

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Kęstutis Budrevičius

CASE veiklos žinių bazės tvarkymo sistema

Magistro darbas

Darbo vadovas
doc. dr. S. Gudas

Kaunas, 2004

Santrauka

Šiuo metu, daugelyje CASE įrankių IS kūrimo gyvavimo ciklo analizės etape surinkta informacija nenaudojama (arba tik dalinai naudojama) projektavimo etapo modeliams generuoti. Tobulinant CASE sistemas siūloma jas papildyti veiklos žinių baze, kuri tampa papildomu informacijos šaltiniu projekto modeliams generuoti.

Darbo metu patikslintas Informacijos sistemų katedroje sukurtas veiklos žinių modelis, suprojektuota ir realizuota veiklos žinių bazės tvarkymo sistema. Sistemos projektavimas atliktas naudojant tokias programines priemones: *Rational Rose Enterprise Edition 2002*, *Microsoft Visio Professional 2002*.

Sukurti vartotojų poreikių modelių (*UCM – Use Case model*) generavimo algoritmai veiklos žinių bazės pagrindu. Sudaryta UCM modelių saugykla ir jų funkcionavimui reikalinga programinė įranga. CASE veiklos žinių bazės tvarkymo sistema sukurta remiantis Informacijos sistemų katedroje sukurtu veiklos meta-modeliu. Realizuota veiklos žinių tvarkymo sistema skirta praplėsti tradicinių CASE sistemų funkcines galimybes.

Realizuota CASE veiklos žinių bazės tvarkymo sistema (*Visual FoxPro 7.0*) leidžia:

- ❖ Įvesti ir koreguoti veiklos žinias - darbų sekų modelių pagrindu aprašytus veiklos elementus: veiklos procesus, veiklos funkcijas, informacinius srautus, materialius srautus, vykdytojus;
- ❖ Susieti veiklos procesus ir veiklos funkcijas, modeliuoti jų hierarchinę struktūrą;
- ❖ Sudaryti ataskaitas apie žinių bazės turinį;
- ❖ Generuoti vartotojų poreikių modelius (UCM) pagal:
 - ✓ vartotojo (analitiko) nurodytą veiklos funkciją;
 - ✓ vartotojo (analitiko) nurodytą veiklos procesą;
 - ✓ vartotojo (analitiko) nurodytą vykdytoją.

Šis darbas yra dalis ISK katedros atliekamo mokslinio tyrimo „Žiniomis grindžiamos IS inžinerijos metodas“.

Summary

Now in most CASE tools the information collected during the stage of IS cycle analysis is not used to generate the models of the designing stage. Improving CASE systems it is offered to enlarge them with an activity knowledge basis, what becomes an additional source of information to generate project model.

Activity knowledge model created in the department of data systems has been specified and activity setting system has been designed and realized during the work. System design has been conducted using programming appliances: *Rational Rose Enterprise Edition 2002*, *Microsoft Visio Professional 2002*.

Use Case model generating algorithms has been created on the grounds of the activity basis. UCM models storage and software needed for its functioning have been worked out. CASE activity knowledge basis ordering system has been created on the grounds of the activity meta-model worked out in the department of data systems. Realized activity knowledge ordering system is used to extend traditional CASE systems functional possibilities.

Realized CASE activity knowledge basis ordering system (*Visual FoxPro 7.0*) lets:

- ❖ To input and adjust activity knowledge – activity elements described on the grounds of work model sequences activity procedures, activity functions, information flow, material flow and users;
- ❖ To relate activity processes, activity functions, simulate their hierarchy structure;
- ❖ To make accounts about the content of knowledge basis;
- ❖ To generate users case models (UCM) according to:
 - ✓ users directed activity function;
 - ✓ users directed activity process;
 - ✓ users directed executor.

This work is a part of the scientific research “IS engineering method grounded on knowledge” carried out in the department of IS.

Turinys

| | |
|---|-----------|
| 1. ĮVADAS | 9 |
| 2. TERMINŲ IR SANTRUMPŲ ŽODYNAS | 11 |
| 3. ANALITINĖ DALIS | 12 |
| 3.1. CASE SISTEMŲ ANALIZĖ APŽVALGA | 12 |
| 3.2. VEIKLOS MODELIŲ ANALIZĖ | 14 |
| 3.2.2. Hierarchinis veiklos modelis | 14 |
| 3.2.3. Veiklos modelio klasių diagrama | 15 |
| 3.2.4. Tarptautinių veiklos modeliavimo standartų analizė..... | 15 |
| 3.3. VEIKLOS METAMODELIS | 18 |
| 3.4. PROGRAMINIŲ PRIEMONIŲ PASIRINKIMAS | 19 |
| 3.5. VARTOTOJO POREIKIŲ MODELIO (UCM) GENERAVIMAS | 19 |
| 3.5.1. Veiklos žiniomis grindžiama IS inžinerija..... | 19 |
| 3.5.2. Panaudojimo atvejų modelio formalus aprašymas | 20 |
| 3.5.3. Praktiniai UCM taikymo ypatumai | 20 |
| 4. TYRIMO DALIS | 21 |
| 4.1. SISTEMOS ARCHITEKTŪRA | 21 |
| 4.1.1. CASE veiklos žinių bazės tvarkymo sistemos loginis vaizdas | 21 |
| 4.1.2. Detalus posistemų aprašymas..... | 22 |
| 4.2. ŽINIŲ BAZĖS LOGINĖS STRUKTŪROS PROJEKTAS..... | 24 |
| 4.2.1. ER modelis..... | 24 |
| 4.2.2. Dalykinės srities klasių diagrama..... | 25 |
| 4.3. UCM GENERAVIMAS | 27 |
| 4.3.1. UCM modelių saugykla..... | 28 |
| 4.3.2. Proceso UCM generavimas..... | 29 |
| 4.3.3. Funkcijos UCM generavimas | 32 |
| 4.3.4. Vykdytojo UCM generavimas..... | 35 |
| 5. EKSPERIMENTINĖ DALIS | 38 |
| 5.1. VARTOTOJO SĄSAJOS PROJEKTAS | 38 |
| 5.2. VARTOTOJO ATMINTINĖ..... | 39 |
| 5.2.1. Sistemos paskirtis..... | 39 |
| 5.2.2. Sistemos galimybės | 39 |
| 5.2.3. Darbas su sistema..... | 40 |
| 5.3. DETALUS SISTEMOS APRAŠYMAS..... | 40 |
| 5.3.1. Sistemos sandara | 41 |
| 5.3.2. Informacija..... | 41 |
| 5.3.3. Procesai..... | 42 |
| 5.3.4. Funkcijos..... | 53 |
| 5.3.5. Procesų ir funkcijų ryšys | 57 |
| 5.3.6. Vykdytojai..... | 60 |
| 5.3.7. Ataskaitos | 62 |

| | | |
|-----------|-----------------------------------|-----------|
| 5.3.8. | <i>Darbo pabaiga</i> | 62 |
| 5.3.9. | <i>Galimos klaidos</i> | 63 |
| 5.3.10. | <i>Pradiniai nustatymai</i> | 64 |
| 5.3.11. | <i>Sistemos diegimas</i> | 65 |
| 5.4. | UCM GENERAVIMAS..... | 66 |
| 6. | IŠVADOS | 67 |
| 7. | LITERATŪRA | 69 |
| 8. | PRIEDAI | 70 |

Paveikslai

| | |
|--|----|
| 1 pav. Egzistuojančių CASE priemonių funkcinė palyginimo diagrama | 12 |
| 2 pav. Hierarchinis veiklos modelis | 14 |
| 3 pav. Veiklos modelio klasių diagrama..... | 15 |
| 4 pav. CEN ENV 4003 standarte naudojama principinė veiklos modeliavimo schema | 16 |
| 5 pav. Organizacijos modeliavimui būtini pagrindiniai konstruktai, apibrėžti CEN ENV 12204 standarte | 16 |
| 6 pav. UEML principinė schema..... | 17 |
| 7 pav. Veiklos metamodelis | 18 |
| 8 pav. UCM metamodelis | 20 |
| 9 pav. CASE veiklos žinių bazės tvarkymo sistemos loginis vaizdas..... | 21 |
| 10 pav. Žinių bazės esybių ryšių diagrama | 24 |
| 11 pav. CASE žinių tvarkymo sistemos dalykinės srities klasių diagrama | 26 |
| 12 pav. CASE žinių bazės tvarkymo sistema | 27 |
| 13 pav. UCM bazės dalykinė srities klasių diagrama..... | 28 |
| 14 pav. Proceso UCM generavimo algoritmas | 29 |
| 15 pav. Funkcijos UCM generavimo algoritmas | 32 |
| 16 pav. Vykdytojo UCM generavimo algoritmas | 35 |
| 17 pav. Vartotojo sąsaja struktūrinė schema..... | 38 |
| 18 pav. Pagrindinis sistemos langas | 40 |
| 19 pav. Sistemos struktūrinė schema..... | 41 |
| 20 pav. Sistemos langas „Autorius“ | 41 |
| 21 pav. Sistemos langas „Procesai“ | 42 |
| 22 pav. Sistemos langas „Naujo proceso įvedimas“ | 42 |
| 23 pav. Pranešimas apie egzistuojantį įrašą | 43 |
| 24 pav. Pranešimas apie įrašytą procesą | 43 |
| 25 pav. Sistemos pranešimai apie šalinimo veiksmų patvirtinimą..... | 43 |
| 26 pav. Pranešimai apie atliktus pašalinimus..... | 43 |
| 27 pav. Sistemos langas „Proceso atributai“ | 44 |
| 28 pav. Sistemos langas „Proceso atributai“ | 44 |
| 29 pav. Pranešimas apie įrašytą atributą | 44 |
| 30 pav. Sistemos langas „Proceso koregavimas“ | 45 |
| 31 pav. Pranešimas apie atliktus pakeitimus..... | 45 |
| 32 pav. Proceso hierarchijos lygio pasirinkimas | 46 |

| | | |
|---------|--|----|
| 33 pav. | Sistemos langas „Pasirinkto lygio procesai“ | 46 |
| 34 pav. | Sistemos langas „Pasirinkto lygio procesai“ – redagavimo režimas | 47 |
| 35 pav. | Sistemos langas „Procesų dekompozicija“ | 48 |
| 36 pav. | Sistemos langas „Procesų dekompozicija“ – procesų priskyrimas | 49 |
| 37 pav. | Pranešimas apie negalimą veiksmą | 49 |
| 38 pav. | Sistemos langas „Procesų įeiga/išeiga“ | 50 |
| 39 pav. | Sistemos langas „Materialaus srauto pasirinkimas“ | 50 |
| 40 pav. | Pranešimas apie jau priskirtą srautą | 51 |
| 41 pav. | Pranešimas apie sėkmingą veiksmą | 51 |
| 42 pav. | Sistemos langas „Materialus srautas“ | 52 |
| 43 pav. | Sistemos langas „Funkcijos“ | 53 |
| 44 pav. | Funkcijų hierarchijos lygių pasirinkimo langas | 54 |
| 45 pav. | Sistemos langas „Pasirinkto lygio funkcijos“ | 54 |
| 46 pav. | Dekomponuojamos funkcijos pasirinkimo langas | 55 |
| 47 pav. | Sistemos langas „Funkcijų dekompozicija“ – funkcijų priskyrimas | 55 |
| 48 pav. | Funkcijų įeigos ir išeigos priskyrimo langas | 56 |
| 49 pav. | Sistemos langas „Informacinis srautas“ | 57 |
| 50 pav. | Sistemos langas „Procesų ir funkcijų ryšys“ | 58 |
| 51 pav. | Sistemos langas „Procesų ir funkcijų susiejimas“ | 58 |
| 52 pav. | Pranešimas apie egzistuojantį ryšį | 59 |
| 53 pav. | Pranešimas apie sėkmingai atliktą veiksmą | 59 |
| 54 pav. | Sistemos langas „Metodas“ | 59 |
| 55 pav. | Proceso hierarchijos lygio pasirinkimas | 60 |
| 56 pav. | Sistemos langas „Vykdytojai“ | 60 |
| 57 pav. | Sistemos langas „Vykdytojai“ – redagavimo režimas | 61 |
| 58 pav. | Reikalavimas patvirtinti savo pasirinkimą | 62 |
| 59 pav. | Pranešimai apie neįvestus laukus | 63 |
| 60 pav. | Pranešimas apie egzistuojantį atributą | 63 |
| 61 pav. | Pranešimas apie ryšius tarp įrašų | 63 |
| 62 pav. | Regiono nustatymas | 64 |
| 63 pav. | Kalbos pasirinkimas | 64 |
| 64 pav. | Atributų nustatymas | 65 |
| 65 pav. | Generuojamo UCM modelio pasirinkimas | 66 |
| 66 pav. | Proceso pasirinkimas | 66 |

Lentelės

| | |
|---|----|
| 1 lentelė. Pagrindiniai organizacijos veiklos modeliavimo standartai | 16 |
| 2 lentelė. Šiuo metu kuriami su organizacijos modeliavimu susiję standartai..... | 17 |
| 3 lentelė. Posistemė – vartotojo sąsaja | 22 |
| 4 lentelė. Posistemė – veiklos aptarnavimo sluoksnis..... | 23 |
| 5 lentelė. Posistemė – veiklos objektai | 23 |
| 6 lentelė. Žiniomis grindžiamo panaudojimo atvejų modelio generavimo variantai..... | 27 |

1. Įvadas

Darbo tikslas – suprojektuoti ir realizuoti CASE veiklos žinių bazės tvarkymo sistemą. Sistema skirta analitikams, modeliuojantiems nagrinėjamos organizacijos veiklą IS inžinerijos tikslais. Žinias apie dalykinę sritį analitikai kaupia naudodami „Veiklos žinių kaupimo posistemį“, kuris surenka ir patikrina veiklos žinias - darbų sekų modelių pagrindu aprašytą veiklos sritį. Realizuota CASE veiklos žinių bazės tvarkymo sistema leidžia kaupti informaciją CASE veiklos modelio žinių bazėje, atlikti sukauptos informacijos analizę, jos koregavimą ir UCM (*Use Case model*) modelių generavimą.

Suprojektuota ir programiškai realizuota CASE sistemos žinių bazė veiklos modelio pagrindu. CASE veiklos žinių bazės tvarkymo sistemos kūrimo darbo etapus galime suskirstyti į veiklos modelio, CASE sistemų, tarptautinių veiklos modeliavimo standartų analizę, kuriamos sistemos duomenų bazės projektavimą, sistemos kūrimą ir UCM modelių generavimą žinių bazėje sukauptos informacijos pagrindu.

Siekiant parinkti veiklos modelio sudėtį ir atskleisti jo inžinerines panaudojimo galimybes (vartotojo poreikių modelio generavimas), darbo metu atlikta CASE sistemose taikomų veiklos modeliavimo priemonių analizė (*Oracle Designer2000, Provision Workbench 3.1, Rational Rose Enterprise Edition, Object Engineering Workbench (OEW), System Architect*) ir tarptautinių veiklos modeliavimo standartų analizė (*ISO 14258, ISO 15704, ISO 10314, ISO/IEC 15288, CEN ENV 40003, CEN ENV 12 204*).

Veiklos analizės metu surinkta informacija tvarkoma kaip žinių struktūra, leidžianti generuoti IS projektinius modelius (UCM, kaip vartotojo poreikių specifikaciją, klasių modelį ir kt.). Tokią žinių struktūrą patalpintą CASE žinių bazėje vadiname veiklos modeliu (angl. *Enterprise model*) [1].

Darbe pateikiama veiklos modelio sudėtis, CASE veiklos žinių bazės struktūra, realizuojamos sistemos ypatybės, IS projekto modelių generavimo galimybės.

Sukurta veiklos žinių bazė susideda iš 17 esybių. Pagrindinės esybės: veiklos procesai, veiklos funkcijos, materialūs, informaciniai srautai, vykdytojai, veiklos, tikslai, įvykiai. Realizuota CASE veiklos žinių bazės tvarkymo sistema (*Visual FoxPro 7.0*) leidžianti:

- ❖ įvesti ir koreguoti veiklos žinias - darbų sekų modelių pagrindu aprašytus veiklos elementus: veiklos procesus, veiklos funkcijas, informacinius srautus, materialius srautus, vykdytojus;
- ❖ susieti veiklos procesus ir veiklos funkcijas, modeliuoti jų hierarchinę struktūrą;
- ❖ nustatyti vykdytojų valdomus veiklos procesus ir veiklos funkcijas;
- ❖ sudaryti ataskaitas apie žinių bazės turinį (veiklos procesų ir funkcijų, vykdytojų vykdomų procesų, vykdytojų, materialių ir informacinių srautų) ir išsamias pasirinktų veiklos procesų ir funkcijų ataskaitas (veiklos procesų ir funkcijų, funkcijos valdomų procesų).

CASE žinių bazės tvarkymo sistema sėkmingai patikrinta įvedant, koreguojant, šalinant gamybinės įmonės modelį. Kontrolinių duomenų apimtis: 4 hierarchijos lygiai, 250 įrašų.

Sukurta žinių bazė panaudojimo atvejų modelių UCM (*Use Case model*) generavimui. Pateikiami UCM generavimo algoritmai pagal:

- ❖ vartotojo (analitiko) nurodytą veiklos funkciją;
- ❖ vartotojo (analitiko) nurodytą veiklos procesą;
- ❖ vartotojo (analitiko) nurodytą vykdytoją.

Sukurtų UCM generavimo algoritmų pagrindu Visual FoxPro 7.0 programinėmis priemonėmis realizuotas UCM generavimo posistemis: UCM modelių saugykla, veiklos proceso UCM generavimo modulis, veiklos funkcijos UCM generavimo modulis, vykdytojo UCM generavimo modulis.

Daugelyje CASE priemonių objektinis IS kūrimas pradedamas nuo panaudojimo atvejų modelių. UCM paprastai sudaro sistemos analitikas, nagrinėdamas kompiuterizuojamąją veiklos sritį bei vartotojo pateiktus funkcinis ir nefunkcinis reikalavimus kuriamai informacijos sistemai (IS). Taip UCM tampa pagrindine ir veiklos, ir vartotojo poreikių modeliavimo priemone [1].

Šis darbas yra antroji kuriamo IS inžinerijos proceso dalis, integruota su vartotojo poreikių modeliavimo ir analizės technologija, kurią atliko magistrantė Sandra Budrevičienė.

Darbo ypatumas – sukurtas ir realizuotas metodas, kuris leidžia naudojant sukurtus algoritmus sukauptą informaciją analizuoti, koreguoti ir generuoti UML projektinius modelius.

2. Terminų ir santrumpų žodynas

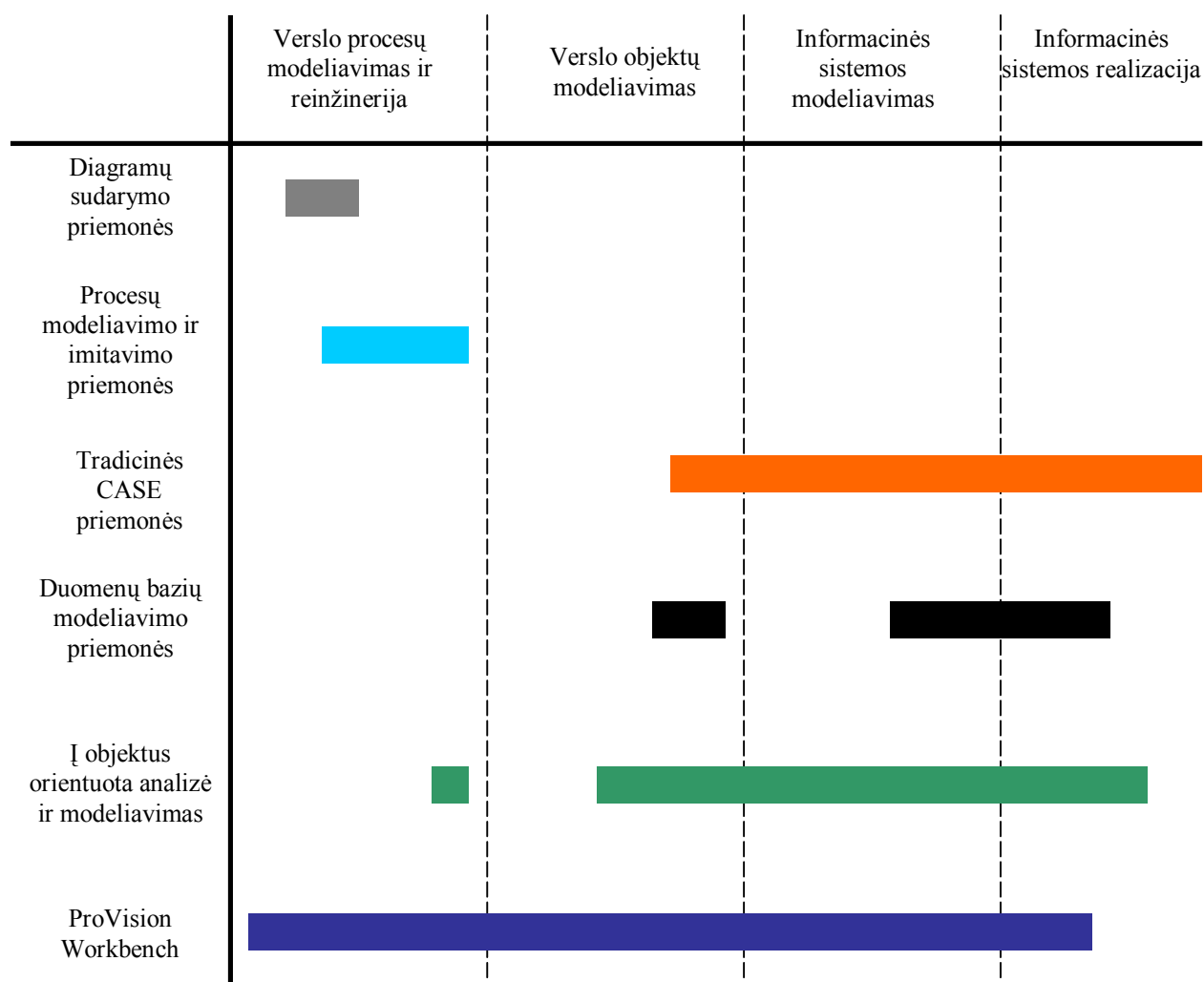
| | | |
|------|---|--|
| CASE | - | kompiuterizuotas IS projektavimas (<i>Computer Aided System Engineering</i>) |
| CD | - | kompaktinė plokštelė |
| DB | - | duomenų bazė |
| DBVS | - | duomenų bazės valdymo sistema |
| DSD | - | duomenų srautų diagrama |
| IS | - | informacinė sistema |
| LAN | - | vienos organizacijos tinklas (<i>Local Area Network</i>) |
| PA | - | panaudojimo atvejis |
| PC | - | personalinis kompiuteris |
| UCM | - | panaudojimo atvejų modelis (<i>Use Case Model</i>) |
| UML | - | vieninga modeliavimo kalba (<i>Unified Modeling Language</i>) |

3. Analitinė dalis

3.1. CASE sistemų analizė apžvalga

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. Būtent tokia sistema suprojektuota ir realizuota darbo metu.

1 paveiksle matome daugelio tipo modeliavimo priemonių palyginimus.



1 pav. Egzistuojančių CASE priemonių funkcinė palyginimo diagrama

Egzistuojančios Modeliavimo priemonės:

- ❖ **Diagramų sudarymo priemonės** (*diagramming tools*). Šios priemonės skirtos tiktai 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.*).
- ❖ **Į objektą orientuota analizė ir modelivimas** (*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.*). Apie 78 procentų CASE priemonių yra į objektą 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).

3.2. Veiklos modelių analizė

3.2.1. Veiklos modelis pagal M. Porter

Organizacijos veiklos funkcijoms ir procesams aprašyti gali būti naudojamas M.Porter pasiūlytas organizacijos vertės grandinės modelis [2]. Šis modelis aprašo organizacijoje vykstančius procesus, jų valdymo funkcijas bei ryšius tarp funkcijų ir procesų. Vertės grandinės modelyje išskiriami procesai, kurie formuoja veiklos produktą ir veiklos valdymo funkcijos.

Procesas – tai veiklos vienetą, nurodantis organizacijos darbų seką. Būtent procesas formuoja organizacijos veiklos produktą. Proceso įeigos ir išeigos srautai yra tik materialūs. Įeigos keitimo į išeigą metu procesas naudoja resursus. Procesai gali būti išskaidomi į smulkesnes dedamąsias: subprocesus, veiklas, užduotis, operacijas.

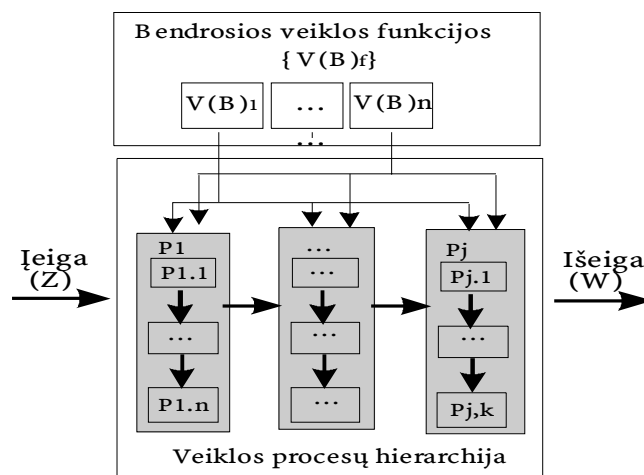
Funkcija – tai organizacijoje vykstančių darbų sekų elementas, skirtas kontroliuoti ir valdyti procesus. Funkcijos srautai gali būti tik informacinio tipo.

Vertės grandinės modelis (pagal M. Porterį) yra organizacijos veiklos modelis, kurio pagrindas – veiklos valdymo funkcijų ir veiklos procesų sąveika [3].

3.2.2. Hierarchinis veiklos modelis

Tradicinis vertės grandinės modelis apibrėžia tik pagrindinius veiklos procesus bei funkcijas. Tokio detalumo nepakanka kad būtų išspręsta veiklos kompiuterizavimo problema. Tikslinga sudaryti išsamesnį vertės grandinės modelį, išskaidant veiklos funkcijas į bendrąsias ir specializuotąsias bei suskirstant veiklos procesus į hierarchinius lygmenis. Specializuotųjų ir bendrųjų informacijos išteklių skirtumas yra tas, kad specializuotieji informacijos ištekliai gali būti naudojami vienam proceso etapui ar po etapui įgyvendinti, o bendrieji informacijos ištekliai reikalingi kiekvieno hierarchijos lygmens procesų etapams [4].

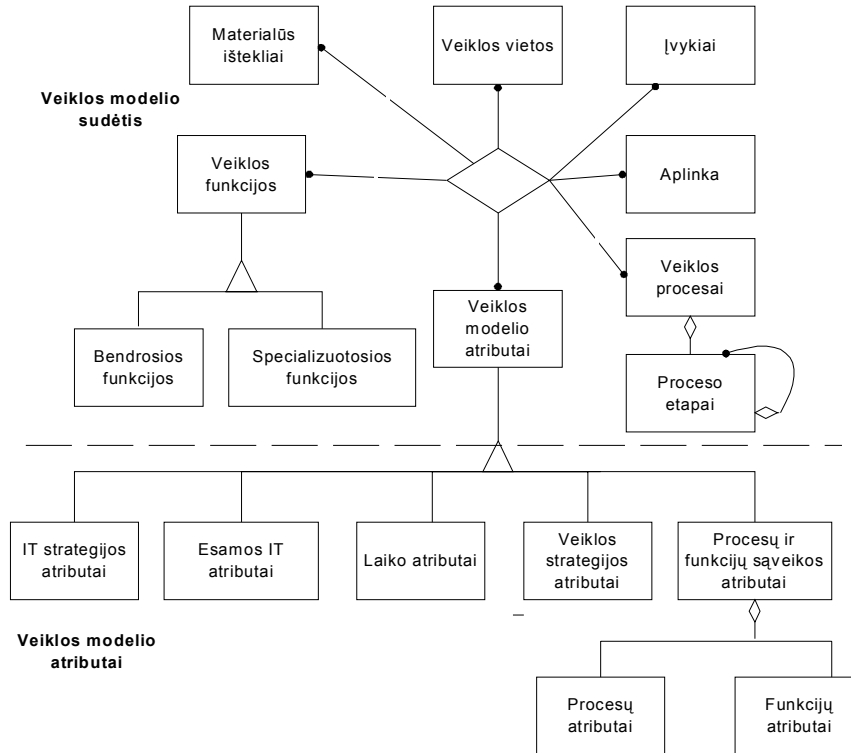
Su vertės grandinės modelio elementais (valdymo funkcijomis ir procesais) galima susieti informacinius išteklius, būtinus veiklos proceso etapų valdymui įgyvendinti (2 pav.) [4].



2 pav. Hierarchinis veiklos modelis

3.2.3. Veiklos modelio klasių diagrama

Šiuolaikinės organizacijos duomenų saugykla turėtų saugoti ne tik veiklos duomenis, bet ir veiklos informacijos išteklių specifikacijas. Vienas iš pagrindinių modelių, kurių objektinė IS kūrimo metodologija rekomenduoja kompiuterizuojamiems procesams ir problemoms struktūrinti, yra klasių modelis. Aptarto veiklos modelio klasių diagrama pateikiama 3 paveiksle.



3 pav. Veiklos modelio klasių diagrama

Ši klasių diagrama naudojama projektuojant organizacijos veiklos duomenų saugyklą.

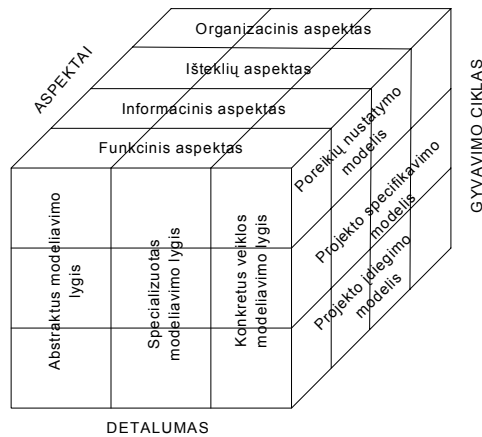
3.2.4. Tarptautinių veiklos modeliavimo standartų analizė

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

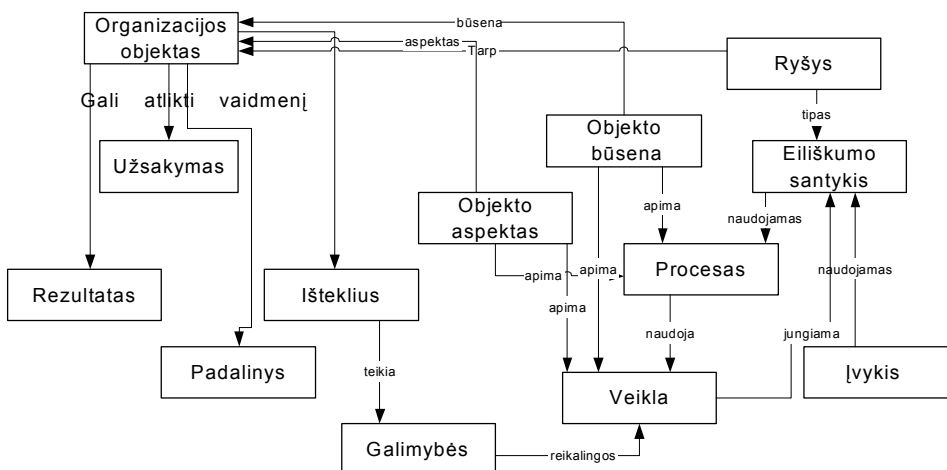
| Standarto kodas | Standarto pavadinimas |
|-----------------|---|
| ISO 14258 | Concepts and Rules for Enterprise Models |
| ISO 15704 | Requirements for Enterprise reference architectures and methodologies |
| ISO 10314 | Shop floor production model |
| ISO/IEC 15288 | System life cycle processes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

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 (4 pav.).



4 pav. CEN ENV 4003 standarte naudojama principinė veiklos modeliavimo schema

CEN ENV 40003 standarte veiklos modeliavimui būtinas sudėtinės dalis (konstruktus) apibrėžia CEN ENV 12204 standartas (5 pav.).

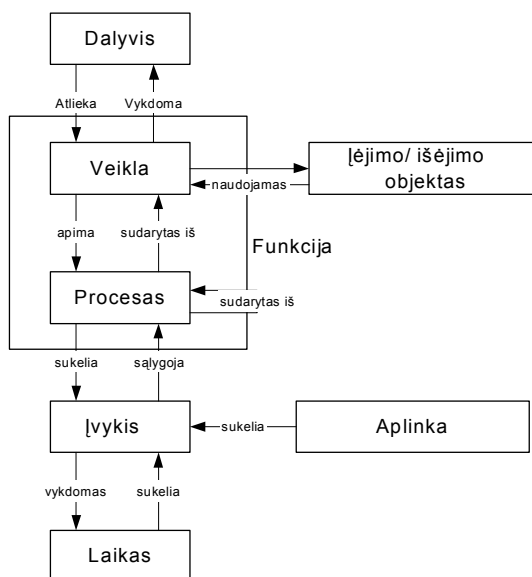


5 pav. Organizacijos modeliavimui būtini pagrindiniai konstruktai, apibrėžti CEN ENV 12204 standarte

Tačiau, kaip parodė praktika, CEN ENV 40003 (*Computer Integrated Manufacturing - Systems Architecture - Framework for Enterprise Modeling*), CEN ENV 12204 (*Advanced Manufacturing Technology - Systems Architecture - Constructs for Enterprise Modeling*) [11] standartai pilnai 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 lentelė) bei kalbos, pvz., UEML [7] (*Unified Enterprise Modeling Language*) (6 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 |

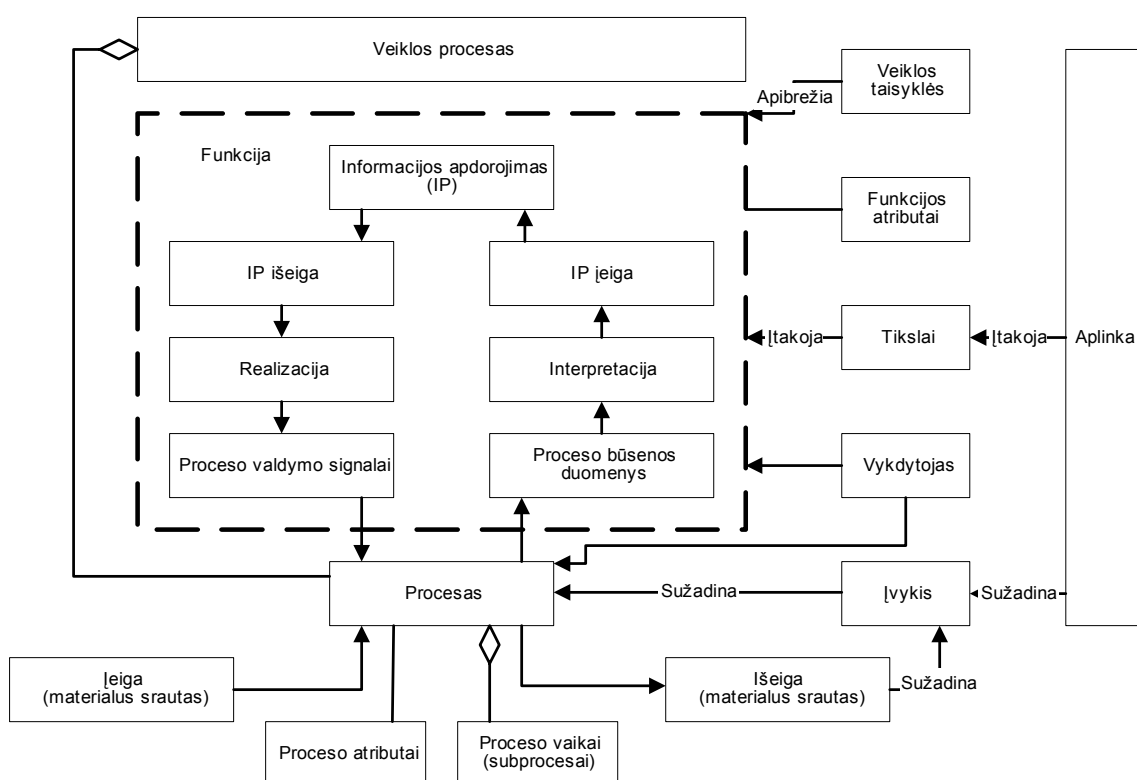


6 pav. UEML principinė schema

3.3. Veiklos metamodelis

Veiklos metamodelis sukurtas atsižvelgiant į tarptautinių standartų ENV 12204 (*Advanced Manufacturing Technology - Systems Architecture - Constructs for Enterprise Modeling*), ENV 4003 (*Computer Integrated Manufacturing - Systems Architecture - Framework for Enterprise Modeling*), UEMML reikalavimus. Veiklos metamodelis apibrėžia veiklos modelio sudėtį. 7 paveikslo veiklos metamodelis sukurtas remiantis:

- ❖ Porter Veiklos grandinės modeliui;
- ❖ Hendersono veiklos ir IT sąsajos modeliui;
- ❖ Tarptautiniais veiklos modeliavimo standartais (UEMML, ENV 12204, ENV 40003, IDEF3).



7 pav. Veiklos metamodelis

Veiklos metamodelio sudarymas yra ISK katedros atliekamo mokslinio tyrimo “Žiniomis grindžiamos IS inžinerijos metodo” dalis. CASE žinių bazės tvarkymo sistema sukurta veiklos metamodelis pagrindu.

3.4. Programinių priemonių pasirinkimas

Dauguma vartotojų naudoja *Windows* operacinę sistemą (*Windows95 /98 /2000 /XP*). Visual programavimo kalbos užtikrina gerą vartotojo sąsają, pagreitina kūrimo procesą, todėl programinės įrangos kūrimui pasirinktas *Visual FoxPro 7.0* paketas.

Sistema projektuota *Rational Rose Enterprise Edition 2002*, *MS Visio Professional 2002* paketų pagalba.

3.5. Vartotojo poreikių modelio (UCM) generavimas

Informacijos sistemų inžinerijos gyvavimo ciklas pradedamas veiklos dalykinės srities analize, vartotojo poreikių analize ir informacinių poreikių specifikacijos sudarymu. Į objektą orientuotoje IS inžinerijoje veiklos analizei gali būti taikoma veiklos (angl. *Activity*) diagrama ir panaudojimo atvejų modelis (angl. *Use Case Model*), kurį labai skirtingai įvairūs autoriai verčia į lietuvių kalbą (užduočių, taikomųjų uždavinių modelis), todėl toliau šį modelį vadinsime UCM (*UML 1.4, 2000*). Daugelyje CASE priemonių objektinis IS kūrimas pradedamas nuo panaudojimo atvejų modelio. UCM paprastai sudaro sistemos analitikas, nagrinėdamas kompiuterizuojamąją veiklos sritį bei vartotojo pateiktus funkcinius ir nefunkcinius reikalavimus kuriamai informacijos sistemai (IS). Taip UCM tampa pagrindine ir veiklos, ir vartotojo poreikių modeliavimo priemone [9].

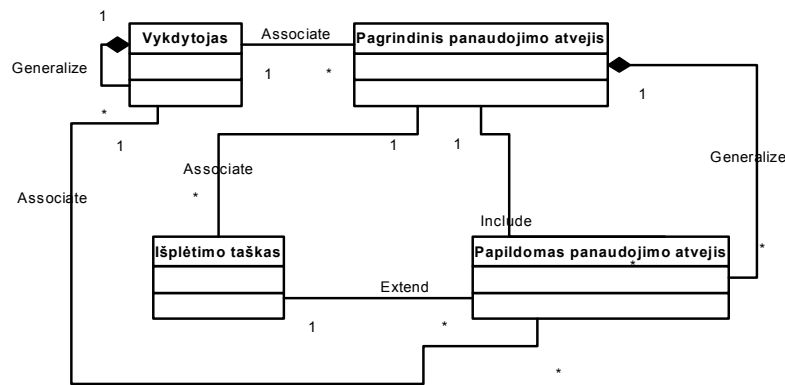
Siekiant tobulinti informacijos sistemos kūrimo procesą, norint sukurti žiniomis grindžiamą IS inžineriją, šios UML priemonės veiklai modeliuoti nepakankamos. Mūsų požiūriu veiklos analizės metu surinkta informacija turi būti sutvarkoma kaip žinių struktūra, leidžianti generuoti IS projektinius modelius (UCM, kaip vartotojo poreikių specifikaciją, klasių modelį ir kt.). Tokią žinių struktūrą vadinsime veiklos modeliu (angl. *Enterprise model*) [9].

3.5.1. Veiklos žiniomis grindžiama IS inžinerija

Šiuo metu, daugelyje CASE įrankių IS kūrimo gyvavimo ciklo analizės etape surinkta informacija nenaudojama (arba tik dalinai naudojama) projektavimo etapo modeliams generuoti. Naudojant „veiklos žinių kaupimo posistemį“ kompiuterizuojamosios dalykinės srities žinios gali būti surenkamos ir kaupiamos CASE veiklos žinių bazėje. Naudojant šią sukaupią informaciją, būtų galima generuoti UML apibrėžtus projektinius modelius. Teoriškai panaudojimo atvejų modeliai gali būti generuojami kiekvienai modelio klasei, tačiau tiksliausia juos generuoti procesų, funkcijų, vykdytojų klasėms.

3.5.2. Panaudojimo atvejų modelio formalus aprašymas

Detaliai UCM sudėtis aprašoma UML specifikacijos 1.4 versijoje (*UML 1.4, 2000*). Šioje specifikacijos versijoje aprašomo UCM metamodelis pateikiamas 8 paveiksle. Šiame paveiksle išsaugoti UML specifikacijoje originalo kalba apibrėžti ryšių pavadinimai. UCM pagrindinės sudėtinės dalys yra panaudojimo atvejis (angl. *Use Case*), vykdytojas (angl. *Actor*) ir juos siejanti asociacija (informacijos srautas). Panaudojimo atvejis suprantamas kaip vartotojo poreikis arba užduotis, kurią reikia kompiuterizuoti. Panaudojimo atvejai tarpusavyje gali būti siejami „*Include*“ arba „*Extends*“ ryšiais. „*Include*“ ryšys nurodo kad panaudojimo atvejis susideda iš dalių - žemesnio lygmens panaudojimo atvejų. „*Extends*“ ryšiu prie pagrindinio panaudojimo atvejo prijungiamas papildomas panaudojimo atvejis, kurio rezultatu naudojasi pagrindinis panaudojimo atvejis. Išplėtimo taškas (angl. *Extention point*) yra nuoroda į vietą, kur, galiojant apibrėžtai sąlygai, gali kreiptis kiti panaudojimo atvejai [9].



8 pav. UCM metamodelis

3.5.3. Praktiniai UCM taikymo ypatumai

Praktikoje UCM gali būti taikomas dviem tikslams, kurie siejasi tarpusavyje:

- ❖ analizuojamos veiklos srities modeliui aprašyti – nurodyti svarbiausias veiklos dalyvių sąveikas (materialias ir informacines) su veiklos uždaviniais (gamybiniais, informacijos apdorojimo);
- ❖ kompiuterizuojamos veiklos srities informaciniams poreikiams specifikuoti – modeliuoti tik informacines sąveikas tarp veiklos dalyvių ir kompiuterizuojamų procesų bei funkcijų (t.y. taikomųjų uždavinių).

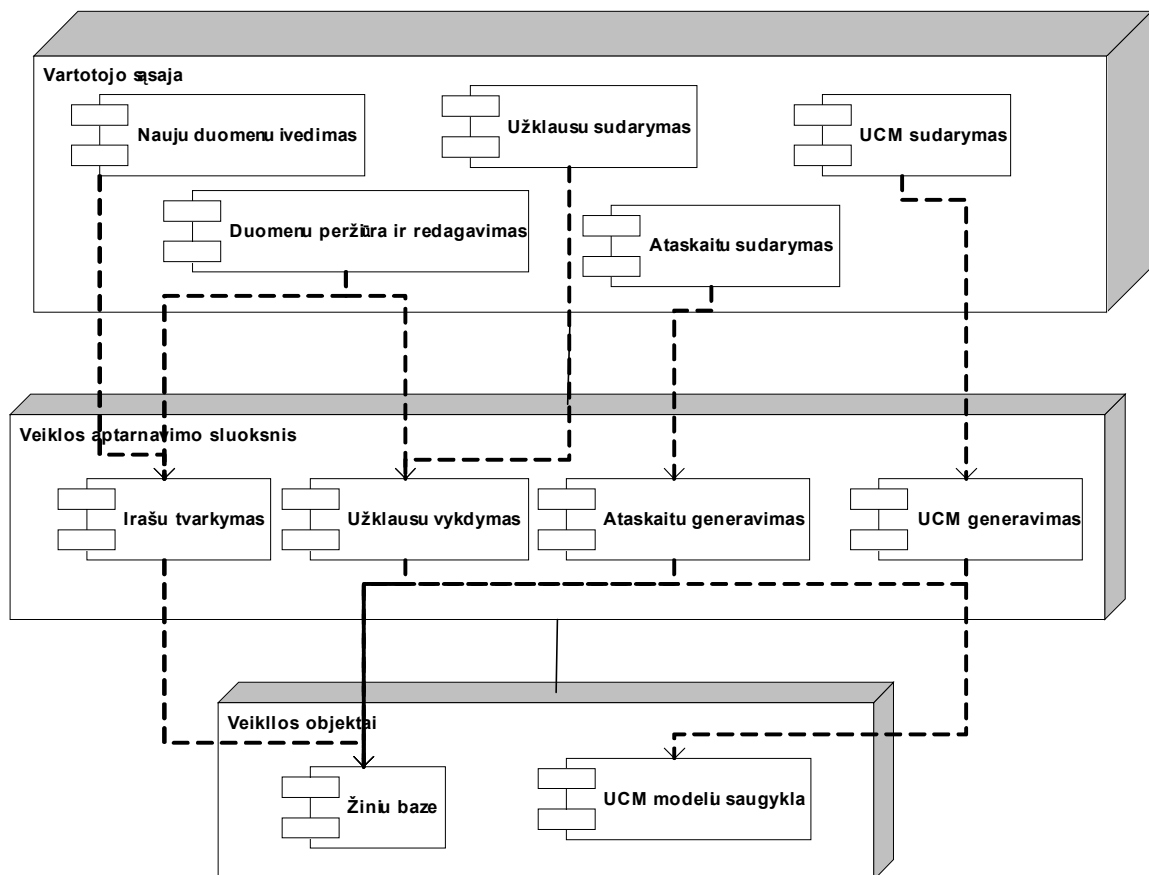
Pirmuoju atveju UCM vadinsime veiklos uždavinių modeliu. Tai apibendrintas modelis, srautai gali būti materialūs ir informaciniai, neįvardinti, o veiklos užduočių turinys neaprašytas. Antruoju atveju UCM vadinsime vartotojo (informacinių) poreikių modeliu, nes visi srautai - informaciniai ir identifikuoti (įvardinti). Vartotojo (informacinių) poreikių modelis gali būti aprašytas detaliai, nurodant informacijos srautų struktūras, taikomųjų uždavinių formulavimus, reikalavimus sprendimo eigai ir rezultatams, jis vadinamas taikomųjų uždavinių modeliu. Tai pakankamai tiksliai vartotojo reikalavimų pageidaujama informacijos sistemai specifikacija.

4. Tyrimo dalis

4.1. Sistemos architektūra

4.1.1. CASE veiklos žinių bazės tvarkymo sistemos loginis vaizdas

„CASE veiklos žinių bazės tvarkymo sistema“ loginiu požiūriu susideda iš trijų pagrindinių sluoksnių: vartotojo sąsaja, veiklos aptarnavimo sluoksnis, veiklos objektai (9 pav.).



9 pav. CASE veiklos žinių bazės tvarkymo sistemos loginis vaizdas

Vartotojo sąsaja susideda iš ribinių klasių langams, kurių pagalba vartotojas bendrauja su sistema. Šis sluoksnis turi visas ribines klases, kurios generuoja vartotojo matomas ekrano formas (tai ekrano formos, formų mygtukų, laukelių išsidėstymas ir t.t.).

Veiklos aptarnavimo sluoksnis susideda iš valdymo klasių, kurios atlieka duomenų koregavimą, įvedimą, filtravimą, vykdo vartotojo suformuotus užklausus, generuoja panaudojimo atvejų modelius (UCM).

Veiklos objektų sluoksnis apima visas esybių klases, kurios reiškia sistemoje egzistuojančius „objektus“ (procesai, funkcijos, vykdytojai, srautai), bei UCM modelių saugyklos ribines klases.

4.1.2. Detalus posistemių aprašymas

Komponentai, pateikti sistemos architektūroje, turi būti detaliai aprašyti (Appleton, 2002) [13]. Kiti žemesnio lygio komponentai turi būti taip pat detalizuoti, t.y. turi būti “lendama” gilyn. Žemiau pateiktose lentelėse (3, 4, 5 lentelės) panaudojant detalios sistemos architektūros šablonus (detailed system design) [13] aprašyta pagrindinės sistemos posistemės.

3 lentelė. Posistemė – vartotojo sąsaja

| Vartotojo sąsaja | |
|-------------------------|--|
| <i>Klasifikacija</i> | Posistemė |
| <i>Apibrėžimas</i> | Šioje posistemėje veikia vartotojo sąsajos komponentai. Ji apima visas vartotojo sąsajos ribines klases. |
| <i>Atsakomybės</i> | Posistemėje realizuota vartotojo sąsaja, formų pasirinkimas, duomenų įvedimas, duomenų atvaizdavimas, užklausų, ataskaitų sudarymą, UCM sudarymas. |
| <i>Struktūra</i> | Posistemė susideda iš šių komponentų: <ul style="list-style-type: none">❖ Duomenų peržiūra ir redagavimas❖ Naujų duomenų įvedimas❖ Užklausų sudarymas❖ Ataskaitų sudarymas❖ UCM sudarymas |
| <i>Sąveikavimas</i> | Posistemė sąveikauja su veiklos aptarnavimo posisteme. |
| <i>Resursai</i> | Ši posistemė naudojami kompiuterio atmintimi, procesoriaus resursais, tai pat programines įrangos bibliotekomis komponentų elementams generuoti (formoms, mygtukams, laukams ir t.t.). Taip pat naudoja tokius įvedimo įrenginius kaip pelė, klaviatūra informacijos įvedimui, bei monitorių, spausdinimo įrenginį informacijos išvedimui. |

4 lentelė. Posistemė – veiklos aptarnavimo sluoksnis.

| Veiklos aptarnavimo sluoksnis | |
|--------------------------------------|---|
| <i>Klasifikacija</i> | Posistemė |
| <i>Apibrėžimas</i> | Posistemė, kurioje veikia veiklos aptarnavimo sluoksnio komponentai. Jis apima visas valdymo klases. |
| <i>Atsakomybės</i> | Ši posistemė yra tarpininkas tarp vartotojo sąsajos ir duomenų bazės. Joje vykdomi pagrindiniai sistemos veiksmai (įvestų, taisyčių duomenų įrašymas į žinių bazę, duomenų atrinkimas, filtravimas, rūšiavimas, UCM generavimas). |
| <i>Struktūra</i> | Posistemė susideda iš tokių komponentų: <ul style="list-style-type: none"> ❖ Įrašų tvarkymas ❖ Užklausų vykdymas ❖ Ataskaitų generavimas ❖ UCM generavimas |
| <i>Sąveikavimas</i> | Posistemė yra tarpininkas tarp vartotojo sąsajos posistemės ir veiklos objektų. |
| <i>Resursai</i> | Ši posistemė naudojami kompiuterio atmintimi, procesoriaus resursais. Tuo atveju kai duomenų bazė yra nutolusiame kompiuteryje, ji naudojami vietiniu (LAN) tinklu. |

5 lentelė. Posistemė – veiklos objektai.

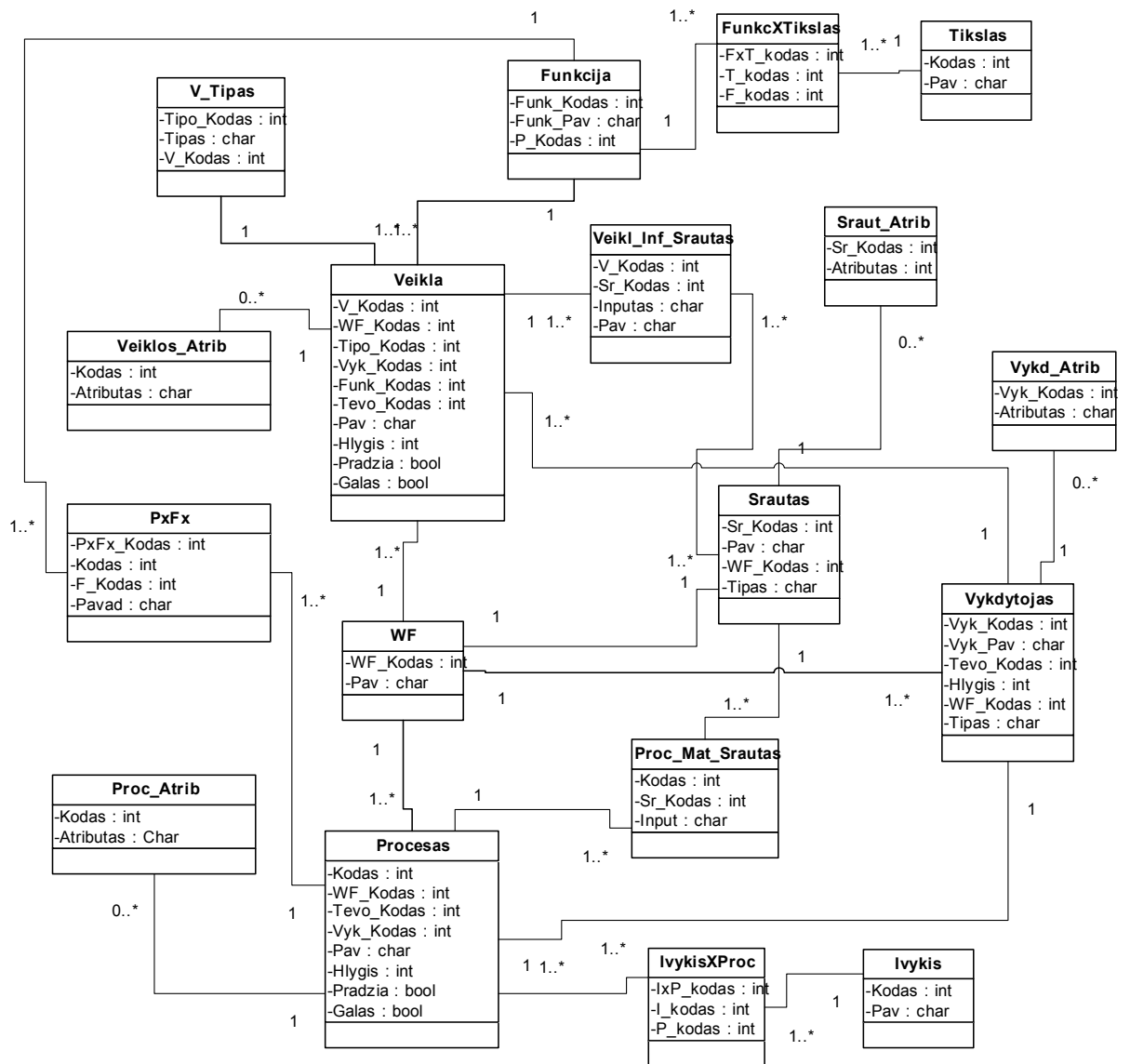
| Veiklos objektai | |
|-------------------------|---|
| <i>Klasifikacija</i> | Posistemė |
| <i>Apibrėžimas</i> | Posistemė, kurioje saugomi žinių bazės ir sugeneruotų UCM modelių duomenys. |
| <i>Atsakomybės</i> | Ši posistemė bus atsakinga už veiklos modelio duomenų saugojimą, duomenų apdorojimą pagal užklausas, sugeneruotų UCM modelių saugojimą. |
| <i>Struktūra</i> | Veiklos objektų posistemė susideda iš veiklos žinių bazės ir UCM modelių saugyklos. |
| <i>Sąveikavimas</i> | Posistemė sąveikauja su veiklos aptarnavimo sluoksnio posisteme. Komunikacijai naudojamos užklausos. |
| <i>Resursai</i> | Bendravimas vyksta per SQL užklausas.(<i>SQL transact</i> protokolas). |

4.2.2. Dalykinės srities klasių diagrama

CASE veiklos žinių bazės saugykloje kaupiamos kompiuterizuojamos dalykinės srities žinios, surenkamos taikant darbų sekų modelius. Jų pilnumas tikrinamas pagal veiklos metamodelis sudėtį. CASE veiklos žinių tvarkymo sistemos dalykinės srities klasių diagrama pateikta 11 paveiksle.

CASE veiklos duomenų bazės loginę schemą sudaro aštuoniolika tarpusavyje susietų lentelių. Pagrindinės lentelės yra *Procesas*, *Funkcija*, *PxFx* (proceso ir funkcijos sankirta), *Veikla*, *V_Tipas*, *Vykdytojas*, *Srautas*.

Lentelėje *Procesas* saugomos žinios apie materialius dalykinėje veiklos srityje vykstančius procesus, o lentelėje *Funkcija* – žinios apie dalykinėje veiklos srityje 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 veikla. Veiklos gali būti trijų tipų – interpretavimas, informacijos apdorojimas ir sprendimų priėmimas (IP) ir realizavimas, kurie nustatomi 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*.



11 pav. CASE žinių tvarkymo sistemos dalykinės srities klasių diagrama

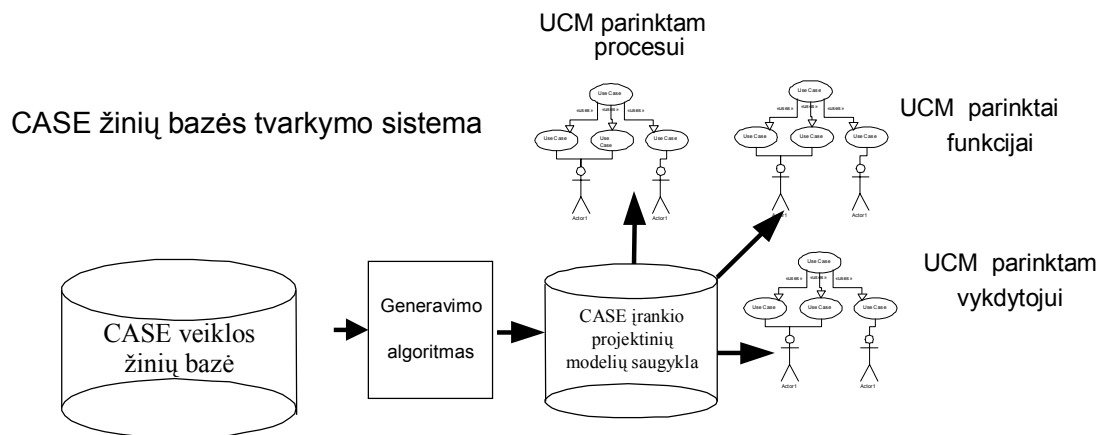
4.3. UCM generavimas

Teoriškai žiniomis grindžiamas UCM gali būti generuojamas kiekvienai modelio klasei, tačiau tikslingiausia ir informatyviausia panaudojimo atvejų modelį generuoti procesų, funkcijų, vykdytojų ir tikslų klasėms. 6 lentelėje pateikiami galimi žiniomis grindžiamo UCM generavimo variantai ir jo sudėtinės dalys.

6 lentelė. Žiniomis grindžiamo panaudojimo atvejų modelio generavimo variantai

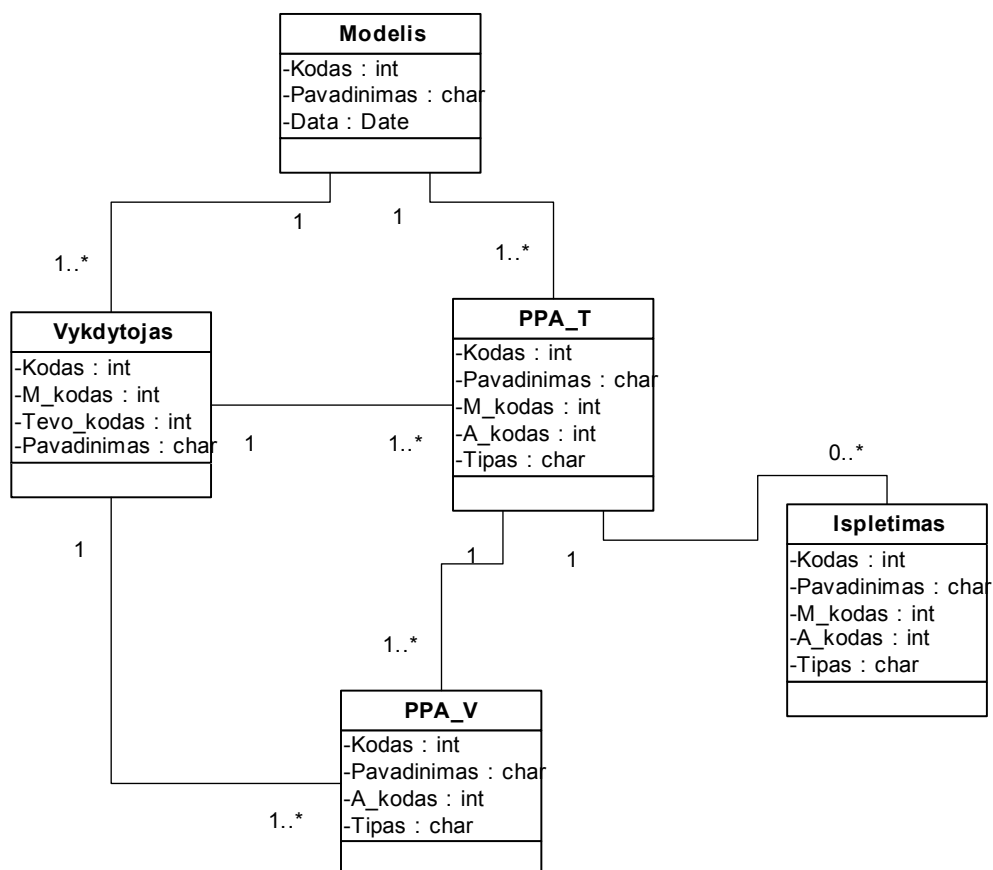
| Use Case pagal | Procesas | Materialus srautas | Funkcija | Informacinė veikla | Informacinis srautas | Vykdytojas |
|--------------------|----------|--------------------|----------|--------------------|----------------------|------------|
| Procesą | + | | | | | + |
| Funkciją | | | + | + | | + |
| Vykdytoją | + | | + | + | | + |
| Materialų srautą | + | + | | | | + |
| Informacijų srautą | | | + | + | + | + |
| Informacinę veiklą | | | + | + | | + |

Detaliau nagrinėsime UCM, skirtą konkrečiai veiklos funkcijai, procesui ir vykdytojui generavimo pagrindinius žingsnius (12 pav.).



12 pav. CASE žinių bazės tvarkymo sistema

4.3.1. UCM modelių saugykla



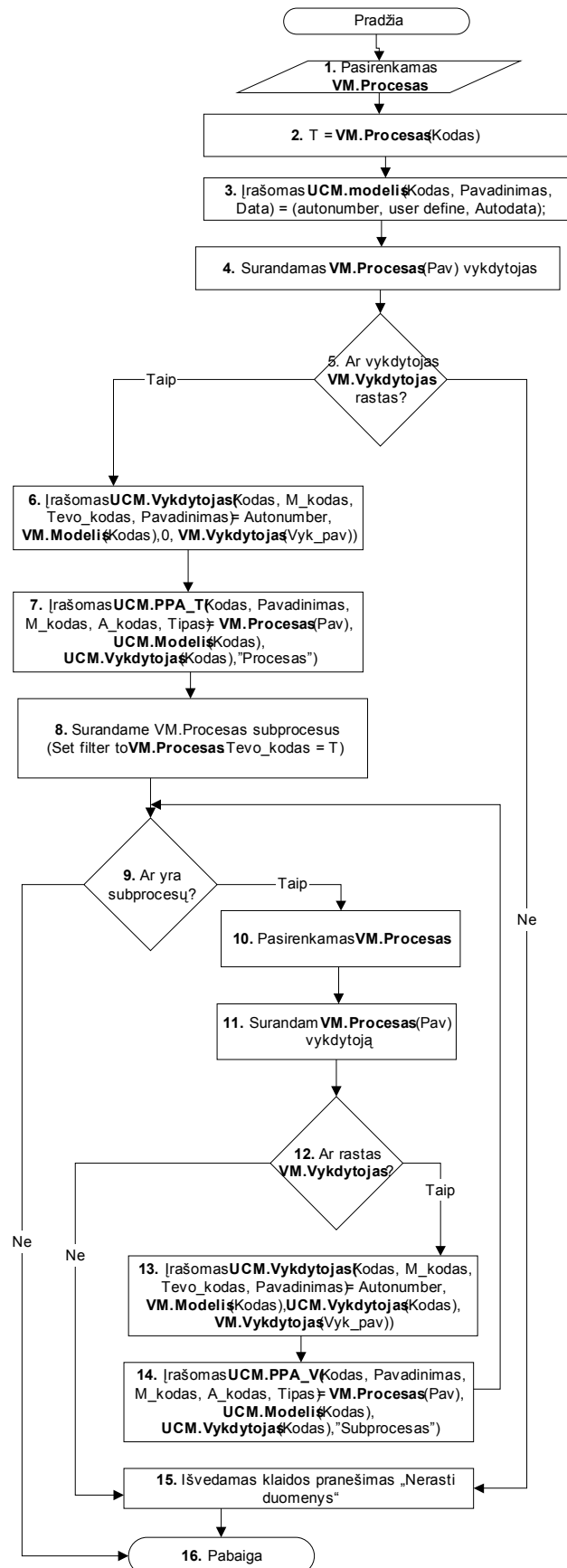
13 pav. UCM bazės dalykinė srities klasių diagrama

UCM bazės dalykinės srities klasių diagramos aprašymas:

- ❖ Modelis - saugomi vartotojo generuojami modeliai;
- ❖ Vykdytojas - saugomi UCM modelio vykdytojai;
- ❖ PPA_T - saugomi pagrindiniai panaudojimo atvejai;
- ❖ PPA_V - saugomi papildomi panaudojimo atvejai;
- ❖ Ispletimas - saugomi išplėtimo taškai.

4.3.2. Proceso UCM generavimas

4.3.2.1. Proceso UCM generavimo algoritmas



14 pav. Proceso UCM generavimo algoritmas

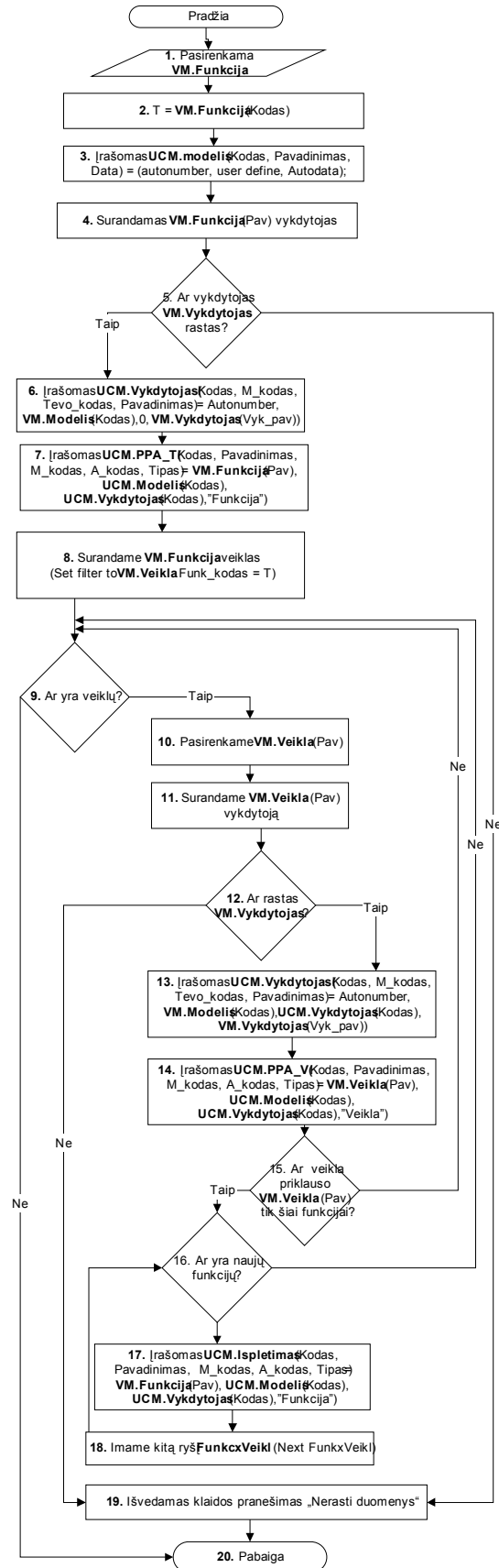
4.3.2.2. Išsamus proceso UCM generavimo algoritmo aprašymas

1. Pasirenkamas procesas kuriam bus generuojamas UCM modelis (**VM.Procesas(Pav)**);
2. Išsaugom proceso kodą (**T = VM.Procesas(Kodas)**);
3. Įrašomas naujas modelis **UCM.Modelis**
 - 3.1. **UCM.Modelis(Kodas) = autonumber;**
 - 3.2. **UCM.Modelis(Pavadinimas) = user define;**
 - 3.3. **UCM.Modelis(Data) = Autodata;**
4. Surandame pasirinkto **VM.Procesas(Pav)** vykdytoją;
5. Tikriname ar vykdytojas **VM.Vykdytojas** rastas:
 - 5.1. Jei vykdytojas rastas atliekame 6 žingsnį;
 - 5.2. Jei vykdytojas nerastas 15 žingsnis (Klaidos pranešimas);
6. Įrašomas naujas Vykdytojas **UCM.Vykdytojas**:
 - 6.1. **UCM.Vykdytojas(Kodas) = Autonumber;**
 - 6.2. **UCM.Vykdytojas(M_kodas) = UCM.Modelis(Kodas);**
 - 6.3. **UCM.vykdytojas(Tevo_kodas) = 0;**
 - 6.4. **UCM.vykdytojas(Pavadinimas) = VM.Vykdytojas(Vyk_pav);**
7. Įrašomas naujas **UseCase_P**:
 - 7.1. **UCM.UseCase_P(Kodas) = Autonumber;**
 - 7.2. **UCM.UseCase_P(Pavadinimas) = VM.Procesas(Pav);**
 - 7.3. **UCM. UseCase_P (M_kodas) = UCM.Modelis(Kodas);**
 - 7.4. **UCM. UseCase_P (A_kodas) = UCM.Vykdytojas(Kodas);**
 - 7.5. **UCM. UseCase_P (Tipas) = "Procesas";**
8. Surandame pasirinkto proceso **VM.Procesas** subprocesus
(**Set filter to VM.Procesas.Tevo_kodas=T**);
9. Tikriname ar yra subprocesų:
 - 9.1. Jei taip vykdomas 10 žingsnis;
 - 9.2. Jei ne vykdomas 16 žingsnis;
10. Pasirenkame subprocesą **VM.Procesas(Pav)**;
11. Surandame pasirinkto **VM.Procesas(Pav)** vykdytoją;
12. Tikriname ar vykdytojas **VM.Vykdytojas** rastas:
 - 12.1. Jei vykdytojas rastas atliekame 13 žingsnį;
 - 12.2. Jei vykdytojas nerastas 15 žingsnis (Klaidos pranešimas);
13. Įrašomas naujas Vykdytojas **UCM.Vykdytojas**:
 - 13.1. **UCM.Vykdytojas(Kodas) = Autonumber;**
 - 13.2. **UCM.Vykdytojas(M_kodas) = UCM.Modelis(Kodas);**

- 13.3. **UCM.vykdytojas(Tevo_kodas) = UCM.Vykdytojas(Kodas);**
- 13.4. **UCM.vykdytojas(Pavadinimas) = VM.Vykdytojas(Vyk_pav);**
- 14. Įrašomas naujas UseCase_C:
 - 14.1. **UCM.UseCase_C(Kodas) = Autonumber;**
 - 14.2. **UCM.UseCase_C(Pavadinimas) = VM.Procesas(Pav);**
 - 14.3. **UCM. UseCase_C (M_kodas) = UCM.Modelis(Kodas);**
 - 14.4. **UCM. UseCase_C (A_kodas) = UCM.Vykdytojas(Kodas);**
 - 14.5. **UCM. UseCase_C (Tipas) = “Subpocesas”;**
 - 14.6. Kartojamas 9 žingsnis;
- 15. Išvedamas klaidos pranešimas (“Nerasti duomenys”);
- 16. Pabaiga.

4.3.3. Funkcijos UCM generavimas

4.3.3.1. Funkcijos UCM generavimo algoritmas



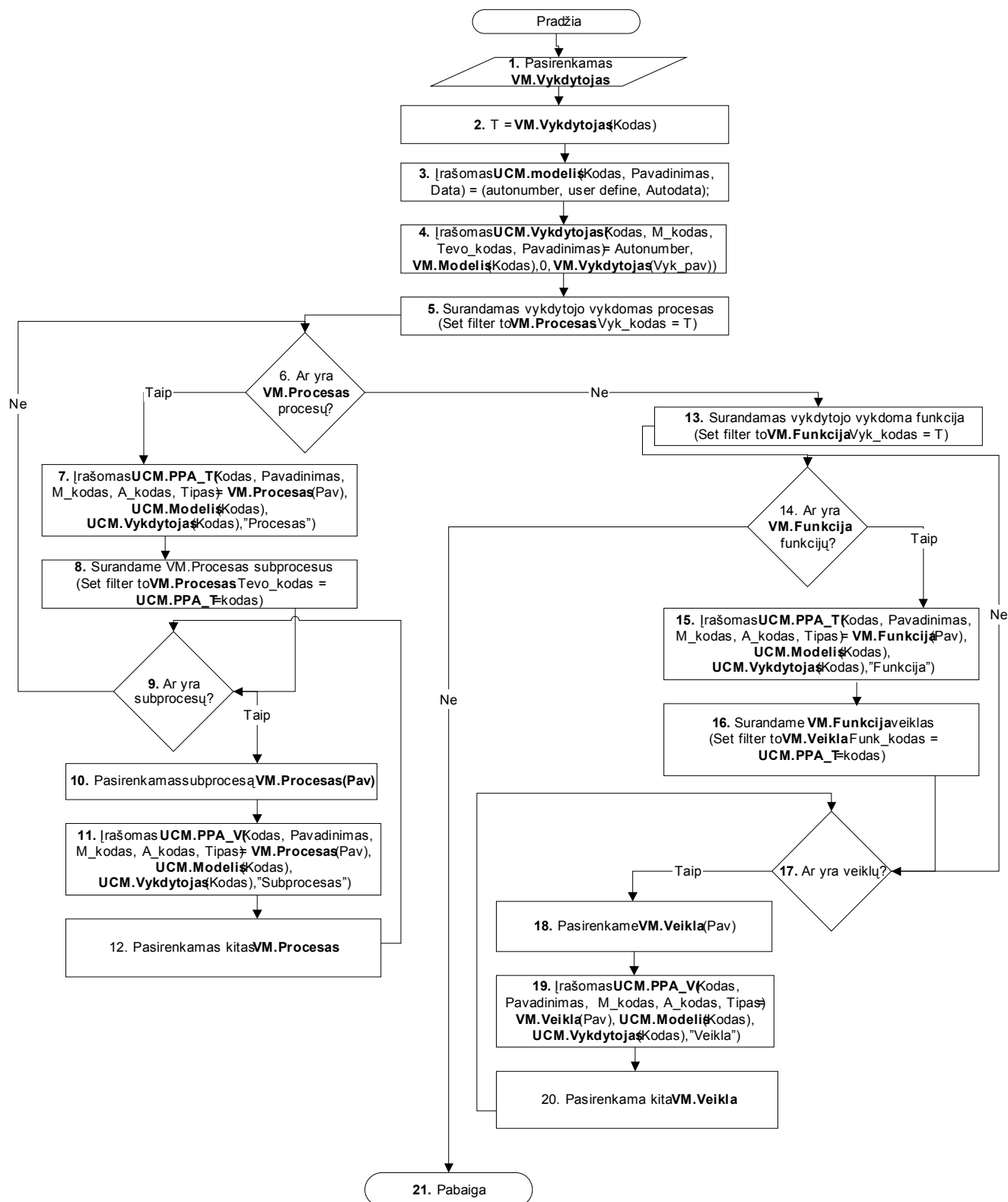
15 pav. Funkcijos UCM generavimo algoritmas

4.3.3.2. Išsamus funkcijos UCM generavimo algoritmo aprašymas

1. Pasirenkama funkcija, kurios atžvilgiu bus generuojamas UCM modelis (**VM.Funkcija(Pav)**);
2. Išsaugomas funkcijos kodas (**T = VM.Funkcija(Kodas)**);
3. Įrašomas naujas modelis **UCM.Modelis**
 - 3.1. **UCM.Modelis(Kodas) = autonumber;**
 - 3.2. **UCM.Modelis(Pavadinimas) = user define;**
 - 3.3. **UCM.Modelis(Data) = Autodata;**
4. Surandame pasirinktos **VM.Funkcija(Pav)** vykdytoją;
5. Tikriname ar vykdytojas **VM.Vykdytojas** rastas:
 - 5.1. Jei vykdytojas rastas atliekame 6 žingsnį;
 - 5.2. Jei vykdytojas nerastas 19 žingsnis (Klaidos pranešimas);
6. Įrašomas naujas Vykdytojas **UCM.Vykdytojas**:
 - 6.1. **UCM.Vykdytojas(Kodas) = Autonumber;**
 - 6.2. **UCM.Vykdytojas(M_kodas) = UCM.Modelis(Kodas);**
 - 6.3. **UCM.vykdytojas(Tevo_kodas) = 0;**
 - 6.4. **UCM.vykdytojas(Pavadinimas) = VM.Vykdytojas(Vyk_pav);**
7. Įrašomas naujas **UCM.PPA_T**:
 - 7.1. **UCM. UCM.PPA_T (Kodas) = Autonumber;**
 - 7.2. **UCM. UCM.PPA_T (Pavadinimas) = VM.Funkcija(Pav);**
 - 7.3. **UCM. UCM.PPA_T (M_kodas) = UCM.Funkcija(Kodas);**
 - 7.4. **UCM. UCM.PPA_T (A_kodas) = UCM.Vykdytojas(Kodas);**
 - 7.5. **UCM. UCM.PPA_T (Tipas) = "Funkcija";**
8. Surandame pasirinktos funkcijos **VM.Funkcija(Pav)** veiklas
(Set filter to VM.Veikla.Funk_kodas = T);
9. Tikriname ar yra veiklų:
 - 9.1. Jei taip vykdomas 10 žingsnis;
 - 9.2. Jei ne vykdomas 20 žingsnis;
10. Pasirenkame veiklą **VM.Veikla(Pav)**;
11. Surandame pasirinktos **VM.Veikla(Pav)** vykdytoją;
12. Tikriname ar vykdytojas **VM.Vykdytojas** rastas:
 - 12.1. Jei vykdytojas rastas atliekame 13 žingsnį;
 - 12.2. Jei vykdytojas nerastas 19 žingsnis (Klaidos pranešimas);
13. Įrašomas naujas Vykdytojas **UCM.Vykdytojas**
 - 13.1. **UCM.Vykdytojas(Kodas) = Autonumber;**

- 13.2. **UCM.Vykdytojas(M_kodas) = UCM.Modelis(Kodas);**
- 13.3. **UCM.vykdytojas(Tevo_kodas) = UCM.Vykdytojas(kodas);**
- 13.4. **UCM.vykdytojas(Pavadinimas) = VM.Vykdytojas(Vyk_pav);**
- 14. Įrašomas naujas **UCM.PPA_V**:
 - 14.1. **UCM. UCM.PPA_V (Kodas) = Autonumber;**
 - 14.2. **UCM. UCM.PPA_V (Pavadinimas) = VM.Veikla(Pav);**
 - 14.3. **UCM. UCM.PPA_V (M_kodas) = UCM.Veikla(Kodas);**
 - 14.4. **UCM. UCM.PPA_V (A_kodas) = UCM.Vykdytojas(Kodas);**
 - 14.5. **UCM. UCM.PPA_V (Tipas) = “Veikla”;**
- 15. Tikriname ar veikla VM.Veikla(Pav) priklauso tik vienai funkcijai:
 - 15.1. Jei taip vykdomas 16 žingsnis;
 - 15.2. Jei ne vykdomas 9 žingsnis;
- 16. Tikriname ar yra naujų funkcijų, kurioms priklauso **VM.Veikla(Pav)**:
 - 16.1. Jei taip vykdomas 17 žingsnis;
 - 16.2. Jei ne vykdomas 9 žingsnis;
- 17. Įrašomas **UCM.Ispletimas**
 - 17.1. **UCM.Ispletimas(Kodas) = Autonumber;**
 - 17.2. **UCM.Ispletimas(Pavadinimas) = VM.Funkcija(Pav);**
 - 17.3. **UCM.Ispletimas(M_kodas) = UCM.Modelis(Kodas);**
 - 17.4. **UCM.Ispletimas(A_kodas) = UCM.Vykdytojas(Kodas);**
 - 17.5. **UCM.Ispletimas(M_kodas) = “Funkcija”;**
- 18. Imamas kitas ryšys **FunkcxVeikl (Next FunkcxVeikl)** ir vykdomas 16 žingsnis;
- 19. Išvedamas klaidos pranešimas (“Nerasti duomenys”);
- 20. Pabaiga.

4.3.4. Vykdytojo UCM generavimas



16 pav. Vykdytojo UCM generavimo algoritmas

4.3.4.1. Vykdytojo UCM generavimo algoritmas

1. Pasirenkamas vykdytojas, kurios atžvilgiu bus generuojamas UCM modelis
(VM.Vykdytojas(Pav));
2. **T = VM.Vykdytojas(Kodas);**
3. Įrašomas naujas modelis **UCM.Modelis:**
 - 3.1. **UCM.Modelis(Kodas) = autonumber;**
 - 3.2. **UCM.Modelis(Pavadinimas) = user define;**
 - 3.3. **UCM.Modelis(Data) = Autodata;**
4. Įrašomas naujas Vykdytojas **UCM.Vykdytojas:**
 - 4.1. **UCM.Vykdytojas(Kodas) = Autonumber;**
 - 4.2. **UCM.Vykdytojas(M_kodas) = UCM.Modelis(Kodas);**
 - 4.3. **UCM.vykdytojas(Tevo_kodas) = 0;**
 - 4.4. **UCM.vykdytojas(Pavadinimas) = VM.Vykdytojas(Vyk_pav);**
5. Surandame pasirinkto **VM.Vykdytojo(Pav)** valdomą procesą;
6. Tikriname ar yra procesų:
 - 6.1. Jei procesų yra atliekame 7 žingsnį;
 - 6.2. Jei procesų nėra atliekame 13 žingsnį;
7. Įrašomas naujas Procesas **UCM.UseCase_P:**
 - 7.1. **UCM.UseCase_P(Kodas) = Autonumber;**
 - 7.2. **UCM.UseCase_P(Pavadinimas) = VM.Procesas(Pav);**
 - 7.3. **UCM. UseCase_P (M_kodas) = UCM.Modelis(Kodas);**
 - 7.4. **UCM. UseCase_P (A_kodas) = UCM.Vykdytojas(Kodas);**
 - 7.5. **UCM. UseCase_P (Tipas) = “Procesas”;**
8. Surandame įrašyto proceso **VM.Procesas** subprocesus
(Set filter to VM.Procesas.Tevo_kodas = UCM.PPA_T(kodas));
9. Tikriname ar yra subprocesų:
 - 9.1. Jei taip vykdomas 10 žingsnis;
 - 9.2. Jei ne vykdomas 6 žingsnis;
10. Pasirenkame subprocesą **VM.Procesas(Pav);**
11. Įrašomas naujas **UseCase_C:**
 - 11.1. **UCM.UseCase_C(Kodas) = Autonumber;**
 - 11.2. **UCM.UseCase_C(Pavadinimas) = VM.Procesas(Pav);**
 - 11.3. **UCM. UseCase_C (M_kodas) = UCM.Modelis(Kodas);**
 - 11.4. **UCM. UseCase_C (A_kodas) = UCM.Vykdytojas(Kodas);**
 - 11.5. **UCM. UseCase_C (Tipas) = “Subpocesas”;**

12. Pasirenkamas kitas **VM.Procesas** ir kartojame 9 žingsnį;
13. Surandame pasirinkto **VM.Vykdytojo(Pav)** valdomą funkciją;
14. Tikriname ar yra funkcijų:
 - 14.1. Jei funkcijų yra atliekame 15 žingsnį;
 - 14.2. Jei funkcijų nėra atliekame 21 žingsnį;
15. Įrašomas naują Funkcija **UCM.UseCase_P**:
 - 15.1. **UCM.UseCase_P(Kodas) = Autonumber;**
 - 15.2. **UCM.UseCase_P(Pavadinimas) = VM.Funkcija(Pav);**
 - 15.3. **UCM. UseCase_P (M_kodas) = UCM.Modelis(Kodas);**
 - 15.4. **UCM. UseCase_P (A_kodas) = UCM.Vykdytojas(Kodas);**
 - 15.5. **UCM. UseCase_P (Tipas) = “Funkcija”;**
16. Surandame pasirinktos funkcijos **VM.Funkcija(Pav)** veiklas
(Set filter to **VM.Veikla.Funk_kodas = UCM.PPA_T(kodas)**);
17. Tikriname ar yra veiklų:
 - 17.1. Jei taip vykdomas 18 žingsnis;
 - 17.2. Jei ne vykdomas 14 žingsnis;
18. Pasirenkame veiklą **VM.Veikla(Pav)**;
19. Įrašomas naujas **UseCase_C**:
 - 19.1. **UCM.UseCase_C(Kodas) = Autonumber;**
 - 19.2. **UCM.UseCase_C(Pavadinimas) = VM.Veikla(Pav);**
 - 19.3. **UCM. UseCase_C (M_kodas) = UCM.Veikla(Kodas);**
 - 19.4. **UCM. UseCase_C (A_kodas) = UCM.Vykdytojas(Kodas);**
 - 19.5. **UCM. UseCase_C (Tipas) = “Veikla”;**
20. Pasirenkama kita **VM.Veikla** ir kartojamas 17 žingsnis;
21. Pabaiga.

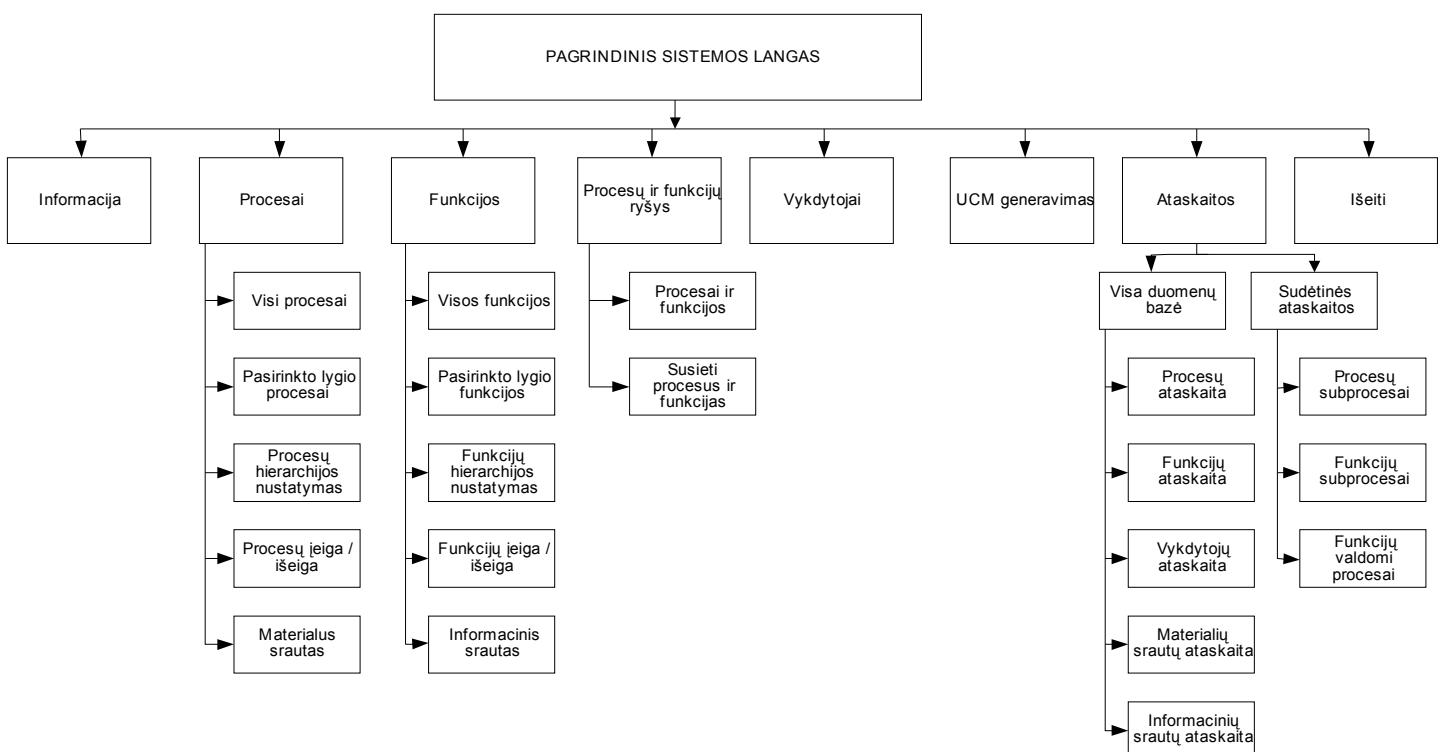
5. Eksperimentinė dalis

5.1. Vartotojo sąsajos projektas

Vartotojo ir sistemos sąsaja yra ženkli visos architektūros dalis. Vartotojo sąsaja turi būti patogi, lengvai naudojama bei suprantama. 17 – me paveiksle pateiktas vartotojo sąsajos modelis, nusakantis sistemos navigavimo planą. Pagrindinis meniu:

- ✓ Informacija;
- ✓ Procesai;
- ✓ Funkcijos;
- ✓ Procesų ir funkcijų ryšys;
- ✓ Vykdytojai;
- ✓ UCM generavimas;
- ✓ Ataskaitos;
- ✓ Išėiti.

Pasirinkus tam tikrą meniu punktą, atidaromas atitinkamas sistemos langas (17 pav.).



17 pav. Vartotojo sąsaja struktūrinė schema

5.2. Vartotojo atmintinė

5.2.1. Sistemos paskirtis

„CASE veiklos žinių bazės tvarkymo sistema” naudojama IS projektavimui, kuriamos sistemos pilnumo tikrinimui, reinžinerijai bei mokymo tikslams. Sukurtąją sistemą galima taikyti praktikoje tobulinant CASE sistemas, papildant jas veiklos žinių baze

Sistemos vartotojais turėtų būti specialistai arba apmokyti vartotojai, kurie atlieka organizacijų projektavimo darbus. Jie turėtų būti susipažinę su tokiais elementariais organizacijos elementais kaip funkcija, procesas, srautas, veiklos taisyklė, žinoti jų tarpusavio skirtumus. Taigi sistemos vartotojais galėtų būti:

- ❖ Analitikai;
- ❖ Projektuotojai;
- ❖ Studentai.

5.2.2. Sistemos galimybės

Mūsų kuriama IS papildo CASE sistemų funkcionalumą, jos tikslas apimti informacinės sistemos (IS) kūrimo gyvavimo ciklą, įskaitant veiklos taisyklių panaudojimą.

Žinių bazės tvarkymo sistemos pagrindinės funkcijas:

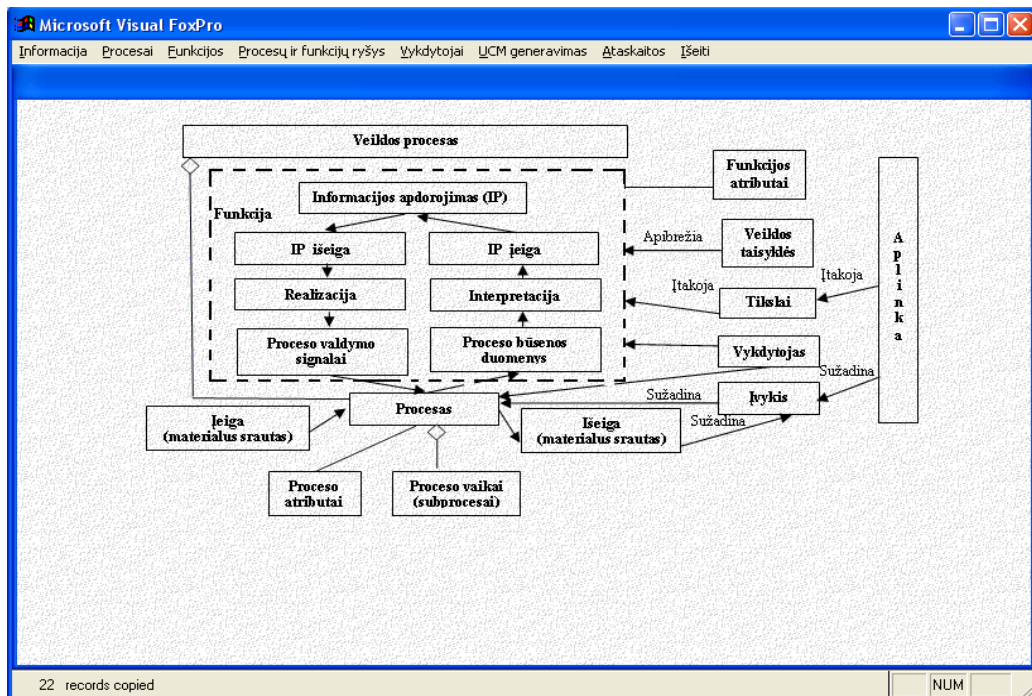
- ❖ Veiklos procesų, funkcijų įvedimas ir koregavimas;
- ❖ Procesų ir funkcijų susiejimas;
- ❖ Materialių, informacinių srautų įvedimas ir koregavimas;
- ❖ Vykdytojų įvedimas, koregavimas, susiejimas su funkcijomis ir procesais;
- ❖ UCM generavimas;
- ❖ Ataskaitų sudarymas.

5.2.3. Darbas su sistema

- ❖ Sistema paleidžiama atidarant „Ziniu_baze.exe“ vykdomąją bylą;
- ❖ Sistemos meniu sudaro aštuoni meniu punktai:
 - „Informacija“;
 - „Procesai“;
 - „Funkcijos“;
 - „Procesų ir funkcijų ryšys“;
 - „Vykdytojai“;
 - „UCM“ generavimas;
 - „Ataskaitos“;
 - „Išeiti“.
- ❖ Sistemos duomenys (procesai, funkcijos, vykdytojai) turi hierarchinę struktūrą.

5.3. Detalus sistemos aprašymas

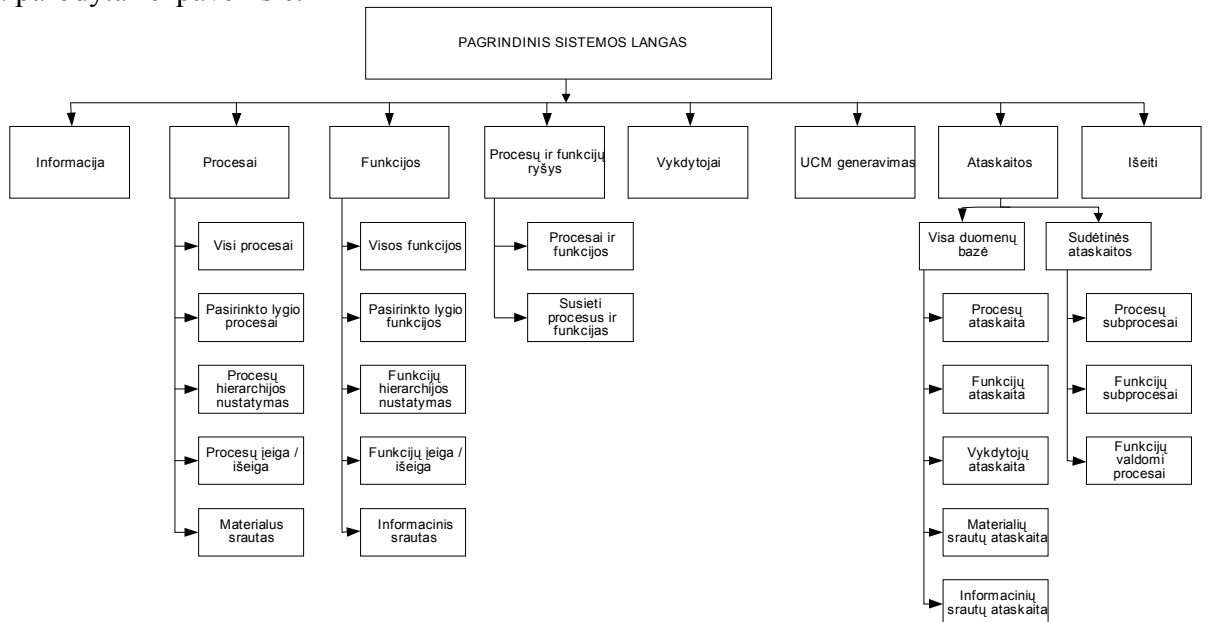
Paleidus programą monitoriaus ekrane išvystame pagrindinį sistemos langą, kuris pateikiamas sekančiame paveiksle (18 pav.). Šio lango schema parodo pagrindines sistemos esybes, taip pat ji tarnauja kaip vartotojo darbo planas (parodo su kuriomis esybėmis vartotojas dirba konkrečiu laiko momentu).



18 pav. Pagrindinis sistemos langas

5.3.1. Sistemos sandara

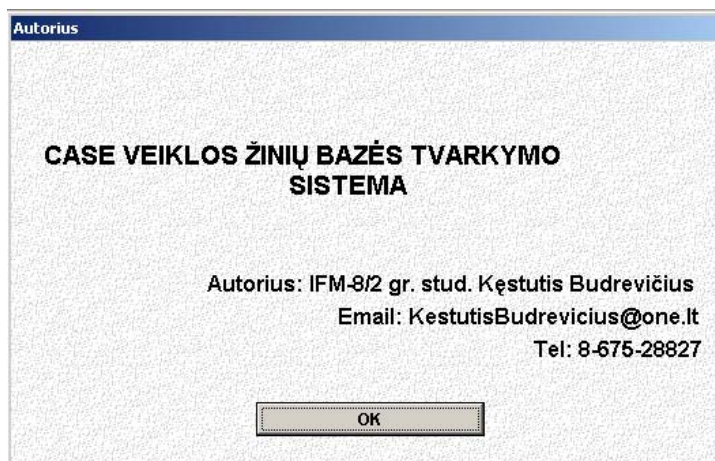
Sistemos meniu sudaro aštuoni punktai: „Informacija“, „Procesai“, „Funkcijos“, „Procesų ir funkcijų ryšys“, „Vykdotojai“, „UCM generavimas“, „Ataskaitos“, „Išeiti“. Sistemos struktūrinė schema parodyta 19 paveiksle.



19 pav. Sistemos struktūrinė schema

5.3.2. Informacija

Pasirinkus meniu punktą „Informacija“ sistema pateikia programos langą „Autorius“ (20 pav.). Jame pateikiama trumpa informacija apie sistemą, jos kūrėją: sistemos, pavadinimas, atlikėjo vardas, pavardė, elektroninis paštas ir telefonas. Norint šį langą uždaryti reikia paspausti mygtuką „OK“.



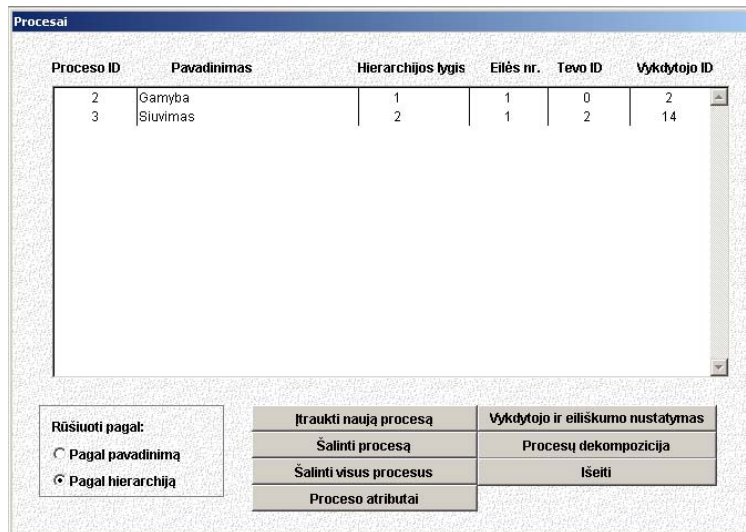
20 pav. Sistemos langas „Autorius“

5.3.3. Procesai

Pasirinkus meniu punktą „Procesai“ sistema pateikia sekantį meniu, sudarytą iš tokių dalių: „Visi procesai“ , „Pasirinkto lygio procesai“, „Procesų dekompozicija“, „Procesų įeiga/išeiga“ ir „Materialus srautas“.

5.3.3.1. Visi procesai

Pasirinkus pirmąjį meniu etapą – „Visi procesai“ sistema pateikia sekantį programos langą (21 pav.):



| Proceso ID | Pavadinimas | Hierarchijos lygis | Eilės nr. | Tevo ID | Vykdytojo ID |
|------------|-------------|--------------------|-----------|---------|--------------|
| 2 | Gamyba | 1 | 1 | 0 | 2 |
| 3 | Siuvimas | 2 | 1 | 2 | 14 |

Rūšiuoti pagal:

- Pagal pavadinimą
- Pagal hierarchiją

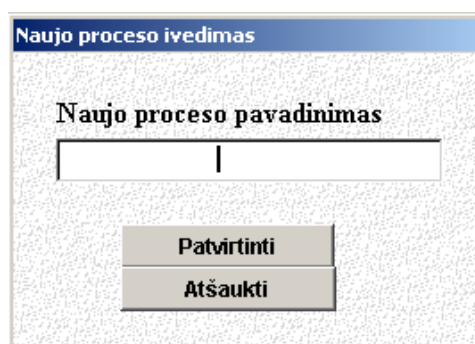
| | |
|------------------------|-----------------------------------|
| Įtraukti naują procesą | Vykdytojo ir eiliškumo nustatymas |
| Šalinti procesą | Procesų dekompozicija |
| Šalinti visus procesus | Išėti |
| Proceso atributai | |

21 pav. Sistemos langas „Procesai“

21 paveiksle pateikto lango funkcijos:

- ❖ Įvesti naujus procesus;
- ❖ Šalinti procesus;
- ❖ Įvesti, pašalinti proceso atributus;
- ❖ Nustatyti procesų vykdytojus ir eilės numerius;
- ❖ Dekomponuoti procesus.

Paspaudus mygtuką „**Įtraukti naują procesą**“ atidaromas langas kuriame reikia įvesti naujo proceso pavadinimą (22 pav.). Pavadinimas negali būti ilgesnis nei 40 simbolių, jame gali būti skaičiai bei simboliai.



Naujo proceso įvedimas

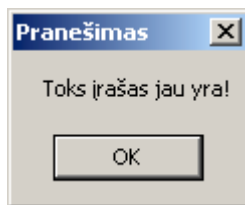
Naujo proceso pavadinimas

Patvirtinti

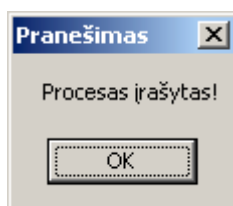
Atšaukti

22 pav. Sistemos langas „Naujo proceso įvedimas“

Įvedus proceso pavadinimą pasirenkame mygtuką „**Patvirtinti**“. Jei toks procesas jau egzistuoja, sistema apie tai informuoja vartotoją išvesdama pranešimą (23 pav.), taip pat sistema informuoja kai procesas įrašomas į duomenų bazę (24 pav.).



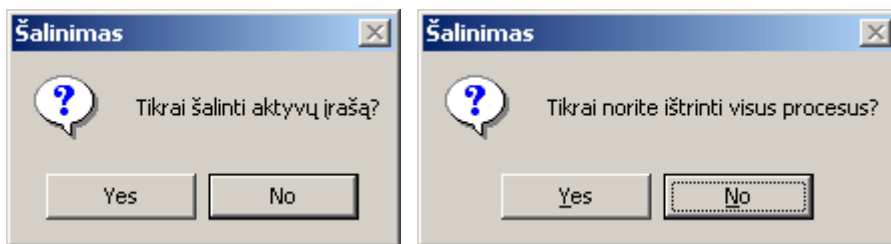
23 pav. Pranešimas apie egzistuojantį įrašą.



24 pav. Pranešimas apie įrašytą procesą.

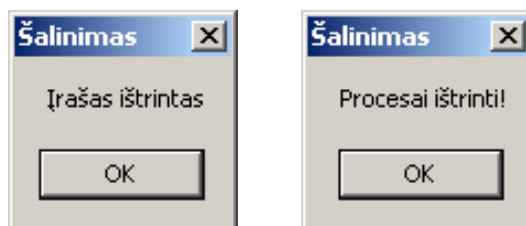
Norint atšaukti naujo proceso įvedimą pasirenkame mygtuką „**Atšaukti**“ .

Norėdami pašalinti pažymėtą procesą pasirenkame mygtuką „**Šalinti procesą**“, jei norime pašalinti visus egzistuojančius procesus pasirenkame mygtuką „**Šalinti visus procesus**“. Sistema paprašo vartotojo patvirtinti veiksmus (24 pav.).



25 pav. Sistemos pranešimai apie šalinimo veiksmų patvirtinimą.

Vartotojui atlikus šalinimo veiksmus, priklausomai nuo to ar buvo pašalintas vienas procesas ar visi procesai sistema pateikia sekančius pranešimus (26 pav.).



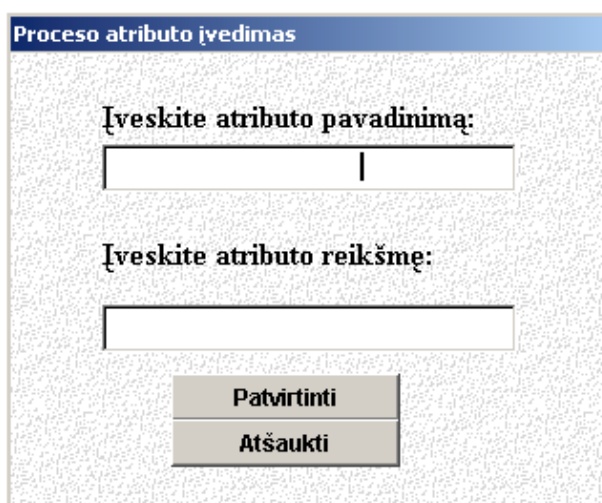
26 pav. Pranešimai apie atliktus pašalinimus

Norint įvesti ar pašalinti proceso papildomus duomenis (atributus) reikia pasirinkti mygtuką „**Proceso atributai**“. Atidaromas naujas langas, kuriame galima įvesti bei pašalinti atributus (27 pav.).



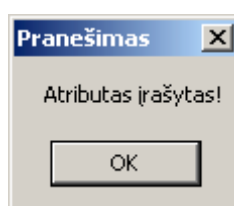
27 pav. Sistemos langas „Proceso atributai“

Norint įvesti atributą pasirenkame mygtuką „**Įterpti naują**“. Sistemos 28 paveikslo lange įvedame atributo pavadinimą, bei jo reikšmę.



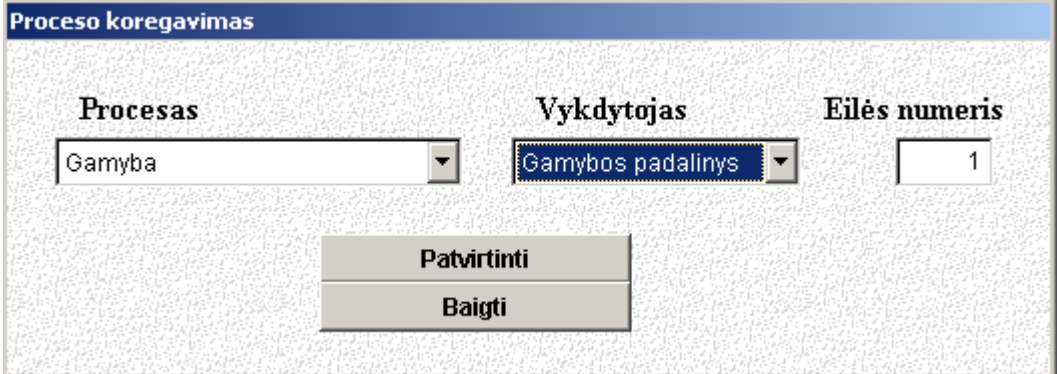
28 pav. Sistemos langas „Proceso atributai“

Įvedus atributo pavadinimą, bei jo reikšmę spaudžiame mygtuką „**Patvirtinti**“. Jei toks atributo pavadinimas jau egzistuoja, sistema apie tai informuoja vartotoją išvesdama pranešimą, taip pat sistema informuoja kai proceso atributas įrašomas į duomenų bazę (29 pav.).



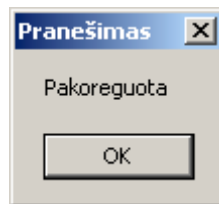
29 pav. Pranešimas apie įrašytą atributą.

Taip pat galima nustatyti procesų vykdytoją bei jo vykdymo eilės numerį. Tai atliekama pasirinkus mygtuką „Vykdytojo ir eiliškumo nustatymas“. Žemiau pavaizduotame lange pasirenkame procesą, nurodome jo vykdytoją bei proceso vykdymo eilės numerį ir spaudžiame mygtuką patvirtinti (30 pav.).



30 pav. Sistemos langas „Proceso koregavimas“

Sistemai atlikus koregavimus apie tai informuojamas vartotojas sistemai išvedant pranešimą (31 pav.).



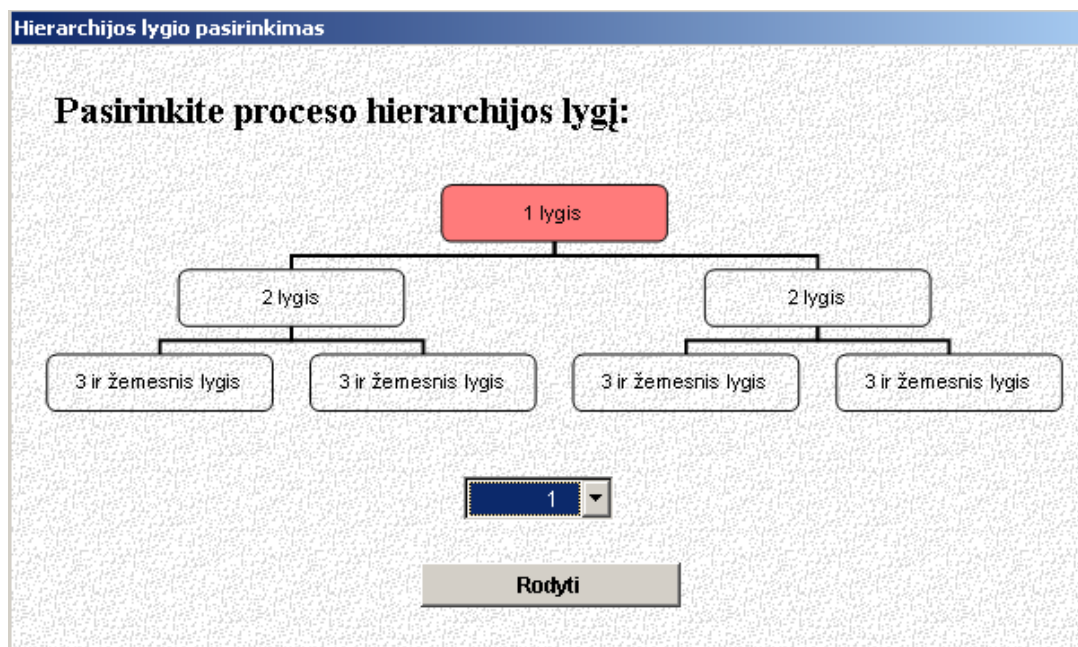
31 pav. Pranešimas apie atliktus pakeitimus

Taip pat galima pasirinkti procesų dekomponavimą (hierarchijos lygio nustatymas). Pasirinkus šį mygtuką sistemą atidaro langą, kuriame galima nustatinėti procesų hierarchijos lygius ir jų priklausomybę kitiems procesams.

Pasirinkus mygtuką „Išeiti“ sistema uždaro langą.

5.3.3.2. Pasirinkto lygio procesai

Pasirinkus sekantį procesų meniu punktą „Pasirinkto lygio procesai“ galima peržiūrėti tik tam tikro lygio procesus. Atsidariusiame lange galima pasirinkti konkretų hierarchijos lygį (sistema leidžia pasirinkti tik tuo metu esančius hierarchijos lygius).



32 pav. Proceso hierarchijos lygio pasirinkimas

Pasirinkus konkretų hierarchijos lygį, sistema išveda langą, kuriame matomi tik pasirinkto lygio procesai (33 pav.).

Procesai

ID: 2 Aukštesnio proceso ID: 0
Pavadinimas: Gamyba Vykdytojo numeris: 1

1 lygio procesai:

| Proceso ID | Proceso pavadinimas | Vykdytojo numeris | Vykdytojo numeris |
|------------|---------------------|-------------------|-------------------|
| 2 | Gamyba | 1 | 2 |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

<< < > >>

Redagavimo režimas
Rodyti proceso "tėvą"
Išėiti

Išterpti **Trinti**

33 pav. Sistemos langas „Pasirinkto lygio procesai“

Šiame lange galima atlikti tokias funkcijas:

- Įvesti naujus procesus;
- Šalinti esamus procesus;
- Redaguoti proceso duomenis.

Pasirinkus mygtuką „**Įterpti**“ sistema leidžia įvesti proceso pavadinimą. Pasirinkus mygtuką „**Trinti**“ paprašoma patvirtinti savo pasirinkimą.

Pasirinkus mygtuką „Redagavimo režimas“ sistema leidžia koreguoti 34 paveiksle pažymėtus lango laukų reikšmes (pavadinimą ir vykdytojo numerį). Šiame režime neleidžiama atlikti kitų veiksmų: naujo proceso įvedimo, šalinimo. Norint juos atlikti reikia pasirinkti mygtuką „**Išeiti iš redagavimo režimo**“.

Procesai

ID: Aukštesnio proceso ID:

Pavadinimas Vykdytojo numeris

1 lygio procesai:

| Proceso ID | Proceso pavadinimas | Vykdytojo numeris | Vykdymo numeris |
|------------|---------------------|-------------------|-----------------|
| 2 | Gamyba | 1 | 2 |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Navigation buttons: << < > >>

Buttons: Įterpti Trinti Išeiti iš redagavimo režimo Rodyti proceso "tėvą" Išeiti

34 pav. Sistemos langas „Pasirinkto lygio procesai“ – redagavimo režimas

5.3.3.3. Procesų dekompozicija

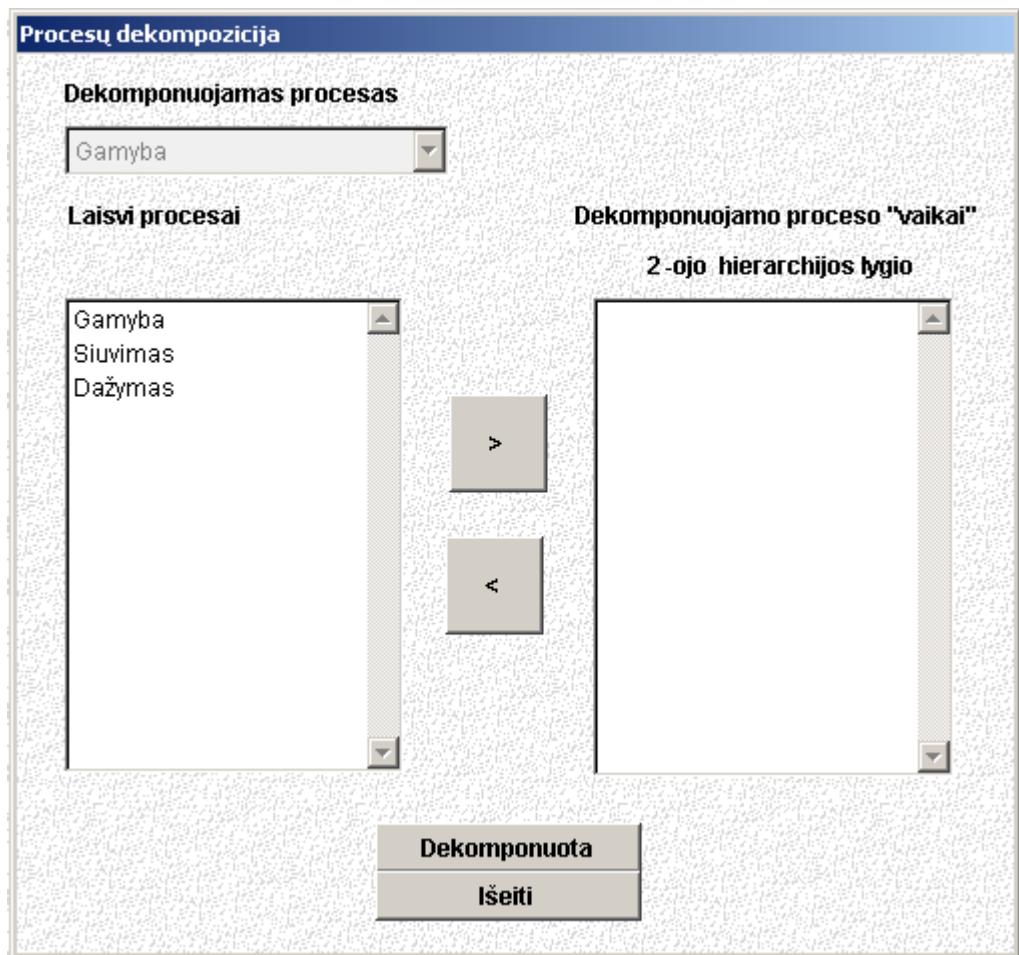
Pasirinkus sekantį procesų meniu punktą „Procesų dekompozicija“ sistema atidaro langą, kuriame galima nustatinėti procesų hierarchijos lygius ir jų priklausomybę kitiems procesams (35pav.).



35 pav. Sistemos langas „Procesų dekompozicija“

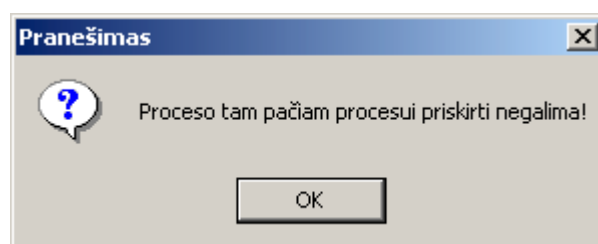
Šiame (35 pav.) lange pasirenkame dekomponuojamą procesą. Pasirinkus procesą, sistema išveda du sąrašus (36 pav.). Viename matomi „laisvi“ procesai, kuriuos galima priskirti pasirinktam (dekomponuojamam) procesui, kitame – pasirinkto proceso subprocesai (jam priklausantys procesai). Iš vieno sąrašo į kitą procesai perkeliami panaudojus klavišus „<“ ir „>“. Kai atliekame reikiamus veiksmus su pasirinktu (dekomponuojamu) procesu spaudžiame mygtuką „**Dekomponuota**“.

Atlikus šį veiksma galime pasirinkti kitą dekomponuojamą procesą.



36 pav. Sistemos langas „Procesų dekompozicija“ – procesų priskyrimas

Jei bandome priskirti pasirinktą procesą sau pačiam sistema informuoja vartotoją, kad to daryti negalima (37 pav.).



37 pav. Pranešimas apie negalimą veiksmą

5.3.3.4. Procesų įeiga/išeiga

Pasirinkus sekantį procesų meniu punktą „Procesų įeiga/išeiga“ sistema atidaro langą, kuriame galima priskirti pasirinktam procesui įeigą ir išeigą (materialius srautus).

| Materialaus srauto pavadinimas | Tipas |
|--------------------------------|--------|
| Žaliavos | Išeiga |
| Žaliavų kiekis | Įeiga |
| | |
| | |
| | |
| | |

38 pav. Sistemos langas „Procesų įeiga/išeiga“

Pasirinkus mygtukus „Priskirti įeigą“ ar „Priskirti išeigą“ sistema išveda langą, kuriame galima pasirinkti materialaus srauto įeigą ar išeigą (priklausomai koks mygtukas buvo paspaustas) .

Pasirinkite materialų srautą:

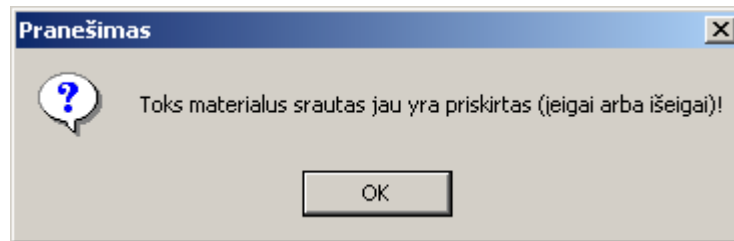
Išmatuota medžiaga

Priskirti

Išeiti

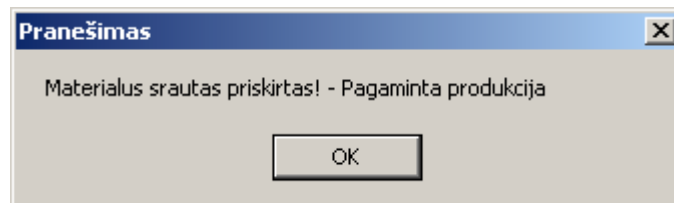
39 pav. Sistemos langas „Materialaus srauto pasirinkimas“

Jei buvo pasirinktas jau priskirtas srautas, sistema išveda pranešimą, kad toks materialus srautas šiam procesui jau yra priskirtas (40 pav.).



40 pav. Pranešimas apie jau priskirtą srautą

Jei šis srautas dar nebuvo priskirtas pasirinktam procesui, tai sistema praneša apie sėkmingą veiksmą (41 pav.).

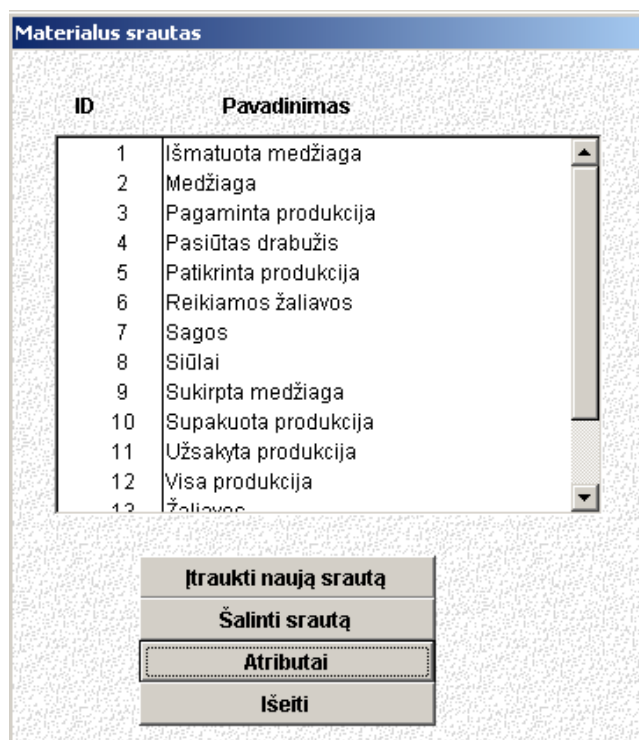


41 pav. Pranešimas apie sėkmingą veiksmą

Pasirinkus mygtuką „Šalinti pažymėtą srautą“ sistema paprašo patvirtinti savo pasirinkimą. Pasirinkus mygtuką „Išeiti“ sistema uždaro „Procesų įeigos/išeigos“ langą.

5.3.3.5. Materialus srautas

Pasirinkus sekantį procesų meniu punktą „Materialus srautas“ sistema atidaro langą, kuriame galima atlikti veiksmus su materialiu srautu (42 pav.).



42 pav. Sistemos langas „Materialus srautas“

Šiame lange galima atlikti tokius veiksmus:

- Įvesti naujus srautus;
- Šalinti esamus srautus;
- Įvesti/ pašalinti srauto atributus.

Paspaudus mygtuką „**Įtraukti naują srautą**“ atidaromas langas kuriame reikia įvesti naujo srauto pavadinimą. Pavadinimas negali būti ilgesnis nei 40 simbolių, jame gali būti skaičiai bei simboliai. Naujo srauto įvedimas analogiškas proceso pavadinimo įvedimui.

Pasirinkus mygtuką „**Šalinti srautą**“ paprašoma patvirtinti savo pasirinkimą. Įrašo šalinimo veiksmai analogiški proceso šalinimui.

Norint įvesti ar pašalinti srauto papildomus duomenis (atributus) reikia pasirinkti mygtuką „**Atributai**“. Atidaromas naujas langas, kuriame galima įvesti bei pašalinti atributus. Veiksmai analogiški proceso atributų įvedimui, bei šalinimui.

Pasirinkus mygtuką „**Išeiti**“ sistema uždaro „Materialių srautų“ langą.

5.3.4. Funkcijos

Pasirinkus meniu punktą „Funkcijos“ sistema pateikia sekantį meniu, sudarytą iš sekančių etapų: „Visos funkcijos“, „Pasirinkto lygio funkcijos“, „Funkcijų dekompozicija“, „Funkcijų įeiga/išeiga“ ir „Informacinis srautas“.

5.3.4.1. Visos funkcijos

Pasirinkus pirmąjį „Funkcijų“ meniu etapą – „Visos funkcijos“ sistema pateikia sekantį programos langą, kuris pavaizduotas 43 paveiksle.

| Funkcijos ID | Pavadinimas | Hierarchijos lygis | Eiliškumas | "Tėvinio" proceso ID | Vykdymo ID. |
|--------------|--------------------------------|--------------------|------------|----------------------|-------------|
| 7 | Užsakymo priėmimas | 1 | 1 | 0 | 18 |
| 6 | Užsakymo patvirtinimas | 1 | 2 | 0 | 18 |
| 12 | Žaliavų užsakymas | 1 | 3 | 0 | 20 |
| 3 | Atsiskaitymas už žaliavas | 1 | 4 | 0 | 1 |
| 4 | Kokybės tikrinimas | 1 | 5 | 0 | 4 |
| 9 | Užsakymo sutikrinimas | 1 | 6 | 0 | 12 |
| 19 | Atsiskaitymas su gamintoju | 1 | 7 | 0 | 1 |
| 13 | Matavimo tikslumo tikrinimas | 2 | 1 | 4 | 5 |
| 14 | Sukirptos madžiagos tikrinimas | 2 | 2 | 4 | 4 |
| 16 | Sutvarkymo patikrinimas | 2 | 3 | 4 | 7 |
| 5 | Susitikimas su tiekėjumi | 2 | 1 | 12 | 17 |
| 11 | Žaliavų kokybės išdėstymas | 2 | 2 | 12 | 16 |
| 10 | Žaliavų išsirinkimas | 2 | 3 | 12 | 16 |
| 8 | Užsakymo sudarymas | 2 | 4 | 12 | 19 |
| 18 | Atlikto darbo tikrinimas | 3 | 1 | 16 | 3 |

Rūšiuoti pagal:

- Pagal pavadinimą
- Pagal hierarchiją

Įtraukti naują funkciją

Šalinti funkciją

Šalinti visas funkcijas

Funkcijos atributai

Vykdymo ir eiliškumo nustatymas

Funkcijų dekompozicija

Išeiti

43 pav. Sistemos langas „Funkcijos“

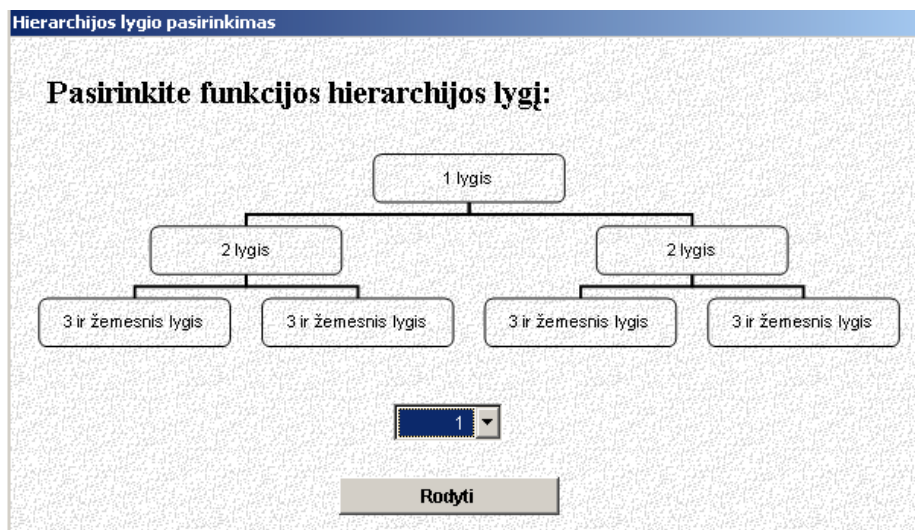
42 paveiksle pateikto lango funkcijos:

- Įvesti naujas funkcijas;
- Šalinti funkcijas;
- Įvesti, pašalinti funkcijos atributus;
- Nustatyti funkcijų vykdytojus ir eilės numerius;
- Dekomponuoti funkcijas.

Kaip matome šis meniu etapas yra analogiškas „Procesų“ meniu etapui „Visi procesai“. Veiksmai, kuriuos galima atlikti šiame žingsnyje yra tokie pat kaip ir meniu etape „Visi procesai“. Dėl šios priežasties jų pakartotinai neaprašinėsime.

5.3.4.2. Pasirinkto lygio funkcijos

Pasirinkus sekantį meniu punktą „Pasirinkto lygio funkcijos“ galima peržiūrėti tik tam tikro lygio funkcijas kaip ir procesus (44 pav.). Atsidariusiame lange galima pasirinkti vieną iš galimų hierarchijos lygių.



44 pav. Funkcijų hierarchijos lygių pasirinkimo langas

Pasirinkus konkretų hierarchijos lygį, sistema išveda langą, kuriame pateikiamos to lygio funkcijos (45 pav.).

Funkcijos

Funkcijos

ID: 3 "Tėvinio" proceso ID 0

Pavadinimas: Atsiskaitymas už žaliavą Eiliškumas 4

1 lygio funkcijos:

| Funkcijos ID | Funkcijos pavadinimas | Eiliškumas | Tevo ID |
|--------------|----------------------------|------------|---------|
| 3 | Atsiskaitymas už žaliavas | 4 | 0 |
| 4 | Kokybės tikrinimas | 5 | 0 |
| 6 | Užsakymo patvirtinimas | 2 | 0 |
| 7 | Užsakymo priėmimas | 1 | 0 |
| 9 | Užsakymo sutikrinimas | 6 | 0 |
| 12 | Žaliavų užsakymas | 3 | 0 |
| 19 | Atsiskaitymas su gamintoju | 7 | 0 |

<< < > >>

Redagavimo režimas

Rodyti funkcijos "tėvą"

Išėiti

Įterpti Trinti

45 pav. Sistemos langas „Pasirinkto lygio funkcijos“

Šio sistemos lango funkcijos:

- Įvesti naujas funkcijas;
- Šalinti esamas funkcijas;
- Redaguoti funkcijų duomenis.

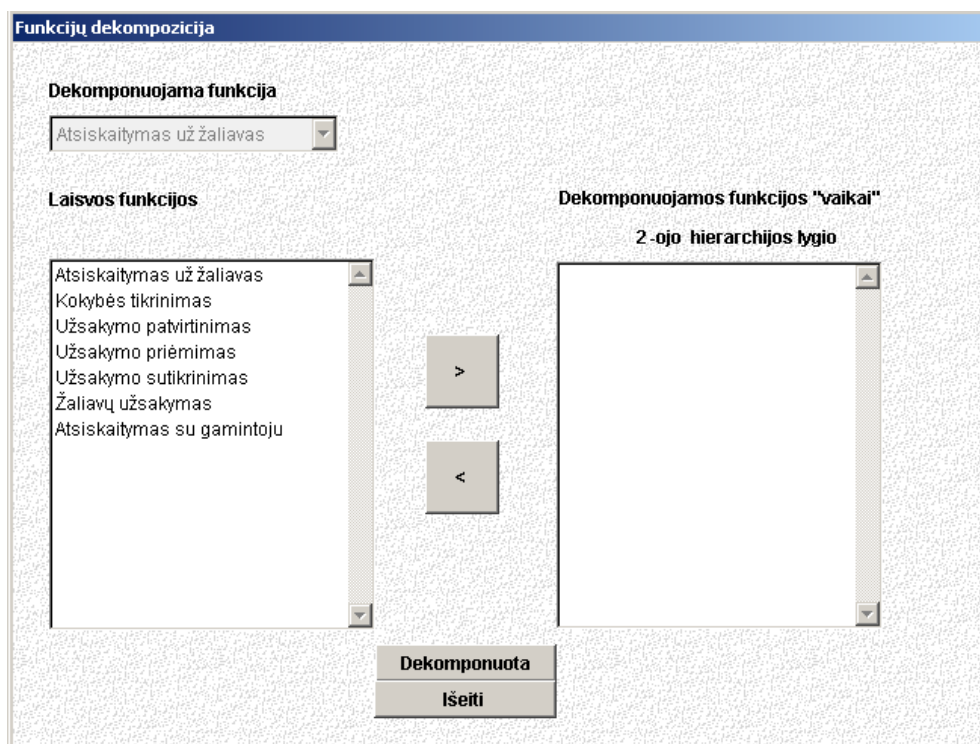
5.3.4.3. Funkcijų dekompozicija

Pasirinkus sekantį meniu punktą „Funkcijų dekompozicija“ sistema atidaro langą, kuriame galima nustatinėti funkcijų hierarchijos lygius ir jų priklausomybę kitiems procesams (46 pav.).



46 pav. Dekomponuojamos funkcijos pasirinkimo langas

Sekantis sistemos langas (47 pav.) yra analogiškas procesų dekompozicijos langui. Tad šio lango funkcijos atitinka procesų dekompozicijos funkcijas.



47 pav. Sistemos langas „Funkcijų dekompozicija“ – funkcijų priskyrimas

5.3.4.4. Funkcijų įeiga/išeiga

Pasirinkus sekantį funkcijų meniu punktą „Funkcijų įeiga/išeiga“ sistema atidaro langą, kuriame galima priskirti pasirinktai funkcijai jos įeigą ir išeigą tai yra informacinius srautus (48 pav.).

Funkcijų įeiga \ išeiga

Funkcijos ID

Funkcijos pavadinimas

Funkcijos informacinis srautas **Tipas**

| Pavad | Input |
|--------------|--------|
| Užsakymas | įeiga |
| Važtaraščiai | išeiga |
| | |
| | |
| | |
| | |
| | |

<< < > >>

Priskirti įeigą

Priskirti išeigą

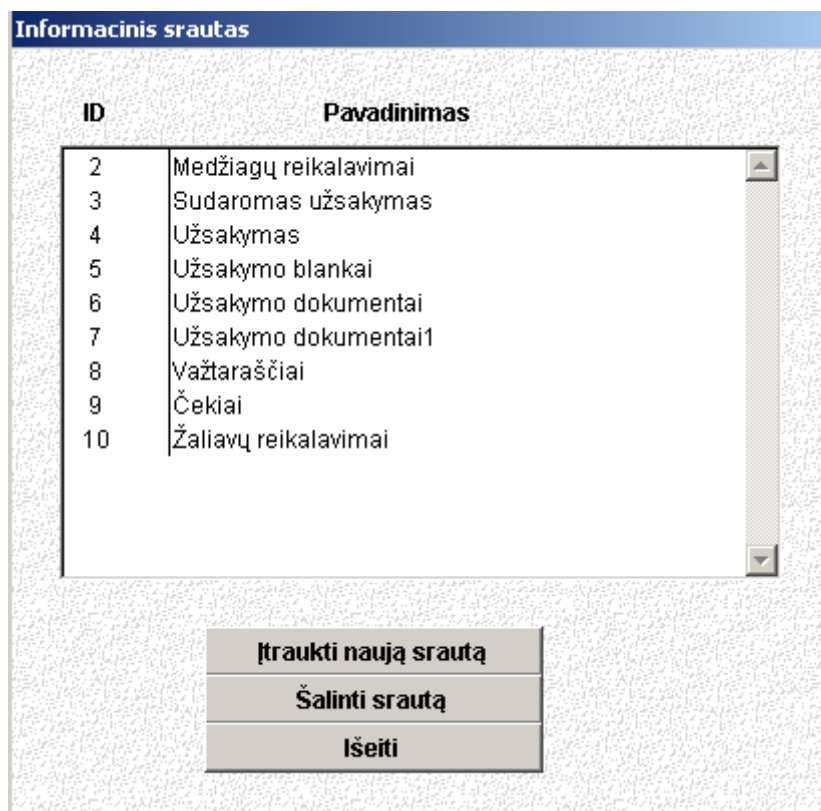
Šalinti pažymėtą srautą

Išeiti

48 pav. Funkcijų įeigos ir išeigos priskyrimo langas

5.3.4.5. Informacinis srautas

Pasirinkus paskutinį funkcijų meniu punktą „Informacinis srautas“ sistema atidaro langą, kuriame galime atlikti žemiau pateiktus veiksmus su informaciniu srautu (49 pav.).



49 pav. Sistemos langas „Informacinis srautas“

Šiame lange galima atlikti tokius veiksmus:

- Įvesti naujus informacinius srautus;
- Šalinti esamus informacinius srautus;
- Įvesti/ pašalinti informacinio srauto atributus.

5.3.5. Procesų ir funkcijų ryšys

Pasirinkus meniu punktą „Procesų ir funkcijų ryšys“ sistema pateikia sekantį meniu, sudarytą iš sekančių etapų: „Procesai ir funkcijos“, „Susieti procesus su funkcijomis“.

5.3.5.1. Procesai ir funkcijos

Pasirinkus pirmąjį „Procesų ir funkcijų ryšys“ meniu etapą – „Procesai ir funkcijos“ sistema pateikia sekantį programos langą, kuris pavaizduotas 50 paveiksle.

Procesų ir Funkcijų ryšys

Proceso ID

Proceso pavadinimas

Procesą valdančios funkcijos:

| Funkcijos ID | Funkcijos pavadinimas |
|--------------|---------------------------|
| 3 | Atsiskaitymas už žaliavas |
| | |
| | |
| | |

50 pav. Sistemos langas „Procesų ir funkcijų ryšys“

Šiame lange vartotojas gali:

- Susieti procesą su funkcija;
- Šalinti valdančią funkciją;
- Koreguoti metodą;
- Nustatyti funkcijos valdomus atributus.

Pasirinkus mygtuką „**Susieti procesą su funkcija**“ sistema išveda langą, kuriame galima priskirti pasirinktam procesui funkciją. Sistema automatiškai išrenka tik to paties hierarchinio lygio funkcijas, kurias vartotojas gali priskirti procesui. Pasirinkus funkciją spaudžiamas mygtukas „**Patvirtinti**“.

Procesų ir funkcijų susiejimas

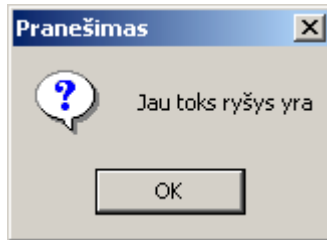
Susiejamas procesas

Proceso hierarchijos lygis

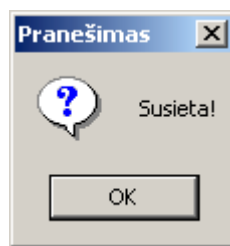
Pasirinkite funkciją:

51 pav. Sistemos langas „Procesų ir funkcijų susiejimas“

Jei pasirinkta funkcija jau susieta su konkrečiu procesu, sistema apie tai informuoja vartotoją išvesdama 52 paveiksle parodytą pranešimą. Kitu atveju vartotojas informuojamas, kad procesas su funkcija susietas (53 pav.).

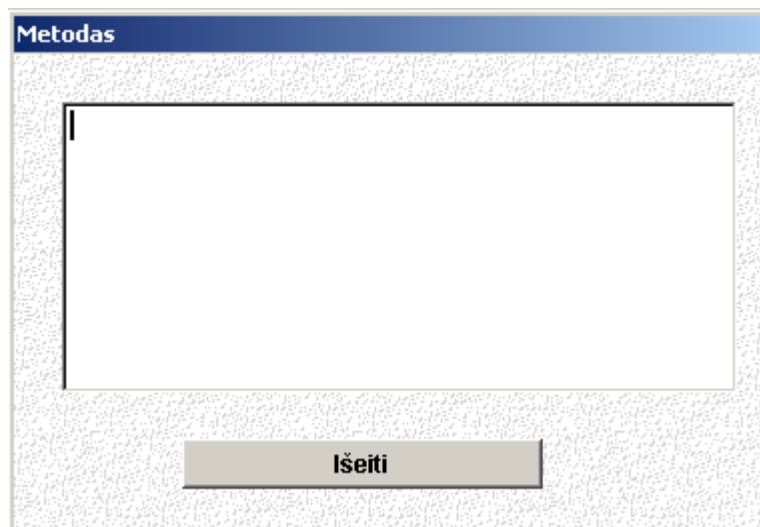


52 pav. Pranešimas apie egzistuojantį ryšį



53 pav. Pranešimas apie sėkmingai atliktą veiksmą

Norint koreguoti metodą pasirenkamas mygtukas „**Metodas**“ ir atsiradusiame 54 paveikslo lange galima keisti metodą.

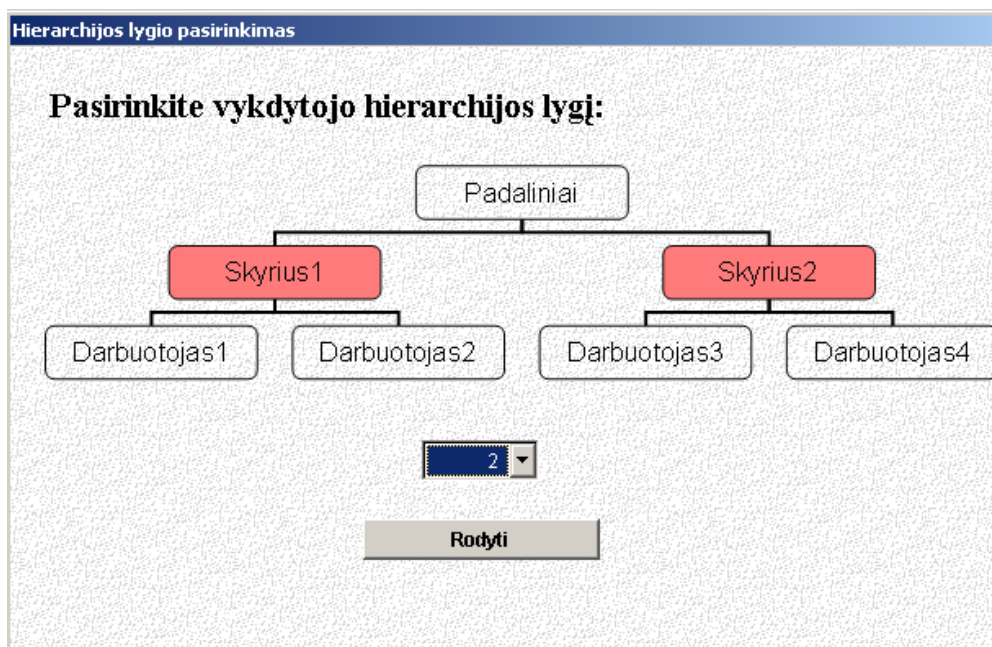


54 pav. Sistemos langas „Metodas“

Norint įvesti ar pašalinti funkcijos valdomus papildomus duomenis (atributus) reikia pasirinkti mygtuką „**Funkcijos valdomi atributai**“. Atidaromas naujas langas, kuriame galima įvesti bei pašalinti atributus. Veiksmai analogiškai proceso atributų įvedimui, bei šalinimui.

5.3.6. Vykdytojai

Pasirinkus meniu punktą „Vykdytojai“ galima peržiūrėti tik tam tikro lygio vykdytojus. Atsidariusiame lange galima pasirinkti konkretų hierarchijos lygį (sistema leidžia pasirinkti tik tuo metu esančius hierarchijos lygius).



55 pav. Proceso hierarchijos lygio pasirinkimas

Pasirinkus konkretų hierarchijos lygį, sistema išveda langą, kuriame matomi tik pasirinkto lygio vykdytojai (56 pav.).

Vykdytojai

ID: 5 Pavadinimas: Matavimo skyrius
Padalinio numeris: 0

| Vykdyt_id | Vykdyt_pav | Tevo_id |
|-----------|---------------------|---------|
| 5 | Matavimo skyrius | 0 |
| 6 | Modellavimo skyrius | 0 |
| 9 | Paruošimo skyrius | 0 |
| 14 | Siuvimo skyrius | 0 |
| 15 | Sukirpimo skyrius | 0 |
| 16 | Žaliavų skyrius | 0 |
| 17 | Tiekėjų skyrius | 0 |
| 19 | Užsakymų skyrius | 0 |

<< < > >>

Redagavimo režimas Rodyti vykdytojo duomenis
Rodyti padalinį Išėiti

Įterpti Trinti

56 pav. Sistemos langas „Vykdytojai“

Šiame lange galima atlikti tokias funkcijas:

- Įvesti naujus vykdytojus;
- Šalinti esamus vykdytojus;
- Redaguoti vykdytojų duomenis;
- Įvesti bei pašalinti papildomus vykdytojų duomenis.

Pasirinkus mygtuką „**Įterpti**“ sistema leidžia įvesti vykdytojo pavadinimą. Pasirinkus mygtuką „**Trinti**“ paprašoma patvirtinti savo pasirinkimą.

Pasirinkus mygtuką „**Redagavimo režimas**“ sistema leidžia koreguoti 57 paveiksle pažymėtus lango laukus (pavadinimą ir vykdytojo numerį). Šiame režime neleidžiama atlikti kitų veiksmų: naujo vykdytojo įvedimo, šalinimo. Norint juos atlikti reikia pasirinkti mygtuką „**Išėiti iš redagavimo režimo**“.

| Vykdyt_id | Vykdyt_pav | Tevu_id |
|-----------|---------------------------------|---------|
| 1 | Atsiskaitymų padalinys | 0 |
| 2 | Gamybos padalinys | 0 |
| 4 | Kokybės tikrinimo padalinys | 0 |
| 8 | Pakavimo padalinys | 0 |
| 11 | Pristatymo padalinys | 0 |
| 12 | Produkcijos tikrinimo padalinys | 0 |
| 13 | Sandėliavimo padalinys | 0 |
| 18 | Užsakymų padalinys | 0 |
| 20 | Žaliavų padalinys | 0 |

57 pav. Sistemos langas „Vykdotojai“ – redagavimo režimas

Norint įvesti ar pašalinti vykdytojų papildomus duomenis (atributus) reikia pasirinkti mygtuką „**Rodyti vykdytojų duomenis**“. Atidaromas naujas langas, kuriame galima įvesti bei pašalinti atributus.

Pasirinkus mygtuką „**Išėiti**“ sistema uždaro „Vykdotojų“ langą.

5.3.7. Ataskaitos

Pasirinkus meniu punktą „Ataskaitos„ sistema pateikia sekančius etapus:

- ❖ Visa duomenų bazė:
 - Procesų ataskaita;
 - Funkcijų ataskaita;
 - Vykdytojų vykdomi procesai;
 - Vykdytojų ataskaita;
 - Materialių srautų ataskaita;
 - Informacinių srautų ataskaita;
- ❖ Proceso subprocesai;
- ❖ Funkcijos subfunkcijos;
- ❖ Funkcijos valdomi procesai.

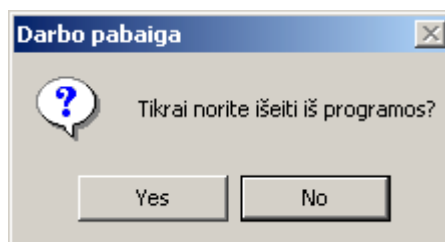
Pasirinkus bet kurį iš šių minėtų etapų sistema pateikia ataskaitą, susijusią su šia dalimi. Procesų kaip ir funkcijų ataskaitoje pateikiama duomenų bazėje esanti informacija, susijusi su procesais ir funkcijomis. Ataskaita „Vykdytojų vykdomi procesai“ pateikia vykdytojus ir kokius procesus jie atlieka. Materialių ir informacinių srautų ataskaitos pateikia srautų sąrašus, esančius duomenų bazėje.

Pasirinkus „Procesų subprocesų“ ir „Funkcijų subfunkcijų“ ataskaitas išvedami atitinkamai pasirinkto proceso subprocesai ar funkcijos subfunkcijos ir jų materialiais arba informaciniais srautais, nurodant koks srautas yra proceso ar funkcijos įeiga ar išeiga.

Pasirinkus ataskaitą „Funkcijos valdomi procesai“ pateikiama ataskaita apie konkrečios pasirinktos funkcijos procesus tai yra kokie procesai priklauso tai funkcijai ir taip pat pateikiama informacija apie tų procesų atributus.

5.3.8. Darbo pabaiga

Pasirinkus ši meniu punktą sistema pasiteirauja ar tikrai vartotojas nori baigti darbą, išvedamas 58 pav. pranešimas. Jei pasirenkamas „Yes“ mygtukas baigiamas darbas su programa.

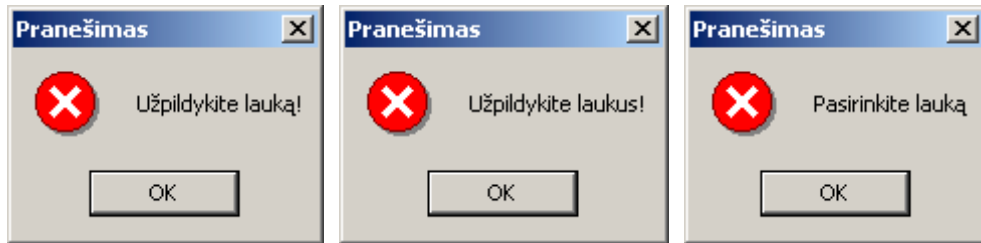


58 pav. Reikalavimas patvirtinti savo pasirinkimą.

5.3.9. Galimos klaidos

Vartotojui atliekant informacijos įvedimo, koregavimo, susiejimo ar kitokius veiksmus sistema gali pateikti pranešimus apie padarytas klaidas.

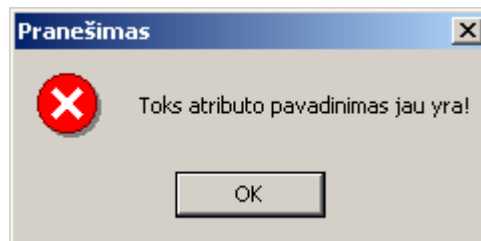
Vienas iš dažniausiai pasitaikančių klaidų – neužpildyti įvedamų duomenų laukai (59 pav.).



59 pav. Pranešimai apie neįvestus laukus

Veiksmai norint išvengti šių pranešimų – vartotojas turi užpildyti visus laukus: įvesti srautų pavadinimus, atributus ir t.t.

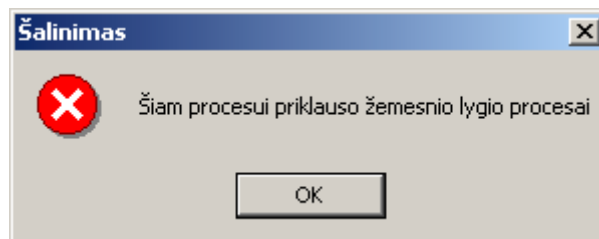
Dažnai pasitaiko, kai vartotojas nori įvesti jau egzistuojantį įrašą (jo pavadinimą). Tokiu atveju programa išveda 60 paveiksle klaidos pranešimą.



60 pav. Pranešimas apie egzistuojantį atributą

Šiuo atveju vartotojui reiktų patikslinti ar jis teisingai įvedė įrašo (proceso, funkcijos, vykdytojo, ar papildomo atributo) pavadinimą, ir patikrinti ar neegzistuoja įrašas su tokiu pavadinimu.

Kita galima klaida pasitaiko vartotojui šalinant neleistinus įrašus (61 pav.).



61 pav. Pranešimas apie ryšius tarp įrašų

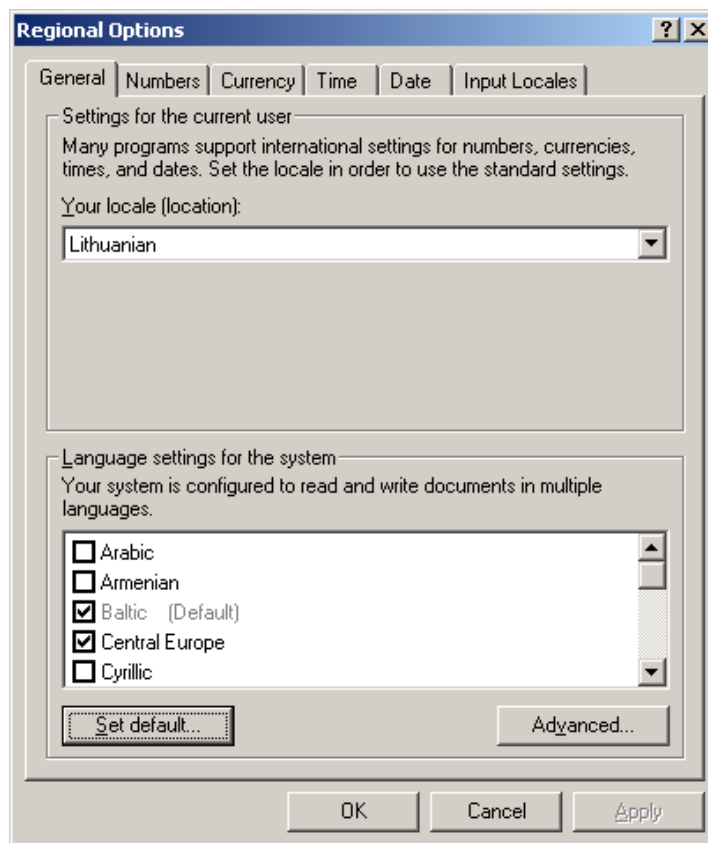
Jeigu procesas, funkcija ar vykdytojas yra susietas atitinkamai su žemesniais procesais, funkcijomis ar vykdytojais, programa neleidžia šių įrašų trinti. Pirmiausiai reiktų ištrinti žemesnio

lygio įrašus, ir tik tada, kai nebelieka hierarchinio ryšio tarp procesų, funkcijų ar vykdytojų galima juos šalinti.

5.3.10. Pradiniai nustatymai

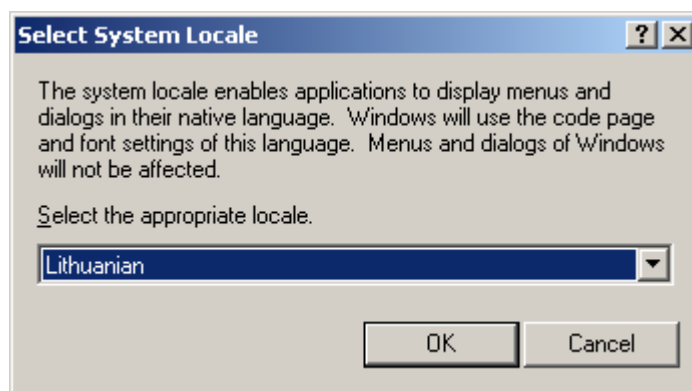
Nustatykite sistemos pagrindinę kalbą – lietuvių kalbą. Tai atliekama vykdant šiuos žingsnius:

- ✓ Atsiverkite pagrindinio regiono nustatymo kortelę (**Start** → **Settings** → **Control Panel** → **Regional Option**) (62 pav.).



62 pav. Regiono nustatymas

- ✓ Kortelėje „General“ paspauskite mygtuką **Set default**. Atsivėrusioje kortelėje pasirinkite lietuvių kalbą ir paspauskite mygtuką **OK** (63 pav.).

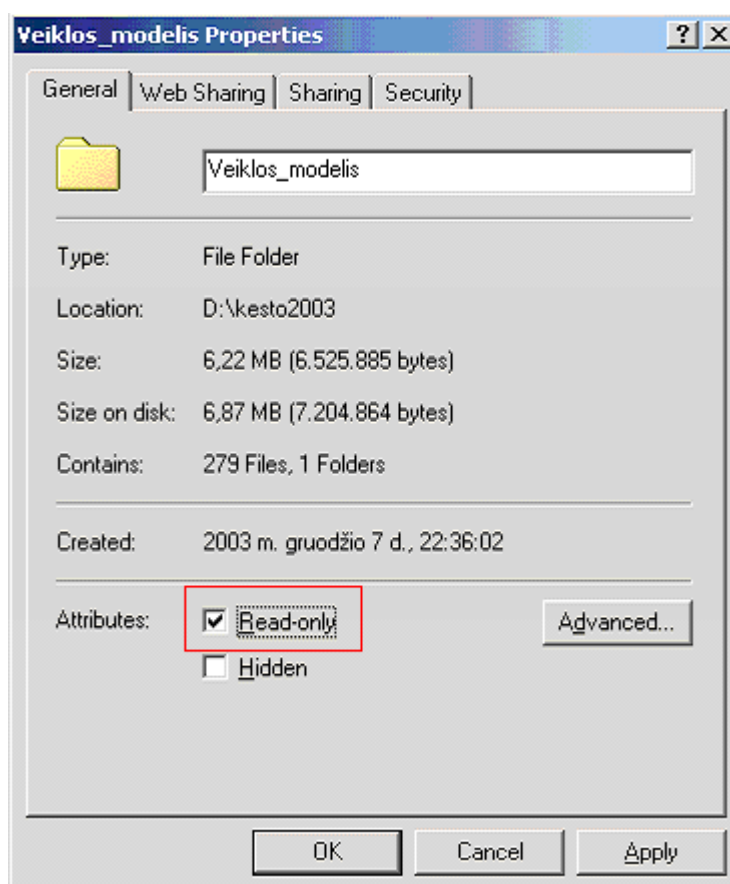


63 pav. Kalbos pasirinkimas

5.3.11. Sistemos diegimas

Sistemos nereikia instaliuoti. Pakanka nukopijuoti sistemos pagrindinį aplanką (ir jame esančius failus) į kompiuterio kietąjį diską. Tai atliekama:

- ❖ Nukopijuojame sistemos aplanką Veiklos_modelis ir jame esančius failus į kompiuterio kietąjį diską;
- ❖ Pažymime į kietą diską perkeltą sistemos aplanką „Veiklos_modelis“ ir žiūrime jo savybes. Reikia įsitikinti, kad nėra uždėtas Read-only atributas. Jei jis uždėtas, tada reikia jį nuimti. Jei šis atributas bus uždėtas sistema negalės keisti reikalingų bylų turinį (64 pav.).



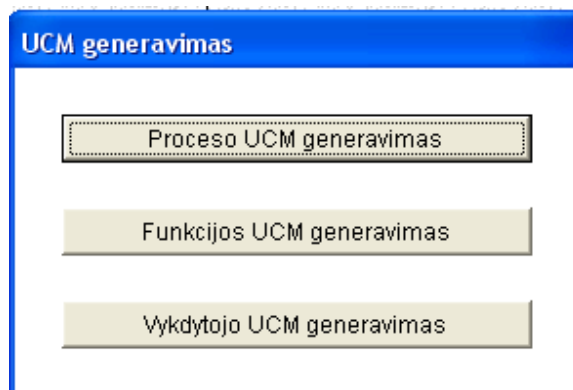
64 pav. Atributų nustatymas

5.4. UCM generavimas

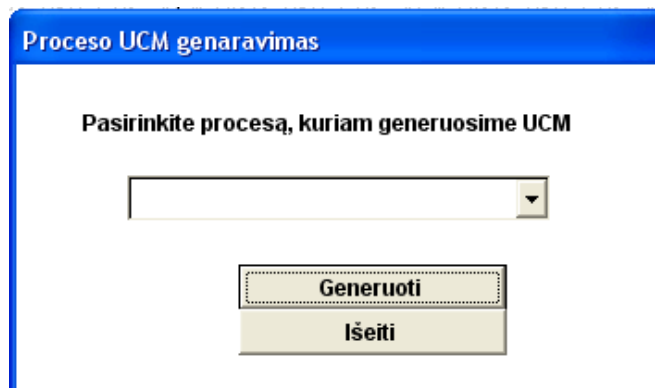
Generuoti vartotojų poreikių modelius (UCM) galima pagal:

- ❖ vartotojo (analitiko) nurodytą veiklos funkciją;
- ❖ vartotojo (analitiko) nurodytą veiklos procesą;
- ❖ vartotojo (analitiko) nurodytą vykdytoją.

Modeliuotojas gali pasirinkti UCM modelio tipą (64 pav.). Reikia nurodyti objektą kurio atžvilgiu bus generuojamas UCM modelis (65 pav.).



65 pav. Generuojamo UCM modelio pasirinkimas



66 pav. Proceso pasirinkimas

6. Išvados

- ❖ Išanalizuoti CASE modeliavimo paketai (*Provision WorkBench v. 4.0.2 Enterprise Pro, MS Visio Professional 2002, ORACLE Designer2000, MagicDraw UML Professional*) ir nustatyta, kad juose negalima automatiškai generuoti panaudojimo atvejų modelių. UCM modeliai kuriami remiantis vartotojo pateikta informacija ir analitiko patirtimi.
- ❖ Išanalizuoti veiklos modeliai (Vertės grandinės modelis (*M.Porter*), Hierarchinis veiklos modelis, UEMML), tačiau jų nepakanka IS inžinerijos tikslams. Šie standartai pilnai netenkina nei organizacijos veiklos projektuotojų nei programinės įrangos gamintojų.
- ❖ Veiklos žinių bazei kurti už pagrindą paimtas Informacijos sistemų katedroje sukurtas veiklos meta-modelis;
- ❖ Darbe sudaryta veiklos modelio klasių diagrama (UML), leidžianti patikslinti veiklos meta-modelio sudėtį nustatant pagrindinius esybių atributus;
- ❖ Suprojektuota ir realizuota (*Rose Enterprise Edition 2002, MS Visio Professional 2002, Visual FoxPro 7.0*), kuri skirta IS inžinerijos tikslams. CASE veiklos žinių bazės tvarkymo sistema galima naudoti organizacijos veiklos modeliavimui;
- ❖ Realizuota CASE žinių bazės tvarkymo sistema, pasižyminti šiomis funkcijomis:
 - ✓ Veiklos procesų ir veiklos funkcijų įvedimas ir koregavimas, papildomų atributų nustatymas;
 - ✓ Veiklos procesų ir veiklos funkcijų ryšio nustatymas, hierarchinės struktūros modeliavimas;
 - ✓ Materialių, informacinių srautų įvedimas ir koregavimas;
 - ✓ Vykdytojų įvedimas, koregavimas, susiejimas su veiklos funkcijomis ir veiklos procesais;
 - ✓ UCM generavimas pagal:
 - vartotojo (analitiko) nurodytą veiklos funkciją;
 - vartotojo (analitiko) nurodytą veiklos procesą;
 - vartotojo (analitiko) nurodytą vykdytoją.
 - ✓ Sudaryti veiklos procesų ir veiklos funkcijų, vykdytojų vykdomų procesų, vykdytojų, materialių ir informacinių srautų ataskaitas ir išsamias pasirinktų veiklos procesų, veiklos funkcijų ir funkcijos valdomų procesų ataskaitas.
- ❖ CASE žinių bazės tvarkymo sistema sėkmingai patikrinta įvedant, koreguojant, šalinant gamybinės įmonės modelį (kontrolinius duomenis);

- ❖ CASE žinių bazės tvarkymo sistema yra žiniomis grindžiamos sistemos prototipas, kuri dar galima tobulinti. Sistema leidžia generuoti UCM modelius veiklos žinių bazės pagrindu, tuo įrodant pačio prototipo ir sukurtos veiklos žinių bazės reikalingumą.
- ❖ CASE žinių bazės tvarkymo sistemą galima taikyti praktikoje tobulinant CASE sistemas, papildant jas veiklos žinių baze.

7. Literatūra

- [1] Lopata A., Gudas S. Vartotojo poreikių modelio generavimas veiklos modelio pagrindu // Kompiuterininkų dienos 2003 : respublikinė konferencija. Vilnius. 2003.
- [2] Framework for Managing Process Improvement, Department of Defence, 1994. Prieiga per internetą: < <http://www/dtic/mil/c3i/bprcd/3003sb.htm> >.
- [3] Lopata A., Gudas S. Organizacijos informacinių išteklių identifikavimo būdas // Informacinės technologijos 2001: konferencijos pranešimų medžiaga. Kaunas. Technologija, 2001.
- [4] Lopata A., Gudas S. Informacijos išteklių identifikavimas, veiklos modelio pagrindu // Informacijos mokslai. Vilniaus universiteto leidykla, T19, Vilnius, 2001, p. 43-50.
- [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/cimoso>.
- [7] Universal Enterprise modeling language, IFAC-IFIP Task Force. [žiūrėta 2003-10- 30]. Prieiga per internetą: < <http://www.cit.gu.edu.au/~bernus/taskforce/archive/UEML-TF-IG.ppt> >.
- [8] 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.
- [9] ProVision “Work Bench” User Manual Guide. Prieiga per internetą: <www.proformacorp.com>.
- [10] 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 >.
- [11] UEML (2001), Universal Enterprise Modeling Language, IFAC-IFIP Task Force. [žiūrėta 2001-10-30]. Prieiga per internetą : <<http://www.cit.gu.edu.au/~bernus/taskforce/archive/UEML-TF-IG.ppt>>.
- [12] UML 1.4 (2000) specification. [žiūrėta 2003-11-10]. Prieiga per internetą : <<http://www.omg.org/cgi-bin/doc?ad/01-02-11>>.
- [13] Appleton, Brad. Welcome to Brad Appleton's Documents. [žiūrėta 2004-05-10]. Prieiga per internetą : < <http://www.cmcrossroads.com/bradapp/> >.

8. Priedai

1 Priedas. Case žinių tvarkymo sistemos duomenų bazės lentelių aprašymas

| Procesas | | |
|------------------|--------------|---|
| Atributas | Tipas | Atributų aprašymas |
| Kodas | Integer | Proceso unikalus numeris (identifikatorius) |
| Pav | Char | Proceso pavadinimas |
| Tevo_kodas | Integer | Proceso aukštesnio proceso („tėvo“) kodas |
| Vykdyt_kodas | Integer | Vykdytojo, kuris vykdo procesą, kodas |
| H_lygis | Integer | Proceso hierarchijos lygis |
| WF_kodas | Integer | Darbų sekų modelio kodas |
| Pradžia | Boolean | Proceso pradžios identifikatorius |
| Galas | Boolean | Proceso pabaigos identifikatorius |

Lentelėje **Procesas** saugomi organizacijos procesai.

| Proc_Atrib | | |
|-------------------|--------------|------------------------------|
| Atributas | Tipas | Atributų aprašymas |
| Kodas | Integer | Proceso numeris |
| Atributas | Char | Proceso atributo pavadinimas |

Lentelėje **Proc_Atrib** saugomi proceso atributai.

| Veikla | | |
|------------------|--------------|---|
| Atributas | Tipas | Atributų aprašymas |
| V_Kodas | Integer | Veiklos unikalus numeris (identifikatorius) |
| Pav | Char | Veiklos pavadinimas |
| Tevo_kodas | Integer | Veiklos aukštesnės veiklos („tėvo“) kodas |
| Vykdyt_kodas | Integer | Vykdytojo, kuris vykdo veiklą, kodas |
| H_lygis | Integer | Veiklos hierarchijos lygis |
| WF_kodas | Integer | Darbų sekų modelio kodas |
| Pradžia | Boolean | Veiklos pradžios identifikatorius |
| Galas | Boolean | Veiklos pabaigos identifikatorius |
| Tipo_kodas | Integer | Kodas, nurodantis veiklos tipą |
| Funk_kodas | Integer | Funkcijos, kuriai priklauso veikla, kodas |

Lentelėje **Veikla** saugomos organizacijos veiklos.

| Veiklos_Atrib | | |
|----------------------|--------------|------------------------------|
| Atributas | Tipas | Atributų aprašymas |
| Kodas | Integer | Veiklos numeris |
| Atributas | Char | Veiklos atributo pavadinimas |

Lentelėje **Veiklos_Atrib** saugomi proceso atributai.

| Funkcija | | |
|------------------|--------------|---|
| Atributas | Tipas | Atributų aprašymas |
| Funk_kodas | Integer | Funkcijos unikalus numeris (identifikatorius) |
| Funk_Pav | Char | Funkcijos pavadinimas |
| P_kodas | Integer | Proceso kodas |

Lentelėje **Funkcija** saugomos organizacijos funkcijos.

| V_tipas | | |
|------------------|--------------|--------------------------------------|
| Atributas | Tipas | Atributų aprašymas |
| Tipo_kodas | Integer | Veikos tipo kodas (identifikatorius) |
| Tipas | Char | Veiklos tipo pavadinimas pavadinimas |

Lentelėje **V_tipas** yra saugomi veiklų tipų sąrašas.

| PxFx | | |
|------------------|--------------|--|
| Atributas | Tipas | Atributų aprašymas |
| PxFx | Integer | Proceso-funkcijos sankirtos kodas (identifikatorius) |
| Kodas | Integer | Proceso kodas |
| F_kodas | Integer | Funkcijos kodas |
| Pavad | Char | Proceso ir funkcijos sąryšio pavadinimas |

Lentelėje **PxFx** saugomi proceso-funkcijos sąryšio duomenys.

| Vykdytojas | | |
|-------------------|--------------|---|
| Atributas | Tipas | Atributų aprašymas |
| Vyk_kodas | Integer | Vykdytojo unikalus kodas (identifikatorius) |
| Vyk_pav | Char | Vykdytojo pavadinimas |
| Tevo_kodas | Integer | Aukštesnio lygio vykdytojo („tėvo“) kodas |
| H_lygis | Integer | Vykdytojo hierarchijos lygis |
| WF_kodas | Integer | Darbų sekų modelio kodas |
| Tipas | Char | Nurodomas vykdytojo tipas |

Lentelėje **Vykdytojas** saugomi organizacijos vykdytojai (skyriai, padaliniai, darbuotojai).

| Vykd_Atrib | | |
|-------------------|--------------|--------------------------------|
| Atributas | Tipas | Atributų aprašymas |
| Vyk_Kodas | Integer | Vykdytojo numeris |
| Atributas | Char | Vykdytojo atributo pavadinimas |

Lentelėje **Vykd_Atrib** yra saugomi vykdytojo atributai.

| Srautas | | |
|------------------|--------------|-----------------------------------|
| Atributas | Tipas | Atributų aprašymas |
| Sr_kodas | Integer | Srauto numeris (identifikatorius) |
| Pav | Char | Srauto pavadinimas |
| WF_kodas | Integer | Darbų sekų modelio kodas |
| Tipas | Char | Srauto tipas (įeiga, išeiga) |

Lentelėje **Srautas** yra saugomi organizacijos srautai.

| Sraut_Atrib | | |
|--------------------|--------------|-----------------------------|
| Atributas | Tipas | Atributų aprašymas |
| Sr_Kodas | Integer | Srauto numeris |
| Atributas | Char | Srauto atributo pavadinimas |

Lentelėje **Sraut_Atrib** yra saugomi srautų atributai.

| Proc_Mat_Srautas | | |
|-------------------------|--------------|---------------------------------------|
| Atributas | Tipas | Atributų aprašymas |
| Kodas | Integer | Proceso numeris |
| Sr_kodas | Integer | Srauto numeris |
| Input | Char | Proceso ir srauto sąryšio pavadinimas |

Lentelėje **Proc_Mat_Srautas** yra saugojamas procesų-materialių srautų sąrašai.

| Veikl_Inf_Srautas | | |
|--------------------------|--------------|---------------------------------------|
| Atributas | Tipas | Atributų aprašymas |
| V_Kodas | Integer | Veikos numeris |
| Sr_kodas | Integer | Srauto numeris |
| Input | Char | Veiklos ir srauto sąryšio pavadinimas |

Lentelėje **Veikl_Inf_Srautas** yra saugojamas veiklų-informacinių srautų sąrašai.

| Ivykis | | |
|------------------|--------------|-----------------------------------|
| Atributas | Tipas | Atributų aprašymas |
| Kodas | Integer | Įvykio numeris (identifikatorius) |
| Pav | Char | Įvykio pavadinimas |

Lentelėje **Ivykis** yra saugomi įvykiai.

| IvykisXProc | | |
|--------------------|--------------|--|
| Atributas | Tipas | Atributų aprašymas |
| IxP_kodas | Integer | Proceso-įvykio sankirtos kodas (identifikatorius) |
| P_kodas | Integer | Proceso numeris |
| I_kodas | Integer | Įvykio numeris |

Lentelėje **IvykisXProc** yra saugomas įvykių – procesų sąrašas. Jis parodo kokie įvykiai sužadina procesus.

| Tikslas | | |
|------------------|--------------|-----------------------------------|
| Atributas | Tipas | Atributų aprašymas |
| Kodas | Integer | Tikslo numeris (identifikatorius) |
| Pav | Char | Tikslo pavadinimas |

Lentelėje **Tikslas** yra saugomi tikslai.

| FunkcXTikslas | | |
|----------------------|--------------|--|
| Atributas | Tipas | Atributų aprašymas |
| FxI_kodas | Integer | Funkcijos-tikslo sankirtos kodas (identifikatorius) |
| F_kodas | Integer | Funkcijos numeris |
| T_kodas | Integer | Tikslo numeris |

Lentelėje **FunkcXTikslas** yra saugomas funkcijų – tikslų sąrašas.

| WF | | |
|------------------|--------------|---|
| Atributas | Tipas | Atributų aprašymas |
| WF_kodas | Integer | Darbų sekų modelio kodas (identifikatorius) |
| Pav | Char | Darbų sekų modelio pavadinimas |

Lentelėje **WF** yra saugomi darbų sekų modelių duomenys.