

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

Laura Jučiūtė

**DUOMENŲ LOGINIŲ STRUKTŪRŲ
IŠSKYRIMAS FUNKCINIŲ REIKALAVIMŲ
SPECIFIKACIJOS PAGRINDU**

Magistro darbas

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

KAUNAS, 2006

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

**DUOMENŲ LOGINIŲ STRUKTŪRŲ
IŠSKYRIMAS FUNKCINIŲ REIKALAVIMŲ
SPECIFIKACIJOS PAGRINDU**

Programų inžinerijos magistro baigiamasis darbas

Kalbos konsultantė

Lietuvių k. katedros lekt.
I. Mickienė
2006-05-23

Vadovas

doc. dr. R. Butleris
2006-05-29

Recenzentė

dr. A. Janavičiūtė
2006-05-29

Atliko

IFM-0/2 gr. stud.
L. Jučiūtė
2006-05-23

KAUNAS, 2006

SUMMARY

Recently the information systems (IS) that computerize various activities in the world are very marketable. Developers of the systems must ensure the effective use of created product. That is why the methods and technologies of IS development are improved constantly. It is desirable that every activity, every stage of the developing process would be as much automated as possible.

Data modeling is one of the initial activities in IS developing process. The quality of data model quality is very important in all information systems' life cycle, also in effective IS exploitation. Referring to results of the nonfiction literature analysis the reasons why the process of data modeling must be automated are introduced, current automatization solutions are described. And as it is the main purpose of master's work an original data modeling method is described and programmable prototype which automates one step of that method – schema integration is introduced.

TURINYS

1. ĮVADAS	6
2. DUOMENŲ STATINĖ STRUKTŪRA IR JOS SUDARYMAS	8
2.1. IS GYVAVIMO CIKLAS IR DUOMENŲ MODELIO REIKŠMĖ	8
2.2. DUOMENŲ MODELIS IR FUNKCINIŲ REIKALAVIMŲ SPECIFIKACIJA.....	8
2.3. ESYBIŲ-RYŠIŲ SCHEMŲ MODELIAVIMAS	10
2.4. DUOMENŲ MODELIAVIMAS, REMIANTIS INFORMACIJOS SRAUTŲ STRUKTŪRA	12
2.5. SCHEMŲ INTEGRAVIMO PROCESAS.....	14
2.5.1. Schemų lyginimas.....	15
2.5.2. Schemų suderinimas.....	18
2.5.3. Schemų sujungimas	18
2.6. ŽINOMI SCHEMŲ INTEGRAVIMO METODAI IR MODELIAI BEI GALIMYBĖS AUTOMATIZUOTI ŠĮ PROCESĄ.....	19
2.7. PROTOTIPO PROGRAMINĖS REALIZACIJOS VIZIJA	20
3. ER SCHEMŲ INTEGRAVIMAS DUOMENŲ SRAUTAIS	22
3.1. ER MODELIŲ INTEGRAVIMO INFORMACIJOS SRAUTŲ SPECIFIKACIJOS PAGRINDU ALGORITMAS	22
3.2. INFORMACIJOS SRAUTŲ SPECIFIKACIJA – RELIACINĖ METADUOMENŲ BAZĖ	24
3.3. PAGRINDINIAI REIKALAVIMAI IR APRIBOJIMAI PROTOTIPUI	33
4. PROTOTIPAS IS DUOMENŲ MODELIO SUDARYTI.....	37
4.1. TRUMPAS PROTOTIPO APRAŠYMAS IR EKRAVINIAI VAIZDAI.....	37
4.2. PROTOTIPO VEIKIMO EKSPERIMENTINIAI REZULTATAI	38
4.3. PROTOTIPO VERTINIMAS STANDARTINIO MS VISIO PROGRAMINIO PAKETO KONTEKSTE 39	
5. IŠVADOS.....	41
LITERATŪRA	43
TERMINŲ IR SANTRUMPŲ ŽODYNAS.....	45

PAVEIKSLĖLIŲ SĄRAŠAS

1 PAV.	DUOMENŲ MODELIO VAIDMUO	8
2 PAV.	IS DUOMENŲ MODELIO SUDARYMAS, REMIANTIS INFORMACIJOS SRAUTŲ STRUKTŪRA – KONCEPCINĖ SCHEMA.....	13
3 PAV.	SCHEMŲ INTEGRAVIMO PROCESAS.....	14
4 PAV.	SCHEMŲ INTEGRAVIMO BŪDAI: (A) N-NARINIS INTEGRAVIMAS, (B) BINARINIS INTEGRAVIMAS.....	15
5 PAV.	ATRIBUTŲ, NEGALINČIŲ BŪTI SINONIMAIS, PAVYZDYS	16
6 PAV.	SEMANTIŠKAI EKVIVALENČIŲ, BET STRUKTŪRIŠKAI SKIRTINGŲ KONSTRUKCIJŲ PAVYZDŽIAI (1)	16
7 PAV.	SEMANTIŠKAI EKVIVALENČIŲ, BET STRUKTŪRIŠKAI SKIRTINGŲ KONSTRUKCIJŲ PAVYZDŽIAI (2)	17
8 PAV.	ER MODELIŲ INTEGRAVIMO INFORMACIJOS SRAUTŲ SPECIFIKACIJOS PAGRINDU ALGORITMAS	23
9 PAV.	INFORMACIJOS SRAUTŲ SPECIFIKACIJOS SAUGYKLOS STRUKTŪRA	25
10 PAV.	INFORMACIJOS SRAUTŲ SAUGYKLA – UŽPILDYTOT „ĖSYBE“ IR „ATRIBUTAS“ LENTELĖS	30
11 PAV.	INFORMACIJOS SRAUTŲ SAUGYKLA – UŽPILDYTA „RYSYS“ LENTELĖ	31
12 PAV.	INFORMACIJOS SRAUTŲ SAUGYKLA – UŽPILDYTOT „DUOM_SALT“ IR „DS_ESYBE“ LENTELĖS	32
13 PAV.	INFORMACIJOS SRAUTŲ SAUGYKLA – UŽPILDYTOT „SRAUTAS_DS“ IR „SR_DS_ELEM“ LENTELĖS	33
14 PAV.	DUOMENŲ MODELIAVIMO VEIKLOS KONTEKSTAS.....	34
15 PAV.	PROTOTIPO PANAUDOJIMO ATVEJŲ DIAGRAMA.....	35
16 PAV.	IS KOMPIUTERIZAVIMO CASE PRIEMONĖ	36
17 PAV.	KOMPONENTŲ DIAGRAMA	36
18 PAV.	PAGRINDINIS PROTOTIPO LANGAS	37
19 PAV.	MS VISIO PROGRAMINIO PAKETO LANGAS IR DUOMENŲ ŠALTINIŲ PARINKIMO LANGAS	37
20 PAV.	ESYBĖS IR RYŠIO SAVYBIŲ REDAGAVIMO LANGAI.....	38
21 PAV.	ATVAIZDUOTA DŠ „UŽSAKYMAS“ IR „SĄSKAITA – FAKTŪRA“ STRUKTŪRA	38
22 PAV.	INTEGRUOTA ER SCHEMA, APIMANTI DŠ „UŽSAKYMAS“ IR „SĄSKAITA – FAKTŪRA“ ER SCHEMAS	39

LENTELIŲ SĄRAŠAS

1 LENTELĖ.	DUOMENŲ SRAUTŲ SAUGYKLOS LENTELĖ „DUOM_SALT“	26
2 LENTELĖ.	DUOMENŲ SRAUTŲ SAUGYKLOS LENTELĖ „SRAUTAS_DS“	26
3 LENTELĖ.	DUOMENŲ SRAUTŲ SAUGYKLOS LENTELĖ „ATRIBUTAS“	26
4 LENTELĖ.	DUOMENŲ SRAUTŲ SAUGYKLOS LENTELĖ „ĖSYBE“	27
5 LENTELĖ.	DUOMENŲ SRAUTŲ SAUGYKLOS LENTELĖ „RYSYS“	27
6 LENTELĖ.	DUOMENŲ SRAUTŲ SAUGYKLOS LENTELĖ „DS_ATRIBUTAS“	28
7 LENTELĖ.	DUOMENŲ SRAUTŲ SAUGYKLOS LENTELĖ „RIB_REIKSME_DS“	28
8 LENTELĖ.	DUOMENŲ SRAUTŲ SAUGYKLOS LENTELĖ „DS_ESYBE“	29
9 LENTELĖ.	DUOMENŲ SRAUTŲ SAUGYKLOS LENTELĖ „SALYGOS_ELEM_DS“	29
10 LENTELĖ.	DUOMENŲ SRAUTŲ SAUGYKLOS LENTELĖ „SR_DS_ELEM“	29
11 LENTELĖ.	DARBE VARTOJAMŲ SANTRUMPŲ PAAIŠKINIMAS	45
12 LENTELĖ.	DARBE VARTOJAMŲ TERMINŲ PAAIŠKINIMAS	45

1. ĮVADAS

Pastaruojų metu itin išaugusi informacijos sistemų (IS), kompiuterizuojančių įvairias veiklos sritis, paklausa. Informacijos sistemų kūrėjai privalo užtikrinti efektyvų sukurto produkto naudojimą. Todėl siekiant išvengti IS eksploatavimo metu atsirandančių klaidų, kurias sąlygoja anksčiau nepastebėti trūkumai, tobulinami egzistuojantys IS kūrimo metodai bei technologijos. Siekiama automatizuoti kuo daugiau kiekvieno kūrimo etapo veiklą.

Duomenų modeliavimas – tai viena pirmųjų IS kūrimo veiklų, kurios rezultatas itin svarbus tolesniam sėkmingam IS kūrimui ir eksploatavimui. Duomenų modelis – tai kuriamos informacijos sistemos duomenų loginės struktūros vaizdas. Pastarasis sudaromas analizės metu ir pateikiamas funkcinių reikalavimų specifikacijos dokumente. Šiame darbe bus nagrinėjamos duomenų modeliavimo veiksmų automatizavimo galimybės.

Svarbiausios probleminės srities sąvokos yra duomenų šaltinis (DŠ), rezultatas (R) ir duomenų srautas (DS). Informacijos sistemos paskirtis – iš tam tikrų įeinančių informacijos srautų (duomenų šaltinių) suformuoti ir pateikti naudotojui išeinančius informacijos srautus (funkcionalumo rezultatus). Trumpai tariant, duomenis rezultatams formuoti teikia vienas arba keli įeinantys informacijos srautai, suprantami kaip to rezultato duomenų šaltiniai. Duomenis iš duomenų šaltinio į rezultatą perduoda duomenų srautas. Kadangi vienas rezultatas dažnai formuojamas, apjungiant keliuose duomenų šaltiniuose esančią informaciją, tam tikri duomenų srautai sieja ne tik DŠ su rezultatais, bet ir pačius duomenų šaltinius tarpusavyje.

Pagrindinis šio darbo tikslas – aprašyti originalų duomenų modeliavimo metodą ir realizuoti programinį prototipą, automatizuojantį vieną iš šio metodo žingsnių – schemų integravimą. Siekiant sukurti prototipą reikia sudaryti algoritmą, kurio pagrindu programinis įrankis veiks. Algoritmas bus sudarytas remiantis prieš tai atlikta duomenų modelio sudarymo principų ir apribojimų analize. Taip pat bus atliekami eksperimentai, padėsiantys patikrinti prototipo veikimo teisingumą.

Antrajame dokumento skyriuje – darbo analizės dalyje parodoma duomenų modelio svarba informacijos sistemos kūrimo ir apskritai gyvavimo ciklo kontekste. Taip pat aprašoma funkcinių reikalavimų specifikacija ir duomenų modelio vieta joje. Supažindinama su esybių-ryšių schemų modeliavimu ir nurodomos šios metodikos pasirinkimo darbe priežastys. Pristatomi duomenų modelio sudarymo principai ir etapai remiantis kompiuterizuojamos organizacijos informacijos srautų struktūra. Detaliai aprašomas schemų integravimo procesas. Analizuojamos duomenų modelio sudarymo proceso automatizavimo galimybės bei aptariami

esami sprendimai. Skyriaus pabaigoje pateikiama planuojamo prototipo programinės realizacijos vizija bei nustatomi sukurto įrankio kokybės vertinimo kriterijai.

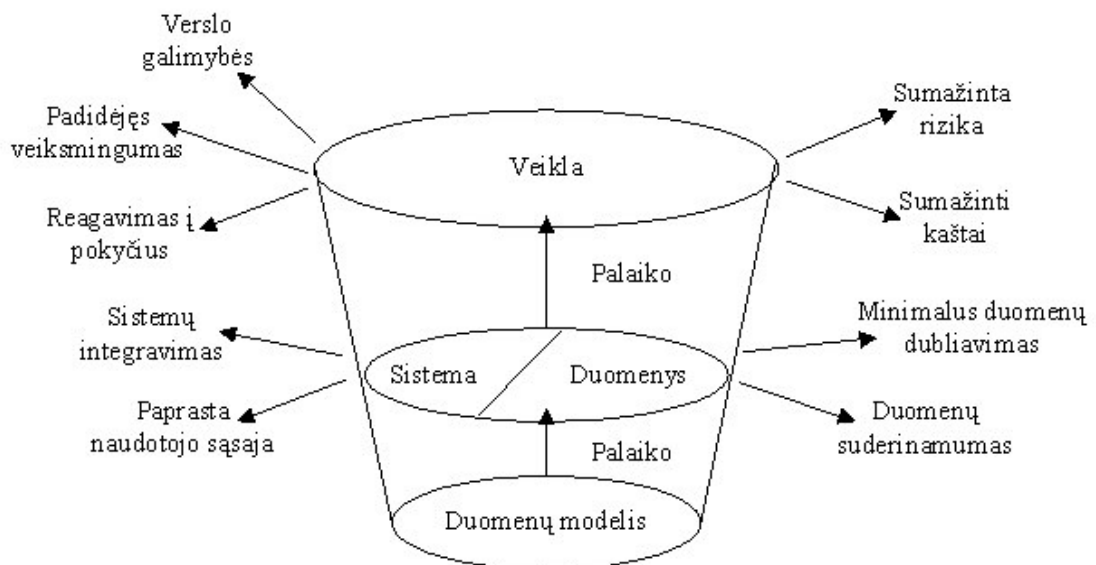
Trečiasis dokumento skyrius – projektinė darbo dalis. Čia nagrinėjami esybių-ryšių (ER) schemų integravimo informacijos šaltų struktūros pagrindų ypatumai. Trumpai pristatomi pagrindiniai prototipo realizacijos sprendimai. Pateikiamas ir išaiškinamas schemų integravimo algoritmas, kurio pagrindu kuriamas įrankis.

Ketvirtajame, eksperimentinio tyrimo, skyriuje pristatomi automatizuoto duomenų modeliavimo įrankio prototipas ir jo eksperimentinio tyrimo rezultatai. Taip pat pateikiami sukurtos programos palyginimo su standartiniu MS Visio paketu pagal išskirtus kriterijus rezultatai.

2. DUOMENŲ STATINĖ STRUKTŪRA IR JOS SUDARYMAS

2.1. IS gyvavimo ciklas ir duomenų modelio reikšmė

Duomenų modelis – tai dažniausiai grafinis konkrečios veiklos srities realių duomenų struktūros vaizdas, kuriuo siekiama standartizuoti atskirus veiklos atvejus apibrėžiant duomenų tipus, ilgius, domenų (galimų reikšmių sritis), ryšius, numatomus kardinalumus (galimų atskirų atvejų skaičių) bei apribojimus (privalomi, pasirenkami, unikalūs ir kt.) [6, 7]. Jis yra vienas svarbiausių faktorių, įtakojančių sėkmingą sukurtos IS naudojimą (1 pav.) [8, 9]. Kuo duomenų modelis kokybiškesnis, tuo mažiau finansinių, laiko ir kitų sąnaudų patiriama dėl vėlesniuose IS kūrimo (gyvavimo ciklo) etapuose rečiau pasitaikančių (ar iš viso nepasitaikančių) klaidų, kurių atsiradimą sąlygoja duomenų modelis. Taip pat žymiai sumažėja rizika patirti nuostolių veikloje, kurios procesus palaiko sukurta informacijos sistema [1].



1 pav. Duomenų modelio vaidmuo

Kadangi esminius ir kitus kompiuterizuojamos veiklos procesus palaikančios informacijos sistemos turi būti patikimos ir efektyvios, geros IS sukūrimui būtina suprasti organizacijoje vykstančius procesus ir remiantis tokios analizės rezultatais sudaryti kokybišką duomenų modelį.

2.2. Duomenų modelis ir funkcinių reikalavimų specifikacija

Funkcinių reikalavimų specifikacija – tai vienas pirmųjų IS kūrimo dokumentų, kuriame apibrėžiamas reikalaujamas kuriamos sistemos naudojimas. Šiame dokumente dėmesys sutelkiamas į naudotojo reikmes. Viena iš sistemos funkcinių reikalavimų

specifikacijos dalių yra duomenų loginės struktūros vaizdas. Pastarąjį galima vadinti duomenų struktūrų žemėlapiu, naudojamu kuriant duomenų bazę projektavimo etape [10-12]. Duomenų modelis išgaunamas reikalavimų analizės etape, o pačiai analizei bendriausiu atveju keliami tokie tikslai:

- apibrėžti duomenis, reikalingus duomenų bazės sudarymui, pradiniais objektų terminais;
- suklasifikuoti ir aprašyti objektus;
- nustatyti ir suklasifikuoti ryšius tarp objektų;
- apibrėžti duomenų bazėje vykdomų operacijų tipus ir sąveiką tarp duomenų bei operacijų;
- nustatyti taisykles, valdančias duomenų vientisumą.

Paprastai informacija, reikalinga duomenų analizei, surenkama peržiūrint turimus specifinės kompiuterizuojamos veiklos procesų dokumentus. Tokiais dokumentais gali būti įvairios formos, ataskaitos, gairės, darbų aprašymai, asmeniniai pasakojimai, pastabos. Popierinė dokumentacija yra labai geras būdas susipažinti su organizacija ir jos vykdomomis veiklomis, kurias reikia sumodeliuoti. Taip pat gali būti rengiami individualūs ar grupiniai interviu su galutiniais naudotojais; peržiūrimos esamos automatizuotos sistemos, jei organizacija tokių turi ir pan. [19].

Kaip jau buvo minėta anksčiau šiame darbe, duomenų modelis – tai konceptualus duomenų struktūrų, reikalingų duomenų bazės kūrimui, vaizdas. Duomenų modelį sudaro duomenų objektai, ryšiai tarp jų ir taisyklės, apibrėžiančios bei valdančios veiksmus objektuose ar tarp jų. Duomenų modelyje akcentuojama, kokie duomenys yra reikalingi ir kaip jie sudaryti. Kokie veiksmai bus atliekami su duomenimis čia nėra taip svarbu. Modelis nepriklauso nuo techninės ar programinės įrangos apribojimų. Dažniausiai naudojamas metodas duomenų loginės struktūros sudarymui yra esybių-ryšių (ER) modelis (žr. 2.3. skyrių). Duomenų modeliavimas, kurio tikslas yra užtikrinti tikslų ir pilną reikalingų duomenų pateikimą, labai imlus darbui ir reikalaujantis daug laiko kūrimo procese. Duomenų modelyje turi būti stengiamasi naudoti lengvai suprantamus žymėjimus ir natūralią kalbą, kad galutiniai kuriamos sistemos naudotojai galėtų juos peržiūrėti ir patvirtinti kaip teisingus. Duomenų modelis turi būti pakankamai detalus ir aiškus, kad projektuotojai galėtų jį panaudoti kuriant duomenų bazę. Duomenų modelio informacija naudojama susijusių lentelių, pirminių ir išorinių raktų, procedūrų ir trigerių apibrėžimui (nustatymui). Prastai sukurta duomenų bazė pareikalaus daugiau darbo, laiko, o tuo pačiu ir finansinių sąnaudų ateityje. Be tikslaus reikalingų duomenų suplanavimo sukurtoje duomenų bazėje gali būti

nejtraukti duomenys, reikalingi svarbių ataskaitų kūrimui, taip pat gali būti gaunami neteisingi, prieštaringi ar nepakankami rezultatai [18].

Apskritai kiekvieną informacijos sistemą galima nagrinėti kaip informacinių srautų sistemą, kurioje būtų išskiriami trys pagrindiniai srautų tipai:

- Įeinantis srautas. Duomenų šaltinis – organizacijos objektas, saugantis duomenis, reikalingus kompiuterizuojamoms veiklos funkcijoms atlikti. Tai gali būti organizacijoje cirkuliuojantys dokumentai, žodiniai pranešimai, kiti informacijos nešėjai.
- Išeinantis srautas. Rezultatas – kompiuterizuotos IS funkcionalumo metu formuojamas rezultatas (ekraninė forma, ataskaita, duomenų srautas elektroniniu formatu), kuris iš dalies pakeičia iki kompiuterizuotos IS įdiegimo organizacijoje cirkuliavusius popierinius dokumentus (ataskaitas, suvestines), žodžiu ar kitomis komunikacijos priemonėmis perduodamus informacijos srautus.
- Vidinis srautas. Duomenų srautas – srautas, perduodamas iš vieno duomenų šaltinio kitam duomenų šaltiniui arba rezultatui. Srauto struktūrą sudarantys elementai parodo, kokio duomenų šaltinio atributo reikšmės bus perduodamos ir koks rezultato arba kito duomenų šaltinio atributas jas gaus. Šios reikšmės suformuoja atitinkamo duomenų šaltinio arba rezultato, kuriam perduodamas srautas, atributų reikšmes [5].

Funkcinių reikalavimų specifikacija, sudaryta operuojant šiomis sąvokomis vadinama informacijos (duomenų) srautų specifikacija. Žinant, kad kompiuterizuotą informacijos sistemą galima įvardinti kaip informacijos srautų apdorojimo sistemą, šiuos srautus galima specifiuoti sudarant duomenų srautų specifikaciją. Šios specifikacijos sudarymo procesas pagrįstas informacijos sistemai keliamų funkcinių reikalavimų specifikavimo metodu (žr. 2.4. skyrių) [2].

2.3. Esybių-ryšių schemų modeliavimas

Esybių-ryšių schematinio vaizdavimo metodas yra grafinis būdas vaizduoti esybes, ryšius, atributus. Toks metodas leidžia naudingai modeliuoti naudotojo reikalavimus informacijai. Pagrindine to priežastimi laikoma tai, kad ER diagramą (ERD) lengva suprasti ne tik sistemų analitikams ar duomenų bazių projektuotojams, bet ir naudotojams. ER diagrama naudinga kaip geras bendravimo tarp sistemų kūrimo profesionalų įrankis informacijos reikalavimų išgavimui iš galutinių sistemos naudotojų. Norėdamas sudaryti

duomenų bazę pagal ER diagramą, duomenų bazės projektuotojas turi ne tik imti interviu iš naudotojų, bet ir analizuoti senos sistemos (jei tokia yra) dokumentaciją ir, svarbiausia, funkcinių naujos sistemos reikalavimų specifikaciją. Dauguma tokio pobūdžio dokumentacijos pateikiama natūralia kalba, kurios turinį dešifruoti į duomenų bazės schemą nusakančią ER diagramą pakankamai sunku [13, 14]. ER diagramos leidžia informacijos sistemų projektuotojams ir naudotojams aprašyti ir susieti savo pasaulio supratimą.

Šiame darbe remiamasi esybių ryšių modeliu. ER modelis yra paprastas naudoti, jį lengva suprasti ne tik modeliuotojams, projektuotojams, bet ir analizėje dalyvaujantiems galutiniams kuriamos sistemos naudotojams. ER modelio diagramos – tai vieninga loginių duomenų struktūrų vaizdavimo technika, nepriklausoma nuo bet kokio duomenų modelio [20]. Šis modelis pagrįstas griežtomis matematinėmis teorijomis ir yra konkretus, šiuo metu dažniausiai naudojamas duomenų modelis. Taip pat ši metodika yra viena pirmųjų ir geriausiai išvystyta.

ER modelis gali būti naudojamas net labai sudėtingų programų projektavimo procese. Pagrindiniai modelio komponentai (esybė ir ryšys) apibrėžti semantiniu aibės pagrindu. Todėl modelis lengvai suprantamas tiek projektuotojų, tiek naudotojų. Teoriniai ER modelio pagrindai yra aibių teorija, algebra, logika ir tinklo teorija. Aibių teorijos pagrindu naudojami tokie ER modelio žymėjimai:

- Esybė e ;
- Esybių aibė E , $e \in E$;
- Atributas v ;
- Atributų aibė V , $v \in V$;
- Ryšys r ;
- Ryšių aibė R , $r \in R$.

Ryšių aibė apibrėžiama kaip matematinis esybių aibių ryšys. $R = \{r_1, r_2, \dots, r_n\}$, kur $r_i = [e_{i1}, e_{i2}, \dots, e_{in}] | e_{i1} \in E_1, \dots, e_{in} \in E_n$ [3, 6].

Esybė – tai organizacijos objektų, pasižyminčių panašiomis savybėmis aibė. Atributas – tai reikšmė, išreiškianti kokybines arba kiekybines esybės savybes. Ryšys – tai jungtis tarp esybių.

ER modelis yra plačiausiai pasaulyje naudojama verslo programų projektavimo procese technika. Šį metodą naudoja daugiau nei 85 % organizacijų [17]. Pagrindinės tai sąlygojančios priežastys:

- ER diagramos turi labai paprastą sintaksę.
- Modeliavimo proceso rezultatą galima naudoti duomenų bazės projektavimui.

- Programiniai įrankiai palengvina darbą.
- Pakeitimai projektavimo eigoje reikalauja mažiau sąnaudų [1, 2].

2.4. Duomenų modeliavimas, remiantis informacijos srautų struktūra

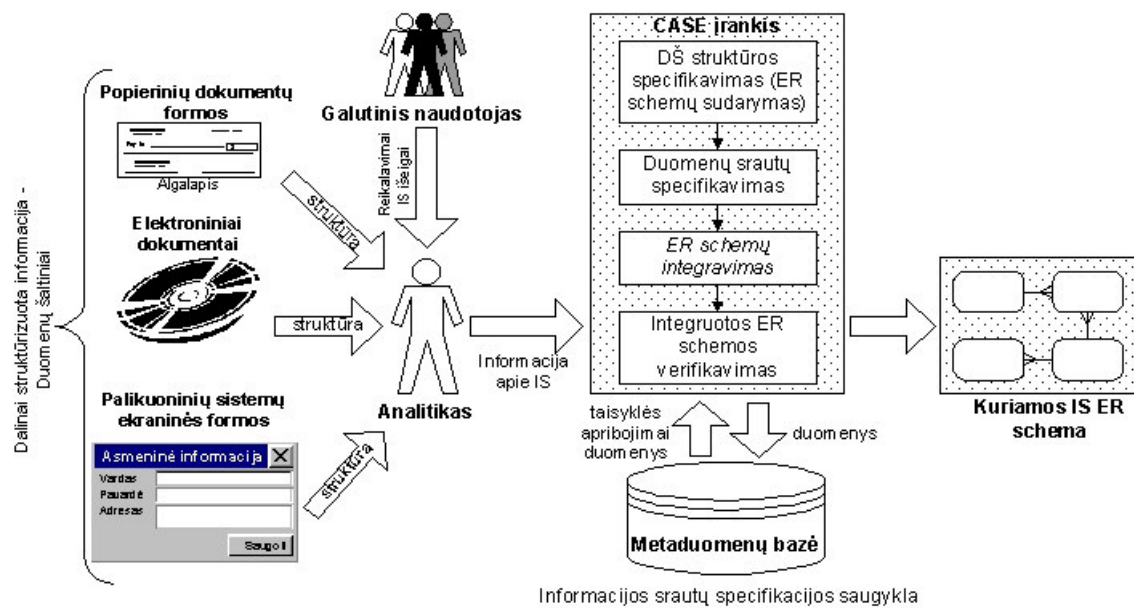
Norėdamas sukurti kokybišką duomenų modelį analitikas privalo gerai suprasti būsimo kuriamos informacijos sistemos naudotojo reikalavimus. Priešingu atveju, IS eksploatavimo metu iškils duomenų prieinamumo, panaudojamumo, teisingumo, vientisumo ir kitų problemų [6, 7].

Suprasti reikalavimus duomenims padeda reikalavimų inžinerija. Kauno technologijos universiteto (KTU) Informacijos sistemų katedra prieš keletą metų pasiūlė ir toliau sėkmingai vysto reikalavimų inžinerijos metodą, kurį taikydamas IS kūrimo procese analitikas analizuoja organizacijoje cirkuliuojančius informacijos srautus (DŠ ir R) ir taip renka reikalavimus kuriamai informacijos sistemai. Pagrindinis šio metodo privalumas yra tai, kad analizė padedama nuo išėtinės informacijos (rezultatų), t.y. nuo būsimojo naudotojo lūkesčių. Būtent laukiami rezultatai lemia, kokius informacijos srautus, t.y. duomenų šaltinius, reikia toliau analizuoti. Duomenų šaltiniai – tai popieriniai ar elektroniniai dokumentai, palikuoninių sistemų ekraninės formos, elektroniniu būdu perduodami duomenų paketai ir pan. Tokia informacija tinkamesnė nei natūrali kalba, nes ji iš dalies jau struktūrizuota. Tai padeda išvengti nesusipratimų tarp analitiko ir būsimojo IS naudotojo.

Palyginus šį metodą su kitais reikalavimų specifikuojamais metodais, (pvz., UML, kai analizė padedama nuo kompiuterizuojamos organizacijos tikslų modelio) galima teigti, kad KTU mokslininkų siūlomas reikalavimų rinkimo būdas yra patrauklesnis, nes jis artimas natūraliam naudotojo lūkesčių išsiaiškinimui. Tačiau yra viena privaloma sąlyga, kuri turi būti tenkinama prieš šio funkcinių reikalavimų kuriamai informacijos sistemai specifikuojamo metodo taikymą. Organizacija jau turi būti veikianti kurį laiką, o jos veiklos – nusistovėjusios.

Taikant šį metodą duomenų modeliavimo procesas išskaidomas į penkis etapus:

- Informacijos srautų (duomenų šaltinių) pavyzdžių surinkimas ir analizė.
- Surinktų duomenų šaltinių struktūrą atitinkančių esybių-ryšių (ER) schemų sudarymas.
- Duomenų srautų specifikuojimas.
- ER schemų integravimas.
- Integruotos ER schemos verifikavimas.



2 pav. IS duomenų modelio sudarymas, remiantis informacijos srautų struktūra – koncepcinė schema

Duomenų modelio kokybė priklauso ne tik nuo pasirinkto reikalavimų specifikavimo metodo, bet nemaža dalimi ir nuo projektuotojo žinių bei patirties. Todėl kaip įmanoma daugiau projektuotojo funkcijų stengiamasi automatizuoti.

Pasiūlymų, kaip automatizuoti duomenų modelio sudarymą organizacijos veikloje naudojamų informacijos srautų pagrindu yra, tačiau geriausias sprendimas dar nerastas. J. Choobineh, J.J. Hwang ir M.V. Mannino, pirmieji iškėlę tokią idėją, pasiūlė dialoginės sistemos variantą, kai ER modelis kuriamas pagal tam tikras organizacijoje naudojamų dokumentų laukų savybes (hierarchinę struktūrą, kardinalumus, užpildymą duomenimis) bei su tais dokumentais susijusius žodinius pranešimus. Vėliau pasiūlymas buvo plėtojamas ir kitų autorių, tačiau labiausiai jį išstobulino B. Wangler. Jo sudarytas algoritmas paremtas naudotojo užpildytų dokumentų formų pavyzdžiais geras tuo, kad į duomenų modelio sudarymo procesą įtraukiamas būsimasis kuriamos IS naudotojas. Tačiau atidžiau panagrinėjus šis metodas nėra patogus, nes naudotojas turi suvesti daug tos pačios dokumento formos pavyzdžių. Šiame darbe pasirinktas duomenų modelio sudarymo būdas nuo B. Wangler plėtoto modelio skiriasi tuo, kad duomenų modeliui sudaryti nenaudojami duomenų šaltinių pavyzdžiai [5].

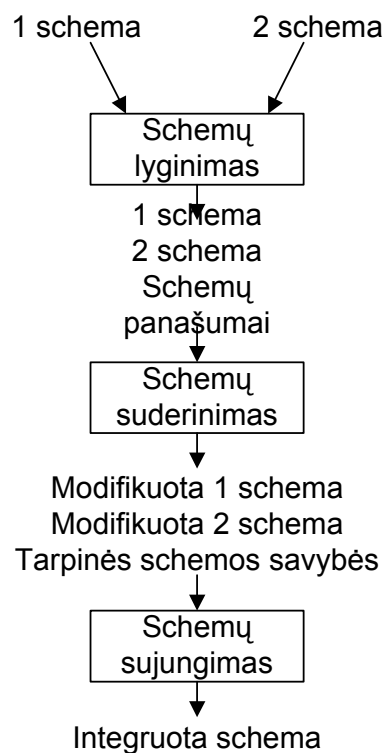
Kuriant didesnes IS įprasta vadovautis „skaldyk ir valdyk“ strategija. Keli projektuotojai ar net jų grupės analizuoja siauresnes kompiuterizuojamos veiklos sritis. Čia, lyginant su kitais, išryškėja dar vienas KTU mokslininkų siūlomo reikalavimų specifikavimo metodo skirtumas. Įprastai, kuriant dideles sistemas, jos išskaidomos į posistemas, o skirtingi asmenys ar grupės analizuoja reikalavimus laukiamam atskirų sistemos dalių funkcionalumui – tiesiog įvardija kompiuterizuotinas funkcijas. Tuo tarpu mūsų metode analizuojami atskirų organizacijos veiklų transakcijas atspindintys DŠ, jų struktūra ir sąsajos, t.y. vykdomi pirmieji

trys duomenų modeliavimo etapai. Šiuo atveju ypač svarbiu tampa ketvirtasis etapas – surinktų duomenų šaltinių struktūrą atitinkančių ER schemų, sudarytų keleto (ar net keliolikos) projektuotojų, integravimas, siekiant gauti vieną bendrą kuriamos IS duomenų modelį [3].

Taigi šiame darbe nuspręsta išnagrinėti ER schemų integravimo automatizavimo galimybes.

2.5. Schemų integravimo procesas

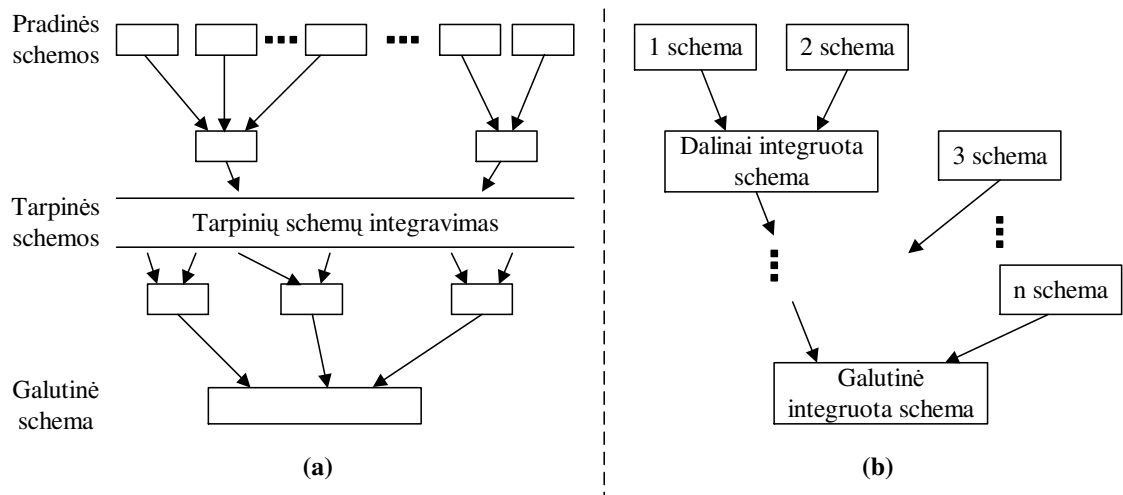
3 paveiksle pateiktas schemų integravimo procesas. Schemų lyginimo etape analizuojami integruojamų schemų panašumai, ypač kreipiamas dėmesys į vienodas, tačiau skirtingai pavaizduotas sąvokas. Šis etapas yra sunkiausias ir reikalauja glaudaus bendravimo tarp integruojamas schemas kūrusių projektuotojų ir būsimųjų sistemos naudotojų. Schemų derinimo metu integruojamos schemas modifikuojamos taip, kad skirtingai pavaizduotos sąvokos būtų suvienodintos. Schemų sujungimo metu integruojamos schemas tiesiog uždedamos viena ant kitos, siekiant gauti vieną bendrą schemą. Šis veiksmas turėtų būti labai nesunkus, jeigu pirmieji etapai buvo atlikti korektiškai.



3 pav. Schemų integravimo procesas

Didelių sistemų kūrimo projektuose gali būti dešimtys ar net šimtai skirtingų schemų, kurias reikia apjungti. Toks didelės apimties integravimo uždavinys reikalauja naudoti tam

tikrą strategiją. Vienas iš galimų integravimo būdų, kai keletas schemų sujungiamos į vieną tarpinę schemą. Tarpinės schemas vėl integruojamos po keletą, kol gaunama galutinė integruota schema (4 pav. (a)). Šios strategijos trūkumas, kad tuo pačiu metu reikia lyginti keletą integruojamų schemų. Remiantis kitu integravimo būdu, vienu metu integruojamos tik dvi schemas. Taip palaipsniui integruojant turimas schemas su gautomis tarpinėmis sukuriama galutinė integruota schema (4 pav. (b)). Šioje strategijoje ypač svarbu pasirinkti schemų integravimo tvarką. Pirmiausiai turėtų būti integruojamos svarbiausios schemas. Svarbiausių išrinkimas turėtų būti atliekamas atsižvelgiant į schemų aktualumą, užbaigtumą, patikimumą. Tokia strategija užtikrina integravimo proceso stabilumą.



4 pav. Schemų integravimo būdai: (a) n-narinis integravimas, (b) binarinis integravimas

2.5.1. Schemų lyginimas

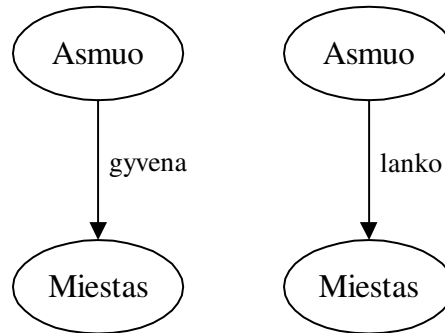
Schemų lyginimo etapo tikslas – aptikti atvejus integruojamose schemose, kai tie patys tikrovės faktai pavaizduoti skirtingai. Šiame etape išskiriamos trys pagrindinės veiklos:

- Vardų lyginimas.
- Struktūrinis lyginimas.
- Tarpinės schemas savybių identifikacija.

Dažnai net ir toje pačioje organizacijoje, tačiau skirtingose srityse dirbantys žmonės vartoja savo terminologiją tiems patiems objektams nusakyti. Tokia realybė daro įtaką vardų pertekliui schemose. Problematiškas sąryšis tarp vardų įvardijamas kaip sinonimas arba homonimas. Sinonimai – tai panašiai įvardyti objektai, kurių prasmė ta pati. Jie dažniausiai nustatomi susipažinus su kontekstu, kuriame objektas naudojamas. Analizuojamas objekto tipo kontekstas: jo atributų aibė, susiję objektų tipai, potipiai, supertipai. Analizuojamas atributo kontekstas: Homonimai – vienodai įvardyti schemų objektai, kurių prasmė skiriasi.

Jie nustatomi, palyginant skirtingoms schemoms priklausančius objektus, turinčius vienodus vardus.

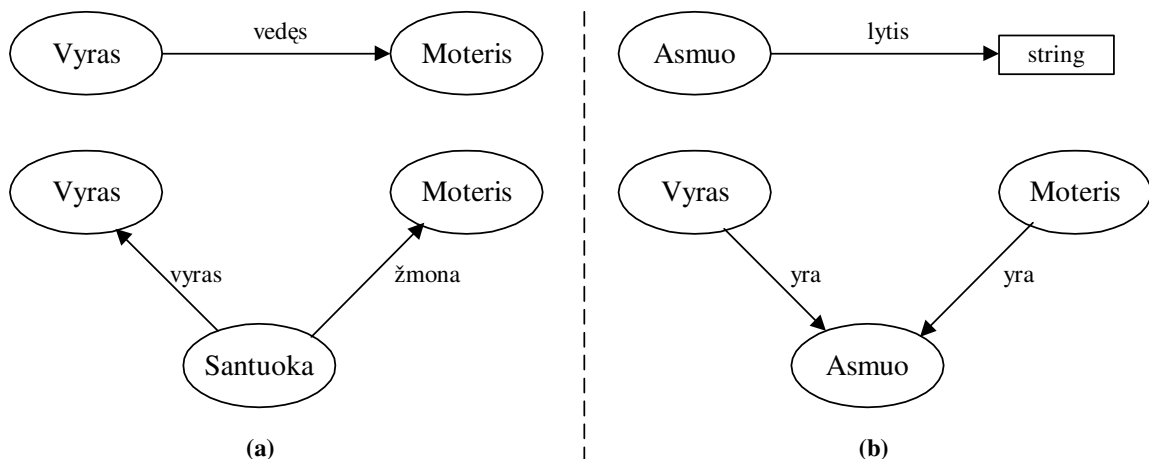
Pasižiūrėjus į 5 paveiksle pateiktas konstrukcijas galima spėti, kad gyvena ir lanko yra sinonimai. Tačiau atributas gyvena gali turėti tik vieną reikšmę, o lanko – daug. Todėl atidžiau išnagrinėjus tampa aišku, kad terminai nėra sinonimai.



5 pav. Atributų, negalinčių būti sinonimais, pavyzdys

Struktūrinius konfliktus aptikti sudėtingiau nei vardų. Struktūrinio lyginimo metu identifikuojami atvejai, kai tas pats realaus pasaulio faktas integruojamose schemose pavaizduotas naudojant skirtingas modelio konstrukcijas. Paprastai šio tipo konfliktai kyla, jeigu sudarant duomenų modelį nebuvo paisoma modeliavimo taisyklių. Žemiau pateikiami svarbiausių ir dažniausiai pasitaikančių tą patį realaus pasaulio faktą vaizduojančių, tačiau struktūriškai skirtingų konstrukcijų pavyzdžiai.

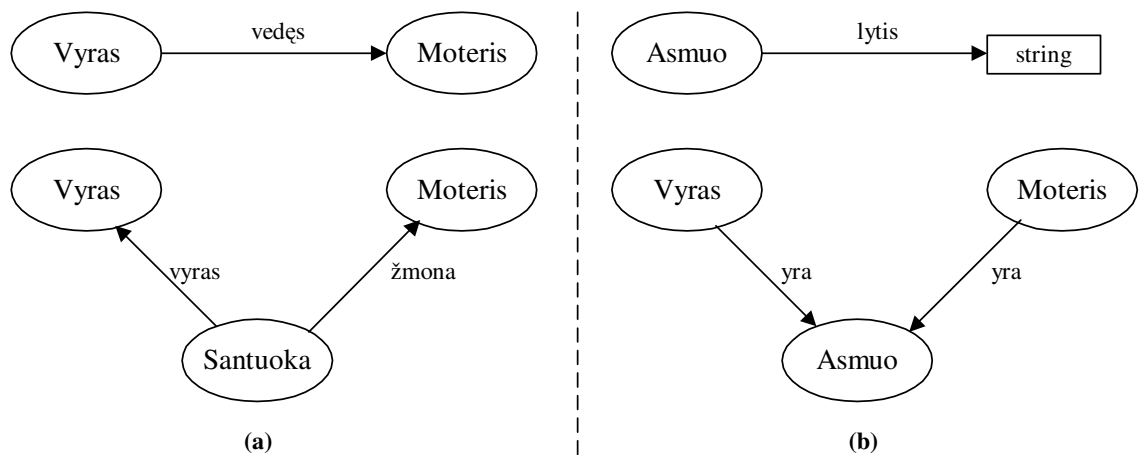
Ryšys tarp objektų gali būti pavaizduotas kaip leksinis (atributas, kategorizuojamas kaip duomenų tipas) arba neleksinis (atributas, kategorizuojamas kaip neleksinis objekto tipas) atributas. 6 paveikslo pirmoji schema leksiniu atributu vaizduoja ryšį tarp knygų ir leidėjų. Antroje schemeje, naudojant papildomą objekto tipą “Leidėjas” ryšys vaizduojamas neleksiniu atributu.



6 pav. Semantiškai ekvivalentių, bet struktūriškai skirtingų konstrukcijų pavyzdžiai (1)

Ryšys tarp realaus pasaulio objektų duomenų modelyje gali būti suprojektuotas tiesiogiai, kaip atributas arba įterpus papildomą objektą, susiejantį tuos realaus pasaulio objektus. Šių skirtingų ryšių pavaizdavimo būdų pavyzdys pateiktas 7 (a) paveiksle. Čia skirtingai sumodeliuota santuokos tarp vyro ir moters situacija. Pirmoje scheme ryšys tarp vyro ir moters pasakomas vienu atributu “vedęs”. Tuo tarpu antroje scheme panaudotas papildomas objektas Santuoka su dviem atributais “vyras” ir “žmona”.

Taip pat yra du būdai konceptualioje scheme vaizduoti ir objektų išskaidymo į kategorijas faktą. Šiuo atveju gali būti naudojami arba tam tikro objekto tipo potipiai, arba atributas su galimomis reikšmėmis, rodančiomis objekto priklausomybę tam tikrai kategorijai. 7 (b) paveiksle pateikiami skirtingi žmonių skirstymo į lytis vaizdavimo modelyje būdai. Pirmoje scheme asmens lytis nusakoma pagal atributo “lytis” reikšmę, o antroje naudojami du atskiri potipiai.



7 pav. Semantiškai ekvivalentių, bet struktūriškai skirtingų konstrukcijų pavyzdžiai (2)

Jeigu integruojamose scheme skirtingos konstrukcijos atrodo ekvivalenčios, reikia išsiaiškinti, ar toms konstrukcijoms taikomi apribojimai neprieštarauja vieni kitiems ir konstrukcijos tikrai ekvivalenčios. Ekvivalentiškumo išaiškinimas tam tikra prasme panašus į vardų palyginimą: tuos pačius apribojimus privaloma taikyti sinoniminiams atributams, objektų tipams.

Ankščiau šiame skyriuje aptartos ekvivalenčios tarpusavyje susijusios dviejų schemų konstrukcijos, t.y. konstrukcijos, kuriomis galima išreikšti tokią pačią informaciją. Tačiau gali būti ir bendresnių analogijų tarp schemų, kai skirtingų schemų konstrukcijoms galioja tam tikri bendri apribojimai. Šios analogijos dažniausiai vadinamos tarpinės schemos savybėmis. Tarpinės schemos savybė – tai formulė, išreikšta probleminių schemų kalbų sąjunga. Galimų tarpinės schemos savybių pavyzdžiai: vienos schemos objekto tipas kitoje scheme yra objekto tipo potipis arba vienos schemos atributas gaunamas iš kitos schemos tam tikros

atributų aibės. Šias savybes reikia detaliai išsiaiškinti schemų lyginimo metu, nes vėliau sujungiant schemas į jas bus atsižvelgiama.

2.5.2. Schemų suderinimas

Būtina schemų integravimo sąlyga – schemų suderinimas, t.y. schemų lyginimo metu nustatytų konfliktų išsprendimas. Sprendžiant konfliktus, schemas modifikuojamos ir taip palengvinamas sujungimo etapo veiksmų atlikimas.

Vardų konfliktai sprendžiami atliekant pervadinimo operacijas. Sinonimų atveju suvienodinami atitinkamų objektų vardai. Atskirais atvejais, kai duomenų modelio, kurį naudoja integravimo metodas, sudarymo taisyklės leidžia, šį konfliktą galima išspręsti išvengiant vardų suvienodinimo. Pavyzdžiui, organizacijos modelyje [3] sinonimų konfliktas sprendžiamas susiejant atitinkamus objektus abipusio paveldėjimo ryšiu. Homonimai pašalinami atitinkamiems objektams suteikiant skirtingus vardus arba prie esamų vardų pridedant priešdėlius. Pasitaikančius raktų konfliktus galima spręsti į integruotą schemą įtraukiant visus raktus arba paliekant integruotoje schemeje tik vieną raktą.

Semantiškai ekvivalenčioms duomenų modelio konstrukcijoms, kurios skiriasi struktūriškai, praktiškai neįmanoma taikyti iš anksto paruoštų griežtų derinimo taisyklių. Dažniausiai tam tikri modelio objektai transformuojami iš vieno tipo į kitą arba sukuriama nauji reikalingi objektai.

2.5.3. Schemų sujungimas

Paskutinis schemų integravimo proceso etapas – schemų sujungimas. Jo metu dvi schemas sujungiamos į vieną. Nors schemas prieš integravimą suderintos, pašalinant neatitikimus, jų sujungimą galima būtų traktuoti kaip paprasčiausią vienos schemas uždėjimo ant kitos veiksmą, konstruojant integruotą schemą atliekamos sujungimo, įtraukimo, susiejimo apibendrinimo ryšiu operacijos.

Batini ir kt. [4] aprašyti integruotos schemas kokybės įvertinimo kriterijai:

- Pilnumas. Integruotoje schemeje turi būti pavaizduota visoms schemoms bendra dalykinė sritis.
- Teisingumas. Integruota schema turi būti teisingai sudaryta iš integruojamų schemų objektų.
- Minimalumas. Jeigu tas pats objektas pavaizduotas keliuose integruojamose schemeose, integruotoje schemeje jis turi būti tik vienas.

- Aiškumas. Integruota schema turi būti lengvai suprantama ir projektuotojui, ir būsimam naudotojui. Tai reiškia, kad iš kelių integravimo rezultatų bus išrinktas labiausiai suprantamas, t.y. aiškiausias.

Gautas integravimo veiksmų rezultatas – integruota schema turi atitikti aukščiau išskirtus ar kitus kokybės kriterijus. Todėl schemas reikia pertvarkyti, pašalinti pertekliškumą, detalizuoti ar pakeisti neaiškias dalis, papildyti schemą naujais objektais ir pan. Integruotoje schemoje gali būti pastebėtos ir suintegruotų pradinių schemų klaidos. Pradinės schemas reikia peržiūrėti ir ištaisyti klaidas. Todėl pats schemų integravimo procesas tampa iteratyviu: įvairia tvarka atliekamas schemų lyginimas, sujungimas, pertvarkymas, klaidų analizė ar perprojektavimas.

2.6. Žinomi schemų integravimo metodai ir modeliai bei galimybės automatizuoti šį procesą

Šiuo metu integravimo metodai pagal specifiką suklasifikuoti į 8 kategorijas:

- Modeliu grindžiami metodai ir euristiniai algoritmai.
- Schemų reinžinerijos metodai.
- Meta duomenų metodai.
- Objektiškai orientuoti metodai.
- Integravimo taikomųjų programų lygyje metodai.
- Integravimo naudojant tarpinę duomenų bazę metodai.
- Dirbtinio intelekto metodai.
- Leksinės semantikos metodai.

Plačiausiai naudojami iš jų yra du: modeliu grindžiami ir meta duomenų metodai. Pirmasis metodas labai mažai automatizuotas ir todėl netinkamas didelės apimties integravimo uždaviniams spręsti. Metaduomenų metodas daugiau skirtas palikuoninėse sistemose specifikuotų schemų integravimui apibrėžiant modelius, kuriuose saugomi metaduomenys [1].

Nors šiame darbe koncentruojamasi ties ER schemų integravimu, tačiau priimant sprendimą galima remtis ir apskritai duomenų bazės schemų integravimo srityje atliktų tyrimų rezultatais bei išvadomis. Pateiksiu keletą mokslinėje literatūroje aprašytų schemų integravimo modelių. ElMasri pirmiausiai siūlo integruojamas schemas transformuoti į esybių-kategorijų-ryšių (EKR) modelį, kuriame be esybių ir ryšių naudojama kategorijų sąvoka. Kategorijos EKR modelyje parodo, kaip aukštesne esybe apibendrinamos žemesnės esybės, arba tiesiog apibrėžia esybės poaibius. Integravimo metu pirmiausiai suintegruojamos

objektų klasės, paskui ryšių klasės ir, galiausiai, sugeneruojamos sąsajos tarp jų. Navathe-Schkolnick modelis, aprašytas Navathe ir Gadgil, sudarytas iš objektų, vaizduojančių esybes ar ryšius, ir jungiamųjų elementų. Pastarieji modeliuoja ryšių savybių įterpimą/pašalinimą ir poabius tarp objektų. Ryšiai išskiriami į tris grupes: nurodantys poabius, atskiriantys ir skirstantys į klases. Integruojant keletą schemų pirmiausiai jos išskiriamos į ekvivalenčius, identiškus ir pavienius vaizdinius. Po to vaizdiniai integruojami nustatant ir išsprendžiant konfliktus. Wiederhold ir ElMasri siūlo prieš integruojant sudaryti struktūrinį modelį, kuriame naudojami ryšiai, vaizduojantys sąsajas tarp esybių klasių. Integruojant du tokius modelius pirmiausiai turėtų būti randamos visos atitinkančių ryšių poros. Tuomet kiekviena pora apjungiama ir gaunama integruota schema [3].

Tokie ir panašūs metodai pateikiami įvairių autorių, tačiau vienas labiausiai sudominusių buvo Kahn pateiktas būtent esybių-ryšių modeliui pritaikytas integravimo būdas. Pati integravimo strategija išskiriama į du etapus: esybių apjungimas ir ryšių apjungimas. Pirmojo etapo metu suformuojamas neperteklinis esybių vaizdinys, antrojo – atsižvelgiant į kardinalumus bei būtinumą apjungiami ryšiai.

Išanalizavus žinomus schemų integravimo metodus bei modelius prieita išvada, kad automatizuotas įrankis turi būti dialoginė sistema, leidžianti analitikui pačiam šiek tiek pakoreguoti integravimo žingsnių seką ar rankiniu būdu patobulinti gautą schemą.

2.7. Prototipo programinės realizacijos vizija

Programinei tyrimo realizacijai bus naudojami įrankiai: MS Visio, MS Access ir MS Visual Basic for Applications. Ši programinė įranga sukurta Microsoft ir veikia plačiau naudojamoje MS Windows operacinėje sistemoje. MS Visio – diagramų kūrimo programa, leidžianti kurti verslo ir technikos diagramas, kuriose sistemingai pateikiamos kompleksinės idėjos, procesai ir sistemos. MS Visio sukurtos diagramos leidžia vizualiai aiškiai, glaustai ir efektyviai pateikti tekstą ir skaičius, kurie atskirai nebūtų tokie informatyvūs [15, 21]. Tyrime bus plečiamas MS Visio funkcionalumas ir programuojama su tai padaryti įgalinančia programa MS Visual Basic for Applications. MS Access šiuo atveju bus panaudota kaip specifikacijos saugykla. MS Access ir MS Visio yra MS Office paketo programos, o MS Visual Basic for Applications užtikrina programinę priėjimą prie kiekvienos MS Office programos funkcionalumo. Kadangi kiekviena MS Office programa pateikia savo funkcionalumą kaip programuojamų objektų rinkinį, Visual Basic for Applications naudojimas leidžia prieiti prie šių objektų ir keisti juos programiškai. Kita Visual Basic for Applications pasirinkimo priežastis: vienoda sintaksė visoms MS Office programoms. Toks

programavimo kalbos vienodumas labai palengvina sprendimų, reikalaujančių atlikti pakeitimus daugiau nei vienoje MS Office programoje, realizavimą [16, 22].

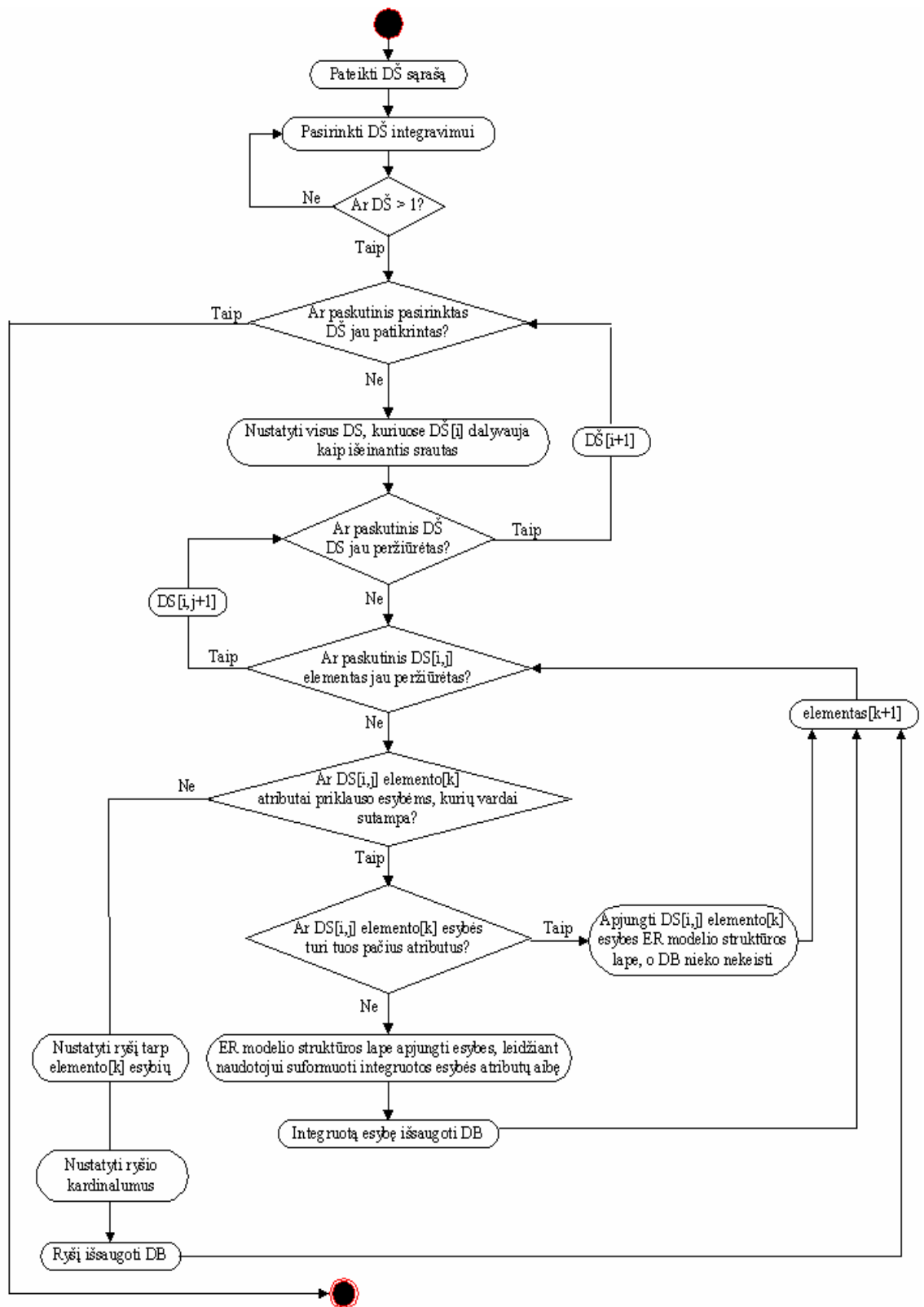
3. ER SCHEMŲ INTEGRAVIMAS DUOMENŲ SRAUTAIS

3.1. ER modelių integravimo informacijos srautų specifikacijos pagrindu algoritmas

Informacijos srautų specifikacijos pagrindu duomenų šaltinių ER modeliai integruojami tokiu būdu:

- 1) Pasirenkami duomenų šaltiniai, kurių ER modeliai bus integruojami. Integravimo procese turi dalyvauti bent du tuo pačiu duomenų srautu susiję duomenų šaltiniai.
- 2) tikrinami visi pasirinkti duomenų šaltiniai. Integravimo procesas baigiamas, patikrinus paskutinį DŠ;
- 3) nustatomi visi duomenų srautai, kuriuose DŠ[i] dalyvauja kaip išeinantis srautas;
- 4) tikrinamas kiekvienas rastas DŠ[i] duomenų srautas DS[i, j]. Jei peržiūrėtas paskutinis to duomenų šaltinio DS[i, j], pereinama prie kito duomenų šaltinio;
- 5) tikrinamas kiekvienas duomenų srauto DS[i, j] elementas. Jei peržiūrėtas paskutinis to duomenų srauto elementas, pereinama prie kito duomenų srauto;
- 6) tikrinama, ar DS[i, j] elemento atributai priklauso esybėms, kurių vardai sutampa. Jeigu taip, tai pereinama prie 7 žingsnio, o jei ne – prie 10 žingsnio;
- 7) tikrinama, ar DS[i, j] elemento atributai priklauso esybėms, kurių atributų aibės sutampa. Jeigu taip, tai pereinama prie 8 žingsnio, o jei ne – prie 9 žingsnio;
- 8) ER modelio struktūros lape esybės apjungiamos į vieną, o duomenų bazėje nedaromi jokie pakeitimai;
- 9) ER modelio struktūros lape esybės integruojamos į vieną, vartotojui leidžiant suformuoti naujos esybės atributų aibę. Integruota esybė išsaugoma duomenų bazėje;
- 10) tarp esybių, kurioms priklauso DS[i, j] elemento[k] atributai, nustatomas ryšys;
- 11) nustatomi naujojo ryšio kardinalumai;
- 12) naujasis ryšys išsaugomas duomenų bazėje.

Grafinis algoritmo vaizdas pateiktas 8 paveiksle.



8 pav. ER modelių integravimo informacijos srautų specifikacijos pagrindu algoritmas

3.2. Informacijos srautų specifikacija – reliacinė metaduomenų bazė

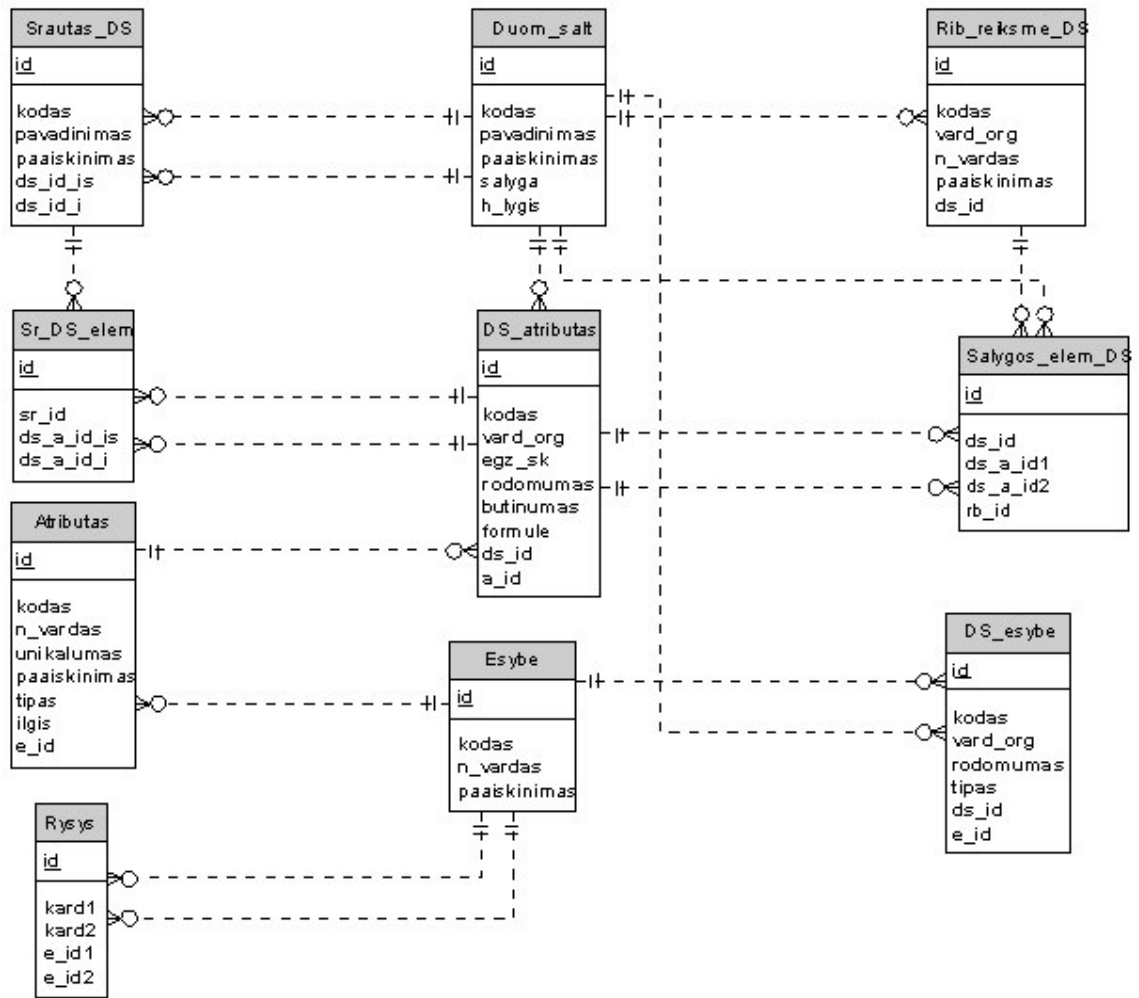
Kaip pagrindas kuriamame įrankyje naudojama informacijos srautų specifikacija, aprašanti kokio tipo ir kokios struktūros turi būti kuriamos IS rezultatus formuojantys duomenų šaltiniai bei juos siejantys duomenų srautai. 9 paveiksle pateikiama informacijos srautų saugyklos struktūra. Lentelėje „Atributas“ saugomi visi nagrinėjamo organizacijos veiklos konteksto atributai. „Esybė“ – visos nagrinėjamo organizacijos veiklos konteksto esybės. „Ryšys“ – visi nagrinėjamo organizacijos veiklos konteksto ryšiai tarp esybių. „Duom_salt“ – informacija apie organizacijos objektus (duomenų šaltinius) saugančius duomenis, reikalingus funkcijoms įvykdyti. „DS_atributas“ – informacija apie organizacijos objektų (duomenų šaltinius) saugančių duomenis, reikalingus funkcijoms įvykdyti, atributus. „Rib_reiksme_DS“ – duomenų šaltinyje esančių atributų egzempliorių ribinių reikšmių informacija. „DS_esybe“ – informacija apie esybes išskirtas iš organizacijos objektų saugančius duomenis, reikalingus funkcijoms įvykdyti. „Salygos_elem_DS“ – informacija apie duomenų šaltinio sudarymo sąlygos elementus. „Srautas_DS“ – informacija apie informacijos srautą tarp duomenų šaltinio ir duomenų šaltinio. „Sr_DS_elem“ – informacija apie informacijos srauto tarp duomenų šaltinio ir duomenų šaltinio sandarą.

Duomenų bazės loginės schemos projektavimui kaip pagrindas naudojama lentelių „Esybe“, „Atributas“ ir „Ryšys“ saugoma informacija, kurios pagrindu sudaromas kuriamos IS duomenų modelio pagrindas. Šiam modeliui gali stigti dalies ryšių arba prireikti detalizuoti esamus. Tai galima atlikti pasitelkiant kitas saugyklos lenteles. Detalizavus modelį gautas duomenų modelis naudojamas kompiuterizuotos IS arba jos prototipo realizavime.

Turima specifikacijos saugykla turi būti užpildyta duomenimis. Saugyklos užpildymo eiga atitinka specifikacijos sudarymo eigą. Kiekvienos lentelės informacija įvedama būtent tokiu eiliškumu, kaip vyksta pats specifikacijos sudarymo procesas. Kiekvienas sekančio žingsnio informacijos įvedimas vyksta pasinaudojant ankstesniu žingsniu išsaugotą informaciją saugykloje.

Specifikavimo metu kiekvienam duomenų šaltiniui pagal tam tikras taisykles sudaromas esybių-ryšių (ER) modelis, apimantis tik to DŠ kontekstą. Toliau specifikuojami duomenų srautai ir jų sudėtis. Duomenų srautus galima įvardinti kaip informacinius ryšius tarp duomenų šaltinių. Viena pagrindinių taisyklių šiems ryšiams specifikuoti yra tai, kad kiekvienam rezultatui formuoti reikalingas bent vienas duomenų šaltinis. Paskutinis etapas yra ryšių, identifikuotų ir aprašytų trečiojo etapo metu perkėlimas į duomenų šaltinių būsenų modelį. Šis etapas leidžia patikslinti jau esančius ryšius tarp duomenų šaltinių. Atlikus visus aukščiau išvardintus etapus sistemos analitikas/projektuotojas turi informacijos srautų

specifikaciją, kurios pagrindu gali būti projektuojamas IS, o pirmiausiai sudaromas duomenų modelis.



9 pav. Informacijos srautų specifikacijos saugyklos struktūra

Toliau šiame skyriuje pateikiami informacijos srautų specifikacijos saugyklos lentelių aprašai.

Lentelėje „Duom_salt“ saugoma informacija apie organizacijos objektus (duomenų šaltinius), saugančius duomenis, reikalingus funkcijoms įvykdyti. Lentelėje fiksuojami duomenys aprašomi 1 lentelėje. Unikalus įrašo id sugeneruojamas DBVS. Unikalią įrašą apibūdinantis kodas sudaromas sistemos naudotojo. Taip pat nurodomas duomenų šaltinio pavadinimas, kuris gali būti paimtas iš nagrinėjamo veiklos konteksto arba sukurtas paties naudotojo. Pateikiama detalizuota informacija apie duomenų šaltinį, žodinė informacija apie sąlygas ir logiką, kurias reikia tenkinti formuojant duomenų šaltinį ir duomenų šaltinio hierarchijos lygis.

1 lentelė. Duomenų srautų saugyklos lentelė „Duom_salt“

Vardas	Duomenų tipas	Pirminis raktas	Būtinai	Unikalūs
id	int	+	+	+
kodas	varchar 10	-	+	+
pavadinimas	varchar 30	-	+	-
paaishkinimas	varchar 200	-	-	-
salyga	varchar 500	-	-	-
h_lygis	int	-	-	-

Lentelėje „Srautas_DS“ saugoma informacija apie informacijos srautą tarp dviejų duomenų šaltinių. Unikalus įrašo id sugeneruoja DBVS. Unikalus įrašą apibūdinantis kodas sudaromas sistemos naudotojo. Nurodomas srauto vardas, kuris gali būti paimtas iš nagrinėjamo veiklos konteksto arba sukurtas paties naudotojo. Pateikiama detali informacija apie srautą ir duomenų šaltinio id, iš kurio išeina srautas, bei duomenų šaltinio id, į kurį įeina srautas.

2 lentelė. Duomenų srautų saugyklos lentelė „Srautas DS“

Vardas	Duomenų tipas	Pirminis raktas	Būtinai	Unikalūs
Id	int	+	+	+
kodas	varchar 10	-	+	+
pavadinimas	varchar 30	-	+	-
paaishkinimas	varchar 200	-	-	-
ds_id_is	int	-	+	-
ds_id_i	int	-	+	-

Lentelėje „Atributas“ saugomi visi nagrinėjamo organizacijos veiklos konteksto atributai. Unikalus įrašo id sugeneruoja DBVS. Unikalus atributą apibūdinantis kodas sudaromas sistemos naudotojo (analitiko/projektuotojo). Atributo vardas gali būti paimtas iš nagrinėjamo veiklos konteksto arba sukurtas paties naudotojo. Nurodomas požymis, rodantis ar atributo reikšmė turi būti unikali; TRUE – reikšmė turi būti unikali, FALSE – reikšmės unikalumas nėra būtinai. Taip pat pateikiama detali informacija apie atributą, atributo duomenų tipas, esybė, kuriai priklauso atributas id.

3 lentelė. Duomenų srautų saugyklos lentelė „Atributas“

Vardas	Duomenų tipas	Pirminis raktas	Būtinai	Unikalūs
id	int	+	+	+
kodas	varchar 10	-	+	+
n_vardas	varchar 50	-	+	-
unikalumas	bool	-	+	-

paaishkinimas	varchar 200	-	-	-
tipas	varchar 30	-	-	-
e_id	int	-	-	-

Lentelėje „Esybe“ saugomos visos nagrinėjamo organizacijos veiklos konteksto esybės. Unikalus įrašo id sugeneruoja DBVS. Unikalus esybę apibūdinantis kodas sudaromas sistemos naudotojo. Esybės vardas gali būti paimtas iš nagrinėjamo veiklos konteksto arba sukurtas paties naudotojo. Taip pat užregistruojama detali informacija apie esybę.

4 lentelė. Duomenų srautų saugyklos lentelė „Esybe“

Vardas	Duomenų tipas	Pirminis raktas	Būtinus	Unikalus
id	int	+	+	+
kodas	varchar 10	-	+	+
n_vardas	varchar 30	-	+	-
paaishkinimas	varchar 200	-	-	-

Lentelėje „Rysys“ saugomi visi nagrinėjamo organizacijos veiklos konteksto ryšiai tarp esybių. Unikalus įrašo id sugeneruoja DBVS. Nurodomas ryšio kardinalumas viename ryšio gale. Kardinalumo aprašymo struktūra: <būtinumas>,<kardinalumas>. Būtinumas gali įgyti šias reikšmes: 0-nebūtinus, 1- būtinus. Kardinalumas gali įgyti šias reikšmes: 1 – vienas, x – daug. Analogiškai nurodomas ir ryšio kardinalumas antrame ryšio gale. Taip pat pateikiami esybių, kurias jungia ryšys id.

5 lentelė. Duomenų srautų saugyklos lentelė „Rysys“

Vardas	Duomenų tipas	Pirminis raktas	Būtinus	Unikalus
id	int	+	+	+
kard1	varchar 3	-	+	-
kard2	varchar 3	-	+	-
e_id1	int	-	+	-
e_id2	int	-	+	-

Lentelėje „DS_atributas“ saugoma informacija apie organizacijos objektų (duomenų šaltinius) saugančių duomenis, reikalingus funkcijoms įvykdyti, atributus. Unikalus įrašo id sugeneruoja DBVS. Unikalus įrašą apibūdinantis kodas sudaromas sistemos naudotojo. Duomenų šaltinio atributo pavadinimas gali būti paimtas iš nagrinėjamo veiklos konteksto. Kiti nurodomi duomenys: konkretaus atributo egzempliorių skaičius įvedamų į konkretų duomenų šaltinį; informacija ar atributas matomas tik specifikacijos modelyje, ar ir originale. (FALSE – originale ir modelyje, TRUE - tik modelyje.); požymis, rodantis ar atributo reikšmės būtinai turi būti įvedamos į duomenų šaltinį. (TRUE – reikšmė turi būti pateikiama, o FALSE - kad reikšmė nebūtina.). Jei atributo reikšmė yra skaičiuojama, išsaugoma

formulė, pagal kurią gali būti skaičiuojam atributo reikšmė. Nurodomas duomenų šaltinio id, kuriam priklauso atributas ir sistemos atributo id, su kuriuo siejamas duomenų šaltinio atributas, nes skirtinguose rezultatuose ar duomenų šaltiniuose tas pats sistemos atributas gali būti skirtingai įvardintas.

6 lentelė. Duomenų srautų saugyklos lentelė „DS_atributas“

Vardas	Duomenų tipas	Pirminis raktas	Būtinas	Unikalus
id	int	+	+	+
kodas	varchar 10	-	+	+
vard_org	varchar 50	-	+	-
egz_sk	int	-	+	-
rodomumas	bool	-	+	-
butinumas	bool	-	+	-
formule	varchar 200	-	-	-
ds_id	int	-	+	-
a_id	int	-	+	-

Lentelėje „Rib_reiksme_DS“ saugoma duomenų šaltinyje esančių atributų egzempliorių ribinių reikšmių informacija. Unikalus įrašo id sugeneruoja DBVS. Unikalus įrašą apibūdinantis kodas sudaromas sistemos naudotojo. Reikšmės pavadinimas gali būti paimamas iš nagrinėjamo veiklos konteksto. Reikšmė, kurią IS siūlo naudoti naudotojas, tai gali būti ir ta pati reikšmė paimta iš nagrinėjamo veiklos konteksto. Taip pat pateikiama detalizuota informacija apie reikšmę; duomenų šaltinio id, kuriame naudojama ši reikšmė.

7 lentelė. Duomenų srautų saugyklos lentelė „Rib_reiksme_DS“

Vardas	Duomenų tipas	Pirminis raktas	Būtinas	Unikalus
id	int	+	+	+
kodas	varchar 10	-	+	+
vard_org	varchar 50	-	+	-
n_vardas	varchar 30	-	+	-
paaiskinimas	varchar 200	-	-	-
ds_id	id	-	+	-

Lentelėje „DS_esybe“ saugoma informacija apie esybes išskirtas iš organizacijos objektų saugančius duomenis, reikalingus funkcijoms įvykdyti. Unikalus įrašo id sugeneruoja DBVS. Unikalus įrašą apibūdinantis kodas sudaromas sistemos naudotojo. Esybės pavadinimas paimamas iš nagrinėjamo veiklos konteksto. Nurodoma, ar esybė matoma tik specifikacijos modelyje, ar ir originale. (FALSE – originale ir modelyje, TRUE - tik modelyje.) Pateikiamas duomenų šaltinio id, kuriam priklauso esybė, sistemos esybės id, su kuria siejama duomenų šaltinio esybė, nes skirtinguose rezultatuose ar duomenų šaltiniuose ta pati sistemos esybė gali būti skirtingai įvardinta.

8 lentelė. Duomenų srautų saugyklos lentelė „DS_esybe“

Vardas	Duomenų tipas	Pirminis raktas	Būtinasis	Unikalus
id	int	+	+	+
kodas	varchar 10	-	+	+
vard_org	varchar 50	-	+	-
rodomumas	bool	-	+	-
ds_id	int	-	+	-
e_id	int	-	+	-

Lentelėje „Salygos_elem_DS“ saugoma informacija apie duomenų šaltinio sudarymo sąlygos elementus. Unikalus įrašo id sugeneruojamas DBVS. Pateikiami duomenų šaltinio, kuriam formuojama sąlyga id, duomenų šaltinio atributo id, kuris įtakoja sąlygos, duomenų šaltinio atributo id, kuris įtakoja duomenų šaltinio atributą. Taip pat nurodomas duomenų šaltinio ribinės reikšmės id, kuri įtakoja rezultato atributą.

9 lentelė. Duomenų srautų saugyklos lentelė „Salygos_elem_DS“

Vardas	Duomenų tipas	Pirminis raktas	Būtinasis	Unikalus
id	int	+	+	+
ds_id	int	-	+	-
ds_a_id1	int	-	+	-
ds_a_id2	int	-	-	-
rb_id	int	-	-	-

Lentelėje „Sr_DS_elem“ saugoma informacija apie informacijos srauto tarp duomenų šaltinio ir duomenų šaltinio sandarą. Unikalus įrašo id sugeneruojamas DBVS. Nurodomas srauto id. Pateikiami duomenų šaltinio iš kurio išeina srautas, atributo id ir duomenų šaltinio atributo id, į kurį pereina informacija iš duomenų šaltinio atributo.

10 lentelė. Duomenų srautų saugyklos lentelė „Sr_DS_elem“

Vardas	Duomenų tipas	Pirminis raktas	Būtinasis	Unikalus
id	int	+	+	+
sr_id	int	-	+	-
ds_a_id_is	int	-	+	-
ds_a_id_i	int	-	+	-

Toliau šiame skyriuje pateikiami užpildytos informacijos srautų specifikacijos saugyklos MS Access ekraniniai vaizdai su trumpais komentarais.

10 paveiksle pateikiamas saugyklos lentelių „Esybe“ ir „Atributas“ vaizdas. Pirmiausiai lentelėje „Esybe“ užregistruojama informacija apie kompiuterizuojamos veiklos objektus, išskirtus kaip esybes. Lentelėje „Atributas“ užregistruojami atributai ir susiejami su konkrečiomis esybėmis lentelėje „Esybe“.

The screenshot shows the Microsoft Access interface with two tables displayed in a grid view. The top table is 'Esybe' and the bottom table is 'Atributas'.

id	kodas	n_vardas	paaiskinimas
1	e1	Darbuotojas	

id	kodas	n_vardas	unikalumas	paaiskinimas	tipas	ilgis
1	a1	Tab_nr	True		integer	
2	a2	Vardas	False		text	20
3	a3	Pavarde	False		text	20
4	a4	Pareigos	False		text	15

2	e2	Mokejimo_rusis	
3	e3	Filialas	
4	e4	Gaminys	
5	e5	Uzsakymas	
6	e6	Gaminio_furnitura	
7	e7	Gaminio_paslauga	
8	e8	Gaminio_zaliava	
9	e9	Zaliava	
10	e10	Paslauga	
11	e11	Furnitura	
12	e12	Uzsakovas	
13	e13	Kasos_aparatas	
14	e14	PVM_saskaita_faktura	
15	e15	Darbuotojas	
16	e16	Masina	
17	e17	Pirkejas	
18	e18	Gaminys	
19	e19	Nurasymo_aktas_darbuotojas	
20	e20	Nurasymo_aktas	
21	e21	Darbuotojas	
22	e22	Zaliava	
23	e23	Nurasoma_zaliava	
24	e24	Sandelis	

10 pav. Informacijos srautų saugykla – užpildytos „Esybe“ ir „Atributas“ lentelės

11 paveiksle pateikiama informacija apie kiekvieną kompiuterizuojamos veiklos esybės jungiantį ryšį.

	id	kard1	kard2	e_id1	e_id2	identifikuojantis
▶	1	1,1	0,x	1	5	<input type="checkbox"/>
	2	1,1	0,x	2	5	<input type="checkbox"/>
	3	1,1	0,x	3	5	<input type="checkbox"/>
	4	1,1	1,x	5	4	<input type="checkbox"/>
	5	1,1	1,x	4	7	<input checked="" type="checkbox"/>
	6	1,1	1,x	4	8	<input checked="" type="checkbox"/>
	7	1,1	0,x	4	6	<input checked="" type="checkbox"/>
	8	1,1	0,x	9	8	<input checked="" type="checkbox"/>
	9	1,1	0,x	10	7	<input checked="" type="checkbox"/>
	10	1,1	0,x	11	6	<input checked="" type="checkbox"/>
	11	1,1	1,x	12	5	<input type="checkbox"/>
	12	1,1	0,x	13	14	<input type="checkbox"/>
	13	1,1	0,x	15	14	<input type="checkbox"/>
	14	1,1	0,x	15	14	<input type="checkbox"/>
	15	1,1	0,x	15	14	<input type="checkbox"/>
	16	1,1	0,x	16	14	<input type="checkbox"/>
	17	1,1	0,x	17	14	<input type="checkbox"/>
	18	1,1	0,x	14	18	<input type="checkbox"/>
	19	1,1	1,x	20	19	<input checked="" type="checkbox"/>
	20	1,1	0,x	21	19	<input checked="" type="checkbox"/>
	21	1,1	1,x	21	20	<input type="checkbox"/>
	22	1,1	0,x	22	23	<input checked="" type="checkbox"/>
	23	1,1	1,x	20	23	<input checked="" type="checkbox"/>
	24	1,1	0,x	24	20	<input type="checkbox"/>
*						<input type="checkbox"/>

Record: 1 of 24
id identifies Rsys NUM

11 pav. Informacijos srautų saugykla – užpildyta „Rsys“ lentelė

12 paveiksle pateikiamas lentelių „Duom_salt“ ir „DS_esybe“ vaizdas. Lentelėje „Duom_salt“ užregistruota pagrindinė duomenų šaltinių informacija. Anksčiau pristatytoje lentelėje „Esybe“ saugomos visos išskirtos kompiuterizuojamos veiklos esybės. Per lentelėje „DS_esybe“ registruojamus duomenis, kiekviena nagrinėjamo konteksto esybė priskiriama tam tikram duomenų šaltiniui.

	id	kodas	pavadinimas	paaiskinimas	salysga	h_lygis
-	1	ds1	Uzsakymas			1
+	2	ds2	PVM_saskaita_faktura			2
+	3	ds3	Nurasymo_aktas			1
*						

Record: 2 of 3

h_lygis is of Duom_salt

NUM

12 pav. Informacijos srautų saugykla – užpildytos „Duom_salt“ ir „DS_esybe“ lentelės

13 paveiksle pateikiamos užpildytos „Srautas_DS“ ir „Sr_DS_elem“ lentelės. Kad galėtų būti automatiškai integruojami, duomenų šaltiniai turi būti tarpusavyje susiję. Kiekvienas DŠ, užregistruotas lentelėje „Duum_salt“ susiejamas su tam tikru kitu duomenų šaltiniu. Šios sąsajos aprašomos lentelėje „Srautas_DS“. Tačiau vien tik srauto, siejančio du duomenų šaltinius, nepakanka. Tai būtų per daug abstraktu. Todėl papildomoje informacijos srautų specifikacijos saugyklos lentelėje „Sr_DS_elem“ kiekvienas srautas detalizuojamas konkrečiais atributais, nešančiais informaciją iš vieno DŠ į kitą.

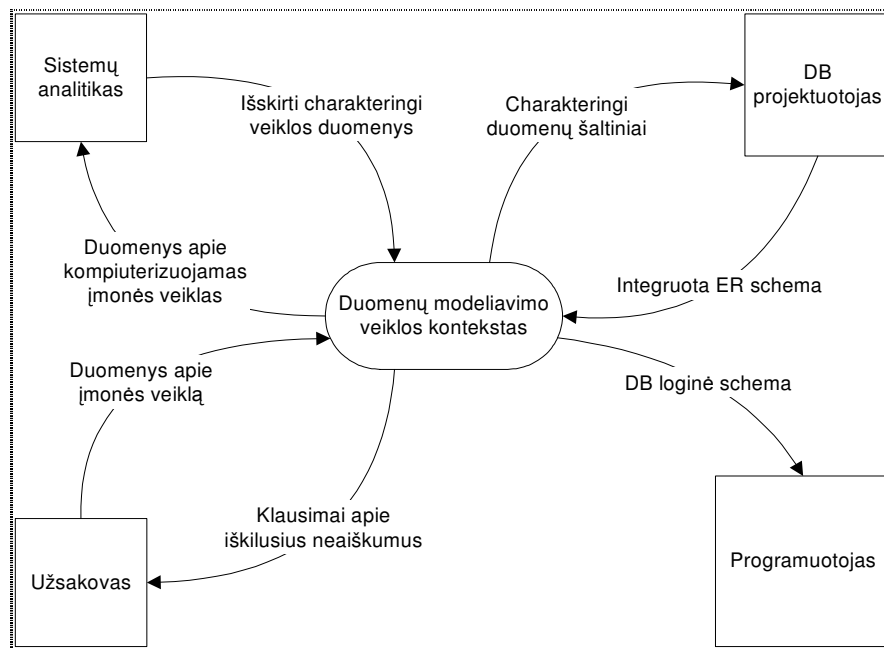
id	kodas	pavadinimas	paaiskinimas	ds_id_is	ds_id_i
1	s1	pirmas	tarp užskymo ir	1	2
169				1	69
270				2	70
371				3	71
472				4	72
4774				47	74
4875				48	75
1178				11	78
1279				12	79
1780				17	80
1881				18	81
1982				19	82

13 pav. Informacijos srautų saugykla – užpildytos „Srautas_DS“ ir „Sr_DS_elem“ lentelės

3.3. Pagrindiniai reikalavimai ir apribojimai prototipui

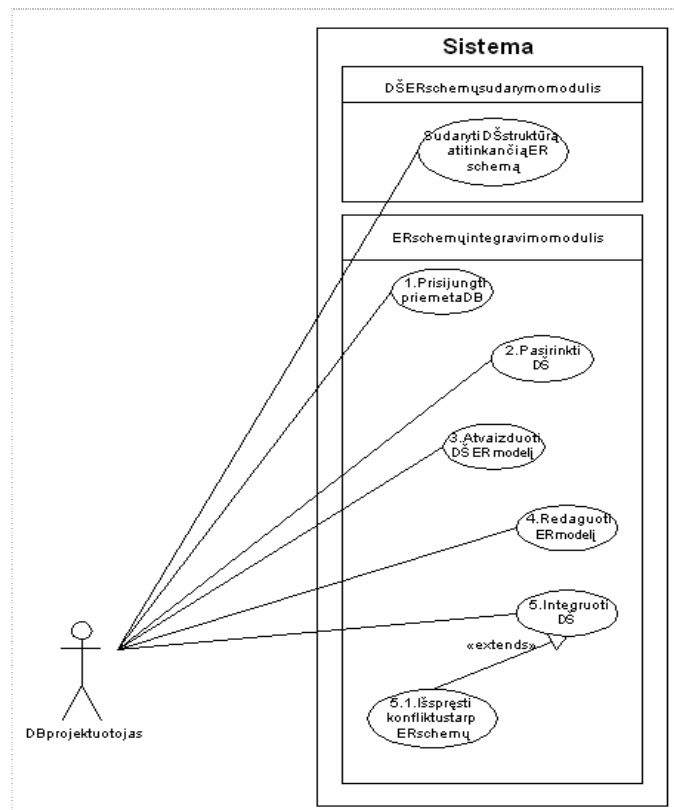
Šiame skyriuje pirmiausiai trumpai pristatomas šiame darbe pristatomo duomenų modeliavimo kontekstas: užsakovas pateikia duomenis apie įmonės veiklą (organizacijos veiklos standartai, įvairūs dokumentai, naudojami žodiniai pranešimai ir pan.); iš užsakovo gautus duomenis apie kompiuterizuojamas įmonės veiklas analizuoja sistemų analitikas; išskiriami charakteringi dokumentai – duomenų šaltiniai; DB projektuotojas gauna analizės duomenis apie charekteringus kompiuterizuojamos veiklos duomenų šaltinius; pagal turimus duomenis projektuojamas kuriamos sistemos duomenų modelis; galiausiai gaunamas galutinis duomenų modeliavimo veiklos rezultatas – integruota ER schema. Sistemos kūrėjai gali

užsakovui bet kada pateikti analizės bei projektavimo metu iškilusius klausimus. Gautas duomenų modelis toliau naudojamas kituose IS kūrimo etapuose.



14 pav. Duomenų modeliavimo veiklos kontekstas

15 paveiksle pateikiami kuriamo prototipo panaudojimo atvejai. Panaudojimo atvejų diagramoje tam, kad geriau būtų suprantamas kompiuterizuojamos veiklos kontekstas, pateikiamas ir anksčiau realizuotas DŠ ER schemų sudarymo modulis. Šiame darbe bus realizuojami penki naujo ER schemų integravimo modulio panaudojimo atvejai.

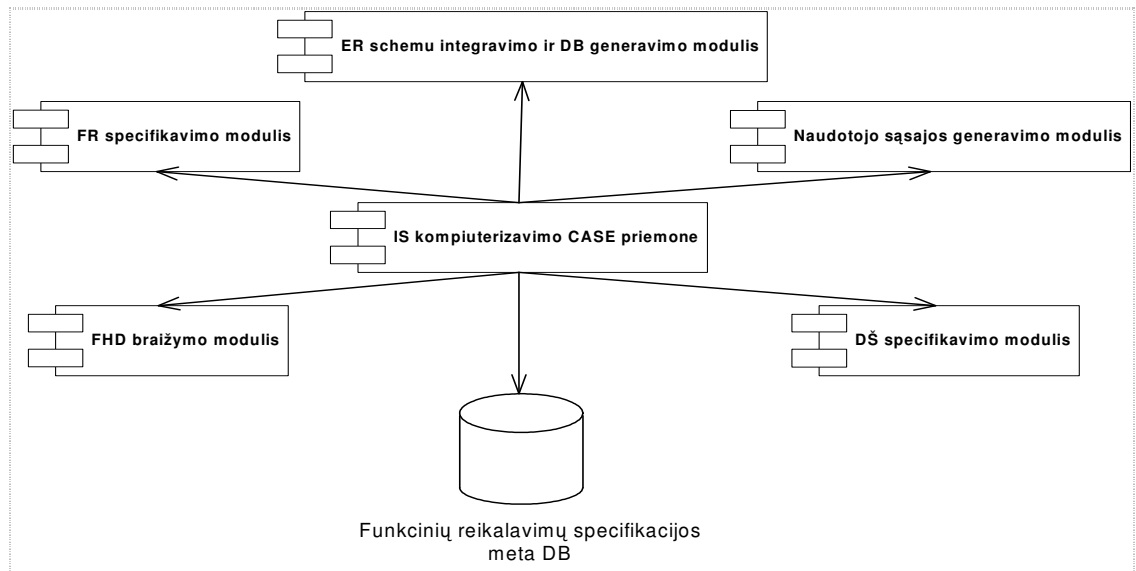


15 pav. Prototipo panaudojimo atvejų diagrama

Kiti svarbūs reikalavimai kuriamam įrankiui:

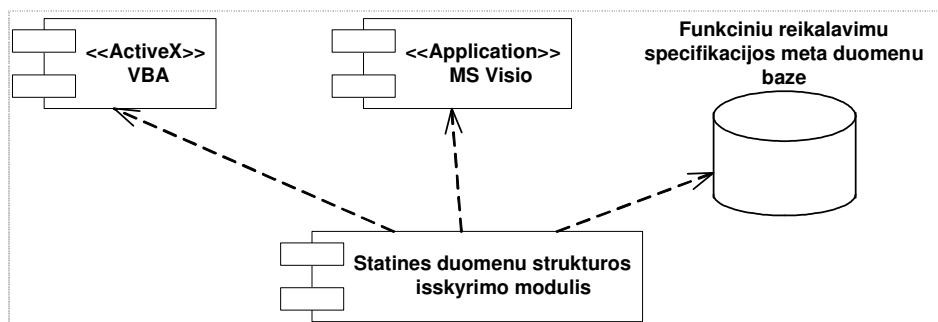
- Lengvai skaitoma sąsaja;
- Imituojanti MS Visio tipinę sąsają;
- Neįkyri sąsaja;
- Nesudėtingas panaudojimas.
- Paprastas naudotis IT inžinieriams (įprasti žymėjimai);
- Paprastai panaudojamas IT specialistų be apsimokymo;
- Nacionalinės kalbos panaudojimas;
- Sutrumpinimų panaudojimas;
- Veiklos našumo prieaugis dėl sistemos diegimo.
- Integravimo proceso rezultatas turi būti patikimas.

16-jame paveiksle pateikta bendra IS kompiuterizavimo CASE priemonė, kur turės būti integruotas kuriamas ER schemų integravimo modulis.



16 pav. IS kompiuterizavimo CASE priemonė

17 paveiksle pateikta modulio komponentų diagrama.

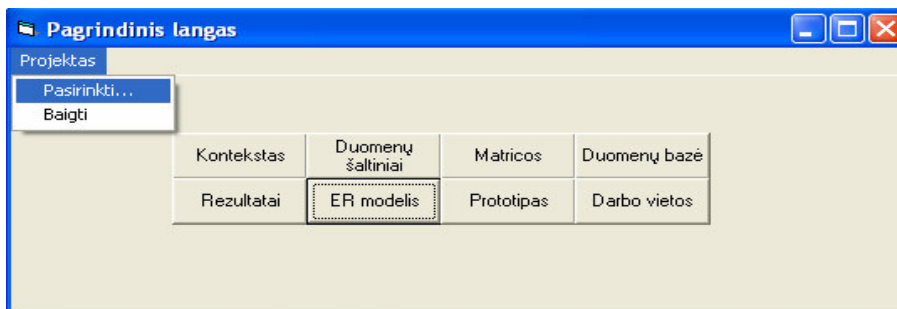


17 pav. Komponentų diagrama

4. PROTOTIPAS IS DUOMENŲ MODELIUI SUDARYTI

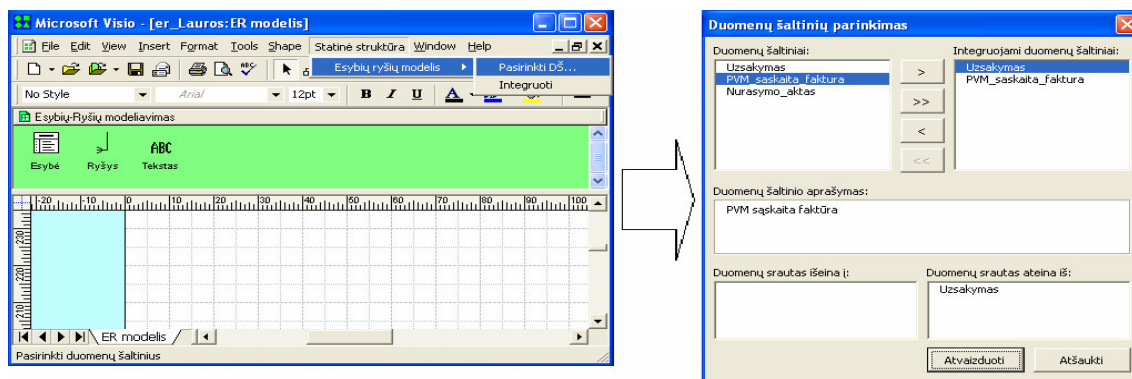
4.1. Trumpas prototipo aprašymas ir ekraniniai vaizdai

KTU Informacijos sistemų katedros mokslo grupė siekia automatizuoti ir kitus IS kūrimo etapus. Visi moduliai yra iškviečiami iš pagrindinės ekraninės formos „Pagrindinis langas“ (18 pav.). Programinis prototipas IS duomenų modeliui sudaryti taip pat iškviečiamas iš šios formos prieš tai pasirinkus metaduomenų bazę, kurioje saugomos duomenų šaltinių struktūrą atitinkančios ER schemas.



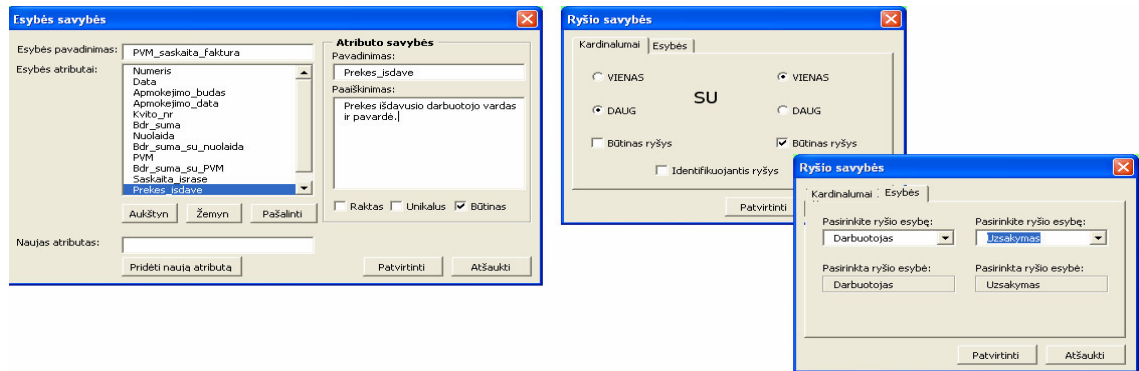
18 pav. Pagrindinis prototipo langas

Pasirinkus meta duomenų bazę atidaromas MS Visio langas su papildomu meniu punktu ir specialia įrankių juosta, skirta esybių-ryšių modelio elementams vaizduoti (19 pav.). Meta duomenų bazėje saugomos ER schemas integruojamos ne visos iš karto, o poromis arba grupėmis. Tai nusprendžia vartotojas. Jis nurodo, kokių DŠ ER schemas integruoti, o sistema patikrina, ar jo sprendimas teisingas, t.y. ar šie duomenų šaltiniai yra susiję duomenų srautu.



19 pav. MS Visio programinio paketo langas ir duomenų šaltinių parinkimo langas

Pasirinkus integruojamus DŠ jų ER schemas pavaizduojamos MS Visio lape (žr. 4.2 skyrių, 21 pav.). Kiekviena atskiros ar jau suintegruotos ER schemas esybė ar kiekvienas ryšys gali būti redaguojami rankiniu būdu (20 pav.).

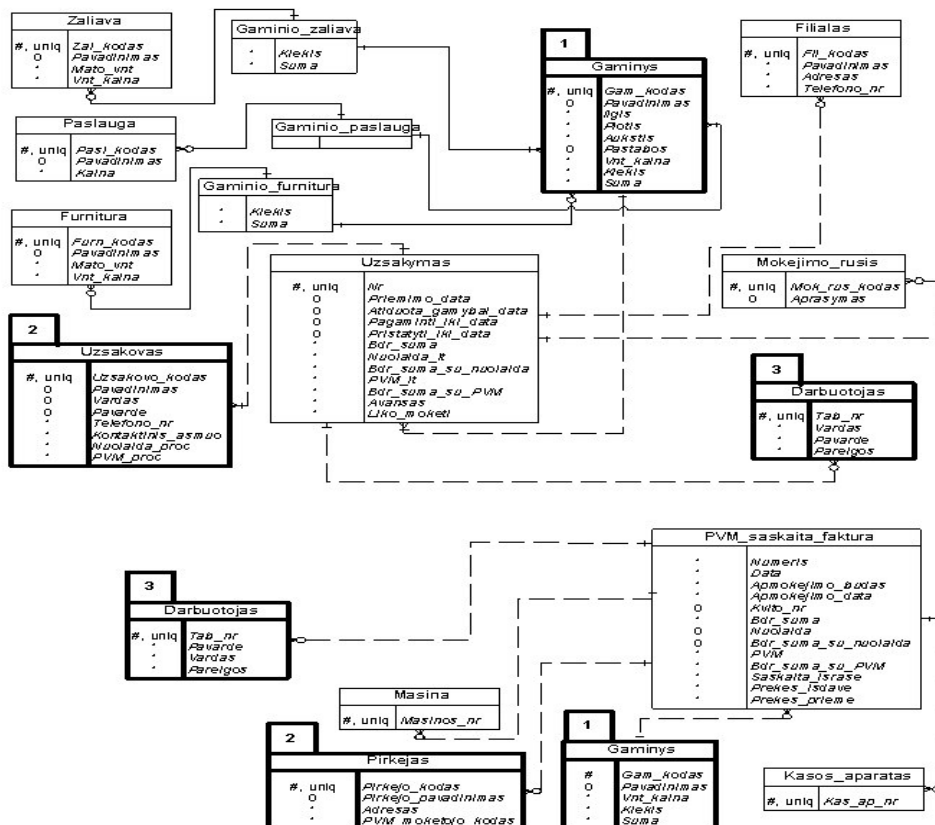


20 pav. Esysbės ir ryšio savybių redagavimo langai

4.2. Prototipo veikimo eksperimentiniai rezultatai

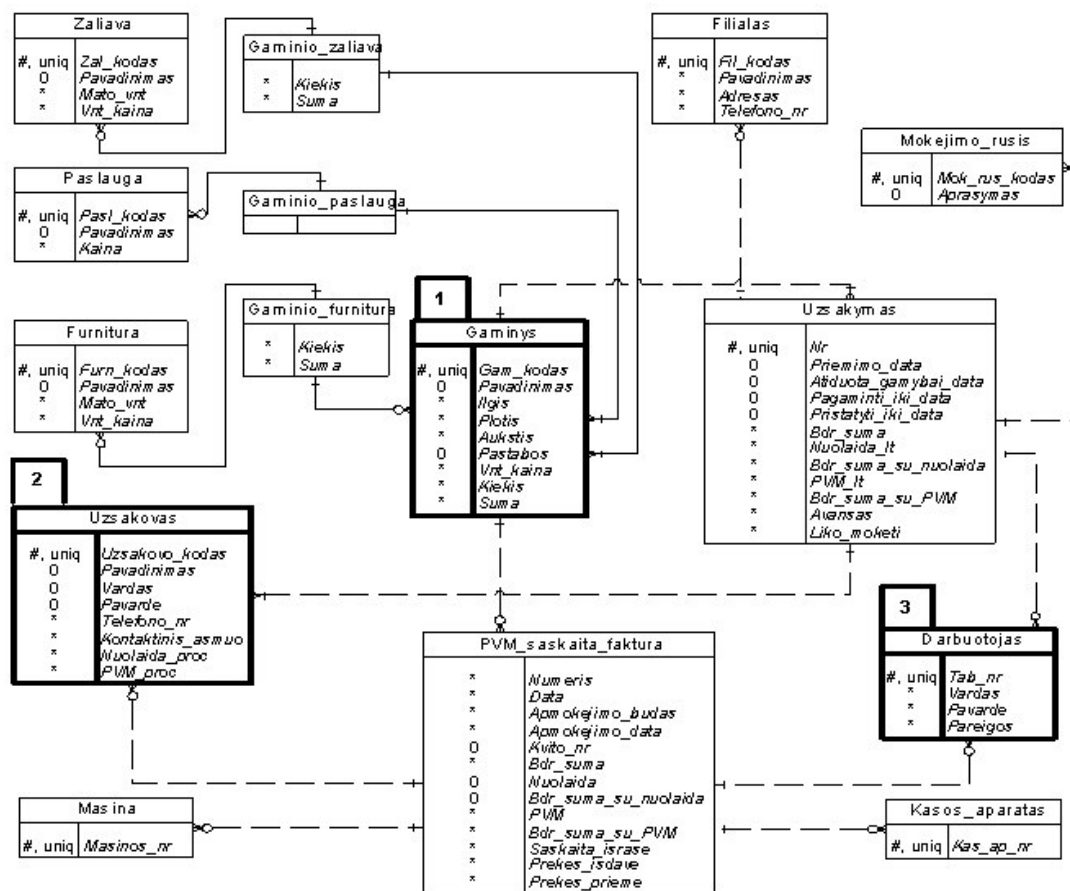
Sukurto prototipo veikimas buvo testuojamas su realiais duomenimis užpildyta informacijos srautų saugykla.

Šiame skyriuje pateikiamos ER schemas, nukopijuotos iš prototipo MS Visio lapo prieš ir po integravimo. Prototipo veikimas buvo patikrintas, naudojantis testiniais duomenimis užpildyta metaduomenų baze. 21 paveiksle pateiktos dviejų duomenų šaltinių – „Užsakymas“ ir „Sąskaita-faktūra“ – ER schemas MS Visio lange. Panašios schemų esybės yra paryškintos ir atitinkamai sunumeruotos.



21 pav. Atvaizduota DŠ „Užsakymas“ ir „Sąskaita – faktūra“ struktūra

22 paveiksle pateikiama suintegruota ER schema. Panaikintas vienodų esybių dubliavimasis, atitinkamai perkelti ryšiai.



22 pav. Integruota ER schema, apimanti DŠ „Užsakymas“ ir „Sąskaita – faktūra“ ER schemas

4.3. Prototipo vertinimas standartinio MS Visio programinio paketo kontekste

Kadangi atliekant šio darbo analizę nebuvo rasta šiuo metu egzistuojančių komercinių ar tiriamųjų programų, integruojančių ER schemas, sukurto prototipo lyginimas su kitais projektavimo įrankiais nėra lygiavertis. Programinei realizacijai buvo pasirinktas MS Visio paketas, todėl sukurtas produktas vertinamas šio projektavimo įrankio kontekste.

Prototipas informacijos sistemos duomenų modeliui sudaryti integruotas į standartinį MS Visio paketą. Taigi liko visos ankstesnės pastarojo paketo funkcijos, leidžiančios kurti verslo ir technikos diagramas, kuriose sistemingai pateikiamos kompleksinės idėjos, procesai ir sistemos. Naujų funkcijų pridėjimas nepadarė neigiamos įtakos įrankio naudojimui. Atliekant šį komercinį paketą įeinančius veiksmus neiškyla jokių programinių klaidų, nulemtų funkcionalumą praplečiančio programinio kodo.

Naujas funkcijas lengva įsisavinti, nes kartu su prototipu pateikiamas ir naudotojo vadovas. Taip pat sukurto įrankio grafinė naudotojo sąsaja imituoja standartinę MS Visio

sąsają. Sąsaja sukurta lietuvių kalba, siekiant išvengti kalbos barjero keliamų keblumų. Žinoma, neįgudęs naudotojas sunkiau įsisavins naują funkcionalumą, tačiau IT inžinieriams tai turėtų būti lengva, nes naudojami standartiniai elementai.

MS Visio pasižymi tolerantiškumu klaidoms, o prototipe ši savybė silpnesnė. Tolerantiškumą klaidoms galima apibūdinti kaip programos gebėjimą nepertraukiamai dirbti tarp atsirandančių klaidų. Taip pat tolerantiškumo klaidoms rodiklis yra ir sistemos reikalavimas ištaisyti nekorektiškai įvestus duomenis rankiniu būdu arba automatinis jų ištaisymas. Taip siekiama išvengti nenumatytų programos sutrikimų. Jeigu nekorektiški duomenys praleidžiami, vėliau vykdant kitus veiksmus sistema gali pateikti naudotojui nesuprantamus pranešimus arba apskritai „nulūžti“. Šiuo aspektu MS Visio programa yra pranašesnė už sukurtą prototipą. Nors ankstesniojo funkcionalumo prototipas ir neįtakoja, tačiau atliekant naujus galimus veiksmus pasitaiko nenumatytų atvejų.

Nors MS Visio pakete yra daugybė įvairių šablonų skirtingoms diagramoms braižyti, sukurtas naujas šablonas palengvina projektuotojui darbą savo konkretumu. Išskirtinai tik ER diagramų elementai leidžia negaišti laiko galvojant kurį elementą pasirinkti.

Prototipas dėl savo programinės struktūros gali būti nesunkiai plečiamas arba sujungiamas su kitais MS Visio funkcionalumą papildančiais moduliais.

Naudojantis tiek standartinio MS Visio, tiek prototipo funkcijomis sukurti dokumentai gali būti eksportuojami į MS Access, MS SQL ar kitas duomenų bazių valdymo sistemas. Taip pat yra galimybė importuoti į MS Visio kitų programų bylas. Prototipo atveju ne tik importuojami duomenys iš informacijos šrautų specifikacijos saugyklos MS Access, bet ir pagal apibrėžtus apribojimus pavaizduojama ER schema.

Įvairių funkcijų praplėtimo ir tyrimo galimybės su MS Visio yra gana didelės. Norint sukurtą prototipą IS duomenų modeliui sudaryti taikyti realiai vykdomuose projektuose, reikalingas tolesnis darbo plėtojimas.

5. IŠVADOS

1. Kadangi esminius ir kitus kompiuterizuojamos veiklos procesus palaikančios informacijos sistemos turi būti patikimos ir efektyvios, geros IS sukūrimui būtina suprasti organizacijoje vykstančius procesus ir, remiantis tokios analizės rezultatais, sudaryti kokybišką duomenų modelį.
2. Kuriamos informacijos sistemos duomenų loginė struktūra turi būti pateikta funkcinių reikalavimų specifikacijoje. Kadangi dokumentas įprastai derinamas su galutiniu naudotoju, duomenų modelį tikslinga vaizduoti plačiai paplitusia, lengvai įsisavinama ir suprantama esybių-ryšių schema.
3. Duomenų modeliavimas remiantis organizacijoje cirkuliuojančių informacijos srautų struktūra lemia didesnę sudaryto duomenų modelio kokybę, nes analizuojama iš dalies jau struktūrizuota informacija (tinkamesnė nei natūrali kalba). Tuo pačiu toks reikalavimų surinkimas artimas natūraliam naudotojo lūkesčių išsiaiškinimui, nes analizė pradedama nuo laukiamų kuriamos IS rezultatų.
4. Kuriant didesnės apimties informacijos sistemas keli projektuotojai ar jų grupės analizuoja atskiras siauresnes veiklos sritis ir atitinkamai joms sudaro po ER schemą. Taip iškyla poreikis korektiškai ir tuo pačiu automatizuotai integruoti keletą ER schemų, siekiant gauti vieną bendrą kuriamos IS duomenų modelį.
5. Automatizuotas ER schemų integravimo įrankis turi būti dialoginė sistema, leidžianti analitikui pačiam šiek tiek pakoreguoti integravimo žingsnių seką ar rankiniu būdu patobulinti gautą schemą. Integruojant turi būti išsprendžiami tarp schemų kylantys konfliktai.
6. Kaip pagrindas sukurtame prototipe naudojama informacijos srautų specifikacija, aprašanti kokio tipo ir kokios struktūros turi būti kuriamos IS rezultatus formuojantys duomenų šaltiniai bei juos siejantys duomenų srautai. Specifikacijos saugykla – MS Access duomenų bazė, pasirinkta dėl nesudėtingo jos naudojimo.
7. Atsižvelgiant į išsiaiškintus integravimo ypatumus, jo metu kylančius konfliktus, sukurtas ir programiškai realizuotas algoritmas. Realizacijai pasirinktas MS Visio programinis paketas, kurį galima lengvai praplėsti naudojant VBA. Tiesa, ne visi aspektai išnagrinėti giliau, įvertinti tik vardų ir iš dalies semantikos konfliktai.
8. Palyginus standartinį MS Visio programinį paketą su prototipo funkcijomis praplėstu įrankiu nustatyta, kad naujas įterptas funkcionalumas nesugadina standartinio funkcionalumo. Turimas patogus įrankis, leidžiantis automatizuotai atlikti dar daugiau projektavimo darbų.

9. Remiantis darbo metu atliktos analizės bei realizuoto prototipo veikimo eksperimentiniais rezultatais, galima konstatuoti, kad efektyvus kokybiško kuriamos IS duomenų modelio sudarymas integruojant atskirų dalių ER schemas įmanomas. Siekiant taikyti šį metodą realiai vykdomuose projektuose, darbas turi būti plėtojamas.
10. Parengtas ir 11-oje tarpuniversitetinėje doktorantų ir magistrantų konferencijoje “Informacinės technologijos 2006” perskaitytas mokslinis straipsnis tema “Automatizuotas informacijos sistemos statinės struktūros sudarymas”. Straipsnis taip pat išspausdintas konferencijos pranešimų medžiagoje.

LITERATŪRA

1. Aleksandravičienė A., Butleris R. *Duomenų modelio sudarymas, integruojant ER schemas*. Informacijos mokslai: mokslo darbai. Vilniaus universitetas. Vilnius: Vilniaus universiteto leidykla, 2005, p. 118-127.
2. Aleksandravičienė A., Butleris R. *Schemų integravimo proceso analizė*. Informacinės technologijos 2006: konferencijos pranešimų medžiaga. Kauno technologijos universitetas. Kaunas: Technologija, 2006, p. 610-618.
3. Batini C., Lenzerini M., Navathe S. B. A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys*, 1986, gruodis, p. 323-397.
4. Boman M., Johannesson P., Bubenko J. jr., Wangler B. *Models, concepts, and information: an introduction to conceptual modelling for information systems development*. Stockholm: Department of Computer and Systems Science, Royal Institute of Technology, 1993, p. 130-144.
5. Bostrøm, Edgar. Experiences from model based Informatics teaching using ER-modeling [interaktyvus]. [žiūrėta 2006 m. balandžio 15 d.]. Prieiga per internetą: <http://www.intermedia.uio.no/cool/WS-03/Bostrom.doc>
6. Brooks F. Three Great Challenges for Half-Century-Old Computer Science. *Journal of the ACM*, 2003, sausis, p. 25-26.
7. Butkienė R. *Informacijos sistemos keliamų funkcinių reikalavimų specifikavimo metodas*. Kauno technologijos universitetas: daktaro disertacijos santrauka. Kaunas: Technologija, 2002, p. 38-44, 82-88.
8. Butleris R., Danikauskas T. *Reikalavimų specifikavimo Oracle CASE terpėje plėtra* [interaktyvus]. February 2004 [žiūrėta 2006 m. balandžio 15 d.]. Prieiga per internetą: <http://www.leidykla.vu.lt/inetleid/inf-mok/19/str6.html>
9. CHEN, Peter; THALHEIM, Bernhard; WONG, Leah. Future Directions of Conceptual Modeling [interaktyvus]. [žiūrėta 2006 m. balandžio 15 d.]. Prieiga per internetą: <http://bit.csc.lsu.edu/~chen/>
10. CHEN, Peter. Past, Present, Future of Data/Information Modeling [interaktyvus]. [žiūrėta 2006 m. balandžio 5 d.]. Prieiga per internetą: <http://www.cis.cornell.edu/iisi/SRDAI-workshop/presentations/chen-p.ppt#16>
11. CHEN, Peter. Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned [interaktyvus]. June 2002 [žiūrėta 2006 m. balandžio 5 d.]. Prieiga per internetą: <http://bit.csc.lsu.edu/~chen/>

12. CHEN, Peter. The Entity-Relationship Model-Toward a Unified View of Data [interaktyvus]. September 1976 [žiūrėta 2006 m. balandžio 5 d.]. Prieiga per internetą: <<http://bit.csc.lsu.edu/~chen/>>
13. DREWRY, Tony. Logical Data Structures. December 2003 [žiūrėta 2006 m. balandžio 20 d.]. Prieiga per internetą: <<http://www.cems.uwe.ac.uk/~tdrewry/lds.htm>>
14. Lawrence R. Schema integration methodologies for multidatabases and the relational integration model – candidacy document. March 22, 1999.
15. McCall J. A. Quality Factors, Encyclopedia of Software Engineering, John Wiley & Sons, Inc., p. 959-969 (1994).
16. Mullins S. C. Getting down to data basics. *The Data Administration Newsletter* [interaktyvus]. 2002, spalio [žiūrėta 2006 m. kovo 17 d.]. Prieiga per internetą: <http://www.tdan.com>.
17. Smith M. A. How to learn data modeling. *The Data Administration Newsletter* [interaktyvus]. 2004, balandis [žiūrėta 2006 m. kovo 17 d.]. Prieiga per internetą: <http://www.tdan.com>.
18. West M., Julian F. Developing high quality data models. *EPISTLE* [interaktyvus]. 1996, kovas [žiūrėta 2006 m. kovo 25 d.]. Prieiga per internetą: <http://www.matthew-west.org.uk/>.
19. Introduction to Data Modeling [interaktyvus]. February 2004 [žiūrėta 2006 m. balandžio 2 d.]. Prieiga per internetą: <http://www.utexas.edu/its/windows/database/datamodeling/dm/overview.html>
20. The Entity Relationship Model [interaktyvus]. [žiūrėta 2006 m. balandžio 13 d.]. Prieiga per internetą: http://www.dcs.bbk.ac.uk/~mark/download/optdb_erd.pdf
21. Microsoft. Visio 2003 apžvalga. [interaktyvus]. [žiūrėta 2006 m. kovo 29 d.]. Prieiga per internetą: <<http://www.microsoft.com/lietuva/office/visio/prodinfo/overview.asp>>
22. Microsoft. [interaktyvus]. Microsoft Office 97/Visual Basic Programmer's Guide. 1997 [žiūrėta 2006 m. kovo 29 d.]. Prieiga per internetą: <<http://www.microsoft.com/officedev/articles/Opg/Intro/Intro.htm>>

TERMINŲ IR SANTRUMPŲ ŽODYNAS

11 lentelė. Darbe vartojamų santrumpų paaiškinimas

Santrumpa	Paiškinimas
DB	Duomenų bazė.
DBVS	Duomenų bazių valdymo sistema.
DS	Duomenų srautas.
DŠ	Duomenų šaltinis.
ER, ER schema	Esybių – ryšių diagrama.
MS	Microsoft
OS	Operacinė sistema
R	Veiklos formuojamas rezultatas.
IS	Informacijos sistema.
VBA	Visual Basic for Application programinis įrankis.

12 lentelė. Darbe vartojamų terminų paaiškinimas

Terminas	Paiškinimas
Atributas	Kiekybinė arba kokybinė tam tikro organizacijos objekto, apie kurį turi būti saugoma informacija, savybė.
Esybė	Realių dalykinės srities objektų aibė arba klasė.
Ribinė reikšmė	Atributų reikšmes ribojantis elementas, kuris yra tam tikra atributo reikšmė.
Ryšys	Dviejų esybių (arba vienos ir tos pačios esybės pačios su savimi) egzempliorių tarpusavio sąryšis (priklausomybė).
Sąlygos elementas	Elementas, parodantis koks šaltinio elementas (atributas ar ribojantis elementas) įtakoja kitą šaltinio elementą (atributą).
Srautas	Informacijos srautas tarp duomenų šaltinio ir rezultato.
Srauto elementas	Elementas, parodantis koks srauto elementas (atributas ar ribojantis elementas) įtakoja kitą srauto elementą (atributą).