

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

Paulius Žydelis

**ERP SISTEMŲ FUNKCIONALUMO IR
VARTOTOJO REIKALAVIMŲ SUTAPATINIMO
MODELIS**

Magistro darbas

**Vadovas
doc. dr. R. Butleris**

KAUNAS, 2006

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

**ERP SISTEMŲ FUNKCIONALUMO IR
VARTOTOJO REIKALAVIMŲ SUTAPATINIMO
MODELIS**

Magistro darbas

Vadovas
doc. dr. R. Butleris

Konsultantė
dokt. B. Misevičiūtė

Recenzentas
doc. dr. V. Pilkauskas

Atliko
IFM 0/4 gr. stud.
P. Žydelis

KAUNAS, 2006

SUMMARY

THE MODEL OF MATCHING ERP SYSTEM FUNCTIONALITY TO CUSTOMER REQUIREMENTS

This work treats the importance of Enterprise Resource Planning (ERP) system and problems that face users. One of the biggest problem is how to choose the right system spending less time and money. The purpose of this work is to design the method of matching requirements raised by system and customer, and to implement this method by creating model which will collect, match and return acceptable results to customer.

TURINYS

ĮVADAS	7
1. ERP SISTEMŲ PRITAIKYMO ORGANIZACIJOS REIKALAVIMAMS PRINCIPŲ ANALIZĖ .	9
1.1. Analizės tikslas	9
1.2. Tyrimo sritis, objektas ir problema	9
1.3. ERP sistemų taikymo srities analizė	11
1.3.1. ERP sistemų vartotojų bendrosios aibės, jų savybės	12
1.3.2. Vartotojų sprendžiamos problemos ir siekiami tikslai.....	13
1.4. Reikalavimų nesutapimo problemos sprendimo metodų analizė	13
1.5. Darbo tikslas ir siekiami privalumai	19
1.5.1. Siekiamo modelio apibrėžimas.....	20
1.5.2. Reikalavimų sutapatinimo sistemos kompiuterizuojamos funkcijos	21
1.6. Kuriamo modelio reikalavimų specifikacija	22
1.6.1. Funkciniai reikalavimai	22
1.6.2. Nefunkciniai reikalavimai.....	23
1.6.3. Rizikos faktorių analizė	25
1.6.4. Rezultato kokybės kriterijai	25
1.6.5. Sistemos vartotojų panaudojimo atvejai.....	26
1.6.5.1. Vartotojo – programuotojo panaudojimo atvejai	26
1.6.5.2. Vartotojo – projektuotojo panaudojimo atvejai	27
1.6.5.3. Vartotojo – analitiko panaudojimo atvejai	27
1.7. Projektavimo metodų, priemonių parinkimas kuriamo modelio realizacijai bei naudojimui.....	29
1.8. Kompiuterizuojamos sistemos varianto parinkimas.....	30
1.9. Išvados.....	31
2. REIKALAVIMŲ SUTAPATINIMO MODELIS IR JO REALIZACIJA.....	33
2.1. Modelio projekto tikslas.....	33
2.2. Kuriamo modelio realizacijos bei veikimo aplinkos apibrėžimas.....	33
2.3. Reikalavimų sutapatinimo modelio architektūra.....	33
2.3.1. Duomenų surinkimo modulis	34
2.3.2. Duomenų apdorojimo, analizės modulis	35
2.3.3. Duomenų saugojimo modulis	36
2.4. Sistemą sudarantys duomenų šrautai.....	36
2.4.1. Įėjimo informacijos struktūra	37
2.5. Reikalavimų aprašymuose naudojamų duomenų struktūrų ir operacijų aprašai.....	41

	4
2.6. Sistemos veiklos modelis	42
2.6. Reikalavimų sutapatinimo sistemos naudojami algoritmai	44
2.6.1. Pradinių duomenų – funkcionalumo ar reikalavimų aprašų - skaitymas.....	45
2.6.2. Reikalavimų aprašų talpinimas duomenų saugykloje bei skaitymas iš jos	46
2.6.3. Reikalavimų aprašų sutapatinimas	51
2.7. Duomenų bazės schema	58
2.8. Vartotojo sąsajos modelis	59
2.9. Sistemos realizacijos modelis	62
2.10. Sistemos naudojimo instrukcija.....	63
2.11. Projekto išvados.....	63
3. Eksperimentinis reikalavimų sutapatinimo modelio tyrimas.....	64
3.1. Eksperimentinis sukurto modelio diegimas	64
3.2. Sistemos naudojimo instrukcija	64
3.3. Sukurto sutapatinimo modelio įvertinimas	66
IŠVADOS.....	69
LITERATŪRA.....	71
PRIEDAI	74
1 PRIEDAS. Paskelbtas magistro darbo straipsnis.....	74
2 PRIEDAS. OPM metodologija, žymėjimai.	78
3 PRIEDAS. Reikalavimų saugojimui naudojamų duomenų bazės lentelių aprašai.....	81
4 PRIEDAS. Reikalavimų sutapatinimo modelio testiniai duomenys	87

Lentelių sąrašas

1.1 lentelė. Nefunkciniai reikalavimai.....	23
1.1 lentelė (tęsinys). Nefunkciniai reikalavimai	24
1.2 lentelė. Rizikos faktoriai ir jų pašalinimo būdai.....	25
1.3 lentelė. Sistemos kokybės kriterijai ir vertinimo faktoriai	25
2.1 lentelė. Loginio objekto elementai	39
2.2 lentelė. Loginės būsenos elementai	40
2.3 lentelė. Loginio proceso elementai.....	40
2.4 lentelė. Duomenų bazės lentelės, jų paskirtis	59
3.1 lentelė. Sutapatavimo modelio kokybės kriterijų įvertinimas	67
3.1 lentelė (tęsinys). Sutapatavimo modelio kokybės kriterijų įvertinimas.....	68

Paveikslėlių sąrašas

1.1 pav. Vartotojo (analitiko) veiklos funkcijos.....	11
1.2 pav. C.Rolland ir N.Prakash siūlomas ERP reikalavimų pritaikymo modelis [1, 2].....	14
1.3 pav. ERP reikalavimų sutapatavimo proceso modelis	15
1.4 pav. „ERP Evaluation Center“ modelio reikalavimų sugretinimo etapai;	16
vartotojo ir sistemos veikimo sritys	16
1.5 pav. AMOS sistemos schema [7]: vartotojo sąsaja, paieškos variklis ir duomenų bazė.....	18
1.6 pav. Schematiškas paieškos algoritmas [8].....	18
1.7 pav. Paketų aprašų principo struktūra atvaizduota UML principu [7]	19
1.8 pav. Planuojamos sistemos architektūra.....	20
1.9 pav. Planuojamos sistemos funkcijos	21
1.10 pav. Informacinės sistemos vartotojų grupės.....	26
1.11 pav. Vartotojų grupės <i>Programuotojas</i> panaudojimo atvejų schema	26
1.12 pav. Vartotojų grupės <i>Projektuotojas</i> panaudojimo atvejų schema	27
1.13 pav. Vartotojų grupės <i>Analitikas</i> panaudojimo atvejų schema.....	28
1.14 pav. Reikalavimų sutapatavimo sistemos uždaviniai	30
1.15 pav. Reikalavimų pritaikymo sistemos uždaviniai	30
2.1 pav. Sutapatavimo modelio architektūros loginė schema.....	34
2.2 pav. Duomenų surinkimo modulio loginė schema	34
2.3 pav. Duomenų apdorojimo, analizės modulio loginė schema	35
2.4 pav. Duomenų saugojimo modulio loginė schema.....	36
2.5 pav. Sistemos duomenų srautų schema	37
2.6 pav. Įėjimo duomenų schema.....	38

2.7 pav. Sutapatinimo modelyje naudojamų duomenų struktūrų ir operacijų aprašai.....	41
2.8 pav. Sistemos veiklos modelis	42
2.9 pav. Nuskaitymo, normalizavimo veiklos loginė schema.....	43
2.10 pav. Nuskaitymo veiklos loginė schema.....	43
2.11 pav. Rezultatų formavimo veiklos loginė schema.....	44
2.12 pav. Pradinių duomenų nuskaitymo algoritmo blokinė schema.....	45
2.13 pav. Pradinių duomenų – funkcionalumo ar reikalavimų aprašų – nuskaitymo algoritmas.....	46
2.14 pav. Duomenų saugyklos užpildymo algoritmo blokinė schema.....	47
2.15 pav. Algoritmas, skirtas duomenų saugyklos užpildymui reikalavimų aprašu.....	48
2.16 pav. Reikalavimų aprašų skaitymo iš duomenų bazės algoritmo blokinė schema.....	49
2.17 pav. Reikalavimų aprašų skaitymo iš duomenų bazės algoritmas.....	50
2.18 pav. Reikalavimų aprašų sutapatinimo algoritmas (1-a dalis).....	51
2.19 pav. Reikalavimų aprašų sutapatinimo algoritmas (2-a dalis).....	52
2.20 pav. Vertinamų sistemų sulyginimo algoritmo schema. Pradžia.....	52
2.21 pav. Reikalavimų sutapatinimas sisteminiame lygmenyje.....	53
2.22 pav. Modulių sutapatinimo metodo blokinė schema.....	54
2.23 pav. Objektų sutapatinimo metodo blokinė schema.....	54
2.24 pav. Esybės tapačių ryšių paieškos metodo blokinė schema	55
2.25 pav. Reikalavimų aprašų sutapatinimo algoritmas (2-a dalis).....	56
2.26 pav. Bendros duomenų bazės loginė schema.....	58
2.27 pav. OPCAT projektavimo įrankio vartotojo sąsaja.....	60
2.28 pav. Suvestų duomenų pateikimo ir įverčių nustatymo langas.....	60
2.29 pav. Bendrų rezultatų pateikimo langas	61
2.30 pav. Sutapatintų rezultatų pateikimo langas	61
2.31 pav. Reikalavimų sutapatinimo sistemos komponentų modelis	62
2.32 pav. Informacinės sistemos diegimo diagrama.....	62
3.1 pav. Modelyje atvaizduotas sistemos funkcionalumo medis.....	65
3.2 pav. Sutapatinimo metodas. Apibendrintieji rezultatai	66

IVADAS

Šiuo metu rinkoje sparčiai populiarėja komerciniai produktai – tokie kaip įmonių išteklių valdymo (angl. *Enterprise Resources Planning* (ERP)) sistemos.

ERP – tai verslo strategija, kurioje integruotos išteklių valdymo, finansų, prekybos bei kitos funkcijos, padedančios stebėti ir optimizuoti įmonių resursus. Šios sistemos sukurtos tobulinant gamybos įmonių poreikių valdymo sistemas (angl. *Manufacturing Requirements Planning* (MRP)) dėka atsiradusių reliacinių duomenų bazių valdymo sistemų, CASE priemonių, ketvirtos-kartos (4GL) kalbų vystymo įrankių ir kliento/serverio architektūros. Įdiegus ERP sistemas, įmonės gali sklandžiau reguliuoti savo veiklos procesus, greičiau ir efektyviau atlikti reikalingas analizes ir gauti atitinkamus rezultatus. Šios sistemos įmonėms leidžia stebėti ir neatsilikti nuo rinkos, įvertinant joje atsiradusius pokyčius.

Pagal Partner Group atliktus ERP sistemų tyrimus, buvo prieita išvadų, kad tokių sistemų naudojimas turi aibę trūkumų:

- 1) didelė kaina;
- 2) sudėtingas bei daug laiko užimantis realizavimas;
- 3) sistemos atnaujinimo būtinybė;
- 4) sudėtingas pritaikymas prie specifinių vartotojų keliamų reikalavimų;
- 5) pritaikymo plačiai auditorijai būtinybė;
- 6) per maža ERP sistemų tiekėjų techninė parama;
- 7) didelė tarptautinių nesuderinimų tikimybė.

Siekiant sumažinti šių trūkumų kiekį, orientuojamasi į vieną iš sričių – šių sistemų sklandų pritaikymą prie vartotojų keliamų reikalavimų. Čia iškyla poreikis išnagrinėti žinomų (prieinamų) ERP sistemų funkcionalumo specifikavimo metodus bei jų diegimo instrukcijas, o taip pat atlikti ir vartotojo reikalavimų analizę.

Ši sritis jau buvo nagrinėta užsienio autorių, tačiau dažniausiai tik teoriškai. Taigi, pasinaudojus autorių anksčiau suformuluotomis teorijomis bei siūlomais problemos sprendimo metodais, buvo siekiama sudaryti patikslintą ERP sistemų funkcionalumo specifikavimo metodą ir sukurti jo programinį prototipą. Sukurta sistema užtikrintų galimybę kompiuterizuotu būdu atlikti ERP sistemos funkcionalumo ir vartotojo reikalavimų sutapatinimą. Iki šiol tai buvo atliekama tik teoriniame lygmenyje.

Šiuo metu, kaskart atsiradus tokios sistemos panaudojimo atvejui, yra atliekama analizė, tiriant konkrečias sistemos bei vartotojų reikalavimų specifikacijas. Dažnai tokia problema sprendžiama vienu iš pateiktų būdų:

- 1) pasirenkama ir įsigyjama sistema, tinkamiausiai deranti prie vartotojų reikalavimų;
- 2) sistema įsigyjama pritaikant atlikti specifines, vartotojo keliamas užduotis.

Ypatingas dėmesys skiriamas pirmajam būdai, kuris pasirenkamas dažniausiai. Čia, prie vartotojų keliamų reikalavimų, priešpastatomi papildomi reikalavimai patiems vartotojams – tokie kaip programinės įrangos pasirinkimo supratimas, egzistuojančios programos įrangos analizė, funkcinės galimybės, sąsaja su vartotoju, suderinamumas su įvairių platformų sistemomis.

Priešingai nei pirmasis, „pritaikomąjį“ būdą naudoja ERP sistemų platintojai, atsakingi už jų diegimą. Praktikoje yra nemažai pavyzdžių, rodančių, kad pritaikymas yra sudėtingas procesas, kuris tuo pačiu yra ir pagrindinė nesėkmingo sistemų diegimo priežastis.

Šiame darbe pateikiama informacija atlikta trim etapais – analizuojant ERP sistemų taikymo sritį, projektuojant bei realizuojant reikalavimų sutapatavimo metodą, jį išpildantį modelį ir įvertinant jo tinkamumą šios problemos sprendimui.

Analizės metu apibrėžiamas darbo uždavinys, tiriami kitų autorių darbai bei analogiškos sistemos, nustatoma galima sistemos architektūra, jos atliekamos funkcijos. Taip pat identifikuojamos vartotojų grupės su savo tikslais. Nurodomi reikalavimai bei kriterijai pagal kuriuos galima vertinti sukurtą modelį.

Projektavimo proceso metu patikslinama sistemos architektūra, nustatomos duomenų struktūros, kuriomis bus dirbama. Sudaromi algoritmai išpildantys ankstesniame etape iškeltas sistemos funkcijas. Projektuojama duomenų bazės lentelių struktūra, kuri tiktų naudojimui darbo metu. Realizuojamas visą tai išpildantis modelis.

Galiausiai, apibrėžiant sistemos naudojimo diegimo bei naudojimo instrukcijas yra atliekamas sukurto sutapatavimo modelio įvertinimas. Pagal šį įvertinimą paaiškės ar darbo metu sukurtas loginis problemos sprendimo būdas bei jį realizuojanti programa yra tinkama naudojimui bei galimam funkciniam praplėtimui.

1. ERP SISTEMŲ PRITAIKYMO ORGANIZACIJOS REIKALAVIMAMS PRINCIPŲ ANALIZĖ

1.1. Analizės tikslas

Prieš darant praktinį darbą svarbu atlikti ERP sistemų funkcionalumo specifikuojančių metodų analizę, apžvelgiant mokslinius šaltinius apie tokių sistemų pritaikymo principus konkrečiam kompiuterizuojamam objektui.

Siekiami išsiaiškinti vartotojų poreikius, kurių pagrindu pasirenkama tinkama ERP sistema. Pagal esamą literatūrą ar gyvenimiškus pavyzdžius bandoma išskirti bendrinius, dažniausiai pasikartojančius, vartotojų keliamus reikalavimus (kompiuterizuojamo objekto reikalavimus) ir, tuo pačiu, reikalavimus, keliamus patiems vartotojams.

Atliktos analizės pagrindu pasiūlomas modelis, apimantis reikalavimus ERP sistemos funkcionalumui bei vartotojo reikalavimams aprašyti tokia forma, pagal kurią būtų galima organizuoti sklandų vartotojo reikalavimų „padengimą“ ERP sistemos galimybėmis, operatyviai išskiriant papildomai reikalingas funkcijas arba esamų funkcijų plėtros ar keitimo poreikius.

1.2. Tyrimo sritis, objektas ir problema

Įmonių išteklių valdymo sistemos (ERP) – tai informacinės sistemos, kurios apjungia daug veiklos procesų, susijusių su įmonių vykdomomis funkcijomis - gamyba, logistika, prekių, paslaugų paskirstymu, finansine apskaita. Įmonės, naudojančios tokio pobūdžio programinius paketus, gali lengviau stebėti bei tvarkyti tokias sritis kaip planavimas, gamyba, pardavimai, sąskaitų tvarkymas, žmoniškųjų resursų valdymas ir kt. ERP sistemų pagalba galima valdyti visos įmonės bei atskirai ją sudarančių padalinių veiklą. Tai laiko visą organizaciją kartu prijungiant naudojamus išorinius verslo procesus.

Pagrindiniai ERP naudojimo privalumai yra ta, kad šio tipo programinė įranga:

- 1) apima verslo procesus;
- 2) yra modulinė;
- 3) yra integruotas paketas;
- 4) apima įmonių veiklą už jos ribų – sąryšius su tiekėjais, klientais bei partneriais;
- 5) apima visas arba bent jau didžiąją dalį įmonių veiklos funkcijų.

ERP sistemos yra vienos iš stambiausių dabar egzistuojančių programinės įrangos paketų. Tai yra dėlto, kad tiekėjai bando savo produktą pritaikyti daugeliui įmonių, kurių veiklos bei valdymo formos dažnai skiriasi vienos nuo kitų.

Taigi, nors tokie produktai turi begalę privalumų, yra ir keletą labai reikšmingų trūkumų:

- 1) brangus sistemos diegimas bei palaikomumas;
- 2) sudėtingas kai kurių sistemų naudojimas;
- 3) problemos vienoje sferoje gali turėti poveikį visoms kitoms sritims.

Dar daugiau esminių trūkumų pastebima, kuomet ERP programiniai paketai yra integruojami su kitomis valdymo sistemomis.

Tokių sudėtingų bei didelių sistemų diegimas dažniausiai reikalauja nemažo analitikų, programuotojų bei vartotojų būrio, o tai sudaro milijonines kompanijų (ypač tarptautinių) išlaidas. Organizacijos, pasirinkusios tokios programinės įrangos naudojimą, prašo sistemų tiekėjų ar kitų konsultacinių įmonių pagalbos. ERP konsultavimas yra skaidomas į du lygius: biznio bei techninį konsultavimą. Pirmajame lygmenyje nagrinėjami įmonės veiklos procesai bei atliekamas jų sutapatinimas su atitinkamais procesais, realizuotais ERP sistemoje. Taip ji yra pritaikoma prie įmonės keliamų poreikių. Techninę paramą dažnai sudaro programavimas bei sistemų konfigūravimas. Siekiant patenkinti savo klientų poreikius, daugelis ERP tiekėjų suteikia galimybę keisti jų sukurtą programinę įrangą pritaikant ją prie keliamų reikalavimų.

Šiuo metu yra nemažai ERP sistemų tiekėjų - tai: Best, AccPac, Open Systems, Microsoft, SAP, Epicor, PeopleSoft, Oracle, SBT. Tai tik pačios populiariausios kompanijos, aprūpinančios verslo įmones jiems reikalinga programine įranga, pasaulyje jų yra kur kas daugiau. Tačiau jų kuriamą produkciją apvienija begalo didelė kaina, siekianti pusę milijono dolerių už programinio paketo licenciją, o kartais ir daugiau. Prie to prisideda dar ne ką mažesnė sistemos pritaikymo bei diegimo suma.

Šiuo darbu ir siekiama atlikti tokių sistemų analizę bei pasiūlyti vartotojams programinės įrangos modelį, kurio pagalba būtų galima surinkti bei saugoti žinomų ERP sistemų funkcionalumo specifikacijas. Asmenys, pasinaudoję šiuo produktu, galėtų sumažinti savo išlaidas bei sutrumpinti laiką tinkamos ir jų organizacijos veiklą labiausiai atitinkančios sistemos paieškai. Tai aktualu todėl, kad specifikuotos programinės įrangos kūrimo kaina yra pakankamai didelė, be to išlieka didelė rizika, kad laikui bėgant dėl atsiradusių rinkos pokyčių gali pakisti įmonės veiklos kryptys, jos gali išsiplėsti, tuomet sukurta sistema nebeatliks arba netinkamai atliks savo funkcijas, o reinžinerijos procesas reikalauja dar didesnių išlaidų.

Apibendrinant, galima apibrėžti atliekamo tyrimo sritis, objektą bei problemą:

Tyrimo sritis - žinomų (prieinamų) ERP sistemų funkcionalumo specifikavimo metodai bei jų diegimo instrukcijos, o taip pat ir vartotojų reikalavimų specifikavimo sfera. Siekiama realizuoti šių dviejų sričių sutapatinimo modelį, suteiksiantį galimybę atlikti sklandų ERP sistemų diegimą.

Tyrimo objektas – ERP sistemų funkcionalumo specifikavimo metodas ir jo programinė realizacija, užtikrinanti galimybę kompiuterizuotu būdu atlikti ERP sistemos funkcionalumo ir

virtotojo reikalavimų sutapatinimą, kompleksiskai panaudojant specializuotą specifikacijos saugyklą bei žinomas sistemų projektavimo priemones (CASE).

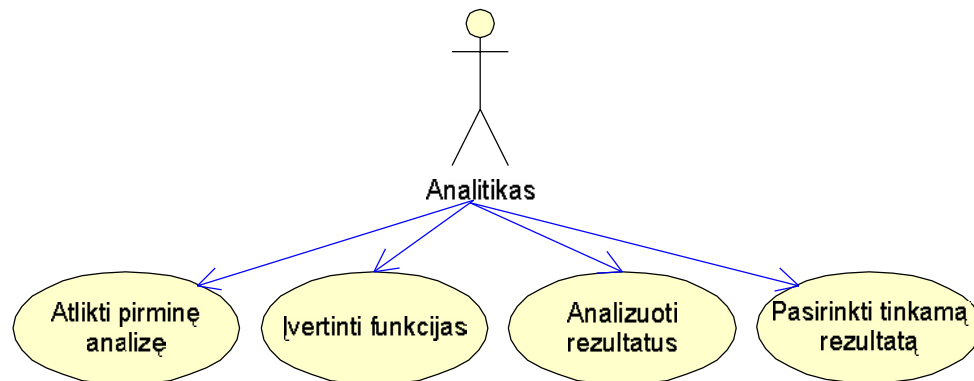
Tyrimo problema - siekiama išspręsti susidarančią nesėkmingą sistemų diegimų grėsmę susidūrus su organizacijų keliamais reikalavimais. Tokia grėsmė iškyła dėl ERP specialistų ir organizacijų atstovų nesusikalbėjimo, kadangi suderinimo procesas yra sutelkiamas į kuriamų sistemų funkcionalumą.

1.3. ERP sistemų taikymo srities analizė

Planuojama sukurti sistemą, kurios pagalba galima būtų specifiukuoti, saugoti ERP sistemų funkcijas bei jas identifikuoti ieškant tinkamo sprendimo pasirenkant reikalingą programinę įrangą.

Sistemos realizavimui reikia sudaryti ERP sistemų funkcionalumo identifikavimo metodiką.

Ši metodika aktuali analitikams, kurie apibrėžia organizacijos poreikius ir juos supaprastina iki ERP teikiamų funkcijų lygio.



1.1 pav. Vartotojo (analitiko) veiklos funkcijos

Stambiausios sritys, su kuriomis analitikas turėtų susipažinti prieš įsigyjant ERP sistemą bei atlikti tyrimus, siekiant išsiaiškinti visus galimus poreikius:

- įmonės valdymas;
- darbuotojų, padalinių kontrolė;
- išteklių kontrolė;
- buhalterija;
- žmogiškųjų resursų valdymas;
- klientų aptarnavimas;
- projektų valdymas;
- elektroninė prekyba;
- prekybos formos;
- pirkimai/pardavimai (marketingas)

- įmonės modeliavimas;
- produkcijos planavimas;
- kaina.

Atlikus analizę, gautos išvados konkretizuojamos ir įvertinamos kaip funkcijos, stengiantis jas priartinti prie ERP sistemų funkcijų. Jei tai nebus padaryta, gali atsirasti informacijos nepanaudojimo atvejais ir planuojamos sistemos gražinta informacija rezultatai gali neatitikti laukiamų rezultatų. Gavus rezultatus atliekama detalesnė kelių pateiktų ERP sistemų analizė, įvertinant smulkesnius reikalavimus, keliamus ERP sistemoms.

1.3.1. ERP sistemų vartotojų bendrosios aibės, jų savybės

ERP sistemos padeda valdyti daugelį verslo sferų. Kiekvienoje iš jų veikia savos vartotojų grupės su savais tikslais ir problemomis. Sistemoje dalyvauja ir tiesiogiai, ir netiesiogiai su įmonės veikla susiję asmenys. Aprėpiant visas sferas, galima išskirti tokias apibendrintas vartotojų grupes: organizatorius, vykdytojus, naudotojus bei analitikus.

Organizatoriai – tai asmenys, atsakingi už įmonės padalinių, konkrečių darbų vykdymą. Konkretizuojant, tai būtų padalinių, projektų vadovai. Jie naudoja sistemoje sukauptą informaciją apie darbus, jų vykdymo tvarką, pobūdį, žmogiškuosius resursus, pildo minėtą informaciją, kuria vėliau gali patys naudotis bei ja naudotusi įmonės savininkai, analitikai, planuotojai bei kiti vartotojai, kuriems ši informacija reikalinga vykdant įmonės veiklą.

Vykdytojai – asmenys, kurie per nustatytą terminą (konkretų laikotarpį ar iki rezultatų gavimo momento) atlieka organizatorių paskirtas užduotis. Jie naudojami sistema joje registruojant atliktus darbus, sugaištą laiką, susidariusias uždavinių sprendimo problemas bei jų pašalinimo būdus. Šiai vartotojų grupei priskiriami programuotojai, projektuotojai, dizaineriai, kiti tiesiogiai ar per kitus asmenys veikiantys bei organizatorių užduotis vykdantys asmenys.

Naudotojai – asmenys, naudojantys sukauptą informaciją apie įmonės veiklą. Čia veikia įmonių savininkai, tiekėjai, klientai bei kiti vartotojai.

Analitikai – vartotojai, kurie naudoja sukauptą informaciją apie įmonės veiklą, darbus, veda finansinę apskaitą, vykdo strateginius planavimus, siekiant pagerinti įmonės veiklą, optimizuojant laiko, išteklių sąnaudas, pildo numatomus planus. Šiai grupei priskiriami finansininkai, įmonių, projektų analitikai ir kiti vartotojai, renkantys bei nagrinėjantys sukauptą informaciją, pateikiantys gautas išvadas ar sąmatas.

Čia pateikti apibendrinti ERP sistemų vartotojai, kurie naudojami jau iš anksto parinkta bei įdiegta tiksliausiai įmonės valdymui tinkančia resursų valdymo sistema. Žinant būsimus sistemos vartotojus galima išskirti ir sistemos pasirinkime dalyvausiančius asmenis. Tai būtų: analitikai, organizatoriai.

Analitikai pagal dabartines įmonės veiklos sritis bei atliktus strateginius ateities planavimus gali parinkti sistemą, kuri realiausiai atitinka įmonės dabarties ir ateities planus.

Tuo tarpu organizatoriai, kurių darbas yra vadovauti vykdomiems projektams, gali patikslinti sistemos pasirinkimą pagal konkrečius ar apibendrintus esminius darbų vykdymo kriterijus.

1.3.2. Vartotojų sprendžiamos problemos ir siejami tikslai

Pagrindinis tokio ERP sistemų funkcionalumo specifikacijų ir vartotojų reikalavimų modelio tikslas yra tas, kad pagal keliamus reikalavimus galima būtų atsirinkti tinkamiausią sistemą, kurios pagalba vėliau būtų greičiau bei optimaliau reguliuoti įmonės veiklą bei resursus. Tačiau prieš atsirenkant tinkamą sistemą, vartotojai susiduria su problemomis, kurių siekiama išvengti atliekant grupę analizių, nagrinėjant tokius klausimus:

- ar tokios sistemos pasirinkimas pagerins iki tol įmonėje vykdytas procedūras?
- ar pasirinkta sistema bus tokia lanksti ir patogi naudojimui, kaip tikimasi?
- ar ji sustiprins esamas silpnąsias veiklos grandis?
- ar pilnai pakeis anksčiau naudotą programinę įrangą?
- ko siekiama – pakeisti visą iki tol naudotą neefektyvią programinę įrangą ar tik porą netinkamų jos modulių?
- ar pasirinktą produktą bus galima priderinti prie kitų naudojamų programinės įrangos sistemų?

Tai būtina išnagrinėti siekiant apsisaugoti nuo netinkamo pasirinkimo, kuris vėliau gali pareikalauti daug išlaidų, o naudos atnešti mažai: sistema gali būti skirta kitoms įmonių resursų valdymo sferoms nei kad reikalinga konkrečiai įmonei.

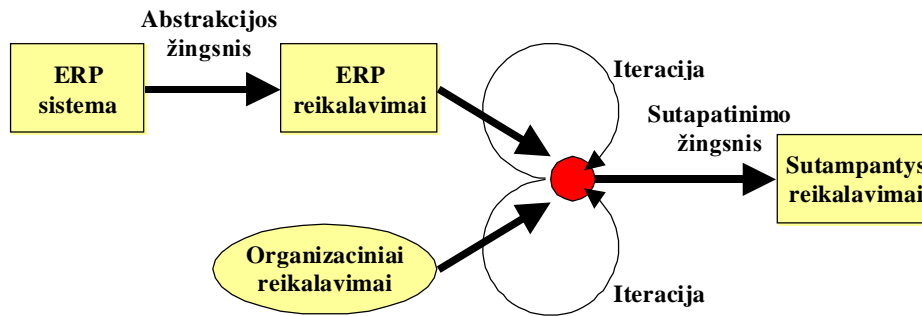
Naudojimas sistema turi būti patogus (taip sudarant sąlygas efektyvesniam darbui), lengviau suprantamas bei greičiau sprendžiamos ankstesnės problemos. Pats apmokymas naudojimosi programiniu paketu turėtų būti nesudėtingas – dažnai vartotojai vengia piniginių bei laiko išlaidų šiam procesui.

1.4. Reikalavimų nesutapimo problemos sprendimo metodų analizė

Nemažai autorių savo literatūroje atkreipė dėmesį į ERP sistemų funkcijų ir vartotojų reikalavimų nesutapimą, tačiau retas iš jų siūlo kokių nors metodinių priemonių šiai problemai spręsti.

Toliausiai šioje srityje pažengė C.Rolland ir N.Prakash. Šių autorių siūlomas sprendimas [1, 2] pasižymi tuo, kad sistemų diegimas būtų orientuotas į vartotojų keliamų reikalavimų, o ne į sistemų funkcinę sferą, kaip tai daro ERP ekspertai. Kad tai būtų galima realizuoti, siūloma sukurti ERP pritaikymo modelį, kurį sudarytų egzistuojančių ERP sistemų funkcinių galimybių bei vartotojų reikalavimų specifikacijos. Šiame modelyje turėtų būti įdiegtas toks funkcionalumas:

- kuriami asociatyviniai ryšiai tarp ERP programinės įrangos sisteminių funkcijų bei keliamų reikalavimų;
- atliekamas ERP sistemos ir vartotojo reikalavimų sutapatinimas;
- automatizuojamas perėjimas nuo vartotojo poreikių prie ERP funkcionalumo savybių, naudojant sudarytus “reikalavimų-funkcionalumo” ryšius.



1.2 pav. C.Rolland ir N.Prakash siūlomas ERP reikalavimų pritaikymo modelis [1, 2]

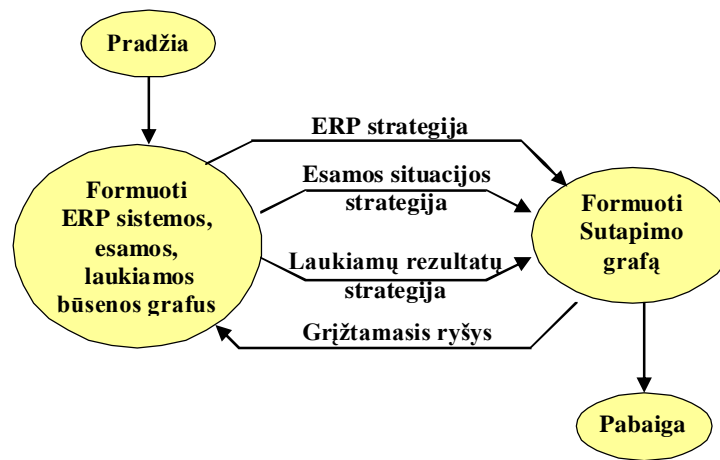
1.2 paveiksle pavaizduotas ERP reikalavimų pritaikymo modelis. Pirmuoju žingsniu atliekamas sistemų funkcionalumo konkretizavimas specifikuojant reikalavimus, kurie realizuoti sistemoje. Antrasis žingsnis - tai sutapatinimo procesas, kuriuo bandoma išskirti ir parinkti reikalavimus, sutampančius tiek iš ERP, tiek iš organizacijos pusės. Sutapatinimo žingsnis yra iteracinis – kiekvieno iteracinio veiksmo metu iš pateiktų reikalavimų aibių yra naudojami tik tie duomenys, kurie dar liko netyrinėti ar nesutapatinti. Šio proceso pabaiga bus laukiamas rezultatas, kai pateikiama reikalavimų aibė, kurių pagrindu bus diegiama ERP sistema.

Surinkti reikalavimai yra išreiškiami *Sutapimo grafu*. Šis grafas sudaromas sugretinant ERP sistemos ir organizacijos keliamus specifinius reikalavimus. Siekiant palengvinti sugretinimo procesą, abiejų pusių reikalavimus siūloma išreikšti grafiškai, vaizduojant vienodomis notacijomis.

ERP reikalavimų sutapatinimo proceso modelis pateiktas 1.3 pav. Jį sudaro keturios atskiros dalys, padalintos į dvi sritis. Pradinėje srityje yra talpinami trys surinktų reikalavimų grafai: ERP sistemos, esamos ir siekiamos situacijos organizacijoje grafai:

- esamos situacijos organizacijoje grafas parodo šiuo metu naudojamomis technologijomis pasiektus rezultatus;
- siekiamos situacijos organizacijoje grafas parodo kokių rezultatų organizacija tikisi įdiegus pasirinktą ERP sistemą;
- ERP sistemos grafas parodo sistemos funkcijas, pačios sistemos keliamus reikalavimus.

Rezultatų srityje talpinamas vienas – sutapimo grafas.

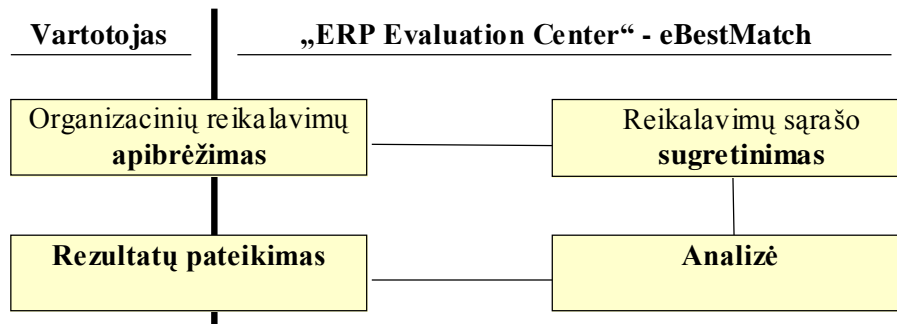


1.3 pav. ERP reikalavimų sutapatinimo proceso modelis

Šie grafai aprašomi trimis žingsniais. Pirmo etapo metu yra sudaromi visi trys pradinės srities grafai. Grafų sudarymas yra pagrįstas kūrimo ir tikrinimo taisyklėmis. Antrojo, sutapatinimo etapo metu sudaromas sutapimo grafas. Pagrindiniai reikalavimai atvaizduoti sutapimo grafą yra išgauti sugretinant reikalavimus, paimtus iš ERP sistemų ir siekiamos situacijos organizacijoje grafų. Kiti, specifiniai įmonės reikalavimai gali būti neišgauti iš ERP sistemos grafo, taigi jie reikalauja specialaus vystymo. Tokiu būdu sutapimo grafas leidžia identifikuoti reikalavimus, dėl kurių organizacijoms gali tekti papildomai rūpintis, nes tai nebus ERP sritis. Sutapimo grafas gali būti sudaromas viena iš trijų strategijų: ERP, esamų ar laukiamų rezultatų įmonėje strategija. Kiekvienos iš šių strategijų tikslas yra nueiti nuo pradinio iki pabaigos taško, atrinkti sutampančius reikalavimus ir patalpinti juos į sutapimo grafą. Trečiuoju ir paskutiniu žingsniu yra atliekamas sutikrinimas, ar sutapimo grafą patalpinti visi reikalavimai, išskirti *Laukiamų rezultatų grafą*. Taigi, taip užbaigiant šį procesą yra užtikrinamas grafo tikslumas.

Štai taip C.Rolland ir N.Prakash siūlo įveikti šią nesuderinamumo problemą, pasirinkdami reikalavimų *sutapatinimo* strategiją. Tokiai strategijai pritaikytas metodas patogus tuo, kad vartotojas gali matyti jam rūpimas veiklos sritis, kurios yra tenkinamos, taip pat išskiriant ir netenkinamas sferas. Tačiau tokią metodiką gali būti pakankamai sunku įsisavinti, be to, pasiruošimo procesas reikalauja ypatingo pasiruošimo bei kruopštumo sudarinėjant reikalavimų grafus, siekiant, kad rezultatas būtų kuo tikslesnis.

Visiškai kitokią – sugretinimo – strategiją pasirinko TechnologyEvaluation.Com grupė (TEC Group), pasiūliusi veikiančią sistemą „ERP Evaluation Center“ [3]. Tai „keturių žingsnių“ sistema (1.4 pav.) – reikalavimų apibrėžimas, sugretinimas, analizė, rezultatų pateikimas.



1.4 pav. „ERP Evaluation Center“ modelio reikalavimų sugretinimo etapai; vartotojo ir sistemos veikimo sritys

Tinkamo įrankio parinktis pradedama organizacijos reikalavimų apibrėžimu. Šiame žingsnyje vartotojui pateikiamas klausimynas su atsakymų variantais. Klausimai sudaryti pagal funkcinis ir techninius kriterijus. Vėlesnio etapo metu atliekamas vartotojo apibrėžtų ir duomenų bazėje saugomų ERP sistemų funkcijų sugretinimas. Šį procesą atlieka TEC grupės ištobulintas sugretinimo variklis „eBestMatch“ [4]. Šis modulis yra naudojamas visose TEC grupės teikiamose sistemose (ERP, CRM, HR ir kitokio pobūdžio programinių paketų). Sugretinimo etapą seka analizė. Čia, pagal iš anksto nustatytas reikšmes, sistema įvertina kiekvieną iš ERP sistemų, fiksuojant jų stipriąsias ir silpnąsias puses kiekvienam pasirinktam kriterijui. Apdorojus duomenis, pateikiamas vartotojų reikalavimus dalinai ar pilnai tenkinančių sistemų sąrašas su analizės metu apskaičiuotomis reikšmėmis. Pasirinkus keletą iš jų yra detalizuojamas kiekvienos sistemos atitikimas pateiktiems reikalavimams.

Sugretinimo strategijos principais veikiančios sistemos turi nemažai privalumų: paprasčiau naudoti, reikalauja mažiau laiko pasiruošimui ir pačiam reikalavimų sugretinimo procesui, nereikia, specifikavimo bei modeliavimo žinių, mažesnės išlaidos. Tačiau rezultatai gali neatspindėti visų vartotojų poreikių – vartotojas yra priverstas tenkintis tik iš anksto apibrėžtais, informacinėje sistemoje išsaugotais ERP sistemų funkcionalumo aprašais. Pateikti rezultatai (ERP sistemos) neatspindės trūkumų – reikalavimų, kurių jos netenkina.

Panašaus į šį darbą projekto ėmėsi ispanų programuotojų grupė. Jų projektas pavadintas AMOS [5], kurio rezultate buvo sukurtas įrankis, leidžiantis suskirstyti į kategorijas, saugoti ir atlikti paieškas tarp atvirojo kodo programinių paketų, programinio kodo ištraukų ar programų testavimo įrankių. Šie paketai yra skirstomi pagal jų reikšminius aprašus. Tai padėtų žmonėms, kuriantiems atvirojo kodo programas, greičiau ir efektyviau išnaudoti produktų kūrimo grafikus, panaudojant anksčiau sukurtą kodo dalis.

Projekto kūrimo metu buvo išanalizuoti galimi programinio kodo paieškos būdai [6], tai:

- Formalūs metodai. Šis būdas gražina tikslus rezultatus, tačiau gali operuoti tik su mažais kodo fragmentais. Reikalingi formalūs aprašai.

- „Dirbtinis intelektas“ yra beveik automatizuotas, naudoja komentarus, funkcijų pavadinimus, tačiau turi ir savų minusų – kartais sunku išgauti suprantamus terminus, sudėtingas pritaikymas kelių kalbų vartojimui, paieškos tikslumas bei rezultatų pateikimas pagal užklausų naudojimo dažnumą nėra labai tikslus.

- Pilno perrinkimo (angl. *brute force*) būdas. Paprasta realizuoti, greita paieška, reikalauja riboto vartotojo įsikišimo. Negali operuoti paketo informacija, painiojasi su sinonimais, dažnai pateikia neteisingus rezultatus.

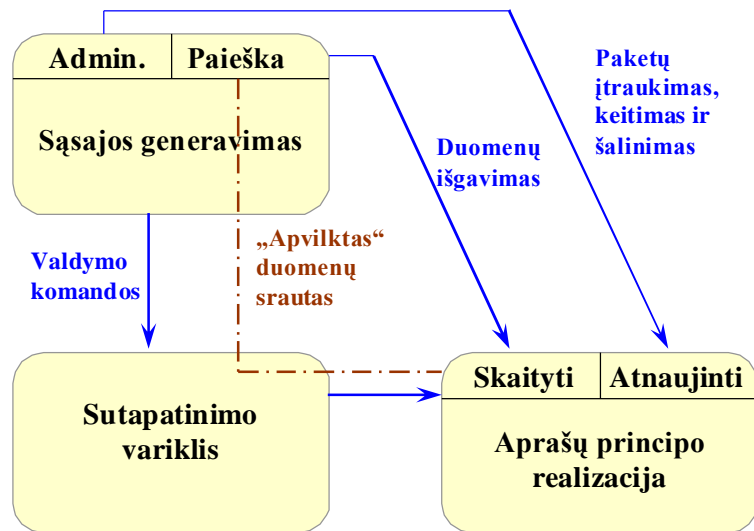
Kūrėjai pasirinko kitą būdą, tuo pačiu suskirstydami projektą į kelias dalis:

- Pasirenkamas programinio kodo aprašymo principas.
- Sukuriamas programinio kodo ir jo ištraukų aprašymo terminų žodynas.
- Realizuojamas paieškos ir sutapatinimo variklis.
- Realizuojama patogi ir stabiliai veikianti duomenų bazė, gebanti saugoti kelis šimtus kodo aprašų bei veikianti kartu su paieškos varikliu.
- Duomenų bazės užpildymas informacija.

Tačiau projektuotojų grupė nurodė ir savo pasirinkto būdo nepatogumus – tai daug laiko užimantis bei sudėtingas neautomatizuotas paketų aprašų ir žodyno sudarymas, visi terminai turi būti įtraukti į žodyną – priešingu atveju jie nebus rasti. Beto šis paieškos mechanizmas yra specializuotas – jis netiktų bendram naudojimui t.y. bendro pobūdžio informacijos paieškai.

Sistemą [7] sudaro trys stambios dalys – vartotojo sąsaja, paieškos variklis ir duomenų bazė.

1.5 paveiksle parodyti ryšiai tarp vartotojo sąsajos, paieškos (sutapatinimo) variklio ir paketų aprašus saugančios duomenų bazės. Rodyklės išreiškia vartotojo sąsajos panaudojimą, priklausomai nuo atliekamų funkcijų.



1.5 pav. AMOS sistemos schema [7]: vartotojo sąsaja, paieškos variklis ir duomenų bazė

Visa ši schema sudaro bendrą *užklausių sąsają*, kuri realizuoja metodus atliekančius tokius veiksmus:

1. Skaito duomenų bazėje esančią informaciją (operacija atliekama per sutapatinimo variklį, paieškos sąsają ir administratoriaus sąsają).
2. Atnaujina duomenų bazėje esančią informaciją (operaciją atliekama per administratoriaus sąsają).
3. Valdo paieškos procesą (operacija atliekama per paieškos sąsają).

1.6 paveiksle pateikta paieškos variklį sudarančio algoritmo schema [8].

```

Pradiniai duomenys:  $R$  --- reikalavimų aibė
Rezultatai:  $(P, R)$  --- rastų paketų ir neišpildytų reikalavimų aibės

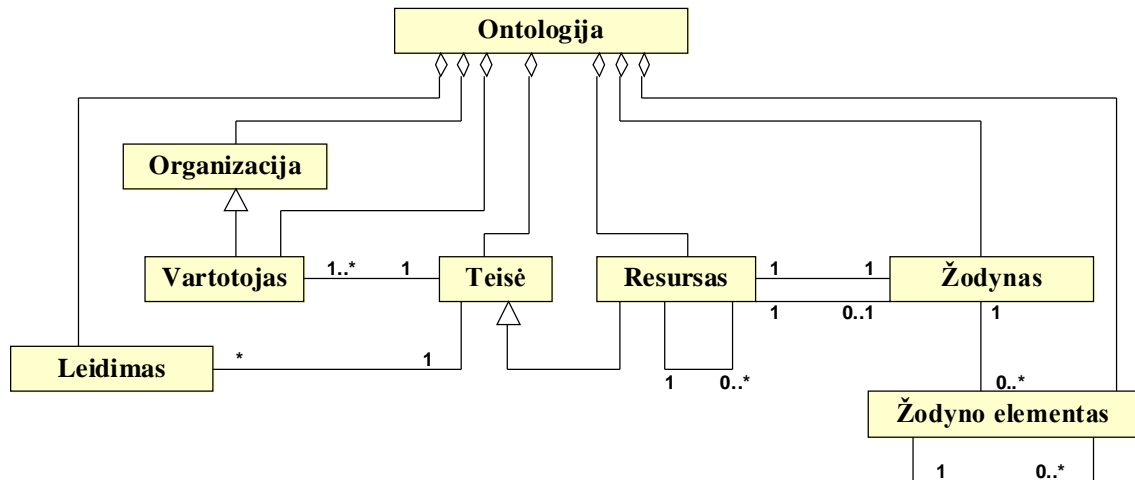
 $F := 0$           --- Tai, kas jau buvo atlikta
 $P := 0$           --- Jau panaudoti paketai
do
  select  $A \in \{A_q^p \mid q \cap R \neq 0\}$ 
   $F := F \cup q$ 
   $R := (R - q) \cup (p - F)$ 
   $P := P \cup A$ 
until <nėra  $A_q^p$  galinčių pakeisti  $R$ >
return  $(P, R)$ 
-----
PASTABA:  $A^p$  išreiškia paketui  $A$  reikiamus paketus, kurių galimybes
žymi  $p$ .  $A_q$  išreiškia paketo  $A$  teikiamas galimybes  $q$ , kurių reikia
kitiems paketams.

```

1.6 pav. Schematiškas paieškos algoritmas [8]

Raktažodis *select* išreiškia paketų išrinkimo neapibrėžtumą: kiekvieno ciklo metu gali būti keli variantai. Kol R aibėje esančius reikalavimus tenkina konkretus duomenų bazėje saugomas paketas A, tol A yra parenkamas. Netenkinami reikalavimai yra gražinami į aibę R kitam paieškos ciklui. Išrinktų paketų informacija yra talpinami į aibę F, kuri saugo rezultatus bei nutraukia paiešką. Procesas gali baigtis ir neišpildžius visų iškeltų reikalavimų.

AMOS sistemos projektuotojai taip pat parodė duomenų aprašų principo (ontologijos) schemą [7].



1.7 pav. Paketų aprašų principo struktūra atvaizduota UML principu [7]

Taigi, AMOS projektas yra analogiškas šiam darbui, išskyrus tai, kad yra skirtinga darbo sritis (nagrinėjami atvirojo kodo paketai, svarstomos jų aprašymo galimybės) ir specializuota į efektyvesnę ir greitesnę atvirojo kodo programų kūrimą. Šios projektuotojų grupės sukurtas paieškos variklis veikia sutapatavimo principu, kai atsizvelgiama ne tik į pateiktus reikalavimus, bet ir ryšius tarp duomenų bazėje saugomų paketų aprašų.

1.5. Darbo tikslas ir siekiami privalumai

Šiuo darbu siekiama sukurti programinį modelį, kuris galėtų palengvinti tinkamos ERP sistemos pasirinkimą. Tikslas – patenkinti vartotojus, ieškančius jų organizacijų veiklą tiksliausiai atspindinčias sistemas. Taip jų paieškoms būtų sutaupoma laiko bei išteklių. Sukurta programa pagal vartotojų pateiktus reikalavimus atrinktų tinkamiausias sistemas, taip sumažinant bereikalingos analizės mastus.

Vartotojas, siekiantis surasti tinkamą sistemą, turi būti susipažinęs su jų funkcionalumu, taip pat žinoti OPM metodologiją. Jam nereikia rūpintis konkrečių įmonių išteklių valdymo sistemų paieška bei atskirai nagrinėti jų galimybes. Tai atliekama sudarant keliamų reikalavimų diagramą. Ši diagrama turi būti sudaroma pagal bendrą ERP sistemos veikimo schemą (vartotojas turi žinoti bent vieną sistemą bei suprasti jos veikimą). Sudarius diagramą, naudojantis OPM metodologija, jos aprašas OPL

kalba turi būti perduotas sutapatinimo programai, kuri atlieka kelias roles: suvestos informacijos apdorojimą, saugomų ir suvestų duomenų sutapatinimą, rezultatų formavimą ir pateikimą.

1.5.1. Siekiamo modelio apibrėžimas

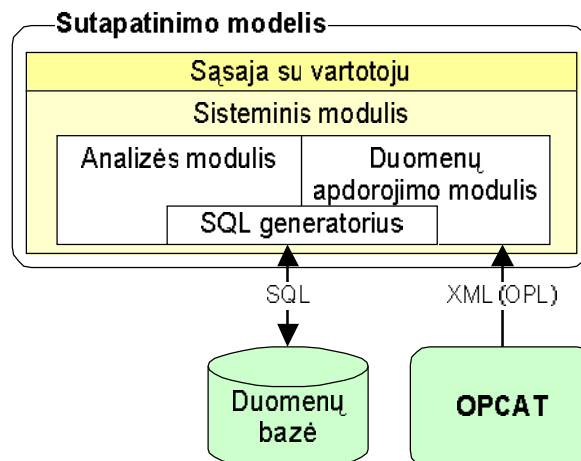
Kuriamą sistemą galima sudaryti iš tokių apibendrintų modulių (1.8 pav.):

- **OPCAT programinės įrangos projektavimo priemonė** – įrankis, palaikantis objektų procesų metodologiją (OPM). Jos apibrėžtomis taisyklėmis galima aprašyti reikalavimus. Reikalavimai gali būti aprašyti grafine forma (objektų-procesų diagramomis (OPD)) bei tekstiniu formatu (objektų-procesų kalba (OPL)). Šis įrankis leidžia išsaugoti diagramas XML formatu. Tuo pasinaudojant galima kurti ERP sistemos funkcionalumo bei organizacinių reikalavimų diagramas bei jas išsaugoti tolesniam apdorojimui.

- **šąsaja su vartotoju** – formos, kurių pagalba bendraujama su vartotoju. Čia atliekamas OPCAT įrankiu surinktų duomenų nuskaitymas bei rezultatų pateikimas vykdant tinkamos ERP sistemos parinktį. Rezultatai gali būti įvertinti nurodant procentinius įverčius, pagal tai, kiek sistema atlieka pageidaujamų funkcijų, o taip pat galima nurodyti kokybinius įverčius, tuo atveju vartotojas turėtų iš anksto nustatyti svarbumo faktorius konkrečioms ERP funkcionalumo aspektams. Vartotojas taip pat gali nustatyti informacijos apdorojimo tikslumą. Taip būtų galima išvengti netikslumų, kuriuos vartotojas galėjo padaryti reikalavimų diagramos sudarymo metu.

- **sisteminis modulis** apjungia dvi dalis: duomenų apdorojimą, kuris naudojamas nuskaitymą informaciją ir užpildant duomenų saugyklą, bei analizatorius, kuriame pagal pateiktus poreikių kriterijus yra atrenkamos tinkamiausios sistemos, o gauti rezultatai per šąsają grąžinami vartotojui. Duomenų apdorojimo ir analizės dalys naudoja duomenų bazėje saugomus duomenis siunčiant užklausas, kurias formuoja SQL užklausų generatorius.

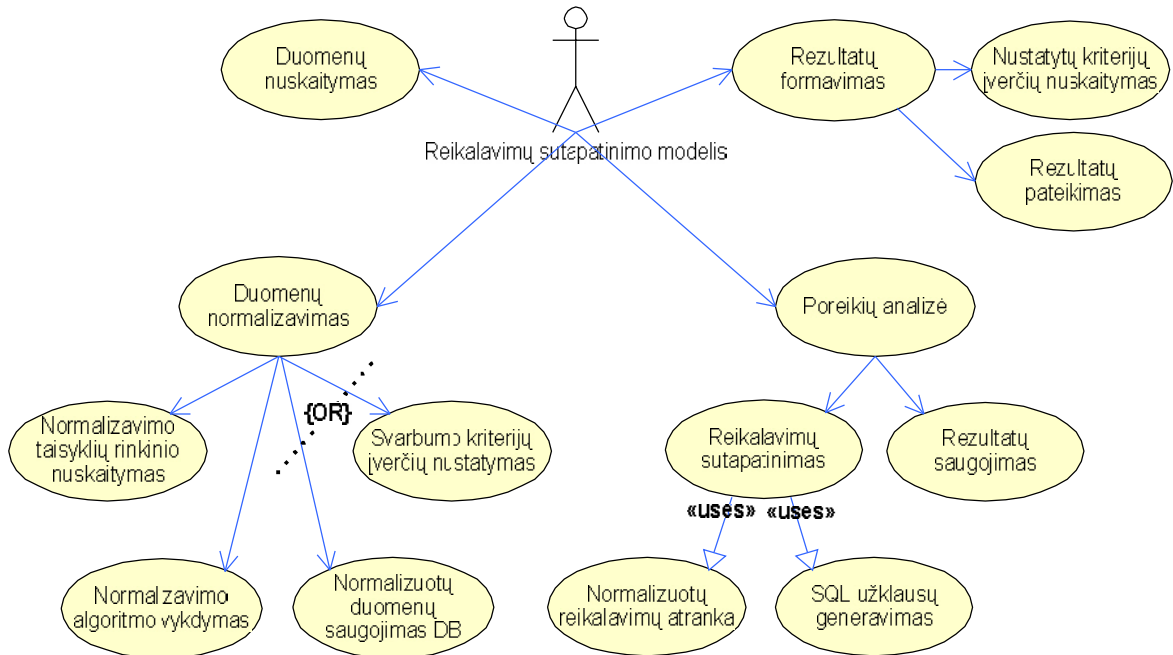
- **duomenų saugykloje** kaupiama informacija apie žinomas ERP sistemas, suskirsčius ją pagal sistemų teikiamas funkcijas.



1.8 pav. Planuojamos sistemos architektūra

1.5.2. Reikalavimų sutapatinimo sistemos kompiuterizuojamos funkcijos

Pasirinktos kompiuterizuoti sistemos numatytos atlikti funkcijos yra pavaizduotos 1.9 paveiksle:



1.9 pav. Planuojamos sistemos funkcijos

Ketrios bendrinės funkcijos yra būtinos tokio pobūdžio sistemai kurti – duomenų nuskaitymas, normalizavimas, reikalavimų analizė bei rezultatų formavimas. Pirmoji funkcija skirta duomenų įkrovimui į sistemą, antroji – nuskaitytų duomenų pritaikymas saugojimui duomenų bazėje arba sutapatinimo procesui. Čia, naudojantis iš anksto sudarytu normalizavimo taisyklių rinkiniu bei normalizavimo algoritmu, atliekamas duomenų supaprastinimas siekiant išvengti galimų nesklandumų atsiradusių dėl klaidingos rašybos ar papildomų duomenų naudojimo. Normalizuotus duomenis siūloma išsaugoti duomenų bazėje arba „paruošti“ sutapatinimo procesui. Pirmas veiksmas yra skirtas ERP sistemų funkcionalumo aprašų saugojimui duomenų bazėje, kai tuo tarpu antruoju veiksmu yra įvertinami organizacijos reikalavimų kriterijai.

Normalizuoti duomenys perduodami poreikių analizei, kur sutapatinimo algoritmo metu atrinkti ir iš duomenų bazės išgauti duomenys yra sulyginami. Gautas rezultatas yra saugomas objektų pavidalų vidinėje programos struktūroje. Baigus sutapatinimo procesą informacija pateikiama vartotojui suprantama forma. Rezultatai įvertinami procentais arba pagal anksčiau nustatytas reikšmes.

1.6. Kuriamo modelio reikalavimų specifikacija

Norint, kad kuriama programa pilnai atliktų nustatytus uždavinius, jos veikimas būtų stabilus, konkrečiose situacijose įvykdytų specialius veiksmus, reikia sudaryti sistemai keliamų reikalavimų sąrašą. Iš anksto numčius sistemos elgseną ypatingų situacijų atvejais yra didinamas kuriamo produkto stabilumo lygis. Tai taip pat leidžia sutaupyti laikinių bei piniginių išlaidų, galinčių atsirasti siekiant patobulinti produktą, kai šis jau yra realizuotas bei problema išaiškėja vėliau.

1.6.1. Funkciniai reikalavimai

Reikalavimai keliami asmeniniam kompiuteriui:

Asmeninio kompiuterio techniniai duomenys priklausys nuo duomenų apimties, vartojamos papildomos operatyvinės bei techninės įrangos.

Sistemoje naudojama duomenų bazė saugos ERP sistemų funkcionalumo aprašus.

Tiksliai nustatyti reikalingos kietojo disko vietos kiekį informacijos saugojimui negalima. Sistemos projektavimo bei kūrimo metu bus naudojama kompiuteriu, kurio techninės charakteristikos yra tokios:

- procesoriaus sparta ne mažesnė nei 800 MHz
- kietasis diskas ne mažesnis nei 1 GB
- operatyvinė atmintis ne mažesnė nei 128 MB

Bus siekiama, kad sukurtas produktas pilnai funkcionuotų esant tokiai techninei konfigūracijai.

Numatoma, kad sistemos kūrimui, testavimams bei nuolatiniam veikimui asmeniniame kompiuteryje turės būti įdiegta tokia programinė įranga:

- Windows operacinė sistema (Windows 2000, Windows XP arba Windows NT)
- Apache HTTP serveris
- MySQL serveris – sistema, valdanti duomenų bazines
- Java programavimo kalbos kompiliatorius (Java 2 SDK)
- JDBC tvarkyklė, kurios pagalba susiejama MySQL duomenų bazė ir Java programa.

Reikalavimai vartotojo sąsajai:

Informacinės sistemos sąsaja turi būti aiški bei patogi naudoti. Komponentai išdėstyti pagal galiojančias vartotojo sąsajos kūrimo rekomendacijas ir standartus. Realizuoti įvedamų duomenų kontrolę, kad sumažinti galimų klaidų skaičių.

Vartotojo sąsajos kalba – pasirinktinai: anglų arba lietuvių.

1.6.2. Nefunkciniai reikalavimai

Čia pateikiami informacinei sistemai keliami nefunkciniai reikalavimai:

1.1 lentelė.

Nefunkciniai reikalavimai

Reikalavimų įtakos sritis	Reikalavimas	Svarba	Reikalavimo išpildymas
Funkcionalumas	Tinkamumas	↑	Kuriama sistema, sutapatinimo metodas turi būti tinkami iškelto reikalavimų sutapatinimo uždavinio sprendimui ir atitinkamų rezultatų pateikimui.
	Tikslumas	↑	Sistema, atlikusi vartotojo pateiktų duomenų analizę bei sutapatinimo procesą, turi pateikti kuo tikslesnius rezultatus, nes tai įtakoja organizacijos išlaidas (pasirenkant siūlomą ERP sistemą). Pateikus neteisingą variantą, tai gali nulemti tolesnę organizacijos veiklą (žlugimas arba veiklos pristabdymas, dėl netinkamo valdymo per ERP sistemą).
	Saugumas	↓	Sutapatinimo sistema nenaudoja jokios konfidencialios informacijos apie organizaciją – ji veikia tik su iš anksto išsaugotais ir vartotojo pateiktais duomenimis. Jos darbo tikslas – patarti pasirinkti tinkamą ERP sistemą. Sistema pritaikyta veikti viename personaliniame kompiuteryje, tad papildomų apsaugos priemonių nenumatoma skirti.
Patikimumas	Užbaigtumas	↑	Sistema turi būti išbaigta, stengiantis numatyti visus galimus klaidos atvejus.
	Tolerancija klaidoms	↑	Sistemos veikimo metu susidarius klaidai vartotojas turi būti informuojamas apie ją, o pati sistema – toliau veikti, neužstrigti.

1.1 lentelė (tęsinys).

Nefunkciniai reikalavimai

Patogumas			Vartotojas turi mokėti sudaryti reikalavimų diagramas bei suprasti sudaromą funkcionalumo schemą iš ERP sistemų pusės. Sudarius reikalavimų schemą, vartotojui papildomų žinių nereikia – tolesnius veiksmus už jį atliks sistema, pabaigoje pateikiant rezultatus. Rezultatai turi būti pateikiami kuo aiškesne forma, kad nesukeltų papildomų pastangų juos aiškinantis ir vertinant.
Suprantamumas		↑	
Efektyvumas			Tinkamos sistemos pasirinkimo procesas turi užimti kuo mažiau laiko. Pagrindinius laiko išteklius koncentruojant į reikalavimų schemas sudarymą (tai priklauso nuo vartotojo žinių). Pati sistema duomenų apdorojimo, sutapatavimo ir rezultatų pateikimo procesus turi atlikti per kuo trumpesnę laiką. Tai turi būti apspręsta per optimalaus algoritmo sudarymą.
			Sutapatavimo modelio procesorius neturi išnaudoti sistemos operatyvinės atminties, taip būtų sudaroma aplinkybė programą naudoti ir paprastuose asmeniniuose kompiuteriuose, ne tik serveriuose. Žinant platų ERP sistemų funkcionalumą reikia sudaryti tinkamas sąlygas jų aprašams saugoti – iš anksto išskirti kietojo disko vietas.
Išteklių naudojimas			
Priežiūros savybės	Analizės savybės (identifikuojant klaidas)	↑	Turi būti sudaryta pagalba vartotojui, klaidos atveju, kad būtų galima atsekti klaidos priežastį ir galimą jos sprendimo būdą.
	Praplečiamumas		Programa turi būti kuriama laikantis objektiškai orientuoto programavimo, taip paliekant galimybę jos praplėtimui.

1.6.3. Rizikos faktorių analizė

Siekiant apsaugoti sistemą nuo galimos nestabilumo ar pažeidžiamumo rizikos, reikia įvertinti tai galinčius iššaukti faktorius. Dalis rizikos faktorių gali atsirasti dėl pasirinktų projekto kūrimo technologijų ar metodologijų, taip pat dėl kompiuteriuose esamų sistemų konfigūracijos ar galimų vartotojų klaidų. Remiantis iš anksto apibrėžtomis rizikomis, dalies jų galima išvengti dar sistemos kūrimo metu.

1.2 lentelė.

Rizikos faktoriai ir jų pašalinimo būdai

Rizikos faktorius	Rizikos faktoriaus eliminavimo būdas
Kintantys reikalavimai	Reikalavimai turi būti tikslinami darbo eigoje.
Kintantys ryšiai su kitomis sistemomis	Numatyti atsarginį atvejį problemų atveju.
Diegimo problemos	Išbandyti sistemos diegimą įvairių konfigūracijų kompiuteriuose, numatyti galimas problemas.
Nekorektiški duomenys	Vartotojui, sudarančiam reikalavimų schemą, pateikti reikalavimų, kaip sąvokų, žodyną.
Sistema neveiks	Numatyti kritinius taškus.
Išaugus sistemos kaina ir jos kūrimo laikas	Numatyti mažesnę sistemos variantą, tenkinantį iškeltą projekto tikslą.

1.6.4. Rezultato kokybės kriterijai

Sukūrus pagalbinę sistemą, padedančią vartotojui pasirinkti tinkamą ERP produktą, bei išpildžius numatytus funkcinius ir nefuncinius reikalavimus, gautas rezultatas turi būti įvertintas kokybės kriterijais:

1.3 lentelė.

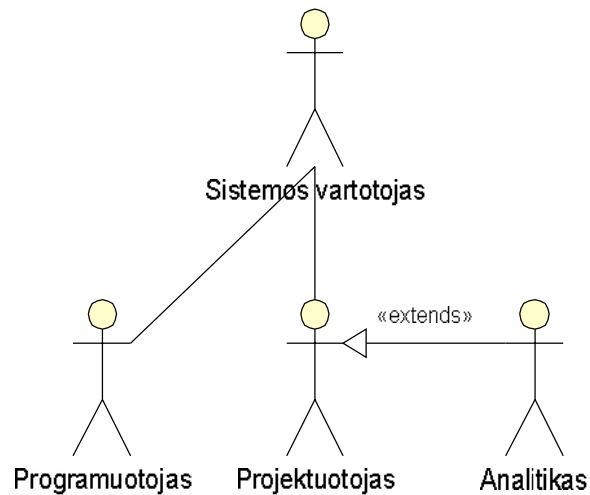
Sistemos kokybės kriterijai ir vertinimo faktoriai

Kokybės kriterijus	Vertinimo faktoriai
Patikimumas	Stabilus sistemos veikimas. Mažesnių nei numatyta techninių resursų poveikis sistemos darbui. Sistemos reakcija į vartotojo duomenų ar netinkamo sistemos naudojimo klaidas.
Efektyvumas	Sistemos atliekamos vykdomos analizės optimalumas (pagal rezultatų kokybę ir jų pateikimo greitį). Tinkamas pagrindinės sistemos resursų naudojimas.
Išsamumas	Visų užsakovo ir vartotojų keliamų reikalavimų išpildymas.

Įvertinus sistemą šiais kriterijais, bus galima prieiti prie išvadų, ar sukurtoji sistema pasiteisino, kiek resursų pareikalavo jos kūrimas bei kiek gali pareikalausti jos praplečiamumas.

1.6.5. Sistemos vartotojų panaudojimo atvejai

Kaip ir minėta ankstesniuose skyriuose, numatoma, kad būsima sistema naudosis trijų tipų vartotojai – tai programuotojai, projektuotojai bei analitikai.

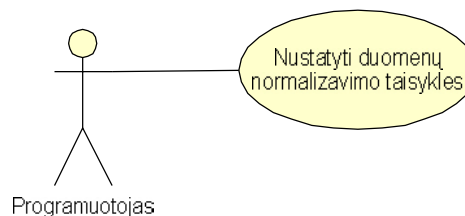


1.10 pav. Informacinės sistemos vartotojų grupės

Sistemos projektuotojai naudodamiesi sistemos pateiktu grafiniu projektavimo įrankiu sudaro esamos ERP sistemos funkcionalumo schemą. Analitikai – asmenys, ieškantys tiksliausiai jų atstovaujama organizaciją tenkinančios sistemos – pilnai padengia projektuotojų atliekamus veiksmus prijungiant rezultatų apdorojimo funkcijas. Trečioji šios informacinės sistemos vartotojų grupė yra programuotojai. Šios grupės būtų galima neminėti, tačiau ją sudarančių vartotojų atliekam veiksmams gali padidinti sistemos skaičiavimų tikslumą.

1.6.5.1. Vartotojo – programuotojo panaudojimo atvejai

Programuotojo panaudojimo atvejai pateikti žemiau esančioje schemeje:

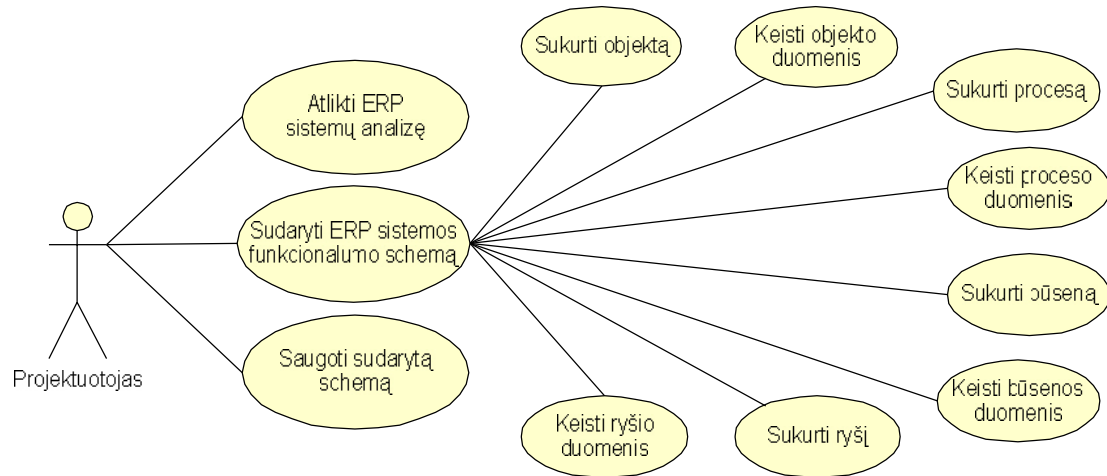


1.11 pav. Vartotojų grupės *Programuotojas* panaudojimo atvejų schema

Programuotojas atlieka tik vieną funkciją – jis gali sudaryti duomenų apdorojimo taisyklių rinkinį. Ši funkcija sistemoje nėra kompiuterizuota – numatoma, kad vartotojas, priklausantis šiai grupei, turi galimybę sudaryti tam tikras taisykles, pagal kurias bus apdorojami duomenys, t.y. specialiųjų simbolių išmetimas ir pan. Taisyklės yra kuriamos pagal sudarytą šabloną, saugant taisyklių duomenis XML formatu.

1.6.5.2. Vartotojo – projektuotojo panaudojimo atvejai

Čia pateikiama vartotojų priklausančių projektuotojų grupei panaudojimo atvejų schema:

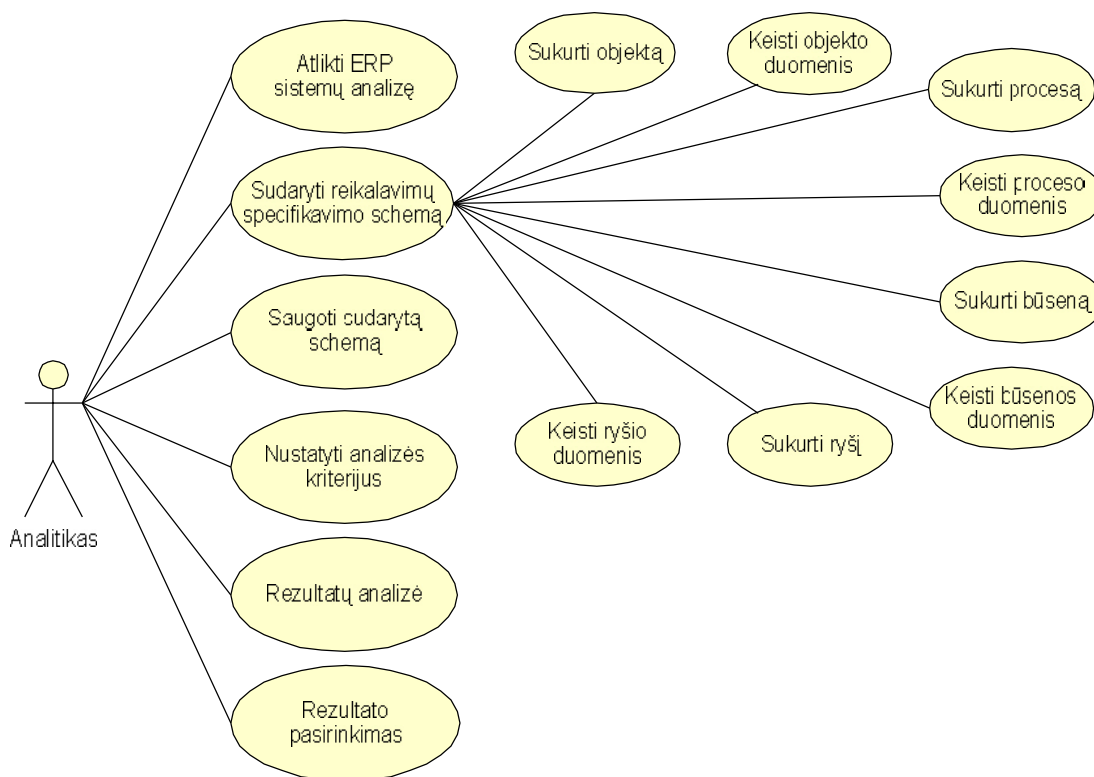


1.12 pav. Vartotojų grupės *Projektuotojas* panaudojimo atvejų schema

Iš pateiktų panaudojimo atvejų kompiuterizuojamos yra tik dvi funkcijos – ERP sistemos funkcionalumo schemas sudarymas bei tos schemas saugojimas duomenų saugykloje. ERP sistemos analizę vartotojas atlieka savarankiškai, tai yra būtina sąlyga, norint tinkamai sudaryti jos funkcionalumo schemą. Vėliau vartotojas naudodamasis sistemos teikiamu grafiniu projektavimo įrankiu sudaro šią schemą. Ją išsaugo bei pritaikius anksčiau sukurtas duomenų apdorojimo taisykles duomenis pateikia sistemos duomenų apdorojimo moduliui. Gauti programiniai objektai yra išsaugomi duomenų bazėje išlaikant sudarytą informacijos hierarchiją.

1.6.5.3. Vartotojo – analitiko panaudojimo atvejai

Kadangi ši sistema yra orientuota į analitikus, tad jie ir atlieka daugiausia funkcijų:



1.13 pav. Vartotojų grupės *Analitikas* panaudojimo atvejų schema

Kaip ir minėta anksčiau, analitikai pilnai padengia projektuotojų grupei priskiriamų vartotojų funkcijas – savarankiškai atlieka ERP sistemų analizę, sudaro norimos ERP sistemos funkcionalumo schemą bei ją išsaugo duomenų bazėje. Pažymėtina tai, kad duomenys duomenų bazėje yra saugomi laikinai, t.y. tik naudojimui analizės metu, vėliau sistema juos pašalina. Tačiau sistema palieka galimybę vartotojui išlaikyti duomenis ir neišnaikinti atlikus darbą. Prieš pradėdant darbą vartotojas gali nustatyti svarbumo įverčius, t.y. pažymėti daugiau ar mažiau svarbias ieškomos ERP sistemos sritis. Pagal šiuos nustatymus reikalavimų sutapatinimo sistema apskaičiuos duomenų bazėje saugomų sistemų atitikimą vartotojo pateiktai schemai. Šis žingsnis gali būti ir praleidžiamas, tuomet sutapatinimo modelis pateiks rezultatus, kuriuose atsispindės kiek vartotojo sudaryta schema atitinka duomenų bazėje saugomus ERP sistemų aprašus. Galiausiai analitikas turės atlikti pateiktų rezultatų analizę ir pasirinkti tinkamiausią sistemą.

Kadangi vartotojas pagrindinę darbo dalį atlieka dirbdamas su grafinių projektavimo įrankiu, todėl čia yra išskirtos vartotojo galimos atlikti funkcijos, sudarant ERP sistemos funkcionalumo schemą. Jos yra: objektų, procesų, būsenų bei ryšių kūrimas/šalinimas, o taip pat ir jau sukurtų duomenų, procesų, būsenų ar ryšių duomenų keitimas.

1.7. Projektavimo metodų, priemonių parinkimas kuriamo modelio realizacijai bei naudojimui

Viena iš sistemos realizavimo priemonių galėtų būti *Objektų-Procesų metodologija* (OPM) [9]. Jos apibrėžtomis taisyklėmis galima aprašyti reikalavimus – ši metodologija naudojama sistemų ir jų specifikacijų apibrėžimams. Tai palanku dėl kūrimo metu vyraujančio požiūrio – kuriama sistema ar jos funkcinis aprašas yra matomas iš sisteminės – funkcijų, struktūros ir elgsenos – pusės. Reikalavimams aprašyti ši modeliavimo kalba tinka dėl savo universalumo ir lankstumo – čia remiamasi dviem pagrindiniais elementais: objektais ir procesais. Reikalavimai gali būti aprašyti grafine forma (objektų-procesų diagramomis (OPD)) bei tekstinio formatu (objektų-procesų kalba (OPL)).

Lyginant su OPM, UML [10], kaip programinių produktų modeliavimo priemonė, yra mažiau priimtina dėl gana sudėtingo modeliavimo – čia projektuojamos sistemos struktūra ir elgsena yra atvaizduojama skirtingais požiūriais, skirtingų tipų diagramomis. Projektavimo metu reikia skirti papildomo atidumo ir pastangų norint išlaikyti vientisumą tarp šių nevienodų tipų diagramų. Tuo tarpu OPM leidžia kurti sistemos struktūros ir elgsenos schemas lygiagrečiai, žvelgiant į viską iš vieno taško. I.Reinhartz-Berger ir D.Dori atliko apklausą [11], siūlydami palyginti UML ir OPM metodologijas. Studentų grupėms pateiktas užduočių ir klausimų sąrašas, kuriuo jie turėjo priimti išvadas, kuri iš šių dviejų metodologijų yra patrauklesnė. Susumavus rezultatus, „laimėtoja“ buvo paskelbta OPM. Dvi pagrindinės priežastys yra tos, kad OPM turi vieno tipo OPD diagramas, kurias lengviau gali suprasti neapmokyti vartotojai, bei tai, kad UML sistemos modelis yra išskaidomas į daugelio požiūrio diagramų – taip sunkiau suprasti projektuojamos sistemos funkcionalumą bei jos vientisumą.

Pasinaudoti OPM metodologijos patogumais leidžia specialiai jai sukurta *CASE* priemonė OPCAT [12]. Šis įrankis leidžia kurti diagramas bei sudaryti tekstinį kuriamo objekto aprašą, taip pat suteikia galimybę sugeneruoti dokumentaciją, programinį kodą (Java kalba), atlikti automatinį aprašų analizę, išsaugoti diagramas bei jų tekstinį aprašą XML formatu. Tuo pasinaudojant galima kurti ERP sistemos funkcionalumo bei organizacinių reikalavimų diagramas bei jas išsaugoti tolesniam apdorojimui.

Kadangi OPCAT yra sukurtas Java kalba, tuo būdu siūloma, kad sisteminis modulis taip pat būtų realizuotas Java kalba. Tai taip pat yra patogu dėl šios programavimo kalbos lankstumo – nedaug pastangų reikalauja perėjimas nuo vieno tipo sisteminės platformos prie kitos; kuriamas produktas gali būti naudojamas tiek internetinių puslapių, tiek paleidžiamojo failo pavidalu.

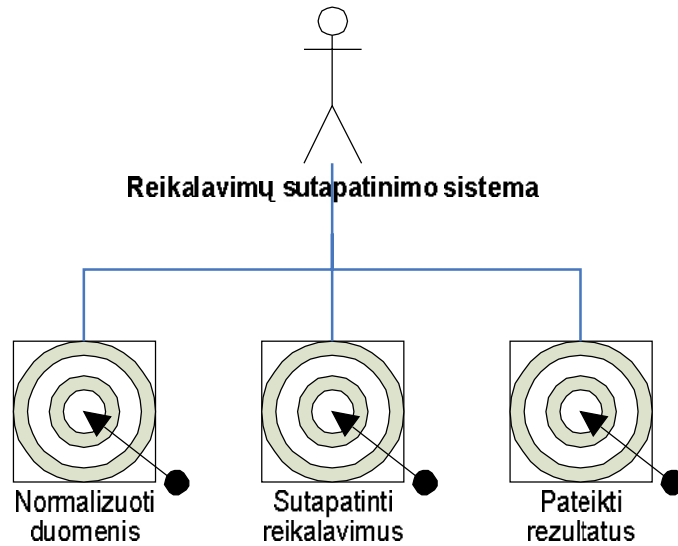
Numatoma, kad be OPM palaikančio įrankio OPCAT, grafinės vartotojo sąsajos, sistemos „variklio“ dar bus naudojama ir saugykla – specializuota duomenų bazė, skirta surinktų reikalavimų

specifikacijai saugoti. Žinant Java galimybes kaip vienas tinkamiausių variantų yra MySQL duomenų bazė.

1.8. Kompiuterizuojamos sistemos varianto parinkimas

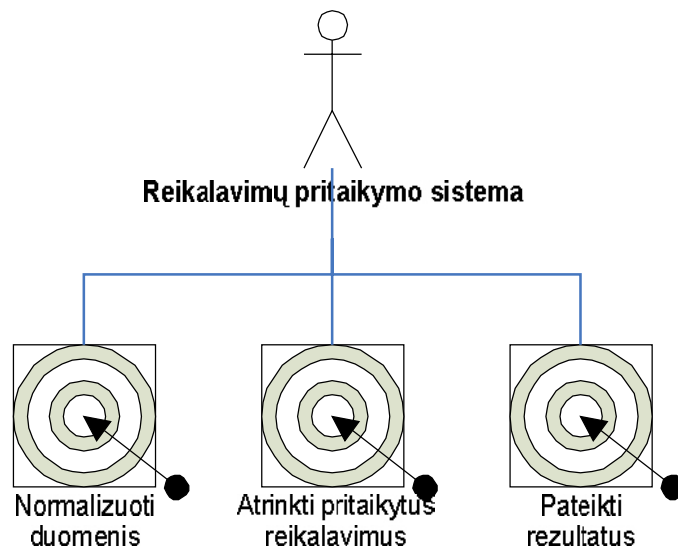
Sistemą galima realizuoti dviem būdais:

a) kuriant reikalavimų sutapatinimo sistemą. Toks programinis produktas turėtų atlikti tris pagrindines užduotis (1.14 pav.): normalizuoti duomenis, juos sulygtinti ir pateikti rezultatus.



1.14 pav. Reikalavimų sutapatinimo sistemos uždaviniai.

b) kuriant reikalavimų pritaikymo sistemą. Tokia programa turėtų gebėti atlikti tokius tris uždavinius (1.15 pav.): normalizuoti duomenis, pateikti vartotojui saugomus duomenis ir pagal atrinktus rezultatus formuoti ataskaitas.



1.15 pav. Reikalavimų pritaikymo sistemos uždaviniai.

Nors antrasis variantas yra tikslesnis rezultatų apdorojimo srityje, tačiau čia neatspindėtų specifiniai vartotojų norai – nesimatyty sistemai keliami reikalavimai, kurie iki šiol nebūtų išsaugoti reikalavimų duomenų bazėje.

Siekiant, kad vartotojui pateikti rezultatai atspindėtų visus – ir padengtus, ir nepadengtus – keltus reikalavimus yra pasirinktas pirmasis variantas – šio darbo rezultate bus sukurta rezultatų sutapatinimo sistema.

Orientuojamasi į pastarosios sistemos uždavinius – pradžioje reikia sukurti normalizavimo algoritmą, kurio dėka būtų galima supaprastinti duomenis. Duomenys pateikiami XML formatu. Čia saugoma OPL kalba aprašyta informacija apie egzistuojančią arba ieškotiną ERP sistemą. Šio uždavinio metu atliekamas rašybos elementų suvienodinimas (pvz.: išmetami specifiniai simboliai: #, š, \$, ..) bei XML duomenų nešmenų atmetimas – pašalinami „<“, „>“ simboliai ir tarp jų esančių OPM objektų pavadinimai. Beje, pastarieji dar naudojami siekiant išsaugoti informaciją duomenų bazėje.

Antrasis uždavinys - sutapatinimas. Čia pagal išsaugotus normalizuotus duomenis turi būti formuojamos užklauskos, siekiant iš duomenų bazės išrinkti ieškomą informaciją. Šiam procesui reikia realizuoti sutapatinimo algoritmą, gebantį atrinkti duomenų porcijas, jas sugretinti bei saugoti paruošiant rezultatus galutiniam apdorojimui.

Paskutiniu – rezultatų pateikimo – veiksmu yra nagrinėjami išsaugoti rezultatai, įvertinami vartotojo pasirinktais kriterijais ir būdais bei pateikiami nustatyta forma (pvz.: iš anksto nustatyta rezultatų struktūra, failo tipu ar vartotojo sąsajos forma).

1.9. Išvados

Atlikus ERP sistemų, jų funkcionalumo vertinimo modelių bei tam skirtų programinių paketų principų analizę, galima daryti tokias išvadas ir pažymėti:

- ERP – brangios bei sudėtingos sistemos, kurių naudojimo poreikis labai didelis, tačiau pritaikymas prie organizacijos reikalavimų reikalauja daug pastangų ir išlaidų.
- Atliktos analizės bei reikalingų spręsti uždavinių planavimo metu orientuotasi į tokią tyrimo sritį - ERP sistemų funkcionalumo specifikuavimo metodai, jų diegimo principai bei vartotojų šioms sistemoms keliamų reikalavimų aspektai. Tyrimo objektu pasirinktas ERP sistemų funkcionalumo specifikuavimo metodas ir jo programinis modelis. Visu darbu siekiama išspręsti susidarančią sistemų nesėkmingų diegimų grėsmę susidūrus su organizacijų keliamais reikalavimais.
- Analizės metu išnagrinėti kitų autorių siūlomi problemos sprendimai. Kaip tinkamiausias yra pasirinktas C.Rolland ir N.Prakash siūlomas ERP reikalavimų pritaikymo modelis. Taip pat galima pasinaudoti AMOS projekte naudojamo paieškos variklio logika.

- Sudarytas ERP reikalavimų pritaikymo modelio eskizinis variantas. Jo realizacijai numatoma naudoti OPM metodologija paremtomis taisyklėmis ir jų pagrindu veikiančia projektavimo priemone.

Pažymėtina, kad šis darbas atitinka šiuolaikinius ERP sistemų pritaikymo poreikius. Iki šiol rinkoje neegzistuoja sistemų, gebančių pakankamu detalumo lygmeniu sugretinti vartotojų ir ERP sistemų funkcionalumo reikalavimus. Šio darbo pagrindu numatomas sukurti programinis prototipas, kuris turi sumažinti vartotojų reikalavimų analizės pasirenkant ERP sistemą mastus bei pačio ERP sistemų diegimo kaštus, parenkant tinkamiausią įmonių išteklių valdymo sistemą.

2. REIKALAVIMŲ SUTAPATINIMO MODELIS IR JO REALIZACIJA

Šioje dalyje pateikiamas ERP sistemų funkcionalumo ir vartotojo pateiktų reikalavimų sutapatinimo modelio realizavimo projektas. Būsimo modelio struktūra, funkcijos, jos panaudojimo atvejai ir atskleidžiami vėlesniuose skyreliuose pateikiant aprašymus bei diagramas.

2.1. Modelio projekto tikslas

Šios informacinės sistemos kūrimo projekto tikslas yra išnagrinėti šios sistemos vartotojų grupes, panaudojimo atvejus, sudaryti loginę programos architektūrą, atlikti duomenų bazės projektavimą, apmastyti kokios galėtų būti realizacijos alternatyvos. Taip pat suprojektuoti sistemos vartotojo sąsają, sudaryti langų navigavimo planą.

2.2. Kuriamo modelio realizacijos bei veikimo aplinkos apibrėžimas

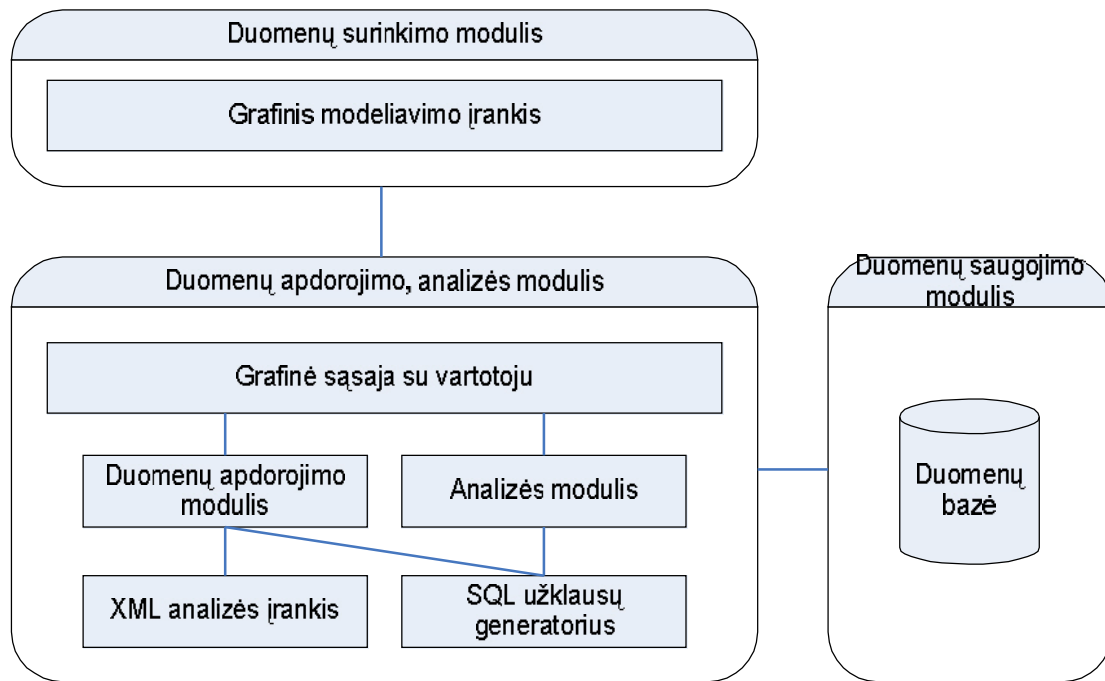
Numatoma, kad sistema bus kuriama bei vėliau ja bus dirbama Windows operacinės sistemos terpėje.

Modelis bus realizuotas Java programavimo kalba – ji pasirinkta ne tik dėl to, kad yra nemokama bei savo plačių galimybių, bet ir dėl to, kad siekiama bendro sukurtos sistemos vientisumo. Priežastis – OPM principais veikiantis programinės įrangos projektavimo įrankis OPCAT, kuris realizuotas Java. Šia priemone naudojama prašyti sistemų funkcionalumą bei vartotojų reikalavimus.

Surinktą informaciją numatoma saugoti MySQL duomenų bazėje. MySQL pasirinkta dėl puikaus suderinamumo su Java programavimo kalba (išlaikomas sistemos vientisumo tikslas) ir nesudėtingo valdymo.

2.3. Reikalavimų sutapatinimo modelio architektūra

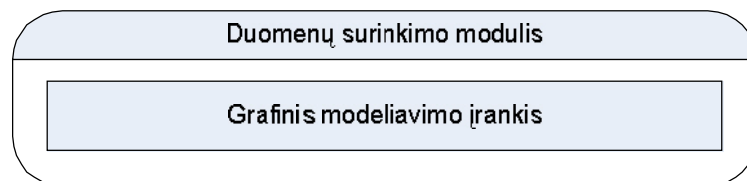
Pagal analizės metu sudarytas išvadas bei projektavimo metu atliktus planavimus yra numatyta, kad ERP reikalavimų ir vartotojų reikalavimų sutapatinimo modelį turėtų sudaryti trys dalys. Jas sudarantys komponentai yra suskirstyti pagal jų atliekamas funkcijas.



2.1 pav. Sutapatavimo modelio architektūros loginė schema

Komponentai yra suskirstyti į duomenų surinkimo, duomenų apdorojimo ir analizės, duomenų saugojimo modulius.

2.3.1. Duomenų surinkimo modulis

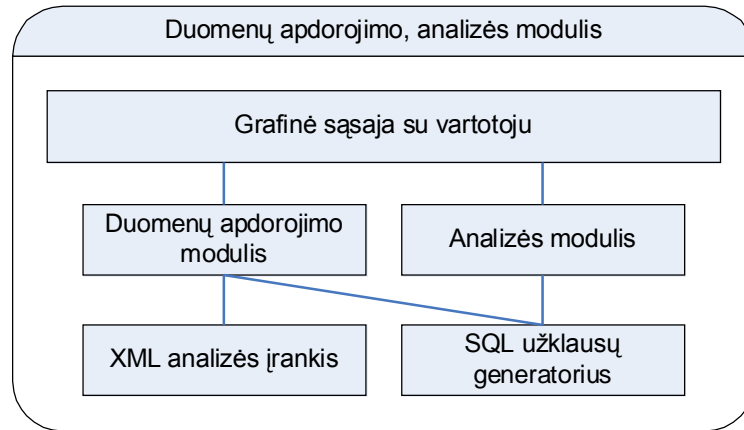


2.2 pav. Duomenų surinkimo modulio loginė schema

Duomenų surinkimo modulyje pažymėtas tik vienas komponentas, tai grafinis modeliavimo įrankis. Tai gali būti bet kokia programinė priemonė, kuri leidžia išreikšti organizacijos reikalavimus ar kompiuterizuotos sistemos galimybes. Veikiausiai tai turėtų būti programinės įrangos projektavimo (CASE) įrankis, leidžiantis išreikšti sistemų funkcijų hierarchijas bei pateikti papildomus duomenis. Kadangi šis darbas yra orientuotas į duomenų apdorojimo ir analizės sferą, tuomet čia yra panaudojamas esamas ir laisvai prieinamas įrankis, kuris dirba su OPM notacijomis – programinės įrangos projektavimo priemonė OPCAT. OPM metodologija buvo pasirinkta dėl savo lankstumo. Vartotojas, besinaudojantis sistemų funkcionalumo ir organizacijų reikalavimų sutapatavimo modeliu, darbą pradėtų sudarymas esamos arba tikėtinos sistemos funkcinę schemą. Tai atliekama naudojantis

OPM principais veikiančiu projektavimo įrankiu OPCAT. Schema turi būti sudaryta iš įprastinių OPM metodologijos komponentų – objektų, procesų, sąryšių tarp jų aibe bei būsenų.

2.3.2. Duomenų apdorojimo, analizės modulis



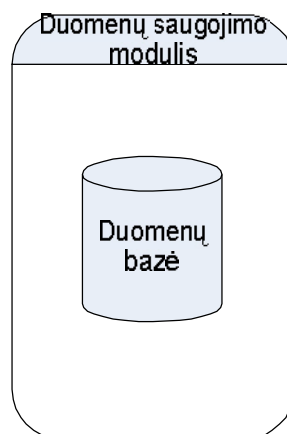
2.3 pav. Duomenų apdorojimo, analizės modulio loginė schema

Sudarius esamos ar tikėtinos ERP sistemos funkcionalumo schemą, toliau tęsiamas darbas su duomenų apdorojimo bei analizės moduliu. Vartotojas čia negali matyti šią dalį sudarančių komponentų, jis su sistema bendrauja tik per grafinę sąsają. Ja naudodamasis jis gali nurodyti tolesnius veiksmus: duomenų nuskaitymą, analizės metu naudojamų įverčių nustatymą, duomenų saugojimą bei pateiktų rezultatų analizavimą.

Šią loginę sistemos dalį sudaro dvi sritys, atliekančios sau būdingus veiksmus – tai duomenų apdorojimo bei analizės moduliai.

Duomenų apdorojimo modulis, naudodamasis anksčiau sudarytomis duomenų normalizavimo taisyklėmis, atlieka duomenų supaprastinimą iki saugotino lygmens. Norint vartotojo pateiktus duomenis išsaugoti bus naudojamos dvi papildomos komponentais – tai XML analizės įrankis bei SQL užklausų generatorius. Pirmasis naudojamas nagrinėjant duotą duomenų bylą, pagal sudarytas notacijas atpažįstant duomenų struktūras bei išgaunant duomenų objektus. Antrasis komponentas yra naudojamas siekiant šiuos surinktus duomenis išsaugoti duomenų bazėje. Jo funkcijos taip pat yra naudojamos analizės metu, kai siekiama išgauti konkrečią informaciją, saugomą duomenų bazėje.

2.3.3. Duomenų saugojimo modulis



2.4 pav. Duomenų saugojimo modulio loginė schema

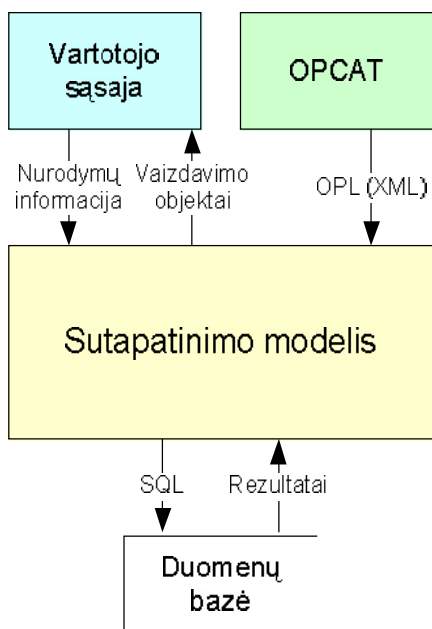
Šią dalį sudaro tik reliacinė duomenų bazė. Joje yra kaupiama ir iš jos išgaunama anksčiau išsaugota informacija. Duomenų saugyklos struktūra yra aprašyta 2.7. punkte.

2.4. Sistemą sudarantys duomenų srautai

Projektuojamoje sistemoje duomenų srautai pasiskirsto tarp sistemos sudėtinių elementų: grafinės projektavimo priemonės (projekte pasirinktos OPCAT), sutapatinimo modelio, duomenų bazės ir vartotojo grafinės sąsajos.

Pagrindinius duomenis sistemai perduoda vartotojas, kuris naudojantis OPCAT įrankiu sudaro reikalavimų schemą. Sudarytų diagramų atitikmeniu – OPL kalba yra išsaugomas į failą. Čia diagramos saugomos XML pavidalu, taip siekiant išlaikyti diagramų struktūrą. Sekantys veiksmai, kaip ir aprašyta 1.5.2. skyriuje „Kompiuterizuojamos sistemos funkcijos“, yra atliekami paeiliui: duomenys nuskaitomi, normalizuojami, talpinami į duomenų bazę arba sulyginami su iš jos išgauta informacija bei gauti rezultatai per grafinę sąsają pateikiami vartotojui.

Duomenų srautų schema pateikta 2.5 paveiksle:



2.5 pav. Sistemos duomenų srautų schema

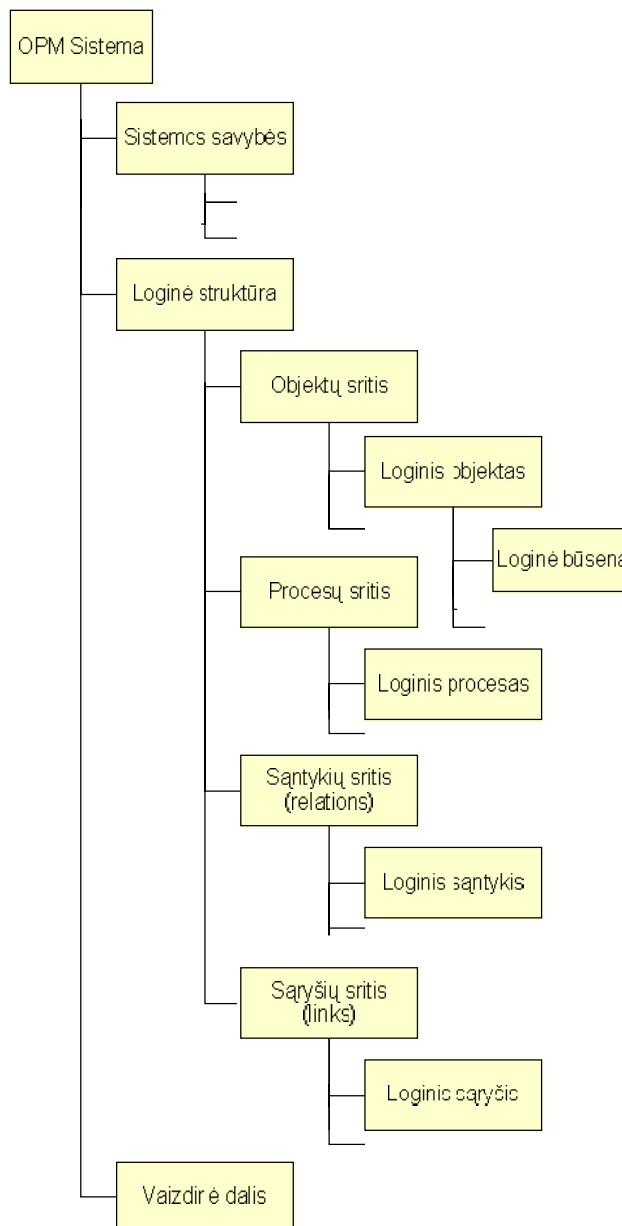
Pažymėtina, kad vartotojo sąsaja ne tik gauna suformuotus rezultatus, kurie saugomi objektų pavidalu, tačiau jos pagalba sistemai galima perduoti nustatytą informaciją (svarbumo kriterijus rezultatų analizei, jų įverčius ir pan.).

2.4.1. Įėjimo informacijos struktūra

Pradžioje vartotojas, pasinaudodamas anksčiau sukurtu reikalavimų specifikavimo šablonu, modeliuoja ieškomos sistemos schemą. Tam jis naudoja grafinį projektavimo įrankį OPCAT. Sudarytą schemą išsaugos XML formatu. Čia OPCAT priemonė išsaugo visą informaciją apie sumodeliuotą sistemą.

XML kalba aprašytame kode išskirtos trys stambios sritys: sistemos (failo) duomenys, projektuotos sistemos loginis ir vaizdavimo informacijos blokai. Loginės informacijos bloką sudaro keturios skiltys, kuriose aprašomi objektai, procesai, struktūriniai bei procedūriniai ryšiai. Būsenos apibrėžiamos objektų skiltyje.

Bendrinė įėjimo duomenų schema yra tokia:



2.6 pav. Įėjimo duomenų schema

Įėjimo duomenys sudaro OPCAT kūrėjų nustatytą struktūrą – duomenų grafą.

Šakninė šio grafo viršūnė yra *OPM Sistema* dalis. Čia trumpai aprašoma informacija apie sudarytą diagramą: autorius, sistemos pavadinimas, sukūrimo data ir pan. Toliau ši dalis aprašo kitas tris stambias sritis: *Sistemos savybės*, *Loginė struktūra*, *Vaizdinė dalis*.

Sistemos savybių skiltyje saugoma informacija, kuri buvo nurodyta sistemos diagramos kūrimo metu. Tai informacija apie pačią diagramą, apribojimo taisykles ir pan.

Pagrindinė ir svarbiausia informacija, kuri bus apdorojama, yra pateikiama *Loginės struktūros* skiltyje. Šiame darbe kuriamas modelis nagrinėja tik loginę reikalavimų specifikaciją – likusių dalių analizė galėtų būti įtraukta sekančiame reikalavimų sutapatavimo modelio plėtos etape. Pradžioje yra *Objektų skiltis*, kurioje vienas po kito išvardinami diagramoje sukurti *Loginiai objektai*. Loginis objektas aprašo duomenų struktūrą – esybę, kurią gali sudaryti tokie elementai:

2.1 lentelė.

Loginio objekto elementai

Elementas	Aprašymas	Galimos loginės reikšmės
Pavadinimas	Projektuotojo nurodytas objekto pavadinimas	
Pradinė reikšmė	Objektui priskirta pradinė reikšmė	
Raktas	Žymi ar objektas yra raktinis. Šiuo elementu žymima ar objektas yra sistemos modulis.	Taip/ne
Tipas	Nurodomas objekto tipas	Sudėtinis (sudėti nurodo projektuotojo), įprastinis objektinis tipas (baitas, skaičius, simbolis, simbolių masyvas,...), projektuotojo sukurtas tipas.
Materija	Žymi ar objektas yra fizinis, ar loginis.	Fizinis/loginis
Kilmė	Parodo ar tai yra aplinkos objektas, ar sisteminis.	Aplinkos/sisteminis
Pastovus	Žymi, kad objektas yra konstanta.	Taip/ne
Aprašymas	Projektuotojo sudarytas aprašymas, apibūdinantis objektą.	

Elementu *Raktas* yra pasinaudojama tam, kad atvaizduoti ir išskirti loginį objektą kaip analizuojamos ERP sistemos ar sudaromų organizacijos reikalavimų sritį (ERP sistemos modulį).

Loginis objektas taip pat gali turėti papildomus elementus – *logines būsenas*. Jei šios buvo apibrėžtos diagramoje, tai čia jas galima panaudoti žinant kuriam konkrečiam objektui jos priklauso. Priešingu atveju šių elementų nebus.

Loginę būseną galima apibūdinti šiais elementais:

2.2 lentelė.

Loginės būsenos elementai

Elementas	Aprašymas	Galimos loginės reikšmės
Pavadinimas	Projektuotojo nurodytas būsenos pavadinimas	
Pradinė	Žymi, kad tai yra pradinė būseną.	Taip/ne
Galinė	Žymi, kad tai yra galinė būseną.	Taip/ne
Įprastinė (default)	Žymi, kad tai yra įprasta būseną.	Taip/ne
Mažiausia / didžiausia trukmė	Aprašo, kiek ilgai gali būseną tęstis.	Begalybė, bet koks kitas apibrėžtas laiko matas.
Aprašymas	Aprašymas, apibūdinantis objektą, jo būseną.	

Procesų skiltyje yra išvardinami diagramoje sukurti ir aprašyti *loginiai procesai*. Loginis procesas – duomenų struktūra, apibūdinanti veiksmą, kurį gali atlikti diagramoje aprašytas objektas arba kuris ta objektą keičia. Šią struktūrą sudaro:

2.3 lentelė.

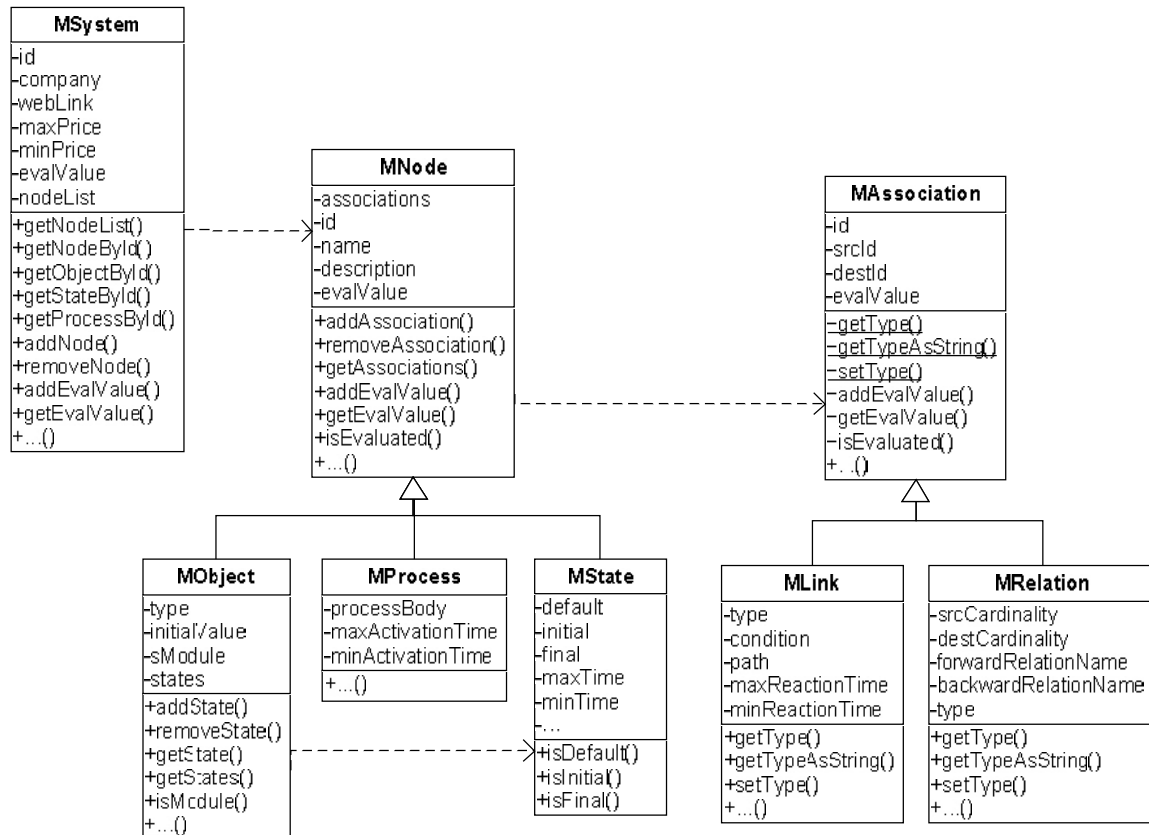
Loginio proceso elementai

Elementas	Aprašymas	Galimos loginės reikšmės
Pavadinimas	Projektuotojo nurodytas proceso pavadinimas.	
Materija	Žymi ar procesas yra fizinis, ar loginis.	Fizinis/loginis
Kilmė	Parodo ar tai yra aplinkos ar sistemos procesas.	Aplinkos/sisteminis
Turinys	Proceso turinys, kūnas.	
Mažiausia / didžiausia trukmė	Aprašo, kiek ilgai procesas gali būti vykdomas.	Begalybė, bet koks kitas apibrėžtas laiko matas.
Aprašymas	Proceso (atliekamo veiksmo) aprašymas.	

Po loginių procesų išvardinimo yra pateikiama dvi ryšiams tarp loginių esybių skirtos sritys – pirmoje pateikiami duomenys apie santykius (*Santykių sritis* (angl. relations)), o antroje – apie sąryšius (*Sąryšių sritis* (angl. links)). Kiekvienas iš ryšio tipų talpina savitą informaciją apie siejamas esybes bei patį ryšį. Sukurtame modelyje naudojama tik tokia ryšių informacija: siejamos esybės bei ryšio tipas. Platesnis ryšių aprašymas pateikiamas 2 priede („OPM metodologija, žymėjimai“).

2.5. Reikalavimų aprašymuose naudojamų duomenų struktūrų ir operacijų aprašai

Naudojantis OPM metodologijos notacijomis bei OPCAT priemonės sugeneruoto diagramų failo struktūra yra numatyta, kad reikalavimų sutapatinimo prototipas dirbs operuodamas tokia duomenų struktūra bei ryšiais tarp jų:



2.7 pav. Sutapatinimo modelyje naudojamų duomenų struktūrų ir operacijų aprašai

MSystem klasė naudojama saugoti informaciją apie sistemą, kurios funkcionalumo specifikacijų schemą siekiama išsaugoti ar sutapatinti. Kiekvienas iš atributų gali būti keičiamas bei skaitomas naudojantis atitinkamais metodais. Toliau atskleidžiamos tik esminės šios duomenų struktūros vietos.

MSystem kintamasis *evalValue* naudojamas sistemos tapatumo įvertinimui saugoti – tai įprasta skaitinė reikšmė. Analogiškai yra ir apibendrintose grafo viršūnių (*MNode*) ir ryšių (*MAssociation*) klasėse. Šioje klasėje saugoma tiek bendra informacija apie sistemą (kompanija, kainos,..), tiek ir visas OPM diagramose atvaizduotas grafas. Duomenys apie grafą yra talpinami į grafo viršūnių sąrašą *nodeList*.

Grafo viršūnėmis gali būti trys OPM metodologijoje naudojamos esybės – objektai (šio tipo struktūrą atspindi klasė *MObject*), procesai (*MProcess*), būsenos (*MState*). Jas apibendrina viršūnės

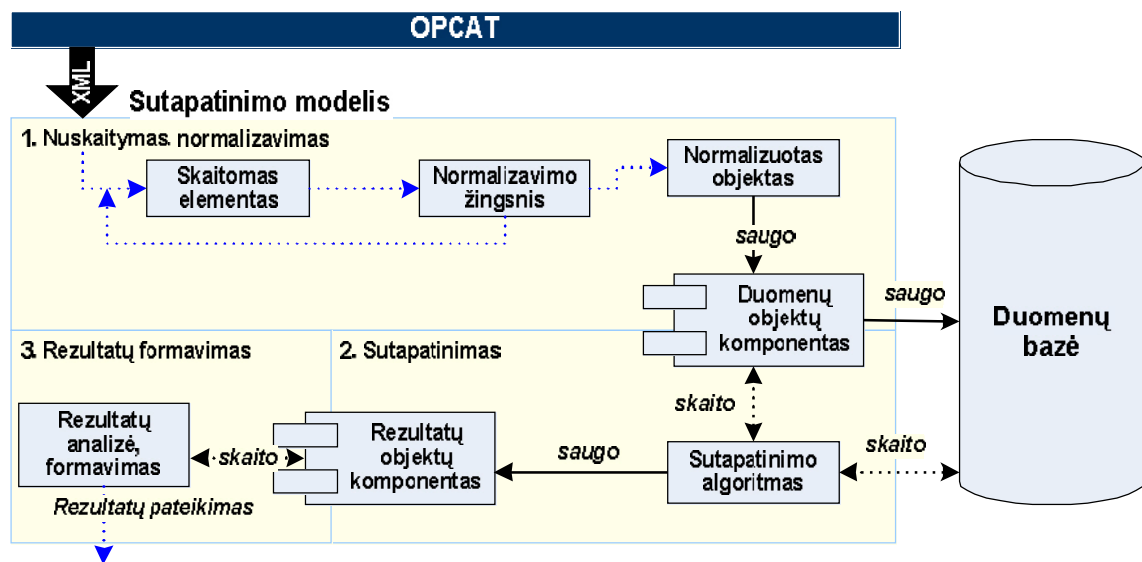
(*MNode*) klasė. Čia saugomi duomenys apie viršūnę, kaip apie patį duomenų elementą. *MObject* klasė iš kitų viršūnių klasių išsiskiria tuo, kad ji gali atspindėti tiek OPM loginį objektą, tiek loginį modulį, apibrėžiantį aprašomos ERP sistemos funkcinę sritį. Identifikavimui, ar tas objektas yra modulis naudojamas metodas *isModule()*. Objektas gali turėti ir jam priskirtų būsenų, kurios saugomos būsenų sąrašė *states*. Klasės *MProcess* ir *MState* skirtos tipinės informacijos, kurios struktūrą nustatyta OPCAT įrankiu, taipinimui.

Apibendrintoje viršūnės klasėje taip pat nurodoma informacija apie ryšį su kitomis viršūnėmis – dirbama su ryšių sąrašiu. Į jį dedami ryšiai gali būti dvejopi – santykiai (*MRelation*) ir sąryšiai/jungtys (*MLink*). Jos atskirtos yra dėl to, kad turi skirtingas prasmes – vieni ryšiai parodo esybių susietumą tarpusavyje, kiti – esybių poveikį viena kitai.

Taigi, ši duomenų struktūra yra sukurta darbui su duomenimis, kurie gali būti tiek grafo, tiek kito pavidalo. Šis pritaikymas atliktas pagal OPM metodologiją.

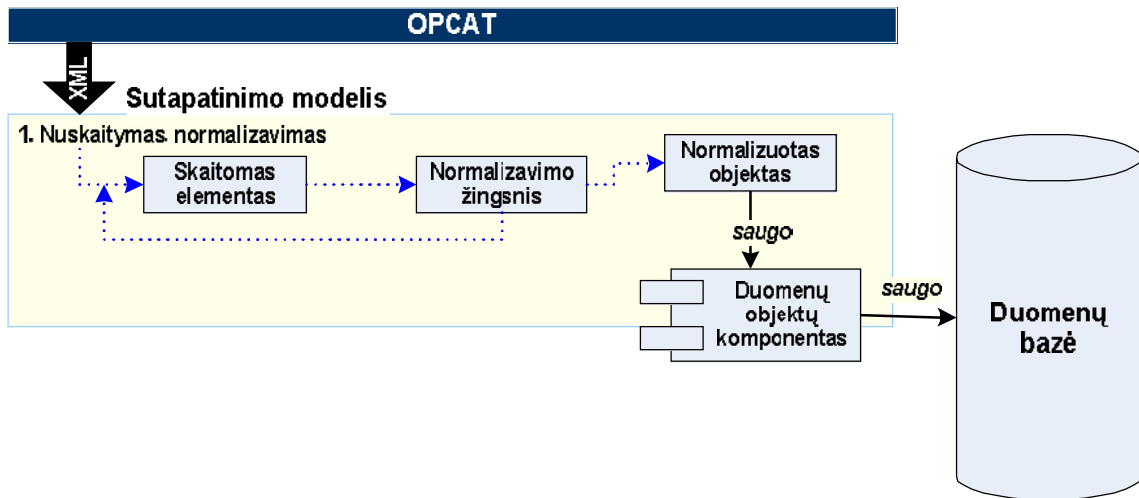
2.6. Sistemos veiklos modelis

Šiame skyriuje pateikiamos schemas, atspindinčios informacinės sistemos darbą ieškant tiksliausiai derančios ERP sistemos prie vartotojo pateiktų duomenų.



2.8 pav. Sistemos veiklos modelis

Darbas pradamas sudarant ieškomos ERP sistemos funkcionalumo schemą naudojantis PĮ projektavimo įrankiu, prie kurio yra pritaikyta šis sutapatinimo modelis.

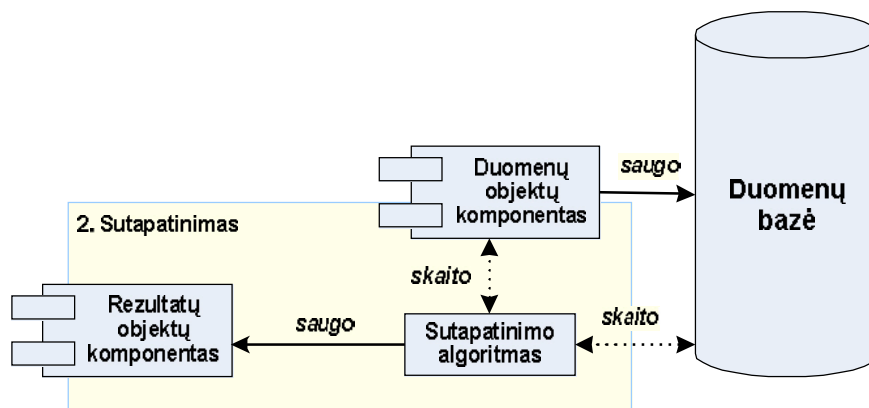


2.9 pav. Nuskaitymo, normalizavimo veiklos loginė schema

Vartotojas, baigęs darbą su OPCAT, pereina prie veiksmų su sutapatinimo modeliu. Pradinis žingsnis yra duomenų nuskaitymas. Šioje dalyje yra atliekamas duomenų failo teisingumo tikrinimas bei surinktos informacijos užkrovimas į kompiuterio atmintį.

Skaitant duomenis iš pradinių duomenų failo yra vykdomi pažingsniniai veiksmai: nuskaityma duomenų porcija, patikrinamas jos teisingumas. Jei jie teisingi toliau atliekamas normalizavimas – tai duomenų supaprastinimas iki sulyginamojo lygmens. Taip pat vykdomas duomenų tikrinimas siekiant išvengti duomenų nesutapimo klaidos. Tai atliekama visai gautai informacijai taikant tas pačias taisykles: šalinant specialiuosius simbolius (#, \$, u, ..), tarpus, sprendžiant didžiųjų/mažųjų raidžių problemą. Normalizavimo žingsnis tai vykdo naudodamasis anksčiau sudarytomis normalizavimo taisyklėmis, kurios priklausomai nuo užsakovo reikalavimų, gali būti saugomos tiek duomenų bazėje, tiek faile (pavyzdžiui, XML aprašu).

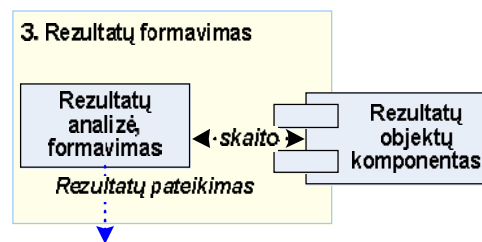
Kai atliktas duomenų nuskaitymas bei normalizavimas ir rezultate turime normalizuotus objektus, šie yra patalpunami į duomenų objektų komponentus. Vėliau juos galima saugoti duomenų bazėje arba atlikti tolesnius veiksmus *Sutapatinimo* pakopoje.



2.10 pav. Nuskaitymo veiklos loginė schema

Po normalizavimo veiksmo turime aibę vartotojo pateiktų duomenų. Jie imami paeiliui ir sulyginami su gautais duomenimis iš duomenų bazės. Pastarieji yra išgaunami sudarant SQL užklaudas, priklausomas nuo sistemoje saugomų normalizuotų duomenų. Po kiekvieno sutapatavimo gausim tokius rezultatus: objektas arba procesas egzistuoja/negzistuoja informacinės sistemos duomenų saugykloje. Pagal tai reikia įvertinti konkretaus objekto/funkcijos rezultatų įvertinimą bei perskaičiuoti jo tėvinio objekto įvertinimą. Įvertinimai gali būti skaičiuojami procentaliai, t.y. pateikiant vartotojui rezultatus, kurie atspindės kiek tiksliai vartotojo pateikta reikalavimų schema atitiko duomenų saugykloje saugomą ERP sistemos funkcionalumo schema. Tačiau įvertiniai gali būti skaičiuojami ir visai kitu būdu, tai galima atlikti prieš sutapatavimo žingsnį, sistemai pateikiant įvertinimus, kurie nurodo dalių didesnę ar mažesnę svarbą.

Galiausiai rezultatai yra perduodami į paskutinį etapą – tai rezultatų apdorojimas.



2.11 pav. Rezultatų formavimo veiklos loginė schema

Čia yra analizuojami išsaugoti rezultatų objektai, sudaromi rezultatų dokumentai, atliekamas jų formavimas bei pateikimas vartotojui.

Vartotojui gavus rezultatus sistemos darbo ciklas baigiasi, čia vartotojas naudojasi tik vartotojo sąsajos teikiama galimybe apžvelgti gautuosius rezultatus.

2.6. Reikalavimų sutapatavimo sistemos naudojami algoritmai

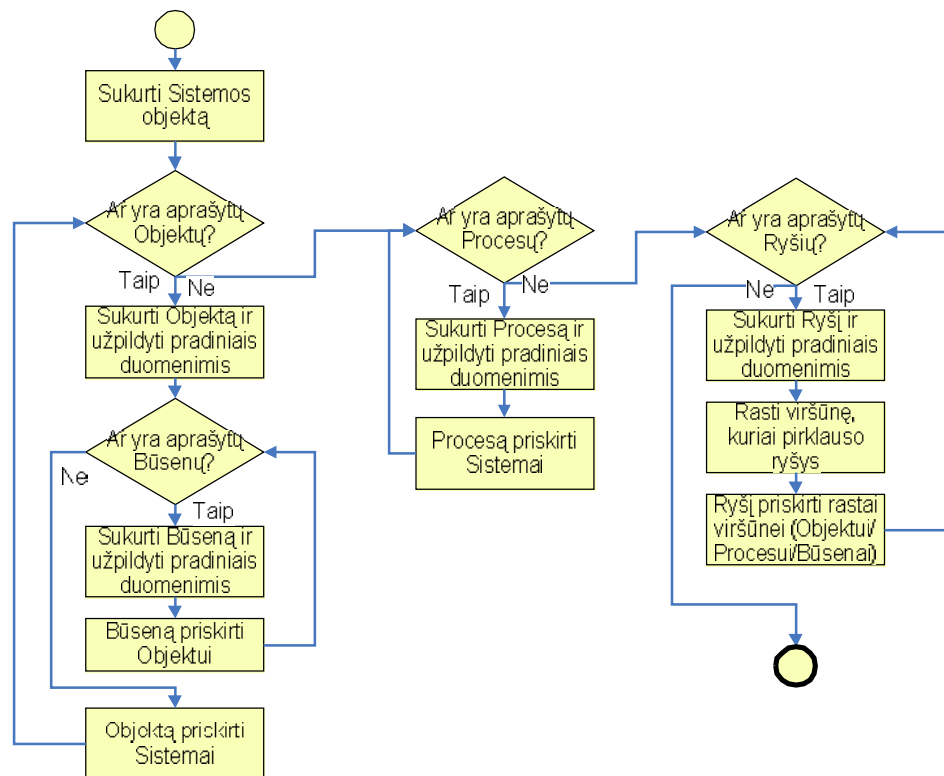
Kuriama sistema naudos kelis, kiekvienai jos daliai specifinius algoritmus. Tai priklauso nuo kiekvienos iš jų atliekamų funkcijų – jos yra: ERP sistemos funkcionalumo arba vartotojų reikalavimų nuskaitymas, darbas su duomenų baze (funkcionalumo/reikalavimų aprašų rašymas į duomenų saugyklą bei skaitymas iš jos), reikalavimų ar sistemų funkcionalumo aprašų sutapatinimas.

Pradžioje reikėtų apibrėžti sisteminius objektus, kuriais bus operuojama, bei jų struktūrą. Visur turi būti dirbama su medžio tipo sistemos aprašu. Taigi šaknine medžio viršūne bus objektas, pavadinimu *Sistema*. Joje ir bus saugoma visa funkcionalumo aprašo struktūra. Žemesnes šakas sudarytų *Moduliai*, kurie atspindi stambiausias, specializuotas ERP sistemų sferas (pavyzdžiui, „Gamyba“, „Finansų apskaita“, „Atsargų apskaita ir valdymas“ ir pan.). Čia talpinami *Objektai*

(esybės), objektų *Būsenos* (būsenos, kurioje gali būti esybės) ir *Procesai* (veiksmai, kuriais transformuojamos esybės ar jų dalis). Šie objektai naudojami atsižvelgiant į naudojamos OPM metodologijos specifiką. Juos sujungdami tarpusavio *Ryšiais* sudaromas sistemos funkcionalumo / vartotojų reikalavimų grafas, šitaip išreiškiant naudojamų duomenų vientisumą.

2.6.1. Pradinių duomenų – funkcionalumo ar reikalavimų aprašų - skaitymas

Pradinių duomenų nuskaitymo algoritmas yra tiesiogiai priklausomas nuo pradinių duomenų struktūros.



2.12 pav. Pradinių duomenų nuskaitymo algoritmo blokinė schema

```

nauja Sistema;

kol (yra aprašytų Objektų) {
    naujas Objektas;
    kol (yra aprašytų Būsenų) {
        nauja Būsena;
        Objektas.pridėti(Būsena);
    }
    Sistema.pridėti(Objektas);
}

kol (yra aprašytų Procesų) {
    naujas Procesas;
    Sistema.pridėti(Procesas);
}

kol (yra aprašytų Ryšių) {
    naujas Ryšys;
    rasti (viršūnę Objektą/Procesą/Būsena,
        kuri jungiasi šiuo ryšiu su kita viršūne);
    viršūnė.pridėtiRyšį(Ryšys);
}

patalpinti Sistemą programos atmintyje;

```

2.13 pav. Pradinių duomenų – funkcionalumo ar reikalavimų aprašų – nuskaitymo algoritmas

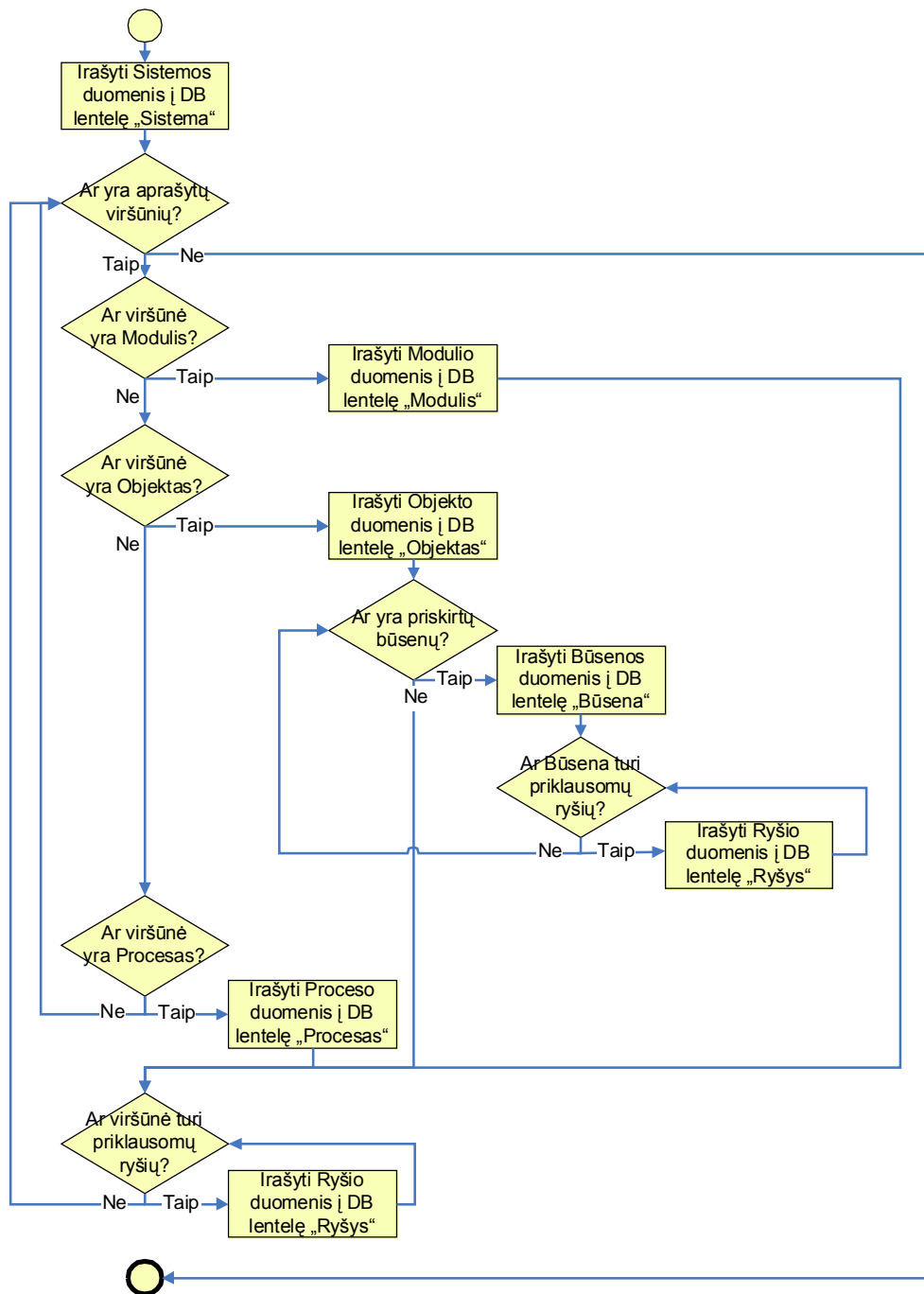
Pabrėžtina, kad po kiekvienos formuojamo grafo viršūnės (Objekto, proceso ar Būsenos) ar ryšio sukūrimo yra priskiriami toms esybėms priklausantys papildomi duomenys, aprašyti OPD reikalavimų diagramos sudarymo metu. Taip būtų tokie laukai kaip pavadinimas, aprašymas, tipas ir pan.

Be to, kiekvieno informacijos gabaliuko nuskaitymą iš duomenų failo lydi duomenų normalizavimo žingsnis. Normalizavimas atliekamas tik iš semantinės (gramatinės) pusės. Šiame etape yra analizuojamas kiekvienas tekstinis simbolis ir sutikrinamas su anksčiau programuotojo specializuotame faile apibrėžtais normalizuotinais simboliais ar jų seka. Jei teksto dalis yra aprašyta normalizuotinių simbolių aibėje, tuomet ji yra pakeičiama kita – siūlytina – reikšme, kurią taip pat yra nurodęs programuotojas. Taip pat gali būti nurodyta, kad kai kuriuos simbolius reikia šalinti nekeičiant kitais.

2.6.2. Reikalavimų aprašų talpinimas duomenų saugykloje bei skaitymas iš jos

Vartotojai, nurodę programai nuskaityti duomenis iš sudaryto duomenų failo, toliau gali rinktis punktą, skirtą jų išsaugojimui duomenų saugykloje.

Čia programa papildo duomenų bazės lenteles nauja informacija.



2.14 pav. Duomenų saugyklos užpildymo algoritmo blokinė schema


```

įrašyti Sistemos informaciją į DB lentelę „Sistema“;

kol (yra priskirtų viršūnių) {

    jei (viršūnė yra Modulis) {
        įrašyti Modulio informaciją į DB lentelę „Modulis“;

    } kitaip jei (viršūnė yra Objektas) {
        įrašyti Objekto informaciją į DB lentelę „Objektas“;
        kol (yra priskirtų būsenų) {
            įrašyti Būsenos informaciją į DB lentelę „Būsena“;
            kol (yra Būsenai priklausomų ryšių)
                įrašyti Ryšio informaciją į DB lentelę „Ryšys“;
        }

    } kitaip jei (viršūnė yra Procesas) {
        įrašyti Proceso informaciją į DB lentelę „Procesas“;

    }

    kol (yra viršūnei priklausomų ryšių) {
        įrašyti Ryšio informaciją į DB lentelę „Ryšys“;
    }

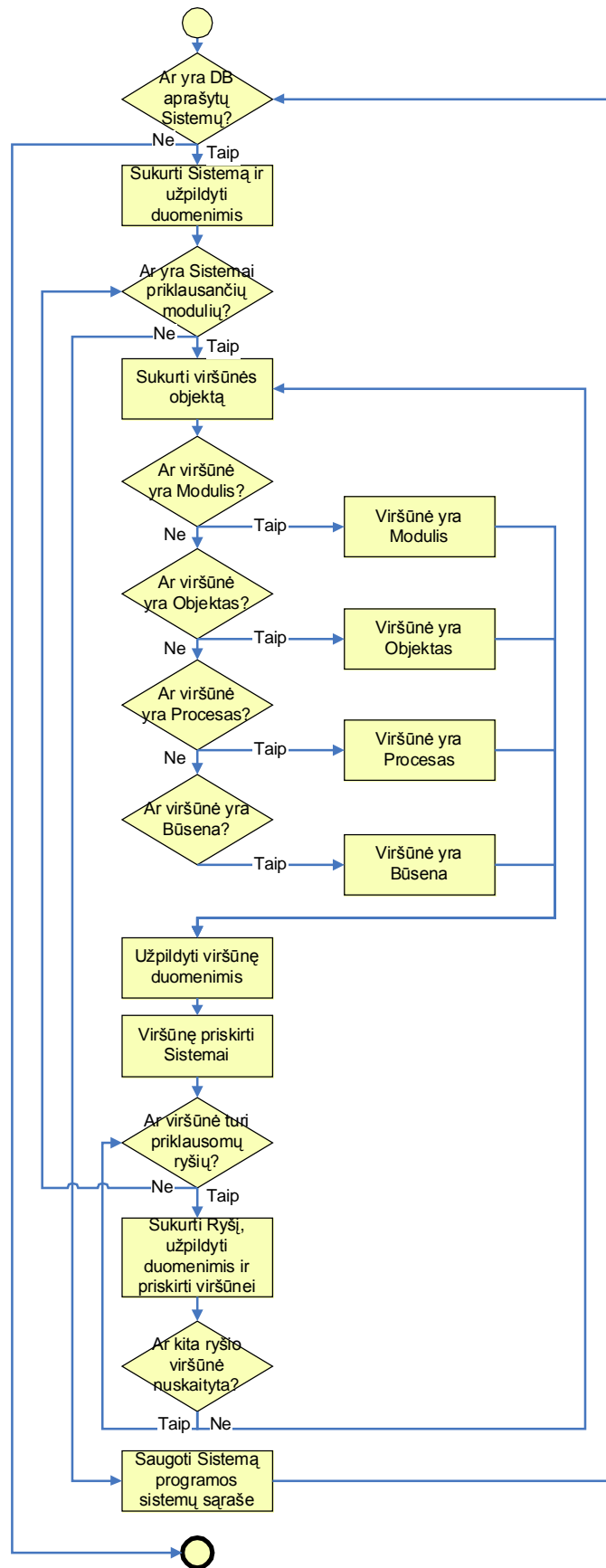
}

```

2.15 pav. Algoritmas, skirtas duomenų saugyklos užpildymui reikalavimų aprašu

Duomenų įrašymui yra naudojamosi objektais aprašyta sistema, neatsižvelgiant į jos medžio tipo struktūrą. Tačiau ji yra išlaikoma, kadangi duomenys rašomi į struktūrizuotą duomenų bazę, pagal kurią vėliau gali tekti juos atstatyti, t.y. nuskaityti iš duomenų bazės ir patalpinti į reikalavimų sutapatavimo programos atmintį.

Visiškai priešingai yra elgiamasi nuskaitant reikalavimų specifikacijas iš duomenų bazės (pavyzdžiui, sutapatavimo etapui). Čia pasirenkama sistema, atrenkami jai priklausantys moduliai, o vėliau pagal tarpusavio ryšius nustatomos ir šių medžio viršūnių šakos su savom priklausomybėm.



2.16 pav. Reikalavimų aprašų skaitymo iš duomenų bazės algoritmo blokinė schema

```

kol (yra DB aprašytų Sistemų) {
    nauja Sistema;
    kol (yra Sistemai priklausančių modulių){
        naujas Modulis;
        Sistema.tirtiViršūnę(Sistema, Modulis);
    }

    talpinti Sistema į programos atmintį;
}

-----

tirtiViršūnę (Sistema, viršūnė){
    Viršūnė v;

    jei (viršūnė yra Modulis){
        v = naujas Modulis;
    } kitaip jei (viršūnė yra Objektas){
        v = naujas Objektas;
    } kitaip jei (viršūnė yra Procesas){
        v = naujas Procesas;
    } kitaip jei (viršūnė yra Būsena){
        v = nauja Būsena;
    }
    Sistema.pridėtiViršūnę(v);

    rasti (v priklausančius ryšius);
    kol (yra v priklausančių ryšių){
        naujas Ryšys;
        v.pridėtiRyšį(Ryšys);
        rasti (kita_ryšio_viršūnė);

        jei (ryšio kita_ryšio_viršūnė nenuskaityta)
            Sistema.tirtiViršūnę(kita_ryšio_viršūnė);
    }
}

```

2.17 pav. Reikalavimų aprašų skaitymo iš duomenų bazės algoritmas

Šio algoritmo metu yra imamas sistemos aprašas ir pagal jį bandoma atstatyti visą reikalavimų aprašo struktūrą. Yra žinomi sistemos moduliai, tad yra atliekama kiekvieno iš jų analizė – sukuriamas pats viršūnės objektas, priskiriama jam priklausanti papildoma informacija bei tikrinami visi jo ryšiai. Radus viršūnei priskirtą ryšį atliekama kitos ryšio viršūnės (antrojo ryšio galo) paieška. Ją atradus veiksmas kartojamas rekursiškai – taip daromas medžio šakojimasis iki žemiausio aprašyto lygio. Galiausiai, kai visos viršūnės yra iširtos, gauname reikalavimų medį, kurį galima įkelti į kompiuterio operatyviąją atmintį ir kur turimais duomenimis bus galima pasinaudoti sekančių žingsnių metu.

Pastaba: šiame loginiame algoritmo aprašyme neatsispindi tam tikros medžio sudarymo taisyklės, t.y. kad viršūnė Būsena gali būti pataļpinta tik į tėvinių objektą, o ne į sistemos viršūnių sąrašą, kaip tai yra daroma su Moduliais, Objektais ir Procesais. Ryšių informacija saugoma atitinkamose viršūnėse, o ne tiesiogiai Sistemos objekte.

2.6.3. Reikalavimų aprašų sutapatinimas

Esminė šio projekto dalis yra ERP sistemų funkcionalumo ir vartotojo reikalavimų aprašų sutapatinimas. Žemiau pateikiamas loginė šio algoritmo išraiška.

```

Sistema s;          -- sistema, kurios duomenis tapatinsim su
                    kitomis sistemomis
kol (yra vertinamų sistemų){
    vertinamaSistema = viena iš vertinamų sistemų;
    vertintiSistemas(s, vertinamaSistema)
}
-----
vertintiSistemas(s, vS){
    kol (s turi modulių){
        double ivertinimas = 0;
        Modulis = vienas iš s modulių;  -- modulis, pagal kurį vertinama
        jei(vS turi modulį tapatų moduliui Modulis)
            ivertinimas = vertintiModulius(Modulis, tapatusModulis);
        vS.perskaiciuotiIvertinimus(ivertinimas);
    }
}
-----
vertintiModulius(m, vM){
    // analogiški veiksmai metoduose vertintiProcesus, vertintiBūsenas
    vM.perskaiciuotiIvertinimus( sulygintiDuomenis(m, vM) );
    vM.perskaiciuotiIvertinimus( vertintiViršūniųRyšius(m, vM) );
    gražinti vM.ivertinimas;
}
-----
vertintiObjektus(o, vO){
    vO.perskaiciuotiIvertinimus( sulygintiDuomenis(o, vO) );
    kol (o turi būsenų){
        double ivertinimas = 0;
        Būsena = viena iš o būsenų;      -- būsena, pagal kuria vertinama
        jei(vO turi būseną tapačią būsenai Būsena)
            ivertinimas = vertintiBūsenas(Būsena, tapatiBūsena);
        vO.perskaiciuotiIvertinimus(ivertinimas);
    }
    vO.perskaiciuotiIvertinimus(vertintiViršūniųRyšius(o, vO) );
    gražinti vO.ivertinimas;
}

```

2.18 pav. Reikalavimų aprašų sutapatinimo algoritmas (1-a dalis)

```

-----
vertintiViršūniųRyšius(v, vV){
  //v - tapatinama viršūnė, vV - vertinama viršūnė
  kol (v turi ryšių){
    double ivertinimas = 0;
    Ryšys = vienas iš v ryšių;          -- ryšys, pagal kurį vertinama
    jei(vV turi ryšį tapatų ryšiui Ryšys)
      ivertinimas = vertintiRyšius(Ryšys, tapatusRyšys);
    vV.perskaiciuotiIvertinimus(ivertinimas);
  }
  gražinti vV.ivertinimas;
}
-----

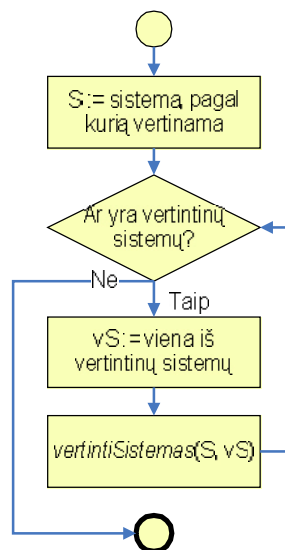
vertintiRyšius(r, vR){
  vR.perskaiciuotiIvertinimus( sulygintiDuomenis(r, vR) );
  jei (ryšio vR.kitas_galas nevertintas) {
    double ivertinimas = 0;
    jei (vR.kitas_galas yra modulis)
      ivertinimas = vertintiModulius (r.kitas_galas, vR.kitas_galas);
    kitaip jei (vR.kitas_galas yra objektas)
      ivertinimas = vertintiObjektus (r.kitas_galas, vR.kitas_galas);
    kitaip jei (vR.kitas_galas yra procesas)
      ivertinimas = vertintiProcesus (r.kitas_galas, vR.kitas_galas);
    kitaip jei (vR.kitas_galas yra būseną)
      ivertinimas = vertintiBūsenas (r.kitas_galas, vR.kitas_galas);
  }
  vR.perskaiciuotiIvertinimus(ivertinimas);

  gražinti vR.ivertinimas;
}

```

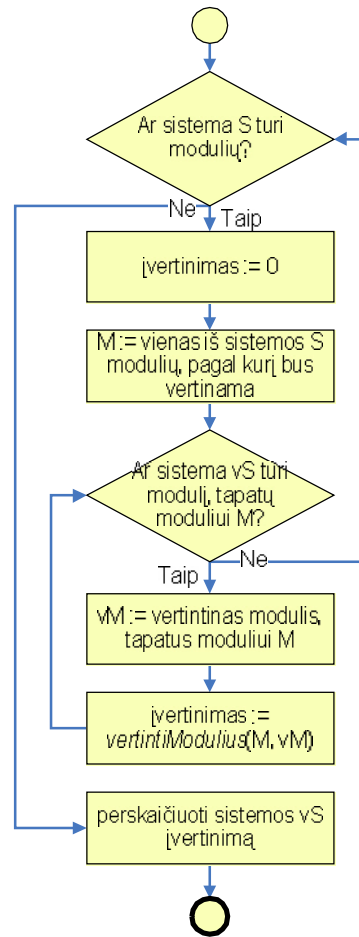
2.19 pav. Reikalavimų aprašų sutapatinimo algoritmas (2-a dalis)

Prieš pradėdant sistemų sutapatinimą, į programą iš duomenų bazės užkraunamos sistemos, kurias reikės įvertinti. Taip pat reikia turėti duomenis „darbinės“ sistemos, su kuria bus atliekamas sulyginimas. Šiuos punktus įvykdžius procesas vykdomas toliau.



2.20 pav. Vertinamų sistemų sulyginimo algoritmo schema. Pradžia

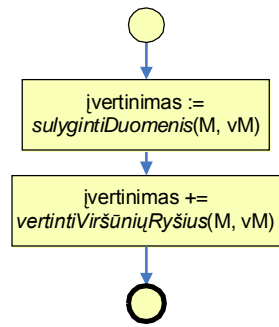
Paeiliui imamos sistemos iš vertinamų sistemų sąrašo bei atliekama nuodugni jų analizė.



2.21 pav. Reikalavimų sutapatinimas sisteminiame lygmenyje

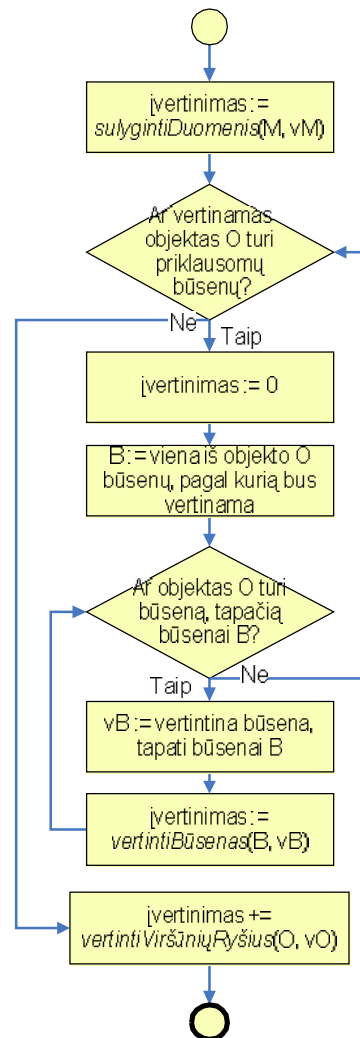
Pradžioje abiejuose sistemų aprašuose ieškoma tapačių viršūnių – modulių. Juos radus sulyginami tokie duomenys kaip pavadinimas, aprašymas. Toliau tapatinami jų ryšiai, rodantys santykius su kitomis viršūnėmis.

Modelių (bei kitų esybių) tapatumas yra fiksuojamas tuomet, kai atitinka esybių pavadinimai. Būtų netikslu jei tikrinimas būtų atliekamas visoje esybių (objektų/procesų/būsėnų) aibėje, tačiau esmė yra tai, kad tai vykdoma tik atitinkančiose struktūrose (t.y. identifikuotame grafo lygyje).



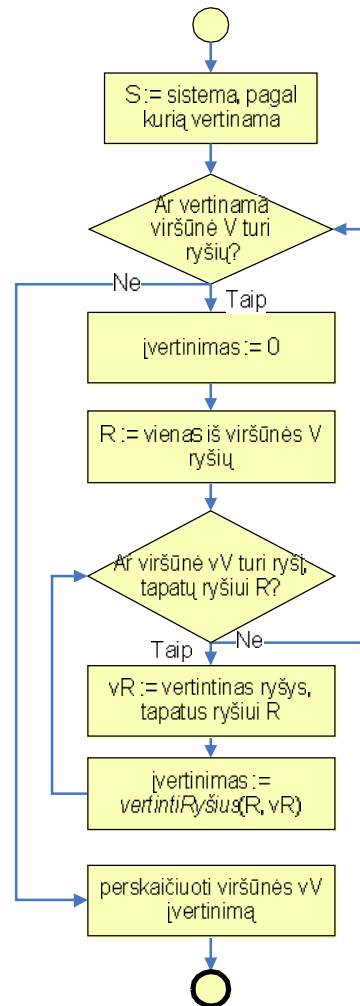
2.22 pav. Modulių sutapatavimo metodo blokinė schema

Šiame etape atliekamas esybių elementų informacijos sulyginimas bei pereinama prie modulių (vertintino ir to, pagal kurį yra vertinama) ryšių tyrimo. 2.22 paveiksle atvaizduotas apibendrintas modulių sulyginimas. Lygiai taip pat atrodo procesų ir būsenų informacijos sutapatinimas, tad jų schemas praleidžiamos.



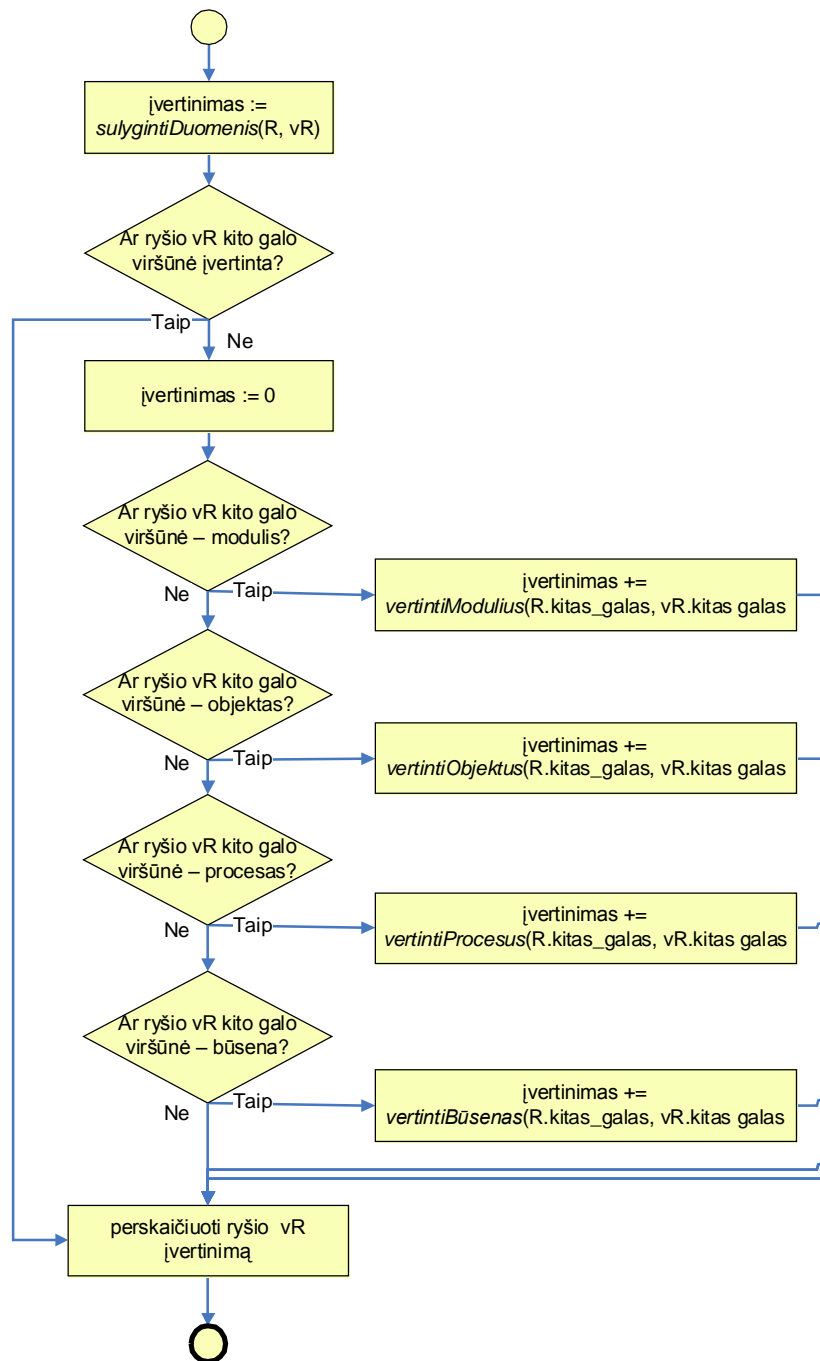
2.23 pav. Objektų sutapatavimo metodo blokinė schema

Objektų sutapatinimas šiek tiek skiriasi nuo kitų esybių sutapatinimo metodų. Tai yra dėl to, kad objektai gali turėti būsenas, tad šių veiksmų sekoje papildomai yra atliekamas būsenų tikrinimas (tapačių būsenų paieška) bei priklausomai nuo rezultato perskaičiuojamas objekto tapatumo įvertinimas.



2.24 pav. Esysės tapačių ryšių paieškos metodo blokinė schema

Šiuo metodu siekiama surasti tapačius ryšius, kurios yra aprašytos abiejose esybėse (grafo viršūnėse). Ryšiai yra tapatūs jei sutampa abiejų galų viršūnės (jų pavadinimai bei pozicija grafe) bei ryšio tipas. Plečiant programos funkcionalumą, galima būtų praleisti lyginimą pagal ryšio tipą, tokiu būdu vartotojas nebūtų „apsunkintas“ tikslaus ryšio pasirinkimu.



2.25 pav. Reikalavimų aprašų sutapatinimo algoritmas (2-a dalis)

Aptikus vienodus ryšius (tokius, kurių pradžios ir galo viršūnės yra panašios), analizuojama jų informacija. Vėliau seka nagrinėjimas viršūnių, kurias jungia modulių ryšiai. Tai atliekama rekursiškai, su paieška ir analize gilyn. Šis procesas vykdomas tol, kol nelieka nė vienos nepatikrintos „darbinės“ sistemos grafo viršūnės ar ryšio. Tačiau gali būti sulyginti ne visos viršūnės, o tuo pačiu ir ryšiai, nes tėvinės objektai (stambiosios sistemos medžio šakos) gali neturėti atitiktens tapatinamos

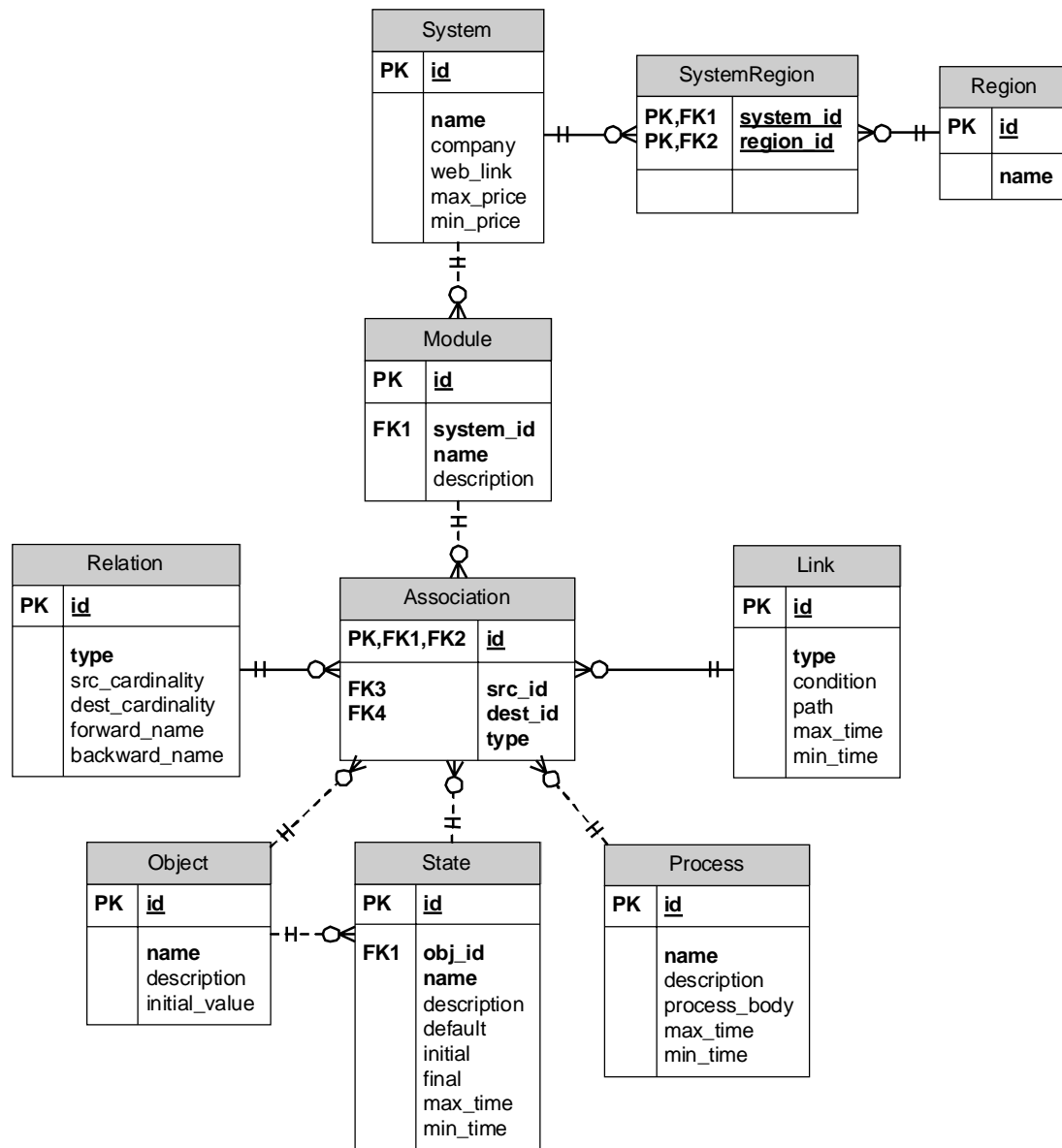
sistemos medyje, tokiu atveju žemesnės, smulkesnės šakos tikrinamos nebus. Tačiau tai vis tiek bus įvertinta.

Pagal modelyje numatytą vertinimo sistemą yra sulyginamas kiekvienas viršūnės ar ryšio elementas (pavyzdžiui, pavadinimas, aprašymas). Yra skaičiuojamas procentinis objektų panašumo įvertinimas. Dažniausiai pasitaikanti situacija gali būti tokia, kad žemiausios viršūnės turės didžiausią įvertinimą, o šakninės viršūnės, o juo labiau ir bendras visos sistemos aprašo įvertinimas bus mažiausias.

2.7. Duomenų bazės schema

Reikalavimų sutapatinimo modelio veikimui reikia suprojektuoti duomenų bazę, kurioje būtų saugomi duomenys bei pagal juos būtų atliekamas anksčiau sumodeliuotų ir vartotojų pateiktų sistemų sulyginimas.

Bus naudojamos reliacinės duomenų bazės, todėl reikia sudaryti lentelių hierarchijas, duomenų saugojimui.



2.26 pav. Bendros duomenų bazės loginė schema

Duomenų bazę sudarančios lentelės yra naudojamos saugoti diagramomis aprašytų reikalavimų specifikacijas, reikalavimus išreikštais diagramų elementais:

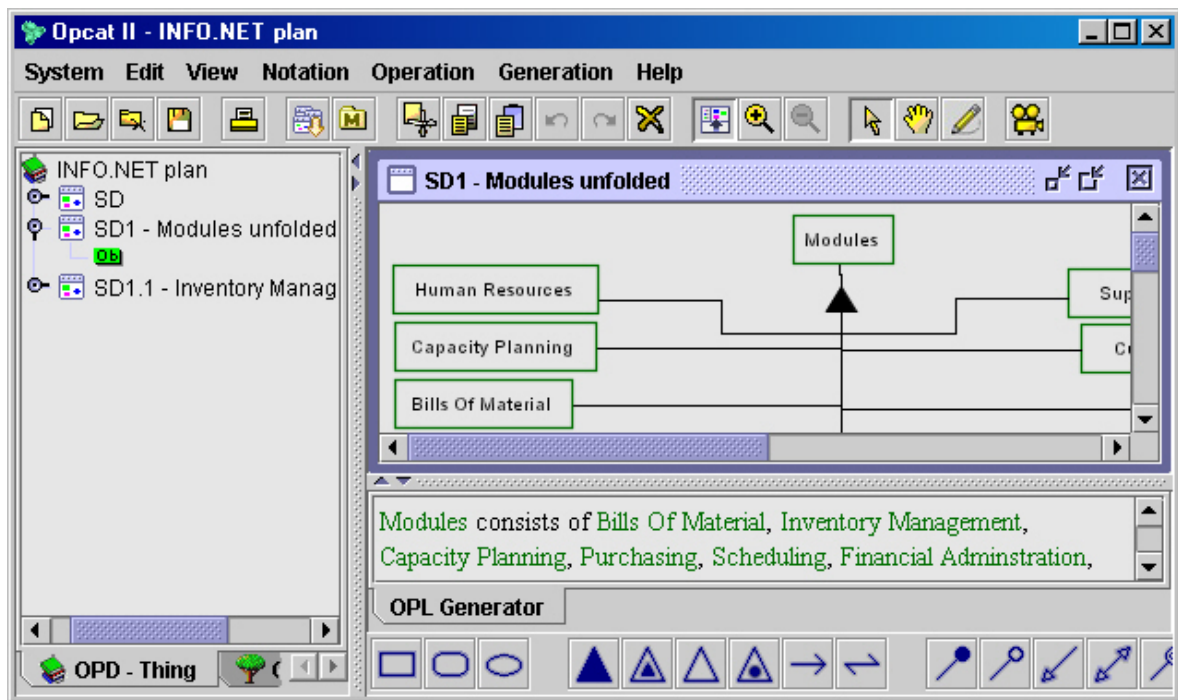
2.4 lentelė.**Duomenų bazės lentelės, jų paskirtis**

DB lentelė	Paskirtis
System	Skirta saugoti bendrinius duomenis apie ERP sistemą.
SystemRegion	Skirta identifikuoti sistemų veikimo geografinius regionus (pavyzdžiui, Concorde sistema gali pilnai tenkinti vartotoją (pagal savo teikiamas funkcijas, dokumentų šablonus ir pan.) Pabaltijo valstybėse, Danijoje ir t.t.).
Region	Skirta saugoti informaciją apie regionus, kuriuose kaip numatyta sistemos kūrėjų sukurtas produktas pilnai atliks savo funkcijas
Module	Saugo informaciją apie ERP sistemų funkcinis blokus-modulius.
Association	Saugomi duomenys yra naudojami sąryšiams tarp esybių (objektų, būsenų, procesų ir modulių) pažymėti.
Relation	Čia saugomi papildomi sąveikų duomenys, kurie papildo lentelėje „Association“ saugomą informaciją apie sąryšius.
Link	Čia saugomi papildomi jungčių duomenys, kurie papildo lentelėje „Association“ saugomą informaciją apie sąryšius.
Object	Naudojama esybių-objektų informacijai saugoti.
Process	Naudojama esybių-procesų informacijai saugoti.
State	Naudojama esybių-objektų būsenoms saugoti.

2.8. Vartotojo sąsajos modelis

Vartotojo sąsajos modelyje pateikiama sistemos vartotojo sąsajos architektūra.

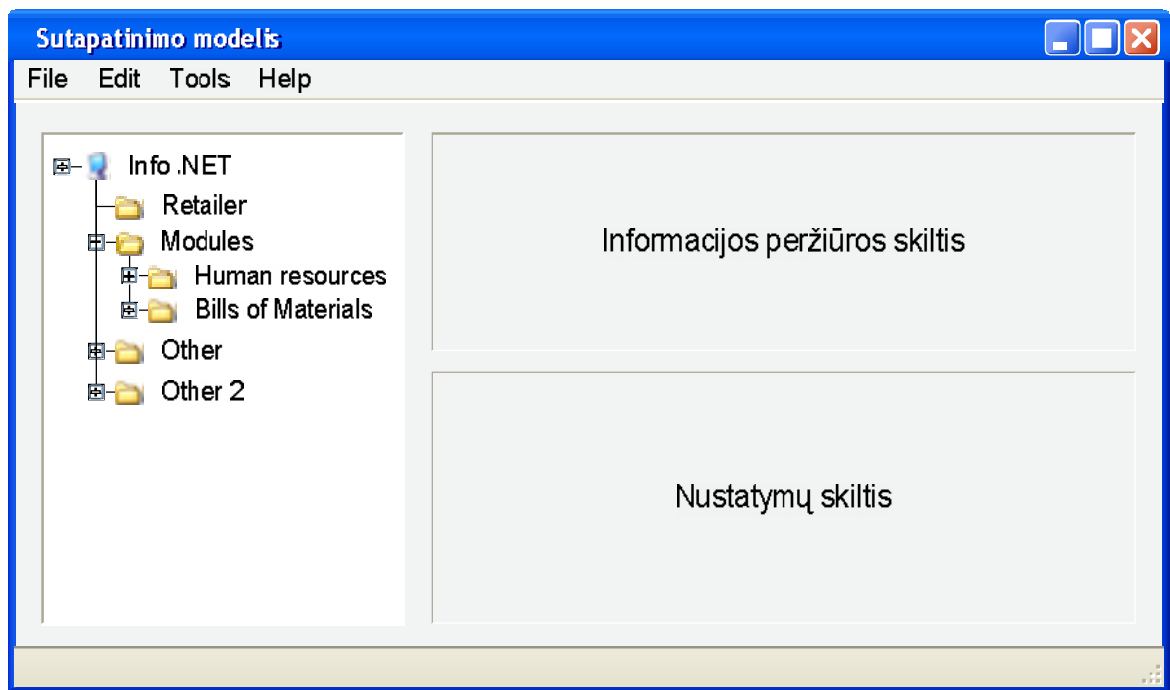
Čia taip pat yra pateikiamas Izraelio technologijos universiteto sukurto programinės įrangos projektavimo įrankio sąsaja.



2.27 pav. OPCAT projektavimo įrankio vartotojo sąsaja

Šis įrankis yra įtraukimas į bendra sistemos darbą – tam pritaikytas duomenų apdorojimo, sutapatinimo modelis bei duomenų saugykla.

Žemiau pateikiamose schemose yra atvaizduojami duomenų apdorojimo, sutapatinimo modelio vartotojo sąsaja.

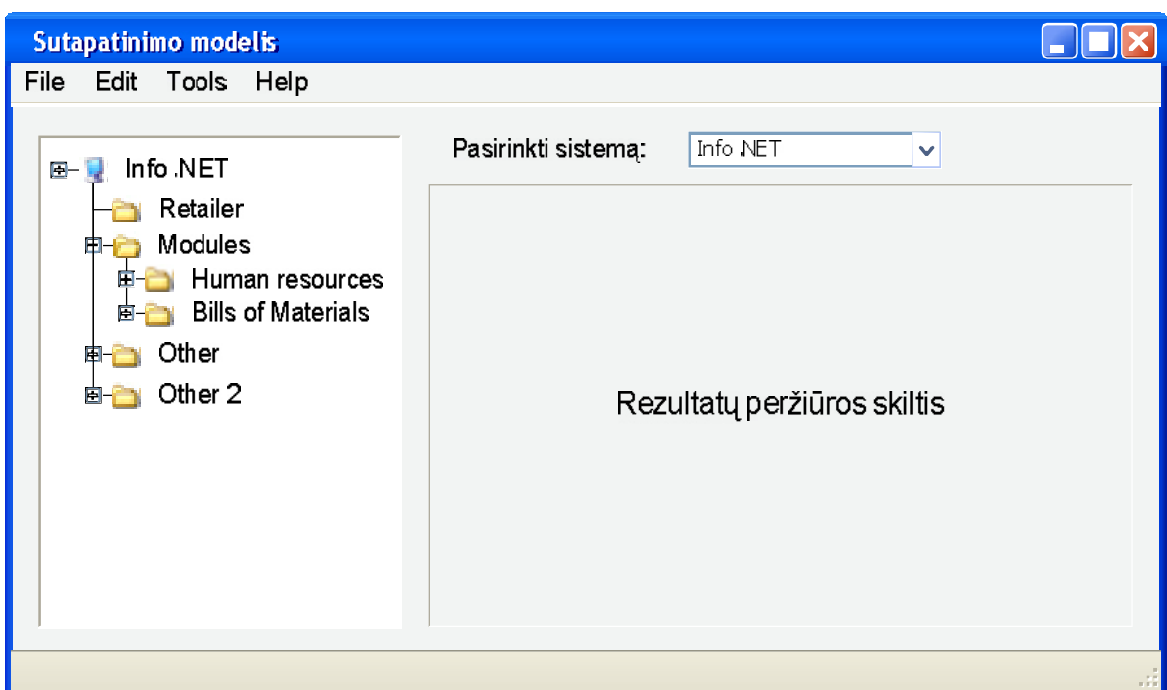


2.28 pav. Suvestų duomenų pateikimo ir įverčių nustatymo langas

The screenshot shows a window titled 'Sutapatinimo modelis' with a menu bar (File, Edit, Tools, Help). The main area contains a table with 6 columns: 'Kriterijai', 'Info .NET', 'Concorde XAL', 'Navision', 'Some other ERP', and 'PeopleSoft'. The text 'Rezultatų peržiūros skiltis' is centered in the table area.

Kriterijai	Info .NET	Concorde XAL	Navision	Some other ERP	PeopleSoft

2.29 pav. Bendrų rezultatų pateikimo langas



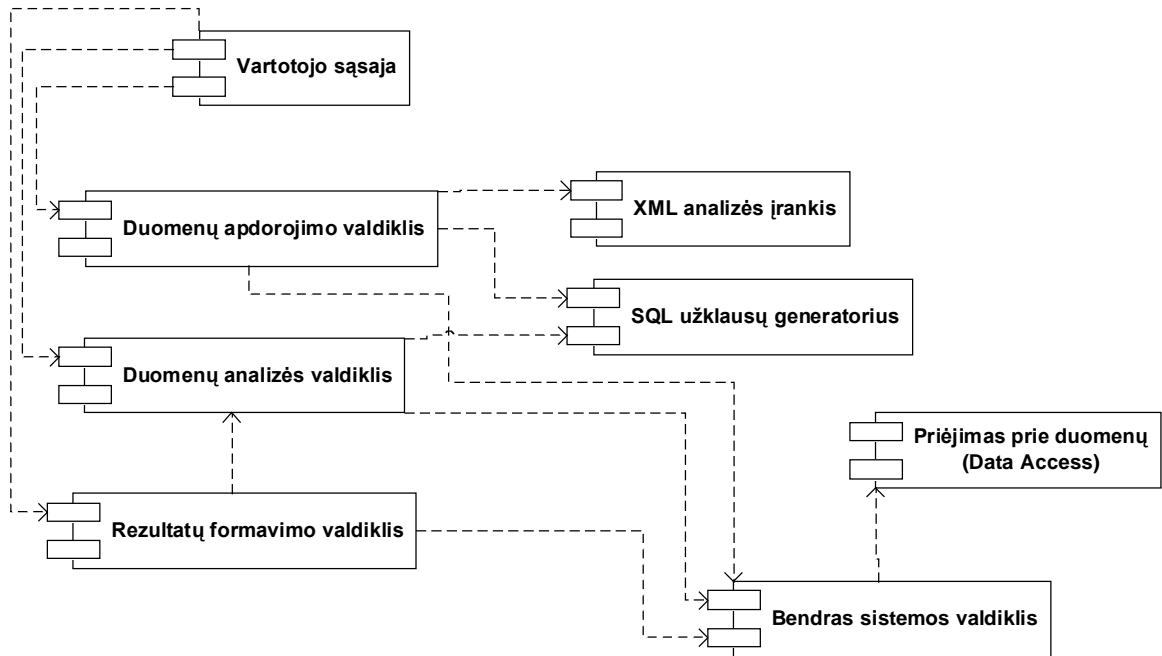
2.30 pav. Sutapatintų rezultatų pateikimo langas

Numatoma, kad rezultatų bus galima peržvelgti tiek sugretinant visas gautas tinkamas ERP sistemas kartu, tiek jas rodant atskirai. Sutapatinimo sistema pradžioje pateikia bendrą statistinį rezultatų dokumentą, kuriame atspindi kiek kiekviena ERP sistema tenkina rodomas funkcijas ar detales apie pačią ERP sistemą. Detalesnis analizės dokumentas yra pateikiamas vartotojui

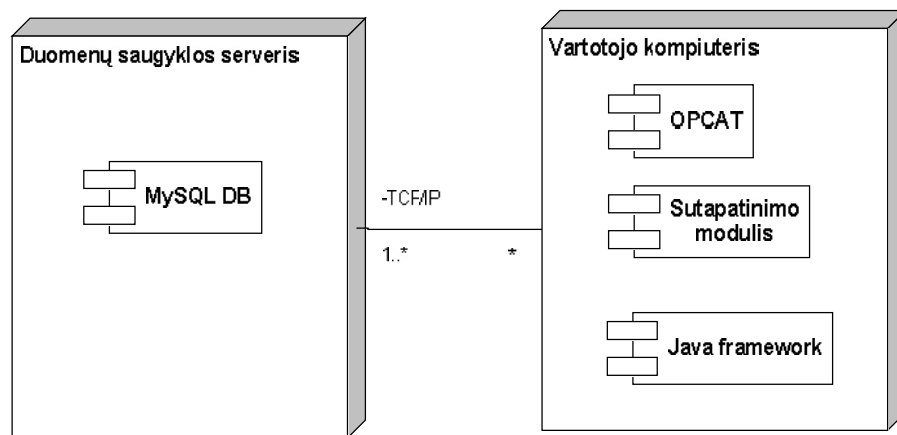
pageidaujant ir pasirinkus šią funkciją. Pagrindinis sistemos valdymas atliekamas meniu pagalba (išskyrus vertinimo dydžių nustatymą).

2.9. Sistemos realizacijos modelis

Realizacijos modelyje pateikiamos sistemos komponentų ir sistemos diegimo diagramos:



2.31 pav. Reikalavimų sutapatinimo sistemos komponentų modelis



2.32 pav. Informacinės sistemos diegimo diagrama

Šiose sistemos komponentų ir diegimo diagramose gerai atsispindi visa sistemos architektūra tiek iš aparatūrinės, tiek iš programinės pusės. Ši informacija yra naudinga IS projektuotojams.

2.10. Sistemos naudojimo instrukcija

Vartotojui, norinčiam naudotis šio projekto rezultate sukurtu modeliu, patartina naudoti autoriaus sudarytą šabloninį reikalavimų aprašymo failą, skirtą naudojimuisi OPM metodologijos principais veikiančia PĮ projektavimo įrankiu OPCAT. Taip galima būtų sumažinti galimų klaidų skaičių.

Reikalavimų schemas sudarymo metu reikėtų naudoti pagalbinę, papildomą sistemos dalį – tai sąvokų žodyną. Tai atskira programa, leidžianti vartotojui pasitikrinti sistemų aprašų duomenų bazėje saugomų objektų pavadinimus. Šį žodyną reikėtų naudoti dėl to pačios priežasties, kad būtų galima sumažinti galimą klaidų kiekį, susidariusį dėl nevisai tiksliai įvestų terminų, kadangi sistemos darbas – paieška, sulyginimai – yra atliekami su objektų pavadinimais.

Taip pat vartotojas gali susiderinti sistemos darbą pagal savo pageidavimus, tokiu atveju jam reikėtų perimti programuotojo funkcijas. Čia yra galimybė pakeisti normalizuotinių simbolių aibes, kurias modelis naudoja užkraudamas iš failo sistemos funkcionalumo / vartotojų reikalavimų aprašą bei jį analizuodamas.

2.11. Projekto išvados

Atlikus vartotojų reikalavimų ir ERP programinių paketų funkcionalumo schemų sutapatavimo sistemos projektavimą galima daryti tokias išvadas:

- Projektuojamoji sistema yra gana sudėtinga, jos veikimui galima pritaikyti įvairius modulius (tokius kaip, rezultatų skaičiavimas pagal vartotojo nustatytus įverčius ir pan.). Juos realizuojant yra palikta galimybė toliau vystyti šią informacinę sistemą – leista plėsti.
- Sistemą realizuota ne kaip viena visuma, o kaip komponentinis junginys. Tai yra patogiau testuojant sistemą – greičiau randamos ir pašalinamos klaidos. Dekompozicijos metu gauti komponentai gali būti panaudojami kuriant kitą programinę įrangą.
- Atliktas OPM metodologija veikiančio PĮ kūrimo priemonės OPCAT tyrimas. Tirti duomenų aprašymo būdai ir galimas jų panaudojimas darbui su kitomis programomis.
- Po reikalavimų specifikacijų ir įrankio OPCAT nagrinėjimo, buvo nuspręsta ir suformuota loginė struktūra, kuria (jos objektais ir tarpusavio ryšiais) būtų patogu naudotis realizuotoje sistemoje.
- Suprojektuota sistema, automatizuosianti ERP sistemų funkcionalumo schemų sutapatinimą su vartotojų pateiktomis reikalavimų schemomis.
- Sukurtas reikalavimų aprašų sutapatinimo algoritmas. Pritaikytas kuriamam modeliui ir įtrauktas į jos darbą.

3. Eksperimentinis reikalavimų sutapatinimo modelio tyrimas

3.1. Eksperimentinis sukurto modelio diegimas

Šio magistrinio darbo metu visuose etapuose, kurių metu planuotas reikalavimų sutapatinimui tinkamas metodas, galimas jį išpildantis modelis, atliktas struktūrų ir jų tarpusavio logikos projektavimas bei realizacija, buvo naudojamosi Windows operacine sistema bei joje veikiančia programine įranga. Taigi pats prototipinis modelis yra pritaikytas veikti Windows aplinkoje.

Sistemos funkcionavimui reikia, kad kompiuteryje būtų įdiegta Java programavimo kalbą (Java2 1.4 versija arba naujesnė) palaikanti aplinka. Čia vartotojui tereikia atlikti standartinį Java aplinkos diegimą, kuris nereikalauja papildomų pastangų.

Toliau reikia įdiegti OPM metodologijos principais veikiančią programinę įrangą OPCAT, kuri galima parsisiųsti iš Izraelio mokslininko D.Dori svetainės [9].

Magistriniam darbe realizuotą reikalavimų sutapatinimo modelį, galima nukopijuoti į bet kurią pageidaujamą vietą diske. Vertėtų atkreipti dėmesį į tai, kad naudojimui reikės ne tik modelio paleidžiamosios bylos, bet ir kelių kitų – tai prisijungimo prie DB, normalizavimo taisyklių informaciją saugančių laikmenų. Jos turėtų būti patalpintos tame pačiame sistemos kataloge kaip ir sutapatinimo programa.

Kad sistema pilnai funkcionuotų ir apdorotus duomenis būtų galima išsaugoti reikia įdiegti darbui su duomenų bazėmis naudojamą sistemą MySQL. Galima atlikti papildomą MySQL sistemos konfigūravimą tam, kad apsaugoti saugomus duomenis nuo poveikio iš išorės.

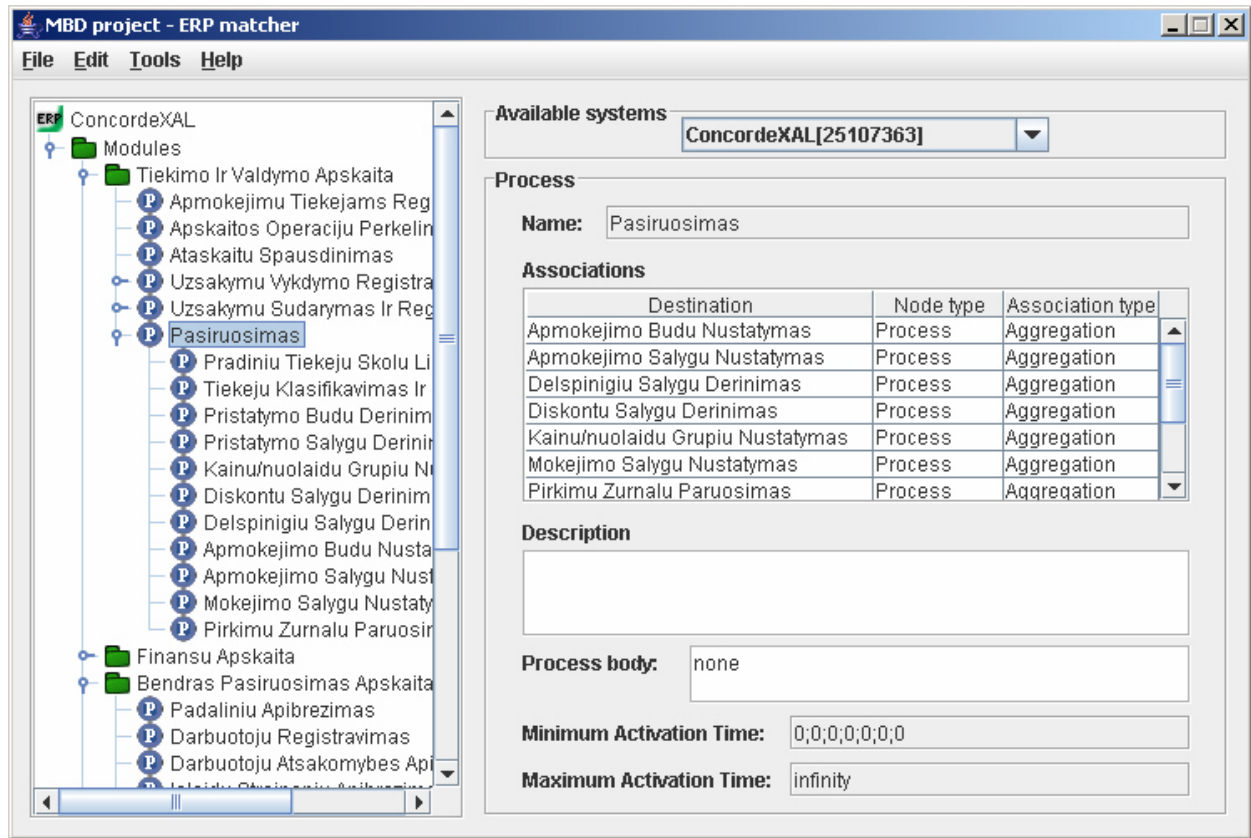
Tai atlikus turėsime veikiančią reikalavimų sutapatinimo modelį, kuriuo naudojantis galima atlikti tinkamos ir organizacijos reikalavimus išpildančią ERP sistemos paiešką.

3.2. Sistemos naudojimo instrukcija

Vartotojas (o tiksliau analitikas) pradėdamas tinkamos ERP sistemos paiešką pirmiausia turi susipažinti su OPM metodologija, jos notacijomis bei sąvokomis. Po to pasinaudodamas autoriaus pateiktu pavyzdžiu-šablonu sudaryti organizacijos poreikius – reikalavimus, keliamus ieškomai sistemai. Sudarinėjant diagramas reikia stengtis, kad aprašomos ERP sistemos funkcionalumas / organizacijos reikalavimai sudarytų grafo tipo struktūrą. Tai patartina dėl to, kad galima būtų identifikuoti sistemą, jos elementus pagal jų esamą poziciją sistemos struktūroje, kad per klaidą nebūtų sutapatinami du panašūs, bet logiškai nesutapatinami elementai.

Sudarius reikalavimų schemą, ši turėtų būti išsaugoma „Opcat2 XML“ formato faile, su galūne *.opx. Iš tiesų kaip jau ir minėta 2.4.1 dalyje, tai gana patogios struktūros sistemų aprašams saugoti.

Toliau paleidžiama magistrinio darbo metu sukurta programa ir pasirenkamas meniu punktas „Nuskaityti sistemos duomenis iš failo“. Tuomet yra vykdoma XML failo analizė – perkeliant duomenis iš failo į kompiuterio operatyviają atmintį.



3.1 pav. Modelyje atvaizduotas sistemos funkcionalumo medis

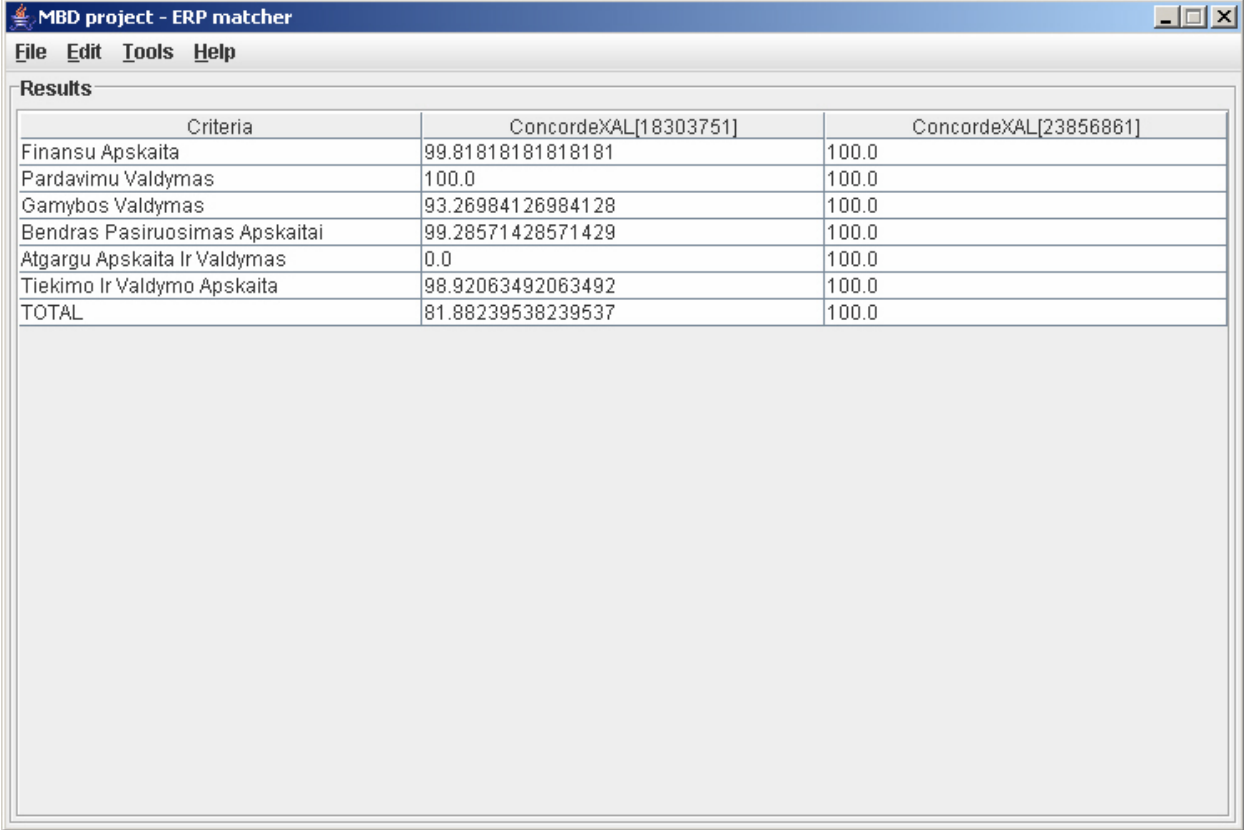
Vartotojui pateikiamas apdorotos sistemos medis, kurį galima peržiūrėti. Moduliai / objektai / procesai / būsenos vaizduojamos atitinkamais simboliais kairiajame lange. Šioje struktūroje nesimatys ar yra ciklinių ryšių – tai galima atsekti tik peržvelgus kiekvienos esybės elementus. Dešinėje lango pusėje pateikiamos į operatyviają atmintį užkrautų sistemų pavadinimai bei informacija apie pasirinktas medžio struktūros elementus.

Spragtelėjus ant ryšių lentelės vartotojui bus pateikta informacija apie konkretų pasirinktą ryšį.

Norint išsaugoti sistemos funkcionalumo aprašą duomenų bazėje reikia pasirinkti meniu punktą „Saugoti sistemą duomenų bazėje“. Tuomet į duomenų bazę bus išvesta informacija apie tuo metu aktyvią (vaizduojamą) sistemą.

Pagrindinis punktas yra „Tapačios sistemos paieška“. Čia vartotojas turi būti į operatyviają atmintį įkėlęs savo sudarytos diagramos struktūrą. Toliau iš duomenų bazės paieškos skaitomos saugomos sistemos yra tapatinamos su vartotojo pateikta ERP sistema. Atlikus sutapatinimo veiksmus

analitikui pateikiami apibendrinti sutapatinimo rezultatai – duomenų bazėje saugotų sistemų modulių atitikmuo vartotojo pateiktiems moduliams. Tai tik bendrinė informacija.



Criteria	ConcordeXAL[18303751]	ConcordeXAL[23856861]
Finansu Apskaita	99.81818181818181	100.0
Pardavimu Valdymas	100.0	100.0
Gamybos Valdymas	93.26984126984128	100.0
Bendras Pasiruosimas Apskaitai	99.28571428571429	100.0
Atgargu Apskaita Ir Valdymas	0.0	100.0
Tiekimo Ir Valdymo Apskaita	98.92063492063492	100.0
TOTAL	81.88239538239537	100.0

3.2 pav. Sutapatinimo metodas. Apibendrintieji rezultatai

Detalesnius duomenis galima gauti pasirinkus norimos įvertintos sistemos duomenų stulpelį. Tuo būtų analitikas galės pamatyti kiek kiekvienas struktūros elementas atitinka jo sudarytą schemą.

Tolesniems perėjimams tarp įvertintų sistemų schemų yra realizuoti meniu punktas „Rodyti bendrinius rezultatus“ bei sistemų pasirinkimo iš operatyviosios atminties laukelis.

3.3. Sukurto sutapatinimo modelio įvertinimas

Pagal anksčiau iškeltus funkcinis, nefunkcinis reikalavimus bei kokybės kriterijus sukurtą sutapatinimo metodą ir jį išpildantį modelį galima vertinti taip:

3.1 lentelė.

Sutapatinimo modelio kokybės kriterijų įvertinimas

Reikalavimas	Reikalavimo išpildymas	Komentarai
Techninės įrangos pakankamumas	+ / -	Modelis projektuotas, realizuotas ir testuotas keliose tarnybinėse stotyse. Nepastebėta jokio poveikio darbui su kompiuteriais, kurių techninės charakteristikos yra nemažesnės nei apibrėžtos reikalavimų specifikacijose (su silpnesniais kompiuteriais neišbandyta)
Programinės įrangos tinkamumas	+	Projektavimo, realizavimo bei vartojimo metu naudojama programinė įranga yra tinkama ir suderinta su sukurtu modeliu.
Vartotojo sąsajos kalba	+	Realizuota anglų kalba pasirinktinai iš anglų ir lietuvių.
Tinkamumas	+	Sukurta sistema, sutapatinimo modelis yra tinkama reikalavimų sutapatinimo problemai spręsti. Rezultate pateikia informaciją apie sulyginamų sistemų tapatumą, kuris išreiškiamas procentinėmis išraiškomis. Rezultatą, jo forma galima keisti realizuojant kitas sutapimo įvertinimo skaičiavimo metodikas (pavyzdžiui, nurodant kiekvienos esybės elemento svarbumo laipsnį).
Tikslumas	+ -	Modelis gali operuoti su gana dideliu tikslumu, nes apibrėžtos duomenų struktūros bei algoritmai leidžia dirbti nepriklausomai nuo duomenų dydžio. Tapatumo įvertinimo skaičiavimo algoritmas yra tikslus lyginant tik duomenų grafų struktūrų sutapimą, tačiau netinka lyginti grafų iš biznio logikos pusės.
Užbaigtumas	+	Sukurta sistema yra pakankamai išbaigta. Numatyti galimi klaidų atvejai.
Tolerancija klaidoms	+	Susidarius klaidai sistema toliau dirba, neužstringa ir apie ją informuojamas vartotojas.

3.1 lentelė (tęsinys).

Sutapatinimo modelio kokybės kriterijų įvertinimas

Patogumas	+ / –	Reikia nemažos analizės, kruopštumo sudarant ERP sistemų funkcionalumo ar organizacijos iškeltų reikalavimų aprašus. Tačiau darbas su sutapatinimo modeliu yra suprantamas ir nesudėtingas– čia aiškiai pateikiama supaprastinta diagramų schema (grafo pavidalu) bei duomenys apie jos elementus. Veiksmų seka yra sudaroma modelio pagalba, tad vartotojas nepaspaus kur nors netyčia ir ne vietoj. Rezultatai yra pateikiami sistemos ekranuose. Jų pateikimo formą galima praplėsti.
Išteklių naudojimas	+ / –	Gali tekti apriboti modelio operatyvinėje atmintyje saugomos informacijos kiekį ir apimtis, jei dirbant su dideliais duomenų grafais paaiškės, kad sistema naudoja per daug išteklių.
Praplečiamumas	+	Modelis objektiškai orientuotas, palikta galimybė plėsti.
Nekorektiški duomenys	+ / –	Naudoti reikalavimų-sąvokų žodyną.
Auganti sistemos kaina ir jos kūrimo laikas	+ / –	Sukurtas mažesnis sistemos variantas, tenkinantis iškeltą projekto tikslą.
Išsamumas	+	Išpildyti visi iškelti reikalavimai sistemai bei jos darbui.

Sukurtas metodas ir modelis išpildo visus iškeltus reikalavimus.

Lyginant sukurtą modelį su rinkoje esančia „ERP Evaluation Center“ sistema, galima teigti, kad suprojektuota bei sukurta sistema yra tikslesnė, nes gali dirbti su daug detalesniais duomenimis bei naudoti norimą, realizuotą sistemų tapatumo įvertinimo algoritmą. Tuo tarpu Internete veikianti sistema tiesiog pateikia rezultatus (rekomendacijas) apie tinkamą sistemą iš vartotojo apklausos, kurią sudaro vos 40 klausimų.

IŠVADOS

Atliekant šį darbą buvo padarytos tokios išvados:

- ERP – brangios, sudėtingos sistemos, kurių pritaikymas prie įmonių keliamų reikalavimų sudaro dideles išlaidas bei pastangas. Tačiau tai daryti verta, kadangi šių sistemų poreikis yra tikrai didelis. Kad pasirinkti tinkamiausią sistemą, reikėtų naudoti kokią nors analitinę priemonę, kuri galėtų įvertinti, kiek įmonės poreikių tenkina viena ar kita pasirinkta sistema.
- Šaltinių analizės metu plačiausiai nagrinėtas teorinis mokslininkų C.Rolland ir N.Prakash siūlytas ERP reikalavimų pritaikymo modelis. Kaip paaikškėjo vėlesniais šio darbo etapais – šis modelis tapo pagrindu kurto programos prototipo struktūrai ir elgsenai.
- Praktinis modelis „ERP Evaluation Center“ taip pat davė idėjų sistemos vystymui, tačiau šio modelio sulyginamoji-įvertinamoji specifiika yra visai nepanaši į šiame magistriniame darbe atliekamą įvertinimą. Tai laisvai Internetu prieinama sistema, kurią galima nemokamai išbandyti. Ištestavus jos laisvai teikiamas galimybes galima teigti, kad „ERP Evaluation Center“ duomenimis operuoja ir rezultatus pateikia tik labai abstrakčiame duomenų lygyje, nepakankamai juos detalizuojant.
- Atlikta realizacijos bei modelio naudojimo priemonių paieška ir analizė. Realizacijai pasirinkta Java kalba, puikiai deranti duomenų saugykla MySQL, Izraelio mokslininkų sukurta ir OPM metodologijos principais veikianti PĮ kūrimo priemonė OPCAT ir kt. Renkantis naudojamą programinę įrangą stengtasi išlaikyti bendrą kuriamo modelio vientisumą, nedalumą.
- Suprojektuotos ir sutapatavimo modelyje realizuotos duomenų struktūros, leidžiančios atvaizduoti ERP sistemos funkcionalumą kaip jungtinių grafą. Duomenų struktūros apibrėžia tam tikras duomenų aprašymo taisykles.
- Nustatytas sistemos veiklos modelis, atskleidžiantis sistemos veiksmų vieno nuo kito priklausomybę bei eiliškumą. Išskirtos pagrindinės sistemos veiklos dalys.
- Sudaryti algoritmai, kurių pagalba galima dirbti su pateikta informacija – ERP sistemos funkcionalumo ar organizacijos reikalavimų aprašu. Nuskaitant XML formatu pateiktus duomenis naudojamosi pradinių duomenų skaitymo algoritmu. Taip pat sukurti algoritmai darbui su duomenų baze – grafo duomenų ir struktūros saugojimui bei skaitymui iš duomenų saugyklos. Vartotojas-analitikas, siekdamas rasti tinkamiausią įmonių išteklių valdymo sistemą, turi naudotis reikalavimų aprašų sutapatavimo algoritmu, kuris šią problemą ir sprendžia.

- Suprojektuotas reikalavimų sutapatinimo modelis, išskiriant esmines sistemos dalis. Apibrėžtos tų dalių funkcijos, struktūros bei tarpusavio priklausomybės. Modelis realizuotas laikantis sudarytos architektūros.
- Suprojektuota, realizuota ir modelio darbo metu panaudota duomenų bazių lentelių struktūra, leidžianti į duomenų bazę perkelti grafo tipo duomenis, išlaikant jų informaciją bei susietumą.
- Kurtas modelis yra komponentinis junginys, bet ne vientisa visuma. Tai leidžia greičiau aptikti netikslumus, atlikti pakartotiną komponentų panaudojimą. Sistema yra „draugiška“ kitiems komponentams. Tikslas – tai ją padarys patrauklesnę vartotojams.
- Realizuotame modelyje panaudotas procentinis sistemų įvertinimas. Tačiau vėliau vartotojui galima būtų sudaryti galimybę pasirinkti sistemų tapatumo įvertinimo skaičiavimo algoritmą. Pavyzdžiui, vartotojas galėtų nustatyti įverčius, žyminčius kiek kuris objektas ar komponentas yra svarbus.
- Sukurto prototipo testavimo metu išbandyta ir patvirtinta jo savybė – stabilumas. Sistema dirba patikimai bei yra tolerantiška klaidoms, jei tokios iškyla, informuodama apie jas vartotoją. Taip pat galima teigti, kad sudarytos duomenų struktūros yra tinkamos darbui su grafu aprašytais duomenimis. Sistemos architektūra patogi darbui su šia informacija.
- Atlikus realizuoto modelio įvertinimą galima tvirtinti, kad išpildyta dauguma iškeltų reikalavimų.
- Šio darbo straipsnis buvo paskelbtas 2005 balandžio 29 dieną vykusioje 10-ojoje tarpuniversitetinėje magistrantų ir doktorantų konferencijoje „Informacinės technologijos'05“.

Šis darbas atitinka šiuolaikinius ERP sistemų pritaikymo poreikius. Iki šiol rinkoje nebuvo programinių paketų, gebančių sutapatinti sistemas iki pakankamo detalumo lygio. Šio darbo rezultate yra sukurtas reikalavimų sutapatinimo modelis, kurį toliau tobulinant galima būtų realizuoti ir išleisti tai sugebančią atlikti sistemą. Tačiau tam reikia papildomos analizės prijungiant kitas sferas ar idėjas, kurios šiame magistro darbe buvo neanalizuotos ir neįtrauktos. Manau, kad gautume patrauklią vartotojams sistemą, leisiančią sumažinti įmonių išteklius ieškant ir renkantis naujas įmonių išteklių valdymo programas.

LITERATŪRA

- [1] C. Rolland, N. Prakash. Matching ERP System Functionality to Customer Requirements. IEEE, Toronto, Canada, psl. 66-75, 2001. [žiūrėta 2005.03.13]. Prieiga per Internetą: <http://doi.ieeecs.org/10.1109/ISRE.2001.948545>
- [2] C. Rolland, N. Prakash. Bridging the Gap Between Organisational Needs and ERP Functionality. 2000. [žiūrėta 2005.03.12]. Prieiga per Internetą: http://www.ida.his.se/~anne/CR_ERP.pdf
- [3] TechnologyEvaluation.Com. ERP Evaluation Center. [žiūrėta 2004.12.20]. Prieiga per Internetą: <http://www.erp-software-comparison.com>
- [4] TechnologyEvaluation.Com. Frequently Asked Questions. [žiūrėta 2005.04.06]. Prieiga per Internetą: <http://www.erpevaluation.com/popups/faq.asp>
- [5] AMOS. A gentle introduction to AMOS. [žiūrėta 2005.09.23]. Prieiga per Internetą: http://clip.dia.fi.upm.es/~amos/AMOS/amos_gentle.html
- [6] C.Daffara. The AMOS Project. Automated Matching of Open Source Code. Free Software and Research, Soissons, France, 2003. [žiūrėta 2005.09.23]. Prieiga per Internetą: <http://clip.dia.fi.upm.es/papers/FSR2003-amos.pdf>
- [7] M.Carro, J.Correas, J.Gómez. The AMOS Project. The Internal Query Language Design. 2002. [žiūrėta 2005.09.23]. Prieiga per internetą: http://www.amosproject.org/Papers/internal_QL_design.pdf
- [8] M.Carro. The Amos Project: an Approach to Reusing Open Source Code. Madrid, Spain, 2003. [žiūrėta 2005.09.24]. Prieiga per Internetą: <http://clip.dia.fi.upm.es/papers/amos-cbd.pdf>
- [9] Object-Process Methodology. [žiūrėta 2005.03.10]. Prieiga per Internetą: <http://www.objectprocess.org>
- [10] OMG. Unified Modeling Language (UML) 1.5 Specification. 2005. [žiūrėta 2005.04.15]. Prieiga per Internetą: <http://www.omg.org/technology/documents/formal/uml.htm>
- [11] I. Reinhartz-Berger, D. Dori. OPM vs. UML - Experimenting with Comprehension and Construction of Web Application Models. Empirical Software Engineering 10, 1, psl. 57-80. 2005. [žiūrėta 2005.05.20]. Prieiga per Internetą: <http://dori2.technion.ac.il/opm/documents/articles/article194.pdf>
- [12] D. Dori, D., I. Reinhartz-Berger, A. Sturm. OPCAT – A Bimodal Case Tool for Object-Process based system development. 5th International Conference on Enterprise Information Systems, psl. 286-291. 2003. [žiūrėta 2005.03.21]. Prieiga per Internetą: <http://dori2.technion.ac.il/opm/documents/articles/article74.pdf> Programinę priemonę galima parsisiųsti iš Interneto: <http://www.technion.ac.il/~erant/opcat/>

- [13] L. Nemuraitė. Vadybos kompiuterizuotos darbo vietos. Kaunas: Technologija, 1999. 232.
- [14] P. Žydelis. ERP sistemų funkcionalumo ir vartotojų reikalavimų sutapatavimo modelis. Informacinės technologijos. 10-oji tarpuniversitetinė magistrantų ir doktorantų konferencija. Konferencijos pranešimų medžiaga. Kaunas: Technologija, 2005. 232, 134-137 p.

TERMINŲ IR SANTRUMPŲ ŽODYNAS

- Apache – HTTP serveris.
 - CASE priemonės (angl. *Computer Aided Software Engineering Tools*) – programinės įrangos projektavimo priemonės.
 - DB – duomenų bazė, duomenų saugykla.
 - ERP sistema (angl. *Enterprise Resource Planning System*) – įmonių išteklių valdymo sistema.
 - Java – objektiškai orientuota programavimo kalba.
 - JDBC – Java ir duomenų bazių sąsajos tvarkyklė.
 - MySQL – duomenų bazės valdymo sistema.
 - OPCAT (angl. *Object-Process Case Tool*) – programinės įrangos projektavimo įrankis, skirtas darbui pagal OPM metodologiją.
 - OPM (angl. *Object-Process Metodology*) – objektų-procesų metodologija.
 - Windows – asmeninio kompiuterio operacinė sistema, turinti grafinę sąsają.
 - XML (angl. *Extended Markup Language*) – interneto dokumentų žymėjimo kalba, galinti atlikti daugiau su verslu susijusių funkcijų nei HTML.
 - Modulis – esybė, atspindinti ERP sistemos sritį, kuri nagrinėja tam tikrą biznio sritį.
- Pavyzdžiui: finansų apskaita, žmogiškieji resursai, ..
- Objektas – esybė, kuris tam tikrą laiko tarpą egzistuoja fizinėje ar loginėje plotmėje.
 - Procesas – veiksmas ar jų visuma, sukuriantis arba sunaikinantis objektą, arba keičiantis objekto būseną.
 - Būsena – Situacija, kurioje gali būti objektas.

PRIEDAI

1 PRIEDAS. Paskelbtas magistro darbo straipsnis

Čia pateikiamas autoriaus straipsnis, kuris buvo paskelbtas 2005 m. balandžio mėn. 29 d. vykusioje tarpuniversitetinėje magistrantų ir doktorantų konferencijoje „Informacinės technologijos’05“ [14].

ERP SISTEMŲ FUNKCIONALUMO IR VARTOTOJO REIKALAVIMŲ SUTAPATINIMO MODELIS

Paulius Žydelis

*Kauno technologijos universitetas, Informacijos sistemų katedra, Studentų g. 50, 51368 Kaunas
(Vadovas: doc. Rimantas Butleris)*

Straipsnyje nagrinėjamos įmonių išteklių valdymo sistemų bendrosios savybės, problemos, susidarantių sistemų diegimo ir naudojimo metu. Analizuojama tokio tipo sistemų funkcionalumo specifikacijų sutapatinimo su vartotojų keliamais reikalavimais galimybė. Remiantis esamais problemos sprendimo būdais kuriamą naujos, vartotojams pagalbą teikiančios sistemos vizija.

Straipsnyje naudojamos santrumpos:

- ERP sistema – įmonių išteklių valdymo sistema (angl. Enterprise Resource Planning System);
- CASE priemonės – programinės įrangos projektavimo priemonės (angl. Computer Aided Software Engineering Tools).
- OPM – objektų procesų metodologija (angl. Object-Process Methodology).

1 Įvadas

Šiuo metu rinkoje populiarėja komerciniai produktai – tokie kaip įmonių išteklių valdymo (ERP) sistemos.

ERP – tai verslo strategija, kurioje integruotos išteklių valdymo, finansų, prekybos bei kitos funkcijos, padedančios stebėti ir optimizuoti įmonių resursus. Šios sistemos sukurtos tobulinant gamybos įmonių poreikių valdymo sistemas (angl. Manufacturing Requirements Planning (MRP)) dėka atsiradusių reliacinių duomenų bazių valdymo sistemų, CASE priemonių, ketvirtos-kartos (4GL) kalbų vystymo įrankių ir kliento/serverio architektūros. Įdiegus ERP sistemas, įmonės gali sklandžiau reguliuoti savo veiklos procesus, greičiau ir efektyviau atlikti reikalingas analizes ir gauti atitinkamus rezultatus. Šios sistemos įmonėms leidžia stebėti ir neatsilikti nuo rinkos, įvertinant joje atsiradusius pokyčius.

Tačiau tokių sistemų panaudojimas turi aiškę trūkumų – tai didelė kaina, sudėtingas bei daug laiko užimantis realizavimas, sistemos atnaujinimo būtinybė, didelė tarptautinių nesuderinimų tikimybė ir panašiai.

Siekiant sumažinti šių trūkumų kiekį, orientuotasi į vieną iš sričių – šių sistemų sklandų pritaikymą prie vartotojų keliamų reikalavimų. Čia išskyla poreikis išnagrinėti žinomų (prieinamų) ERP sistemų funkcionalumo specifikavimo metodus bei jų diegimo instrukcijas, o taip pat atlikti ir vartotojo reikalavimų analizę.

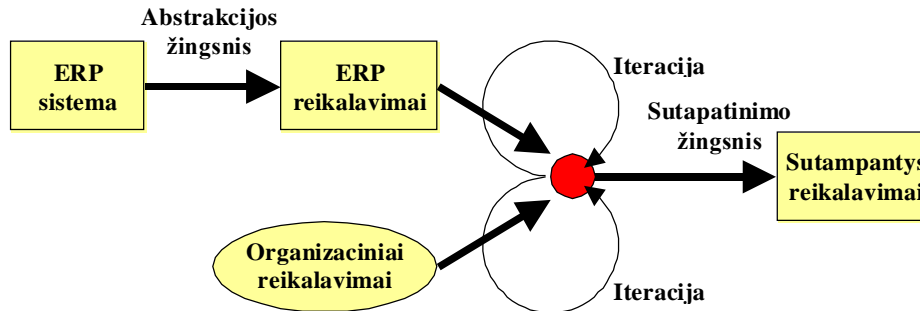
Ši sritis jau buvo nagrinėta užsienio autorių, tačiau tik teoriškai. Taigi, pasinaudojus autorių anksčiau iškeltomis teorijomis bei siūlomais problemos sprendimo metodais, siekiama sudaryti patikslintą ERP sistemų funkcionalumo specifikavimo metodą ir sukurti jo programinį prototipą. Sukurta sistema užtikrintų galimybę kompiuterizuotu būdu atlikti ERP sistemos funkcionalumo ir vartotojo reikalavimų sutapatinimą. Iki šiol tai buvo atliekama tik teoriniame lygmenyje.

2 ERP sistemų pritaikymo organizacijos reikalavimams principų analizė

Nemažai autorių savo literatūroje atkreipė dėmesį į ERP sistemų funkcijų ir vartotojų reikalavimų nesutapatimą, tačiau retas iš jų siūlo kokių nors metodinių priemonių šiai problemai spręsti.

Toliausiai šioje srityje pažengė C.Rolland ir N.Prakash. Šių autorių siūlomas sprendimas [1,2] pasižymi tuo, kad sistemų diegimas būtų orientuotas į vartotojų keliamų reikalavimų, o ne į sistemų funkcinę sferą, kaip tai daro ERP ekspertai. Kad tai būtų galima realizuoti, siūloma sukurti ERP pritaikymo modelį, kurį sudarytų egzistuojančių ERP sistemų funkcinių galimybių bei joms priskirtų reikalavimų specifikacija. Šiame modelyje turėtų būti įdiegtas toks funkcionalumas:

- kuriami asociatyviniai ryšiai tarp ERP programinės įrangos sisteminių funkcijų bei keliamų reikalavimų;
- atliekamas ERP sistemos ir vartotojo reikalavimų sutapatinimas;
- automatizuojamas perėjimas nuo vartotojo poreikių prie ERP funkcionalumo savybių, naudojant sudarytus “reikalavimų-funkcionalumo” ryšius.



1 pav. C.Rolland ir N.Prakash siūlomas ERP reikalavimų pritaikymo modelis [1, 2]

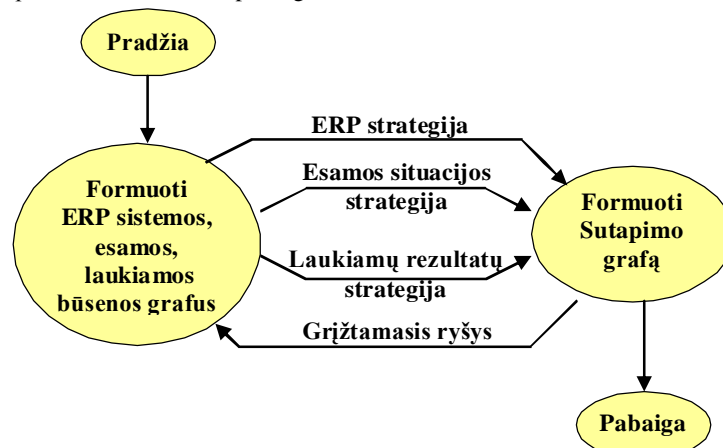
1 paveiksle pavaizduotas ERP reikalavimų pritaikymo modelis. Pirmuoju žingsniu atliekamas sistemų funkcionalumo konkretizavimas sudarant reikalavimus, kurių siekia pati sistema. Antrasis žingsnis - tai sutapatinimo procesas, kuriuo bandoma išskirti ir parinkti reikalavimus, sutampančius tiek iš ERP, tiek iš organizacijos pusės. Sutapatinimo žingsnis yra iteracinis – kiekvieno iteracinio veiksmo metu iš pateiktų reikalavimų aibė yra naudojami tik tie duomenys, kurie dar liko netyrinėti ar nesutapatinti. Šio proceso pabaiga bus laukiamas rezultatas, kai pateikiama reikalavimų aibė, kurių pagrindu bus diegiama ERP sistema.

Surinkti reikalavimai gali būti išreikšiami *Sutapimo grafu*. Šis grafas gali būti sudaromas sugretinant ERP sistemos ir organizacijos keliamus specifinius reikalavimus. Siekiant palengvinti sugretinimo procesą, abiejų pusių reikalavimai gali būti išreikšti grafiškai.

ERP reikalavimų sutapatinimo proceso modelis pateiktas 2 pav. Jį sudaro keturios atskiros dalys, padalintos į dvi sritis. Pradinėje srityje yra talpinami trys surinktų reikalavimų grafai: ERP sistemos, esamos ir siekiamos situacijos organizacijoje grafai:

- esamos situacijos organizacijoje grafas parodo šiuo metu naudojamomis technologijomis pasiektus rezultatus;
- siekiamos situacijos organizacijoje grafas parodo kokių rezultatų organizacija tikisi įdiegus pasirinktą ERP sistemą;
- ERP sistemos grafas parodo sistemos funkcijas, pačios sistemos keliamus reikalavimus.

Rezultatų srityje talpinamas vienas – sutapimo grafas.



2 pav. ERP reikalavimų sutapatinimo proceso modelis

Šie grafai aprašomi trimis žingsniais. Pirmo etapo metu yra sudaromi visi trys pradinės srities grafai. Grafų sudarymas yra pagrįstas kūrimo ir tikrinimo taisyklėmis. Antrojo, sutapatinimo etapo metu sudaromas sutapimo grafas. Pagrindiniai reikalavimai atvaizduoti sutapimo grafе yra išgauti sugretinant reikalavimus, paimtus iš ERP sistemų ir siekiamos situacijos organizacijoje grafų. Kiti, specifiniai įmonės reikalavimai gali būti neišgauti iš ERP sistemos grafo, taigi jie reikalauja specialaus vystymo. Tokiu būdu sutapimo grafas leidžia identifikuoti reikalavimus, dėl kurių

organizacijoms gali tekti papildomai rūpintis, nes tai nebus ERP sritis. Sutapimo grafas gali būti sudaromas viena iš trijų strategijų: ERP, esamų ar laukiamų rezultatų įmonėje strategija. Kiekvienos iš šių strategijų tikslas yra nueiti nuo pradinio iki pabaigos taško, atrinkti sutampančius reikalavimus ir patalpinti juos į sutapimo grafą. Trečiuoju ir paskutiniu žingsniu yra atliekamas sutikrinimas, ar sutapimo grafe patalpinti visi reikalavimai, išskelti *Laukiamų rezultatų grafe*. Taigi, taip užbaigiant šį procesą yra užtikrinamas grafo tikslumas.

3 Kuriamo reikalavimų sutapatinimo modelio vizija

Analizės metu panaudota jau egzistuojanti sistema „ERP Evaluation Center“ [3], kurią vartotojams pateikia *TechnologyEvaluation.Com* grupė. Joje vartotojams pateikiamas klausimynas su atsakymų variantais. Apdorojus duomenis, pateikiamas vartotojų reikalavimus dalinai ar pilnai tenkinančių sistemų sąrašas. Vėliau pasirinkus keletą iš jų yra detalizuojamas kiekvienos sistemos atitiktas pateiktiems reikalavimams.

Tačiau atliktos analizės pagrindu galima daryti išvadą, kad tokių – sistemų funkcinius ir vartotojų reikalavimus apjungiančių – sistemų yra vos kelios ir ši sritis tik plėtojama. Šiame darbe ir siekiama sukurti ERP sistemų funkcionalumo ir vartotojo reikalavimų sutapatinimo modelį bei jo programinį prototipą, skirtą ERP sistemų diegimui palengvinti. Tikslas – patenkinti vartotojus, ieškančius jų organizacijų veiklą tiksliausiai atspindinčias sistemas. Taip jų paieškoms būtų sutaupoma daug laiko bei išteklių. Sukurtas programinis prototipas pagal vartotojų pateiktus reikalavimus turi nustatyti atitinkamas ERP tinkamumą, o kartu ir atitikimą vartotojo reikalavimams, taip sumažinant bereikalingos analizės mastus.

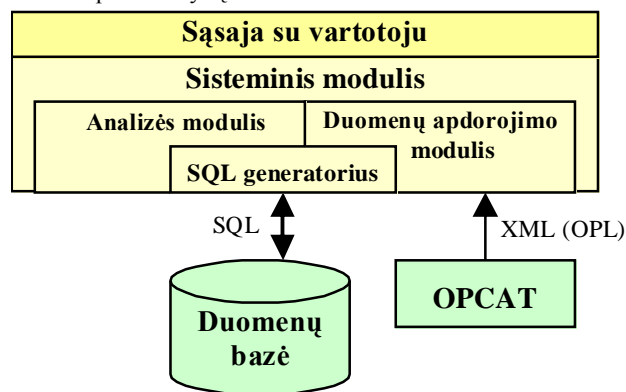
Viena iš planuojamų sistemos realizavimo priemonių yra *Objektų-Procesų metodologija* (OPM) [5]. Jos apibrėžtomis taisyklėmis galima aprašyti reikalavimus. Tai tinka dėl šios modeliavimo kalbos universalumo ir lankstumo – čia remiamasi dviem pagrindiniais elementais: objektais ir procesais. Reikalavimai gali būti aprašyti grafine forma (objektų-procesų diagramomis (OPD)) bei tekstiniu formatu (objektų-procesų kalba (OPL)). OPM yra palaikoma OPCAT *Case* priemone. Šis įrankis leidžia išsaugoti diagramas XML formatu. Tuo pasinaudojant galima kurti ERP sistemos funkcionalumo bei organizacinių reikalavimų diagramas bei jas išsaugoti tolesniam apdorojimui.

Modelis kuriamas pradedant nuo reikalavimų sutapatinimo algoritmo sudarymo. Jo pagrindu yra abstrakcijos žingsniai, kurių metu reikia surinkti pradinis duomenis – išskirti ERP sistemų funkcijas ir organizacinius reikalavimus, bei sutapatinimo žingsnis – atliekamų veiksmų seka reikia susieti surinktus reikalavimus. Čia remiamasi teoriniu C.Rolland ir N.Prakash metodu [1,2].

Numatoma, kad be OPM palaikančio įrankio OPCAT, grafinės vartotojo sąsajos, sistemos „variklio“ dar bus naudojama ir saugykla – specializuota duomenų bazė, skirta surinktų reikalavimų specifikacijai saugoti. Numatytos šios sistemos funkcijos. Pradedama reikalavimų specifikacijos parengimu – vartotojai-analitikai, ieškančys sau tinkamos sistemos, turėtų sudaryti grafinį reikalavimų medį, nurodant ko tikisi iš naujos sistemos. Vėliau seka reikalavimų išgavimas iš išsaugotų duomenų, jų normalizavimas. Duomenų bazėje išsaugojus normalizuotus reikalavimus bei jų tarpusavio sąryšius, atliekamas reikalavimų sutapatinimas, pritaikant anksčiau sudarytą sutapatinimo algoritmą. Galiausiai lieka rezultatų apdorojimas ir pateikimas vartotojui.

Galimos dvi modelio vystymo kryptys: naudojant sutapatinimo ir/arba pritaikymo būdą. Sutapatinimo būdas yra aprašytas aukščiau – jis grindžiamas tuo, kad vartotojas turi gerai išmanyti OPM metodologiją ir naudodamasis OPCAT sudaryti reikalavimų medį. Siekiant išvengti nesuderinamumą kuriant diagramas, galinčių iškilti dėl sąvokų nesuderinamumo, turi būtų naudojamas reikalavimų-funkcijų žodynas. Vėliau duomenys apdorojami sutapatinimo programa ir pateikiami rezultatai.

Antrasis, pritaikymo būdas, paprastesnis vartotojui. Su vartotoju bendraujama dinaminių dialogų pagalba. Dialogai sudaromi pagal anksčiau surinktas ERP sistemų funkcijas. Šis būdas geras tuo, kad neatsiranda nesusipratimų dėl sąvokų, ERP sistemos parinkimas užima mažiau laiko. Tačiau yra ir savų minusų: vartotojas matys tik tokias funkcijas, kokios yra užregistruotos iš anksčiau, pateikti rezultatai (ERP sistemos) neatspindės trūkumų – reikalavimų, kurių jos netenkina. Taigi konstruktyvesnis yra pirmasis - sutapatinimo būdas – tačiau antrasis sprendimo variantas taip pat naudingas ir jį vartotojas galėtų naudoti kaip alternatyvą.



3 pav. Planuojamos sistemos architektūra

Galiausiai, vartotojui yra pateikiami atrinkti rezultatai – organizacijos reikalavimus tenkinančios ERP sistemos funkcijos bei reikalavimų sąrašas, kurie nėra pilnai arba visai nepadengti ERP funkcionalumu. Rezultatai gali būti įvertinti nurodant procentinius įverčius, pagal tai, kiek sistema atlieka pageidaujamų funkcijų, o taip pat galima nurodyti kokybinius įverčius, tuo atveju vartotojas turėtų iš anksto nustatyti svarbumo faktorius konkreitiems ERP funkcionalumo aspektams.

4 Išvados

Atlikus ERP sistemų, jų funkcionalumo vertinimo modelių bei tam skirtų programinių paketų principų analizę, galima daryti tokias išvadas ir pažymėti:

- ERP – brangios bei sudėtingos sistemos, kurių naudojimo poreikis labai didelis, tačiau pritaikymas prie organizacijos reikalavimų reikalauja daug pastangų ir išlaidų.
- Atliktos analizės bei reikalingų spresti uždavinių planavimo metu orientuotasi į tokią tyrimo sritį - ERP sistemų funkcionalumo specifikavimo metodai, jų diegimo principai bei vartotojų šioms sistemoms keliamų reikalavimų aspektai. Tyrimo objektu pasirinktas ERP sistemų funkcionalumo specifikavimo metodas ir jo programinis modelis. Visu darbu siekiama išspręsti susidarantią sistemų nesėkmingų diegimų grėsmę susidūrus su organizacijų keliamais reikalavimais.
- Analizės metu išnagrinėti kitų autorių siūlomi problemos sprendimai. Kaip tinkamiausias yra pasirinktas C.Rolland ir N.Prakash siūlomas ERP reikalavimų pritaikymo modelis.
- Sudarytas ERP reikalavimų pritaikymo modelio eskizinis variantas. Jo realizacijai numatoma naudotis OPM metodologija paremtomis taisyklėmis ir jų pagrindu veikiančia projektavimo priemone.

Pažymėtina, kad šis darbas atitinka šiuolaikinius ERP sistemų pritaikymo poreikius. Iki šiol rinkoje neegzistuoja sistemų, gebančių pakankamu detalumo lygmeniu sugretinti vartotojų ir ERP sistemų funkcionalumo reikalavimus. Šio darbo pagrindu numatomas sukurti programinis prototipas, kuris turi sumažinti vartotojų reikalavimų analizės pasirenkant ERP sistemą mastus bei pačio ERP sistemų diegimo kaštus, parenkant tinkamiausią įmonių išteklių valdymo sistemą.

Literatūra

- [1] C. Rolland, N. Prakash. Matching ERP System Functionality to Customer Requirements. Toronto, Canada, 2001. [žiūrėta 2005.03.13]. Prieiga per Internetą: <http://doi.ieeeecs.org/10.1109/ISRE.2001.948545>
- [2] C. Rolland, N. Prakash. Bridging the Gap Between Organisational Needs and ERP Functionality. 2000. [žiūrėta 2005.03.12]. Prieiga per Internetą: http://www.ida.his.se/~anne/CR_ERP.pdf
- [3] TechnologyEvaluation.Com. ERP Evaluation Center. [žiūrėta 2004.12.20]. Prieiga per Internetą: <http://www.erp-software-comparison.com>
- [4] M. Themistocleous, Z. Irani, ir kiti. ERP Problems and Application Integration Issues: An Empirical Survey. Brunel University, Uxbridge, UK, 2001. [žiūrėta 2004.12.10]. Prieiga per Internetą: http://www.hicss.hawaii.edu/HICSS_34/PDFs/ST2EA03.pdf
- [5] Object-Process Methodology. [žiūrėta 2005.03.10]. Prieiga per Internetą: <http://www.objectprocess.org>

THE MODEL OF MATCHING ERP SYSTEM FUNCTIONALITY TO CUSTOMER REQUIREMENTS

Paulius Žydelis

The article treats the importance of Enterprise Resource Planning (ERP) system and problems that face users. One of the biggest problem is how to choose the right system spending less time and money. The purpose of this work is to design the method of matching requirements raised by system and customer, and to implement this method by creating model which will collect, match and return acceptable results to customer.












2 PRIEDAS. OPM metodologija, žymėjimai.

Objektų-procesų metodologijos elementai yra esybės ir ryšiai.

OPM metodologijoje yra naudojamos šių tipų esybės:

1 lentelė.





OPM metodologijos esybės

Esybė	Esybės tipas	Esybės simbolis	Aprašymas
Objektas (angl. Object)	Sisteminis loginis objektas		Esybė, kuris tam tikrą laiko tarpą egzistuoja fiziniėje ar loginėje plotmėje.
	Aplinkos loginis objektas		
	Sisteminis fizinis objektas		
	Aplinkos fizinis objektas		
Procesas (angl. Process)	Sisteminis loginis procesas		Veiksmas ar jų visuma, sukuriantis arba sunaikinantis objektą, arba keičiantis objekto būseną.
	Aplinkos loginis procesas		
	Sisteminis fizinis procesas		
	Aplinkos fizinis procesas		
Būsena (angl. State)	Pradinė būsena		Situacija, kurioje gali būti objektas.
	Galinė būsena		
	Paprasta būsena		











Taip pat naudojami tokie sąryšių tarp esybių žymėjimai:

2 lentelė.

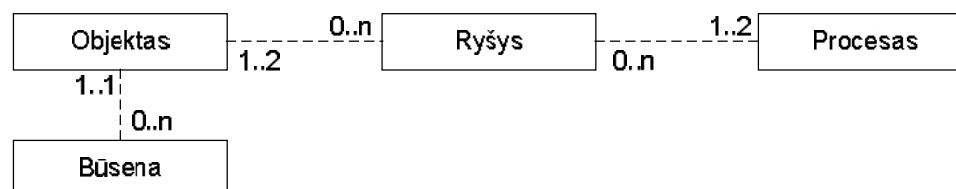
OPM metodologijoje naudojami ryšiai

Ryšių grupė	Ryšio tipas	OPD žymėjimas	Paiškinimas
Struktūriniai ryšiai	Ryšiai, nurodantys pastovų santykį tarp esybių (objektų, procesų).		
Pagrindiniai struktūriniai ryšiai	Agregacija (angl. Aggregation/Participation)		Žymi, kad esybė yra sudaryta iš kitų esybių.
	Charakteristika (angl. Exhibition/Characterization)		Parodo, kad viena esybė apibūdina kitą esybę.
	Generalizacija (angl. Generalization/Specialization)		Žymi faktą, kad esybė apibendrina specializuotų esybių grupę.
	Klasifikacija (angl. Classification/Instantiation)		Apjungia esybes (klases) į bendrą esybę, kurią jos apibūdina.

OPM metodologijoje naudojami ryšiai

Bendrieji struktūriniai ryšiai	Dvikryptis (angl. Bidirectional link)		
	Vienkryptis (angl. Unidirectional link)		
Procedūriniai ryšiai	Ryšiai tarp procesų ir objektų, kuriuos jie keičia arba tarp procesų ir objektų būsenų.		
Aktyvavimo ryšiai	Veiksnių ryšys (angl. Agent link)		Rodo, kad proceso įvykdymui / veikimui reikalingas išorinis veiksnys (agentas) ar jų grupė.
	Įrankio ryšys (angl. Instrument link)		Nurodo, kad proceso įvykdymui / veikimui reikalingas objektas („pradiniai duomenys“) ar jų grupė.
	Sąlyginis ryšys (angl. Condition link)		Nurodo, kad proceso įvykdymui reikalinga sąlyga.
Transformavimo ryšiai	Rezultato/naikinimo ryšys (angl. Result/Consumption link)		Naudojamas, nurodyti, kad procesas sukuria / keičia / sunaikina objektą
	Pasekmės ryšys (angl. Effect link)		Parodo, kad procesas keičia objektą.
Įvykių ryšiai	Įvykio-įrankio ryšys (angl. Instrument event link)		Nurodo, kad objektas tapęs tam tikros būsenos aktyvuoja procesą.
	Įvykio naikinimo ryšys (angl. Consumption event link)		Žymi, kad objektas tapęs tam tikros būsenos aktyvuoja procesą, kuris pakeičia objektą, jo būsenas.
	Iššaukimo ryšys (angl. Invocation link)		Žymi, kad procesas aktyvuoja (iššaukia) kitą procesą.

OPM metodologija apibendrintai galima išreikšti tokiu meta modeliu:



1 pav. OPM metodologijos meta-modelis

Šios schemos paprastumas išreiškia OPM metodologijos paprastumą.

Tam, kad būtų galima sudaryti, kuo didesnes, o tuo pačiu ir aiškesnes, diagramas, OPM metodologija naudoja „mastelio keitimo“ mechanizmus. Šie mechanizmai pagrįsti tuo, kad leidžia sudaryti diagramas išlaikant tarp jų ryšius. Mastelio keitimas apima diagramų detalizavimą bei supaprastinimą. Detalizavimas suteikia daugiau informacijos apie vieną ar kelis schemas elementus. Supaprastinimas naudojamas paprastesnių, aiškesnių, apibendrintų diagramų gavimui.

OPM yra realizuoti du pagrindiniai mastelio keitimo mechanizmai: *Zooming-in/Zooming-out* ir *Unfolding/Folding*.

3 lentelė.

Mastelio keitimo mechanizmų priklausomybės

Tipas	Tikslas ir naudojami mechanizmai		Pirminis operandas	Antrinis operandas
	Detalizavimas	Supaprastinimas		
Matomumo mastelio keitimas	Zooming-in	Zooming-out	procesas	objektas
Hierarchijos mastelio keitimas	Unfolding	Folding	objektas	procesas

Zooming-in – sumažina matomumo mastelį taip, kad žemesniame lygyje apibrėžti elementai tampa nematomais. *Zooming-out* – padidina matomumo mastelį, kad būtų žemesnio lygio esybės taptų matomomis ir jomis būtų galima naudotis atliekant projektavimo veiksmus.

Unfolding – atskleidžia žemesnio lygio esybes, kurios hierarchiškai susietos su aukštesnio lygio esybėmis. Jo rezultatas yra medis, kurio viršūne yra sudėtinis objektas.

3 PRIEDAS. Reikalavimų saugojimui naudojamų duomenų bazės lentelių aprašai

Čia pateiktos duomenų bazę sudarančios lentelės, kurios pagal savo struktūrą ir ryšius padeda saugoti reikalavimų specifikacijas, o vėliau su ja ir dirbti.

- Lentelė „System“ skirta saugoti bendrinius duomenis apie ERP sistemą.

System	
PK	<u>id</u>
	name company web_link max_price min_price

1 pav. Duomenų bazės lentelė *System*

id – sistemos identifikacijos numeris (pirminis lentelės raktas),

name – sistemos pavadinimas,

company – kompanijos, sukūrusios sistemą, pavadinimas,

web_link – WWW nuoroda, kur galima rasti šį produktą ar aprašymą apie jį,

max_price – didžiausia sistemos kaina,

min_price – mažiausia sistemos kaina.

Kainos yra saugomos tam, kad vartotojas galėtų pagal ją pasirinkti sistemą. Pažymėtina, kad vertėtų saugoti du kainų dydžius (mažiausią ir didžiausią), kadangi organizacijos, kuriančios įmonių išteklių valdymo sistemas, numato įvairias kainas, kurios priklauso nuo sistemoje naudojamų modulių kiekio. Modulus kiekvienas užsakovas gali pasirinkti pats.

- Lentelė „SystemRegion“ skirta saugoti sistemų veikimo geografinius regionus (pavyzdžiui, Concorde sistema gali pilnai tenkinti vartotoją (pagal savo teikiamas funkcijas, dokumentų šablonus ir pan.) Pabaltijo valstybėse, Danijoje ir t.t.).

SystemRegion	
PK	<u>system_id</u>
PK	<u>region_id</u>

2 pav. Duomenų bazės lentelė *SystemRegion*

system_id – sistemos identifikacijos numeris (antrinis lentelės raktas),
region_id – geografinio regiono identifikacijos numeris (antrinis lentelės raktas).

- Lentelė „Region“ skirta saugoti informaciją apie regionus, kuriuose kaip numatyta sistemos kūrėjų sukurtas produktas pilnai atliks savo funkcijas.

Region	
PK	<u>id</u>
	name

3 pav. Duomenų bazės lentelė *Region*

id – regiono identifikacijos numeris (pirminis lentelės raktas),
name – regiono pavadinimas.

- Lentelė „Module“ skirta saugoti informacija apie ERP sistemų funkcinis blokus-modulius.

Module	
PK	<u>id</u>
	system_id
	name
	description

4 pav. Duomenų bazės lentelė *Module*

id – sistemos modulio identifikacijos numeris (pirminis lentelės raktas),
system_id – sistemos identifikacijos numeris (antrinis lentelės raktas),
name – modulio pavadinimas,
description – modulio aprašymas.

- Lentelėje „Association“ saugomi duomenys yra naudojami sąryšiams tarp esybių (objektų, būsenų, procesų ir modulių) pažymėti.

Association	
PK	<u>id</u>
PK	<u>src_id</u>
PK	<u>dest_id</u>
	type

5 pav. Duomenų bazės lentelė *Association*

id – esybių sąryšio identifikacijos numeris (pirminis lentelės raktas),
src_id – pradinės esybės (šaltinio) identifikacijos numeris (antrinis lentelės raktas),
dest_id – galinės esybės (tikslas) identifikacijos numeris (antrinis lentelės raktas),
type – sąryšio tipo žymėjimas (žymi ar tai sąveika (angl. Relation), ar jungtis (angl. Link)).

- Lentelėje „Relation“ esantys duomenys papildo lentelėje „Association“ saugomą informaciją apie sąryšius. Čia saugomi papildomi sąveikų duomenys.

Relation	
PK	<u>id</u>
	type src_cardinality dest_cardinality forward_name backward_name

6 pav. Duomenų bazės lentelė *Relation*

id – esybių sąryšio identifikacijos numeris (pirminis lentelės raktas),
type – sąryšio tipas (pvz.: Apibendrinimas, charakteristika ir t.t.),
src_cardinality – šaltinio esybės kardinalumas,
dest_cardinality – tikslo esybės kardinalumas,
forward_name – sąryšio, jungiančio esybės kryptimi šaltinis-tikslas, pavadinimas,
backward_name – sąryšio, jungiančio esybės kryptimi tikslas-šaltinis, pavadinimas.

- Lentelėje „Link“ esantys duomenys papildo lentelėje „Association“ saugomą informaciją apie sąryšius. Čia saugomi papildomi jungčių duomenys.

Link	
PK	<u>id</u>
	type condition path max_time min_time

7 pav. Duomenų bazės lentelė *Link*

id – esybių sąryšio identifikacijos numeris (pirminis lentelės raktas),

type – sąryšio tipas (pvz.: Rezultato, pasekmės ir t.t.),

condition – žymi būseną, kuri tęsiasi tol kol dvi esybės yra sujungtos atitinkamu ryšiu,

path – ryšio pavadinimas, naudojamas išskirti ryšį iš jų aibės,

max_time – žymi ilgiausią laiko tarpą per kurį turi būti pradėtas proceso vykdymas (maksimalus reakcijos laikas),

min_time – žymi trumpiausią laiko tarpą per kurį turi būti pradėtas proceso vykdymas (minimalus reakcijos laikas).

- Lentelėje „Process“ yra saugomi duomenys apie sistemos procesus bei jų ryšius su kitomis esybėmis (procesais, objektais ar būsenomis).

Process	
PK	<u>id</u>
	name description process_body max_time min_time

8 pav. Duomenų bazės lentelė *Process*

id – sistemos proceso identifikacijos numeris (pirminis lentelės raktas),

name – proceso pavadinimas,

description – proceso aprašymas,

process_body – proceso turinys (žodinis ar kodinis veiksmo aprašymas),

max_time – žymi ilgiausią laiko tarpą per kurį procesas yra įvykdomas (maksimalus veikimo laikas),

min_time – žymi trumpiausią laiko tarpą per kurį procesas yra įvykdomas (minimalus veikimo laikas).

- Lentelė „Object“ naudojama esybių-objektų informacijai saugoti.

Object	
PK	id
	name description type initial_value

9 pav. Duomenų bazės lentelė *Object*

id – objekto identifikacijos numeris (pirminis lentelės raktas),

name – objekto pavadinimas,

description – objekto aprašymas,

type – objekto tipas (pvz.: loginis kintamasis *boolean*, skaičius *int*, *double*,...),

initial_value – pradinė objekto reikšmė.

- Lentelė „State“ naudojama esybių-objektų būsenoms saugoti.

State	
PK	id
	obj_id name description default initial final max_time min_time

10 pav. Duomenų bazės lentelė *State*

id – būsenos identifikacijos numeris (pirminis lentelės raktas),

obj_id – tėvinio objekto, kuriam priklauso ši būsena identifikacijos numeris (antrinis lentelės raktas),

name – būsenos pavadinimas,

description – būsenos aprašymas,

default – žymi, kad būseną yra įprastinė pirminė,

initial – žymi, kad būseną yra pradinė,

final – žymi, kad būseną yra galinė,

max_time – žymi ilgiausią laiko tarpą, kurį objektas gali būti šioje būsenoje (maksimalus aktyvumo laikas),

min_time – žymi trumpiausią laiko tarpą, kurį objektas gali būti šioje būsenoje (minimalus aktyvumo laikas).

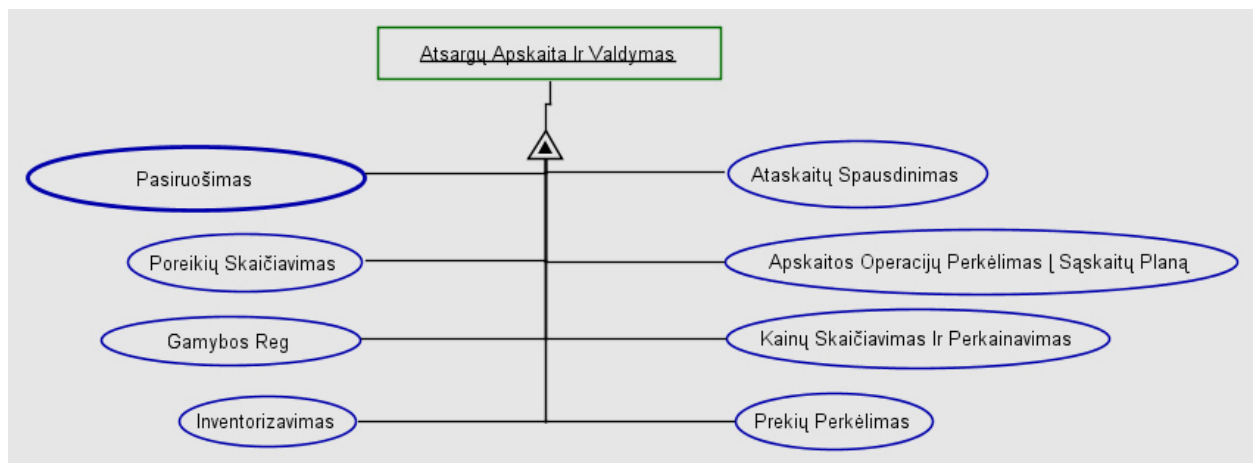
4 PRIEDAS. Reikalavimų sutapatinimo modelio testiniai duomenys

Šio magistrinio darbo buvo nemažai naudotasi KTU dr. doc. L. Nemuraitės mokomąja knyga „Vadybos kompiuterizuotos vietos“ [13]. Pagal knygoje pateiktas schemas, sprendžiamus uždavinius buvo išskirtos ten nagrinėjamos ERP sistemos dalys, pagal kurias buvo galima sudaryti sistemos grafą (tiksliau jo dalį) atitinkančią sistemos struktūrą. Analizuota ir konstruota ConcordeXAL sistemos struktūra.



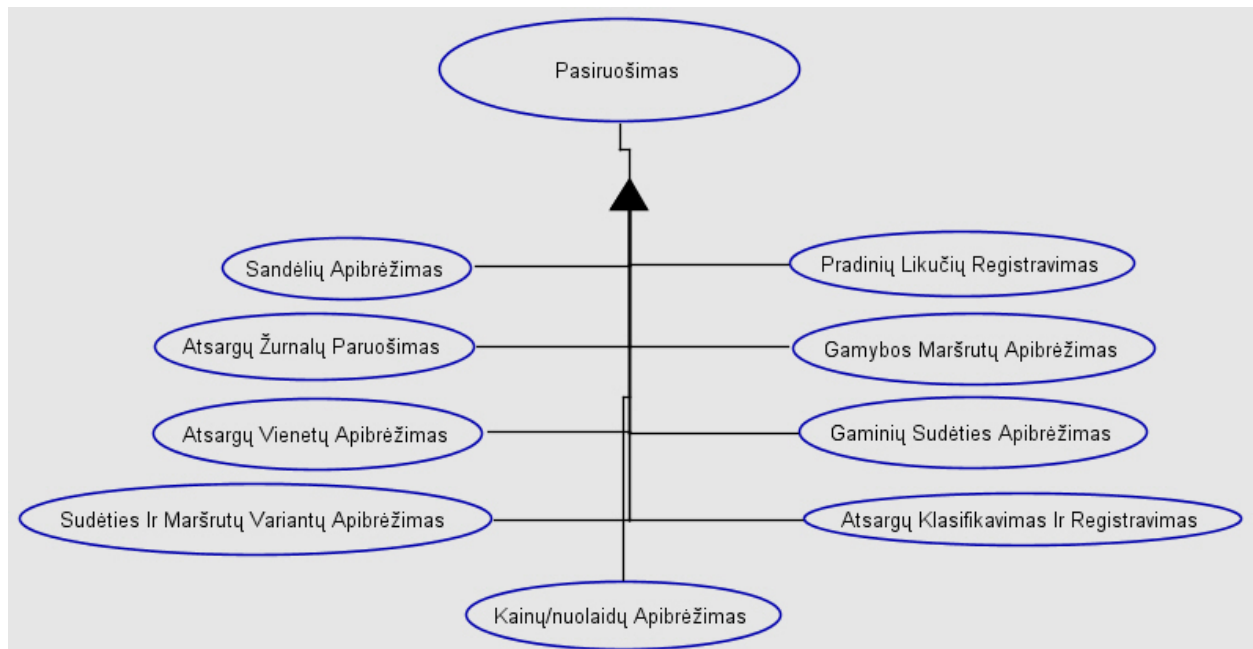
1 pav. Išskirti ConcordeXal sistemos moduliai

1 paveiksle parodyta informacija, kad abstraktų objektą „Modules“ sudaro kiti objektai – moduliai, kurie žymi ERP nagrinėjamas verslo sritis.

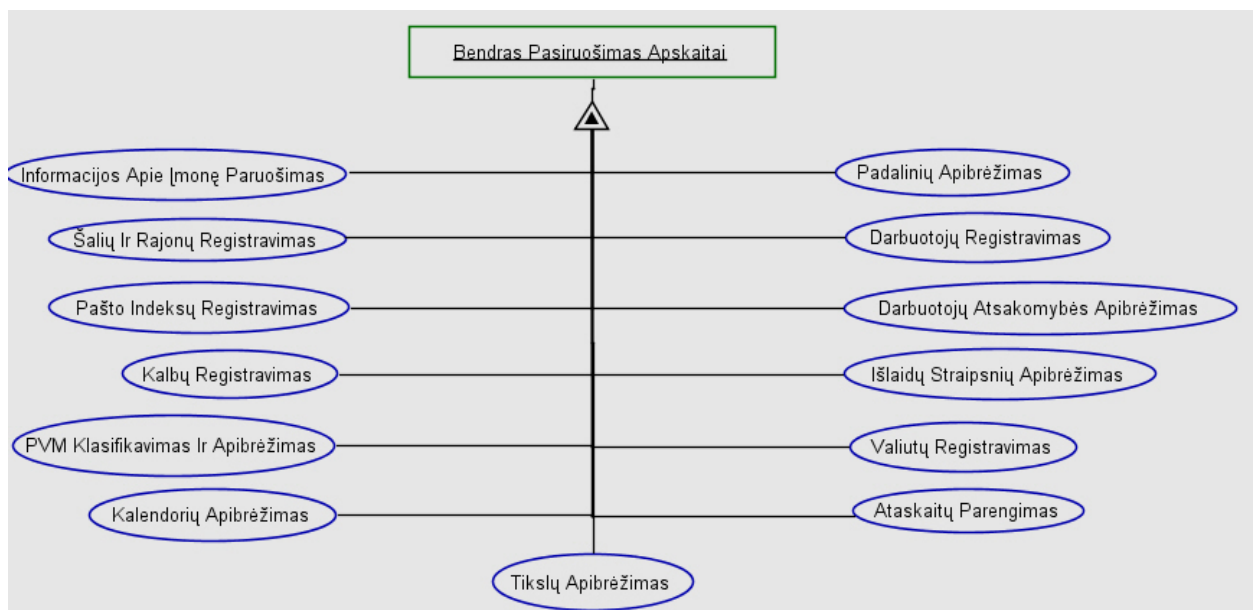


2 pav. Modulio „Atsargų Apskaita ir Valdymas“ struktūra

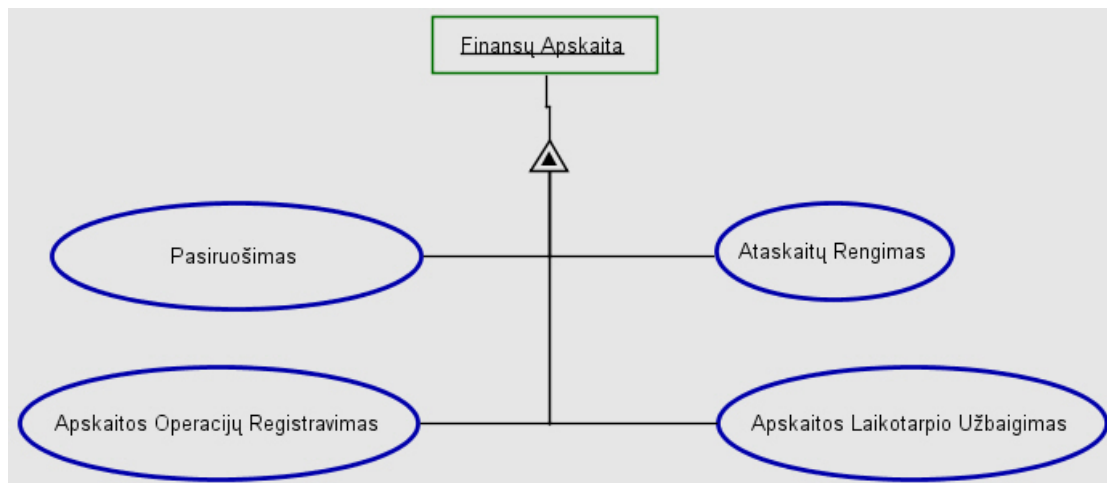
2 paveiksle pateikta schema nurodo, kad objektas-modulis „Atsargų apskaita ir valdymas“ yra charakterizuojamas grupe abstrahuotų procesų: „Pasiruošimas“, „Poreikių skaičiavimas“, „Inventorizavimas“ ir kiti.



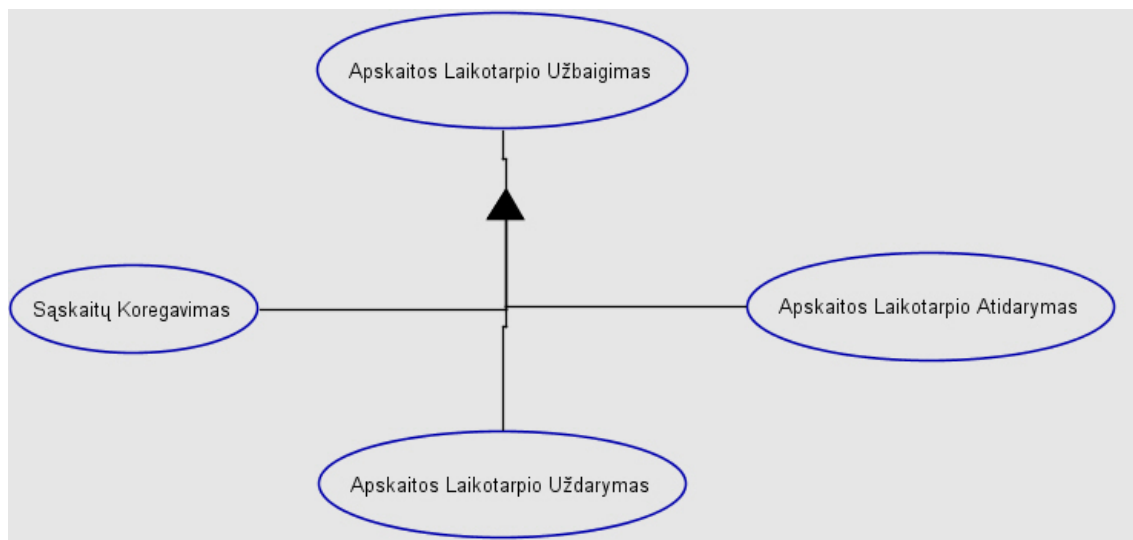
3 pav. Modulo „Atsargų Apskaita ir Valdymas“ procesą „Pasiruošimas“ sudarantys procesai



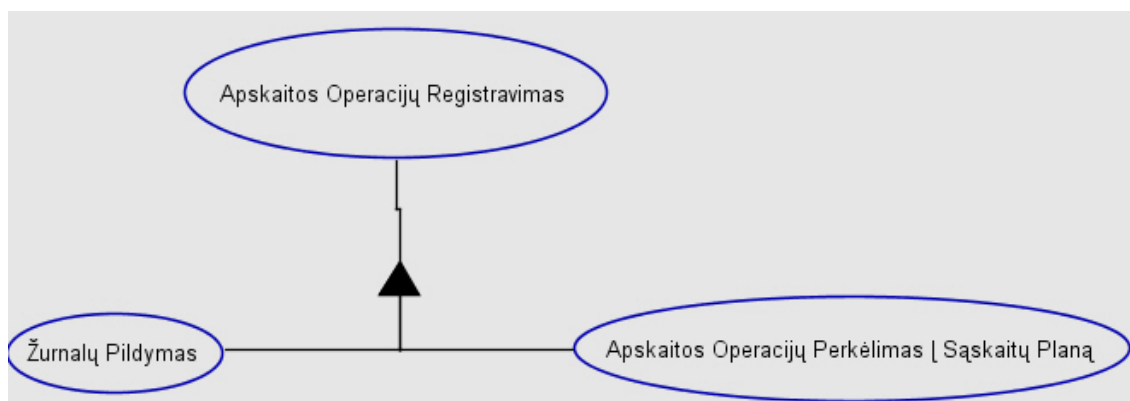
4 pav. Modulo „Bendras Pasiruošimas Apskaitai“ struktūra



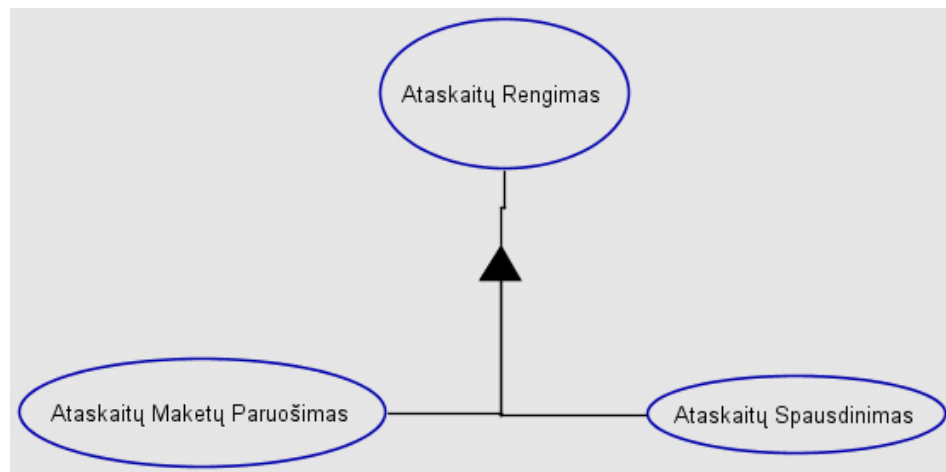
5 pav. Modulio „Finansų Apskaita“ struktūra



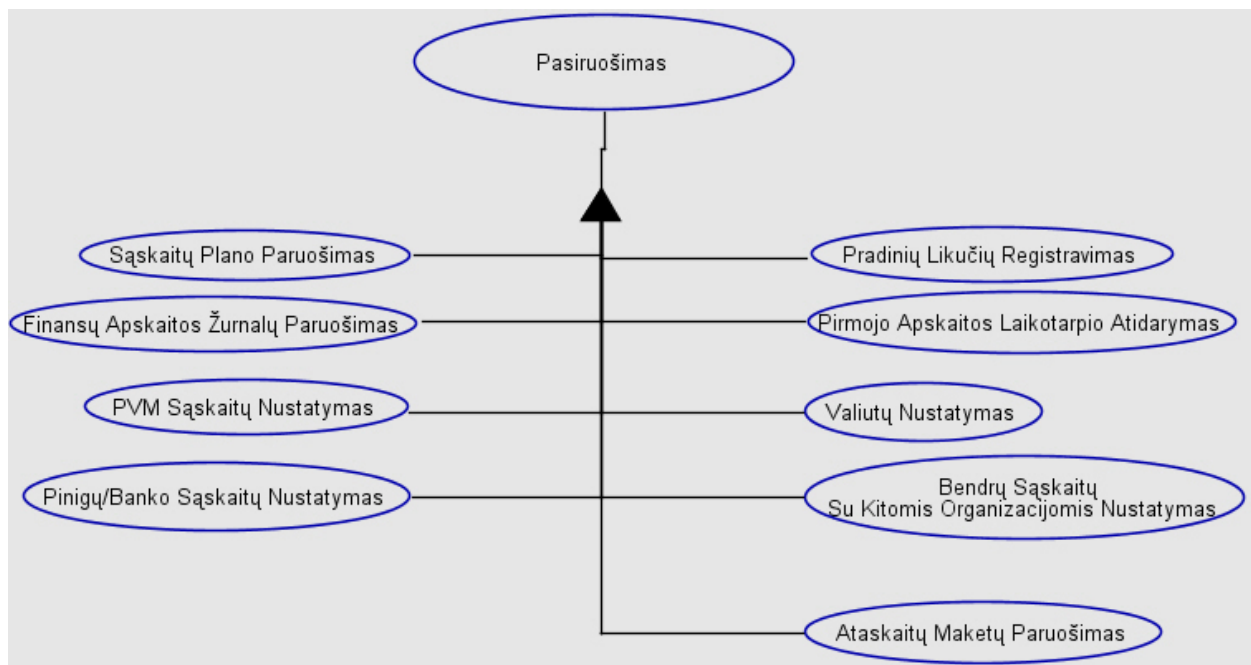
6 pav. Proceso sudedamosios dalys



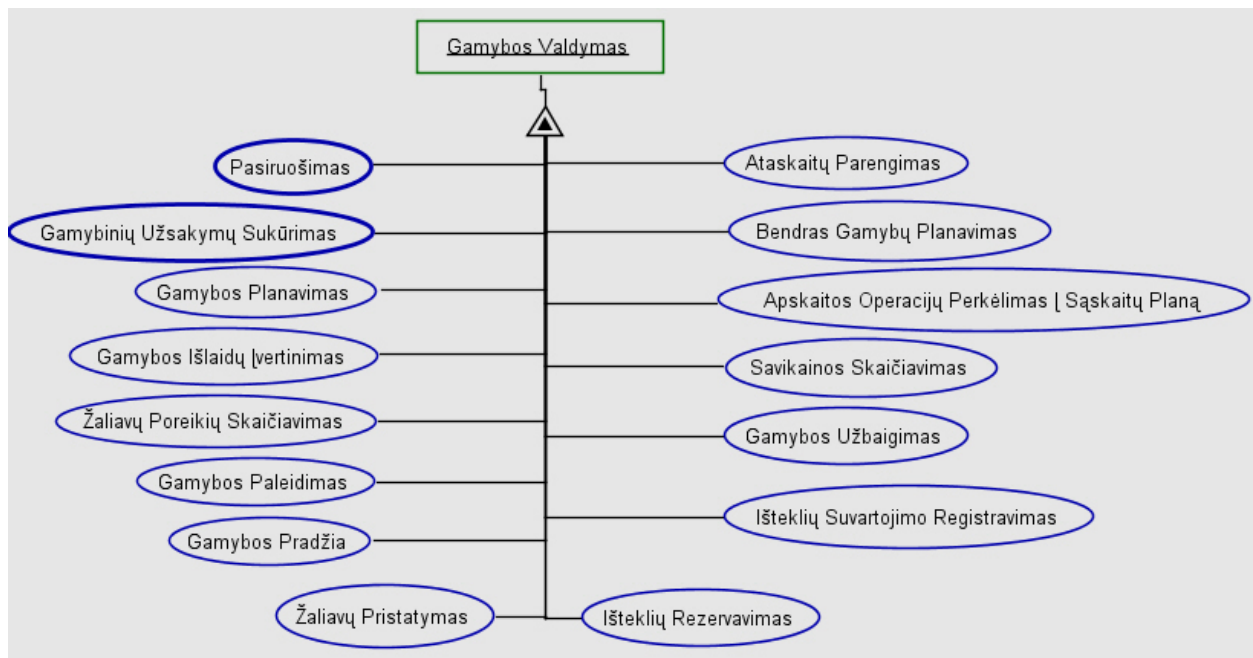
7 pav. Proceso „Apskaitos Operacijų Registravimas“ struktūra



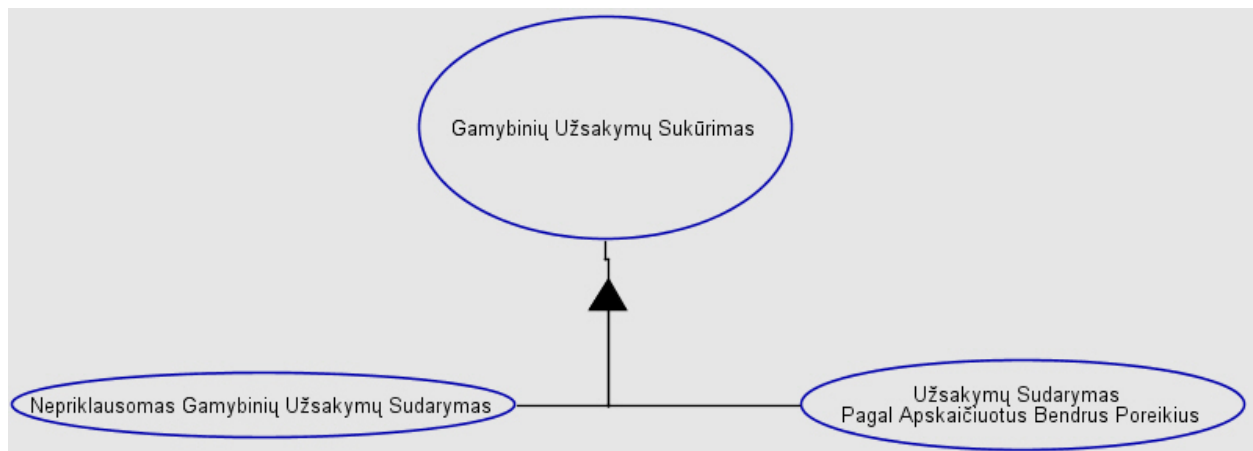
8 pav. Proceso „Ataskaitų rengimas“ sudėtinės dalys



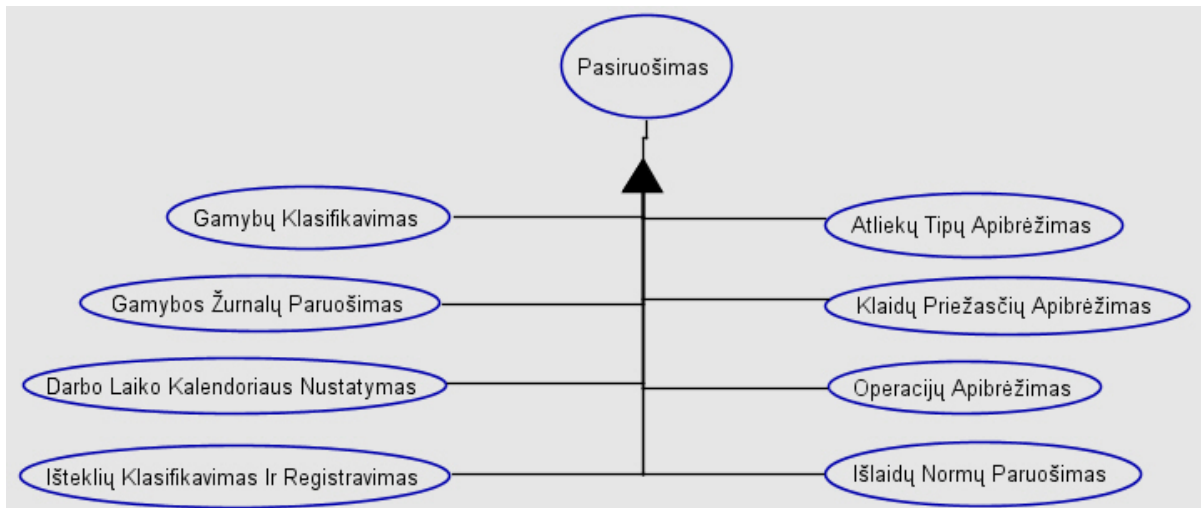
9 pav. Modulo „Finansų apskaita“ proceso „Pasiruošimas“ struktūra



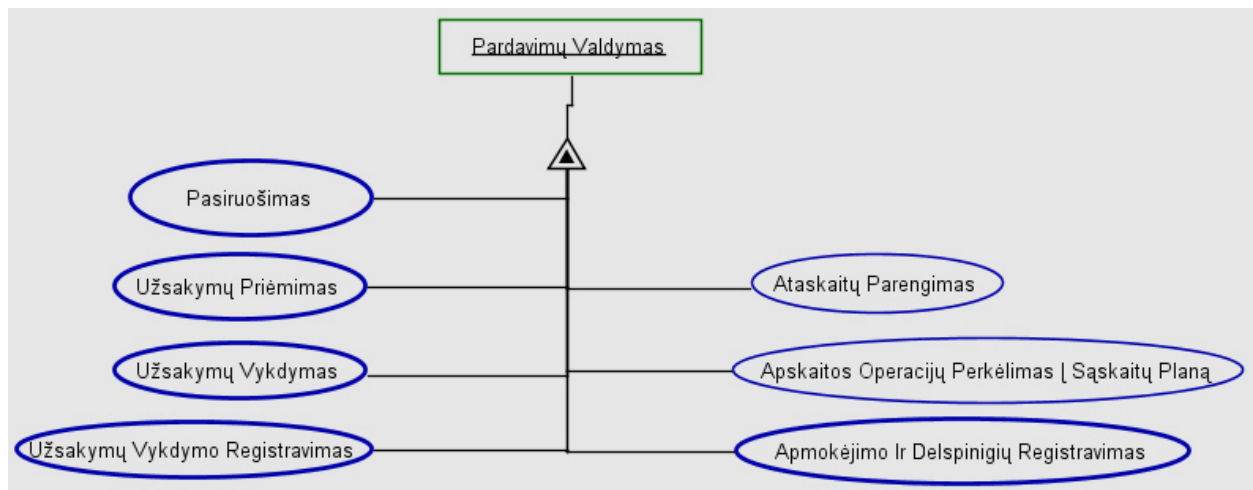
10 pav. Modulio „Gamybos valdymas“ struktūra



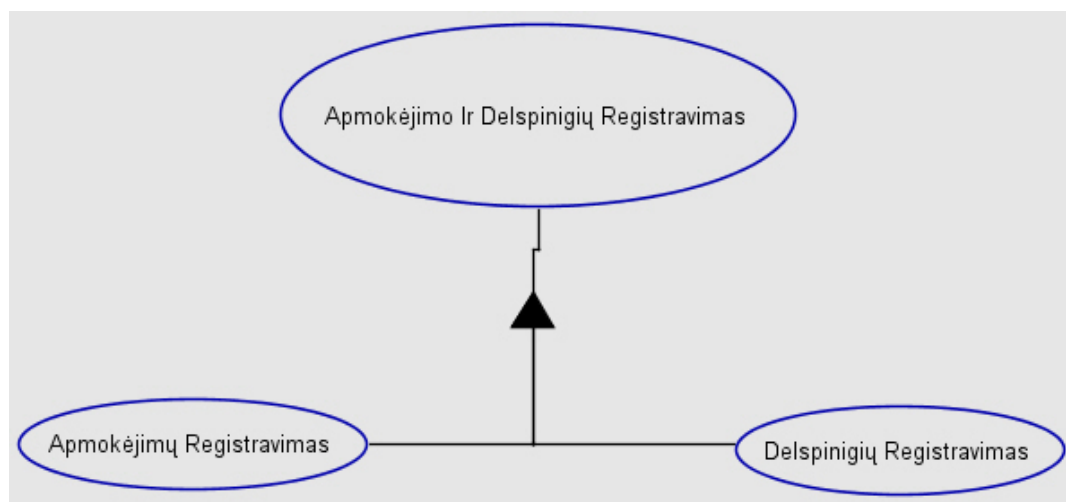
11 pav. Proceso „Gamybos Valdymas“ → „Gaminų Užsakymų Sukūrimas“ struktūra



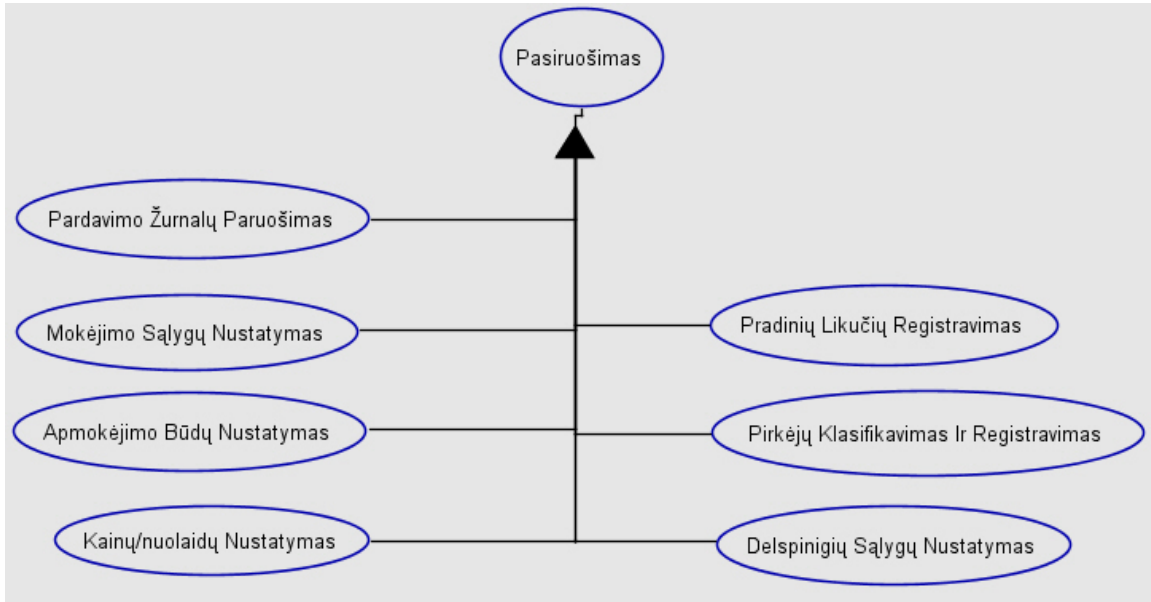
12 pav. „Pasiruošimo“ sudėtis. „Pasiruošimas“ priklauso „Gamybos valdymui“



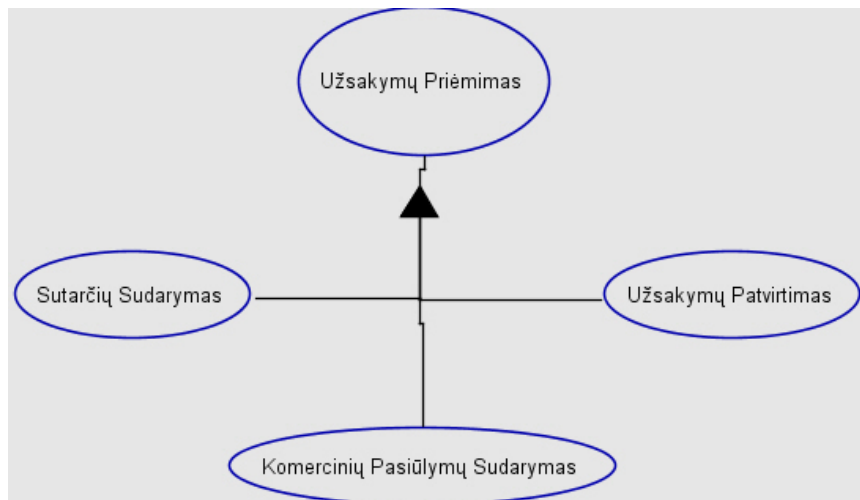
13 pav. Modulio „Pardavimų Valdymas“ struktūra



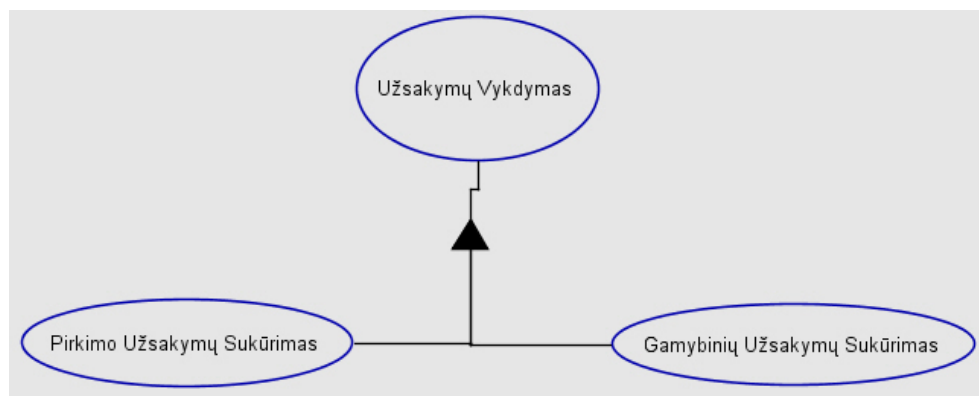
14 pav. Proceso struktūra



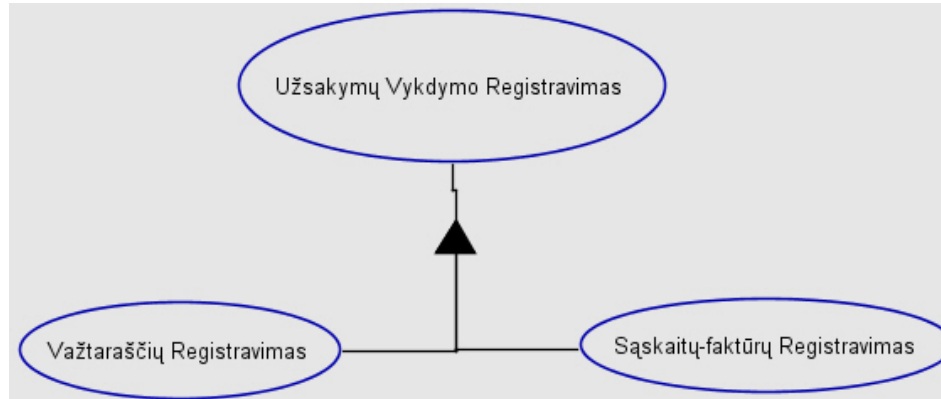
15 pav. Proceso struktūra



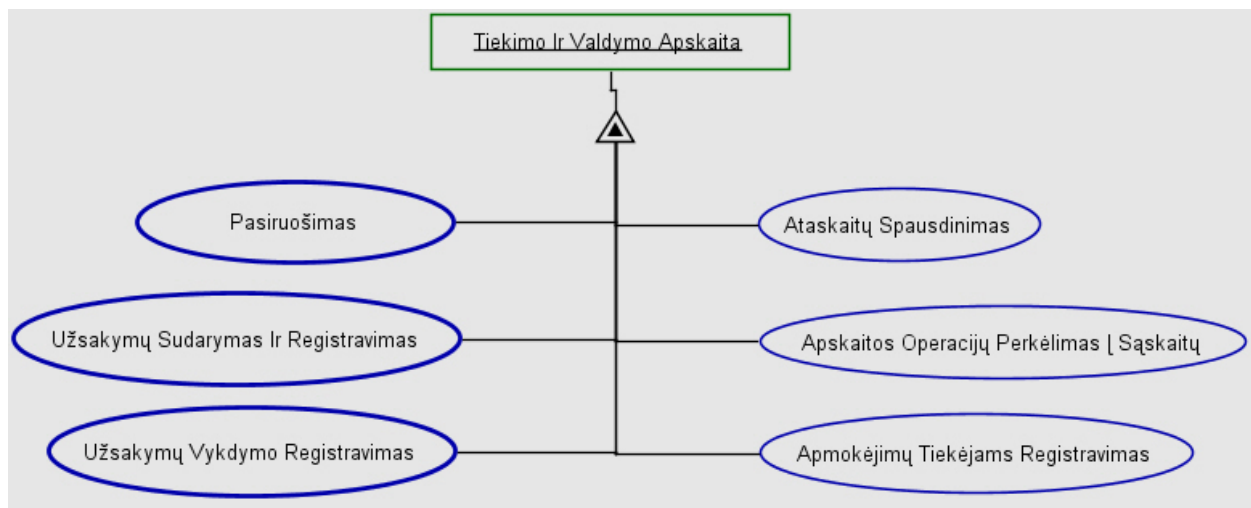
16 pav. Proceso struktūra



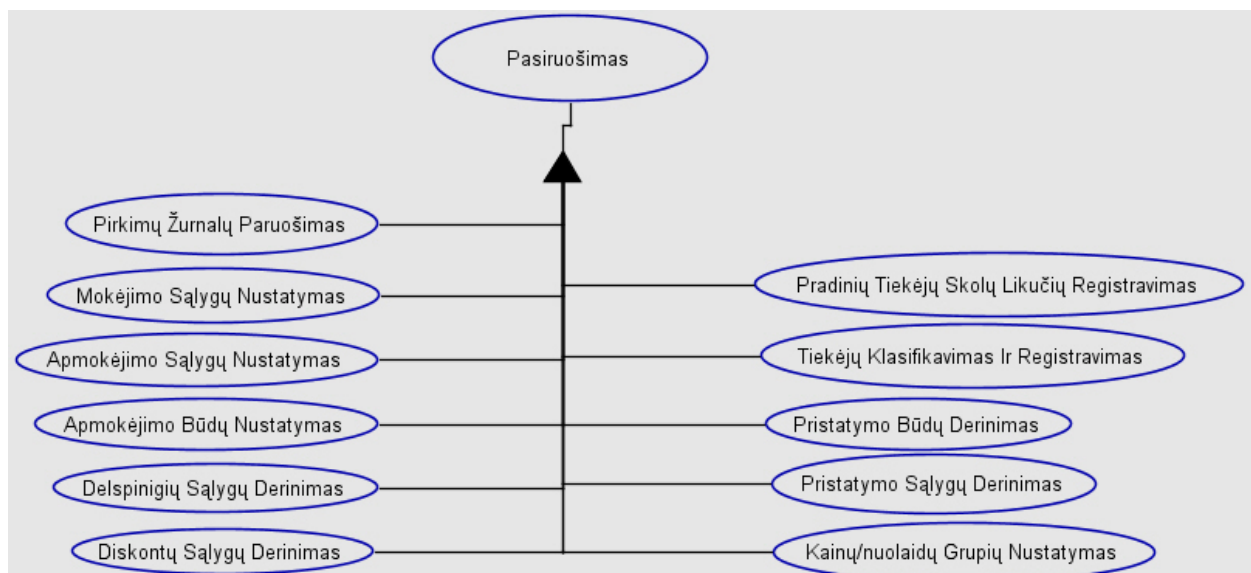
17 pav. Proceso struktūra



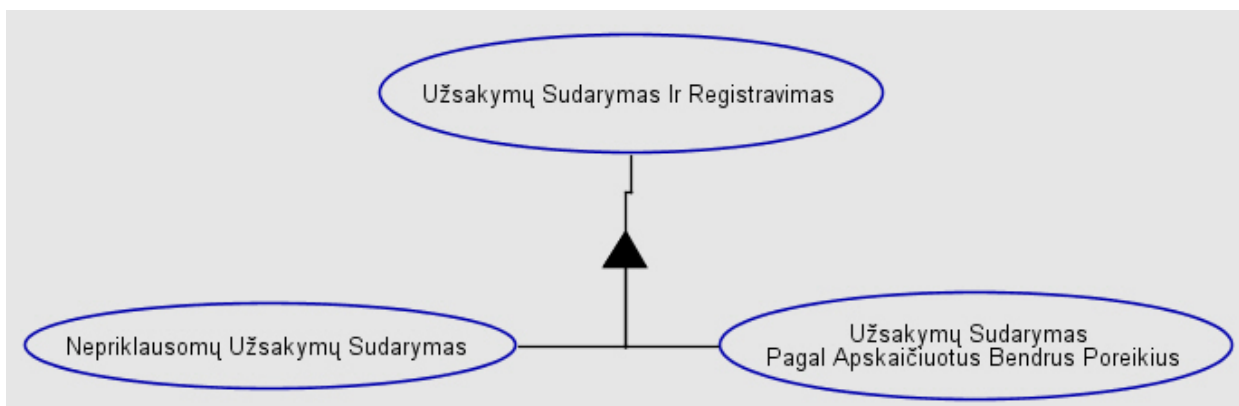
18 pav. Proceso struktūra



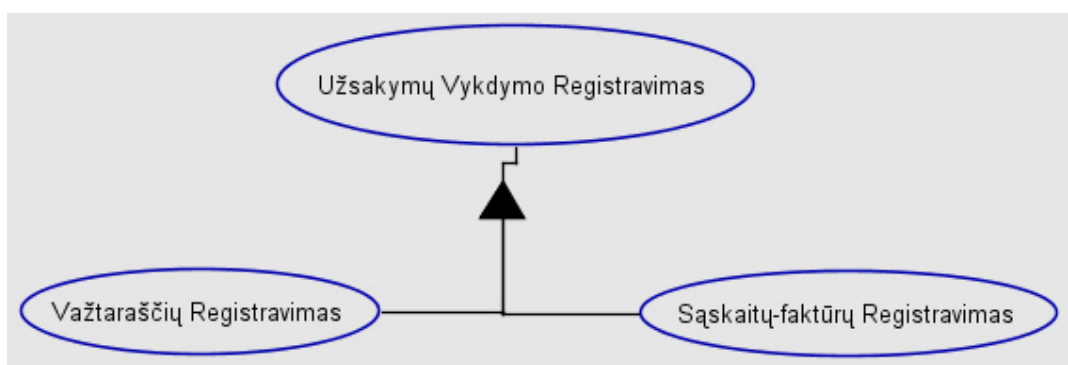
19 pav. Modulio „Tiekimo Ir valdymo Apskaita“ struktūra



20 pav. „Tiekimo Ir Valdymo Apskaita“ „Pasiruošimas“



21 pav. „Užsakymo Sudarymas ir Registravimas“ sudėtis



22 pav. „Užsakymų Vykdyimo Registravimas“

Taip aprašytą sistemą užtenka išsaugoti XML formatu apibrėžtomis OPCAT rezultatų failo sudarymo taisyklėmis ir tuomet naudojantis sukurtu reikalavimų sutapatavimo modeliu atlikti duomenų analizę bei panaudojimą.