



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

Informacijos sistemų katedra

Automatizuotas informacinės sistemos vartotojo sąsajos projektavimas

Magistro darbas

Darbo vadovė:

dr. doc. Rita Butkienė

KAUNAS, 2007

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

Informacijos sistemų katedra

Katedros vedėjas

doc. dr. R. Butleris

2007-01-09

Automatizuotas informacinės sistemos vartotojo sąsajos projektavimas

Magistro darbas

Recenzentas

doc. dr. E. Karčiauskas

2007-01-09

Vadovė

doc. dr. R. Butkienė

2007-01-09

Atliko

IFM-1/4 gr. stud.

A. Kupis

2007-01-09

KAUNAS, 2007

Turinys

1 Įvadas.....	4
2 Analitinė dalis.....	5
2.1 Tyrimo sritis.....	5
2.2 Analizės metodų, priemonių parinkimas.....	6
2.3 Grafinė vartotojo sąsajos kūrimo proceso analizė.....	6
2.3.1 Veiklos tikslų modelis.....	7
2.3.2 Veiklos objektų modelis.....	8
2.4 Grafinės sąsajos elementų naudojimas.....	10
2.5 Grafinės sąsajos projektavimo principai.....	13
2.6 Sąsajos projektavimo metodai.....	14
2.7 CASE įrankiai grafinės sąsajos kūrimui.....	15
2.8 Projekto tikslas.....	20
2.9 Kompiuterizuojamos sistemos varianto pasirinkimas.....	21
2.10 Analizės išvados.....	22
3 Reikalavimai GUI automtizuoto įrankio kūrimui.....	23
3.1 GUI automatizuoto įrankio reikalavimų specifikacija.....	23
3.2 Dalykinės srities modelis.....	28
4 Sąsajos generavimo įrankio projektas.....	30
4.1 Sprendimo pagrindimas.....	30
4.2 Sistemos architektūra.....	30
4.2.1 Loginė sistemos architektūra.....	30
4.2.2 Vartotojo, veiklos, duomenų paslaugų klasių diagramos.....	31
4.3 Detalus projektas.....	33
4.4 Sistemos elgsenos modelis.....	35
4.4.1 Sistemos veiklos modelis.....	35
4.4.2 Sekų diagramos.....	36
4.4.3 Navigacijos planas.....	39
4.5 Duomenų bazės schema.....	40
4.6 Testavimo modelis, pavyzdiniai duomenys.....	42
5 Eksperimentinis tyrimas.....	48
6 Darbo rezultatai.....	49
7 Išvados.....	50
8 Literatūra.....	51

1 Įvadas

Suprojektuoti grafinę vartotojo sąsają yra vienas iš pagrindinių uždavinių projektuojant kompiuterizuotą informacinę sistemą. Grafinei vartotojo sąsajai suprojektuoti taikomi įvairūs principai, metodikos, įrankiai. Bet esminę įtaką grafinės vartotojo sąsajos struktūrai turi vartotojų reikalavimai kuriamos sistemos funkcionalumui.

Informacijos sistemų katedroje plėtojamas CASE įrankis, kurio pagalba vartotojo funkciniai reikalavimai kompiuterizuotai informacinei sistemai specifikuojami ODRES (Output Driven REquirements Specification) metodu. Šiuo metodu specifikuoti reikalavimai yra sustruktūrinti, o CASE įrankis leidžia juos išsaugoti reikalavimų saugykloje. Reikalavimų saugykloje sukauptos informacijos panaudojimas būtų efektyvesnis, jei ji automatiniu arba automatizuotu būdu būtų panaudojama kituose sistemos kūrimo etapuose (pavyzdžiui, projektavime). Vienas iš tokių panaudojimo būdų būtų – automatinis grafinės vartotojo sąsajos projekto sukūrimas iš saugykloje sukauptos informacijos.

Šio darbo tikslas yra sukurti įrankį, kuris:

- automatiškai sukurtų grafinės vartotojo sąsajos projektą iš ODRES metodu specifikuotų ir CASE įrankio saugykloje saugomų reikalavimų,
- automatizuotų automatiškai sukurto grafinės vartotojo sąsajos projekto tobulinimo procesą,
- išsaugotų sukurtą grafinės sąsajos projektą formoje, leidžiančioje automatiškai realizuoti suprojektuotą sąsają.

Įrankis, grafinei vartotojo sąsajai projektuoti, realizuotas Microsoft Visio 2003 paketu, naudojant Visual Basic for Applications programavimo kalbą. Programiškai valdomi Visio įrankiai – formų, įvedimo ir išvedimų laukų grafiniai simboliai.

Darbo aprašo antrajame skyriuje apibrėžtas tyrimo objektas, pateiktas reikalavimų specifیکavimo metodų ir priemonių pasirinkimo analizė. Išanalizavus tyrimo sritį, suformuluotas projekto tikslas. Trečiajame skyriuje pateiktas automatizuoto grafinės sąsajos kūrimo įrankio reikalavimai, iš kurių seka ketvirtajame skyriuje plėtojamas šio įrankio projektas.. Penktajame skyriuje pateikiamas sistemos eksperimentas.

2 Analitinė dalis

2.1 Tyrimo sritis

Projektuojant kompiuterizuotą informacinę sistemą vienas iš uždavinių yra suprojektuoti grafinę vartotojo sąsają (*GUI - graphical user interface*).

Šiame darbe bus nagrinėjamas kompiuterizuotos informacinės sistemos grafinės sąsajos projektavimo procesas. Grafinėi sąsajai suprojektuoti taikomi įvairios metodikos, principai, bet pagrindinis dėmesys bus skiriamas įrankiams, automatizuojantiems grafinės sąsajos projektavimo procesą. Taip pat bus nagrinėjamas KTU Informacijos sistemų katedroje plėtojamas CASE įrankis, skirtas specifikuoti vartotojo reikalavimus kompiuterizuotai informacinei sistemai ODRES metodu.[1]

Sąsajos kūrimo procesas užima nemažą viso projekto kūrimo dalį. Siekiant kuo labiau sutrumpinti kūrimo laiką pasitelkiami įvairūs įrankiai gebantys sąsają generuoti automatiškai. Tarpiniuose sistemos projektavimo etapuose, generuojant sąsają, galima stebėti projektą, ar jis atitinka vartotojo nurodytus reikalavimus. Vieni metodai sąsajos prototipą sugeneruoja visiškai automatiškai, kiti tai daro dialogų pagalba. Skirtingi metodai naudoja skirtingus duomenų šaltinius – vieni sąsają generuoja iš UML diagramų, kiti iš duomenų bazės lentelių pavadinimų, arba metaduomenų ir pan.

Automatizuoti grafinės sąsajos kūrimo įrankiai atlieka pagalbinę funkciją programų inžinieriams, transformuodamas grafinėi sąsajai keliamus reikalavimus į ekrane išdėstomus grafinius elementus. Šis įrankis padeda spręsti tiek paprastas, tiek sudėtingesnes užduotis. Kiekvienam generavimo atvejui taikomos apibendrintos taisyklės, kurios ne visada tinka specifiniams uždaviniams. Dėl šios priežasties tokio įrankio kūrimas tampa sudėtingu. Generuojant sąsają, ekrane pateikiami visi įvedimo ar išvedimo laukai, aprašyti susietoje bazėje. Dažniausiai visų elementų vaizduoti nereikia, jei sistema naudojami keletas savos srities specialistų.

Pagrindinis uždavinys norint sukurti įrankį, leidžiantį automatizuoti vartotojo sąsajos generavimą – tinkamai suformuluoti reikalavimus sąsajai. Vartotojo reikalavimai kuriamai programų sistemos sąsajai nurodomi programinės įrangos reikalavimų specifikacijoje. Jie apibrėžia sistemos funkcionalumą bei pradinius duomenų šaltinius. Reikalavimai išgaunami tiesiogiai bendraujant su sistemos galutiniais vartotojais (savos srities žinovais). Tik teisingai suformulavus reikalavimus keliamus GUI, galimas jos projektavimas.

Pasinaudojant iškeltais vartotojo sąsajos reikalavimais, CASE įrankis turi pasiūlyti kuo tikslesnę galutiniam variantui sąsajos versiją, su tinkamais elementų išdėstymais, langais bei

jų struktūromis. Reikalavimų specifikavimui skirtas įrankis, kuris automatizuoja Informacijos sistemų katedroje sukurtą ODRES metodą, leidžia sukaupti pakankamai duomenų, leidžiančių automatiškai sukurti grafinės sąsajos projektą.

2.2 Analizės metodų, priemonių parinkimas

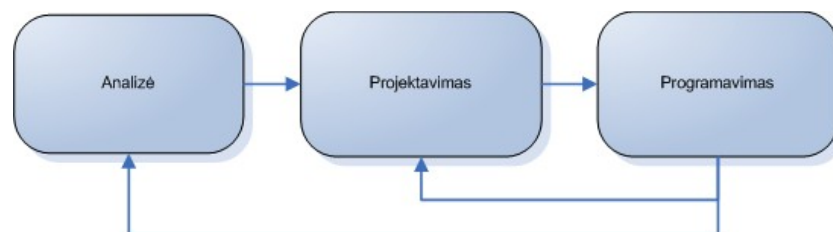
Srities vartotojų reikalavimų analizė ir sistemos projektavimas atliekamas naudojant RUP metodą, kuris remiasi UML. Ši specifikavimo kalba pasižymi dideliu išraiškingumu bei notacijos įvairove, tuo užtikrindama galimybes aprašyti įvairiausias dalykinės srities charakteristikas bei vartotojo reikalavimų niuansus. UML naudojamas klasių modelis suorientuoja sistemos specifikaciją objektiniu aspektu.

Tačiau naudojant UML, vartotojo funkcinis reikalavimus betarpiškai tenkinantys rezultatai – formos, ataskaitos, meniu, kurie dažniausiai išreiškiami per sąsajos elementus, nėra pilnai specifikuojami, o specifikacijos forma neadekvati vartotojo turimam jų įvaizdžiui.

UML notacija realizuota viename iš labiausiai paplitusių ir palaikomų programinės įrangos kūrimo metodų – vieningo Rational proceso (RUP). RUP pagrįstas užduotimis, reikalavimai išreiškiami per užduotis. Kuriant sistemą svarbu sukonstruoti patikimą architektūrą. Kokybiška architektūra yra kokybiškos sistemos pagrindas. Projektas suskaldomas į smulkesnes dalis (sub-projektus), kurios taip pat laikomos projektais. Kiekvienos dalelės įgyvendinimas prisideda prie viso projekto įgyvendinimo

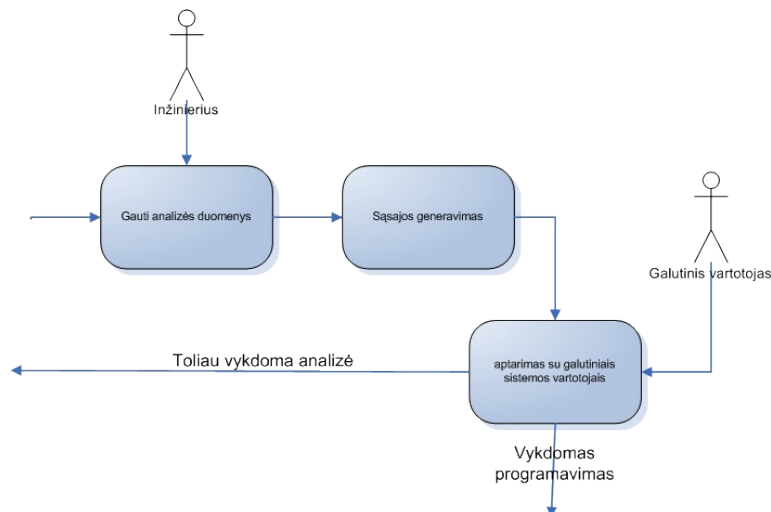
2.3 Grafinė vartotojo sąsajos kūrimo proceso analizė

Informacinė sistema kuriama keliais etapais (1-as pav.). Pirmiausia suformuluojami reikalavimai kuriamai sistemai, po to projektuojama sistemos architektūra ir programinės įrangos architektūra, kuri vėliau kiekviename etape detalizuojama.[2]



Pav. 1 IS kūrimo etapai

Projektavimo etape vykdomas grafinės sąsajos projektavimas. (2-as pav.)



Pav. 2 GUI kūrimo procesas

Grafinė vartotojo sąsaja įdiegta kiekvienoje modernioje operacinėje sistemoje – Microsoft Windows, Linux (su atskirai pasirenkama KDE, GNOME), MacOS. Lyginant su komandinės eilutės sąsaja, pradėjus naudoti grafinę sąsają milžiniškai padidėjo programų efektyvumas, bet nežymiai sumažėjo funkcionalumas (kai programą galima valdyti komandomis ir komandų parametrais, kurių visų nereikia atvaizduoti ekrane)

Grafinė vartotojo sąsaja – tai priemonė, padedanti programai vizualiai bendrauti su vartotoju. GUI suteikia programai išskirtinę išvaizdą bei funkcionalumą ir grafinių elementų (mygtukų, įvedimo laukų ir pan.) pagalba, atvaizduoja tas funkcijas. Tačiau ne kiekvienam specifikacijos metu analizuojamame duomenų šaltinyje naudojamam elementui būtinas susietas grafinis elementas. Ne kiekvienai funkcijai būtina, o kartais ir negalima sukurti konkretų kontrolės elementą.

Visa GUI sudaroma iš atskirų komponentų – objektų, kuriais vartotojas bendrauja pelės ar klaviatūros pagalba

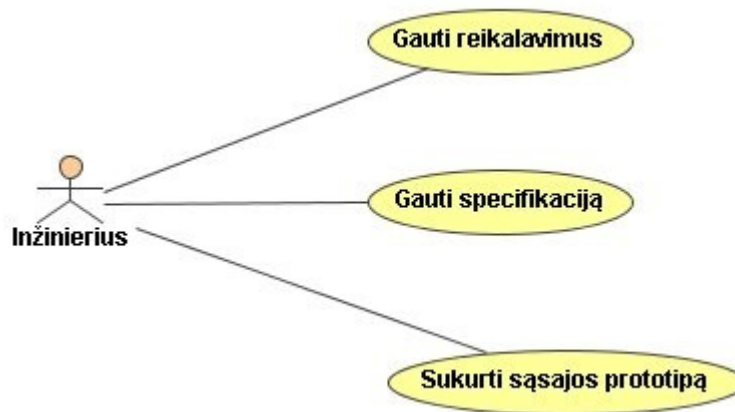
Pagal susiformavusį standartinį lango šabloną visos sudėtingesnės programos naudojami šiais komponentais – viršutinė meniu juosta (*menu bar*), mygtukų meniu juosta (*toolbar menu*) padedanti greičiau pasiekti komandas iš viršutinės juostos bei būsenos juosta (*status bar*)

2.3.1 Veiklos tikslų modelis

Pagrindinis inžinieriaus veiklos tikslas yra sukurti informacinės sistemos vartotojo sąsają. Automatizuotas sąsajos generavimas išreiškiamas šiais panaudojimo atvejais (2 pav.):

- Gauti reikalavimus
- Gauti specifikaciją

- Sukurti sąsajos prototipą



Pav. 3 Reikalavimų inžinieriaus veiklos panaudojimo atvejai

2.3.2 Veiklos objektų modelis

Skiriami du bendravimo su grafinais elementais būdai:

- Objektas – Veiksmas: pirmiausia pasirenkamas objektas su kuriuo bus atliekamas veiksmas (pažymėti įvedimo eilutę ir tik po to ją nukopijuoti)
- Veiksmas – Objektas: pirmiausia pasirenkamas veiksmas kuris bus atliktas su vėliau parenkamu objektu (komandinės eilutės atveju pirmiausia pasirenkamas veiksmas kopijuoti, po to objektas – ką kopijuoti) [4]

Dėl grafinės vartotojo sąsajos prigimties, tai kad ji sudaryta iš atskirų objektų, vartotojas su sistema bendrauja pirmuoju būdu, ir kiekvienas elementas atlieka jam skirtą veiksmą.

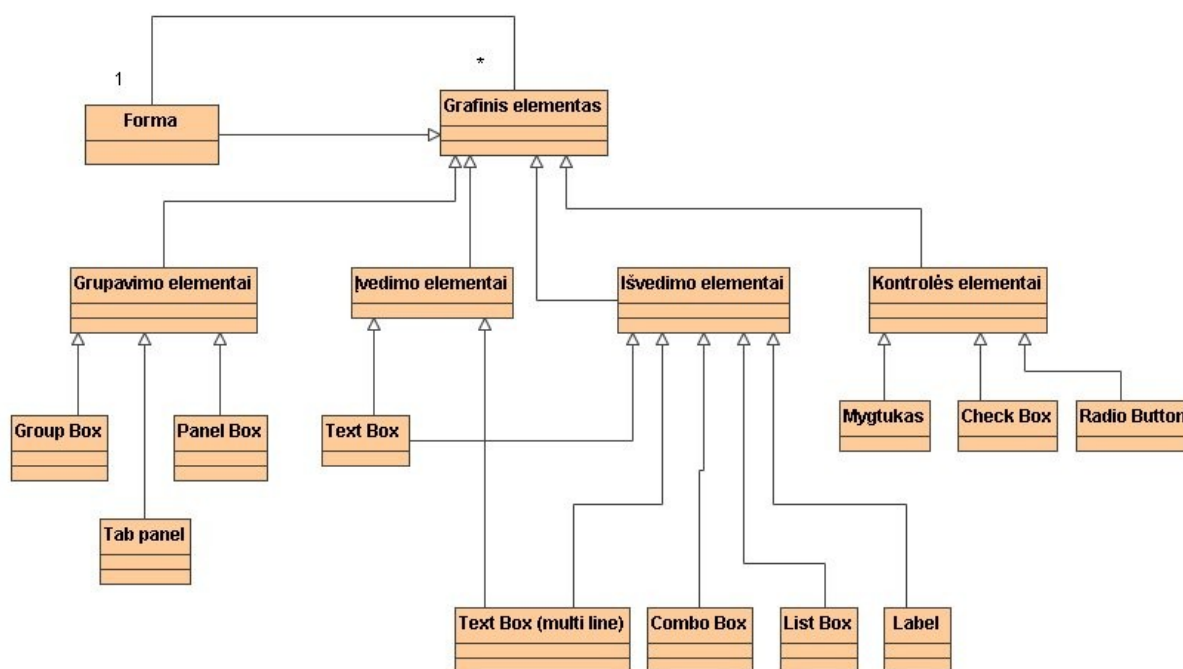
Elementas – tai neatskiriama vartotojo sąsajos dalis kuri neturi sub-elementų. Skiriami pagrindiniai elementai[4]:

- Duomenys (*data*) – elementas kuris vizualiai atvaizduoja savąją reikšmę
- Grafika (*graphic*) – elementai – paveikslukai
- Objektai (*object*) – elementas turintis su savimi susietą funkciją kurios vizualiai neišreiškia
- Piktogramos, simboliai (*icon*) – objektas atvaizduojamas grafiškai, dažnai susietas su aprašymu
- Kontroliniai elementai (*control*) – elementai kurių elgsena iš karto numatyta (open document, save document mygtukai), arba kuriems programiškai priskiriamos atliekamos funkcijos.
- Pasirinkimo (*choise*)

- Kombinuoti (*composite*) – tai pavyzdžiui vaizdo kontrolės elementai, vienodos elgsenos elementai
- Darbo erdvės (*workspace*) – tai talpykla, kurioje talpinami kiti elementai – langai, pranešimo žinutės, piktogramos.
- Langai (*window*) – elementas esantis darbo erdvėje, atliekantis dialogo bendravimo funkciją su vartotoju.

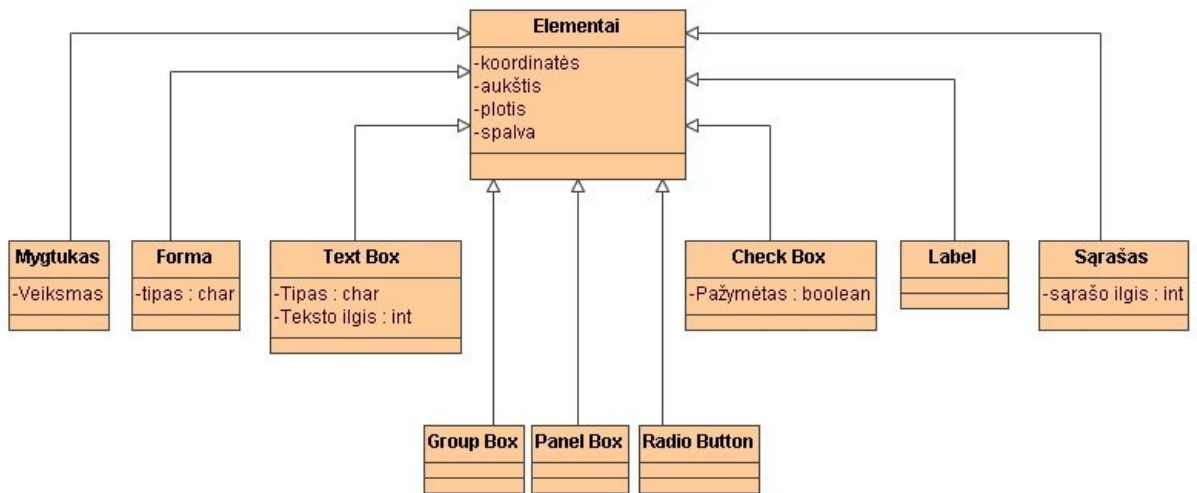
Plačiau apie kontrolės elementus. Šie elementai pagal savo atliekamas funkcijas grupuojami taip (3 pav.):

- Reikšmių nustatymo,
- Interaktyvi kontrolė
- Duomenų įvedimo, vaizdavimo
- Formų valdomi elementai
- Grupavimo
- Sąrašų sudarymo



Pav. 4 Elementų grupavimas pagal tipus

Kiekvienas elementas turi jį apibūdinančius atributus: aukštį, plotį, padėties koordinates, ir panašiai. Pavyzdžiui teksto įvedimo lauką apibūdina jo koordinatės, nurodomos reikšmėmis x ir y nuo formos viršutinio kairiojo kampo, įvedimo lauko dydis aprašomas reikšmėmis ilgis ir plotis, pagal tipus jis gali būti slaptažodžiui įvesti (kai įvedamas tekstas pakeičiamas kitais simboliais, dažniausiai ,*'), vienos eilutės įvedimo laukas arba teksto (kelių eilučių) įvedimo laukas. Mažiau reikšmingas atributas – spalva. Pagal savo būklę įvedimo laukas gali būti redaguojamas arba tik skaitomas (*enabled/disabled*), matomas arba nematomas.



Pav. 5 Įvedimo lauką apibūdinantys parametrai

2.4 Grafinės sąsajos elementų naudojimas

Grafinė sąsaja sudaroma iš grafinių elementų. Formoje elementai išdėstomi pagal tam tikras taisykles. Apibendrintą elementų išdėstymo tvarką nusako Geštalo psichologija.

Geštalo psichologija - psichologijos kryptis, teigianti, kad žmogaus suvokimas ir mąstysena yra orientuoti į objektų vientisą suvokimą (*geštaltą*). Pagal šią teoriją, žmogus nesąmoningai apjungia visus suvokiamus dalykus į tam tikras grupes, papildydamas trūkstamas detales [5]. Apjungimas vyksta remiantis įvairiomis taisyklėmis:

- Artumas – elementai kurie naudojami arti vienas kito laikomi elementų grupe. Ši išdėstymo savybė naudojama elementų aprašymų ar objektų grupavimui ekrane.
- Panašumas – žmogaus smegenys panašius objektus grupuoja panašiai kaip arti esančius. Šis principas labiausiai tinkamas lentelių vaizdavimui.
- Paprastumas – objektai grupuojami į nesudėtingas figūras pagal jų simetriją, kampuoatumą ar glotnumą.

Simetrija – žmogaus smegenys grupuoja simetriškus objektus. Programose simetrija panaudojama pateikiant du grafinius duomenų vaizdus, taip supaprastinant skirtumą vieno nuo kito.

Lentelė 1 Elementų išdėstymo būdai

Grafinės sąsajos elementų išdėstymo būdas	Aprašymas	Naudojimo atvejai
Tinklinis išdėstymas	<p>Ekraną vaizdo padalinimas tinkleliu į keletą zonų padeda išlaikyti nuoseklų sąsajos stilių. Tai taip pat supaprastina patį dizainą ir kūrimo procesą. Tinklinio išdėstymo rezultatas, kad vartotojas gali nuspėti kurioje vietoje rasti atitinkami grafiniai elementai.</p> <p>Tinklelis dalina ekraną į mažesnius vienetus, ir artumo, panašumo, simetrijos, lygiavimo pagalba sukuriamas vientisas dizainas.</p>	<p>Tinklinis išdėstymas atskiria kontrolės elementus nuo duomenų įvedimo / išvedimo elementų. Metabazėje naudojamos esybės su savais atributais atsitinka vieną elementų grupę.</p>
Svarbių elementų paryškinimas	<p>Elementų dydis ar kontrastas pabrėžia elementų svarbą. Kontrastas – tai vizualus skirtumas tarp vieno elemento ir kitų ekrane esančių elementų. Tai gali būti elemento dydis, forma, spalva, padėtis ekrane.</p>	<p>Sistemos vartotojai gali nurodyti svarbesnius (arba būtinus užpildyti) nei kitus elementus. Tokių elementų išskyrimas vyksta sugeneravus sąsają.</p>
Grupavimas į funkcines zonas	<p>Grupavimas vykdomas grafiniais objektais – linijomis, stačiakampiais, spalva, tuščia erdve arba pagal Geštalo teoriją. Nesvarbu kokių stiliumi naudojama, zonų atskirimas turi būti lengvas. Artimai susiję elementai turi būti grupuojami kartu, pavyzdžiui banko duodamoje formoje sąskaitos numeris ir banko pavadinimas turėtų būti sugrupuoti, kitu atveju pildant formą, informaciją apie banką ir sąskaitą tektų peržiūrėti ne vieną kartą. Kitos funkcinės zonos turėtų būti skirtos duomenų navigacijai, mažiau reikšmingai informacijai, teminėms grupėms taip pat išskirta zona funkciniam interaktyviam bendravimui</p>	<p>Metabazėje vienas duomenų šaltinis gali turėti vieną arba daugiau esybių. Esybės su jai priklausančiais atributais sudaro vieną funkcinę zoną. Funkcinė zoną metabazėje atitinka lentelė Formos Dalis.</p>

<p>Kontroliniams elementams atskira zona</p>	<p>Projektuojant sąsają atskirą zoną reikia išskirti kontroliniams elementams. Valdymo ir pasirinkimo mygtukai turėtų būti patalpinti išskirtinėje ekrano vietoje. Pavyzdžiui interneto puslapiuose dauguma vartotojų navigacijos mygtukus tikisi rasti ekrano viršuje arba kairėje.</p> <p>Svarbiausia informacija tikėtina kad bus pateikiama ekrano centre. Be to labai svarbu išlaikyti vientisumą tarp formų ir elementams suteikti pastovias vietas.</p>	<p>Kontrolės elementai būtini langų valdymui (iškvietimui, uždarymui), duomenų valdymui (duomenų užkrovimas, ištrynimas), įrašų valdymui (pildant sąrašą – eilutės ištrynimas, naujos įterpimas, koregavimas)</p>
<p>Stilių nuoseklumas</p>	<p>Projektuojant gerą sąsają svarbu laikytis stilių nuoseklumo. Toje pačioje formoje panašūs mygtukai neturėtų turėti skirtingus stilius, pavyzdžiui toje pačioje formoje nepatartina naudoti stačiakampius mygtukus ir ovalius tuo pačiu metu. Tai nesuteikia estetiško vaizdo, bet ir blaško vartotoją</p>	<p>Pagrindinėje ir pagalbinėse formose išlaikomas tas pats šrifto stilius, elementų dydis.</p>

2.5 Grafinės sąsajos projektavimo principai

Gera sąsaja paremta kūrybinio proceso principais sukonzentruotais į vartotoją ir jo užduotis.[6]

Centre – vartotojas

Svarbus principas tai, kad vartotojas visada turi jausti kontrolę, o nesijausti kontroliuojamas. Šį principą galima išskirstyti smulkiau:

- Vartotojas, o ne kompiuteris ar programinė įranga, pradeda veiksmus. Galima automatizuoti užduotis, bet pirmiausia automatizavimą reikia įterpti tarp nuosekliai žmogaus vykdomų komandų.
- Dėl skirtingų vartotojų gabumų ir poreikių, patys vartotojai turi turėti galimybę prisitaikyti sąsają savo poreikiams.
- Sąsaja turi būti kuo jautresnė ir intuityvesnė. Kuo mažiau naudoti režimus. Režimas tai tokia būklė, į kurią patenkama atlikus kokią nors funkciją, ir pilna sistemos kontrolė atgaunama išėjus iš šios būklės.

Atvirumas

Sąsajos vartotojai turi jausti galimybę tiesiogiai manipuluoti programos atvaizduojama informacija.. Ar tai būtų kokio nors objekto (pvz. teksto) iškirpimas ar pertempimas dokumente, vartotojai ekrane turėtų matyti visus objektų pokyčius. Matoma informacija ir pasirinkimai palengvina vartotojo užduotis.

Nuoseklumas

Nuoseklumas leidžia vartotojams pritaikyti žinomus metodus greitesniam naujų užduočių vykdymui. Visa tai vyksta todėl, jog nereikia skirti papildomai laiko norint įsiminti skirtingus su sąsaja bendravimo būdus. Kuriant sistemą stabilesnę, ji taip pat tampa labiau nuspėjama.

Nuoseklumas svarbus visais grafinės sąsajos kūrimo aspektais – komandų vardais, vaizdiniu informacijos atvaizdavimu, operatyviniu veikimu bei elementų išdėstymu ekrane ir languose:

- Dažnai pasikartojančios komandos sugrupuotos į savas grupeles.
- Kuriamos programinės įrangos sąsaja turėtų būti kuo panašesnė į operacinės sistemos sąsają.

Užmaršumas

Vartotojai mėgsta tyrinėti sąsają ją bandydami ir taip netyčia sukeldami klaidas. Gera sąsaja leidžia dialoginį tyrinėjimą. Ji pateikia tik tinkamus vartotojui pasirinkimus ir perspėja apie rizikingus veiksmus, kuriais būtų galima prarasti duomenis.

Net ir pati geriausia sąsaja neapsaugoma nuo klaidų. Tos klaidos netyčinės – susipainiojus kuris mygtukas kurią funkciją atlieka, ar tiesiog nejučiom paspaudus ant mygtuko.

Estetika

Vizualus dizainas svarbi sąsajos dalis. Vizualūs atributai kai kuriems objektams suteikia tikslesnę reikšmę, naudojant atitinkamai ženklus ar užuominas. Tuo pačiu, kiekvienas ekrane pasirodęs vizualusis elementas, traukia vartotojo dėmesį ir per didelis jų kiekis vartotoją tiesiog išblaško.

Paprastumas

Grafinė sąsaja turėtų būti paprasta, bet ne per daug supaprastinta, lengvai išmokstama ir lengvai naudojama. Ji taip pat turėtų pateikti visą programos funkcionalumą. Sudėtingiausias darbas kuriant grafinę sąsają, tai laviruoti tarp maksimalaus funkcionalumo ir aplinkos paprastumo.

Vienas būdas įgyvendinant paprastumo reikalavimą – tai vaizduojamos informacijos sumažinimas iki pakankamo veikimui minimumo (pvz. mygtukuose užtenka naudoti paveikslėlius vietoj žodinių jų aprašymų)

Kaip jau minėta nuoseklumo reikalavime, paprastumas siejasi su panašumu. Formos panašios į realias naudojamas kasdienėj raštvedyboj atrodo daug paprastesnėmis.

Pačių vartotojų įvairovė apsunkina tikslą sukurti kuo bendresnę sąsają, pavyzdžiui mygtukai su techniniais paveikslukais gali būti vienaip suprantami technikams ir visiškai skirtingai medikams.

2.6 Sąsajos projektavimo metodai

Yra ne vienas grafinės sąsajos projektavimo būdų, kiekvienas jų turi savų privalumų ir trūkumų. Pagrindiniai bruožai pasirenkant projektavimo būdą būtų:

- Laikas skirtas projektavimui

Prototipo kūrimo laikas užima nemažą viso projekto kūrimo dalį. Siekiama sumažinti laiko išlaidas, ir likusį laiką išnaudoti kitose projekto stadijose (testavimui, analizei).

- Išlaidos skirtos projektavimui

Kiekvienu prototipo kūrimo būdu naudojami skirtingi kaštai. Atitinkamai, kuo kaštai mažesni tuo labiau apribotos galimybės kuriant sudėtingas formas.

- Sudėtingumas

Kuriamo sąsajos prototipo sudėtingumas siejasi su išlaidomis skirtomis to prototipo kūrimui. Sudėtingomis sąsajomis įvardinama kompiuterinių žaidimų sąsajos ir pan.

Palyginimui pateikta 2-a lentelė su skirtingais projektavimo būdais:

Lentelė 2 Prototipo kūrimo būdai

Prototipo kūrimo būdas	Laiko sąnaudos	Išlaidos	Sudėtingumas
Popierinis prototipo kūrimas (Tinka labai ankstyvose projekto stadijose)	Užima daug laiko	Naudojamas tik popierius ir pieštukas	Sudėtinga sukurti komplikuotas formas, neįmanoma interaktyvumas
Kompiuterinis prototipo kūrimas:			Sudėtingas vartotojo ir projektuotojo bendravimas
Storyboard – kompiuterių pagalba piešiamos formos ir jos sujungiamos nuorodomis.	Greitas kūrimas ir koregavimas	Išlaidos susideda tik iš kompiuterinės ir programinės įrangos kainų	Tinkamas sudėtingiems projektams, tačiau nėra interaktyvių elementų
Interaktyvių vykdomųjų formų kūrimas	Greitas kūrimas ir koregavimas		Ribojama pateiktų elementų įvairovė. Netinkamas labai sudėtingoms formoms.

Efektyviam sąsajos projektavimui siekiama sumažinti laiko sąnaudas bei didinti funkcionalumą.

2.7 CASE įrankiai grafinės sąsajos kūrimui

Projektuojamai programai keliant funkcinius reikalavimus, pasitelkiami grafiniai įrankiai sistemos prototipo vaizdavimui grafiniais elementais. Taip padedama sistemos galutiniam vartotojui išivaizduoti kaip atrodys sukurta programinė įranga. Sistemų kūrėjai šiuos įrankius naudoja automatizuotam funkcinių reikalavimų transformavimui, ir/arba kodo generavimui, Šie CASE įrankiai gali būti naudojami kartu arba atskirai.

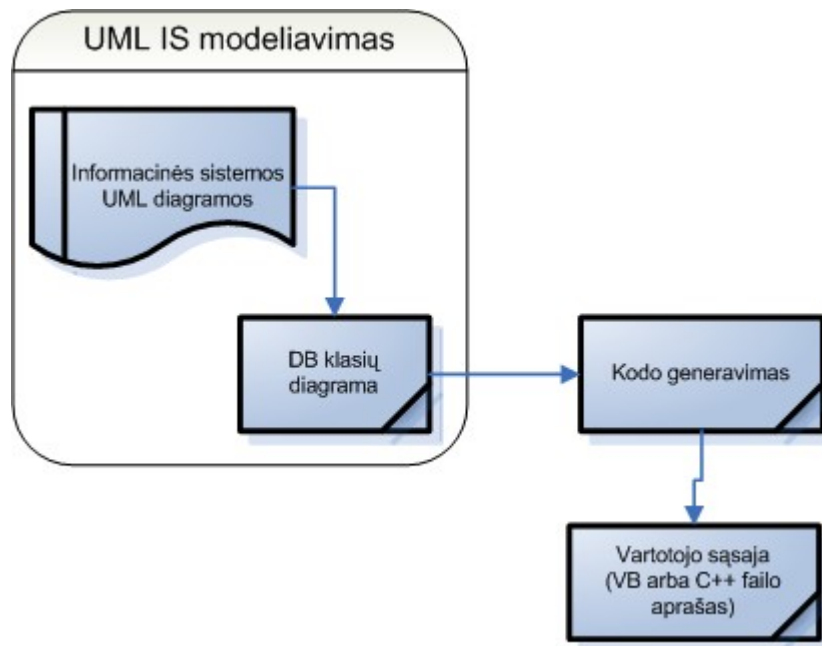
Egzistuoja nemažai programinės įrangos paketų įdiegusių langų (window) generatorių duomenų įvedimui, taip pat pavienių programų skirtų grafinės sąsajos kūrimui (naudojant kaip piešimo įrankį, ar automatiškai generuojant iš duomenų bazės (informacinės sistemos)).

Analizei pasirinkau tris pagrindinius įrankius formų generavimui, su kuriais buvau susipažinęs universitete:

- ProVision Workbench
- Microsoft Access
- Oracle Designer

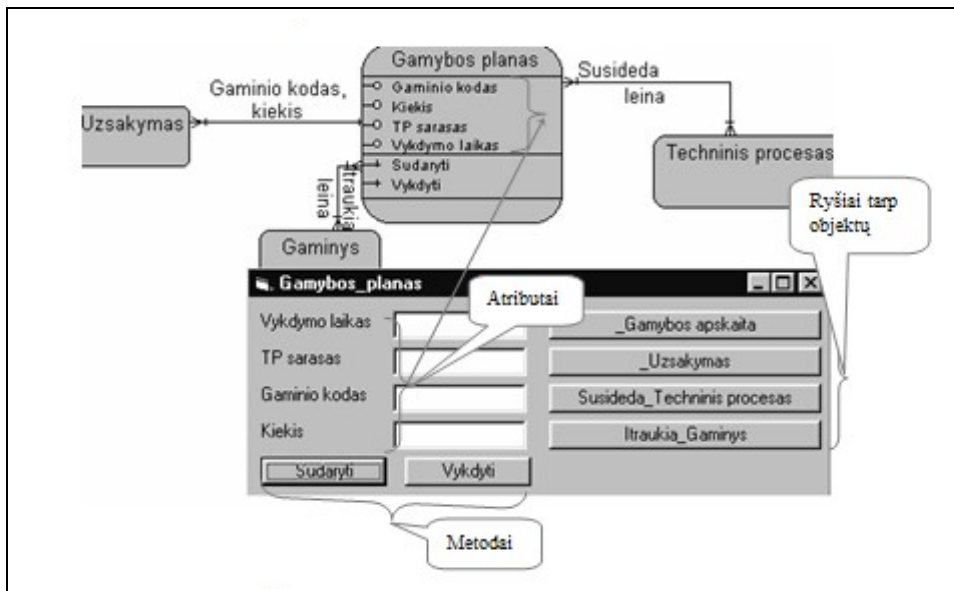
ProVision Workbench

ProVision Workbench – tai organizacijos modeliavimo priemonė, kuri apjungia Business Process Reengineering (BPR) ir Object-Oriented Analysis and Design (OO A&D) į vieną, integruotą modeliavimo priemonę



ProVision Workbench sąsajos generavimo schema

Paketas palaiko biznio modeliavimo galimybes, ypač unikalus biznio sąveikų modelis ir populiarūs darbų sekos ir biznio objektų modeliai. Tai leidžia apimti BPR ir biznio objektų modeliavimo etapus. Palaikant reliacinių lentelių modelį, vartotojo sąsajos ir kitų UML modelių kūrimą, paketas apima visą informacinės sistemos modeliavimo etapą. Taip pat dėka galimybės generuoti programinius kodus, paketas patenka ir į realizacijos etapo dalį.



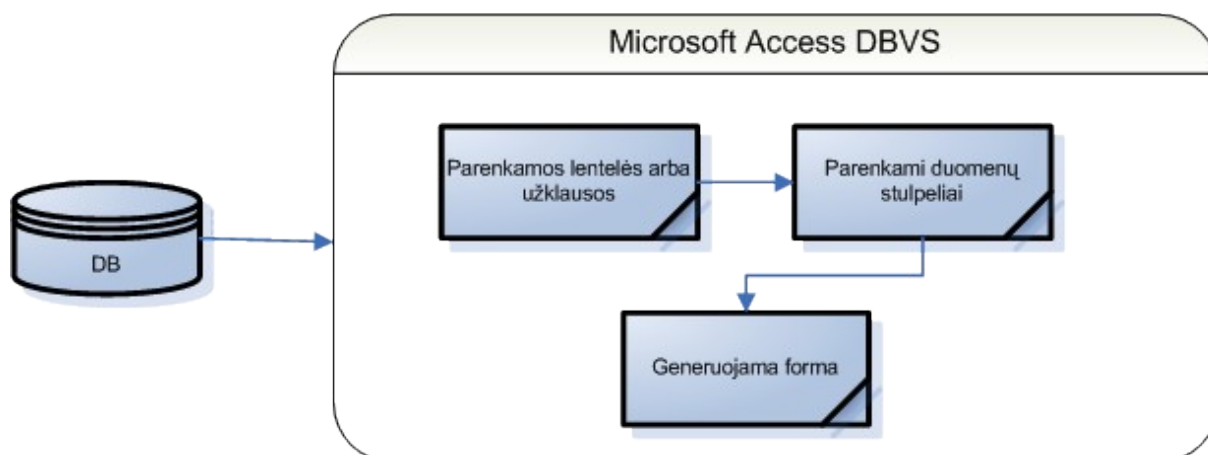
Pav. 6 ProVision Workbench aplinka

Pagrindinis šaltinis generuojamiems vartotojo sąsajos langų prototipams – biznio objektų modelyje apibrėžiami objektai, jų metodai, atributai bei ryšiai tarp objektų. Vartotojo sąsaja kuriama iš UML modelių, bet ne iš konkrečios užpildytos DBVS.

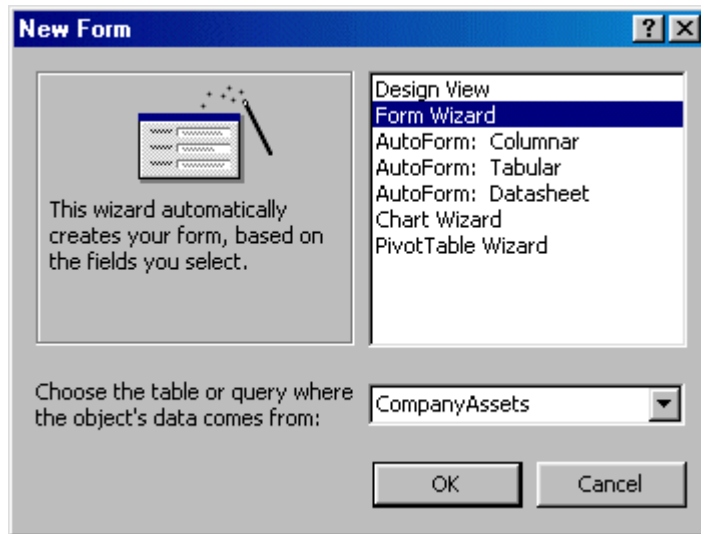
Microsoft Access

Kompanijos Microsoft kuriamame Office pakete esanti Access duomenų bazių valdymo sistema turi įdiegtą sąsajos (formų) generavimą. Sąsaja generuojama vedlio pagalba. Pirmajame vedlio lange pasirenkama kokio tipo ir kaip formą kurti – savarankiškam režime, vedlio pagalba, automatiškas generavimas (duomenys pateikiami stulpeliu, lentele ir pan.), diagramų formą.

Užklausų metu pasirenkamos lentelės iš kurių kuriama forma, įdedami valdymo mygtukai. Duomenys pildomi iš pasirinktų lentelių. Formų laukų pavadinimai imami iš lentelių stulpelių pavadinimų.



Microsoft Access formų generavimo schema

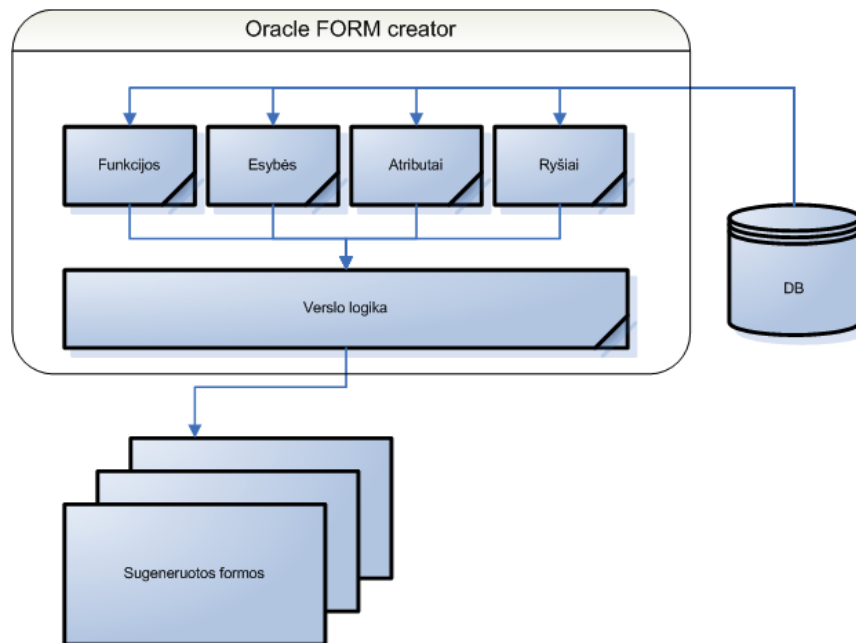


Pav. 7 Formų kūrimo vedlio pirmasis langas

Access DBVS projektui netinka, nes sąsaja kuriama iš lentelių, o ne lentelėse esančių duomenų.

Oracle

Oracle yra programinės įrangos paketas duomenų modeliavimui, transformavimui, specifikavimui. Jis skirtas reikalavimams kompiuterizuotai IS specifiuoti, KIS projektuoti ir realizuoti. Visas didelis paketas sudarytas iš atskirų mažesnių komponentų. Oracle Designer priemonėmis galima sugeneruoti ataskaitas, formas, meniu bei kitas interaktyvias ekranines formas. Čia specialūs generatoriai, naudodamiesi sudarytais projektais bei aplinkos parametrais generuoja atitinkamas pilnai funkcionuojančias kompiuterines programas su išėities tekstais.



Oracle 9i formų generavimo schema

Forma padaroma vykdomuoju failu, kurį vėliau naudoja Oracle formų modulis. Be lentelėse pateiktų duomenų formose taip pat naudojami įvairūs GUI elementai – meniu, mygtukai, grafiniai elementai ir pan.[3]

ODRES CASE įrankis

Informacijos sistemų katedroje plėtojamas ODRES metodas, skirtas automatizuotu būdu specifikuoti reikalavimus kompiuterizuotai informacinei sistemai. KIS kūrimo, taikant ODRES metodą, procesą sudarytų tokie etapai:

- Organizacijos veiklos modeliavimas
- KIS modeliavimas
- KIS projektavimas
- KIS realizavimas

Grafinė vartotojo sąsają būtų galima sukurti remiantis ODRES metode specifikuotais duomenų šaltiniais (pradiniais dokumentais).

Lentelėje pateikiami palyginimai tarp sąsajos kūrimo įrankių:

Lentelė 3 Įrankių palyginimai

	Ms Access	Oracle	Provision	ODRES
Duomenų šaltinis	Lentelės	Lentelės	UML diagramos	Metaduommenys
Priklausomybė nuo kūrimo aplinkos	Yra	Yra	Nėra. <i>Sąsaja susiejama su C++ arba VB kodu, kuris gali būti vykdomas kitose aplinkose.</i>	Nėra. Naudojamas sąsajos tekstinis aprašas.
Elementų keitimas	Yra	Yra	Nėra	-
Kūrimo automatizavimas	Dialogų pagalba	Oracle Designer / From generator	Automatiškas	-

2.8 Projekto tikslas

Šio projekto tikslas – sukurti programinę įrangą, kuri leistų automatiškai sukurti informacinės sistemos grafinės vartotojo sąsajos prototipą iš reikalavimų specifikacijos, sudarytos remiantis Informacijos sistemų katedroje pasiūlytu reikalavimų specifikavimo ODRES metodu. Sukurtą sąsajos prototipą leisti koreguoti rankiniu būdu bei jį išsaugoti.

Sukurtoji programinė įranga turi būti Informacijos sistemų katedroje kuriamo CASE įrankio, skirto reikalavimų specifikacijai sudaryti, dalimi. Ji suteiks papildomą funkcionalumą prie šio įrankio.

Pagrindinis projekto kokybės kriterijus – sukurtos programinės įrangos funkcionalumas. Pagrindinė sukurtoji programinės įrangos funkcija yra – automatiškai sukurti IS GUI prototipą iš duomenų bazėje pasirinktos specifikacijos. Duomenų bazėje saugoma tik vienos sistemos reikalavimų specifikacija. Programa ją turi išanalizuoti, sugeneruoti sąsajos prototipą, kurį vėliau galima būtų koreguoti ir išsaugoti toje pačioje bazėje.

Taip pat turi būti patikrinta, ar sukurtoji programinė įranga:

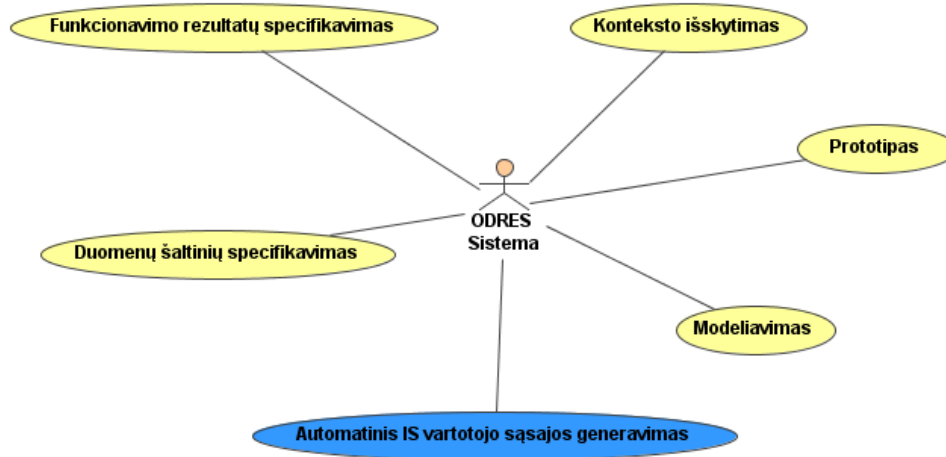
- integruojasi į Informacijos sistemų katedroje kuriamą CASE paketą,
- yra išbaigta, t. y., ar atitinka visus jai iškeltus funkcinius reikalavimus,
- yra tolerantiška klaidoms.
- Yra funkcionali:
 - Skaito duomenis
 - Išsaugo duomenis
 - Leidžia koreguoti elementus

- Perkeliamumas, t.y., ar galima suprojektuotą sistemą perkelti į kitą aplinką.

2.9 Kompiuterizuojamos sistemos varianto pasirinkimas

Pagrindinis kompiuterizuojamas uždavinys – KIS grafinės sąsajos prototipo kūrimas.

Šis uždavinys, tai vienas ODRES metodu kuriamos IS etapas:



Plėtojant šį uždavinį turi būti tenkinami projekto tiksluose išskelti reikalavimai.

2.10 Analizės išvados

1. Atlikus pasirinktų grafinės sąsajos generavimo įrankių (Oracle, Provision WorkBench, Ms Access) analizę, nustatyta, kad jie susieti su konkrečiomis duomenų bazių valdymo sistemomis ir dėl to negali būti panaudoti ODRES metodo automatizuojančiam įrankiui.

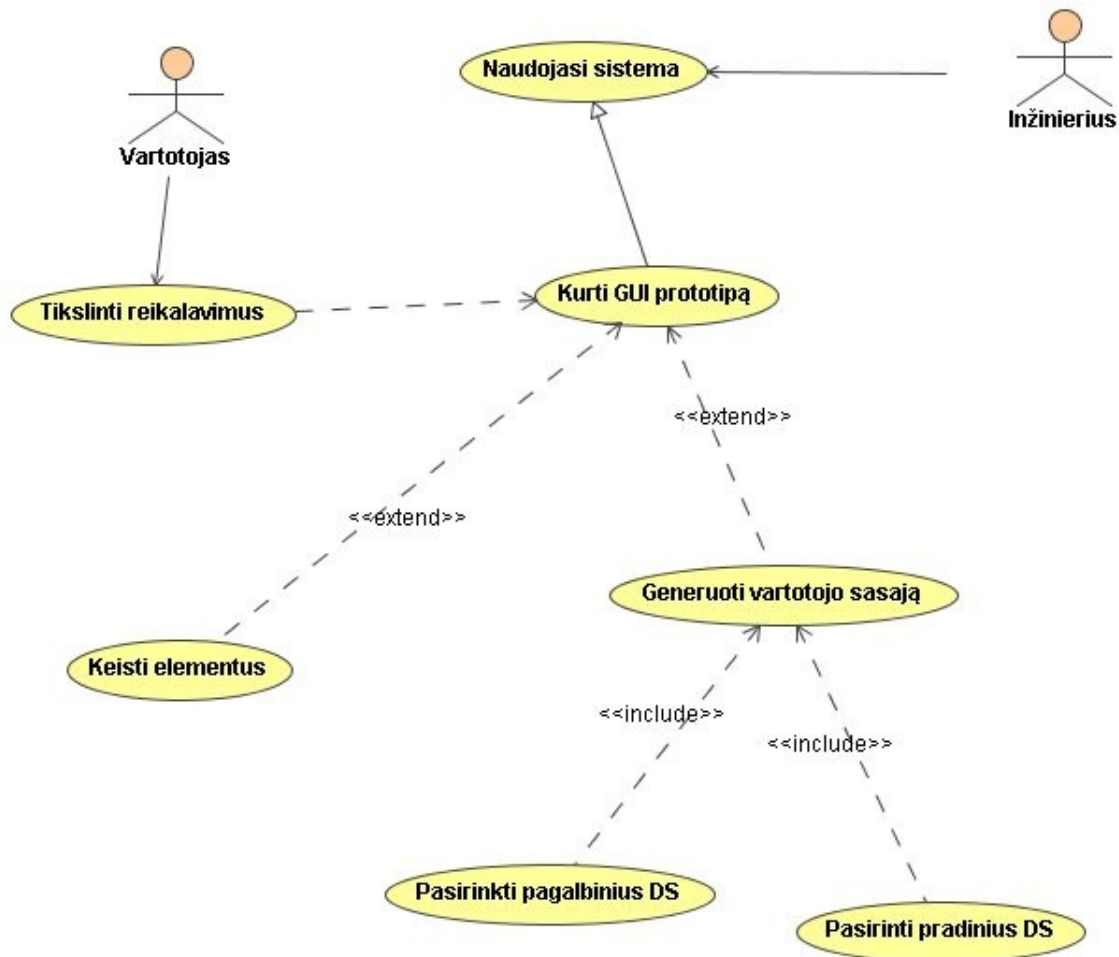
2. Generuojamų formų duomenų šaltiniams įrankiai naudoja bazėse saugomas duomenų lenteles, bet ne pačius duomenis. ODRES metodo automatizuojančiam įrankiui reikalingi bazėje saugomi metaduomenys..

3. Minėtais įrankiais sugeneruotas formas negalima išsaugoti tose pačiose saugyklose. ODRES metodu išgautos specifikacijos vartotojo sąsajos prototipui generuoti reikalingas įrankis galintis generuoti sąsają iš konkrečios (šiuo metodu suprojektuotos) specifikacijos saugyklos metaduomenų. O rezultata – sąsajos aprašą, reikia patalpinti toje pačioje duomenų saugykloje

3 Reikalavimai GUI automtizuoto įrankio kūrimui

3.1 GUI automatizuoto įrankio reikalavimų specifikacija

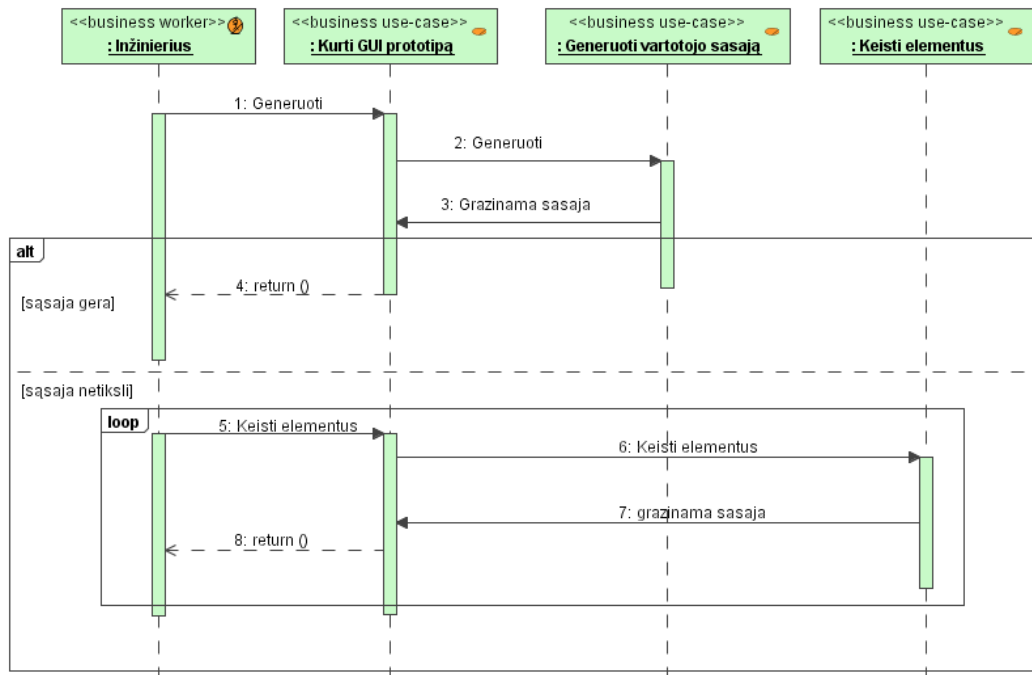
Pasirinktas kompiuterizuojamas etapas – grafinės vartotojo sąsajos generavimo įrankis. Paveiksle pavaizduoti šio etapo panaudojimo atvejai:



Pav. 8 Pasirinkto kompiuterizuojamo uždavinio panaudojimo atvejai

Sistemos vykdyme dalyvauja du aktoriai – inžinierius, kuris inicijuoja sąsajos generavimą, ir vartotojas, kuris gali keisti sugeneruotą sąsają. Pagrindinis panaudojimo atvejis pasirinktam kompiuterizuojamam etapui yra kurti GUI prototipą. Šį panaudojimo atvejį išplečia panaudojimo atvejai „keisti elementus“ ir „generuoti vartotojo sąsają“.

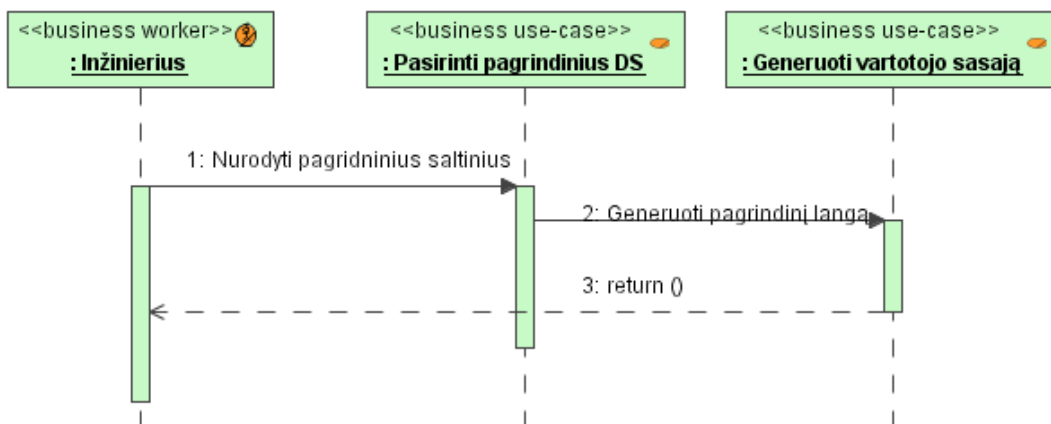
Sekos diagrama panaudojimo atvejui „generuoti sąsają“:



Pav. 9 Sekų diagrama panaudojimo atvejui „Generuoti sąsają“

PA „Generuoti sąsają“		
Prieš sąlyga	Vartotojas inžinierius Užpildyta DB	
Sužadinimo sąlyga	Inžinierius nori sugeneruoti vartotojo sąsają	
Susiję panaudojimo atvejai	Išplečia PA	Kurti prototipa
	Apima PA	
	Specializuoja PA	
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai	
1. Parenkamas duomenų šaltinis	1.1 Sistema pateikia raktinių elementų išrinkimo langą	
2. Parenkami raktiniai	2.1 Sukuriama forma su raktiniais	
	2.2 Tikrinamas kitų elementų	
	2.3 Parenkami elementų įvedimo tipai <ul style="list-style-type: none"> • Įvedimo laukas • Data Grid • Sąrašas 	
3. Koreguojamos padėtys		
4. Išsaugoma	4.1 Į DB įrašomos elementų padėtys	
Po sąlyga:	Pateikiama sugeneruota vartotojo sąsają	

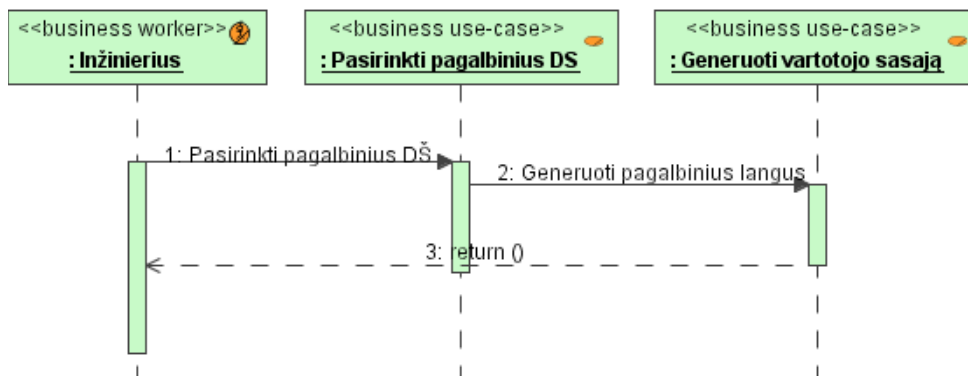
Sekų diagrama panaudojimo atvejui "Pasirinkti pagrindinius DŠ"



Pav. 10 Sekų diagrama panaudojimo atvejui "Pasirinkti pagrindinius DŠ"

PA „Pasirinkti pagrindinius DŠ“		
Prieš sąlyga	Vartotojas inžinierius Užpildyta DB	
Sužadinimo sąlyga	Inžinierius nori sugeneruoti vartotojo sąsają	
Susiję panaudojimo atvejai	Išplečia PA	Kurti prototipą
	Apima PA	
	Specializuoja PA	
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai	
1. Pasirenkamas duomenų šaltinis	1.1 Sistema išrenka šiam šaltiniui priklausančius atributus ir esybes.	
	1.2 Pateikiamas pagrindinis langas	
Po sąlyga:	Pateikiama sugeneruota vartotojo sąsają	

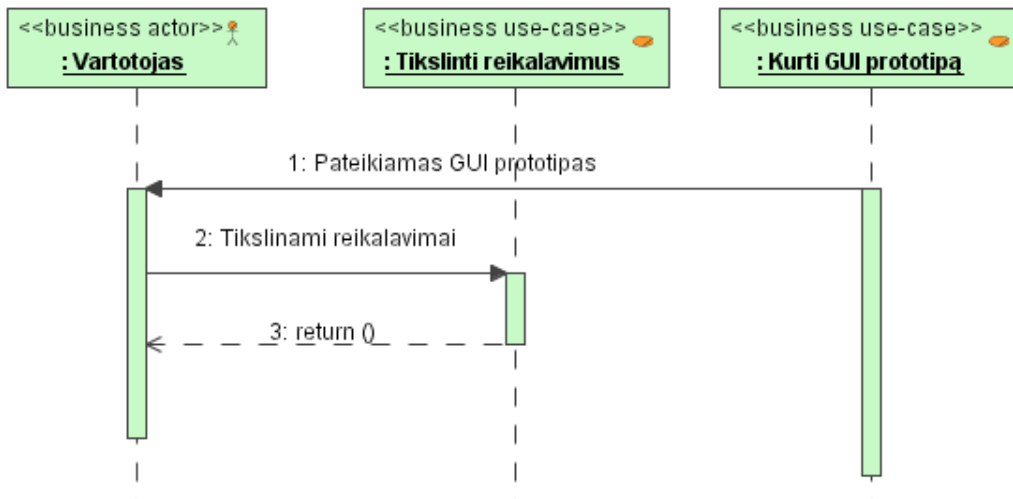
Sekos diagrama panaudojimo atvejui " Pasirinkti pagalbinus DŠ"



Pav. 11 Sekų diagrama panaudojimo atvejui " Pasirinkti pagalbinus DŠ"

PA „Pasirinkti pagalbinus DŠ“		
Prieš sąlyga	Vartotojas inžinierius Užpildyta DB	
Sužadinimo sąlyga	Inžinierius nori sugeneruoti vartotojo sąsają	
Susiję panaudojimo atvejai	Išplečia PA	Kurti prototipą
	Apima PA	
	Specializuoja PA	
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai	
1. Pasirenkamas duomenų šaltinis	1.1 Sistema išrenka šiam šaltiniui priklausančius atributus ir esybes.	
	1.2 Pateikiamas pagalbinis langas	
Po sąlyga:	Pateikiama sugeneruota vartotojo sąsają	

Sekų diagrama panaudojimo atvejui "Tikslinti reikalavimus"



Pav. 12 Sekų diagrama panaudojimo atvejui "Tikslinti reikalavimus"

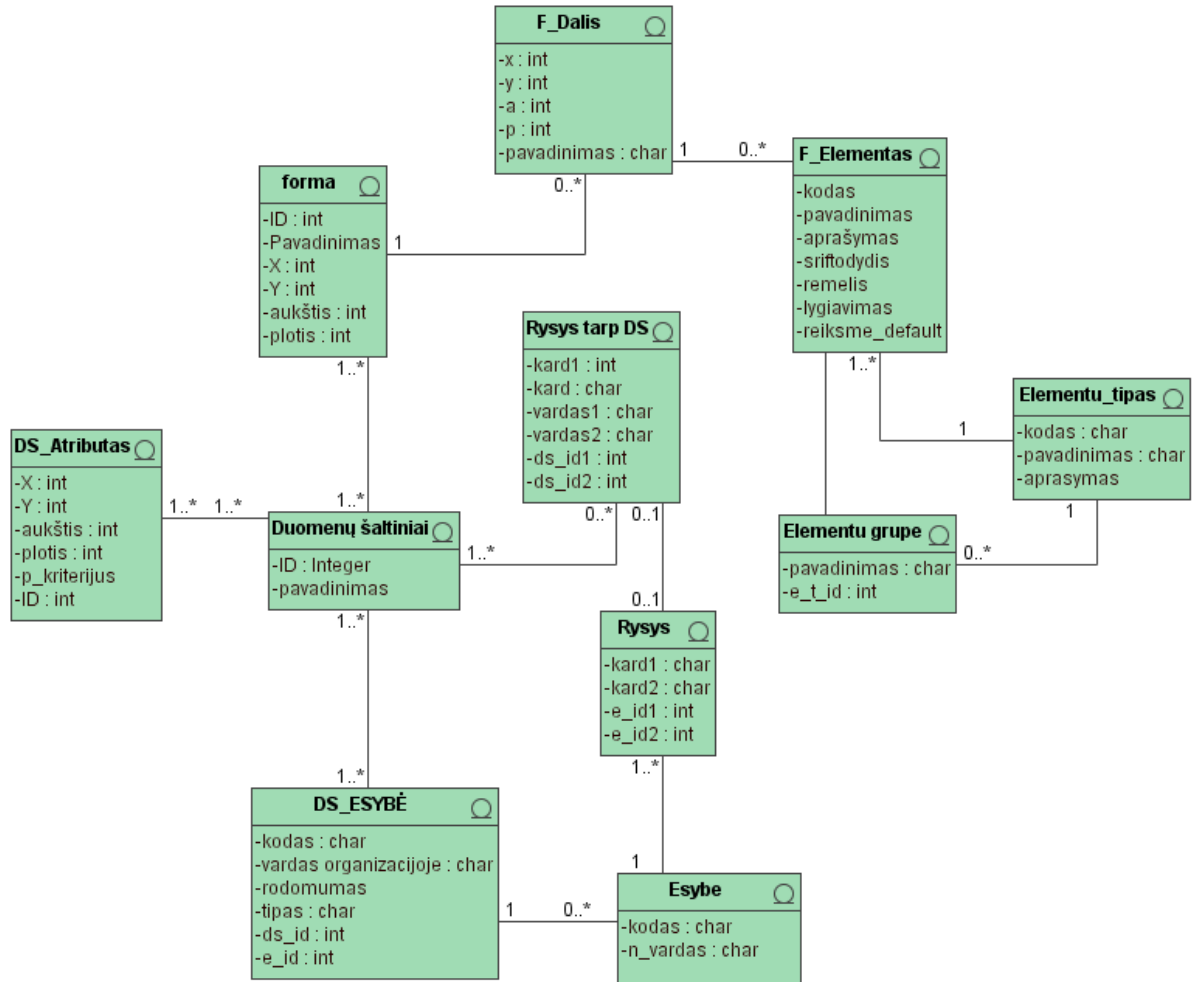
PA „Tikslinti reikalavimus“	
Prieš sąlyga	Vartotojas inžinierius Užpildyta DB
Sužadinimo sąlyga	Inžinierius nori sugeneruoti vartotojo sąsaja
Susiję panaudojimo atvejai	Išplečia PA Kurti prototipą
	Apima PA
	Specializuoja PA
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Pasirenkami koreguojami elementai	1.1 Pateikiami galimi keitimo variantai
2. pasirenkami nauji	2.1 išsaugomi duomenys metabazėje
Po sąlyga:	Pateikiama sugeneruota vartotojo sąsaja

3.2 Dalykinės srities modelis

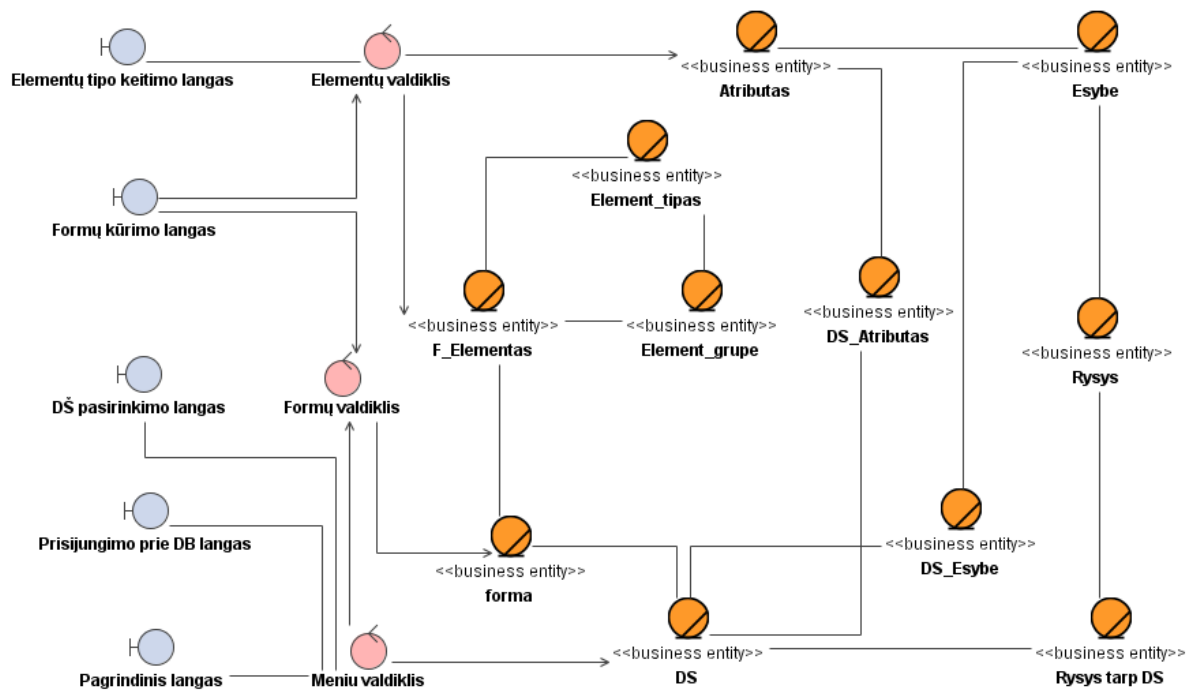
Vartotojo sąsajos generavimo dalykinės srities modelis pavaizduotas 13-ame pav.

Šiame modelyje pavaizduotos esybės būtinos vartotojo sąsajos generavimui. Pagrindinė esybė – Duomenų šaltiniai, kuriuos aprašo duomenų šaltinių atributas su savais elementais.

Kiekvienas duomenų šaltinis gali turėti nuosavą formą.



Pav. 13 Dalykinės srities modelis



Pav. 14 Sąsajos generavimo analizės modelis

Analizės modelyje dalyvauja langai – pagrindinis langas kuriame, menu pagalba parenkami netikslių elementų atributai ar specifikuojama trūkstama informacija sėkmingam sąsajos generavimui.

Formų kūrimo lange pateikiama jau sugeneruota sąsaja, kurią galima koreguoti. Už koregavimą atsakingi formų ir elementų valdikliai. Elementų tipai, padėtys ir parametrai koreguojami Elementų tipo keitimo lange.

DŠ pasirinkimo lange pasirenkami naudojami duomenų šaltiniai.

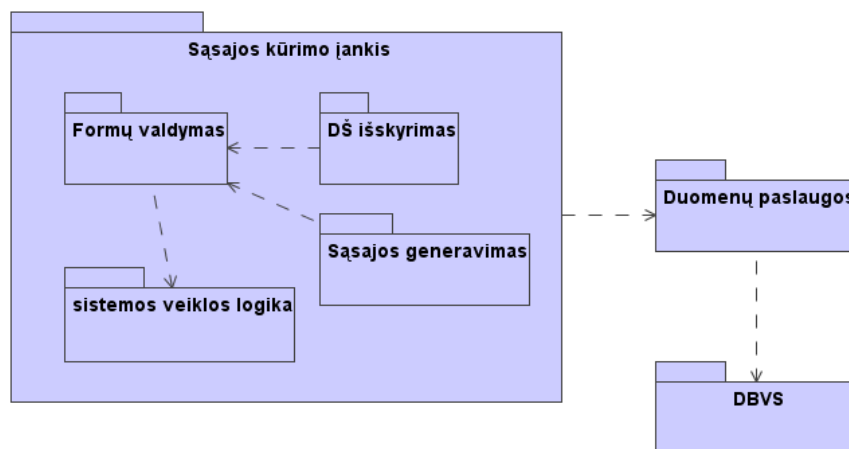
4 Sąsajos generavimo įrankio projektas

4.1 Sprendimo pagrindimas

Sąsajos generavimo įrankis susietas su ODRES saugykla, kur saugomi duomenys apie sąsajos elementus. Įrankis susietas tik su konkrečia saugykla, ir jo įnašas į šį metodą – sugeneruoti grafiniai elementai, kurie toliau bus naudojami prototipo kūrimui ir susiejimui su reliacine duomenų baze. Įrankio realizavimo aplinka pasirinkta MS Visio 2003, kadangi joje jau egzistuoja daugelis standartinių formų ir formų elementų, bei tai aplinka, kurioje kuriami kiti ODRES metodo etapai.

4.2 Sistemos architektūra

4.2.1 Loginė sistemos architektūra



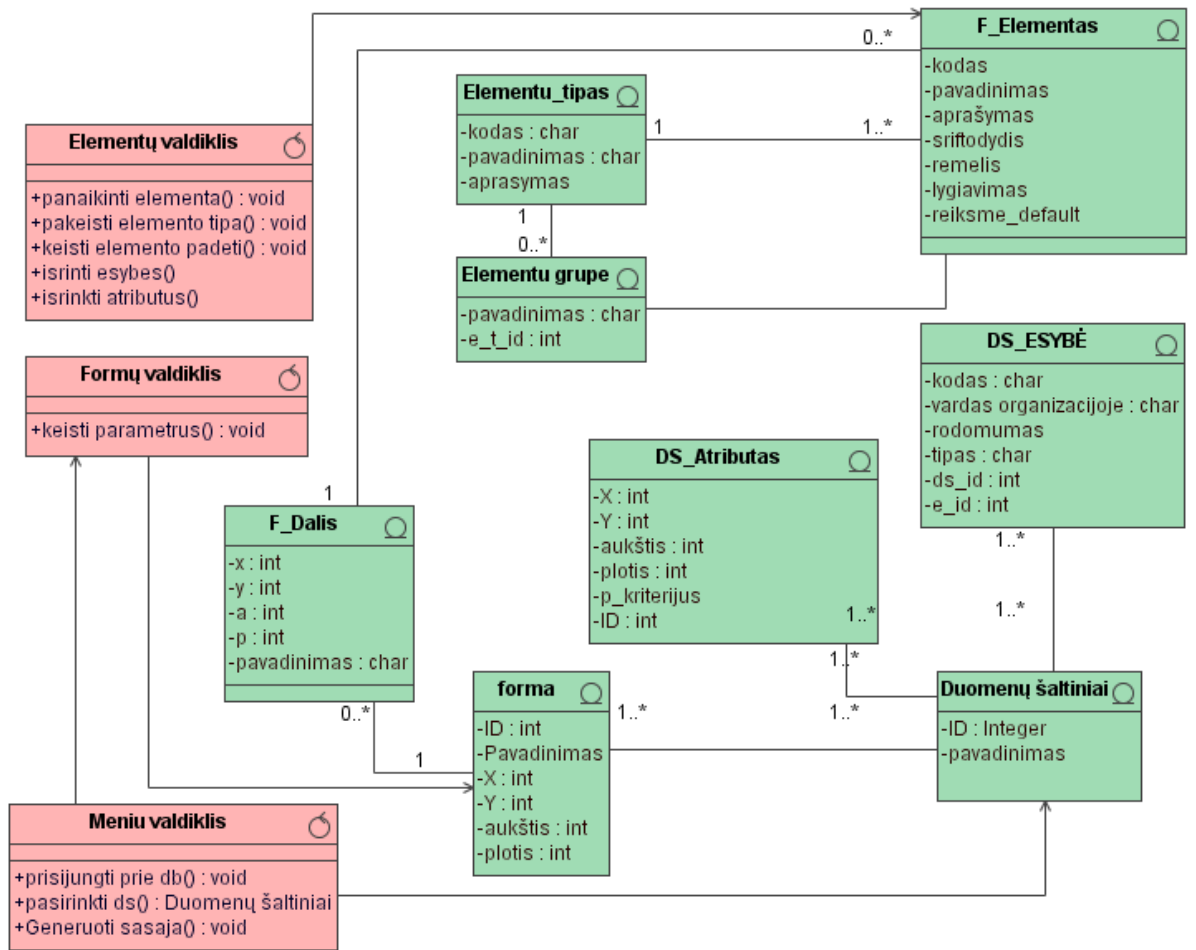
Pav. 15 Sistemos architektūra

Sąsajos kūrimo įrankis valdomas dialogų pagalba, išskirti esminiai dialogai :

- Duomenų šaltinių išskydimo dialogas
- Sąsajos generavimo dialogas

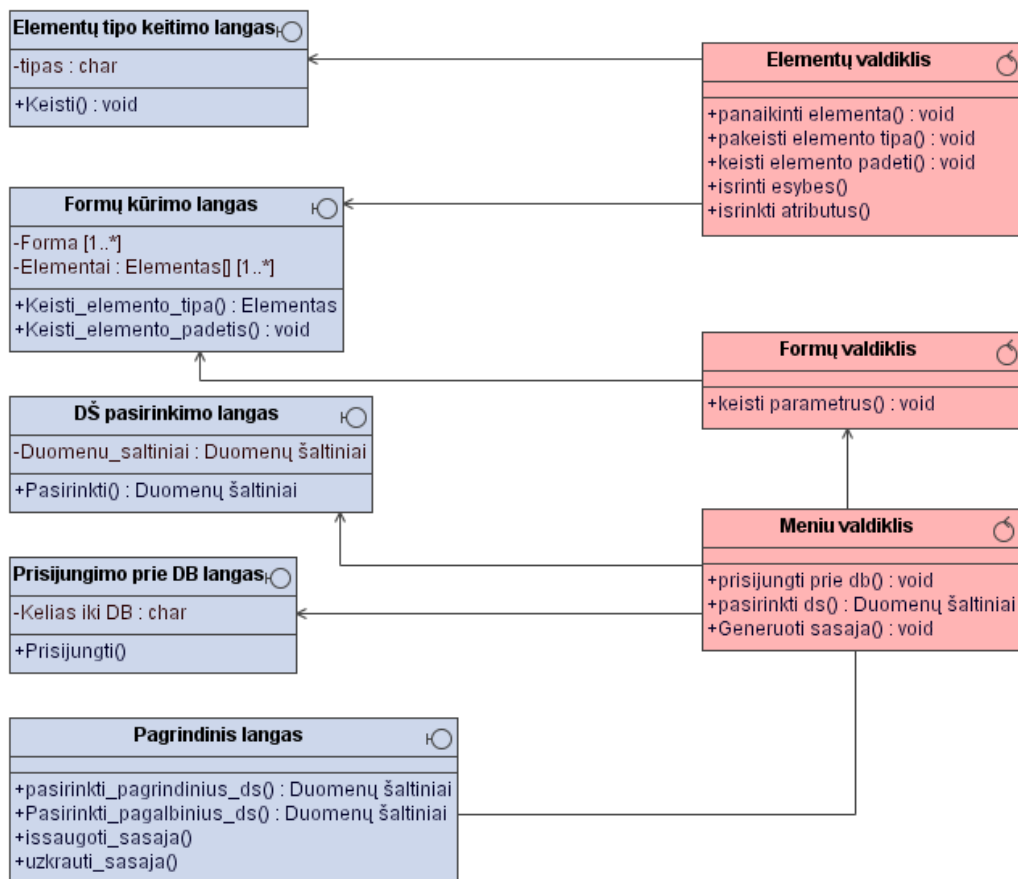
Vykdam projektą naudojama išorinė duomenų prieiga per duomenų paslaugų paketą susisiekiant su Duomenų baze.

4.2.2 Vartotojo, veiklos, duomenų paslaugų klasių diagramos



Pav. 16 Sąsajos generavimo veiklos logikos klasių diagrama

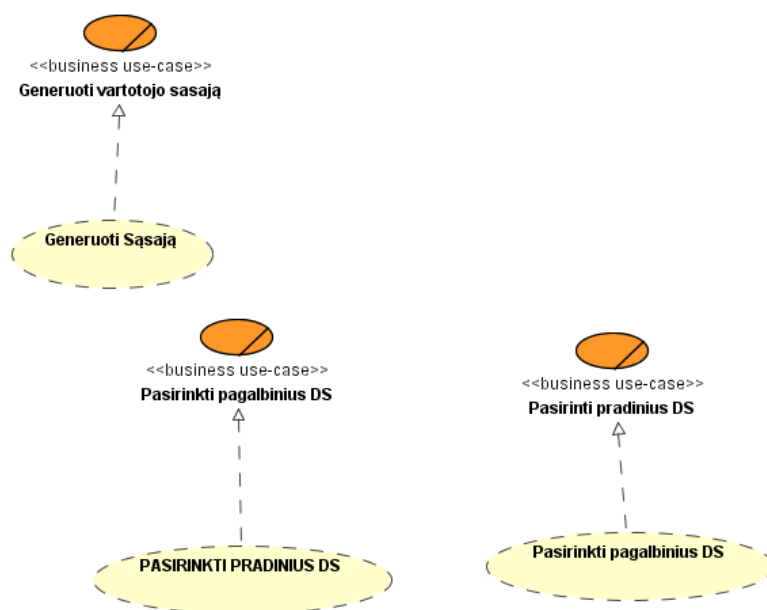
Diagramoje parodomi ryšiai tarp duomenų klasių ir valdiklių klasių.



Pav. 17 Sąsajos generavimo vartotojo logikos klasių diagrama

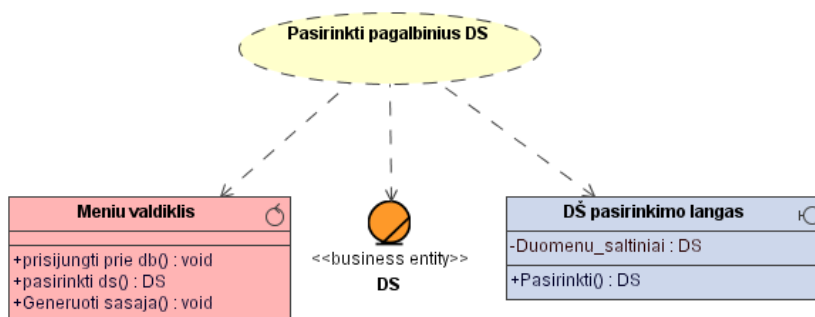
Šioje diagramoje pateikiami valdikliai ir formos dalyvaujantys sąsajos generavimo procese. Nurodomi atributai ir operacijos.

4.3 Detalus projektas



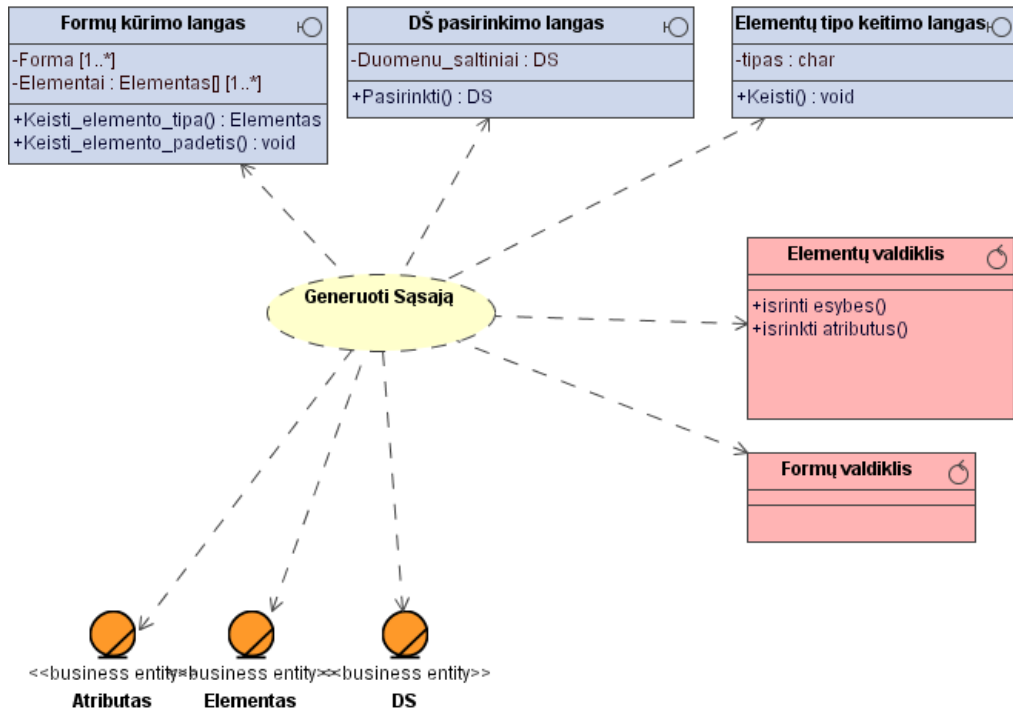
Pav. 18 Praplėsti panaudojimo atvejai

Pateikti pagrindiniai kompiuterizuojami panaudojimo atvejai, kurie susiejami su veiklos panaudojimo atvejais.



Pav. 19 Pasirinkti pagalbinus DŠ

Panaudojimo atvejis „Pasirinkti pagalbinus DŠ“ naudoja menui valdiklį, DŠ pasirinkimo langą ir DŠ esybę.

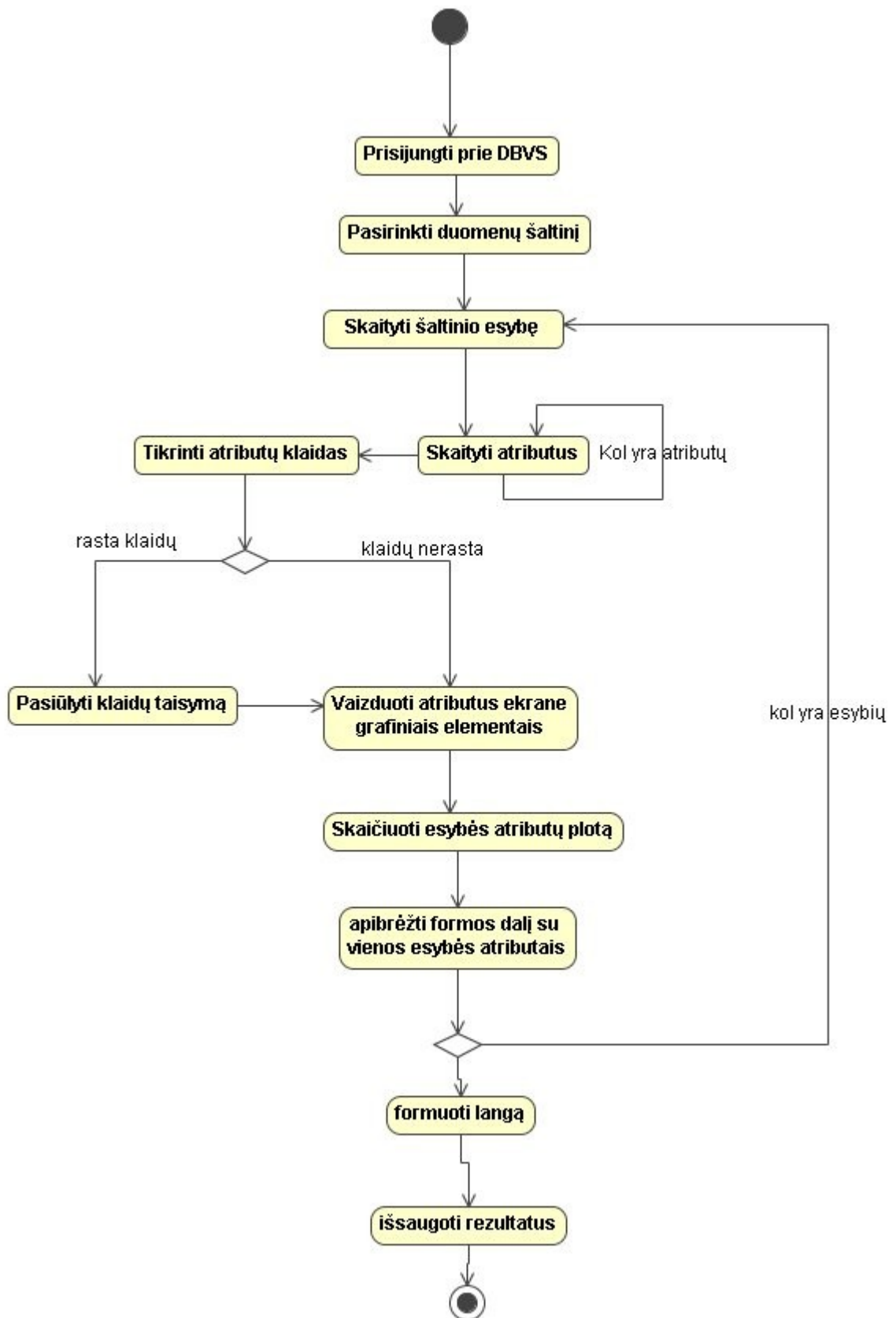


Pav. 20 Generuoti sąsaja

Praplėstoje panaudojimų atvejų diagramoje nurodyta su kokiomis klasėmis siejasi panaudojimo atvejis „Generuoti sąsaja“.

4.4 Sistemos elgsenos modelis

4.4.1 Sistemos veiklos modelis



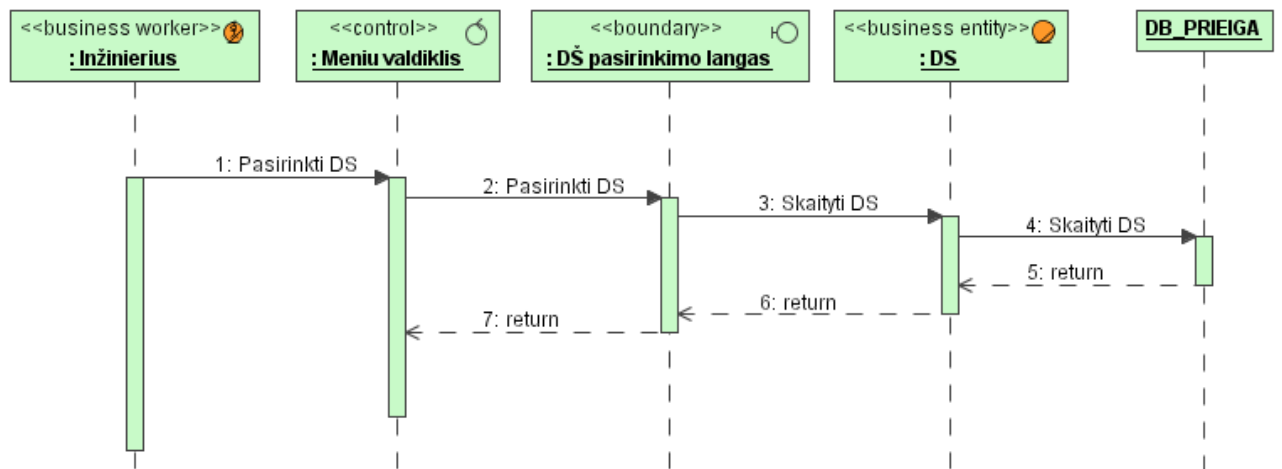
Pav. 21 Veiklos modelis

Veiklos diagramos pirmas žingsnis – tikrinti metabazės pilnumą, tai yra išsiaiškinama ar anksčiau buvo generuota sąsaja ir ar jau užpildyti elementų atributai. Jie sąsaja buvo generuota, skaitomi sąsajos lementų duomenys ir jie vaizduojami ekrane (veikla „Generuoti sąsają“). Kita sąlyga, jei sąsaja nebuvo generuota ir elementų atributų laukeliai tušti pereinama prie veiklos „Pasirinkti DŠ“. Čia pasirenkami esminiai duomenų šaltiniai kurie būtų vaizduojami sugeneruotos sąsajos pagrindiniame lange. Pasirinkus duomenų šaltinius, iš metabazės skaitomi pasirinktų DŠ atributai ir esybės. Veikla „vaizduoti DŠ elementus ekrane“ į lapą sudeda reikalingus atributus, o veikla „Generuoti sąsają“ juos išdėsto pagal logišką tvarką. Išdėsčios ekrane esminius duomenų šaltinių elementus pasirenkami mažiau aktualūs elementai kurie gali būti iškviečiami mygtukų ar kortelių pagalba.

Sugeneravus galutinę sąsają inžinierius gali keisti elementų tipus (pvz. teksto įvedimo lauką Textbox(single line) galima keisti į Textbox (multi line))

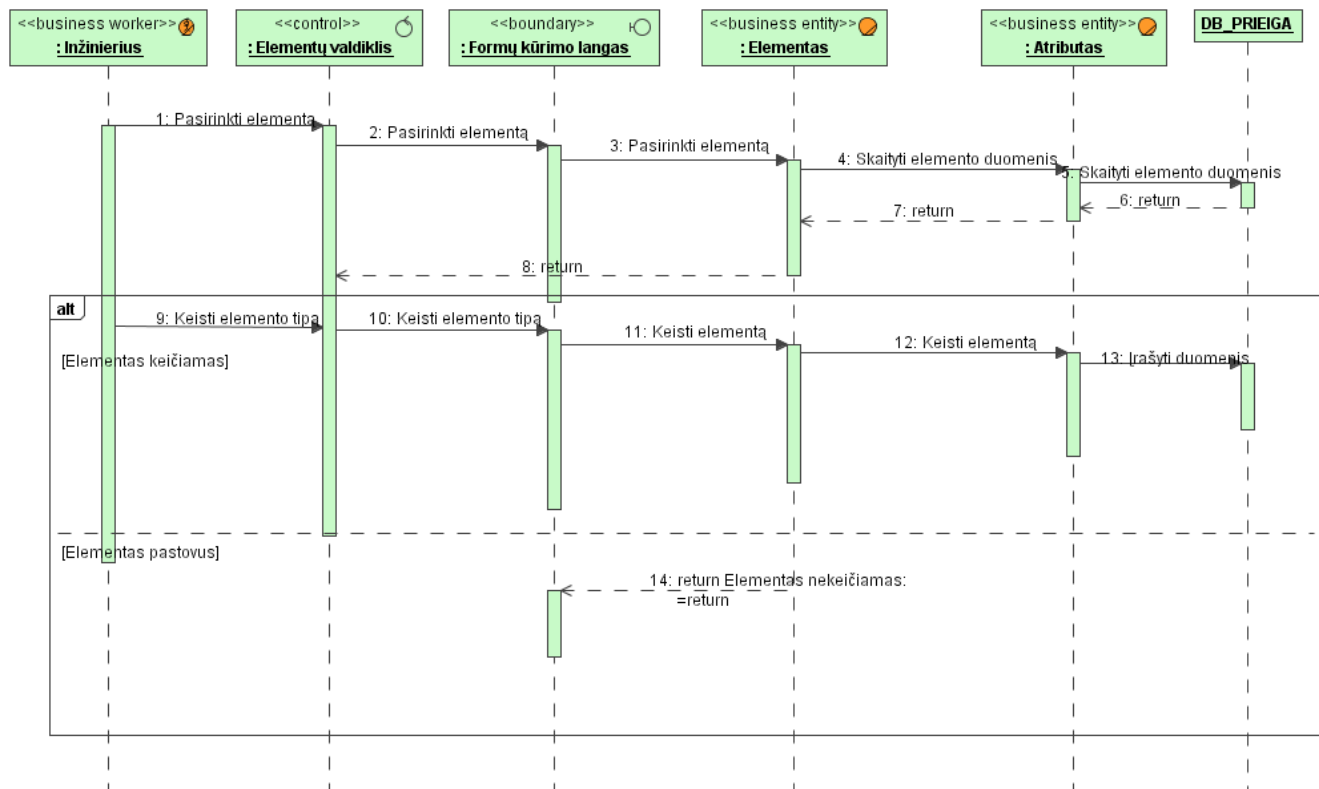
Sugeneruotos sąsajos rezultatai išsaugomi naudojamoje metabazėje.

4.4.2 Sekų diagramos



Pav. 22 Duomenų šaltinio pasirinkimo sekos diagrama

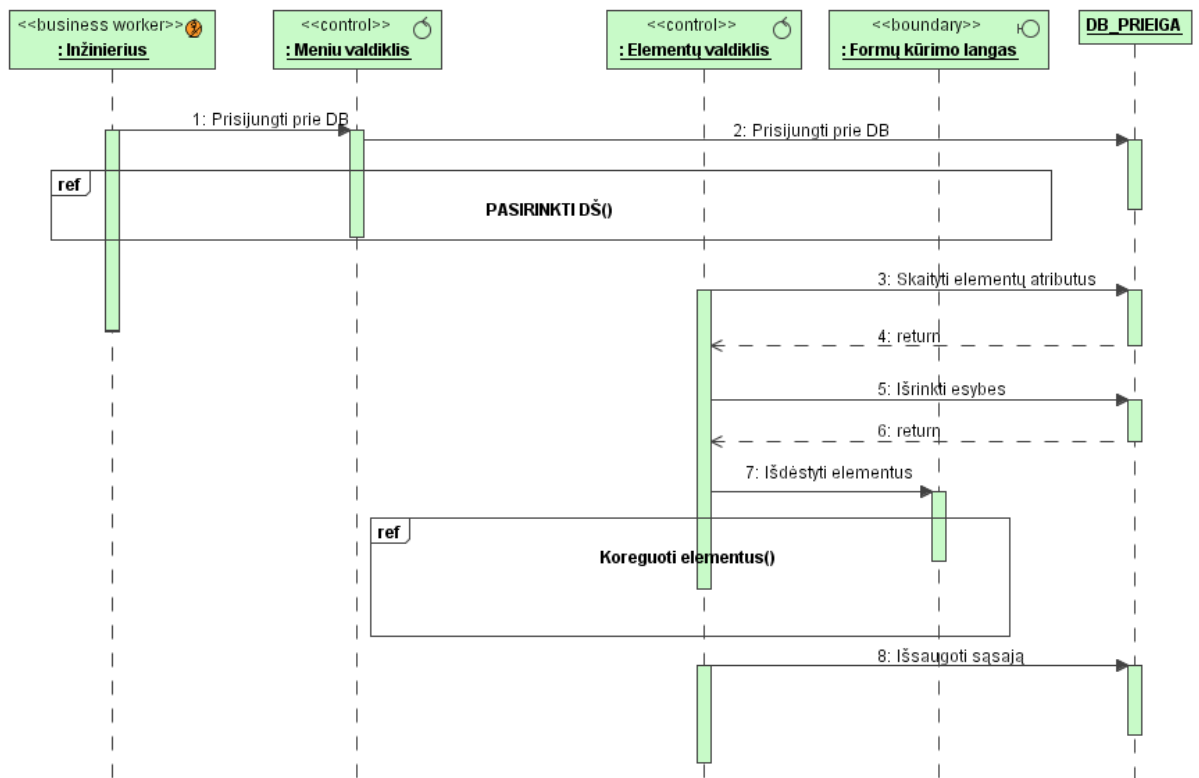
Ši sekos diagrama vaizduoja veiksmų seką atliekama pasirenkant duomenų šaltinius.



Pav. 23 Elementų keitimo sekos diagrama

Formų kūrimo lange pasirinkus elementą, išskviečiamas kontekstinis meniu, kurio pagalba galima keisti elemento tipą. Elementų keitimo eigoje dalyvauja elementų valdiklis, formų kūrimo langas, elementas ir atributas programinės klasės.

Keičiant elementą atsižvelgiama dvi alternatyvas, ar elementas gali būti keičiamas ar ne.



Pav. 1 Sąsajos generavimo sekos diagrama

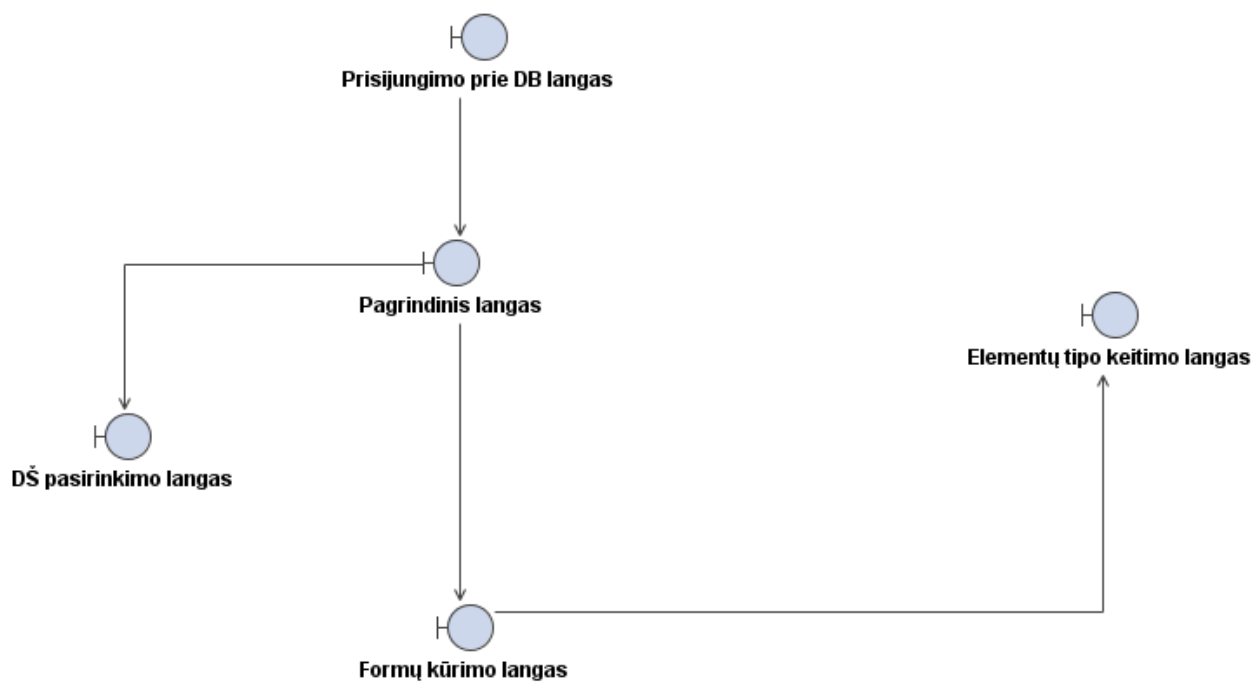
Ši sekos diagrama apima elementų keitimo / koregavimo bei duomenų šaltinių pasirinkimo klasių diagramas.

4.4.3 Navigacijos planas

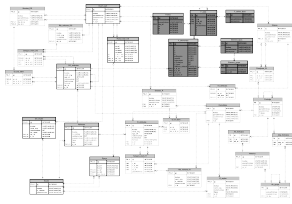
Navigacijos plane parodomi projekte esantys dialogo langai ir jų tarpusavio sąveika.

Prisijungimo prie duomenų bazės lange leidžiama pasirinkti duomenų saugyklą iš kurios skaitomi metaduomenys. Pagrindiniame lange galima pasirinkti veiksmus kuriuos atlieka grafinės sąsajos kūrimo įrankis – skaityti duomenis, tikrinti pilnumą, generuoti sąsają, išsaugoti duomenis.

Generuojant sąsają būtina pasirinkti kuriam duomenų šaltiniui ji generuojama, tai atliekama DŠ pasirinkimo lange. Pasirinkus duomenų šaltinį pereinama į formų kūrimo langą, kuriame pateikiamas sugeneruotos sąsajos eskizas ir kurį galima redaguoti.



Pav. 24 Navigacijos planas



Lentelėje Duom_salt saugomi duomenų šaltiniai, kiekvienas duomenų šaltinis atitinka sąsajoje sugeneruotą formą. Generuojant sąsają parenkamas vienas pagrindinis duomenų šaltinis, iš kurio generuojama valdančioji forma.

Esybe		
PK,11	id	INTEGER
	kodas	VARCHAR(10)
	n_vardas	VARCHAR(30)
	paaiskinimas	VARCHAR(200)

Lentelėje esybė saugomos pasirinkto duomenų šaltinio esybės. Esybė su savais atributais sudaro formos dalį.

DS_atributas		
PK,13	id	INTEGER
	kodas	VARCHAR(10)
	vard_org	VARCHAR(50)
	egz_sk	INTEGER
	rodumumas	BIT
	butinumas	BIT
	formule	VARCHAR(200)
FK2,12	ds_id	INTEGER
FK1,11	a_id	INTEGER

Iš lentelės DS_atributas generuojami sąsajos įvedimo elementai. Laukelis egz_sk parodo ar tai vieno lauko įvedimo elementas ar sąrašo elementas.

4.6 Testavimo modelis, pavyzdiniai duomenys

Duomenų bazėje turime tokius duomenis:

Lentelė 4 Duom_salt

	id	kodas	pavadinimas	paaiskinimas	salyga	h_lygis
▶	+	1 DS1	Sutartis	Sutarties duomenys		
	+	2 DR	Techninis pasas	draudžiamas objektas		
	+	3 AG	Darbo sutartis	Draudimo agentai		
	+	4 DRR	Asmens dokumentas	Draudiku duomenys		
*						

Lentelėje saugomi duomenys apie pradinis duomenų šaltinius. Pavyzdžiui naudosime draudimo informacinės sistemos duomenis. Duomenų šaltiniais laikomi: sutarties duomenys, techninis automobilio pasas, darbo sutartis (kominisinių sakičiavimui), asmens dokumentas (draudėjo duomenys)

Lentelė 5 Ds_esybe

	id	kodas	vard_org	rodomumas	ds_tipas	ds_id	e_id
▶		1 nr	Poliso numeris	-1	1	1	1
		2 drd	Draudejas	-1	1	1	2
		3 drobj	Objektai	-1	1	1	3
		4 es4	identifikavimo n	-1	1	2	4
		5 sut	Sutartis	-1	1	3	5
*							

Lentelė 6 Esybe

	id	kodas	n_vardas	paaiskinimas
▶	+	1 nr	Sutartis	Poliso serija ir numeris
	+	2 drd	Draudejas	Draudėjo duomenys
	+	3 obj	Draudimo objektai	Apdrausti objektai
	+	4 msn	Masina	Masinos duomenys
	+	5 sut	Sutartis	Darbo sutarties numeris
*				

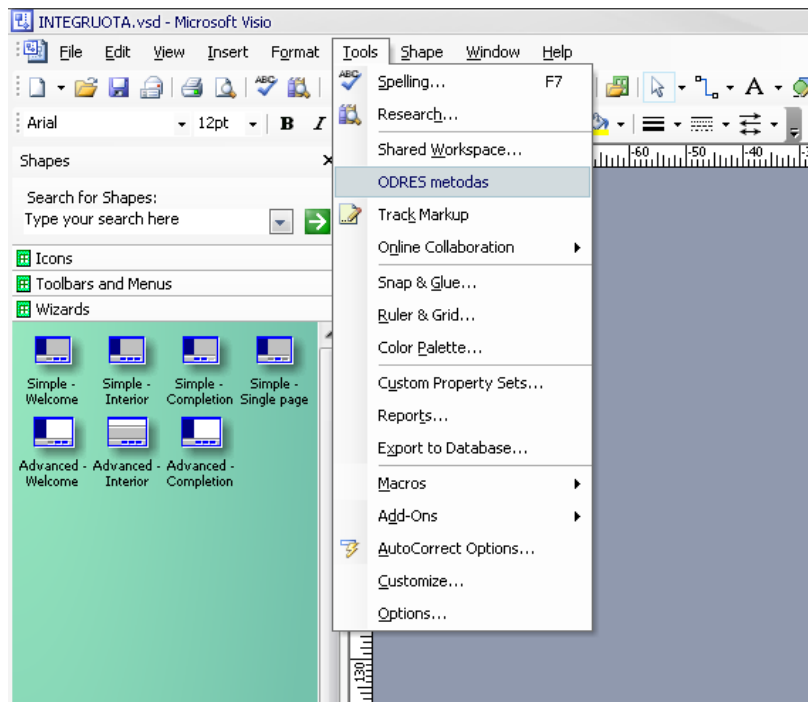
Lentelė 7 Ds_ atributas

	id	kodas	vard_org	egz_sk	rodomumas	butinumas	formule	ds_id	a_id
▶	+	1 id1	Nuo	1	-1	0		1	1
	+	2 id2	Iki	1	-1	0		1	2
	+	3 id3	Trukme	1	-1	0	Iki - NUO	1	4
	+	4 id4	Komisas	1	0	0		1	3
	+	5 id5	Kodas	1	-1	0		1	5
	+	6 id6	Vardas	1	-1	0		1	6
	+	7 id7	Pavardė	1	-1	0		1	7
	+	8 id8	Valstybinis numeris	15	-1	0		1	8
	+	9 id9	Marke	5	-1	0		1	9
	+	10 id0	Modelis	5	-1	0		1	10
	+	11 ie1	Identifikavimo numeris	5	-1	0		1	11
	+	12 ie2	Adresas	1	-1	0		1	12
	+	13 ie3	Ident. numeris	1	-1	0		2	13
	+	14 ie4	Pagaminimo m.	1	-1	0		2	14
	+	15 ie5	Valstybinis Nr.	1	-1	0		2	15
	+	16 e16	Numeris	1	-1	0		3	16
	+	17 e17	Vardas	1	-1	0		3	17
	+	18 e18	Pavardė	1	-1	0		3	18
	+	19 e19	kodas	1	-1	0		3	19
	+	20 e20	rusis	6	-1	0		3	20
	+	21 e21	komisas	6	-1	0		3	21
*									

Lentelė 8 Atributas

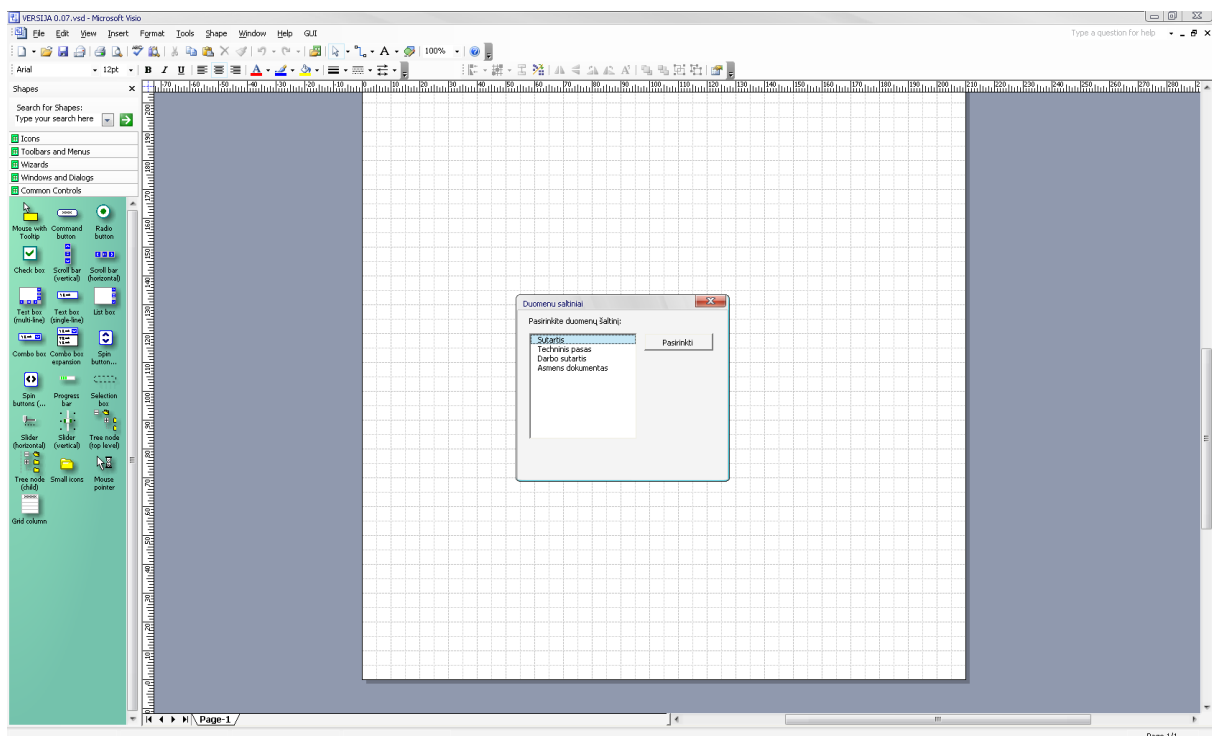
	id	kodas	n_vardas	unikalumas	paaiskinimas	tipas	ilgis	e_id
▶	+	1 atr01	Nuo	0	Galioja nuo	Date	10	1
	+	2 atr02	Iki	0	Galioja iki	Date	10	1
	+	3 atr03	Komisas	0	Komisiniu dydis	Integer	2	1
	+	4 atr04	Trukme	0	Sutarties trukmė	Date	10	1
	+	5 atr05	Kodas	-1	Asmens kodas	String	15	2
	+	6 atr06	Vardas	0	Draudėjo vardas	String	15	2
	+	7 atr07	Pavardė	0	Draudėjo pavardė	String	15	2
	+	8 atr08	kodas	0	Objektai	String	15	3
	+	9 atr09	marke	0	Marke	String	15	3
	+	10 atr10	modelis	0	modelis	string	15	3
	+	11 atr11	Identifikavimo n	-1	identifik.	String	15	3
	+	12 atr12	Adresas	0	Draudejo adresas	String	50	2
	+	13 atr13	Identifikavimo n	-1	identifikavimo n	String	15	4
	+	14 atr14	Metai	0	pagaminimo metai	Date	10	4
	+	15 atr15	Valst Nr.	0	valstybinis numeris	String	10	4
	+	16 atr16	Nr.	-1	Sutarties numeris	String	10	5
	+	17 atr17	Vardas	0		String	10	5
	+	18 atr18	Pavardė	0		String	10	5
	+	19 atr19	Kodas	-1	asmens kodas	String	10	5
	+	20 atr20	rusis.	0		String	10	5
	+	21 atr21	komis	0	komisas	double	3	5
*								

Programa paleidžiama per „Tools“ meniu punktą „Odres metodus“:



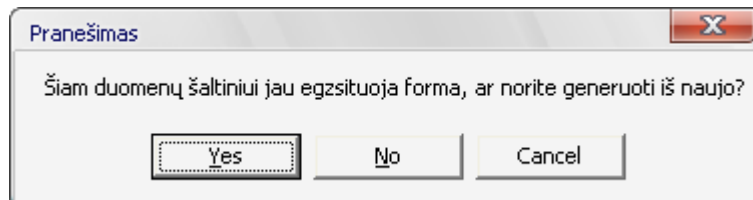
Pav. 26 Paleidimas

Vykdamt programą pirmiausiai duodama pasirinkti pagrindinį duomenų šaltinį iš kurio būtų generuojamas pagrindinis sąsajos langas:



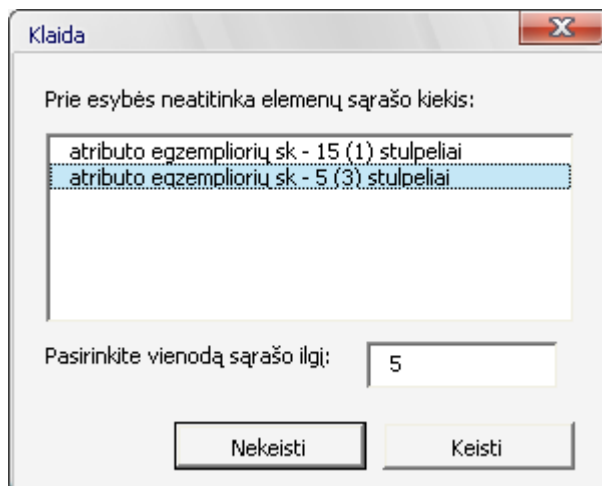
Pav. 27 Duomenų šaltinių pasirinkimo langas

Tikrinama ar nebuvo išsaugoti ankstesni duomenys:



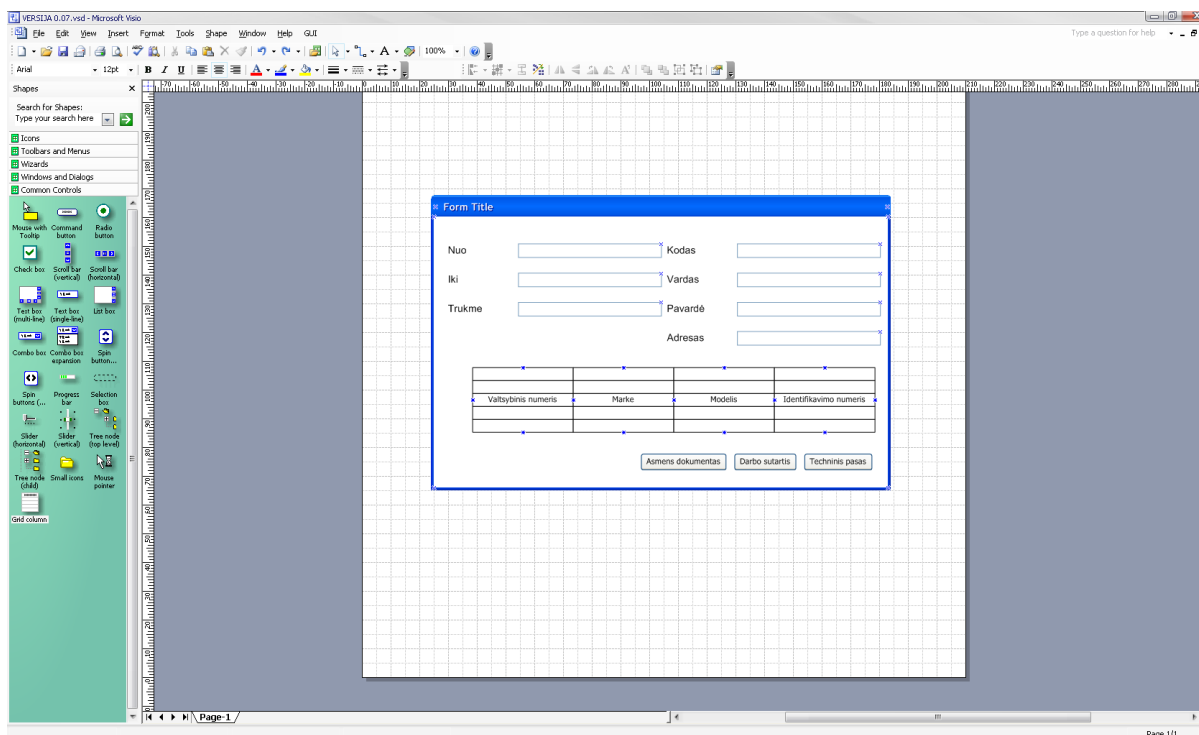
Pav. 28 Pranešimas apie anksčiau generuotą formą

Toliau vykdant generavimą, tikrinamos klaidos esančios pačioje duomenų saugykloje:



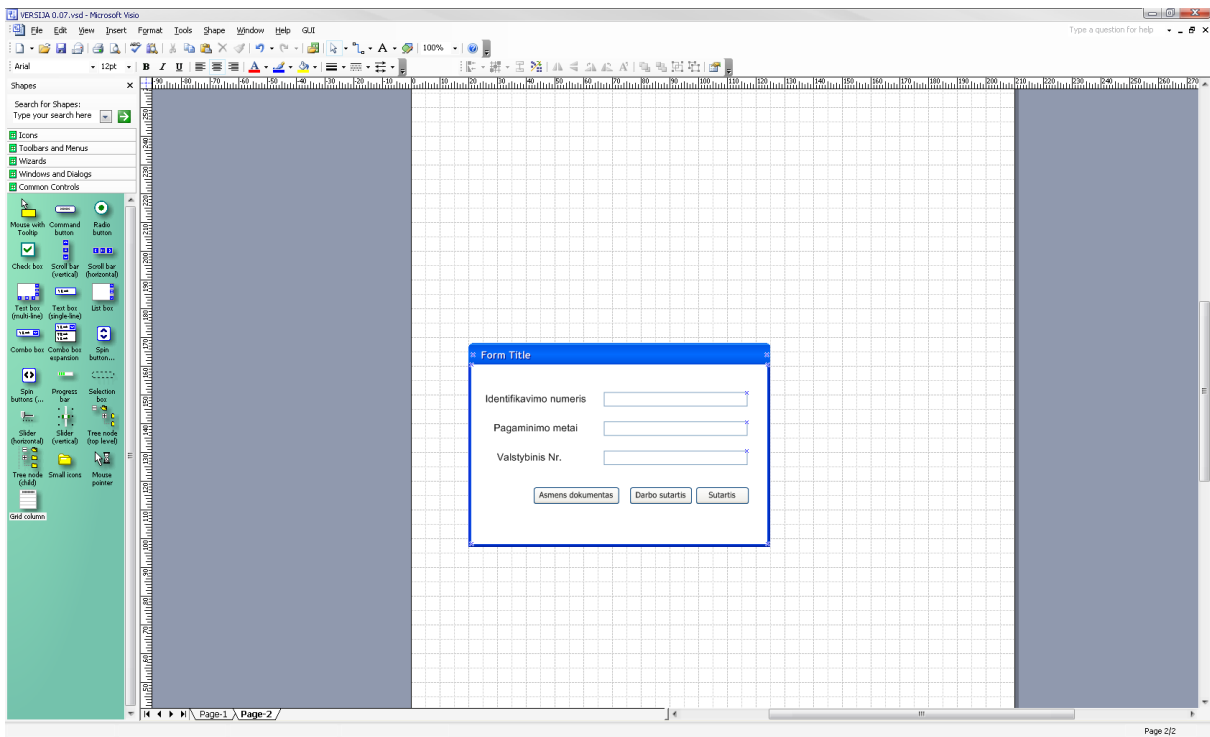
Pav. 29 Pranešimas apie klaidingai užpildytą sąrašo elementą

Ms Visio lange pateikiamas sąsajos eskizas:

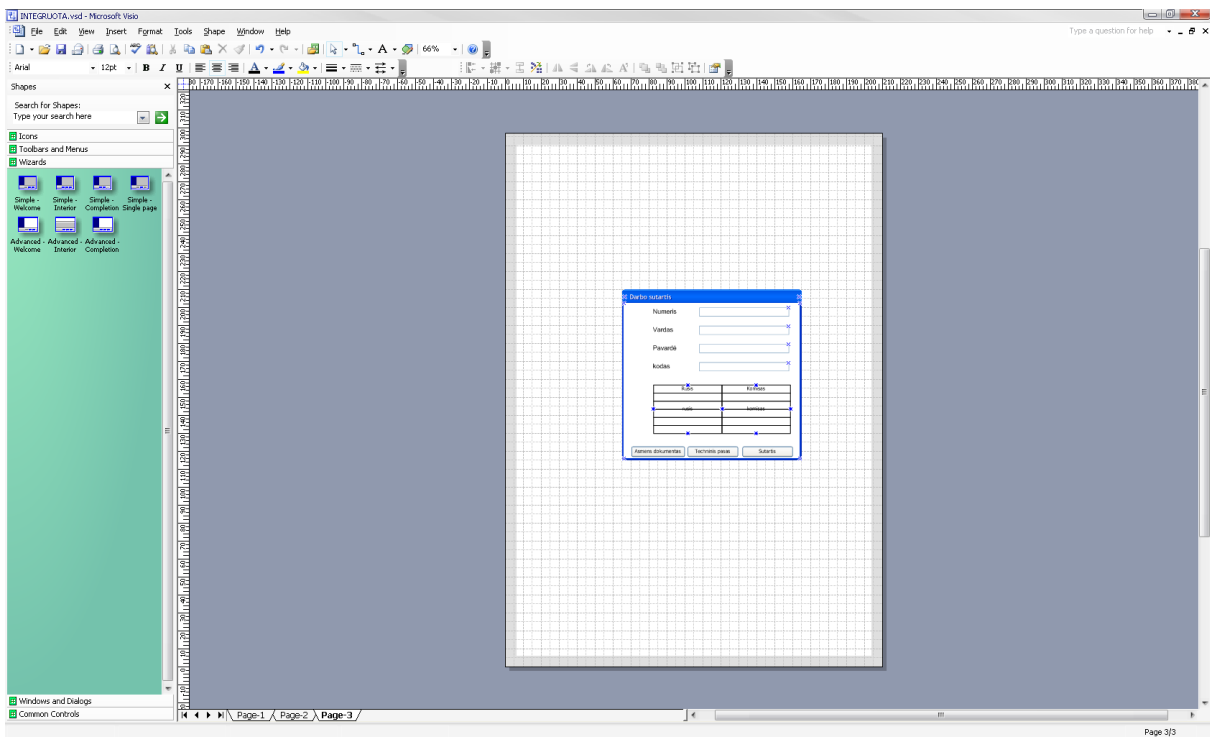


Pav. 30 Pagrindinio šaltinio sugeneruota forma

Toliau vykdant programą generuojami pagalbiniai duomenų šaltiniai:



Pav. 31 Pagalbinio šaltinio sugeneruota forma



Pav. 32 Pagalbinio šaltinio sugeneruota forma

Kiekvieno sugeneruoto sąsajos elemento duomenis galima pažiūrėti pasirinkus elemento *custom properties* kontekstinį meniu:

Pav. 33 Pasirinkto elemento duomenys

Duomenys išsaugomi toje pačioje duomenų saugykloje:

Lentelė 9 Forma

		id	pavadinimas	ds_id	a	p	x	y
▶	+	16	Techninis	2	130	154	76	25
	+	120	Sutartis	1	60	85	153	65
*								

Lentelėje forma išsaugomi formos duomenys

Lentelė 10 F_Elementas

id	x	y	a	p	sriftodydis	remelis	lygiavimas	p_kriterijus	p_tipas	reiksme_default	f_d_id	e_t_id	ds_a_id	grupes_id
1	10	60	10	40	12	0	0	0	0.0		1	1	1	1
2	10	80	10	40	12	0	0	0	0.0		1	1	2	1
3	10	102	10	40	12	0	0	0	0.0		1	1	3	1
4	130	60	10	40	12	0	0	0	0.0		1	1	4	1
5	130	80	10	40	12	0	0	0	0.0		1	1	5	1

Kiekvieno grafinės sąsajos elemento duomenys išsaugomi lentelėje F_elementas.

Lentelė 11 Element_tipas

		id	kodas	pavadinimas	aprasymas
	+	1	1	Text box (single	
▶	+	2	2	Label	
	+	3	3	Grid Column	
*					

5 Eksperimentinis tyrimas

Sukurtos sistemos kokybę įvertiname pagal 2.8 skyriuje apibrėžtus kokybės kriterijus 5 balų sistema. 1 – sistema visiškai neatitinka iškelto kriterijaus, 2 – sistema blogai atitinka kriterijų, 3 – sistema patenkinamai atitinka kriterijų, 4 – sistema gerai atitinka kokybės kriterijų, 5 – sistema puikiai atitinka kriterijų.

PĮ IS prototipui kurti atitikimas kokybės kriterijams

	1	2	3	4	5
Funkcionalumas				+	
o Skaitomi užpildyti duomenys			+		
o Išsaugomi duomenys				+	
o Leidžiama koreuoti elementus				+	
Integracija į ODRES CASE įrankį			+		
Išbaigtumas			+		
Tolerancija klaidoms				+	
Perkeliamumas				+	

6 Darbo rezultatai

1. Išanalizuoti sąsajos elementų išdėstymo būdai ir jų panaudojimo darbe atvejai
2. Apibrėžtos viena elementą apibūdinančios savybės, būtinos išsaugant elemento aprašą metabazėje.
3. Palygintas plėtojamas įrankis su kitais alternatyviais sąsajos kūrimo įrankiais.
4. Specifikuoti reikalavimai keliami plėtojamam grafinės sąsajos automatizuotas kūrimo įrankiui
5. Suprojektuota GUI automatizuoto įrankio kūrimo sistemos architektūra.
6. GUI automatizuoto įrankio sistema realizuota Microsoft VISIO paketu, naudojant VBA programavimo kalbą.

7 Išvados

1. Išanalizavus pasirinktus *Provision Workbench*, *Oracle Designer* ir *Ms Access* grafinės vartotojo sąsajos generavimo įrankius, padaryta išvada, kad reikalingas įrankis galintis generuoti sąsają iš metaduomenų, kadangi minėti įrankiai naudoja duomenis ir funkcijas.
2. Atlikus eksperimentą nustatyta, kad sukurtu automatizuotu programinės įrangos kūrimo įrankiu galima generuoti vartotojo sąsają iš pasirinktos ODRES metodo metabazės
3. Sukurtas įrankis gali būti integruojamas į ODRES CASE įrankį.

8 Literatūra

[1] Butkienė, R. Informacijos sistemai keliamų funkcinių reikalavimų specifavimo metodas. Daktaro disertacija. Kaunas, 2002.

[2] <http://gedmin.as/study/inf98/pkpm/Pagrindiniai%20gyvavimo%20ciklo%20procesai.ppt>

[3] Informacijos sistemų projektavimas Oracle Designer/2000 priemonėmis / Rita Butkienė ; Kauno technologijos universitetas.

[4] <http://www.cs.umd.edu/class/fall2002/cmsc838s/tichi/oai.html>

[5] http://lt.wikipedia.org/wiki/Ge%C5%A1talt_psichologija

[6] <http://docs.hp.com/en/B1171-90148/ch01s02.html>

Priedai

Programos išeities kodas

Pagrindinis modulis

```
Dim ds As String
Dim rs As ADODB.Recordset
Dim items() As Variant
Public frmX, frmY, frmW, frmL As Double
Public lastID As Long
Public kelias As String
Public sarasas As Integer

Sub main()
lastID = 1
sarasas = 0
OPENDB
Saltiniai.Show

End Sub

Sub OPENDB()

Set conn = New ADODB.Connection
With conn
.Provider = "Microsoft.Jet.OLEDB.4.0"
.ConnectionString = "data source=" & kelias & ";"
.Open
End With

Set rs = New ADODB.Recordset
rs.Open "SELECT * FROM Duom_Salt", conn
strVariable1 = rs("Pavadinimas")

Do Until rs.EOF
Saltiniai.Duom_salt.AddItem      rs("Pavadinimas")      'Lentele
uzpildoma duomenu saltiniais
rs.MoveNext
Loop

End Sub

Formų generavimo modulis
Dim esybes, atributai, atrib2 As ADODB.Recordset
Dim dsindex, esybes_count, atributai_count, tmpi As Integer
Dim dspavadinimas As String
Dim langas, el1, el2 As Visio.Shape
Dim langoX, langoY As Integer 'lango parametrai milimetrais
Dim elx, ely As Double
Dim srx, sry As Double
Public filename As String
```

```

Dim pkaire As Double, pdesine As Double, pvirsus As Double,
papacia As Double, tmpkaire As Double, tmpdesine As Double,
tmpvirsus As Double, tmppapacia As Double
Dim es(50) As Variant
Dim el(50, 1500) As Variant

Private Sub CommandButton1_Click()
Saltiniai.Hide

filename = Visio.ActiveDocument.Name

esybes_count = 0

Set conn = New ADODB.Connection
With conn
.Provider = "Microsoft.Jet.OLEDB.4.0"
.ConnectionString = "data source=" & kelias
.Open
End With

Dim formos As ADODB.Recordset

elx = 2.5
ely = 6

Dim i As Integer
For i = 0 To Duom_salt.ListCount

If Duom_salt.Selected(i) Then

        Set rs = New ADODB.Recordset
        rs.Open "SELECT * from Duom_salt where pavadinimas
= '" & Duom_salt.Value & "'", conn
        Do Until rs.EOF
            dsindex = rs("id")
            dspavadinimas = rs("pavadinimas")
            rs.MoveNext
        Loop

        Set formos = New ADODB.Recordset
        formos.Open "SELECT * FROM Forma WHERE (((Forma.ds_id)=" &
dsindex & "));", conn
        If formos.EOF Then

Else

        Dim klausimas As Integer
        klausimas = MsgBox("Šiam duomenų šaltiniui jau egzistuoja
forma, ar norite generuoti iš naujo?", vbYesNoCancel,
"Pranešimas")
        If klausimas = vbYes Then
conn.Execute "DELETE FROM Forma WHERE ds_id=" & dsindex & ";"

Else

```

```
End
End If
```

```
'End If
End If
```

```
'ESYBIU PRIKLAUSANCIU PASIRINKTAM SALTIIUI KIEKIS
Dim cspr As Integer
```

```
Set esybes = New ADODB.Recordset
esybes.Open "SELECT Esybe.id AS Esybe_id, Esybe.kodas,
Esybe.n_vardas, Esybe.paaiskinimas FROM Esybe INNER JOIN
(Duom_salt INNER JOIN DS_esybe ON Duom_salt.id =
DS_esybe.ds_id) ON Esybe.id = DS_esybe.e_id WHERE
(((Duom_salt.id)=" & dsindex & "));", conn
```

```
Do Until esybes.EOF
    esybes_count = esybes_count + 1
    tmpi = 1
    'atributai priklausantys pasirinktai esybei
```

```
        Set atributai = New ADODB.Recordset
        atributai.Open "SELECT DS_atributas.* FROM Esybe,
atributas, ds_atributas WHERE Atributas.e_id = Esybe.id and
rodomumas=true and DS_atributas.egz_sk=1 and
DS_atributas.id=atributas.id and Esybe.id=" &
esybes("Esybe_id"), conn
        Do Until atributai.EOF
            'IDEDAMAS IVEDIMO LAUKAS
```

```
                Set el1 =
Application.ActiveWindow.Page.Drop(Application.Documents.Item(
"UICTRL_M.VSS").Masters.ItemU("Text box (single-line)"), elx,
ely - atributai_count * 0.4)
```

```
        Dim labeltxt As Visio.Characters
```

```
                Set labeltxt =
Application.ActiveWindow.Page.Shapes.ItemFromID(el1.ID).Charac
ters
```

```
        labeltxt.Begin = 0
        labeltxt.End = 30
        labeltxt.Text = ""
```

```
        cspr = custompr(el1, esybes("Esybe_id"))
```

```
        'IDEDAMA ANTRASTE
```

```
        Application.Windows.ItemEx(filename).Activate
```

```
                Set el2 =
Application.ActiveWindow.Page.Drop(Application.Documents.Item(
"PEANNT_M.VSS").Masters.ItemU("10pt. text"), elx - 0.45, ely -
atributai_count * 0.4)
```

```
        cspr = custompr(el2, esybes("Esybe_id"))
```

```

        Application.ActiveWindow.Page.Shapes.ItemFromID(el
2.ID).CellsSRC(visSectionCharacter, 0,
visCharacterSize).FormulaU = "10 pt"
        Application.ActiveWindow.Page.Shapes.ItemFromID(el
2.ID).CellsSRC(visSectionParagraph, 0, visHorzAlign).FormulaU
= "0"

        Set labeltxt =
Application.ActiveWindow.Page.Shapes.ItemFromID(el2.ID).Charac
ters
        labeltxt.Begin = 0
        labeltxt.End = 10
        labeltxt.Text = "" & atributai("vard_org") & ""

        atributai_count = atributai_count + 1

        If ely > sry Then
            sry = ely
        End If

        atributai.MoveNext
    Loop

        sry = sry - atributai_count * 0.4
        atributai_count = 0

        'reikia padaryt jei DAUGIAU NEI DVI ESYBES
        If esybes_count = 1 Then
            elx = elx + 3
            ely = ely + atributai_count * 0.4

        ElseIf esybes_count < 3 And Not esybes_count = 1 Then
            elx = elx + 3
            ely = ely + atributai_count * 0.4

        End If
        'saraso formavimas

        Dim srsplt As Integer
        Dim atrib2tmp As ADODB.Recordset

        Set atrib2 = New ADODB.Recordset
        atrib2.Open "SELECT DS_atributas.* FROM Esybe,
atributas, ds_atributas WHERE Atributas.e_id = Esybe.id and
rodomumas=true and DS_atributas.egz_sk>1 and
DS_atributas.id=atributas.id and Esybe.id=" &
esybes("Esybe_id"), conn
        Set atrib2tmp = atrib2

        srsplt = skaiciuoti_eilutes(atrib2tmp)
        atrib2.Close
        Set atrib2tmp = Nothing

```

```

Dim atr_ctrl(50) As Variant
tmpi = 0
    atrib2.Open "SELECT DS_atributas.* FROM Esybe,
atributas, ds_atributas WHERE Atributas.e_id = Esybe.id and
rodomumas=true and DS_atributas.egz_sk>1 and
DS_atributas.id=atributas.id and Esybe.id=" &
esybes("Esybe_id"), conn
    Do While Not atrib2.EOF
        'elementu kiekio kontrole

        tmpi = atr_ctrl(atrib2("egz_sk"))
        tmpi = tmpi + 1
        atr_ctrl(atrib2("egz_sk")) = tmpi

    atrib2.MoveNext

Loop

Dim tmpii As Integer
For tmpi = 0 To 50
    For tmpii = 0 To 50
        If atr_ctrl(tmpi) And atr_ctrl(tmpii) Then
            If atr_ctrl(tmpii) < atr_ctrl(tmpi) Then
                ' MsgBox atr_ctrl(tmpii)
                Elkontrole.ListBox1.AddItem ("atributo
egzempliorių sk - " & tmpii & " (" & atr_ctrl(tmpii) & ")
stulpeliai")
                Elkontrole.ListBox1.AddItem ("atributo
egzempliorių sk - " & tmpi & " (" & atr_ctrl(tmpi) & ")
stulpeliai")
                Elkontrole.TextBox1.Text = tmpi
                Elkontrole.Show
            End If
        End If
    Next tmpii
Next tmpi
atrib2.Close
    Plotas (0)
    srx = pkaire
    sry = papacia - 0.3

    atrib2.Open "SELECT DS_atributas.* FROM Esybe,
atributas, ds_atributas WHERE Atributas.e_id = Esybe.id and
rodomumas=true and DS_atributas.egz_sk>1 and
DS_atributas.id=atributas.id and Esybe.id=" &
esybes("Esybe_id"), conn

    Do While Not atrib2.EOF
        Dim elplt As Double

```



```

        If srsplt > 0 Then

elplt = 1

End If
        Application.Windows.ItemEx(filename).Activate

                                Set el1 =
Application.ActiveWindow.Page.Drop(Application.Documents.Item(
"UICTRL_M.VSS").Masters.ItemU("Grid column"), srx + elplt /
25.4, sry)
                elplt = ((pdesine - pkaire) * 25.4) / srsplt
                cspr = custompr(el1, esybes("Esybe_id"))

                Dim vsoCharacters4 As Visio.Characters
                                Set vsoCharacters4 =
Application.ActiveWindow.Page.Shapes.ItemFromID(el1.ID).Charac
ters
                vsoCharacters4.Begin = 0
                vsoCharacters4.End = 0
                vsoCharacters4.Text = "" & atrib2("vard_org")
& ""

                intPropRow2 = 0
                If sarasas = 0 Then
                        el1.CellsSRC(visSectionProp, intPropRow2,
visCustPropsValue).FormulaU = "" & atrib2("egz_sk") & ""
                Else
                        el1.CellsSRC(visSectionProp, intPropRow2,
visCustPropsValue).FormulaU = "" & sarasas & ""
                End If
                                el1.CellsSRC(visSectionObject,
visRowXFormOut, visXFormWidth).FormulaU = Int(elplt) & " mm"

                'Next

                srx = srx + elplt / 25.4
                atrib2.MoveNext

                Loop
                Set atrib2 = Nothing

esybes.MoveNext
Loop

papacia = 5
pvirus = 0
pdesine = 5
pkaire = 0

Plotas (0)

'NUSTATYTI LANGO PARAMETRUS

```

```

Set                                langas                                =
Application.ActiveWindow.Page.Drop(Application.Documents.Item(
"UIDLGS_M.VSS").Masters.ItemU("Blank form"), pkaire +
((pdesine - pkaire) / 2) - 0.2, papacia + (pvirsus - papacia)
/ 2 - 0.1)
lastID = langas.ID

Application.ActiveWindow.Page.Shapes.ItemFromID(lastID).CellsS
RC(visSectionObject, visRowXFormOut, visXFormWidth).FormulaU =
Int((1 + pdesine - pkaire) * 25) & " mm"
Application.ActiveWindow.Page.Shapes.ItemFromID(lastID).CellsS
RC(visSectionObject, visRowXFormOut, visXFormHeight).FormulaU
= Int((1.05 + pvirsus - papacia) * 25) & " mm"
ActiveWindow.Select
Application.ActiveWindow.Page.Shapes.ItemFromID(lastID),
visSelect
Application.ActiveWindow.Selection.SendToBack

conn.Execute "INSERT INTO Forma (id, pavadinimas,
ds_id,a,p,x,y) values (" & lastID & ", " &
Left(dspavadinimas, 10) & ", " & dsindex & ", " & Int(pkaire
* 25.4) & ", " & Int(pvirsus * 25.4) & ", " & Int((pdesine -
pkaire) * 25.4) & ", " & Int((pvirsus - papacia) * 25.4) & ")"

End If
Next

srx = pdesine - 0.5
sry = papacia - 0.4

For i = 0 To Duom_salt.ListCount
If Duom_salt.Selected(i) Then

        Dim vsoCharacters5 As Visio.Characters
        Set vsoCharacters5 =
Application.ActiveWindow.Page.Shapes.ItemFromID(lastID).Charac
ters

        vsoCharacters5.Begin = 0
        vsoCharacters5.End = 50
        vsoCharacters5.Text = "" & Duom_salt.Value &
""

Set rs = New ADODB.Recordset
rs.Open "SELECT * from Duom_salt where pavadinimas not like '"
& Duom_salt.Value & "'", conn

Do Until rs.EOF
Application.Windows.ItemEx(filename).Activate
        Set ell =
Application.ActiveWindow.Page.Drop(Application.Documents.Item(
"UICTRL_M.VSS").Masters.ItemU("Command button"), srx, sry)

```

```

        cspr = custompr(ell, -1)
Application.ActiveWindow.Page.Shapes.ItemFromID(ell.ID).CellsS
RC(visSectionObject, visRowXFormOut, visXFormWidth).FormulaU =
"28 mm"
        Dim vsoCharacters1 As Visio.Characters
                Set vsoCharacters1 =
Application.ActiveWindow.Page.Shapes.ItemFromID(ell.ID).Charac
ters
        vsoCharacters1.Begin = 0
        vsoCharacters1.End = 20
        vsoCharacters1.Text = "" & rs("pavadinimas") & ""

        srx = srx - 1.2

dsindex = rs("id")

rs.MoveNext
Loop
End If
Next

conn.Close

Duom_salt.Clear

End
        Dim vsoPage1 As Visio.Page
        Set vsoPage1 = ActiveDocument.Pages.Add
        vsoPage1.Background = False

Saltiniai.Show

End Sub

Private Function skaiciuoti_eilutes(ByVal rsADO As
ADODB.Recordset) As Integer
Dim iCount As Integer

        iCount = 0

        If rsADO.EOF Then
                GetADORecordCount = 0
                Exit Function
        End If

        Do While Not rsADO.EOF
                iCount = iCount + 1
                rsADO.MoveNext
        Loop

```

```
    skaiciuoti_eilutes = iCount
End Function
```

```
Private Function koregotix(ByVal elid As Integer) As Double
Dim elvisx, elpinx As Double
```

```
elvisx =
Application.ActiveWindow.Page.Shapes.ItemFromID(elid).CellsSRC
(visSectionObject, visRowXFormOut, visX).Result(Visio.visNone)
elpinx =
Application.ActiveWindow.Page.Shapes.ItemFromID(elid).CellsSRC
(visSectionObject, visRowXFormOut,
visXFormLocPinX).Result(Visio.visNone)
```

```
If elvisx = elpinx Then
    koregotix = 0
Else
    koregotix = elpinx
    MsgBox elvisx & " - " & elpinx & " = " & koregotix
End If
```

```
End Function
```

```
Private Function koregotiy(ByVal elid As Integer) As Double
Dim elvisx, elpinx As Double
```

```
elvisx =
Application.ActiveWindow.Page.Shapes.ItemFromID(elid).CellsSRC
(visSectionObject, visRowXFormOut, visY).Result(Visio.visNone)
elpinx =
Application.ActiveWindow.Page.Shapes.ItemFromID(elid).CellsSRC
(visSectionObject, visRowXFormOut,
visYFormLocPinY).Result(Visio.visNone)
```

```
If elvisx = elpinx Then
    koregotiy = 0
Else
    koregotiy = elpinx
End If
```

```
End Function
```

```
Private Function Plotas(ByVal esid As Integer)
'plotas kuriuose randasi elementai
Dim sheip As Visio.Selection
Application.ActiveWindow.SelectAll
Set sheip = ActiveWindow.Selection
sheip.BoundingBox visBBoxDrawingCoords + visBBoxUprightWH,
pkairė, papacia, pdesine, pvirsus
End Function
```

```
Elementų atributų surašymo modulis
```

```

Private Function custompr(ByVal vsoshapel As Shape, ByVal
esybe As Integer) As Integer
    Dim x As Double, y As Double, a As Double, p As Double

    Dim intPropRow As Integer
    Dim intPropRow2 As Integer
        x = vsoshapel.CellsSRC(visSectionObject, visRowXFormOut,
visX).Result(Visio.visNone) - koreguotix(vsoshapel.ID)
        y = vsoshapel.CellsSRC(visSectionObject, visRowXFormOut,
visY).Result(Visio.visNone) + koreguotiy(vsoshapel.ID)

        intPropRow = vsoshapel.AddRow(visSectionProp, visRowLast,
visTagDefault)
            vsoshapel.CellsSRC(visSectionProp, intPropRow,
visCustPropsLabel).FormulaU = ""esybe""
            vsoshapel.CellsSRC(visSectionProp, intPropRow,
visCustPropsType).FormulaU = "2"
            vsoshapel.CellsSRC(visSectionProp, intPropRow,
visCustPropsFormat).FormulaU = ""0""
            vsoshapel.CellsSRC(visSectionProp, intPropRow,
visCustPropsLangID).FormulaU = "1033"
            vsoshapel.CellsSRC(visSectionProp, intPropRow,
visCustPropsCalendar).FormulaU = ""
            vsoshapel.CellsSRC(visSectionProp, intPropRow,
visCustPropsPrompt).FormulaU = ""
            vsoshapel.CellsSRC(visSectionProp, intPropRow,
visCustPropsValue).FormulaU = esybe
            vsoshapel.CellsSRC(visSectionProp, intPropRow,
visCustPropsSortKey).FormulaU = ""

            intPropRow2 = vsoshapel.AddRow(visSectionProp,
visRowLast, visTagDefault)
                vsoshapel.CellsSRC(visSectionProp, intPropRow2,
visCustPropsLabel).FormulaU = ""x""
                vsoshapel.CellsSRC(visSectionProp, intPropRow2,
visCustPropsType).FormulaU = "2"
                vsoshapel.CellsSRC(visSectionProp, intPropRow2,
visCustPropsFormat).FormulaU = ""0""
                vsoshapel.CellsSRC(visSectionProp, intPropRow2,
visCustPropsLangID).FormulaU = "1033"
                vsoshapel.CellsSRC(visSectionProp, intPropRow2,
visCustPropsCalendar).FormulaU = ""
                vsoshapel.CellsSRC(visSectionProp, intPropRow2,
visCustPropsPrompt).FormulaU = ""
                vsoshapel.CellsSRC(visSectionProp, intPropRow2,
visCustPropsValue).FormulaU = Str(x * 24.5)
                vsoshapel.CellsSRC(visSectionProp, intPropRow2,
visCustPropsSortKey).FormulaU = ""

                intPropRow3 = vsoshapel.AddRow(visSectionProp,
visRowLast, visTagDefault)
                    vsoshapel.CellsSRC(visSectionProp, intPropRow3,
visCustPropsLabel).FormulaU = ""y""

```

```

        vsoshapel.CellsSRC(visSectionProp, intPropRow3,
visCustPropsType).FormulaU = "2"
        vsoshapel.CellsSRC(visSectionProp, intPropRow3,
visCustPropsFormat).FormulaU = ""0""
        vsoshapel.CellsSRC(visSectionProp, intPropRow3,
visCustPropsLangID).FormulaU = "1033"
        vsoshapel.CellsSRC(visSectionProp, intPropRow3,
visCustPropsCalendar).FormulaU = ""
        vsoshapel.CellsSRC(visSectionProp, intPropRow3,
visCustPropsPrompt).FormulaU = ""
        vsoshapel.CellsSRC(visSectionProp, intPropRow3,
visCustPropsValue).FormulaU = Str(y * 24.5)
        vsoshapel.CellsSRC(visSectionProp, intPropRow3,
visCustPropsSortKey).FormulaU = ""

```

End Function

```

Public Function issaugoti()
Dim conObj As Visio.Connect
Dim i As Integer
Dim ReportingType As String
Dim vsoPage As Visio.Page
Dim VsoShp As Visio.Shape
Dim propval As String
Dim sql As String

```

```

Dim esybe As Integer, x, y, a, p As Integer

```

```

For Each vsoPage In ActiveDocument.Pages
    For Each VsoShp In vsoPage.Shapes
        If Not VsoShp.OneD Then

```

```

            nrows = VsoShp.RowCount(Visio.visSectionProp)

```

```

            For i = 0 To nrows - 1

```

```

                propval =
VsoShp.CellsSRC(Visio.visSectionProp, i,
visCustPropsLabel).ResultStr(Visio.visNone)

```

```

                Select Case propval
                Case "esybe"

```

```

                    esybe =
VsoShp.CellsSRC(Visio.visSectionProp, i,
visCustPropsValue).ResultStr(Visio.visNone)

```

```

                    Case "x"
                        x = VsoShp.CellsSRC(Visio.visSectionProp,
i, visCustPropsValue).ResultStr(Visio.visNone)
                    Case "y"
                        y = VsoShp.CellsSRC(Visio.visSectionProp,
i, visCustPropsValue).ResultStr(Visio.visNone)
                    Case "a"

```

```

                a = VsoShp.CellsSRC(Visio.visSectionProp,
i, visCustPropsValue).ResultStr(Visio.visNone)
                Case "p"
                    p = VsoShp.CellsSRC(Visio.visSectionProp,
i, visCustPropsValue).ResultStr(Visio.visNone)
                End Select

            Next i
        End If
        sql = "insert into f_elementas(id, x,y,a,p) values ("
& VsoShp.ID & ", " & x & " , " & y & ", " & a & ", " & p & ")"
        conn.Execute sql
    Next VsoShp

Next vsoPage

End Function

```