

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Eglė Malinauskienė

**Rekonstrukcijos metodų analizė modernizuojant
informacinę sistemą**

Magistro darbas

Darbo vadovas
doc. V. Jusas

Kaunas, 2004

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

TVIRTINU

Katedros vedėjas
dr. E. Bareiša

2004 05

**Rekonstrukcijos metodų analizė modernizuojant
informacinę sistemą**

Informatikos mokslų magistro darbas

Kalbos konsultantė

Lietuvių kalbos katedros lektorė
dr. J. Mikelionienė

2004 05

Vadovas

doc. V. Jusas

2004 05

Recenzentas

doc. A. Lenkevičius

2004 05

Atliko

IFM-8/2 gr. stud.
E. Malinauskienė

2004 05

Kaunas, 2004

SANTRAUKA

Šiame magistro darbe nagrinėjami paveldėtųjų sistemų rekonstrukcijos metodai. Paveldėtoji sistema – tai seniai sukurta sistema, sunkiai suderinama su šiuolaikinėmis technologijomis ir naudojama tik todėl, kad dėl ilgo savo naudojimo tapo neatskiriama įmonės veiklos dalimi. Tokio tipo sistemos yra didelės, monolitiškos ir sunkiai modifikuojamos, o jų pakeitimo išlaidos ir rizika sunkiai įvertinami.

Programų inžinerijos mokslas siūlo laipsnišką informacinių sistemų modernizaciją rekonstruojant paveldėtas sistemas. Programinės įrangos rekonstrukcijos tikslas – pagerinti arba transformuoti programinę įrangą taip, kad ši taptų lengviau suprantama, kontroliuojama ir panaudojama iš naujo, kartu išsaugojant jos naudingas, laiko patikrintas funkcijas.

Skiriami tokie pagrindiniai rekonstrukcijos metodai: kodo retransliavimas, atvirkštinė rekonstrukcija ir duomenų rekonstrukcija. Šiame magistro darbe aprašoma minėtų rekonstrukcijos metodų analizė, atlikta modernizuojant miškų ūkio produkcijos gamybos-pardavimo apskaitos sistemą. Ištirta kiekvieno metodo pritaikymo ir įgyvendinimo laiko sąnaudos. Pateikiamas panaudotų rekonstrukcijos metodų reikšmė sistemos patikimumo, našumo, panaudojamumo ir kt. charakteristikų pagerinimui.

Raktiniai žodžiai: paveldėtosios sistemos, sistemų tobulinimas, programinės įrangos rekonstrukcija, atvirkštinė rekonstrukcija, kodo retransliacija, duomenų rekonstrukcija

ABSTRACT

This master thesis covers re-engineering methods of legacy systems. Legacy system is an old system, which is hardly compliant with modern technologies and used only because it has become an integral part of organization business process support during the long period of its maintenance. These systems are large, monolithic and difficult to modify, and cost and risk of their replacement are difficult to predict.

The science of software engineering offers an incremental modernization of information systems applying the re-engineering of legacy software. The main goal of software re-engineering is to transform the software in the way, it would become easier to understand, maintain and re-use, at the same time preserving its useful, time trusted functions.

The main re-engineering methods are source code translation, reverse engineering and data re-engineering. This thesis covers the analysis of these methods, which was made during the re-engineering of wood production and sales accounting system. The adoption and realization time rate of every method was examined. The influence of the applied re-engineering methods to the system reliability, efficiency, usability and other quality metrics is given.

Key words: legacy systems, system evolution, software re-engineering, reverse engineering, source code translation, data re-engineering

TURINYS

1	ĮVADAS	9
2	TERMINŲ IR SANTRUMPŲ ŽODYNAS.....	11
3	PAVELDĖTOSIOS SISTEMOS IR JŲ TOBULINIMO METODAI.....	12
3.1	Paveldėtosios sistemos.....	12
3.1.1	Paveldėtosios sistemos samprata	12
3.1.2	Paveldėtoji sistema – sociotechninė sistema	13
3.1.3	Paveldėtųjų sistemų struktūra	15
3.1.4	Paveldėtųjų sistemų įvertinimas	16
3.1.4.1.	Dviejų matavimų paveldėtosios sistemos vertės nustatymo modelis.....	17
3.1.4.2.	Trijų matavimų paveldėtosios sistemos vertės nustatymo modelis	19
3.2	Paveldėtųjų sistemų tobulinimas.....	23
3.2.1	Paveldėtųjų sistemų tobulinimo proceso sunkumai	23
3.2.2	Paveldėtųjų sistemų tobulinimo metodai	24
3.2.2.1.	Paveldėtosios PĮ pakeitimas nauja PĮ.....	26
3.2.2.2.	PĮ palaikymas.....	27
3.2.2.3.	PĮ modernizavimas	28
4	PROGRAMINĖS ĮRANGOS REKONSTRUKCIJA.....	32
4.1	Rekonstrukcijos samprata.....	32
4.2	Rekonstrukcijos modelis.....	32
4.3	Rekonstrukcijos proceso etapai.....	33
4.4	Rekonstrukcijos metodai	34
4.4.1	Kodo retransliavimas	35
4.4.2	Atvirkštinė rekonstrukcija	35
4.4.3	Programos restruktūrizavimas	36
4.4.4	Programos modularizacija	36
4.4.5	Duomenų rekonstrukcija	36
4.5	Rekonstrukcijos įgyvendinimo būdai.....	37
4.6	Rekonstrukcijos metodo rizikos	38
5	SISTEMOS REKONSTRUKCIJOS REALIZACIJA.....	40
5.1	Sistemos paskirtis.....	40
5.2	Reikalavimų specifikacija.....	41

5.2.1	Veiklos kontekstas	41
5.2.2	Produkto veiklos sfera	42
5.2.3	Funkciniai reikalavimai	43
5.2.4	Nefunkciniai reikalavimai	47
5.3	Architektūros specifikacija.....	48
5.3.1	Sistemos komponentų aprašymas	48
5.3.2	Sistemos klasių diagramos	50
5.3.3	Duomenų bazės schema	54
6	PAVELDĖTOSIOS IR REKONSTRUOTOS SISTEMŲ TYRIMAS	56
6.1	Paveldėtosios sistemos kokybės tyrimas.....	56
6.1.1	Kokybės įvertinimo metodo aprašymas	56
6.1.2	Kokybės įvertinimo rezultatai.....	57
6.1.3	Paveldėtosios sistemos rekonstrukcijos galimybių analizė	62
6.2	Naujos sistemos versijos kokybės tyrimas	63
6.2.1	Kokybės įvertinimo metodo aprašymas	63
6.2.2	Kokybės įvertinimo rezultatai.....	64
6.2.3	Tolimesnis sistemos tobulinimas	66
7	REKONSTRUKCIJOS METODŲ EKSPERIMENTINIS ĮVERTINIMAS.....	67
7.1	Rekonstrukcijos metodų pritaikymas nagrinėjamai sistemai.....	67
7.1.1	Atvirkštinė rekonstrukcija	67
7.1.2	Kodo retransliacija	68
7.1.3	Duomenų migracija.....	71
7.2	Rekonstrukcijos metodų efektyvumo įvertinimas	73
8	IŠVADOS.....	75
	LITERATŪRA	76
	PRIEDAI	78
A.	Dalykinės srities klasių aprašymas	78
B.	Valdymo klasių aprašymas.....	83
C.	Rekonstruotos sistemos grafinės vartotojo sąsajos pavyzdžiai	86
D.	Dalykinės srities terminų žodynas.....	90

LENTELIŲ SĄRAŠAS

1 lentelė. Priežastiniai paveldėtosios sistemos vertės kriterijai	21
2 lentelė. Paveldėtosios sistemos vertę įtakojantys poveikiai.....	21
3 lentelė. PĮ rekonstrukcijos rizikos	38
4 lentelė. PĮ rekonstrukcijos rizikų sumažinimo būdai	39
5 lentelė. Pirminių dokumentų rekvizitų priklausomybė antgalvei bei kūneliui	46
6 lentelė. Rekvizitų priklausymo išlaidavimo dokumentams lentelė	47
7 lentelė. Sistemos DOS versijos vertinimo rezultatai	61
8 lentelė. Rekonstrukcijos metodų sąsaja su sistemos charakteristikomis	63
9 lentelė. <i>BROWSE</i> kreipinio sąsaja su Visual FoxPro programavimo kalba	70
10 lentelė. <i>ACTIVATE WINDOW</i> sąsaja su Visual FoxPro programavimo kalba.....	70
11 lentelė. Rekonstrukcijos metodų įtaka sistemos charakteristikoms	73

PAVEIKSLŲ SĄRAŠAS

1 pav. Paveldėtosios sistemos komponentai	13
2 pav. Sluoksninis paveldėtosios sistemos modelis	14
3 pav. Sluoksninis paveldėtosios PĮ modelis, orientuotas į jos vaidmenį įmonėje	15
4 pav. Dviejų matavimų paveldėtosios sistemos vertės nustatymo modelis	17
5 pav. Trijų matavimų paveldėtosios sistemos vertės nustatymo modelis	20
6 pav. PĮ tobulinimo metodų hierarchija	26
7 pav. Paveldėtųjų sistemų struktūra	29
8 pav. PĮ rekonstrukcijos modelis	33
9 pav. Rekonstrukcijos įgyvendinimo procesas	34
10 pav. Atvirkštinės rekonstrukcijos procesas	35
11 pav. Miškų ūkio produkcijos gamybos-pardavimo veiklos kontekstas	42
12 pav. Sistemos panaudojimo atvejų modelis	42
13 pav. Sistemos komponentų diagrama	49
14 pav. Sistemos klasių suskirstymas į paketus	50
15 pav. Produkcijos priėmimo posistemės valdymo klasių diagrama	51
16 pav. Produkcijos išlaidavimo posistemės valdymo klasių diagrama	52
17 pav. Produkcijos priėmimo posistemės dalykinės srities klasių diagrama	53
18 pav. Produkcijos išlaidavimo posistemės dalykinės srities klasių diagrama	54
19 pav. Sistemos naudojama duomenų bazės schema	55
20 pav. Sistemos modernizacijos procesas	56
21 pav. Senos ir naujos sistemos versijų kokybės palyginimas	66
22 pav. Atvirkštinės rekonstrukcijos pritaikymas nagrinėjamai sistemai	68
23 pav. Kodo retransliacijos pritaikymas nagrinėjamai sistemai	69
24 pav. Duomenų sąsajos pokyčiai migruojant iš senos sistemos į naują	72
25 pav. Biržių sąrašo ekrano forma	86
26 pav. Biržės įvedimo ekrano forma	87
27 pav. Taškavimo lapo įvedimo ekrano forma	88
28 pav. Medienos priėmimo kortelės įvedimo ekrano forma	88
29 pav. Sąskaitos-faktūros įvedimo ekrano forma	89

1 ĮVADAS

Vienas iš aktualių programų inžinerijos tikslų šiuo metu – pagreitinti perėjimą prie evoliucionuojančių sistemų kūrimo modelio, kuriame dirbtinai sukurta riba tarp programų realizacijos ir palaikymo būtų pakeista nuolat besivystančios sistemos samprata. Pagrindinė kliūtis pasiekti šiam tikslui yra plačiai daugelyje organizacijų naudojamos paveldėtosios sistemos.

Informacinių technologijų vystymosi ankstyvajame periode, verslo įmonės buvo priverstos pačios daug investuoti, kad koku nors būdu realizuotų jų veiklos procesus įgyvendinančias informacines sistemas. Ilgainiui jos buvo gerai ištestuotos, todėl veikė patikimai, ir tokiu būdu tapo neatsiejama įmonių valdymo sistemos dalimi. Tuo pačiu metu sparčiai besivystant informacinėms technologijoms, atsirado didelis atotrūkis tarp jų siūlomų galimybių modernizuoti verslo procesą ir uždaru, tik įmonėje naudojamų sistemų kūrimo priemonių.

Dauguma organizacijų nori naujas technologijas suderinti su savo senomis sistemomis, kurios dėl jau atgyvenusių kūrimo priemonių, dokumentacijos trūkumo ir pablogėjusios struktūros, sunkiai integruojasi su siūlomais sprendimais. Norėdamos išspręsti šią problemą, organizacijos priverstos ieškoti būdų, kaip su minimaliom sąnaudom ir rizika, atnaujinti savo turimas sistemas ir pritaikyti jas tolimesniems patobulinimams.

Vienas iš plačiausiai naudojamų paveldėtųjų sistemų problemos sprendimo metodų yra programinės įrangos rekonstrukcija. Didžiausias jos privalumas – pakartotinas turimos sistemos dokumentacijos, kodo ir prie jos dirbančių žmonių patirties panaudojimas.

Dauguma tyrimo darbų rekonstrukcijos srityje skirti Fortran ir COBOL programavimo kalbomis sukurtoms sistemos. Lietuvoje aštuntajame ir devintajame praėjusio amžiaus dešimtmetyje dauguma verslo sistemų DOS terpei buvo kuriamos naudojant DBASE, Clipper, FoxPro ir panašias į DB orientuotas programavimo kalbas. Tačiau nelengva rasti metodinių nurodymų, kaip atlikti paveldėtosios sistemos, sukurtos tokiomis priemonėmis, rekonstrukciją.

Todėl šio magistro darbo tikslas buvo atlikti rekonstrukcijos metodų analizę minėto tipo sistemų atveju. Kaip konkretus pavyzdys buvo pasirinkta miškų ūkio produkcijos gamybos-pardavimų apskaitos sistema, realizuota FoxPro priemonėmis.

Darbo metu buvo susipažinta su paveldėtųjų sistemų problema ir jos sprendimo metodais, didžiausią dėmesį skiriant programinės įrangos rekonstrukcijai.

Remiantis atliktos analizės rezultatais, buvo realizuota miškų ūkio produkcijos gamybos-pardavimo apskaitos sistemos rekonstrukcija ir atliktas šio proceso tyrimas.

Tyrimo metu įvertinta paveldėtosios ir rekonstruotos sistemų kokybė, pateiktas naudotų rekonstrukcijos metodų efektyvumo įvertinimas, atsižvelgiant į jų įtaką sistemos charakteristikų pagerinimui ir jų modifikacijas nagrinėjamos sistemos atveju.

2 TERMINŲ IR SANTRUMPŲ ŽODYNAS

Paveldėtoji sistema – tai kritinė organizacijos veiklai programinė įranga, kuri negali būti efektyviai tobulinama.

Programinės įrangos rekonstrukcija – tai paveldėtosios sistemos, kuri tapo per brangi palaikyti ir kurios architektūros arba realizacijos priemonės yra atgyvenusios, perrašymas panaudojant šiuolaikines PĮ ir/arba kompiuterinės įrangos technologijas.

Kodo retransliavimas – tai paprasčiausias PĮ rekonstrukcijos metodas, nereikalaujantis programinės įrangos suvokimo. Jo metu paveldėtosios sistemos kodas pertransliuojamas į kitos, naujesnės programavimo kalbos kodą – programos struktūra ir organizacija išlieka nepakitusios.

Atvirkštinė rekonstrukcija – tai programos analizės metodas, vykdomas tam, kad nustatytus sistemos komponentus ir jų sąryšius, sukurti naują programos atvaizdavimo formą, kuri būtų aukštesnio abstrakcijos lygmens negu programos kodas.

Duomenų rekonstrukcija – tai duomenų struktūros analizės ir reorganizavimo procesas.

MKF – miško kirtimo fondas

DB – duomenų bazė

DBVS – duomenų bazių valdymo sistema

PĮ – programinė įranga

CASE – Computer-Aided System Design

3 PAVELDĖTOSIOS SISTEMOS IR JŲ TOBULINIMO METODAI

3.1 Paveldėtosios sistemos

3.1.1 Paveldėtosios sistemos samprata

Nepaisant to, kad dauguma programinės įrangos (PI) specialistų intuityviai supranta, kas tai yra „paveldėtoji sistema“, nėra lengva surasti vienodą jos apibrėžimą: skirtingi šaltiniai pateikia skirtingas šios sąvokos interpretacijas.

Pagal I. Sommerville, „paveldėtosios sistemos yra senos sistemos, kurios vis dar teikia esmines verslo paslaugas“ (2001). Organizacijos nemažai pinigų investuoja į PI kūrimą ir tam, kad šios investicijos atsipirktų, ji privalo tarnauti eilę metų. Nors kiekvienos PI gyvavimo ciklas yra skirtingas, didelės sistemos naudojamos ne mažiau kaip 10, o kartais ir daugiau metų. Taigi verslo sėkmė šiais laikais labai daug priklauso nuo naudojamos PI, ir todėl bet koks jos funkcionavimo sutrikimas gali neigiamai veikti verslą.

K. H. Bennett teigia, kad paveldėtosios sistemos, tai „didelės apimties PI sistemos, su kuriomis nežinome, kaip tvarkytis, bet kurios yra gyvybiškai svarbios mūsų organizacijai“ (Gold, 1998). Tokios sistemos pasižymi tuo, kad:

- yra parašytos pasenusiomis ir dažnai nebenaudojamomis programavimo kalbomis;
- buvo kuriamos naudojant atgyvenusią programų inžinerijos metodiką;
- atlieka kritines organizacijai funkcijas;
- yra didelės apimties;
- jas sunku suprasti, taigi ir sunku palaikyti.

Nemokamas kompiuterijos terminų žodynas internete (*The Free On-Line Dictionary of Computing*) taip apibrėžia paveldėtasias sistemas: „Kompiuterinė sistema arba taikomoji programa, kuri naudojama tik dėl to, kad jos pakeitimo arba perprojektavimo išlaidos būtų labai didelės, nepaisant jos mažo konkurentabilumo ir nesuderinamumo su šiuolaikinėmis identiškėmis sistemomis. Kitaip sakant, tokio tipo sistema yra didelė, monolitiška ir sunkiai modifikuojama“ (Gold, 1998).

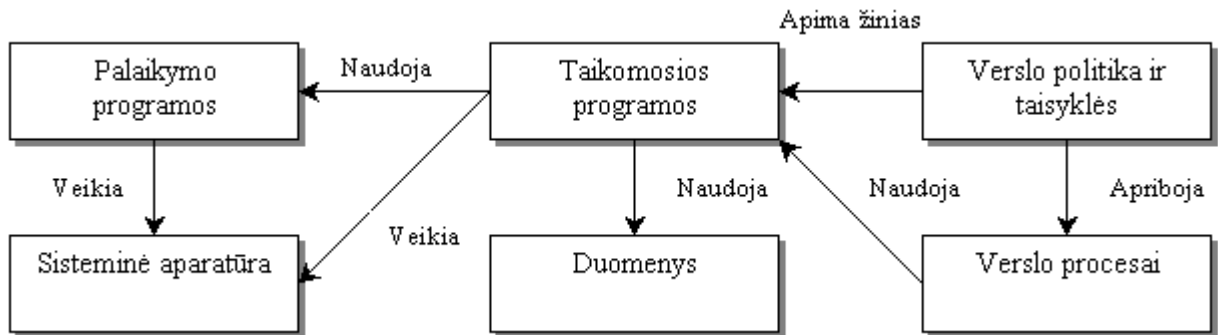
Paveldėtoji sistema, anot Brodie ir Stonebraker, yra „bet kokia informacinė sistema, kuri sunkiai pasiduoda modifikacijoms ir vystymuisi, kad patenkintų naujus ir nuolat besikeičiančius verslo reikalavimus“ (Gold, 1998).

Remiantis anksčiau pateiktais paveldėtųjų sistemų apibrėžimais, galima teigti, kad visos tokio tipo sistemos yra PĮ sistemos, kritinės organizacijos veiklai. Jas reikia nuolat modifikuoti dėl nuolat kintančių verslo aplinkos reikalavimų, tačiau tai padaryti dažnai yra labai sunku ir brangu. Taigi „paveldėtoji PĮ yra kritinė PĮ, kuri negali būti efektyviai modifikuojama“ (Gold, 1998).

3.1.2 Paveldėtoji sistema – sociotechninė sistema

Pagal ankstesniame skyriuje pateiktą paveldėtosios sistemos apibrėžimą galima būtų ją prilyginti PĮ, tačiau iš tikrųjų PĮ yra tik viena sudedamoji paveldėtosios sistemos dalis.

Plačiąja prasme paveldėtosios sistemos yra sociotechninės sistemos, sudarytos iš PĮ, kompiuterinės įrangos, duomenų ir verslo procesų (Sommerville, 2001).



1 pav. Paveldėtosios sistemos komponentai

Pav. 1 parodytos paveldėtosios sociotechninės sistemos sudedamosios dalys ir jų tarpusavio ryšiai:

1. Sisteminė aparatūra: dauguma paveldėtųjų sistemų buvo sukurtos dar didiesiems kompiuteriams, o jų palaikymas brangiai kainuoja.
2. Palaikymo programos: paveldėtoji sistema gali naudoti įvairias palaikymo programas, pradedant operacine sistema ir baigiant kompiliatoriais, naudojamais sistemos vystymo procese. Šios priemonės taip pat gali būti jau atgyvenusios ir nebepalaikomos savo gamintojų.
3. Taikomosios programos: paveldėtoji sistema dažnai sudaryta iš daugelio atskirų programų, sukurtų skirtingu metu ir realizuojančių skirtingas verslo funkcijas.
4. Duomenys: tai duomenys, kuriuos apdoroja taikomoji PĮ. Per paveldėtųjų sistemų gyvavimo laikotarpį susikaupia didžiuliai duomenų kiekiai, ir dažnai šie duomenys būna nesuderinami tarpusavyje bei atsikartoja skirtinguose failuose.

5. Verslo procesai: tai procesai, kurie vykdomi tam, kad pasiekti užsibrėžtų verslo tikslų.
6. Verslo politika ir taisyklės: verslo politika apibrėžia, kaip turi būti vykdomi verslo procesai, o taisyklės nurodo jų vykdymo metu taikomus apribojimus. Paveldėtoji sistema taip pat gali būti įvardinta kaip verslo politikos ir taisyklių dalis.

Šiuos paveldėtosios sistemos komponentus galima suskirstyti į atitinkamus sluoksnius, kaip parodyta pav. 2 (Sommerville, 2001).

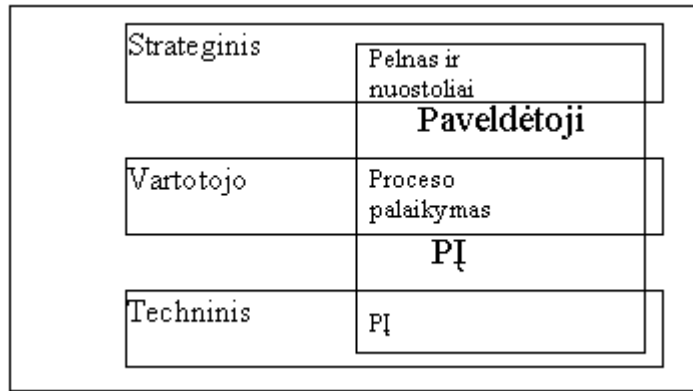
Verslo procesai
Taikomoji PĮ
Palaikančioji PĮ
Kompiuterinė įranga

2 pav. Sluoksninis paveldėtosios sistemos modelis

Kiekvienas sluoksnis priklauso nuo žemiau esančiojo bei sąveikauja su juo. Jeigu sąsajos tarp šių sluoksnių palaikomos, tuomet įmanoma pakeisti konkretų sluoksnį, nepaveikiant likusiųjų, tačiau praktikoje dažniausiai pakeitimai viename sistemos sluoksnyje pareikalauja atitinkamų pakeitimų ir kituose, esančiuose tiek žemiau, tiek ir aukščiau jo. Tai sąlygojama įvairių priežasčių:

1. Kartais keičiant vieną sistemos sluoksnį įdiegiamos naujos priemonės, todėl gali tekti pakoreguoti ir kitus sluoksnius, kad jie pajėgtų išnaudoti šių priemonių teikiamus privalumus.
2. PĮ keitimas sulėtina sistemos darbą, todėl tenka įsigyti naują kompiuterinę įrangą, kuri leistų pagerinti sistemos našumo charakteristikas, tačiau kartu ji leidžia realizuoti taikomosios PĮ sluoksnyje ir tas funkcijas, kurios naudojant senąją įrangą buvo neįmanomos.
3. Dažnai yra sunku palaikyti kompiuterinės įrangos sąsajas, ypač jeigu pasiūlomi radikalūs pokyčiai naujai įrangai.

Aprašytas sluoksninis paveldėtosios sistemos modelis atspindi inžinierių požiūrį, tačiau galimas ir kitas jo variantas, orientuotas į paveldėtos PĮ reikšmę ją naudojančioje įmonėje (žr. pav. 3) (Gold, 1998).



3 pav. Sluoksninis paveldėtosios PĮ modelis, orientuotas į jos vaidmenį įmonėje

Aukščiausias, strateginis, lygmuo parodo įmonės direktorių ir kito valdančiojo personalo požiūrį: jie paveldėtają PĮ traktuoja atsivėlgdami į jos įtaką įmonės pelnei arba nuostoliams.

Vidurinis, vartotojų, lygmuo atspindi požiūrį vartotojų, kurie specifikuoja ir naudoja sistemą kaip savo darbo įrankį.

Žemiausias, techninis, lygmuo pateikia PĮ palaikančiųjų darbuotojų požiūrį, kurie ją vertina pagal jos palaikomumo charakteristiką.

3.1.3 Paveldėtųjų sistemų struktūra

Dauguma paveldėtųjų sistemų buvo sukurtos dar prieš atsirandant objektiškai orientuotam projektavimo metodui, todėl dažnai paveldėtoji PĮ susideda ne iš tarpusavyje sąveikaujančių objektų, bet iš atskirų programų, kurių kiekviena sudaryta iš tam tikro funkcijų rinkinio. Kiekviena šių funkcijų realizuoja dalį sistemos atliekamų veiksmų bei yra iškviečiama kitų funkcijų pagalba.

Paveldėtųjų sistemų kūrime naudota funkcinio projektavimo metodologija pagrįsta programos dekompozicija į rinkinį tarpusavyje sąveikaujančių funkcijų, įskaitant sistemos būseną, kurios informacija jos dalinasi (vidinė funkcijų būsenos informacija yra palaikoma tik jų vykdymo metu).

Funkcinio projektavimo atveju algoritmas saugomas pačioje funkcijoje, o sistemos būsenos informacija žinoma visiems programos komponentams. Kai kuriose situacijose tokia architektūra kelia problemų, nes galimi atvejai, kuomet kuri nors funkcija pakeičia sistemos būseną taip, kaip kitos funkcijos to nesitiki. Tai yra vienas pagrindinių paveldėtosios PĮ trūkumų, nes ji dažniausiai buvo modifikuojama skirtingų žmonių, o vienas žmogus negali pilnai suvokti, kaip tarpusavyje sąveikauja skirtingos programos dalys.

Funkcinis projektavimo metodas yra efektyviausias projektuojant sistemą, kur saugomos informacijos kiekis apie jos būseną minimalus ir yra aiškus informacijos dalijimasis tarp jos komponentų. Todėl sistemos, kurių reakcija priklauso tik nuo vieno įėjimo ir kurios yra nepriklausomos nuo ankstesnių įėjimų reikšmių natūraliai orientuotos į funkcinę projektavimo ir programavimo paradigmą. Tokių sistemų kategorijai priklauso dauguma verslo sistemų, kurios apdoroja pavienius įrašus, o vieno įrašo apdorojimas nepriklauso nuo prieš tai buvusių įrašų apdorojimo rezultatu. Taigi visai natūralu tokių sistemų projektavimui naudoti funkcinį metodą.

Verslo sistemos sudaro didžiausią dalį paveldėtųjų sistemų klasėje. Egzistuoja dvi pagrindinės verslo sistemų grupės:

1. Paketus apdorojančios sistemos: įėjimo duomenys ir gauti rezultatai jose yra paduodami iš failų paketų pavidalu. Tokių sistemų pavyzdžiais galėtų būti algalapių sudarymo, sąskaitų išrašymo sistemos ir kt.
2. Transakcijas atliekančios sistemos: įėjimo duomenys ir gauti rezultatai jose paduodami kaip eilė transakcijų tarp DB ir vartotojo.

Tiek paketus apdorojančios, tiek ir transakcijas atliekančios sistemos funkcionuoja pagal įėjimo-apdorojimo-rezultatu modelį. Jos surenka įėjimo duomenis iš vieno ar daugiau šaltinių, juos apdoroja ir grąžina gautus rezultatus.

Vis dar eksploatuojant paveldėtas sistemas, naudojamas ir funkcinis projektavimo metodas. Tačiau jis vis dar gali būti taikomas ir naujai kuriamoms sistemoms, kai:

1. Bus realizuojamos duomenų apdorojimo sistemos, pagrįstos transakcijų vykdymu ir DB atnaujinimu. Tokio tipo sistemoms nereikia turėti informacijos apie anksčiau įvykdytas transakcijas, taigi objektai, saugantys privačius duomenis, nereikalingi.
2. Įmonė daug investavo į funkcinio projektavimo metodus, juos palaikančias automatizuoto programų kūrimo (*Computer-Aided System Design – CASE*) priemones ir personalo apmokymą. Perėjimo prie objektiškai orientuoto metodo rizika ir išlaidos gali tokiu atveju ir neatsipirkti.

Taigi svarbiausia šiais laikais užtikrinti, kad sistemos, suprojektuotos funkcinio metodu, galėtų sėkmingai būti integruotos su objektiškai orientuotomis sistemomis.

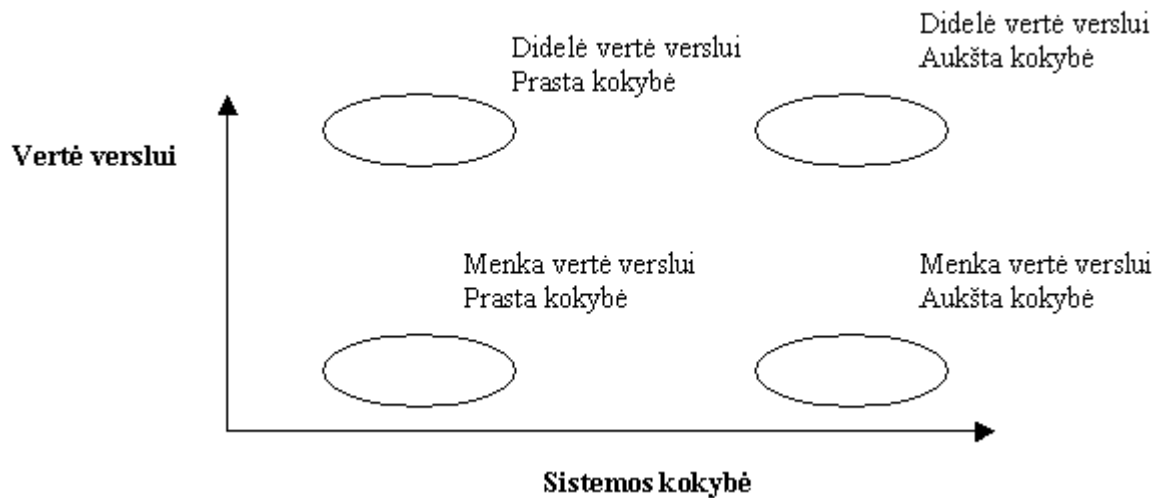
3.1.4 Paveldėtųjų sistemų įvertinimas

Viena iš didžiausių problemų, išskylanti naudojant paveldėtas sistemas, yra jų vertės nustatymas, leidžiantis įmonei nuspręsti, kokią tolimesnio sistemos vystymo strategiją pasirinkti.

Šiame skyriuje bus pasiūlyti du paveldėtosios sistemos įvertinimo modeliai: dviejų ir trijų matavimų.

3.1.4.1. Dviejų matavimų paveldėtosios sistemos vertės nustatymo modelis

Dviejų matavimų modelis remiasi sistemos įvertinimu verslo ir kokybės atžvilgiu (Sommerville, 2001).



4 pav. Dviejų matavimų paveldėtosios sistemos vertės nustatymo modelis

Galimi 4 tokio įvertinimo rezultatai (žr.pav. 4):

1. Prasta kokybė, menka vertė verslui: toliau naudoti tokias sistemas yra per brangu, o jų naudojimas versle neatsiperka. Geriausia jas išmesti.
2. Prasta kokybė, didelė vertė verslui: tokios sistemos yra neatsiejama verslo dalis, todėl jų išmesti negalima. Tačiau prasta jų kokybė rodo, kad jas sunku ir brangu palaikyti, todėl reikia atlikti jų rekonstrukciją arba pakeisti į panašaus tipo naują sistemą, jeigu tokia yra prieinama.
3. Aukšta kokybė, menka vertė verslui: tai tokios sistemos, kurios daug naudos verslui neatneša, tačiau dėl aukštos kokybės, jų palaikymas yra paprastas ir nebrangus. Būtų per didelė rizika jas pakeisti nauja sistema, todėl geriausiai tęsti jų palaikymą arba tiesiog jų atsisakyti.
4. Aukšta kokybė, didelė vertė verslui: šios sistemos yra labai naudingos verslui, o dėl aukštos kokybės jas palaikyti nebrangu, todėl geriausia tiesiog tęsti jų palaikymą.

Toliau bus aprašyta, kaip atlikti sistemos vertės nustatymą verslo ir kokybės atžvilgiu.

Vertės verslui nustatymas

Paveldėtosios sistemos vertės verslui nustatymas dažnai remiasi subjektyvia nuomone, todėl patikimų objektyvių metodų, taikytinų šiame procese, nėra. Vis dėlto atliekant sistemos vertės verslui nustatymą, derėtų remtis ne viena kokia nors nuomone apie sistemą, bet apklausti kuo daugiau su ja susijusių darbuotojų, ir galutinę jos vertę nustatyti apibendrinus visus gautus vertinimus.

Patartina išsiaiškinti, kaip į analizuojamą sistemą žiūri šie ja suinteresuoti asmenys:

1. Galutiniai sistemos vartotojai: kaip efektyviai sistema padeda jiems vykdyti verslo procesus? Kiek procentų sistemos funkcijų jie naudojami?
2. Įmonės klientai: ar sistemos naudojimas yra klientams nejaučiamas, o gal jų veiksmai vis dėlto yra jos ribojami? Ar jie priversti laukti dėl sistemos įtakos? Ar sistemoje pasitaikančios klaidos turi jiems tiesioginę įtaką?
3. Vadybininkai: ar jų manymu sistema efektyviai prisideda prie jų skyriaus sėkmės? Ar sistemos palaikymo išlaidos pateisinamos? Ar sistemos apdorojimo duomenys yra kritiniai jų skyriaus veiklai?
4. Informacinių technologijų vadybininkai: ar sunku surasti žmonių, sugebančių dirbti su sistema? Ar sistema nenaudoja resursų, kurie galėtų būti efektyviau išnaudoti kitų sistemų?
5. Įmonės vadovai: ar sistema ir su ja susiję verslo procesai padeda siekti užsibrėžtų verslo tikslų?

Sistemos kokybės įvertinimas

Kaip buvo rašyta ankstesniuose skyriuose, paveldėtoji sistema – tai ne vien PĮ, bet ir verslo procesai, kompiuterinė įranga, palaikančioji PĮ. Todėl vertinant sistemos kokybę, reikia pirmiausia įvertinti visus jos komponentus, ir tik po to – visą sistemą.

Verslo procesų kokybės vertinimas yra artimas paveldėtosios sistemos vertės verslui nustatymo procesui. Vis dėlto kadangi verslo procesai yra paveldėtosios sistemos dalis, gali prireikti daugiau informacijos, norint nustatyti jų kokybės lygį: prastos kokybės procesus reikia gerinti, o tai reiškia, kad gali tekti koreguoti ir pačią paveldėtają sistemą.

Vertinant verslo procesų, kuriuos padeda atlikti paveldėtoji sistema, kokybę vėl reikia apklausti su sistema susijusius asmenis. Šį kartą rekomenduojama gauti atsakymus į tokius klausimus:

1. Ar egzistuoja apibrėžtas proceso modelis, ir ar yra atitinkamos procedūros, padedančios patikrinti, ar šiuo modeliu naudojama?

2. Ar skirtingi organizacijos padaliniai naudoja tuos pačius procesus toms pačioms funkcijoms atlikti?
3. Kaip procese dalyvaujantys žmonės adaptavo jį savo vykdomai veiklai?
4. Ar yra būtini sąryšiai su kitais verslo procesais? Jeigu taip, ar tie sąryšiai akivaizdūs procese dalyvaujantiems asmenims?
5. Ar procesas efektyviai palaikomas paveldėtosios PĮ? Ar ji pateikia visą procesui reikalingą informaciją? Ar procesas reikalauja tuos pačius duomenis įvesti kelis kartus skirtingose vietose?

Įvertinus verslo procesų kokybę, reikia įvertinti paveldėtosios sistemos naudojamą kompiuterinę įrangą ir palaikančią PĮ, nes veikimo aplinkos veiksniai dažnai būna paveldėtosios PĮ keitimo priežastimi.

Vertinant sistemos veikimo aplinkos kokybę, reikia įvertinti kompiuterinės įrangos ir palaikančios PĮ priežiūros išlaidas, kompiuterinės įrangos gedimų skaičių per tam tikrą laiko periodą, palaikančios PĮ pataisymų dažnį. Taip pat reikia atsižvelgti į kompiuterinės įrangos ir palaikančios PĮ tiekėjų stabilumą – jeigu jie daugiau nebevykdo komercinės veiklos, tai gali reikšti, kad iš jų įsigyti produktai nebebus aptarnaujami.

Galiausiai reikia įvertinti pačią paveldėtąją PĮ. Šiuo atveju reikia atlikti tokių jos kokybei įtakos turinčių veiksnių analizę: PĮ suprantamumo, turimos dokumentacijos, naudojamo duomenų modelio, našumo, programavimo kalbos, konfigūracijos valdymo, testavimo duomenų ir su sistema dirbančio personalo kvalifikacijos.

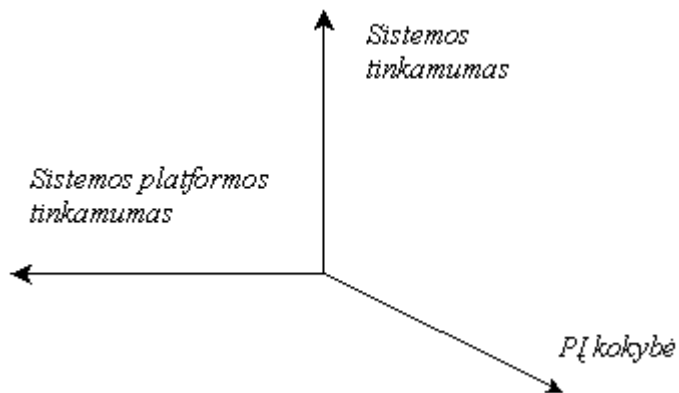
Vertinant visos sistemos kokybę, reikia atsižvelgti į:

1. Prašymų koreguoti sistemą skaičių: atliekami pakeitimai gadina sistemos struktūrą ir apsunkina kitų pakeitimų įgyvendinimą. Kuo šis skaičius didesnis, tuo prastesnė sistemos kokybė.
2. Skirtingų vartotojo sąsajų skaičių: šis veiksnys svarbus ekrano formomis pagrįstose sistemose, kur kiekviena forma gali būti laikoma, kaip atskira vartotojo sąsaja. Kuo labiau viena forma skiriasi nuo kitos, tuo didesnė tikimybė, kad jose bus nesutapimų ir perteklinės informacijos.
3. Duomenų kiekį: kuo didesnis duomenų (failų skaičius, DB dydis) kiekis, tuo sistema sudėtingesnė.

3.1.4.2. Trijų matavimų paveldėtosios sistemos vertės nustatymo modelis

Trijų matavimų paveldėtosios sistemos vertės nustatymo modelis remiasi sistemos tinkamumo, sistemos platformos tinkamumo ir sistemos kokybės įvertinimu (O'Byrne, 2000). Jis leidžia įvertinti paveldėtosios sistemos atgyvenimo laipsnį, analizuojant priežastinius paveldėtųjų sistemų vertės kriterijus bei jos vertę įtakančius poveikius.

Kiekvienas vektorius šiame modelyje (žr. pav. 5) vaizduoja vieną iš trijų minėtų matavimų. Kuo paveldėtosios sistemos vertė mažėja tam tikro matavimo atžvilgiu, tuo jam skirtos vektoriaus koordinatės tolsta nuo centrinio taško. Jeigu sistemos vertė to matavimo atžvilgiu išauga, vektoriaus koordinatė artėja prie centrinio taško. Centrinio taško koordinatės yra (0,0,0) – tai rodo, kad paveldėtosios sistemos atgyvenimo laipsnis yra nulinis.



5 pav. Trijų matavimų paveldėtosios sistemos vertės nustatymo modelis

Pirmasis paveldėtosios sistemos vertės matavimo kriterijus yra sistemos tinkamumas. Jis padeda įvertinti santykį tarp verslo ir informacinių technologijų strategijos bei įmonės vidinių taisyklių, apimančių dirbantį personalą, tradicijas ir vykdomus procesus.

Antrasis matavimo kriterijus yra sistemos platformos tinkamumas. Jis apima sistemos naudojamų kompiuterinės įrangos ir palaikančios PĮ įvertinimą, bet į jį neįeina platformos tinkamumo įvertinimas organizacijos aplinkai – šis veiksnys priklauso pirmajam matavimui. Antrasis kriterijus vertina tuos techninius sistemos aspektus, kurie yra valdomi informacinių technologijų skyriaus, bet kurių pats skyrius nekuria.

Trečiasis matavimas yra PĮ kokybė. Jis apima tik paveldėtąją sistemą sudarančios taikomosios PĮ vertinimą, t.y. programos kodo, architektūros ir konfigūracijos valdymo kokybės įvertinimą.

Šie trys matavimai lentelėje 1 yra suklasifikuoti ir susieti su priežastiniais paveldėtosios sistemos vertės kriterijais.

1 lentelė. Priežastiniai paveldėtosios sistemos vertės kriterijai

Priežastiniai paveldėtosios sistemos vertės kriterijai	
Sistemos tinkamumas	Sistemos tinkamumas verslo procesams
	Verslo procesų atitikimas organizacijos tikslams
	Sistemos technologijos tinkamumas įmonės vidinėms taisyklėms
Sistemos platformos tinkamumas	Kompiuterinės įrangos tinkamumas
	Operacinės sistemos tinkamumas
	Tinklų tinkamumas
	Sistemos kūrimo aplinkos tinkamumas
	Duomenų valdymo priemonių tinkamumas
PĮ kokybė	Konfigūracijų valdymo kokybė
	Architektūros kokybė
	Programos kodo kokybė

Paveldėtosios sistemos vertę apsprendžia ir įvairūs poveikiai, kuriuos būtų galima suskirstyti į keturias grupes: sistemos vertė, naudojimo paprastumas, palaikymo paprastumas ir migracijos paprastumas. Į kiekvieną šių grupių įeina atitinkami poveikiai, pateikti lentelėje 2.

Sistemos vertė – tai jos vertė organizacijai, padedant jai atlikti tam tikros dalykinės srities funkcijas. Jeigu ji mažėja, mažėja organizacijos konkurentabilumas atliekant tas funkcijas.

Į naudojimo paprastumo grupę įeinantys poveikiai įtakoja galutinius sistemos vartotojus, peržiūras vykdančių ir sistemą palaikančių personalą.

Palaikymo paprastumo grupė apima poveikius, susijusius su nuolatiniais pokyčiais, atliekamais sistemai tam, kad ji atitiktų verslo keliamus reikalavimus.

Migracijos paprastumo grupę sudaro poveikiai, susiję su stambiais arba strateginiais pokyčiais, kuriuos reikia atlikti sistemai tam, kad ji galėtų tenkinti naujus verslo poreikius, būtų perkelta į naują platformą ir t.t.

2 lentelė. Paveldėtosios sistemos vertę įtakoiantys poveikiai

Paveldėtosios sistemos vertę įtakoiantys poveikiai	
Sistemos vertė	Paskirties kritiškumas

2 lentelė. Paveldėtosios sistemos vertę įtakojantys poveikiai

Paveldėtosios sistemos vertę įtakojantys poveikiai	
	Patikimumas
Naudojimo paprastumas	Vartotojų pasitenkinimas
	Testavimo ir peržiūrų paprastumas
Palaikymo paprastumas	Palaikymo išlaidos
	Palaikymo resursų prieinamumas
	Programos dydis ir sudėtingumo laipsnis
	Priklausomybė nuo atskirų darbuotojų
Migracijos paprastumas	Naujų technologijų panaudojimo paprastumas
	Išplečiamumas

Remiantis **1** ir **2** lentelėse pateiktais duomenimis, galima apibrėžti, kas tai yra paveldėtosios sistemos atgyvenimo laipsnis: „Paveldėtosios sistemos atgyvenimo laipsnis yra sistemos trūkumų matas, išreikštas sistemos tinkamumo, sistemos platformos tinkamumo ir taikomosios PĮ kokybės lygiu, įvertinant tai, kad sistemos vertė mažėja, taip pat kaip mažėja jos naudojimo, palaikymo ir migracijos paprastumas“ (O’Byrne, 2000).

Paveldėtosios informacinės sistemos atgyvenimo laipsnį leidžia nustatyti LACE šablonas ir metodika, sudaryta iš tokių dalių:

- 1) paveldėtosios sistemos atgyvenimo laipsnį įtakančių poveikių nustatymo (*Legacy Effect Determination – LED*) šablono;
- 2) priežastinių paveldėtosios sistemos atgyvenimo laipsnio kriterijų nustatymo (*Legacy causal crITeria – LIT*) šablono;
- 3) paveldėtosios sistemos atgyvenimo laipsnio kriterijų/poveikių sąryšių nustatymo (*Legacy stAtus Cause/Effect – LACE*) šablono.

LED skirtas identifikuoti poveikius, turinčius įtaką informacinės sistemos atgyvenimo laipsniui; LIT pagalba nustatomos priežastys, kodėl naudojamos informacinės sistemos atgyvenimo laipsnis auga; LACE šablonas leidžia susieti tarpusavyje poveikius ir priežastis.

LED šablonas pagrįstas lentele **2**, papildžius ją dar vienu papildomu stulpeliu „Yra, Nėra arba Nenustatytas“, skirtu parodyti tam tikro paveldėtosios sistemos atgyvenimo laipsnį įtakančio kriterijaus buvimą, nebuvimą arba neapibrėžtumą nagrinėjamoje informacinėje sistemoje.

LIT šablonas pagrįstas lentele 1, papildyta papildomu stulpeliu, kuriame prie kiekvieno poveikio pažymima, ar sistema veikiama jo, ar ne. Jeigu poveikio įtaka sistemai dar neaiški, šiame stulpelyje reikia pažymėti, kad poveikis dar tiriamas.

Kuomet pilnai užpildomos LED ir LIT šablonų lentelės (jose turi būti tik reikšmės „Yra“ ir „Nėra“, o reikšmių, kad poveikis/kriterijus dar tiriami būti negali), sudaroma LACE šablono lentelė, kuri leidžia susieti kriterijus su poveikiais. Joje kiekviena eilutė skirta priežastiniam kriterijui, o kiekvienas stulpelis – poveikiui. Jeigu kriterijaus eilutėje ties tam tikrų poveikių stulpeliais pažymėtas X, tai reiškia, kad tas priežastinis kriterijus sukelia pažymėtą poveikį.

LACE šablonas ir metodologija leidžia įvertinti, kokie veiksniai lemia, kad informacinė sistema tampa paveldėtąja, tuo pačiu nustatant jos atgyvenimo laipsnį.

3.2 Paveldėtųjų sistemų tobulinimas

3.2.1 Paveldėtųjų sistemų tobulinimo proceso sunkumai

Informacinės technologijos leidžia komercinėms ir valstybinėms organizacijoms iš pagrindų pakeisti savo vykdomą veiklą, padidinant veiklos našumą bei įsisavinant naujas verslo kryptis. Tai padaryti padeda informacinės sistemos, kurios realizuoja naujus verslo procesus, apdoroja ir analizuoja verslui svarbius duomenis. Jų svarba skatina įmones daug dėmesio skirti informacinių sistemų tobulinimo proceso valdymui, kad šios atitiktų besikeičiančius reikalavimus ir naudotų naujausias informacines technologijas, leidžiančias padidinti įmonių konkurentabilumą rinkoje. Vis dėlto informacinių sistemų tobulinimas tampa vis sunkesnis joms senstant. Taip yra dėl keleto priežasčių:

1. Skirtingos sistemos dalys buvo sukurtos skirtingų žmonių, todėl joje nėra nusistovėjusio programavimo stiliaus.
2. Dalis arba visa sistema buvo sukurta naudojant pasenusią programavimo kalbą, todėl gali būti sunku surasti ją dar mokančių žmonių ir gali tekti sistemos palaikymą perduoti trečiajai šaliai už nemažą kainą.
3. Sistemos dokumentacija dažnai yra nepilna ir pasenusi. Kartais yra galimi tokie atvejai, kai vienintelė sistemos dokumentacija yra PĮ kodas, tačiau kartais ir jo įmonė gali neturėti.
4. Ilgi palaikymo metai gali būti pakenkę sistemos struktūrai ir padarę ją nepaprastai sunkiai suvokiamą. Sistema gali būti papildyta naujomis programomis ir sąveikauti su kitomis sistemos dalimis *ad hoc* metodų pagalba.

5. Sistema gali būti sukurta taip, kad užimtų mažiau vietos arba greičiau funkcionuotų, o ne taip, kad ją būtų galima lengviau suprasti. Tai sukelia nemažai problemų programuotojams, pripratusiems prie modernių PĮ kūrimo metodų ir nežinantiems anksčiau naudotų programavimo triukų.
6. Sistemos apdorjami duomenys gali būti saugomi skirtinguose, nesuderinamos struktūros failuose, todėl gali pasitaikyti duomenų dubliavimas, jie gali būti pasenę, netikslūs ir nepilni.

Dėl anksčiau išvardintų priežasčių, dauguma verslo įmonių, naudojančių paveldėtąsias sistemas, nuolat susiduria su fundamentalia dilema. Jeigu jos ir toliau naudosis paveldėtosiomis sistemomis bei atliks joms reikalingus pakeitimus, sistemų naudojimo išlaidos neišvengiamai išaugs. Jeigu jos nuspręs pakeisti paveldėtąsias sistemas naujomis, tai irgi nemažai kainuos, tačiau gali būti, kad naujai įdiegtos sistemos nebus taip sėkmingai naudojamos ir naudingos verslui, kaip senosios. Visi šie veiksniai skatina įmones nuolat ieškoti naujų programų inžinerijos metodų, kurie galėtų padėti joms prailginti paveldėtųjų sistemų gyvavimo ciklą bei sumažinti jų palaikymo išlaidas.

Toliau skyriuje yra aprašomi dažniausiai praktikoje naudojami paveldėtųjų sistemų tobulinimo būdai, didžiausią dėmesį skiriant jų modernizavimo metodams.

3.2.2 Paveldėtųjų sistemų tobulinimo metodai

Neįmanoma sukurti sistemos, kurios nereikėtų keisti eksploataavimo metu. Kai PĮ pradeda naudoti, vartotojams atsiranda naujų reikalavimų arba reikia keisti pradinius, keičiantis verslo taisyklėms. Dalis PĮ komponentų turi būti modifikuojami, norint juose ištaisyti esančias klaidas, padidinti našumą ar pagerinti kitas nefunkcines charakteristikas. Taigi po PĮ pristatymo vartotojams, ji nuolat turi būti tobulinama, kad patenkintų jų poreikius. Egzistuoja daug skirtingų PĮ tobulinimo metodų.

Sommerville pateikia tris PĮ tobulinimo būdus (Sommerville, 2001):

1. PĮ palaikymas: jo metu atliekami smulkūs pakeitimai, išplaukiantys iš besikeičiančių reikalavimų, tačiau jie nedaro įtakos sistemos architektūrai.
2. Architektūros transformavimas: tai radikalesnis PĮ tobulinimo metodas, nes jis apima reikšmingus jos architektūros pokyčius, dažniausiai pervedant sistemą nuo centralizuotos prie kliento-serverio architektūros.
3. PĮ rekonstrukcija: ji skiriasi nuo kitų būdų tuo, kad jos metu sistema nepapildoma naujomis funkcijomis – ji keičiama tik tam, kad taptų lengviau suprantama ir modifikuojama.

Rekonstrukcijos metu gali būti šiek tiek pakoreguota sistemos architektūra, tačiau tai nebus esminės modifikacijos.

J. Bisbal įvardina keturis PĮ tobulinimo metodus (Bisbal, 1995):

1. PĮ „apgaubimas“ (*wrapping*): naudojant šį metodą, esami sistemos duomenys, palaikančioji PĮ, taikomosios programos ir sąsajos apgaubiamos naujai sukurtomis sąsajomis. Tai leidžia toliau naudoti senus komponentus, papildyti jų atliekamas funkcijas naujomis ir pakartotinai panaudoti gerai ištestuotus komponentus, tokiu būdu išsiverčiant be papildomų investicijų į naujų kūrimą.
2. PĮ palaikymas
3. PĮ migracija: jos metu sistema pervedama į naują platformą, išlaikant visą jos funkcionalumą ir kiek įmanoma mažiau paveikiant egzistuojančią funkcinę ir verslo aplinką.
4. PĮ sukūrimas nuo pradžių (*redevelopment*): jo metu sukuriamas egzistuojančios paveldėtosios sistemos atitikmuo, pradedant nuo pat pradžių: naudojama visiškai kita programavimo kalba, DB valdymo sistema, kompiuterinės įrangos platforma ir kt.

PĮ tobulinimo metodus dar galima suskirstyti ir į tokias tris grupes (Comella-Dorda, 2000):

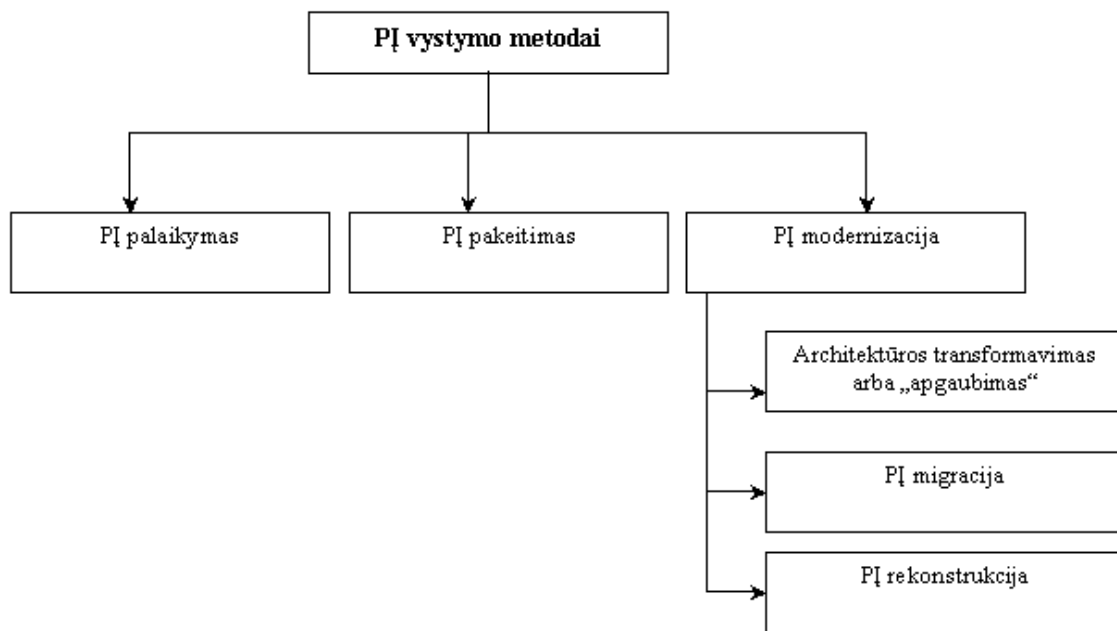
1. PĮ palaikymas
2. PĮ pakeitimas į panašią naują sistemą: jis atliekamas tuomet, kai paveldėtoji sistema jau nebeatitinka jai keliamų reikalavimų, o jos modernizacija yra neįmanoma arba labai brangi. Šis būdas naudojamas tuomet, kai paveldėtoji sistema yra nedokumentuota, atgyvenusi arba neišplečiama.
3. PĮ modernizacija: ji apima didesnius pakeitimus negu PĮ palaikymas, tačiau taip pat išsaugo žymesnę dalį senos sistemos. Atliekami pakeitimai gali būti sistemos restruktūrizacija, svarbūs funkciniai papildymai arba naujų PĮ atributų įvedimas.

Apibendrinant anksčiau pateiktas PĮ tobulinimo metodų klasifikacijas, galima sudaryti tokią jų hierarchiją, kaip pavaizduota pav. 6.

Visuose šaltiniuose, kaip vienas iš PĮ tobulinimo metodų, buvo įvardintas jos palaikymas, todėl jis pateikiamas ir hierarchijoje.

Antrasis metodas – PĮ pakeitimas. J. Bisbal, kaip vieną iš PĮ palaikymo būdų, pateikia jos sukūrimą iš naujo, tačiau jis nemini apie galimybę patiems naujos sistemos nekurti, o jeigu tai įmanoma, įsigyti jau sukurtą panašią PĮ, kaip nurodoma trečiuoju atveju. Kadangi abu šie metodai iš esmės nusako paveldėtosios sistemos atsisakymą ir jos pakeitimą kita, nesvarbu kurta ar piršta, juos galima sujungti į vieną – PĮ pakeitimo metodą.

Trečiuoju atveju pateiktas modernizacijos metodas iš esmės apima tokius smulkesnius metodus, kaip architektūros transformavimą (arba „apgaubimą“), migraciją bei rekonstrukciją. Nors kai kurie šaltiniai migraciją ir rekonstrukciją laiko vienu ir tuo pačiu dalyku, visgi J. Bisbal įvardija kai kuriuos esminius skirtumus tarp šių dviejų metodų (Bisbal, 1995). Jo manymu, dauguma rekonstrukcijos metodų siūlo visišką sistemos perdarymą iš naujo, todėl sistemos rekonstrukcija yra ne kas kita, kaip senos sistemos analizė ir perdarymas taip, kad atstatyti ją naujoje formoje. Dėl šios priežasties J. Bisbal laiko rekonstrukciją artimesne PĮ pakeitimo metodui. Tačiau tai nėra tas pats, nes sistemos nuo pat pradžių kurti nereikia, o ir sistema naujoje formoje turi senosios branduolį, kuris daugeliu atveju yra daug patikimesnis negu kad būtų visiškai iš naujo sukurtos sistemos atveju. Todėl rekonstrukcija kartu su migracija buvo priskirta PĮ modernizacijos metodų grupei.



6 pav. PĮ tobulinimo metodų hierarchija

3.2.2.1. Paveldėtosios PĮ pakeitimas nauja PĮ

Paveldėtoji PĮ gali būti sukurta iš naujo arba pakeista to paties funkcionalumo naujai įsigytu variantu, jeigu toks egzistuoja. Ši technologija taikoma toms paveldėtosioms sistemoms, kurios jau nebeatitinka verslo reikalavimų ir kurių modernizacija neįmanoma arba per brangi. Vis dėlto sustabdyti paveldėtųjų sistemų naudojimą ir pakeisti jas naujomis yra gana rizikinga dėl daugelio priežasčių:

1. Labai retai įmonė turi išsamią paveldėtosios sistemos dokumentaciją, kurioje detaliai būtų aprašyti visi joje atlikti pakeitimai. Dėl to labai sunku būtų parašyti specifikaciją naujai sistemai taip, kad ši savo funkcionalumu pilnai atitiktų senąją.
2. Verslo procesai ir paveldėtosios sistemos funkcijos yra labai tarpusavyje susiję, nes verslo procesai buvo pakeisti taip, kad galėtų pasinaudoti sistemos teikiamomis galimybėmis bei apeiti jos trūkumus. Pakeitus sistemą nauja, šiuos procesus vėl tektų keisti, o adaptacijos išlaidos ir pasekmės yra sunkiai prognozuojami.
3. Sistemoje gali būti įdiegtos svarbios verslo taisyklės, apie kurias neužsiminta turimoje dokumentacijoje. Jeigu pakeitus sistemą, šios taisyklės taps nebepalaikomos, įmonė gali patirti rimtų nuostolių.
4. Naujos sistemos sukūrimas reikalauja daug žmogiškųjų resursų, tačiau jie dažniausiai yra pilnai paskirstyti palaikymo užduotims ir yra nesusipažinę su naujomis technologijomis, kurios gali būti panaudotos naujoje sistemoje.
5. Naujos sistemos sukūrimas pats per save yra rizikingas procesas, todėl su nauja sistema gali iškilti nenumatytų problemų: vėluoti jos įdiegimas, viršijamas skirtas biudžetas, užsitęsęs kūrimo procesui gali paaiškėti, kad naujoji sistema naudoja jau pasenusias technologijas, o verslo procesai pasikeitę taip, kad ji nebetenkina jai keliamų reikalavimų.

3.2.2.2. PĮ palaikymas

PĮ palaikymas yra įprastas sistemos keitimas po to, kai ji buvo pristatyta vartotojui. Keitimas apima specifikavimo, projektavimo, programavimo metu padarytų klaidų ištaisymą, bei naujų reikalavimų realizaciją. Jis atliekamas modifikuojant egzistuojančius sistemos komponentus arba pridedant naujus, jeigu to reikia.

Egzistuoja trys PĮ palaikymo tipai (Sommerville, 2001):

1. PĮ klaidų ištaisymas: ištaisyti programavimo metu padarytas klaidas yra santykinai nebrangu. Projektavimo klaidos yra sunkiau ištaisomos, nes jos gali apimti keleto sistemos komponentų perdarymą. Pačios brangiausios taisymo atžvilgiu – reikalavimų specifikacijos klaidos, nes jos gali sąlygoti visos sistemos perprojektavimą.
2. PĮ pritaikymas kitai veikimo aplinkai: jis gali būti reikalingas kuomet keičiasi kai kurie sistemos veikimo aplinkos elementai, pvz., kompiuterinė įranga, operacinė sistema arba palaikančioji PĮ.

3. PĮ funkcionalumo papildymas arba modifikavimas: jis atliekamas tuomet, kai keičiasi sistemos reikalavimai, įvykus pasikeitimams įmonėje arba jos vykdomuose verslo procesuose. Šis palaikymo tipas praktikoje atliekamas dažniausiai.

Nors PĮ palaikymas reikalingas bet kokios sistemos tobulinimo procese, jis turi kai kurių apribojimų paveldėtųjų sistemų atveju:

- 1) naujų technologijų panaudojimas, siekiant papildyti sistemą verslo konkurentabilumą didinančiomis funkcijomis, yra labai apribotas;
- 2) palaikymo išlaidos ilgainiui labai išauga, tai pat vis sunkiau surasti atgyvenusias technologijas išmanančių specialistų;
- 3) dažnai dėl sistemos konceptualaus integralumo mažėjimo, bendra smulkių pakeitimų įtaka sistemai yra daug didesnė negu vieno pakeitimo. Informacinės sistemos laikui bėgant išsiplečia, kadangi nenaudojamo kodo, apsunkinančio sistemos palaikymą, pašalinimui neskiriama pakankamai dėmesio ir lėšų.

Dėl šių apribojimų, paveldėtosioms sistemoms palaikymas yra retai atliekamas, o pasak J. Bisbal, jeigu sistemą galima palaikyti, tai ji nėra paveldėtoji (Bisbal, 1995).

3.2.2.3. PĮ modernizavimas

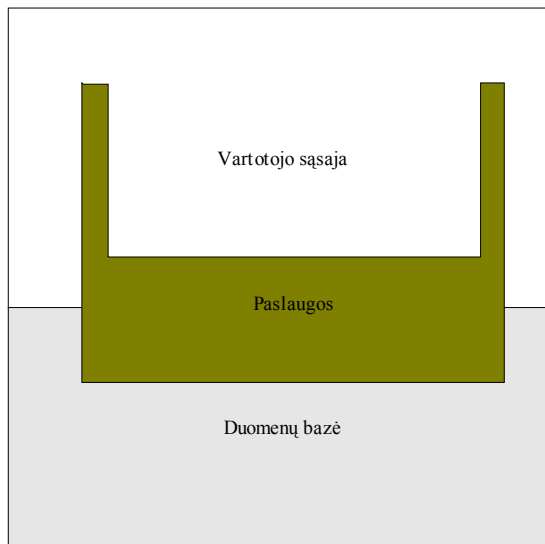
PĮ modernizacija gali būti atlikta vienu iš trijų metodų: architektūros transformacija arba „apgaubimu“, PĮ migracija arba PĮ rekonstrukcija.

Architektūros transformacija arba „apgaubimas“ dažniausiai taikomas norint sistemą iš centralizuotos architektūros pervesti į kliento-serverio. Šio paveldėtosios sistemos modernizacijos metodo pasirinkimą lemia žemiau išvardintos priežastys:

1. Kompiuterinės įrangos išlaidos: paskirstytos kliento-serverio sistemos įsigyjimas ir palaikymas kainuoja mažiau negu analogiško galingumo didelės mašinos (*mainframe*) įsigyjimas.
2. Vartotojo sąsajos viltys: dažniausiai paveldėtosios sistemos turi tekstinėmis formomis pagrįstą vartotojo sąsają. Tačiau dauguma šiuolaikinių vartotojų tikisi grafinės vartotojo sąsajos, kuri padeda paprasčiau bendrauti su sistema. Tokio tipo sąsaja reikalauja nemažai skaičiavimų ir yra įmanoma tik kliento-serverio architektūroje.
3. Paskirstytas priėjimas prie sistemų: dauguma įmonių paskirsto savo padalinius skirtingose vietose, todėl ir jų informacinės sistemos turi būti pasiekiamos iš skirtingų vietų ir su skirtinga įranga.

Kaip ten bebūtų, modifikuoti paveldėtosios sistemos architektūrą yra rimtas iššūkis ir tuo pačiu labai brangus procesas, todėl prieš jį pradėdamos organizacijos turėtų kruopščiai įvertinti savo paveldėtas sistemas, kad įsitikinti, jog iš architektūros transformacijos bus gauta nemaža nauda.

Sunkiausia architektūros transformacijos procese yra tai, kad dauguma paveldėtųjų sistemų turi tokią struktūrą, iš kurios nelengva išskirti nepriklausomus komponentus. Dažniausiai vartotojo sąsaja, sistemos atliekamos paslaugos bei DB yra tarpusavyje susipynę, o ne atskiri paveldėtosios sistemos sluoksniai (žr. pav. 7).



7 pav. Paveldėtųjų sistemų struktūra

Tokiose situacijose transformuoti sistemos architektūrą iš centralizuotos į paskirstytą būtų per sudėtinga, todėl tam naudojamas „apgaubimo“ metodas. Jo pagalba egzistuojantys duomenys, palaikančioji PĮ, taikomosios programos ir naudojamos sąsajos yra „apgaubiamos“ naujomis sąsajomis. „Apgaubti“ komponentai funkcionuoja kaip serveris, kuris atlieka funkcijas, reikalingas išoriniam klientui, kuriam nereikia žinoti, kaip tos funkcijos realizuotos. „Apgaubimas“ leidžia organizacijoms pakartotinai panaudoti gerai ištestuotus senus komponentus, kuriais jos pasitiki ir į kuriuos buvo daug investavusios.

Kuomet PĮ pakeitimas kita yra per daug rizikingas, o architektūros transformacija arba „apgaubimas“ netinka, paveldėtosios sistemos gyvavimo ciklui prailginti pasirenkamas PĮ migracijos metodas. Nors jis yra daug sudėtingesnis už architektūros transformacijos metodą, sėkmės atveju jo nauda būna daug didesnė nei anksčiau aprašytų metodų. PĮ migracija leidžia

pernešti paveldėtą sistemą į naują platformą, išlaikant visą jos funkcionalumą ir kiek įmanoma mažiau keičiant veikimo bei verslo aplinkas.

Kai kurios migracijos metu atsirandančios problemos yra bendros visiems programų inžinerijos projektams ir yra detalai ištyrinėtos. Tai sistemos kūrimas, testavimas, DB modelio parinkimas. Tačiau kai kurie klausimai yra būdingi tik migracijos procesui; dažniausiai pasitaikantys iš jų yra DB užpildymas senais duomenimis, atnaujinamos sistemos testavimas bei paveldėtosios sistemos naudojimo sustabdymas.

Prieš pradėdant migracijos procesą reikia gerai išanalizuoti turimą sistemą, kad išsiaiškinti jos galimybes, sąveikas su kitomis sistemomis ir pan. To nepadarius yra didelė tikimybė, kad reikalavimų specifikacija naujai sistemai bus nepilna arba neteisinga, vadinasi, migracija bus nesėkminga.

Pirmiausia migracijos metu reikia paveldėtosios sistemos duomenis perkelti į naują sistemą. Tam reikia palyginti seną duomenų struktūrą su planuojama kurti nauja ir atlikti reikiamas duomenų transformacijas. Duomenis galima perkelti visus iš karto arba palaipsniui, padalinant juos į atskirus fragmentus. Jeigu senieji duomenys prastos kokybės, gali tekti atlikti jų išvalymą.

Apie 80 proc. viso migracijos proceso laiko užima galutinės sistemos testavimas. Įvertinant paveldėtos sistemos kritiškumą, galutinės sistemos funkcijų testavimo metu gaunami rezultatai turi būti lygiai tokie patys, kaip ir vykdant analogiškas paveldėtosios sistemos funkcijas. Todėl nepatartina migracijos metu kurti naujas funkcijas, kad būtų įmanoma sulyginti senos ir naujos sistemos darbo rezultatus. Jeigu vis dėlto naujo funkcionalumo reikia, jį patartina prie galutinės sistemos prijungti tik atlikus pradinę senosios sistemos migraciją.

Paskutinis migracijos etapas yra paveldėtosios sistemos naudojimo sustabdymas. Egzistuoja trys būdai, kaip atlikti šį etapą.

Pirmasis būdas – sustabdyti ir paleisti (*cut-and-run*) – siūlo sustabdyti paveldėtosios sistemos naudojimą ir pradėti naudoti atnaujintą jos variantą. Daugeliu atveju šis būdas yra labai pavojingas, nes naujosios sistemos blogo veikimo atveju, organizacija rizikuoja netekti svarbių jai duomenų bei vilkinti svarbių verslo procesų vykdymą.

Antrasis būdas – fazinė sąveika (*phased interoperability*) – rekomenduoja atsisakyti paveldėtosios sistemos palaipsniui: nustoti naudoti vieną jos dalį ir ją pakeisti atitinkama naujos sistemos dalimi, paskui atsisakyti sekančios senos sistemos dalies ir t.t. Šis būdas yra per daug sudėtingas dėl monolitinės paveldėtųjų sistemų struktūros, pavaizduotos pav. 7

Trečiuoju – lygiagretaus veikimo (*parallel operations*) – būdu paveldėtoji ir atnaujinta sistema naudojamos kartu ir tik gerai ištestavus atnaujintą sistemą bei įsitikinus jos korektišku veikimu, paveldėtosios sistemos naudojimas sustabdomas. Šis metodas apsunkina sistemos vartotojų darbą, nes jie turi atlikti tuos pačius veiksmus tiek su senąja, tiek ir su naująja sistema.

Kadangi visi šie būdai turi trūkumų, dažniausiai atliekant paveldėtųjų sistemų migraciją naudojama jų kombinacija.

PI rekonstrukcija gali būti ne vien tik atskiras PI modernizavimo būdas, bet ir aukščiau aprašytų metodų dalis – plačiau apie ją rašoma sekančiame skyriuje

4 PROGRAMINĖS ĮRANGOS REKONSTRUKCIJA

4.1 Rekonstrukcijos samprata

PĮ rekonstrukcijos esmė yra pagerinti arba transformuoti egzistuojančią PĮ taip, kad ji taptų lengviau suprantama, kontroliuojama ir panaudojama iš naujo. Šis metodas leidžia surasti bei pakartotinai panaudoti vertingas paveldėtosios sistemos dalis, sumažinti aukštas jos palaikymo išlaidas ir sukurti sąlygas tolimesniam tobulinimui.

Iš esmės, rekonstrukcija – tai paveldėtosios sistemos, kuri tapo per brangi palaikyti ir kurios architektūros arba realizacijos priemonės yra atgyvenusios, perrašymas panaudojant šiuolaikines PĮ ir/arba kompiuterinės įrangos technologijas (Rosenberg, 1996). Sudėtingiausia rekonstrukcijos procese yra paveldėtosios sistemos suvokimas, nes dažniausiai reikalavimų, architektūros ir išeities kodo aprašymų nėra arba jie jau pasenę.

PĮ rekonstrukcijos tikslas yra suvokti paveldėtąją sistemą (jos specifikaciją, architektūrą, realizaciją) ir perrašyti ją, kad pagerinti funkcionalumą, našumą arba realizaciją. Galima būtų įvardinti 4 pagrindinius rekonstrukcijos siekius (Rosenberg, 1996): PĮ paruošimas jos funkcionalumo išplėtimui, PĮ palaikymo pagerinimas, migracija į naują platformą ir patikimumo pagerinimas.

Pagrindinės PĮ rekonstrukcijos pranašumo prieš kitus anksčiau nagrinėtus PĮ tobulinimo metodus priežastys yra šios (Sommerville, 2001):

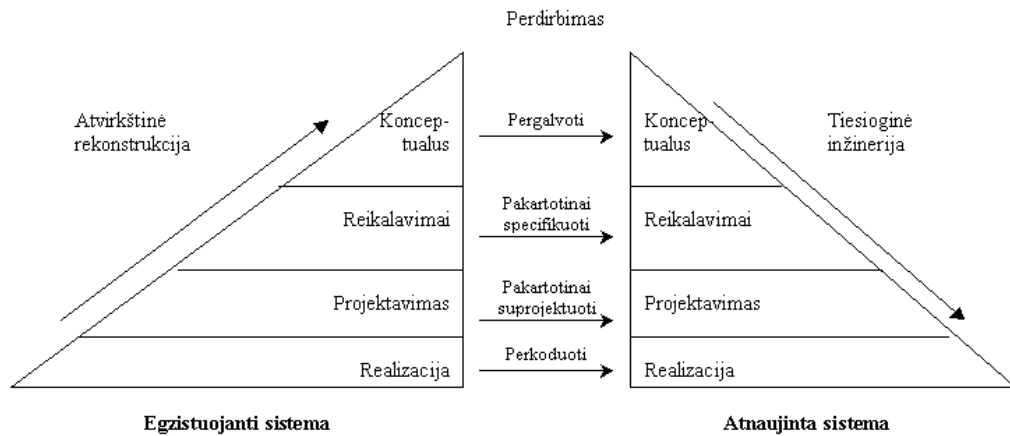
- 1) mažesnė rizika: kurti organizacijai gyvybiškai svarbią sistemą iš naujo visuomet yra labai rizikinga, nes atsiranda klaidų tikimybė specifikacijoje, gali iškilti įvairios realizacijos problemos ir kt.
- 2) mažesnės išlaidos: rekonstrukcijos išlaidos yra žymiai mažesnės negu naujos sistemos kūrimo.

Taip pat rekonstrukcijos metodą galima rinktis ir dėl to, kad (Renaissance Framework):

- 1) galima geriau išnaudoti turimo personalo žinias ir nereikia samdyti daug naujų darbuotojų;
- 2) galima išsiaiškinti verslo taisykles, kurios yra įterptos į rekonstruojamą sistemą;
- 3) galima atlikti paveldėtosios sistemos atnaujinimą palaipsniui, kuomet yra pakankamai resursų ir taip, kad vartotojai galėtų prisitaikyti prie pasikeitusių sistemos modulių.

4.2 Rekonstrukcijos modelis

Pav. 8 pavaizduotas PĮ rekonstrukcijos modelis, kuris susieja rekonstrukcijos procesus su įprastais PĮ kūrimo procesais (Rosenberg, 1996).



8 pav. PĮ rekonstrukcijos modelis

Atvirkštinės inžinerijos metu yra keičiamas paveldėtosios sistemos atvaizdavimo abstrakcijos lygmuo: jis palaipsniui didėja, kylant aukštyn sistemos abstrakcijos lygiais.

Perdirbimo metu atliekami pakeitimai sistemos atvaizdavime, nekeičiant abstrakcijos lygmens. Tam tikra sistemos charakteristika perdirbama sistemos abstrakcijos lygmenyje, kuriame ji yra aiškiausiai išreikšta. Norint pakeisti programos kodą, perkodavimas atliekamas realizacijos lygmenyje. Šiuo atveju atvirkštinės rekonstrukcijos atlikti nereikia. Tačiau norint pakeisti sistemos architektūrą (ją pakartotinai suprojektuoti), reikalavimus arba konceptualų modelį, prieš tai atitinkamame abstrakcijos lygmenyje įvykdoma atvirkštinė rekonstrukcija.

Tiesioginės inžinerijos metu egzistuojanti informacija apie sistemą pakeičiama perdirbtąja, pamažu mažinant abstrakcijos lygį.

4.3 Rekonstrukcijos proceso etapai

Visas PĮ rekonstrukcijos procesas gali būti suskaidytas į 5 etapus (Rosenberg, 1996):

- 1) rekonstrukciją vykdančio personalo parinkimas;
- 2) projekto įvykdomumo analizė;
- 3) paveldėtosios sistemos analizė ir rekonstrukcijos planavimas;
- 4) rekonstrukcijos įgyvendinimas;
- 5) testavimas ir perėjimas prie atnaujintos sistemos.

Rekonstrukciją vykdančias personalas valdys visą jos procesą nuo pradžios iki pabaigos, todėl jam reikia suteikti žinių apie PĮ rekonstrukciją, technologinių pakeitimų valdymą ir PĮ priežiūros procesų naudojimą. Šis personalas bus atsakingas už tokias užduotis, kaip rekonstrukcijos

įgyvendinimo plano sudarymas, jai atlikti skirtų įrankių įsigijimas bei įsisavinimas ir rekonstrukcijos proceso stebėjimas.

Projekto įvykdomumo analizės metu reikia įvertinti organizacijos poreikius ir tikslus, kuriuos tenkina egzistuojanti sistema, nes labai svarbu, kad pasirinkta rekonstrukcijos strategija atitiktų organizacijos kultūrinės normas. Taip pat reikia įvertinti paveldėtųjų sistemų vertę, kad būtų galima įsitikinti, ar PĮ rekonstrukcija atsipirks.

Paveldėtosios sistemos analizės ir rekonstrukcijos planavimo etapą galima suskaidyti į 3 žingsnius: paveldėtosios sistemos analizę, atnaujintos sistemos charakteristikų identifikavimą ir atnaujintos sistemos funkcionalumo įvertinimo kriterijų rinkinio sudarymą.

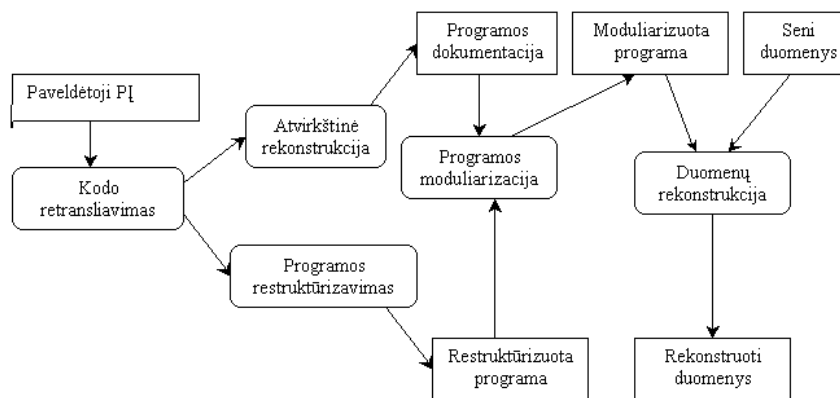
Rekonstrukcijos įgyvendinimo etapo metu atliekama atvirkštinė rekonstrukcija ir tiesioginė inžinerija, t.y. atnaujinama paveldėtoji sistema.

Testavimo ir perėjimo prie naujos sistemos metu, atnaujinta sistema testuojama ir diegiama. Testavimas niekuo nesiskiria nuo to, kuris atliekamas naujai kuriamai sistemai. Tačiau rekonstrukcijos atveju galima panaudoti tuos testų rinkinius, kurie buvo naudojami paveldėtajai sistemai testuoti, nes taip lengviau įsitikinti, kad atnaujintos sistemos funkcijos gražina tuos pačius rezultatus, kaip ir jas atitinkančios paveldėtosios sistemos funkcijos.

4.4 Rekonstrukcijos metodai

PĮ įgyvendinimo etapą galima išskaidyti taip, kaip parodyta pav. 9 (Sommerville, 2001). Tačiau rekonstrukcijos metu gali būti atliekami nebūtinai visi minėtame paveikslėlyje parodyti veiksmai: galbūt gali reikėti tik pertransliuoti programos kodą arba restruktūrizuoti duomenis.

Žemiau esančiuose skyriuose detaliam aprašomi kiekvienas iš pav. 9 pateiktų rekonstrukcijos metodų.



9 pav. Rekonstrukcijos įgyvendinimo procesas

4.4.1 Kodo retransliavimas

Tai yra paprasčiausias PĮ rekonstrukcijos metodas, nes jis nereikalauja PĮ supratimo. Jo metu paveldėtosios sistemos kodas pertransliuojamas į kitos, naujesnės programavimo kalbos kodą – programos struktūra ir organizacija išlieka nepakitusios.

Retransliuoti kodą gali reikėti dėl šių priežasčių:

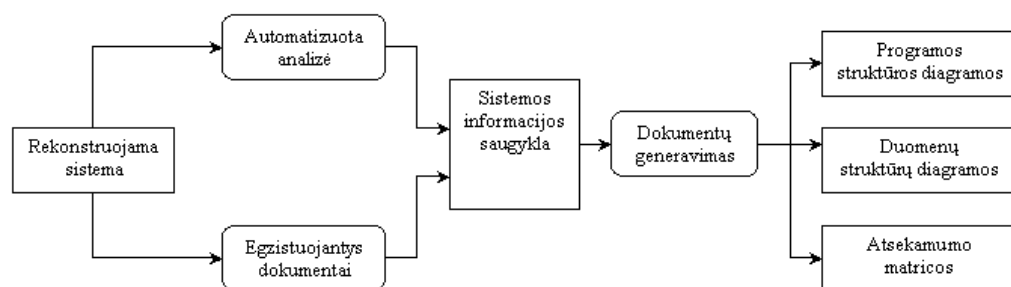
1. Kompiuterinės įrangos platformos pasikeitimas: organizacija gali norėti pakeisti naudojamą kompiuterinės įrangos platformą, kuriai naudojamos programavimo kalbos kompiliatoriai bus neprieinami.
2. Personalo įgūdžių stoka: įmonės darbuotojai gali stokoti naudojamos programavimo kalbos žinių, todėl jiems sunku ją palaikyti.
3. PĮ palaikymo stoka: naudojamos programavimo kalbos kompiliatorių tiekėjai gali būti bankrutavę arba nutraukę jų palaikymą.

Dažniausiai kodo retransliavimas atliekamas automatizuotų priemonių pagalba, tačiau visiškai šio proceso automatizavimas nėra įmanomas, tad dalį automatiškai retransliuoto kodo gali tekti pakoreguoti ranka.

4.4.2 Atvirkštinė rekonstrukcija

Atvirkštinė rekonstrukcija – tai programos analizės metodas, vykdomas tam, kad nustatytus sistemos komponentus ir jų sąryšius, sukurti naują programos atvaizdavimo formą, kuri būtų aukštesnio abstrakcijos lygmens negu programos kodas. Nuo jos efektyvumo priklauso viso PĮ rekonstrukcijos proceso rezultatai.

Atvirkštinės rekonstrukcijos procesas pavaizduotas pav. 10.



10 pav. Atvirkštinės rekonstrukcijos procesas

Atvirkštinės rekonstrukcijos procesas prasideda reikalavimų ir architektūros informacijos išgavimu iš programos išeities teksto ir egzistuojančių dokumentų. Iš gautos informacijos

generuojamos duomenų ir programos struktūros diagramos, naudojamos tolimesniame rekonstrukcijos procese.

4.4.3 Programos restruktūrizavimas

Kadangi paveldėtosios sistemos buvo palaikomos daugelį metų, jų struktūra pakito nuo dažnai atliekamų pakeitimų. Todėl dažnai reikia atlikti jų restruktūrizaciją, kad supaprastinti sudėtingas sąlygines išraiškas bei sunkiai suvokiamą valdymo kontrolę.

Atliekant restruktūrizaciją susiduriama su tokio pobūdžio problemomis, kaip komentarų, dokumentacijos praradimas, galingos kompiuterinės įrangos poreikiu, kad galėtų sėkmingai dirbti restruktūrizaciją atliekantys įrankiai. Pastarieji gali ir klysti, todėl šiame procese taip pat neapsieinama be rankinio darbo.

4.4.4 Programos modularizacija

Programos modularizacija yra programos reorganizavimo procesas, kurio metu susijusios programos dalys yra surenkamos į vieną modulį. Gali būti sukurti skirtingi modulių tipai:

- 1) duomenų abstrakcija gauta apjungus panašias duomenų struktūras ir jas apdorojančias funkcijas;
- 2) kompiuterinės įrangos moduliai, apjungiantys funkcijas, aptarnaujančias konkrečių kompiuterinės įrangos modulių;
- 3) funkciniai moduliai surenka į vieną vietą funkcijas, kurios atlieka tas pačias arba labai panašias užduotis;
- 4) procesą palaikantys moduliai apjungia visas funkcijas ir duomenis, palaikančius tam tikrą verslo procesą.

Programos modularizacija dažniausiai atliekama rankomis, analizuojant ir koreguojant programos kodą. Tam kad kokybiškai modularizuoti programą, reikia gerai išsiaiškinti jos komponentų tarpusavio sąveiką ir žinoti, ką kiekvienas iš jų daro.

4.4.5 Duomenų rekonstrukcija

Nors dauguma programų tobulinimo metodų orientuoti į PĮ, tačiau labai dažnai reikia keisti ir naudojamą duomenų modelį. Duomenų struktūros analizės ir reorganizavimo procesas vadinamas duomenų rekonstrukcija.

Duomenų rekonstrukcija atliekama tada, kai:

1. Pastebima duomenų degradacija: atliekant paveldėtųjų sistemų palaikymą, neišvengiamai tenka koreguoti ir naudojamą duomenų struktūrą. Tačiau laikui bėgant, jos pakeitimai gali sukelti klaidas, duomenų dubliavimąsi ir panašias problemas.
2. Apribojimai apdorojamų duomenų kiekiui: daugelyje paveldėtųjų sistemų buvo nustatyti apribojimai apdorojamam duomenų kiekiui, tačiau per eilę metų, duomenų kiekis išaugo ir reikia atlikti rekonstrukciją, kad tuos apribojimus panaikinti.
3. Architektūros evoliucija: perkeliant sistemą iš centralizuotos į paskirstytą architektūrą, reikia, kad duomenis būtų galima pasiekti iš daugelio nutolusių klientų. Tam gali tekti pakeisti duomenų modelį ir/arba duomenų bazių valdymo sistemą.

Skiriami tokie duomenų rekonstrukcijos metodai: duomenų išvalymas, duomenų išplėtimas ir duomenų migracija.

4.5 Rekonstrukcijos įgyvendinimo būdai

Egzistuoja trys rekonstrukcijos įgyvendinimo būdai, kurie vienas nuo kito skiriasi siūloma strategija, kaip sustabdyti paveldėtosios sistemos naudojimą ir pradėti naudoti atnaujintą jos variantą (Rosenberg, 1996): „didžiojo sprogo“ , palaipsninis ir evoliucinis.

„Didžiojo sprogo“ metodas siūlo pakeisti paveldėtąją sistemą jos atnaujintu variantu iš karto. Jo privalumas yra tas, kad paveldėtoji sistema iš karto pakeičiama į atnaujintą, taigi nereikia kurti jokių sąsajų tarp senų ir naujų jos komponentų. Tačiau jis turi nemažai trūkumų. Pirmiausia, jeigu atnaujinama didelė sistema, gali prireikti nemažai resursų, norint atlikti jos rekonstrukciją iš karto. Antra, didelės sistemos rekonstrukcijai prireiks daug laiko, ir jeigu bus siekiama ją atlikti vienu ypu, gali būti, kad ją užbaigus gauta rekonstruota sistema bus pasenusi kaip ir jos pirmtakė. Trečia, gali iškilti nemažų problemų, jeigu rekonstruota sistema neveiks taip, kaip paveldėtoji ir organizacija liks be tinkamos jos darbui informacinės sistemos.

Palaipsninis metodas siūlo suskaidyti paveldėtąją sistemą į atskiras dalis ir atlikti kiekvienos jų rekonstrukciją atskirai, o baigus tam tikros dalies atnaujinimą, ja pakeisti atitinkamą paveldėtosios sistemos dalį. Šio metodo plusas yra tas, kad kritinės paveldėtosios sistemos dalys gali būti pakeistos greičiau, be to, jeigu naujose dalyse atsirastų klaidų, jas bus lengviau ištaisyti. Pagrindiniai metodo trūkumai būtų tokie:

- reikia labai kruopščiai atlikti konfigūracijų valdymą, nes sistema atnaujinama per daugelį versijų;
- visos sistemos struktūra negali būti pagerinta – restruktūrizuojami tik atitinkami jos elementai.

Evoliucinio metodo atveju sistema taip pat suskirstoma į atskiras dalis, kurios vėliau yra pakeičiamos naujais savo variantais. Jis skiriasi nuo palaipsninio metodo tuo, kad sistemos dalys išskiriamos pagal jų atliekamą funkcionalumą, o ne remiantis programos struktūra. Taigi šis metodas leidžia modularizuoti sistemą bei sumažinti atskirų jos komponentų dydį. Vienas iš trūkumų būtų tas, kad reikia sistemoje surasti visas jos dalis, realizuojančias panašias funkcijas ir apjungti jas į vieną modulį.

4.6 Rekonstrukcijos metodo rizikos

PĮ rekonstrukcija dažniausiai naudojama tuomet, kai reikia sumažinti paveldėtosios sistemos palaikymo riziką ir išlaidas. Tačiau renkantis šį metodą taip pat reikia įvertinti tam tikras, tik jam būdingas rizikas.

Lentelėje pateikiama PĮ rekonstrukcijos rizikų grupės ir jų detalizacija.

3 lentelė. PĮ rekonstrukcijos rizikos

Rizikų grupė	Rizikos
Procesas	Labai aukštos rankiniu būdu atliekamos rekonstrukcijos išlaidos Rekonstruojamoms dalims pasirinktas neteisingas rekonstrukcijos metodas Rekonstruota sistema funkcionuoja blogai Rekonstrukcijos metu sukuriama daug kainuojančios dokumentacijos Pasirinkti rekonstrukcijos metodai neleidžia pasiekti jai iškeltų tikslų Paveldėtosios sistemos funkcionalumas pasikeičia atliekant jo rekonstrukciją
Atvirkštinė rekonstrukcija	Paveldėtoji sistema realizuota programavimo kalba, kuri neleidžia išgauti informacijos apie sistemos struktūrą tam, kad sudaryti jos reikalavimų ir architektūros specifikacijas Iš programos kodo išgauta objektų informacija yra nepilna arba neteisinga Prarandamos verslo taisyklės, „užsiūtos“ programos kode
Tiesioginė inžinerija	Atsiranda nauji reikalavimai ir sistema papildoma naujomis funkcijomis Sunku atlikti senų duomenų migraciją į atnaujintą sistemą
Personalas	Neturi žinių ir patirties rekonstrukcijos srityje Programuotojai specialiai atlieka savo darbą lėčiau, kad nemėgstamą rekonstrukcijos projektą padarytų mažiau efektyvų
Įrankiai	Priklausomybė nuo įrankių, kurie neatlieka visų prižadėtų funkcijų Branda – įrankio tiekėjo ir paties įrankio kokybės stabilumas

3 lentelė. PĮ rekonstrukcijos rizikos

Rizikų grupė	Rizikos
Strategija	Iš karto susitarta rekonstruoti visą sistemą Bandoma siekti nerealių tikslų Nėra rekonstrukcijos proceso plano

Lentelėje 4 pateikiami siūlymai kaip tam tikras PĮ rekonstrukcijos rizikas sumažinti (Santhanakrishnan, 2001).

4 lentelė. PĮ rekonstrukcijos rizikų sumažinimo būdai

Rizika	Sumažinimo būdas
Didelė dalis rekonstrukcijos turi būti atliekama rankiniu būdu	Atlikti vienos kurios nors pasirinktos funkcijos rekonstrukciją, kad įsitikinti, jog pasirinktas teisingas metodas. Atlikti gautų rezultatų įvertinimą ir susikurti priemones, kurios padėtų atlikti rekonstrukciją bent dalinai automatizuotai.
Suderinamumas tarp senų ir naujų duomenų	Gali tekti dubliuoti duomenis Sukurti priemones, kurios padėtų perkelti senus duomenis į atnaujintą sistemą ir atvirkščiai.
Rekonstruotos sistemos testavimas	Turi būti nustatyti aiškūs kriterijai, pagal kuriuos bus nustatomas rekonstruotos sistemos pilnumas ir teisingumas. Naudoti tuos pačius testavimo atvejus kaip ir paveldėtajai sistemai, kad įsitikinti, jog analogiškos rekonstruotos sistemos funkcijos veikia teisingai

5 SISTEMOS REKONSTRUKCIJOS REALIZACIJA

Magistro darbo projektinės dalies tikslas buvo atlikti Lietuvos miškų urėdijose naudojamos MKF programos rekonstrukciją bei paruošti visos miškų ūkio produkcijos gamybos-pardavimo apskaitos sistemos tobulinimo projektą.

Pirmiausia buvo atlikta turimos ir urėdijose naudojamos produkcijos gamybos-pardavimo apskaitos sistemos analizė ir pagal jos rezultatus paruošti sistemos reikalavimų bei architektūros specifikacijų dokumentai. Remiantis sukaupta informacija, MKF programos DOS versija buvo retransliuota į Visual FoxPro programavimo kalbą, tuo pačiu perkeliant ją į operacinės sistemos Windows platformą bei pakeičiant tekstinėmis formomis pagrįstą vartotojo sąsają į grafinę.

Atnaujintos MKF programos versija taip pat buvo papildyta nauju moduliu, leidžiančiu atlikti miškų urėdijos ūkinės veiklos analizę ir naudojančiu naujai suprojektuotą DB (seni moduliai pradės naudoti naująjį DB variantą tik tuomet, kai jie visi bus pilnai perkelti į Windows platformą). Naujoji sistemos versija buvo pilnai ištestuota ir detaliai aprašyta vartotojo vadove.

Šiame skyriuje pateikiamas sutrumpintas projektinės dalies dokumentacijos variantas, į kurį įeina reikalavimų, sistemos architektūros ir detalios architektūros specifikacijos.

5.1 Sistemos paskirtis

Šiuo metu valstybiniuose miškuose miško ruošą vykdo rangovų įmonės, kurios pagamintą produkciją perduoda miškų urėdijoms. Miškų urėdijų darbuotojai (dažniausiai – girininkai) periodiškai atlieka pagamintos produkcijos priėmimą. Jie priėmimo duomenis surašo į specialius popierinius dokumentus: apvaliosios medienos taškavimo lapą, iš kurio paskui sudaro apvaliosios medienos priėmimo kortelę.

Panašiai vykdomas ir medienos pardavimas. Urėdijos pardavimų vadybininkas, įvertindamas produkcijos atsargas įvairiose miško vietose, išduoda pirkėjams atitinkamus orderius pasiimti užsakytai medienai. Pirkėjui atvykus į nurodytą vietą, girininkas sudaro apvaliosios medienos taškavimo lapą, o pabaigoje – medienos išvežimo važtaraštį arba sąskaitą-faktūrą. Medienos tūrio ir kainos skaičiavimai atliekami kalkuliatoriumi, dėl ko gaištamas laikas bei padaroma klaidų. Be to, dėl to prastovi ir pirkėjų transportas.

Duomenys iš sudarytų pagamintos ir parduotos produkcijos dokumentų yra perkeliami į girininkijos DB. Taigi girininkas atlieka dvigubą darbą, pirmiausia informaciją apie produkciją surašydamas į popierinius dokumentus, o po to ją įvesdamas į kompiuterį.

Girininkijos gamybos-pardavimų duomenis privalo perduoti miškų urėdijai. Šiuo metu daugumoje urėdijų tai atliekama tik kartą per mėnesį, o toks periodiškumas trukdo operatyviai planuoti savo gamybos apimtį ir pardavimus. Operatyviam planavimui naudojama informacija dabar dažniausiai surenkama telefono pranešimų pagalba, todėl yra netiksli ir nepatikima.

Dalį įvardintų problemų padėtų išspręsti MKF programos rekonstrukcija: atnaujinta sistema sudarytų sąlygas miškų urėdijų ir girininkijų darbuotojams patogiau suvesti duomenis apie pagamintą ir parduotą medieną, be to, ji veiktų stabiliau negu DOS versija.

Rekonstrukcijos metu esamos sistemos architektūra būtų paruošta tolimesniam sistemos tobulinimui, įgyvendinant naujas funkcijas, kurios užtikrintų, kad miškų urėdijos DB būtų naujausia informacija apie pagamintą produkciją. Tai leistų joms greičiau aptarnauti savo klientus: medienos užsakovams tiksliai ir greitai būtų galima nurodyti, kuriose girininkijose yra jų pageidaujami sortimentai, o pardavimo vietoje paspartėtų tūrio ir kainų skaičiavimai, dėl ko pirkėjas greičiau gautų reikiamus pardavimo dokumentus ir išvengtų transporto prastovų.

Visų čia įvardintų tikslų įgyvendinimas padėtų miškų urėdijoms padidinti iš miškų ūkio produkcijos gaunamą pelną.

5.2 Reikalavimų specifikacija

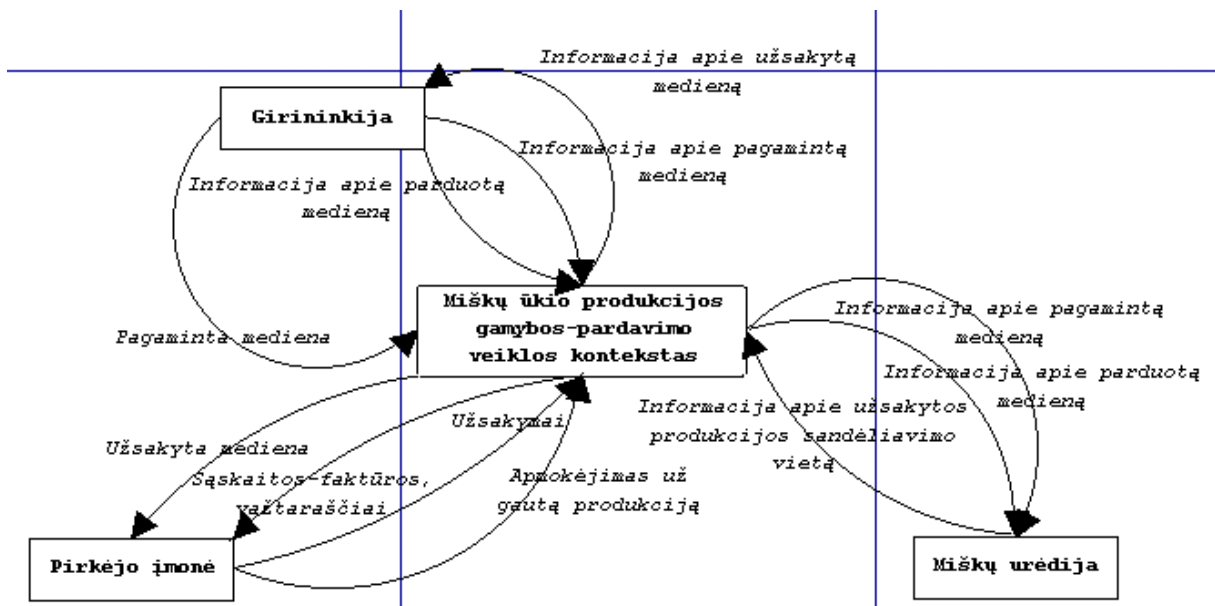
5.2.1 Veiklos kontekstas

Pav. 11 pavaizduotas miškų ūkio produkcijos gamybos-pardavimo veiklos, į kurią įsilies kuriama ir projektuojama sistemos, kontekstas.

Girininkija suformuoja du materialius (parduota mediena ir pagaminta mediena) ir du informacinius (dokumentai su informacija apie pagamintą ir parduotą medieną) srautus. Taip pat iš miškų urėdijos ji sužino apie tai, kokia produkcija užsakyta ir kas ją užsakė.

Pirkėjo įmonė užsisako urėdijos gaminamos produkcijos, apmoka už ją, o iš girininkijos gauna užsakytą produkciją ir pirkimą liudijančius dokumentus (sąskaitas-faktūras ir važtaraščius).

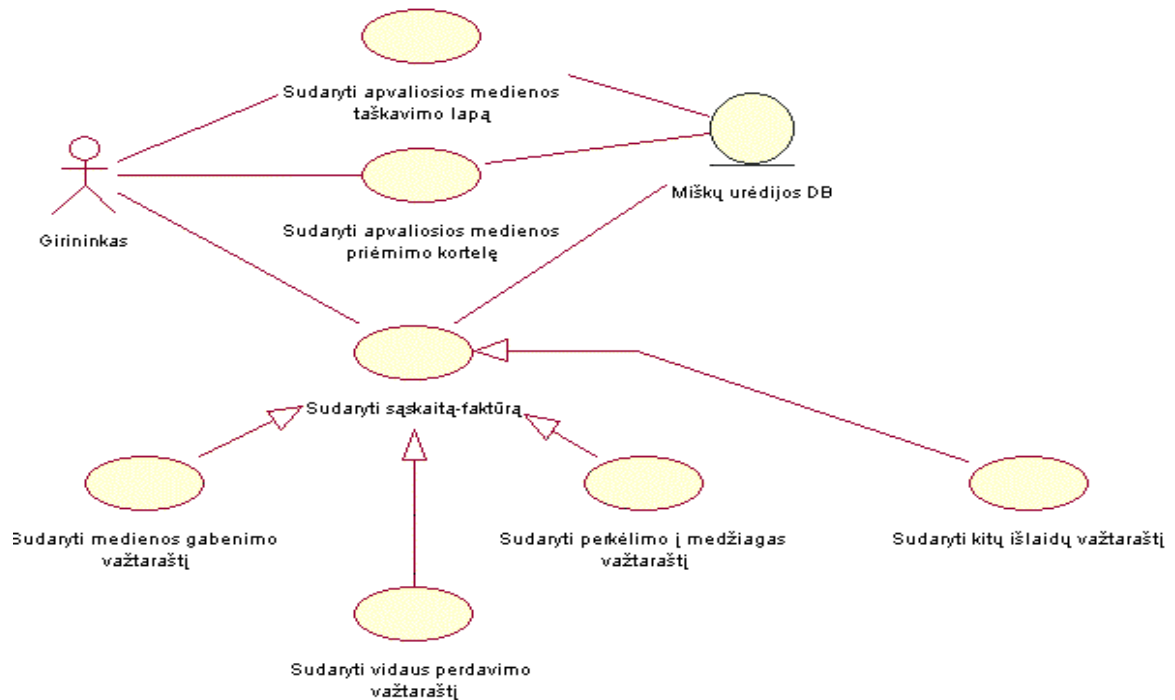
Miškų urėdija, gavusi informaciją apie pagamintą ir parduotą medieną, informuoja pirkėją apie tai, kur sandėliuojama jo pageidaujama produkcija ir ar urėdija jos turi.



11 pav. Miškų ūkio produkcijos gamybos-pardavimo veiklos kontekstas

5.2.2 Produkto veiklos sfera

MKF programos panaudojimo atvejų modelis, gautas atlikus jos DOS versijos analizę, pateikiamas pav. 12.



12 pav. Sistemos panaudojimo atvejų modelis

5.2.3 Funkciniai reikalavimai

Reikalavimas 1: atlikti urėdijoje pagamintos, parduotos ir turimos produkcijos analizę pasirinktam laiko periodui.

Reikalavimas 2: vartotojas turi turėti galimybę pasirinkti laiko periodą, kuriam nori atlikti urėdijos ūkinės veiklos analizę.

Reikalavimas 3: urėdijos ūkinės veiklos informacija pateikiama tokia hierarchija:

1. Girininkija

1.1. Girininkijos sandėliai

1.1.1. Sandėlyje turimos produkcijos rūšys

1.1.1.1. Sandėlyje esantys tam tikros produkcijos rūšies sortimentai

1.2. Girininkijos produkcijos rūšys

1.2.1. Tam tikros produkcijos rūšies sortimentai

1.2.1.1. Sandėliai, kuriuose yra konkretus sortimentas

Reikalavimas 4: vartotojui pasirinkus hierarchijos lygmenį “1. Girininkijos”, jam turi būti pateiktas urėdijos girininkijų sąrašas ir kiek kiekvienoje iš jų buvo pagaminta produkcijos, kiek jos buvo parduota ir kiek liko.

Reikalavimas 5: vartotojui pasirinkus hierarchijos lygmenį “1.1. Girininkijos sandėliai”, jam turi būti pateiktas pasirinktos girininkijos sandėlių (biržių) sąrašas ir kiek kiekviename iš jų pagaminta produkcijos, parduota ir koks jos likutis.

Reikalavimas 6: vartotojui pasirinkus hierarchijos lygmenį “1.1.1. Sandėlyje turimos produkcijos rūšys”, jam turi būti pateiktas pasirinktos girininkijos pasirinktame sandėlyje esančių produkcijos rūšių sąrašas ir kiek kiekvienos iš jų pagaminta, parduota ir koks jos likutis.

Reikalavimas 7: vartotojui pasirinkus hierarchijos lygmenį “1.1.1.1. Sandėlyje esantys tam tikros produkcijos rūšies sortimentai”, jam turi būti pateiktas pasirinktos girininkijos pasirinktame sandėlyje pasirinktai produkcijos rūšiai priklausančių sortimentų sąrašas ir kiek kiekvieno iš jų pagaminta, parduota ir koks jų likutis.

Reikalavimas 8: vartotojui pasirinkus hierarchijos lygmenį “1.2. Girininkijos produkcijos rūšys”, jam turi būti pateiktas pasirinktoje girininkijoje esančių produkcijos rūšių sąrašas ir kiek kiekvienos iš jų buvo pagaminta, parduota ir kiek liko.

Reikalavimas 9: vartotojui pasirinkus hierarchijos lygmenį “1.2.1. Tam tikros produkcijos rūšies sortimentai”, jam turi būti pateiktas pasirinktoje girininkijoje pasirinktos

produkcijos rūšies sortimentų sąrašas ir kiek kiekvieno iš jų buvo pagaminta, parduota ir kiek liko.

Reikalavimas 10: vartotojui pasirinkus hierarchijos lygmenį “1.2.1.1. Sandėliai, kuriuose yra konkretus sortimentas”, jam turi būti pateiktas sandėlių sąrašas, kuriuose yra pasirinktas sortimentas ir kiek jo kiekviename iš tame sąraše esančių sandėlių yra.

Reikalavimas 11: vartotojui pageidaujant, turi būti sudaroma analizės rezultatus atspindinti ataskaita, kurią galima peržiūrėti ekrane, išsisaugoti faile ir atsispausdinti.

Reikalavimas 12: vartotojui pageidaujant, analizės rezultatus turi būti galima atvaizduoti diagramų pavidalu (t.y. grafiškai).

Reikalavimas 13: sudaryti apvaliosios medienos taškavimo lapą. Duomenys, skirti taškavimo lapui sudaryti, yra surenkami miške: dažniausiai naudojamas laisvos formos pirminis dokumentas (girininkas nusibraižo lentelę sąsiuvinyje). Jo duomenys vedami į kompiuterį. Įvedimo metu programa turi automatiškai skaičiuoti tūrį.

Taškavimo lapo dokumentą sudaro dokumento antgalvė ir kūneliai

Kūneliai yra 2 tipų:

- medienos priėmimui vienetiniu metodu;
- medienos priėmimui grupiniu metodu.

Reikalavimas 14: įvesti apvaliosios medienos taškavimo lapo antgalvę. Ji sudaryta iš tokių laukų: numeris, sudarymo data, girininkija, kvartalo nr., sklypų nr., biržės nr., leidimo miškui kirsti nr., medieną perdavusio asmens tabelio nr., medieną priėmusio asmens tabelio nr.

Reikalavimas 15: įvesti apvaliosios medienos taškavimo lapo kūnelius vienetiniu metodu. Apskaitant medieną vienetiniu metodu, įvedama tokia informacija: sortimentas; ilgis, m; skersmuo, cm; apskaitomų sortimentų kiekis; tūris.

Reikalavimas 16: įvesti apvaliosios medienos taškavimo lapo kūnelį grupiniu metodu. Apskaitant medieną grupiniu metodu, įvedama tokia informacija: sortimentas; plotis, m; ilgis, cm; aukštis; tūris, erdm; glaudumo koeficientas; tūris, ktm.

Reikalavimas 17: įvesti apvaliosios medienos priėmimo kortelę. Apvaliosios medienos priėmimo kortelės dokumentas sudarytas iš antgalvės ir kūnelių. Kūneliai gali būti užpildomi dviem būdais:

- automatiškai, kortelei priskiriant sudarytus taškavimo lapus;
- įvedami rankiniu būdu sudaryti medienos priėmimo kortelės duomenys (bus realizuojama vėliau).

Taškavimo lapai medienos priėmimo kortelei gali būti priskiriami taip pat dviem būdais:

- jeigu pasirinktai medienos priėmimo vietai egzistuoja taškavimo lapai, kūneliams informacija priskiriama iš jų, arba gali būti sudaromi ir nauji taškavimo lapai;
- jeigu pasirinktai medienos priėmimo vietai nesudarytas nė vienas taškavimo lapas, kūneliams užpildyti sudaromi nauji taškavimo lapai.

Reikalavimas 18: įvesti apvaliosios medienos priėmimo kortelės antgalvę. Ji sudaryta iš tokių laukų: nr.; išrašymo data; girininkija; kvartalo nr.; biržės nr.; leidimo miškui kirsti nr.; sklypų nr.; kirtimų rūšis; ruošėjas (rangovas); priimamos medienos priėmimo periodas; produkciją priėmęs asmuo; taškavimo lapų, įtrauktų į kortelę, skaičius; bendras priimtas produkcijos kiekis.

Reikalavimas 19: turi būti galimybė apvaliosios medienos priėmimo kortelei priskirti taškavimo lapus.

Reikalavimas 20: sudaryti bazinį produkcijos kainininką. Prieš sudarinėjant pirminius išlaidavimo dokumentus, reikia įsivesti produkcijos kainininką. Kainininko įvedimas susideda iš jo antgalvės ir kūnelių įvedimo.

Reikalavimas 21: įvesti bazinio kainininko antgalvę. Ji sudaryta iš tokių laukų:

- sandėlio tipas: K, M,G (kirtavietė, miško sandėlys, Galutinis sandėlys);
- galiojimo pabaiga: įvedama kainininko galiojimo pabaigos data.

Reikalavimas 22: įvesti bazinio kainininko kūnelį. Jis sudarytas iš tokių laukų:

- sortimento kodas: įvedamas sortimento kodas. Jį galima pasirinkti arba iš sortimentų sąrašo, arba įvesti ranka;
- kaina, Lt.: įvedama sortimento kaina litais.

Reikalavimas 23: įvesti sąskaitos-faktūros antgalvę. Ji sudaryta iš tokių laukų: dokumento serija; dokumento nr.; išrašymo data; apskaitos mėnuo; siuntėjas; girininkija; produkcijos kategorija; sandėlio tipas; kvartalas; sandėlis; pirkėjas; kainininko ID; valiuta; apmokėjimo terminas; apmokėjimo būdas; išvežimas; dokumentą sudaręs asmuo.

Reikalavimas 24: įvesti sąskaitos-faktūros kūnelį. Jis sudarytas iš tokių laukų: sandėlio nr.; produkcijos kodas; produkcijos pavadinimas; storis; mato vnt.; kiekis; kaina; suma; suma su PVM; nuolaida.

Reikalavimas 25: sudaryti medienos gabenimo važtaraštį, vidaus perdavimo važtaraštį, perkėlimo į medžiagas važtaraštį, kitų išlaidų važtaraštį.

Kadangi šių produkcijos išlaidavimo dokumentų įvedimas yra lygiai toks pat, kaip ir anksčiau aprašytos sąskaitos-faktūros, toliau pateikiami tik jiems privalomi rekvizitai (žr. lentele 7 ir 8).

5 lentelė. Pirminių dokumentų rekvizitų priklausomybė antgalvei bei kūneliui

Rekvizito pavadinimas	Vieta
Išrašymo data	A
Dokumento Serija Nr.	A
Apskaitos mėnuo	A
Girininkija	A
Siuntėjas (padalinys atkraunantis medieną)	A
Pakrovimo vieta	A
Kvartalas ir/arba Biržės Nr. arba sandėlio Nr.	A
Pirkėjas	A
Sutarties Nr.	A
Apmokėjimo terminas.	A
Apmokėjimo būdas	A
Išvežimo būdas	A
Važtaraščio Nr. (jei papildomai sudaromas medienos gabenimo važtaraštis)	A
Dokumentą sudaręs asmuo	A
Kvartalas ir/arba Biržės Nr. arba sandėlio Nr.	K
Produkcijos kodas	K
Produkcijos pavadinimas	K
Storis (jei naudojam nestandartinės storio klasės)	K
Mato vienetas	K
Kiekis	K
Pardavimo kaina	K
Suma	K
PVM	K
Nuolaida	K

Čia: A – antgalvė, K – kūnelis

6 lentelė. Rekvizitų priklausymo išlaidavimo dokumentams lentelė

Rekvizito lauko vardas	PVM Sąskaita- faktūra	Medienos gabenimo važtaraštis	Perkėlimo į medžiagas važtaraštis	Vidaus pardavimo važtaraštis	Kitų išlaidų važtaraštis
Išrašymo data	+	+	+	+	+
Dokumento Serija	+	+			
Dokumento Nr.	+	+	+	+	+
Apskaitos mėnuo	+	+			
Girininkija	+	+	+	+	+
Siuntėjas	+	+	+	+	+
Pakrovimo vieta	+	+	+	+	+
Sandėlio Nr.	+	+	+	+	+
Pirkėjas (gavėjas)	+	+	+	+	+
Sutarties Nr	+				
Apmokėjimo terminas	+				
Apmokėjimo būdas	+				
Išvežimo būdas	+	+			
Važtaraščio Nr.	+	+			
Dokumentą sudariusio asmens Tabelio Nr	+	+	+	+	+
Sandėlio Nr.	+	+	+	+	+
Produkcijos kodas	+	+	+	+	+
Produkcijos pavadinimas	+	+	+	+	+
Storis	+	+			
Mato vnt.	+	+	+	+	+
Kiekis	+	+	+	+	+
Pardavimo kaina	+				
Suma	+				
Nuolaida	+				

5.2.4 Nefunkciniai reikalavimai

Reikalavimas 26: ryšių su pirkėjais palaikymo programos pagrindinis langas turi būti sudarytas iš dviejų dalių: nustatymų lango ir produkcijos gamybos-pardavimų rezultatų lentelės.

Reikalavimas 27: ryšių su pirkėjais palaikymo programos nustatymų langas turi būti išskviečiamas per meniu.

Reikalavimas 28: rezultatų lange pagrindinių objektų gamybos-pardavimų informacija gali būti išskleista, panaudojant "medžio" struktūrą.

Reikalavimas 29: naujos MKF programos vartotojo sąsaja turi atitikti jos DOS programos versiją.

Reikalavimas 30: sukurtą sistemą sudarančios taikomosios programos turi būti patogios naudoti savo vartotojui.

Reikalavimas 31: kuriamą sistemą sudarančias taikomasias programas turi lengvai įsisavinti urėdai, girininkai ir pardavimų vadybininkai.

Reikalavimas 32: sukurtos taikomosios programos turi atlikti savo funkcijas pakankamai greitai.

Reikalavimas 33: atsiradus klaidoms sukurtoje sistemoje, jos turi būti ištaisytos nedelsiant.

Reikalavimas 34: periodiškai turi būti išleidžiamos naujos sistemos versijos.

Reikalavimas 35: iškilus klausimams dėl naudojimosi sistema, vartotojai turi būti konsultuojami telefonu.

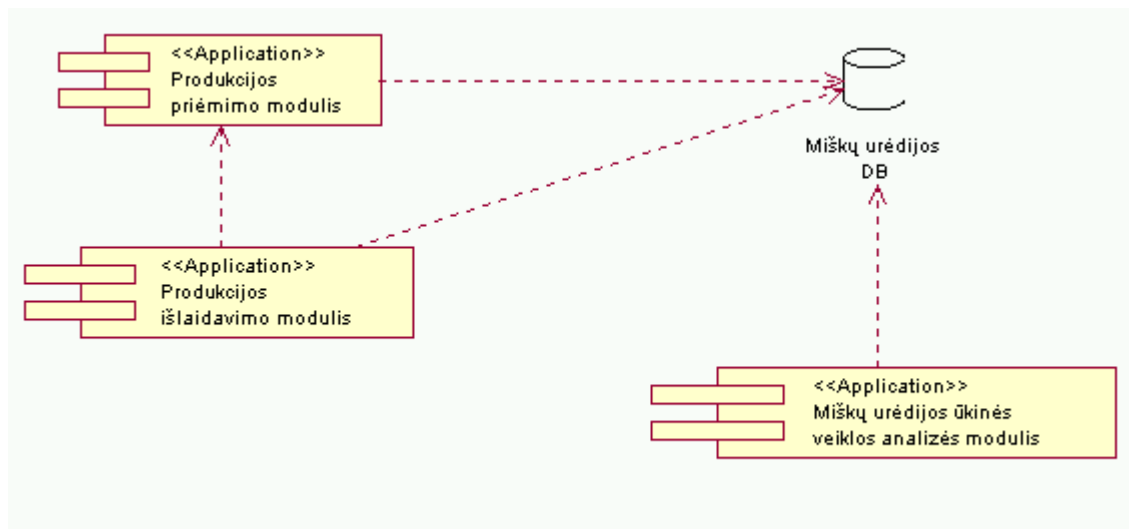
Reikalavimas 36: sukurta sistema turi turėti vartotojų prisijungimo vardus ir slaptažodžius, kad pašaliniai asmenys negalėtų įvesti, peržiūrėti ir koreguoti jų duomenų.

Reikalavimas 37: sukurtoji sistema turės būti suderinama su Lietuvos Respublikos Aplinkos ministro įsakymu dėl apvaliosios medienos apskaitos tvarkos (Aplinkos ministerija/Įsakymas/441/2001 08 31/Įsigalioja nuo 2001 09 08/Valstybės žinios'2001 Nr.77-2714), miškų įstatymo pakeitimo įstatymu (Lietuvos Respublikos Seimas/Įstatymas/IX-240/2001 04 10/Įsigalioja nuo 2001 07 01/Valstybės žinios'2001 Nr.35-1161) bei Generalinės miškų urėdijos prie Aplinkos ministerijos generalinio miškų urėdo įsakymu dėl miškų ūkio darbų bei apvaliosios medienos klasifikatorių patvirtinimo ir taikymo.

5.3 Architektūros specifikacija

5.3.1 Sistemos komponentų aprašymas

Šiame skyriuje detalai aprašyti sistemą sudarantys komponentai, pavaizduoti pav. 13.



13 pav. Sistemos komponentų diagrama

Produkcijos priėmimo modulis

Klasifikacija: posistemė.

Apibrėžimas: skirtas informacijos apie pagamintus sortimentus įvedimui.

Atsakomybės: ši posistemė leis vartotojui:

- sudaryti medienos taškavimo lapą pagamintai medienai, apskaitant ją vienetiniu arba grupiniu metodais;
- sudaryti medienos priėmimo kortelę.

Produkcijos išlaidavimo modulis

Klasifikacija: posistemė.

Apibrėžimas: skirtas įvesti informacijai apie miškų urėdijos parduodamą produkciją.

Atsakomybės: ši posistemė leis vartotojui:

- sudaryti bazinį produkcijos kainininką;
- sudaryti sąskaitą-faktūrą, medienos gabenimo važtaraštį, vidaus perdavimo važtaraštį, perkėlimo į medžiagas važtaraštį ir kitų išlaidų važtaraštį.

Miškų urėdijos ūkinės veiklos analizės modulis

Klasifikacija: posistemė.

Apibrėžimas: skirta atlikti miškų urėdijos ūkinės veiklos analizę pagal girininkijas, jose esančius sandėlius bei jose gaminamos produkcijos rūšis.

Atsakomybės: ši posistemė leis vartotojui sužinoti pagamintos/parduotos/esamos produkcijos kiekius pagal sandėlius arba pagal produkcijos rūšis pasirinktam laiko periodui.

Miškų urėdijos DB

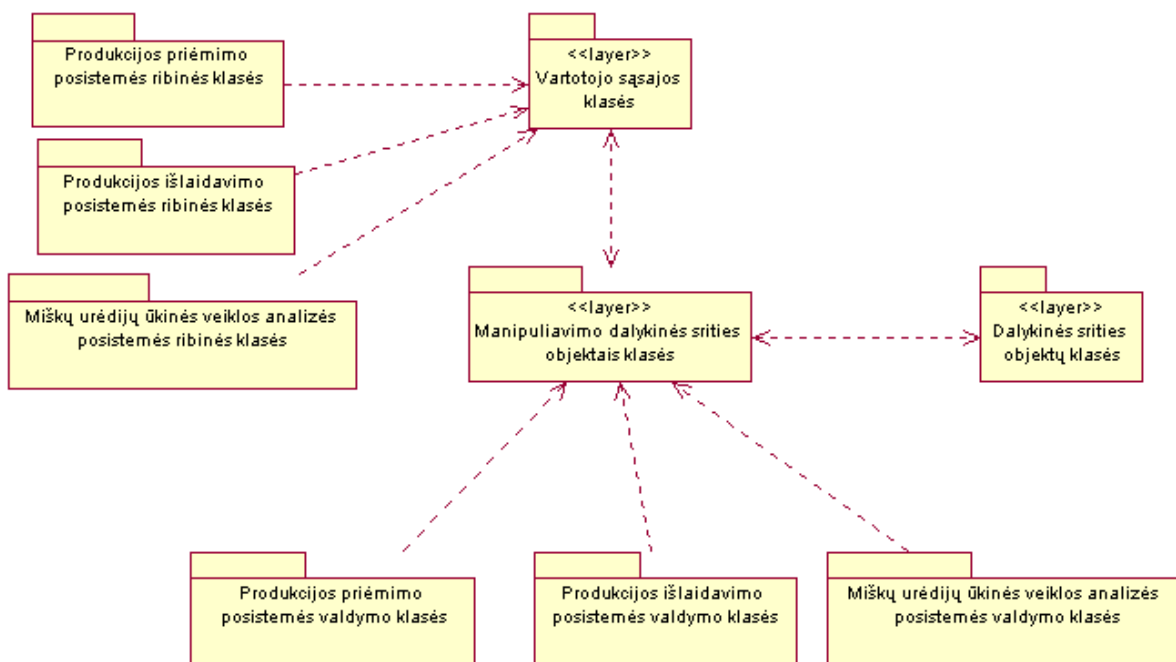
Klasifikacija: duomenų bazė.

Apibrėžimas: atlieka miškų urėdijos duomenų saugojimo funkciją.

Atsakomybės: šis komponentas kaupis informaciją apie miškų urėdijos pagamintą ir parduotą miškų ūkio produkciją.

5.3.2 Sistemos klasių diagramos

Sistemos klases būtų galima suskirstyti į tokius paketus: vartotojo sąsajos, dalykinės srities objektų ir manipuliavimo su dalykinės srities objektais. Vartotojo sąsajos paketui priklauso ribinės klasės, kurių pagalba vartotojas sąveikauja su sistema. Dalykinės srities objektų paketui priklauso dalykinės srities objektus aprašančios klasės, atitinkančios DB lenteles. Manipuliavimo su dalykinės srities objektais paketo klasės atlieka veiksmus su dalykinės srities objektais ir yra tarpininkais tarp vartotojo sąsajos ir dalykinės srities objektų klasių. Klases apjungiančių paketų diagrama pavaizduota pav. 14.



14 pav. Sistemos klasių suskirstymas į paketus

Vartotojo sąsajos klasių paketui atskiros diagramos braižomos nebuvo, nes vartotojo sąsaja nėra loginis sistemos architektūros vaizdas – ji priklauso fiziniam sistemos realizacijos

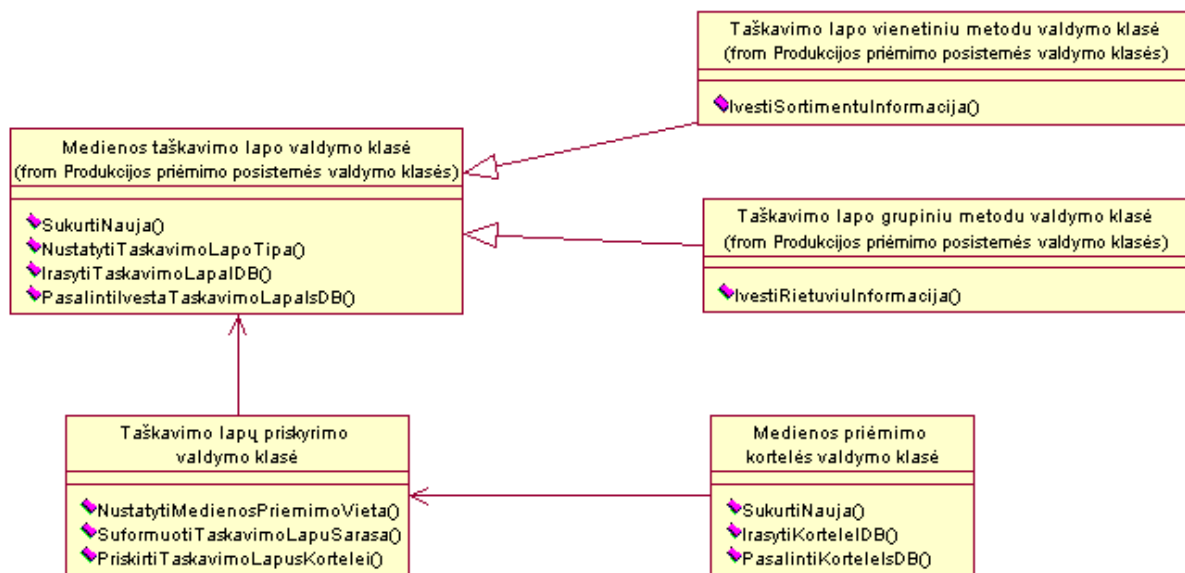
lygmeniui. Tačiau buvo pabandyta išskirti kai kuriuos galimus pagrindinių posistemų vartotojo sąsajos elementus.

Produkcijos priėmimo posistemės vartotojo sąsają sudarys meniu; apvaliosios medienos taškavimo lapų ir apvaliosios medienos priėmimo kortelių sąrašų ekrano formos; apvaliosios medienos taškavimo lapo antgalvės, kūnelių vienetiniu bei grupiniu metodu įvedimo/koregavimo ekrano formos; apvaliosios medienos priėmimo kortelės antgalvės bei kūnelių įvedimo/koregavimo ekrano formos.

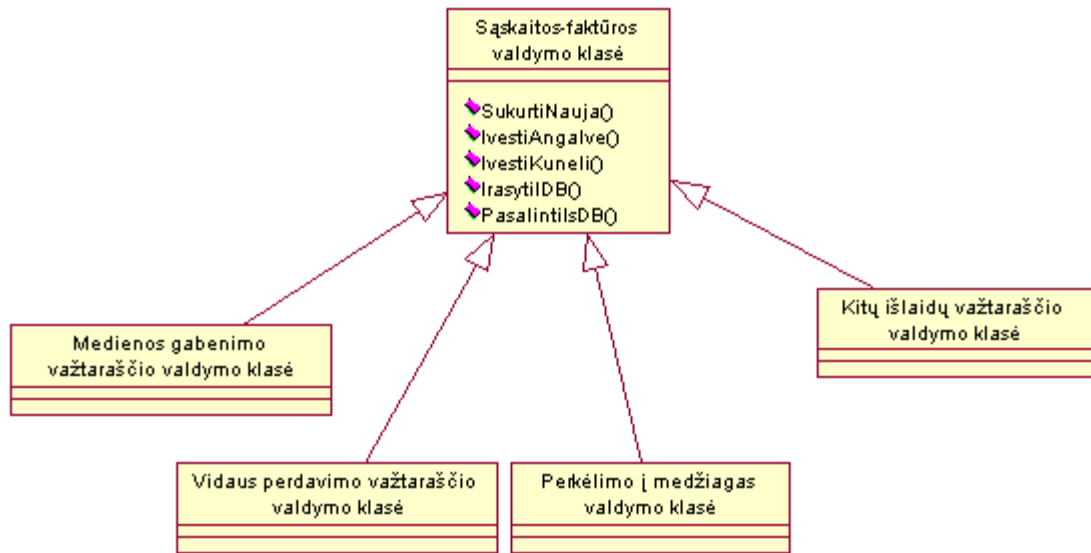
Produkcijos išlaidavimo posistemės vartotojo sąsają sudarys meniu; sąskaitos-faktūros, medienos gabenimo važtaraščio, vidaus perdavimo važtaraščio, perkėlimo į medžiagas važtaraščio ir kitų išlaidų važtaraščio antgalvės bei kūnelių įvedimo ekrano formos.

Miškų urėdijų ūkinės veiklos analizės posistemės vartotojo sąsają sudarys meniu; laiko periodo nustatymo ekrano forma; pagrindinis langas, sudalintas į dvi dalis: nustatymų ir rezultatų. Nustatymų lange medžio struktūros pavidalu bus pateiktas girininkijų, jose esančių sandėlių ir produkcijos sąrašas, o rezultatų lange bus rodoma, kiek kokios produkcijos buvo pagaminta, parduota ir yra likę.

Pav. 15 - 16 pateiktos valdymo klasių diagramos.

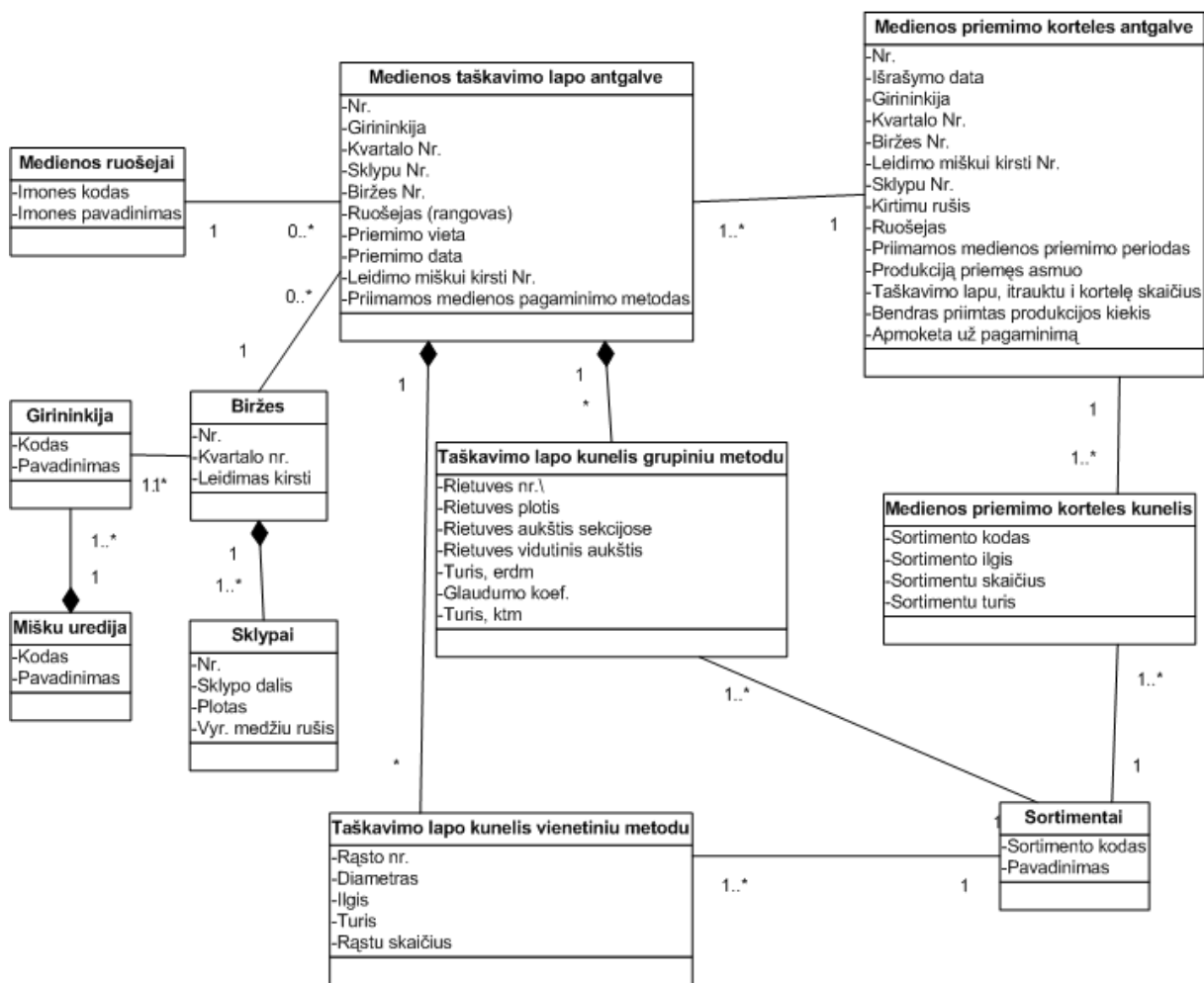


15 pav. Produkcijos priėmimo posistemės valdymo klasių diagrama

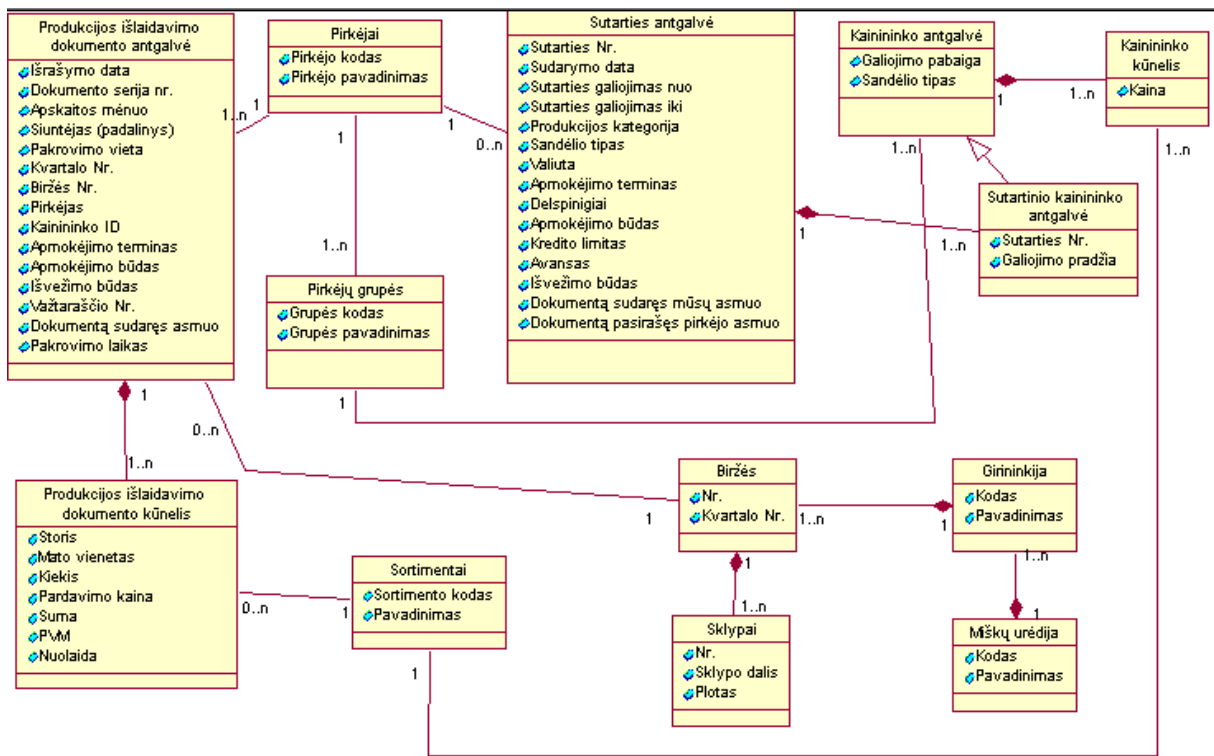


16 pav. Produkcijos išlaidavimo posistemės valdymo klasių diagrama

Pav. 17 – 18 išskirtos klasės priklauso dalykinės srities objektų klasių paketui. Produkcijos priėmimo ir produkcijos išlaidavimo posistemių dalykinės srities objektų klasėmis naudojasi miškų urėdijų ūkinės veiklos analizės posistemė, todėl jai atskira diagrama nepateikiama. Dalykinės srities objektų klasių diagramų pagrindu bus projektuojama sistemos duomenų bazė.



17 pav. Produkcijos priėmimo posistemės dalykinės srities klasių diagrama

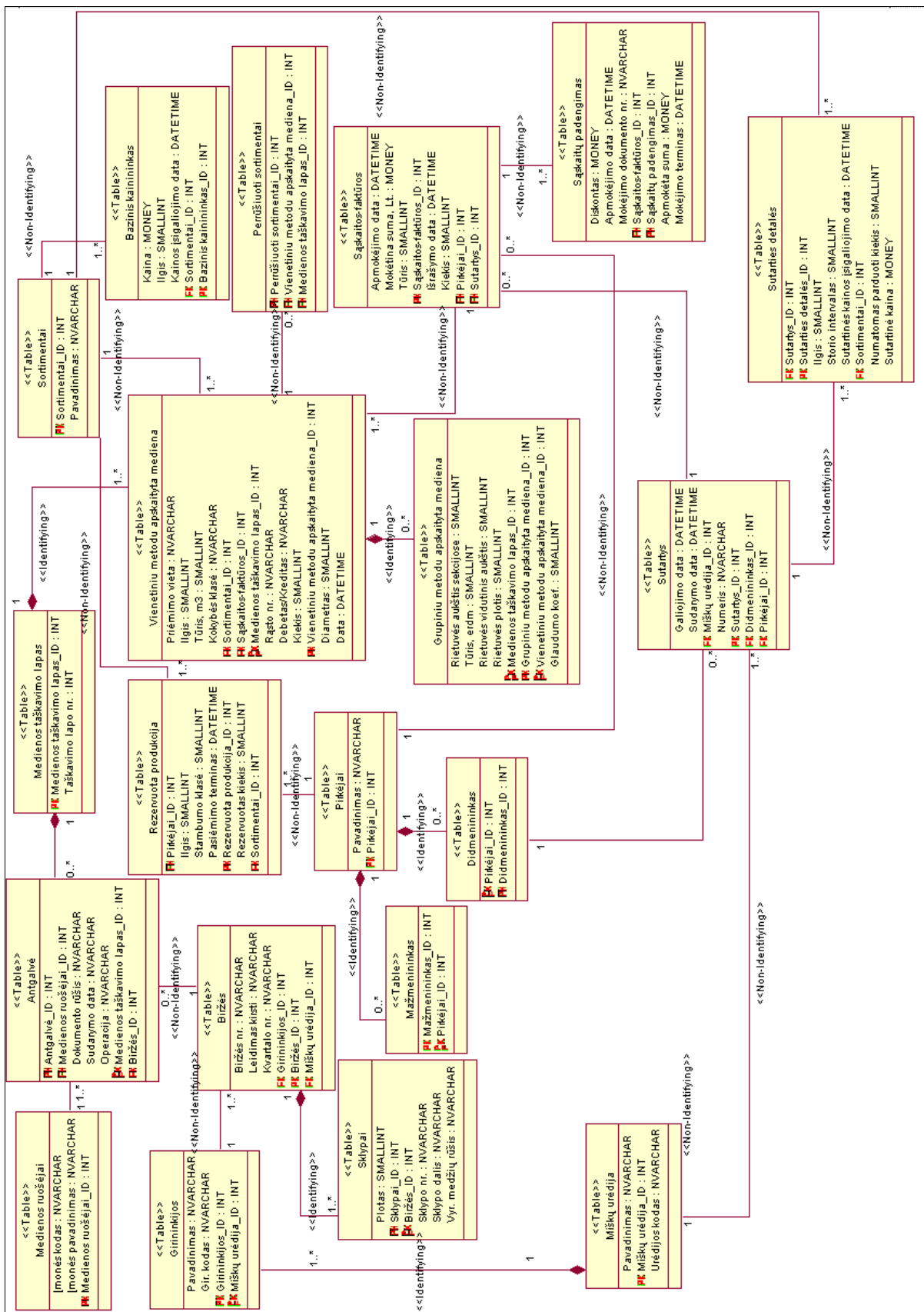


18 pav. Produkcijos išlaidavimo posistemės dalykinės srities klasių diagrama

Sistemos klasių detalus aprašymas pateikiamas šio dokumento prieduose A ir B.

5.3.3 Duomenų bazės schema

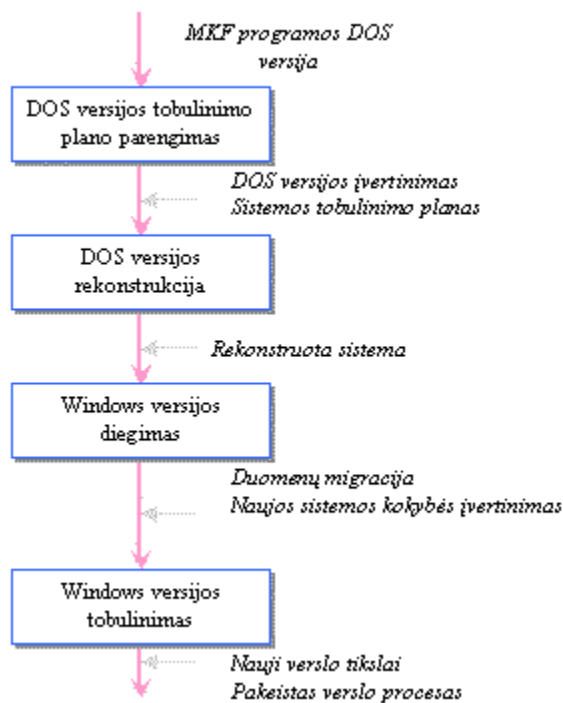
Pav. 20 pateikta DB schema, sugeneruota iš dalykinės srities klasių diagramų. Agregacijos, kompozicijos ir paveldėjimo ryšiai yra tipo <<Identifying>>, asociacijos ryšiai – <<Non-Identifying>>. Paveldėjimo ryšiai buvo transformuoti į kompozicijos ryšius. Kiekviena klasė, tapusi DB lentele, įgavo pirminį raktą (PK), o kiekviena lentelė, esanti kardinalumo "1..*" arba "0..*" pusėje, paveldi tėvinės lentelės pirminį raktą (FK).



19 pav. Sistemos naudojama duomenų bazės schema

6 PAVELDĖTOSIOS IR REKONSTRUOTOS SISTEMŲ TYRIMAS

MKF programos DOS versijos modernizacijos procesas buvo sudarytas iš 4 etapų: tobulinimo plano parengimo, rekonstrukcijos, sistemos Windows versijos diegimo ir jos tolimesnio tobulinimo¹ (žr. pav. 20).



20 pav. Sistemos modernizacijos procesas

Šiame skyriuje aprašyta atlikto darbo tiriamoji dalis, apimanti sistemos DOS versijos tobulinimo plano parengimo, Windows versijos diegimo ir tobulinimo etapus.

6.1 Paveldėtosios sistemos kokybės tyrimas

6.1.1 Kokybės įvertinimo metodo aprašymas

Prieš pradėdant MKF programos rekonstrukciją, buvo atliktas jos įvertinimas pagal dviejų matavimų paveldėtosios sistemos vertės nustatymo modelį (žr. skyrių 3.1.4.1), papildant jį *Renaissance* metodo nurodymais (Renaissance Consortium, 1998 birželis).

Visą naudotą MKF programos įvertinimo metodą galima suskirstyti į tokius etapus:

¹ MKF programos DOS versijos rekonstrukcija buvo atliekama remiantis *Renaissance* metodu (Renaissance Consortium, 1998 spalio)

1. Sudaromos anketos, skirtos įvertinti sistemos vertę verslui bei jos kokybę.
2. Pagal anketose esančius klausimus atliekama projekto užsakovo, galutinių sistemos vartotojų, Lietuvos miškų urėdijose dirbančių vadybininkų bei jų vadovų – urėdų – apklausa.
3. Gauti apklausos rezultatai apibendrinami, kiekvieną anketos klausimą įvertinant keturių balų sistemoje, kur:
 - 1 – klausimu vertinama sistemos savybė yra labai prastoje būklėje;
 - 2 – klausimu vertinama sistemos savybė yra prastoje, bet dar nekritinėje būklėje;
 - 3 – klausimu vertinama sistemos savybė gali būti vertinama vidutiniškai;
 - 4 – klausimu vertinama sistemos savybė vertinama puikiai.
4. Vertę verslui ir kokybę nustatančių klausimų įvertinimai susumuojami atskirai, o gauti rezultatai palyginami su rodikliu $(n*4)/2$, kur n – anketos klausimų skaičius kiekvienoje grupėje. Priklausomai nuo palyginimo rezultatų, parenkamas tolimesnis sistemos tobulinimo kelias.

Jeigu vertė verslui ir sistemos kokybė neperžengia už rodiklio $(n*4)/2$ ribų, sistemą galima palaikyti toliau, nieko joje nekeičiant. Jeigu vertė verslui yra didesnė už rodiklį $(n*4)/2$, tačiau kokybė peržengia jo ribas, sistemą reikia rekonstruoti. Jeigu vertė verslui yra mažesnė už rodiklį $(n*4)/2$, o kokybė didesnė, sistemą galima arba toliau palaikyti, arba jos atsisakyti. Jeigu vertė verslui ir sistemos kokybė mažesnės už rodiklį $(n*4)/2$, sistemą reikia pakeisti kita, nes jos rekonstrukcija per brangiai kainuos.

Sekančiame skyriuje pateikiami aprašyto metodo pritaikymas MKF programos DOS versijai ir jo metu gauti rezultatai.

6.1.2 Kokybės įvertinimo rezultatai

Pirmiausia buvo išsiaiškinta MKF programos įtaka ir vertė miškų urėdijų veiklai. Žemiau pateikiami atsakymai į klausimus, leidžiančius nustatyti nagrinėjamos programos vertę verslui, o lentelėje 7 – šių atsakymų skaitinis įvertinimas.

V1. Ar sistema efektyviai leidžia vykdyti verslo procesus?

Taip, nes joje yra realizuoti visi pagrindiniai miškų urėdijose atliekami procesai: biržių informacijos tvarkymas, medienos priėmimas, medienos išlaidavimas ir kt.

V2. Kiek procentų visos sistemos funkcijų naudoja vartotojai?

Tik 25% sistemos vartotojų naudojami visomis jos funkcijomis; likę dirba tik su 10% programos galimybių.

V3. Ar sistemos naudojimas neapriboja urėdijų klientų veiksmų?

Klientai tiesiogiai nejaučia neigiamos sistemos įtakos jų verslo santykiams su miškų urėdija, tačiau realizavus sistemoje galimybę duomenis apie priimtą ir išlaiduotą medieną įvedinėti tiesiog miške, klientų aptarnavimo laikas paspartėtų, ir jie taip pat gautų naudos iš urėdijose naudojamos PĮ.

V4. Ar sistema efektyviai prisideda prie girininkijų veiklos sėkmės?

Taip, nes ji panaikina rankinį skaičiavimą biržių įvertinimo, medienos priėmimo bei medienos išlaidavimo procesų metu, kuris dažnai būna ne tik lėtas, bet ir netikslus.

V5. Ar sistemos apdorojami duomenys yra kritiniai girininkijų veiklai?

Taip, nes šie duomenys naudojami planuojant produkcijos gamybą, aptarnaujant klientus bei analizuojant gaunamus veiklos rezultatus.

V6. Ar sunku surasti žmonių, sugebančių dirbti su sistema?

Apmokyti vartotojus dirbti su turima sistema nėra sunku, tačiau nauji vartotojai jau nelinkę įsisavinti darbo su programomis DOS aplinkoje. Urėdijos turi problemų ieškant sistemą prižiūrinčių specialistų, kadangi dauguma jų neturi žinių apie DOS platformai skirtų programų palaikymą.

Vertinant MKF programos kokybę, visi klausimai buvo suskirstyti į tris grupes:

- 1) verslo procesų kokybės nustatymo;
- 2) veikimo aplinkos kokybės nustatymo;
- 3) visos sistemos kokybės nustatymo.

Verslo procesų kokybės įvertinimas.

P1. Ar egzistuoja apibrėžtas verslo procesų modelis? Jei taip, ar yra atitinkamos procedūros, padedančios patikrinti naudojimąsi šiuo modeliu?

Miškų urėdijose vykdomi verslo procesai reglamentuojami LR įstatymų, kurių vykdymą kontroliuoja Generalinė miškų urėdija. Tačiau su vykdomų verslo procesų modeliu gerai susipažinę tik patys vartotojai, negalintys pateikti formalaus proceso modelio aprašo, o tai labai apsunkina naudojamos sistemos tobulinimą.

P2. Ar yra būtini sąryšiai su kitais verslo procesais?

Miškų urėdijų vykdoma produkcijos gamybos-pardavimo apskaita tiesiogiai susijusi su buhalterinės apskaitos procesu, kuris naudojasi joje surenkamais duomenimis.

P3. Ar procesas efektyviai palaikomas esančios sistemos?

Sistema realizuoja visas funkcijas, reikalingas verslo procesui. Ji pateikia visą jam reikalingą informaciją bei leidžia išvengti tų pačių duomenų įvedimo kelis kartus skirtingose vietose. Tačiau sunku užtikrinti šios sistemos sąsają su kitomis, ne mažiau verslo procesui svarbiomis sistemomis, be to, tampa problematiška realizuoti naujas arba patobulinti jau esamas funkcijas, norint pagerinti vykdomą verslo procesą. Duomenų mainai tarp atskirų sistemos komponentų yra neautomatizuoti, todėl negalima garantuoti pilno duomenų patikimumo.

Veikimo aplinkos kokybės nustatymas.

E1. Ar palaikančios PĮ tiekėjas dar vykdo savo veiklą ir atlieka savo produktų palaikymą?

Naudojama MKF versija yra sukurta duomenų bazių valdymo sistemos (DBVS) FoxPro, veikiančios DOS aplinkoje, priemonių pagalba. DOS platformai skirtos FoxPro priemonės jau nebetobulinamos ir nebepalaikomos savo kūrėjų.

E2. Ar yra problemų egzistuojančiai sistemai sąveikauti su kitomis sistemomis?

Naudojama sistema nėra susieta su urėdijose naudojamomis buhalterinės apskaitos programomis, taip pat dėl sistemoje naudojamų pasenusių technologijų būtų sudėtinga užtikrinti jos sąsajas su duomenų mainus tarp girininkijų ir urėdijos, bei urėdijos ir Generalinės miškų urėdijos atliekančiu modulių, mobilia girininko darbo vieta ir kt.

E3. Koks yra egzistuojančios sistemos išeities kodo suprantamumas?

Egzistuojančios sistemos išeities kodo suprantamumo lygis yra pakankamai aukštas dėl naudojamų komentarų, prasmingų kintamųjų vardų, dauguma kodo gerai struktūrizuota.

E4. Kokio lygio turima sistemos dokumentacija?

Vienintelė turima sistemos dokumentacija yra jos vartotojo vadovas, išsamiai aprašantis bazines funkcijas, tačiau sistemą iš programavimo pusės aiškinančių dokumentų, tokių kaip reikalavimų, architektūros specifikacija, testavimo planai, nėra.

E5. Kokios būklės yra sistemos duomenų modelis?

Naudojama sistema dirba „laisvųjų“ DB lentelių pagrindu, kurios nėra sujungtos į vieningą DB. Visos DB funkcijos užtikrinamos per duomenų žodyną bei programos autoriaus sukurtas funkcijas. Kai kurias savybes, kurias leidžia palaikyti šiuolaikinės DBVS, realizuoti tokių metodų pagalba yra sunku ir dažnai per brangu.

E6. Kokia programavimo kalba naudojama sistemai palaikyti?

Kaip jau minėta anksčiau, egzistuojanti sistema yra sukurta FoxPro, veikiančio DOS aplinkoje, priemonėmis, kurios naudojamos ir jos palaikymo procese. Šios priemonės yra jau

nebepalaikomos savo kūrėjų, todėl tampa vis sunkiau turimą sistemą pritaikyti prie urėdijose naudojamų operacinių sistemų, be to, turimų priemonių pagalba problematiška realizuoti šiuolaikiškai atrodančias ataskaitas ar vartotojo sąsają.

E7. Ar egzistuoja sistemos konfigūracijos valdymas?

Egzistuoja dviejų lygių sistemos versijų valdymo kontrolė: naujų modulių licencijų ir vartotojų licencijų. Modulių licencijų atveju vartotojas įsigyja teisę naudoti konkretų modulį ir ši teisė išlieka sekančiose sistemos versijose. Vartotojų licencijų atveju, specialiai versijų valdymo kontrolei skirtame faile yra saugoma informacija apie tai, su kokia versija gali dirbti koks vartotojas.

Tačiau užsakovo įmonėje nebuvo naudojama kitų sistemai svarbių elementų (reikalavimų, projektavimo, testavimo dokumentų) konfigūracijos valdymas.

E8. Ar yra sistemos testavimo dokumentacija?

Sistemos testavimo dokumentacijos nėra. Testavimui buvo naudojami iš vartotojų gauti duomenys, tačiau informacija apie įvykdytus testavimo atvejus bei jiems naudotus duomenis ir gautus rezultatus saugoma nebuvo.

E9. Ar yra žmonių, galinčių palaikyti ir suprasti turimą sistemą?

Prie MKF programos dirbo tik užsakovo įmonė, kurios savininkas ir yra sistemos kūrėjas, todėl tai palengvina jos palaikymą. Vis dėlto susirasti naujų darbuotojų, kurie galėtų dirbti DOS FoxPro priemonėmis, yra sunku bei brangu.

Visos sistemos kokybės įvertinimas.

S1. Koks yra prašymų pakoreguoti sistemą skaičius?

Per metus iš aktyviausių sistemos vartotojų buvo gaunama apie 20 pasiūlymų tobulinti sistemą bei nuolatos tekdavo taisyti menkas klaidas, pastebėtas sistemoje. Todėl dėl didelio keitimų skaičiaus kai kurios sistemos dalys išgadino sistemos struktūrą, be to, pakeitimai yra nedokumentuoti ir dalis jų daryti skubiai, siekiant patenkinti vartotojų poreikius, kas taip pat neigiamai paveikė naudojamą sistemą.

S2. Kiek sistemoje yra skirtingų vartotojo sąsajų?

Naudojamoje sistemoje yra virš 30 skirtingas funkcijas atliekančių ekrano formų, o tai apsunkina sistemos palaikymą, nes reikia užtikrinti kiekvienos formos teisingą veikimą.

S3. Koks yra sistemos apdorojamų duomenų kiekis?

Sistemoje yra 182 DB lentelės, skirtos duomenims saugoti, bei 79 klasifikatoriai. Vidutiniškai vienoje duomenų saugojimo lentelėje yra apie 4000 įrašų, susidarančių per vienerius

metus, todėl sistemos apdorojami duomenų kiekiai yra ištiesų dideli ir jai reikia tikrai gero ir patikimo duomenų modelio ir DBVS.

Apibendrinus visus pateiktus atsakymus, jie buvo įvertinti keturių balų sistemoje. Įvertinimo rezultatai pateikti lentelėje 7.

7 lentelė. Sistemos DOS versijos vertinimo rezultatai

Klausimo grupė	Vertinama savybė	Klausimo Nr.	Skaitinė vertė	
Vertė verslui	Svarba galutiniams vartotojams	V1.	4	
		V2.	2	
	Svarba urėdijų klientams	tik su 10% programos galimybių.	V3.	3
		Svarba girininkijų veiklai	V4.	4
	V5.		4	
	Svarba urėdijų personalo vadybininkams	V6.	2	
		ΣV	19	
	$(n*4)/2$	12		
Kokybės lygis	Kokybė verslo atžvilgiu	P1.	3	
		P2.	2	
		P3.	2	
	Tiekėjas	E1.	1	
	Suderinamumas	E2.	1	
	Išeities kodas	E3.	3	
	Dokumentacija	E4.	2	
	Duomenys	E5.	2	
	Programavimo kalba	E6.	1	
	Konfigūracijos valdymas	E7.	2	

7 lentelė. Sistemos DOS versijos vertinimo rezultatai

Klausimo grupė	Vertinama savybė	Klausimo Nr.	Skaitinė vertė
	Testavimo dokumentacija	E8.	1
	Personalo kvalifikacija	E9.	2
	Prašymų pakeisti sistemą skaičius	S1.	1
	Vartotojo sąsajų skaičius	S2.	2
	Duomenų apimtys	S3.	1
		ΣK	26
		$(n*4)/2$	30

Taigi pagal lentelėje 7 pateiktus rezultatus matyti, kad MKF programos vertė verslui yra didelė ($\Sigma V > (n*4)/2$), o jos kokybė maža ($\Sigma K < (n*4)/2$). Vadinasi, jos atsisakyti arba sukurti pradedant nuo pat pradžių būtų per daug rizikinga. Mažiausia rizika, būtų šią PĮ rekonstruoti.

6.1.3 Paveldėtosios sistemos rekonstrukcijos galimybių analizė

Remiantis atliktu sistemos DOS versijos įvertinimu, nuspręsta modernizuoti MKF programą rekonstrukcijos pagalba. Šis būdas pasirinktas dėl tokių priežasčių:

1. MKF programos palaikymas tapo sudėtingas ir brangus dėl naudojamų priemonių ribotumo.
2. Naudojamos programos vertė verslui yra labai didelė, todėl būtų per daug rizikinga visą sistemą kurti nuo pat pradžių, nes tuomet padidėtų tikimybė, kad naujasis variantas neturės visų reikalingų senosios sistemos funkcijų arba šias funkcijas atliks klaidingai.
3. Rekonstrukcijos procese dalyvaus MKF programos autorius bei žmonės, gerai susipažinę su šia programa, todėl nereikės samdyti naujų žmonių, tokiu būdu sumažinant programos atnaujinimo išlaidas.
4. Rekonstrukcija galima atlikti palaipsniui, iš pradžių modernizuojant svarbiausius MKF programos modulius.

Pasirinktas evoliucinis rekonstrukcijos įgyvendinimo metodas (žr. skyrių 4.5), MKF programą suskirstant į atskirus modulius pagal jų atliekamas funkcijas, atskirai juos rekonstruojant ir palaipsniškai apjungiant juos į naują sistemą. Rekonstrukcijai pasirinkti

svarbiausi MKF programos moduliai: biržių informacijos, medienos priėmimo ir medienos išlaidavimo.

Analizuojant MKF programos vertės tyrimo rezultatus paaiškėjo, kad jos kokybė nukenčia dėl naudojamų atgyvenusių kūrimo priemonių, duomenų modelio ribotumo bei dokumentacijos nebuvimo. Todėl nuspręsta pasirinkti šiuos rekonstrukcijos metodus: kodo retransliavimą, atvirkštinę rekonstrukciją ir duomenų rekonstrukciją. Kiekvieno metodo pagalba siekiama pagerinti ir tam tikras sistemos savybes, nukenčiančias dėl anksčiau minėtų veiksnių.

Rekonstrukcijos metodų sąsaja su konkrečiomis paveldėtos sistemos charakteristikomis pateikta lentelėje 8. Šios lentelės pagrindu bus atliekamas ir atnaujintos sistemos kokybės įvertinimas, aprašytas sekančiame skyriuje.

8 lentelė. Rekonstrukcijos metodų sąsaja su sistemos charakteristikomis

Charakteristika	Naudojami rekonstrukcijos metodai		
	Atvirkštinė rekonstrukcija	Kodo retransliacija	Duomenų rekonstrukcija
Patikimumas		*	*
Našumas		*	*
Panaudojamumas		*	
Palaikomumas	*	*	*
Pakartotinis panaudojimas	*		*
Išplečiamumas	*	*	*

6.2 Naujos sistemos versijos kokybės tyrimas

6.2.1 Kokybės įvertinimo metodo aprašymas

Naujos sistemos kokybės įvertinimas buvo atliktas dviem etapais. Pirmame etape įvertinta, kaip po rekonstrukcijos pasikeitė sistemos charakteristikos, įvardintos lentelėje 8. Antrame etape palyginta, kaip pasikeitė sistemos kokybės rodikliai lyginant su senąja sistema. Pastarojo tyrimo metu buvo pasinaudota senajai sistemai įvertinti skirtais klausimais (žr. skyrių 6.1.2). Viso sistemos kokybės tyrimo metu dalyvavo projekto užsakovas, o analizuojant sistemą iš vartotojo pozicijų buvo apklausiami sistemos galutiniai vartotojai.

6.2.2 Kokybės įvertinimo rezultatai

Pirmasis kokybės vertinimo etapas

Šio etapo metu buvo įvertinta, kaip panaudojus atvirkštinę rekonstrukciją, kodo retransliavimą ir duomenų rekonstrukciją, buvo pagerintos sistemos charakteristikos, pateiktos lentelėje 8.

Patikimumo charakteristika pagerėjo 20%. Senojoje sistemoje DB lentelės nebuvo apjungtos į vieningą DB, o saugomos kaip „laisvosios“ lentelės. Dėl to programoje pasitaikydavo duomenų „dauginimasis“ atliekant skaičiavimus, formuojant ataskaitas. Šio tipo klaidos ypač pasireikšdavo esant didesniam duomenų kiekiui. Be to, senojoje sistemoje duomenų šalinimo ar įterpimo metu nebūdavo pilnai užtikrinamas DB integralumas, dėl ko taip pat atsirasdavo programos darbo sutrikimai. Naujoje sistemoje DB integralumą užtikrina Visual FoxPro DBVS, o unikalūs identifikatoriai duomenų lentelėse padeda išvengti jų „dauginimosi“ programos darbo metu.

Našumo charakteristika pagerėjo apie 5%. Šis pasikeitimas pastebimas tik programos paleidimo atžvilgiu, nes Windows operacinė sistema lėčiau startuodavo DOS aplinkai skirtas programas. Kadangi pats programos kodas nebuvo optimizuojamas, siekiant pagerinti sistemos našumą, o jos našumas ir nebuvo labai žemas, rekonstrukcija šios charakteristikos ir negalėjo labai įtakoti.

Panaudojamumo charakteristikos pagerėjimas yra net 45%. Naujoji sistemos versija tapo daug patrauklesnė vartotojui dėl šių priežasčių:

1. Pereita nuo tekstinėmis formomis pagrįstos vartotojo sąsajos prie grafinės, kuri šiuolaikiniams PĮ naudotojams yra daug įprastesnė už pirmąjį variantą.
2. Supaprastėjo programos instaliavimo procesas: norint dirbti su senąja sistema, reikėdavo atlikti tam tikrus operacinės sistemos nustatymus, o tai paprastiems vartotojams yra per sudėtinga.
3. Rekonstrukcijos metu buvo atlikta senosios programos versijos meniu reorganizacija: buvo pakeistas jo punktų išdėstymas, orientuotas į atliekamas funkcijas. Šis išdėstymas buvo jau išsigadinęs, atliekant nemažą pakeitimų skaičių ir plečiant programos funkcionalumą. Dabartinis meniu variantas leidžia lengviau orientuotis netgi naujokui, neskaičiusiam vartotojo instrukcijų.

Palaikomumas turėtų išaugti net iki 70%, nes buvo sukurta išsami sistemos dokumentacija, aprašytas programos testavimo procesas ir jo metu gauti rezultatai, paruoštas detalus vartotojo vadovas bei pereita prie šiuolaikiškos programavimo kalbos. Parengta DB schema leis paprasčiau

realizuoti duomenų įterpimo ir šalinimo operacijas, o programuotojams nereikės rūpintis kiekvienos lentelės atidarymu prieš panaudojimą, nes jos bus atidaromos kartu su DB programos darbo pradžioje.

Pakartotinio panaudojimo tikimybė išaugo apie 15%. Didžiausia tikimybė, kad pakartotinai bus panaudojama sistemos DB, nes ji tiks projektuojant mobilią girininko darbo vietą, kuriant papildomus sistemos modulius. Pakartotinai gali būti panaudoti ir surinkti reikalavimai, architektūros specifikacija, testavimo planai, ateityje prireikus iš esmės keisti sistemos veikimo aplinką.

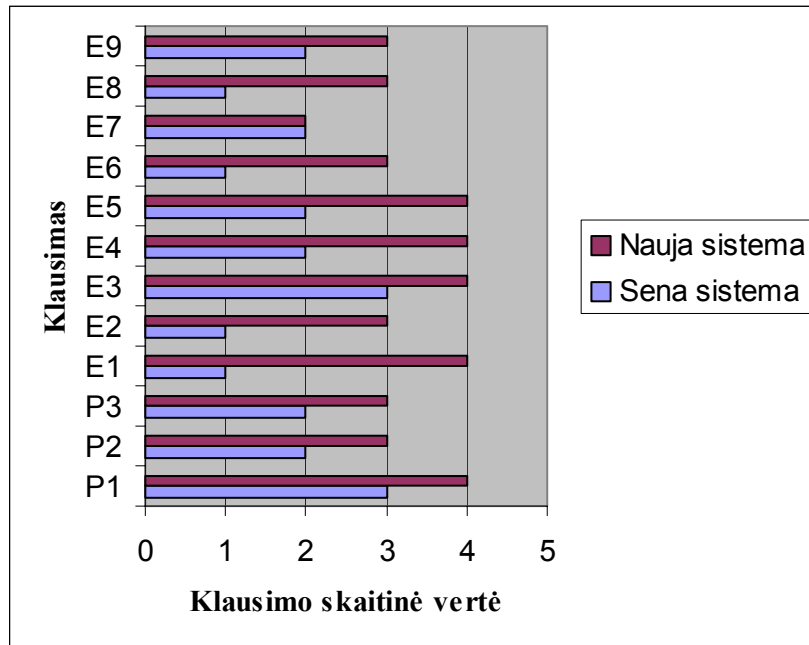
Išplečiamumas pagerėjo net 65%. Turint išsamią dokumentaciją ir perėjus prie šiuolaikinių programavimo technologijų, sistemą galima lengvai papildyti naujais moduliais, paprasčiau užtikrinti sąsają su kitomis sistemomis bei atlikti perėjimą prie ateityje planuojamos bendros Lietuvos miškų urėdijų DB.

Antrasis kokybės vertinimo etapas

Jo metu buvo atliktas sistemos kokybės įvertinimas pagal anketas, naudotas senos jos versijos vertės nustatymui (žr. skyrių 6.1.2). Kokybės analizei pasirinkti tik sistemos techninę kokybę nustatinėjantys klausimai, nes jos vertė verslui ir prieš rekonstrukciją buvo didelė, tad naujos versijos sukūrimas padėjo tik ją išlaikyti, o ateityje leis ir padidinti.

Vertinami buvo tik sistemos verslo proceso ir veikimo aplinkos kokybę atspindintys klausimai, nes sunku pradinėje naudojimo stadijoje atsakyti į klausimą, koks bus prašymų patobulinti skaičius, o sistemos langų skaičius ir duomenų apimtys naujoje sistemoje išliko tokie patys kaip ir jos pirmtakėje.

Gautų atsakymų įvertinimai buvo palyginti su to paties eksperimento senosios sistemos vertei nustatyti rezultatais. Paaiškėjo, kad sistemos kokybė rekonstrukcijos pagalba išaugo 1,5 karto (žr. pav. 21), ypač pagerėjus sistemos dokumentacijos ir kūrimo priemonių kokybei – ji tapo lankstesnė ateityje planuojamiems tobulinimams.



21 pav. Senos ir naujos sistemos versijų kokybės palyginimas

6.2.3 Tolimesnis sistemos tobulinimas

Naujoji MKF versija bus pagrindas sistemai, kuri ateityje galėtų būti pritaikyta šių uždavinių sprendimui:

1. Miškų ūkio produkcijos gamybos-pardavimo duomenų įvedimui lauko sąlygomis.
2. Stambūs medienos pirkėjai, o ypač eksportuotojai, turėtų galimybę gauti elektroninę perkamos medienos specifikaciją, kurią po to galėtų panaudoti gautos medienos priėmimo, ginčų su tiekėju atvejais, o vėliau – ir medienos judėjimo savo įmonėje kontrolei.
3. Esant centralizuotam serveriui, pirkėjas, pateikęs savo prisijungimo duomenis, galėtų atsisiųsti informaciją apie nupirktą medieną, tiesiogiai nekontaktodamas su atskirų miškų urėdijų informacinėmis sistemomis.
4. Įteisinus elektroninius pirkimo-pardavimo dokumentus, būtų galima operatyviai pereiti prie jų.
5. Sistema leistų urėdijoms išsamius duomenis apie savo parduodamą medieną pateikti medienos biržai, kuomet tokia atsirastų.

7 REKONSTRUKCIJOS METODŲ EKSPERIMENTINIS ĮVERTINIMAS

Užbaigus MKF programos rekonstrukciją, buvo atliktas naudotų rekonstrukcijos metodų tyrimas. Jo metu buvo vertinama:

- 1) kiek laiko reikėjo skirti kiekvieno metodo pritaikymui ir įgyvendinimui;
- 2) kokią įtaką kiekvienas jų turėjo sistemos charakteristikų pagerinimui.

Atlikto eksperimento rezultatais galės pasinaudoti ir kiti, su panašaus tipo paveldėtosiomis sistemomis dirbantys specialistai.

Pagrindinės nagrinėtos sistemos charakteristikos yra:

- ji buvo parašyta DOS FoxPro programavimo priemonėmis;
- duomenys buvo saugojami „laisvosiose“ DB lentelėse, neapjungtose į vieningą DB;
- vienintelė prieinama sistemos dokumentacija buvo jos vartotojo vadovas.

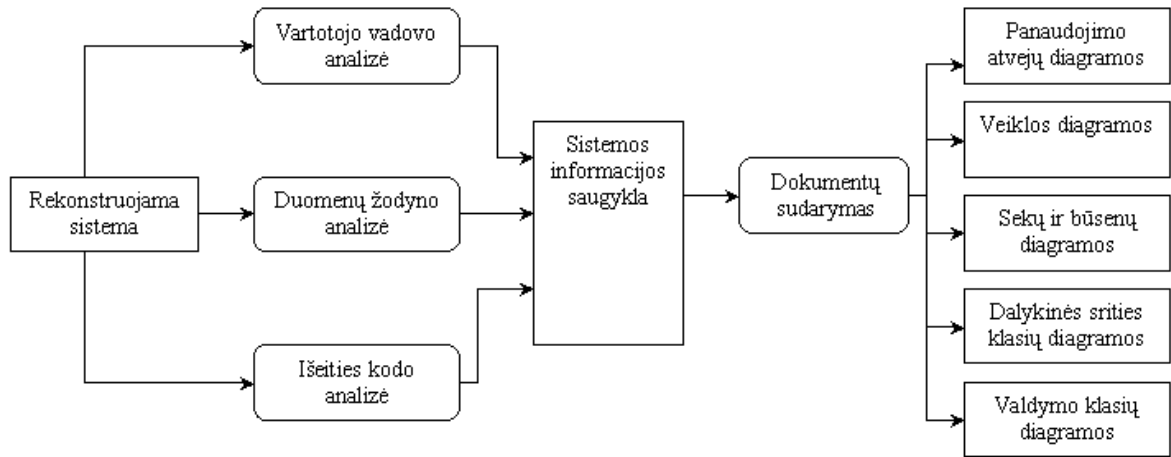
Šiame skyriuje nagrinėjama, kurie metodai buvo efektyviausi tokio tipo sistemos atveju. Pateikiami nurodymai, kaip kiekvieną iš naudotų metodų pritaikyti panašioms sistemoms.

7.1 Rekonstrukcijos metodų pritaikymas nagrinėjamai sistemai

7.1.1 Atvirkštinė rekonstrukcija

MKF programos atvirkštinės rekonstrukcijos tikslas buvo pagal turimą dokumentaciją ir programos kodą sudaryti jos struktūrą bei veikimo principus vaizduojančias UML (*Unified Modeling Language*) diagramas.

Atvirkštinės rekonstrukcijos procesą automatizuoti leidžiančių priemonių, kurios analizuotų FoxPro programavimo kalba parašytą išeities tekstą, rasti nepavyko, todėl ji buvo atliekama rankiniu būdu. Atliktą atvirkštinės rekonstrukcijos procesą galima būtų pavaizduoti taip, kaip parodyta pav. 22.



22 pav. Atvirkštinės rekonstrukcijos pritaikymas nagrinėjamai sistemai

Atlikus senosios sistemos vartotojo vadovo analizę, buvo sudarytos skyriuje 5.2.2 pateikiamos panaudojimo atvejų bei veiklos diagramos.

Remiantis turimu programos duomenų žodynu, buvo išskirtos dalykinės srities klasės (žr. skyrių 5.3.2), kurių diagramų pagalba Rational Rose paketas automatiškai sugeneravo naujosios sistemos DB schemą (žr. skyrių 5.3.3).

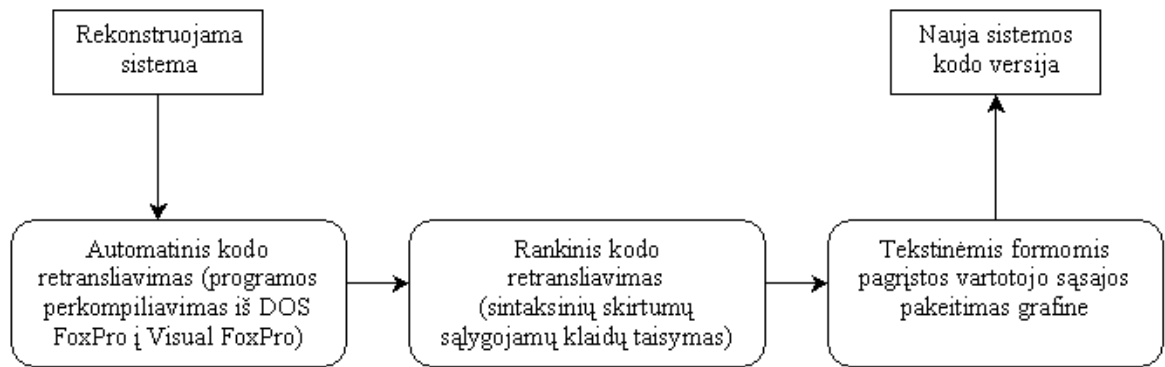
Analizuojant senosios programos išėities tekstus, buvo suformuotos naujosios sistemos valdymo klasių diagramos (žr. skyrių 5.3.2) bei sekų ir būsenų diagramos.

Atlikus nagrinėjamos sistemos atvirkštinę rekonstrukciją, galima padaryti išvadą, jog didelė dalis šio proceso programoms, sukurtoms tokiomis programavimo kalbomis kaip FoxPro, yra rankinis. Tad neturint jokios paveldėtosios sistemos dokumentacijos, duomenų modelio aprašo bei kokybiško išėities kodo, atvirkštinė rekonstrukcija gali užtrukti labai ilgai ir neduoti norimų rezultatų.

7.1.2 Kodo retransliacija

Kodo retransliacijos proceso tikslas buvo pervesti DOS FoxPro programos kodą į Visual FoxPro programavimo kalbą.

Kodo retransliacija buvo atliekama trimis etapais (žr. pav. 23). Pirmame etape DOS FoxPro programos kodas automatiškai buvo perkompiliuotas į Visual FoxPro programos kodą. Antrojo etapo metu programa buvo vykdoma ir rankomis taisomos dėl DOS FoxPro ir Visual FoxPro kalbų sintaksinių skirtumų atsiradusios klaidos. Trečiajame etape tekstinėmis formomis pagrįsta vartotojo sąsaja buvo pakeista į grafinę vartotojo sąsają.



23 pav. Kodo retransliacijos pritaikymas nagrinėjamai sistemai

Pirmasis ir antrasis etapas buvo atlikti nesunkiai ir su nedidelėmis laiko sąnaudomis, nes abiejų programavimo kalbų versijos yra labai panašios.

Problematiškiausias buvo trečiasis etapas, nes reikėjo išsiaiškinti, koku principu atlikti senąją vartotojo sąsają formuojančio kodo pritaikymą grafinės vartotojo sąsajos formavimui. Buvo išskirti trys perkeliamų vartotojo sąsajos elementų tipai: meniu, ekrano formos ir ataskaitos.

Menu elementų formavimas nebuvo perdarytas. Pastebėta, kad DOS versijos meniu formavimo variantas yra daug patogesnis negu Visual FoxPro, kadangi pastarajame naudojama vizualinė meniu kūrimo priemonė labiau apriboja programuotojo įtaką formuojamo meniu ir jo punktų valdymo atžvilgiu.

Ataskaitos buvo atnaujinamos dviem būdais. Pirmiausia teko pertransliuoti jose naudojamus simbolius (panaudota užsakovo įmonės sukurta programa). Po to ataskaitų elementų išdėstymas buvo koreguojamas rankiniu būdu Visual FoxPro ataskaitų kūrimo priemonės *Report Designer* pagalba.

Atliekant ekrano formų rekonstrukciją buvo išanalizuoti du formų variantai:

- 1) duomenys pateikiami lentelės *Browse* pavidalu;
- 2) duomenys atvaizduojami skirtinguose formos elementuose (*TextBox*, *ComboBox* ir kt.).

Pirmąjį ekrano formos variantą galima pailiustruoti tokiu pavyzdžiu. DOS FoxPro programavimo kalboje jis formuojamas šio kreipinio pagalba:

```

BROWSE FIELDS ;
GR :H='Gir' :R,;
KNR :H='B.Nr.' :R,;
KV :H='Kv' :R,;
NOEDIT NOMENU NODELETE LOCK 3;
FOR &br_salyga ;
TITLE wtitle_f
  
```

Šio kreipinio sąsaja su Visual FoxPro programavimo kalba pateikta lentelėje 9.

9 lentelė. *BROWSE* kreipinio sąsaja su Visual FoxPro programavimo kalba

DOS kodo elementas	DOS elemento reikšmė	Windows elemento atitikmuo
BROWSE FIELDS	Iškviečiama duomenų peržiūra lentelės pavidalu	Ekrano forma su <i>Grid</i> komponentu
GR : BNR : KV :	<lentelės atributo vardas>:	Kiekvieno <i>Grid</i> elemento <i>Column</i> savybei <i>ControlSource</i> nurodyti lentelės atributo vardą
H='Gir' H='B.Nr.' H='Kv'	H='<peržiūros lentelės stulpelio pavadinimas>'	Kiekvieno <i>Grid</i> elemento <i>Header</i> savybei <i>Caption</i> priskirti atitinkamą peržiūros lentelės stulpelio pavadinimą
FOR &br_salyga	FOR <lentelės filtravimo sąlyga>	Prieš iškviečiant ekrano formą, reikia nustatyti naudojamai lentelei filtrą: <i>SET FILTER TO &br_salyga</i>
TITLE wtitle_f	TITLE <ekrano formos pavadinimas>	Iškviešti ekrano formą, paduodant jai jos pavadinimo parametą: <i>DO form form_birsar with wtitle_f</i> Ekrano formos įvykyje <i>INIT EVENT</i> reikia jos savybei <i>Caption</i> priskirti paduoto parametro reikšmę.

Antrąjį ekrano formos variantą galima pailiustruoti tokiu pavyzdžiu. DOS FoxPro programavimo kalboje jis formuojamas šio kreipinio pagalba:

```

ACTIVATE WINDOW kirt_b
@ 1,2 SAY 'Išrašymo data:' GET m.IVDATA
@ 1,$+1 SAY 'Pratešta iki:' GET m.PRATES
VALID V_PRATEST()
@ 2,2 SAY 'Leidimo Nr.: ' GET m.kbnr ;
VALID v_kbnr()

```

Šio kreipinio sąsaja su Visual FoxPro programavimo kalba pateikta lentelėje 10.

10 lentelė. *ACTIVATE WINDOW* sąsaja su Visual FoxPro programavimo kalba

DOS kodo elementas	DOS elemento reikšmė	Windows elemento atitikmuo
ACTIVATE WINDOW kirt_b	Inicializuojama ekrano forma, pavadinimu kirt_b	Ekrano forma be <i>Grid</i> komponento
@ 1,2 @ 1,\$+1 @ 2,2	@ <ekrano formos elemento koordinatės>	Visual FoxPro atveju nenaudojama
SAY 'Išrašymo data:' SAY 'Pratešta iki:'	SAY '<ekrano formos elemento pavadinimas>'	Kiekvieno ekrano formos elemento savybei <i>Caption</i>

10 lentelė. *ACTIVATE WINDOW* sąsaja su Visual FoxPro programavimo kalba

DOS kodo elementas	DOS elemento reikšmė	Windows elemento atitikmuo
SAY 'Leidimo Nr.: '		priskirti atitinkamą pavadinimą
GET m.IVDATA GET m.PRATES GET m.kbnr	GET <ekrano formos naudojamų duomenų elementas>	Kiekvieno ekrano formos elemento <i>ControSource</i> savybei priskirti atitinkamą duomenų elementą
VALID V_PRATEST() VALID v_kbnr()	VALID <duomenų kontrolės veiksmus atliekančios procedūros vardas>	Kiekvieno ekrano formos elemento įvykyje <i>VALID EVENT</i> , nurodyti atitinkamą duomenų kontrolę atliekančią procedūrą

Naujos sistemos versijos grafinės vartotojo sąsajos langų pavyzdžiai pateikiami šio dokumento priede C.

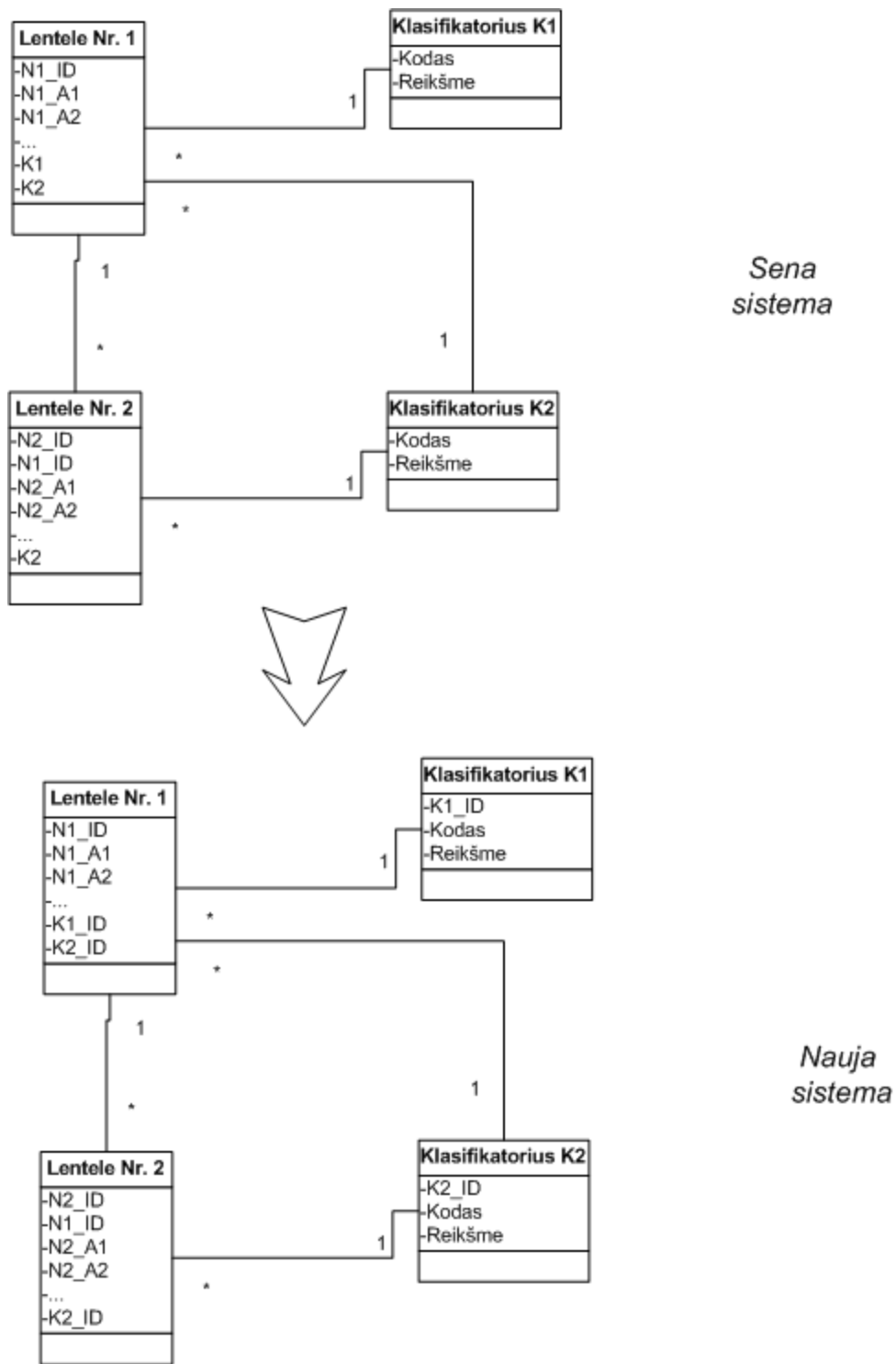
Apibedrinant atliktą kodo retransliaciją, galima teigti, kad esant tai pačiai programavimo kalbai, tik skirtingoms jos versijoms, ši rekonstrukcijos etapą galima atlikti beveik automatiškai. Sunkiausia yra perkelti vartotojo sąsajos komponentus: šiuo atveju reikia atlikti nemažai rankinio darbo bei kruopštų naujųjų komponentų testavimą.

7.1.3 Duomenų migracija

Sistemos Windows versijos diegimo metu didžiausia problema buvo senos sistemos duomenų perkėlimas į naująją.

Senosios sistemos DB lentelės galima suskirstyti į du tipus: pagrindinio naudojimo ir sąrašų (klasifikatorių). Pagrindinio naudojimo lentelės tarpusavyje siejamos unikalių identifikatorių pagalba, o ryšiai tarp jų ir sąrašų lentelių užtikrinami sąrašų lentelių lauko „Kodas“ pagalba (žr. pav. 24).

Naujai suprojektuotoje DB buvo išlaikyti senieji lentelių ir jų atributų vardai tam, kad palengvinti būsimąją duomenų migraciją. Skirtumas tarp senosios sistemos naudojamo duomenų modelio ir naujosios DB yra tas, kad pastarojoje ryšiai tarp visų lentelių yra užtikrinami tik per unikalius identifikatorius (žr. pav. 24). Todėl pagrindinio naudojimo lentelių, nesusijusių su sąrašais, duomenų perkėlimas nesukėlė didelių sunkumų – duomenys paprasčiausiai buvo įrašyti iš senųjų lentelių į naujuosius jų atitikmenis. Sąrašų lentelių duomenys buvo perkelti tokiu pačiu principu.



24 pav. Duomenų sąsajos pokyčiai migruojant iš senos sistemos į naują

Sudėtingiausia buvo perkelti pagrindinio naudojimo lentelių, susijusių su sąrašais, duomenis. Tam buvo sukurta speciali programa, atliekanti duomenų importą. Trumpai jos veikimo principą būtų galima aprašyti taip:

1. Atrinkami senoje lentelėje naudojamų sąrašų lentelių pavadinimai bei konkrečios panaudotos sąrašo lentelės lauko „Kodas“ reikšmės.
2. Pagal atrinktus sąrašų lentelių pavadinimus, iš kiekvienos jų į tarpinę lentelę surašomi lauko „Kodas“ reikšmės atitinkantys unikalūs identifikatoriai.
3. Atrinkti identifikatoriai įrašomi į perkeliama lentelę vietoj lauko „Kodas“ kartu su senosios lentelės duomenimis.

Vertinant duomenų migracijos procesą, galima teigti, kad duomenys iš senosios sistemos į naująją buvo perkelti be klaidų ir nepatiriant informacijos nuostolių. Duomenų migracijos procesą palengvino tai, kad naujoji DB buvo projektuojama išlaikant senojoje DB naudojamus lentelių ir jų atributų vardus – priešingu atveju, būtų buvę sunku užtikrinti automatizuotą duomenų perkėlimo procesą. Lentelių tarpusavio sąryšiams nustatyti buvo naudojamas senosios sistemos duomenų žodynas.

Taigi norint užtikrinti greitą ir patikimą duomenų migraciją, reikia labai kruopščiai išanalizuoti naudojamą paveldėtosios sistemos duomenų modelį, sudaryti jo ryšių su naujai projektuojamu modeliu diagramą, kuri palengvintų automatizuoto duomenų perkėlimo planavimą.

7.2 Rekonstrukcijos metodų efektyvumo įvertinimas

Rekonstrukcijos metodų efektyvumas nagrinėjamos sistemos atveju buvo įvertintas atsižvelgiant į jų įtaką sistemos charakteristikoms bei į kiekvieno rekonstrukcijos metodo laiko sąnaudas.

Kiekvieno naudoto rekonstrukcijos metodo įtakos sistemos charakteristikoms įvertinimas pateikiamas lentelėje 11.

11 lentelė. Rekonstrukcijos metodų įtaka sistemos charakteristikoms

Charakteristika	Naudojami rekonstrukcijos metodai		
	Atvirkštinė rekonstrukcija	Kodo retransliacija	Duomenų rekonstrukcija
Patikimumas		5%	15%
Našumas		2,5%	2,5%
Panaudojamumas		45%	
Palaikomumas	40%	10%	20%
Pakartotinis	5%		10%

11 lentelė. Rekonstrukcijos metodų įtaka sistemos charakteristikoms

Charakteristika	Naudojami rekonstrukcijos metodai		
	Atvirkštinė rekonstrukcija	Kodo retransliacija	Duomenų rekonstrukcija
panaudojimas			
Išplečiamumas	40%	5%	20%
Iš viso:	85%	67,5%	67,5%

Naudojamų rekonstrukcijos metodų įtaka įvardintoms sistemos charakteristikoms buvo skaičiuojama atsižvelgiant į šių charakteristikų kokybės tyrimo rezultatus (žr. skyrių 6.2.2). Kaip matyti iš lentelės **11**, didžiausią įtaką sistemos charakteristikų pagerinimui turėjo atvirkštinės inžinerijos procesas, šiek tiek mažiau – kodo retransliacija ir duomenų rekonstrukcija.

Laiko atžvilgiu, daugiausia sąnaudų buvo patirta atliekant atvirkštinę ir duomenų rekonstrukciją, nes šiems metodams automatizuoti nebuvo rasta tinkamų priemonių (būtų galima teigti, kad atvirkštinė rekonstrukcija užėmė 40%, o duomenų rekonstrukcija – 20% viso rekonstrukcijos proceso laiko). Kodo retransliacija buvo dalinai automatizuota, todėl truko šiek tiek trumpiau – apie 40%.

Taigi galima teigti, jog nagrinėjamos sistemos atveju efektyviausias buvo atvirkštinės ir duomenų rekonstrukcijos metodai, o kodo retransliacija ženkliai pagerino tik sistemos panaudojamumo charakteristiką (daugiausia vartotojo sąsajos atžvilgiu).

8 IŠVADOS

Magistro darbo metu rekonstruota paveldėtoji sistema, veikianti DOS aplinkoje ir parašyta FoxPro programavimo kalba. Atliktas rekonstrukcijos metodų pritaikymo tokio pobūdžio sistemoms tyrimas. Remiantis jo rezultatais, galima suformuluoti šias išvadas:

1. Rekonstrukcija tinka paveldėtųjų sistemų tobulinimo problemoms spręsti tuomet, kai jų vertė verslui yra didelė, tačiau kokybė – prasta.
2. Prieš atliekant paveldėtosios sistemos rekonstrukciją, patartina išanalizuoti kitų įmonių patirtį, sukauptą rekonstruojant panašias ir nagrinėjamą sistemas. Tai leidžia iš anksto nustatyti rekonstrukcijos metodus, įsigyti arba sukurti jiems įgyvendinti reikalingas priemones bei išvengti tokiam procesui būdingų rizikų.
3. Rekonstrukcijos procesas turi būti pradėtas nuo nagrinėjamoms paveldėtoms sistemoms kokybės įvertinimo. Jo metu taikomi metodai leidžia ne tik nustatyti tiriamos sistemos stovį, tačiau ir suformuluoti tikslus, kuriuos reikės pasiekti rekonstrukcijos pagalba. Be to, tyrimo metu surinkta informacija apie sistemos išeities kodą, turimą dokumentaciją, naudojamą duomenų modelį padeda efektyviau organizuoti patį rekonstrukcijos procesą.
4. Magistro darbo metu buvo pasirinkti trys rekonstrukcijos metodai: atvirkštinės rekonstrukcija, kodo retransliacija ir duomenų rekonstrukcija.
5. Sistemos kokybei didžiausią įtaką turėjo atvirkštinė ir duomenų rekonstrukcijos. Atvirkštinės rekonstrukcijos pagalba buvo sudaryta detali sistemos dokumentacija, kuri bus naudojama tolimesniam jos tobulinime. Duomenų rekonstrukcija apjungė naudotas DB lenteles į vieningą DB, o tai labai padidino sistemos patikimumą.
6. Kodo retransliacija buvo vienas paprasčiausių metodų realizacijos požiūriu, tačiau mažiausiai efektyvus sistemos kokybės atžvilgiu. Jo metu buvo sukurta grafinė vartotojo sąsaja, o sistema perkelta į naujesnės programavimo kalbos aplinką. Šie pakeitimai teikia tik laikiną naudą ir labiau juntami vartotojui, negu sistemą palaikantiems specialistams.
7. Rekonstrukcija yra tik būdas, leidžiantis atnaujinti paveldėtąją sistemą ir pritaikyti ją ateities patobulinimams. Tačiau jeigu naujos sistemos palaikymo procese nebus laikomasi bendrų programų inžinerijos principų, labai greitai ji ir vėl gali tapti paveldėtąja.
8. Tolimesnis šiame darbe pateiktos metodikos tobulinimas turėtų orientuotis į rekonstrukcijos proceso rizikos ir sąnaudų įvertinimą.

LITERATŪRA

1. Bazian, M.; Booth, J; Long J. **Использование Visual FoxPro™ 6.0.** – Москва, Санкт-Петербург, Киев, 1999.
2. Bisbal, J.; Lawless, D.; Wu, B.; Grimson, J. **Legacy Information Systems: Issues and Directions.** – Iš *IEEE Software*, Vol. 12, 1995. – [žiūrėta 2004-04-02]. Prieiga per internetą: <<http://www.informatik.uni-trier.de/~ley/db/journals/software/software12.html>>
3. Comella-Dorda, S.; Walnau, K.; Seacord, R. C.; Robert, J. **A Survey of Legacy System Modernization Approaches.** – Iš *Software Engineering Institute*, 2000. – [žiūrėta 2004-03-24]. Prieiga per internetą: <http://www.sei.cmu.edu/publications/documents/00_reports/00tn003.html>
4. Cotterell, M.; Hughes B. **Software Project Management.** – International Thomson Computer Press, 1995.
5. Fowler, M.; Scott, K. **UML Distilled: A Brief Guide to the Standard Object Modeling Language.** – Addison-Wesley, 2000.
6. Gold, N. **The Meaning of Legacy Systems.** – SABA Project Report, 1998.
7. Горев А. **Visual FoxPro 5.0. Книга для программистов.** - Журнал „FoxTalk“ ТОО „Эдэль“, 1997.
8. Mazanec, M. **Reengineering the „Enterprise“ Game to the Web.** – Iš *Scientific Literature Digital Library*, 2001. – [žiūrėta 2004-05-10]. Prieiga per internetą: <<http://citeseer.ist.psu.edu/543541.html>>
9. Miller, R. **Practical UML™: A Hands-On Introduction for Developers.** – [žiūrėta 2004-01-03]. Prieiga per internetą: <<http://community.borland.com/article/0,1410,31863,00.html>>
10. O’Byrne, P.; Wu, B. **LACE Frameworks and Technique – Identifying the Legacy Status of a Business Information System from the Perspectives of its Causes and Effects.** – [žiūrėta 2004-02-22]. Prieiga per internetą: <http://www.comp.dit.ie/pobyrne/Masters/1_obyrne_p.pdf>
11. Renaissance Consortium. **Renaissance Framework.** – Iš *Renaissance Web*, 1997. – [žiūrėta 2004-03-05]. Prieiga per internetą: <<http://www.comp.lancs.ac.uk/computing/research/cseg/projects/renaissance/pdf/D411.pdf>>
12. Renaissance Consortium. **Evolution Planning.** – Iš *Renaissance Web*, 1998 birželis. – [žiūrėta 2004-03-05]. Prieiga per internetą: <<http://www.comp.lancs.ac.uk/computing/research/cseg/projects/renaissance/RenaissanceWeb/project/Documents2.html>>

13. Renaissance Consortium. **Renaissance Method**. – Iš *Renaissance Web*, 1998 spal. – [žiūrėta 2004-03-05]. Prieiga per internetą:
<<http://www.comp.lancs.ac.uk/computing/research/cseg/projects/renaissance/RenaissanceWeb/project/Documents3.html>>
14. Rosenberg, L. H. **Software Re-engineering**. – Iš *NASA Software Assurance Technology Center SATC-TR-95-1001*, 1996. – [žiūrėta 2004-03-02]. Prieiga per internetą:
<<http://satc.gsfc.nasa.gov/support/reengrpt.PDF>>
15. Santhanakrishnan, S. **Re-engineering Legacy Applications**. – Iš *TATA Consultancy Services White Papers*, 2001. – [žiūrėta 2004-04-02]. Prieiga per internetą:
<http://www.tcs.com/0_whitepapers/htdocs/Rengg_legacy.pdf>
16. Sommerville, I. **Software Engineering**. – Addison-Wesley, 6th edition, 2001.]
17. Tilley, S.R.; Smith, D.B. **Perspectives on Legacy System Reengineering**. – Iš *Software Engineering Institute*, 1995. – [žiūrėta 2004-02-24]. Prieiga per internetą:
<<http://www.sei.cmu.edu/reengineering/lsysree.html>>

PRIEDAI

A. Dalykinės srities klasių aprašymas

Žemiau detaliai aprašytos sistemos architektūros specifikacijos dokumente pateiktoje dalykinės srities klasių diagramoje įvardintos klasės (žr. pav. 17 – 18).

Miškų urėdija

Klasifikacija: dalykinės srities klasė.

Apibrėžimas: klasė skirta dalykinės srities objektų duomenims saugoti, vėliau ji bus transformuota į DB lentelę.

Atsakomybės: klasė saugo duomenis apie miškų urėdiją.

Struktūra: klasė turi šiuos atributus:

- Urėdijos kodas (tipas:String; žymi urėdijos, kaip įmonės, kodą);
- Pavadinimas (tipas:String; žymi urėdijos pavadinimą).

Girininkijos

Klasifikacija: dalykinės srities klasė.

Apibrėžimas: klasė skirta dalykinės srities objektų duomenims saugoti, vėliau ji bus transformuota į DB lentelę.

Atsakomybės: klasė saugo duomenis apie miškų urėdijai priklausančias girininkijas.

Struktūra: klasė turi šiuos atributus:

- Gir. kodas (tipas:String; žymi girininkijos kodą);
- Pavadinimas (tipas:String; žymi girininkijos pavadinimą).

Biržės

Klasifikacija: dalykinės srities klasė.

Apibrėžimas: klasė skirta dalykinės srities objektų duomenims saugoti, vėliau ji bus transformuota į DB lentelę.

Atsakomybės: klasė saugo duomenis apie girininkijai priklausančias biržes.

Struktūra: klasę sudaro šie atributai:

- Biržės nr. (tipas:String; žymi biržės nr.);
- Kvartalo nr. (tipas:String; žymi kvartalo, kuriam priklauso biržė, numerį);
- Leidimas kirsti (tipas:String; žymi biržei išduoto leidimo kirsti numerį).

Sklypai

Klasifikacija: dalykinės srities klasė.

Apibrėžimas: klasė skirta dalykinės srities objektų duomenims saugoti, vėliau ji bus transformuota į DB lentelę.

Atsakomybės: klasė saugo duomenis apie girininkijai priklausančių biržių sklypus.

Struktūra: klasę sudaro šie atributai:

- Sklypo nr. (tipas:String; žymi sklypo nr.);
- Sklypo dalis (tipas:String; žymi sklypo dalį, kuri priklauso sklypo numeriui; šio lauko užpildyti nebūtina, nes sklypo nr. gali dalies ir neturėti);
- Plotas (tipas:Real; žymi sklypo plotą);
- Vyr. medžių rūšis (tipas:String; žymi sklype vyraujančią medžių rūšį).

Medienos ruošėjai

Klasifikacija: dalykinės srities klasė.

Apibrėžimas: klasė skirta dalykinės srities objektų duomenims saugoti, vėliau ji bus transformuota į DB lentelę.

Atsakomybės: klasė saugo duomenis apie medienos ruošėjus, kurie atlieka medienos gamybos darbus miškų urėdijai.

Struktūra: klasę sudaro šie atributai:

- Įmonės kodas (tipas:String; žymi medienos ruošėjo įmonės kodą);
- Įmonės pavadinimas (tipas:String; žymi medienos ruošėjo įmonės pavadinimą).

Taškavimo lapo antgalvė

Klasifikacija: dalykinės srities klasė.

Apibrėžimas: klasė skirta dalykinės srities objektų duomenims saugoti, vėliau ji bus transformuota į DB lentelę.

Atsakomybės: klasė saugo medienos taškavimo lapo apibendrintus duomenis.

Struktūra: klasę sudaro šie atributai:

- Nr. (tipas: Integer, taškavimo lapo numeris);
- Girininkija (tipas: String, girininkijos kodas);
- Kvartalo Nr. (tipas: String, kvartalo numeris);
- Sklypu Nr. (tipas: String, sklypų, įeinančių į nurodytą biržę ir kvartalą, sąrašas);
- Biržės Nr. (tipas: String, biržės numeris);
- Ruošėjas (tipas: String, ruošėjo kodas);

- Priėmimo vieta;
- Leidimo miškui kirsti Nr. (tipas: String, leidimo miškui kirsti Nr.).

Sortimentai

Klasifikacija: dalykinės srities klasė.

Apibrėžimas: klasė skirta dalykinės srities objektų duomenims saugoti, vėliau ji bus transformuota į DB lentelę.

Atsakomybės: klasė saugo informaciją apie galimus gaminti sortimentus.

Struktūra: klasę sudaro atributas pavadinimas (tipas:String; žymi sortimento pavadinimą, kuris susideda iš jo paskirties ir medžių rūšies).

Taškavimo lapo kūnelis vienetiniu metodu

Klasifikacija: dalykinės srities klasė.

Apibrėžimas: klasė skirta dalykinės srities objektų duomenims saugoti, vėliau ji bus transformuota į DB lentelę.

Atsakomybės: klasė saugo informaciją apie pagamintą medieną, kuri yra apskaitoma vienetiniu metodu.

Struktūra: klasę sudaro šie atributai:

- Rąsto nr. (tipas:String; žymi apskaitomo rąsto numerį);
- Diametras (tipas:Real; žymi apskaitomo rąsto skersmenį);
- Ilgis (tipas:Real; žymi apskaitomo rąsto ilgį);
- Tūris, m³ (tipas:Real; žymi apskaitomo rąsto tūrį, jeigu apskaitomi tokių pačių charakteristikų rąstai);
- Rąstų skaičius (tipas:Real; žymi apskaitomų rąstų kiekį).

Taškavimo lapo kūnelis grupiniu metodu

Klasifikacija: dalykinės srities klasė.

Apibrėžimas: klasė skirta dalykinės srities objektų duomenims saugoti, vėliau ji bus transformuota į DB lentelę.

Atsakomybės: klasė saugo informaciją apie pagamintą medieną, kuri yra apskaitoma grupiniu metodu.

Struktūra: klasė sudaryta iš tokių atributų:

- Rietuvės nr.;
- Rietuvės plotis (tipas:Real; žymi apskaitomos rietuvės plotį);

- Rietuvės aukštis sekcijomis (tipas:Real; žymi apskaitomos rietuvės plotį sekcijomis);
- Rietuvės vidutinis aukštis (tipas:Real; žymi apskaitomos rietuvės vidutinį aukštį);
- Tūris, erdm (tipas:Real; žymi apskaitomos rietuvės tūrį);
- Glaudumo koef. (tipas:Real; žymi apskaitomos rietuvės glaudumo koeficientą);
- Tūris, ktm.

Produkcijos išlaidavimo dokumento antgalvė

Klasifikacija: dalykinės srities klasė.

Apibrėžimas: klasė skirta dalykinės srities objektų duomenims saugoti, vėliau ji bus transformuota į DB lentelę.

Atsakomybės: klasė saugo informaciją apie sąskaitas-faktūras, medienos gabenimo važtaraščius, vidaus perdavimo važtaraščius, perkėlimo į medžiagas važtaraščius ir kitų išlaidų važtaraščius

Struktūra: klasę sudaro šie atributai:

- Išrašymo data (tipa: Date);
- Dokumento serija ir numeris;
- Apskaitos mėnuo;
- Siuntėjas;
- Pakrovimo vieta;
- Kvartalo Nr. (tipas: String, kvartalo numeris);
- Sklypu Nr. (tipas: String, sklypų, įeinančių į nurodytą biržę ir kvartalą, sąrašas);
- Biržės Nr. (tipas: String, biržės numeris);
- Pirkėjas;
- Kainininko ID;
- Apmokėjimo terminas;
- Apmokėjimo būdas;
- Išvežimo būdas;
- Dokumentą sudaręs asmuo.

Produkcijos išlaidavimo dokumento kūnelis

Klasifikacija: dalykinės srities klasė.

Apibrėžimas: klasė skirta dalykinės srities objektų duomenims saugoti, vėliau ji bus transformuota į DB lentelę.

Atsakomybės: klasė saugo informaciją apie pirminių išlaidavimo dokumentų detales: kas parduota, už kiek ir pan.

Struktūra: klasę sudaro šie atributai:

- Tūris (tipas:Real; žymi, koks rąstų tūris buvo nupirktas);
- Kiekis (tipas:Real; žymi, kiek buvo nupirkta rąstų);
- Pardavimo kaina (tipas: Real; žymi sortimento pardavimo kainą);
- Suma, Lt. (tipas:Currency; žymi, kiek reikia mokėti už įsigytą kiekį rąstų);
- PVM, Lt. (tipas: Currency; žymi, kiek reikia mokėti PVM už įsigytą rąstų kiekį);
- Nuolaida (tipas: Currency).

Kainininko antgalvė; kainininko kūnelis

Klasifikacija: dalykinės srities klasės.

Apibrėžimas: klasės skirta dalykinės srities objektų duomenims saugoti, vėliau ji bus transformuota į DB lentelę.

Atsakomybės: klasės saugo informaciją apie sortimentų bazinės kainas.

Struktūra: klasės sudaro šie atributai:

- Galiojimo pabaiga;
- Sandėlio tipas;
- Kaina (tipas:Currency; žymi sortimento kainą).

Pirkėjai

Klasifikacija: dalykinės srities klasė.

Apibrėžimas: klasė skirtos dalykinės srities objektų duomenims saugoti, vėliau ji bus transformuota į DB lentelę „Pirkėjai“.

Atsakomybės: klasė saugo informaciją apie urėdijos pirkėjus.

Struktūra: klasę „Pirkėjai“ sudaro atributas pavadinimas (tipas:String; žymi pirkėjo įmonės pavadinimą).

B. Valdymo klasių aprašymas

Šiame skyrelyje detaliai aprašomos pav. 15 – 16 pavaizduotos sistemos valdymo klasės.

Medienos taškavimo lapo valdymo klasė

Klasifikacija: valdymo klasė.

Apibrėžimas: klasė skirta atskirti vartotojo sąsajos klases nuo dalykinės srities klasių – ji atlieka valdiklio vaidmenį.

Atsakomybės: klasė realizuoja šiuos galimus veiksmus su taškavimo lapu: naujo taškavimo lapo sukūrimą, jo tipo nustatymą, įrašymą į DB bei pašalinimą iš DB.

Struktūra: klasę sudaro šie metodai:

- SukurtiNauja();
- NustatytiTaskavimoLapoTipa(type:String);
- IrasytiTaskavimoLapaIDB(taškavimo lapo duomenys);
- PasalintiTaskavimoLapaIsDB(key:String).

Skaičiavimai: klasė savo metodus vykdo tokiu eiliškumu:

1. Vartotojas pasinaudodamas medienos taškavimo lapo forma, sukuria naują medienos taškavimo lapą (SukurtiNauja()).
2. Užpildęs taškavimo lapo antgalvės laukus, vartotojas pasirenka taškavimo lapo tipą (nustatoma parametro type reikšmė) ir programa automatiškai sugeneruoja įvedimo laukus pasirinkto tipo medienos taškavimo lapui (NustatytiTaskavimoLapoTipa(type)).
3. Užpildęs taškavimo lapą, vartotojas jį išsaugo DB (IrasytiTaskavimoLapaIDB(taškavimo lapo duomenys)).

Taškavimo lapo vienetiniu metodu valdymo klasė

Klasifikacija: valdymo klasė.

Apibrėžimas: klasė skirta atskirti vartotojo sąsajos klases nuo dalykinės srities klasių – ji atlieka valdiklio vaidmenį.

Atsakomybės: klasė realizuoja taškavimo lapo kūnelio vienetiniu metodu įvedimą.

Struktūra: klasę sudaro šie metodas IvestiSortimentuInformacija().

Skaičiavimai: klasė savo metodus vykdo tokiu eiliškumu:

1. Vartotojas sudaro medienos taškavimo lapą (vykdomi medienos taškavimo lapo valdymo klasės metodai).

2. Naudodamasis medienos taškavimo lapo forma vartotojas iškviečia metodą `IvestiSortimentuInformacija()`.

Medienos priėmimo kortelės valdymo klasė

Klasifikacija: valdymo klasė.

Apibrėžimas: klasė skirta atskirti vartotojo sąsajos klases nuo dalykinės srities klasių – ji atlieka valdiklio vaidmenį.

Atsakomybės: klasė realizuoja šiuos galimus veiksmus su taškavimo lapu: naujos medienos priėmimo kortelės sukūrimą, jos įrašymą į DB bei pašalinimą iš DB.

Struktūra: klasę sudaro šie metodai:

- `SukurtiNauja()`;
- `IrasytiKorteleIDB(priemimo kortelės duomenys)`;
- `PasalintiKorteleIsDB(key:String)`.

Skaičiavimai: klasė savo metodus vykdo tokiu eiliškumu:

1. Vartotojas pasinaudodamas medienos priėmimo kortelės forma, sukuria naują medienos priėmimo kortelę (`SukurtiNauja()`).
2. Užpildęs kortelės antgalvės laukus, vartotojas priskiria kortelei taškavimo lapus (tai vykdo valdymo klasės “Taškavimo lapų priskyrimas” metodai).
3. Įvedęs priėmimo kortelę, vartotojas ją išsaugo DB (`IrasytiKorteleIDB(kortelės duomenys)`).

Sąskaitos-faktūros valdymo klasė

Klasifikacija: valdymo klasė.

Apibrėžimas: klasė skirta atskirti vartotojo sąsajos klases nuo dalykinės srities klasių – ji atlieka valdiklio vaidmenį.

Atsakomybės: klasės metodai realizuoja visų pirminių produkcijos išlaidavimo dokumentų sudarymo funkcijas.

Struktūra: klasę sudaro šie metodai:

- `SukurtiNauja()`;
- `IvestiAntgalve()`;
- `IvestiKuneli()`;
- `IrasytiIDB()`;
- `PasalintiIsDB()`.

Skaičiavimai: klasės metodai iškviečiami tokiu eiliškumu:

1. Vartotojas iškviesdamas pirminio išlaidavimo dokumento įvedimo formą, sukuria naują įrašą miškų urėdijos DB (SukurtiNauja()) ir atidaro dokumento antgalvės įvedimo formą (IvestiAntgalve()).
2. Užpildžius antgalvės laukus, įvedama sąskaitos-faktūros kūnelis (IvestiKuneli()).
3. Įvedus pirminį dokumentą, jis išsaugojamas DB (IrasytiIDB()).

C. Rekonstruotos sistemos grafinės vartotojo sąsajos pavyzdžiai

Biržių sąrašas(2003)														
Gir.	B. Nr.	Kv.	Plot.	V.m.r	Naud.	Dtg	Atk.	Kert. t.	Vid.st	Likvid.	Padar.	Būklė	Sklypai	Data
01	0003	0090	2.4	D	1	SA		570.5	0.553	534.4	235.7	T	16a,15a	2001.09.25
01	0101	0008	2.8		1	SA		372.8	0.468	327.4	290.4	B	5a	2002.09.18
01	0102	0010	2.7		1	SA		679.9	0.407	614.5	464.8	T	1,5	2002.09.18
01	0103	0010	0.6		1	SA		62.3	0.366	55.1	39.5	B	8a	2002.09.18
01	0104	0032	1.1		1	SA		269.8	0.600	239.2	215.7	B	17	2002.09.18
01	0105	0056	1.2		1	SA		233.8	0.618	210.0	149.0	B	4a	2002.09.18
01	0106	0058	3.7		1	SA		691.1	0.431	598.2	480.2	T	6a	2002.09.18
01	0107	0069	2.1		1	SA		423.5	0.345	390.0	213.4	B	5a	2002.09.18
01	0108	0075	3.3	D	1	SA	1	724.9	0.140	661.1	361.8	B	3a	2002.09.18
01	0109	0079	4.9		1	SA		808.8	0.452	742.8	346.6	B	3,4a	2002.09.18
01	0110	0087	3.2	A	1	SA		672.7	1.015	620.2	284.2	T	1a	2002.10.09
01	0111	0092	4.7		1	SA		963.9	0.313	842.1	632.8	T	3b	2002.09.18
01	0112	0095	4.9		1	SA		852.8	0.539	732.9	619.6	T	8a,16a	2002.09.18
01	0113	0101	1.9		1	SA		323.2	0.977	287.9	186.8	T	6,7a	2002.09.18
01	0114	0100	2.3		1	SA		448.9	0.522	389.6	309.9	T	1	2002.09.18
01	0115	0012	3.1	E	2	NC	1	42.9	0.083	41.6	3.3	T	13	2002.09.26
01	0116	0023	1.1	BT	2	NC	1	24.8	0.089	24.8	0.0	B	16	2002.09.26
01	0117	0023	2.7	L	2	NC	1	43.1	0.066	43.1	0.0	B	9	2002.09.26
01	0118	0086	12.1	B	2	NC	1	494.9	0.147	438.7	161.9	T	5	2002.09.26
01	0119	0100	0.9	E	2	NC	1	45.9	0.179	33.6	27.5	T	13	2003.01.30
01	0120	0106	0.7	E	2	NC	1	43.6	0.164	31.1	26.1	T	3	2003.01.30

25 pav. Biržių sąrašo ekrano forma

Biržės Nr. 3 aprašas

Girininkija	D1	Gustonių	
Kvartalas	90		
Kirtimo metai		2003	
Apsauginė kategorija	4	Ūkiniai	
Sklypo Nr.	16	a	
Vyr. medžių rūšis	D	Drebulynas	
Naudojimo būdas	1	Pagrindinio naudojimo	
Kirtimo būdas	1	Plynas	
Kertamas plotas	2.4	Ištraukimo atstumas, m	0
Medžių skaičius		964	
Dirvožemio TG	SA		
Atkūrimo būdo šifras			
Saug. pomiškio plotas		0.0	
Skalsumas	0.7		
Gamintojas			Išvežimas
Būklė	T	Įtaksuota biržė	
Įvykdymo data			

26 pav. Biržės įvedimo ekrano forma

Apvaliosios medienos taškavimo lapas

Numeris Sudarymo data

Girininkija

Biržės Nr. Kv. Nr. Sklypo Nr.

Medienos ruošėjas

Leidimo miškui kirsti Nr.

Sortimento tūriai, apskaičiuoti:

Viso: vnt. m³ riet. m³

Perdavė Priėmė

27 pav. Taškavimo lapo įvedimo ekrano forma

Apvaliosios medienos priėmimo kortelė

Serija Numeris Išrašymo data

Girininkija

Biržės Nr. Kv. Nr. Sklypo Nr.

Kirtimų rūšis Medienos ruošėjas

Leidimo miškui kirsti Nr.

Už laikotarpį nuo iki

Operacija Prod. vieta

Perdavė Priėmė

28 pav. Medienos priėmimo kortelės įvedimo ekrano forma

Sąskaita-faktūra

Serija Numeris Išrašymo data

Siuntėjas Būdos g-ja Prod. kat.

Pakr. v. Galutinis sandėlys Biržės Nr. Kv. Nr.

Pirkėjas UAB "Runika"

Apmok. terminas, d. Apmok. būdas Kreditas

Išvežimas

Sąskaitos-faktūros turinys

	Kiekis	<input type="text" value="15.000"/>	Viso be PVM	<input type="text" value="2025.00"/>
<i>Viso:</i>			PVM	<input type="text" value="18 %"/>
			Viso su PVM	<input type="text" value="2389.5000"/>

29 pav. Sąskaitos-faktūros įvedimo ekrano forma

D. Dalykinės srities terminų žodynas

Apvalioji mediena – nukirstas ir nugenėtas medis be viršūnės, kuris gali būti skersai supjaustytas arba ne.

Apvaliosios medienos išlaidavimas (pardavimas) – skiriamos dvi apvaliosios medienos išlaidavimo rūšys: mažmeninis pardavimas ir didmeninis pardavimas. Mažmeninis medienos pardavimas vyksta visus metus urėdijos (pardavėjo) nustatytais mažmeninėmis kainomis. Didmeninis pardavimas vykdomas metinio pardavimo konkursuose, pirkėjo ir pardavėjo susitarimu, aukcionuose.

Apvaliosios medienos priėmimas – medienos matavimas, tūrio nustatymas, klasifikavimas (rūšiavimas) pagal medžių rūšis, sortimentus ir kokybę, ženklinimas įrašant reikiamus duomenis į medienos priėmimo dokumentus. Nepriimta mediena negali būti parduodama pirkėjams.

Apvaliosios medienos priėmimo kortelė – griežtos atskaitomybės dokumentas, turintis seriją ir numerį. Kiekvienam medienos priėmimui užpildoma atskira apvaliosios medienos priėmimo kortelė. Duomenys apie priimtą medieną įrašomi iš taškavimo lapo, atskirai užpildomo, matuojant medieną vienetiniu arba grupiniu metodais.

Apvaliosios medienos taškavimo lapas – pirminis medienos apskaitos dokumentas, pildomas vienu egzemplioriumi atskirai matuojamiems vienetiniu ir grupiniu metodais sortimentams.

Generalinė miškų urėdija prie Aplinkos ministerijos – valstybinių miškų, priskirtų miškų urėdijoms, ūkinio valdymo institucija, organizuojanti bei koordinuojanti šių miškų atkūrimą, priežiūrą, apsaugą ir miško išteklių naudojimą.

Kompleksinė miškų ūkio veikla – veikla, apimanti miškų atkūrimą, priežiūrą, apsaugą, racionalų miškų išteklių naudojimą ir prekybą mediena bei miško ištekliais.

Miškų urėdija – valstybės įmonė, turto patikėjimo teise valdanti, naudojanti valstybinius miškus ir jais disponuojanti įstatymų nustatyta tvarka, taip pat vykdanči juose kompleksinę miškų ūkio veiklą ir kitą įmonės nuostatuose numatytą veiklą.

Medienos atkrovimas – medienos pakrovimas į pirkėjo transportą.

Medienos gabenimo važtaraštis – apvaliosios medienos išlaidavimo dokumentas, sudaromas tuomet, kai girininkai vykdo tik medienos atkrovimą, bet neišrašo pirkėjui sąskaitų-faktūrų.

Medienos perklasifikavimas (perrūšiavimas) – medienos kokybės klasės (rūšies) pakeitimas arba priskyrimas kitam sortimentui (paskirčiai).

Perkėlimo į medžiagas važtaraštis – apvaliosios medienos išlaidavimo dokumentas, sudaromas tuomet, kai mediena sunaudojama urėdijos technologinėms reikmėms, pvz., klojant klojinius, kad lengviau miške pravažiuotų urėdijos transportas.

Sortimentas – nustatytos paskirties apvalioji mediena.

Sortimentų apskaitymas grupiniu metodu – sortimentai, matuojami grupiniu metodu, priimami sukrauti į rietuves kirtavietėse, miško sandėliuose, galutiniuose sandėliuose bei pakrauti į transporto priemones. Priimant medieną transporto priemonėse, pirmiausia užpildomas Apvaliosios medienos taškavimo lapas (matuojant grupiniu metodu), tik po to išrašomi medienos gabenimo dokumentai.

Sortimentų apskaitymas vienetiniu metodu – sortimentai, matuojami vienetiniu metodu, paprastai priimami kirtavietėse (biržėse, valksnose, šalia valksnų). Kai kirtavietėse priimti visų sortimentų neįmanoma, jie priimami miško sandėliuose. Taikant medienos ruošos technologiją stiebais, sortimentai priimami galutiniuose sandėliuose.

Vidaus perdavimo važtaraštis – apvaliosios medienos išlaidavimo dokumentas, sudaromas tuomet, kai mediena pergabenama iš vieno urėdijos sandėlio į kitą.