM** bazę (**P\_WFM.Proc\_Atrib(P\_kodas, P\_atributas) := VP\_WFM.Proc\_Atrib(VP\_kodas, VP\_atributas)**);
- 10) Imamas naujas **VP\_WFM.V\_Procesas\_Atrib.VP\_Atributas** ir kartojamas 8 žingsnis;
- 11) Įrašome vykdytoją ir visus jo atributus į **P\_WFM** bazę (**P\_WFM.Vykdytojas(P\_vykdyt\_kodas, P\_WF\_Kodas, P\_Vykdyt\_Pav, P\_Vykdyt\_Tevo\_Kodas, P\_Vykdyt\_Hlygis):= VP\_WFM.Vykdytojas(VP\_vykdyt\_kodas, VP\_WF\_Kodas, VP\_Vykdyt\_Pav, VP\_Vykdyt\_Tevo\_Kodas, VP\_Vykdyt\_Hlygis)**);
- 12) Tikriname ar įrašytas vykdytojas (**P\_WFM.Vykdytojas**) turi papildomų atributų;
  - a. Jei vykdytojas turi atributų vykdomas 15 žingsnis;
  - b. Jei ne vykdomas 13 žingsnis;
- 13) Įrašome vykdytojo papildomą atributą į **P\_WFM** bazę

- P\_WFM.Vykd\_Atrib(P\_kodas, P\_atributas) := VP\_WFM.Vykd\_Atrib(VP\_kodas, VP\_atributas);**
- 14) Imamas naujas **VP\_WFM.V\_Vykd\_Atrib.VP\_Atributas** ir kartojamas 12 žingsnis;
- 15) Įrašome pasirinktą srautą ir visus jo atributus į P\_WFM bazę  
**(P\_WFM.Mat\_Srautas(P\_Sraut\_Kodas, P\_WF\_Kodas, P\_Sraut\_Pav) := VP\_WFM.Mat\_srautas(VP\_Sraut\_Kodas, VP\_WF\_Kodas, VP\_Sraut\_Pav));**
- 16) Tikriname ar įrašytas srautas (**P\_WFM.Mat\_Srautas**) turi papildomų atributų:
- Jei procesas turi atributų vykdomas 19 žingsnis;
  - Jei ne vykdomas 17 žingsnis;
- 17) Įrašome srauto papildomą atributą į P\_WFM bazę  
**P\_WFM.Mat\_Sraut\_Atrib(Mat\_Sraut\_Kodas, Mat\_Sraut\_Atrib\_Pav) := VP\_WFM.VP\_Sraut\_Atrib(VP\_Sraut\_Kodas, VP\_Sraut\_Atrib\_Pav);**
- 18) Imamas naujas **VP\_WFM.Mat\_Sraut\_Atrib.VP\_Atributas** ir kartojamas 16 žingsnis;
- 19) Susiejame procesas su pasirinktu srautu:  
**P\_WFM.Proc\_Mat\_Srautas(P\_Kodas, P\_Sraut\_Kodas, Input):= VP\_VFM.VP\_Proc\_Sraut(VP\_Kodas, VP\_Sraut\_Kodas, Input)**
- 20) Įrašome “workflow” ir visus jo atributus į P\_WFM bazę  
**(P\_WFM.P\_WF(P\_WF\_kodas, P\_WF\_Pav) := VP\_WFM.WFM(VP\_kodas, VP\_WF\_Kodas, VP\_Tevo\_Kodas, VP\_Vykdyt\_Kodas, VP\_Pav, VP\_hlygis, Pradzia,Galas));**
- 21) Tikriname ar yra naujų veiklų (**VP\_WMF.VP\_Veikla**) susijusių su pasirinktu srautu (**VP\_WMF.VP\_Srautas**):
- Jei egzistuoja nauja veikla, vykdomas 22 žingsnis;
  - Jei neegzistuoja vykdomas 36 žingsnis;
- 22) Įrašome veiklą ir visus jos atributus į F\_WFM bazę  
**(V\_WFM.Veikla(V\_kodas, V\_WF\_Kodas, V\_Tevo\_Kodas, V\_Vykdyt\_Kodas, V\_Pav, V\_hlygis, Pradzia,Galas):= VP\_WFM.Veikla(VP\_kodas, VP\_WF\_Kodas, VP\_Tevo\_Kodas, VP\_Vykdyt\_Kodas, VP\_Pav, VP\_hlygis, Pradzia,Galas));**
- 23) Tikriname ar įrašyta veikla (**P\_WFM.Veikla**) turi papildomų atributų:
- Jei veikla turi atributų vykdomas 24 žingsnis;
  - Jei ne vykdomas 26 žingsnis;
- 24) Įrašome veiklos papildomą atributą į P\_WFM bazę  
**P\_WFM.Veikla\_Atrib(V\_kodas, V\_atributas) := VP\_WFM.Veikla\_Atrib(VP\_kodas, VP\_atributas);**
- 25) Imamas naujas **VP\_WFM.V\_Veikla\_Atrib.VP\_Atributas** ir kartojamas 23 žingsnis;
- 26) Įrašome vykdytoją ir visus jo atributus į V\_WFM bazę

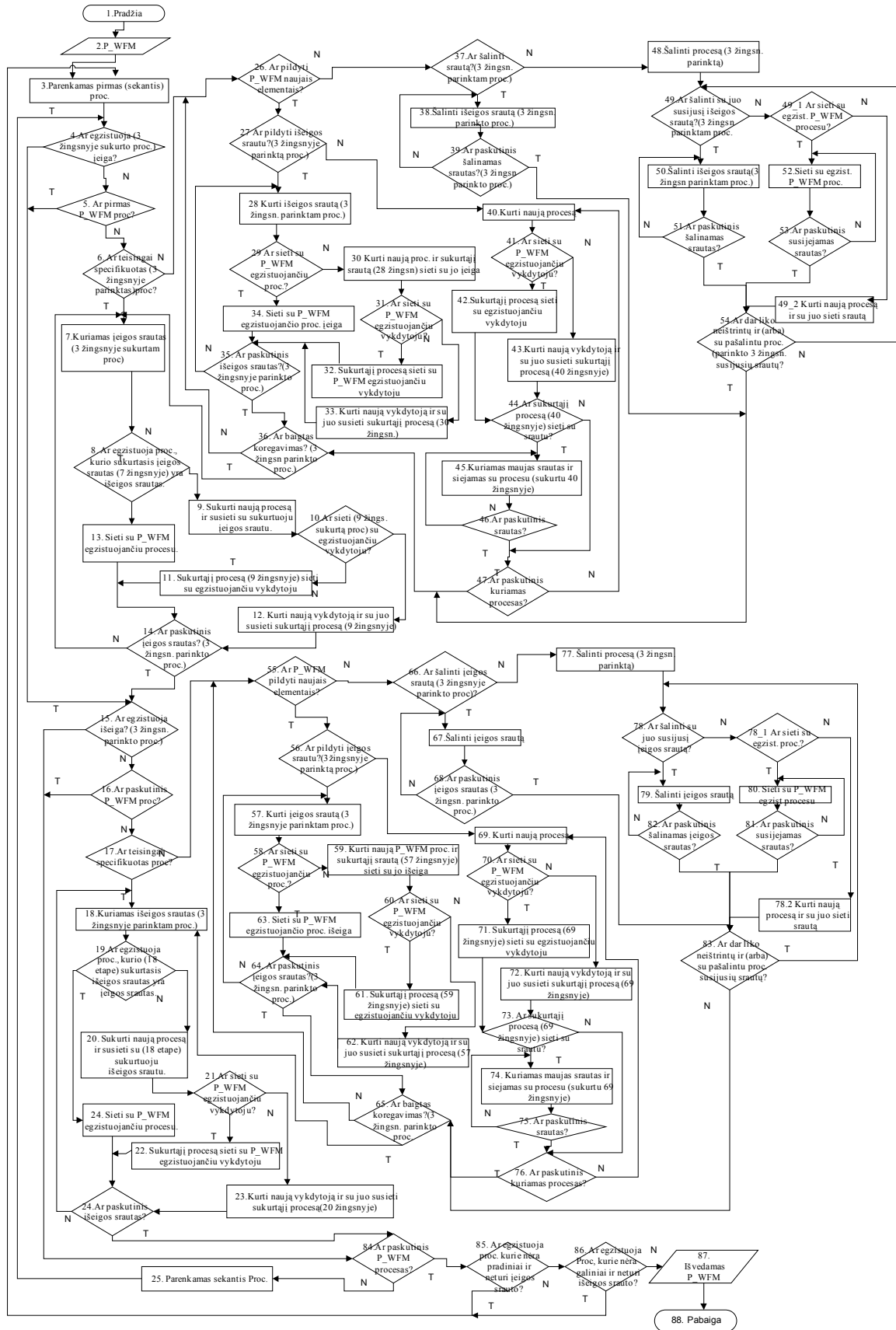
- (**V\_WFM.Vykdytojas(V\_vykdyt\_kodas, V\_WF\_Kodas, V\_Vykdyt\_Pav, V\_Vykdyt\_Tevo\_Kodas, V\_Vykdyt\_Hlygis):= VP\_WFM.Vykdytojas(VP\_vykdyt\_kodas, VP\_WF\_Kodas, VP\_Vykdyt\_Pav, VP\_Vykdyt\_Tevo\_Kodas, VP\_Vykdyt\_Hlygis);**)
- 27) Tikriname ar įrašytas vykdytojas (**V\_WFM.Vykdytojas**) turi papildomų atributų:
- Jei vykdytojas turi atributų vykdomas 28 žingsnis;
  - Jei ne vykdomas 30 žingsnis;
- 28) Įrašome vykdytojo papildomą atributą į V\_WFM bazę
- V\_WFM.Vykd\_Atrib(V\_kodas, V\_atributas) := VP\_WFM.Vykd\_Atrib(VP\_kodas, VP\_atributas);**
- 29) Imamas naujas **VP\_WFM.V\_Vykd\_Atrib.VP\_Atributas** ir kartojamas 27 žingsnis;
- 30) Įrašome pasirinktą srautą ir visus jo atributus į P\_WFM bazę
- (P\_WFM.Inf\_Srautas(Inf\_Sraut\_Kodas, Inf\_WF\_Kodas, Inf\_Sraut\_Pav) := VP\_WFM.Inf\_srautas(VP\_Sraut\_Kodas, VP\_WF\_Kodas, VP\_Sraut\_Pav));**
- 31) Tikriname ar įrašytas srautas (**V\_WFM.Mat\_Srautas**) turi papildomų atributų:
- Jei turi atributų vykdomas 32 žingsnis;
  - Jei ne vykdomas 34 žingsnis;
- 32) Įrašome srauto papildomą atributą į V\_WFM bazę
- V\_WFM.Mat\_Sraut\_Atrib(Inf\_Sraut\_Kodas, Inf\_Sraut\_Atrib\_Pav) := VP\_WFM.VP\_Sraut\_Atrib(VP\_Sraut\_Kodas, VP\_Sraut\_Atrib\_Pav);**
- 33) Imamas naujas **VP\_WFM.VP\_Sraut\_Atrib.VP\_Atributas** ir kartojamas 31 žingsnis;
- 34) Susiejamas informacinis srautas su veikla:
- F\_WFM.Veikl\_Inf\_Srautas(V\_Kodas, Inf\_Sraut\_Kodas, Input):= VP\_VFM.VP\_Proc\_Sraut(VP\_Kodas, VP\_Sraut\_Kodas, Input)**
- 35) Įrašome “workflow” ir visus jo atributus į V\_WFM bazę
- (V\_WFM.P\_WF(P\_WF\_kodas, V\_WF\_Pav) := VP\_WFM.WFM(VP\_kodas, VP\_WF\_Kodas, VP\_Tevo\_Kodas, VP\_Vykdyt\_Kodas, VP\_Pav, VP\_hlygis, Pradzia,Galas));**
- 36) Tikriname ar tai paskutinis srautas:
- Jei taip vykdomas 38 žingsnis;
  - Jei ne vykdomas 37 žingsnis;
- 37) Pasirenkame kitą srautą - **VP\_WFM.VP\_Srautas.VP\_Sraut\_Kodas** ;
- 38) Tikriname ar tai paskutinis **VP\_WFM**;
- Jei taip vykdomas 40 žingsnis;
  - Jei ne vykdomas 39 žingsnis;
- 39) Imamas naujas **VP\_WFM**;
- 40) Rezultatas – **P\_WFM** ir **F\_WFM** bazės; 41)Pabaiga.

## 4.6. Trūkio taškų šalinimas

### 4.6.1. Procesų ir funkcijų darbų sekų modelių trūkio taškų šalinimas

Išskaidžius VP\_WFM į F\_WFM ir P\_WFM yra didelė tikimybė, kad F\_WFM ir P\_WFM modeliuose egzistuoja trūkio taškai. Procesų darbų sekų modelio trūkio taškas, tai materialaus srauto, nutraukiančio nuoseklų procesų sekos vykdymą nebuvimas. Trūkio taško egzistavimas parodo, kad vartotojas neteisingai pateikė pradinę informaciją tai yra žinias apie kompiuterizuojamą dalykinę veiklos sritį. P\_WFM modelio trūkio taškams šalinti skirtas P\_WFM trūkio taškų aptikimo ir šalinimo algoritmas. O F\_WFM modelio trūkio taškams šalinti taikomas F\_WFM trūkio taškų aptikimo ir šalinimo algoritmas (analogiškas P\_WFM algoritmui). Trūkio taško šalinimas, nepriklausomai nuo jo šalinimo būdo, papildo procesų darbų sekų modelį realiai dalykinėse srityje egzistuojančiomis, tačiau klaidingai specifikuotomis arba visai nspecifikuotomis žiniomis (procesais, materialiais srautais ir vykdytojais). P\_WFM trūkio taškai aptinkami analizuojant kiekvieno materialaus proceso įeigos ir išeigos srautus. Jeigu materialus procesas nėra pradinis P\_WFM elementas, ir neturi materialaus įeigos srauto, vadinasi P\_WFM egzistuoja trūkio taškas. Jeigu materialus procesas nėra paskutinis P\_WFM elementas ir neturi materialaus išeigos srauto, vadinasi P\_WFM modelyje taip pat egzistuoja trūkio taškas. Prieš P\_WFM trūkio taškų šalinimo algoritme naikinant trūkio tašką, P\_WFM pildant reikiamais materialiais srautais ir (arba) procesais, tikrinama ar tas procesas teisingai specifikuotas. Aptikus trūkio tašką (neteisingai specifikuota proceso įeiga) nustatoma, kad procesas specifikuotas neteisingai.

### 4.6.2. Trūkio taškų šalinimo algoritmo schema



44 pav. Procesų trūkio taškų šalinimo algoritmas

## 4.7. Informacijos importavimas

### 4.7.1. Informacijos importavimas

Surinktos informacijos importavimo algoritmas apima sutvarkytos informacijos importavimą į veiklos žinių bazę (VP). Šį algoritmą dar galime vadinti veiklos žinių bazės formavimo algoritmu, nes būtent jis užpildo veiklos žinių bazę nagrinėjamos organizacijos duomenimis, kurie buvo surinkti ir patikslinti ankstesnių algoritmų pagalba. Informacijos importavimas į veiklos žinių bazę pradedamas nuo informacinių ir materialių srautų tvarkymo - pirmuoju žingsniu perkeliama informaciniai veiklų srautai. Sekančiu žingsniu perkeliama materialūs veiklos procesų srautai. Importavus procesų materialius ir veiklų informacinius srautus importuojame veiklų ir procesų vykdytojus. Kai veiklos žinių bazėje užpildėme organizacijos procesų ir veiklų materialiais ir informaciniais srautais ir vykdytojais, sekančiu žingsniu importuojame organizacijos veiklas ir procesus. Veiklos žinių bazėje jau importuota visa informacija, nusakanti organizacijos veiklos procesus ir funkcijas, tačiau dar nenurodytas ryšis, siejantis procesus ir veiklas su materialiais ir informaciniais srautais ir jų vykdytojais. Tad sekančiu žingsniu susiejame procesus su materialiais srautais ir atitinkamais procesų vykdytojais, ir taip pat veiklas su informaciniais srautais ir jų vykdytojais.

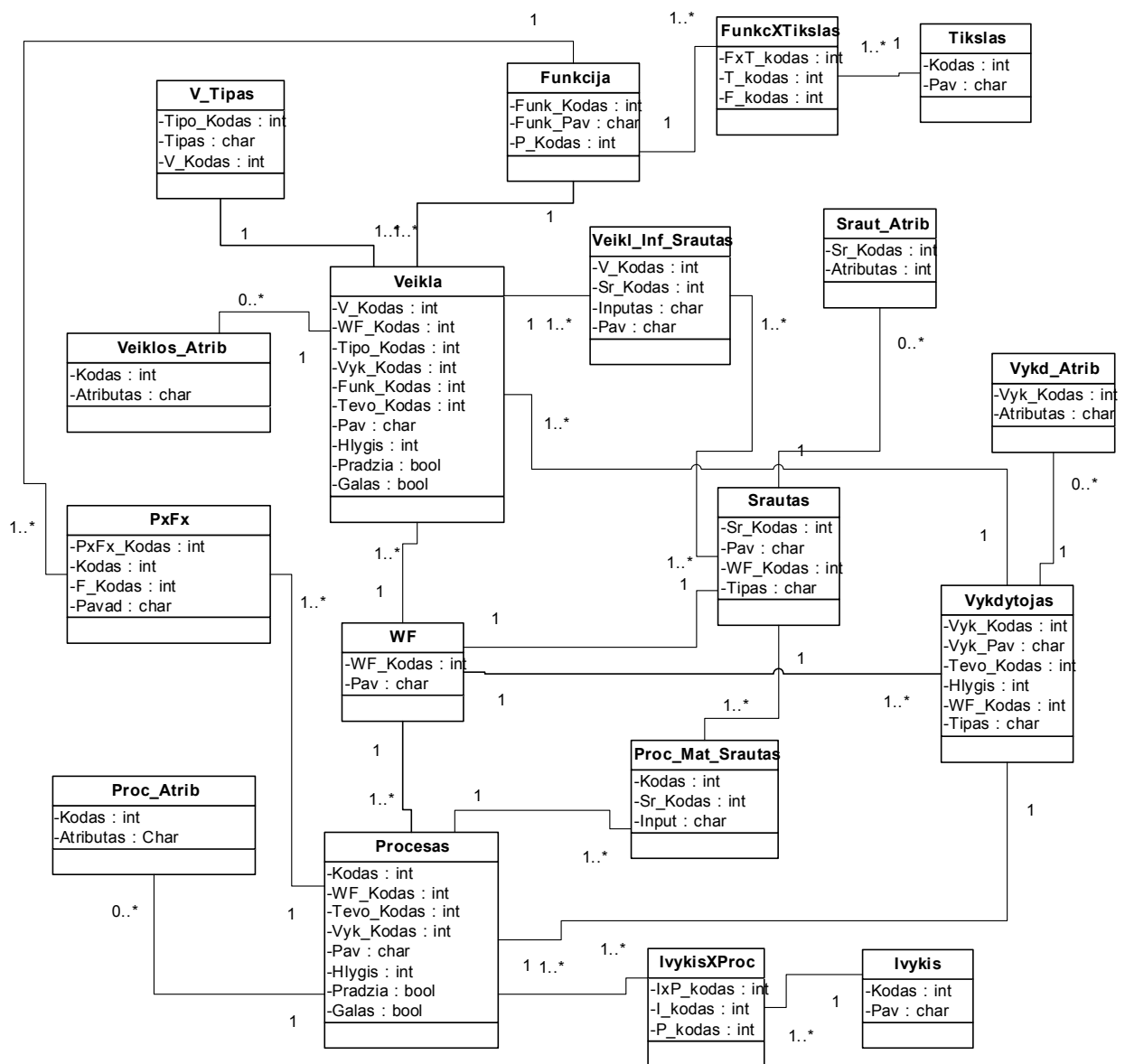
Principinis importavimo algoritmas pateiktas 47 paveiksle.

### 4.7.2. Veiklos žinių bazė

Veiklos žinių bazę sudaro aštuoniolika tarpusavyje susietų lentelių (45 pav.). Pagrindinės lentelės yra šios - *Procesas*, *Funkcija*, *PxFx* (proceso ir funkcijos sankirta), *Veikla*, *V\_Tipas*, *Vykdytojas*, *Srautas*.

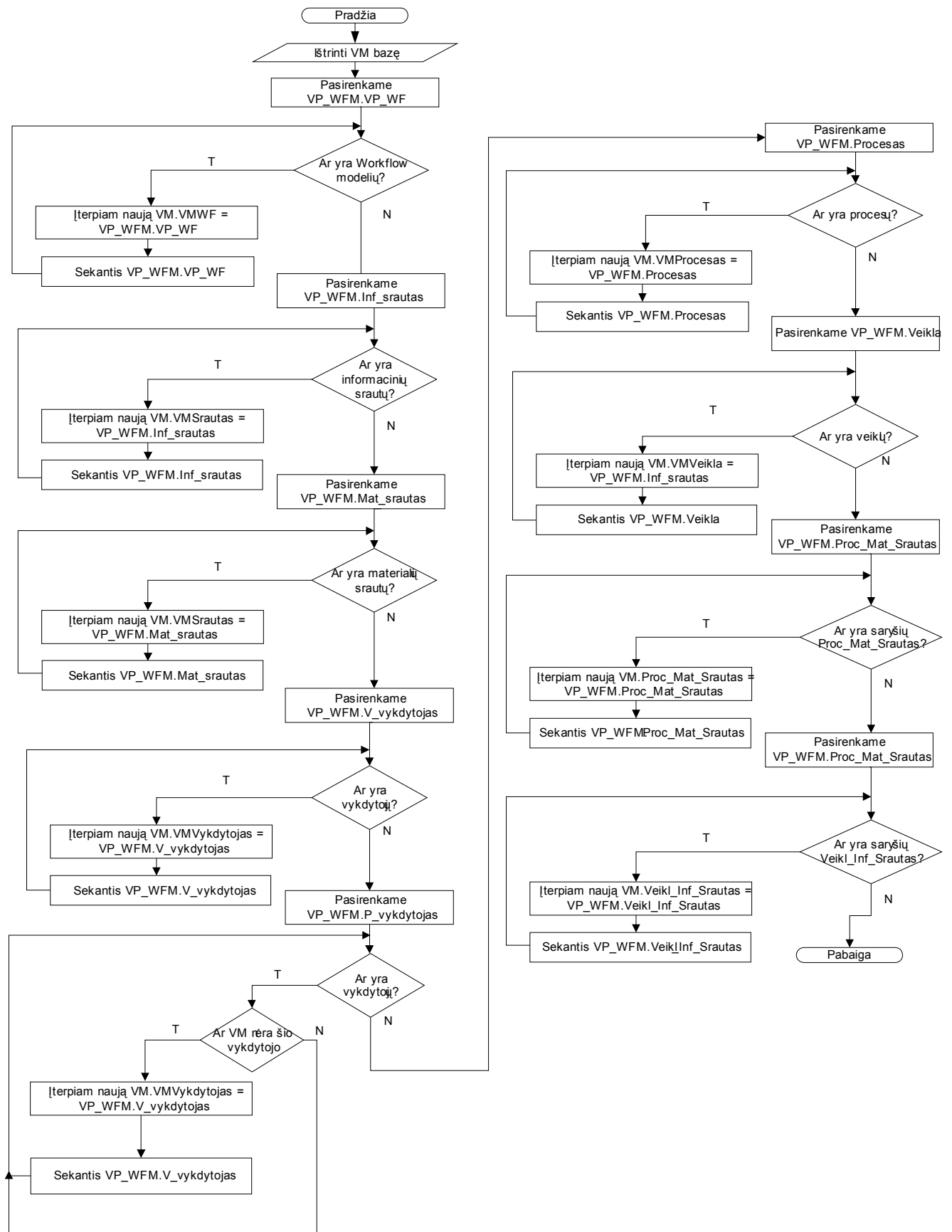
Lentelėje *Procesas* saugomos žinios apie organizacijoje vykstančius procesus, o lentelėje *Funkcija* – žinios apie egzistuojančias funkcijas. *PxFx* saugo procesų ir funkcijų sąsajos informaciją, t.y. kokius procesus valdo konkreti funkcija. Lentelėje *Veikla* saugoma informacija apie organizacijos veiklas. Sekanti lentelė *V\_Tipas* nurodo kokio tipo yra veiklos. Veiklos gali būti trijų tipų - Interpretavimo, Informacijos apdorojimo ir sprendimų priėmimo (IP) ir Realizavimo, kurios nustatomos pagal veiklos įeigos ir išeigos srautus [8]. Lentelė *Vykdytojas* kaupia informaciją apie organizacinę struktūrą: procesų ir funkcijų vykdytojus t.y. padalinius, cechus ar asmenis. Lentelėse *Srautas* saugoma informacija apie materialius ir informacinius kompiuterizuojamos dalykinės srities srautus. Procesai, procesų ir funkcijų sankirta, vykdytojai, materialūs ir informaciniai srautai gali turėti papildomus informacinius atributus. Jie saugomi atitinkamose lentelėse - *Proc\_Atrib*, *Veiklos\_Atrib*, *Vykd\_Atrib*, *Sraut\_Atrib* ir *Inf\_Sraut\_Atrib*.





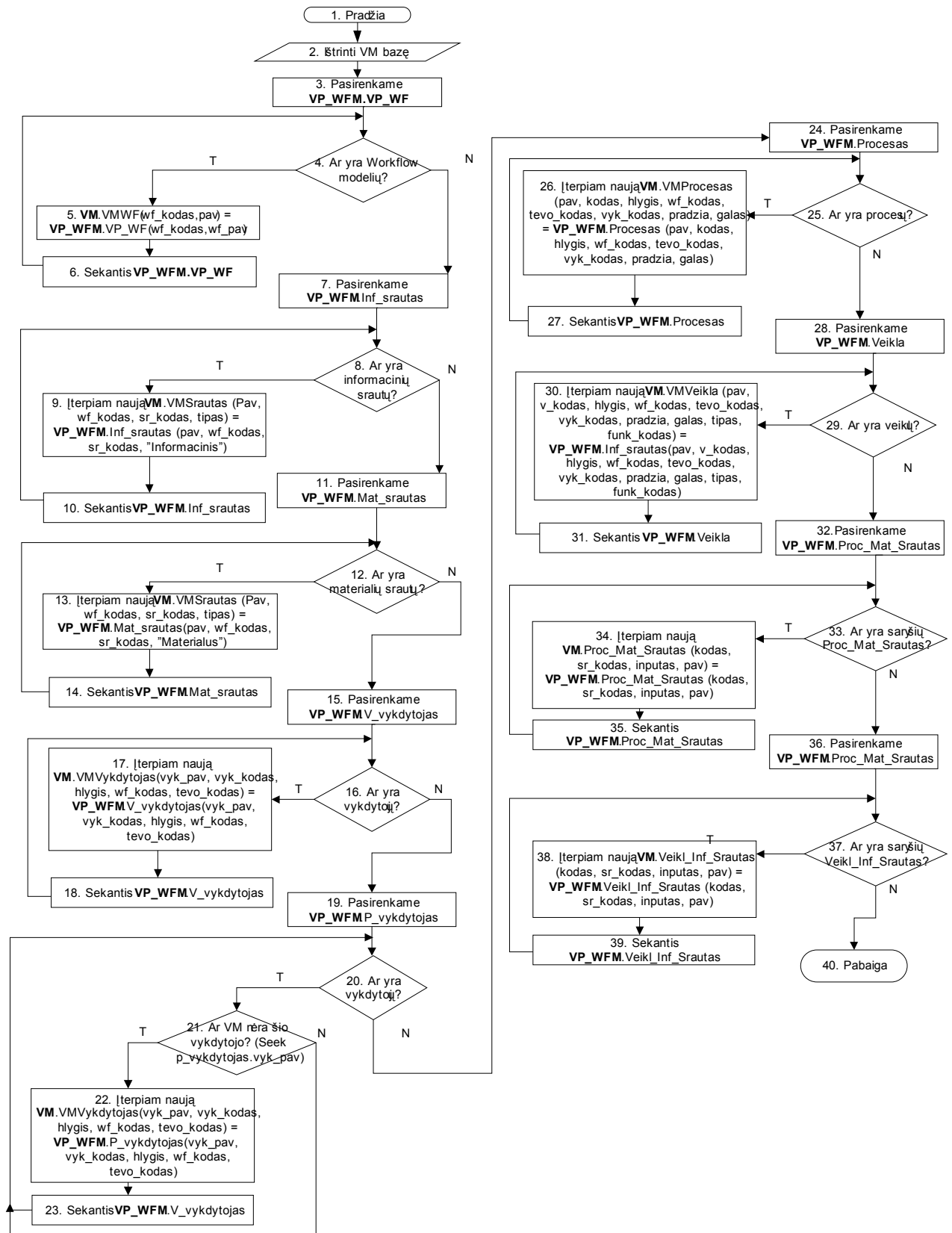
45 pav. Veiklos žinių bazė

### 4.7.3. Informacijos importavimo algoritmo schemas



46 pav. Informacijos importavimo algoritmas

## Detalus informacijos importavimo algoritmas:



47 pav. Detalus informacijos importavimo algoritmas

#### 4.7.4. Informacijos importavimo algoritmo aprašymas

47 –ame paveiksle pateikto informacijos importavimo algoritmo aprašymas:

- 1) Pradžia;
- 2) Ištriname VM bazėje egzistuojančius duomenis;
- 3) Pasirenkame **VP\_WFM.VP\_WF** lentelę;
- 4) Tikriname ar yra „*Workflow*” modelių?
  - a. Jei „*Workflow*” egzistuoja **VP\_WFM** bazėje vykdome 5 žingsnį;
  - b. Jei „*Workflow*” neegzistuoja vykdome 7 žingsnį;
- 5) Įrašome į VM bazę pasirinktą „*Workflow*” **VM.VM\_WF(wf\_kodas, pav) = VP\_WFM.VP\_WF(wf\_kodas, pav);**
- 6) Pasirenkame kitą **VP\_WFM.VP\_WF** „*Workflow*”;
- 7) Pasirenkame **VP\_WFM.Inf\_srautas** informacinį srautą;
- 8) Tikriname ar yra informacinių srautų?
  - a. Jei srautas egzistuoja **VP\_WFM** bazėje vykdome 9 žingsnį;
  - b. Jei srautas neegzistuoja vykdome 11 žingsnį;
- 9) Įrašome į VM bazę pasirinktą srautą **VM.VMSrautas (Pav, wf\_kodas, sr\_kodas, tipas) = VP\_WFM.Inf\_srautas (pav, wf\_kodas, sr\_kodas, "Informacinis");**
- 10) Pasirenkame kitą **VP\_WFM.Mat\_srautas** materialų srautą;
- 11) Pasirenkame **VP\_WFM.Mat\_srautas** materialų srautą;
- 12) Tikriname ar yra materialų srautų?
  - a. Jei srautas egzistuoja **VP\_WFM** bazėje vykdome 13 žingsnį;
  - b. Jei srautas neegzistuoja vykdome 15 žingsnį;
- 13) Įrašome į VM bazę pasirinktą srautą **VM.VMSrautas (Pav, wf\_kodas, sr\_kodas, tipas) = VP\_WFM.Mat\_srautas (pav, wf\_kodas, sr\_kodas, "Materialus");**
- 14) Pasirenkame kitą **VP\_WFM.Mat\_srautas** materialų srautą;
- 15) Pasirenkame **VP\_WFM.V\_vykdytojas** veiklos vykdytoją;
- 16) Tikriname ar yra vykdytojų?
  - a. Jei vykdytojas egzistuoja **VP\_WFM** bazėje vykdome 17 žingsnį;
  - b. Jei vykdytojas neegzistuoja vykdome 19 žingsnį;
- 17) Įrašome į VM bazę pasirinktą srautą **VM.VMvykdytojas(vyk\_pav, vyk\_kodas, hlygis, wf\_kodas, tevo\_kodas) = VP\_WFM.V\_vykdytojas (vyk\_pav, vyk\_kodas, hlygis, wf\_kodas, tevo\_kodas);**
- 18) Pasirenkame kitą **VP\_WFM.V\_vykdytojas** vykdytoją;
- 19) Pasirenkame **VP\_WFM.P\_vykdytojas** proceso vykdytoją;

- 20) Tikriname ar yra vykdytojų?
- Jei vykdytojas egzistuoja **VP\_WFM** bazėje vykdomė 21 žingsnį;
  - Jei vykdytojas neegzistuoja vykdomė 24 žingsnį;
- 21) Tikriname ar **VM** nėra šio vykdytojo? (Seek **P\_vykdytojas.vyk\_pav**)
- Jei vykdytojo nėra **VM** bazėje vykdomė 22 žingsnį;
  - Jei vykdytojas neegzistuoja vykdomė 20 žingsnį;
- 22) Įrašome į VM bazę pasirinktą vykdytoją **VM.VM\_Vykdytojas(vyk\_pav, vyk\_kodas, hlygis, wf\_kodas, tevo\_kodas) = VP\_WFM.P\_vykdytojas (vyk\_pav, vyk\_kodas, hlygis, wf\_kodas, tevo\_kodas)**;
- 23) Pasirenkame kitą **VP\_WFM.P\_vykdytojas** vykdytoją;
- 24) Pasirenkame **VP\_WFM.Procesas** procesą;
- 25) Tikriname ar yra procesų?
- Jei procesas egzistuoja **VP\_WFM** bazėje vykdomė 26 žingsnį;
  - Jei vykdytojas neegzistuoja vykdomė 28 žingsnį;
- 26) Įrašome į VM bazę pasirinktą procesą **VM.VM\_Procesas (pav, kodas, hlygis, wf\_kodas, tevo\_kodas, vyk\_kodas, pradzia, galas) = VP\_WFM.Procesas (pav, kodas, hlygis, wf\_kodas, tevo\_kodas, vyk\_kodas, pradzia, galas)**;
- 27) Pasirenkame kitą **VP\_WFM.VM\_Procesas** procesą;
- 28) Pasirenkame **VP\_WFM.Veikla** veiklą;
- 29) Tikriname ar yra veiklų?
- Jei veikla egzistuoja **VP\_WFM** bazėje vykdomė 30 žingsnį;
  - Jei vykdytojas neegzistuoja vykdomė 32 žingsnį;
- 30) Įrašome į VM bazę pasirinktą veiklą **VM.VM\_Veikla (pav, v\_kodas, hlygis, wf\_kodas, tevo\_kodas, vyk\_kodas, pradzia, galas, tipas, funk\_kodas) = VP\_WFM.Inf\_srautas(pav, v\_kodas, hlygis, wf\_kodas, tevo\_kodas, vyk\_kodas, pradzia, galas, tipas, funk\_kodas)**;
- 31) Pasirenkame kitą **VP\_WFM.VM\_Veikla** veiklą;
- 32) Pasirenkame **VP\_WFM.Proc\_Mat\_Srautas** ryšį;
- 33) Tikriname ar yra procesų ir materialių srautų ryšių?
- Jei ryšys egzistuoja vykdomė 34 žingsnį;
  - Jei ryšys neegzistuoja vykdomė 3š žingsnį;
- 34) Įrašome į VM bazę pasirinktą ryšį **VM.Proc\_Mat\_Srautas (kodas, sr\_kodas, inputas, pav) = VP\_WFM.Proc\_Mat\_Srautas (kodas, sr\_kodas, inputas, pav)**;
- 35) Pasirenkame kitą **VP\_WFM.Proc\_Mat\_Srautas** ryšį;
- 36) Pasirenkame **VP\_WFM.Veikl\_Inf\_Srautas** ryšį;

- 37) Tikriname ar yra veiklų ir informacinių srautų ryšių?
- Jei ryšys egzistuoja vykdome 38 žingsnį;
  - Jei ryšys neegzistuoja vykdome 40 žingsnį;
- 38) Įrašome į VM bazę pasirinktą ryšį **VM.Veikl\_Inf\_Srautas (kodas, sr\_kodas, inputas, pav) = VP\_WFM.Veikl\_Inf\_Srautas (kodas, sr\_kodas, inputas, pav);**
- 39) Pasirenkame kitą **VP\_WFM. Veikl\_Inf\_Srautas** ryšį;
- 40) Pabaiga.

## 4.8. Funkcijos sudėties nustatymas

### 4.8.1. Funkcijos sudėties nustatymas

Nustačius procesą valdančios funkcijos sudėtį, vykdomas dalykinės srities žinių surinkimo ir pilnumo tikrinimo algoritmo etapas, formuojantis funkcijos sudėties darbų sekų modelį. Šiame modelyje funkcijų darbų sekų modelio veiklos, informaciniai srautai ir vykdytojai vaizduojami kaip veiklos metamodelio sudėtinės dalys. Kiekvienas funkcijos sudėties darbų sekų modelis, atspindi dalykinėje srityje egzistuojančius informacinius srautus bei veiklas, priklausančias vienai konkrečiai funkcijai, bei valdančias vieną ar kelis materialius procesus. Kiekviena funkcijų darbų sekų modelyje egzistuojanti veikla gali būti vienos arba daugiau funkcijų sudėties darbų sekų modelių sudėtinė dalis. Kiekvieną procesą gali valdyti viena arba daugiau funkcijų, specifikuotų funkcijų sudėties darbų sekų modeliuose.

6 lentelėje pateikiamos funkcijos sudėtinės dalys, nustatomos pagal funkcijų darbų sekų modelyje egzistuojančių veiklų įeigos ir išeigos srautus. Pagal veiklos metamodelyje apibrėžtą funkcijos sudėtį egzistuoja trijų tipų veiklos:

- Interpretavimas;
- Informacijos apdorojimas ir sprendimų priėmimas (IP);
- Realizavimas.

6 lentelė. Funkcijų darbų sekų modelyje galimos veiklos įeigos ir išeigos kombinacijos

Veiklos išeigos tipas Veiklos įeigos tipas	Process Output (P_output)	IP Input	IP Output	Process Input
<b>1. Process Output</b>	Toks veiklos tipas negalimas	Veiklos tipas yra "Interpretation"	Veiklos sudėtyje egzistuoja "Interpretavimas" ir "IP"	Veiklos sudėtyje egzistuoja "Interpretavimas", "IP" ir "Realizavimas"
<b>2. IP Input</b>	Toks veiklos tipas negalimas	Toks veiklos tipas negalimas	Veiklos tipas yra "Interpretavimas"	Veiklos sudėtyje yra "IP" ir "Realizavimas"
<b>3. IP Output</b>	Toks veiklos tipas negalimas	Toks veiklos tipas negalimas	Toks veiklos tipas negalimas	Veiklos tipas yra "Realizavimas"
<b>4. Process Input</b>	Ta ctivity yra PROCESSas	Toks veiklos tipas negalimas	Toks veiklos tipas negalimas	Toks veiklos tipas negalimas

Jeigu veiklos ir įeiga, ir išeiga yra „*Process Output*“, vadinasi funkcijų darbų sekų modelyje aptikta negalima veikla, nes pagal veiklos metamodelio struktūrą, nei viena funkcijos sudėtinė dalis negali turėti to paties tipo įeigos ir išeigos srautų. Analogiškai negali egzistuoti veiklos, kurių įeigos ir išeigos (*IP Input*, *IP Output*, *Process Input*). Jeigu veiklos įeiga „*Process Output*“, o išeiga „*IP Input*“, pagal veiklos metamodelio struktūrą nustatoma, kad ši veikla yra funkcijos dalis- „Interpretavimas“. Interpretavimas- tai taisyklių rinkinys, skirtas transformuoti informacinį srautą „*Process Output*“ į duomenų apdorojimo ir sprendimų priėmimo funkcijos dalies „*IP*“ apdorojimui paruoštą pradinę įeigą „*IP Input*“. Interpretavimas yra būtina funkcijos sudėtinė dalis, nes „*Process Output*“ informacijos srautas, gali neatitikti duomenų apdorojimo ir sprendimų priėmimo funkcijos dalies įeigai „*IP Input*“ nustatyto duomenų formato. Jeigu veiklos įeiga yra „*IP Input*“, o išeiga „*IP Output*“, nustatoma, kad ši veikla yra funkcijos sudėtinė dalis „*IP*“. „*IP*“- funkcijos dalis, kurios pagrindinė paskirtis atlikti procesą valdančius informacijos apdorojimo ir sprendimo priėmimo veiksmus. Jeigu veiklos įeiga yra „*IP Output*“, o išeiga „*Process Input*“ nustatoma, kad ši veikla yra funkcijos dalis- „Realizavimas“. Realizavimas- tai funkcijos dalis atliekanti veiksmą atvirkštinį (priešingą) interpretavimui. Realizavimas „*IP*“ etape apdorotus „*IP Output*“ duomenis transformuoja į tiesioginiam proceso valdymui tinkamą formatą „*Process Input*“.

Galimi atvejai, kai pagal veiklos įeigos ir išeigos srautus, nustatoma, kad veikla funkcijų darbų sekų modelyje apibrėžia kelias funkcijos sudėtines dalis. Jeigu veiklos įeiga yra „*Process Output*“, o išeiga „*IP Output*“, vadinasi šios veiklos sudėtyje egzistuoja funkcijos sudėtinės dalys „*IP*“ ir „Interpretavimas“, bei juos jungiantis informacinis srautas „*IP Input*“. Jei veiklos įeiga „*Process Output*“, o išeiga „*Process Input*“, nustatoma, kad veiklos sudėtyje egzistuoja ne tik „Interpretavimas“, „*IP*“ ir „Realizavimas“, bet ir juos jungiantys informaciniai srautai „*IP Input*“ ir „*IP Output*“, o tai

reiškia, kad ši veikla yra funkcija. Egzistuojant veiklos įėjai „IP Input“, galimos dviejų tipų išeigos: „IP Output“ ir „Process Input“. Esant veiklos išėjai „Process Input“, veiklos sudėtyje nustatomas „Realizavimas“ ir „IP“, bei juos jungiantis informacinis srautas „IP Output“.

Esant veiklos įėjai „Process Input“, o išėjai „Process Output“, nustatoma, kad funkcijų darbų sekų modelyje egzistuoja klaida, nes ši veikla yra ne funkcija, ar jos dalis, o materialus procesas.

Galimos funkcijos sudėtinės dalys, nustatomos pagal veiklos įeigos ir išeigos srautus pateiktos 7 lentelėje.

7 lentelė. Galimos funkcijos sudėtinės dalys, nustatomos pagal veiklos įeigos ir išeigos srautus.

	Identifikuotas activity tipas	Trūkstanti funkcijos komponentai (pažymėti +)						
Activity Įeiga/Išeiga	Activity tipas	Process Output	Interpretavimas	IP Input	IP	IP Output	Realization	Process Input
Process Output/ IP Input	Interpretavimas				+	+	+	+
Process Output/ IP Output	Sudėtinė activity: IN + IP		+(tarpinis)	+(tarpinis)	+		+	+
Process Output/ Process Input	Funkcija		+(tarpinis)	+(tarpinis)	+(tarpinis)	+(tarpinis)	+(tarpinis)	
IP Input/ IP Output	IP	+	+				+	+
IP Input/ Process Input	Sudėtinė activity: IP + realisation	+	+		+(tarpinis)	+(tarpinis)	+(tarpinis)	
IP Output/ Process Input	Realizavimas	+	+	+	+			





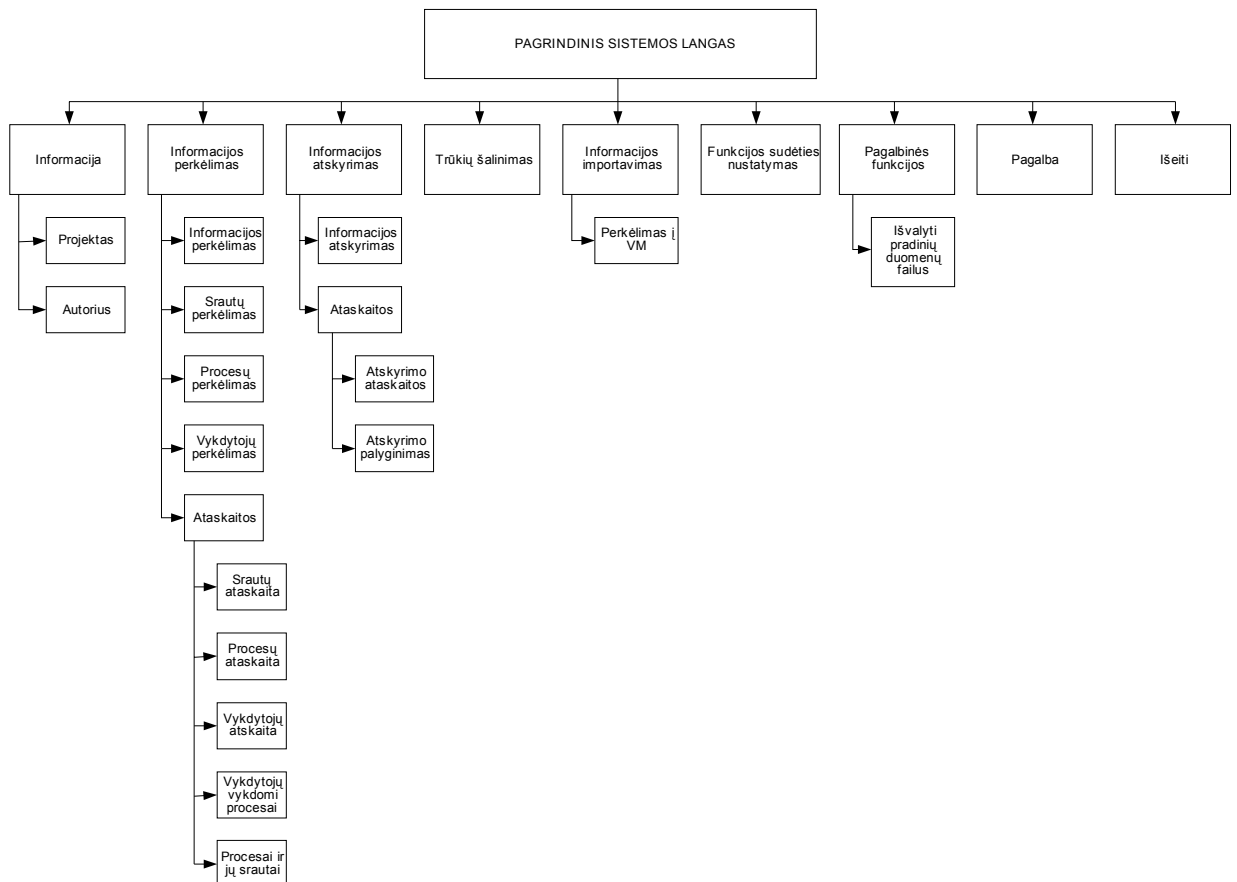
## 5. Eksperimentinė dalis

### 5.1. Vartotojo sąsajos projektas

Vartotojo ir sistemos sąsaja yra ženkli visos architektūros dalis. Vartotojo sąsaja turi būti patogi, lengvai naudojama bei suprantama. 49 – me paveiksle pateiktas vartotojo sąsajos modelis, nusakantis sistemos navigavimo planą. Pagrindinis meniu:

- ✓ Informacija;
- ✓ Informacijos perkėlimas;
- ✓ Informacijos atskyrimas;
- ✓ Trūkių šalinimas;
- ✓ Informacijos importavimas;
- ✓ Funkcijos sudėties nustatymas;
- ✓ Pagalbinės funkcijos;
- ✓ Pagalba;
- ✓ Išėiti.

Pasirinkus tam tikrą meniu punktą, atidaromas atitinkamas puslapis ar forma (49 pav.).



49 pav. Vartotojo sąsaja

## 5.2. Vartotojo dokumentacija

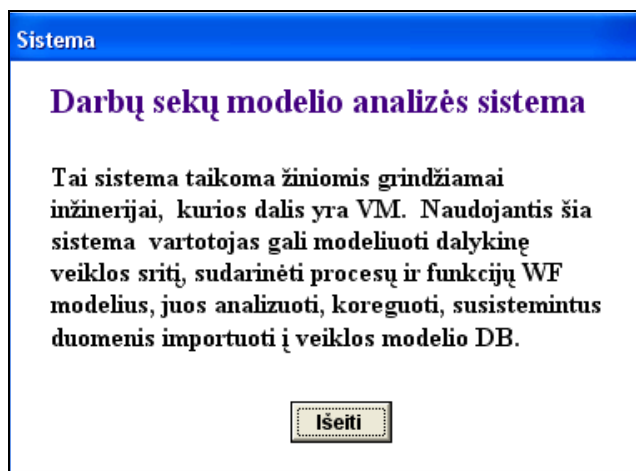
### 5.2.1. Sistemos funkcinis aprašymas

Sistemos galimybės:

- ✓ Informacija – vartotojas gali sužinoti bendrą informaciją apie sistemą ir jos autorių.
- ✓ Informacijos perkėlimas – atliekamas vartotojo pateiktos pradinės informacijos perkėlimas į tarpinę žinių bazę iš darbų sekų modelių. Informacijos perkėlimas gali būti atliekamas vienu žingsniu pasirenkant „ Informacijos perkėlimą“ ir trimis žingsniais, pasirenkant „ Srautų perkėlimą“, „Procesų perkėlimą“ ir „Vykdymo perkėlimą“. Vartotojas taip pat gali pasirinkti meniu punktą „Ataskaitos“, kuriame pateikiamos tokio tipo ataskaitos: srautų ataskaita, procesų ataskaita, vykdytojų ataskaita, vykdytojų vykdomi procesai ir procesai ir jų srautai.
- ✓ Informacijos atskyrimas – atliekamas perkeltos informacijos atskyrimas į procesų darbų sekų modelį (P\_WFM) ir funkcijų darbų sekų modelį (F\_WFM). Vartotojas taip pat gali pasirinkti meniu punktą „Ataskaitos“. Atskirtą informaciją vartotojas gali išvysti ataskaitų pavidale „ Atskyrimo ataskaitos“ arba sistemos lange „Atskyrimo palyginimas“.
- ✓ Trūkių šalinimas - išskyrus atitinkamus darbų sekų modelių komponentus į P\_WFM ir F\_WFM vartotojas gali atlikti analizę ar sistemoje neatsirado trūkio taškų, t.y. ar pilnai aprašyta specifikuota kompiuterizuojamoji dalykinė sritis. Jei taip tai vartotojas turi pašalinti atsiradusius trūkio taškus.
- ✓ Informacijos importavimas – atliekamas informacijos perkėlimas į veiklos žinių bazę.
- ✓ Funkcijos sudėties nustatymas - šiame etape vartotojas nustato funkcijų sudėtį.
- ✓ Pagalbinės funkcijos – šiame etape vartotojui pateikiama pagalbinė sistemos funkcija – „ Išvalyti pradinių duomenų failus“.
- ✓ Pagalba – pateikiama bendra informacija apie sistemą.
- ✓ Išėiti – meniu punktas, kuris vykdomas norint baigti darbą su sistema.

### 5.2.2. Informacija

Vartotojui pateikiama informacija apie sistemą (50 pav.) ir apie sistemos kūrėją (51 pav.):



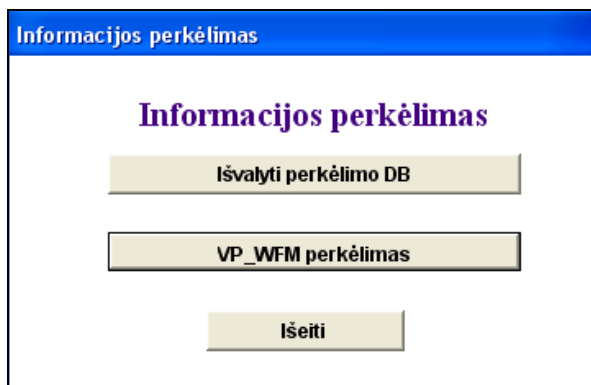
50 pav. Sistemos langas „Sistema“



51 pav. Sistemos langas „Autorius“

### 5.2.3. Informacijos perkėlimas

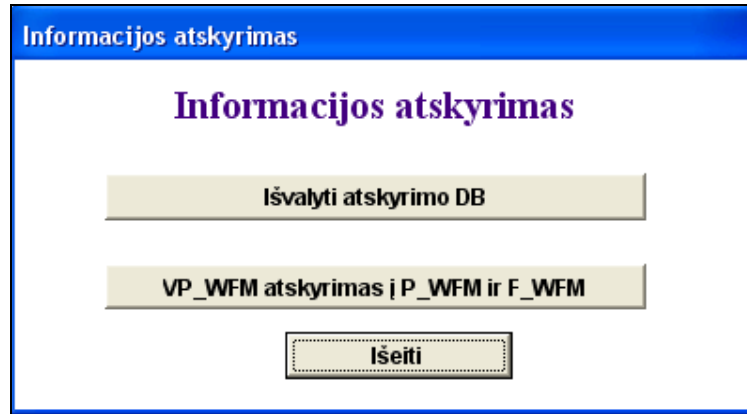
Vykdomas informacijos perkėlimas į tarpinę veiklos žinių bazę (52 pav.). Prieš perkėlimą reikia atlikti duomenų bazės išvalymą, pasirenkant mygtuką „Išvalyti perkėlimo DB“. Įvykdžius informacijos perkėlimą mygtuku „VP\_WFM perkėlimas“ sistemos langas uždaromas pasirenkant sekantį mygtuką „Išeiti“.



52 pav. Sistemos langas „Informacijos perkėlimas“

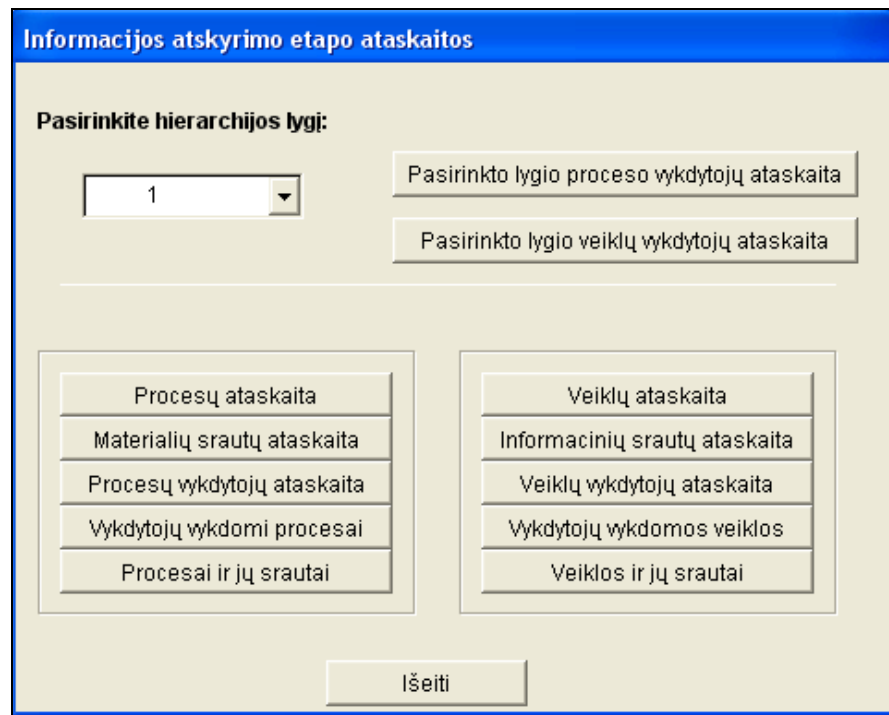
#### 5.2.4. Informacijos atskyrimas

Vykdomas perkeltos informacijos atskyrimas į procesų darbų sekų modelį (P\_WFM) ir funkcijų darbų sekų modelį (F\_WFM). Prieš pasirenkant mygtuką „VP\_WFM atskyrimas į P\_WFM ir F\_WFM“ taip pat reikia atlikti duomenų bazės išvalymą, pasirenkant mygtuką „Išvalyti atskyrimo DB“. Įvykdžius informacijos atskyrimą šis sistemos langas uždaromas pasirenkant sekantį mygtuką „Išeiti“.



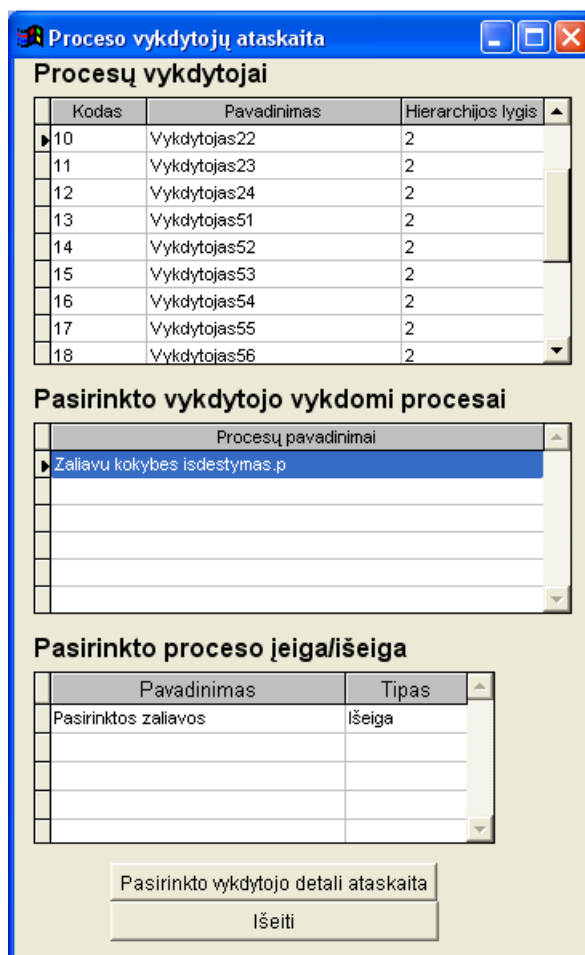
53 pav. Sistemos langas „Informacijos atskyrimas“

Pasirinkus ir įvykdžius mygtuką „VP\_WFM atskyrimas į P\_WFM ir F\_WFM“ sistema pateikia sekantį langą, kuriame pateikiamos galimos generuojamos ataskaitos (54 pav.).



54 pav. Sistemos langas „Informacijos atskyrimo etapo ataskaitos“

Norint generuoti ataskaitas visų pirma turime pasirinkti tą hierarchijos lygį, kurio ataskaitos bus generuojamos. Tada spaudžiame mygtuką „Pasirinkto lygio proceso vykdytojų ataskaita“ arba „Pasirinkto lygio veiklų vykdytojų ataskaita“. Pasirinkus vieną iš dviejų minėtų mygtukų, sistema pateikia sekantį langą - „Proceso vykdytojų ataskaita“ (55 pav.). Lentelėje „Procesų vykdytojai“ pasirinkus vykdytoją sekančiose dvejose lentelėse „Pasirinkto vykdytojo vykdomi procesai“ ir „Pasirinkto proceso įeiga/išeiga“ pateikiami pasirinkto vykdytojo visi vykdomi procesai, bei nurodomi tų procesų šrantai tai yra procesų įeiga ir procesų išeiga. Pasirinkus mygtuką „Pasirinkto vykdytojo detali ataskaita“ ta pati informacija pateikiama spausdinimui paruoštu formatu. Sistemos langas uždaromas mygtuku „Išeiti“. Sistemos lange „Informacijos atskyrimo etapo ataskaitos“ galima generuoti ne tik pasirinkto hierarchijos lygio ataskaitas, bet visų procesų, veiklų, materialių ir informacinių šrautų, procesų vykdytojų ataskaitas ir kitas pasirenkant atitinkamus mygtukus sistemos lange. Sistemos langas uždaromas mygtuku „Išeiti“.



55 pav. Sistemos langas „Proceso vykdytojo ataskaita“

Pasirinkus meniu: *Informacijos atskyrimas -> Ataskaitos -> Atskyrimo palyginimas* pateikiamas sistemos langas, susidedantis iš dviejų dalių : „Perkelta informacija“ (Puslapis 1) ir „Atskirta informacija“ (Puslapis 2). Pirmajame puslapyje (56 pav.) pateikiama informacija, kuri buvo

perkelta į tarpinę veiklos žinių bazę, o antrajame puslapyje pateikiama kaip ta informacija jau yra atskirta į procesus, jų vykdytojus ir materialius srautus bei veiklas, jų vykdytojus ir informacinius srautus (57 pav.).

Informacijos atskyrimas

### Perkelta informacija

Puslapis 1

Vykdytojai	
Pavadinimas	H lygis
Vykdytojas1	1
Vykdytojas2	1
Vykdytojas3	1
Vykdytojas4	1
Vykdytojas5	1
Vykdytojas6	1
Vykdytojas7	1
Vykdytojas8	1
Vykdytojas21	2
Vykdytojas22	2
Vykdytojas23	2
Vykdytojas24	2
Vykdytojas51	2
Vykdytojas52	2

Srautai	
Pavadinimas	Tipas
Kiekiai	Informacinis
Saskaitos fakturos1	Informacinis
Saskaitos-fakturos	Informacinis
Sertifikatai	Informacinis
Sutarties dokumentai	Informacinis
Uzsakymas	Informacinis
Uzsakymo blankai	Informacinis
Uzsakymo dokumentai	Informacinis
Uzsakymo dokumentai2	Informacinis
Vaztarasciai	Informacinis
Gaminio detales	Materialus
Gaminys	Materialus
Pagaminta produkcija	Materialus

### Atskirta informacija

Puslapis 2

Veiklos procesai	
Pavadinimas	H lygis
Atsiskaitymas su gamintoju	1
Atsiskaitymas uz zaliavas	1
Kokybes tikrinimas	1
Produkcijos gamyba	1
Produkcijos pakavimas	1
Produkcijos pristatymas uzsako	1
Produkcijos sandeliavimas	1
Uzsakymo patvirtinimas	1
Uzsakymo priemimas	1
Uzsakymo sutikrinimas	1
Zaliavu pristatymas	1
Zaliavu sandeliavimas	1
Zaliavu uzsakymas	1
Drabuzio siuvinimas	2
Drabuzio sutvarkymas	2
Medziagos ismatavimas	2
Medziagos sukirpimas	2
Produkto modelivimas	2
Sustikimas su tiekejumi	2
Uzsakymo sudarymas	2
Zaliavu gavimas	2
Zaliavu issirinkimas	2
Zaliavu kokybes isdestymas	2
Daliu susiuvimas	3
Sagu susiuvimas	3
Siulu parinkimas	3

Visos atskyrimo etapo atsakaitys

Išėiti

56 pav. Sistemos langas „Informacijos atskyrimas, Perkelta informacija“

Informacijos atskyrimas

### Perkelta informacija

Puslapis 1

Procesų vykdytojai	
Pavadinimas	H lygis
Vykdytojas22	2
Vykdytojas23	2
Vykdytojas24	2
Vykdytojas51	2
Vykdytojas52	2
Vykdytojas53	2
Vykdytojas54	2
Vykdytojas55	2
Vykdytojas56	2

Materialūs srautai	
Pavadinimas	
Gaminio detales	
Gaminys	
Pagaminta produkcija	
Pasirinktos zaliavos	
Patikrinta produkcija	
Reikiamos zaliavos	
Siulai	
Sukirpta medziaga	
Supakuota produkcija	
Susiuotos dalys	
Uzsakyta produkcija	
Uzsakytos zaliavos	
Visa produkcija	

### Atskirta informacija

Puslapis 2

Procesai	
Pavadinimas	H lygis
Zaliavu kokybes isdestymas.p	2
Medziagos ismatavimas	2
Zaliavu gavimas.p	2
Uzsakymo sudarymas.p	2
Produkto modelivimas.p	2
Siulu parinkimas	3
Sagu susiuvimas	3
Daliu susiuvimas	3

Veiklių vykdytojai	
Pavadinimas	H lygis
Vykdytojas1	1
Vykdytojas2	1
Vykdytojas3	1
Vykdytojas4	1
Vykdytojas6	1
Vykdytojas8	1
Vykdytojas21	2
Vykdytojas22	2
Vykdytojas23	2
Vykdytojas24	2
Vykdytojas51	2
Vykdytojas52	2

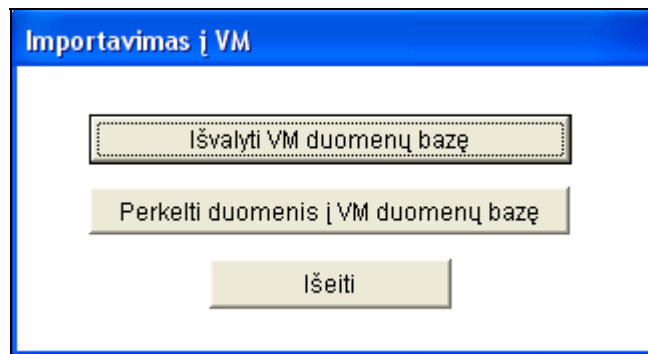
Veiklos	
Pavadinimas	H lygis
Atsiskaitymas su gamintoju	1
Produkcijos pristatymas uzsakov	1
Atsiskaitymas uz zaliavas.v	1
Zaliavu sandeliavimas.v	1
Zaliavu uzsakymas.v	1
Uzsakymo patvirtinimas	1
Uzsakymo priemimas	1
Uzsakymo sutikrinimas.v	1
Kokybes tikrinimas.v	1
Zaliavu gavimas.v	2
Produkto modelivimas.v	2
Zaliavu kokybes isdestymas.v	2
Sustikimas su tiekejumi	2
Uzsakymo sudarymas.v	2
Zaliavu issirinkimas.v	2

Informaciniai srautai	
Pavadinimas	
Kiekiai	
Saskaitos fakturos1	
Saskaitos-fakturos	
Sertifikatai	
Sutarties dokumentai	
Uzsakymas	
Uzsakymo blankai	
Uzsakymo dokumentai	
Uzsakymo dokumentai2	
Vaztarasciai	

57 pav. Sistemos langas „Informacijos atskyrimas, Atskirta informacija“

### 5.2.5. Informacijos importavimas

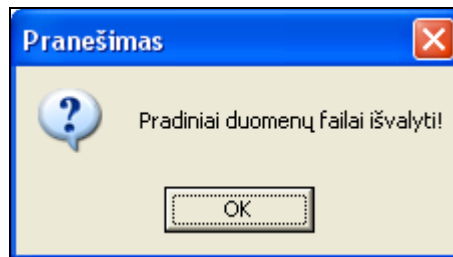
Vykdomas informacijos perkėlimas į veiklos žinių bazę. Prieš pasirenkant mygtuką „Perkelti duomenis į VM duomenų bazę“ reikia atlikti duomenų bazės išvalymą, pasirenkant mygtuką „Išvalyti VM duomenų bazę“. Įvykdžius informacijos perkėlimą į VM duomenų bazę sistemos langas uždaromas pasirenkant mygtuką „Išeiti“.



58 pav. Sistemos langas „Importavimas į VM“

### 5.2.6. Pagalbinės funkcijos

Šiame etape vartotojui pateikiama pagalbinė sistemos funkcija – „Išvalyti pradinių duomenų failus“. Įvykdžius pasirinktą funkciją sistema pateikia pranešimą (59 pav.).



59 pav. Pranešimas „Pradiniai duomenų failai išvalyti!“



## 6. Išvados

- ✓ Sukurtas ir realizuotas veiklos žinių kaupimo metodas, grindžiamas darbų sekų modelių transformavimo technologija, kuri leidžia aptikti nenuoseklumus (trūkius) vartotojo pateiktoje informacijoje
- ✓ Trūkliai veiklos procesų sekose ir valdymo funkcijų modeliuose identifikuojami formalių kriterijų atžvilgiu (formalizuoto veiklos modelio atžvilgiu).
- ✓ Skaidant veiklos procesų darbų sekų modelius į procesų ir funkcijų darbų sekų modelius, atsiskiria materialus veiklos domeną nuo informacinio ir kartu paaiškėja loginės spragos surinktoje informacijoje.
- ✓ Papildomai apklausiant analitiką šie trūkliai šalinami. Taip veiklos žinių bazės pagrindu sudaroma vartotojo pateiktos pirminės informacijos tikslinimo iteracija.
- ✓ Procesų ir funkcijų darbų sekų modeliai naudojami kaip papildomas žinių šaltinis (greta vartotojo), kuris leidžia patikslinti veiklos modelio sudėtį.
- ✓ Realizuojant veiklos modelio užpildymo algoritmus:
  - Sukurtas ir realizuotas surinktų žinių perkėlimo algoritmas;
  - Patikslinti procesų ir funkcijų skaidymo bei veiklos žinių bazės formavimo algoritmai;
  - Įrodytas šių algoritmų naudingumas, nes realizacijos metu gauti teigiami rezultatai modeliuojant konkrečią veiklos sritį.
- ✓ „Veiklos žinių kaupimo posistemis” sėkmingai patikrintas užpildant jį gamybinės įmonės „Linas” kontroliniais duomenimis.
- ✓ Šis veiklos žinių kaupimo posistemis yra žiniomis grindžiamos sistemos prototipas, kadangi modelių analizei panaudojamas veiklos metamodelis kaip iš anksto apibrėžta struktūra (kriterijus).

## 7. Literatūra

- [1] Bubenko J. Informatikon anglysis and conceptual modeling in Databases (Ed. By J.Predaens), Academic Press, London, 1987. - 139 -193 p.
- [2] Gudas S. Organizacijų veiklos modeliavimas. – Kaunas: Technologija , 2000. - 9-21p.
- [3] Boman M., Bubenko J., jr., Johansson P., Wangler Models, concepts, and informatikon. – Department of Computer and Systems Sciences, Royal Institute of Technology and Stockholm University, 1993. - 214 p.
- [4] Licker P.S.. Fundamentals of Systems Anglysis with Application Design. – Boston: Boyd&Fraser Publishing Company, 1987.- 681 p.
- [5] Lopata A. Veiklos modelių sudėties analizė // Informacinės technologijos 2002: konferencijos pranešimų medžiaga . Kaunas, Technologija, 2002.
- [6] Tham K.D. CIM-OSA: Enterprise modelling. –Enterprise integration laboratory, University of Toronto. [žiūrėta 2001-07-15]. Prieiga per internetą:  
<[www.ie.utoronto.ca/EIL/entmethod/cimos](http://www.ie.utoronto.ca/EIL/entmethod/cimos)>.
- [7] Universal Enterprise modeling language, IFAC-IFIP Task Force. [žiūrėta 2001-10- 30]. Prieiga per internetą:
- [8] < <http://www.cit.gu.edu.au/~bernus/taskforce/archive/UEML-TF-IG.ppt> >.
- [9] Lopata A., Gudas S. Modifikuotų darbų sekų modelių taikymas dalykinės srities žinioms specifikuoti // Informacinės technologijos 2003 : konferencijos pranešimų medžiaga. Kaunas, Technologija, 2003, p. 36-40 .
- [10] Lopata A., Gudas S. Informacijos išteklių identifikavimas, veiklos modelio pagrindu // Informacijos mokslai. Vilniaus universiteto leidykla, T19, Vilnius, 2001, p. 43-50.
- [11] Gudas S., Skersys T., Lopata A. Domain Knowledge Integration For Information Systems Engineering // Informacinės technologijos verslui 2002. Kaunas. 2002.
- [12] Robertson J., Volere Requirements Specification Template.  
Prieiga per internetą:< <http://www.systemsguild.com/GuildSite/Robs/Template.html>>.
- [13] ProVision “Work Bench” User Manual Guide. Prieiga per internetą:  
<[www.proformacorp.com](http://www.proformacorp.com)>.
- [14] ENV 12204: Advanced Manufacturing Technology - Systems Architecture - Constructs for Enterprise Modelling, CEN TC 310/WG1, 1996.

- Prieiga per internetą: < [http://www.pera.net/Standards/ENV\\_12204.html](http://www.pera.net/Standards/ENV_12204.html) >.
- [15] Kradolfer M. „A Work Flow Metamodel Supporting Dynamic, Reuse- Based Model Evolution" Zurich, 2000. [žiūrėta 2003-04-20]. Prieiga per internetą: <[www.ifi.unizh.ch/ifiadmin/staff/rofrei/Dissertationen/Jahr\\_2000/thesis\\_kradolfer.pdf](http://www.ifi.unizh.ch/ifiadmin/staff/rofrei/Dissertationen/Jahr_2000/thesis_kradolfer.pdf)>.
- [16] Lopata A., Gudas S. Veiklos funkcijų darbų sekų modelis // Informacinės Technologijos verslui 2003: konferencijos pranešimų medžiaga. Kaunas, 2003, p.45-48.
- [17] Budrevičienė S., Lopata A., Gudas S. Veiklos žinių kaupimo posistemis // Informacinės technologijos 2004: konferencijos pranešimų medžiaga. Kaunas, Technologija, ISBN 9955-09-588-1.2004, p.581-584.
- [18] Appleton, Brad. Welcome to Brad Appleton's Documents. [žiūrėta 2002 10 01] Prieiga per internetą: < <http://www.enteract.com/~bradapp/> >.

# Priedas

## VEIKLOS ŽINIŲ KAUPIMO POSISTEMIS

**Sandra Budrevičienė, Audrius Lopata, Saulius Gudas**

*Informacijos sistemų katedra, Kauno technologijos universitetas*

*Studentų 50-309, LT-3031 Kaunas*

*Straipsnyje aprašyti IS inžinerijai skirto veiklos modelio užpildymo konkrečios srities žiniomis technologijos pagrindiniai etapai. Pateikiami veiklos žinių kaupimo posistemo programinės įrangos vartotojo sąsajos pavyzdžiai, aptariamoms šio programinės įrangos prototipo vystymo galimybės.*

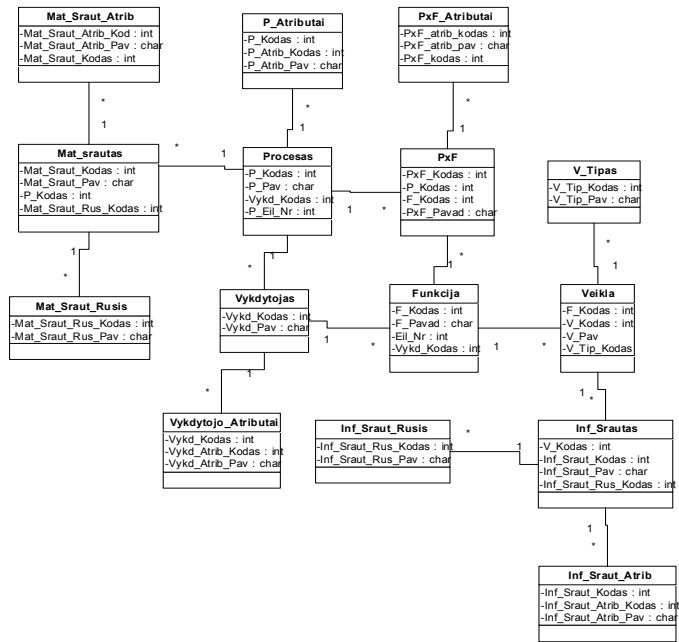
### **Įvadas**

Informacijos sistemų inžinerijos gyvavimo ciklas pradedamas veiklos dalykinės srities analize, vartotojo poreikių analize ir informacinių poreikių specifikacijos sudarymu. Mūsų požiūriu veiklos analizės metu surinkta informacija turi būti sutvarkoma kaip žinių struktūra, leidžianti generuoti IS projektinius modelius (UCM, kaip vartotojo poreikių specifikaciją, klasių modelį ir kt.). Tokią žinių struktūrą vadinsime *veiklos modeliu* (angl. *Enterprise model*) [1,3,4,5]. Veiklos modelis detaliam nagrinėjamas [2]. Straipsnio tikslas yra apžvelgti veiklos modelio užpildymo konkrečios srities žiniomis pagrindinius etapus ir pateikti sukurtos programinės realizacijos aprašymą.

### **Veiklos modelio loginė DB schema**

Veiklos modelio duomenų bazės loginę schemą sudaro penkiolika tarpusavyje susietų lentelių. Pagrindinės lentelės yra *Procesas*, *Funkcija*, *PxF* (proceso ir funkcijos sankirta), *Vykdytojas*, *Mat\_srautas*, *Inf\_srautas*.

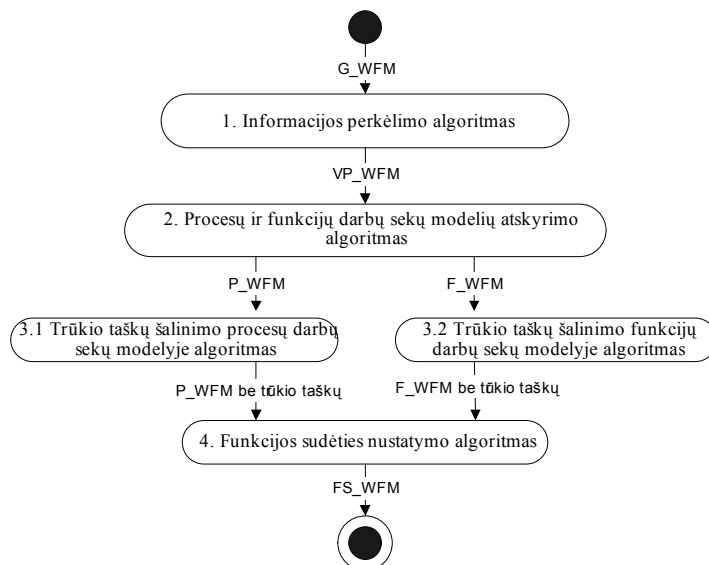
Lentelėje *Procesas* saugomos žinios apie materialius dalykinėje veiklos srityje vykstančius procesus, o lentelėje *Funkcija* – žinios apie dalykinėje veiklos srityje vykstančias funkcijas. *PxF* saugo procesų ir funkcijų sąsajos informaciją, t.y. kokius procesus valdo konkreti funkcija. Lentelėje *Vykdytojas* saugoma informacija apie organizacinę struktūrą: procesų ir funkcijų vykdytojus t.y. padalinius, cechus ar asmenis. Lentelėse *Mat\_srautas* ir *Inf\_srautas* saugoma informacija apie materialius ir informacinius kompiuterizuojamos dalykinės srities srautus srautus. Procesai, procesų ir funkcijų sankirta, vykdytojai, materialūs ir informaciniai srautai gali turėti papildomus informacinius atributus. Jie saugomi atitinkamose lentelėse *P\_Atributai*, *PxF\_Atributai*, *Vykdytojo\_Atributai*, *Mat\_Sraut\_Atrib* ir *Inf\_Sraut\_Atrib*. Materialaus ar informacinio srauto rūšis nurodoma *Mat\_Sraut\_Rusis* ir *Inf\_Sraut\_Rusis* lentelėse. Funkciją gali sudaryti daug veiklų, sudarančių veiklų hierarchija saugoma lentelėje *Veikla*, o veiklų tipai lentelėje *V\_Tipas*.



1 pav. Veiklos Modelio loginė duomenų bazės schema

### Veiklos modelio užpildymo etapai

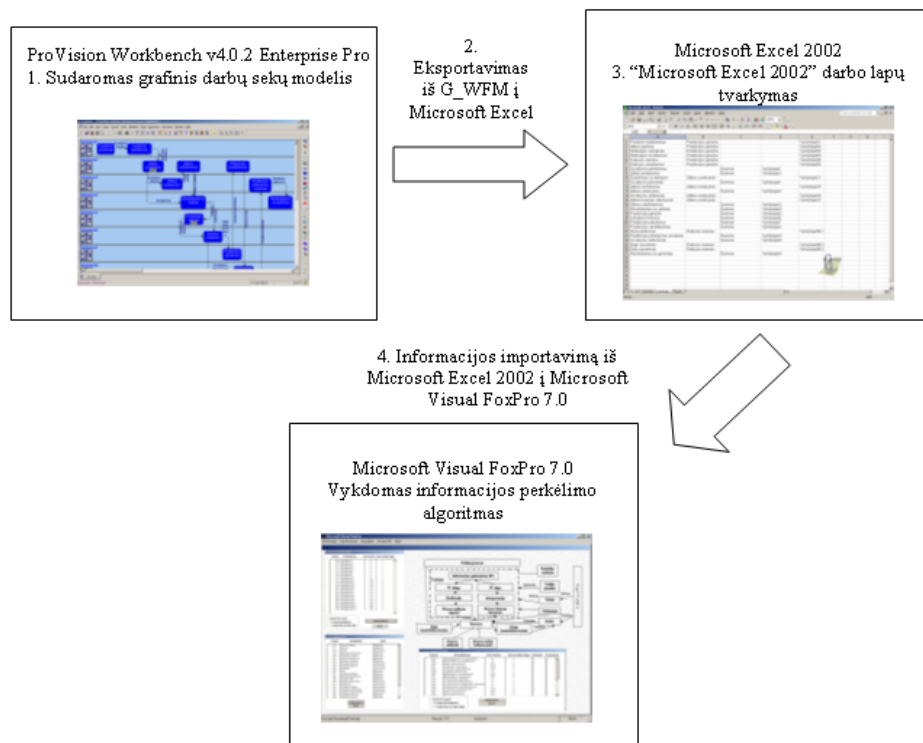
Dalykinės srities žinių surinkimo etape taikomi šešių tipų modifikuoti darbų sekų modeliai: veiklos procesų darbų sekų modelis (VP\_WFM), procesų darbų sekų modelis (P\_WFM), funkcijų darbų sekų modelis (F\_WFM), procesų darbų sekų modelis be trūkio taškų, funkcijų darbų sekų modelis be trūkio taškų ir funkcijos sudėties darbų sekų modelis (FS\_WFM). Šių modelių kūrimui ir žinių transformavimui į veiklos modelį taikomi penkių tipų algoritmai: informacijos perkėlimo algoritmas, procesų ir funkcijų darbų sekų modelių atskyrimo algoritmas, trūkio taškų šalinimo procesų darbų sekų modelyje algoritmas, trūkio taškų šalinimo funkcijų darbų sekų modelyje algoritmas, funkcijos sudėties nustatymo algoritmas ir žinių transformavimo į veiklos modelį algoritmas.



2 pav. Informacijos tvarkymo algoritmas

## Pirmasis etapas – Informacijos perkėlimo algoritmas

Pirmasis darbo etapas susideda iš penkių žingsnių. Pirmuoju žingsniu paketu „ProVision Workbench v4.0.2 Enterprise Pro“ sudaromas grafinis darbų sekų modelis (G\_WFM), specifikuojantis kompiuterizuojamą dalykinę veiklos sritį. VP\_WFM skirtas nagrinėjamo veiklos proceso eigai aprašyti. Kai jau darbų sekų modelis - G\_WFM -sudarytas, galime atlikti sekantį pirmojo etapo žingsnį tai yra - eksportavimą į „Microsoft Excel 2002“ paketą. Šis paketas naudojamas kaip tarpinis informacijos perkėlimui iš „ProVision Workbench v 4.0“ į „Microsoft Visual FoxPro 7.0“. Šis paketas pasirinktas todėl, kad grafinio modeliavimo būdas leidžia informacijos eksportavimą į šį ir dar į keletą paketų. Apsispręsta pasirinkti „Microsoft Excel 2002“, nes būtent iš šio paketo įmanomas informacijos importavimas į „Microsoft Visual FoxPro 7.0“, kurio pagalba realizuojame VM valdymo informacinę sistemą. Taigi antru žingsniu atliekame - *Informacijos eksportavimą iš G\_WFM į „Microsoft Excel 2002“* darbo lapus. Atlikus minėtą žingsnį gautus „Microsoft Excel 2002“ darbo lapus reikia pasiruošti sekantiems žingsniams t.y. trečiasis pirmojo etapo žingsnis – „Microsoft Excel 2002“ darbo lapų tvarkymas (panaikiname juose esančius nereikalingus laukus). Šis žingsnis atliekamas todėl, kad dalis laukų minėtuose darbų lapuose yra visiškai tušti. Atlikus trečiąjį žingsnį galime vykdyti ketvirtą - *Informacijos importavimą iš „Microsoft Excel 2002“ į „Microsoft Visual FoxPro 7.0“*, nes visas darbas bus atliekamas būtent šiuo paketu. Importavus duomenis į „Microsoft Visual FoxPro 7.0“ DB, gauname dvi nepriklausomas DB lenteles, kuriose ir yra patalpinta visa reikiama informacija. Minėtu paketu taip pat vykdomas ir *Informacijos perkėlimo algoritmas*.



3pav. Informacijos tvarkymo algoritmo pirmasis etapas - informacijos perkėlimo algoritmas

### **Antrasis etapas – Procesų ir funkcijų darbų sekų modelių atskyrimo algoritmas**

Antruoju etapu VP\_WFM esanti informacija yra išskiriama į dvi dalis: tai yra į P\_WFM ir F\_WFM. Šis etapas realizuojamas analizuojant VP\_WFM veiklos procesų įeigos ir išeigos srautus. Jei veiklos proceso įeigos ir (arba) išeigos srautas yra materialus tai procesas, kartu su materialiais įeigos ir išeigos srautais P\_WFM atvaizduojamas kaip materialus procesas, o jei įeigos ir (arba) išeigos srautas yra informacinis – veiklos procesas, kartu su juo susijusiais informaciniais srautais, F\_WFM atvaizduojamas kaip informacinė veikla. Taip visi veiklos procesai yra suskirstomi į dvi grupes ir šis jų padalinimas realizuojamas „Microsoft Visual Studio 7.0“ pakete P\_WFM ir F\_WFM atskyrimo algoritmo pagalba.

### **Trečiasis etapas – Trūkio taškų šalinimo procesų ir funkcijų darbų sekų modelyje algoritmas**

Išskyrus atitinkamus VP\_WFM komponentus į P\_WFM ir F\_WFM reikia ištirti ar organizacijos veiklos modeliuose, sudarytuose pagal analizės metu surinktą (vartotojo pateiktą) informaciją, nėra trūkio taškų, t.y. ar pilnai aprašyta specifikuota kompiuterizuojamoji dalykinė sritis. Trūkio taškas- tai nuosekliai vykdomų procesų ar veiklų grandinės neegzistavimas, t.y. informacinių ar materialių srautų tarp procesų ar veiklų, būtinų informacijos ar materialių išteklių aprūpinimui trūkumas. P\_WFM trūkio taškams šalinti taikomas procesų darbų sekų modelio trūkio taškų šalinimo algoritmas. F\_WFM trūkio taškams šalinti taikomas funkcijų darbų sekų modelio trūkio taškų šalinimo algoritmas. Tai analogiški algoritmai, tik taikomi skirtingų tipų darbų sekų modeliams. P\_WFM ir F\_WFM trūkio taškų šalinimo algoritmų rezultatas- P\_WFM be trūkio taškų ir F\_WFM be trūkio taškų.

### **Ketvirtas etapas – Funkcijos sudėties nustatymo algoritmas**

Šiame etape nustatoma funkcijų sudėtis. Pagal veiklos metamodelyje apibrėžtą funkcijos sudėtį egzistuoja trijų tipų veiklos – *Interpretavimas*, *Informacijos apdorojimas ir sprendimų priėmimas (IP)* ir *Realizavimas*, kurios nustatomos pagal veiklos įeigos ir išeigos srautus. Jeigu veiklos įeiga “*Process Output*”, o išeiga “*IP Input*”, pagal veiklos metamodelio struktūrą nustatoma, kad ši veikla yra funkcijos dalis - “*Interpretavimas*”. Jeigu veiklos įeiga yra “*IP Input*”, o išeiga “*IP Output*”, nustatoma, kad ši veikla yra funkcijos sudėtinė dalis “*Informacijos apdorojimas ir sprendimų priėmimas*” (IP). Jeigu veiklos įeiga yra “*IP Output*”, o išeiga “*Process Input*”, nustatoma, kad ši veikla yra funkcijos dalis - “*Realizavimas*”. Galimas variantas, kad negali būti nustatytas konkrečios veiklos tipas. Ši veikla turi vieno tipo veiklos įeigą ir kito tipo veiklos išeigą. Tokiu atveju, panaudojant egzistuojantį – funkcijos sudėties nustatymo algoritmą, surandama kokių tipų veiklų trūksta ir šios veiklos bei jų trūkstami ryšiai įvedami vartotojo.

## **Išvados**

Naudojant „Provision WorkBench v. 4.0.2 Enterprise Pro“, „Microsoft Excel 2002“ ir „Microsoft Visual Studio 7.0“ programinius paketus realizuoti informacijos perkėlimo bei procesų ir funkcijų darbų sekų modelių atskyrimo algoritmai. Sukurta veiklos modelio žinių tvarkybos informacinė sistema. Taip intelektualizuojant informacijos sistemos kūrimo procesą veiklos žinių pagrindu, gali būti sutaupomas IS projektavimo darbo laikas, gerinama sprendimų kokybė – veiklos modelio sukauptos jau patikrintos žinios apie probleminę sritį.

## **Literatūra**

1. A. Lopata Veiklos modelių sudėties analizė. „Informacinės Technologijos 2002“, Kaunas, 2002 psl. 377-381
2. S.Gudas, T. Skersys, A. Lopata Domain Knowledge Integration For Information Systems Engineering. „Informacinės technologijos verslui 2002“ Kaunas, Technologija 2002. psl.56-59
3. A. Lopata, S. Gudas Informacijos išteklių identifikavimas, veiklos modelio pagrindu. Informacijos mokslai, T19, Vilnius. Vilniaus universiteto leidykla, 2001, p.43-50
4. E. Turban, E. McLean, J. Wetherbe Information Technology for Management, John Wiley&Sons.Ins., 1999.
5. Kenneth C.Laudon, Jane P.Laudon Management Information Systems, Prentice-Hall, 2000