

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Vilma Eidukynaitė

**Vartotojo sąsajos modeliavimas duomenų srautų
specifikacijos pagrindu**

Magistro darbas

Darbo vadovas

doc. dr. Rimantas Butleris

Kaunas, 2006

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Vilma Eidukynaitė

Vartotojo sąsajos modeliavimas duomenų srautų specifikacijos pagrindu

Magistro darbas

Kalbos konsultantė

Lietuvių k. katedros lekt.
doc. J. Mikelionienė

2006-05-25

Vadovas

doc. dr. Rimantas Butleris

2006-05-29

Recenzentas

Atliko

IFM 0/2 gr. stud.
Vilma Eidukynaitė

2006-05-26

2006-05-25

Kaunas, 2006

User Interface Design Based on the Data Flow Specification

SUMMARY

The user interface is the direct mediator between the user and the system. It is one of main factors which influences how fluently and with what time resources system could be integrated into business process and how fast systems deployment could be performed. User interface is one of the most important points in software design, because it determines quality and rate of project implementation.

Software design methodologies, based on Unified Modeling Language (UML), Oracle CASE, introduced by C. Finkelstein, D. J. Anderson, V. Balasubramanian, A. Granlund, D. Lafreniere, D. Carr are analyzed in this paper.

The user interface modeling method based on data flow specification is presented in this work; the software prototype of modeling user interface based on this method is implemented.

TURINYS

| | |
|---|----|
| ĮVADAS | 9 |
| 1 VARTOTOJO SĄSAJOS MODELIAVIMAS..... | 11 |
| 1.1 Vartotojo sąsajos projektavimo metodikos..... | 12 |
| 1.1.1 Oracle CASE | 12 |
| 1.1.2 UML grindžiama metodika | 13 |
| 1.1.3 C. Finkelstein metodika..... | 14 |
| 1.1.4 A. Grunlund, D. Lafreniere, D. Carr šablonu paremtas vartotojo sąsajos modeliovimo procesas..... | 15 |
| 1.1.5 V. Balasubramanian vartotojo sąsajos projektavimo metodas pagrįstas nuorodų sistemos funkcionalumu | 16 |
| 1.1.6 D. J. Anderson modelis..... | 18 |
| 1.2 Vartotojo sąsajos modeliovimo technologijos | 18 |
| 1.2.1 IS specifikacijos panaudojimas galutinėje programinės įrangos realizavimo fazėje Oracle Designer/2000 priemone..... | 18 |
| 1.2.2 Magic Draw paketas, palaikantis UML diagramas | 19 |
| 1.2.3 MS Visio 2003 funkcionalumo išplėtimas | 19 |
| 1.3 Analizės rezultatai..... | 20 |
| 2 DUOMENŲ SRAUTŲ SPECIFIKACIJA GRINDŽIAMAS VARTOTOJO SĄSAJOS MODELIS..... | 22 |
| 2.1 Duomenų srautų specifikacija ir specifikavimo proceso eiga | 22 |
| 2.2 Duomenų srautų specifikacijos panaudojimas vartotojo sąsajai modeliuoti | 25 |
| 2.3 Vartotojo sąsajos modelio šablonas..... | 27 |
| 2.4 Konceptualus vartotojo sąsajos sudarymo modelis | 29 |
| 3 VARTOTOJO SĄSAJOS MODELIAVIMO PROTOTIPO REALIZACIJA | 35 |

| | | |
|-------|---|----|
| 3.1 | Panaudojimo atvejai..... | 35 |
| 3.2 | Veiklos kontekstas | 37 |
| 3.3 | Bendra prototipo struktūra | 38 |
| 3.4 | Taisyklės vartotojo sąsajos modelio teisingumo tikrinimui | 39 |
| 3.5 | Vartotojo sąsajos prototipo veikimas..... | 39 |
| 3.5.1 | Duomenų šaltinio pasirinkimas | 39 |
| 3.5.2 | Atributams vartotojo sąsajos elementų parinkimas | 40 |
| 3.5.3 | Apribojimų tikrinimas | 42 |
| 3.5.4 | Vartotojo sąsajos elementų ataskaita..... | 43 |
| 3.6 | Lyginamoji analizė ir praktinė nauda | 44 |
| | IŠVADOS | 46 |
| | TERMINŲ IR SANTRUMPŲ ŽODYNAS | 48 |
| | LITERATŪRA | 49 |
| | PRIEDAI..... | 52 |
| | Testiniai duomenys..... | 52 |
| | Duomenų bazės modelio lentelių aprašymas..... | 54 |
| | Publikuoti straipsniai | 62 |

LENTELIŲ SĄRAŠAS

| | |
|---|----|
| 1 lentelė. Pasiūlyto, Oracle CASE, UML, D. J. Anderson metodų palyginimas | 44 |
| 2 lentelė. Pasiūlyto, šablonu paremto, nuorodų sistemos funkcionalumu pagrįsto metodų palyginimas | 45 |
| 3 lentelė. Metamodelio lentelės „Duom_salt“ testiniai duomenys | 52 |
| 4 lentelė. Metamodelio lentelės „Atributas“ testiniai duomenys | 52 |
| 5 lentelė. Metamodelio lentelės „DS_atributai“ testiniai duomenys | 53 |
| 6 lentelė. Metamodelio lentelės „UIE elementai“ testiniai duomenys | 53 |

PAVEIKSLŲ SĄRAŠAS

| | |
|---|----|
| 1 pav. Vartotojo sąsajos specifikavimo eiga ir rezultatai | 11 |
| 2 pav. UML ir vartotojo sąsajos projektavimas..... | 14 |
| 3 pav. C. Finkelstein metodologija ir vartotojo sąsajos projektavimas | 15 |
| 4 pav. Šablonu paremto metodo struktūra | 16 |
| 5 pav. Vartotojo sąsajos projektavimo užduotys | 17 |
| 6 pav. Anderson modelis vartotojo sąsajai projektuoti..... | 18 |
| 7 pav. Duomenų srautų specifikacijos modeliai | 23 |
| 8 pav. Duomenų srautų specifikacijos panaudojimas IS projektavimo procese..... | 25 |
| 9 pav. Vartotojo sąsajos kūrimo procesas | 26 |
| 10 pav. Konceptualaus vartotojo sąsajos modelio šablonas | 27 |
| 11 pav. Duomenų srautų specifikacijos metamodelis..... | 29 |
| 12 pav. DSS metamodelio lentelės, reikalingos duomenų šaltiniui užpildyti | 30 |
| 13 pav. Vartotojo sąsajos modelio sudarymo procesas | 32 |
| 14 pav. DSS metamodelio lentelės, reikalingos rezultatui užpildyti | 33 |
| 15 pav. Vartotojo sąsajos modeliavimo prototipo panaudojimo atvejai | 35 |
| 16 pav. Konteksto diagrama | 37 |
| 17 pav. CASE įrankio struktūrinė schema..... | 38 |
| 18 pav. Komponentų diagrama..... | 39 |
| 19 pav. Duomenų šaltinio pasirinkimas | 40 |
| 20 pav. Vartotojo sąsajos elementų pasirinkimas..... | 41 |
| 21 pav. Automatiškai sugeneruota vartotojo sąsaja..... | 41 |
| 22 pav. Kontekstinis meniu vartotojo sąsajos elemento tipui keisti..... | 41 |
| 23 pav. Vartotojo sąsajos elemento keitimas..... | 42 |
| 24 pav. Eiliškumo nustatymas objektams..... | 42 |

| | |
|--|----|
| 25 pav. MS Visio meniu išplėtimas..... | 43 |
| 26 pav. Vartotojo sąsajos elementų ataskaita | 44 |

ĮVADAS

Vartotojo sąsaja yra tiesioginis tarpininkas tarp sistemos ir vartotojo. Ji yra vienas iš faktorių, lemiančių kaip sklandžiai ir su kokiomis laiko, lėšų sąnaudomis sistema įsilies į verslo procesą, kaip greitai bus atliktas sistemos diegimas. Vartotojo sąsajos prototipų sudarymas yra naudojamas jau ankstyvosiose programinės įrangos projektavimo fazėse reikalavimų analizei, specifikacijų validavimui ir vartotojų programinės įrangos sistemų pripažinimui didinti. Programinės įrangos prototipų sudarymas yra susijęs ir su sparčiu vartotojo sąsajos kūrimu, ir su planuojamų sistemų funkcionalumu. Tam, kad būtų naudingas, prototipas turi būti sukurtas greitai ir sumodeliuotas tokiais priemonėmis, kad galėtų būti greitai modifikuojamas. Vartotojo sąsajos modelis turėtų būti atskirtas nuo programos realizacijos [29].

Vartotojo sąsajos ir jos modeliavimo svarbą patvirtina ir skirtingų autorių pateikiamų metodikų ir sprendimų vartotojo sąsajai modeliuoti įvairovė [22]. *Oracle CASE* metodas ir jo pagrindu sukurta CASE priemonė turi išplėtotas vartotojo sąsajos reikalavimų specifikavimo galimybes, leidžia išskirti veiklos funkcijas, jas detalizuoti funkcijų hierarchijos diagramomis (angl. *Function hierarchy diagram*), išskirti reikšmingas organizacijos esybes, ryšius tarp jų, bei jų atributus esybių-ryšių diagrama ER (angl. *Entity Relationship diagram*). ER ir funkcijų hierarchijos diagramos reikalingos vartotojo sąsajai specifiuoti. Vartotojo sąsajai modeliuoti gali būti taikoma UML (angl. *Unified Modeling Language*) kalba [27, 28]. UML apibrėžia 12 diagramų tipų, užtikrinančių vartotojo sąsajos modeliavimo proceso eigos specifikavimą. Šiame formalių uždavinio procesus aprašančių diagramų rinkinyje išvelgiamas ėjimas nuo uždavinio koncepcijos iki duomenų bazės schemas ir uždavinio sprendimo algoritmo formalaus aprašymo [18].

Darbe analizuojamos C. Finkeltein metodika, autorių A. Grunlund, D. Lafreniere, D. Carr pateiktas šablonu paremtas vartotojo sąsajos modeliavimo procesas, V. Balasubramanian vartotojo sąsajos projektavimo metodas pagrįstas nuorodų sistemos funkcionalumu, D. J. Anderson modelis.

Galima išskirti keletą esminių problemų srityje:

- Funkcinių bei nefuncinių vartotojo reikalavimų specifikavimo ir atvaizdavimo išsamus užtikrinimas.

- Vartotojo sąsajos ir veiklos logikos integracija. Specifikuojant reikalavimus informacijos sistemai, veiklos logika būna pilnai specifikuota, tačiau kuriant vartotojo sąsają sunku atskirti, kokia dalis veiklos logikos yra neatsiejama nuo vartotojo sąsajos ir kuri gali būti palikta, kaip vidinių IS procesų dalis.
- Palaikomumas, t. y., kad sugeneruoti rezultatai galėtų būti lengvai redaguojami ir atliktos korekcijos lengvai išsaugomos specifikacijoje arba būtų galimybė iš sugeneruotų rezultatų atlikti atgalinę inžineriją [23].

Darbe siekiama:

- Atlikti vartotojo sąsajos modeliavimo metodų analizę.
- Išskirti kriterijus, kuriuos turėtų tenkinti visavertis vartotojo sąsajos modeliavimo metodas.
- Pateikti vartotojo sąsajos modeliavimo duomenų srautų specifikacijos pagrindu metodą ir pagal jį realizuotą vartotojo sąsajos modeliavimo prototipą;
- Vartotojo sąsajos modeliavimo metodą palyginti su kitais, išskiriant metodo privalumus ir trūkumus bei pateikti metodo įvertinimą.

Pirmame skyriuje siekiama apžvelgti vartotojo sąsajos projektavimo metodus ir metodologijas, pateikti jų privalumus ir trūkumus bei išskirti, kokie kuriamos sistemos aspektai turi būti įvertinti modeliuojant vartotojo sąsają.

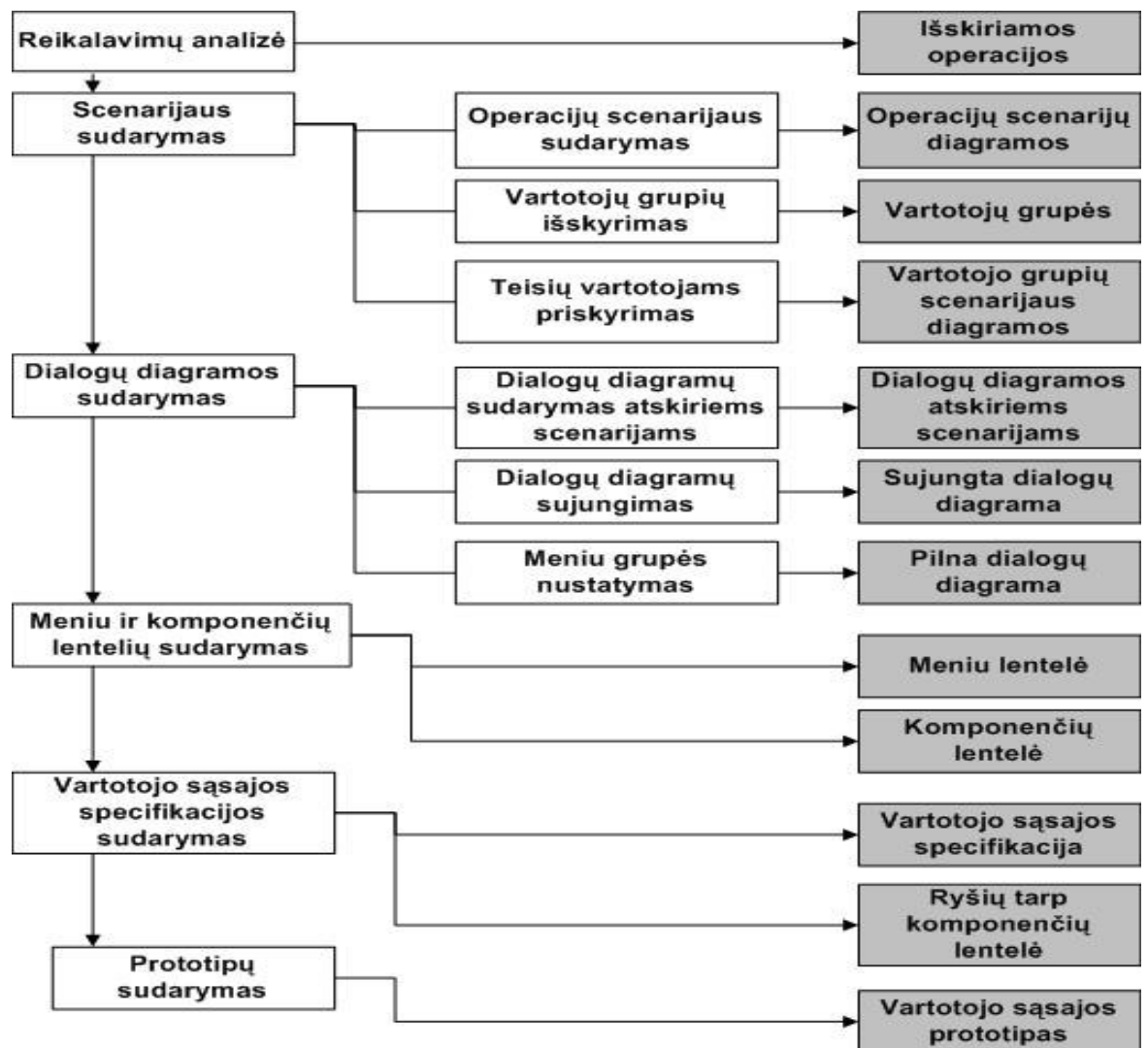
Antrame skyriuje pateikiama duomenų srautų specifikacijos sudėtis ir specifikavimo proceso eiga, atkreipiant dėmesį į vartotojo sąsajos modeliavimo metodo vietą specifikacijoje, naudojamus duomenų srautus, vartotojo sąsajos modeliavimo duomenų srautų specifikacijos pagrindu modelis, konceptualus modelio sudarymo procesas.

Trečiame skyriuje aprašoma vartotojo sąsajos modeliavimo prototipo realizacija, panaudojimo atvejai, apribojimai. Pademonstruojamas prototipo veikimas. Pateikiami lyginamosios analizės rezultatai.

1 VARTOTOJO ŠAŠAJOS MODELIAVIMAS

Prototipų sudarymas yra naudojamas ankstyvosiose programonės įrangos projektavimo fazėse reikalavimų analizei, specifikacijų validavimui ir vartotojų programinės įrangos sistemų pripažinimui didinti. Programinės įrangos prototipų sudarymas yra susijęs ir su sparčiu vartotojo sąsajos kūrimu, ir su planuojamų sistemų funkcionalumu [19]. Tam, kad būtų naudingas, prototipas turi būti sukurtas greitai ir sumodeliuotas tokiomis priemonėmis, kad galėtų būti greitai modifikuojamas.

Vartotojo sąsajos prototipo kūrimas apima reikalavimų analizės, scenarijaus sudarymo, dialogų diagramos sudarymo, meniu ir komponentių lentelių sudarymo, vartotojo specifikacijos sudarymo etapus [25]. Vartotojo sąsajos specifikavimo eiga ir rezultatai pateikiami 1 paveiksle.



1 pav. Vartotojo sąsajos specifikavimo eiga ir rezultatai

1.1 Vartotojo sąsajos projektavimo metodikos

Svarbu panagrinėti vartotojo sąsajos projektavimo metodikas tam, kad išskirti pagrindinius kriterijus, į kuriuos svarbu atsižvelgti kuriant vartotojo sąsają. Bus analizuojamos šios metodikos: Oracle CASE, UML paremta, autorių A. Grunlund, D. Lafreniere, D. Carr pateikta šablonu paremta vartotojo sąsajos modeliavimo metodika, D. J. Anderson modelis, V. Balasubramanian metodas pagrįstas nuorodų sistemos funkcionalumu.

1.1.1 Oracle CASE

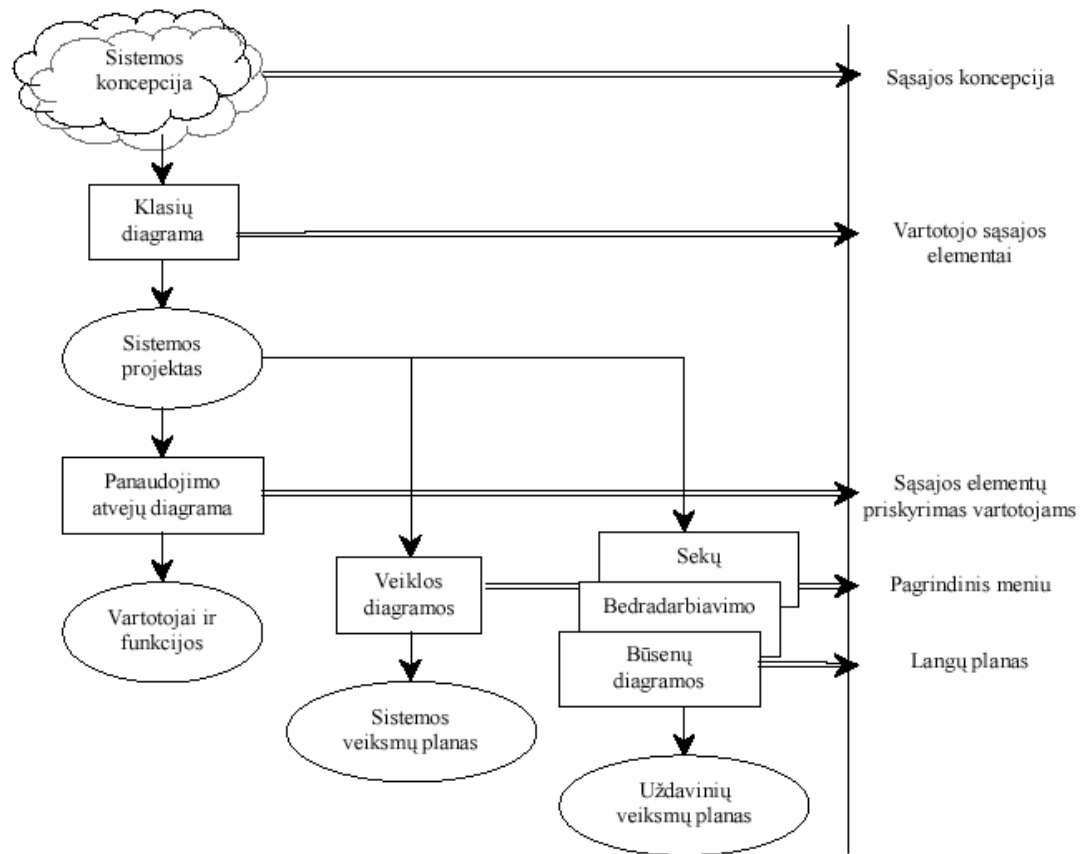
Informacijos sistemų kompiuterizuotas projektavimas – vienas iš vartotojo reikalavimų analizės ir specifikavimo etapo kokybės užtikrinimo metodų. CASE (CASE – *computer-aided software engineering*) priemonės leidžia atsiriboti nuo painaus programinio kodo, pirmuosius sistemos kūrimo darbus atliekant konceptualių lygmeniu, kuriame sistemos architektūros specifikacija ir projektavimo rezultatai yra daug vaizdingesni, lengviau suprantami ir modifikuojami. Tai iš dalies priartina IS kūrimą prie vartotojo lygio. CASE priemonės naudojamos visame sistemos kūrimo cikle. Šiomis priemonėmis organizuojamas ir valdomas programinės įrangos kūrimas, tai ypač naudinga dideliems projektams, kadangi padeda susisteminti IS kūrimą ir griežčiau jį kontroliuoti. CASE priemonių pagrindas yra įvairūs programinės įrangos kūrimo metodai: kai kurių priemonių pagrindas – struktūrinis metodas, tačiau pastaruoju metu orientuojamasi į objektines programinės įrangos kūrimo metodologijas.

Oracle korporacija bene labiausiai išstobulino IS kūrimo priemones. Oracle CASE metodo pagrindu įdiegta Oracle Designer priemonė kartu su kitomis Oracle priemonėmis suteikia galimybę kompiuterizuotai atlikti daugelį IS kūrimo darbų. Oracle siūloma technologija yra viena iš tobuliausių; ji užtikrina integruotą sistemos projektavimą bei programinės įrangos kūrimą [5, 6].

Oracle CASE metodas ir jo pagrindu sukurta CASE priemonė turi išplėtotas vartotojo sąsajos reikalavimų specifikavimo galimybes, leidžiančias pateikti vartotojui specifikuotus reikalavimus tokiu pavidalu: formų, ataskaitų, meniu. išskirti veiklos funkcijas, jas detalizuoti funkcijų hierarchijos diagramomis (angl. *Function hierarchy diagram*), išskirti reikšmingas organizacijos esybes, ryšius tarp jų, bei jų atributus esybių-ryšių diagrama ER (angl. *Entity Relationship diagram*). Vartotojo sąsajai specifikuoti reikia šių diagramų: esybių ryšių (ER) ir funkcijų hierarchijos [12].

1.1.2 UML grindžiama metodika

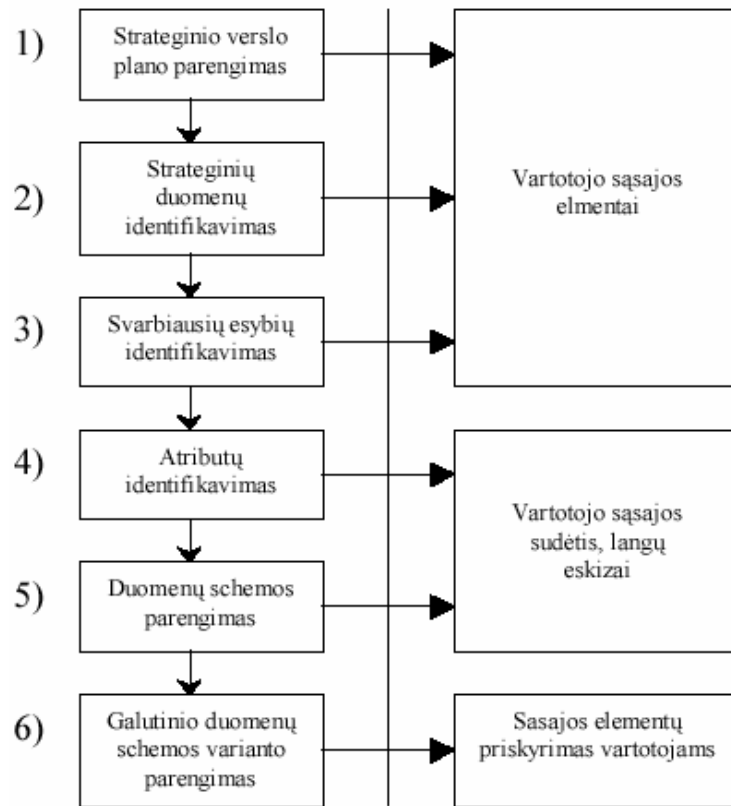
UML (*Unified Modeling Language*) yra programų sistemų projektavimo rezultatų vaizdavimo, specifikavimo, konstravimo ir dokumentavimo kalba [21]. UML apibrėžia 12 diagramų tipų, suskirstytų į tris kategorijas [26]: keturi diagramų tipai (klasių, objektų, komponentų, įdiegimo diagramos) vaizduoja statinę struktūrą, penki (panaudojimo atvejų, sekų, veiklos, bendradarbiavimo, būsenų diagramos) – skirtingus dinaminės elgsenos aspektus, trys (paketai, posistemės, moduliai) – modulių organizavimo ir valdymo būdus. Šiame formalių uždavinio procesus aprašančių diagramų rinkinyje išvelgiamas ėjimas nuo uždavinio koncepcijos iki duomenų bazės schemas ir uždavinio sprendimo algoritmo formalaus aprašymo [18]. 2 paveiksle parodytas galimas šių diagramų sujungimas projektuojant ir išskirti sąsajos projektavimo žingsniai, kurie turėtų būti atlikti bendroje sistemos projektavimo veiksmų sekoje [20]. Klasių diagrama yra pradinė ir pagrindinė, nes jai sudaryti reikia žinoti uždavinio formuluotę ir esminius struktūrinius elementus. Ši diagrama leidžia aiškiai suprasti, ko yra siekiama uždaviniu ir konceptualiai parodyti ryšiai tarp uždavinio struktūrinių elementų (klasių). Klasių diagramą galima laikyti apibendrintu sistemos projektu, todėl ją kuriant paaiškėja atskiri vartotojo sąsajos elementai. Sudarius klasių diagramą, galima vystyti sistemos aprašymą panaudojimo atvejų (angl. *use case*) diagramą, kurioje sistema yra paskirstoma „menamiems“ vartotojams įsivaizduojant jų funkcijas sistemoje. Taigi tokia diagrama leidžia įsivaizduoti atskirų vartotojų sąsajas, bet tik apibendrintame lygyje. Sistemos projektas, aprašytas klasių diagramose, leidžia detalizuoti sprendžiamais uždaviniais (veiklos diagrama) ir atlikti šių uždavinių detalizaciją (būsenų, sekų, bendradarbiavimo diagramos). Šios keturios diagramos vadinamos elgsenos diagramomis. Pirmoji leidžia sudaryti pagrindinį sąsajos meniu, o kitos trys – vartotojo sąsajos langų planą.



2 pav. UML ir vartotojo sąsajos projektavimas

1.1.3 C. Finkelstein metodika

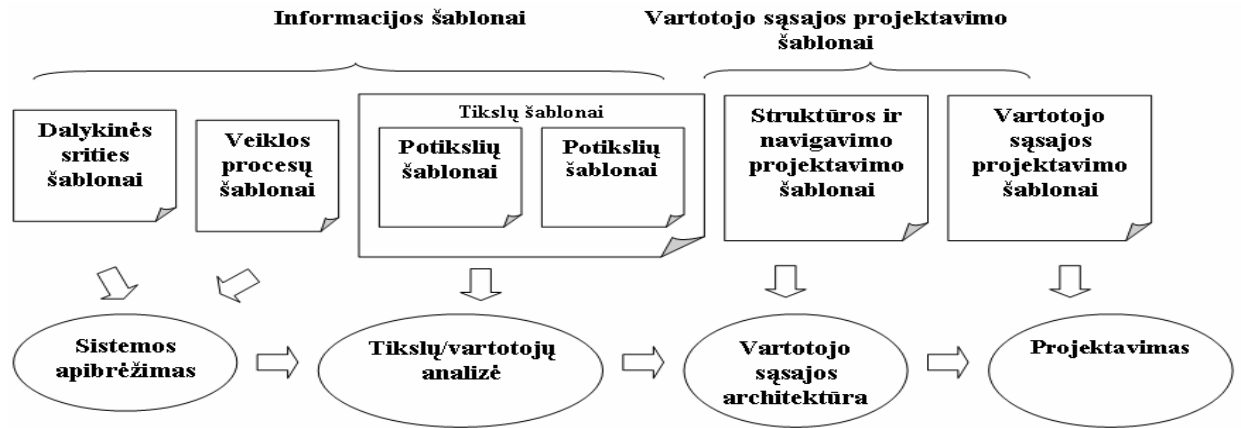
Galima analogiška C. Finkelstein pasiūlytos projektavimo metodikos analizė. Informacijos inžinerija apima ne visos sistemos, o tik jos informacinės dalies projektavimą. Čia neatsispindi tai, kas aprašoma elgsenos diagramose ir praktiškai apsiribojama tuo, kas yra klasių diagramoje, tačiau čia daug detaliau aprašoma informacinė pusė: strateginiai duomenys, esybės, atributai ir jų ryšiai. Integruojant šią metodiką su vartotojo sąsajos projektavimu (žr. 3 pav.), praktiškai galima veiksmų sekoje išskirti žingsnius nuo sąsajos koncepcijos iki informacinių langų eskizų [20].



3 pav. C. Finkelstein metodologija ir vartotojo sąsajos projektavimas

1.1.4 A. Grunlund, D. Lafreniere, D. Carr šablonu paremtas vartotojo sąsajos modeliavimo procesas

A. Grunlund, D. Lafreniere, D. Carr pateikė šablonu paremtą vartotojo sąsajos modeliavimo procesą (angl. *A Pattern-Supported Approach*) [17]. Šablonas yra formalizuotas įrodytos koncepcijos, kuri išreiškia netrivialius vartotojo sąsajos modeliavimo problemos sprendimus, aprašymas. Pirminis šablono tikslas yra apžvelgti sprendimus, kurie padeda vartotojo sąsajos projektuotojams išspręsti dažnai pasitaikančias ir sudėtingas vartotojo sąsajos vystymo problemas. Šablonu paremto metodo struktūra pavaizduota 4 paveiksle.



4 pav. Šablonu paremta metodo struktūra

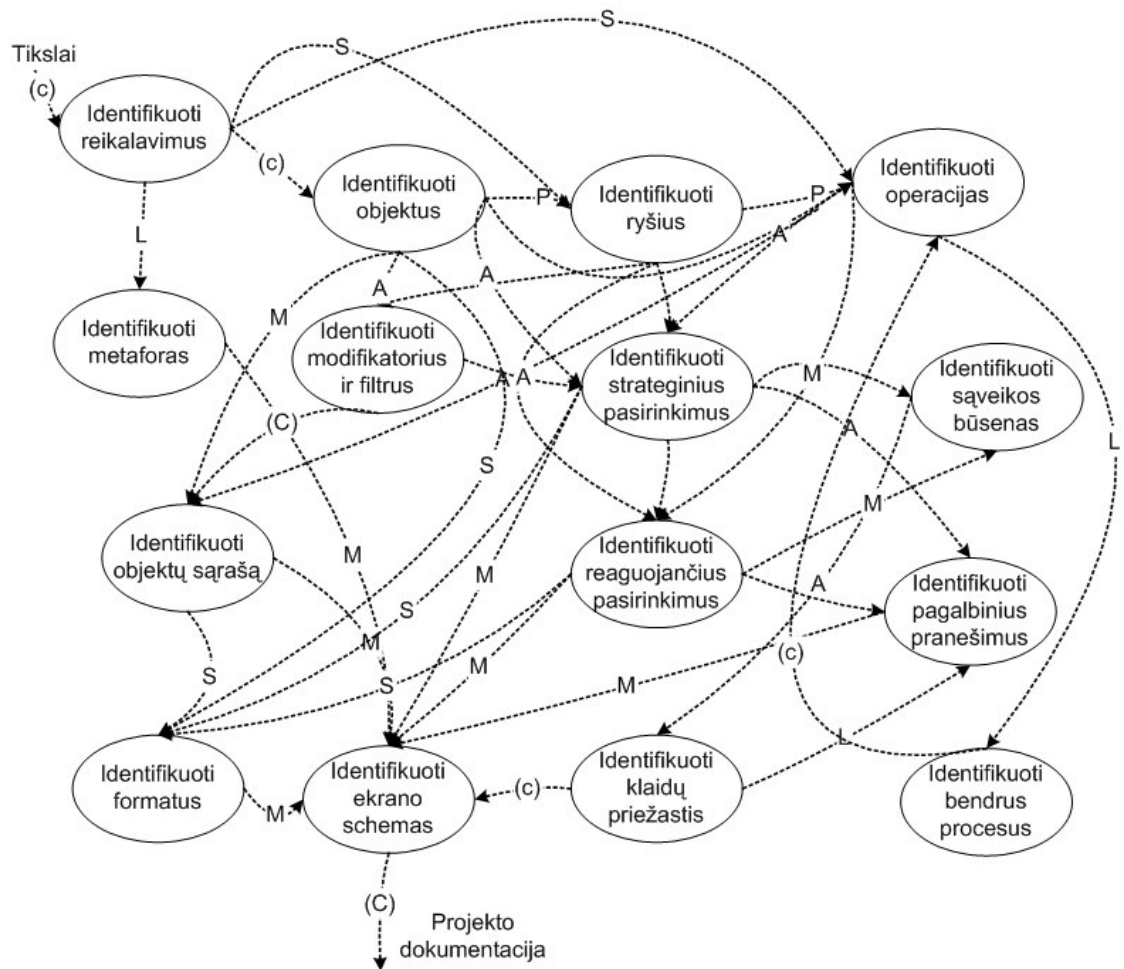
1.1.5 V. Balasubramanian vartotojo sąsajos projektavimo metodas pagrįstas nuorodų sistemos funkcionalumu

Balasubramanian pateikė vartotojo sąsajos projektavimo metodą, pagrįstą nuorodų sistemos funkcionalumu [4]. Vartotojo sąsajos projektavimo procesas nagrinėjamas kaip projektavimo užduočių aibė, transformuojanti vartotojo reikalavimus į vartotojo sąsajos projektavimo specifikaciją. Vartotojo sąsajos projektavimas gali būti padalintas į tris pagrindines fazes: planavimą ir idėjų generavimą, prototipų kūrimą ir projektavimą, ir kokybės analizę. Kadangi vartotojo sąsajos projektavimas kūrybiškas procesas, jis negali būti nuoseklus. Taigi šio metodo pagrindas – nuorodų sistemos tinklas, sudarytas iš penkiolikos nenuoseklių, stipriai tarpusavyje susietų projektavimo užduočių, kurios gali būti traktuojamos kaip mazgai. Tai:

- Identifikuoti aplikacijos reikalavimus.
- Identifikuoti metaforą ar metaforas.
- Identifikuoti objektus, kurie sudaro aplikaciją.
- Identifikuoti šalutinius klasifikacijas ar ryšius tarp objektų.
- Identifikuoti veiksmus ar funkcijas, kurias gali atlikti objektai. Tai gali būti bendri veiksmai, aiškiai nusakyti veiksmai arba valdymo funkcijos.
- Identifikuoti modifikatorius ar filtrus, kurie pažymi objektų posistemas.
- Identifikuoti strateginius pasirinkimus, kurie leidžia vartotojui pritaikyti sąveiką specifinių tikslų atlikimui.
- Identifikuoti objektų formatus, objektų dalis, meniu ir k. t.
- Identifikuoti objektų, kurie yra semantiškai susiję arba paremti specifiniais pasirinkimų kriterijais, sąrašą.

- Identifikuoti reaguojančius pasirinkimus, kurie gali būti atlikti sąrašo ar objektų rinkinio. Tai pasirinkimai, kurie numato pilną specifinių tikslų vykdymą.
- Identifikuoti besidalinančius procesus ar funkcijas.
- Identifikuoti vartotojo sąveikos būsenas.
- Identifikuoti būtiną pagalbą per visą sistemą.
- Identifikuoti galimus pranešimus (atgal vartotojui) ir klaidų sąlygas.
- Identifikuoti ekrano schemą – darbo plotą, valdymo sritį, būsenų sritį, pranešimų sritį.

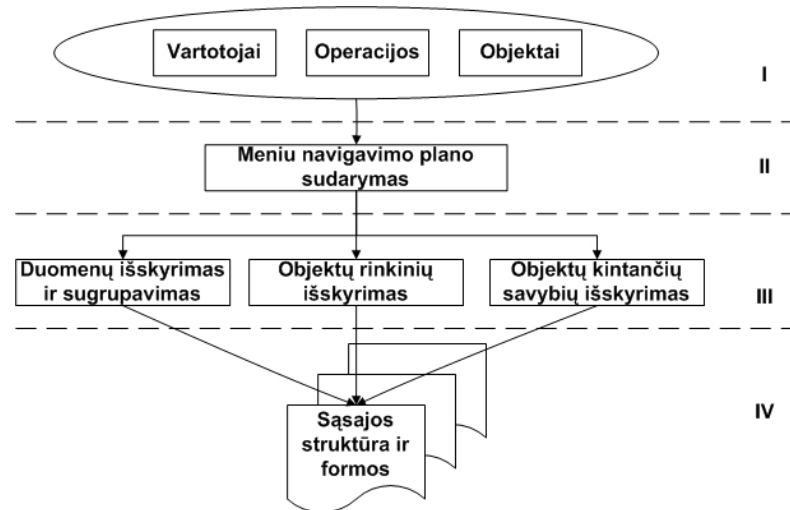
Proceso mazgų, jų įėjimų aibių ir transformuojamų išėjimų aibių kombinacija sudaro nuorodų sistemos tinklą. Jis įgalina sąveiką ir grįžimą atgal prie įvairių užduočių. Vartotojo sąsajos projektavimo užduotys pavaizduotos 5 paveiksle.



5 pav. Vartotojo sąsajos projektavimo užduotys

1.1.6 D. J. Anderson modelis

Bene plačiausiai naudojamas D. J. Anderson [3] modelis pateikiamas 6 paveiksle.



6 pav. Anderson modelis vartotojo sąsajai projektuoti

Tai keturių etapų procesas, kur pirmajame etape suformuojama sąsajos koncepcija ir išryškinami vartotojo sąsajos elementai, antrajame suformuojamas navigacijos planas, trečiajame etape atliekama sąsajos elementų detalizacija, ketvirtajame suformuojama galutinė vartotojo sąsajos versija [1, 2]. Šios projektavimo fazės pateikiamos atsietai nuo sistemos projektavimo, todėl jos prašosi integruojamos į praeitame skyrelyje nagrinėtąsias metodikas. Gretinant jas, matyti:

- 1) Sąsajos koncepcijos sukūrimas ir elementų išryškinimas integruotūsi į pradinis sistemos projektavimo žingsnius.
- 2) Navigacija ir detalizacija, t. y. pagrindinis meniu, langai ir jų planas, integruotūsi į elgsenos diagramų sudarymo etapą.
- 3) Vartotojo sąsajos struktūra ir forma sistemos projektavimo metodikose neatsispindi [20].

1.2 Vartotojo sąsajos modeliavimo technologijos

1.2.1 IS specifikacijos panaudojimas galutinėje programinės įrangos realizavimo fazėje Oracle Designer/2000 priemone

Šiuo metu labiausiai paplitusi CASE sistema – Oracle Designer/2000. Vienas iš jos privalumų – galimybė betarpiškai perduoti sistemos objektų saugykloje esančią IS specifikaciją į

galutinę programinės realizacijos fazę, naudojant kitas Oracle sistemos priemones. Tuo būdu užtikrinama nepertraukiama IS kūrimo technologija su viena ir ta pačia sistema ir padidinamas kuriamosios sistemos specifikacijos panaudojimo laipsnis programiškai realizuojant siūlomus projektinius sprendimus [7].

Oracle Designer formų generatoriumi galima sukurti funkciškai turtingą grafinę vartotojo sąsajos programą. Formų generatorius įgalina generuoti Form Builder 5.0 programų modulius (formas ir meniu) iš veiklos taisyklių ir numatytų reikalavimų įrašytų Oracle Designer saugykloje.

Turint apibrėžtus modulius modulių saugykloje ir išskirtą modulių duomenų vartoseną ir reikalavimų schemas, galima toliau kontroliuoti išvaizdą ir aplikacijų egliseną pagal apibūdintus objektus objektų bibliotekoje, iš kurių formų generatorius galima suskirstyti į posistemes arba kopijuoti objektus į generuojamas formas.

1.2.2 Magic Draw paketas, palaikantis UML diagramas

MagicDrawUML yra CASE priemonė, skirta objektinei analizei, projektavimui ir kodo generavimui. Sistema remiasi keliomis technologijomis – Booch , OMT, UML ir kt. Visos jos naudojamos bendram tikslui – parodo grafiškai programinės įrangos struktūrą keliais pjūviais; Magic Draw palaiko UML kalbą, o UML kalba remiasi diagramomis. Magic Draw palaiko visas UML diagramas:

- panaudojimo atvejų – sistemos funkcijos vartotojo požiūriu;
- klasių – statinė sistemos struktūra atvaizduota klasėmis ir jų ryšiais;
- būsenų – klasių būsenų diagrama;
- veiksnių – aibė veiksnių, priklausančių tam tikrai operacijai;
- sekų – atvaizduoja objektų sąveiką;
- bendradabavimo – objektų sąveika;
- realizacijos – fiziniai komponentai (failai, aparatūra ir pan.).

1.2.3 MS Visio 2003 funkcionalumo išplėtimas

Microsoft Visio paketas turi automatizavimo įrankį. Automatizavimo įrankis – tai mechanizmas, įgalinantis išplėsti Microsoft Visio funkcionalumą arba įtraukti Visio mechanizmą kaip grafinį mechanizmą individualioms programoms. Automatizavimas gali būti panaudotas piešiniams sukurti ar duomenims iš vidinių šaltinių atnaujinti, perskaityti piešinius ir išgauti informaciją iš jų, arba išplėsti piešimo elgseną *Microsoft Visual Basic for Application (VBA)*

aplikacijoms kurti. VBA projektas susideda iš modulių, klasių modulių ir vartotojo formų. Modulis – tai apibrėžimų rinkinys, sekantis iš programos vykdomų – infrastruktūrų sąrašo. Klasių modulis apibrėžia objektus, jų savybes ir metodus. Klasės modulio dokumentas kaip šablonas, iš kurio objekto pavyzdys sukuriamas vykdymo metu. Kiekvienas Visio paremtas VBS projektas susideda iš klasių modulio, kuris pateikia MS Visio dokumento savybes, metodus, įvykius. Vartotojo formos susideda iš tokių vartotojo sąsajos valdymo, komandinių mygtukų ir teksto laukų. Paprastas projektas gali reikalauti atskirų vartotojo formų ar modulių, sudėtingi projektai gali sudaryti daugialypius modulius, klasių modulius ir vartotojo formas. Visio/2003 failai ir šablonai gali būti naudojami specifikacijos grafinei notacijai saugoti [9].

1.3 Analizės rezultatai

Susipažinus su žinomais metodais ir metodologijomis vartotojo sąsajai modeliuoti, galima išskirti reikalavimus, kurių laikomasi, kuriant vartotojo sąsajos metodikas. Juos galima sugrupuoti į penkias klases. Tai:

- 1) Informacijos poreikių reikalavimai: kokios informacijos vartotojui reikės eksploatuojant sistemą;
- 2) Funkcionalumo realizavimo reikalavimai: uždavinio sprendimo algoritmo realizacija;
- 3) Duomenų registracijos reikalavimai: kaip bus užpildoma duomenų bazė;
- 4) Racionaliausios vartotojo sąsajos alternatyvos reikalavimai: sukurti pirmuosius tris punktus tenkinančią racionaliausią vartotojo sąsajos alternatyvą;
- 5) Vartotojo nefunkciniai reikalavimai.

Po atliktos analizės galima išskirti tokius atskirų metodikų privalumus ir trūkumus:

- 1) Oracle CASE priemonės neužtikrina išsamios reikalavimų specifikacijos sudarymo poreikių surinkimo etape, tačiau Oracle CASE pagrindu įdiegta Oracle Designer priemonė kartu su kitomis priemonės teikia galimybę specifikuoti vartotojo sąsajos reikalavimus.
- 2) Vartotojo reikalavimų specifikavimas naudojant *UML* teikia daugiau galimybių nei *Oracle CASE* metodas, kuris neužtikrina išsamios reikalavimų specifikacijos sudarymo. Oracle CASE metodu tiesiogiai neskiriama dėmesio poreikių surinkimo etapui.

- 3) Tiek nuorodų sistema paremtas vartotojo sąsajos projektavimo metodas, tiek ir vartotojo sąsajos modeliavimo procesas remiantis šablonais neturi išvystytų priemonių šiai metodikai įgyvendinti.
- 4) Naudojant *UML* vartotojo funkcinius reikalavimus tiesiogiai tenkinantys rezultatai – formos, ataskaitos, meniu, kurie paprastai išreiškiami per sąsajos elementus, – nėra visiškai specifikuojami, o specifikacijos forma neadekvati vartotojo susikurtam jų įvaizdžiui.

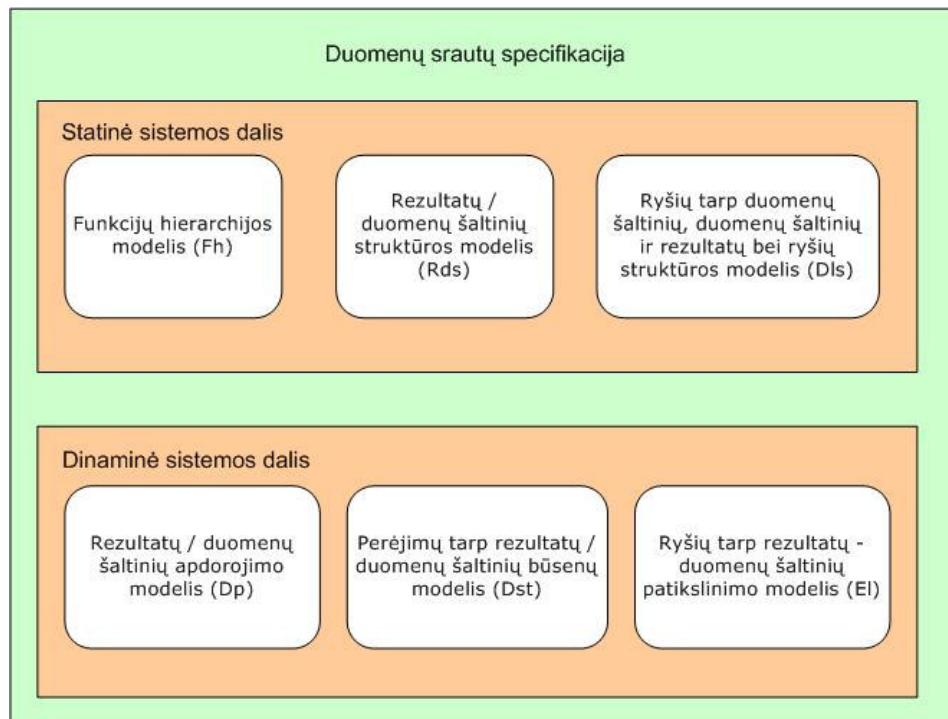
2 DUOMENŲ SRAUTŲ SPECIFIKACIJA GRINDŽIAMAS VARTOTOJO SĄSAJOS MODELIS

Įeinantys ir išeinantys informaciniai duomenų srautai egzistuoja kiekvienoje organizacijoje. Įeinančių srautų apdorojimas sukuria išeinančius srautus. Daugelis iš organizacijoje esančių srautų turi dokumentinę ar kitokią standartizuotą formą, kuri paprastai naudojama organizacijoje. Informacijos sistemos tikslas yra informacinių srautų apdorojimas ir efektyvus tų srautų valdymas. Duomenų srautų specifikacija yra funkcinių reikalavimų specifikavimo metodo (FRSM) rezultatas [8]. FRSM pagrindinė idėja – organizacijos duomenų srautų analizė. Duomenų srautų specifikacijos pagrindu modeliuojama vartotojo sąsaja, todėl svarbu apžvelgti duomenų srautų specifikacijos sudėtį ir specifikavimo eigą.

2.1 Duomenų srautų specifikacija ir specifikavimo proceso eiga

Duomenų srautų specifikacija (DSS) yra reikalavimų informacijos sistemai specifikacija, kuri gali būti aprašyta kaip sistema $\langle ISS \rangle ::= (\langle Fh \rangle, \langle Rds \rangle, \langle Dls \rangle, \langle Dp \rangle, \langle Dst \rangle, \langle El \rangle)$ susidedanti iš žemiau išvardintų modelių (žr. 7 pav.):

- Fh – Funkcijų hierarchijos modelis;
- Rds – Rezultatų / duomenų šaltinių struktūros modelis;
- Dls – Ryšių tarp duomenų šaltinių, duomenų šaltinių ir rezultatų bei ryšių struktūros modelis;
- Dp – Rezultatų / duomenų šaltinių apdorojimo modelis;
- Dst – Perėjimų tarp rezultatų / duomenų šaltinių būsenų modelis;
- El – Ryšių tarp rezultatų – duomenų šaltinių patikslinimo modelis.



7 pav. Duomenų srautų specifikacijos modeliai

DSS modeliai aprašo ir statinę, ir dinaminę kuriamos sistemos dalį (žr. 7 pav.). Statinė sistemos dalis aprašoma: Fh, Rds, Dls modeliais, o sistemos elgsena (dinamika): Dp, Dst, El modeliais. Visi šie modeliai turi grafinę notaciją, tačiau, siekiant specifikavimo ir sistemos projektavimo procesą automatizuoti, visa informacija saugoma specifikacijos saugykloje. 11 paveikslėlyje pateikiamas DSS saugyklos metamodelis.

Pagrindinis FRSM darbo principas yra dalykinės srities analizė pradedant nuo rezultatų ir tęsiant analizę link duomenų šaltinių. Šis principas atitinka natūralią kompiuterizuojamos veiklos analizės eigą.

Pirmiausiai apibrėžiamas dalykinės srities kontekstas, kuris bus nagrinėjamas kuriant IS. Konteksto apibrėžimui naudojamas modifikuotas funkcijų hierarchijos modelis. Sudarius šį modelį suformuojamas sąrašas veiklos konteksto rezultatų, kuriuos turės pateikti sukurta IS. Sudaromas ryšių tarp duomenų šaltinių, duomenų šaltinių ir rezultatų, bei ryšių struktūros modelis. Modelio sudarymo metu nustatomi duomenų šaltiniai, kurie reikalingi atrinktiems pirmojo etapo metu rezultatams suformuoti. Modelis parodo ryšio informacinio srauto kryptį, bei srauto struktūrą, kuri specifikuojama, kai sudaromi duomenų šaltinių bei rezultatų struktūros modeliai.

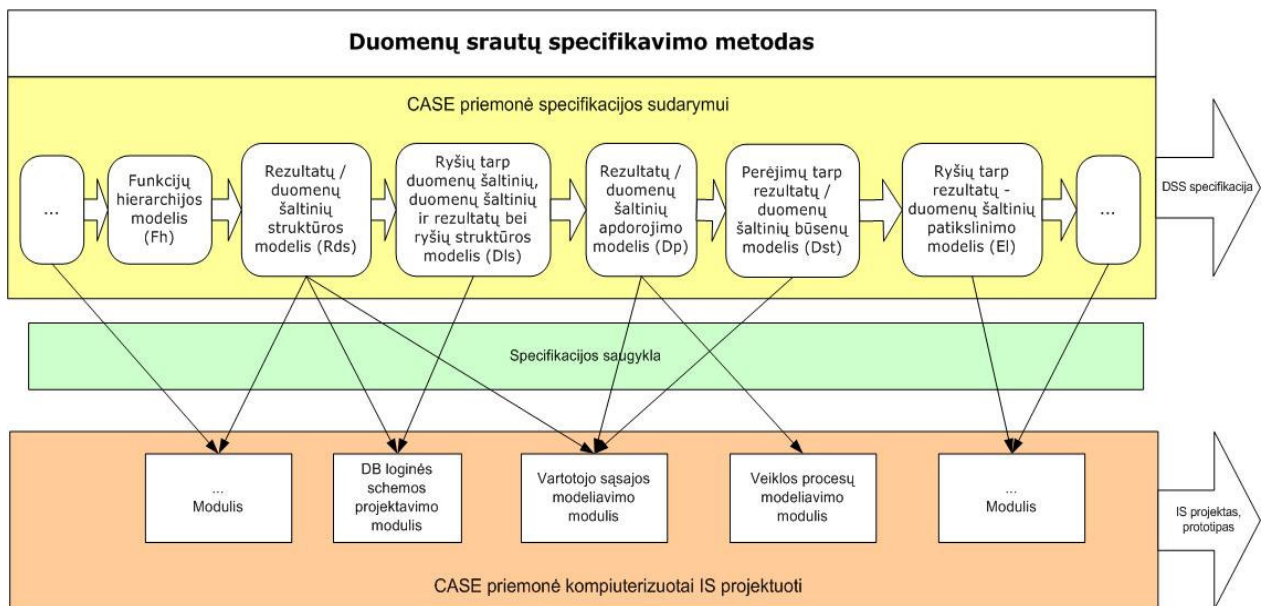
Struktūros modelio sudarymui naudojamas panašus principas, kaip ir formų analizės pagrindu sudarant duomenų modelį. Struktūros modelis yra sudarytas iš dviejų sluoksnių: į vartotoją orientuotas sluoksnis ir į projektuotoją orientuotas sluoksnis. Šių skirtingų sluoksnių sudarymo tikslas yra atotrūkio tarp vartotojo ir IS projektuotojo sumažinimas [15]. Iš projektuotojo sluoksnio sudaromas lokalus, ribojamas vieno rezultato arba duomenų šaltinio konteksto ER modelis.

Sekančio etapo metu detalizuojamas DIs modelis, specifikuojant ryšių struktūrą. Tam panaudojami sudaryti struktūros modeliai bei sugeneruotos lokaliai ER schemas. Detalizuota ryšio struktūra parodo priklausomybes tarp rezultatų ir duomenų šaltinių. Kiekvienam rezultatui ir duomenų šaltiniui taip pat sudaromas apdorojimo modelis. Dp modelis specifikuoja veiksmus, kurie turi būti atlikti apdorojant duomenų šaltinį ar generuojant rezultatą, bei tų veiksmų seką. Dp modelyje taip pat specifikuojami ir aktoriai, kurie dalyvauja apdorojimo procese.

Sekančio žingsnio metu turi būti sudarytas perėjimų tarp rezultatų / duomenų šaltinių būsenų modelis. Dst modelis specifikuoja visas galimas būsenas, į kurias gali pakliūti rezultatas arba duomenų šaltinis apdorojimo arba generavimo proceso metu. Šiam modeliui buvo pasirinkti veiklos (angl. *enterprise*) modeliavimo principai, leidžiantys integruoti statines ir dinamines priklausomybes.

Galiausiai Dst ir DIs modelių pagrindu atliekamas ryšių tarp duomenų šaltinių, duomenų šaltinių ir rezultatų patikslinimas. Ryšių patikslinimas yra būtinas, nes ryšiai, kurie nustatyti DIs modelyje, yra silpni, kadangi jie nėra visada egzistuojantys. Taip yra todėl, kad duomenų šaltiniai arba rezultatai gali būti susiejami tik būdami tam tikroje būsenoje. Šiuo modeliu baigiamas funkcinių reikalavimų informacijos sistemai specifikacijos sudarymas.

Specifikacijos sudarymo procesas ir kompiuterizuotos IS projektavimo procesas gali būti realizuotas kaip atskiri CASE įrankio komponentai, dirbantys su ta pačia saugykla (žr. 8 pav.)

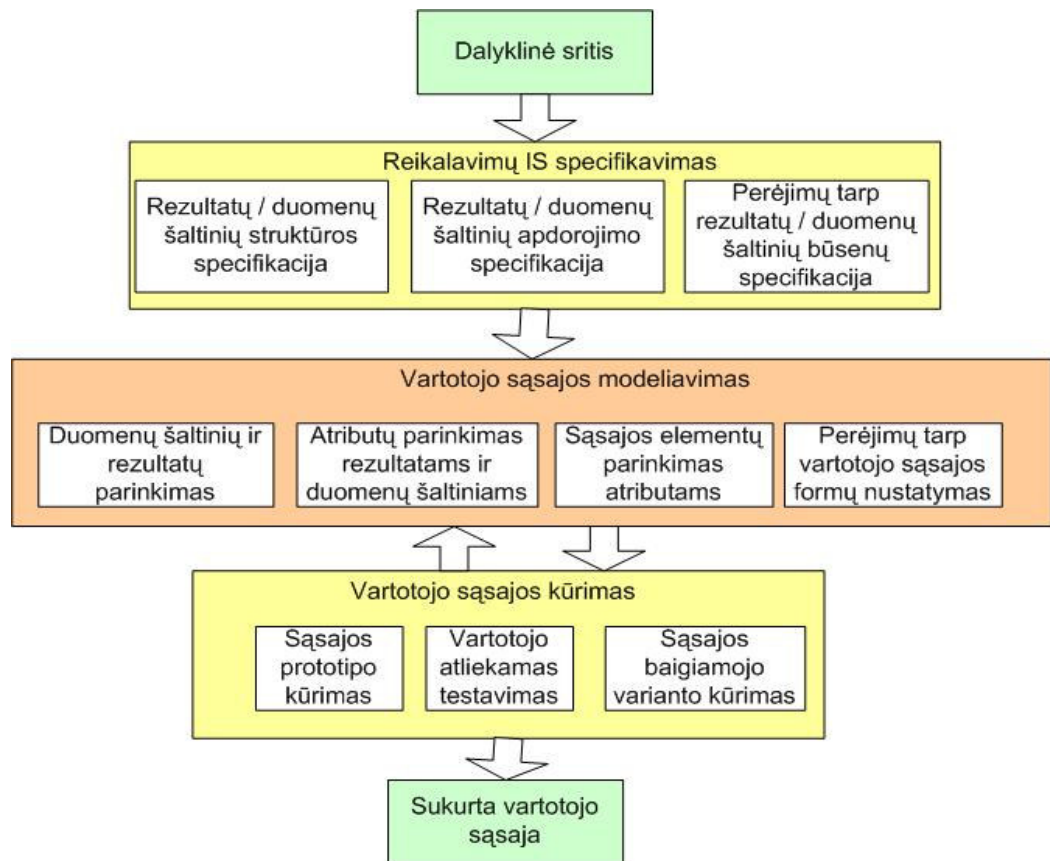


8 pav. Duomenų srautų specifikacijos panaudojimas IS projektavimo procese

Specifikacija saugoma ne tik grafinės notacijos pavidale, bet ir meta duomenų bazėje – saugykloje. Saugykla leidžia praturtinti specifikaciją informacija.

2.2 Duomenų srautų specifikacijos panaudojimas vartotojo sąsajai modeliuoti

FRSM pagrindu sudarytos specifikacijos naudojamos vartotojo sąsajos modeliavimo procese (žr. 9 pav.). Vartotojo sąsajos kūrimas prasideda nuo dalykinės srities nagrinėjimo, pagal ją specifikuojami reikalavimai DSS saugykloje. Remiantis DSS yra modeliuojama vartotojo sąsaja: parenkami duomenų šaltiniai ir rezultatai, jiems parenkami atributai. Parinktiems atributams yra priskiriami vartotojo sąsajos elementai. Sumodeliavus objektams (duomenų šaltiniams ir rezultatams) vartotojo sąsajas, nustatomas eiliškumas tarp šių sąsajos langų. Vartotojo sąsajos kūrimo procesas pateiktas 9 paveiksle.



9 pav. Vartotojo sąsajos kūrimo procesas

Vartotojo sąsajai modeliuoti naudojami šie modeliai:

- Rezultatų / duomenų šaltinių struktūros modelis;
- Rezultatų / duomenų šaltinių apdorojimo modelis;
- Perėjimų tarp rezultatų / duomenų šaltinių būsenų modelis.

Pirmasis modelis suteikia informacijos apie vartotojo sąsajos vieno lango (formos – duomenų šaltinis, ataskaitos – rezultatas) struktūrą. Tam būtent naudojamas į vartotoją orientuotas sluoksnis.

Antrasis modelis suteikia informacijos apie šaltinio arba rezultato apdorojimo procesą, kokiame etape kokie atributai apdorojami, koks aktorius ir kokius veiksmus atlieka su konkrečiu duomenų šaltiniu arba rezultatu.

Trečiasis modelis suteikia informacijos apie būsenas, kurios gali susidaryti darbo metu su šaltiniu ar rezultatu.

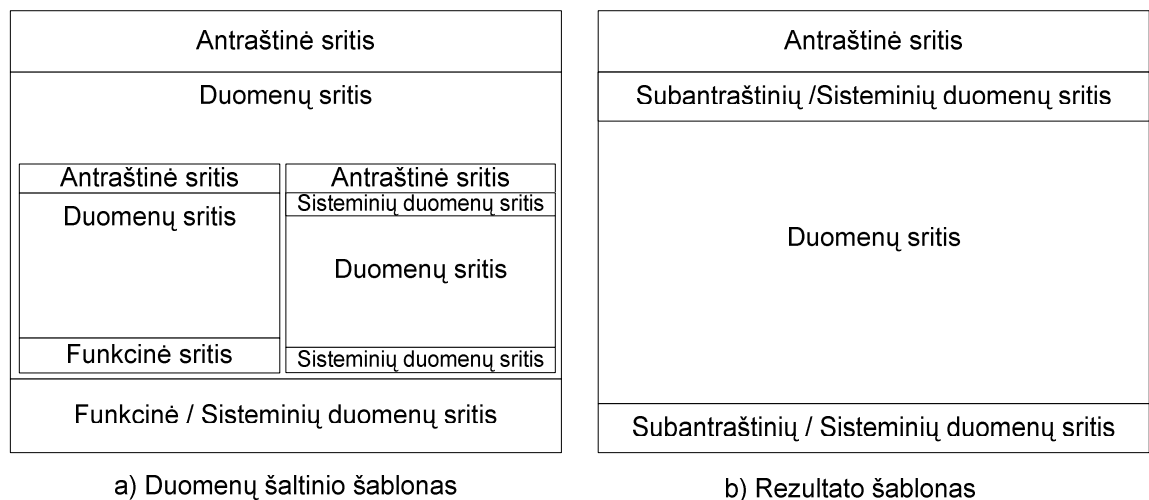
2.3 Vartotojo sąsajos modelio šablonas

Vartotojo sąsajos modeliavimas apima duomenų įvedimo, redagavimo, peržiūrėjimo bei įvairiausių rezultatų (ataskaitų) modelių sudarymą. Vartotojo sąsajos modelių sudarymą galima suskirstyti į dvi fazes:

- Abstraktaus modelio sudarymas;
- Detalaus modelio sudarymas (detalus projektavimas);

Kiekvienam DSS specifikuotam duomenų šaltiniui ir rezultatui turi būti sudarytas abstraktus vartotojo sąsajos modelis ir detalus, kuris sudaromas detalizuojant abstraktųjį modelį. Abstrakčiam modeliui sudaryti reikalingas konceptualus vartotojo sąsajos modelio šablonas, nes visą DSS sudarymo ir panaudojimo kuriamos sistemos projektavime procesą norima automatizuoti. Kiekvienam specifikuotam ir projektavimui etapui sukuriamas programinė realizacija, kurių visuma sudarys CASE įrankio prototipą.

Vartotojo sąsajos atveju, konceptualus šablonas užpildomas automatizuotai kiekvienam DSS-oje specifikuotam duomenų šaltiniui ar rezultatui. Toliau detalaus modelio sudarymo metu šablonas detalizuojamas. 10 paveiksle pateikiami vartotojo sąsajos modelių šablonai. Kairiajame paveiksle pateiktas duomenų šaltinio šablonas, dešiniajame – rezultato šablonas.



10 pav. Konceptualaus vartotojo sąsajos modelio šablonas

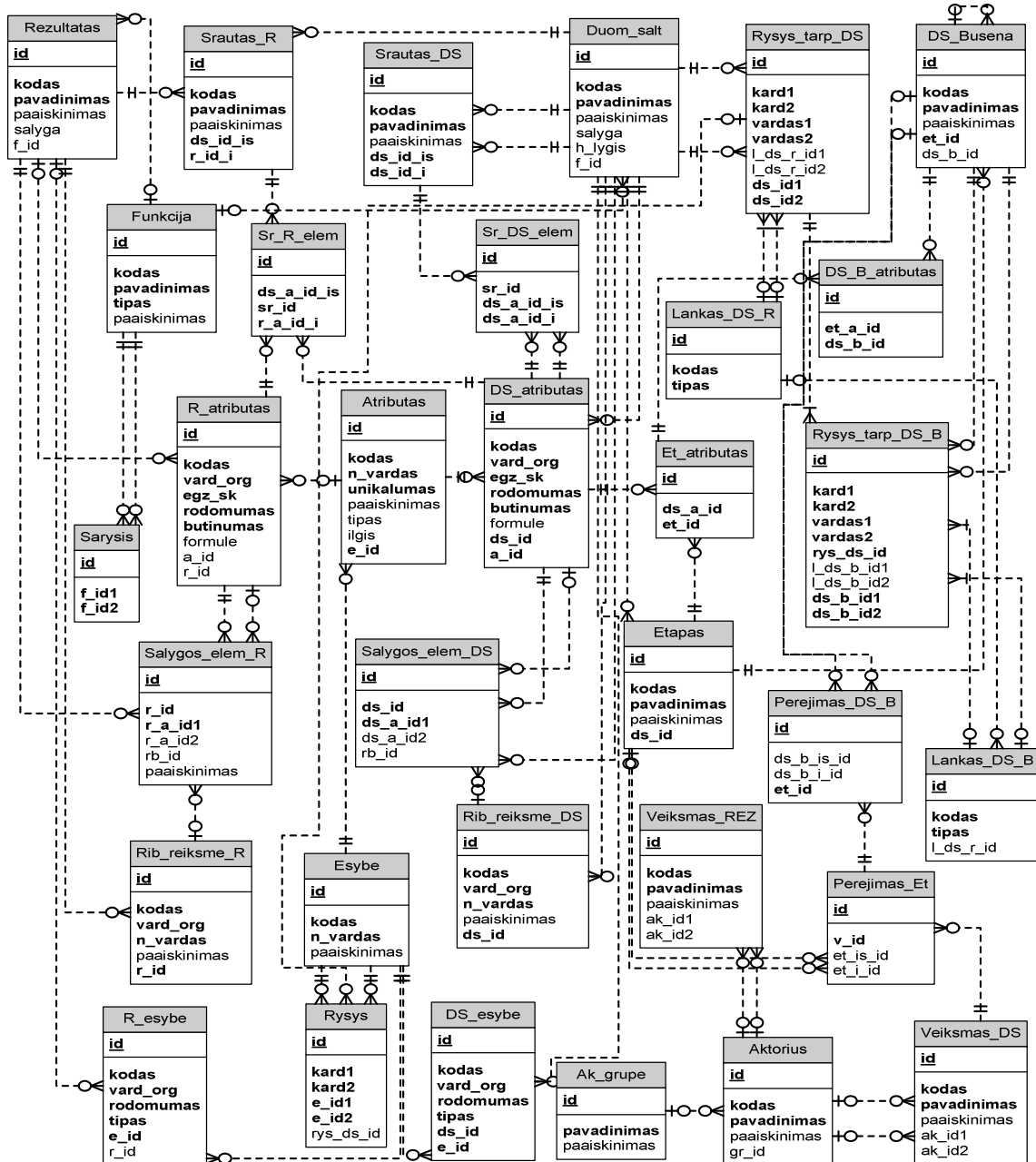
Tiek duomenų šaltinio, tiek rezultato šablonas turi antraštines sritis, kuriose talpinama informacija, leidžianti identifikuoti pagrindinę pateikiamo vartotojo sąsajos lango paskirtį, funkciją. Toliau duomenų šaltinio šablone yra duomenų sritis, kurioje vyksta darbas su duomenimis (įvedimas, redagavimas, paieška, pavaizdavimas). Funkcinė/sisteminių duomenų sritis skirta funkcinių / navigacinių elementų patalpinimui. Taip pat sisteminės informacijos

pateikimui tokios, kaip IS būseną, vartotojo sisteminė informacija, sisteminiai pranešimai ir kt. Priklausomai nuo duomenų šaltinio struktūros, duomenų srityje gal būti talpinamas vėl kitas duomenų šaltinio ar rezultato šablonas.

Tokiu principu modeliuojama formos ir subformos (šablono ir subšablono) struktūra. Subšablonams antraštinės, funkcinės ir sisteminės sritys neprivalomos. Rezultato šablone po antraštinės srities seka Subantraštinių / sisteminių duomenų sritis, kurioje talpinama: rezultato duomenų antraštės, duomenų agregavimo ir apibendrinimo rezultatai bei visa sisteminė informacija, kuri buvo paminėta ir duomenų šaltinio šablono atveju. Duomenų srityje talpinami rezultatui atrinkti duomenys arba tų duomenų agregacijos rezultatai. Subantraštinių / sisteminių duomenų sritis, nėra būtina, todėl modelio sudarymo eigoje ji gali būti eliminuojama.

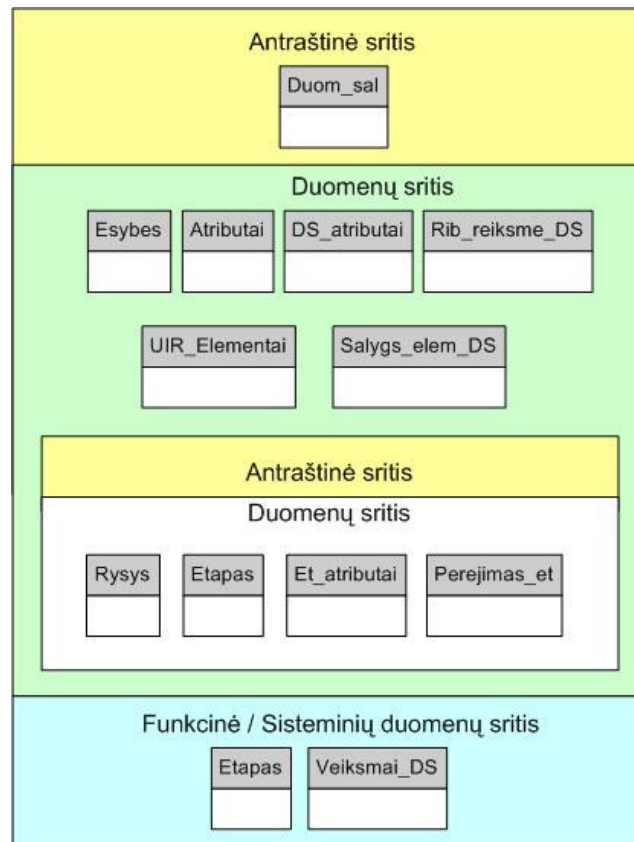
2.4 Konceptualus vartotojo sąsajos sudarymo modelis

Šiame skyriuje pristatomas konceptualaus vartotojo sąsajos modelio sudarymo procesas, kuris bus naudojamas, kaip pagrindas automatizuotame vartotojo sąsajos modelio sudarymo modulyje. Todėl proceso aprašyme naudojamos DSS meta-modelio (žr.11 pav.) meta-esybės ir meta-atributus.



11 pav. Duomenų srautų specifikacijos metamodelis

Šablonas (žr. 10 pav.) užpildomas automatizuotai kiekvienam DSS-oje specifikuotam duomenų šaltiniui ar rezultatui. Duomenų šaltiniui užpildyti reikalingas poaibis visų metamodelio lentelių (žr. 12 pav.).

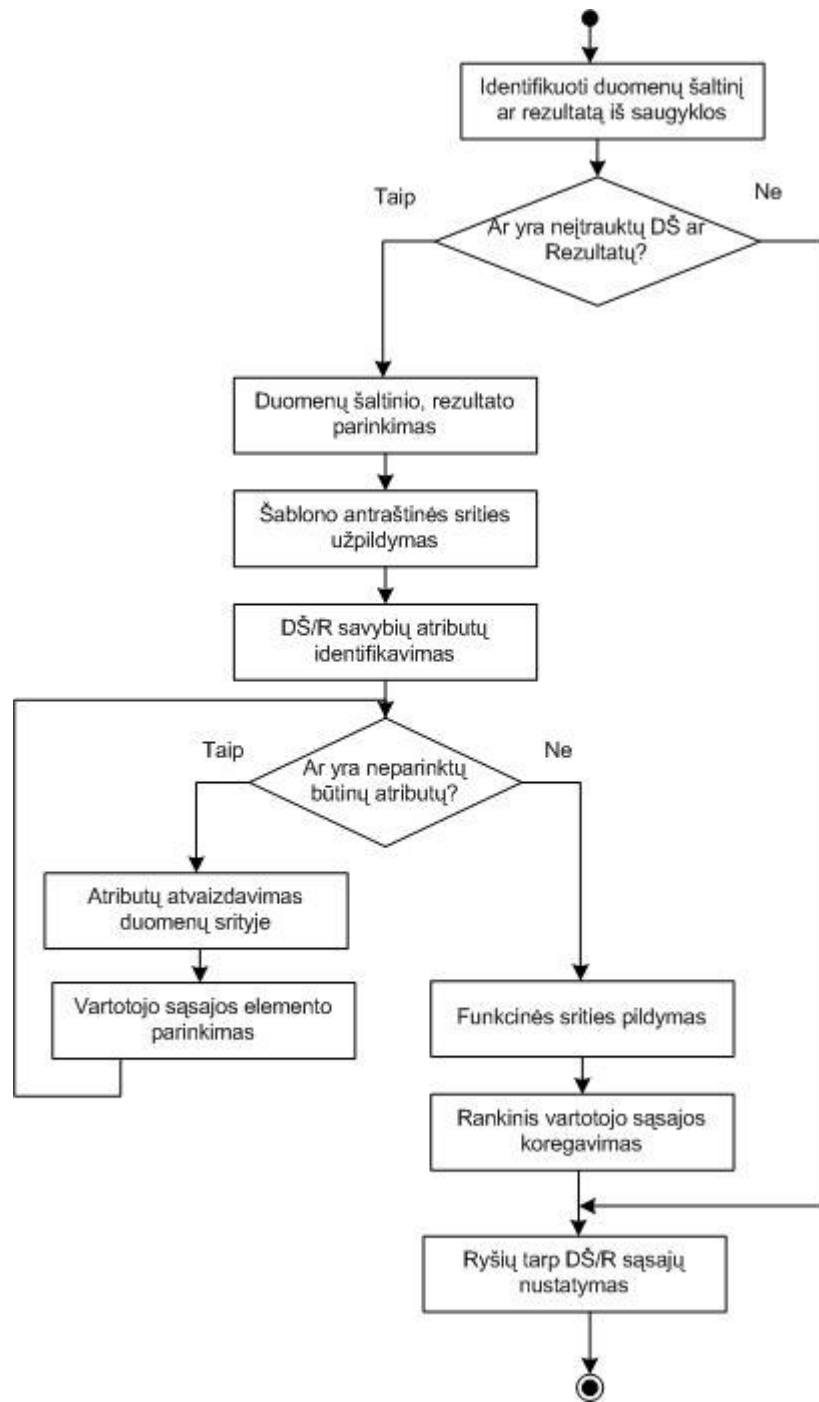


12 pav. DSS metamodelio lentelės, reikalingos duomenų šaltiniui užpildyti

Vartotojo sąsajos modeliavimo duomenų šaltiniui proceso žingsniai:

- 1) Pasirenkamas duomenų šaltinis arba rezultatas, kuriam bus sudaromas vartotojo sąsajos modelis. Meta-modelyje informacija saugoma *Duom_sal* ir *Rezultatas* lentelėse.
- 2) Pagal pasirinkto objekto pavadinimą ir paaiškinamąją informaciją sudaromas pavadinimas, kuris patalpinamas į šablono antraštinę sritį.
- 3) Pasirinktam objektui – duomenų šaltiniui – atrenkamos jo esybės ir atributai iš turimo IS duomenų modelio. Informacija paimama iš *Esybės* ir *Atributai* lentelių.
- 4) Pasirenkama pagrindinė duomenų šaltinio esybė. Šablono duomenų srityje atvaizduojami visi jos atributai (*DS_atributai.var_dorg*), kuriems nustatyta atvaizdavimo savybė (*DS_atributai.rod_mumas*).

- 5) Prie kiekvieno atvaizduoto atributo duomenų srityje sukuriamas jo egzemplioriaus atvaizdavimo vartotojo sąsajos elementas. Elemento tipas parenkamas sekančiu būdu:
- Jei atributas nėra išorinis raktas, jam parenkamas tekstinio įvedimo laukelio tipas;
 - Jei atributas yra išorinis raktas, jam suteikiamas paprasto arba iškrentančio sąrašo tipas, kur sąrašo stulpelių laukai yra esybės, su kurios pirminių rakto atributu susisieja duotasis atributas, atributai;
 - Jei atributas turi nustatytą duomenų aibę arba intervalą ar kitus apribojimus (Rib_reiksme_DS, Sąlygos_elem_DS), jam taip pat gali būti parenkamas paprasto, iškrentančio sąrašo tipas arba kitas vartotojo sąsajos elemento tipas, leidžiantis kontroliuoti įvedamas reikšmes.
- 6) Jei pagrindinė duomenų šaltinio esybė turi sąryšį su kitomis duomenų šaltinio esybėmis ryšių *1:daug* (*Rsysys.kard1*, *Rsysys.kard2*), tai besirišančioms duomenų šaltinio esybėms duomenų srityje sukuriama nauji duomenų šaltinio arba rezultato šablonai (subšablonai) priklausomai nuo to, kokias funkcijas norima atlikti su šios esybės duomenimis. Funkcionalumą galima sužinoti pagal duomenų šaltinio apdorojimo etapų informaciją saugomą meta-modelio esybėse: *Etapas*, *Et_atributai* ir *Perėjimas_et*.
- 7) Pildoma Funkcinė/Sisteminių duomenų sritis tiek pagrindinio šablono, tiek subšablonų. Pagal meta-modelio lentelių *Etapas* ir *Veiksmi_DS* informaciją sužinomos visos funkcijos, kurios atliekamos su duomenų šaltinio informacija. Gautos informacijos pagrindu srityje įvedami funkciniai ir navigacijos mygtukai. Sistemine informacija pildoma bendrojo modelio detalizavimo fazės metu.



13 pav. Vartotojo sąsajos modelio sudarymo procesas

Rezultatui užpildyti reikalingas poaibis metamodelio lentelių pateiktas 14 paveiksle.

| | | | |
|--|-----------|---------------|-------------|
| Antraštinė sritis | | | |
| Rezultatas | | | |
| Subantraštinių / Sisteminių duomenų sritis | | | |
| Esybes | Atributai | R_esybes | R_atributai |
| | | | |
| Duomenų sritis | | | |
| Rysys | | UIR_Elementai | |
| | | | |
| Esybes | Atributai | R_esybes | R_atributai |
| | | | |
| Subantraštinių / Sisteminių duomenų sritis | | | |
| Etapas | | Veiksmai_REZ | |
| | | | |

14 pav. DSS metamodelio lentelės, reikalingos rezultatui užpildyti

Vartotojo sąsajos modeliavimo rezultato proceso žingsniai:

- 1) Pasirenkamas duomenų šaltinis arba rezultatas, kuriam bus sudaromas vartotojo sąsajos modelis. Meta-modelyje informacija saugoma *Duom_sal* ir *Rezultatas* lentelėse.
- 2) Pagal pasirinkto objekto pavadinimą ir paaiškinamąją informaciją sudaromas pavadinimas, kuris patalpinamas į šablono antraštinę sritį.
- 3) Pasirinktam objektui (rezultatui) atrenkamos jo esybės ir atributai ir turimo IS duomenų modelio. Informacija paaimama iš *Esybės* ir *Atributai* lentelių.
- 4) Pagal pasirinkto rezultato aprašą (*Rezultatas.paaiskinimas*) ir duomenų atvaizdavimo (*R_esybes.rodomumas*, *R_atributai.rodomumas* ir *R_atributai.formulė*) informaciją atrenkami rezultato atributai ir eilute išdėstomi Subantraštinių /Sisteminių duomenų srities viršutinėje dalyje (žr. 10 pav.).
- 5) Duomenų srityje suformuojama lentelė, kurios stulpelių skaičių atitiks 4 žingsnio metu atrinktų atributų skaičius, o eilučių skaičių atitiks meta-modelio *R_atributai.egz_sk* atributo maksimali reikšmė pasirinkto rezultato atributų aibėje.

Atlikus šiuos žingsnius turimas abstraktus vartotojo sąsajos modelis, kuris toliau detalizuojamas. Modelio detalizavimas, tai jau rankinis sistemos analitiko/projektuotojo darbas.

Šios fazės metu modelyje papildomos Funkcinė / Sisteminių duomenų ir Subantraštinių / Sisteminių duomenų sritys sisteminė informacija, kuri išgauna iš Rezultatų / duomenų šaltinių struktūros grafinio modelio pirmojo ir antrojo sluoksnių [14]. Pagal šių sluoksnių informaciją gali būti detalizuojamas ir kitos modelio sritys. Galiausiai modelyje turi būti įvertinami būsimos sistemos vartotojo nefunkciniai reikalavimai, kurie turi būti surenkami atskiro proceso metu, nes DSS meta-modelyje nefunkciniai reikalavimai nesaugomi.

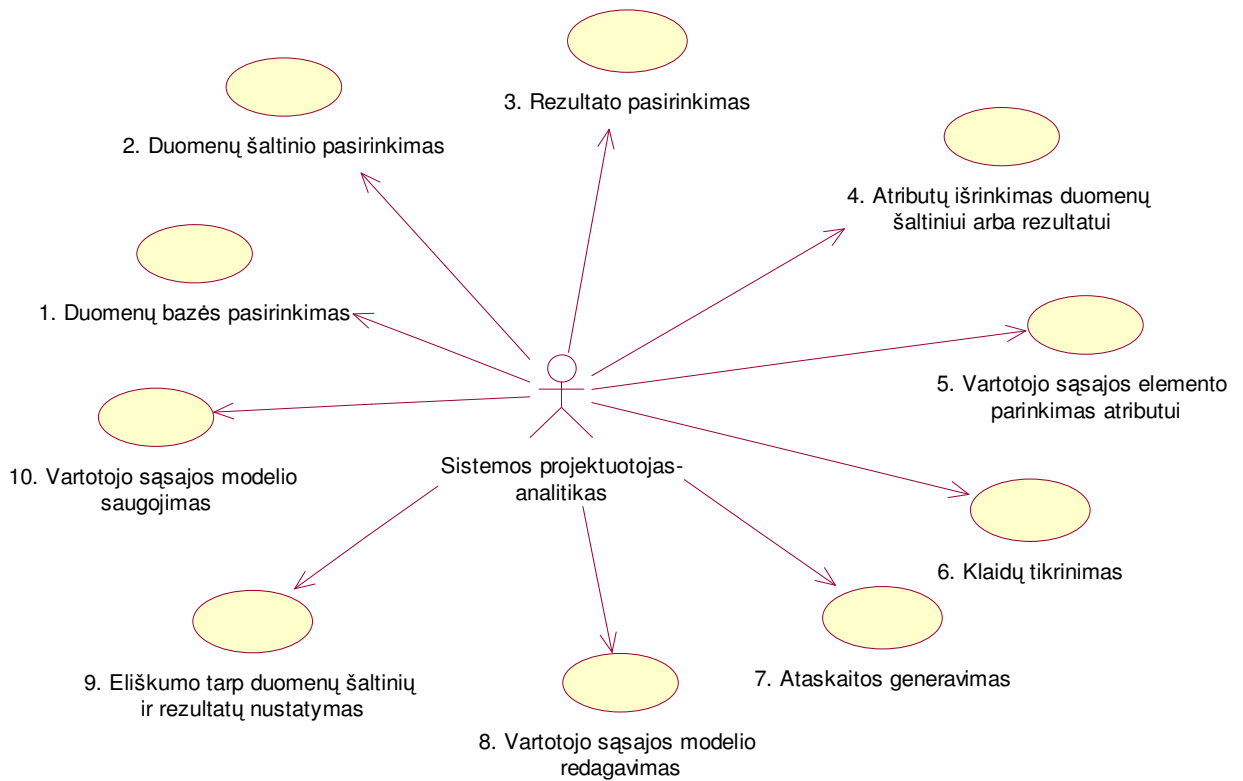
Gautas vartotojo sąsajos modelis gali būti vėliau vėl modifikuotas pasikeitus tiek funkciniais, tiek nefunkciniams reikalavimams, nes DSS sudarymo ir panaudojimo procesas yra iteratyvus.

3 VARTOTOJO SĄSAJOS MODELIAVIMO PROTOTIPO REALIZACIJA

2 skyriuje nagrinėtam vartotojo sąsajos modelių sudarymo duomenų srautų specifikacijos pagrindu metodo ir procesui įgyvendinti buvo sukurtas CASE įrankis.

3.1 Panaudojimo atvejai

Sistemos panaudojimo atvejai pateikti 15 paveiksle.



15 pav. Vartotojo sąsajos modeliavimo prototipo panaudojimo atvejai

1) Duomenų bazės pasirinkimas.

Skirtingiems projektams saugoti naudojamos skirtingos duomenų bazės, todėl svarbu pasirinkti duomenų bazę, kurioje saugoma informacija.

2) Duomenų šaltinio pasirinkimas.

Pateikiamas duomenų šaltinių sąrašas, iš kurio pasirenkamas duomenų šaltinis, kuriam norima modeliuoti vartotojo sąsają.

3) Rezultato pasirinkimas.

Pateikiamas rezultatų sąrašas, iš kurio pasirenkamas duomenų šaltinis, kuriam norima modeliuoti vartotojo sąsają.

4) Atributų išrinkimas duomenų šaltiniui arba rezultatui.

Pasirinkus duomenų šaltinį ar atributui, pateikiami pasirinkto objekto (duomenų šaltinio ar rezultato) atributai. Į vartotojo sąsajos modelį gali būti įtraukti ne visi atributai. Tie atributai, kurie metaduomenų bazės lentelių laukų *DS_ atributai.rodomumas* ir *R_ atributai.rodomumas* turi teigiamą reikšmę, turi būtinai būt įtraukti į vartotojo sąsajos modelį.

5) Vartotojo sąsajos elemento parinkimas atributui

Parinkamas vartotojo sąsajos elementas iš pateikto sąrašo. Jei atributas nėra išorinis raktas, jam parenkamas tekstinio įvedimo laukelio tipas. Jei atributas yra išorinis raktas, jam suteikiamas paprasto arba iškrentančio sąrašo tipas, kur sąrašo stulpelių laukai yra esybės, su kurios pirminių rakto atributu susisieja duotasis atributas, atributai. Jei atributas turi nustatytą duomenų aibę arba intervalą ar kitus apribojimus (*Rib_reiksme_DS*, *Sąlygos_elem_DS*), jam taip pat gali būti parenkamas paprasto, iškrentančio sąrašo tipas arba kitas vartotojo sąsajos elemento tipas, leidžiantis kontroliuoti įvedamas reikšmes.

6) Klaidų tikrinimas

Tikrinamas atributų būtinumas. Duomenų šaltinio atributai su būtinumo savybe turi būtinai būti įtraukti į sąsajos modelį. Tikrinamas atributų dubliavimasis. Sistema leidžia patikrinti ar atributai iš duomenų šaltinių nėra įtraukti į sąsajos modelį kelis kartus. Tikrinama, ar pagal 5 punkte aprašytus kriterijus parinktas teisingas vartotojo sąsajos tipas.

7) Ataskaitos generavimas.

Sistema leidžia sugeneruoti ataskaitas apie panaudotus vartotojo sąsajos elementus ir jų kiekį į šiuos formatus: xls, txt.

8) Vartotojo sąsajos modelio redagavimas

Sumodeliuotas vartotojo sąsajos modelis gali būti redaguojamas – keičiama sąsajos elementų tvarka, vieta, tipas. Pakoregavus modelį, duomenys išsaugomi grafinėje notacijoje ir saugykloje.

9) Eliškumo tarp duomenų šaltinių ir rezultatų nustatymas

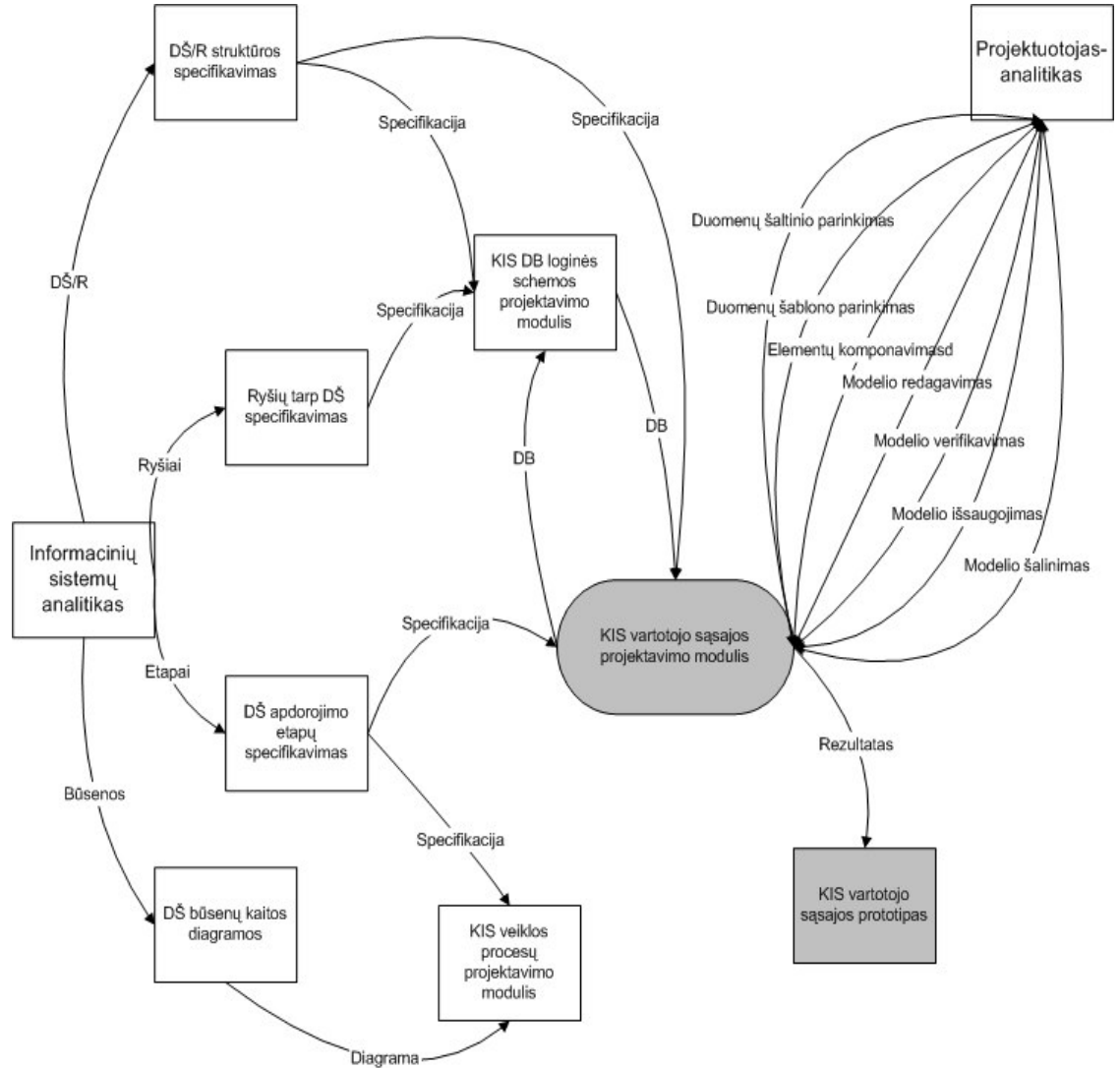
Nustatomas eiliškumas tarp duomenų šaltiniams ir rezultatams sukurtų vartotojo sąsajų.

10) Vartotojo sąsajos modelio saugojimas.

Modelį galima išsaugoti į saugyklą – duomenų bazę arba grafinėje notacijoje – MS Visio faile.

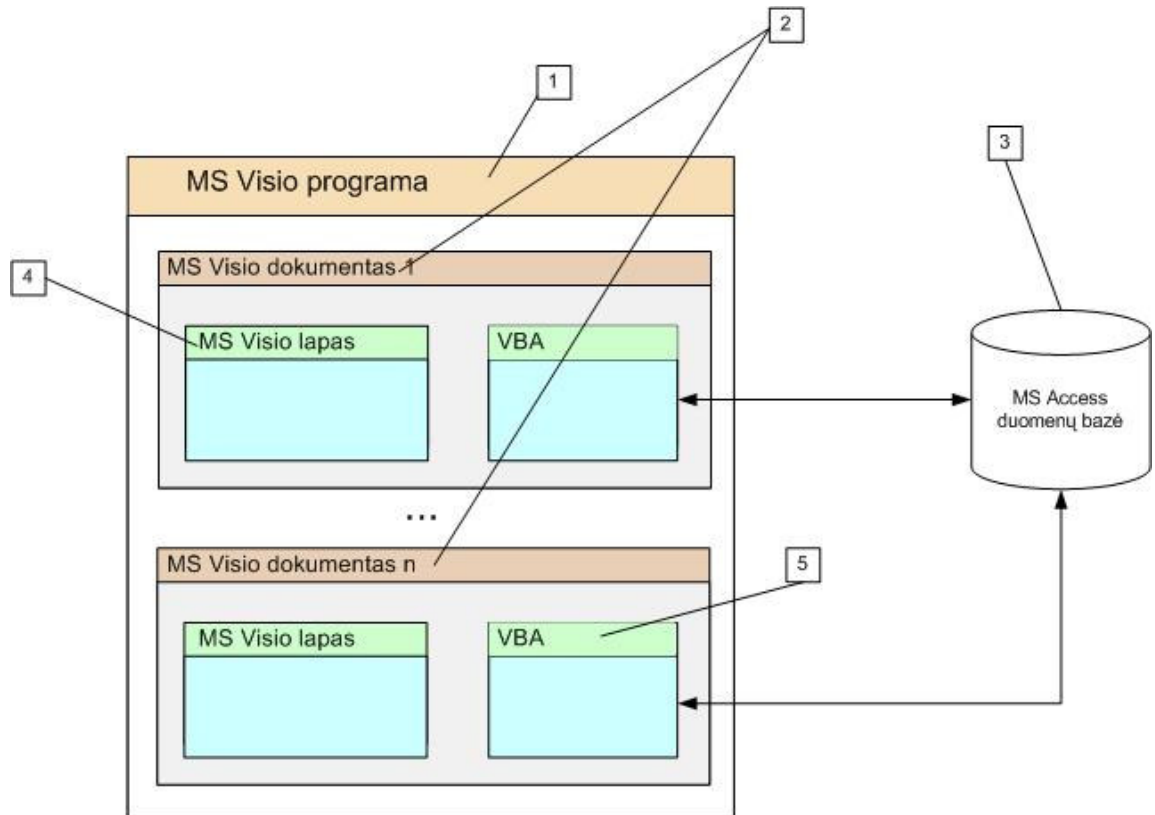
3.2 Veiklos kontekstas

Veiklos konteksto diagramoje apibrėžiamos sistemos ribos, išorinės esybės, kurios bendrauja su sistema, bei pagrindiniai informacijos srautai tarp sistemos ir išorinių esybių (16 pav.).



16 pav. Konteksto diagrama

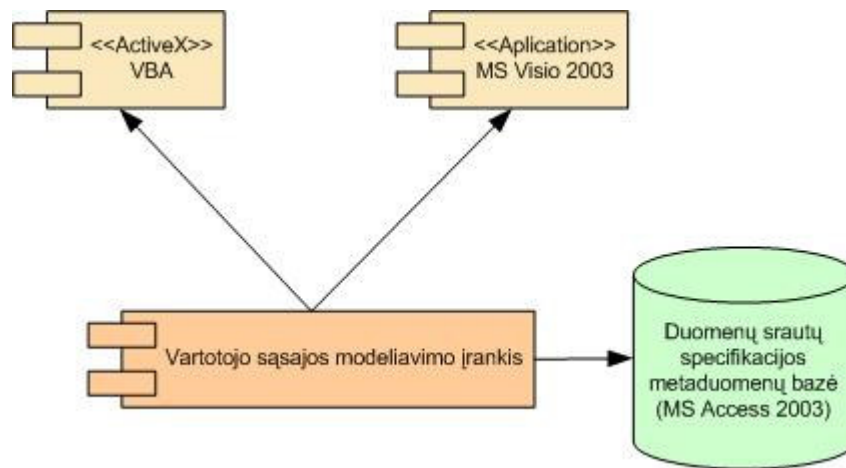
3.3 Bendra prototipo struktūra



17 pav. CASE įrankio struktūrinė schema

Įrankis veikia MS Visio 2003 aplinkoje (17 pav., 1), kuri leidžia sukurti pageidaujamus grafinius objektus, o trūkstamas funkcijas bei procedūras realizuoti VBA (Visual Basic for Application) (17 pav., 5). Duomenų srautų specifikacijos saugykla yra sukurta MS Access 2003 duomenų bazių valdymo sistema (17 pav., 3). Sumodeliuota vartotojo sąsajos saugykla yra ne vienalytė. Ją sudaro MS Access 2003 duomenų bazė (17 pav., 3) ir MS Visio failai (17 pav., 4), saugantys specifikaciją grafinėje notacijoje (17 pav., 2). Įrankio vartotojas – sistemos analitikas.

Vartotojo sąsajos modeliavimo prototipo komponentų diagrama pateikta 18 paveiksle.



18 pav. Komponentų diagrama

3.4 Taisyklės vartotojo sąsajos modelio teisingumo tikrinimui

Sukūrus vartotojo sąsajos modelį būtina jį patikrinti. Tai atliekama pasirinkus meniu punktą „Tikrinti klaidas“.

Tikrinimo sąlygų aprašymai:

- Tikrinamas atribūtų būtinumas. Duomenų šaltinio atributai su būtinumo savybe turi būtinai būti įtraukti į sąsajos modelį.
- Tikrinamas atribūtų dubliavimasis. Sistema leidžia patikrinti ar atributai iš duomenų šaltinių nėra įtraukti į sąsajos modelį kelis kartus.
- Tipų tikrinimas

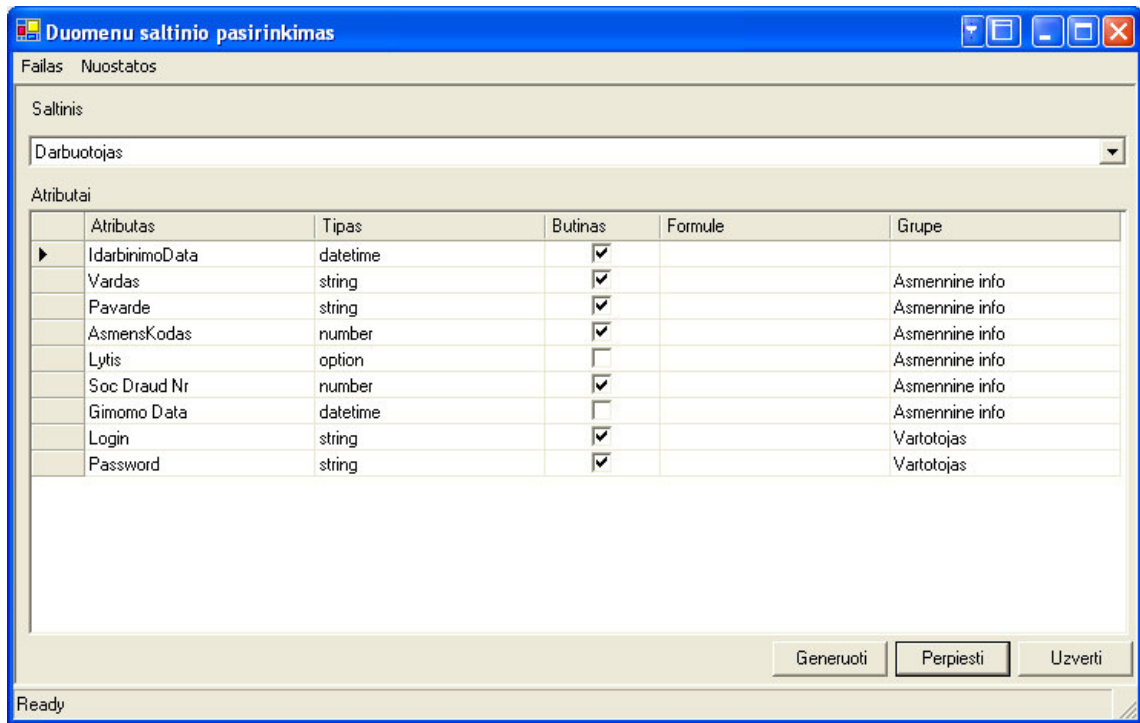
3.5 Vartotojo sąsajos prototipo veikimas

Vartotojo sąsajos modeliavimo prototipo veikimas pademonstruotas pasirinkus testinius duomenis, pagal kuriuos sumodeliuota vartotojo sąsaja. Testiniais duomenimis užpildoma saugykla. Metamodelio lentelių testiniai duomenys pateikiami priede 3, 4, 5 lentelėse.

3.5.1 Duomenų šaltinio pasirinkimas

Duomenų bazės lentelėse jau yra informacija apie duomenų šaltinius. Galima pasirinkti, kurio duomenų šaltinio atributus vaizduosite vartotojo sąsajos modelyje. Dialogo lange «**Duomenų šaltinio pasirinkimas**» spragtelėkite ant pasirinkimo mygtuko ir programa pateiks sąrašą šaltinių iš kurių vienu metu galite pasirinkti tik vieną (žr. 19 pav.). Pasirinkus duomenų

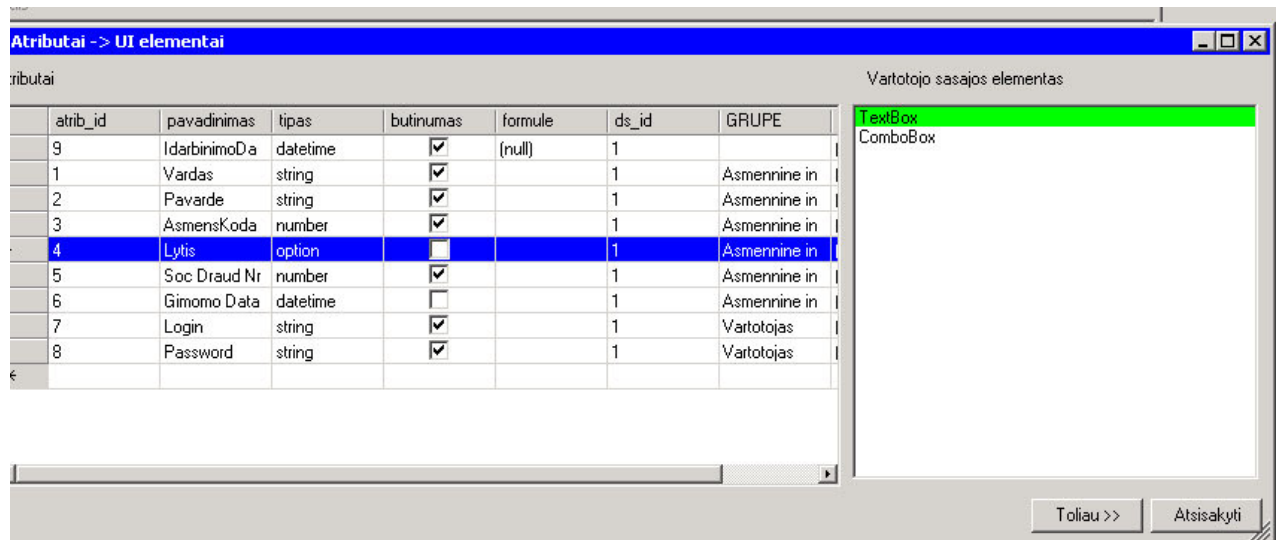
šaltinį, srityje Atributai bus atvaizduoti visi to šaltinio atributai iš duomenų bazės lentelės. Prieš renkantis duomenų šaltinį turi būti pasirinkta duomenų bazė.



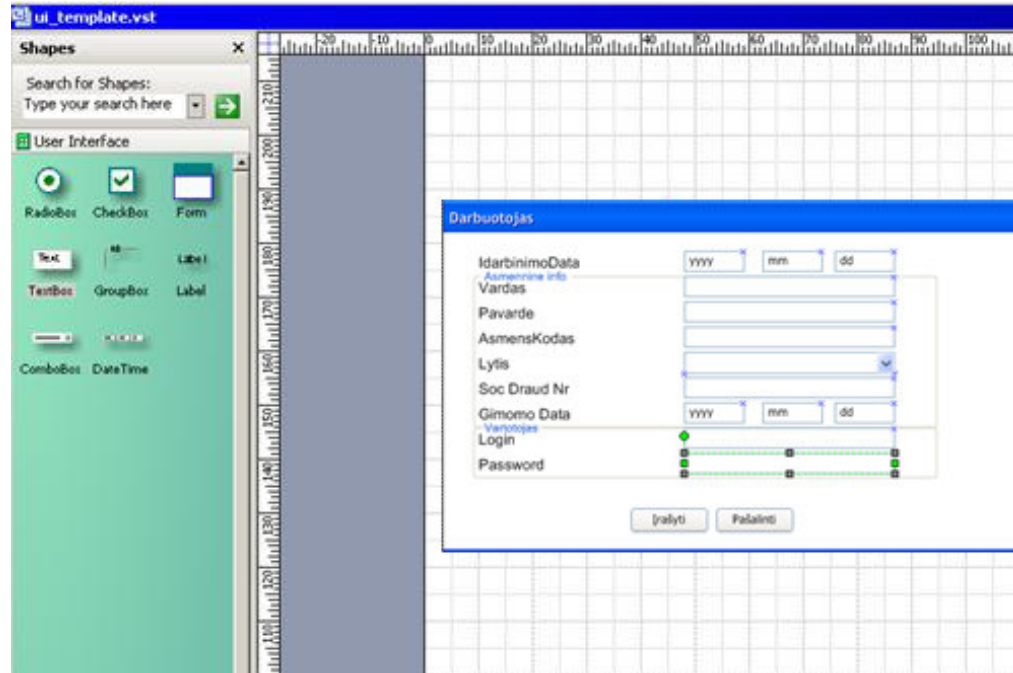
19 pav. Duomenų šaltinio pasirinkimas

3.5.2 Atributams vartotojo sąsajos elementų parinkimas

Vartotojo sąsajos elementų parinkimas atributams įgalina iš pateikto sąrašo išsirinkti norimą vartotojo sąsajos elementą (žr. 20 pav.) ir spragtelėjus mygtuką «Toliau» automatiškai sugeneruojamas vartotojo sąsajos langas (žr. 21 pav.).

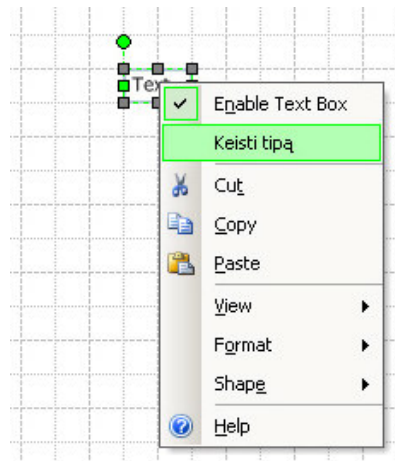


20 pav. Vartotojo sąsajos elementų pasirinkimas

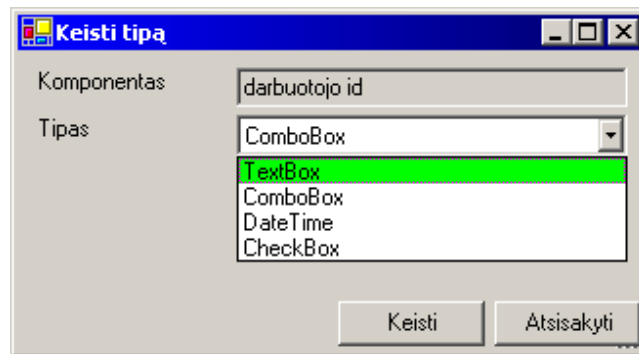


21 pav. Automatiškai sugeneruota vartotojo sąsaja

Detalizuojant automatiškai sugeneruotą modelį, galima pakeisti vartotojo sąsajos elemento tipą. Tam naudojamas kontekstinis meniu, iškviečiamas spragtelėjus ant sąsajos elemento ir pasirinkus punktą „Keisti tipą“ (žr. 22 pav.) leidžiama pasirinkti iš pateiktų sąrašo kitą sąsajos elemento tipą (žr. 23 pav.).

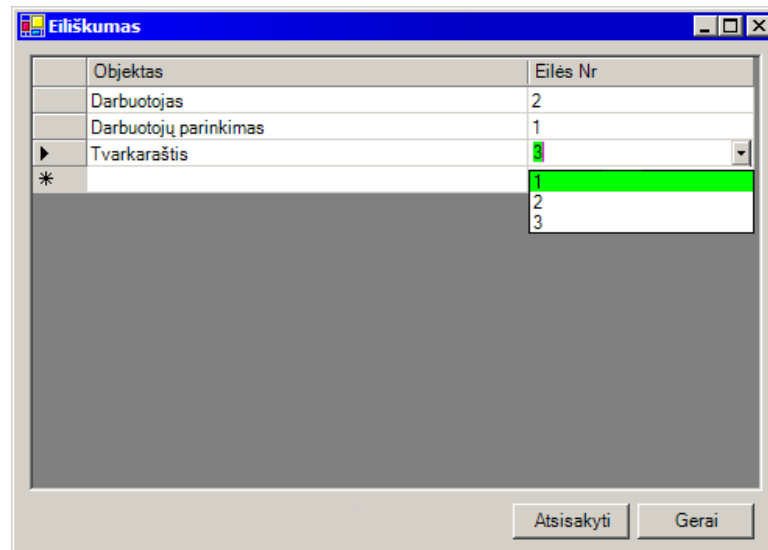


22 pav. Kontekstinis meniu vartotojo sąsajos elemento tipui keisti



23 pav. Vartotojo sąsajos elemento keitimas

Sumodeliavus vartotojo sąsają objektams (duomenų šaltiniams ir rezultatams), nustatomas eiliškumas tarp duomenų šaltiniams ir rezultatams sukurtų vartotojo sąsajų (žr. 24 pav.).

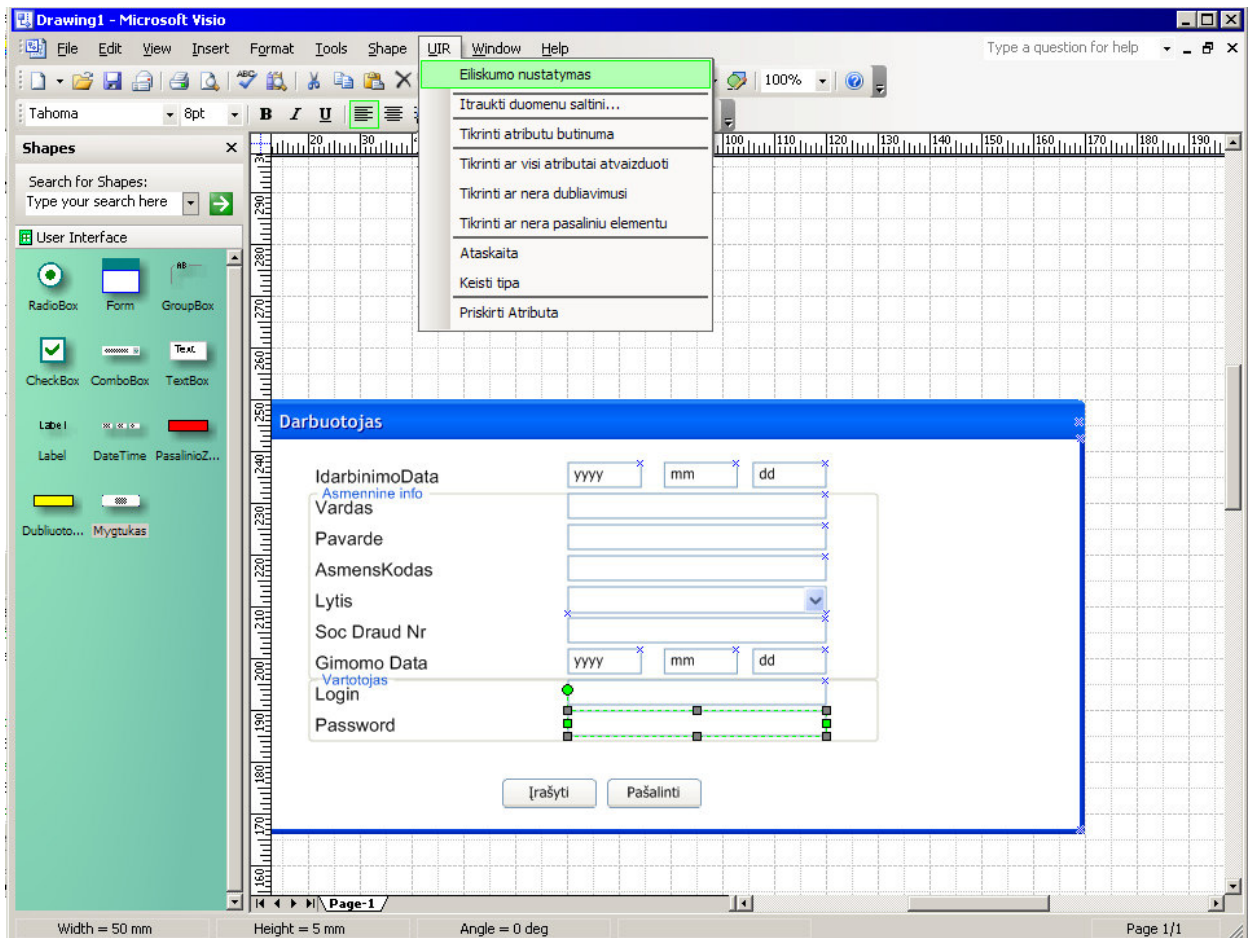


24 pav. Eiliškumo nustatymas objektams

3.5.3 Apribojimų tikrinimas

MS Visio programos meniu yra išplėstas papildomomis įrankio savybėmis. Meniu punktas UIR skirtas apribojimų vartotojo sąsajai patikrinti (žr. 25 pav.). Galimi tokie tikrinimai:

- Tikrinamas atributo būtinumas
- Tikrinama ar visi būtini atributai yra atvaizduoti.
- Tikrinama ar nėra dubliavimosi.
- Tikrinama ar nėra pašalinių elementų.



25 pav. MS Visio meniu išplėtimas

3.5.4 Vartotojo sąsajos elementų ataskaita

Ataskaitoje pateikiami visi pavaizduoti vartotojo sąsajos elementai, jų kiekis, duomenų šaltiniai (žr. 26 pav.). Ataskaita iškviečiama iš praplėsto meniu UIR MS Visio programos lange.

The screenshot shows the 'FormReport' window displaying a report titled 'Komponentų pasiskirstymas' dated 22/12/2005. The report contains a table with the following data:

| Duomenų šaltinis | ComboBox | RadioButton | ListBox | CheckBo | DateTime | Other | Visible | Totall |
|--------------------|----------|-------------|---------|---------|----------|-------|---------|--------|
| Pajamavimo orderis | 9 | 5 | 5 | 3 | 5 | 13 | 32 | 40 |
| Vartotojas | 10 | 6 | 1 | 0 | 0 | 5 | 5 | 22 |
| Faktura | 9 | 5 | 5 | 3 | 5 | 13 | 32 | 40 |

The status bar at the bottom indicates 'Current Page No: 1', 'Total Page No: 1', and 'Zoom Factor: 100%'.

3.6 Lyginamoji analizė ir praktinė nauda

1 ir 2 lentelėje pateiktas metodų palyginimas.

1 lentelė. Pasiūlyto, Oracle CASE, UML, D. J. Anderson metodų palyginimas

| Eil. Nr. | Kriterijaus pavadinimas | Pasiūlytas modelis | Oracle CASE metodas | UML grindžiama metodika | D. J. Anderson modelis |
|----------|---|--------------------|---------------------|-------------------------|------------------------|
| 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Programinė realizacija | + | + | +/- | - |
| 2 | Vartotojo sąsajos modeliavimas naudojant sąsajos elementus | + | +/- | +/- | + |
| 3 | Specifikacijos saugykla | + | + | + | - |
| 4 | Abipusis ryšys tarp saugyklos ir modelio | + | +/- | - | - |
| 5 | Galimybė išplėsti duomenų saugyklą | + | + | - | - |
| 6 | Vartotojo sąsajos modelio adekvatumas vartotojo susikurtam įvaizdžiui | + | +/- | +/- | + |
| 7 | Reikalavimų specifikavimo proceso eigos atitikimas natūraliam bendravimui | + | +/- | +/- | + |
| 8 | Vartotojo nefunkcinių reikalavimų saugojimas | + | - | + | + |
| 9 | Standartų palaikymas | - | + | + | - |
| 10 | Kodo generavimas | - | + | +/- | - |

2 lentelė. Pasiūlyto, šablonu paremto, nuorodų sistemos funkcionalumu pagrįsto metodų palyginimas

| Eil. Nr. | Kriterijaus pavadinimas | Šablonu paremtas metodas | C. Finkelstein metodika | Nuorodų sistemos funkcionalumu pagrįstas metodas |
|----------|---|--------------------------|-------------------------|--|
| 1 | 2 | 3 | 4 | 5 |
| 1 | Programinė realizacija | - | - | - |
| 2 | Vartotojo sąsajos modeliavimas naudojant sąsajos elementus | + | + | - |
| 3 | Specifikacijos saugykla | - | - | - |
| 4 | Vartotojo sąsajos modelio adekvatumas vartotojo susikurtam įvaizdžiui | + | + | - |
| 5 | Abipusis ryšys tarp saugyklos ir modelio | - | - | - |
| 6 | Galimybė išplėsti duomenų saugyklą | - | - | - |
| 7 | Reikalavimų specifikavimo proceso eigos atitikimas natūraliam bendravimui | +/- | + | + |
| 8 | Vartotojo nefunkcinių reikalavimų saugojimas | + | + | - |
| 9 | Standartų palaikymas | - | - | - |
| 10 | Kodo generavimas | - | - | - |

IŠVADOS

- 1) Išanalizavus žinomus vartotojo sąsajos modeliavimo metodus, išskirti reikalavimai, kurių laikomasi kuriant vartotojo sąsają. Tai informacijos poreikių, funkcionalumo, duomenų registracijos, racionaliausios vartotojo sąsajos alternatyvos, vartotojo nefunkciniai reikalavimai.
- 2) Oracle CASE pagrindu įdiegta Oracle Designer priemonė teikia galimybę specifiuoti vartotojo sąsajos reikalavimus, tačiau kaip ir UML atveju vartotojo sąsajos elementai nėra visiškai pilnai specifiuojami. Tiek C. Finkelstein metodika, tiek nuorodų sistema paremtas, tiek šablonais grindžiamas modeliavimo metodai neturi išvystytų priemonių metodikai įgyvendinti.
- 3) Pateiktas vartotojo sąsajos modeliavimo duomenų srautų specifikacijos pagrindu metodas, įvertinantis funkcinis reikalavimus, kurie saugomi duomenų srautų specifikacijos saugykloje, o nefunkciniai reikalavimai – grafinėje notacijoje. Modeliavimas atliekamas remiantis atskirai kiekvieno duomenų šaltinio apdorojimo etapų ir struktūros informacija saugykloje, todėl dar neturint pilno kompiuterizuoto IS loginės schemos modelio galima turėti vartotojo sąsajos elementų prototipą.
- 4) Duomenų saugykloje yra saugojami vartotojo sąsajos rezultatai – IS formų ir ataskaitų šablonai. Egzistuoja abipusis ryšys tarp saugyklos ir modelio, todėl pakoregavus šablonus galima saugykloje išsaugoti atliktas korekcijas ir jas susieti su saugyklos elementais.
- 5) Sukurtas vartotojo sąsajos modeliavimo duomenų srautų specifikacijos pagrindu koncepcijos programinis prototipas. Programinio prototipo kūrimui pasirinkta MS Visio 2003 aplinka, trūkstamos funkcijos bei procedūros realizuotos VBA (Visual Basic for Application). Duomenų srautų specifikacijos saugyklai pasirinkta MS Access 2003.
- 6) Su realizuotu vartotojo sąsajos prototipu atliktas eksperimentinis tyrimas ir lyginamojo analizė parodė, kad sukurtas prototipas atitinka daugumą kriterijų – yra abipusis ryšys tarp modelio ir saugyklos, galima išplėsti duomenų saugyklą, vartotojo sąsajos modelis adekvatus vartotojo susikurtam įvaizdžiui, tačiau negeneruoja programos kodo iš vartotojo sąsajos modelio, ribotos galimybės pilnai surinkti visus standartinius vartotojo sąsajos elementus.

- 7) Dalyvauta 9-oje tarpuniversitetinėje magistrantų ir doktorantų konferencijoje Informacinė visuomenė ir universitetinės studijos, skaitytas pranešimas „Vartotojo sąsajos modeliavimas funkcinių reikalavimų pagrindu“ [16]; konferencijoje Informacinės technologijos 2005 skaitytas pranešimas „Vartotojo sąsajos modelių sudarymas informacijos srautų specifikacijos pagrindu“ [13]; konferencijoje Informacinės technologijos – 2006 skaitytas pranešimas „MDA Case įrankių analizė“ [24].

TERMINŲ IR SANTRUMPŲ ŽODYNAS

CASE

(CASE – *computer-aided software engineering*) programinė įranga ar programų paketai, skirti supaprastinti programų sistemų kūrimą ir palaikymą, įvairūs įrankiai skirti skirtingiems programų kūrimo ciklo etapams – projekto valdymui, verslo ar funkcinei analizei, projektavimui, kodo saugojimui ir versijų sudarymui, kompiliavimui, testavimui ir pan.

Dls

Ryšių tarp duomenų šaltinių, duomenų šaltinių ir rezultatų bei ryšių struktūros modelis.

Dp

Rezultatų / duomenų šaltinių apdorojimo modelis.

Dst

Perėjimų tarp rezultatų / duomenų šaltinių būsenų modelis.

DSS

Duomenų srautų specifikacija yra reikalavimų informacijos sistemai specifikacija, susidedanti iš modelių: funkcijų hierarchijos; rezultatų / duomenų šaltinių struktūros; ryšių tarp duomenų šaltinių, duomenų šaltinių ir rezultatų bei ryšių struktūros; rezultatų / duomenų šaltinių apdorojimo; perėjimų tarp rezultatų / duomenų šaltinių būsenų; ryšių tarp rezultatų – duomenų šaltinių patikslinimo.

EI

Ryšių tarp rezultatų – duomenų šaltinių patikslinimo modelis.

Fh

Funkcijų hierarchijos modelis.

FRSM

Funkcinių reikalavimų specifikavimo metodas, kurio pagrindinė idėja – organizacijos informacinių srautų analizė.

Rds

Rezultatų / duomenų šaltinių struktūros modelis.

UML

Programų sistemų projektavimo rezultatų vaizdavimo, specifikavimo, konstravimo ir dokumentavimo kalba., apibrėžianti 12 diagramų tipų.

LITERATŪRA

- 1 Anderson, D. J. *User Interface Modeling*. [Žiūrėta 2003-11-10]. Prieiga per internetą: <http://www.uidesign.net/1999/papers/UI_Modelling1.html>.
- 2 Anderson, D. J. *Extending UML for UI*. [Žiūrėta 2003-10-21]. Prieiga per internetą: <<http://www.uidesign.net/2000/papers/TUPISproposal.html>>.
- 3 Andrijauskas, A.; Butleris, R. Vartotojų reikalavimų sąsajai specifikuojimo modelis. Kompiuterininkų dienos -99: devintosios mokslinės-praktinės kompiuterininkų konferencijos ir ketvirtosios mokyklinės informatikos konferencijos mokslo darbai, Biržtonas, 1999 m. rugsėjo 16-18 d. D. 1. Vilnius, 1999, p. 9-13.
- 4 Balasubramanian, V. *Supporting the User Interface Design Process with Hypertext Functionality*. [Žiūrėta 2003-11-05]. Prieiga per internetą: <<http://eies.njit.edu/~turoff/Papers/uihtf/uihtf.html>>
- 5 Barker, R.; Longman, R. *Case Method, function and process modeling*. Addison-Wesley, 1992.
- 6 Barker, R. *Case Method, Tasks and deliverables*, Addison-Wesley, 1989.
- 7 Butkienė, R. *Informacijos sistemų projektavimas Oracle Designer/200 priemonėmis: mokomoji knyga*. Kaunas, 1998. - 166 p.
- 8 Butkienė, R; Butleris, R. *The Approach for the User Requirements Specification*. 5th East-European conference ADBIS'2001, Research Communications, Ed. by A.Čaplinskis, J.Eder, Vilnius, 2001, p. 225-240.
- 9 Butkienė, R; Savickis, S. *Reikalavimų KIS įvedamai sistemai ir išvedamai informacijai specifikuojimas*. Informacinės technologijos 2001: konferencijų pranešimų medžiaga. Kaunas, 2001, p.401-407, ISBN 9955-09-119-3
- 10 Butleris, R.; Danikauskas, T. *Conceptual Data model Design Using Functional Requirements Specification Method*. Proceedings of EMMSAD'04 workshop at CaiSE'04 conference. Riga, Latvija, 2004, p. 221-232.

- 11 Butleris, R.; Danikauskas, T.; Misevičiūtė, B. *Informacijos sistemos kompiuterizuotų darbo vietų projektavimas*. Informacinės technologijos '2004: konferencijos pranešimų medžiaga. Kaunas, 2004, p. 533.
- 12 Danikauskas, T; Butleris, R.. *Reikalavimų specifikavimo Oracle CASE terpėje plėtra*. Informacijos mokslai: mokslo darbai. Kaunas, 2001, T.19, p. 51-62, ISSN 1392-0561.
- 13 Danikauskas, T.; Eidukynaitė, V.; Butleris, R. *Vartotojo sąsajos modelių sudarymas informacijos srautų specifikacijos pagrindu*. Informacinės technologijos 2005: konferencijos pranešimų medžiaga. T. 2. Kaunas, 2005, p. 610-617, ISBN 9955-09-788-4.
- 14 Danikauskas, T.; Misevičiūtė, B. *Reikalavimų specifikacijos modeliais grindžiamas IS vartotojo sąsajos projektavimas*. Informacinės technologijos verslui – 2004: konferencijos pranešimų medžiaga. Kaunas, VU KHF, 2004, T. 1., p. 59-64, ISBN 9955-09-649-7.
- 15 Díaz, J. S.; Pastor, O.; Fons, J. J. *From User Requirements to User Interfaces: A Methodological Approach*, Lecture Notes In Computer Science, 2001, Vol. 2068 archive Proceedings of the 13th International Conference on Advanced Information Systems Engineering, p. 60-75.
- 16 Eidukynaitė, V.; Butleris, R. *Vartotojo sąsajos modeliavimas funkcinių reikalavimų pagrindu*. Informacinė visuomenė ir universitetinės studijos: 9-oji tarpuniversitetinė magistrantų ir doktorantų konferencijos pranešimų medžiaga. Kaunas, 2004, p. 187-193.
- 17 Granlund, Å; Lafrenière, D; Carr, D. A. *A Pattern-Supported Approach to the User Interface Design Process*. [Žiūrėta 2003-11-05]. Prieiga per internetą: <<http://citeseer.nj.nec.com/granlund01patternsupported.html>>.
- 18 Harmelen, M. V. *Object Modeling and User Interface Design*. Edison-Wesley, 2001.
- 19 Hasselbring, W.; Kröber, A. *Combining OMT with a prototyping*. The Journal of System and oftware, 1998, lapkritis, 43 tomas, 3 numeris, p. 187 – 196.
- 20 Kiauleikis, M. *Vartotojo sąsaja duomenų inžinerijoje*. Informacinė visuomenė ir universitetinės studijos 2001: 6-oji magistrantų ir doktorantų konferencija. Kaunas, 2001, p. 110-114.

- 21 Maciaszek, L. A. *Requirements Analysis and System Design Developing Information Systems*. Adison-Wesley, 2001.
- 22 Myers, B.A. *User-interface tools: introduction and survey*. Software, 1989, IEEE Volume 6, Issue 1, p. 15-3.
- 23 Molina, P.J. *A Review to Model-Based User Interface Development Technology*. IUI/CADUI 2004 workshop, Electronic proceedings, Portugal, Madeira, 2004.
- 24 Packevičius, Š.; Eidukynaitė, V.; Ušaniov, A. *MDA Case įrankių analizė*. Informacinės technologijos – 2006. T. 1. Kaunas, 2005, p. 161-166.
- 25 Ravid, A; Berry, D. M. *A Method for Extracting and Stating Software Requirements that a User Interface Prototype Contains*. Requirement Engineering. 5 tomas, 4 numeris, p. 225-241.
- 26 UML Resource Page. [Žiūrėta 2003-11-19]. Prieiga per internetą <<http://www.omg.org/technology/uml/index.htm>>.
- 27 Saiedian, H.; Dale, R. *Requirements engineering: making the connection between the software developer and customer*. Information and Software Technology. Vol.42, Elsevier, 2000, p. 419-428.
- 28 Silva, P. P; Paton, N. W. *User Interface Modelling with UML*. Information Modelling and Knowledge Bases XII - 10th European-Japanese Conference on Information Modelling and Knowledge Representation, Saariselka, Finland, May 2000). Amsterdam, 2001 p. 203-217.
- 29 Six, H. W.; Voss, J. *User Interface Development: Problems and Experiences*. Symposium on New Results and New Trends in Computer Science, Graz/Austra, 1991. [Žiūrėta 2006-04-03]. Prieiga per internetą <<http://www.informatik.fernuni-hagen.de/pi3/PDFs/Graz.pdf>>.

PRIEDAI

Testiniai duomenys

3 lentelė. Metamodelio lentelės „Duom_salt“ testiniai duomenys

| Ds_id | Pavadinimas | Piln_pavadinimas | Paaishkinimas | H_lygis |
|-------|-------------|------------------|--------------------|---------|
| 1 | Darb | Darbuotojas | Projektuotojas | |
| 2 | PVM_S | PVM Sąskaita | Užpildoma sąskaita | |

4 lentelė. Metamodelio lentelės „Atributas“ testiniai duomenys

| id | n_vardas | unikalumas | Paaishkinimas | Tipas | ilgis |
|----|----------------|------------|---------------------------------|----------|-------|
| 1 | Vardas | 0 | Vardas | String | |
| 2 | Pavarde | 1 | Pavarde | String | |
| 3 | AsmensKodas | 1 | Asmens kodas | String | |
| 4 | Lytis | 0 | Lytis | String | |
| 5 | Soc Draud Nr. | 1 | Socialinio draudimo pažymėjimas | number | |
| 6 | GimimoData | 1 | Gimimo data | datetime | |
| 7 | Login | 1 | Prisijungimo vardas | string | |
| 8 | Password | 1 | Slaptažodis | string | |
| 9 | Įdarbinimodata | 0 | Įdarbinimo data | datetime | |

5 lentelė. Metamodelio lentelės „DS_atributai“ testiniai duomenys

| id | Vard_org | rodomumas | butinumas | Ds_id |
|----|----------|-----------|-----------|-------|
| 1 | 2 | 1 | 1 | 3 |
| 2 | 1 | 1 | 1 | 5 |
| 3 | 3 | 1 | 1 | 4 |
| 4 | 4 | 1 | 0 | 6 |

6 lentelė. Metamodelio lentelės „UIE elementai“ testiniai duomenys

| Id | Elementas | x | y | Plotis |
|-----|------------|-------|-------|---|
| 680 | TextBox.17 | 70 mm | 76 mm | BOUND(50 mm;0;FALSE;User.TextWidth;1e100) |
| 681 | TextBox.6 | 70 mm | 70 mm | BOUND(50 mm;0;FALSE;User.TextWidth;1e100) |
| 682 | TextBox.10 | 70 mm | 64 mm | BOUND(50 mm;0;FALSE;User.TextWidth;1e100) |
| 683 | TextBox.14 | 70 mm | 58 mm | BOUND(50 mm;0;FALSE;User.TextWidth;1e100) |
| 684 | ComboBox | 70 mm | 52 mm | BOUND(50 mm;0;FALSE;User.TextWidth;1e100) |
| 685 | TextBox.28 | 70 mm | 46 mm | BOUND(50 mm;0;FALSE;User.TextWidth;1e100) |
| 686 | TextBox.32 | 70 mm | 40 mm | BOUND(50 mm;0;FALSE;User.TextWidth;1e100) |
| 687 | TextBox.36 | 70 mm | 34 mm | BOUND(50 mm;0;FALSE;User.TextWidth;1e100) |
| 688 | TextBox.40 | 70 mm | 28 mm | BOUND(50 mm;0;FALSE;User.TextWidth;1e100) |

Duomenų bazės modelio lentelių aprašymas

| Vardas | Duomenų tipas | Pirminis raktas | Būtinas | Unikalus | Paaishkinimas |
|------------------|---------------|-----------------|---------|----------|--|
| Atributas | | | | | Lentelėje saugomi visi nagrinėjamo organizacijos veiklos konteksto atributai. |
| id | int | + | + | + | Unikalus įrašo id, kuri sugeneruoja DBVS. |
| kodas | varchar 10 | | + | + | Unikalus atributą apibūdinantis kodas, kodą sudaro sistemos vartotojas (analitikas/projektuotojas). |
| n_vardas | varchar 50 | | + | | Atributo vardas, kuris gali būti paimtas iš nagrinėjamo veiklos konteksto arba sukurtas paties vartotojo. |
| unikalumas | bool | | + | | Požymis rodantis ar atributo reikšmė turi būti unikali. TRUE – reikšmė turi būti unikali, o FALSE - kad reikšmės unikalumas nėra būtinas. |
| paaishkinimas | varchar 200 | | | | Detali informacija apie atributą. |
| tipas | varchar 30 | | | | Atributo duomenų tipas. |
| e_id | int | | | | Esybės, kuriai priklauso atributas id. |
| Esybe | | | | | Lentelėje saugomos visos nagrinėjamo organizacijos veiklos konteksto esybės. |
| id | int | + | + | + | Unikalus įrašo id, kurį sugeneruoja DBVS. |
| kodas | varchar 10 | | + | + | Unikalus esybę apibūdinantis kodas, kodą sudaro sistemos vartotojas. |
| n_vardas | varchar 30 | | + | | Esybės vardas, kuris gali būti paimtas iš nagrinėjamo veiklos konteksto arba sukurtas paties vartotojo. |
| paaishkinimas | varchar 200 | | | | Detalizuota informacija apie esybę. |
| Rysys | | | | | Lentelėje saugomi visi nagrinėjamo organizacijos veiklos konteksto ryšiai tarp esybių. |
| id | int | + | + | + | Unikalus įrašo id, kuri sugeneruoja DBVS. |
| kard1 | varchar 3 | | + | | 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. |
| kard2 | varchar 3 | | + | | Ryšio kardinalumas antrame ryšio gale. Kardinalumo aprašymo struktūra: <būtinumas>,<kardinalumas>. Būtinumas gali |

| Vardas | Duomenų tipas | Pirminis raktas | Būtinis | Unikalus | Paaiškinimas |
|--------------------|---------------|-----------------|---------|----------|--|
| | | | | | įgyti šias reikšmes: 0-nebūtinis, 1- būtinis. Kardinalumas gali įgyti šias reikšmes: 1 – vienas, x – daug. |
| e_id1 | int | | + | | Esybių, kurias jungia ryšys id. |
| e_id2 | int | | + | | |
| rys_ds_id | int | | | | |
| Rezultatas | | | | | Lentelėje saugomi informacija apie KIS funkcionalumo metu formuojamus rezultatus arba žodžiu perduodamus informacijos srautus. |
| id | int | + | + | + | Unikalus įrašo id, kuri sugeneruoja DBVS. |
| kodas | varchar 10 | | + | + | Unikalus įrašą apibudinantis kodas, kodą sudaro sistemos vartotojas. |
| pavadinimas | varchar 30 | | + | | Rezultato pavadinimas, kuris gali būti paimtas iš nagrinėjamo veiklos konteksto arba sukurtas paties vartotojo. |
| paaiskinimas | varchar 200 | | | | Detalizuota informacija apie rezultatą. |
| salyga | varchar 500 | | | | Užrašoma žodinė informacija apie sąlygas, kurias reikia tenkinti formuojant tam tikrą rezultatą. |
| R_atributas | | | | | Lentelėje saugomi informacija apie KIS funkcionalumo metu formuojamų rezultatų atributus. |
| id | int | + | + | + | Unikalus įrašo id, kuri sugeneruoja DBVS. |
| kodas | varchar 10 | | + | + | Unikalus įrašą apibudinantis kodas, kodą sudaro sistemos vartotojas. |
| vard_org | varchar 50 | | + | | Rezultato atributo pavadinimas, kuris paimtas iš nagrinėjamo veiklos konteksto, toks koks pateikiamas originaliame rezultate. |
| egz_sk | int | | + | | Konkreto atributo egzempliorių skaičius išvedamų į rezultatą. |
| rodomumas | bool | | + | | Ar atributas matomas tik specifikacijos modelyje ar ir originale. FALSE – originale ir modelyje, TRUE - tik modelyje. |
| butinumas | bool | | + | | Požymis rodantis ar atributo reikšmės būtinai turi būti pateikiama rezultate. TRUE – reikšmė turi būti pateikiama, o FALSE - kad reikšmė nebūtina. |
| formule | varchar 200 | | | | Jei atributo reikšmė yra skaičiuojama, išsaugoma formulė, pagal kuria gali būti skaičiuojam atributo reikšmė. |
| r_id | int | | + | | Rezultato id, kuriam priklauso atributas. |
| a_id | int | | + | | Sistemos atributo id, su kuriuo siejamas rezultato atributas, nes skirtinguose |

| Vardas | Duomenų tipas | Pirminis raktas | Būtinai | Unikalus | Paaiškinimas |
|-----------------------|---------------|-----------------|---------|----------|---|
| | | | | | rezultatuose ar duomenų šaltiniuose tas pats sistemos atributas gali būti skirtingai įvardintas. |
| Rib_reiksme_R | | | | | Lentelėje saugoma rezultate esančių atributų egzempliorių ribinių reikšmių informacija. |
| id | int | + | + | + | Unikalus įrašo id, kuri sugeneruoja DBVS. |
| kodas | varchar 10 | | + | + | Unikalus įrašą apibūdinantis kodas, kodą sudaro sistemos vartotojas. |
| vard_org | varchar 50 | | + | | Reikšmė, kuris paimtas iš nagrinėjamo veiklos konteksto. |
| n_vardas | varchar 30 | | + | | Reikšmė, kuria KIS siūlo naudoti vartotojas, tai gali būti ir ta pati reikšmė paimta iš nagrinėjamo veiklos konteksto |
| paaiskinimas | varchar 200 | | | | Detalizuota informacija apie reikšmę. |
| r_id | int | | + | | Rezultato id, kuriame naudojama ši reikšmė. |
| R_esybe | | | | | Lentelėje saugomi informacija apie KIS funkcionalumo metu formuojamų rezultatų išskirtas esybes. |
| id | int | + | + | + | Unikalus įrašo id, kuri sugeneruoja DBVS. |
| kodas | varchar 10 | | + | + | Unikalus įrašą apibūdinantis kodas, kodą sudaro sistemos vartotojas. |
| vard_org | varchar 50 | | + | | Esybės pavadinimas, kuris paimtas iš nagrinėjamo veiklos konteksto. |
| rodomumas | bool | | + | | Ar esybė matoma tik specifikacijos modelyje ar ir originale. FALSE – originale ir modelyje, TRUE - tik modelyje. |
| tipas | varchar 30 | | + | | |
| r_id | int | | + | | Rezultato id, kuriam priklauso esybė. |
| e_id | int | | + | | Sistemos esybės id, su kuria siejama rezultato esybė, nes skirtinguose rezultatuose ar duomenų šaltiniuose ta pati sistemos esybė gali būti skirtingai įvardinta. |
| Salygos_elem_R | | | | | Lentelėje saugoma informacija apie rezultato sudarymo sąlygos elementus. |
| id | int | + | + | + | Unikalus įrašo id, kuri sugeneruoja DBVS. |
| r_id | int | | + | | Rezultato, kuriam formuojam sąlyga id. |
| r_a_id1 | int | | + | | Rezultato atributo id, kuris įtakojamas sąlygos. |
| r_a_id2 | int | | | | Rezultato atributo id, kuris įtakoja rezultato atributą. |
| rb_id | int | | | | Rezultato ribinės reikšmės id, kuri įtakoja rezultato atributą. |

| Vardas | Duomenų tipas | Pirminis raktas | Būtinasis | Unikalus | Paaiškinimas |
|---------------------|---------------|-----------------|-----------|----------|--|
| Duom_salt | | | | | Lentelėje saugomi informacija apie organizacijos objektus (duomenų šaltinius) saugančius duomenis, reikalingus funkcijoms įvykdyti. |
| id | int | + | + | + | Unikalus įrašo id, kuri sugeneruoja DBVS. |
| kodas | varchar 10 | | + | + | Unikalus įrašą apibūdinantis kodas, kodą sudaro sistemos vartotojas. |
| pavadinimas | varchar 30 | | + | | Duomenų šaltinio pavadinimas, kuris gali būti paimtas iš nagrinėjamo veiklos konteksto arba sukurtas paties vartotojo. |
| paaiškinimas | varchar 200 | | | | Detalizuota informacija apie duomenų šaltinį. |
| salyga | varchar 500 | | | | Užrašoma žodinė informacija apie sąlygas ir logiką, kurias reikia tenkinti formuojant duomenų šaltinį. |
| h_lygis | int | | | | Duomenų šaltinio hierarchijos lygis. |
| DS_atributas | | | | | Lentelėje saugomi informacija apie organizacijos objektų (duomenų šaltinius) saugančių duomenis, reikalingus funkcijoms įvykdyti, atributus. |
| id | int | + | + | + | Unikalus įrašo id, kuri sugeneruoja DBVS. |
| kodas | varchar 10 | | + | + | Unikalus įrašą apibūdinantis kodas, kodą sudaro sistemos vartotojas. |
| vard_org | varchar 50 | | + | | Duomenų šaltinio atributo pavadinimas, kuris paimtas iš nagrinėjamo veiklos konteksto. |
| egz_sk | int | | + | | Konkreto atributo egzempliorių skaičius įvedamų į konkretų duomenų šaltinį. |
| rodumas | bool | | + | | Ar atributas matomas tik specifikacijos modelyje ar ir originale. FALSE – originale ir modelyje, TRUE - tik modelyje. |
| butinumas | bool | | + | | Požymis rodantis ar atributo reikšmės būtinai turi būti įvedama į duomenų šaltinį. TRUE – reikšmė turi būti pateikiama, o FALSE - kad reikšmė nebūtina. |
| formule | varchar 200 | | | | Jei atributo reikšmė yra skaičiuojama, išsaugoma formulė, pagal kuria gali būti skaičiuojam atributo reikšmė. |
| ds_id | int | | + | | Duomenų šaltinio id, kuriam priklauso atributas. |
| a_id | int | | + | | Sistemos atributo id, su kuriuo siejamas duomenų šaltinio atributas, nes skirtinguose rezultatuose ar duomenų šaltiniuose tas pats sistemos atributas gali būti skirtingai įvardintas. |

| Vardas | Duomenų tipas | Pirminis raktas | Būtinasis | Unikalus | Paaiškinimas |
|------------------------|---------------|-----------------|-----------|----------|--|
| Rib_reiksme_DS | | | | | Lentelėje saugoma duomenų šaltinyje esančių atributų egzempliorių ribinių reikšmių informacija. |
| id | int | + | + | + | Unikalus įrašo id, kuri sugeneruoja DBVS. |
| kodas | varchar 10 | | + | + | Unikalus įrašą apibudinantis kodas, kodą sudaro sistemos vartotojas. |
| vard_org | varchar 50 | | + | | Reikšmė, kuris paimtas iš nagrinėjamo veiklos konteksto. |
| n_vardas | varchar 30 | | + | | Reikšmė, kuria KIS siūlo naudoti vartotojas, tai gali būti ir ta pati reikšmė paimta iš nagrinėjamo veiklos konteksto |
| paaiskinimas | varchar 200 | | | | Detalizuota informacija apie reikšmę. |
| ds_id | id | | + | | duomenų šaltinio id, kuriame naudojama ši reikšmė. |
| DS_esybe | | | | | Lentelėje saugomi informacija apie esybes išskirtas iš organizacijos objektų saugančius duomenis, reikalingus funkcijoms įvykdyti. |
| id | int | + | + | + | Unikalus įrašo id, kuri sugeneruoja DBVS. |
| kodas | varchar 10 | | + | + | Unikalus įrašą apibudinantis kodas, kodą sudaro sistemos vartotojas. |
| vard_org | varchar 50 | | + | | Esybės pavadinimas, kuris paimtas iš nagrinėjamo veiklos konteksto. |
| rodomumas | bool | | + | | Ar esybė matoma tik specifikacijos modelyje ar ir originale. FALSE – originale ir modelyje, TRUE - tik modelyje. |
| tipas | varchar 30 | | + | | |
| ds_id | int | | + | | Duomenų šaltinio id, kuriam priklauso esybė. |
| e_id | int | | + | | Sistemos esybės id, su kuria siejama duomenų šaltinio esybė, nes skirtinguose rezultatuose ar duomenų šaltiniuose ta pati sistemos esybė gali būti skirtingai įvardinta. |
| Salygos_elem_DS | | | | | Lentelėje saugoma informacija apie duomenų šaltinio sudarymo sąlygos elementus. |
| id | int | + | + | + | Unikalus įrašo id, kuri sugeneruoja DBVS. |
| ds_id | int | | + | | Duomenų šaltinio, kuriam formuojam sąlyga id. |
| ds_a_id1 | int | | + | | Duomenų šaltinio atributo id, kuris įtakojamas sąlygos. |
| ds_a_id2 | int | | | | Duomenų šaltinio atributo id, kuris įtakoja duomenų šaltinio atributą. |
| rb_id | int | | | | Duomenų šaltinio ribinės reikšmės id, kuri |

| Vardas | Duomenų tipas | Pirminis raktas | Būtinas | Unikalus | Paaiškinimas |
|---------------------|---------------|-----------------|---------|----------|--|
| | | | | | įtakoja rezultato atributą. |
| Et_atributas | | | | | Lentelėje saugoma informacija apie duomenų šaltinių apdorojimo etapo metu įvedamus arba modifikuojamus atributus. |
| id | int | + | + | + | Unikalus įrašo id, kuri sugeneruoja DBVS. |
| ds_a_id | int | | + | | Duomenų šaltinio atributo id, kuris to etapo metu apdorojamas. |
| et_id | int | | + | | Etapo id. |
| Etapas | | | | | Lentelėje saugoma informacija apie duomenų šaltinio apdorojimo etapus. |
| id | int | + | + | + | Unikalus įrašo id, kuri sugeneruoja DBVS. |
| kodas | varchar 10 | | + | + | Unikalus įrašą apibudinantis kodas, kodą sudaro sistemos vartotojas. |
| pavadinimas | varchar 30 | | + | | Vardas, kuris gali būti paimtas iš nagrinėjamo veiklos konteksto arba sukurtas paties vartotojo. |
| paaiskinimas | varchar 200 | | | | Detali informacija apie etapą. |
| ds_id | int | | + | | Duomenų šaltinio id, kuriam priklauso etapas. |
| Perėjimas_Et | | | | | Lentelėje saugoma informacija apie perėjimus tarp duomenų šaltinio apdorojimo etapų. |
| id | int | + | + | + | Unikalus įrašo id, kuri sugeneruoja DBVS. |
| v_id | int | | + | + | Veiksmo id, kurį vykdant atliekamas perėjimas. |
| et_is_id | int | | | | Etapo id iš kurio įvyksta perėjimas |
| et_i_id | int | | | | Etapo id į kurį įvyksta perėjimas |
| Veiksmas_DS | | | | | Lentelėje saugoma informacija apie visus galimus duomenų šaltinių apdorojimo veiksmus. |
| id | int | + | + | + | Unikalus įrašo id, kuri sugeneruoja DBVS. |
| kodas | varchar 10 | | + | + | Unikalus įrašą apibudinantis kodas, kodą sudaro sistemos vartotojas. |
| pavadinimas | varchar 30 | | + | | Vardas, kuris gali būti paimtas iš nagrinėjamo veiklos konteksto arba sukurtas paties vartotojo. |
| paaiskinimas | varchar 200 | | | | Detali informacija apie veiksmą. |
| ak_id1 | int | | + | | Aktorius id, kuris yra agentas t.y. atlieka veiksmą arba jį inicijuoja. |
| ak_id2 | int | | + | | Aktorius id, kuris yra veiksmo rezultatų (srauto) gavėjas. |

| Vardas | Duomenų tipas | Pirminis raktas | Būtinis | Unikalus | Paiškinimas |
|---------------------|---------------|-----------------|---------|----------|---|
| Aktorius | | | | | Lentelėje saugoma informacija apie visus aktorius esančius nagrinėjamos veiklos kontekste. |
| id | int | + | + | + | Unikalus įrašo id, kuri sugeneruoja DBVS. |
| kodas | varchar 10 | | + | + | Unikalus įrašą apibudinantis kodas, kodą sudaro sistemos vartotojas. |
| pavadinimas | varchar 30 | | + | | Vardas, kuris gali būti paimtas iš nagrinėjamo veiklos konteksto arba sukurtas paties vartotojo. |
| paiškinimas | varchar 200 | | | | Detali informacija apie aktorį. |
| Elementas | | | | | Lentelėje saugomi informacija apie vartotojo sąsajos objektus (elementus) saugančius duomenis, reikalingus vartotojo sąsajai modeliuoti. |
| id | int | + | + | + | Unikalus įrašo id, kuri sugeneruoja DBVS. |
| kodas | varchar 10 | | + | + | Unikalus įrašą apibudinantis kodas, kodą sudaro sistemos vartotojas. |
| pavadinimas | varchar 30 | | + | | Vartotojo sąsajos elemento pavadinimas. |
| paiškinimas | varchar 200 | | | | Detalizuota informacija apie elementą. |
| salyga | varchar 500 | | | | Užrašoma žodinė informacija apie sąlygas ir logiką, kurias reikia tenkinti pasirenkant elementą. |
| | | | | | |
| EL_atributas | | | | | Lentelėje saugomi informacija apie organizacijos objektų (duomenų šaltinius) saugančių duomenis, reikalingus funkcijoms įvykdyti, atributus. |
| id | int | + | + | + | Unikalus įrašo id, kuri sugeneruoja DBVS. |
| kodas | varchar 10 | | + | + | Unikalus įrašą apibudinantis kodas, kodą sudaro sistemos vartotojas. |
| tipas | varchar 50 | | + | | Elemento tipas . |
| egz_sk | int | | + | | Konkreto atributo egzempliorių skaičius įvedamų į konkretų duomenų šaltinį. |
| rodomumas | bool | | + | | Ar atributas matomas tik specifikacijos modelyje ar ir |

| Vardas | Duomenų tipas | Pirminis raktas | Būtinis | Unikalus | Paiškinimas |
|--------------------|---------------|-----------------|---------|----------|---|
| | | | | | originale. FALSE - originale ir modelyje, TRUE - tik modelyje. |
| butinumas | bool | | + | | Požymis rodantis ar atributo reikšmės būtinai turi būti įvedama į duomenų šaltinį. TRUE - reikšmė turi būti pateikiama, o FALSE - kad reikšmė nebūtina. |
| el_id | int | | + | | Elemento id, kuriam priklauso atributas. |
| Veiksmas_EL | | | | | Lentelėje saugoma informacija apie visus galimus duomenų šaltinių apdorojimo veiksmus. |
| id | int | + | + | + | Unikalus įrašo id, kuri sugeneruoja DBVS. |
| kodas | varchar 10 | | + | + | Unikalus įrašą apibudinantis kodas, kodą sudaro sistemos vartotojas. |
| Eil_nr | varchar 30 | | + | | Numeris, rodantis kurioje sąsajos vietoje yra elementas. |
| paaishkinimas | varchar 200 | | | | Detali informacija apie veiksmą. |
| el_id1 | int | | + | | Elemento id, su kuriuo susiejamas. |

Publikuoti straipsniai